universität
wien

DISSERTATION

Titel der Dissertation

Improving the characterisation of fractured hydrocarbon reservoir dolomites (Vienna Basin, Austria) by 3D and 2D micro-imaging and permeability measurements

Verfasser

Maarten Hidde Voorn, MSc.

angestrebter akademischer Grad

Doktor der Naturwissenschaften (Dr.rer.nat.)

Wien, 2014

Studienkennzahl lt. Studienblatt:	A 791 426
Dissertationsgebiet lt. Studienblatt:	Erdwissenschaften
Betreuerin / Betreuer:	Univ. Prof. Mag. Dr. Bernhard Grasemann Dr. Ulrike Exner

Table of contents

Acknowledgements 5

Abstract 7

Zusammenfassung 8

A. Introduction 9

B. Geological background and sampling 11
  B.1. Geology and tectonics 11
  B.2. Dolomite fault rock classifications 15

C. X-ray Micro-Computed Tomography 19
  C.1. µCT data acquisition 19
    C.1.1. Background and theory 19
    C.1.2. µCT on Hauptdolomit samples: possibilities and limitations 23
  C.2. µCT data processing: an introduction 28
    C.2.1. Effects in the data 28
    C.2.2. Extracting petrophysical properties from datasets 29

D. µCT data processing: Multiscale Hessian fracture filter (MSHFF) 31
  D.1. Introduction 32
  D.2. Methods and results 34
    D.2.1 Introduction to feature detection using the Hessian matrix 34
    D.2.2 Implementation 37
    D.2.3 Multiscale Hessian fracture filtering 38
    D.2.4 Calibration, connectivity, and segmentation 40
  D.3. Discussion 43
    D.3.1 Advantages and drawbacks 43
    D.3.2 Other considerations and limitations 46
    D.3.3 Orientation information from the Hessian matrix 46
  D.4. Conclusions 47

E. Additional µCT data processing and analysis 49
  E.1. Addendum and improvements to the MSHFF 49
    E.1.1. Narrow feature enhancement and thresholding 49
    E.1.2. Island removal 51
    E.1.3. Summary of segmentation sequence 52
  E.2. Additional MSHFF performance tests 52
  E.3. µCT data analysis 53

F. Dolomite fault rock analysis 63
  F.1. Introduction 64
  F.2. Geological background and samples 66
  F.3. 3D and 2D imaging methods 70
    F.3.1. 3D X-Ray micro-Computed Tomography (µCT) 70
    F.3.2. Thin sections 73
  F.4. Permeability experiments 75
    F.4.1. Gas permeability 75
    F.4.2. Water permeability 76

F.5. Results and discussion 79
  F.5.1. Porosity 79
  F.5.2. Aperture 81
  F.5.3. Fracture density 83
  F.5.4. Fracture orientation 83
  F.5.5. Potential other parameters from imaging data 83
  F.5.6. Permeability experiments 84
  F.5.7. Permeability models 88
  F.5.8. Synthesis and recommendations 89
F.6. Conclusion 90

G. Additional thin section techniques 93
  G.1. Thin section overviews, image processing and porosity analysis 93
  G.2. Cathodoluminescence (CL) microscopy 96
  G.3. Focussed Ion Beam – Scanning Electron Microscopy (FIB-SEM) tomography 99
    G.3.1. Introduction and data acquisition 99
    G.3.2. FIB-SEM Tomography image processing 103
    G.3.3. FIB-SEM Tomography image analysis 103
    G.3.4. FIB-SEM Tomography results 104
    G.3.5. FIB-SEM conclusions 109

H. Permeability 111

I. Summary and conclusions 115

References 117

Appendix X1: MSHFF code user guide 125
  X1.1. Included data 126
  X1.2. Preparation checklist before running any code 127
  X1.3. Determining analysis parameters 128
  X1.4. Selecting ROI and control lines settings 136
  X1.5. Running the multiscale Hessian fracture filter 139
  X1.6. Calibration 141
  X1.7. Connectivity filtering (in MATLAB®) 147
  X1.8. Binarisation 147

Appendix X2: MSHFF code 149
  X2.1. MSHFF Preparation (FIJI Macro) 149
  X2.2. MSHFF Main code (FIJI Macro) 152
  X2.3. MSHFF Calibration (FIJI Macro) 163
  X2.4. Connectivity filtering (MATLAB® script) 167

Appendix X3: Curriculum Vitae 169

Acknowledgements

I wish to thank everyone who helped making it possible to work on and finish this thesis. This especially involves my supervisors Ulrike Exner and Bernhard Grasemann, as well as my paper co-authors Auke Barnhoorn, Patrick Baud and Thierry Reuschlé. Without them, most of the research and writing would not have been possible. I also wish to thank Alex Rath and Stefan Hoyer for the co-operations and nice discussions within the project.

Many people have helped in “technical support”, but from the people not mentioned yet, especially Christian Gusenbauer (FH Wels, µCT), Gerlinde Habler (University of Vienna, FIB-SEM) and Jan Etienne (Delft University, permeability experiments) have contributed greatly in obtaining the data presented in this study. I would also like to thank all other colleagues and friends at the University of Vienna for great scientific discussions, and moreover crazy lunches, barbecues and “extracurricular activities”!

Furthermore, I would like to thank all friends and family in the Netherlands, Austria and elsewhere for their support. Special thanks and love goes out to my girlfriend Resi Veeningen. Without her, the great “Viennese experience” would very likely never have happened, which has shown us even more that we make a great couple!

Finally (hopefully I have not forgotten anyone accidentally), I would like to thank the Austrian oil company OMV for the main funding of this project, and particularly Philipp Strauss as the main contact within OMV. Without their funding, ideas and samples this study could not have been conducted. Part of the permeability experiments have been funded over the ÖAD WTZ (Amadée programme) project FR 05/2014.

Abstract

In hydrocarbon reservoirs, naturally fractured rocks can occur and influence the porosity and permeability characteristics significantly. When focussing on small samples (i.e. cores and plugs) from such reservoirs, there are several drawbacks on methods commonly applied (such as laboratory porosity and permeability tests, and thin section analysis). For example, methods may not provide enough detail, only 2D information and/or only bulk information. In an effort to overcome these drawbacks, this thesis investigates the use of 3D X-ray micro-Computed Tomography (µCT) on plug samples of fractured dolomite from the pre-Neogene basement of the Vienna Basin (Austria). µCT is a non-destructive technique, and allows imaging the internal structure of the rock samples in 3D. This image data is processed using a Hessian based filtering technique to extract the fracture network from the data, and from this parameters such as porosity, aperture and fracture orientation in the sample can be determined.

The 3D µCT data were complemented and benchmarked with a range of other analytical techniques. Thin sections imaged in 2D by Scanning Electron Microscopy (SEM) with Back-Scattered Electrons (BSE) contrast provide a more detailed view of the rocks' microstructure and mineral content. Furthermore, serial slicing by Focussed Ion Beam - Scanning Electron Microscopy (FIB-SEM) gives a very detailed view in 3D, and can be useful in assessing microporosity. Finally, permeability measurements under increasing confining pressure provide a better link to the possible behaviour of the rocks in more realistic reservoir in-situ conditions.

In this thesis, the possibilities and limitations of the above techniques are shown, as well as the most efficient ways of processing and combining the different sources of information. The results and techniques can therefore be readily applied and standardised in more extensive studies on fractured reservoir rocks.

Zusammenfassung

In Kohlenwasserstoff-Lagerstätten können Gesteine auftreten, die natürliche Klüfte aufweisen, welche die Porositäts- und Permeabilitätseigenschaften deutlich beeinflussen. Bei der Untersuchung kleiner Proben (d.h. Bohrkerne und Plugs) aus solchen Lagerstätten existieren allerdings mehrere Nachteile bei der Anwendung konventioneller Methoden (z.B. Labormessungen von Porosität und Permeabilität, Dünnschliffanalysen). Unter anderem können diese Methoden z.B. nicht ausreichend detailliert sein oder nur zweidimensionale oder gemittelte Informationen liefern. Mit dem Ziel diese Nachteile zu umgehen, prüft die vorliegende Arbeit die Anwendbarkeit von 3D Röntgen-Microcomputertomographie (µCT) an kleinen Bohrkernen (Plugs) aus Proben eines zerklüfteten Dolomitgesteins, das aus dem prä-neogenen Untergrund des Wiener Beckens stammt. µCT ist eine nicht-destruktive Methode und erlaubt die Darstellung der internen Struktur von Gesteinsproben in 3D. Die gewonnenen Daten werden mittels einer eigens entwickelten Technik zur Filterung (Hessian Filtering) prozessiert, um das Kluftnetzwerk zu extrahieren und daraus Parameter wie Porosität, Kluftweite oder Orientierung der Klüfte zu bestimmen.

Weiters werden die 3D µCT Daten durch eine Reihe anderer analytischer Methoden ergänzt und kalibriert. So geben Gesteinsdünnschliffe in 2D im Rasterelektronenmikroskop (SEM) mittels rückgestrahlter Elektronen (BSE) einen detaillierteren Einblick in Mikrostruktur und Mineralbestandteile der Proben. Außerdem wurden durch Ionenstrahl-Rasterelektronen-mikroskopie (FIB-SEM) serielle Schnitte im Mikrometer-Bereich erzeugt, die ein noch detaillierteres Bild bieten und zur Bestimmung der Mikroporosität herangezogen werden. Darüber hinaus wurden Laborexperimente zur Messung der Permeabilität unter steigendem Umgebungsdruck durchgeführt, die eine Annäherung an das Verhalten der Gesteine unter realistischen Bedingungen in der Lagerstätte selbst bieten können.

In dieser Dissertation werden sowohl Möglichkeiten und Grenzen der oben genannten Methoden, als auch die effizientesten Abläufe zur Bearbeitung und Kombination der verschiedenen Informationsquellen aufgezeigt. Die Ergebnisse und Techniken sollen folglich auch in umfassenderen Studien an geklüfteten Lagerstättengesteinen angewandt und standardisiert werden.

A. Introduction

With an increasing worldwide need for hydrocarbons (oil and gas) there is an increasing need for discovering new reservoirs, as well as a need for producing more efficiently from existing or unconventional reservoirs. Finding and applying new and more detailed methods of analysing rocks from such reservoirs is part of these developments. The research presented in this thesis is focussed on a case study of reservoirs consisting of fractured dolomite basement rocks in the Vienna Basin in Austria (their geological background will be explained in more detail in Chapter B). Exploration and production from these reservoirs has been ongoing since the 1960s with varying success (Zimmer & Wessely, 1996), after already 30 years of exploration and production from siliciclastic reservoirs in overlying Neogene units. Because the dolomite rocks from the dolomite reservoirs range from being relatively solid or cemented to being densely fractured – sometimes within the metre-scale – analyses on the field and reservoir scale (e.g. by seismics) or on outcrop analogues are not always successful in explaining differences in production. Also borehole information such as by Formation Micro Imaging (FMI) in the newer boreholes does not always provide enough detail for analysis. For these reasons, cores have always been an important source of information for the reservoirs in this area. However, also common analyses on these cores have drawbacks.

First, hand specimen analysis along the core gives a good overview, but is limited in detail, since one can only observe the outside of the core in limited resolution. Second, common laboratory analyses on core and plug samples provide useful bulk values of for example porosity and permeability, but usually limited information about the internal structure of the rocks. Techniques that do provide more information about the structure, such as Mercury Intrusion Porosimetry (MIP), can be very useful, but limit the use of the samples afterwards. Third, thin section analysis on core samples is often carried out and provides a great amount of detail, with a large range of techniques (from optical microscopy to Scanning Electron Microscopy (SEM) analysis) and analyses (e.g. contents analysis and detailed porosity analysis) being available. Despite being very useful, most of these techniques are inherently 2D, and partially destructive to the samples as the thin sections need to be created. Fifth and finally, for fractured samples various other techniques can be applied such as detailed analysis of fracture surfaces by for example surface profiling, or the generation of theoretical models of fracture networks. The detail and applicability of these methods can however be problematic for analysing complex natural fracture networks on the core and plug scale.

The main limitations of the research usually carried out on fractured reservoir samples are thus that they may (1) give only low levels of detail, (2) are not able to provide details on the internal structure of samples, (3) provide only two-dimensional information, (4) are (partially) destructive on the samples, and (5) are not applicable or representative for complex natural fracture networks. The ideal solution would be to have one single method that does not suffer from any of the above drawbacks, or at least provides a good trade-off between them. The best possible technique for this might be X-ray Micro-Computed Tomography (µCT), as it (1) provides good levels of detail, depending on the samples size, (2) images the internal structure of samples, (3) does this in 3D, (4) is non-destructive and (5) is able to image a full fracture network in a samples (but related to the resolution), and hence does not only provide an approximation. For these reasons, this thesis will focus on whether and how µCT can be efficiently used for the analysis of fractured dolomite samples, or fractured samples in general. The details of the technology and several important advantages and drawbacks will be discussed in Chapter C. Chapter D and E will focus on the technique termed Multiscale Hessian fracture filter (MSHFF) developed for the efficient processing (segmentation) of the data obtained by µCT scanning. In the paper in Chapter F the main analytical techniques and results (on for example porosity, aperture, fracture orientation and fracture density data) for the samples covered in this study will be shown. Chapter F can therefore be seen as the main and summarising chapter in this thesis.

Next to rock characteristics mentioned before, an important parameter for hydrocarbon production is the permeability of samples. Because this cannot be directly derived from µCT data, computer models on the data or different tests on the samples are required. To also obtain a better link to the behaviour of the studied rocks at depth in the reservoir, we chose to apply permeability experiments under confining pressure, which will be discussed in detail in Chapter F. Chapter H will additionally discuss numerical modelling on the obtained µCT data briefly.

Beside the large focus on µCT and permeability experiments, thin section analysis is also applied in this thesis. Such thin section analyses are very informative and can – for the rocks in this study – often not be reached by µCT imaging in terms of detail. Thin sections thus remain an important source of information in this thesis, and these results are covered in Chapters F and G. In Chapter G, also Focussed Ion Beam Scanning Electron Microscopy (FIB-SEM) is shortly discussed, as this can provide 3D information at a much higher detail than µCT. However, because of this high detail, its applicability on the fracture networks shown here is limited.

With information coming from very different techniques (µCT, laboratory flow tests, and thin section analyses), the combination and comparison of results is required to show the power – but also the drawbacks – of these techniques. The main combinations and comparisons are shown in the paper in Chapter F, and eventually summarised in Chapter I. The results on the limited amount of samples here show that although µCT is certainly not a replacement for the more common research techniques applied on core samples, it may provide a very good link between different scales and types of analyses, especially when complex fracture networks on the hand specimen scale are involved.

B. Geological background and sampling

B.1. Geology and tectonics

A short overview of the geology and tectonic setting is given in this section. For more detailed descriptions – including the broader geological setting and details on different reservoirs in the area – the reader is referred to published literature (e.g. Arzmüller et al., 2006; Wessely, 2006a; Wessely, 2006b; Zimmer & Wessely, 1996). Background and descriptions focussing on Hauptdolomit – both in the Vienna basin as outside of it – can be found e.g. in Fruth & Scherreiks (1984) and Masaryk & Lintnerová (1997).

The rocks analysed in this study all origin from the Norian (Upper Triassic) Hauptdolomit formation, and have been drilled from the pre-Neogene Vienna Basin subsurface. Here, the Hauptdolomit rocks are present in different tectonic nappes of the Northern Calcareous Alps (e.g. Wessely 2006a); the latter itself being a component of the larger allochtonous Alpine-Carpathian thrust complex (e.g. Arzmüller et al., 2006). Figure B.1.1 shows an overview of the Vienna Basin within the Alpine-Carpathian thrust belt. An overview of the stages of Vienna Basin development is shown in Figure B.1.2. An overview of the different nappes of the Calcareous Alps is shown in Figure B.1.3.

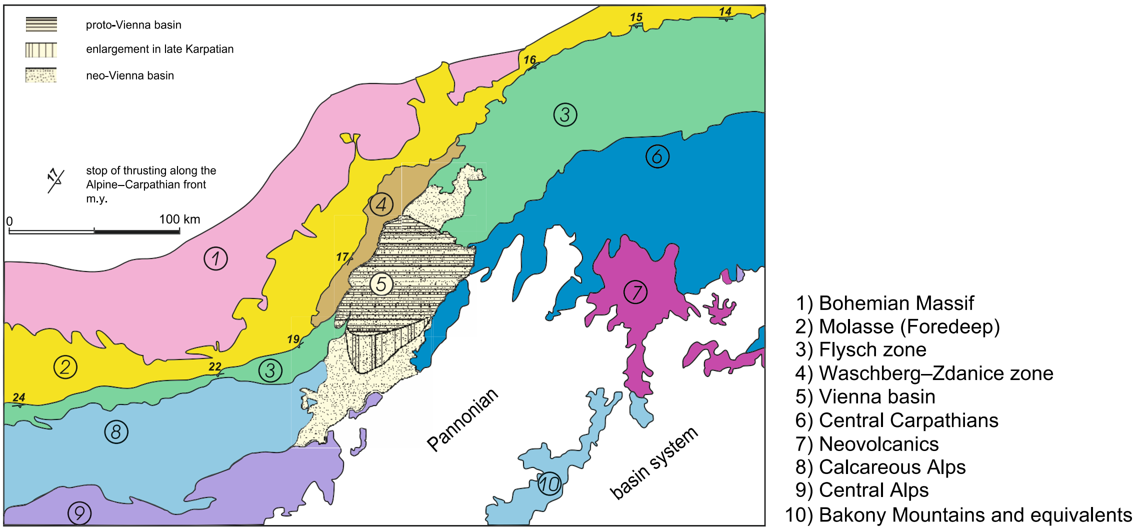


Figure B.1.1. The Vienna Basin in the Alpine-Carpathian thrust belt (From: Arzmüller et al., 2006).

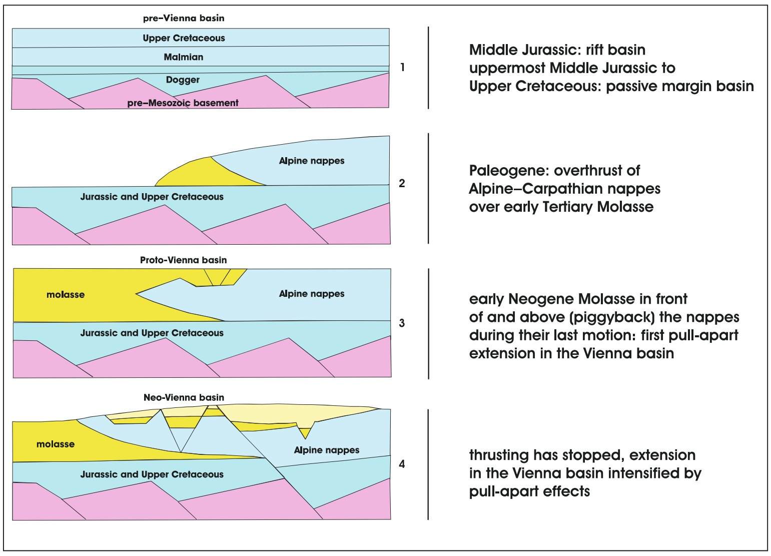

pre–Vienna basin
Upper Cretaceous
Malmian
Dogger
pre–Mesozoic basement
1
Middle Jurassic: rift basin
uppermost Middle Jurassic to
Upper Cretaceous: passive margin basin
Alpine nappes
Jurassic and Upper Cretaceous
2
Paleogene: overthrust of
Alpine–Carpathian nappes
over early Tertiary Molasse
Proto-Vienna basin
molasse
Alpine nappes
Jurassic and Upper Cretaceous
3
early Neogene Molasse in front
of and above (piggyback) the nappes
during their last motion: first pull-apart
extension in the Vienna basin
Neo-Vienna basin
molasse
Alpine nappes
Jurassic and Upper Cretaceous
4
thrusting has stopped, extension
in the Vienna basin intensified by
pull-apart effects

Figure B.1.2. Stages of basin development in the Vienna Basin (From: Arzmüller et al., 2006).

Juvavicum	
Tirolicum	Göller nappe (+J)
	Ötscher nappe
	Reisalpen nappe
Bajuvaricum	Sulzbach nappe
	Lunz nappe
	Frankenfels nappe

Figure B.1.3. Schematic overview of tectonic blocks in the Calcareous Alps. The Juvavicum, Tirolicum and Bajuvaricum are the major units, of which some can be subdivided. The "+J" for the Göller nappe indicates some parts assigned to the Juvavicum have been tectonically emplaced inside the Göller nappe. (Created after: Wessely 2006a.)

Figure B.1.4 shows a geological sketch map of the pre-Neogene rocks under the Vienna Basin, showing some important oil and gas fields in the Frankenfels-Lunz (Bajuvaricum) and Tirolicum nappes. The profile lines indicated on this map are shown in Figure B.1.5, displaying the complex folding and deformation patterns in the nappes, and the main presence of oil and gas (apart from the Neogene basin infill) in rocks of the Hauptdolomit formation.

Deformation of the different thrust sheets below the Vienna Basin, in a complex polyphase brittle manner, has been investigated in detail by Peresson & Decker (1997), with the most important deformation phases summarised as:

a) *Heteroaxial NW- and N-directed fold-and-thrust shortening during thrusting of the Austroalpine allochthon over the underlying Flysch units and the European margin (Cretaceous to Miocene).*
b) *Lateral extrusion of the allochthon along ENE-striking wrench faults (Oligocene to Miocene).*
c) *Sinistral strike-slip faulting and formation of the Miocene Vienna pull-apart basin on top of the Alpine-Carpathian thrust sheets.*
d) *NE-SW-directed extension in the Pliocene and Lower Quaternary.*

These phases of deformation have led to fracture networks with various mutually cross-cutting faulting phases, as well as overprinting microstructures. This complex pattern of faults and microstructures is what is studied in this research project by the use of various analysis techniques, with a focus on 3D imaging of rock samples. The classification of these structures in dolomites is discussed in more detail in the following section.

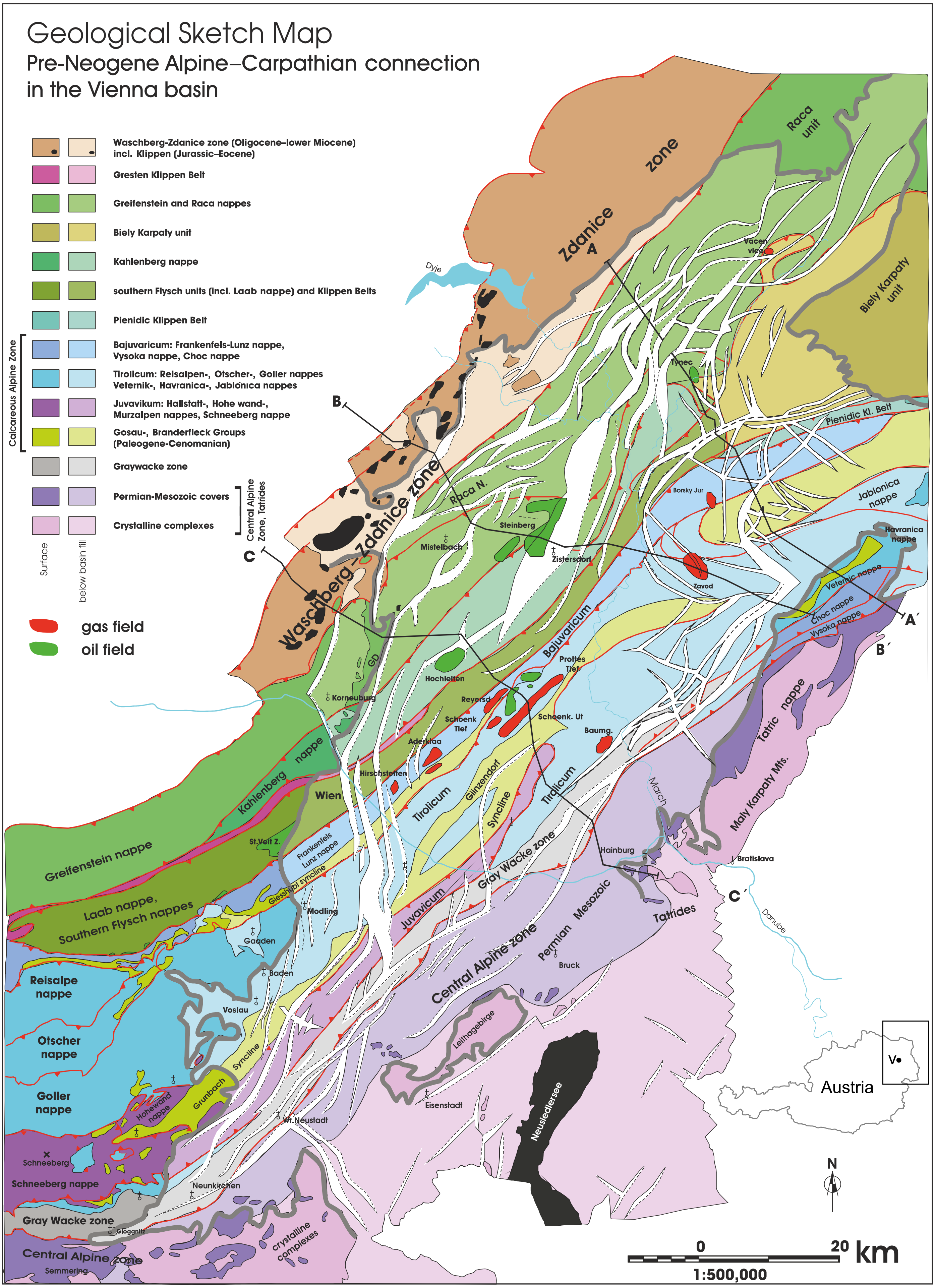


Figure B.1.4. Geological sketch map showing the Alpine-Carpathian connection under the Neogene fill of the Vienna Basin. Small inset shows location of the map relative to Austria (V=Vienna). (After: Arzmüller et al., 2006)

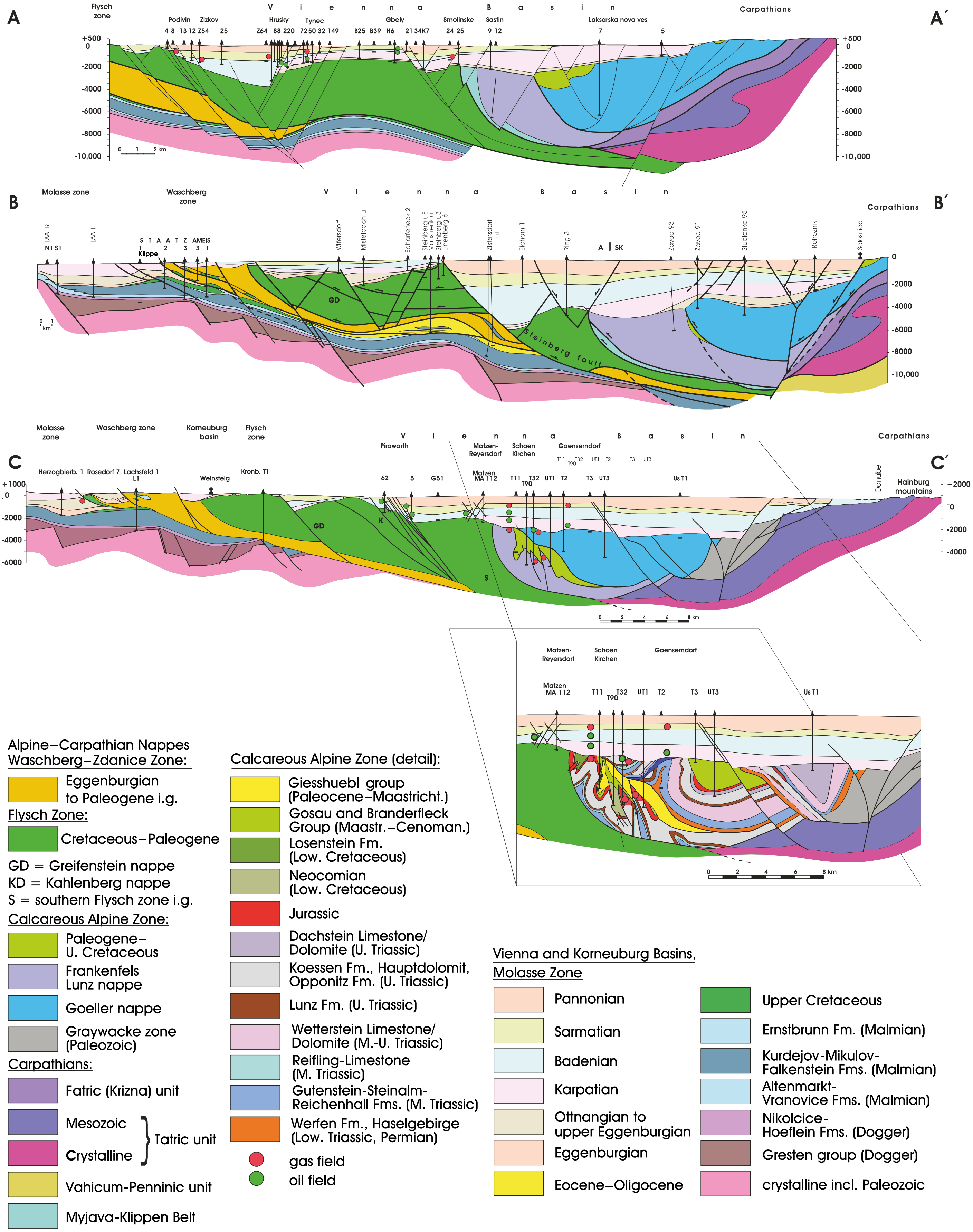


Figure B.1.5. Geological cross sections through the Vienna Basin and its pre-Neogene floor. Profile locations: see Figure B.1.4. (From: Arzmüller et al., 2006)

B.2. Dolomite fault rock classifications

All samples in this study are generally regarded as “fractured dolomite” or “narrowly fractured dolomite”. However, the fracturing in the rocks is clearly not homogenous. For the description of the samples in more detail, we require a terminology that deals with this inhomogeneity effectively. It is difficult to set up such a terminology ourselves using core data only, as one cannot always appreciate the larger structures from the core. Creating a terminology is even less feasible using the data from smaller samples, such as the plugs used for µCT scanning in this study. Because of these limitations, we decided to adopt a terminology based on outcrop studies of fractured carbonates and dolomites, which can be applied on our small samples.

The terminology we apply is taken from previously published literature (e.g. Billi et al., 2003; Billi, 2010), in an adapted form used in internal reports of the OMV and by Bauer (2010) and Schröckenfuchs (2012), with a clear focus on dolomites. The general structure of a fault in dolomite is shown in Figure B.2.1, based on field studies of outcrops of Hauptdolomit and other dolomite formations of the Northern Calcareous Alps. In the course of this project, some fieldwork to reservoir analogues next to the Vienna basin has been carried out as well, but no detailed results could be obtained during these (hence are not included in this thesis). However, even these limited observations do support the following descriptions.

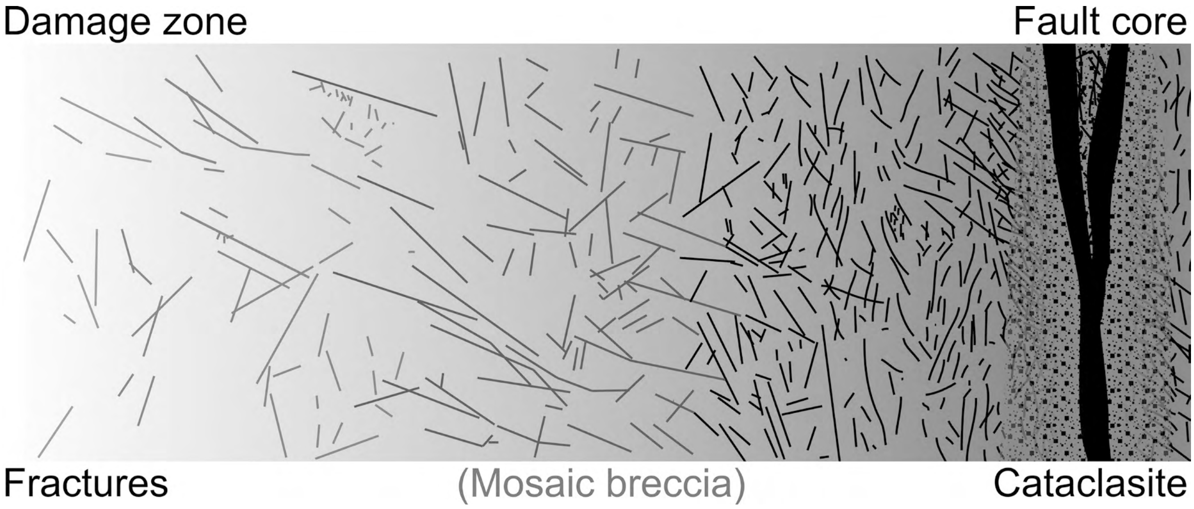


Figure B.2.1. Sketch of the structure of a fault zone, divided in fault core and damage zone, in dolomites from the Northern Calcareous Alps. The width covered in this sketch is in the 10s of metres, but note these structures can exist on different scales. The dolomites observed in the outcrop studies are generally characterised by a single fault core, containing cataclasites of various degrees. The fault core is surrounded by a damage zone, displaying a decreasing degree of deformation (lower fracture density) further away from the fault core. In some cases, cemented mosaic or dilation breccias are observed near the fault core, often acting as a transitional structure between the fault core cataclasites and the fractured damage zone rocks. After Bauer (2010) and Schröckenfuchs (2012).

In Figure B.2.1, a clear division is made between rocks of the fault core and rocks of the damage zone. Rocks of the damage zone are characterised by an increasing fracture density when located closer to the fault core. This increasing fracture density can be divided into several classes or fracture facies, as shown in Figure B.2.2. In the current study, fracture facies 4, characterised by “extremely closely (<1 cm) jointed and micro-fractured dolomite with randomly oriented joints” (OMV internal report), is the most important, as this is the only type for which it is possible to distinguish it definitively in plug-sized samples. Such samples are therefore also simply referred to as “fractured” later in this thesis.

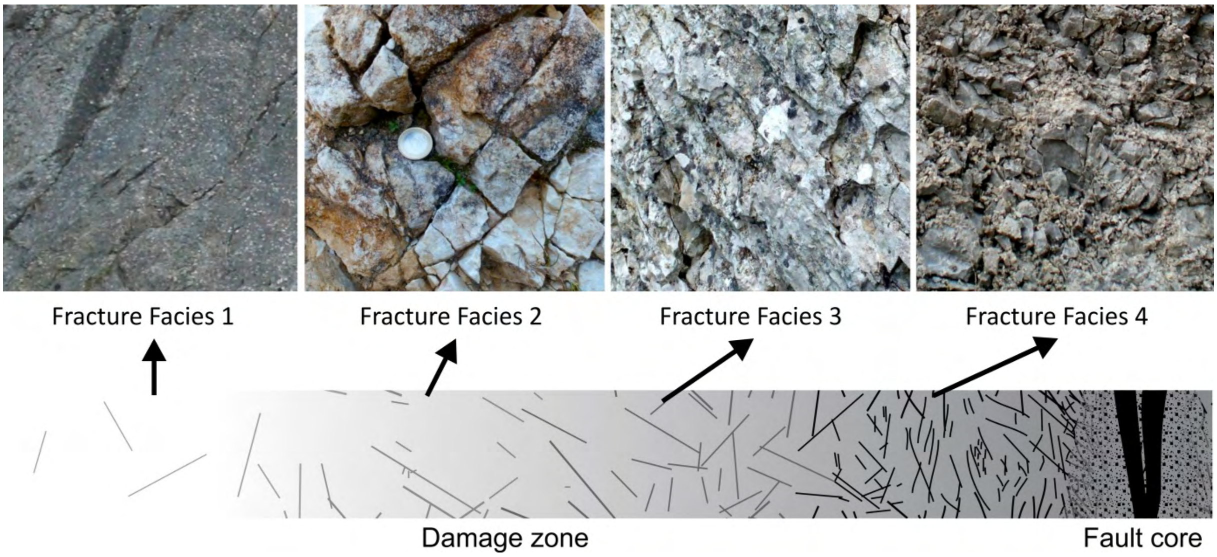


Figure B.2.2. Fracture facies types in the damage zone. The images show examples in outcrops, all taken at a similar scale (see coin in second figure). With every consecutive fracture facies, the fracture density increases. Additionally, the orientation of fractures gets more random. Fracture facies 4 is the most important in this study, as it represents the only facies that can be identified in plug-sized samples. Photographs courtesy of K. Decker., H. Bauer and T. Schröckenfuchs. After Bauer (2010) and Schröckenfuchs (2012).

The fault core rocks are divided into several types. The main distinction is made between breccias (0-10% matrix) and cataclasites (10-100% matrix). The distinction between matrix and grains is made at around 4 µm (Flügel 2004, p. 73). The cataclasites are subdivided into three categories (Billi 2010; Bauer, 2010; Schröckenfuchs, 2012):

- Embryonic cataclasite: A fine-grained matrix with angular and slightly rounded host rock components (1 mm to 1-2 cm in size), in poor sorting. The amount of matrix is <50%. Sometimes the different fragments still show a puzzle-like fit.
- Intermediate cataclasite: Smaller fragments (<1 cm) than in an embryonic cataclasite, with more rounding and better sorting. The relative amount of matrix increases to > 50%.
- Mature cataclasite: Small (mostly < 2 mm), very well sorted and often well rounded fragments in a matrix. The relative matrix constituent is higher (>80%) than in embryonic and intermediate cataclasites.

Besides cataclasites, breccias are found in field studies close to the fault cores. Mosaic or dilation breccias are the most important, and characterised by broken host rock displaying a puzzle-like fit between many fragments (mode I opening). In the reported outcrop studies, mosaic breccias are in many cases cemented by calcite, but this is not often observed in the samples in current study. The term mosaic breccia is used here instead of dilation breccia most of the time, as in field studies, dilation breccias often displayed preferred orientations (Woodcock et al. 2006, Tarasewicz et al., 2005). The different fault core rock types discussed here are shown in Figure B.2.3.

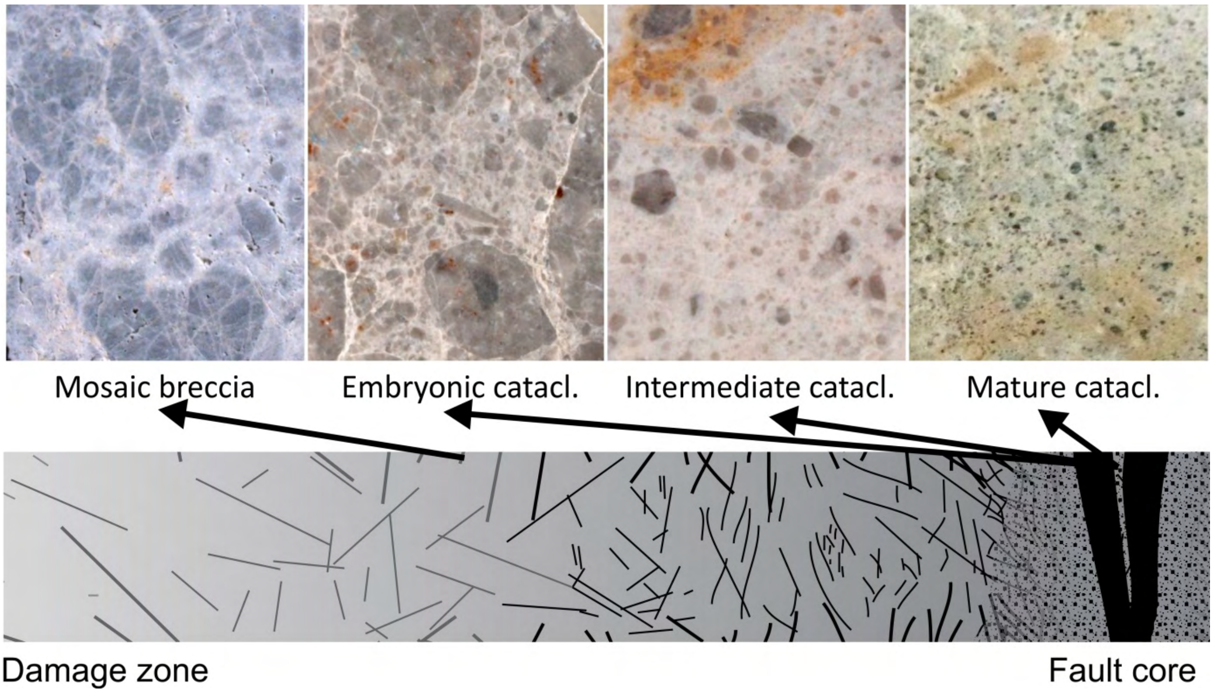
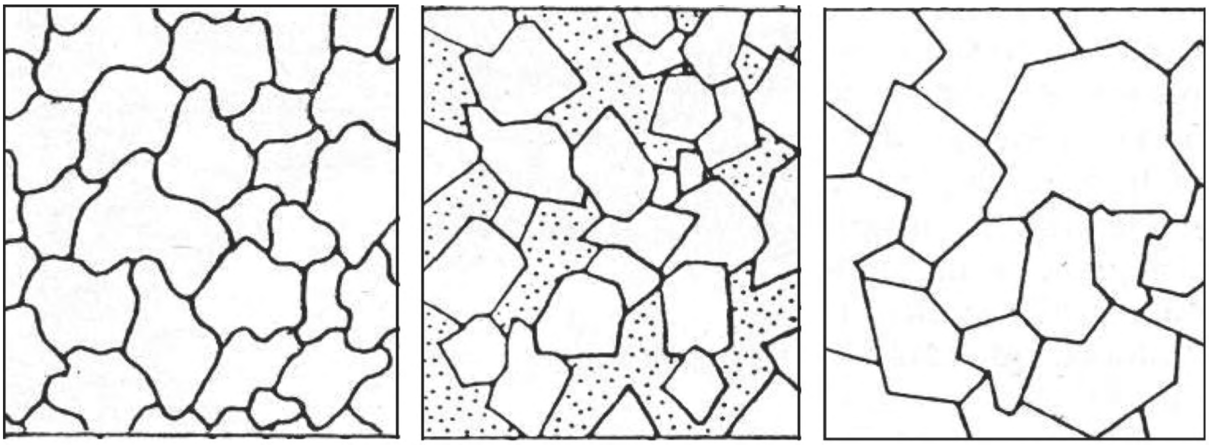


Figure B.2.3. Different rock types associated with the fault core of a fault zone. Closest to the fault core, cataclasites of increasing degree of deformation (grain size, grain rounding, sorting) are present. Other fault rocks associated with the fault zone are mosaic or dilation breccias, exhibiting a puzzle-like fit between host rock fragments. Photographs courtesy of K. Decker, H. Bauer and T. Schröckenfuchs. After Bauer (2010) and Schröckenfuchs (2012).

The dolomite fault rock descriptions shown here are usually sufficient for describing the observed rocks in this study, as we mainly focus on fracture patterns. These descriptions are suitable for having at least an idea of where in a fault zone the studied rocks might come from. One should however keep the various scales of fault zones and degrees of intensity of faulting in mind, meaning that different reservoirs might have very different sizes of fault cores characterised by cataclasis.

If any descriptions are required regarding the dolomite host rock fabric and porosity itself, classifications and terminology according to Flügel (2004) and Sibley & Gregg (1987) are used here. The most important terminology is shown in Figure B.2.4 and Figure B.2.5.



Nonplanar: closely packed anhedral crystals with mostly curved, lobate, serrated, or otherwise irregular intercrystalline boundaries. Preserved crystal-face junctions are rare and crystals often have undulatory extinction under crossed nicols.

Planar-e (euhedral): most dolomite crystals are euhedral rhombs; crystal-supported with intercrystalline area filled by another mineral or porous (as in sucrosic texture).

Planar-s (subhedral): most dolomite crystals are subhedral to anhedral with straight, compromise boundaries and many crystal-face junctions. Low porosity and/or low intercrystalline matrix.

Figure B.2.4. Classification of dolomite fabrics. From Flügel (2004), after Sibley and Gregg (1987).

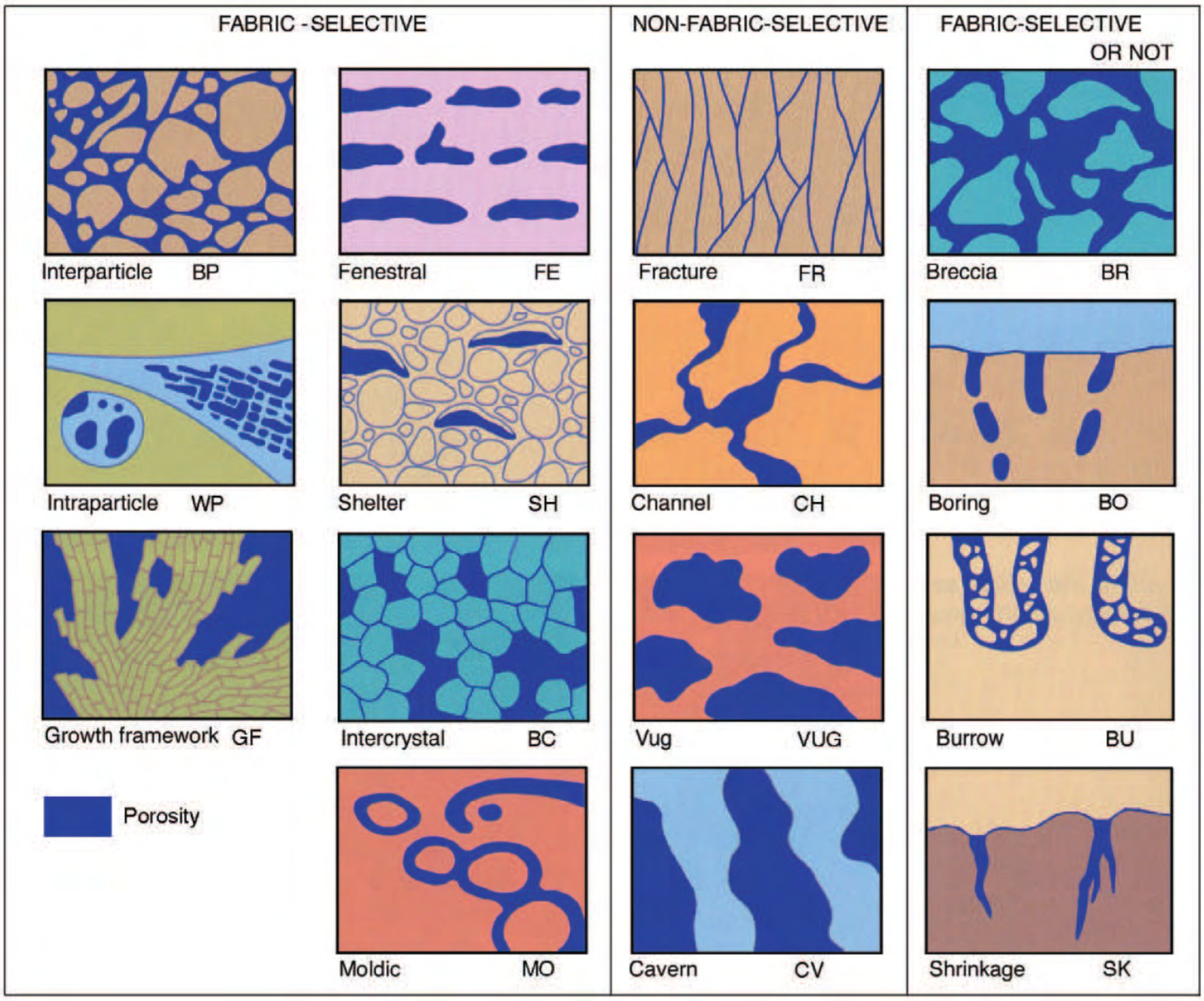


Figure B.2.5. Pore types. For more details, see the original reference. From Flügel (2004), after Choquette and Pray (1970).

C. X-Ray Micro-Computed Tomography (µCT)

C.1. µCT data acquisition

C.1.1. Background and theory

Computed Tomography (CT) is a technique to non-destructively image the internals of objects. It uses a series of two-dimensional X-ray images taken from different viewpoints around the sample, which is then computed (reconstructed) into a three-dimensional volume. Medical CT scanners were originally developed in the 1970s, and were the main purpose of the technique. They have a limited spatial resolution, because the application on human bodies requires a limited radiation dose combined with large investigated volumes. X-ray Micro-Computed Tomography (µCT) has been developed for industrial purposes, and is less limited in its capabilities. This means, higher radiation doses and more focussed X-rays (microfocus) can be applied on smaller sample volumes. This results in higher resolutions and better image contrast. Furthermore, µCT systems can be better developed or adjusted to specific applications. In industry, µCT devices are for example used in investigating damage in machine parts (e.g. metal fatigue cracks), and for investigating internal properties of building materials. The use of CT and µCT in geosciences has greatly increased over the last two decades, due to a wide range of applications for 3D imaging of rocks, the clear advantages over destructive sampling, and a growing number of available µCT scanners.

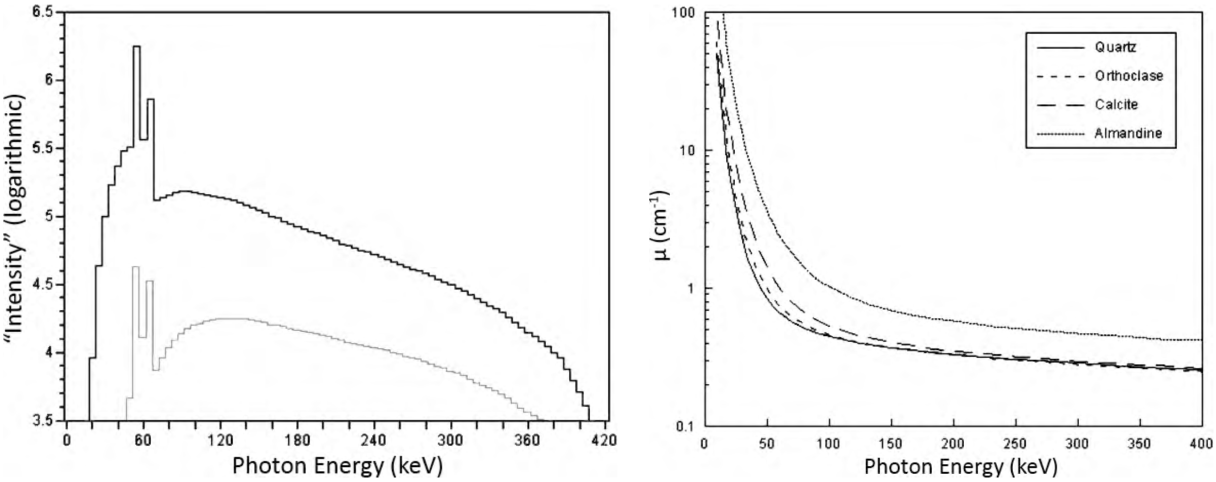


Figure C.1.1. Left – The black line shows the theoretical Energy Spectrum for a 420 kV X-ray source on a tungsten target. Note that all energies lower than the source output that is set (for example, 200 kV), are in fact outputted when scanning: the X-ray source is polychromatic. The grey line shows the spectrum after passing through 5 cm of quartz: the X-rays are attenuated, and the *average* energy level has *increased*. **Right** – Linear attenuation coefficient µ as a function of X-ray energy, for different minerals. Note the differential attenuation (a higher µ means more attenuation) of different X-ray energy levels, caused by various interaction mechanisms at play. The diagram also shows that at low energy levels, it is possible to distinguish between for example calcite and quartz. At higher energy levels this however becomes impossible. From: Ketcham & Carlson 2001.

For all CT devices, imaging is based on the concept of X-ray attenuation. For every material, the intensity (amount of photons) of incoming X-ray radiation is diminished inside the sample by various atomic scale interactions. See Equation C.1.1. hereafter. The longer the pathway x of the radiation through a sample, the more the radiation is attenuated. The rate at which this occurs is determined by the attenuation coefficient µ. The higher this µ, the more the X-rays are attenuated (when comparing the same sample). The attenuation coefficient depends, among others, on the X-ray energy E, the material density ρ, and the atomic number Z of the material. For rock samples, a relatively high X-ray energy E is required (above 100 keV), and therefore the attenuation coefficient µ is primarily a function of density ρ (meaning little atomic number Z dependence). Figure C.1.1 shows intensity spectra of X-rays for different energy levels and the dependence of µ on the energy level.

$$I = I_0 \cdot e^{-\mu x} \text{ (Beer-Lambert Law) with } \mu = \rho\left(a + \frac{bZ^{3.8}}{E^{3.2}}\right) \qquad \text{(Equation C.1.1)}$$

(I = Intensity, I_0 = Initial intensity, μ = Attenuation coefficient, x = Path length through sample, ρ = Density, Z = Atomic number, E = Energy, a,b = Instrument dependent parameters).

To avoid confusion, note that in a microfocus X-ray tube, the X-ray photons are generated by the collision of electrons with a target. The acceleration of the electrons is generated by applying an electrical potential, for example 200 kV. Per definition, every one of these electrons now has an energy of 200 keV. This energy is (partly) transferred in the collision to X-ray photons, bearing energies in keV, with in this case a maximum of 200 keV. Due to this, the terms kV and keV are sometimes interchanged in X-ray literature.

The µCT systems used in this study are of the so-called third generation, cone beam scanning type. In this setup, the X-ray source and detector are fixed, and the sample is mounted on a rotating and movable plate between source and detector. The sample rotates 360° in certain predefined steps, and after each rotational step, a 2D X-ray image (radiogram) is recorded. These radiograms are then reconstructed into a 3D volume, where ideally, the grey-scale of every volume element (voxel) represents the characteristics (for rocks, mainly the density) of the material at that point in the sample. A diagram of this process is shown in Figure C.1.2. In this study, the full 3D dataset eventually has to be processed so the porosity can be extracted, which can then serve as input for further analysis and modelling.

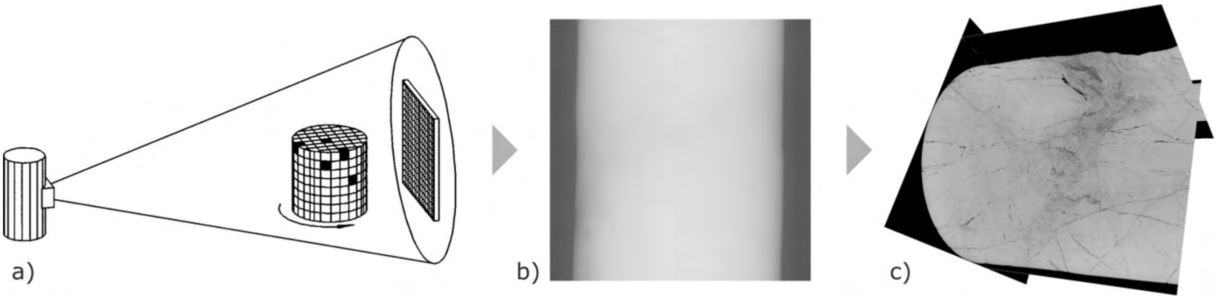


Figure C.1.2 a) Schematic drawing of a cone beam scanning setup. The sample rotates stepwise within a cone-shaped X-ray beam, generated by the X-ray source. For each rotational step, a so-called radiogram is recorded by the detector. From: Klobes et al. 1997. **b)** Example of a radiogram, representing one single rotational step in the µCT scan. All radiograms of the µCT scan are combined by a process termed reconstruction into 3D volumes and/or 2D stacks of slices. **c)** Slice-based representation of sample Prottes TS1 after reconstruction.

Several problems have to be considered in the process of µCT data acquisition and processing. To ensure a good understanding of the possibilities and limitations of applying µCT on Hauptdolomit samples, some of the most important problems are explained here:

Acquisition

- *Resolution, noise, penetration, and contrast.* Several trade-offs need to be considered in the scanning process. For proper imaging, the X-ray beams have to be focussed on the sample. Smaller focal spots give sharper results (i.e. a higher resolution). High intensity or power (high tube voltage in kV and high tube current in µA) provides better counting statistics, meaning less image noise. However, the maximum power on the focal spot is limited in the machine, to prevent thermal damage to the metal target inside the X-ray source. This means the spot size has to be increased when applying a higher power. Another important point is that higher energy X-rays (controlled by the tube voltage in kV) have more sample penetrating power. However, these higher energy X-rays are also attenuated less inside the sample, therefore resulting in less image contrast. To summarise, the described effects mean that for big and/or dense samples, the required X-ray energies are high, resulting in high power, thereby requiring a large spot size. So the bigger and/or denser the sample, the lower the spatial resolution, and most likely also the lower the contrast. The detector exposure times are often changed for obtaining more signal (e.g. "compensating" for a low current in µA), but this also has drawbacks in terms of noise and an increase of scan times.

- *Beam hardening.* Most X-rays sources (as the ones used in this study) are polychromatic, meaning they produce X-rays with a range of different energies (as seen in Figure C.1.1). Different energy X-rays are attenuated in a different manner in the sample. Low energy X-rays are quickly attenuated completely. Higher energy X-rays are attenuated less and thus penetrate the sample better. Therefore, for short X-ray paths through the sample (e.g. at the edges of a rock core) more low energy X-rays finally reach the detector, than for long X-ray paths (e.g. through the centre of a rock core). This results in a gradient in greyscales across 2D radiograms and in the final 3D reconstruction, even if the samples are completely homogenous. It also means that if X-ray energies are not high enough, penetration can be limited in sample centres, showing limited contrast in these areas. Figure C.1.3 shows a schematic drawing of the beam hardening effect. An example in one specimen is shown in Figure C.1.5. Note that due to the conical shape of the X-ray beam, also some beam hardening effects exist in the z-direction (i.e. the length direction of the sample cylinder) – See for example Figure C.1.6. Beam hardening effects can be reduced by pre-filtering the X-rays from the source, e.g. by a copper plate, reducing the amount of low energy X-rays. Naturally, this also reduces contrast and X-ray intensity, increasing the amount of noise.

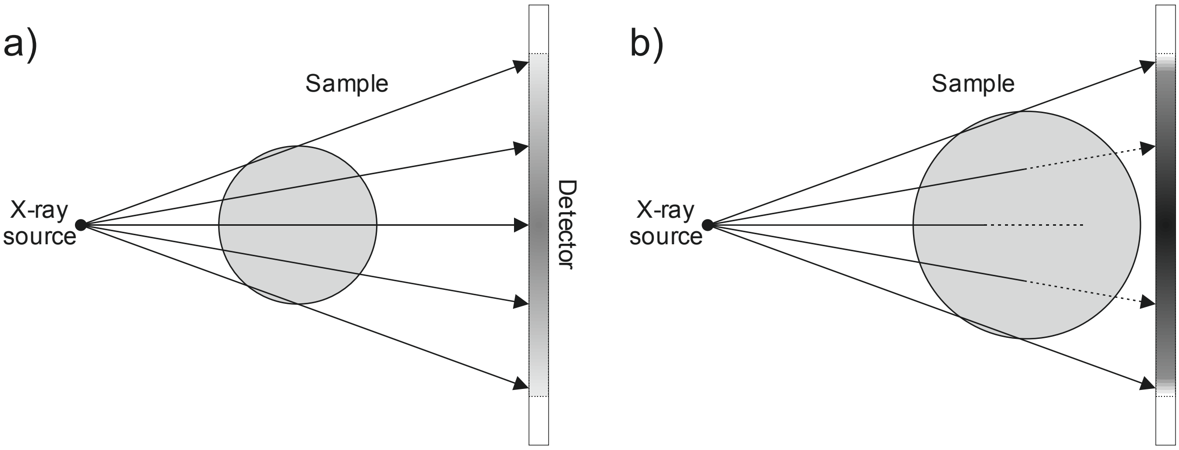


Figure C.1.3 Schematic drawings of beam hardening problems. **a)** Slight beam hardening in a small and/or not very dense sample. At the edges, the X-ray paths through the sample are short, and only the very lowest energy X-rays are attenuated significantly. Through the centre of the sample, more attenuation takes place. Overall, this results in a small gradient in greyscales on the detector (even if the sample is completely homogenous). **b)** Severe beam hardening in a large and/or dense sample. The effects at the edges are similar as before, but towards the centre, less and less X-ray beams can pass through. In the extreme case drawn here, no X-rays at all can pass through the centre anymore. The result is a large greyscale gradient on the detector, and worse, less or even no information towards the centre. Beam hardening and sample penetration problems will show up in the final reconstruction as well. Pre-filtering the X-rays and software corrections are used to reduce the effects.

- *Technical scanner variability.* Various setups exist for µCT scanners. These can, in combination with the reconstruction process (see below), give variable results for the same samples. Two different µCT devices are used in this study: A Viscom X8060 NDT at the Department of Anthropology of the University of Vienna, and a Rayscan 250 E at the University of Applied Sciences of Upper Austria (Fachhochschule Oberösterreich, FHOÖ) in Wels (Figure C.1.4). Theoretically these machines have quite similar capabilities (especially for rock samples), but they differ in their appearance, detailed characteristics, their normal field of use, and the hardware and software used for reconstruction and post-processing. Better results for the Hauptdolomit samples have been obtained at the FHOÖ Wels scanner: The most pronounced differences are caused by the different detectors used: shorter acquisition times at the FHOÖ Wels scanner allow for e.g. obtaining the same radiogram 5 times, to then average these images. This is a very effective way to reduce noise in the final reconstructed result. Therefore the scanner at FHOÖ Wels is the main source for µCT data in this study.

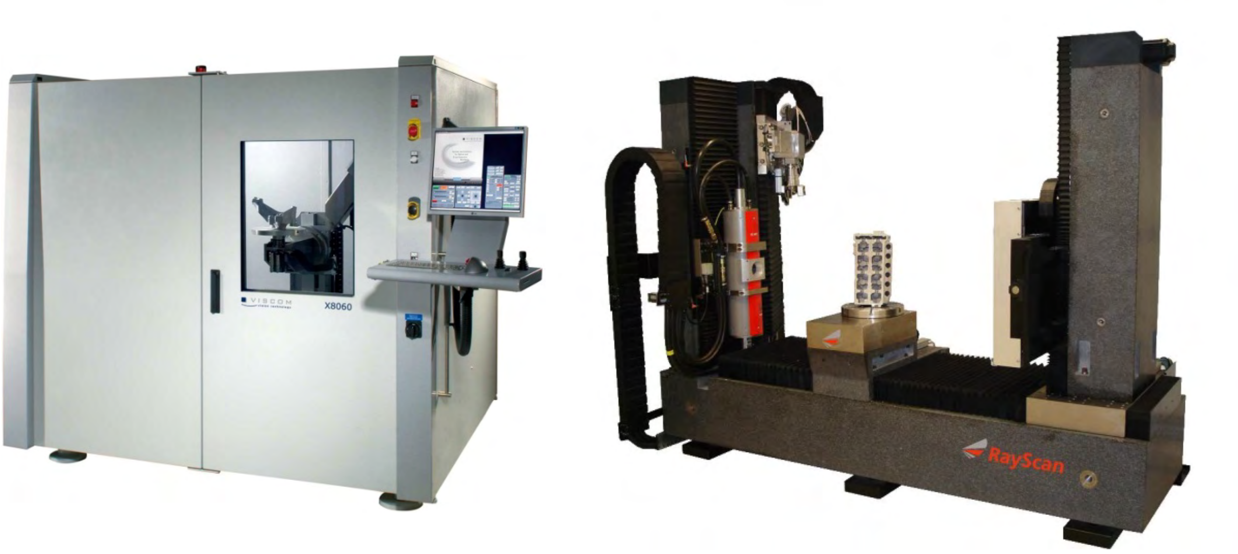


Figure C.1.4 Different µCT scanners used in this study. **Left** – Viscom X8060 NDT Scanner, as at the Department of Anthropology of the University of Vienna. The device is ca. 1.9 m in height. **Right** – RayScan 250E, as at the FHOÖ Wels. The width of the machine is ca. 2.5 m.

Reconstruction

- *Software variability.* The 2D radiograms are combined by software into 3D volumes by a mathematical process called "Filtered Back Projection". Different variations on this algorithm are differently applied in various software packages. Also other processing steps, e.g. noise reduction and software-based beam hardening correction, are applied differently in each software package. The problem is, there is often little control on parameters, and in many cases the processing steps and parameters are not documented (i.e. are "company secrets"). There is however a large variation in reconstructed output from the different software packages, sometimes even showing clear erroneous results (e.g. the same fracture is visible twice in a sample). Examples of reconstruction differences are shown in Figure C.1.7 and Figure C.1.10.
- *Ring artefacts.* Pixel defects or bad calibration can cause certain pixels on the µCT detector to behave differently from how they should. If this occurs in every radiogram, the reconstructed image may show clear circular features called ring artefacts. Most reconstruction software packages try to compensate for this effect, with variable success. Ring artefacts may cause problems in processing the data in the next step for porosity calculation.
- *Streaking artefacts and scattering radiation.* Related to beam hardening effects, sudden high density gradients (e.g. a piece of metal in the sample) can cause ghost lines and scattering stars to occur in the reconstructed result. See Figures C.1.8 and C.1.9 for examples. A careful choice of scanning parameters and software post-processing (or using different software packages) can reduce these effects to some extent. However, if possible, trying to avoid these problems before scanning is always the best choice.

C.1.2. µCT on Hauptdolomit samples: possibilities and limitations

For displaying purposes, most of the images shown here are 2D slices through the full 3D datasets. All datasets have already been obtained at the highest possible resolution for the sample, considering the technical limitations mentioned before.

10 cm cores:
Problematic beam hardening and sample penetration

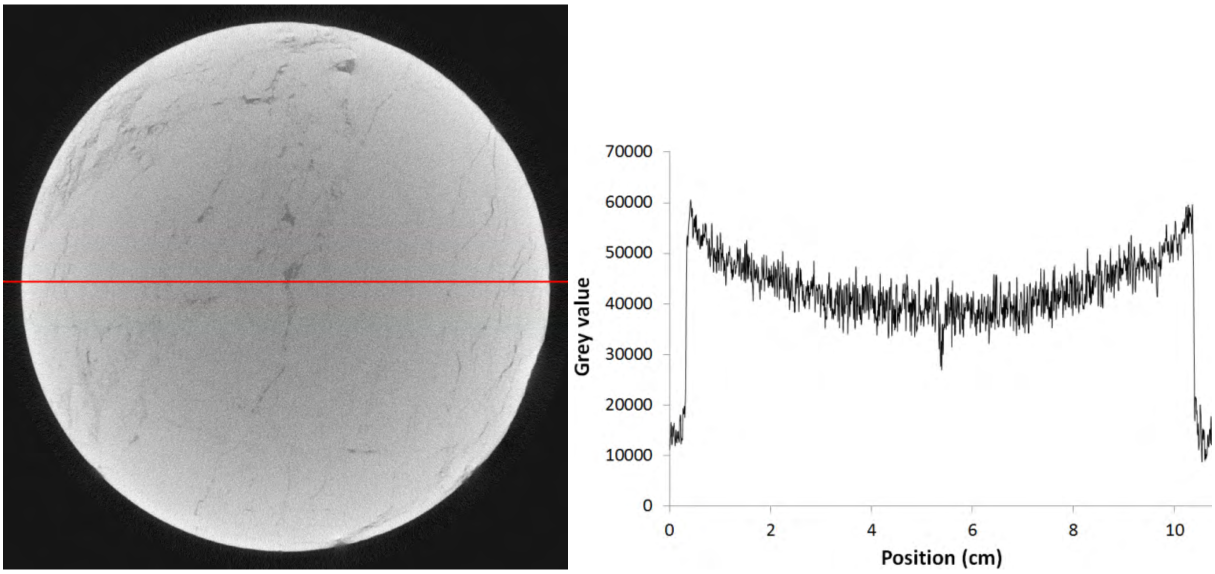


Figure C.1.5 10 cm core sample Schönkirchen T91-5, scanned and reconstructed with (55 µm)^3/voxel resolution at FHOÖ Wels. The sample shows limited penetration in the centre, and severe beam hardening (brighter rim) as shown by the greyvalues profile.

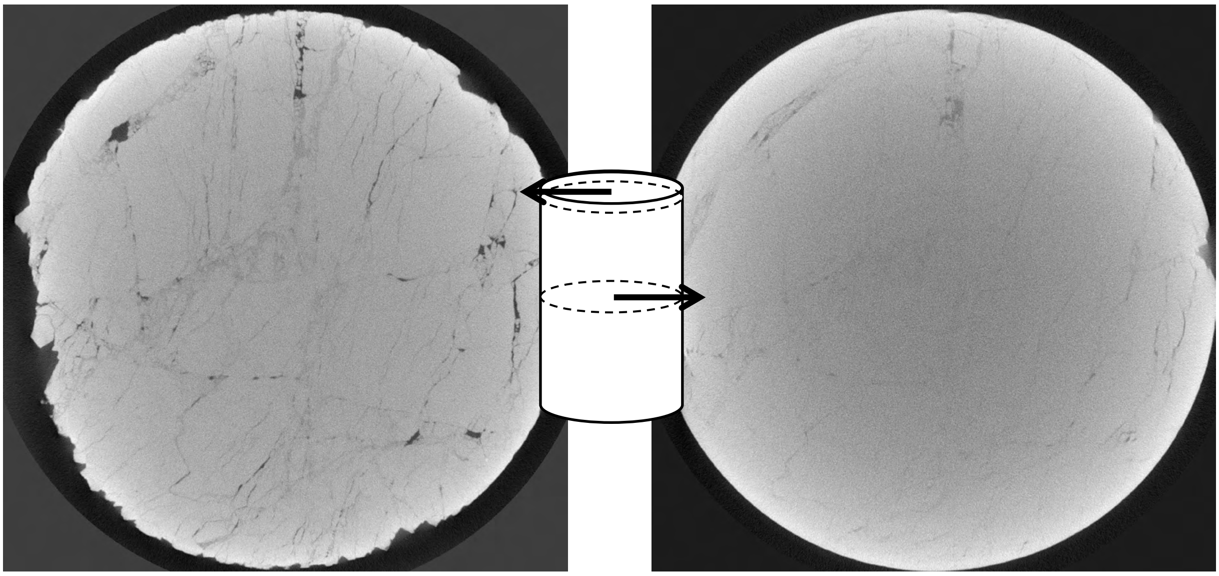


Figure C.1.6 10 cm core sample Schönkirchen T91-5, scanned and reconstructed with (55 µm)^3/voxel resolution at FHOÖ Wels. A slice from the top of the core (left) shows quite some detail and contrast. A slice from the centre of the core shows very little detail and contrast (right). This illustrates another problem with penetration depth and beam hardening. Due to the conical shape of the X-ray beam, X-rays passing through the top or bottom of the core are inclined to the core's orientation. Therefore, some X-rays will leave the sample quickly, hence are not attenuated that much. This results in less beam hardening and a higher contrast. In the centre of the core, X-rays are not inclined to the sample, and thus always have to pass through the entire core. This results in more beam hardening so a limited contrast. From this core, a 3 cm plug has been drilled to investigate the fractures in a smaller area.

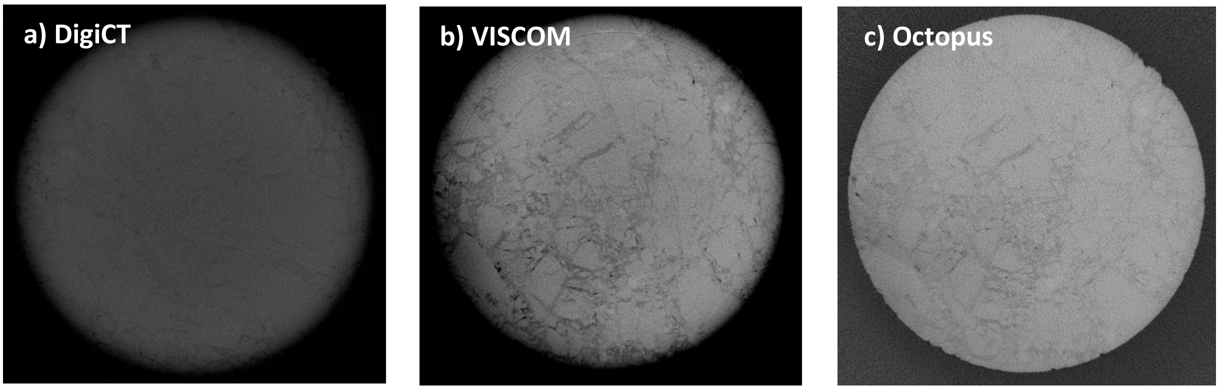


Figure C.1.7 10 cm core sample Prottes TS1-Nr1-Slice5, scanned and reconstructed with (50 µm)^3/voxel resolution at the µCT lab at the University of Vienna. All slices a-c originate from the same µCT scan, represent (approximately) the same position at the edge of the core, but are reconstructed with different software packages. **a)** DigiCT reconstruction, showing extremely little detail, a lot of noise and over-corrected beam hardening (the edge is darker than the centre). **b)** VISCOM reconstruction, showing quite a lot of detail, but also a lot of noise, and over-corrected beam hardening. Also, one ring artefact is very prominent, mainly in the left-top corner. **c)** Octopus reconstruction, showing some detail, but limited contrast, and quite some noise. Beam hardening effects are not apparent. Some blurry edges may indicate doubling of features. This doubling has been observed for all different reconstruction software packages in different slices.

6 cm core Ebental T1:
May show sufficient X-ray penetration, but suffers from beam hardening, ring artefacts and noise. Additionally, severe streaking and scattering effects by a dense phase.

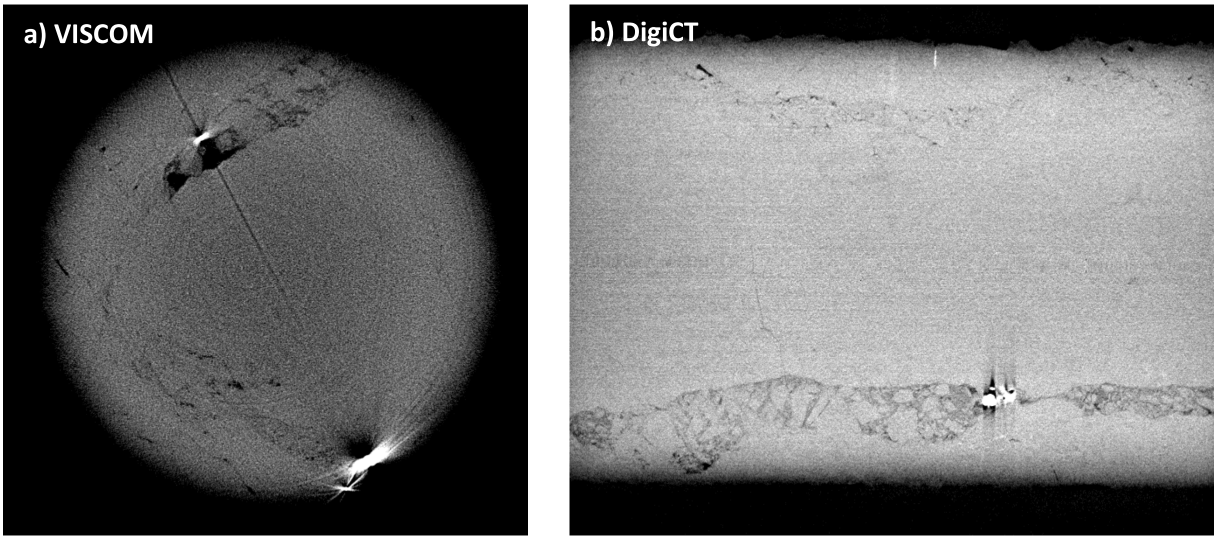


Figure C.1.8 6 cm core sample Ebenthal T1, scanned and reconstructed with (50 µm)^3/voxel resolution at the µCT Lab at the University of Vienna. **a)** VISCOM reconstructed slice. **b)** DigiCT reconstructed slice in a different orientation (cut along the core's length direction). Similar problems as for the 10 cm diameter cores occur, but some more detail is observable due to the smaller sample size. However, the scan suffers from very limited contrast due to dense phases (most likely pyrite), resulting in a lot of noise, streaking and scattering artefacts.

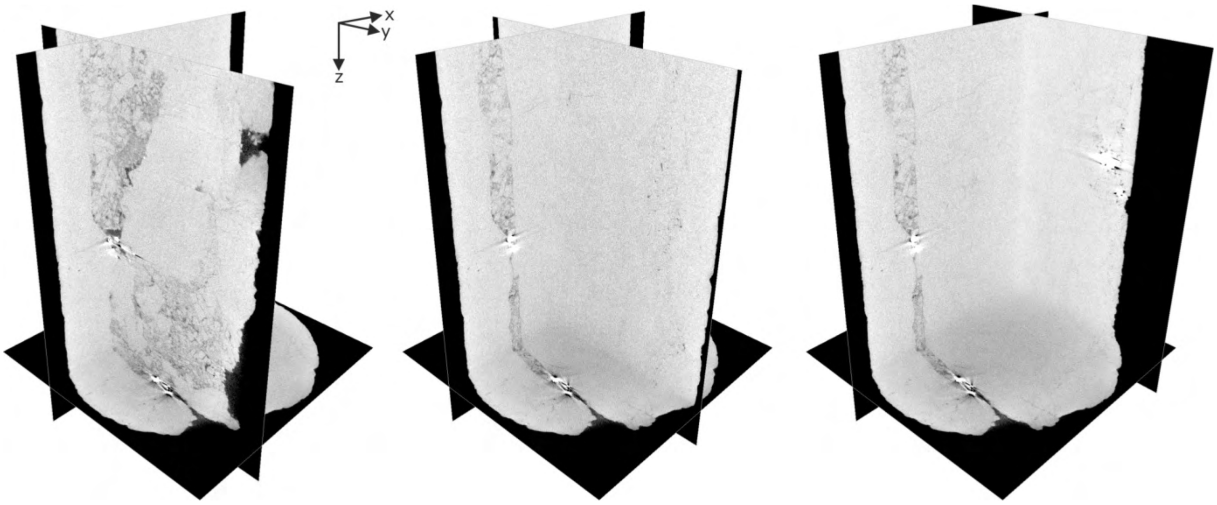


Figure C.1.9 6 cm diameter sample Ebenthal T1, in a test scan without averaging, reconstructed with (50 µm)^3/voxel resolution at FHOÖ Wels. Slice-based 3D visualisation of the data, using the Amira® visualisation software, moving in the positive x-direction. The results for this sample are clearly better than from the Vienna µCT lab, but several problems remain. Although some structures are well visible, large artefacts are still introduced by a dense mineral phase present in the fractures (most likely pyrite). Additionally, especially in the centre of the dataset, a lot of noise is present and almost no structure can be observed. These problems unfortunately hamper viable postprocessing of the data.

3 cm plugs:
Good results from FHOÖ Wels µCT.

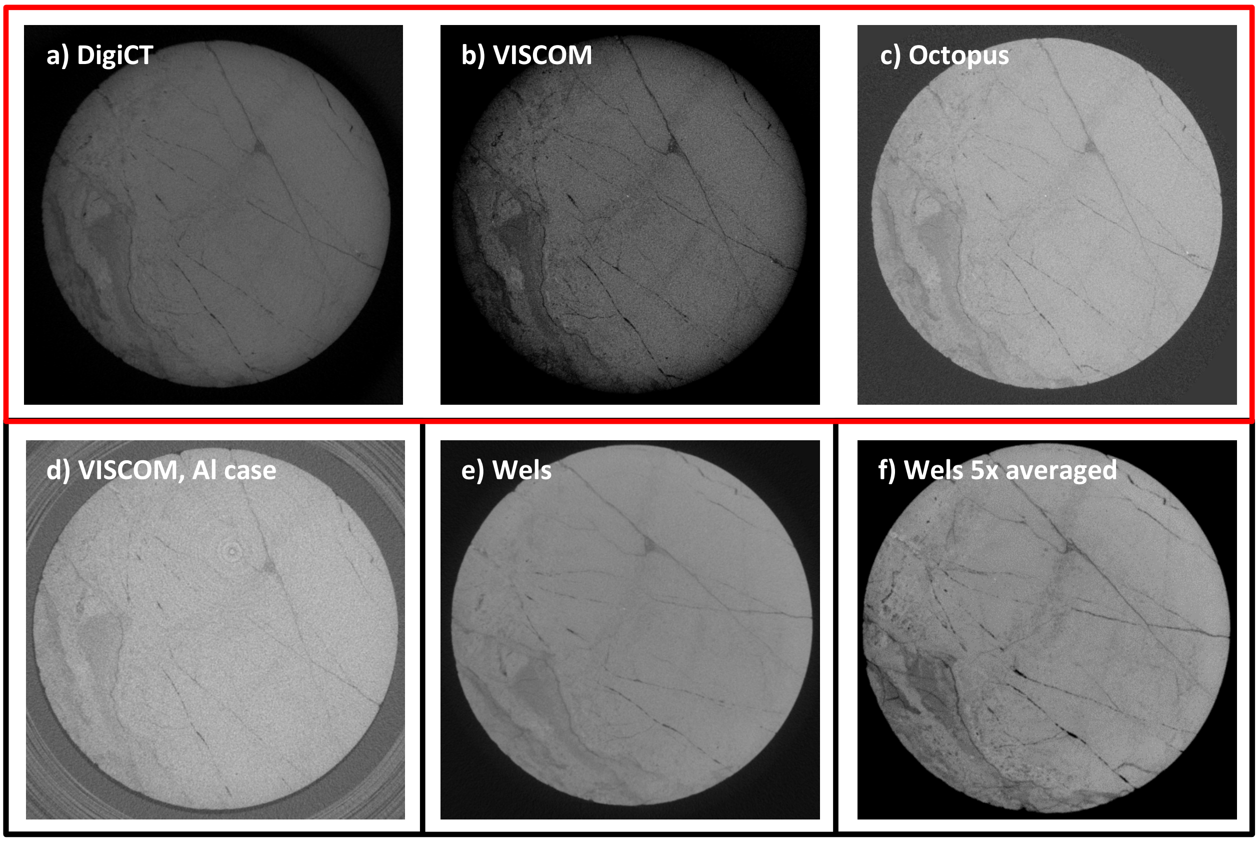


Figure C.1.10 3 cm plug sample Prottes TS1, different scans and reconstructions. **a-c)** Different reconstructions of the same (25 µm)^3/voxel resolution scan at the µCT lab at the University of Vienna. Sample penetration is quite good, due to the small sample size, but noise, beam hardening (or overcorrection of it), ring artefacts and doubling of features are evident. **d)** Same sample, but wrapped in an aluminium cover, scanned and reconstructed with (25 µm)^3/voxel resolution at the µCT lab at the University of Vienna. The aluminium casting reduces beam hardening (as it was intended for), but also results in less contrast and more noise, and prominent ring artefacts. **e)** Scanned and reconstructed with (25 µm)^3/voxel resolution at FHOÖ Wels. Shows virtually no beam hardening, less noise, but lacks some contrast and sharpness. **f)** Scanned with 5x averaging (i.e. making 5 radiograms for each position and average these out for noise reduction and contrast increase) and reconstructed with (18.9 µm)^3/voxel resolution at FHOÖ Wels. The averaging results in a lot more contrast, sharpness and detail. Virtually no beam hardening, some ring artefacts, and some noise. This dataset represents the best result obtained for this sample, and a similar approach is used for other samples.

2 cm plugs:
Good results from FHOÖ Wels µCT.
Very similar to results from 3 cm plugs, but having a better resolution.

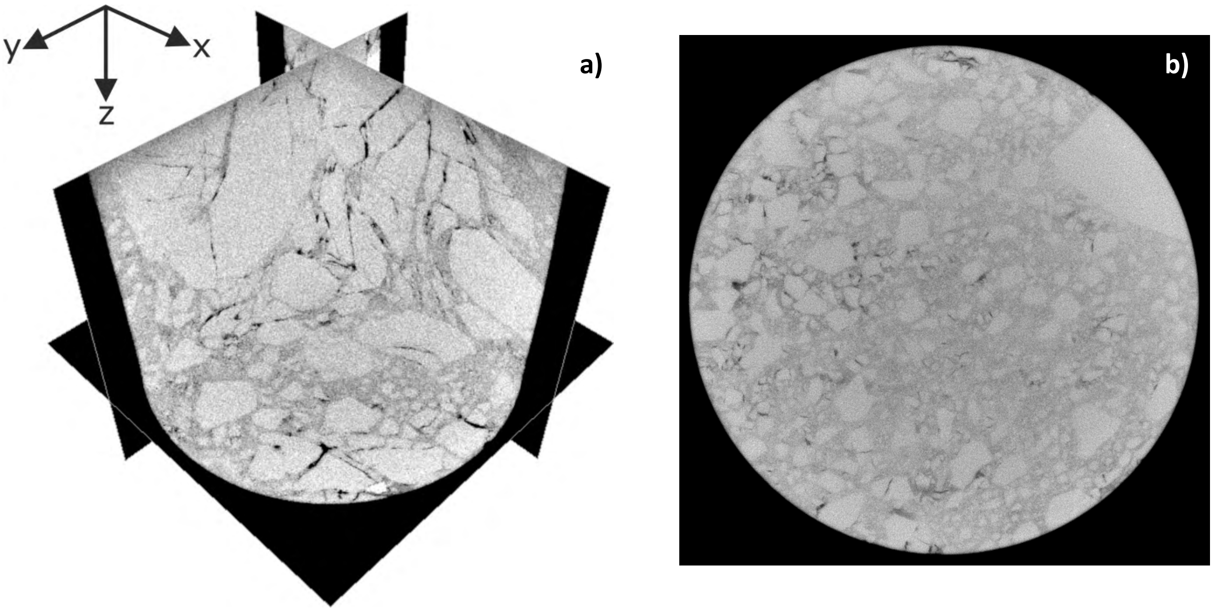


Figure C.1.11 Examples of µCT scans of 2 cm diameter plugs, performed at FHOÖ Wels. **a)** Sample Prottes TS1-3sp, scanned and reconstructed at (12.5 µm)^3/voxel. Slice-based 3D visualisation of the data, using the Amira® visualisation software. **b)** Sample Strasshof T4-2sp, scanned and reconstructed at (12.5 µm)^3/voxel. Slice outtake from the 3D µCT scan.

C.2. µCT data processing: an introduction

The final and workable reconstructions of the Hauptdolomit µCT data need to be processed in order to extract the porosity, after which for example porosity distribution, fracture distribution, and permeability can be assessed. Several problems are encountered in this process, and very different approaches can be taken. A short introduction is given here.

C.2.1. Effects in the data

- *Partial Volume Effect.* An important effect to consider is inherent to all µCT datasets, and termed Partial Volume Effect (PVE). Ideally, the greyvalue of every reconstructed voxel represents the material characteristics of the respective point in the sample. This is however not the case, even when undesirable effects, e.g. noise, ring artefacts or beam hardening, are reduced to a minimum. The problem lies in features that are smaller than the spatial resolution of the dataset, e.g. very narrow fractures or small pores. Due to this, the displayed greyvalue of the voxel is a mixture of the actual greyvalue of the narrow feature and the greyvalue of the surroundings (Figure C.2.1). This happens on multiple levels and scales, thereby complicating the analysis of the data.
- *Data blurring.* An effect that can be of importance for the narrow fractures in the Hauptdolomit is data blurring. Please note this effect is sometimes grouped with the PVE. Signals originating from one point (e.g. in a fracture) are blurred over their surroundings to some extent. Due to this effect, fractures that are in reality narrower than the spatial resolution of the scan can appear blurred but broader on the final scan results (Figure C.2.2). Promising attempts are made to use and quantify these effects, providing true fracture apertures (Ketcham et al. 2010; Ketcham & Hildebrandt, 2014). These are however not yet applicable on full µCT data sets of natural fractures.

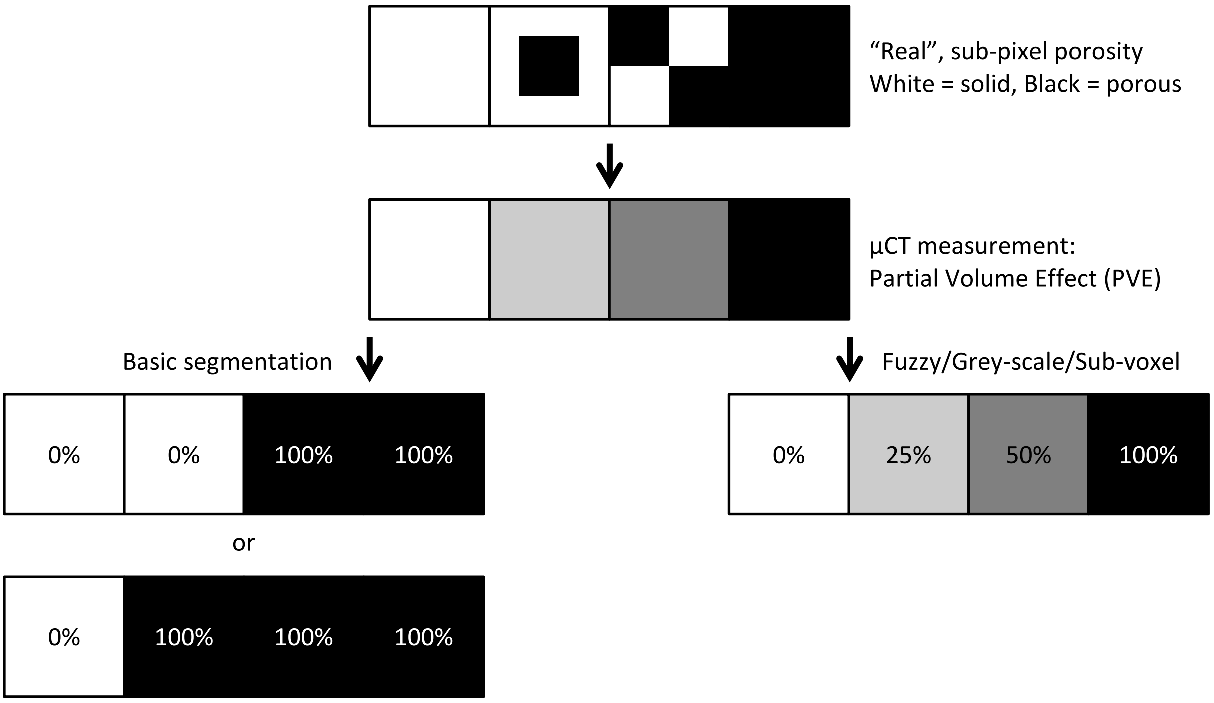

"Real", sub-pixel porosity
White = solid, Black = porous
µCT measurement:
Partial Volume Effect (PVE)
Basic segmentation
Fuzzy/Grey-scale/Sub-voxel
0%
0%
100%
100%
or
0%
100%
100%
100%
0%
25%
50%
100%

Figure C.2.1 (previous page) Two-dimensional illustration of the Partial Volume Effect, and the difference between segmentation (left) and fuzzy approaches (right) on the data. Fuzzy approaches are also called grey-scale methods or sub-voxel methods. Percentages indicate the amount of inferred porosity. Segmentation depends greatly on the settings and the chosen algorithm. One has to assign a voxel to either pore space or solid. The boundaries set can be very ambiguous, especially when applying thresholding only. For fuzzy techniques, many different approaches can be taken. Here, only a simple example of a linear approach is given. More sophisticated fuzzy techniques usually require a lot of computational power, and are often limited in their application. Created and extended after: Vergés et al. 2011.

C.2.2. Extracting petrophysical properties from datasets

- *Segmentation*. One obvious approach to process the 3D µCT data set is to extract the fracture network from the host rock by identifying the fractures as one phase, and the host rock as the other (in this case: converting the dataset to binary). Such a process is termed segmentation; a widely used technique, for which a tremendous amount of different approaches and algorithms exists. A rough division in these techniques is two-fold:
    - *Segmentation by thresholding.* The easiest way to try to segment a dataset is setting a so-called threshold. This means one creates a binary dataset by setting one single greyvalue where e.g. every greyvalue below it is attributed to fractures, and every greyvalue above it to host rock. In manual thresholding, the interpreter sets this value by themselves, which is of course quite ambiguous. Many algorithms exist that try to find the "optimal" threshold value by statistical analysis of the data (e.g. the Otsu threshold; Otsu 1979). This has the advantage over manual thresholding of being reproducible, but it is still ambiguous to choose which of these methods is the best. Furthermore, when for example beam hardening and noise effects are visible in the data, applying a threshold becomes particularly difficult. Narrow features induce even more problems (see PVE and data blurring). Direct thresholding is thus not applicable to the Hauptdolomit datasets. Also see Figure C.2.1 and Chapter D.
    - *Adaptive techniques and feature tracking.* More sophisticated segmentation routines use a wide range of different approaches to get to a more satisfying result. Many of these techniques still use thresholding at some stage. Some so-called locally adaptive techniques for example start from an initial conservatively thresholded image, and then elaborate on the differences to the direct neighbourhood of an already segmented feature. Other methods for example use algorithms to track particular features in the dataset. Such a technique is also used for the Hauptdolomit data sets, using the multiscale Hessian fracture filter (MSHFF) (Chapter D).

  It has been shown in literature that a clear flaw of segmentation methods is that they can produce very different results; in one study porosity estimates for the same sample varied from approximately 25 to 75% for different segmentation techniques and operators (Baveye et al. 2010, Figure C.2.3). However, applying a reasonable way to segment the data still is the most viable approach to work on the Hauptdolomit data sets.
- *Fuzzy/Grey-scale/Sub-voxel approaches*. As an alternative to full segmentation, approaches termed fuzzy (often applied in GIS and cartography), grey-scale or sub-voxel can be taken to process a dataset. Very different techniques techniques can be applied in this way. Some of these techniques try to calculate petrophysical properties such as porosity and permeability directly from the reconstructed data sets without segmentation and human input. Although sometimes successful, the application of these techniques is at the moment very limited. Moreover, the mathematics can become very complex, and the required computational power is large. Next to the direct calculation of properties, fuzzy approaches can also be used as a "soft" segmentation technique, where for example not only two classes are defined (e.g. porous vs. non-porous), but a whole range in between (see Figure C.2.1). The multiscale Hessian fracture filtering (MSHFF) approach as applied in this study (Section C.2.3) is capable of such output. However, we decided to still apply a full segmentation on the data, because the calibration of fuzzy data introduces additional problems, and further processing and analysis steps often turn out not to work without a full segmentation.

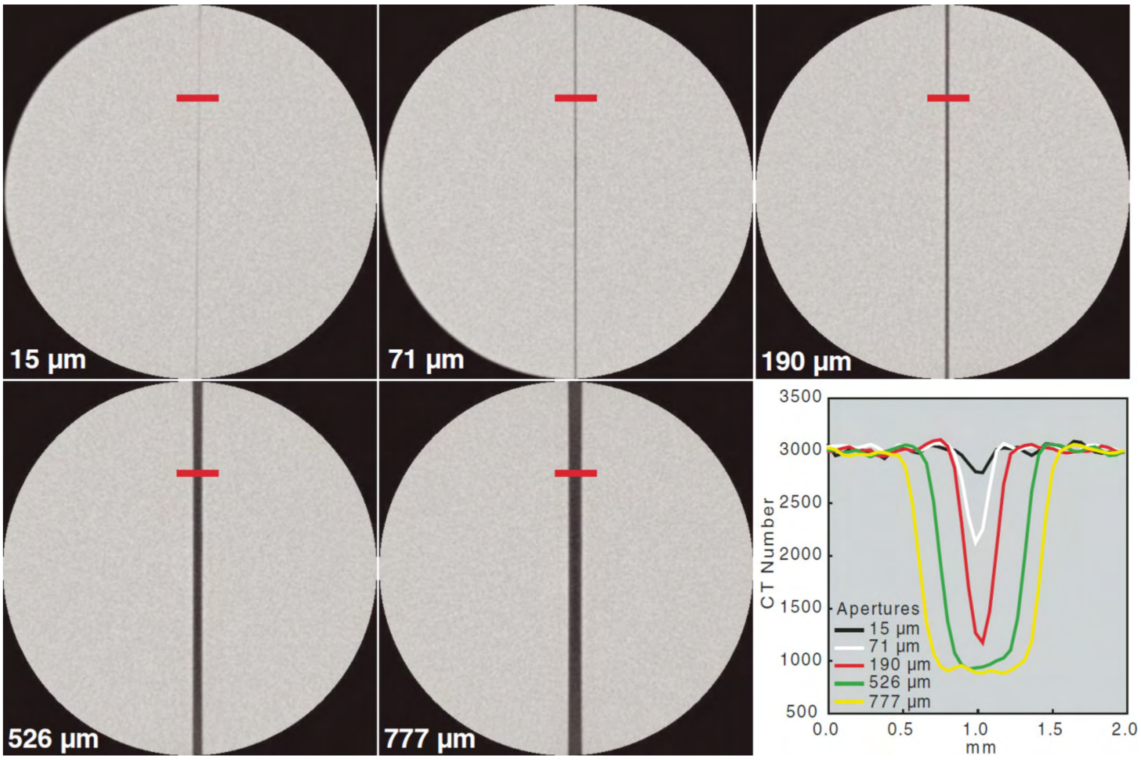


Figure C.2.2 Calibrated real fracture apertures represented in a µCT dataset. The spatial resolution of the different scans is 46,9 µm / voxel side. A narrower fracture of 15 µm however still shows up in the dataset due to the partial volume effect (PVE). In addition this fracture also appears wider than it should (i.e. even wider than 1 voxel side), due to blurring effects. Clearly, both the PVE and blurring can have large effects on the narrowly fractured Hauptdolomit data sets. From: Ketcham et al. 2010.

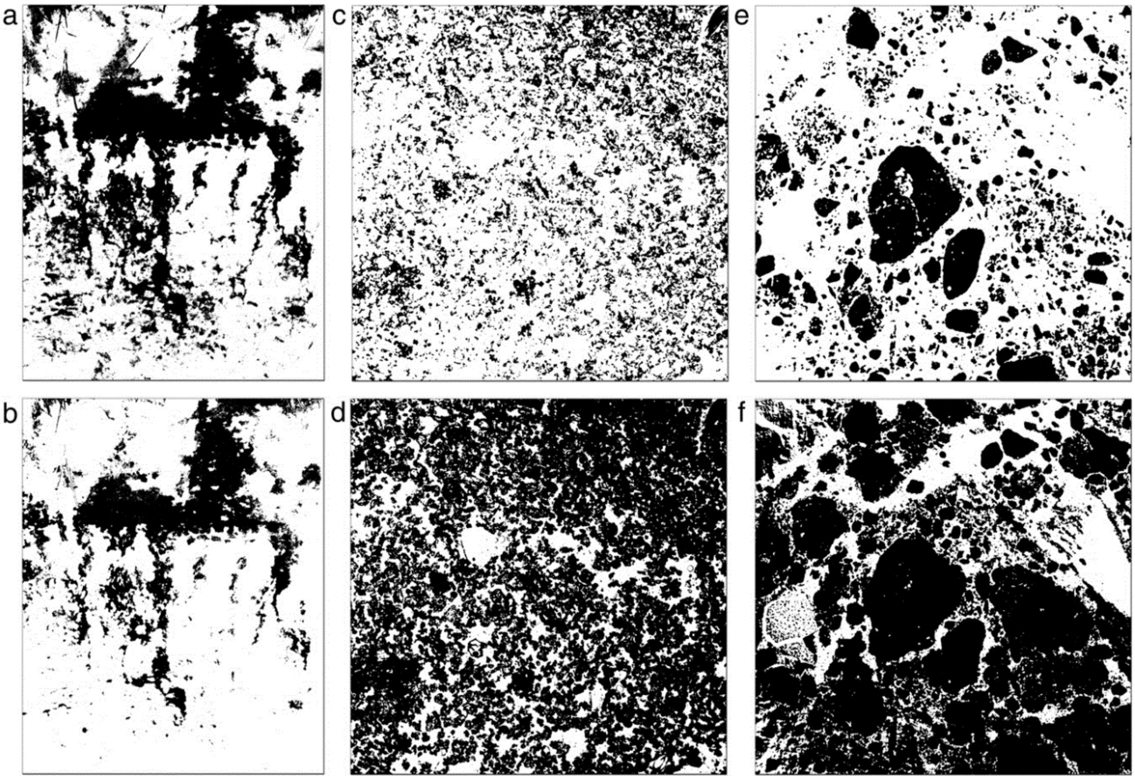


Figure C.2.3 Differences in porosity segmentation performed on photos from soil samples by different experts. The top and bottom rows represent the two extreme ends (under- and oversegmentation) as found in the study. The largest difference reported was ca. 25% versus 75% porosity for the same sample. From: Baveye et al. 2010.

D. µCT data processing: Multiscale Hessian fracture filter (MSHFF)

The following section has been published as a paper as:

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

The original text is presented here completely, to ensure internal cohesion within the paper. For the same reason, several improvements and adaptations to the method described in the paper are covered in Chapter E. The reference list to the paper is implemented in the complete reference list of this thesis.

Multiscale Hessian fracture filtering for the enhancement and segmentation of narrow fractures in 3D image data

Abstract

Narrow fractures - or more generally narrow planar features - can be difficult to extract from 3D image datasets, and available methods are often unsuitable or inapplicable. A proper extraction is however in many cases required for visualisation or future processing steps. We use the example of 3D X-ray micro-Computed Tomography (µCT) data of narrow fractures through core samples from a dolomitic hydrocarbon reservoir (Hauptdolomit below the Vienna Basin, Austria). The extraction and eventual binary segmentation of the fractures in these datasets is required for porosity determination and permeability modelling.

In this paper, we present the multiscale Hessian fracture filtering technique for extracting narrow fractures from a 3D image dataset. The second-order information in the Hessian matrix is used to distinguish planar features from the dataset. Different results are obtained for different scales of analysis in the calculation of the Hessian matrix. By combining these various scales of analysis, the final output is multiscale; i.e. narrow fractures of different apertures are detected. The presented technique is implemented and made available as macro code for the multiplatform public domain image processing software ImageJ. Serial processing of blocks of data ensures that full 3D processing of relatively large datasets (example dataset: 1670 x 1670 x 1546 voxels) is possible on a desktop computer. Here, several hours of processing time are required, but interaction is only required in the beginning. Various post-processing steps (calibration, connectivity filtering, and binarisation) can be applied, depending on the goals of research.

The multiscale Hessian fracture filtering technique provides very good results for extracting the narrow fractures in our example dataset, despite several drawbacks inherent to the use of the Hessian matrix. Although we apply the technique on a specific example, the general implementation makes the filter suitable for different types of 3D datasets and different research goals.

D.1. Introduction

With an increasing availability of imaging techniques in the geosciences, there is an increasing need for methods to extract information from such datasets. Three-dimensional imaging techniques, such as X-ray micro-Computed Tomography (µCT), nuclear magnetic resonance (NMR) tomography or Focussed Ion Beam - Scanning Electron Microscopy (FIB-SEM) tomography, generally provide a lot of data that need to be processed efficiently. As very specific and various problems can be addressed using such data, image processing and image analysis techniques need to be applied accordingly. In many cases, it is desired to define features in a dataset by simplifying the original (greyscale image) information. This is often done by segmentation, where a distinct set of classes is created and every single datapoint is assigned to a class. The simplest example of segmentation is binarisation, where every datapoint is assigned to either background (for example: solid material) or foreground (for example: pores). Such a simplified dataset can then serve as input for future analysis, such as permeability modelling. Besides segmentation, fuzzy classifications or other filtering techniques can be applied to simplify the data or improve visualisation, all depending on the final desired goals with the datasets.

In this paper, we cover the specific problem of narrow fractures – or more generally speaking narrow features with a planar shape – in 3D datasets. In our case, the fractures are of importance for the storage (porosity) and transport (permeability) of gas in a fractured dolomitic hydrocarbon reservoir (Hauptdolomit) below the Vienna Basin, Austria. From drillcores taken from these rocks at depth, cylindrical sample plugs of 2 and 3 cm in diameter were selected, and scanned using µCT. The details of these scans are not of importance for the objectives of this paper, since we only focus on the extraction of features from a 3D dataset. For a (geoscientific) overview on µCT scanning and important initial image processing steps required, the reader is referred to e.g. Stock (1999), Van Geet et al. (2000), Ketcham and Carlson (2001), Mees et al. (2003) and Cnudde et al. (2006). Since µCT data on rock samples is mainly a representation of material density, fractures are dark, and solid material is brighter, in greyscale images of such data.

The fractures in our datasets often have an aperture of <10 voxels, and various apertures are present throughout every single dataset. With features being that narrow, the partial volume effect (PVE) and blurring affect the reconstructed µCT data, resulting in the fractures showing up brighter ("having a higher density") than they theoretically should have, as empty space. This reduced contrast to the surroundings makes the extraction of these fractures difficult or impossible by global, greyscale based segmentation techniques. An example of simple global thresholding (binarisation by setting a single grey value, above which all voxels are assigned to one material, and below which all voxels are assigned to another material) on one of our µCT datasets is shown in Figure D.1. Clearly, using such a global threshold does not provide satisfying results for such datasets. This approach is hampered even more by other artefacts often present in µCT data, such as beam hardening effects, remnants of ring artefacts, and of course noise.

Because global thresholding techniques are usually not successful on extracting narrow fractures from a dataset, and manual digitising is not practical in such large datasets (billions of voxels), local and/or structure-based techniques are more viable. Various approaches exist in literature to process fractures in 3D (µCT) datasets (e.g. Keller, 1998; Van Geet and Swennen, 2001; Landis et al., 2003; Sellers et al., 2003; Karpyn et al., 2007; Christe, 2009; Ketcham et al., 2010), of which some indeed apply such a more sophisticated technique. However, problems with these techniques are that they often still do not work for narrow fractures as in our example datasets. For example, they are not generally applicable (e.g. only work on one single fracture in a whole dataset), have no published codes or require expensive commercial software packages, require a lot of processing power, or only process data in two dimensions. Because of these limitations, in this paper we present a different approach, based on the Hessian matrix, for processing narrow fractures in 3D datasets. This approach allows us to extract the fractures of various apertures efficiently from our µCT datasets, without the requirement of extensive computational power. The automated implementation is available for use in the multiplatform public domain software ImageJ (Rasband, 2012). Note that although we use the example of µCT data here, the technique can potentially be used for any 3D dataset.

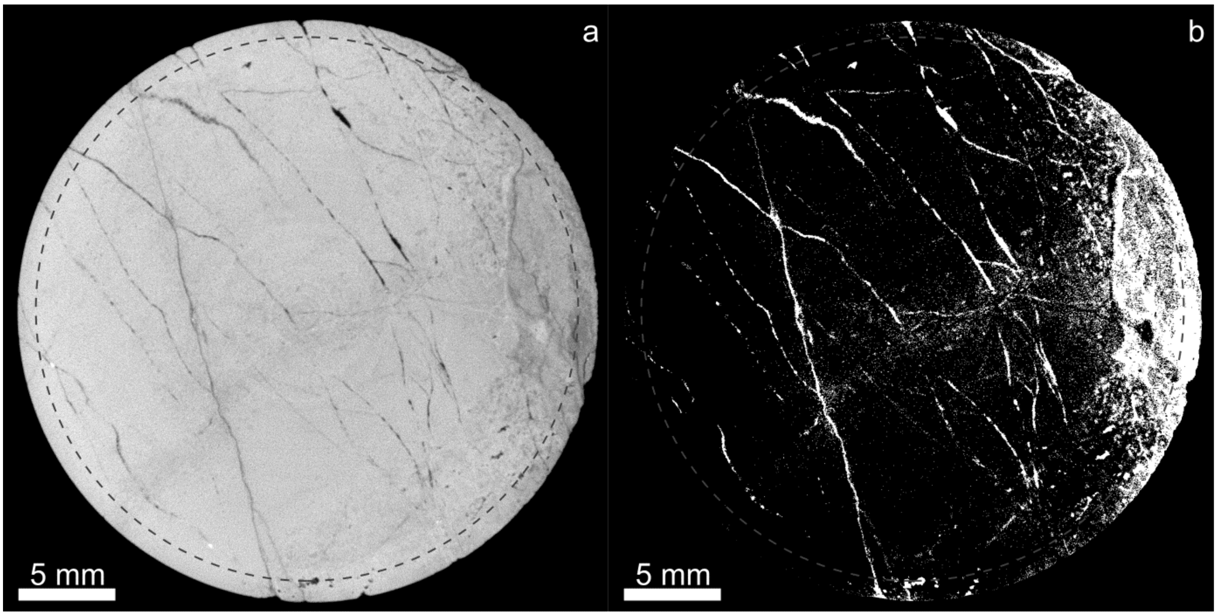


Figure D.1. Input data and binary thresholding. The dashed circle indicates the final ROI as used in the Hessian analysis, to allow comparison to Figures D.3 and D.4. Contrast and brightness settings have been altered for display. (a) 2D slice from 3D µCT dataset Prottes TS1, reconstructed at $(18.9\ \mu m)^3$ per voxel. Fractures crosscut a mainly dolomitic host rock. Fracture apertures vary, and larger porous areas can be observed. Material with a lower density, interpreted as more clay rich in hand specimen, is visible on the right hand side of the image. (b) Binarisation of the same slice as in (a) by global thresholding. The here chosen threshold value may be quite high, as especially the right hand side is heavily oversegmented. However, at lower threshold values, undersegmentation becomes a problem in different regions.

D.2. Methods and results

D.2.1 Introduction to feature detection using the Hessian matrix

D.2.1.1 Hessian matrix characteristics

The Hessian matrix is, for 3D image data, a 3x3 symmetric matrix, containing the second order partial derivatives of the inputted image data *I(x,y,z):*

$$H = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \qquad \text{(D.1)}$$

The Hessian matrix therefore "describes the second order structure of intensity variation around each point of a 3D image" (Sato et al., 1997). Since second order information describes curvature, another way to think about it is that the Hessian matrix represents the local curvature of the data in a small neighbourhood surrounding each voxel (e.g. Descoteaux et al., 2005).

Above interpretations can be better understood when taking the method of determining the elements of the Hessian matrix into account. An often applied approach (e.g. Lorenz et al., 1997; Sato et al., 1997; Frangi et al., 1998), comes from the linear space scale theory (overview in Lindeberg, 1998), stating that the second derivative of an image can be obtained by convolving the original image with the derivatives of Gaussians. For a single element of the Hessian matrix, this comes down to for example (for a point *I(x,y,z)*):

$$I_{xx} = \left(B \cdot \frac{\partial^2}{\partial x^2} G(x, y, z, s) \right) * I(x, y, z) \qquad \text{(D.2)}$$

where *B* is a factor that can be used for normalisation (not important here), *s* is a factor of scale (see Section D.2.1.2), and *G* represents a Gaussian function. In one dimension, the Gaussian function is defined as:

$$G(x, s) = C \cdot e^{-\frac{x^2}{2s^2}} \qquad \text{(D.3)}$$

C is again a factor that can be used for normalisation (not important here). The second derivative of this Gaussian function is displayed in Figure D.2a, and represents a probe kernel for contrast in the image (Frangi et al., 1998). Simply speaking, when carrying out the convolution of Equation D.2, the image data *I* is "compared" to the probe kernel (the second derivative of *G*). Therefore, the highest positive response is recorded in the case of a 100% match. This means, a high positive response is recorded for a dark feature on a brighter background (at the correct orientation and scale) (Figure D.2b). A high negative response is recorded for a bright feature on a darker background (Figure D.2c). Additionally, responses to lesser contrast features (Figure D.2d), narrower features (Figure D.2e) and broader features (Figure D.2f) are displayed for comparison.

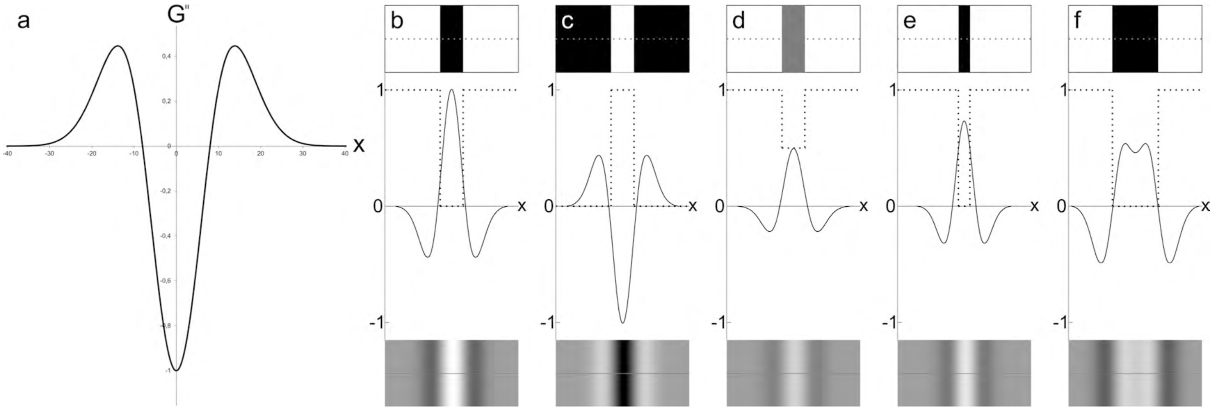


Figure D.2. One-dimensional second derivative of a Gaussian curve, and example results of the convolution to generate one element of the Hessian matrix. In all figures, the same scale setting (*s* = 8) is used for the Gaussian. The vertical axes are all normalised between -1 and 1. (a) Second derivative of a Gaussian curve, in one dimension (Equation D.2). Note the scale of 8 corresponds to the intersection of the curve with the x-axis (at -8 and 8). For the convolution, the best matches for this Gaussian curve will thus be of features of width 2*s* (in this case for a width of 16). (b-f) Top row: Input images (duplicated vertically for display). White = 1, Black = 0. Bottom row: Output of the 1-dimensional Gaussian convolution (duplicated vertically for display). White = 1, Black = -1. Middle row: Graphs displaying the normalised greyscale information of the input data (dashed line) and the output of the 1-dimensional Gaussian convolution (solid line). Note the same normalisations have been carried out for all figures, so the curves and images are directly comparable. (b) Maximum contrast input and maximum response of the convolution, for a feature with a matching width to the Gaussian convolution curve (2*s*; 16 in the example calculation). Dark feature on a bright background. A strong positive response is present for the dark feature. (c) Inverse case of (b). (d) Similar to (b), but for a feature with less contrast to the background (at intensity 0.5). Note the output curve is equally wide as for (b), but has half the intensity. (e) Similar to (b), but for a feature with half the width (*s*). Note the decrease in intensity of the output, as well as a rather broad representation of the feature: where the two curves in (b) crosscut near the x-axis, the two curves intersect much higher up in this graph. (f) Similar to (b), but for a feature with twice the width (4*s*). Note the decrease in intensity, as well as the dip in the centre of the output curve. This situation shows that for features that are very broad relative to the Gaussian scale, two parallel responses will occur.

When extending this concept to three dimensions, the convolution is carried out in multiple directions, and all elements of the Hessian matrix can be determined. Note that, as emphasised by Lorenz et al. (1997), the second-order nature of the Hessian matrix makes it invariant to grey value offsets, scaling and linear greyscale variations throughout a dataset. This means, only the local contrasts are assessed, but the actual greyscale values are not that important. For example, a feature with greyscale 10 on a background of 20, will give the same result as a feature with greyscale value 30 on a background of 40.

After all elements of the Hessian matrix are defined, the eigenvectors and eigenvalues of the matrix can be determined. The eigenvectors of the Hessian matrix describe the local principal directions of curvature, with corresponding eigenvalues λ for their magnitude (Frangi et al., 1998). The eigenvalues thus represent the magnitude of the largest local contrast change, as well as the magnitudes of the local contrast changes in the other two, orthogonal principal directions. The eigenvalue decomposition of the Hessian matrix can therefore be used to distinguish between blob-like, tube-like and plane-like features in a dataset. Table D.1 shows the relations between the eigenvalues for such different features. For our narrow fractures with a low intensity on a brighter background, we are interested in the case where one eigenvalue has a high and positive magnitude, and the other two have a small magnitude, as emphasised in Table D.1. The eigenvector corresponding to this largest eigenvalue is then the normal to the planar feature.

Table D.1. Possible feature patterns for Hessian matrix eigenvalues.

2D		**3D**			**Orientation pattern**
λ_1	λ_2	λ_1	λ_2	λ_3	
N	N	N	N	N	Noisy, no orientation pattern
		L	L	H+	Plate-like structure (dark) *
		L	L	H-	Plate-like structure (bright)
L	H+	L	H+	H+	Tubular structure (dark)
L	H-	L	H-	H-	Tubular structure (bright)
H+	H+	H+	H+	H+	Blob-like structure (dark)
H-	H-	H-	H-	H-	Blob-like structure (bright)

$|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$. H = high value. L = low value. N = noise. +/- = positive or negative eigenvalue. Dark = dark feature on a bright background. Bright = bright feature on a dark background. The fracture representation we are interested in is indicated with a *. After Frangi et al. (1998).

D.2.1.2 Multiscale processing

In the datasets, one should account for features with various apertures. This is possible by altering a parameter when generating the Hessian matrices. As shown in Equations D.2 and D.3 and Figure D.2, the used Gaussian functions introduce a parameter of scale, s. In the Hessian matrix calculation with the plugin FeatureJ, as used in our implementation (see Section D.2.2), this is the smoothing scale, defined as the standard deviation of the Gaussian derivative kernels (Meijering, 2010). A certain width scale eventually gives the highest responses for features with a specific width in the Hessian analysis. The highest responses in any dataset are obtained for maximum contrast features with a width of exactly 2x the applied Gaussian analysis scale (in the FeatureJ definition of scale. Also see Figure D.2). So, with higher smoothing scales, corresponding broader features give a high response in the case of planes. Of course, if one would be looking for blob-like or tube-like features, one would put an emphasis on respectively larger blobs (in 3 directions) and larger diameter (2 directions) tubular features. By combining the results of Hessian analyses on several scales, one gets a good representation of the features over a range of sizes.

D.2.1.3 Applications of Hessian feature detection

Approaches using the Hessian information of datasets for feature extraction are widespread in the medical sciences (e.g. Lorenz et al., 1997; Sato et al., 1997; Frangi et al., 1998; Descoteaux et al., 2005; Fornaro et al., 2010), but also in other research fields such as material sciences (e.g. Ehrig et al., 2011; Stoessel et al., 2012). The authors have not yet encountered it in geosciences-focussed literature involving feature enhancement or segmentation. The multiscale approach as taken in this paper has its main basis in the medical sciences, where a major application is the detection of (blood) vessels in 3D datasets, as applied in Lorenz et al. (1997); Sato et al. (1997) and Frangi et al. (1998). In their applications, one looks for tubular structures, and the main output for every voxel in the approach of Frangi et al. (1998) is a so-called vesselness value, which essentially indicates the likelihood of the presence of a vessel structure at that point in the dataset. Besides for vessel detection, the multiscale approach has already been adapted for the detection of planar features in 3D datasets, for the detection and segmentation of thin bone structures (Descoteaux et al., 2005). Instead of a vesselness value, a similarly defined sheetness value comes out of this approach. Although this sheetness approach has some similarities with the method presented in our paper, and would most likely provide good results too, our method differs in the order and nature of the calculations performed, and the way of calibration. Our method is characterised by simpler mathematics and later stage calibration. In both the vesselness and the sheetness approaches, the main calibration is introduced by sensitivity parameters defined at the start of the filtering process. Since analysis takes long on large datasets, we choose not to predefine all these sensitivity parameters at the start, and thereby ensure to keep some leeway after the most time consuming processing steps are taken. Additionally, this order of data processing is more suitable for a blockwise serial processing of the datasets.

D.2.2 Implementation

The hereafter discussed multiscale Hessian fracture filtering method is implemented and made available in macro code, with a graphical user interface (GUI), for the multiplatform public domain image processing software ImageJ (Rasband, 2012), based on Java. For the calculation of the Hessian matrices from the input data, the plugin FeatureJ (Meijering, 2010) is required. This plugin comes packed with the open source software FIJI (Schindelin et al., 2012), a distribution of ImageJ with additional plugins. Note that although ImageJ and FIJI support multi-processor and parallel processing, the Hessian matrix determination in FeatureJ does not.

The Hessian matrix for 3D image data contains 6 independent components for every voxel due to its symmetry. As this means that 7 (input data + 6 components), 32-bit floating point numbers for every voxel of input data need to be saved in the RAM memory, memory issues may occur for large datasets on computers with insufficient resources. As we wish for the method to still work on such computers (e.g. desktop PCs), we choose to apply a blockwise serial approach. This means, a suitably-sized (dataset and computer dependent) block of data is read into the RAM memory, on which the 3D calculations are performed, and the results are saved to the hard drive. After this step, the next block is read into the memory, with sufficient overlap to the previous dataset to ensure all final results are truly 3D processed. After all initial Hessian calculations are performed, the data is saved as a series of 2D images. Due to the introduction of so-called control lines (see Section D.2.3) to all this processed data, most normalisation, calibration and combination steps after the 3D Hessian calculations can be performed slice-based in 2D. This provides a clear advantage for the speed of processing, while the most important actual fracture detection is performed in 3D. The importance of such 3D processing, and the problems with 2D-only processing of 3D datasets, have been emphasised by for example Elliot and Heck (2007) and Iassonov and Tuller (2010).

The described approach ensures the analysis works on desktop PCs with limited resources (see Section D.3.1 for details). It is important to notice that because the intermediate steps are saved to the hard drive, quite some space may be required there. This can go up to several 100s of GBs, depending on the size of the input dataset and the chosen settings. After some initial setup and control steps, the entire filtering process runs without user intervention.

D.2.3 Multiscale Hessian fracture filtering

The initial step in the multiscale Hessian fracture filtering approach is a preparation step of the data. As input, a stack of 2D images is required, that together make up the 3D dataset. This input data should preferentially be stored as 8 bit or 16 bit greyscale data. A region of interest (ROI) is defined, and the exterior of this region is set to Not a Number (NaN), so that it does not influence the following filtering steps. At the same time, a set of control lines, with widths in accordance with the desired analysis scales (see Section D.2.1.2), is added to every 2D data slice. The contrast of these control lines to their background is calibrated to the contrast of the fractures in the image data by the user. Figure D.3a shows an example of the result of these preparation steps on the data.

After data preparation, the Hessian matrices are determined and the eigenvalues are calculated (output of the FeatureJ filter). The data is processed in the blockwise manner described in Section D.2.2. On every block of data, the desired scales of the Hessian matrix are calculated consecutively. The calculations performed on the output eigenvalues are simple. We are only interested in the case of one large positive eigenvalue and two (absolute) small ones (Table D.1). By setting the definition of the output eigenvalues to $\lambda_1 \leq \lambda_2 \leq \lambda_3$ (hence NOT absolute as in Table D.1), we can use: $A_s = \lambda_{3,s} - |\lambda_{2,s}| - |\lambda_{1,s}|$ if this gives $A_s > 0$; else $A_s = 0$. A is the output intensity, and s represents the different (Gaussian) scales. In this analysis, A_s gets the highest values for the clearest planar features at that scale, that have one single large positive eigenvalue and two near-zero eigenvalues, and takes a consecutively lower value for all less clear planar features.

After the above calculation is performed on every voxel in the dataset for several scales, a normalisation step is carried out. For every image, the following normalisation is performed: $A_s \rightarrow \hat{A}_s$, where $0 \leq \hat{A}_s \leq 1$.This normalisation is done relative to the maximum value in the area with the control lines. This maximum value is always present at the control line of which the width corresponds best to the calculated scale. Throughout the image stack, this value remains constant for every scale. Figure D.3b and D.3c show the normalised output on two different analysis scales.

After the normalisation, the scales are combined by $\hat{A} = \max(\hat{A}_s)$, for $s_{min} \leq s \leq s_{max}$. This means that for every voxel in the dataset, the maximum response of the Hessian analysis over all scales is taken, thereby combining the various apertures. The result of this is shown in Figure D.3d, and represents the final output of the (initial) filtering.

A complete overview of the parameters required for the multiscale Hessian fracture filtering approach is shown in Table D.2.

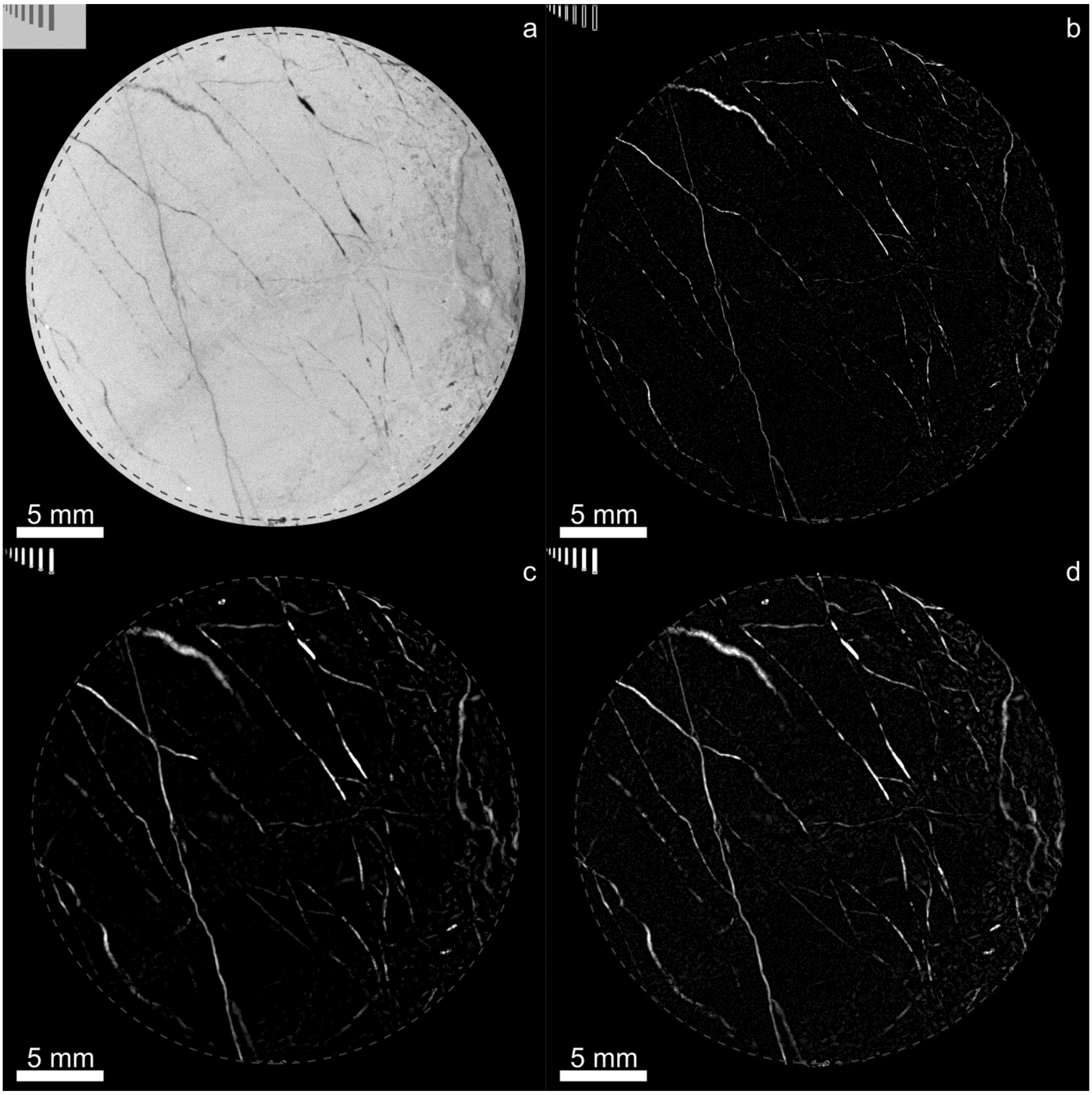


Figure D.3. Steps in the multiscale Hessian fracture filtering procedure. 2D slice representations from the 3D processing procedure. Same slice as displayed in Figure D.1, with the same final ROI indicated by the dashed circle. Contrast and brightness settings have been altered for display. (a) Preparation of the input data before the Hessian filtering. A ROI is defined, where the outside is set to NaN. Note this ROI is larger than the one indicated by the dashed circle, as the final ROI is shrunk to ensure a full 3D output. Control lines are added with a representative contrast for the fractures relative to the surroundings in the image data, for a range of scales. Here, control lines are added for scales 1 to 8, with a stepsize of 1. (b) Normalised results of the combination of the Hessian eigenvalues for fracture enhancement (Section D.2.3), for scale 2 (the lowest included in this analysis). Narrow planar features come out clearly, but noise is also apparent. (c) Similar to (b), but for scale 5 (the highest included in this analysis). In comparison to (b), the noise effect is lower, and fractures appear wider. (d) Combined scales 2 to 5, with a stepsize of 1. This image represents the output of the filtering procedure. In our example, the most important planar features stand out clearly, but (low-intensity) noise remains present.

Table D.2. Multiscale Hessian fracture filtering input parameters.

Name/symbol	Parameter	Description
s_{min}	Minimum Gaussian smoothing scale	Relates to minimum aperture to detect. Maximum response for apertures of 2x the scale.
s_{max}	Maximum Gaussian smoothing scale	Relates to maximum aperture to detect. Maximum response for apertures of 2x the scale.
s_{step}	Steps between Gaussian smoothing scales	Stepsize can be varied relative to the apertures in the data or to speed up the process.
blocksize	Blocksize in amount of slices	Amount of data to be analysed per consecutive step. An optimum can be defined.
*avgmat**	Average material greyscale	Greyscale value that characterises the material outside the fractures. Used for control lines.
*consthresh**	Conservative threshold of fractures	Greyscale value that estimates the greyscales present in the fractures. Used for control lines.
*maxmat**	Maximum material greyscale	Greyscale value that can be used to remove unusually bright spots in the dataset to prevent artefacts in the final output.
padding	Percentage of padding to be performed at start and end of input stack	First and last slices can be copied several times and included in the analysis. Extends the range of output slices but introduces "pseudo-3D" information, increasingly worse at higher percentages. Best kept at zero.

*Parameters that can remain "unspecified" by settings values corresponding to the maximum range of greyscales possible in the dataset (e.g. 0-255 for 8-bit, 0-65355 for 16-bit data). If specified, these values can remain approximate as minor variations do not influence the final result by much. Also note that because of the invariance of the Hessian matrix to the actual greyscale values (only the contrast is of importance), the main specification set by consthresh and avgmat is the average contrast of fractures to their surroundings. See user guide to the code (supplementary material) for more details.

D.2.4 Calibration, connectivity, and segmentation

Although the output of the described multiscale Hessian fracture filtering approach may in some cases be directly useable, we apply some additional postprocessing steps for our datasets. The descriptions hereafter are thus rather specific to our data type, and should therefore be seen more as a case study than previous sections. Calibration is performed to better visualise the data and to reduce noise. Additional noise reduction can be achieved by filtering throughout the dataset for connected clusters. Finally, binary segmentation is often required for data processing in a later stage, such as permeability modelling.

D.2.4.1 Calibration and including a conservative threshold

Calibrating the output of the multiscale Hessian fracture filter can be done in various ways, and is strongly dependent on the desired final output. The calibration we apply results in intensity values from 0 to 100, which can be seen as a fuzzy classification of porosity (e.g. 100 = 100% porous, 0 = 0% porous). For noise reduction and for emphasising high intensity values, we take a sigmoid curve based on the cosine function, with a lower and an upper cut-off value (see the online supplementary material for details). Because such a calibration requires additional user input, it is quite ambiguous, but this is very difficult to prevent in an analysis like the one presented here. Combined with information from different analyses (in our case for example porosity values determined from laboratory measurements or thin section analysis), a better calibration may be achievable. Different calibration curves may also prove more suitable in different analyses.

For the datasets used in our research, an additional step is taken together with the calibration. Besides narrow fractures, also some broader and blob-like porous areas are present in the data. Only the edges of these are detected by the multiscale Hessian fracture filtering. Their internal areas however have very low intensity values, and can easily be selected by a simple global threshold. We therefore select these areas from the original input data, often by using the conservative threshold already defined earlier (Table D.2), and include them with the calibration data as 100% porous areas. The combination of this and the calibration is shown in Figure D.4a. Whether such an addition of a conservative threshold is desired by the user is dependent on the input data.

D.2.4.2 Connectivity filtering

For the examples presented here, the effective porosity and the permeability are eventually of main interest. Because of this, we can use connectivity filtering to improve the representation of the data. In this method, we filter for connected porous clusters throughout the sample, in a certain direction. Included with the supplementary material is a MATLAB® script, which uses the *bwconncomp* function of the MATLAB® Image Processing Toolbox to determine and label clusters of connected data (26-connected neighbourhoods by default) in the dataset. By comparing the labels of the clusters present in the first and the last image slice, the labels of clusters of data that are connected between these two slices can be selected. By just keeping the clusters with these labels in the entire dataset, only the connected voxels are left, and retain their values from the calibration. All voxels in non-connected clusters are set to be non-porous (0). Generally, noise is effectively removed by this method, but depending on the assessed problem, it may have the undesirable effect of removing some planar features entirely. An example of the calibrated results after connectivity filtering is shown in Figure D.4b, as well as in the 3D visualisation in Figure D.5.

D.2.4.3 Binary segmentation

Completely binary segmented results may be desired depending on the problem being assessed. There is clearly no single answer on how to do this for every dataset (for comparisons of binarisation techniques, see for example Sezgin and Sankur (2004), Iassonov et al. (2009) and Porter and Wildenschild (2010)), and it is beyond the scope of this paper to go into detail on this. For our data, cross-calibrations with data from different methods may again be of importance here to decide on the best possible obtainable results. An example of binary segmentation on one of our datasets is shown in Figure D.4c.

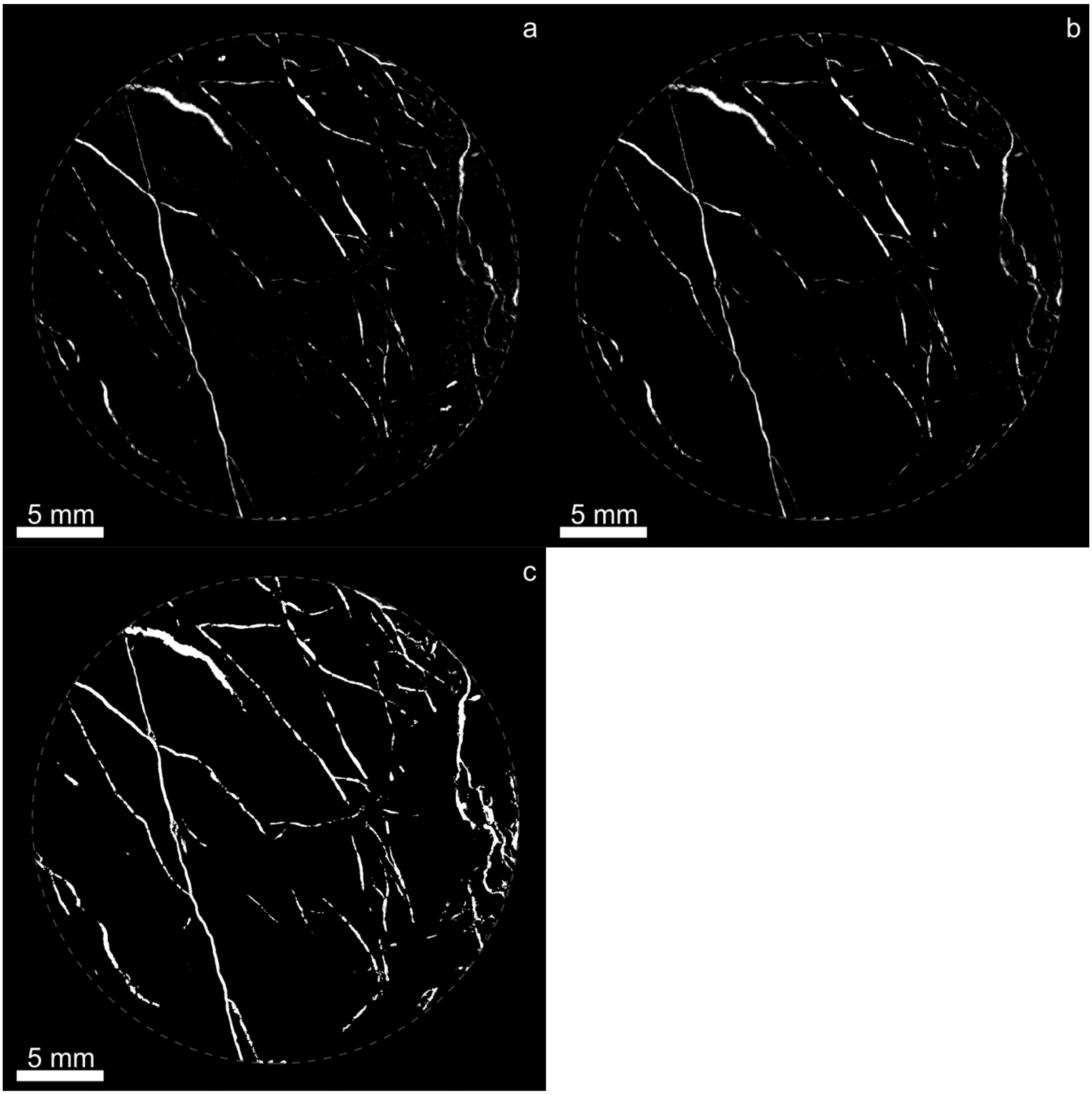


Figure D.4. Calibration and post-processing steps of the data after the multiscale Hessian fracture filtering procedure. The same data is shown as in Figures D.1 and D.3, and the same remarks apply. (a) Calibration of the data using a cosine function (sigmoid curve) between two user-selected values (see supplementary material for details). The calibration shows efficient noise reduction, as well as higher intensities for clear planar features. Additionally, the calibrated result has been combined with a conservative threshold on the original input data, to include larger, non-planar porous areas as well. Output values range from 0 (black) to 100 (white). This can be seen as a fuzzy approach to defining the porosity (for the mentioned output values, a range from 0% to 100% porosity). The performance in the darker area (visible in Figure D.1a) is good (few false positives). Some features appear rather broad in this 2D slice based representation (e.g. left top corner), due to their inclined nature (better visible in 3D). (b) Same dataset as in (a), but 3D filtered for connected porous clusters (intensities of 1 to 100 here considered as porous) between the first and the last slices of data, using MATLAB®. Compared to (a), some remaining noise and isolated pores are removed, as well as some (parts of) fractures that are apparently not connected to the main porous cluster, as resolvable at this resolution. (c) Binarisation attempt of the data shown in (b). The lowest possible threshold value of 1 is chosen here, explaining why some features may appear too broad. Of main importance is however the comparison to Figure D.1b, clearly demonstrating the better performance of the multiscale Hessian fracture filtering technique over such a simple, direct thresholding approach. For a true aperture analysis, a better calibration and threshold setting may be required, for example by using microscope analyses on thin sections as an input for calibration. In such a case, most likely a higher threshold will be chosen, reducing the apertures of the fractures to more realistic values.

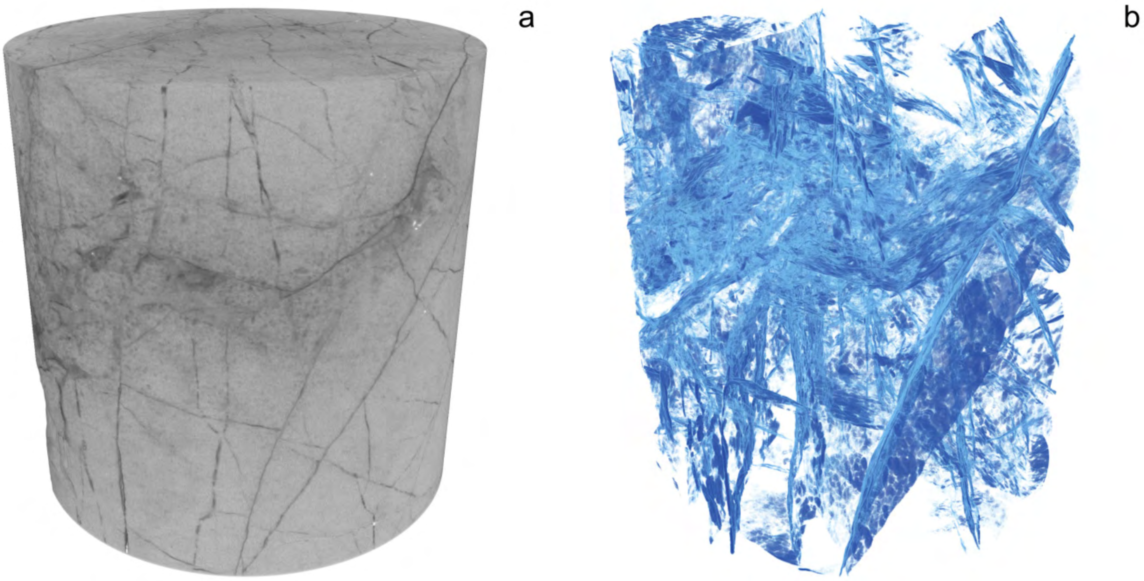


Figure D.5. 2D outtake of a 3D visualisation of input data and the corresponding multiscale Hessian fracture filtering results, created with the visualisation software Amira®. A motion clip, in colour, can be found with the online supplementary material. Same dataset as in Figure D.1, D.3 and D.4. (a) Surface rendering of the dataset, with the final ROI applied. This shows the surface representation of the fractures, as well as material variations throughout the dataset. (b) Volume rendering of the fractures in the dataset, in the same orientation as (a). The same steps as for Figure D.4b have been taken (Hessian filtering, calibration and connectivity filtering). The chosen intensities and opacity are mainly for visualisation, but generally speaking, darker areas are more porous. Some fractures are very clear, whereas in other areas the visualisation is less clear. This is usually due to the presence of a lot of closely spaced and crosscutting fractures, other pores, and/or some remaining noise. Gaps in fracture planes mostly represent partial closure of the fracture (or: an aperture below the detection limit for the Hessian filtering), which can be observed in the original raw data as well.

D.3. Discussion

D.3.1 Advantages and drawbacks

The multiscale Hessian fracture filtering technique described in this paper has some very clear advantages for emphasising and eventually extracting narrow fractures in 3D datasets. When comparing Figure D.4b to Figure D.1b, the advantage over global thresholding is evident. The 3D implementation of the Hessian approach ensures that fractures in all orientations can be detected. This is a clear advantage over 2D methods, especially for fractures that are near-parallel to the slicing direction in slice based datasets.

The full approach is implemented in public domain software and can be run easily, with or without GUI. Some user input is required at the start of the filtering process, which then runs automatically without interruption. Serial processing of the data ensures that even on desktop systems with limited resources, the filtering technique works. The fastest processing time achieved (with optimum settings) for a 1670 x 1670 x 1546 voxels dataset on a desktop with an 8-core Intel Core i7 2600 @ 3.4 GHz processor and 16 GB RAM memory is slightly over 10 hours. Note that in fact only one of the processing cores was used. The Hessian matrix was calculated on 6 Gaussian scales, from 2 to 7, and 134 GB of HDD space was required. The same analysis takes over 31 hours on a desktop with a 2-core (one core used) Intel Core2Duo E8400 @ 3.0 GHz processor and 4 GB RAM, but this time reduces to a more reasonable 14 hours when only taking Gaussian scales 2-5. This displays the large influence of both RAM memory and chosen Gaussian scales. First, the available RAM memory limits the size of the data blocks that can be processed. Second, higher scales take longer to calculate (as then the Gaussian smoothing at every voxel has to account for more surrounding data), and require more overlap between consecutively analysed data blocks. This overlap yields higher memory inefficiency, so more blocks need to be processed eventually. Naturally, the calibration and connectivity filtering steps require some additional processing time, but are separated from the main Hessian analysis.

The described approach requires some user input, depending on the desired outcome and planned future analyses on the dataset. In cases where only a qualitative enhancement of narrow fractures in a dataset is required, the amount of required human input is limited. In this case, next to a user defined ROI, only the Gaussian scales to be analysed need to be chosen. Important to notice is that these scales can influence the filtering results negatively. Including a too low scale can cause noise artefacts to become abundant. A too high scale can lead to an unrealistically broad representation of the fractures in a dataset. An example can be seen in Figure D.2e. Although the curves displayed in the figure do not represent the full output of the multiscale Hessian fracture filter, it is clear the output of the Gaussian convolution results in a curve broader than the input feature. On the other hand, if not high enough scales are included, very broad features might be outputted as two parallel features (for example, Figure D.2f). In the latter case, this usually will be compensated by including a conservative threshold in the calibration step.

Next to the scales, additional user input may be required. For a more reasonable normalisation of the results to the actual contrasts in a dataset, several greyscale values (*consthresh, avgmat* and *maxmat* in Table D.2) should be chosen. More user input ambiguity is of course also included by the final calibration and binarisation steps, as well as when turning on data padding (*padding* in Table D.2). Although it is always better to reduce human input in image processing to a minimum, there are very few approaches available in literature by which this can be achieved, and especially not for narrow fractures. The described vesselness and sheetness approaches (Frangi et al., 1998; Descoteaux et al., 2005) for example also do not perform better in this sense, as they require sensitivity parameters that control the output to be set by the user.

From Figures D.3 and D.4 it is clear most fractures are detected by our method. An emphasis should however be put on some fractures that are not displayed in the results. The main reason for faint fractures not to be present in the output is noise. For (synthetic) input data without any noise, even the faintest features are detected. As in the calibration no noise reduction is required, these faint features will remain visible. However, for real data, this is often not the case, as it is not possible to retain these features without also retaining a lot of noise. Of course, features can also be so thin, that no Gaussian kernel is appropriate to detect them, without again including a lot of additional noise. Because of the above, it is not possible to pinpoint the exact resolution limits of the presented method, as there is a strong correlation to the quality of the input images.

Another point regarding the fracture detection is that some narrow fractures show up rather broad in the final result. This is especially clear in data thresholded using the lowest possible threshold setting, as in Figure D.4c. This broadening of aperture is of course introduced by having to include Gaussian scales too broad for some features, to successfully detect the truly broader features. This effect is inevitable, but better results can be obtained for better calibrations and threshold settings. For example, a comparison to microscope analyses on thin sections can be very useful in this aspect, to set a better threshold before performing an aperture analysis. Another approach would be to calibrate using data of a fracture with a known and controlled aperture (such an approach has for example been applied in Ketcham et al., 2010). For the example Figure D.4c, this would mean taking a much higher threshold value, to decrease the apertures to more realistic values.

As a final point an important disadvantage, inherent to our Hessian approach, should be mentioned here. Fracture intersections may cause problems in the analysis. The intersection of two fractures is not a planar feature, but tube-like, with the structure being the most tubular when two orthogonal planes intersect. In synthetic data, such an intersection has a value of zero in the results of the Hessian filtering (see Figure D.6). The effect decreases with lower intersection angles of two planes, i.e. the intersection becomes more planar in nature and gives a response higher than 0 in the Hessian fracture filtering results (also see Figure D.6). In our datasets, intersections are often characterised by such lower angles. Additionally, the contrast to the surroundings at intersections is often higher than between single fracture and surroundings, which may increase the response in the Hessian filtering result further. If it contains low enough greyscale values, the intersection may also even be output as porous when the low global threshold on the input data is combined with the calibration results (Section D.2.4.1). Therefore, in our data, the effect of the intersection problem does not result in severe artefacts. It is however important to notice the effect when attempting the multiscale Hessian fracture filter on a different type of data. Also note that similar problems as for intersections may occur at fractures that terminate suddenly, as the tip of a fracture is also not a planar feature. Additionally, very closely spaced fractures may interfere in the Hessian filtering step.

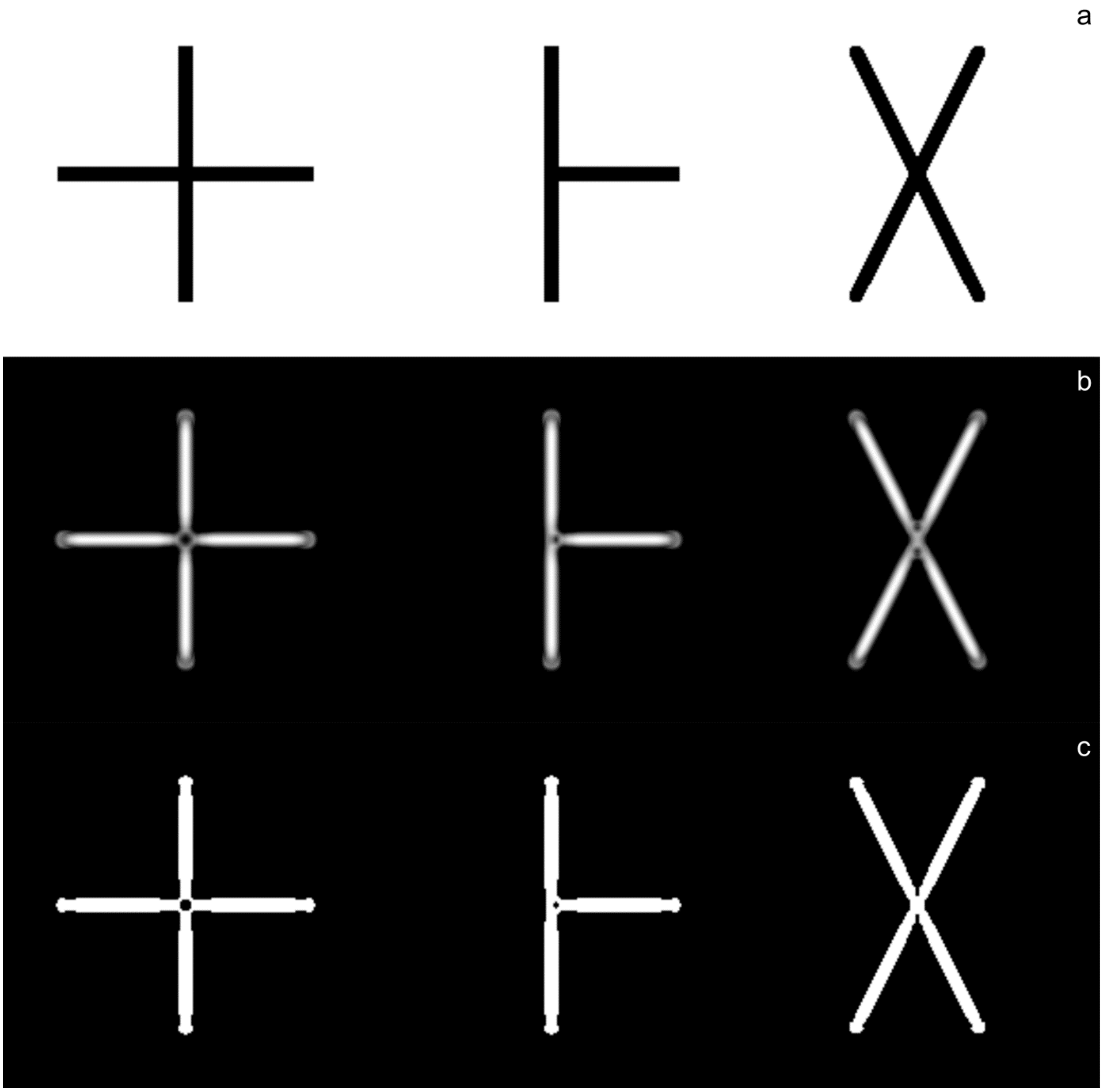


Figure D.6. Results of the multiscale Hessian fracture filtering for synthetic data with intersections and line terminations. Contrast and brightness settings have been altered for display. (a) Input data, the width of the features is originally 8 pixels. (b) Multiscale Hessian fracture filtering output (combination of Gaussian scales 2, 3 and 4). (c) Threshold of (b). It is clear from these figures, line terminations and intersections can pose problems for the filtering, as they do not represent planar data. Line terminations are represented in a bulging way, and line intersections become rounded with little response at the intersection core. The worst results for the intersections are shown for 90 degree angle full intersections (left), whereas the effects become less for terminating intersections (middle) and lower angle intersections (right). These examples can be seen as worst-case scenarios for synthetic data. See the main text for discussion why the influence of these intersections is often less problematic in real data.

D.3.2 Other considerations and limitations

There are some other considerations before applying the presented filtering technique that cannot clearly be termed advantageous or disadvantageous.

It is important to notice the filtering works best on fractures cutting through an otherwise homogenous material. For example for µCT data, if the base material is very heterogeneous, with different phases showing up in a different intensity, artefacts may occur. This is because the Hessian filtering is sensitive to contrast, so also the boundary between two different phases with a high contrast is detected as a planar feature. Of course, the severity of such artefacts depends on the dataset (i.e. the contrast between different phases and the way the fractures appear), and in some cases a suitable preprocessing or calibration step afterwards may reduce the problems.

An important characteristic required for input data is spatial isotropy. The presented filtering technique does not support spatially anisotropic data. Every voxel should therefore be cubic, and the voxel dimensions should remain constant throughout a dataset. Although FeatureJ supports anisotropy, we choose not to support this in our approach, as this severely complicates or even prohibits the usage of the control lines. Proper assessment of the input data and proper conversions, if necessary, are thus important before starting the filtering. Other considerations (including data types, possible downsampling and the choice of the ROI) are explained in the supplementary material.

D.3.3 Orientation information from the Hessian matrix

This paper focusses on the enhancement of fractures or other planar features, using intensities only, by using the Hessian matrix. Hence, only the information on magnitudes (the eigenvalues) has been used so far. We do want to point out however that also orientations can be extracted from the Hessian matrix, by using its eigenvectors. One can take the eigenvector orientation of the largest magnitude eigenvalue, for every voxel that is considered to be part of a planar feature by the filtering process. This eigenvector represents the normal direction to these planar features. These orientations can then for example be displayed and analysed using a stereographic projection. Tests where this technique for orientation analysis was used work well, but as FeatureJ does not natively output the full Hessian matrix, and additional processing is required, it is beyond the scope of this paper to elaborate on the orientation aspect here.

D.4. Conclusions

Narrow fractures in 3D computed tomography data can be problematic to extract, whereas this extraction is often necessary for future processing of the data. In this paper the multiscale Hessian fracture filtering approach is presented, providing an efficient way of extracting such narrow fractures from a dataset. Both the fuzzy output of the filtering approach, suitable for visualisation and porosity determination, and the fully binarised output display the good performance of this technique. This is especially apparent when comparing to direct global threshold binarisation on a dataset, but also for other non-structure based segmentation techniques. The multiscale Hessian fracture filtering approach and the sequential postprocessing steps produce a result with little noise, in which even very narrow fractures are clearly emphasised. Furthermore, the multiscale Hessian fracture filtering technique is not hampered by most greyscale variations throughout datasets. The technique has some drawbacks, mainly inherent to using the Hessian matrix. The most important are the requirement of user input, and possible problems at fracture intersections and terminations. In our data examples, these effects however do not seem to introduce large artefacts.

The multiscale Hessian fracture filtering technique is implemented in the public domain software ImageJ, and provided as supplementary material to this paper. The way of implementation ensures the data is processed in 3D, without user intervention after the initial setup, and without requiring extensive computational power (i.e. possible on a modern desktop PC). These properties are a clear advantage of the approach compared to other approaches in literature. The application of the technique has been shown on our µCT data of narrow fractures in reservoir dolomites. Besides the usage on these datasets, the general implementation of the technique provides a great potential for its usage on different 3D datasets, even when very different goals might be aspired.

Acknowledgements

The authors would like to thank the OMV for the funding of the research presented in this paper. Additional thanks goes to Christian Gusenbauer (University of Applied Sciences, Wels, Austria) for performing the µCT scans and reconstructions. The reviews of this paper, especially by Richard Ketcham, were very helpful to improve the message presented here. Editor-in-Chief Jef Caers and the other editors of Computers and Geosciences are thanked for their work for getting this paper published. Finally, the authors would like to acknowledge the developers of the used software FeatureJ (Erik Meijering), ImageJ (Wayne Rasband) and FIJI, for making the efforts of their work publicly available.

Supporting information

Supplementary information associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cageo.2013.03.006. The MHSFF code can be found in Appendices X1 and X2 and is also available at https://github.com/cageo/Voorn-2013.

E. Additional µCT data processing and analysis

E.1. Addendum and improvements to the MSHFF

E.1.1. Narrow feature enhancement and thresholding

In Chapter C it was already discussed that a full (binary) segmentation of the fractured Hauptdolomit data is required. Segmented data are, relative to fuzzy or other data representations, more useful for porosity determinations and further data analysis techniques (for example for permeability modelling). Regarding this, the previously discussed implementation of the multiscale Hessian fracture filter (Chapter D) has some drawbacks. Although the fuzzy results are quite reasonable (e.g. Figure D.4a, D.4b, D.5b), the finally segmented data (e.g. Figure D.4c) display often too broad features. This is because the final threshold needs to be set relatively low, which has the advantage of including quite narrow (faintly displayed) fractures, but the disadvantage of including larger ranges around broader (clearer displayed) fractures.

A solution to the above problem was realised after the publication of the paper and will be described in the following section. Before combining the different analysis scales with each other, the different normalised scales (Section D.2.3) are multiplied by a factor, depending on the Gaussian scale:

$$B_s = \frac{\sqrt{2}}{\sqrt{s}} \cdot \hat{A}_s \qquad \text{(Equation E.1)}$$

B_s is now the new image, s represents the Gaussian scale, and $\hat{A}_s$ is the normalised result obtained earlier. The combination of the different scales now comes down to:

$$B = \max(B_s), \text{ for } s_{min} \leq s \leq s_{max} \qquad \text{(Equation E.2)}$$

This approach has an empirical basis: in Figure D.2, one can see that a feature with half the intensity also has 50% of the full response (Figure D.2d). However, a feature with half the width (=2x too broad Gaussian scale) has around 71% of the full response (Figure D.2e). The correction explained above sets the response for this twice too broad Gaussian scale to approximately 50%. By doing this, also the influence of larger Gaussian scales for narrow features is reduced. Smaller Gaussian scales keep influencing the response for broader features, but this is of course no problem for the final result. An example, parallel to Figure D.2, can be seen in Figure E.1.1. Note that while this correction enhances the results for narrow features, it decreases the response for broad features in synthetic data. However, as we can observe in our real data, this effect is usually counteracted by the broader features having a higher contrast to the surroundings to start with. Therefore, the correction does not lead to new erroneous results.

The solution discussed has the additional advantage of not having to perform a calibration step (Section D.2.4.1) anymore, so binary segmentation is directly possible. Selecting a proper threshold value however remains difficult. The approach taken for the results shown in this study requires the threshold to be set manually. The choice made here is to set the threshold in such a way that noise levels in the result are still tolerable, while including most features visible in the input data. Unfortunately, this is a source of ambiguity, and depends greatly on the operator. One should therefore keep in mind that the thresholded results are subject to an error margin, as emphasised in Figure E.1.2 (and earlier in Chapter C).

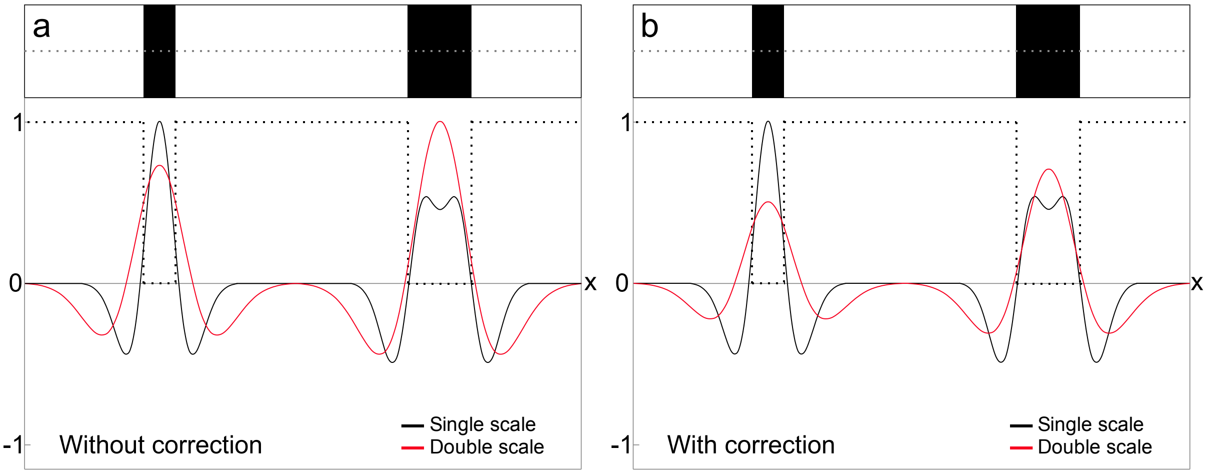


Figure E.1.1. Example results of the convolution to generate one element of the Hessian matrix, in the original (a, using Equation D.2) and the corrected way (b, using Equations D.2 and E.1). Also see Figure D.2 for more information. Figure (a) shows the original processing result. The black curve shows the response of the convolution for a Gaussian scale giving the highest response for the narrow black feature. The red curve shows the response of the convolution for a Gaussian scale that is twice that broad, and gives the highest response for the broad feature. When these analyses would be combined, the narrow feature could get a too broad representation because of the red curve. Figure (b) shows the correction shown above. The black curve retains its position (for display purposes). The red curve is reduced. Clearly, the red curve now influences the response for the narrow feature less than in Figure (a).

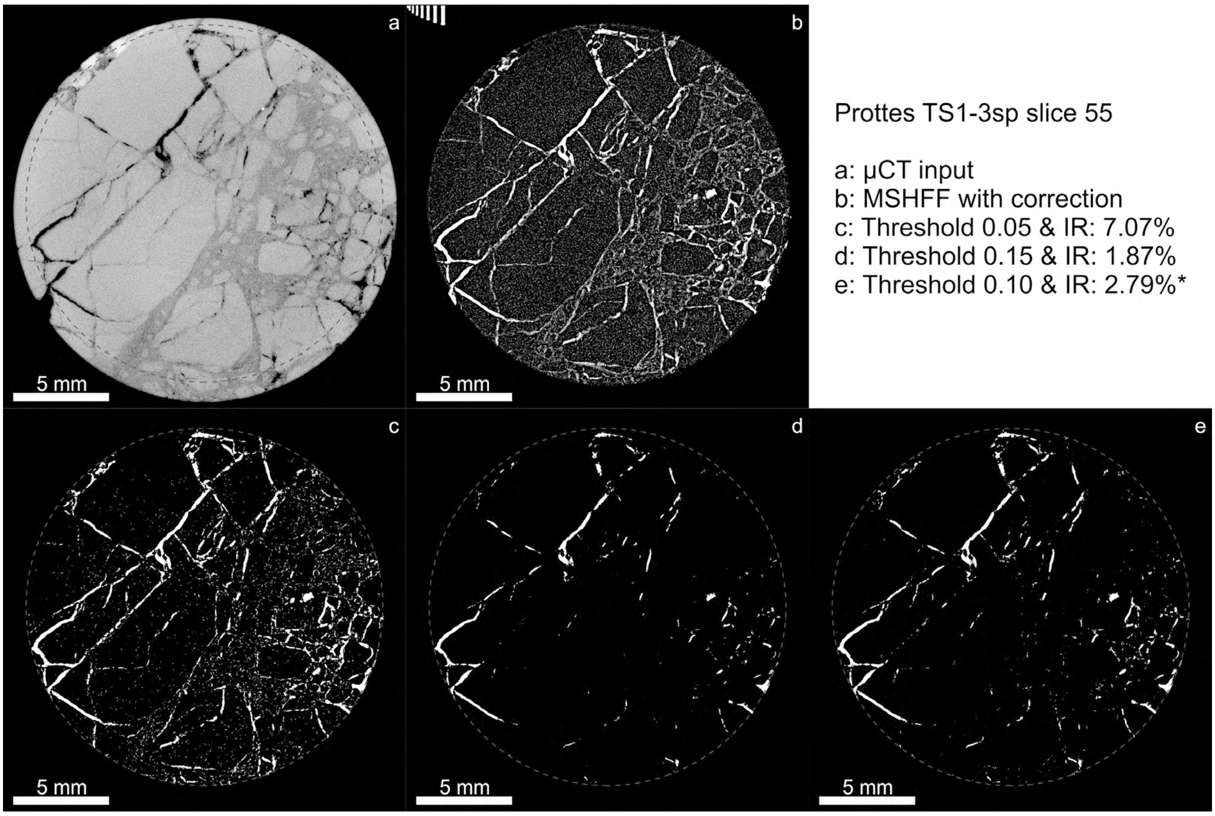

a
b
Prottes TS1-3sp slice 55
a: µCT input
b: MSHFF with correction
c: Threshold 0.05 & IR: 7.07%
d: Threshold 0.15 & IR: 1.87%
e: Threshold 0.10 & IR: 2.79%*
c
d
e
5 mm
5 mm
5 mm
5 mm
5 mm

Figure E.1.2. Example of threshold selection on µCT data. An example is shown from dataset Prottes TS1-3sp, reconstructed at (12.5 µm)^3/voxel. The descriptions of the separate images are noted in the figure. IR means Island Removal, and the noted percentages would be the porosity as displayed for the slices in c, d and e. A very low threshold of 0.05 (Figure c) leads to oversegmentation and an unrealistically high porosity value for this sample. A value of 0.015 (Figure d) is better, but misses some of the features. The threshold chosen for this particular sample is 0.10, shown in Figure e and indicated with a star.

E.1.2. Island removal

After thresholding of the MSHFF filtered data, some noise still affects the end result. In order to remove this noise, connectivity filtering as described in Section D.3.2.3 could be applied. However, it has been noted in combination with thin section analysis that such connectivity filtering can remove porous clusters that are in fact connected, but only below the µCT resolution. It is therefore better for the final goal of this data (i.e. application in hydrocarbon research) not to remove such possibly-connected clusters in an early stage of data processing. To be able to still remove most of the remaining noise in the data, an island filtering procedure is applied instead of connectivity filtering. Island filtering is achieved by making a simple alteration to the connectivity MATLAB® script described in Section D.3.2.3. Using the *bwconncomp* module, clusters of connected voxels (of one segmented group) are labelled. Subsequently all clusters that contain less connected voxels than a certain threshold value are removed. The choice for this threshold is of course arbitrary, but in our datasets most noise was removed efficiently by using a threshold of 20. In this way, all clusters containing less than 20 voxels are removed from the segmented dataset. Some examples of the segmented results after island removal have already been shown in Figure E.1.2.

In the island filtering process, it is important to define the connectivity type. In 3D, there are 3 main options (also see Figure E.1.3):

1. 6-connected (also termed 7-connected) neighbourhood: for every voxel, only the voxels that touch faces are considered to be connected, when they belong to the same segmented group (i.e. are white on a black background).
2. 18-connected (also termed 19-connected) neighbourhood: for every voxel, only the voxels touching faces and edges are considered to be connected, when they belong to the same segmented group.
3. 26-connected (also termed 27-connected) neighbourhood: for every voxel, all touching voxels (faces, edges and corners) are considered to be connected, when they belong to the same segmented group.

In the MATLAB® scripts for connectivity and island filtering, we choose to apply the most commonly used 26-connectivity.

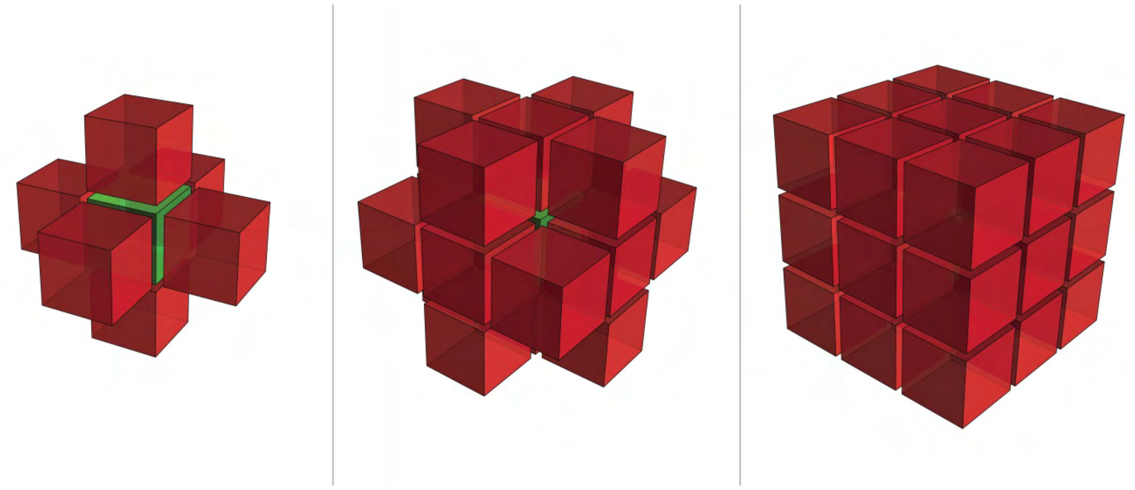


Figure E.1.3. Comparison of the different pixel connectivity possibilities (from left to right: 6-connected, 18-connected and 26-connected). From: Rossi, 2010.

E.1.3. Summary of segmentation sequence

- Check input data for strange features (e.g. scattering by high density phases or reconstruction errors), possible delete part of the range if the data quality is insufficient.
- Apply multiscale Hessian fracture filter (MSHFF), using the most applicable parameters defined in the input data (average contrast between features and background, expected range of width of features, exclusion of bright spots, et cetera).
- Combine different Hessian scales using the square root correction, emphasising the narrower fractures.
- Segment the combined Hessian result by setting a threshold, including the features of interest but excluding most of the noise.
- Apply an island removal step on the segmented data, to delete most noise left after previous step.
- Check if the sample volume is properly defined. Regions without any input data (for example a chipped off part of a plug) that are included in the segmented result should be removed.

E.2. Additional MSHFF performance tests

In Sections D.2 and D.3.1, some performance tests on the multiscale Hessian fracture filter have already been shown. Naturally, more orientation tests like these – including tests to verify the same behaviour on planar features in x, y and z directions – have been carried out. In these tests, the features to be detected by the Hessian filter were always clear lines with a distinct, single-valued contrast to the surroundings. For completeness, some tests on a more realistic image situation (compared to real CT data) are shown here. For this, test figures such as Figure D.6 are taken and distorted by various degrees of salt-and-pepper and (Gaussian) noise using ImageJ (note this is done in 3D). These images are consecutively processed by the standard MSHFF routine as applied on the µCT images. Two extreme examples are shown below. First, Figure E.2.1 shows a situation with some noise on the input image (Figure E.2.1a), in which the feature of interest is still clearly visible. Already in the raw MSHFF output (Figure E.2.1b) the feature of interest is clearly recognisable. After segmentation (Figure E.2.1c), the feature is recovered with a similar quality as shown in Figure D.6. Second, Figure E.2.2 shows an extreme case in which in the input image (Figure E.2.2a) the feature of interest is barely visible. In the raw MSHFF output (Figure E.2.2b), the traces of the feature of interest are however quite recognisable. Segmentation of such an image does however not provide a suitable result (Figure E.2.2c). This example is therefore over the limits of the MSHFF routine, but the natural fractures showing up in µCT scans generally have a higher contrast to their surroundings than this synthetic data. In this specific example (Figure E.2.2a), one could argue the feature of interest might not even be spotted in a completely manual segmentation routine. In conclusion, planar features should be recognised quite efficiently by the MSHFF, even up to rather high levels of noise.

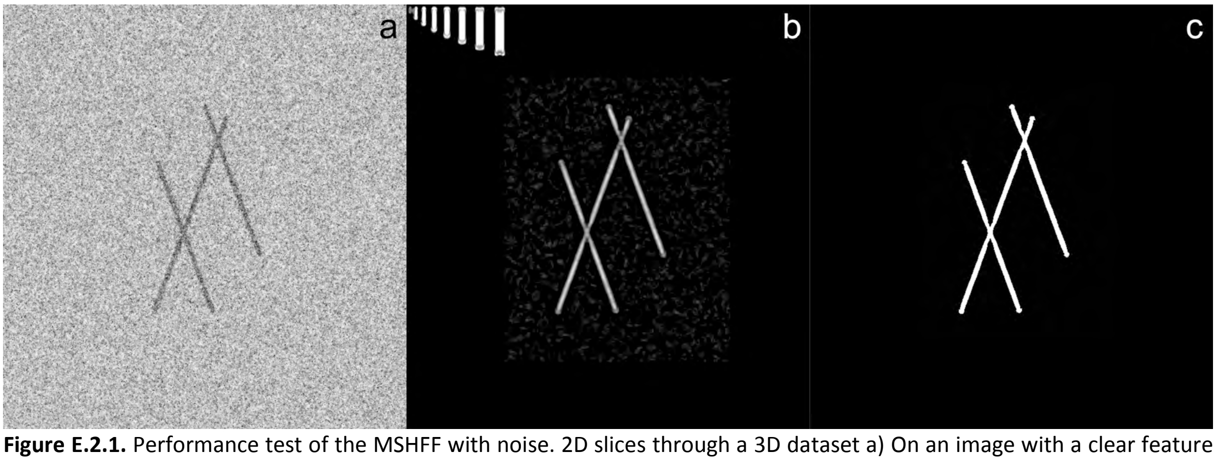


Figure E.2.1. Performance test of the MSHFF with noise. 2D slices through a 3D dataset a) On an image with a clear feature of interest noise is added. b) Raw MSHFF output (brightness and contrast adjusted for visualisation). c) Segmentation result. No island filtering has been applied on this sample.

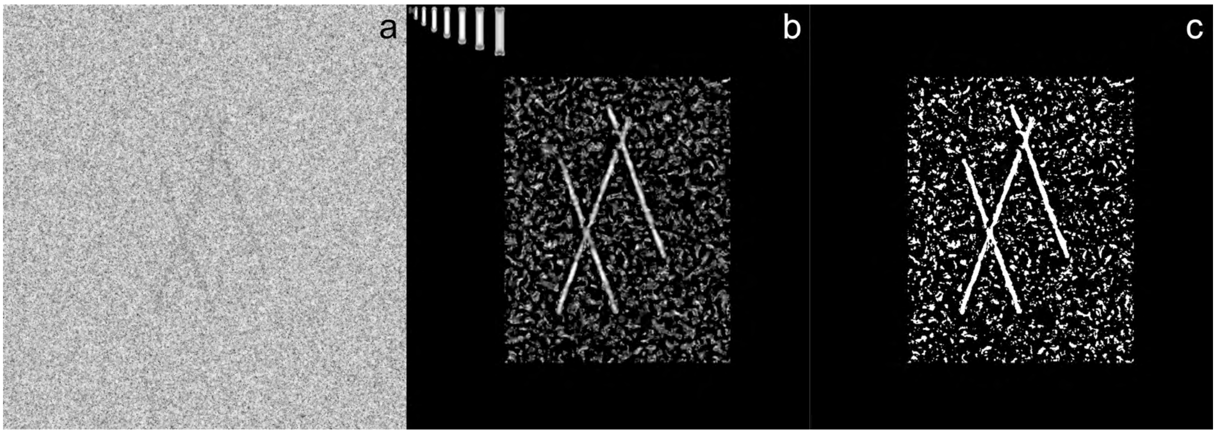


Figure E.2.2. Performance test of the MSHFF with noise. 2D slices through a 3D dataset. Similar to Figure E.2.1, but with a much more extreme noise pattern a) Input image of feature of interest with noise b) Raw MSHFF output (brightness and contrast adjusted for visualisation). The feature of interest becomes better visible. c) Segmentation result. The example shown here is without additional island filtering, but also applying this does not lead to the desired result. This example thus shows the upper noise limit of the MSHFF routine.

E.3. µCT data analysis

After the full segmentation routine by the multiscale Hessian fracture filter is completed, the segmentation results can be used for various analyses. The input data and segmented results are shown in Figure E.3.1 to E.3.7 for the main samples of this study (see Chapter F for details on the samples). The easiest to determine is the porosity, as this can be readily calculated from the amount of voxels accounted to solid and background. Different parameters such as fracture aperture, fracture density and fracture orientation require more extended routines. Details on the methodology for determining such parameters and results on fractured dolomite samples will be discussed in Chapter F. Also permeability can be determined by modelling approaches. Although it is not the scope of this thesis, some remarks and attempts on permeability modelling on Hauptdolomit samples will be covered in Chapters F and H.

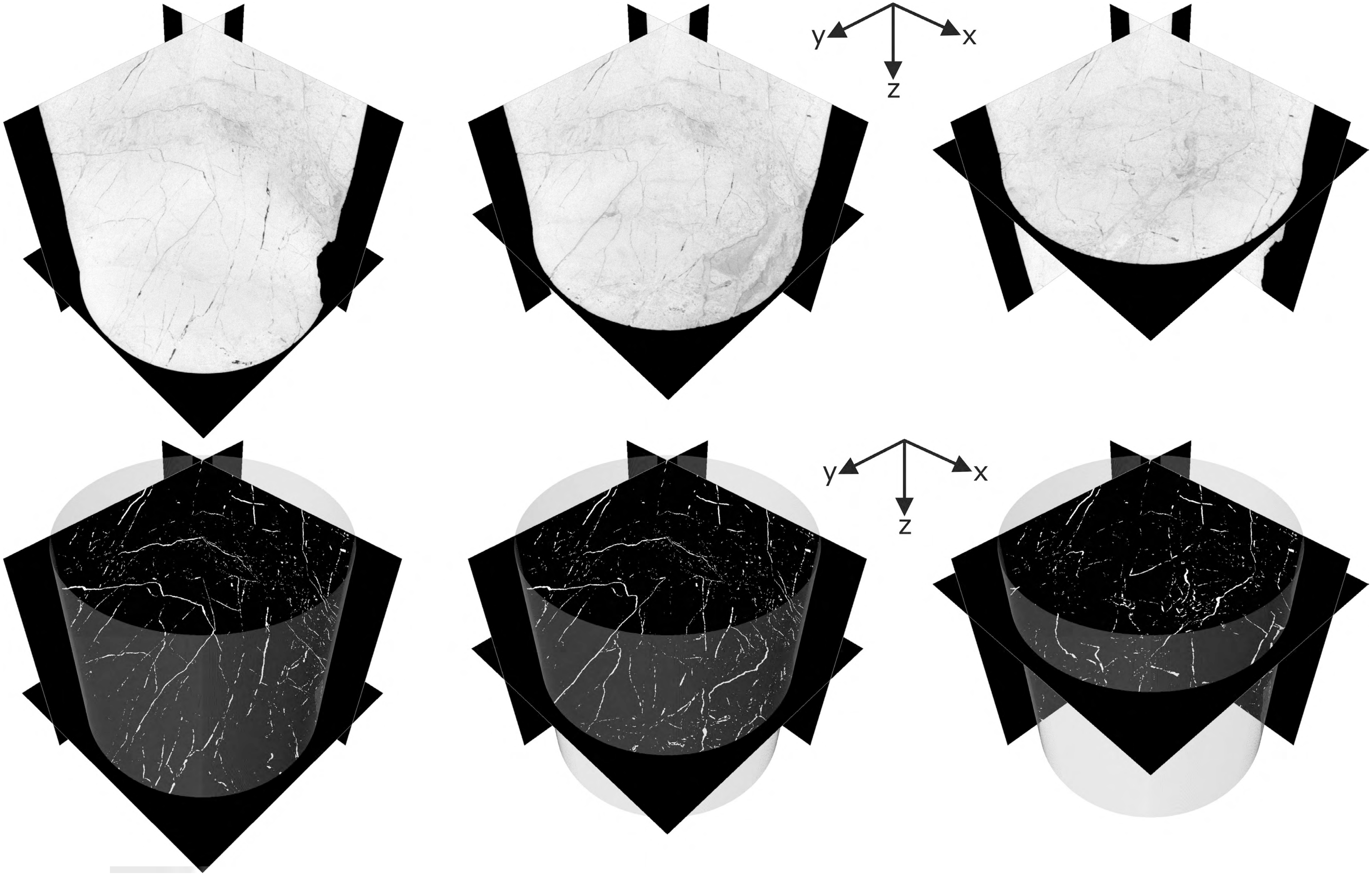


Figure E.3.1. µCT data and segmentation results for Ø 3 cm sample Prottes TS1 at (18.9 µm)^3 / voxel resolution.

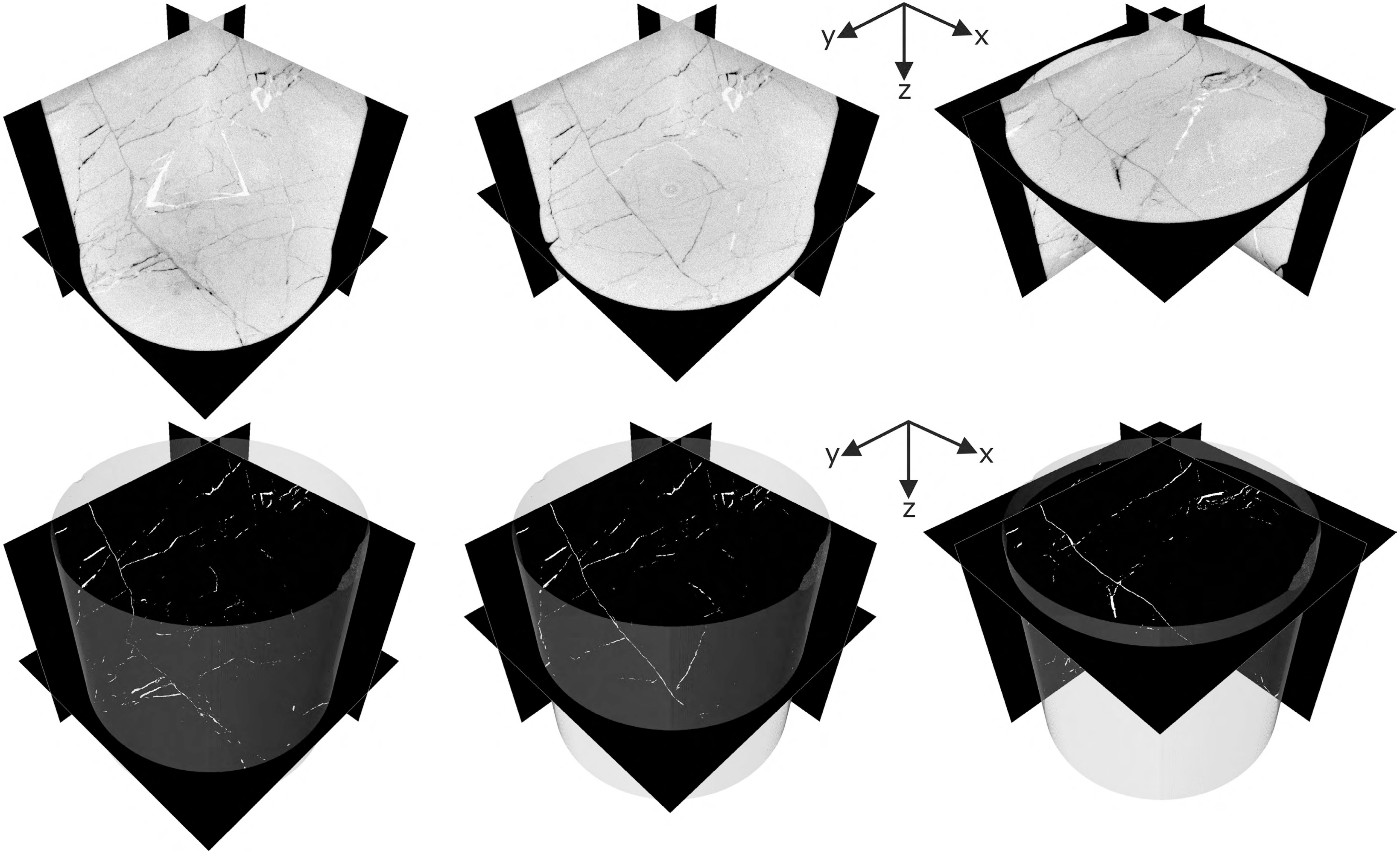


Figure E.3.2. µCT data and segmentation results for Ø 2 cm sample Prottes TS1-8sp at $(12.5\ \mu m)^3$ / voxel resolution.

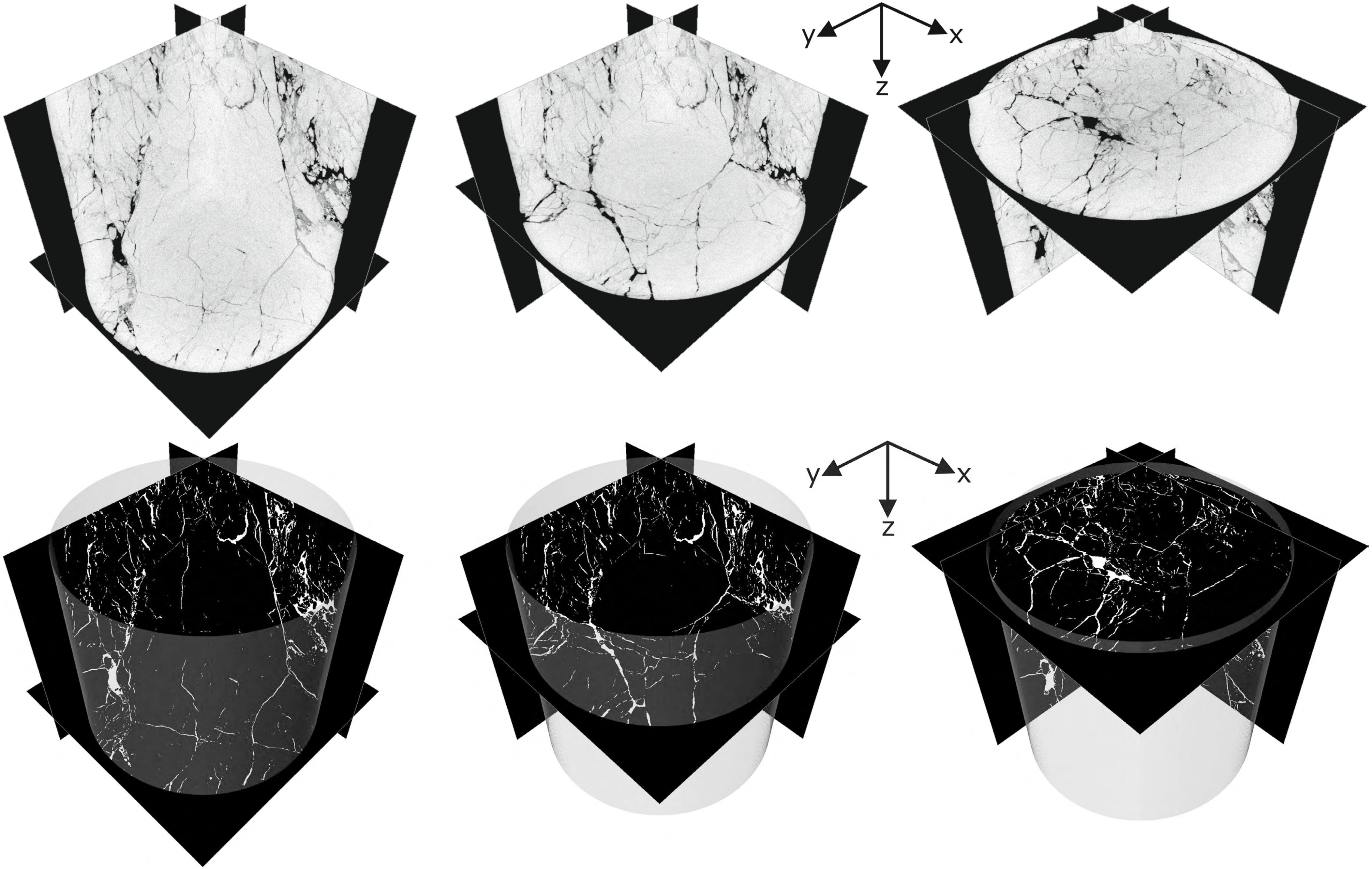


Figure E.3.3. µCT data and segmentation results for Ø 3 cm sample Schönkirchen T91-5p at $(19.0\ \mu m)^3$ / voxel resolution.

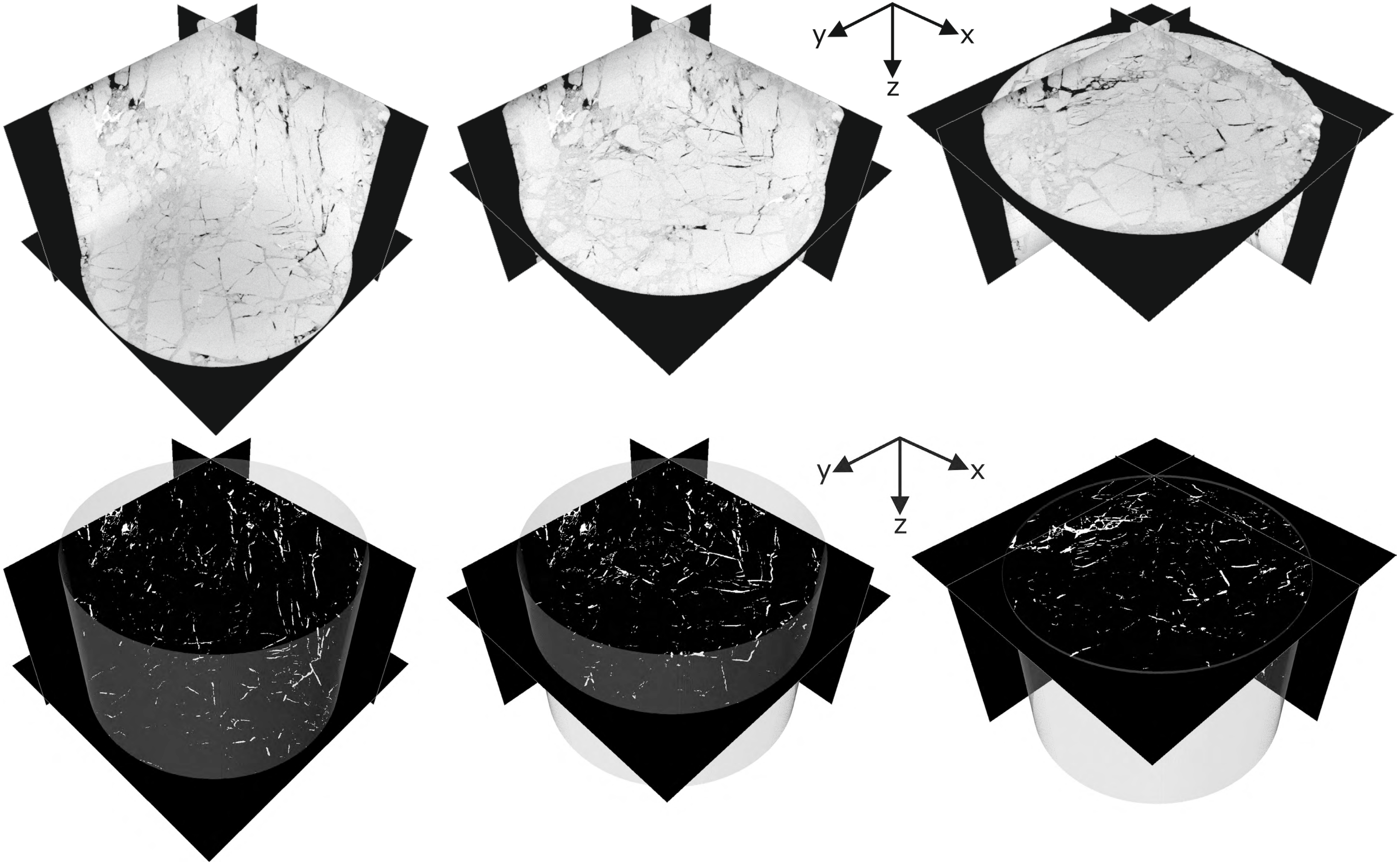


Figure E.3.4. µCT data and segmentation results for Ø 3 cm sample Prottes TS1-4p at $(19.0\ \mu m)^3$ / voxel resolution.

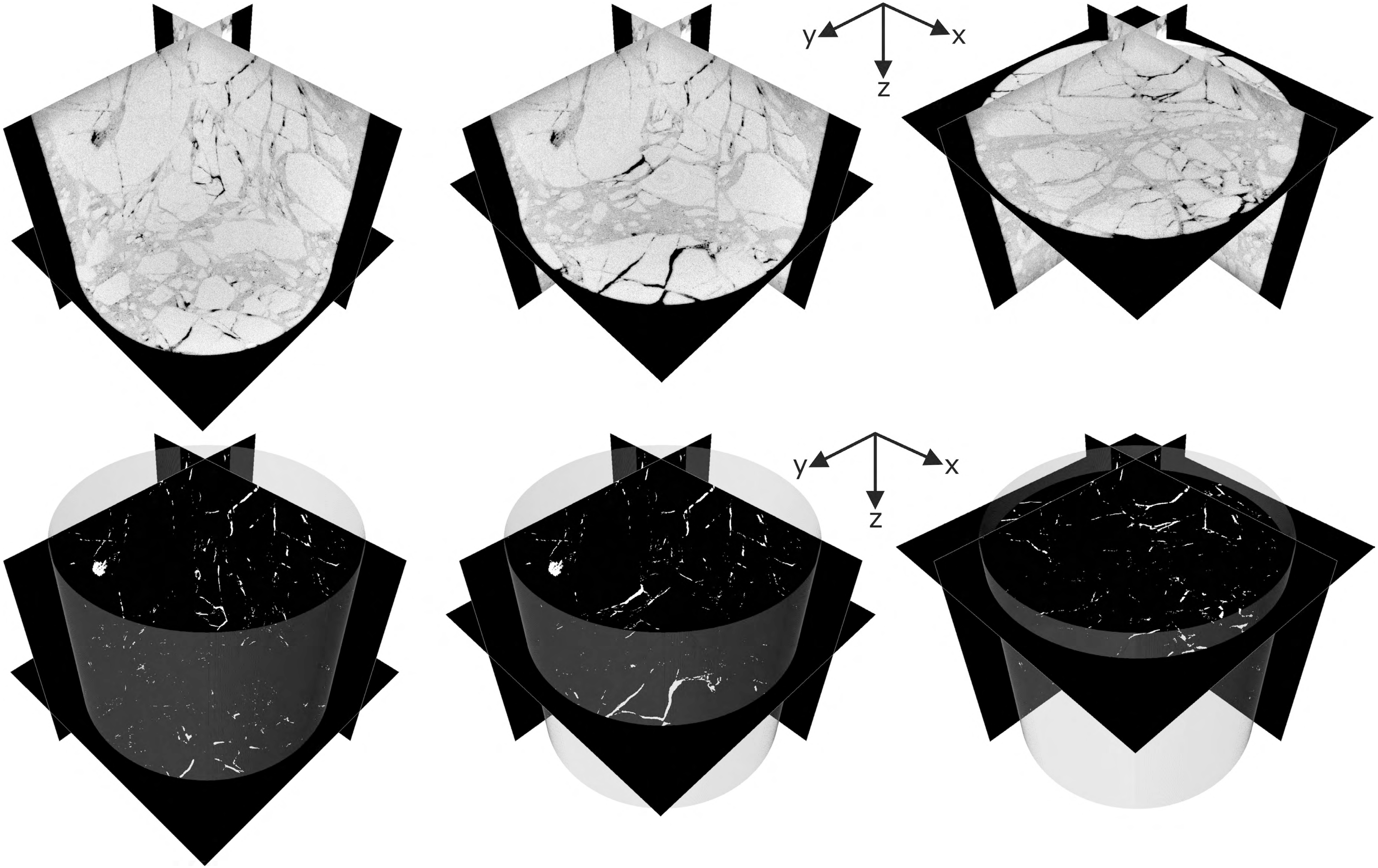


Figure E.3.5. µCT data and segmentation results for Ø 2 cm sample Prottes TS1-3sp at (12.5 µm)^3 / voxel resolution.

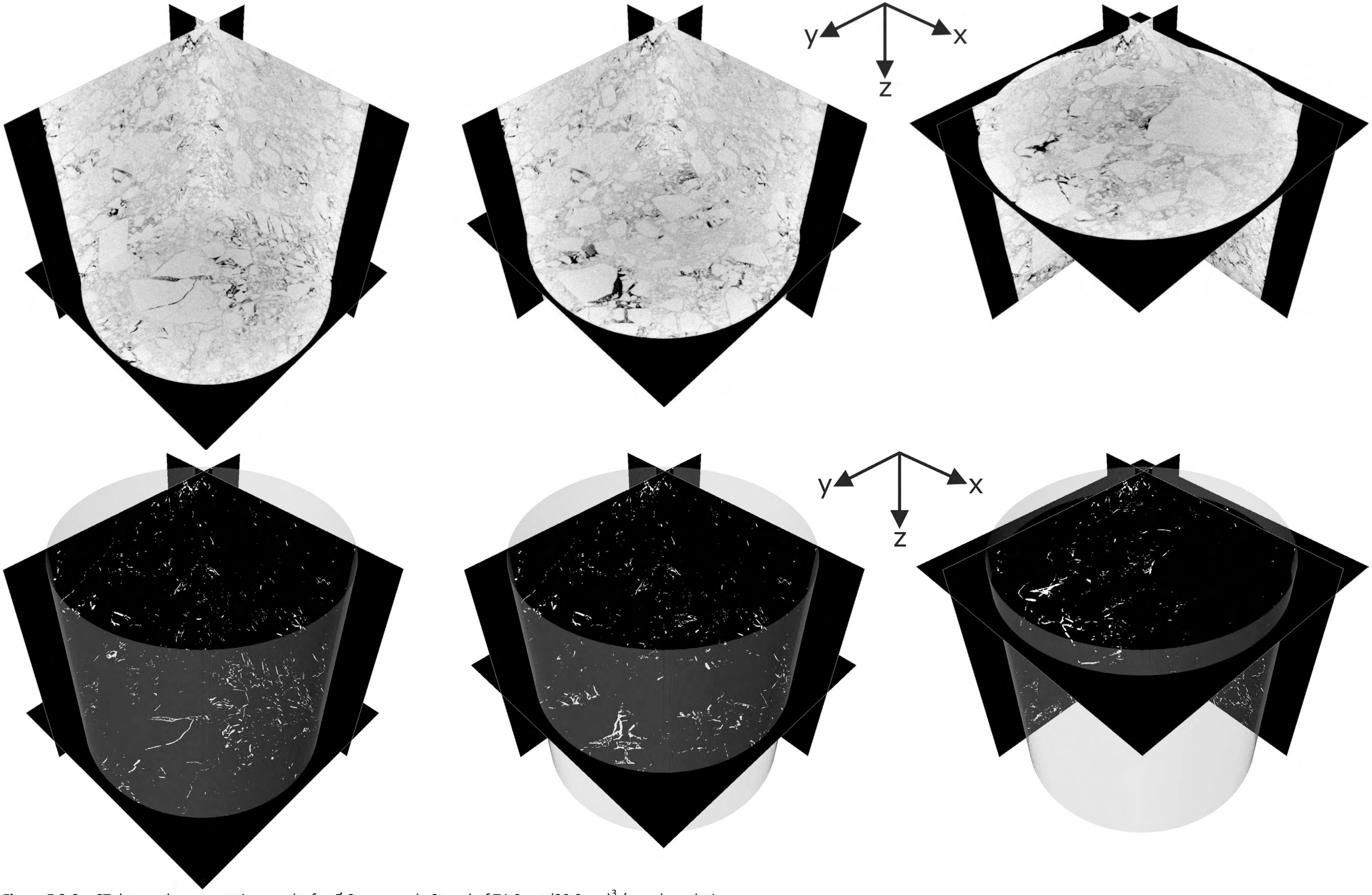


Figure E.3.6. µCT data and segmentation results for Ø 3 cm sample Strasshof T4-3p at $(20.0\ \mu m)^3$ / voxel resolution.

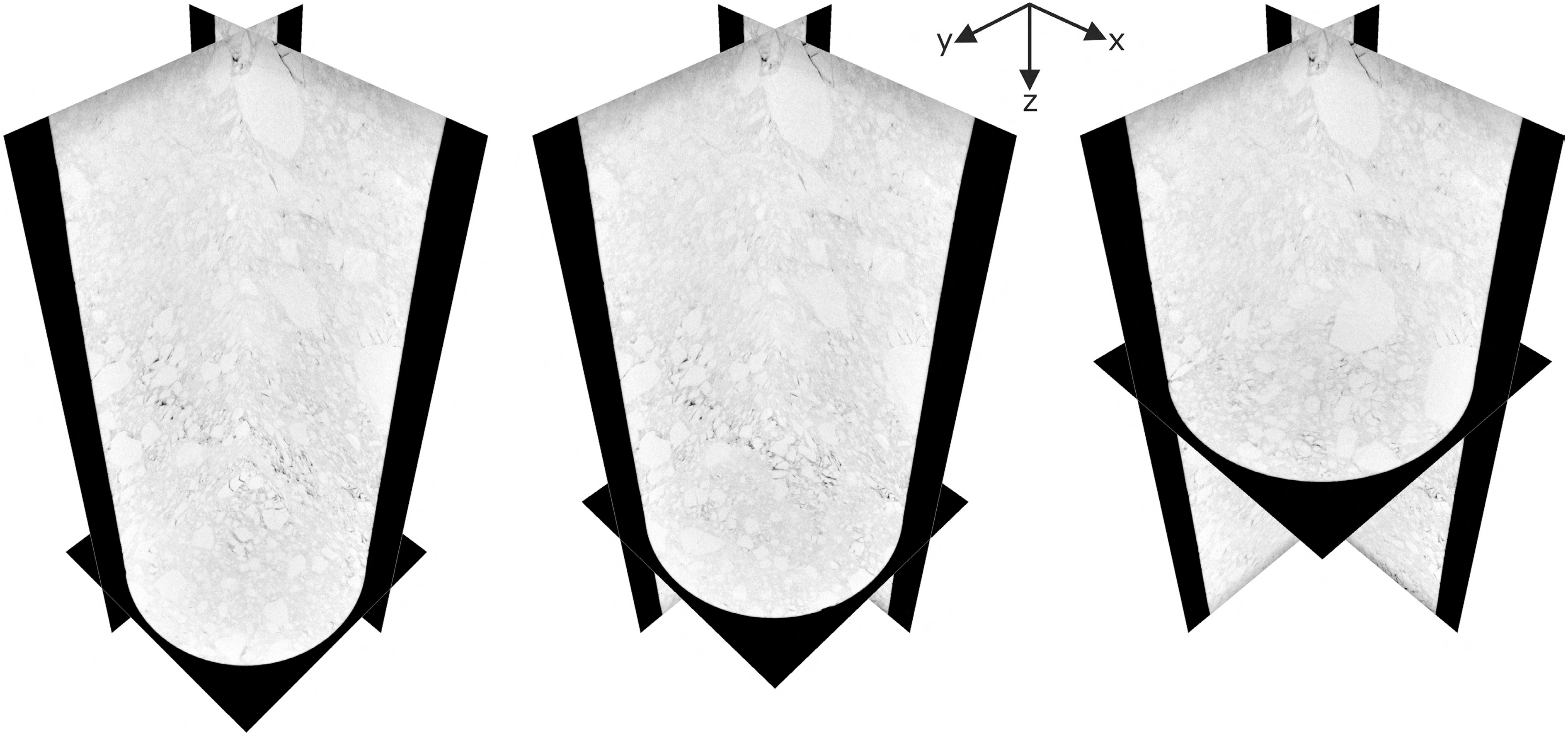


Figure E.3.7a. µCT data results for Ø 2 cm sample Strasshof T4-2sp at (12.5 µm)^3 / voxel resolution.

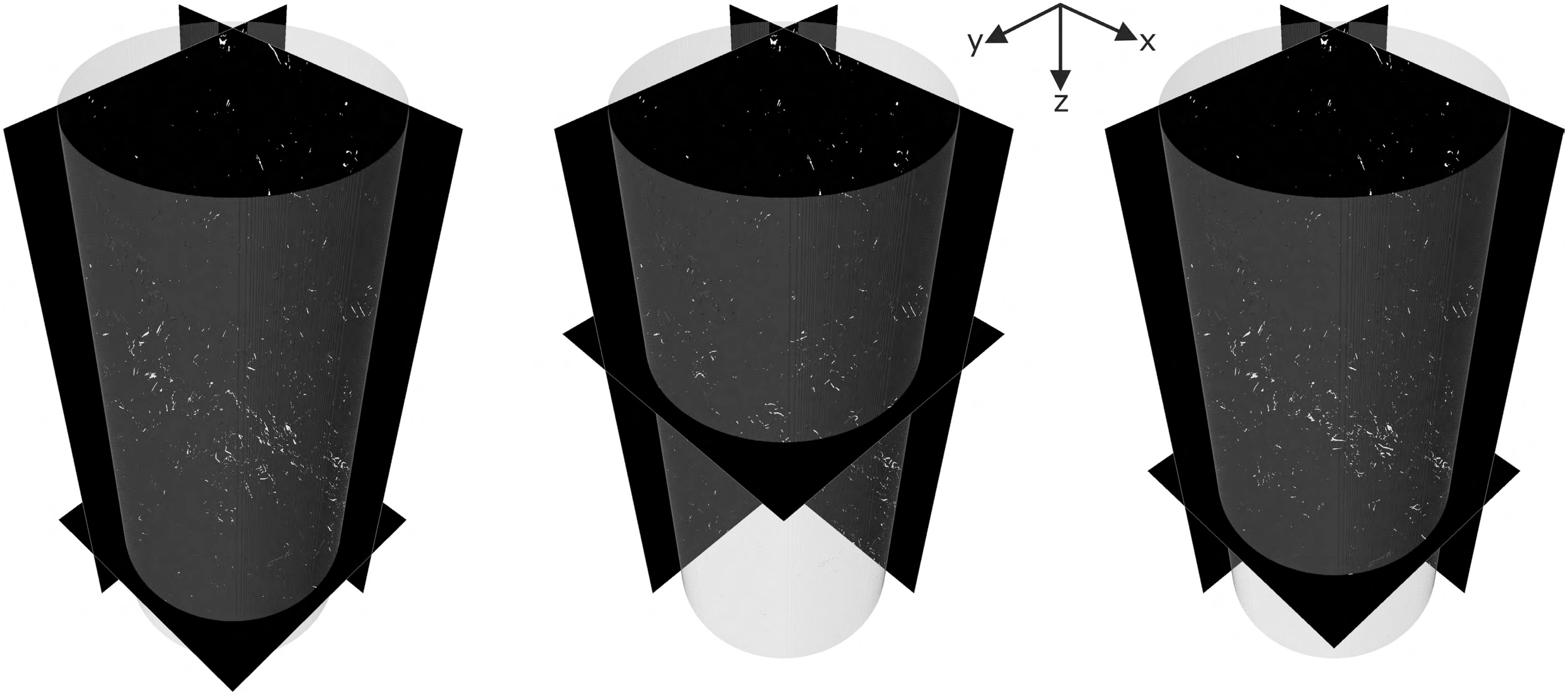


Figure E.3.7b. Segmentation results for Ø 2 cm sample Strasshof T4-2sp at (12.5 µm)^3 / voxel resolution.

F. Dolomite fault rock analysis

The following section has been submitted as a paper to the *Journal of Petroleum Science and Engineering*:

Voorn, M., Exner, U., Barnhoorn, A., Baud, P., Reuschlé, T., 2014. Porosity, permeability and 3D fracture network characterisation of dolomite reservoir rock samples.

The original text is presented here completely, to ensure internal cohesion within the paper. The reference list to the paper is implemented in the complete reference list of this thesis.

Porosity, permeability and 3D fracture network characterisation of dolomite reservoir rock samples

Abstract

With fractured rocks making up an important part of hydrocarbon reservoirs worldwide, detailed analysis of fractures and fracture networks is essential. Common analyses on drill core and plug samples taken from such reservoirs (including hand specimen analysis, thin section analysis and laboratory porosity and permeability determination) however suffer from various problems, such as having a limited resolution, providing only 2D and no internal structure information, being destructive on the samples and/or not being representative for full fracture networks. In this paper, we therefore explore the use of an additional method – non-destructive 3D X-ray micro-Computed Tomography (µCT) – to obtain more information on such fractured samples. Seven plug-sized samples were selected from narrowly fractured rocks of the Hauptdolomit formation, taken from wellbores in the Vienna basin, Austria. These samples span a range of different fault rocks in a fault zone interpretation – ranging from damage zone to fault core. We process the 3D µCT data in this study by a Hessian-based fracture filtering routine and can successfully extract porosity, fracture aperture, fracture density and fracture orientations – in bulk as well as locally. Additionally, thin sections made from selected plug samples provide 2D information with a much higher detail than the µCT data. Finally, gas- and water permeability measurements under confining pressure provide an important link (at least in order of magnitude) towards more realistic reservoir conditions.

This study shows that 3D µCT can be applied efficiently on plug-sized samples of naturally fractured rocks, and that although there are limitations, several important parameters can be extracted. µCT can therefore be a useful addition to studies on such reservoir rocks, and provide valuable input for modelling and simulations. Also permeability experiments under confining pressure provide important additional insights. Combining these and other methods can therefore be a powerful approach in microstructural analysis of reservoir rocks, especially when applying the concepts that we present (on a small set of samples) in a larger study, in an automated and standardised manner.

F.1. Introduction

Fractured reservoirs constitute a significant part of the world's hydrocarbon potential. As pointed out by Nelson (2001), the presence and topology of natural fractures in reservoirs have been and are too often ignored or oversimplified in their interpretation, leading to wrong decisions in exploration and production plans. Natural fractures cannot only contain significant volumes of hydrocarbons (connected porosity), but are in many reservoirs vital for allowing flow at sufficient rates for production (permeability), so analysing and understanding the complexity of fractures and fracture networks is crucial. A great deal of research on natural fractures has been carried out over the years, ranging from the scale of a full reservoir down to the analysis of micro- and nanofractures. Nelson (2001) and Aguilera (1995) have given excellent overviews of fractured reservoirs, their characteristics, and their analysis over a wide range of geological scales. Here, we will focus on fracture analyses at the scale of a drill core and a plug. Analyses carried out on such fractured drill cores and plugs may however not always provide the desired detail on fracture network characteristics, or may have other drawbacks, which will be shortly discussed here.

After recovery, initial research on drill cores and plugs is mostly done at hand specimen scale, in the form of logs. General interpretations can be extracted for example about composition or fracture density and orientation, and these data can be compared to borehole information like Formation Micro Imaging (FMI). The information at these scales is very important to obtain an overview of the various rock types and large structures present in the reservoir, but the resolution of the above techniques is limited, and does not provide details about the internal structure of the reservoir rocks.

After hand specimen analysis, cores and smaller drilled out plugs are selected to determine their (among others) porosity and permeability. This can be done for non-fractured as well as for (some) fractured samples. Standardised methods for laboratory determination of the connected porosity (for example by helium porosimetry or water saturation) and permeability determination (for example by nitrogen gas flow experiments) provide useful values for reservoir evaluation, and these values are also used in this study. They are however bulk values, and give no information about the internal structure (e.g. distribution of fractures) of a core or plug sample. Some laboratory methods like Mercury Intrusion Porosimetry (MIP) do provide information about the internal structure of samples, but are invasive in nature, and samples may not be suitable anymore for other analyses afterwards.

Following bulk laboratory measurements, thin section analysis on drill core samples and wall cuttings is often carried out, if a more detailed analysis of their internal structure is required. Thin sections provide very detailed, generally two-dimensional views of the microstructure of the rocks, and can provide crucial information on for example the diagenetic history of rocks or the microporosity. Thin section analysis, especially using Scanning Electron Microscopy (SEM), is therefore still an important part of our research. However, creating thin sections is destructive on the original samples, and the 2D view can be limiting, especially when looking at fracture networks.

Finally, several (less common) analyses can be carried out to characterise fractures on the core and plug scale, or are at least potentially applicable on cores and plugs. For example, fracture surface, fracture roughness and fracture aperture have been analysed using varying methods (for example by physical and optical surface profiling, Plouraboué et al., 1995; Isakov et al., 2001; Kulatilake et al., 2007; Sharifzadeh et al., 2008; Neuville et al., 2012), and can provide important clues about how fluid flows in fractures. These methods are however very detailed, and therefore usually only applied on single (also often artificially created) fractures. Another option is to generate theoretical models of fracture networks, by applying stochastic methods to create fractures with a realistic spacing, connectivity, aperture of flow properties (for example Discrete Fracture Networks, e.g. Singhal & Gupta, 2010; Sahimi, 2011). Such methods are especially useful for comparing the behaviour of rocks on various scales, and for example for upscaling to the complete reservoir scale, but may not be fully representative for the actual situation and complexity found in nature. Furthermore, if such models are applied, there is still a requirement for realistic input data, which is something the methods applied in this study can potentially provide.

The main drawbacks of conventional research on fractured core and plug samples by the methods mentioned above are thus that they (1) give only low levels of detail, (2) provide no information on the internal structure, (3) give two-dimensional information only, (4) are destructive on the samples, and/or (5) do not apply on, or are not representative for full fracture networks. There is thus a potential for improving core and plug analysis, ideally by a single method that does not possess any of the above drawbacks. A great candidate for such a single method is X-ray micro-Computed Tomography (µCT), since it allows us to image naturally fractured plug samples in 3D in a non-destructive manner. This means, µCT allows us to extract image data on these fractures – and properties we can derive from this (for example fracture connectivity, fracture aperture, fracture orientation) – which otherwise would remain uncovered. Although the use of µCT has its own drawbacks and limitations, especially when compared to the results of single methods mentioned above, it remains a very good compromise between them. Moreover, the use of µCT data becomes even more powerful when complementing it with other analytical methods, which will be discussed in this paper as well.

CT and µCT are nowadays commonly used in various branches of geosciences. For an overview on (µ)CT scanning, applications in the geosciences, and important parameters and processing steps to consider, the reader is referred to Ketcham and Carlson (2001), Mees et al. (2003), Cnudde et al. (2006) and Kaestner et al. (2008). Quite some (µ)CT research has been carried out on samples containing fractures (e.g. Keller, 1998; Bertels et al., 2001; Van Geet & Swennen, 2001; Sellers et al., 2003; Muralidharan et al., 2004; Karpyn et al., 2007; Zabler et al., 2008; Wennberg et al., 2009; Ellis, 2012; Watanabe et al., 2011). These papers cover a wide range of lithologies and scales, and contain interesting concepts about the processing and analysis of the obtained data, but have in common that they focus only on single fractures or relatively simple combinations of fractures. (µ)CT has been applied less on more complex (natural) fracture networks, but examples can be found in literature, for example on building materials (Landis et al., 2003; Ehrig et al., 2011), coals and clays (Montemagno & Pyrak-Nolte, 1999; Bossie-Codreanu et al., 2004; Lenoir et al., 2007) as well as on carbonates (Christe, 2009; Barnhoorn et al., 2010; Zalewska et al., 2011; Fusi & Martinez-Martinez, 2013; Jia et al., 2013). There is a large variation within these papers in their goals and on the complexity of their data acquisition and processing routines, but one of the main obstacles throughout the literature is the efficient 3D processing and analysis of fractures, and especially of narrow fractures. An important focus of this study lies therefore on the extraction of narrow fracture networks and other porous structures from µCT data, using the method introduced by Voorn et al. (2013). In that paper, the technical details of a Hessian-based filtering technique for the efficient extraction of planar features from 3D datasets are outlined. Although this filtering principle can be applied on any kind of 3D dataset with (narrow) planar features, the examples shown in the paper by Voorn et al. (2013) correspond to some of the fractured dolomite samples of current study. Here, we will expand on this initial processing step, and focus on the determination of important parameters for the hydrocarbon industry, such as porosity, fracture aperture, fracture density and fracture orientation.

The main question we try to answer here is thus how to efficiently analyse natural fracture networks in dense rocks, using various methods (2D and 3D imaging, and laboratory porosity and permeability determination), with an emphasis on 3D µCT. These descriptions will be primarily based on 7 dolomite plug samples from the basement of the Vienna Basin (Austria), with different fracture and porosity characteristics. What is not yet taken into account with these methods is the behaviour of the rocks at depth, instead of under surface conditions only. We therefore subjected most plug samples to permeability experiments under confining pressure. The good overview of the plug samples and their relation to a fault core-damage zone structure combined with these permeability measurements at elevated confining pressures can help in understanding the behaviour of the different rock types at depth, which in term can serve as an important input for permeability and reservoir modelling.

<u>F.2. Geological background and samples</u>

The rocks analysed in this study origin from the Upper Triassic (Norian) Hauptdolomit formation, a major formation present throughout the Alps and built up of mainly dolostone. In Italian-speaking regions, the same formation is termed Dolomia Principale, although large lateral and vertical variations do exist. In literature, the rocks of the Hauptdolomit formation are consistently thought to have formed by syn-sedimentary or very early dolomitisation (e.g. Fruth & Scherreiks, 1984; Blendinger, 1997; Masaryk & Lintnerová, 1997; Antonellini & Mollema, 2000; Meister et al., 2013), and are affected by partial dedolomitisation in some areas (Masaryk & Lintnerová, 1997).

All samples in this study are taken from wellbores drilled in the pre-Neogene basement of the Vienna Basin (Austria), with the depths of origin listed in Table F.1. In the Vienna Basin basement, the Hauptdolomit formation rocks are present in different tectonic nappes of the Northern Calcareous

Alps (NCA), with the NCA itself being a component of the larger allochtonous Alpine-Carpathian thrust complex. Both the Alpine nappe stacking in this thrust complex up to the end of the Paleogene, and the formation of the Vienna Basin by pull-apart extension from the Early Neogene (Miocene) onwards, have affected the Hauptdolomit rocks (Zimmer & Wessely, 1996; Arzmüller et al., 2006), leading to a complex dense fracture network over a wide range of scales (Peresson & Decker, 1997). From these densely fractured rocks, hydrocarbon (mainly gas) exploration and production have been successful since the early 1960s – circa three decades after the first hydrocarbon exploration in the basin infill of the Vienna Basin (Zimmer & Wessely, 1996). However, some of the fractured reservoirs produce better than others, and the reasons for this are not always well understood; partially due to the drawbacks of the analysis methods mentioned in the introduction. Also, because several of the wellbores are quite old (drilled in the 1960s and 1970s), logging information is not as extensive as for more modern wellbores, and are therefore not always useful for detailed microstructural analysis. The information gathered from µCT imaging may therefore help to understand the rocks – and the differences between the reservoirs – in a better way.

To be able to place our results into a better perspective than from the core alone, we are adopting a classification scheme for fault rocks based on fieldwork studies (Figures F.1a and F.5a). Most fault zones in carbonate rocks are characterised by a fault core, surrounded by a damage zone (e.g. Billi et al., 2003). In dolomite rocks of the Hauptdolomit and similar formations, there is often a single master fault containing the most intensely damaged rocks (Bauer, 2010; Schröckenfuchs, 2012). Furthest away from this master fault, virtually undamaged rocks (at least by the deformation phase(s) associated with the fault system of interest) are present. When moving towards the fault core, the fracture density increases, up to a point where the rock contains randomly orientated fractures with a spacing of less than 1 cm (Figure F.1b-d). Moving further towards the fault core, a transition zone between the damage zone and the actual fault core can be found. This transition zone is – depending on the fluid regimes that acted in the past – often characterised by mosaic breccias with partial calcite cementation (Woodcock et al., 2006; Figure F.1e-g). Finally, after the transition zone, one arrives at the actual fault core, characterised by cataclasites, where the degree of cataclasis tends to increase when getting closer to the master fault (Figure F.1h-j). A distinction can be made here between embryonic (<50% matrix), intermediate (50-80% matrix) and mature cataclasites (>80% matrix), where also the amount of rounded grains increases from embryonic to mature, although it is not always possible to unequivocally distinguish between the cataclasite types (Billi 2010). The boundary between matrix and grains is set at around 4 µm in this classification (Flügel, 2004).

The classification discussed above can be applied directly on the hand specimen samples we obtain, and if necessary it can be adapted after µCT and/or thin section analysis. Because we are working with small samples, and do not have a good overview of all the surrounding rocks as you would have in a fieldwork study, we have to stress the classification is mainly a tool to discriminate between the different rock types we observe, and not an indisputable evidence the rocks in question come from a certain part of a fault zone. Additionally, we can only sample rocks that do contain enough porosity and fractures to visualise using µCT, but do not readily disintegrate upon sample acquisition or analysis. This of course may introduce an (unavoidable) bias in the sampling, to which we will return in the discussion. Another point of discussion is that the selected samples should contain only natural fractures as much as possible. Although we cannot exclude the presence of some drilling-induced fractures completely, and although it is likely the samples underwent some changes due to unloading during drilling, the samples considered here do not show the clear indications for induced fractures as listed by Nelson (2001).

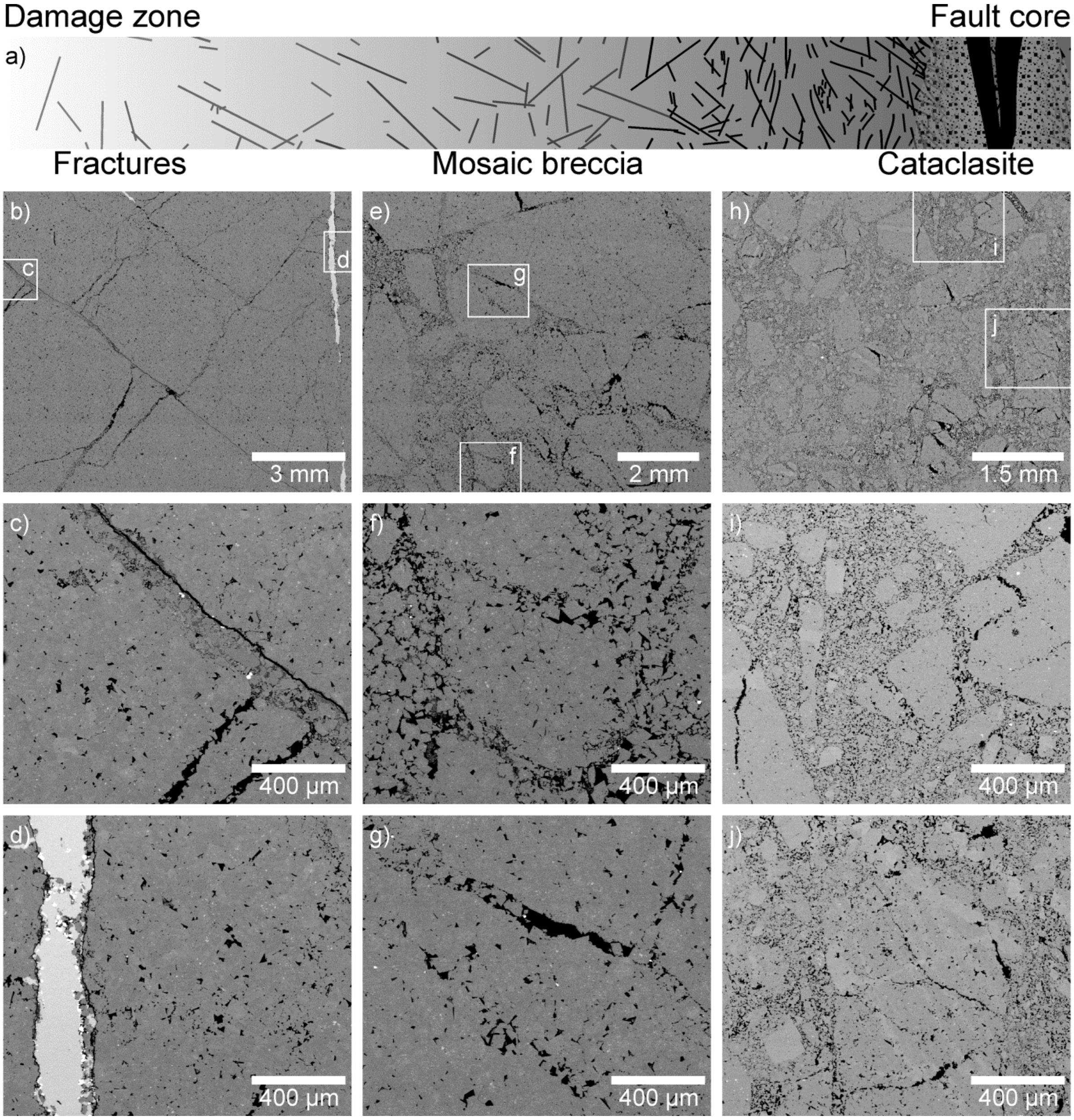


Fig. F.1. Microstructure examples of different fault rocks (SEM-BSE images). a) Fault zone structure as interpreted from fieldwork, showing the transition from damage zone towards fault core, with a gradual change from fractured host rock, via breccias, to finally fine-grained cataclasite in the fault core. b) Part of a stitched SEM-BSE overview on fractured sample Prottes TS1-8sp. Clear open fractures are visible in various orientations. Some are marked by increased clay content and fractured host rock. Additionally, in this sample several calcite veins are present. c) Detailed view in figure (b), showing a fracture crosscutting a possible cataclastic zone with increased clay content, and two fractures crosscutting dolomite host rock only. The main dolomite texture is interpreted as euhedral with interstitial micropores. Single dolomite crystals show an internal structure with small (remnant) calcite crystals and pores. d) Detailed view in figure (b), showing similar dolomite host rock as in figure (c), as well as a calcite vein (light grey) marked by fractures and framboidal pyrite (bright/white spots). e) Part of a stitched SEM-BSE overview on brecciated sample Prottes TS1-3sp. Large dolomite host rock fragments are visible, with finer grained dolomite and pores in between. Various fractures can be observed that crosscut host rock fragments. f) Detailed view in figure (e), showing several dolomite host rock fragments and finer material (sometimes nearly cataclastic) in between. g) Detailed view in figure (e), showing open fractures in dolomite host rock that are very similar to the ones in figures (b) and (c). h) Part of a stitched SEM-BSE overview on cataclasite sample Strasshof T4-2sp. Broken host-rock fragments are visible in a fine-grained cataclastic matrix. i) Detailed view in figure (h), showing several host rock fragments embedded in the cataclastic matrix . j) Detailed view in figure (h), showing a host rock fragment with several dolomite veins and fractures in various orientations.

The 7 dolomite plug samples scanned using µCT are listed in Table F.1, ordered from damage zone to fault core using the dolomite fault rock classification scheme. Table F.1 also lists several other characteristics of the samples, such as the dolomite texture and the cement and clay content present next to the dominant dolomite composition of the rocks. The information in Table F.1 is mainly based on thin section analysis (following the terminology of Flügel, 2004) on the specific samples themselves (non-italic), or inferred from µCT and thin section information from similar samples (italic).

Table F.1 Properties of samples scanned using µCT in this study

Sample	**Dimensions (Ø x l, cm)**	**Depth (m)**[1)]	**Fault zone interpretation**[2)]	**Dolomite Texture** [3)]	**Cements (veins and pores)**	**Clay content** [4)]
Prottes TS1	3.0 x 7.0	2795	Fractured (D)	Mainly subhedral with interstitial pores	Some calcite and pyrite pore filling and small veins.	High in matrix and along fractures.
Prottes TS1-8sp	2.0 x 2.2	2795	Fractured (D)	Mainly euhedral with interstitial pores	Large calcite veins, some pyrite veins. Locally abundant pyrite pore filling.	Some, mainly along fractures.
Schönk. T91-5p	3.0 x 5.7	3008	Fractured + mosaic breccia (D-C)	*Mainly nonplanar*	*Low pyrite content in matrix and fractures.*	*Abundant as pore and fracture filling, higher in brecciated parts.*
Prottes TS1-4p	3.0 x 2.2	2795	Mosaic breccia + cataclasite zones (C)	*Mainly euhedral with interstitial pores*	*Some calcite and pyrite pore filling and small veins.*	*Abundant in cataclasite zones.*
Prottes TS1-3sp	2.0 x 2.2	2795	Mosaic breccia + cataclasite zones (C)	Mainly euhedral with interstitial pores	Few calcite veins, low pyrite content in pores.	Abundant in cataclasite zones.
Strasshof T4-3p	3.0 x 7.0	3320	Embryonic-intermediate cataclasite (C)	*Nonplanar*	*Dolomite veins, possibly abundant. Low pyrite content in matrix.*	*Abundant in matrix.*
Strasshof T4-2sp	2.0 x 5.0	3320	Embryonic-intermediate cataclasite (C)	Nonplanar	Abundant dolomite veins and pore cements. Low pyrite content in matrix.	Abundant in matrix, some continuous bands.

Italic text means interpretation from µCT data or thin sections on a similar sample (not thin sections on sample itself). 1) Minimum depth of core as reported on core boxes. 2) D = damage zone, C = fault core. 3) Classification scheme of Sibley & Gregg (1987). 4) Main clay components: illite and kaolinite.

F.3. 3D and 2D imaging methods

F.3.1. 3D X-Ray micro-Computed Tomography (µCT)

F.3.1.1. µCT scanning details

The plug samples listed in Table F.1 were scanned using a Rayscan 250E X-ray micro-Computed Tomography (µCT) scanner at the Research Group Computed Tomography of the University of Applied Sciences of Upper Austria (Fachhochschule Oberösterreich, FHOÖ) in Wels, Austria. The used X-ray source is a 225 kV Viscom XT9225 DED microfocus X-ray tube with a minimum focal spot of circa 8 µm, and the device is equipped with a 2048x2048 pixels 16 bit a-Si flatbed detector. A copper pre-filter of 0.5 to 1.5 mm thickness (depending on the sample size) was used to reduce beam hardening effects. Source voltages of 120-165 kV were selected (hence resulting in maximum X-ray energies of 120-165 keV), and current and exposure time were adjusted empirically to ensure the best radiogram images. For all samples, between 1080 and 1800 projections were taken, and 4 to 5 radiograms of the same projection (i.e. same sample orientation in the scanner) were averaged. This averaging of course increases the scanning time per sample dramatically, but results in a much better image quality with less noise, which is especially useful when working with narrow features in dense materials. After scanning, the data was reconstructed (i.e. calculating a 3D image volume from the 2D radiograms) with a filtered back-projection algorithm using the devices' included Rayscan reconstruction software, with a resolution of (12.5 µm)^3 (2 cm diameter samples) up to (20 µm)^3 (3 cm diameter samples) per voxel. A voxel is an abbreviation for a volume element, and is the 3D equivalent of a pixel. Additional filtering was kept to a minimum (e.g. no median filter was applied on the final reconstructed data), but some software filtering to reduce the effects of beam hardening and ring artefacts was inevitable.

Additionally to the plug samples of Table F.1, quite a few other samples were scanned, also using different µCT scanners. Most of these scans were unsuccessful due to various reasons, such as too little structure being visible at the µCT resolution, or too much noise being present in the final reconstructed result. In this process, we also attempted to scan larger core samples, of 6 and 10 cm in diameter. However, X-ray penetration turns out to be insufficient at the energy levels mentioned above, and higher energy settings are no option due to a decline in resolution, which would make visualisation of the narrow fractures in these samples impossible.

F.3.1.2. µCT data processing: MSHFF filtering and segmentation

As indicated in the introduction, processing µCT data – especially when containing narrow features relative to the resolution – can be problematic. The most promising approaches currently available in literature attempt to automatically fit a point spread function (PSF) to the µCT data in order to calculate the actual size of visible features (e.g. fracture aperture), often using a physical calibration (Ketcham et al., 2010; Ketcham & Hildebrandt, 2014). Such techniques are unfortunately still difficult to apply automatically – due to computational reasons – on full datasets with complex features, such as a fracture network. We therefore chose to apply a 3D voxel-based segmentation, using the multiscale Hessian fracture filter (MSHFF) of Voorn et al. (2013) – based on the approach for vessel filtering by Frangi (1998). The MSHFF method has been implemented as macro code for the open-source software FIJI (Schindelin et al., 2012) – using the FeatureJ plugin (Meijering, 2010) for the calculation of the Hessian matrices – and runs on a desktop computer for the datasets displayed in this paper, without having to downsample these datasets first. The technical details and examples of segmentation results on fractured dolomite rocks are shown in Voorn et al. (2013). This routine has been slightly altered, leading to a better combination of different scales of analysis, a better result on the narrowest features extractable from a dataset, and no requirement to use an empirical "calibration curve". After the filtering is finished, a final binarised result (i.e. the data is divided into "solid" and "fractures and porosity (above the resolution limit)") is reached by defining a single threshold value on the filtered data. The segmentation results shown in this paper are not filtered for connectivity throughout the plug samples, as the µCT resolution cannot account for possible connectivity below its resolution (e.g. at the matrix-scale). Instead, an island filtering approach is used to remove noise, by deleting small clusters (less than 20 connected voxels in a 27-connected geometry) from the dataset. The final binary segmentation using the MSHFF makes it easier to analyse the data afterwards and perform further calculations on them.

F.3.1.3. µCT data analysis

F.3.1.3.1. Porosity

After binary segmentation, the bulk porosity was determined by dividing the amount of voxels assigned to be porous fraction by the total amount of voxels in the dataset, and normalising to 100%. Of course, the value of the porosity is very dependent on the initial data quality, the filtering process and the final segmentation result, and does hence partially reflect the choices made by the operator. This is however a problem for most available segmentation routines (with some performing better than others).

F.3.1.3.2. Aperture

The aperture of fractures is an important parameter in for example permeability modelling attempts, and should be extracted from the available datasets. Using a calibration approach, apertures and aperture distributions may be extracted from µCT data without needing to segment these (e.g. Keller, 1998; Montemagno & Pyrak-Nolte, 1999; Muralidharan et al., 2004; Ketcham et al., 2010). However, as indicated before, such calibration approaches are generally difficult to run on fracture networks, and additional µCT scanning time is required for the calibration. We will therefore work with the segmented µCT data instead. A possible approach on the available data would be to try to discretise the visualised fractures and determine the aperture (or aperture distribution) for every fracture. However, discretising the fractures in the complex fracture networks shown in this study is difficult (if not impossible), and apertures clearly vary along the fractures, leaving little meaning for single aperture values per fracture. We therefore extract the aperture distribution for the whole sample (i.e. for the whole fracture network), while at the same time assigning a value for maximum aperture at every voxel belonging to the fracture network.

To get this aperture information, the so-called *Local Thickness* plugin (Dougherty & Kunzelmann, 2007) for FIJI is run on the segmented datasets. Similar approaches with different implementations can be found in literature (e.g. Yang et al., 2009 for MATLAB ((The Mathworks, Inc., 2011)), but the Local Thickness plugin could be applied the easiest on large datasets. In this approach the largest possible spheres (thus 3D) are fitted at every location inside the selected features in the dataset. For porous rocks, such an approach can be used to get a distribution of pore size. For fractures however, the diameter of the spheres is a representation of aperture at the location of the sphere. Once the full process is finished, the 3D dataset with the local aperture distributions is saved (see example in Figure F.2), and a histogram of the sample's aperture distribution can be made. To compensate for volume bias (i.e. larger spheres contain many more voxels), we divide the distribution by the volume of each sphere size. Finally, in order to be able to compare the samples, the aperture distribution in voxels is multiplied with the µCT scanning resolution.

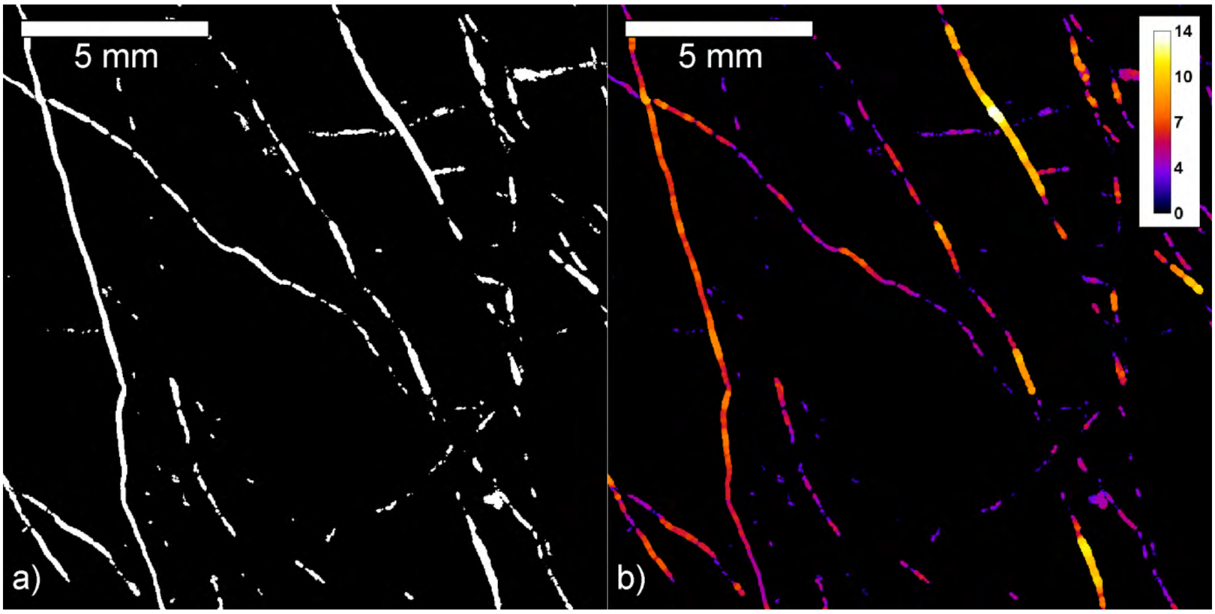


Fig. F.2. Example of the *Local Thickness* approach (Dougherty and Kunzelmann, 2007) for aperture determination on a fractured dolomite sample. a) 2D slice through a 3D binary segmented dataset of µCT data, at (18.9 µm)^3 per voxel resolution. The 3D binary segmented data serves as the input for the *Local Thickness* approach. b) Local aperture distribution, shown as a 2D slice of the same area of figure (a), taken from the full 3D processed dataset. To obtain this result, the porous areas shown in figure a (white) are fitted with spheres of the largest possible diameter using the *Local Thickness* algorithm. For every voxel, the largest possible sphere diameter at that location is recorded and plotted in the figure. For fractures, these sphere diameters can be regarded as a local measure for aperture. The sphere diameters in the example figure are shown in units of voxels.

F.3.1.3.3. Fracture density

Fracture density is a parameter that can be expressed in various ways, depending on the analysis method and scale. For the samples shown here, and for reasons mentioned before (because separating the fractures efficiently is not possible), we apply the simplest bulk definition available in literature: fracture density [m^{-1}] = total fracture surface [m^2] / sample volume [m^3] (Singhal & Gupta, 2010). To fill in this equation, sample volume is of course readily available from the µCT data. To calculate the total fracture surface, we use the results from the aperture determination approach. We assume the total number of voxels in an aperture size class can be divided by its aperture, to provide the fracture surface of that aperture size class. This assumption can be made for fractures since the fitted spheres of the aperture determination approach overlap completely. The fracture surfaces from different size classes are summed up and divided over the volume, and the µCT scanning resolution is used to convert the fracture density to the proper units of m^{-1}.

The obtained values for fracture density of the method described above are higher than generally seen in for example fieldwork studies (Singhal & Gupta, 2010), because of the much higher level of detail available in the µCT scans. The results on the µCT scans can therefore not always be readily used to compare with fracture density results from other studies, but can be used to compare different samples scanned with µCT. Also, when looking in detail on thin sections, and applying a similar approach for fracture density determination in 2D, results in the same order of magnitude are obtained, so the values from µCT and thin sections may be used concurrently.

F.3.1.3.4 Fracture orientation

Knowing the orientation of fractures in a dataset can be very useful input in reservoir evaluation, especially if small-scale fractures display similar behaviour as the larger-scale network. In this case, due to the age of the wellbores, well orientation information is unfortunately not available for comparison to the datasets. Extracting the orientations from image data is a relatively simple task when discretised fractures are available. For the voxel-based data of complex fracture networks – where discretising the fractures is virtually impossible – a different solution is required. We decided to use the concepts of the Hessian based filtering and segmentation method described in Section F.3.1.2. In that filtering routine, only the eigenvalues of the Hessian matrix are used. However, the complete Hessian matrix is being calculated in the process, meaning one can also extract the eigenvectors for the voxels belonging to the extracted fracture network. The eigenvector belonging to the largest Hessian eigenvalue represents the normal direction to the extracted fracture plane. When these normal orientations are plotted in a stereographic projection and contoured for density, this provides a clear representation of the fracture orientations in the sample. Since this is a voxel-based approach, orientations of larger fractures (i.e. fractures containing more voxels) are automatically assigned more weight in the final stereographic projection.

In the process of obtaining the stereographic projections of fracture orientations, several steps are taken. In principle, it would be possible to perform the orientation analysis on the raw input µCT data. However, the segmented results from the MSHFF filtering process are much easier to process, and result in less noise compared to using the raw input data. We therefore decided to run the orientation analysis on the already segmented results. In this calculation step, an adapted version of the MSHFF filtering approach in FIJI outputs the full Hessian matrix at every voxel. This data is then loaded into MATLAB (The Mathworks, Inc., 2011), where the normal direction to fracture planes is extracted from an eigenvector analysis on the datasets. These orientations are then plotted on an equal area stereographic projection, using density contouring relative to a Kalsbeek grid (a regularly spaced grid in equal area projection). It is important to use an effective calculation for performing this density contouring, since for a single dataset, tens to hundreds of millions of orientations are available. Finally, in order to be able to compare the different samples of this study to each other, the density contouring of the stereographic projection is normalised to a single relative maximum value suitable for all samples.

F.3.2. Thin sections

The limited resolution of µCT on plug-sized samples as shown in this study necessitates the use of higher resolution methodologies to gain a better understanding of for example matrix porosity, cement contents, dolomite texture, et cetera. One option would be to create small subsamples from the original plugs and scan these at better resolution using µCT or synchrotron X-ray CT. We however decided to create 2D thin sections after plug scanning, as this is a relatively cheaper and simpler method to apply, but can still provide a lot of detailed insight.

The µCT scans help in selecting the most suitable areas for creating thin sections. After thin section generation, we produced stitched thin section overviews from Scanning Electron Microscopy (SEM) images with back-scattered electrons (BSE) contrast, since in this study we are mainly interested in the comparison of the full thin sections to the µCT data. As a stitching routine, we either used the automatic Microsoft Research Image Composite Editor (ICE; Microsoft Research, 2011) or a self-written linear translation stitching script in FIJI, including some brightness and shading corrections. The main results on for example microstructure and mineral contents from these overviews – as well as more detailed analysis on some thin sections – are shown in Table F.1.

Figure F.3 shows a comparison between µCT and SEM-BSE data, for an approximately matched 2D view from the 3D µCT data and a 2D thin section, in which the difference in resolution is clear. Especially the fine porosity in the matrix is not resolvable using the µCT resolution only. In order to assign a value to the porosity visible in the thin section, we perform a binary segmentation on the SEM-BSE thin section overview. As indicated before, binary segmentation is always prone to error and can be a reflection of the choices made by the operator. For this reason, we do not select a threshold manually, but try to prevent as much bias as possible using automated binary segmentation by the Otsu threshold (Otsu, 1979). The Otsu threshold – although still not ideal –

provides good, repeatable results for the samples, due to the similar image acquisition and processing technique for the different samples in this study. It therefore allows us to compare the results on the samples readily.

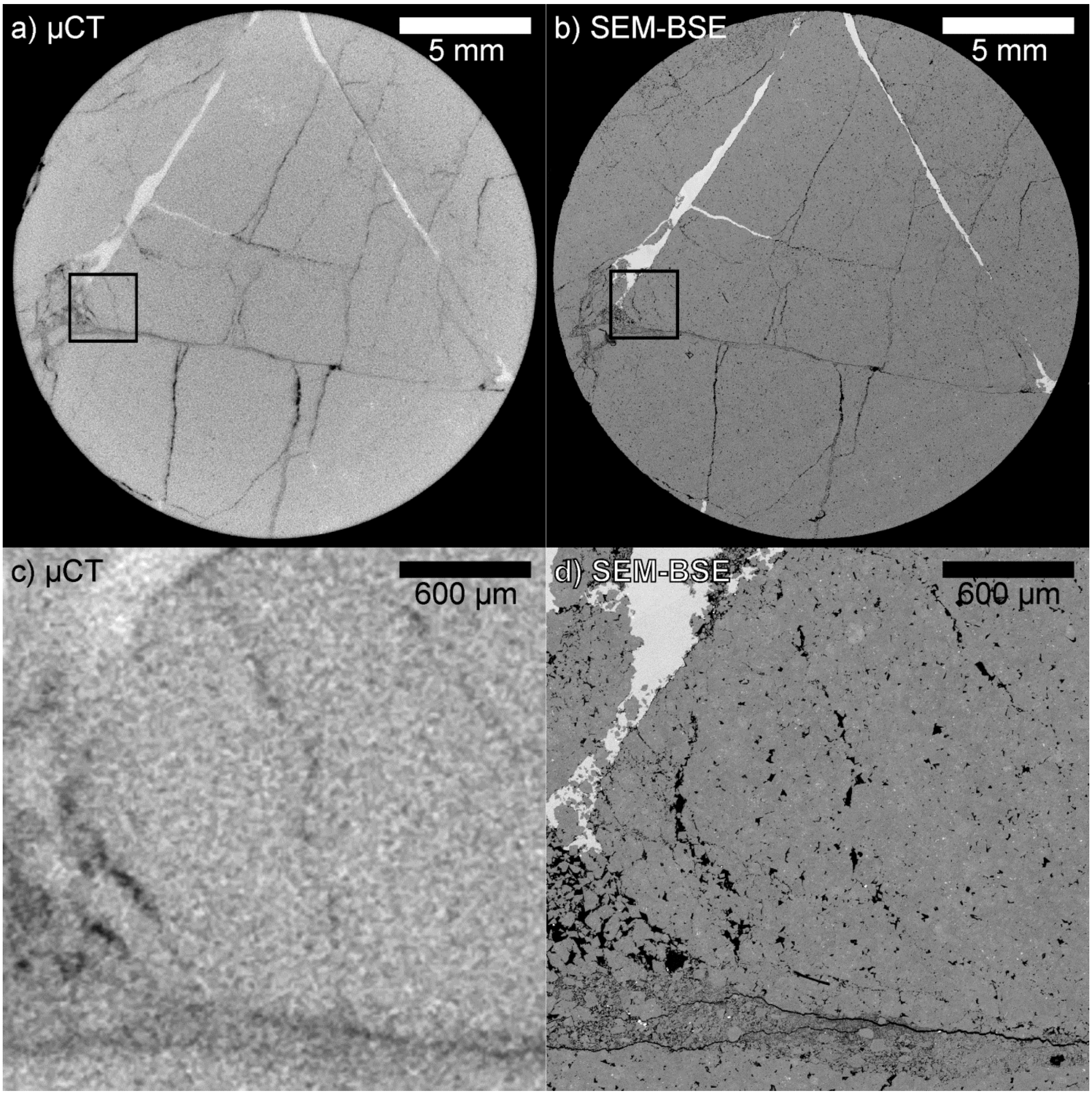


Fig. F.3. Comparison of µCT and thin section data on a fractured dolomite plug sample of 2 cm diameter. Figures (a) and (c) are taken from a 2D slice through a 3D µCT dataset at $(12.5\ \mu m)^3$ per voxel resolution. Figures (b) and (d) show thin section SEM-BSE imaging at $(1.45\ \mu m)^2$ per pixel resolution that have been (approximately) matched to the data in figures (a) and (c). The overview in figure (b) is a stitched result from 292 SEM-BSE images. In overview (figures (a) and (b)), the sample shows several clear fractures, as well as some more brecciated areas (e.g. left of the black squares) and calcite vein filling (bright material). When zooming in on the black squares, figures (c) and (d) show the clear difference in detail between the µCT and SEM-BSE thin section data. Although larger features such as big fractures, large enough open pores and calcite vein filling are still recognisable in the µCT data, the thin section data allows a much more detailed analysis. For example, the interaction between the calcite vein and the dolomite host rock is clearly visible in the thin section data, and so are the details in the cataclastic and clay-rich area around the fractures at the bottom of figure (d). Furthermore, many small pores and the dolomite texture (here interpreted as mainly euhedral) can be observed, which is not possible in the µCT data. Finally, an internal structure in the dolomite grains is visible, consisting of small pores and calcitic areas, interpreted as a remnant of dolomitisation.

F.4. Permeability experiments

As stated in the introduction, all previously described methods provide detailed information on different parameters (porosity, aperture, fracture orientation etc.), but all only under atmospheric (or near-atmospheric) conditions. In order to obtain a better link to the behaviour of the rocks at depth (i.e. more realistic reservoir conditions) we apply flow experiments under confining pressure. Here we used single phase steady state flow under the assumption that Darcy's law was valid, using either gas or water as pore fluids at room temperature, in two different experimental setups. In the following, these two types of permeability experiments are discussed in more detail.

F.4.1. Gas permeability

Gas permeability experiments under confining pressure were carried out on 2 cm diameter plug samples at the École et Observatoire des Sciences de la Terre at the University of Strasbourg, France. The samples were prepared in such a way that the two planar surfaces (top and bottom of sample) are parallel to each other and perpendicular to the long axis of the plug sample. After drying, the samples were put in a jacket and placed in the permeameter. In this permeameter setup, distilled water acts as the confining medium, and nitrogen gas as the pore fluid. A sketch of the experimental setup can be found in Heap et al. (2014).

In the gas permeability setup, the permeability can be determined using the following derivation of Darcy's law, for a compressible gas (Scheidegger, 1974; Tanikawa & Shimamoto, 2006):

$$\kappa_{app} = \frac{\mu dx}{A} Q \frac{2P_{down}}{(P_{up})^2-(P_{down})^2} \text{ (Equation F.4.1)}$$

Where κ_{app} is the apparent permeability [m^2], μ the dynamic viscosity of the pore fluid [Pa s], *dx* the sample length [m], *A* the cross sectional area to flow [m^2], *Q* the flow rate [m^3/s], P_{down} the downstream pore pressure [Pa] and P_{up} the upstream pore pressure [Pa]. In the experiments, P_{down} was kept constant at atmospheric pressure (0.1 MPa), and P_{up} was varied up to a maximum of 0.5 MPa. At any given confining pressure (up to 50 MPa in this setup), the flow rate *Q* was measured at at least 4 controlled variations in P_{up}, and these measurements are used to derive a single permeability determination at this confining pressure.

In Equation F.4.1, the permeability is noted as an apparent permeability, because effects of gas slippage are not yet compensated for. Although permeability is supposed to be an intrinsic value of the porous medium, differences in experimental determination of the permeability can occur when using different pore fluids. For gas, higher permeabilities than the true intrinsic permeability are often recorded because the flow velocity of the gas along the walls of a porous medium is not zero. To compensate for this effect termed Klinkenberg gas slippage, the Klinkenberg correction is carried out (Klinkenberg, 1941), relating the apparent permeability to the mean pore fluid pressure. For permeabilities above circa $1 \cdot 10^{-15}$ m^2 the Forchheimer correction is carried out, relating the measured permeability to the flow rate. After correction, the permeability κ should represent the true intrinsic permeability for the sample. This permeability value is then used in plots relative to the confining pressure. Because the pore fluid pressures in these experiments were relatively low compared to the used confining pressures, we neglect the influence of the pore fluid pressure, and consider the confining pressures to represent effective pressures.

For the 3 cm diameter plug samples, gas permeability experiments could not be carried out in the above setup. However, nitrogen gas permeability measurements (under atmospheric conditions only) were done at OMV LEP, before µCT scanning. These gas permeability measurements will be displayed as an example to the water permeability experiments on these 3 cm diameter plugs (Figure 5g), but it should be noted no Klinkenberg or Forchheimer correction was made for these samples.

Because no such correction could be carried out, a possible error on these measurements of 25% should be regarded.

F.4.2. Water permeability

Several 3 cm plug samples of this study were used in water permeability experiments under confining pressure at the Faculty of Civil Engineering and Geosciences at the Delft University of Technology, The Netherlands. Unfortunately, one of the fractured samples (Prottes TS1) disintegrated after µCT scanning and could therefore not be used anymore for flow experiments. A different fractured sample (Schönkirchen T32-430) was used as a replacement, to have at least one fractured sample available in the experimental results.

Similarly as for the gas permeability samples, the plugs were prepared in a plane parallel fashion. The samples were first water-saturated under a vacuum, before being mounted in a jacket in the pressure vessel. This experimental system is similar to the gas permeability setup, but with one important difference: the axial pressure is controlled separately from the radial pressure in the setup. We will return to the influence of this layout later in this section. As a pore fluid, tap water was used, and as a confining medium for the radial pressure, an incompressible oil (Multitherm PG-1®) was used.

The permeability in the water flow experiment setup can be determined using the linear form of Darcy's law (Tanikawa & Shimamoto, 2006):

$\kappa = \frac{\mu dx}{A}\frac{Q}{DP}$ (Equation F.4.2)

DP in this equation is the differential pore pressure in Pa, similar to p_{up}-p_{down} or p_{down}-p_{up} (depending on the flow direction) in Equation F.4.1. In the experimental setup, a sensitive differential pressure meter is present that measures this pressure difference. Because of this, and because the pore fluid pumps in the system are more sensitive to flow rate, we control and vary the flow rate *Q* in the setup, and keep the pore pressures on one side of the sample constant. After every controlled change in flow rate, we wait for the DP to stabilise with time before taking the actual measurement. For every chosen confining pressure, there are several (minimum 4) measurements of *DP* at a certain *Q* that are combined into a single permeability value at that confining pressure, unless stated otherwise. Note this measurement routine is the inverse of the gas flow experiments (where *Q* was measured relative to a controlled varied pore fluid pressure), but this should not have any additional effects on the final permeability values.

Several observations were made on the samples regarding the measurement routine. First of all, all samples show a marked drop in permeability during the initial flow stages, without any changes in confining pressure. After some period of time, the permeability measurements stabilise, and it is hence important to wait for this initial stage to be over. Second, some tests show the response of permeability to several parameters, namely the axial pressure only, radial pressure only, the inversion of flow and time-dependency without flow. These results are shown in Figure F.4. Note that in all subfigures, single measurements of permeability are shown (i.e. at a single *Q* and *DP*). Figure F.4a and b show that although in all experiments we tried to keep the axial pressure as close as possible to the chosen radial pressure at all times, it is clear this fractured sample seems to respond significantly more to changes in radial pressure. We therefore take the radial pressure in all measurement series as the actual confining pressure. Furthermore, inverting the flow (Figure F.4c) does not seem to have any significant effect on the permeability. If for example particle blockage is the main cause of permeability reduction in a sample, one would expect a short rise in permeability after flow inversion (e.g. Engler, 2010). Finally, keeping the sample pressurised (confining pressure; Figure F.4d) for several hours without any flow does not have a large influence on the permeability, indicating no or only minor time-dependent effects are acting in the samples during these experiments. These tests (Figure F.4) thus indicate the increasing confining pressure is the dominant mechanism for water permeability reduction shown in the following sections.

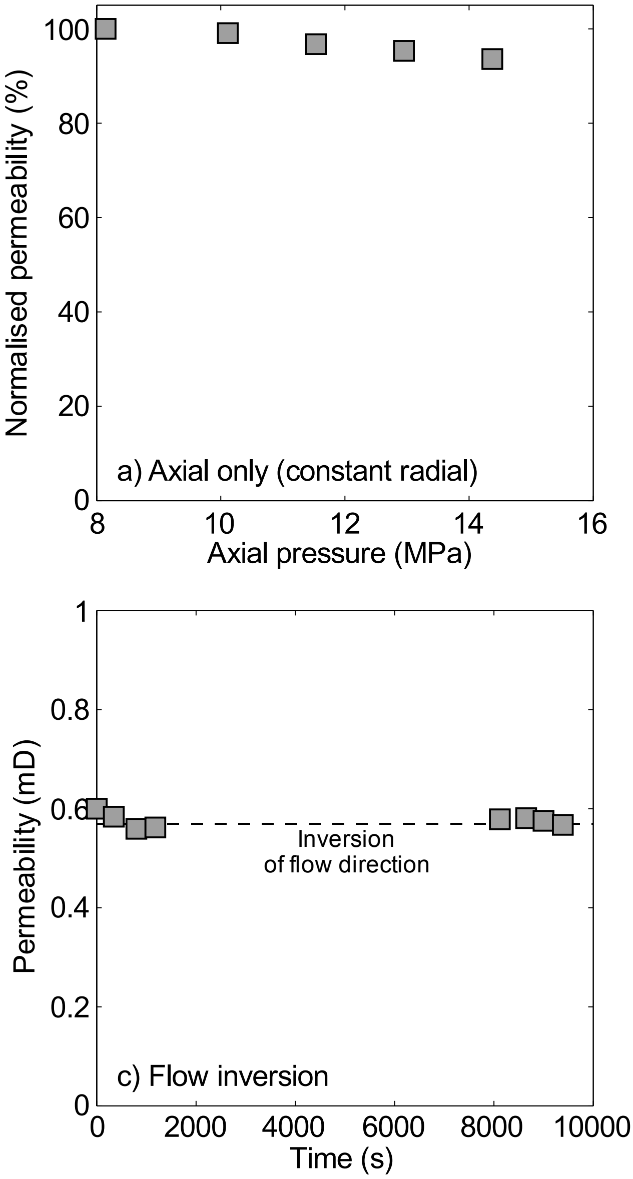


Fig. F.4. Response of permeability to changes in several parameters, for a water permeability experiment on a fractured dolomite sample. The measurements shown here represent single permeability measurements (i.e. one *Q* and *DP* in Equation 4.2). a) Permeability response to changes in axial pressure. The radial pressure was kept constant at circa 7.4 MPa. b) Permeability response to changes in radial pressure. The axial pressure was kept constant at circa 9.7 MPa. c) Influence of inverting the flow direction, while keeping all other parameters constant. d) Influence of time without flow. After the first series of measurements, the flow was turned off, but all other parameters (confining pressure, pore fluid pressure) were kept constant for several hours. When turning the flow on again, no significant change in permeability was observed. Especially the first measurement after flow inversion is completely in the range of the first series of measurements.

Table F.2 Samples, symbols, porosity and fracture density-porosity relationship

Sample	**Symbols in diagrams**	**Lab-Φ1)**	**µCT-Φ**	**TS-Φ2)**	**Fracture density / µCT porosity**	**Comments**
Schönkirchen T32-430	▨	1.7 % (He)	N.A.	N.A.	N.A.	Fractured sample, no µCT scan available
Prottes TS1	■	9.7 % (He)	3.0 %	14.1 %	259	Fractured sample, no permeability data
Prottes TS1-8sp	□	3.8 %	0.9 %	4.2 %	389	
Schönkirchen T91-5p	◆	5.1 % (He)	4.2 %	N.A.	202	
Prottes TS1-4p	●	8.9 % (He)	2.6 %	N.A.	230	
Prottes TS1-3sp	○	6.7 %	2.0 %	9.2 %	304	
Strasshof T4-3p	▲	5.1 % (He)	1.8 %	N.A.	302	
Strasshof T4-2sp	△	5.8 %	0.9 %	12.8%	481	

1) Porosities measured by Helium porosimetry are indicated by "He", the other porosities are measured by water saturation porosimetry. 2) Thin section porosities, averaged out over several 2D thin sections taken from the plugs.

F.5. Results and discussion

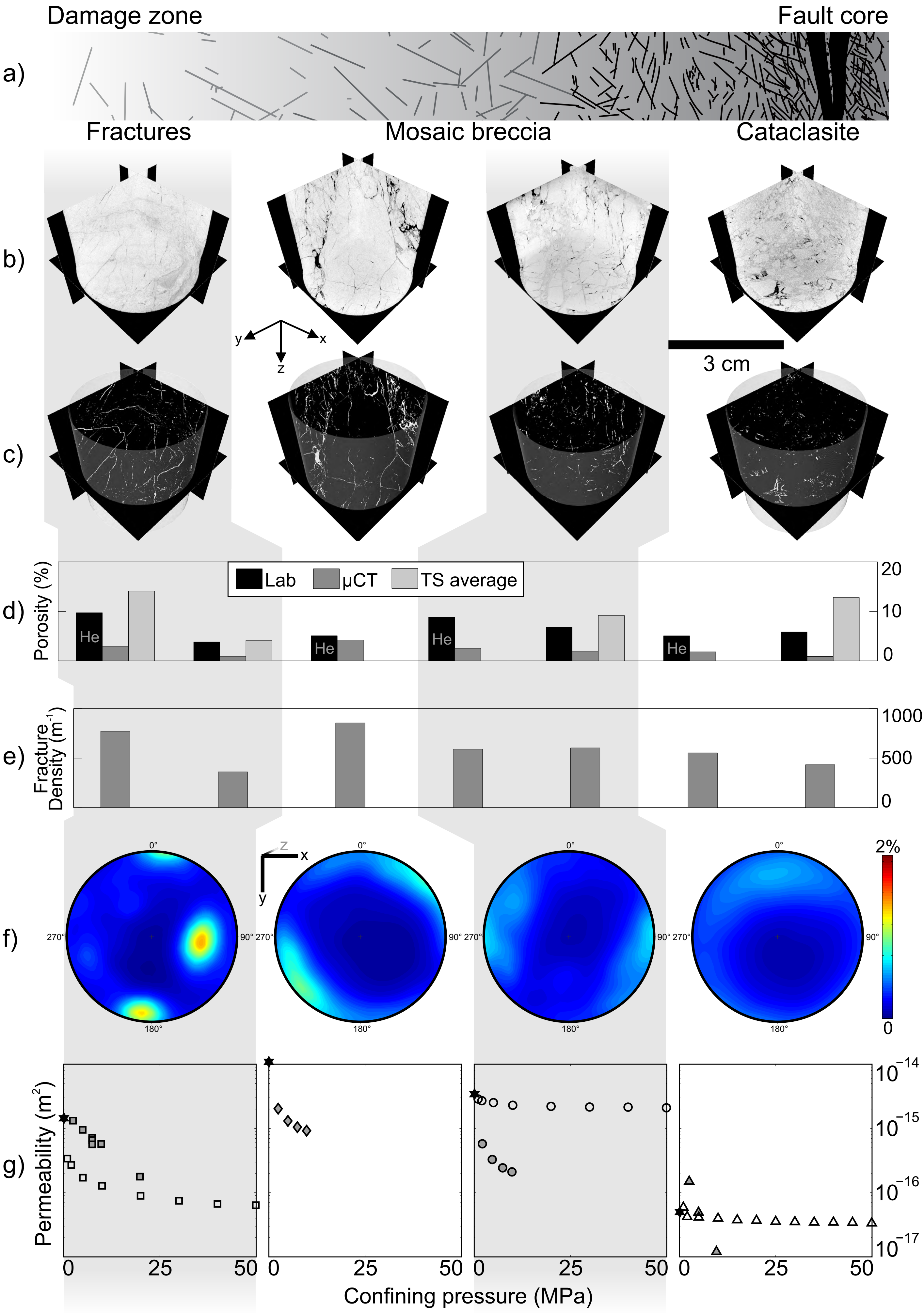
The main results of this study are shown in Figure F.5, linked to the damage zone-fault core structure in dolomites derived from fieldwork studies (Figure F.5a). Four examples of µCT scans (on 3 cm diameter plug samples) are shown (Figure F.5b), portraying the differences in structure when moving from fractured host rock in the damage zone (1st and leftmost column), via a mixture of fractures and breccia (2nd column) and mosaic breccias (3rd column), towards cataclasites in the fault core (4th and rightmost column), as well as variations within the samples themselves (for example, locally enhanced clay content in the matrix, or discontinuous vein fillings). When taking the MSHFF segmentation into account (Figure F.5c), the differences between the porous fracture networks (and separate fractures) in different parts of the fault zone become even clearer. Note that in these segmentation results, apparent mismatches with the input data (Figure F.5) may appear to be present, but are often due to the contrast and brightness settings required to display the input data properly here. The various parameters that can be extracted from such datasets (as listed in the Section F.3), as well as the related laboratory results, will be discussed in more detail in the following sections.

Note that in the results figures, we attempt to use the same symbols and symbol fillings for the same samples. Samples with a similar fault zone interpretation have a symbol of equal shape. The used symbols are listed in Table F.2.

F.5.1. Porosity

The results of porosity determination by laboratory methods (helium porosimetry for the 3 cm diameter samples, water saturation porosimetry for the 2 cm diameter samples), µCT and thin section analysis are shown in Figure F.5d and listed in Table F.2. Thin sections have not been made for all the available samples in Tables F.1 and F.2, and are thus not shown for every sample.

Fig. F.5 (next page). Compilation of the main results of this study. For clarity, the examples and results are grouped using grey and white vertical bands in the background. The four groups from left to right are 1) fractured-only samples, 2) mixtures of fractures and breccias, 3) mosaic breccias, and 4) cataclasites. See main text in Section 5 for a detailed discussion. a) Fault zone structure as interpreted from fieldwork, showing the transition from damage zone towards fault core, with a gradual change from fractured host rock, via breccias, to finally fine-grained cataclasite in the fault core. b) Orthoslice based views through four exemplary 3D µCT datasets of 3 cm diameter plug samples. The resolutions for these samples are, in order from left to right: $(18.9, 19, 19 \text{ and } 20\ \mu\text{m})^3$ per voxel. c) MSHFF binary segmented results of the same samples as in figure (b). The fracture network and pores are shown in white. The boundary of the 3D segmentation results is shown by the light-grey cylinder. d) Bulk porosity for the different samples. Generally speaking, the µCT values (imaging resolutions between $(12.5\ \mu\text{m})^3$ and $(20\ \mu\text{m})^3$ per voxel) are the lowest, laboratory values are intermediate, and SEM-BSE thin section values (imaging resolutions between $(1.1\ \mu\text{m})^2$ and $(1.7\ \mu\text{m})^2$ per pixel, full thin section overviews only) the highest. Note the laboratory values are derived from helium porosimetry (indicated by “He”, 3 cm diameter samples) and water saturation porosimetry (2 cm diameter samples). e) Fracture density values for the same samples as in (d). To obtain these results, the aperture distributions from the *Local Thickness* approach (Sections 3.1.3.2 and 5.2) have been used. f) Examples of fracture orientation. The pole figures are equal area, lower hemisphere projections of the poles to the fracture planes, at every porosity voxel in the dataset. For comparison, the four pole figures have been plotted to the same scale. From left to right, the number of elements is $N = 60{\cdot}10^6$, $226{\cdot}10^6$, $91{\cdot}10^6$ and $130{\cdot}10^6$. g) Permeability values during confining pressure loading experiments, grouped per sample type. Open symbols represent gas permeabilities, and closed symbols water permeabilities; the symbol shapes (Table F.2) refer to the different fault zone interpretations for the samples. Gas permeability experiments without confining pressure – on the same samples with the closed symbols (water permeability experiments) – are shown as stars. No correction has been applied on these latter measurements so a 25% error margin should be regarded.


Damage zone
Fault core
a)
Fractures
Mosaic breccia
Cataclasite
b)
y
x
z
3 cm
c)
d)
Porosity (%)
Lab
µCT
TS average
He
He
He
He
20
10
0
e)
Fracture Density (m^{-1})
1000
500
0
f)
0°
270°
90°
180°
z
x
y
2%
0
g)
Permeability (m^2)
10^{-14}
10^{-15}
10^{-16}
10^{-17}
0
25
50
Confining pressure (MPa)

The overall trend in the porosity comparison is that the µCT results show the lowest porosity, followed by the laboratory results, and finally the thin section results. This is a distinction that can be used to quantify the different contributions to porosity in the sample. First of all, the µCT results only show the porosity in the fracture network and pores above the limited resolution of µCT (12.5-20 µm voxel size). The structures above this resolution limit are likely to be the main contributors to permeability in the samples, as connectivity calculations on the µCT data (Voorn et al., 2013) show most extracted fractures and pores are indeed connected from top to bottom in the samples. An exception to this are the two cataclasite samples that show no or very low connectivity above this resolution, so for them permeability may be controlled by smaller-scale porosity. Second, the lab porosity displays higher values than the µCT results, because the "resolution" of these laboratory methods is better than for µCT; i.e. smaller porous structures can be reached with these laboratory techniques. However, the lab values portray connected porosity only, since only porous structures that can be accessed by a fluid can be used in the standard laboratory methods. However, for the lab measurements – if dry mass and sample dimensions are available – theoretical non-connected porosity can be estimated by assuming an average density for the rock forming minerals. When taking pure dolomite, this results in a fraction for connected/non-connected porosity of 0.6-0.95 (4 samples). Third and finally, the thin section SEM-BSE results show the porosity down to a much better resolution (imaging resolutions between $(1.1\ \mu m)^2$ and $(1.7\ \mu m)^2$ per pixel) than for µCT. Note that only full stitched overviews of thin sections have been taken, i.e. using the largest possible surface area for the porosity determination. If more thin sections were available for a sample, the average of their results has been taken. Contrary to the laboratory methods thin sections actually include non-accessible (i.e. non-connected) porosity as well. The values for thin section porosity are therefore the highest, and the connected/non-connected porosity fractions appear to match the theoretical values from the lab measurements. Note that the thin section creation may have led to some additional sample alteration and porosity change, hence some caution on values obtained by this method is in place. Also, one should keep in mind that whereas the lab and µCT porosities are inherently 3D, the thin section results are not. Still, using thin sections (or any other imaging technique at a better resolution than full plug µCT), the complete connected and non-connected porosity and structure of the samples can determined, which may be important for example for enhanced oil or gas recovery.

F.5.2. Aperture

For aperture distributions, the comparison between the different samples is of main interest, and therefore shown in the separate Figure F.6. The same symbols are used for the same samples in other diagrams (see Table F.2). The general trend in literature for both single fractures and fracture networks, derived from different methodologies (e.g. CT, optical profiling, numerical modelling), is that aperture distributions show a log-normal distribution (e.g. Keller, 1998; Montemagno & Pyrak-Nolte, 1999; Karpyn et al., 2007; Sahimi, 2011). A skewed log-normal distribution can indeed fit the aperture distributions shown here, although also an exponential decay curve could be fitted. The main limiting factor for the aperture distributions from µCT is however the lack of information at the lower end (i.e. the smallest apertures), making a decision on the distribution difficult, but we argue that we most likely deal with similar fracture aperture distributions as shown before in literature.

In terms of fracture aperture (Figure F.6), the Schönkirchen T91-5p sample, which is a mixture between a mosaic breccia and pervasive fractures, differs from the other samples. This sample has the largest open pores of all samples of this study (readily visible in the second column of images of Figure F.5b and c), and this comes out clearly of the aperture determination. The other samples are relatively similar to each other. Because of this similarity, aperture distributions may not be very useful in automated routines for sample characterisation (i.e. division between a fractured sample from the damage zone and a cataclasite from the fault core). However, since the local thickness approach also provides us with a value for aperture everywhere within the 3D image data, these results may be very useful in permeability modelling, or as an input parameter for example for Discrete Fracture Networks.

A final important observation on Figure F.6 is the consistently lower aperture distributions of the 2 cm diameter plug samples (open symbols). Of course, some sample heterogeneity is expected to play a role in the difference to comparable 3 cm plug samples (i.e. samples plotted with equal symbols, but filled). We however think the consistently lower aperture is an effect of µCT resolution and blurring within the µCT data, meaning the real sample apertures may be somewhat lower, and hence shows an important drawback of using aperture information from µCT. This effect of sample size shows that to be able to compare different samples in the best way in future studies (possibly in automated routines), it is recommended to use similar samples sizes throughout. Another option would be to use a calibrated routine that takes µCT blurring into account (e.g. Ketcham et al., 2010; Ketcham & Hildebrandt, 2014), but as mentioned before, such routines may still be hard to apply on complete fracture networks.

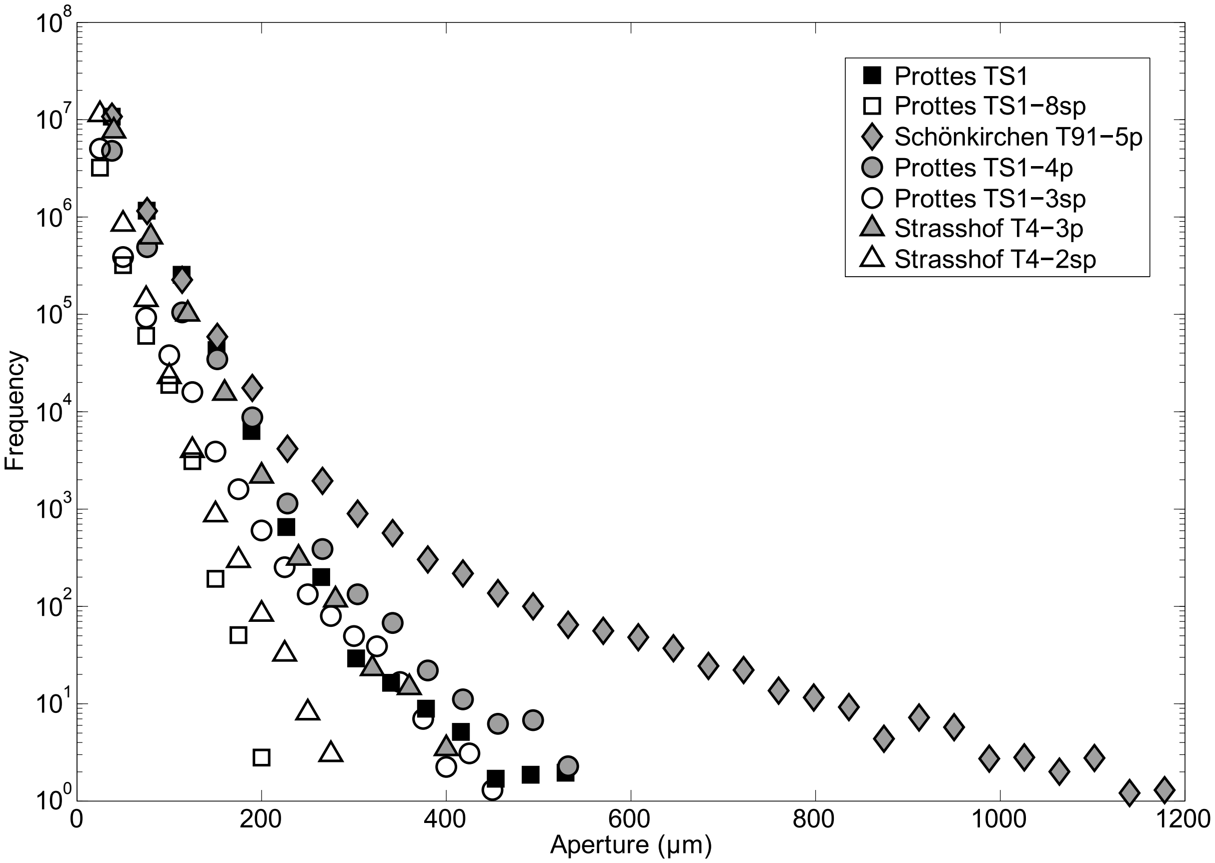


Fig. F.6. Aperture distributions determined from µCT data. Shown here are normalised aperture values (compensated for sphere volume in the Local Thickness approach) in µm. The symbols used are the same as in other figures – Prottes TS1 is plotted with a distinct symbol filling because this fracture sample is not represented in the permeability diagrams (see Table F.2). Most samples have a relatively similar aperture distribution, but brecciated and fractured sample Schönkirchen T91-5p clearly stands out, having large open pores in the µCT data. Another important observation is that the 2 cm diameter samples (open symbols) consistently have a lower distribution than the similar 3 cm diameter samples.

F.5.3. Fracture density

Figure F.5e shows results on fracture density, using the aperture distributions as described in Section F.3.1.3.3. The fracture density values provide an independent parameter to characterise the rocks – possibly also in an automated way. For these samples, a rather good correlation exists to the µCT porosity results, but it should be noted this correlation is not constant for the different samples (i.e. fracture density/µCT porosity ≠ constant or related; see Table F.2). This is because the aperture distributions for the different samples are rather similar (see Section F.5.2), meaning that with a higher porosity also more fractures – i.e. a higher fracture density – must be present. For series of samples with more varying aperture distributions, the fracture density should show less correlation to porosity. For exploration purposes, the fracture density can be an important parameter, especially when implemented into a model, combined with several of the other parameters derived in this study.

F.5.4. Fracture orientation

Some examples of the fracture orientation distribution are shown in Figure F.5f, normalised to the same colour scheme throughout the samples. The fracture orientation is the last parameter that we could extract using the available µCT data. As a general trend, some clear and distinct fracture orientations are visible for the samples that are only fractured, the furthest away from the fault core. For the brecciated samples, the orientations are more spread out and less strong than for the fractured samples. Finally, for the cataclasites, the fracture orientations that could still be extracted show a large spread and no or only a very weak dominant orientation, as expected.

The results from the orientation determination can be used in (automated) classification schemes when applied on multiple samples. Of course, the comparison of the drillcore orientation to fracture orientation data from other sources (e.g. FMI, if the resolutions of the techniques overlap, or to determine if similar structures are present at different scales of observation) can be useful to put the samples into perspective. Unfortunately, in this study the samples were taken randomly from the drillcores, making such a relation to different scales not feasible.

F.5.5. Potential other parameters from imaging data

Several other parameters would be interesting to extract from µCT datasets, but attempts for the samples shown in this study were unfortunately not successful. For completeness and possible application in other studies, we mentioned them here briefly.

Fracture spacing is a very important parameter in for example Discrete Fracture Network modelling. For reasons explained before, it is complex to extract proper, representative distinct fractures from the complete fracture networks shown in this study. We attempted a filtering by fracture orientation (using the orientation data of Section F.5.4) which was partially successful for some of the samples, but not sufficient to extract proper fracture spacing data. For samples with more consistent, parallel sets of distinct fractures, such an approach could however be useful. The same arguments for fracture spacing are also valid for fracture length.

Fracture roughness can be of interest for flow models, since the roughness of fracture may have large consequences on the permeability and the flow mechanism (e.g. causing turbulent flow) through a sample. µCT data can potentially provide information about fracture roughness, but the resolution and fracture apertures in our samples are not sufficient to get a good result. If fracture roughness is a necessary parameter to determine, we would advise to use thin sections (2D), better resolution µCT data on smaller samples (3D) or different techniques such as optical profiling (2D or 3D).

Self-similarity and self-affinity of fractures and fracture networks can be useful in upscaling data on small samples to larger scales, and potentially even up to reservoir scale. We have attempted a box counting technique on the fracture network for our samples, but the results are inconclusive. The main reason is the lack of scales available within a single sample, and the limited resolution. If a box counting technique is applied, more scales (e.g. smaller and larger samples) should be included as well. However, because of the previously discussed problems with using smaller (for example creating them without destroying the sample, and picking the right areas) and larger samples (for example insufficient resolution for viewing the fractures), we have not been able to apply this successfully in this study.

F.5.6. Permeability experiments

5.6.1. Loading only

Figure F.5g shows the results from the gas and water permeability experiments under confining pressure, separated into four graphs of different rock types. The flow has only been measured parallel to the plug axis. For the 3 cm diameter plugs, the available gas permeability experiments without confining pressure (and without correction, so a 25% error margin should be regarded) are shown as black stars for comparison. We argue when comparing the permeability results in Figure F.5g to the other determined parameters, there are some correspondences, but no clear direct links, so for example no simple porosity-permeability relationship that could cover all samples. This lack of a porosity-permeability relationship is show in Figure F.7, for different methods of porosity analysis.

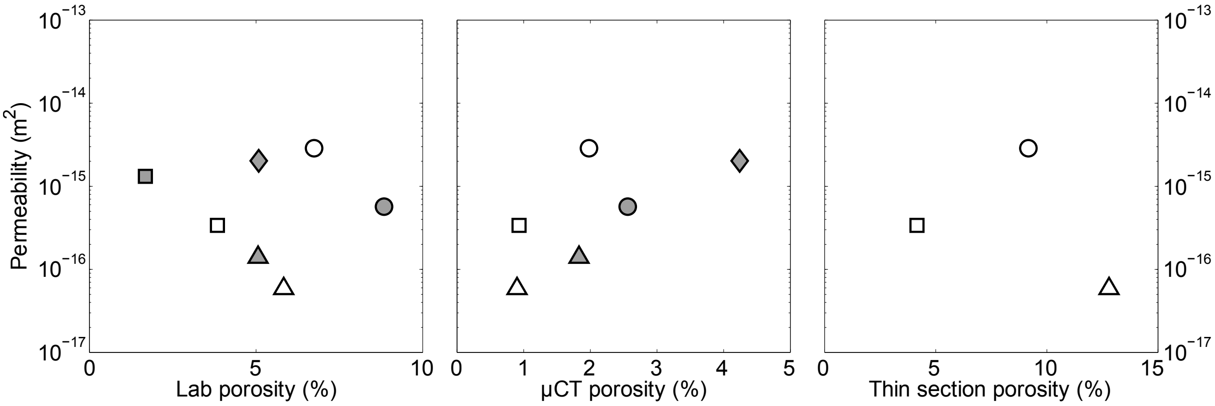


Fig. F.7. Porosity to permeability relationship for the samples shown in this study. The same symbols have been used as in the other figures (see Table F.2). The permeability values are taken at the lowest pressure measurement in the experiments under confining pressures. The porosities chosen are from a) laboratory (helium porosimetry and water saturation porosimetry), b) µCT bulk values (imaging resolutions between (12.5 µm)^3 and (20 µm)^3 per voxel) and c) thin section SEM-BSE porosities (imaging resolutions between (1.1 µm)2 and (1.7 µm)2 per pixel, full thin section overviews only). For all three plots, the data are scattered, meaning there is no simple porosity-permeability relationship: not for similar rock types and definitely not for all rock types together. The best correlation is visible for the µCT porosity data, but with still too much scatter to argue for a single porosity-permeability correlation.

The general trend when comparing the permeability results during loading on all samples is that first of all the various breccias (middle two columns in Figure F.5g) show the highest initial and remaining permeabilities with increasing confining pressure. This can be explained by the complex variations in fracture orientation and apertures. We propose that because of this geometry, it is difficult to close the open porosity structure in an efficient way. Second, the samples belonging to the damage zone with predominately fractures (left column in Figure F.5g) show somewhat lower permeabilities, and relatively the largest decrease in permeability with confining pressure. This can be explained since the geometry of the fracture network is a bit simpler (e.g. there are clear, dominant fracture orientations) than for the breccia samples, so the fractures can close more efficiently. Third and finally, the cataclasite samples (right column in Figure F.5g) show the lowest permeabilities.

Regarding the drop with confining pressure, very different behaviour is observed for the gas and water experiments for these cataclasite samples. The gas permeability experiments show a fast initial drop, followed by virtually no change in permeability. This can be explained by the large spread in orientation of fractures and the overall low porosity in the sample. For the water permeability experiment however, a much more severe drop is observed. This may be true behaviour, meaning the geometry of fractures in this sample is apparently suitable for a quick and efficient permeability reduction. However, there is also a chance that at these low permeability values, some non-steady state flow effects were still acting during the experiment. Another option is that the sample has been altered due to sample preparation before performing the experiments. In summary, the results of the water permeability experiment on this cataclasite sample are the least convincing of all experiments, but are still shown here for comparison.

Regarding the differences between the gas and water permeability experiments on similar samples, there is no clear trend observable. Because of this, we cannot interpret the possible differences between gas and water permeability measurements in detail. Although there is a chance gas permeability measurements could consecutively lead to higher permeability values than water permeability experiments (as for example observed by Tanikawa & Shimamoto (2006) and Faulkner & Rutter (2000) on different rock types), there is no way to know this for certain when not performing experiments with both pore fluids on exactly the same sample. In the case of the results presented here, there is an at least equally large chance we only observe the heterogeneity present between the different samples.

For the gas permeability experiments without confining pressure (black stars, non- corrected) compared to the water permeability experiments on the same samples, there is also no clear trend, but a large error on the non-corrected values should be regarded. These results mainly show that on any permeability measurement, a large possible variation should be kept in mind. Measured permeabilities can thus definitely not be considered as a single true value for permeability for the rock (or fracture class type).

When taking all experiments into account, the general trend is that most permeability drop in these samples seems to be accounted for before reaching 20 MPa. For the gas permeability experiments, this can immediately be seen, since in these experiments the trend through the data “flattens out” quickly (i.e. the permeability stays nearly constant) with increasing confining pressure. It is reasonable to assume a similar trend line for the water permeability experiments. Most likely, the fast drops in permeability at relatively low confining pressures represent a fast closing of open pores/fractures until asperities prevent further closing. Probably, also some elastic effects are at play in the samples. Some interesting observations can be made about these processes when also regarding unloading and reloading of samples, which will be discussed in the following Section F.5.6.2.

The reported pressures in the diagrams are effective confining pressures, but the translation to reservoir depths – without knowing the measured fluid pressures in the reservoirs (data not available from drilling), is difficult. However, it is expected that the reservoirs are definitely at higher pressures than the 20 MPa before which most permeability drop is accounted for. In future studies, if the stress field in the reservoir is known, it may prove to be more useful to apply a proxy to such a stress field than only an isotropic confining pressure. Since we did not possess that information here, a confining pressure was the simplest and easiest to apply.

F.5.6.2. Loading and unloading

In the permeability experiments with water as pore fluid, also the permeability response to unloading and reloading (confining pressure) was investigated. Figure F.8 shows the results of one series of measurements on fractured sample Schönkirchen T32-430, which contains one dominant fracture. The sample was first loaded in steps up to around 10 MPa confining pressure, then unloaded to 5 MPa, and then consecutively loaded up to 20 MPa confining pressure. Finally, the sample was unloaded to near-atmospheric conditions. Clearly, the order of loading and unloading steps has an effect on the measured permeability values. Following for example an exponential decay curve, one could argue that continuous loading beyond the maximum intermediate confining pressure would lead to results such as shown by arrow a in the figure. In a similar manner, one could argue that with only immediate unloading after maximum confining pressure, behaviour such as for arrow b could be expected (this is observed in other samples). However, the chosen paths of loading and unloading lead to different results here: a lower permeability at 20 MPa than for arrow a, and a lower permeability after final unloading than for arrow b.

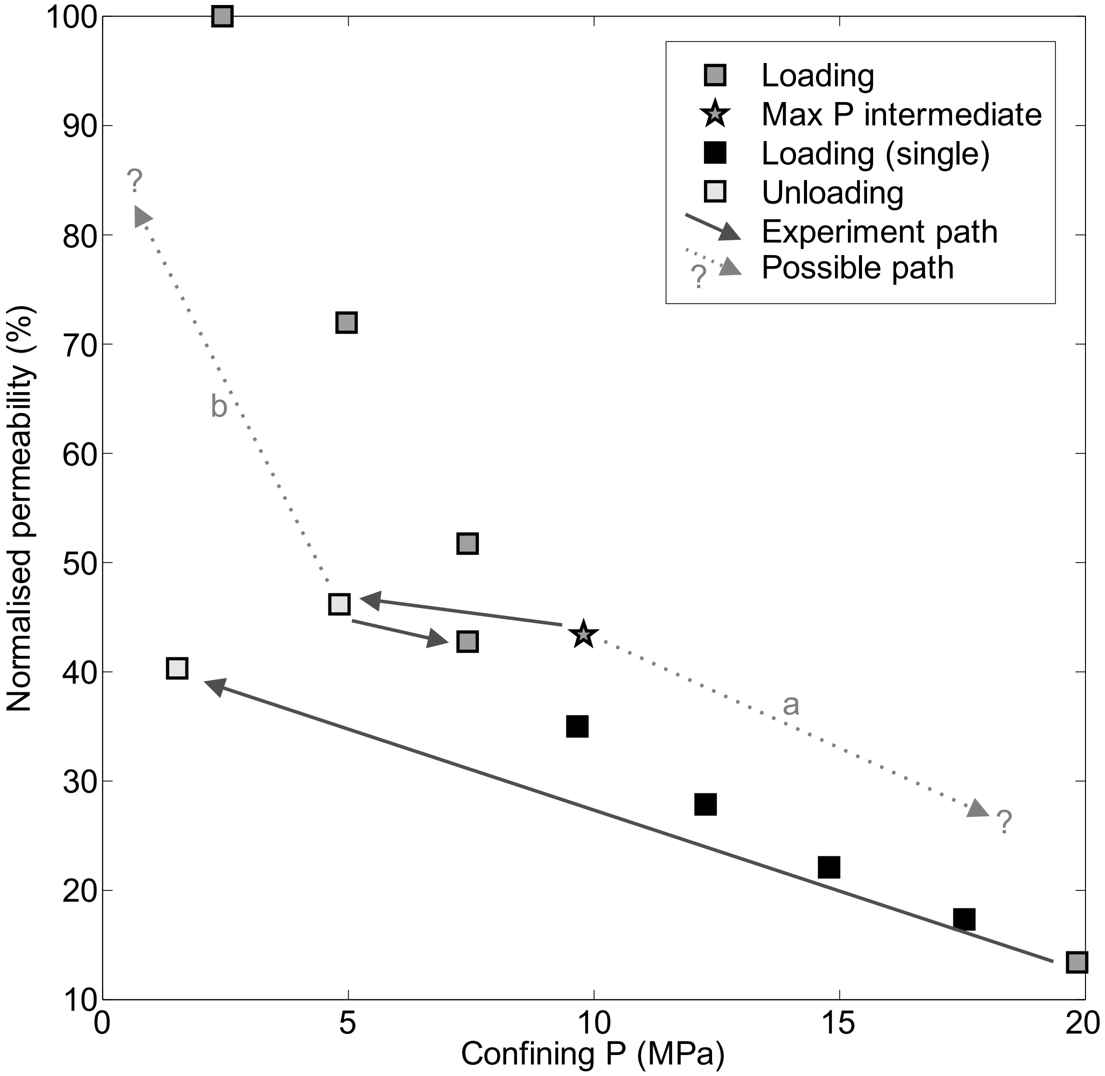


Fig. F.8. Permeability results on a single fracture dolomite plug sample, during various loading and unloading stages with water permeability. The order and choices made in loading and unloading have an influence on the finally obtained results. A normalised measure for permeability was used to allow comparison to the loading and unloading behaviour shown in Figure F.9f. Note that in the diagram, the dark-grey filled data points ("Loading" and "Max P") correspond to the grey-filled squares in the leftmost diagram of Figure F.5g. Furthermore, not all plotted points represent a series of minimum 4 measurements with different flow rates: the black filled symbols marked "Loading (Single)" only represent one measurement at one flow rate, but are still shown to get a better appreciation of the loading path between 7.5 and 20 MPa. After loading to an initial maximum intermediate confining pressure, the sample was unloaded. If loading would however have continued, the path shown by arrow a could be expected. If unloading would have continued after the first unloading step, possibly path b (as seen for the other samples) may have been followed. The real path of reloading and final unloading is however shown in the datapoints.

The dependence on the actual loading- and unloading path is something that is observed often, in various rock types (e.g. Faulkner & Rutter, 2000; Selvadurai & Glowacki, 2008). The behaviour described by Selvadurai & Glowacki (2008), where the confining pressure during the second loading step first has to exceed the intermediate maximum pressure, before extra permeability reduction can occur, is not observed here. Cycling of loading and unloading until no path-dependent behaviour is visible anymore – such as in Faulkner & Rutter (2000) – would be an experimental possibility. We however think such a minimum permeability measure is not relevant for our reservoir samples, because the samples have already undergone unloading when taken from the reservoir. The minimum permeability measurement after cyclic loading and unloading is therefore not likely to be a better representation of actual reservoir permeability conditions than the path-dependent measurements we did perform.

Several reasons could cause the path-dependent behaviour we see in these experiments. For example, consecutive partial destruction of weak asperities or small shear movements along cracks could be acting, and also residual other fluid phases might play a role. Also permeability reduction by particle blockage, where maybe new particles become available during unloading could play a role, although our flow direction inversion experiments (Figure F.4c) suggest particle blockage does not play a major role in these samples. The actual mechanism causing loading-path dependent behaviour can thus not be determined by these experiments only, but it is important to realise such results put even larger uncertainty on single permeability measurements for a sample. Furthermore, since these experiments show that loading and unloading cannot be simply inverted (but in fact show hysteresis-like behaviour), the permeability determined at maximum loading may not reflect actual reservoir conditions well, since the sample has already been unloaded during drilling and recovery (not even taking pore fluid pressure effects in the sample into account). These permeability experiments under confining pressure may therefore provide an approximate order of magnitude in reservoirs, but actual reservoir permeability when the sample was still at depth could be quite a bit higher or lower.

F.5.6.3. µCT after permeability experiments

For a single sample (Prottes TS1-4p), a µCT scan was made before and after a water permeability experiment under confining pressure. By registering (matching) the two datasets in 3D, a comparison between the two can be readily made. Figure F.9 shows the results at a two-dimensional slice through the two datasets, as well as the loading-unloading curve for this sample. When overlaying the two datasets, virtually no difference can be observed. Only a few small differences are visible at the planar edges of the sample (not shown in Figure F.9), but these are most likely related to sample handling after the permeability experiments. These µCT results show that the changes in the sample (note there is an absolute change in permeability between the start and end-points of the permeability experiments) must be below the resolution of the µCT scan. This means, the permeability reduction may for example be accomplished by aperture reductions below the µCT resolution, or changes in pores which are not observable by µCT. If a visualisation of the permeability reducing effects is required, a technique with a much better resolution should thus be used. Another option to obtain even more detailed information would be to image the sample using CT during compression and flow (as done by e.g. Watanabe et al. (2001)). Such a setup is however complex, and not possible to apply within the scope of this study.

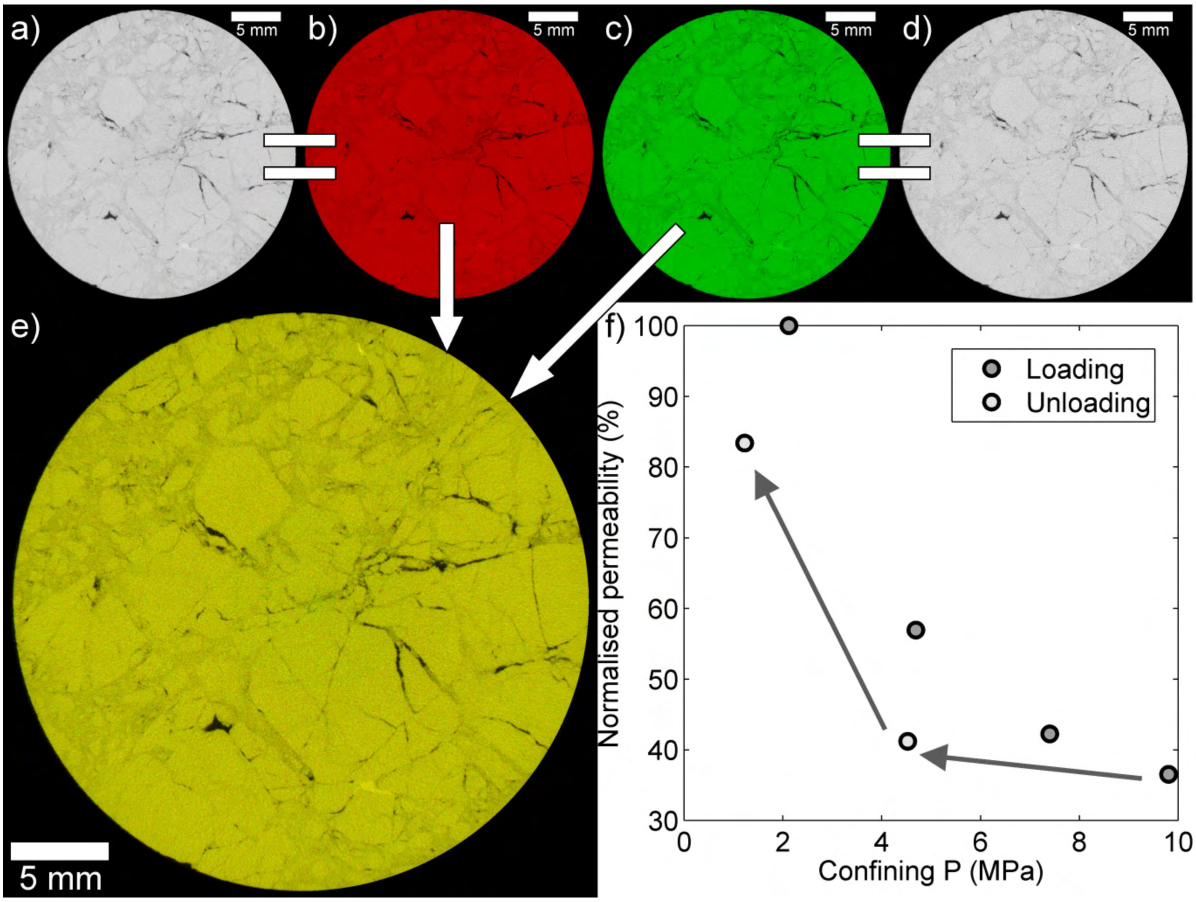


Fig. F.9. µCT scans before and after a permeability experiment under confining pressure, on a 3 cm diameter plug. Contrary to the µCT scans shown before, these scans were made at the Faculty of Civil Engineering and Geosciences at the Delft University of Technology, The Netherlands, using a Phoenix Nanotom µCT, and reconstructed with a resolution of (15 µm)3 per voxel. The scans before and after have been 3D registered to obtain a perfect match. In the examples, 2D slices of matching areas through the full 3D µCT datasets are shown. a) µCT scan before permeability experiment. b) Figure (a) with intensity set as red channel only. c) Figure (d) with intensity set as green channel only. d) µCT scan after permeability experiment. e) RGB combination of figures (b) and (c). If clear differences between the scans would exist, they would show up as distinct green or red areas. However, all combined image information shows up as yellow, meaning no differences between the two scans can be observed. f) Loading and unloading behaviour of the sample. After unloading, the permeability is lower than at the initial stage. A normalised measure for permeability was used to allow comparison to the loading and unloading behaviour shown in Figure F.8.

F.5.7. Permeability models

The permeability results from laboratory experiments should ideally be connected to permeability determined from the imaging results. Here, we will investigate the use of a statistical method defined by Gueguen and Dienes (1989), where a model of cylindrical 2D cracks is proposed, which is expressed as follows in an isotropic case:

$$\kappa = \frac{4\pi}{15} f \frac{\bar{w}^3 \bar{c}^2}{\bar{l}^3} \text{ (Equation F.5.1)}$$

f is a factor between 0 and 1 depicting the fraction of connected fractures. $\bar{w}$ is the average half-crack aperture, $\bar{c}$ the average crack radius and $\bar{l}$ the average crack spacing. As indicated before, the complexity of the fracture networks in the datasets presented in this study does not allow an effective determination of crack radius (or fracture length) and fracture spacing. However, when taking a definition for porosity in the crack model into account, they derive the following from Equation F.5.1 (Gueguen and Dienes (1989):

$\kappa = \frac{2}{15} f \, \overline{w}^2 \Phi$ (Equation F.5.2)

Where Φ is the porosity fraction. Taking the geometric mean of the aperture, f=1 and the porosity from different methods (Table F.2), the results of the model permeability are very different relative to the measured permeability values, with overestimations of a factor (= $\kappa_{\text{model}}/\kappa_{\text{measured}}$) in the order of magnitude of 10^3-10^5 (when using µCT porosities, but similar estimates are found using porosity determinations by different methods). Even when taking a lower value for f, a lower geometric mean for the aperture (compensating for possible overestimation effects of the aperture), and the lowest possible porosity values, the overestimation of permeability by this model stays significant. Among the other reasons this particular model may not work are a) the models assumes isotropy for simplicity, and this is not likely for these samples; b) the variables in the model are statistically independent, whereas here, we observe larger apertures for larger fractures and c) in the model, connectivity is assumed to be by cracks only, but these samples are likely to have significant matrix connectivity.

We therefore argue that to obtain more realistic values for permeability from µCT data, more complex models than in Equation F.5.2 should be applied, or more likely, numerical simulations by for example Finite Element Modelling (e.g. Hoyer et al., 2012) or a Lattice Boltzmann approach (e.g. Degruyter et al., 2010).

F.5.8. Synthesis and recommendations

In the previous sections, the results for various parameters from several methodologies have been shown and discussed in detail. With the available image data, it is possible to visualise the fracture networks and porous structure in the rock samples (above the resolution of the imaging technique in question). Although many details can be extracted from thin section information, µCT imaging is the main source of information in this study, as it provides 3D datasets. On µCT data of a small set of 7 differently fractured dolomite samples, we have been able to extract bulk values and distributions for porosity, fracture aperture, fracture density and fracture orientation, while also being able to look at the local distribution of these parameters, and the relationships between them. Unfortunately, this bulk and voxel-based approach we have taken here does not allow us to separate the fractures into single features, preventing among others analyses on fracture spacing and fracture roughness. The big advantage is however that fairly little manual interpretation and -processing is required, making the shown bulk and voxel-based analyses suitable for an automation routine. Especially when samples of comparable composition and sample size are scanned with the same µCT scanner under similar conditions, an automated analysis with only a few manual steps could be implemented. Such an automated analysis could be suitable for example for an extended fractured reservoir study, where the core samples can be assigned to the different positions in a fracture zone interpretation using the various extracted parameters from the image data (Figure F.5). Another option would be to combine an extended µCT study with a fieldwork study in which a great deal of information is available, to improve both the results obtainable by µCT as well as the fieldwork interpretations. Such an interpretation may be useful on the reservoir scale if a “reservoir analogue” fieldwork area is chosen. The investments required for a large µCT study should however be kept in mind; i.e. it has to be possible to obtain important information that cannot be derived using other, less demanding techniques, or it should be possible to confirm (or disprove) or calibrate the outcome from different techniques by using the µCT results.

The data on the samples presented in this study show that small variations or heterogeneities can already have a large influence on sample parameters such as porosity and permeability. This is on the one hand an important result, since it shows that even rocks that have been sampled very close to each other may be very different in their (microscale) behaviour. No single sample can thus be considered representative for a large area. On the other hand, it also limits the use of the samples shown here, since the small sample sizes that were possible for µCT scanning do clearly not serve as a Representative Elementary Volume (REV). A related limitation on the REV is due to the sampling itself, since we can only use samples that do not disintegrate upon plug creation. On the other end of the scale, also very fine-scaled features (that may be of importance for e.g. hydrocarbon storage and transport) are not included due to the resolution limits of µCT, again limiting the possibilities of obtaining an REV with µCT only. To overcome these problems with REV, for example datasets of adjacent samples could be combined, or clever links to different-scaled data (for example combining the fracture orientation data from µCT with fracture orientations at hand-specimen scale) could be used. Such approaches for upscaling may be successful, but most likely requires additional modelling on the data.

For any upscaling or linking to the reservoir scale, the actual flow behaviour at depth is important. In this study, we have attempted to get a better view on the flow behaviour through these fractured dolomites under confining pressure. The results show that most permeability reduction is already obtained with only several MPa of pressure. The loading-path dependent behaviour of the permeability, the unknown actual stress field and pore pressures at depth, and uncertainties about the rock recovery from the reservoir however put limits on the use of such laboratory measured values for actual reservoir conditions: variations up to several orders of magnitudes can be expected. Still, these values and especially the responses to loading can serve as a very important calibration input in permeability models (e.g. Hoyer et al., 2012) that use the porous structure and other parameters extracted from µCT data.

F.6. Conclusion

The main question we tried to answer in this paper was how to efficiently analyse natural fracture networks in dense rocks. We have shown that 3D µCT imaging is possible on small (2 to 3 cm diameter) plug samples of narrowly fractured dolomites, and that this image data can be successfully processed using the MSHFF routine. Using this processed data, the internal fracture and pore structure of the samples can be visualised, and parameters such as porosity, fracture aperture, fracture density and fracture orientation can be extracted. µCT has important limitations (mainly on sample size and resolution), and can therefore not serve as a complete replacement for other analysis techniques. It can however serve as a valuable addition, especially when applied on larger sets of data. As an addition to µCT imaging, we obtained thin section data for more detailed analyses on the samples, as well as to obtain an additional measure for porosity. Finally, permeability measurements on the samples under confining pressure provide insights on the flow behaviour of the samples at depth, but also show that such measurements can only provide an order of magnitude approximation for the natural situation.

When combining the results on the samples shown in this study with a fault zone interpretation from damage zone to fault core, the differences between the samples become clear. Small variations in microstructure can have a large influence on the observable parameters. Fractured samples (the furthest away from the fault core) generally have relatively speaking intermediate porosity and permeability, and clear distinct fracture orientations. Brecciated samples (from closer to the fault core) have the highest porosities and permeabilities, and a wider spread in fracture orientations. Their fracture densities are high. Finally, cataclasite samples have the lowest porosities and permeabilities, low fracture densities and no fracture orientation preference. The analysis steps taken and the results shown here can be used in more extensive studies, and show that (partial) automation of sample analysis is possible. Moreover, results as the ones shown in this study can serve as input for modelling and simulations, where especially the obtained 3D information and the flow behaviour at depth are important parameters that may be difficult to characterise using other methods.

Acknowledgements

The authors would like to thank the OMV for the funding of the research and providing the samples presented in this paper. Ulrike Exner was funded by the Austrian Science Fund (FWF Project V151-N22). Permeability experiments in Strasbourg were conducted in the ÖAD WTZ (Amadée programme) project FR 05/2014. We would also like to thank Christian Gusenbauer (FHOÖ, Wels, Austria) for performing the main µCT scans and reconstructions, as well as Jan Etienne and Wim Verwaal (Delft University of Technology, The Netherlands) for their great technical support. Finally, we would like to thank Theresa Schröckenfuchs and Helene Bauer (University of Vienna, Austria) for their help with the fault zone interpretation and diagrams based on their fieldwork studies.

G. Additional thin section techniques

In this study, thin sections have been used to provide some details on dolomite texture and other components (e.g. clay and pyrite content) in the samples, which could not be resolved using µCT alone. The main results of these thin section analyses have already been summarised in Chapter F. However, several other analyses have been done on thin sections, and will be discussed below.

<u>G.1. Thin section overviews, image processing and porosity analysis</u>

The most important thin section information in this study is provided by full overviews of thin sections. On thin sections made after µCT scanning, such overviews are particularly useful for the visual comparison to µCT data. Initially, images were made by optical microscopy (both plane polarised light (PPL) and cross polarised light (XPL) used), and stitched together using the automated stitching software Microsoft Research Image Composite Editor (ICE; Microsoft Research, 2011). An example of the results is shown in Figure G.1.1b and c. Although these overviews can be useful for visual inspection, the details of optical microscopy – also because the full vertical thickness of the thin sections is integrated in the view – are not satisfying. Furthermore, optical microscopy thin section imaging is not suitable for porosity analysis here, because the samples have not been dyed in their preparation.

Scanning electron microscopy (SEM) with back-scattered electron (BSE) contrast is a more suitable technique, in particular for inspecting the porosity. The excitation volume on the sample is small in comparison to optical microscopy, meaning that a more realistic view of the thin section surface can be obtained. In order to make overviews of the complete thin section, adjacent images (taken with a constant shift relative to each other) were taken at a high enough magnification to prevent image distortion along the edges. This means for thin sections of 2 cm plug samples, at least several 100s of images need to be made. During such long operating times, the instability of the electron beam can lead to some variations in brightness and contrast throughout the set of images. Therefore, in a further processing step, these variations need to be corrected. Normally, automated stitching software such as Microsoft ICE should cope with these variations, but on the detailed SEM-BSE overviews, stitching by automated software was not successful (and also led to additional image distortion). Because of this, an own routine using MATLAB (The Mathworks, Inc., 2011) and FIJI (Schindelin et al., 2012) was implemented for image correction and image stitching by image translation only. The result of such a routine is shown in Figure G.1.1d and G.1.2b. For the comparison to µCT data, the image data is rotated, and in the case of a slightly inclined thin section, an inclined slice through the µCT data is taken. More details on these samples have also been discussed in Chapter F.

To determine the porosity values for every thin section, a threshold was applied on them. It should be noted here that, as for µCT data, an image does *not* equal a single porosity value! Choices need to be made and the final determined value depends on these choices. First some preprocessing was applied on the images. Dust imaged in the SEM is excluded from the images, or replaced by the average of its surroundings. The greyvalues of bright veins (e.g. calcite) are replaced by the average of the dolomite ground mass, as the threshold value between porosity (black) and solid material (grey), in which we are interested, is otherwise distorted. After this pre-processing, the Otsu threshold (Otsu 1979) was applied, which defines the threshold based on the image histogram, by minimising the intra-class variance of the finally defined two classes (porosity and solid). The choice for this specific method is somewhat ambiguous, but chosen as it is one of the most commonly applied techniques in literature, and often provides good results. A comparison of the value taken by the Otsu threshold relative to other thresholds on the image is shown in Figure G.1.3, and the final results on the available samples have been shown in Chapter F.

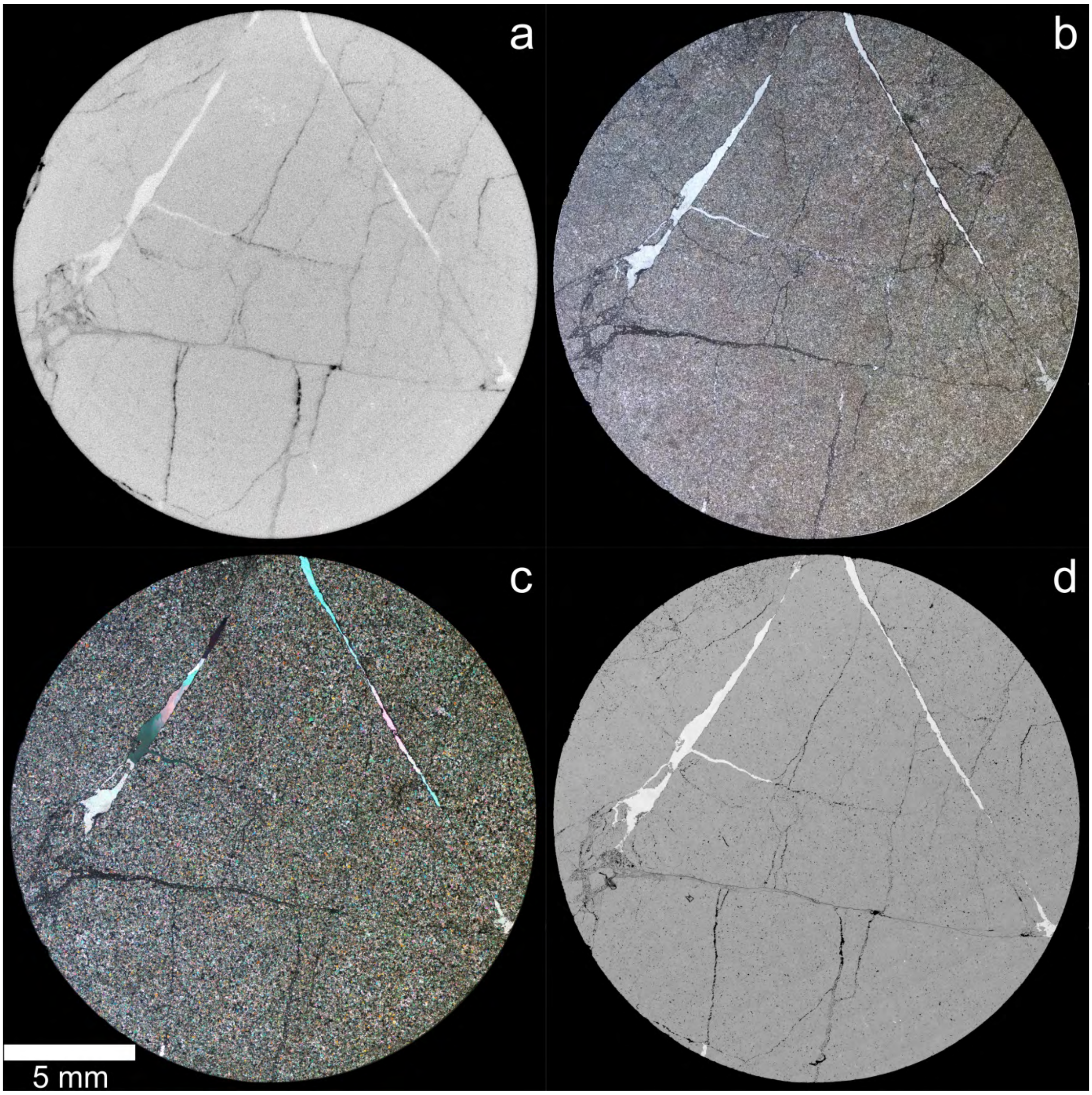


Figure G.1.1. µCT and thin section overview comparison for sample Prottes TS1-8sp (thin section A). The scalebar in the left lower corner is valid for all images. **a)** Approximately matched slice through the µCT data. **b)** Optical microscopy – plane polarised light (PPL) stitched overview. **c)** Optical microscopy – cross polarised light (XPL) stitched overview. D) SEM-BSE stitched overview. The separately taken images are mildly visible in the sample. The thin section results are more detailed that the µCT data and can be used for constituent analysis. More details are discussed in Chapter F.

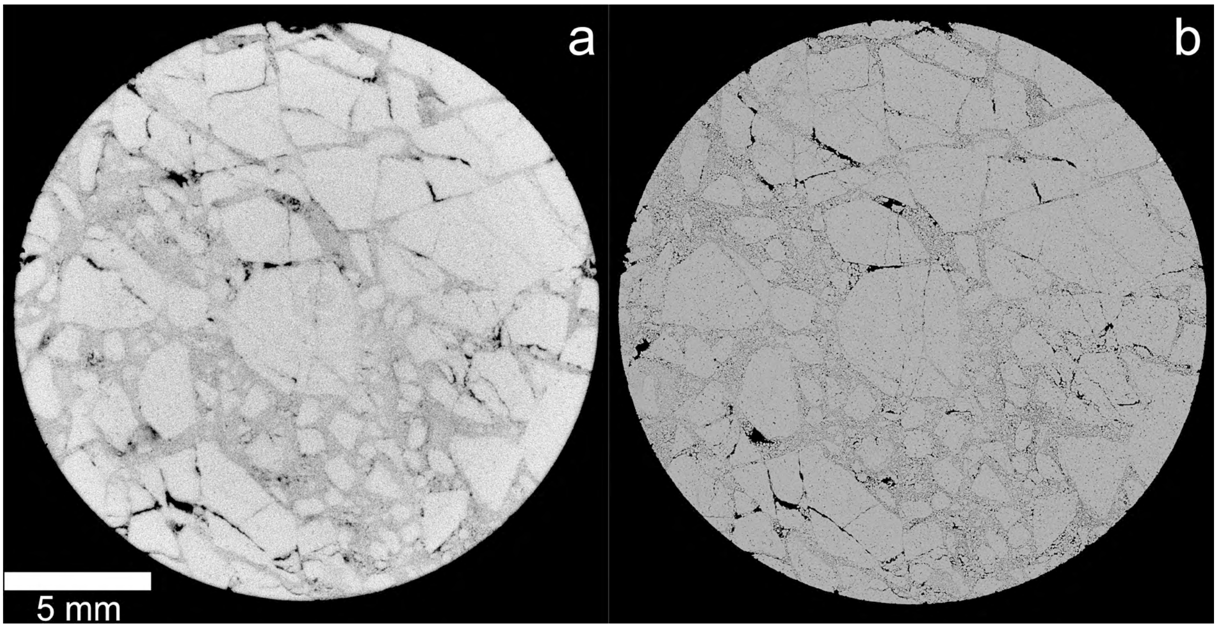


Figure G.1.2. Different example of µCT to thin section overview comparison for sample Prottes TS1-3sp (thin section A). The scalebar in the left lower corner is valid for both images. **a)** Approximately matched slice through the µCT data. **b)** SEM-BSE stitched overview. Especially the finer grained (near-cataclasite) areas in the sample show clearly more detail in the thin section overview.

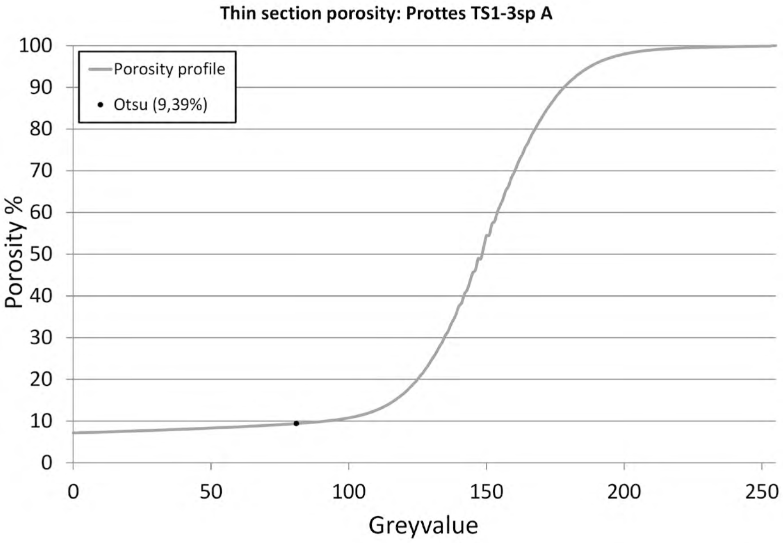


Figure G.1.3. Porosity curve example for a thin section SEM-BSE overview. The porosity value that would be obtained with the threshold at a certain greyvalue is plotted with the grey line. The Otsu threshold (see text) is indicated using the black dot. It is easy to imagine, a range of values around this Otsu determined threshold would probably also yield good segmentation results, but would give a slightly different porosity value.

G.2. Cathodoluminescence (CL) microscopy

Cathodoluminescence (CL) is a phenomenon in which visible light is expelled from a target due to electrons hitting it. The signal, i.e. the frequency and intensity of the generated light, is dependent on the elements in a sample, which are often minor and trace elements. For carbonates, the main elements controlling the CL signal are Fe and Mn. In thin sections, CL can help to visualise growth zoning, cements, matrix, and vein generations due to small chemical differences between adjacent areas. With normal optical microscopy, this is generally not possible. SEM-BSE imaging might be able to grasp some of these small chemical variations in some cases, but the signal and detail are often much stronger in CL imaging.

The source for CL data here is so-called hot-cathode optical CL microscope at the Department of Lithospheric Research of the University of Vienna. In such a device, the thin section sample is mounted upside down in a sample chamber. The sample chamber is put to a vacuum, after which the sample is targeted from below with electrons from an electron gun. Interactions of the electrons with the sample generate the CL signal. This CL signal passes through the sample glass towards a conventional optical microscope fitted with a camera mounted above it. Finally, images are recorded on a computer. One can choose to switch between CL mode and normal optical microscopy mode to be able to compare the datasets. A disadvantage here is that the obtained resolutions are not very good since all light has to pass through quite some materials due to the upside down mounting of the sample in the sample chamber. The finally recorded image is therefore an integration of the CL signal from all the material the rays passed through (“3D effect”). For this reason, samples need to be relatively thin (thinner than the 30 µm of common optical microscopy thin sections). Unfortunately, this was not the case for most thin sections made after µCT scanning. For these reasons, CL has not been used widely in this study, but some examples from other thin sections are shown in Figure G.2.1 to show the power of the technique for investigating fracture generations and cement forming. If a combination with µCT data would have been possible, such CL could be helpful in distinguishing fracture generations, although care should be taken in the application of such 2D data on the 3D µCT datasets.

In principle, it is also possible to use CL on some scanning electron microscopes. In such a setup, the advantage is that the beam does not have to pass through the entire sample, so that higher resolutions can be obtained compared to hot-cathode optical CL. A disadvantage is that colour information (in the available setup at the University of Vienna) is only recordable by combining the intensity-signal of differently colour-filtered detectors, leading to some colour ambiguity. Also very long scanning times and distortion due to calcite content in the samples were at play during this study, which made it not possible to produce viable SEM-CL data here.

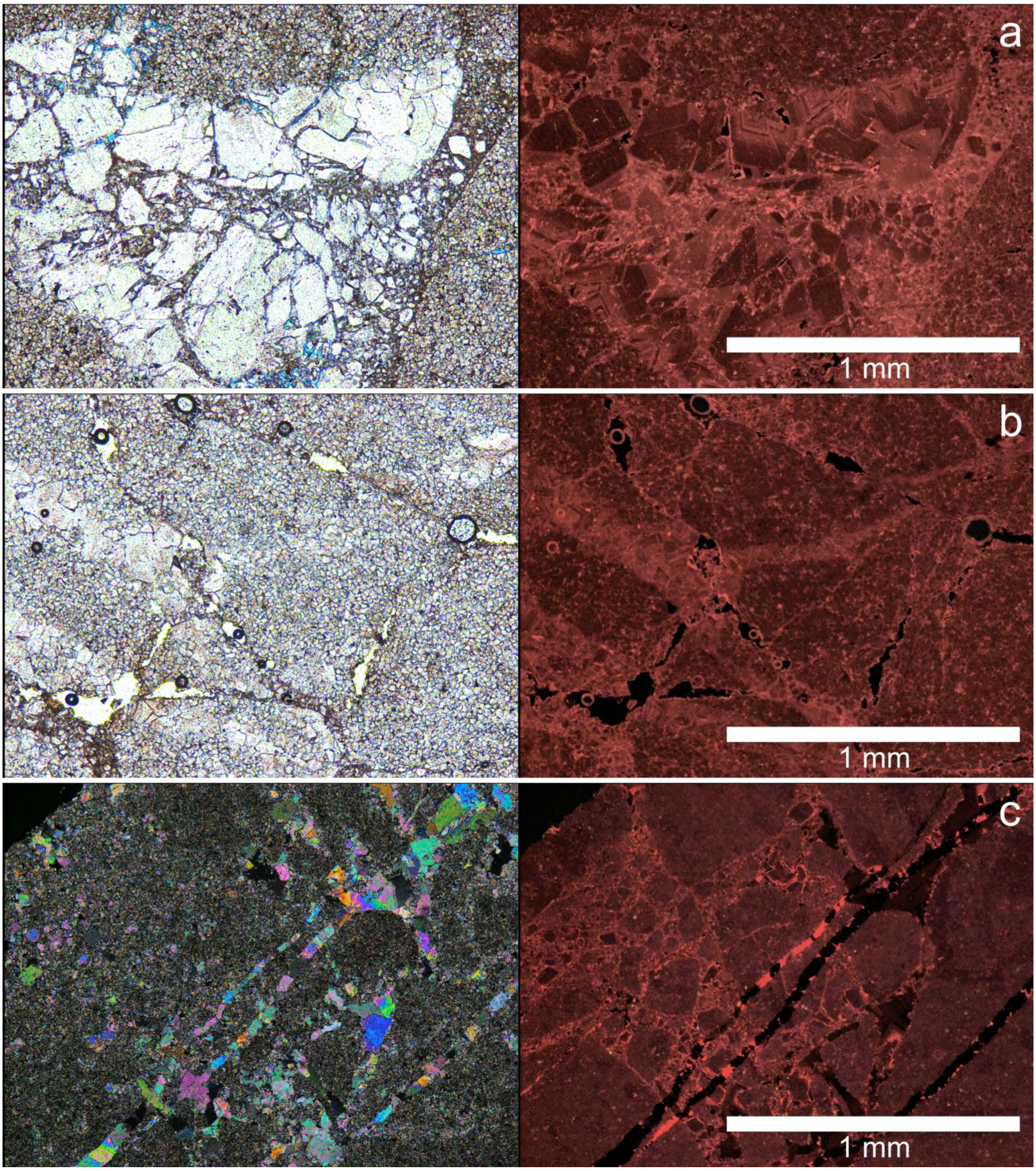


Figure G.2.1. Examples of optical microscopy (left side) and hot optical CL (right side). **a)** PPL – CL pair in thin section STRT4-1-3320,10. The larger crystals are part of a remnant vein in the host rock. In CL, the different crystal structure of the vein versus the finer crystalline host rock can easily be seen. **b)** PPL – CL pair in the same sample as figure a. In CL, several veins are visible that cannot (or only very difficult) be observed in the PPL image. Some veins can be traced into neighbouring objects, showing a breccia-like behaviour. **c)** XPL-CL pair in thin section SKT32-6. The CL image clearly shows fragments in a finer matrix that are not easy to observe in the XPL image. Furthermore, several veins are clearly visible in the XPL image, but their chemical composition differs, as witnessed by the bright versus the dull corresponding responses in the CL image.

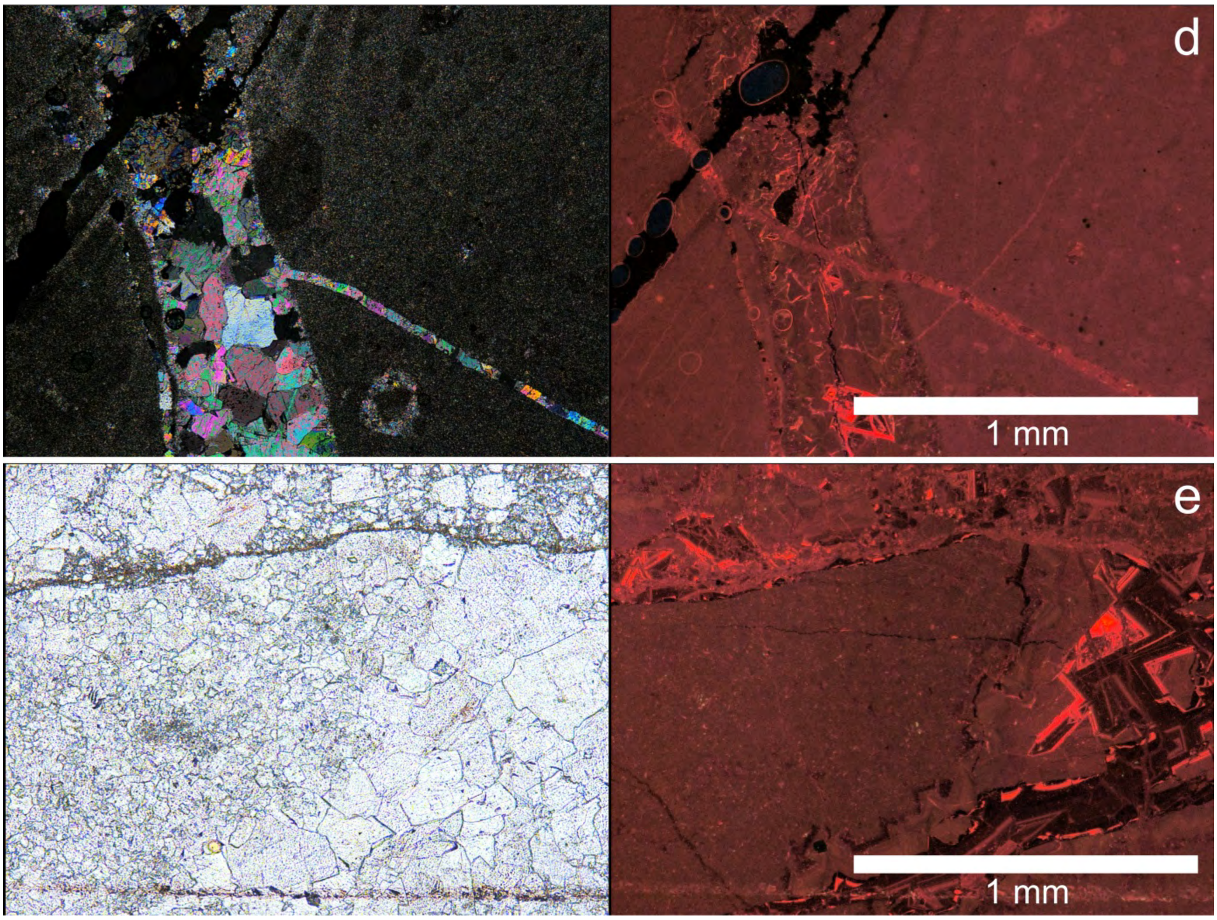


Figure G.2.1. (continued). d) XPL-CL pair in thin section SKT38CB. In XPL, a small vein coming from the right appears to terminate on the larger vein. In CL, one can however clearly observe the thin vein seems to continue within the large vein. Also additional veins can be observed in CL relative to the XPL image. **e)** PPL-CL pair in thin section SKT38SS. Euhedral crystals that are larger than the surrounding rock can already be observed in PPL. In CL, the growth and cementation phases of these euhedral crystals can be observed (growth zoning).

G.3. Focussed Ion Beam – Scanning Electron Microscopy (FIB-SEM) tomography

G.3.1. Introduction and data acquisition

Focussed Ion Beam - Scanning Electron Microscopy (FIB-SEM) is a technique by which µm scale perpendicular sections can be made on the surface of a thin section, which can be imaged and analysed. A Focussed Ion Beam (FIB) column is quite similar to a Scanning Electron Microscope (SEM) column, with the difference that an SEM uses a focussed beam of (negatively charged) electrons for imaging samples, whereas a FIB uses (positively charged) ions. Using an SEM on a sample is usually not or only to a very limited extent destructive for the sample surface. The greater mass and energy of ions when using a FIB on the other hand has the effect that material from the sample is expelled, a process called sputtering. With high beam currents, a lot of material can be removed by sputtering. This property allows one to very precisely "dig holes" on the µm scale in a sample, a process termed milling. By consecutive more precise milling, one can create a smooth perpendicular section to a thin section surface. FIBs are also used for other micromechanical work, for example for making so-called lamellae for Transmission Electron Microscopy (TEM).

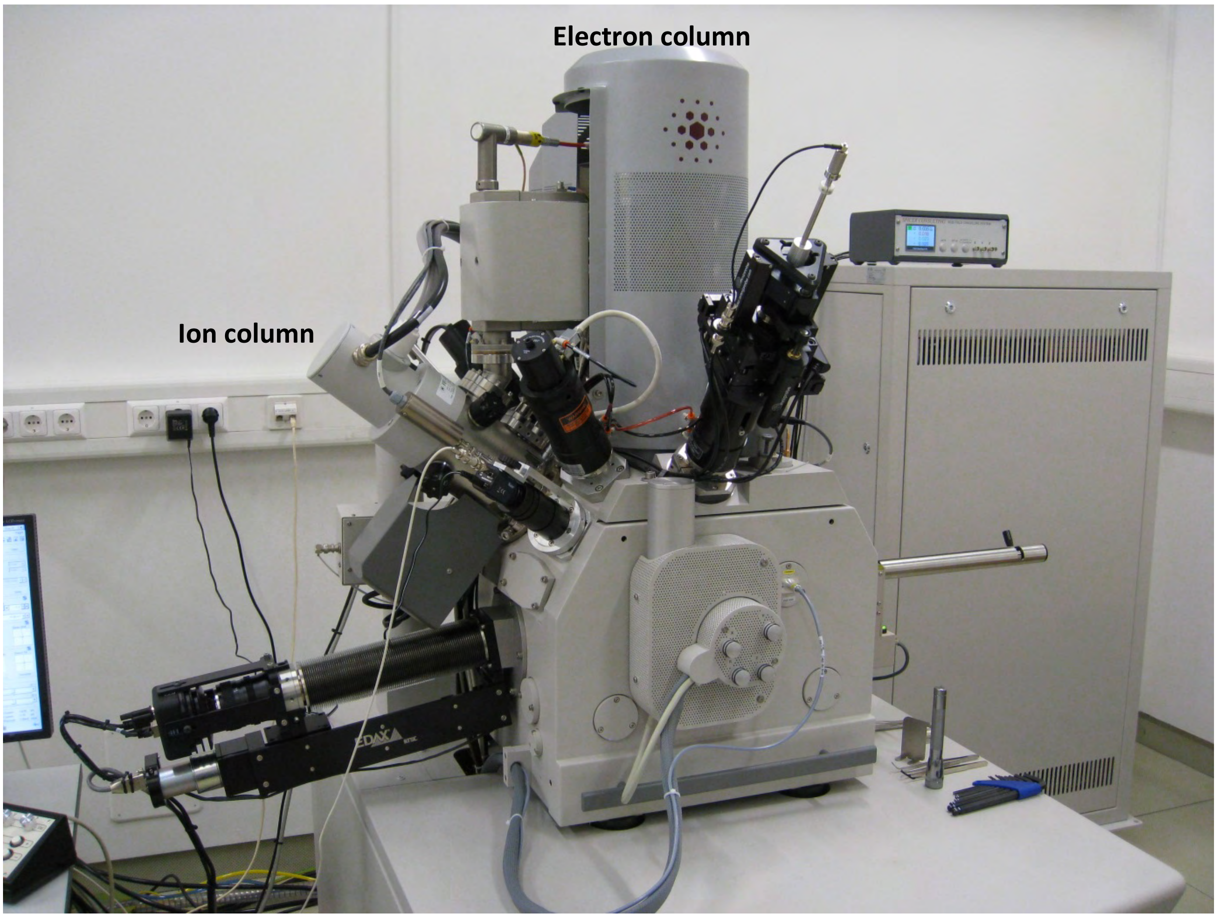


Figure G.3.1. The Quanta 3D FEG DualBeam at the Faculty of Earth Sciences of the University of Vienna. The ion column and electron column are indicated separately. Other detectors and manipulators are clear as well. The front of the image shows the sample door, behind which the sample chamber lies.

At low beam currents the amount of sputtered material is limited when using a FIB, and imaging can be done with for example secondary electron and/or secondary ion detectors. However, the FIB's destructive nature on the sample is not desired for making high quality images of a perpendicular section. An often seen combination is therefore a FIB-SEM, in which both an ion column and an electron column are implemented. In such a setup, the FIB is used for milling only, although some occasional quick imaging using the FIB is still required. The SEM is used for identifying regions of interest, checking the milling process, and high quality imaging of the thin section surface and the generated perpendicular sections. For SEM imaging, samples need to be coated with a conductive layer. This conductive layer is unfortunately also destroyed when using the FIB. However, the Gallium (Ga^{3+}) Liquid Metal Ion Source (LMIS) used in the Quanta 3D DualBeam FEG at the Faculty of Earth Sciences of the University of Vienna (Figure G.3.1) has the advantage of implanting Gallium ions in the surfaces generated by the FIB. This acts as a new conductive layer, which reduces the effects of charging when imaging with the SEM.

Since two particle columns have to be used in the same device, they are tilted with respect to each other. In the Quanta 3D, the electron column is vertical, and the ion column is positioned at an angle of 52° to it. The sample thus has to be tilted inside the vacuum chamber to align properly for either milling with the FIB, or imaging with the SEM. Also see Figure G.3.2 for a schematic view of FIB-SEM at the sample surface.

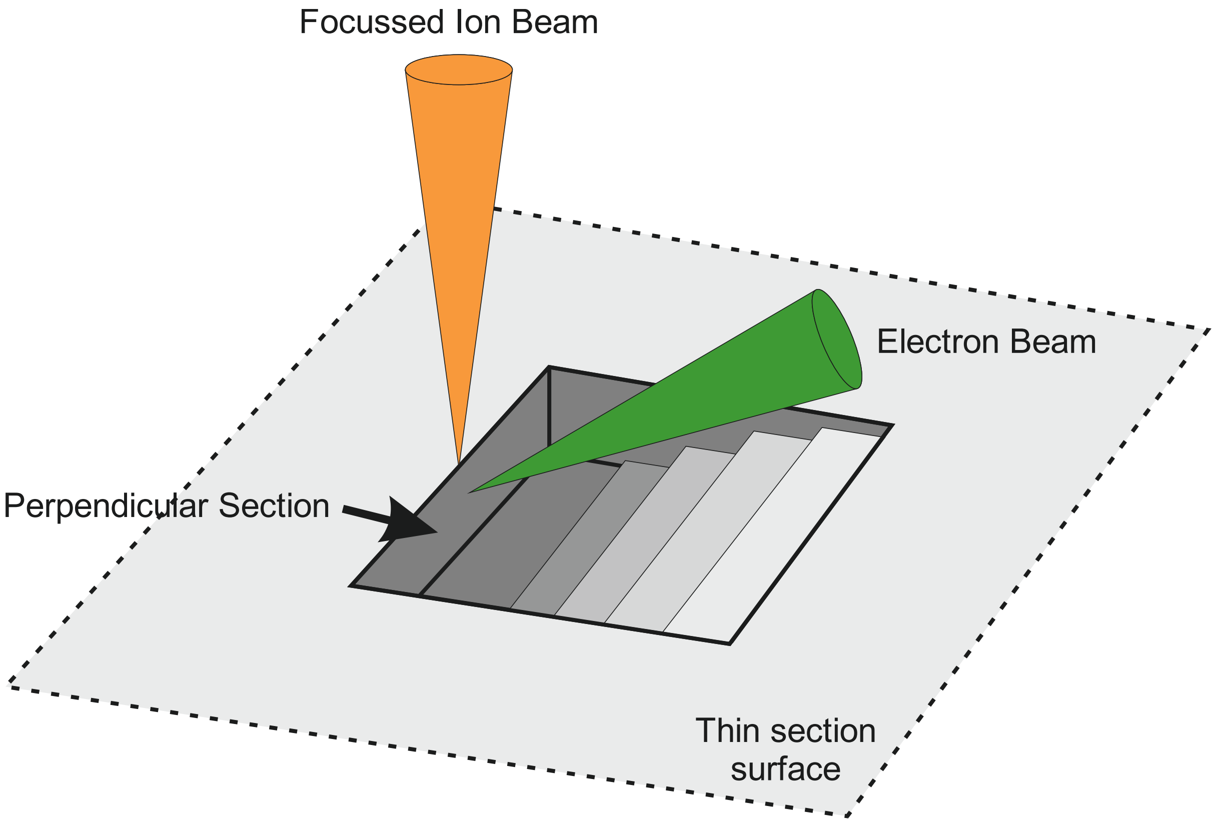


Figure G.3.2. Schematic drawing illustrating the principle of making a perpendicular section on a thin section using a FIB-SEM system. The hole in the thin section surface is generated in steps using the ion beam. The milling, and especially the late-stage fine milling, is done approximately tangential to the ion beam, by tilting the sample to a suitable position. The electron beam, which stands vertical in the Quanta 3D device, is then used to image the perpendicular surface generated with the FIB. Of course also for this step the sample needs to be tilted to a suitable position.

For the creation of a perpendicular section on a thin section, one takes several steps in the FIB-SEM (also see Figure G.3.3). Once a region of interest has been found, several alignment steps are done for the sample and the beams. Then, a thin bar or layer of a conductive material is deposited on the thin section sample surface. This is done to keep control over the position of the region of interest, to improve the consistency of the milling speed and depth over the entire section, and to increase the conduction for using the SEM in a later stage. In the used Quanta 3D, one has the choice between depositing a conductive layer of Platinum or Carbon. This is done by injecting a gas of one of these materials, which is then deposited using the ion beam. After deposition, one can start the milling. First a rough cut is made, followed by increasingly finer milling steps. This requires a lot of quality control and manual settings and adjustments, therefore taking quite some time. For example, one problem that needs to be assessed is that milled material is again deposited in the vicinity of the section. This cannot be prevented, but the redeposited material should not block the view or attach on the freshly created surface of interest. One also needs to make sure material is milled away as equally as possible to end up with a good, flat surface. This is difficult due to the effect of preferential milling, as some materials are milled away easier than others. When eventually a proper perpendicular section is created, the sample is tilted to a position suitable for SEM imaging.

In addition to creating only a single perpendicular surface, one can also create more consecutive perpendicular sections. This is done by imaging the first perpendicular section, followed by milling off a tiny layer, imaging this new surface, and repeating these steps. This process is termed *serial sectioning* or FIB-SEM tomography. After the initial perpendicular section is generated by the manual approach explained earlier, serial sectioning can be automated in the Quanta 3D.

The advantages of using FIB-SEM over using SEM on the surface only is clear: one gets a view in the third dimension, so not limited to the 2D flat surface anymore. This can be used to investigate for example the consistency or extent of features seen at the surface in the 3rd dimension as well. Furthermore, when using serial sectioning properly, this information can be combined into a complete sampled volume of data. This can then for example be processed in a similar manner as the 3D datasets from µCT.

A clear disadvantage of using FIB-SEM is the size that can be sampled (typical perpendicular sections are 20 µm wide). This in some cases puts question marks on whether such a section is representative for the sample on a larger scale. This, combined with the process being quite time-consuming, results in that FIB-SEM analysis has only been applied on a single sample in this study (STRT4-1-3320,1). Also note that an additional drawback of FIB-SEM (tomography), is that the milling is destructive. One cannot repeat the procedure or reassess how the surface view looks. It is therefore very important that high quality data on the surface that will be milled is generated in advance, and that areas of investigation are chosen well.

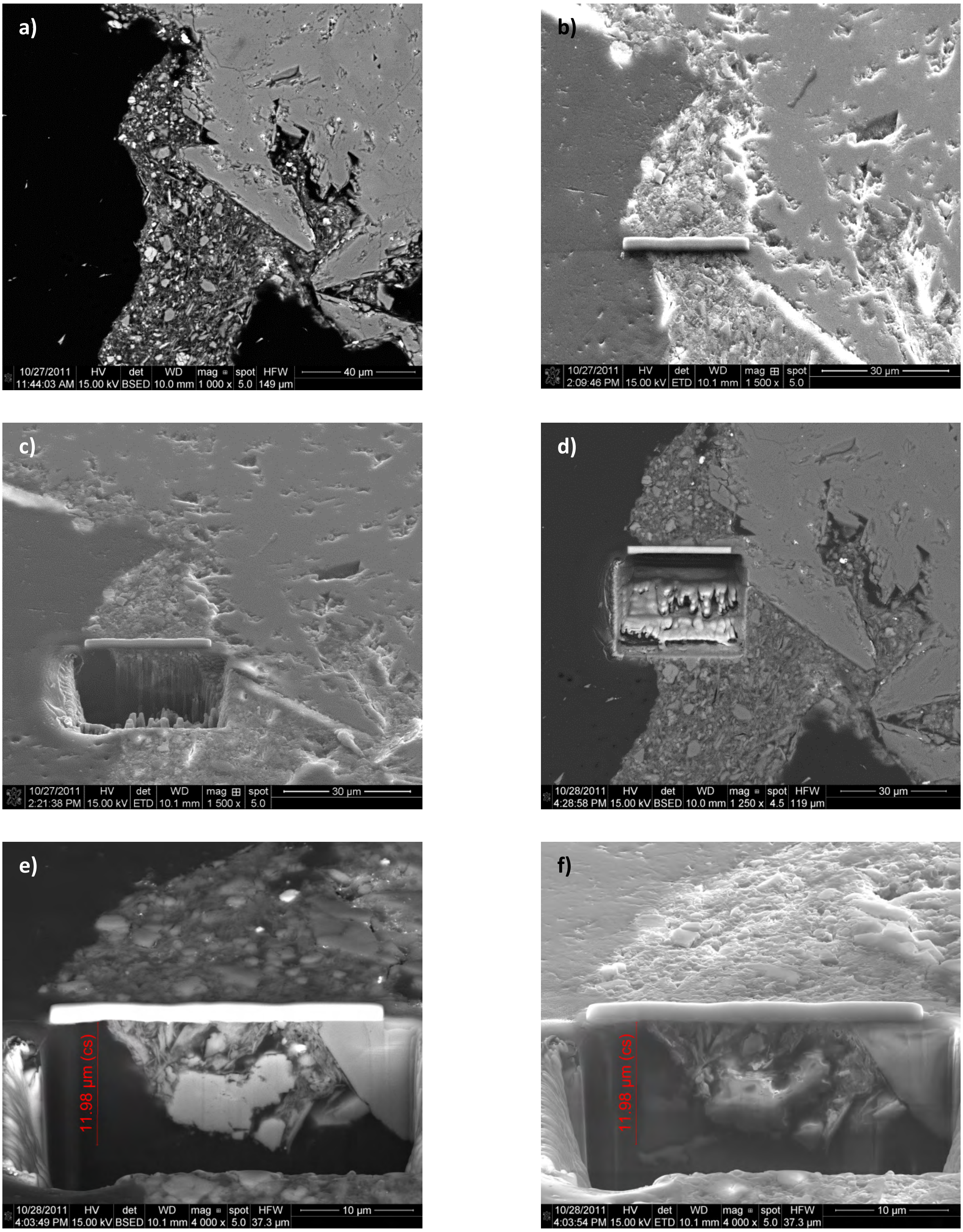

a)
10/27/2011 11:44:03 AM HV 15.00 kV det BSED WD 10.0 mm mag 1 000 x spot 5.0 HFW 149 µm
40 µm
b)
10/27/2011 2:09:46 PM HV 15.00 kV det ETD WD 10.1 mm mag 1 500 x spot 5.0
30 µm
c)
10/27/2011 2:21:38 PM HV 15.00 kV det ETD WD 10.1 mm mag 1 500 x spot 5.0
30 µm
d)
10/28/2011 4:28:58 PM HV 15.00 kV det BSED WD 10.0 mm mag 1 250 x spot 4.5 HFW 119 µm
30 µm
e)
11.98 µm (cs)
10/28/2011 4:03:49 PM HV 15.00 kV det BSED WD 10.1 mm mag 4 000 x spot 5.0 HFW 37.3 µm
10 µm
f)
11.98 µm (cs)
10/28/2011 4:03:54 PM HV 15.00 kV det ETD WD 10.1 mm mag 4 000 x spot 5.0 HFW 37.3 µm
10 µm

Figure G.3.3. Several steps in creating a single perpendicular section in sample STRT4-1-3320,1 using FIB-SEM. **a)** Backscattered Electron (BSE) image of a region of interest on the sample surface. The region shows a material aggregate in resin (black on the image), bordering dolomite host rock. Scale bar 40 µm. **b)** Secondary Electron (SE) image of the region of interest with a ca. 20 µm wide deposited bar of carbon. Scale bar 30 µm. **c)** SE image during the milling process. The surface is not yet clear, and distinct columns of material are visible. Scale bar 30 µm. **d)** BSE top view of the final result. Some columns and redeposited material are visible, but these are not blocking the final created surface. Scale bar 30 µm. **e)** BSE view of the final perpendicular surface. Horizontal scale bar 10 µm. **f)** SE view of the final perpendicular surface. Horizontal scale bar 10 µm.

G.3.2. FIB-SEM Tomography image processing

Some important processing steps are required for the data produced in the FIB-SEM technique, especially for sets of images from FIB-SEM tomography:

- First of all, the data that comes out of the serial slicing process is not yet fully aligned. A (not always trivial) step of image alignment, termed "registration", is thus needed to be able to work with the data. We apply a SIFT-based linear stack alignment approach in FIJI, and compensate for introduced (constant) vertical shifts manually.
- Second, due to the inclination of the electron beam to the scanned surface, and the slicing step size that needs to be set, the voxel dimensions of the produced 3D dataset differ in all 3 directions. The x and y voxel dimensions can be calculated using the original metadata of the produced images. For the z-direction, the voxel dimensions can be estimated when control markers were placed during data acquisition.
- As a third comment, the inclination also often generates a gradient in greyscales throughout the data. Also in the slicing direction, variations in greyscale can be introduced. Extra image processing steps are thus required to remove these gradients. We apply slice-based median contrast and brightness correction to obtain the best possible results.
- Fourth, charging introduces additional light and dark bands in the data, which can be removed for improving later processing steps.
- Fifth and finally, it is important to realise that every imaged "2D" slice also contains some 3D information, due to the interaction volume of the electron beam for BSE imaging. This can result in artefacts in the final data volume.

From the above, it should be clear the processing of FIB-SEM tomography datasets is not a trivial task. Even if a dataset can be produced the processing and analysis steps may be limited because of the produced data. This should therefore be kept in mind when using FIB-SEM tomography.

G.3.3. FIB-SEM Tomography image analysis

First of all, after all possible pre-processing steps, the data need to be segmented. Because of the image quality, several steps were taken here, including a lot of manual processing. Software packages applied in this process are Blob3D® (Ketcham, 2005), Amira®, and FIJI. Island removal is performed using MATLAB®, in the same way as discussed in Section E.1.2. Also, parallel to the approach in Section D.2.4.2, the connectivity of the segmented porosity throughout the sample can be investigated with MATLAB®. The size of the pores is analysed using the Local Thickness approach, as discussed for aperture determination in Chapter F. Finally, the orientation and size of the pores can be investigated by using for example the particle analysis plugin for FIJI. Here, best fitting ellipsoids (so 3D) are created around pores. Using the defined axes of the pores, it is possible to plot the pore orientation into a stereographic projection, in order to determine whether a preferred orientation is present. It should however be noted, that especially for the orientation analysis, the data acquisition technique might have a large influence.

G.3.4. FIB-SEM Tomography results

FIB-SEM was carried out on 1 thin section sample in this study – STRT4-1-3320.1 – which is a cataclasite that stems from a piece of core several centimetres adjacent to the two Strasshof T4 samples scanned by µCT (see Chapter F). Two FIB-SEM serial sections have been made on the sample: in an area within the cataclastic matrix, and an area with in larger fragment.

Figure G.3.4 shows the top view of the area where the serial section in a cataclastic matrix has been made. A porous fragment/aggregate was chosen here, since the actual matrix has too large pore sizes relative to the size of the volume that can be analysed. Figure G.3.5 shows the results of the FIB-SEM tomography. In this area, the matrix likely contributes to the effective porosity and permeability. The image quality however hampers a good segmentation of the fractures and other pores from this sample. Therefore, no further analyses have been performed on this dataset.

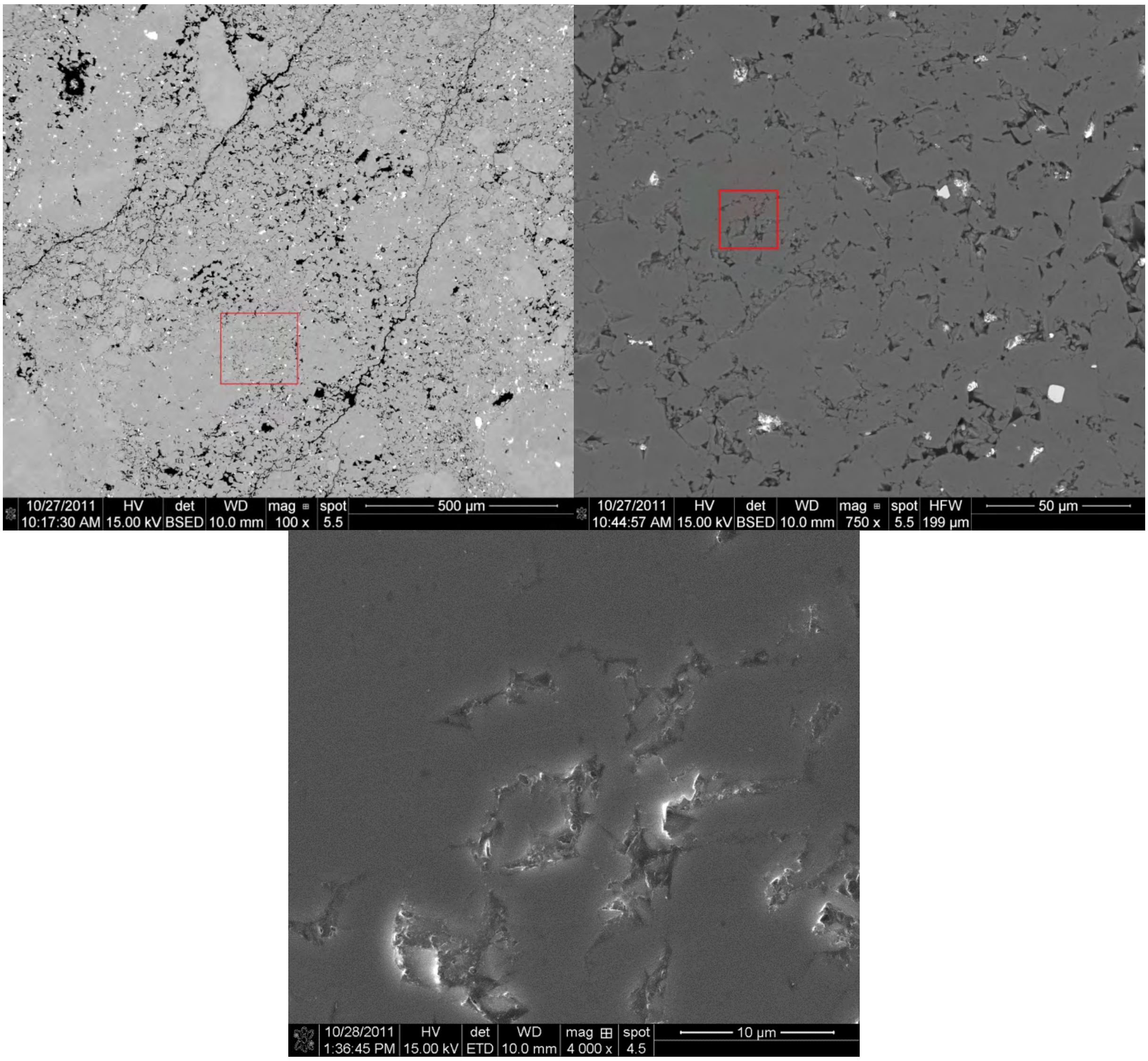


Figure G.3.4. Surface view (top two images: SEM-BSE contrast, bottom image: SEM-SE contrast) of the area used for FIB-SEM tomography in a fragment in the porous cataclastic matrix. The red square in the left top image shows the location of the right top image. The red square in the right top image shows the location of FIB-SEM milling, shown enlarged in SEM-SE contrast in the bottom image.

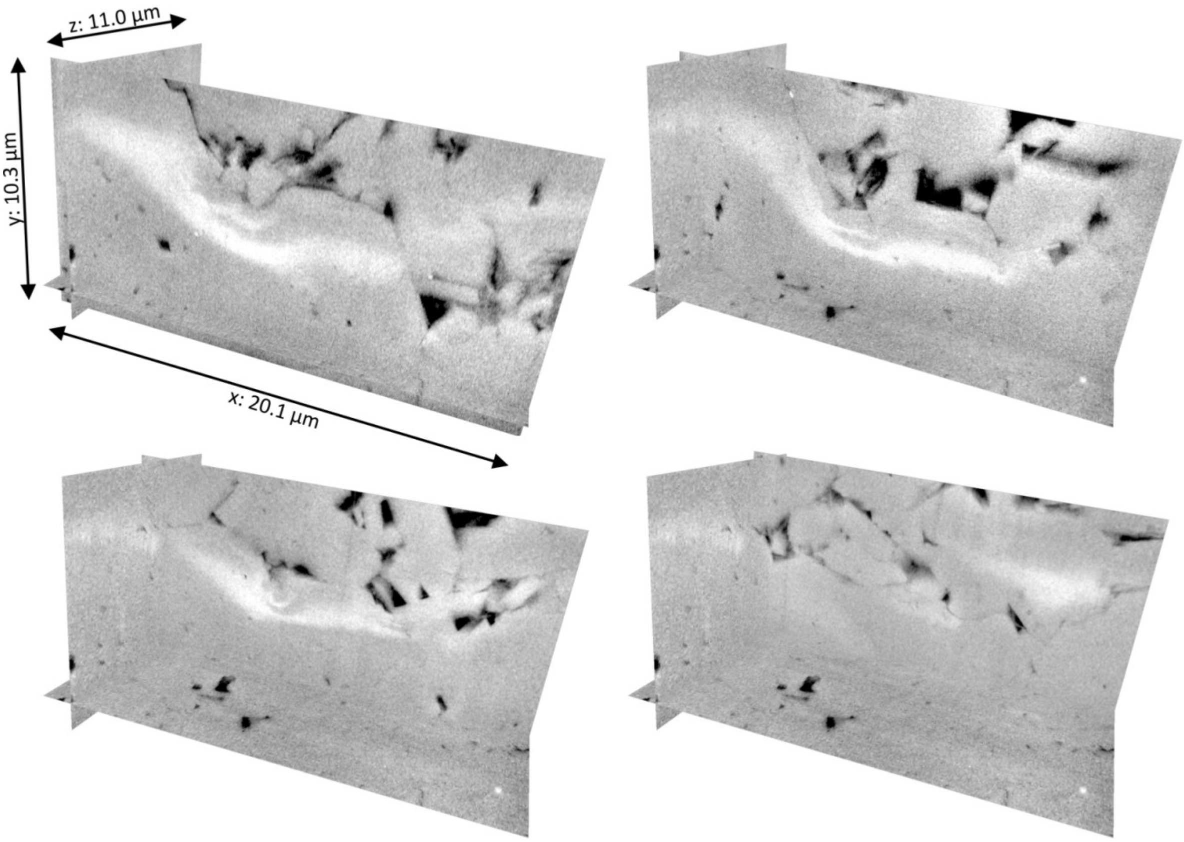


Figure G.3.5. Orthoslice based 3D views created in the Amira® visualisation software for the registered, intensity corrected and correctly scaled FIB-SEM tomography dataset. The milling was performed at the location indicated in Figure G.3.4. Charging areas by the electron beam (bright) have not been removed. The dataset crosscuts a larger fragment at the bottom, but shows a porous and connected structure in the top areas.

A second FIB-SEM section was made in a different area of the thin section STRT4-1-3320,1, inside a larger dolomite grain. Figure G.3.6 shows the area of milling within the chosen fragment in the sample. The resulting FIB-SEM tomography in the area is shown in Figure G.3.7, in which small pores are clearly visible. A manual segmentation attempt on the data is shown in Figure G.3.8.

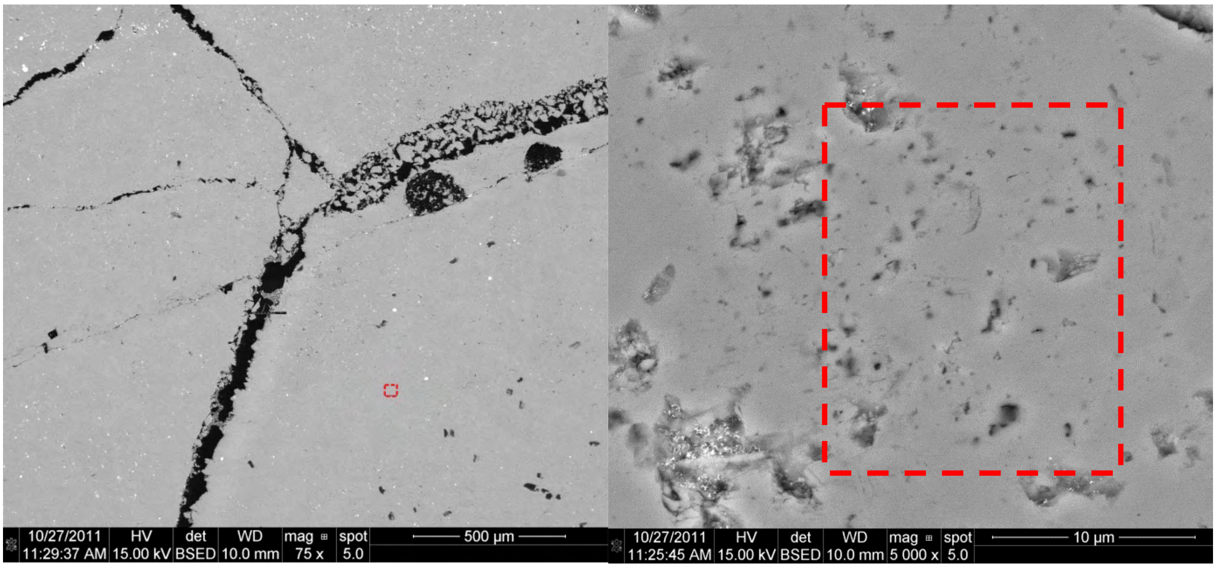


Figure G.3.6. Surface views (SEM-BSE contrast) of the area used for FIB-SEM tomography inside a more solid dolomite fragment in sample STRT4-4-1-3320.1. The red square in the left image shows the location of the right top image. The red square in the right image shows the area of milling.

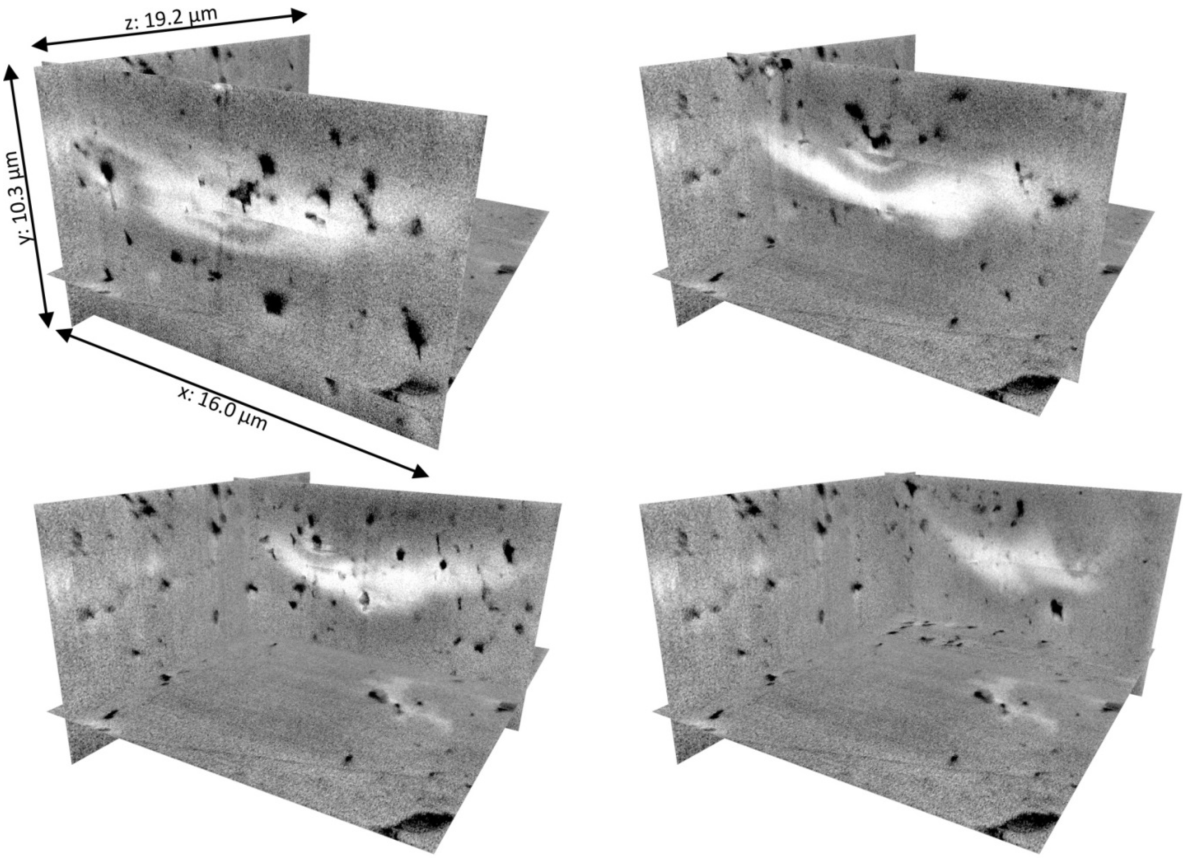


Figure G.3.7. Orthoslice based 3D views created in the Amira® visualisation software for the registered, intensity corrected and correctly scaled FIB-SEM tomography dataset. The milling was performed at the location indicated in Figure G.3.6. Charging areas by the electron beam (bright) have not been removed. In the data, small “intragrain” pores are clearly visible as black spots.

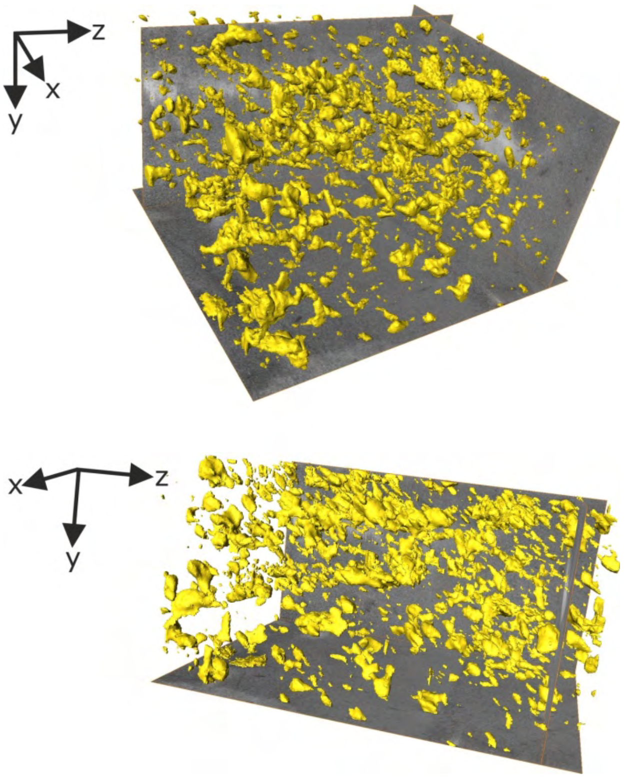


Figure G.3.8. Manual segmentation attempt on the dataset shown in Figure G.3.7, using a combination of Blob3D®, Amira®, and FIJI, and displayed in the Amira® visualisation software. The porosity is shown using surface visualisation in yellow. These pores constitute a porosity of 2.6%, but a connectivity analysis shows the displayed pores are not connected (at the resolution shown here).

The porosity (2.6%) displayed in Figure G.3.8 is not connected in any direction according to a connectivity analysis in MATLAB®. We can therefore argue the tiny pores within fragments most likely do not contribute to the effective porosity and permeability of the samples. Something else that can be observed in Figure G.3.8 is a small preferred orientation of the pores (dropping in the YZ direction). Possibly, this is an effect of the milling and aligning steps taken, but this cannot be confirmed on this dataset.

Besides visualisation and porosity calculation, some extra parameters can be extracted. First of all, pore size analysis can be performed using a similar 3D “Local Thickness” analysis as discussed in Chapter F. These results are shown in Figure G.3.9. Additionally, the pores can be analysed using a particle analysis, where ellipsoids are fitted inside the pore space. Figure G.3.10 shows a Flinn diagram of the ellipsoids fitted on the dataset shown in Figure G.3.7. Although most data plots close to the origin, there is a slight preference for elongated pores. This corresponds to the observations in Figure G.3.7. Similarly, the orientation of the long axes of the pores displays a preferred orientation, as shown in Figure G.3.11.

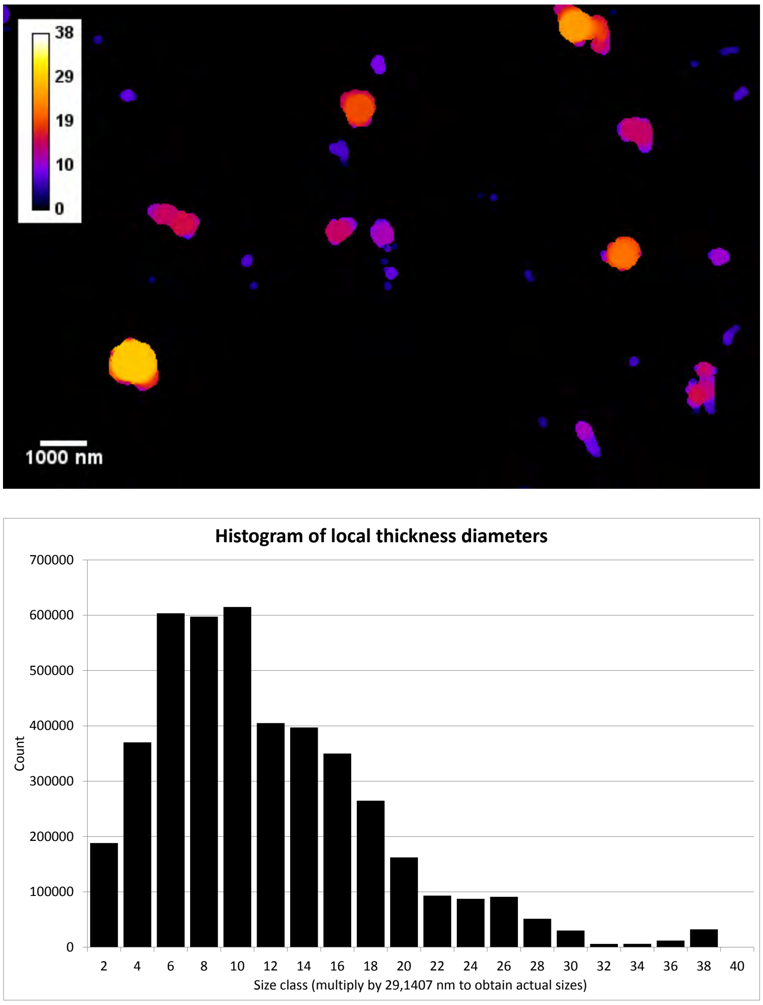


Figure G.3.9. 3D local thickness approach on segmented FIB-SEM tomography data. The top image shows the output of the local thickness technique on the image. The lower image shows a histogram of the recorded diameters. Since no corrections have been carried out, there is a bias to smaller pore sizes. However, there still seems to be a true peak from pore sizes of ca. size classes 6-12 voxels, corresponding to a true diameter of ca. 175-350 nm (i.e. pore volumes for this spherical representation of circa $2{,}8 \cdot 10^6 – 2{,}2 \cdot 10^7$ nm^3, or $0{,}0028 – 0{,}022$ μm^3).

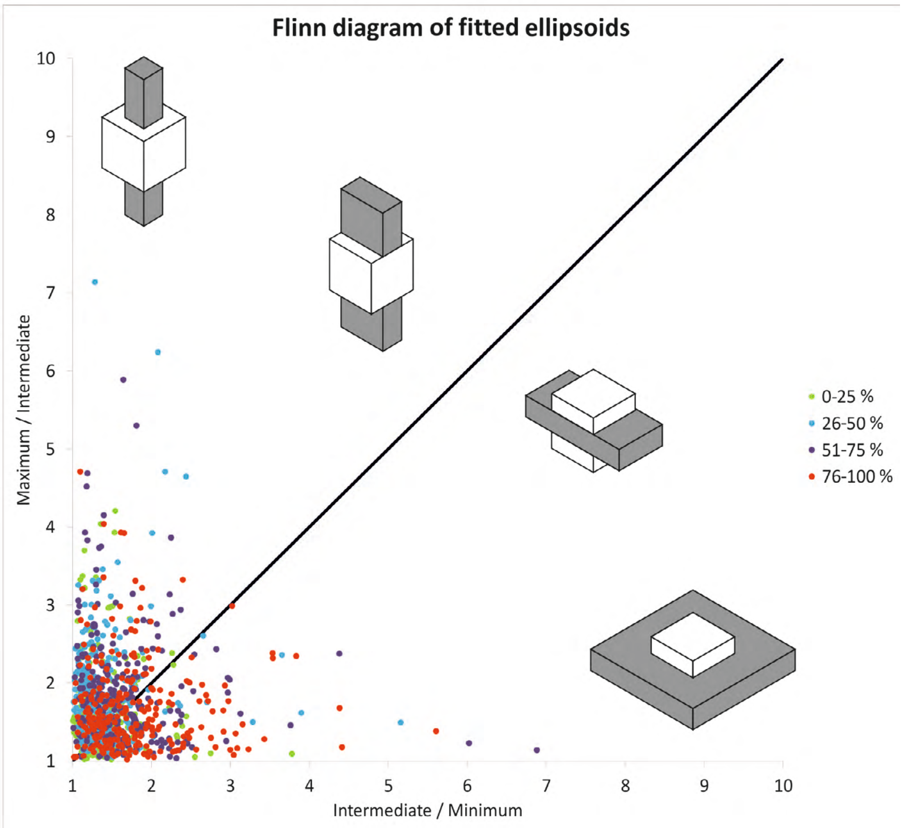


Figure G.3.10. Flinn diagram of ellipsoids fitted to the segmented pores displayed in Figure G.3.7. Although most data plots close to the origin, there seems to be a slight preference for elongated pores. The 4 classes shown represent size classes (e.g. 0-25% are the 25% smallest pores, et cetera).

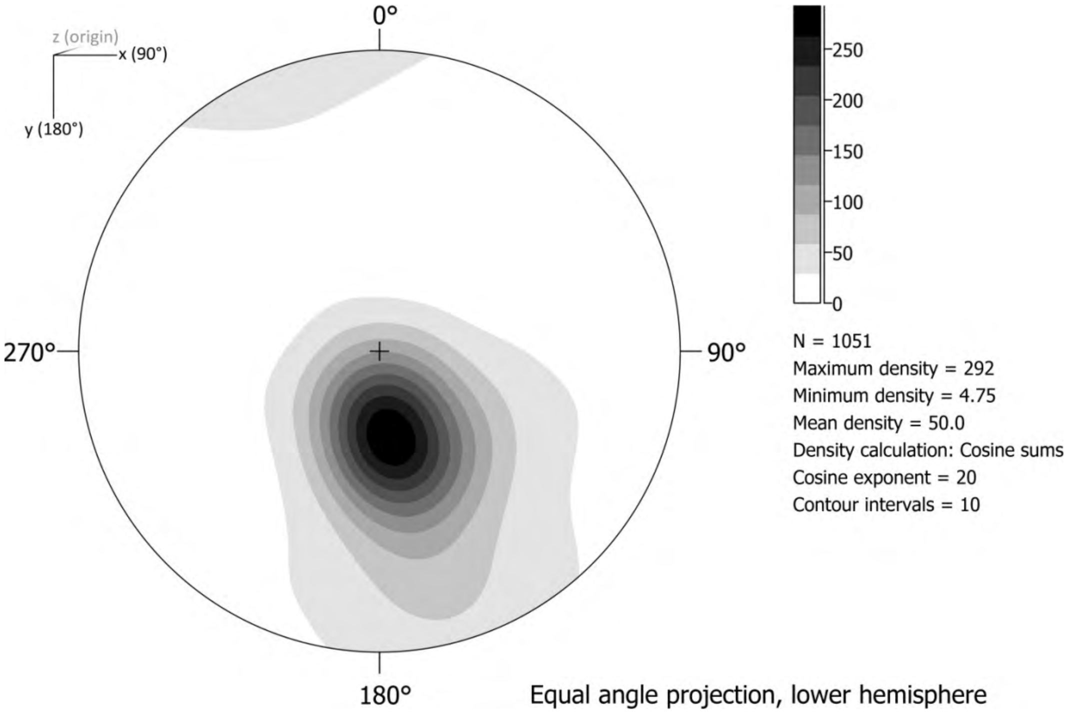


Figure G.3.11. Equal area, lower hemisphere projection of the orientation of the long axes of the pores displayed in Figure G.3.7, after particle analysis. There appears to be a clear preferred orientation, which is in line with the observations in Figure G.3.7. Stereographic projection created using Stereo32 (Röller & Trepmann, 2008).

G.3.5. FIB-SEM conclusions

In the previous sections, some examples of FIB-SEM tomography on the dolomite samples of this study are shown. FIB-SEM tomography can clearly be a very powerful technique if high-resolution 3D data is required. With successful segmentation, porosity, connectivity, size and orientation analysis and possibly permeability modelling are feasible. However, data acquisition and processing are very resource-consuming, and hence should only be used if having such high-resolution 3D information is a clear advantage. For the fractured dolomite samples shown in this study this is not the case, especially when maintaining a research focus on hydrocarbon applications: even though the porosity and connectivity at the FIB-SEM level might be of importance, larger fractures and pores are a priority. Furthermore, even with available FIB-SEM tomography data, the link to the available µCT data is very difficult, as a single FIB-SEM tomography dataset is approximately the same size as a single voxel (!) in the µCT datasets shown in previous chapters. Linking the FIB-SEM tomography data to this different scale is therefore – without any intermediate 3D datasets – a near-impossible task.

H. Permeability

In Chapter F, the main results for permeability determination by flow experiments in the lab have been shown. Such experiments, especially when carried out under different confining pressures, appear to be the most suitable for determining the flow and behaviour at depth for the samples shown in this study. However, with segmented µCT data and other parameters available, also permeability modelling on the µCT data – i.e. on the large fracture network and large pores – would be of interest. Within the scope of this thesis no full permeability modelling has been carried out, but some attempts have still been undertaken. Of these approaches, the *Lattice Boltzmann Method* (LBM) is the most promising.

In conventional Computational Fluid Dynamics (CFD) the Navier-Stokes equations are generally solved directly, by solving equations of conservation of mass, energy et cetera. In order to do this, one discretises the equations, by simplifying the geometry of a dataset by e.g. finite difference or finite element methods. This also reduces the computational power required. In contrast, with the Lattice Boltzmann Method (LBM) one does not alter the original geometry of a dataset, but discretises the initial kinetics of the problem. This is done by modelling the distribution and collisions of artificial particles flowing through the dataset. This modelling involves solving the discrete Boltzmann equation, after which the Navier-Stokes equations can be solved (e.g. Sukop and Thorne 2006). A clear advantage of the LBM is that it is very suitable for complex boundaries, as it can use the geometry from segmentation on µCT data directly, without requiring additional processing or large assumptions about the pore geometry (e.g. Sahimi, 2011). In addition, the concept is relatively easy to understand. A disadvantage is that applying the LBM requires quite some computational power (compared to CFD). However, due to increased available computer power over time, the LBM is gaining popularity.

In this study, an attempt has been made to see if the LBM is applicable on an available Hauptdolomit dataset. To this end, the open source software Palabos (www.palabos.org) has been used, which implements a large range of LBM applications, combined with a Palabos-code written by De Gruyter et al. (2010) – originally designed for fluid flow through volcanic gas pumices. This model uses a so-called D3Q19 lattice, which defines 3D connectivity by an 19-connected neighbourhood), but this can be altered.

A small (300x300x300 voxels) subvolume has been taken from segmented data from the Hauptdolomit µCT scan of Prottes TS1. Multiplied with the resolution this means a ca. 5.6 x 5.6 x 5.6 mm subvolume. An area with a high fracture density has been selected to test the applicability of the method, as shown in Figure H.1. When compared to different permeability modelling approaches on the same dataset, the permeability (ca. 10 D) lies within the same order of magnitude. Because this is a subset of a full µCT scan, no direct comparison to laboratory data can be made, but this value is of course very high compared to full plug permeabilities. These high values can however be expected for small well-connected small subsets, and possibly some overestimation of the aperture also plays a role. The particular modelling run shown here takes about 30.5 hours on the PC described in Chapter D, using a single core. (Much) shorter times can be expected on successful parallel implementing of such code and/or on more powerful computers.

With these computational times, no LBM could be applied on full samples without heavy downsampling. An example on sample Schönkirchen T91-5p with 4x downsampling in every direction is shown in Figure H.2. Although in principle the method still works, many details in the sample have been lost by the downsampling, leading to many potential flow paths being interrupted and other artefacts. Since sample Schönkirchen T91-5p has the largest apertures of all µCT scanned samples in this study, this effect will be even worse for other samples. Applying the LBM on downsampled data is thus not a desirable approach here.

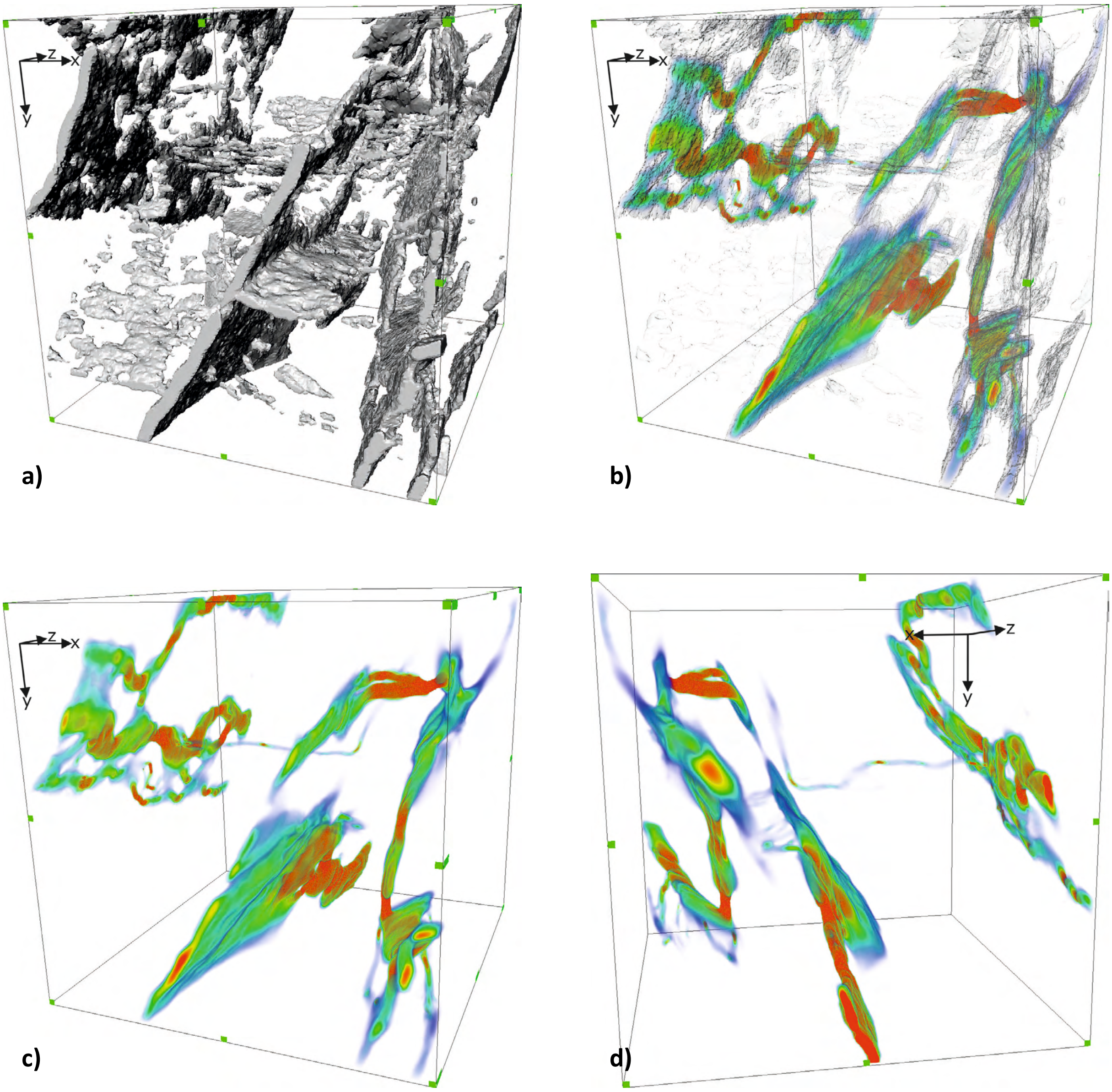


Figure H.1 Visualisation of LBM modelled flow through a 300x300x300 voxels (=ca. 5.6 x 5.6 x 5.6 mm) segmented subvolume of sample Prottes TS1, using the Amira® visualisation software. Flow is in the z-direction only, which is here horizontally (i.e. approximately along the main fractures). **a)** Surface rendering of the segmented fractures. Note also non-connected porosity is included in this model. **b)** Volume rendering of fluid flow patterns through the fractures in the z-direction. High flow speeds are in red, lower flow speeds (i.e. in the broader areas) in green and blue. The surface rendering of the fractures is shown transparently for comparison. **c)** Same as b, but flow pattern only. **d)** Identical flow pattern as c, but visualised from a different angle.

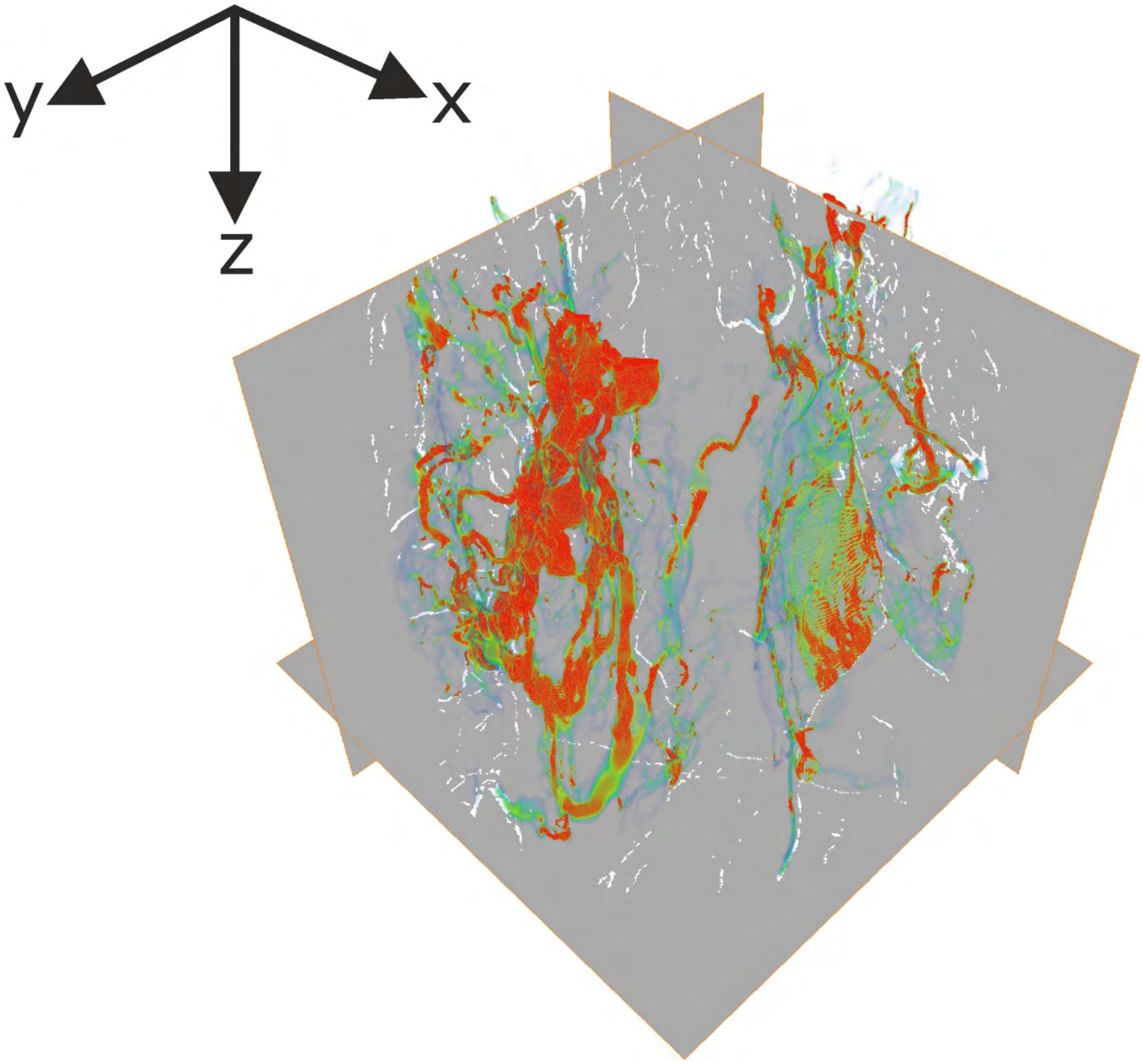


Figure H.2 Visualisation of LBM modelled flow through a 4x downsampled (xyz: 256 x 256 x 366 voxels) segmented dataset of 3 cm diameter plug sample Schönkirchen T91-5p, using the Amira® visualisation software. Flow is in the z-direction only, (i.e. approximately along the main fractures). The orthoslices show the segmented data in the background, and the volume rendering shows the flow speeds in colour, with red being high and blue being low. Part of the fracture network is visualised in the flow paths. However, due to the high downsampling, not all connections present in the original data are available here. Additionally, the apertures get very narrow (up to 1 or 2 voxels) in the downsampled data, leading to artefacts such as the stairstepping visible in the right of image (alternating red and green lines).

I. Summary and conclusions

The central theme of this thesis has been a survey of suitable (new) techniques and combinations of techniques for the analysis of fractured samples for the petroleum industry, with a particular focus on drill core samples from hydrocarbon reservoirs. In this research, several samples from fractured dolomite (“Hauptdolomit”) reservoir drill cores originating from the basement of the Vienna Basin (Austria) were used, but many of the concepts discussed here can potentially be used on other rock types. For the sample rocks, a (crude) classification of fault zone location is useful at the beginning of the analysis routine, and came here from fieldwork insights on outcropping rocks that serve as a reservoir analogue.

After initial laboratory porosity and permeability determination, µCT scanning was carried out on the samples. With current technical limits, suitable results on narrowly fractured samples could only be obtained on 2 and 3 cm diameter plugs. To ensure good image quality with a reduction in noise, scan averaging was carried out (i.e. scanning the sample multiple times in the same orientation, and using all this data as a single radiogram). After obtaining suitable 3D µCT reconstructions, processing of the data was necessary to be able to extract petrophysical characteristics in a later stage. In this processing step, it was realised the most common existing methods for segmentation for geological samples could not be applied successfully, because of various reasons (ranging from simply not working to being very promising but computationally too demanding). An alternative method based on concepts from image processing for medical imaging was developed and termed multiscale Hessian fracture filtering. Using this methodology, thin fractures in the sample data could be extracted successfully without high computational demands (i.e. working on full, non-downsampled µCT datasets on a powerful desktop computer, but compensating by disk space and calculation time) and with a minimum of human bias.

After segmentation, the µCT data was analysed in order to extract parameters that can be useful in hydrocarbon research. Porosity, aperture, fracture density and fracture orientation could be successfully determined in this way, and show clear differences between samples from different localities in a fault zone. A (semi-) automated extraction of such parameters can hence be useful in analysis routines in larger studies, to automatically classify the investigated rock samples. The complex fracture networks of the examples studied in this research unfortunately prevented the separation of sets of fractures in 3D datasets, making the analysis of several other interesting parameters (e.g. fracture spacing and fracture length) not feasible.

Besides µCT, thin sections were used for further investigation of the samples at a better resolution. Thin sections were used as a source of constituent and microstructural analysis, and also show clear differences between rocks at various locations in a fault zone. Complete thin section overviews by SEM-BSE were the most useful in the comparison to µCT, as well as in providing an additional value for porosity. Cathodoluminescence (CL) could only be applied on some thin sections, but is the most suitable technique for unravelling different fault generations and vein and cement growth. However, the 2D information remains a limiting factor. To obtain better resolution 3D information, 3D Focussed Ion Beam – Scanning Electron Microscopy (FIB-SEM) tomography was investigated on thin sections. Although FIB-SEM tomography could be successfully applied, the technique is very resource-consuming and covers such a small area (e.g. a single or a few voxels of the µCT data in this study) that links to different scales are difficult to make. Therefore, FIB-SEM tomography is only useful if a very clear goal at a very specific sample location is defined.

Next to the “static” parameters discussed above, permeability is one of the most important parameters for the hydrocarbon industry. Within the scope of this research, complete permeability modelling on 3D datasets has not been performed, although some tests models applying the Lattice Boltzmann Method (LBM) show several promising results. The main information on permeability however stems from laboratory flow experiments on selected samples. Flow tests using water or gas as a fluid with the sample being under varying confining pressure provide the most interesting results. The decrease of permeability with increasing confining pressure gives a better link to the actual conditions inside a reservoir, and may also be continued to for example aperture and porosity reduction at depth. For the rocks from different locations in the fault zone, different degrees of permeability reduction could be observed, which is again useful in characterising the rocks. This kind of information is valuable in trying to upscale information to a larger scale (e.g. reservoir scale), but it should be noted that this is difficult for the complex small fracture networks in plug-sized samples (i.e. these plugs are not a representative elementary volume).

In conclusion, this study shows that µCT can be a suitable tool in researching narrowly fractured rock samples for hydrocarbon research. Although µCT can certainly not serve as a replacement for more common techniques applied, it can serve as an interesting addition, also in linking different types of investigation. Especially the 3D detailed information obtained by µCT is of value in that perspective. The processing and analysis of such data comes with inherent difficulties, but after developing the concepts and routines shown in this thesis these can easily be applied in larger and (semi-) automated workflows. Additionally, the extractable parameters can serve as important natural input data to for example fracture and flow models of narrowly fractured rocks.

References

Aguilera, R., 1995. Naturally Fractured Reservoirs (2nd ed.). PennWell Books, Tulsa.

Antonellini, M., Mollema, P.N., 2000. A natural analog for a fractured and faulted reservoir in dolomite: Triassic Sella group, Northern Italy. AAPG Bulletin 84(3), 314-344.

Arzmüller, G., Buchta, S., Ralbovský, E., Wessely, G., 2006. The Vienna Basin. In: Golonka, J., Picha, F.J. (eds), The Carpatians and their foreland: Geology and hydrocarbon resources. AAPG Memoir 84, pp. 191-204.

Barnhoorn, A., Cox, S.F., Robinson, D.J., Senden, T., 2010. Stress- and fluid-driven failure during fracture array growth: implications for coupled deformation and fluid flow in the crust. Geology 38(9), 779-782. doi: 10.1130/G31010.1.

Bauer, H., 2010. Deformationsprozesse und hydrogeologische Eigenschaften von Störungszonen in Karbonatgesteinen. M.Sc Dissertation, Earth Sciences, University of Vienna, Austria, 102pp.

Baveye, P.C. et al., 2010. Observer-dependent variability of the thresholding step in the quantitative analysis of soil images and X-ray microtomography data. Geoderma, 157(1-2): 51-63.

Bertels, S.P., DiCarlo, D.A., Blunt, M.J., 2001. Measurement of aperture distribution, capillary pressure, relative permeability, and in situ saturation in a rock fracture using computed tomography scanning. Water Resources Research 37(3), 649-662.

Billi, A., 2010. Microtectonics of low-P low-T carbonate fault rocks. Journal of Structural Geology 32, 1392-1402.

Billi, A., Salvini, F., Storti, F., 2003. The damage zone-fault core transition in carbonate rocks: implications for fault growth, structure and permeability. Journal of Structural Geology 25, 1779-1794.

Blendinger, W., 1997. Dolomitization of the Dolomites (Triassic, Northern Italy): Pilot study. Neues Jahrbuch für Geologie und Paläontologie-Abhandlungen 204(1), 83-110.

Bossie-Codreanu, D., Wolf, K.-H. A A., Ephraim, R., 2004. A new characterization method for coal bed methane. Geologica Belgica 7(3-4), 137-145.

Choquette, P.W., Pray, L.C., 1970. Geologic nomenclature and classification of porosity in sedimentary carbonates. AAPG Bulletin 54(2), pp. 207-244.

Christe, P.G., 2009. Geological Characterization of Cataclastic Rock Samples Using Medical X-ray Computerized Tomography: Towards a Better Geotechnical Description. Ph.D. Dissertation, Faculté Environnement Naturel, Architectural et Construit. École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, 338pp.

Cnudde, V., Masschaele, B., Dierick, M., Vlassenbroeck, J., Hoorebeke, L., Jacobs, P., 2006. Recent progress in X-ray CT as a geosciences tool. Applied Geochemistry 21(5), 826-832. doi:10.1016/j.apgeochem.2006.02.010.

Degruyter, W., Burgisser, A., Bachmann, O. and Malaspinas, O., 2010. Synchrotron X-ray microtomography and lattice Boltzmann simulations of gas flow through volcanic pumices. Geosphere, 6(5): 470-481

Descoteaux, M., Audette, M., Chinzei, K., Siddiqi, K., 2005. Bone enhancement filtering: application to sinus bone segmentation and simulation of pituitary surgery. In: Duncan, J.S., Gerig, G. (Eds.), Proceedings Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2005, Palm Springs, California, USA. Lecture Notes in Computer Science 3749, pp. 9-16.

Dougherty, R.P., Kunzelmann, K.-H., 2007. Computing Local Thickness of 3D Structures with ImageJ. Microscopy & Microanalysis Meeting, August 2007, Ft. Lauderdale, Florida. Conference Presentation. http://www.optinav.com/LocalThicknessEd.pdf [Accessed 22 May 2014]

Ehrig, K., Goebbels, J., Meinel, D., Paetsch, O., Prohaska, S., Zobel, V., 2011. Comparison of Crack Detection Methods for Analyzing Damage Processes in Concrete with Computed Tomography. In: Proceedings International Symposium on Digital Industrial Radiology and Computed Tomography (DIR) 2011, Berlin, Germany, 8pp.

Elliot, T.R., Heck, R.J., 2007. A comparison of 2D vs. 3D thresholding of X-ray CT imagery. Canadian Journal of Soil Science 87(4), 405-412. doi: 10.4141/CJSS06017.

Ellis, B.R., 2012. Geologic carbon sequestration in deep saline aquifers: brine acidification and geochemical alterations of reactive leakage pathways. Ph.D. Dissertation, Faculty of Princeton University, Department of Civil and Environmental Engineering. Princeton, 118pp.

Engler, T.W., 2010. Fluid flow in porous media. Petroleum Engineering 524 Lecture Notes, New Mexico Tech. http://infohost.nmt.edu/~petro/faculty/Engler524/ [Accessed 22 May 2014]

Faulkner, D.R., Rutter, E.H., 2000. Comparisons of water and argon permeability in natural clay-bearing fault gouge under high pressure at 20°C. Journal of Geophysical Research 105(B7), 16415-16426.

Flügel, E., 2004. Microfacies of Carbonate Rocks: Analysis, Interpretation and Application. Springer, Berlin, Heidelberg, New York.

Forchheimer, P., 1901. Wasserbewegung durch Boden. Zeitschrift des Vereines deutscher Ingenieure 45, 1782-1788.

Fornaro, J., Székely, G., Harders, M., 2010. Semi-automatic Segmentation of Fractured Pelvic Bones for Surgical Planning. In: Bello, F., Cotin, S. (Eds.), Proceedings 5th International Symposium on Biomedical Simulation (ISBMS) 2010, Phoenix, Arizona, USA. Lecture Notes in Computer Science 5958, pp. 82-89.

Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A., 1998. Multiscale Vessel Enhancement Filtering. In: Wells, W.M., Colchester, A., Delp, S.L. (Eds.), Proceedings Medical Image Computing and Computer-Assisted Intervention (MICCAI) 1998, Cambridge, Massachusetts, USA. Lecture Notes in Computer Science 1496, pp. 130-137.

Fruth, I., Scherreiks, R., 1984. Hauptdolomit - Sedimentary and Paleogeographic Models (Norian, Northern Calcareous Alps). Geologische Rundschau 73, 305-319.

Fusi, N., Martinez-Martinez, J., 2013. Mercury porosimetry as a tool for improving quality ofmicro-CT images in low porosity carbonate rocks. Engineering Geology 166, 272-282. doi:10.1016/j.enggeo.2013.10.002.

Gueguen, Y., Dienes, J., 1989. Transport properties of rocks from statistics and percolation. Mathematical Geology 21(1), 1-13

Heap, M.J., Lavallée, Y., Petrakova, L., Baud, P., Reuschlé, T., Varley, N.R., Dingwell, D.B., 2014. Microstructural controls on the physical and mechanical properties of edifice-forming andesites at Volcán de Colima, Mexico. Journal of Geophysical Research: Solid Earth 119(4), 2925-2963. doi: 10.1002/2013JB010521

Hoyer, S., Exner, U., Voorn, M., Rath, A., 2012. 3D Modeling of fracture flow in core samples using µ-CT data. Proceedings COMSOL Conference 2012 , Milan, Italy. http://www.comsol.com/offers/conference2012papers/papers/presentation/area/geophysics/id/13295/ [Accessed 22 May 2014]

Iassonov, P., Gebrenegus, T., Tuller, M., 2009. Segmentation of X-ray computed tomography images of porous materials: A crucial step for characterization and quantitative analysis of pore structures. Water Resources Research 45, W09415, 12pp. doi:10.1029/2009WR008087.

Iassonov, P., Tuller, M., 2010. Application of Segmentation for Correction of Intensity Bias in X-Ray Computed Tomography Images. Vadose Zone Journal 9(1), 187-191. doi:10.2136/vzj2009.0042.

Isakov, E., Ogilvie, S.R., Taylor, C.W., Glover, P.W.J., 2011. Fluid flow through rough fractures in rocks I: high resolution aperture determinations. Earth and Planetary Science Letters 191, 267-282.

Jia, L., Chen, M., Jin, Y., 2013. 3D imaging of fractures in carbonate rocks using X-ray computed tomography technology. Carbonates and Evaporites. doi: 10.1007/s13146-013-0179-9.

Kaestner, A., Lehmann, E., Stampanoni, M., 2008. Imaging and image processing in porous media research. Advances in Water Resources 31, 1174-1187. doi:10.1016/j.advwatres.2008.01.022.

Karpyn, Z.T., Grader, A.S., Halleck, P.M., 2007. Visualization of fluid occupancy in a rough fracture using micro-tomography. Journal of colloid and interface science 307(1), 181-187. doi:10.1016/j.jcis.2006.10.082.

Keller, A., 1998. High Resolution, Non-destructive Measurement and Characterization of Fracture Apertures. International Journal of Rock Mechanics and Mining Sciences 35(8), 1037-1050. doi:10.1016/S0148-9062(98)00164-8.

Ketcham, 2005. Computational methods for quantitative analysis of three-dimensional features in geological specimens. Geosphere 1(1), pp32-41. doi:10.1130/GES00001.1

Ketcham, R., Slottke, D.T., Sharp, J.M.J., 2010. Three-dimensional measurement of fractures in heterogeneous materials using high-resolution X-ray computed tomography. Geosphere 6(5), 499-514. doi: 10.1130/ges00552.1.

Ketcham, R.A., Carlson, W.D., 2001. Acquisition, optimization and interpretation of X-ray computed tomographic imagery: applications to the geosciences. Computers & Geosciences 27(4), 381-400. doi:10.1016/S0098-3004(00)00116-3.

Ketcham, R.A., Hildebrandt, J., 2014. Characterizing, measuring, and utilizing the resolution of CT imagery for improved quantification of fine-scale features. Nuclear Instruments and Methods in Physics Research B 324, 80-87.

Klinkenberg, L. J., 1941. The permeability of Porous media to liquids and gases. American Petroleum Institute, Drilling and Productions Practices, 200–213.

Klobes, P., Riesemeier, H., Meyer, K., Goebbels, J. and Hellmuth, K.-H., 1997. Rock porosity determination by combination of X-ray computerized tomography with mercury porosimetry. Fresenius' Journal of Analytical Chemistry, 357(5): 543-547.

Kulatilake, P.H.S.W., Park, J., Balasingam, P., Morgan, R., 2008. Quantification of aperture and relations between aperture, normal stress and fluid flow for natural single rock fractures. Geotechnical and Geological Engineering 26, 269-281. doi: 10.1007/s10706-007-9163-2.

Landis, E.N., Nagy, E.N., Keane, D.T., 2003. Microstructure and fracture in three dimensions. Engineering Fracture Mechanics 70(7-8), 911-925. doi:10.1016/S0013-7944(02)00157-1.

Lenoir, N., Bornert, M., Desrues, J., Bésuelle, P., Viggiani, G., 2007. Volumetric digital image correlation applied to X-ray microtomography images from triaxial compression tests on argillaceous rock. Strain 43, 193-205.

Lindeberg, T., 1998. Edge Detection and Ridge Detection with Automatic Scale Selection. International Journal of Computer Vision 30(2), 117-154. doi:10.1023/A:1008097225773.

Lorenz, C., Carlsen, I.-C., Buzug, T.M., Fassnacht, C., Weese, J., 1997. Multi-scale Line Segmentation with Automatic Estimation of Width, Contrast and Tangential Direction in 2D and 3D Medical Images. In: Troccaz, J., Grimson, E., Mösges, R. (Eds.), Proceedings First Joint Conference, Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery (CVRMed-MRCAS) 1997, Grenoble, France. Lecture Notes in Computer Science 1205, pp. 233-242.

Masaryk, P., Lintnerová, O., 1997. Diagenesis and porosity of the Upper Triassic carbonates of the pre-neogene Vienna basin basement. Geologica Carpathica 48, 371-386.

Mees, F., Swennen, R., Van Geet, M., Jacobs, P. (Eds.), 2003. Applications of Computed Tomography in the Geosciences, Special Publications (215). Geological Society, London, 250pp. ISBN: 1862391394.

Meijering, E.H.W., 2010. FeatureJ 1.6.0, Biomedical Imaging Group Rotterdam, Erasmus MC - University Medical Center Rotterdam, The Netherlands, 2002-2010. http://www.imagescience.org/meijering/software/featurej/ [Accessed 22 May 2014]

Meister, P., McKenzie, J.A., Bernasconi, S.M., Brack, P., 2013. Dolomite formation in the shallow seas of the Alpine Triassic. Sedimentology 60, 270-291. doi: 10.1111/sed.12001.

Microsoft Research, 2011. Image Composite Editor (ICE) v.1.4.4. http://research.microsoft.com/en-us/um/redmond/groups/ivm/ICE/ [Accessed 22 May 2014]

Montemagno, C.D., Pyrak-Nolte, L.J., 1999. Fracture network versus single fractures: measurement of fracture geometry with X-ray tomography. Physics and Chemistry of the Earth A, 24(7), 575-579.

Muralidharan, V., Chakravarthy, D., Putra, E., Schechter, D.S., 2004. Investigating fracture aperture distributions under various stress conditions using X-ray CT scanner. 5th Canadian Petroleum Society International Petroleum Conference, Calgary, Alberta, Canada, 2004.

Nelson, R.A., 2001. Geologic Analysis of Naturally Fractured Reservoirs (2nd ed.). Gulf Professional Publishing, Boston, Oxford, Auckland, Johannesburg, Melbourne, New Delhi. ISBN: 0-88415-317-7.

Neuville, A., Toussaint, R., Schmittbuhl, J., Koehn, D., Schwarz, J.-O., 2012. Characterization of major discontinuities from borehole cores of the black consolidated marl formation of Draix (French Alps). Hydrological Processes 26, 2085-2094. doi: 10.1002/hyp.7984.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Transactions on systems, man and cybernetics, SMC9(1): 62-66.

Peresson, H., Decker, K., 1997. The Tertiary dynamics of the northern Eastern Alps (Austria): changing palaeostresses in a collisional plate boundary. Tectonophysics 272, pp. 125-157.

Plouraboué, F., Kurowski, P., Hulin, J.-P., Roux, S., Schmittbuhl, J., 1995. Aperture of rough cracks. Physical Review E 51(3), 1675-1685.

Porter, M.L., Wildenschild, D., 2010. Image analysis algorithms for estimating porous media multiphase flow variables from computed microtomography data: a validation study. Computational Geosciences 14(1), 15-30. doi:10.1007/s10596-009-9130-5.

Rasband, W.S., 2012. ImageJ, Rasband, W.S., U. S. National Institutes of Health, Bethesda, Maryland, USA, 1997-2012. http://imagej.nih.gov/ij/ [Accessed December 13, 2012]

Röller, K. and Trepmann, C.A., 2008. Stereo32 v1.01 Software and helpfile. Ruhr-Universität Bochum, Institut für Geologie, Mineralogie & Geophysik.

Rossi, L., 2010. Medial Surface Extraction for 3D Shape Representation. Course presentation Università Ca' Foscari di Venezia, 17 May 2010. http://www.dsi.unive.it/~srotabul/files/vision/seminario_3DSkel.pdf [Accessed August 21, 2013]

Sahimi, M., 2011. Flow and transport in porous media and fractured rock: from classical methods to modern approaches (2nd ed.). Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim. ISBN: 978-3-527-40485-8.

Sato, Y., Nakajima, S., Atsumi, H., Koller, T., Gerig, G., Yoshida, S., Kikinis, R., 1997. 3D Multi-scale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images. In: Troccaz, J., Grimson, E., Mösges, R. (Eds.), Proceedings First Joint Conference, Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery (CVRMed-MRCAS) 1997, Grenoble, France. Lecture Notes in Computer Science 1205, pp. 213-222.

Scheidegger, A.E., 1974. The physics of flow through porous media (3rd ed.). University of Toronto Press, Toronto. 353pp.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.Y., White, D.J., Hartenstein, V., Eliceiri, K., Tomancak, P., Cardona, A., 2012. Fiji: an open-source platform for biological-image analysis. Nature methods 9(7), 676-682. doi:10.1038/nmeth.2019.

Schröckenfuchs, T.-C., 2012. Mikrotektonische Analyse und Reservoireigenschaften von karbonatischen Störungsgesteinen. M.Sc Dissertation, Earth Sciences, University of Vienna, Austria, 111pp.

Sellers, E., Vervoort, A., Van Cleynenbreugel, J., 2003. Three-dimensional visualization of fractures in rock test samples, simulating deep level mining excavations, using X-ray computed tomography. In: Mees, F., Swennen, R., Van Geet, M., Jacobs, P. (Eds.), Applications of X-ray Computed Tomography in the Geosciences, Special Publications (215). Geological Society, London, pp. 69-80. doi:10.1144/GSL.SP.2003.215.01.07.

Selvadurai, A.P.S., Glowacki, A., 2008. Permeability hysteresis of limestone during isotropic compression. Groundwater 46(1), 113-119. doi: 10.1111/j.1745-6584.2007.00390.x.

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

Sharifzadeh, M., Mitani, Y., Esaki, T., 2008. Rock joint surfaces measurement and analysis of aperture distribution under different normal and shear loading using GIS. Rock Mechanics and Rock Engineering 41(2), 299-323. doi: 10.1007/s00603-006-0115-6.

Sibley, D.F., Gregg, J.M., 1987. Classification of dolomite rock textures. Journal of Sedimentary Petrology 57(6), 967-975.

Singhal, B.B.S., Gupta, R.,P., 2010. Applied hydrogeology of fractured rocks (2nd ed.). Springer Science+Business Media B.V., Dordrecht, Heidelberg, London, New York. ISBN: 978-9048187890.

Stock, S.R., 1999. X-ray microtomography of materials. International Materials Reviews 44(4), 141-164. doi:10.1179/095066099101528261.

Stoessel, R., Wirjadi, O., Godehardt, M., Schlachter, A.-L., Liebscher, A., 2012. Analysis of inner fracture surfaces in CFRP based on µ-CT image data. In: Proceedings Conference on Industrial Computed Tomography (ICT) 2012, Wels, Austria, pp. 33-40.

Sukop, Michael C. and Thorne, Daniel T. Jr., 2006. Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers. Springer Berlin, Heidelberg, New York. 172 pp.

Tanikawa, W., Shimamoto, T., 2006. Klinkenberg effect for gas permeability and its comparison to water permeability for porous sedimentary rocks. Hydrology and Earth System Sciences Discussions 3, 1315-1338.

Tarasewicz, J.P.T., Woodcock, N.H., Dickson, J.A.D., 2005. Carbonate dilation breccias: Examples from the damage zone to the Dent Fault, northwest England. Geological Society of America Bulletin 117, pp.736-745. doi:10.1130/B25568.1

The Mathworks, Inc., 2011. MATLAB® R2011a, Natick, Massachusetts, USA. http://www.mathworks.com/products/matlab/ [Accessed 22 May 2014]

Van Geet, M., Swennen, R., 2001. Quantitative 3D-fracture Analysis by Means of Microfocus X-ray Computer Tomography (µCT): an example from coal. Geophysical Research Letters 28(17), 3333-3336. doi:10.1029/2001GL013247.

Van Geet, M., Swennen, R., Wevers, M., 2000. Quantitative analysis of reservoir rocks by microfocus X-ray computerised tomography. Sedimentary Geology 132(1-2), 25-36. doi:10.1016/S0037-0738(99)00127-X.

Vergés, E., Tost, D., Ayala, D., Ramos, E. and Grau, S., 2011. 3D pore analysis of sedimentary rocks. Sedimentary Geology, 234(1-4): 109-115.

Visualization Sciences Group (VSG), 2012. Amira® 5, Merignac, France. http://www.amira.com/ [Accessed December 13, 2012]

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

Watanabe, N., Ishibashi, T., Ohsaki, Y., Tsuchiya, Y., Tamagawa, T., Hirano, N., Okabe, H., Tsuchiya, N., 2011. X-ray CT based numerical analysis of fracture flow for core samples under various confining pressures. Engineering Geology 123, 338-346. doi:10.1016/j.enggeo.2011.09.010.

Wennberg, O.P., Rennan, L., Basquet, R., 2009. Computed tomography scan imaging of natural open fractures in a porous rock; geometry and fluid flow. Geophysical Prospecting 57, 239-249. doi: 10.1111/j.1365-2478.2009.00784.x.

Wessely, G., 2006a. Kalkalpen, In: Wessely, G. (Ed.), Geologie der österreichischen Bundesländer: Niederösterreich. Verlag Geologische Bundesanstalt, Vienna, pp. 105-166.

Wessely, G., 2006b. Wiener Becken, In: Wessely, G. (Ed.), Geologie der österreichischen Bundesländer: Niederösterreich. Verlag Geologische Bundesanstalt, Vienna, pp. 189-224.

Woodcock, N.H., Omma, J.E., Dickson, J.A.D., 2006. Chaotic breccia along the Dent Fault, NW England: implosion or collapse of a fault void? Journal of the Geological Society 163, 431-446. doi:10.1144/0016-764905-067

Yang, Z., Peng, X.-F., Lee, D.-J., Chen, M.-Y., 2009. An Image-Based Method for Obtaining Pore-Size Distribution of Porous Media. Environmental Science & Technology, 43(9):3248-3253.

Zabler, S., Rack, A., Manke, I., Thermann, K., Tiedemann, J., Harthill, N., Riesemeier, H., 2008. High-resolution tomography of cracks, voids and micro-structure in greywacke and limestone. Journal of Structural Geology 30, 876-887. doi:10.1016/j.jsg.2008.03.002.

Zalewska, J., Kaczmarczyk, J., Lykowska, G., 2011. Use of X-ray computed microtomography for analysis of drill cores. Nafta-Gaz, Instytut Nafta I Gazu - Polish Oil and Gas Natural Research Institute.

Zimmer, W., Wessely, G., 1996. Exploration Results in Thrust and Subthrust Complexes in the Alps and Below the Vienna Basin in Austria, In: Wessely, G., Liebl, W. (Eds.), Oil and Gas in Alpidic Thrustbelts and Basins of Central and Eastern Europe. EAGE Special Publication, London, pp. 81-107.

Appendix X1: MSHFF code user guide

The following chapter is part of the original appendix to the paper about multiscale Hessian fracture filtering shown in Chapter D. The actual code (ImageJ/FIJI macros and MATLAB® scripts) is shown in Appendix X2. All material of Appendices X1 and X2 is also available digitally at https://github.com/cageo/Voorn-2013. The original text is presented here completely, to ensure internal cohesion within the appendix.

Multiscale Hessian fracture filtering – User guide

The following manual is intended for explaining how to use implementations of the multiscale Hessian fracture filter and other data processing steps, in ImageJ/FIJI and MATLAB®. The main details of the technique are explained in the paper (Chapter D):

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

Some basic prior knowledge on the usage of ImageJ/FIJI and MATLAB® can be an advantage, but is not required. All ImageJ/FIJI macros can run with a GUI, the MATLAB® script does not.

Sections X1.1 to X1.5 cover the main part of the multiscale Hessian fracture filtering and the required preparations. Later sections cover optional additional steps (calibration, connectivity filtering and binarisation).

The consecutive numbering in the file names will – if all goes well – always correspond to the input files. For example, an input file named (e.g.) “Data_0100.tif” corresponds to output containing “Data_0100” in its filename as well.

X1.1. Included data

X1.1.1 Files for running the code, the preparations and optional additional steps

• MSHFF_Prep.ijm	ImageJ macro for ROI and control lines preparation
• MSHFF.ijm	Main multiscale Hessian fracture filtering ImageJ macro
• MSHFF_Cal.ijm	ImageJ macro for calibration using a cosine curve
• MSHFF_Conn.m	MATLAB® script for connectivity filtering
• MSHFF_Block.xls	Spreadsheet file for determining the most suitable block size

X1.1.2 Other data

• Testdata	Folder containing 100 test slices of data
• Testdata_results	Folder containing results of an analysis on above test data

X1.2. Preparation checklist before running any code

X1.2.1 Image files

- 3D dataset should be divided into a series of 2D slices, with consecutive numbering.
- Filenames should start with letters or numbers.
- 8 bit and 16 bit greyscale are supported. Other formats may work, but conversion is advised.
- "Pseudo 8 bit greyscale", 24 bit RGB images should be converted to true 8 bit greyscale first!
- The TIFF file format is recommended, although other formats may work.
- The features of interest (e.g. fractures) should be dark (have a lower greyscale value) than the surrounding material. If not, one can try to invert the image data first.
- The input data should be spatially isotropic, i.e. the resolution is equal in all 3 directions (x, y, z). This means all voxels are cubic. Anisotropy is NOT supported. During the filtering process every voxel is regarded to have a dimension of 1x1x1. The true resolution is thus not of importance for the filtering.

X1.2.2 Downsampling

- Downsampling data speeds up the filtering process severely. In the example of a 2-fold downsampling in all three directions, there is 8 times less data to process in the filtering routine. We have reached very reasonable results with filtered downsampled data compared to its filtered, full scale counterpart. From a data perspective, to prevent additional artefacts or loss of information, it is however not generally advised to use downsampling. It is therefore best to use it only for memory reasons or testing purposes. Downsampling is not included in this code, so should be carried out manually if desired.

X1.2.3 Disk space

- Make sure enough HDD space is available. The minimum amount of required disk space for the Hessian filtering process can be calculated by:

  $$Diskspace(MB) = \frac{32}{8 \cdot 1024 \cdot 1024} \cdot pixelsize\ x \cdot pixelsize\ y \cdot number\ of\ slices \cdot (number\ of\ scales + 3)$$

  The *number of scales* is the number of scales used for the Hessian filtering, i.e. the number of different apertures considered ($(s_{min}$-$s_{max})/s_{step}$, as determined in a later stage). Note the amount of required HDD space may run into the 100s of GBs, for large image datasets.

X1.2.4 Folder structure

- Create an empty folder for the processing steps. Hereafter this is termed *rootfolder*. Short paths are advised.
- Create a subfolder termed *Input*, and copy the stack of images into this folder. Do not put any other files than the image files into this folder.
- The macro codes and scripts can be copied into the *rootfolder*.

X1.2.5 Software

- Download (fiji.sc) and install FIJI.
- Open the programme and go to *Edit* → *Options* → *Memory & Threads*.
- Set the maximum memory high, i.e. as close as possible to your computer's RAM memory. The higher the memory setting, the faster the multiscale Hessian fracture filter will work.
- Close FIJI. (This is done after every step in this analysis to prevent conflicts and memory issues.)
- If connectivity filtering is required, make sure MATLAB® with the Image Processing Toolbox is installed.

X1.3. Determining analysis parameters

X1.3.1 List of chosen parameter values (to be filled in when going through the following sections):

Parameter	Value	Value to use when unspecified	Section X1.***
s_{min}		-	3.2, 3.2.1
s_{max}		-	3.2, 3.2.1
s_{step}		-	3.2, 3.2.1
$uses_{min}$		$=s_{\mathrm{min}}$	3.2.1
$uses_{max}$		$=s_{\mathrm{max}}$	3.2.1
$uses_{step}$		$=s_{\mathrm{step}}$	3.2.1
blocksize		-	3.2, 3.2.2
avgmat		Maximum greyscale (8 bit: 255, 16 bit: 65535)	3.3, 3.3.1
consthresh		0	3.3, 3.3.2
maxmat		Maximum greyscale (8 bit: 255, 16 bit: 65535)	3.3, 3.3.3
padding		0 (recommended)	3.3, 3.3.4

NOTE: The values need to be ordered $s_{min} \leq s_{max}$, $uses_{min} \leq uses_{max}$, $s_{min} \leq uses_{min}$, $uses_{max} \leq s_{max}$, $s_{step} \leq uses_{step}$ and *constresh* < *avgmat* ≤ *maxmat*. Furthermore, *blocksize* ≤ total amount of slices. Else, the values have not been determined correctly.

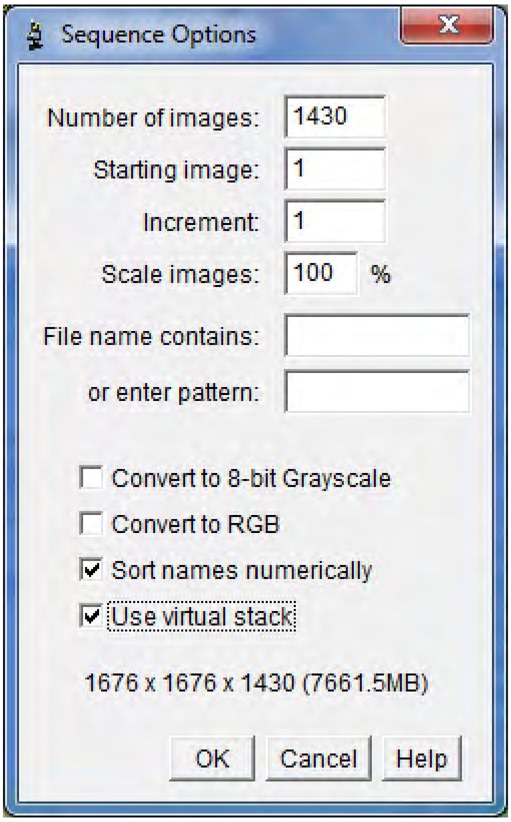
X1.3.2 Required parameters

- The following parameters are required in every analysis:

Name/symbol	Parameter	Description	Section X1.***
s_{min}	minimum Gaussian smoothing scale	relates to the minimum aperture to detect	3.2.1
s_{max}	maximum Gaussian smoothing scale	relates to the maximum aperture to detect	3.2.1
s_{step}	stepsize between Gaussian smoothing scales	controls the intervals between the apertures to detect	3.2.1
blocksize	blocksize in amount of slices	amount of data to be analysed per consecutive step	3.2.2

X1.3.2.1 Smoothing scales

- Open FIJI
- Go to *File* → *Import* → *Image Sequence*
- Browse to the *Input* folder
- Press *Open*
- In the dialog, leave all settings as they are but check *Use virtual stack*



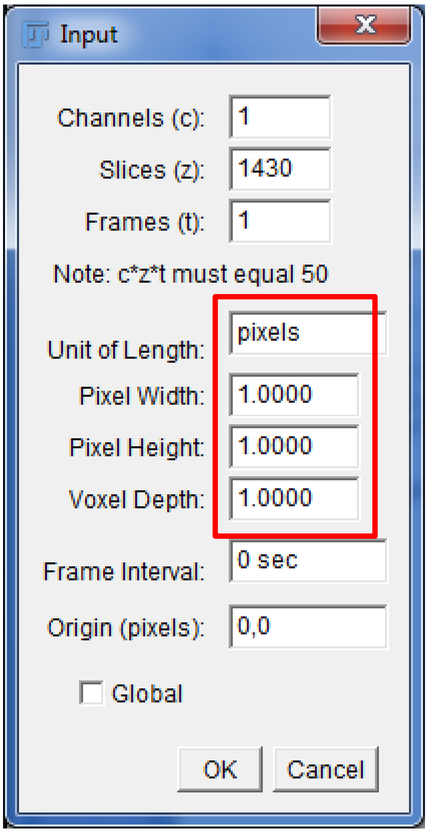
Number of images: 1430
Starting image: 1
Increment: 1
Scale images: 100 %
File name contains:
or enter pattern:

☐ Convert to 8-bit Grayscale
☐ Convert to RGB
☑ Sort names numerically
☑ Use virtual stack

1676 x 1676 x 1430 (7661.5MB)

- Press OK. The image stack is opened. The slider at the bottom can be used to go through all the images.

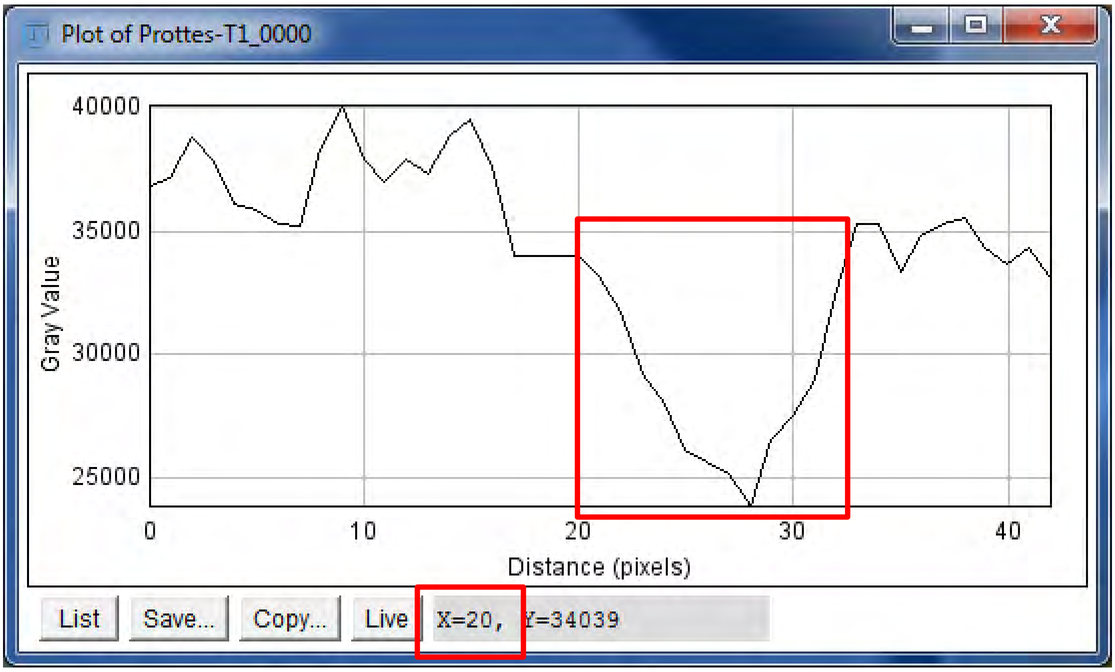
- Make sure the image properties are set to isotropic voxels. This is required for determining the correct Gaussian smoothing scales. To do this, go to *Image* → *Properties*. The following dialog box appears:



Channels (c): 1
Slices (z): 1430
Frames (t): 1
Note: c*z*t must equal 50
Unit of Length: pixels
Pixel Width: 1.0000
Pixel Height: 1.0000
Voxel Depth: 1.0000
Frame Interval: 0 sec
Origin (pixels): 0,0
☐ Global

- In the indicated area, the *Unit of Length* should be set to *pixels* (this can be typed in), and the *Pixel Width*, *Pixel Height* and *Voxel Depth* should all be set to 1. Press OK.

- Use the straight line tool to draw a line over one fracture or other planar feature of interest.
- Press CTRL + K (or *Analyze* → *Plot Profile*). A line plot appears:



- Drag the cursor over the plot. On the lower end, the relative X position can be read.
- Determine the width of the feature of interest, i.e. the width of the “valley” in the plot. In the example, this width is ca. 12 pixels.
- The best-fitting Gaussian smoothing scale for a feature is its width DIVIDED BY 2. In the example, a suitable smoothing scale would thus be 6.
- Close the plot.
- Repeat the procedure for several (representative) features and write down the results.
- Determine a suitable range of Gaussian scales (minimum scale to maximum scale, and the stepsize in between; e.g., 2 to 6 with a stepsize of 1: scales 2, 3, 4, 5 and 6 are calculated).

NOTE: Although fractions are supported, it is best to keep the smoothing scales integer. This also has a theoretical basis as voxels are discrete too.

NOTE: Of course, this approach is NOT exact and the chosen values can differ per user. This will affect the final output of the technique, but variations can be minute. Of main importance is NOT to choose too low smoothing scales (these will introduce noise) or too high smoothing scales (these will result in unrealistically broad features). The user is of course also free to use a different or more sophisticated technique to find a suitable range of smoothing scales.

NOTE: One has the option in the code to calculate more Gaussian scales than are actually used eventually. This can be beneficial for comparing for example the inclusion of an additional scale in the analysis. Every additional added scale takes longer to calculate, but calculating an additional scale separately later will take even longer! The combination of several scales can be run separate from the main code, in a later stage.

s_{min}, s_{max} and s_{step} define the scales that are calculated.
$uses_{min}$, $uses_{max}$ and $uses_{step}$ define the scales that are actually included in the final, combined result. If you do not wish these to differ, use the same values as for the calculated scales.

- Write down the chosen ranges in the table at Section X1.3.1.

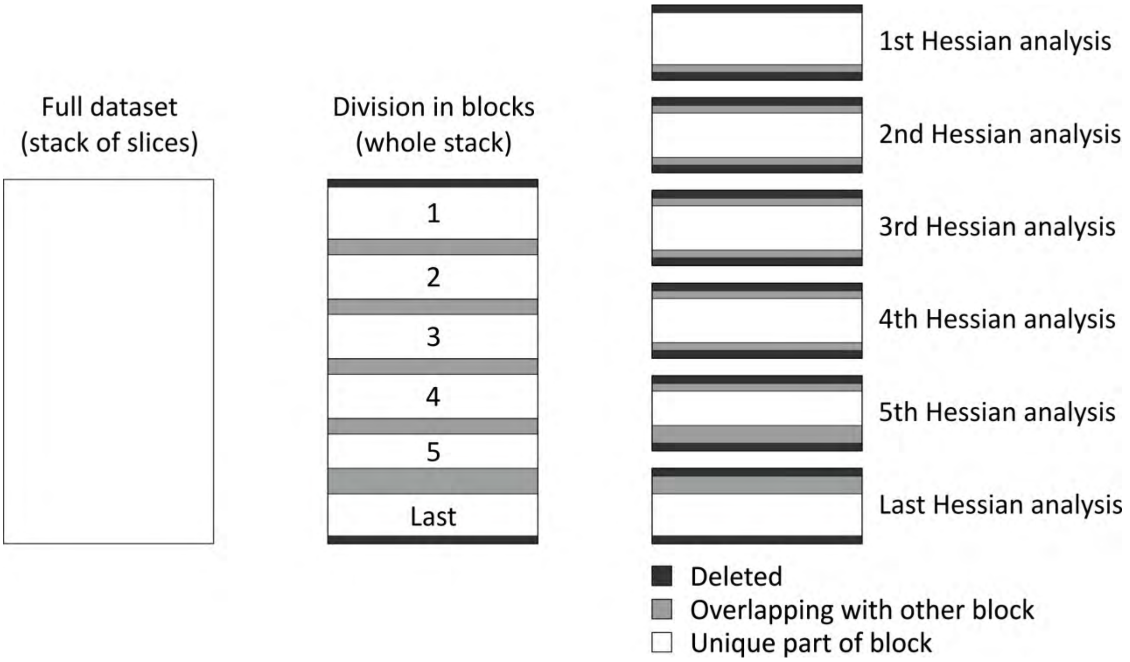
X1.3.2.2 Blocksize

- The *blocksize* can be chosen but more optimum values can be calculated.
- Open the spreadsheet MSHFF_Block.xls
- Enter the required values (RAM memory as set in FIJI (better: slightly lower), number of files (=number of slices), pixelsize x and pixelsize y, plus the maximum Gaussian smoothing scale chosen above (s_{max})).
- The maximum possible, most efficient *blocksize* will appear in green. If a value higher than the number of files is given, one should of course use this number of files (*blocksize* ≤ number of files)

NOTE: If the multiscale Hessian fracture filter still crashes on memory issues, the blocksize should be lowered. Such crashes are possible if the "optimum" blocksize results in a RAM usage very close to the maximum in FIJI, since memory overhead is difficult to assess. Therefore, to prevent issues, one can choose to set the memory in the spreadsheet slightly lower.

- Write down the chosen *blocksize* in the table at section X1.3.1.

BACKGROUND: Visual explanation of the blockwise analysis
The filtering process is characterised by a blockwise serial approach for a full 3D analysis of a large dataset, basically compensating for lack of RAM memory with disk space. Generally speaking the Hessian part of the analysis is characterised by:



The number of deleted slices is set at 2x the maximum Gaussian scale calculated (s_{max}), corresponding to the significant range of influence of the Gaussian curve on the end result in this analysis. This ensures the combined output stack is fully 3D processed.

X1.3.3 Optional parameters

- The following parameters are optional, but can improve the analysis:

Name/symbol	Parameter	Description	Section X1.***
avgmat	Average material greyscale	Greyscale value that characterises the material outside the fractures. Used for control lines.	3.3.1
consthresh	Conservative threshold of fractures	Greyscale value that estimates the greyscales present in the fractures. Used for control lines.	3.3.2
maxmat	Maximum material greyscale	Greyscale value that can be used to remove unusually bright spots in the dataset to prevent artefacts in the final output.	3.3.3
padding	Percentage of padding to be performed at start and end of input stack	First and last slices can be copied several times and included in the analysis. Extends the range of output slices but introduces "pseudo-3D" information.	3.3.4

X1.3.3.1 Average material greyscale (avgmat)

- If not opened anymore, open the image stack in FIJI as in Section X1.3.2.1
- Use one of the area selection tools □ ◯ ▱ ♡ to select a region of average material greyscale, hence NOT including fractures or porosity.
- Press CTRL + M (or *Analyze* → *Measure*). A *Results* window opens.
- In the *Results* window, there is a column termed *Mean*.

NOTE: If the mean is not shown, one can turn this measurement on under *Analyze* → *Set Measurements* from the main window of FIJI. Check *Mean gray value*, and press OK.

- Repeat the procedure for different areas, at different positions in the image stack. New measurements are added to the list.
- After taking several (at least 10-20) measurements, determine an overall mean. This can be used as the average material greyscale (*avgmat*).
- Write down *avgmat* in the table at Section X1.3.1.

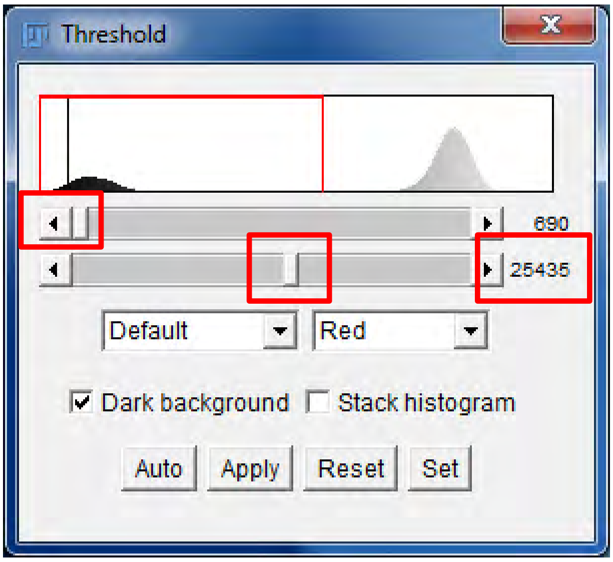
NOTE: The main use of choosing an average material greyscale is for the control lines. The results of normalisation performed during the multiscale Hessian fracture filtering are therefore more realistic relative to the input data, and easier to visualise. However, with calibration steps performed afterwards, the actual influence of the average material greyscale on the result is limited. An exact value is therefore not required.

NOTE: For the Hessian analysis, only contrast is of interest. One can therefore also choose to set the *avgmat* to the greyscale difference of the material relative to the fractures, and keep the *consthresh* value at 0. This may also be useful when a dataset contains a lot of shading, where a true *avgmat* is difficult to define

NOTE: When no *avgmat* is specified, use the values 255 (8 bit data) or 65535 (16 bit data).

X1.3.3.2 Conservative threshold for clear fractures (consthresh)

- If not opened anymore, open the image stack in FIJI as in Section X1.3.2.1.
- Press CTRL + SHIFT + A (or *Edit* → *Selection* → *Select none*) to ensure no selection is active.
- Go to *Image* → *Adjust* → *Threshold*. The following dialog appears:



690

25435

Default

Red

☑ Dark background ☐ Stack histogram

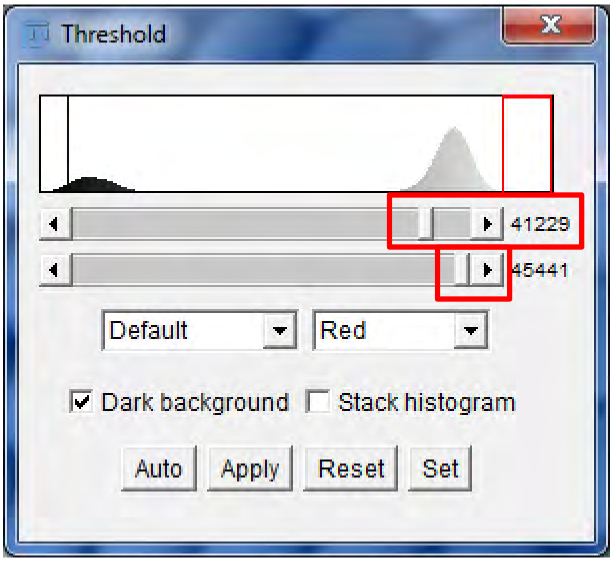
- In this dialog, one can find threshold values.
- Set the slider displayed at the top to its minimum.
- Shift the slider displayed at the bottom until a point is found where only the clearest fractures and open porosity are selected. Hence, this threshold should be very conservative.
- Test the chosen value throughout multiple locations in the image stack.
- When a proper setting is found, write down the value displayed next to the lower slider at *consthresh*, in the table at Section X1.3.1.

NOTE: When not specifying a conservative threshold here, use the value 0 in further processing steps. Also use the value 0 when using a relative value for *avgmat* (see note in Section X1.3.3.1).

NOTE: As for *avgmat*, the value used here is of main importance for the control lines and the normalising, and does therefore not have to be exact. However, this value can also be used in the calibration step, if one chooses to include a conservative threshold there. In that case, extra care for selecting a proper value (including no noise at all) should be taken here.

X1.3.3.3 Maximum material greyscale, for removing bright spots (maxmat)

- One can find in a similar way as in Section X1.3.3.2 a maximum material greyscale that can be set to remove bright spots in the data, which can otherwise introduce artefacts in the final outcome.
- Open the threshold dialog:



41229

45441

Default

Red

☑ Dark background ☐ Stack histogram

- Set the displayed at the top to its maximum.
- Shift the displayed at the bottom until only anomalously bright areas are selected.
- Test the chosen value throughout multiple locations in the image stack.
- When a proper setting is found, write down the value displayed next to the upper slider at *maxmat* in the table at Section X1.3.1.
- During the multiscale Hessian fracture filtering, the greyvalues of the here selected areas are replaced by the *maxmat* value itself.
- Close FIJI.

NOTE: When no *maxmat* is specified, use the values 255 (8 bit data) or 65535 (16 bit data).

NOTE: Although artefacts due to bright spots are reduced by this approach, they can still appear. For data with a lot of bright areas, the multiscale Hessian fracture filter may not be suitable.

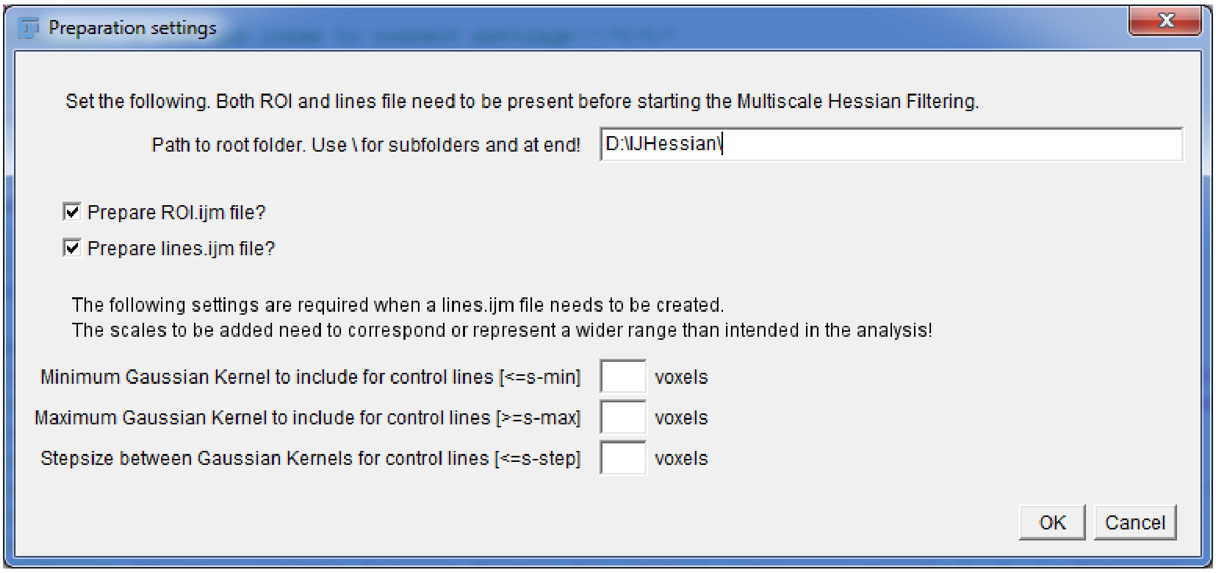
X1.3.3.4 Padding

- *Padding* can be used to extend the range of output data. This reduces the amount of deleted slices at the start and end of the stack in the output data.
- When *padding* is turned on, the first and last slices of the full stack are copied several times.
- The amount of *padding* is set as a percentage of the overlap (=the amount of deleted slices at each end, so 2x s_{max}). 0% *padding* thus means no copied slices, 100% *padding* means the first and last slices are both copied (2x s_{max}) times.
- With an increasing *padding* setting, one increases the amount of “pseudo-3D” data output at the start and end of the stack. Therefore, to ensure 3D output, it is advised to set the *padding* to 0.

NOTE: It is best to leave the *padding* at 0% to ensure 3D output. On small stacks of data, one might however choose to turn *padding* on, to not delete a relatively large amount of the stack.

X1.4. Selecting ROI and control lines settings

- Open FIJI.
- Drag and drop the macro MSHFF_Prep.ijm on the main FIJI screen (or open it manually in FIJI)
- Press *Run*. The following dialog box appears:



Set the following. Both ROI and lines file need to be present before starting the Multiscale Hessian Filtering.

Path to root folder. Use \ for subfolders and at end! D:\IJHessian\

☑ Prepare ROI.ijm file?

☑ Prepare lines.ijm file?

The following settings are required when a lines.ijm file needs to be created.
The scales to be added need to correspond or represent a wider range than intended in the analysis!

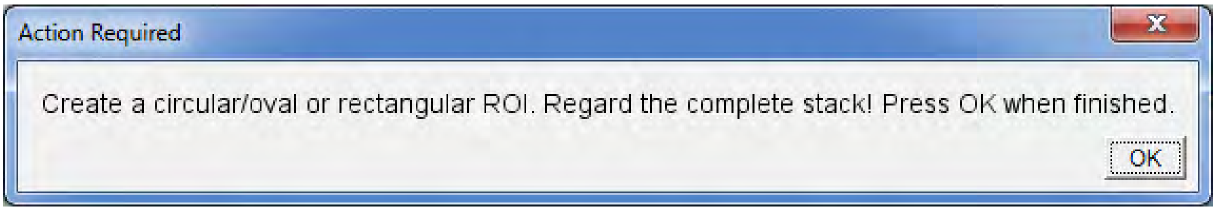
Minimum Gaussian Kernel to include for control lines [<=s-min] ☐ voxels
Maximum Gaussian Kernel to include for control lines [>=s-max] ☐ voxels
Stepsize between Gaussian Kernels for control lines [<=s-step] ☐ voxels

NOTE: When restarting the macro, previously chosen settings are NOT remembered.

- Enter the *rootfolder* (hence, WITHOUT the *Input* folder). Be sure to put a " \ " at the end of the folder name!
- Leave the two options checked. Both files are required for the multiscale Hessian fracture filtering.
- Enter a minimum and maximum Gaussian smoothing scale, and the stepsize in between.

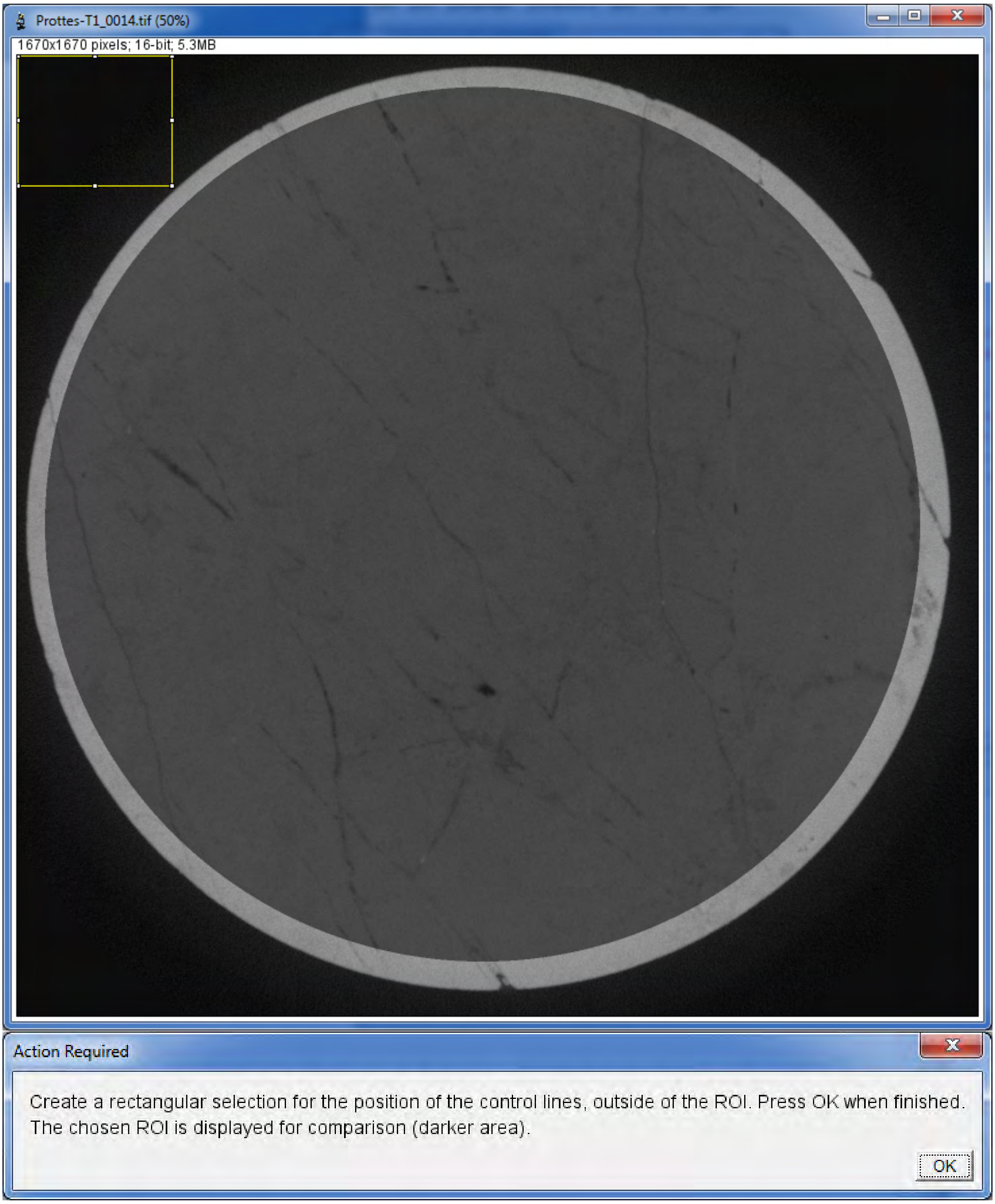
NOTE: The minimum needs to be equal or smaller than s_{min}, the maximum needs to be equal or larger than s_{max}, and the stepsize needs to be equal or an integer division of s_{step}. It is advised to keep the stepsize equal to s_{step}, and add two or more additional lines outside the range s_{min}-s_{max}.

- Press OK.
- The stack will open and a dialog box will appear.



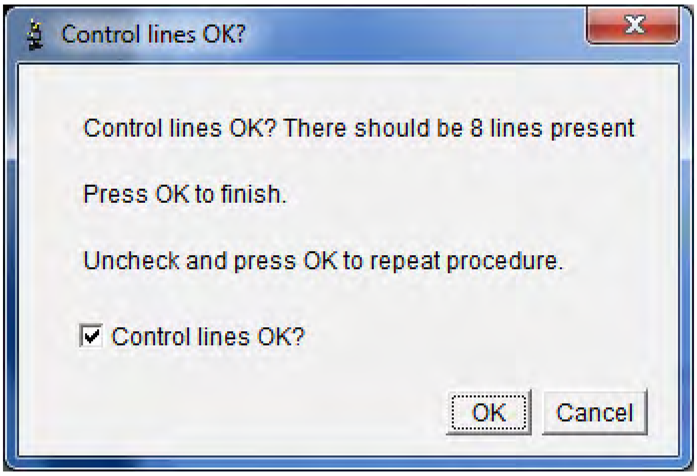
Create a circular/oval or rectangular ROI. Regard the complete stack! Press OK when finished.

- Draw a suitable ROI (oval or rectangular using ▭ ◯; keep the SHIFT key pressed for a perfect circle or square). Check if the ROI is suitable throughout the whole stack. If one wants to include the most possible data, one should create the largest possible fitting ROI. Press OK.
- The selected ROI is indicated (dark) in an image and a new dialog box appears:



Create a rectangular selection for the position of the control lines, outside of the ROI. Press OK when finished.
The chosen ROI is displayed for comparison (darker area).

- Draw a rectangular selection outside the ROI. It should be sufficiently large for the control lines (example: in image above). The best position is at the left upper corner, and completely in this corner. Press OK.
- An example is shown, as well as a dialog box:



Control lines OK? There should be 8 lines present

Press OK to finish.

Uncheck and press OK to repeat procedure.

☑ Control lines OK?

- The amount of control lines that should be present is indicated. Check if these are indeed present in the image. If all is well, leave the *Control lines OK?*-check on and press OK.
- If the choices are not correct, uncheck *Control lines OK?* and press OK. The control lines procedure will restart.
- Close FIJI. All preparation files are now created.

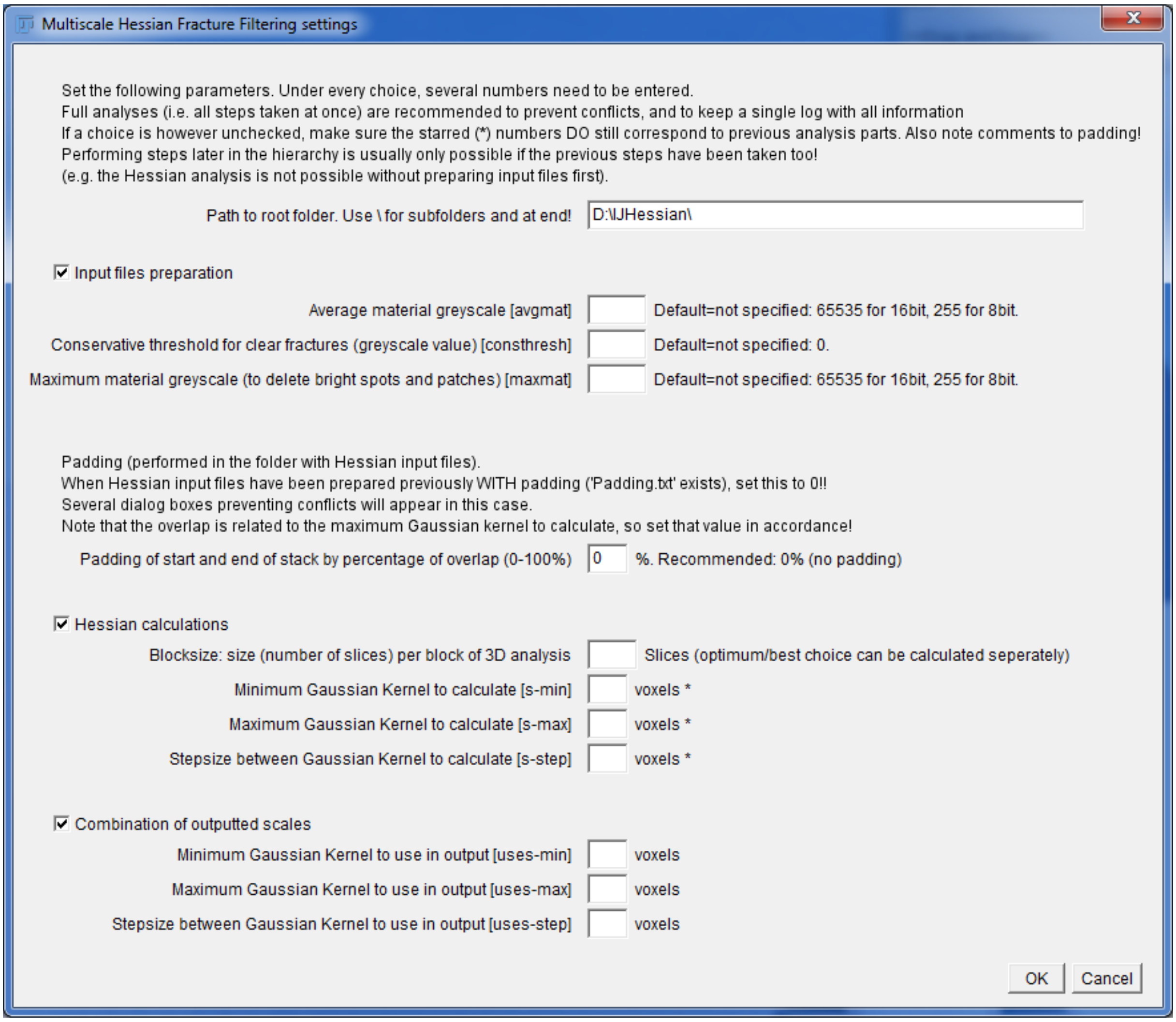
NOTE: The preparation of the input files itself is not yet done. This will be performed in the main multiscale Hessian fracture filtering step (see next section). The outside of the ROI will be set to NaN (not a number), to prevent any non-3D information to influence the end result. Note that due to this, the final ROI is somewhat smaller than the input one.

NOTE: If the control lines cannot be drawn outside the ROI, it is best to shrink the ROI, or to add an additional empty area to the input files first to make space for the control lines box.

NOTE: If the ROI is very small compared to the full images, it is advised to crop the input data first manually to a more reasonable size. This is for memory reasons. Since FeatureJ does not support calculations in a limited area only, applying the ROI does not mean the calculations are only performed inside this area. Every voxel, including all NaNs outside the ROI, require being loaded into the RAM memory in the main multiscale Hessian fracture filtering code. If cropping is applied, remember to also recalculate the *blocksize* and to redo this Section X1.4.

X1.5. Running the multiscale Hessian fracture filter

- Open FIJI.
- Drag and drop the macro MSHFF.ijm on the main FIJI screen (or open it manually in FIJI)
- Press *Run*. The following dialog box appears:



Multiscale Hessian Fracture Filtering settings

Set the following parameters. Under every choice, several numbers need to be entered.
Full analyses (i.e. all steps taken at once) are recommended to prevent conflicts, and to keep a single log with all information
If a choice is however unchecked, make sure the starred (*) numbers DO still correspond to previous analysis parts. Also note comments to padding!
Performing steps later in the hierarchy is usually only possible if the previous steps have been taken too!
(e.g. the Hessian analysis is not possible without preparing input files first).

Path to root folder. Use \ for subfolders and at end! D:\IJHessian\

☑ Input files preparation

Average material greyscale [avgmat] Default=not specified: 65535 for 16bit, 255 for 8bit.
Conservative threshold for clear fractures (greyscale value) [consthresh] Default=not specified: 0.
Maximum material greyscale (to delete bright spots and patches) [maxmat] Default=not specified: 65535 for 16bit, 255 for 8bit.

Padding (performed in the folder with Hessian input files).
When Hessian input files have been prepared previously WITH padding ('Padding.txt' exists), set this to 0!!
Several dialog boxes preventing conflicts will appear in this case.
Note that the overlap is related to the maximum Gaussian kernel to calculate, so set that value in accordance!

Padding of start and end of stack by percentage of overlap (0-100%) 0 %. Recommended: 0% (no padding)

☑ Hessian calculations

Blocksize: size (number of slices) per block of 3D analysis Slices (optimum/best choice can be calculated seperately)
Minimum Gaussian Kernel to calculate [s-min] voxels *
Maximum Gaussian Kernel to calculate [s-max] voxels *
Stepsize between Gaussian Kernel to calculate [s-step] voxels *

☑ Combination of outputted scales

Minimum Gaussian Kernel to use in output [uses-min] voxels
Maximum Gaussian Kernel to use in output [uses-max] voxels
Stepsize between Gaussian Kernel to use in output [uses-step] voxels

NOTE: When restarting the macro, previously chosen settings are NOT remembered.

NOTE: Although the three separate parts of the analysis (Input files preparation, Hessian calculations, Combination of outputted scales) can be run separately, it is strongly advised to run all analyses at once. Although some checks are built in the code, there remains a possibility of erroneous output when options are unchecked. In some cases, it may be necessary to run part of the code separately (for example, if the combination of several scales needs to be altered afterwards). In that case, make sure the same settings as used in previous analysis steps are taken!

- Set the following parameters (Assumes a full analysis. Settings as entered in Section X1.3.1, but note that the order is different):
    - *rootfolder* (hence, WITHOUT the *Input* folder). Be sure to put a " \ " at the end of the folder name!
    - *avgmat*
    - *consthresh*
    - *maxmat*
    - *padding percentage*
    - *blocksize*
    - s_{min}
    - s_{max}
    - s_{step}
    - $uses_{min}$
    - $uses_{max}$
    - $uses_{step}$
- Press OK. The analysis will start.
- Several dialog boxes will appear to check some of the settings. This also includes warnings about possible non-empty folders. On severe errors in settings or input files, the macro will terminate and (usually) display the reason.
- A detailed log-file is created and shown on screen. This log-file is automatically saved in the *rootfolder*, including the chosen settings and the progress of the code.

NOTE: This main part of the multiscale Hessian fracture filtering can take very long and require a lot of HDD space, depending on the amount and size of the input files, the chosen settings, and of course the specifications of the computer. Several examples of runtimes are given in the paper, but with a very different range of settings possible, it is difficult to calculate the runtime in advance. If the analysis really seems to take too long, consider including less high Gaussian smoothing scales, or resampling the input data.

- When finished, close FIJI to clear the RAM memory.

FOLDERS: The *rootfolder* now contains the following subfolders (assuming the full process has been performed):

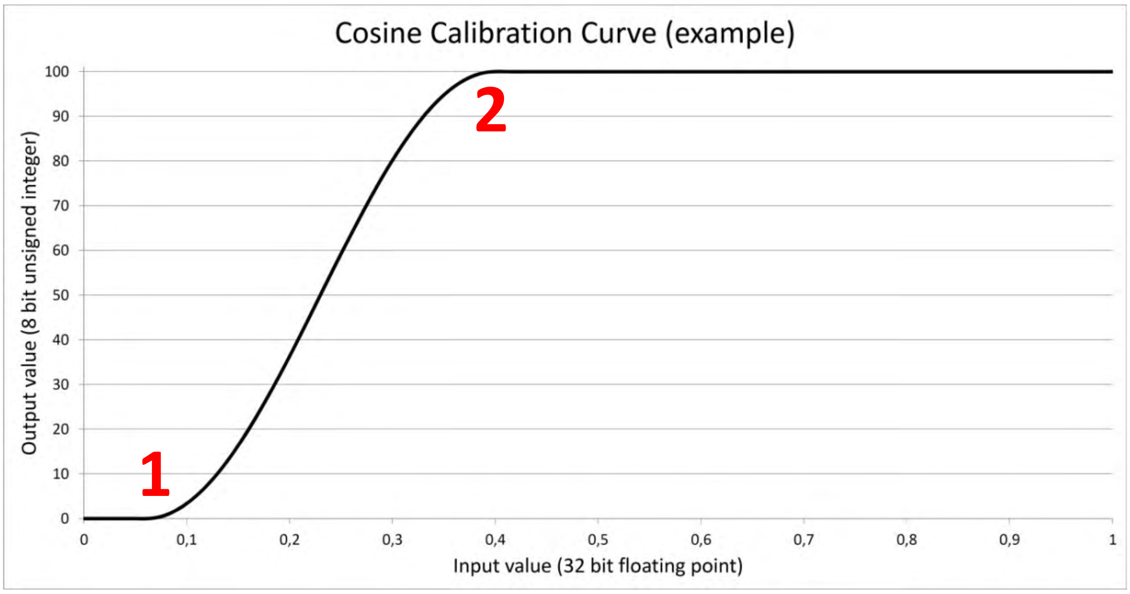
-*Input*:	Original input files.
-*InputHess*:	Altered input files with ROI applied and control lines added.
-*Calc*:	Normalised Hessian filtering results, per Gaussian smoothing scale. This folder *could* be deleted for space reasons, if the results in the *Output* folder are satisfying.
-*Output*:	Combined Hessian scales, according to the chosen settings.

X1.6. Calibration

X1.6.1 Background and required parameters

NOTE: Calibration is not required, but in many cases it is desirable. In this implementation, a cosine curve is applied on the data, and values between 0 and 100 are saved in 8 bit greyscale output files. This reflects the link to porosity for our research, and is hence tailored to this (see paper for more details).

- A curve similar to the following will be used for the calibration:

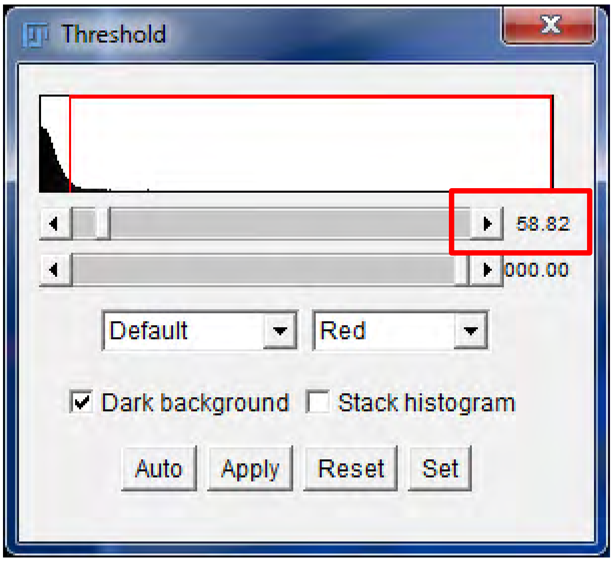


- As visible in the curve, the values from 0 to 1 in the 32 bit floating point data are replaced by output values between 0 and 100. The S-shaped curve (cosine from π to 2π) ensures softening at its start and end points.
- Two values are required:
  1) A lower cut-off for noise (*mincal*)
  2) An upper cut-off for "maximum porosity" (*maxcal*)
- The values are obtained when following the instructions below. Write them down in this table:

Parameter	**Value (between 0 and 1)**
1) *mincal*	
2) *maxcal*	

NOTE: The values require user input and introduce some ambiguity. There is no single answer on how to choose these values, and it is depending on the desired results. The following is an empirical approach to determine these values, which works rather well on the data we have.

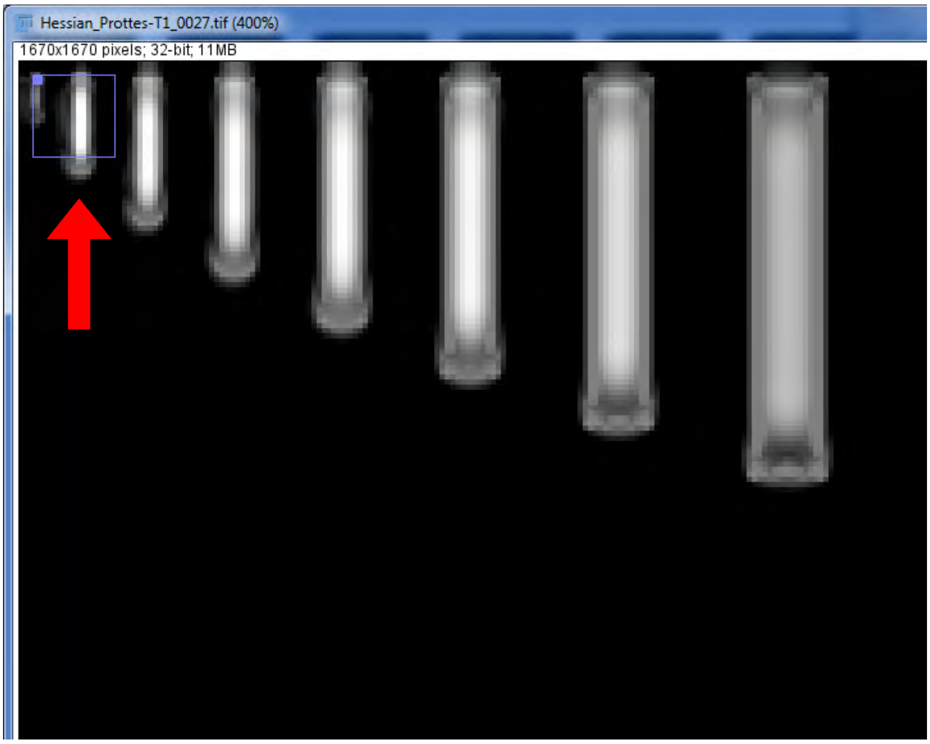
- Open FIJI.
- Open the data from the *Output*-folder (see Section X1.3.2.1). Do not use a virtual stack. If there is not enough memory to open all data, open a smaller, but representative part of the stack.
- Go to *Process* → *Math* → *Multiply*.
- Type in 1000 and press OK. This makes the selection of the values easier.
- For 1:
    - Use the Threshold sliders as in Sections X1.3.3.2 and X1.3.3.3 (*Image* → *Adjust* → *Threshold…*) to find a cut-off value for the noise. This means, finding a value where noise levels are just yet tolerable (select the fractures plus a tolerable amount of noise). For example:



58.82
000.00
Default
Red
☑ Dark background ☐ Stack histogram

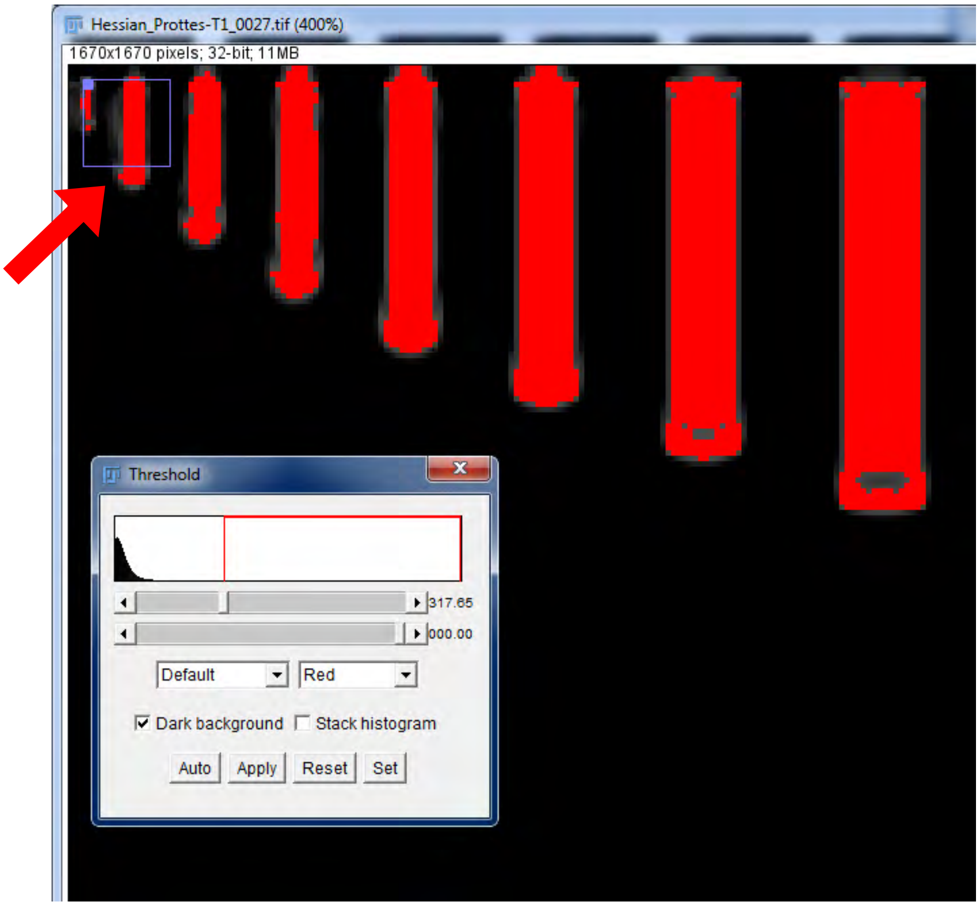
    - Note that due to the shape of the calibration curve, the effect is softened near this value, so even values slightly above the cut-off value will be outputted as very low (near 0 on a scale of 0-100).
    - Divide the found value again by 1000, and write this value down in the table at *mincal*.

- For 2:
    - Empirically, the “last threshold value before the lowest included Gaussian smoothing scale (uses$_{min}$) gets too wide” is a reasonable point for the upper cut-off value. Intensities above this value will all be set to fully porous in the calibrated output.
    - Use the zoom tool to zoom in at the control lines. For example:


Hessian_Prottes-T1_0027.tif (400%)
1670x1670 pixels; 32-bit; 11MB

-
    - In this example, 8 control lines from 1-8 were added, with a stepsize of 1. In the example output however, only scales 2-5 (uses$_{min}$-uses$_{max}$) were included. Hence, the lowest included Gaussian smoothing scale was 2, so this corresponding control line should be checked. This is indicated with the arrow.

- Use the threshold slider as in Section X1.3.3.3 to fill up the control lines with red.
- Continue until the control line of interest is just as wide as it should be. In the case of control line 2, which was originally 4 pixels wide, it should as well be 4 pixels wide.
- For example, the following is <u>too wide</u> (note the little red pixel on the left bottom):



Hessian_Prottes-T1_0027.tif (400%)

1670x1670 pixels; 32-bit; 11MB

Threshold

317.65

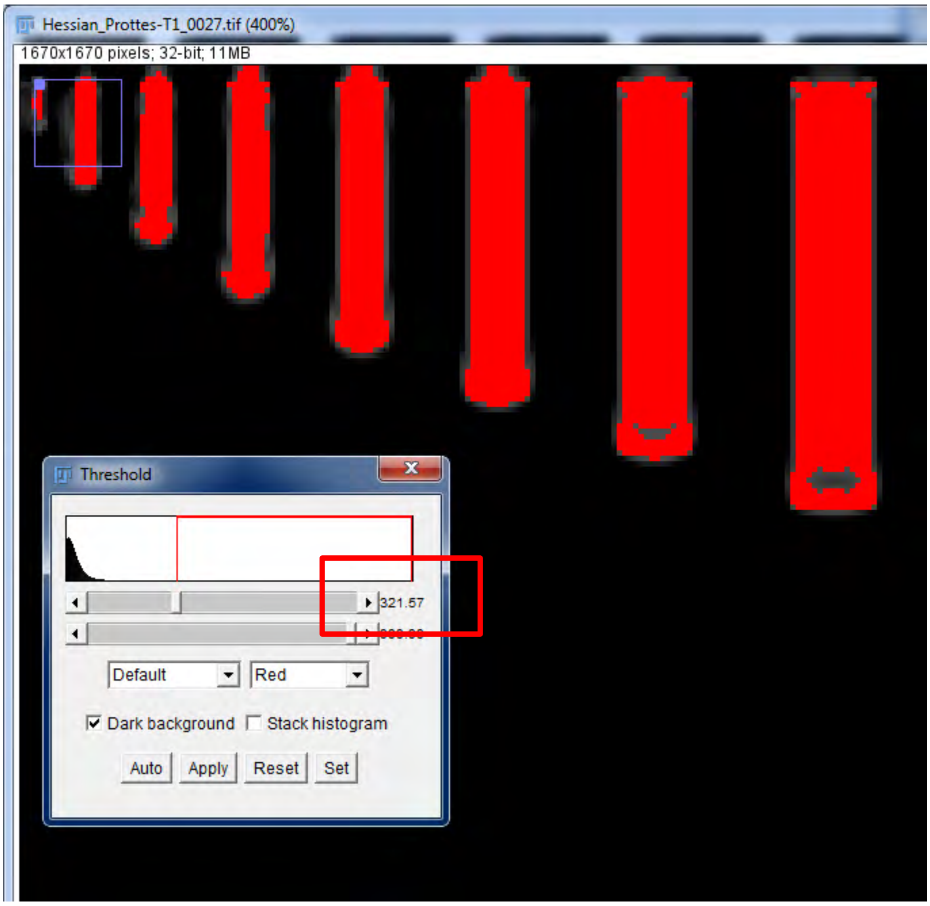
000.00

Default

Red

☑ Dark background ☐ Stack histogram

- Whereas the following (one threshold step further) is just correct:



Hessian_Prottes-T1_0027.tif (400%)

1670x1670 pixels; 32-bit; 11MB

Threshold

321.57

Default | Red

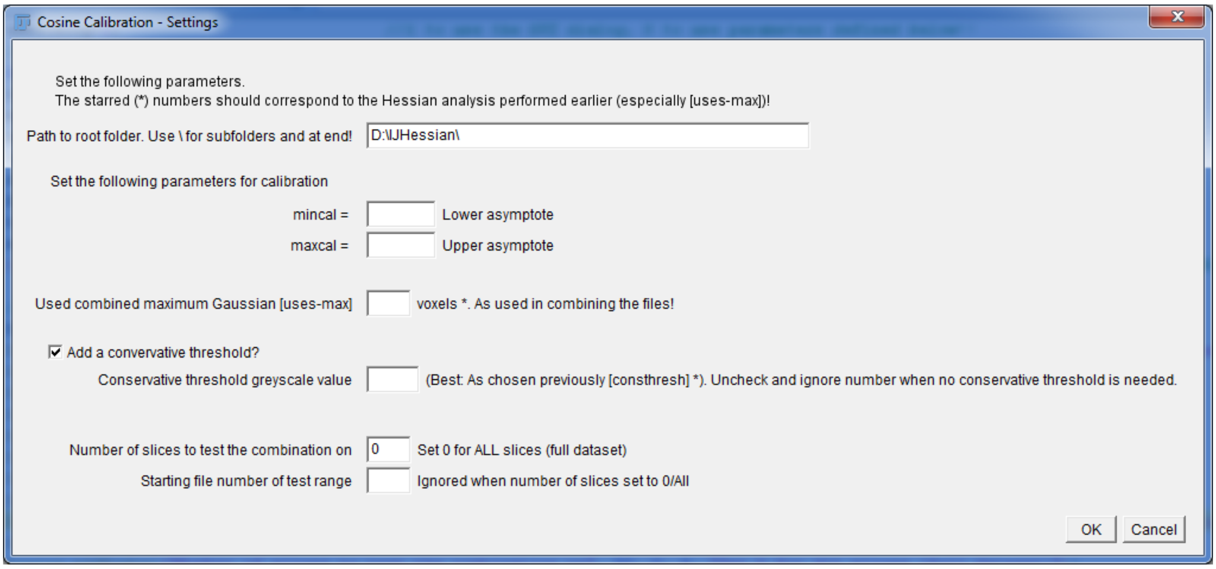
☑ Dark background ☐ Stack histogram

- In this case, one would thus find an upper cut-off of 321.
- Divide the found value again by 1000, and write this value down in the table as *maxcal*.
- Close FIJI.

NOTE: Especially for larger control lines, there may be some distortion at the end points. In that case, look mainly at the centre of the control line for determining the upper cut-off value.

X1.6.2 Applying the calibration

- Open FIJI.
- Drag and drop the macro MSHFF_Cal.ijm on the main FIJI screen (or open it manually in FIJI)
- Press *Run*. The following dialog box appears:



Set the following parameters.
The starred (*) numbers should correspond to the Hessian analysis performed earlier (especially [uses-max])!

Path to root folder. Use \ for subfolders and at end! D:\IJHessian\

Set the following parameters for calibration

mincal = Lower asymptote
maxcal = Upper asymptote

Used combined maximum Gaussian [uses-max] voxels *. As used in combining the files!

☑ Add a convervative threshold?
Conservative threshold greyscale value (Best: As chosen previously [consthresh] *). Uncheck and ignore number when no conservative threshold is needed.

Number of slices to test the combination on 0 Set 0 for ALL slices (full dataset)
Starting file number of test range Ignored when number of slices set to 0/All

NOTE: When restarting the macro, previously chosen settings are NOT remembered.

- Enter the *rootfolder*. Be sure to put a “ \ ” at the end of the folder name!
- Enter the found *mincal* and *maxcal* values.
- Enter the used maximum Gaussian smoothing scale ($uses_{max}$). This is required to shrink the ROI accordingly (to ensure 3D output).
- A conservative threshold can be added. If desired, leave the checkmark on and enter a suitable value. In most cases, one can use *consthresh*.
- The calibration can be tested. If this is desired, enter an amount of test slices, and where to start in the stack. The test is performed and displayed on screen. If not testing (i.e. processing of full dataset), set the number of test slices to 0.
- Press OK. The calibration is performed. The ROI (shrunk because of non-3D edge effects) is applied, setting the material outside the ROI to 0 as well. The control lines are removed too.
- A detailed log-file is created and shown on screen. This log-file is automatically saved in the *rootfolder*, including the chosen settings.
- Close FIJI after the calibration is finished.
- The calibrated output is saved in the folder *CosineCal*.

X1.7. Connectivity filtering (in MATLAB®)

X1.7.1 Background and required parameters

NOTE: The included MATLAB® script is provided "as-is". No additional information on the usage of MATLAB® will be given here. The script assumes all previous processing steps have been taken, including a calibration (i.e. input data must be 8 bit TIFF). Current implementation DOES require a lot of RAM memory and may hence not work on every dataset.

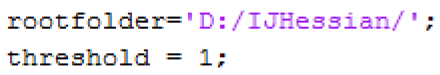
- A threshold is set to binarise the data for the use in MATLAB.
- The MATLAB® script looks for connected clusters in the z (consecutive slices) direction of the input data, using *bwconncomp*. All clusters are labelled.
- Only the clusters present in both the first and last slice of data are kept. These are connected throughout the stack.
- After only the fully connected clusters remain, the intensities from the calibration are taken again (instead of the binary intermediate step). Output files are saved.

- To find a suitable threshold, use the threshold slider in FIJI on the calibrated data. If any noise is left in the data, this should be excluded by setting a threshold higher than this noise.
- The lowest possible – and default – threshold value is here 0.9. This assumes, all calibrated output values of 1 to 100 are considered porous for the connectivity testing.

NOTE: In MATLAB®, the threshold value itself is NOT included in the selected material, whereas in ImageJ and FIJI, the threshold value IS included. This means for example on 8 bit data, a threshold of 0.9 in MATLAB® yields the same result as a threshold of 1 in ImageJ and FIJI (in both cases, all values above 1, including 1 itself, are selected).

X1.7.2 Applying the connectivity filter

- Open the script MSHFF_Conn.m in MATLAB®.
- Enter the *rootfolder* (with forward slashes /, including one at the end) and the desired threshold (Section X1.7.1). For example:



```
rootfolder='D:/IJHessian/';
threshold = 1;
```

- Run the script (▶).
- The progress of the script is indicated.
- When the script is finished, MATLAB® can be closed.
- The connectivity filtered output files are saved in the subfolder *Connected*.

X1.8. Binarisation

No comments are made here for the best way to binarise the data eventually, if this is desired. Many different approaches exist, and different ways can be suitable for different datasets. To apply a threshold, one can open a stack in FIJI, choose a threshold value (as done in previous sections), and press *Apply*. Go to *File* → *Save As* → *Image Sequence* to save the output manually. One can enter a name or use the names of the input files.

Appendix X2: MSHFF code

The following subsections show the ImageJ/FIJI Macro and MATLAB® codes. A description of these codes is shown in Appendix X1.1.1. All material of Appendices X1 and X2 is also available digitally at https://github.com/cageo/Voorn-2013

X2.1. MSHFF Preparation (FIJI Macro)

```
//MSHFF_Prep.ijm

//Multiscale Hessian fracture filtering: Preparation
//Maarten Voorn, December 2012

requires("1.47b");
usedialog=1; //1 is use dialog, 0 is use settings below.

if (usedialog==0) {
	root='D:\\Rootfolder\\';
	doroifile=1;
	dolinesfile=1;
	ming=1;
	maxg=8;
	stepg=1;
} else {
	Dialog.create("Preparation settings");
	Dialog.addMessage("Set the following. Both ROI and lines file need to be
present before starting the Multiscale Hessian Filtering.");
	Dialog.addString("Path to root folder. Use \\ for subfolders and at end!",
"D:\\Rootfolder\\",50);
	Dialog.addCheckbox("Prepare ROI.ijm file?", true);
	Dialog.addCheckbox("Prepare lines.ijm file?", true);
	Dialog.addMessage("The following settings are required when a lines.ijm file
needs to be created.\nThe scales to be added need to correspond or represent a
wider range than intended in the analysis!");
	Dialog.addNumber("Minimum Gaussian Kernel to include for control lines [<=s-
min]", 1, 0, 2, "voxels");
	Dialog.addNumber("Maximum Gaussian Kernel to include for control lines [>=s-
max]", 8, 0, 2, "voxels");
	Dialog.addNumber("Stepsize between Gaussian Kernels for control lines [<=s-
step]", 1, 0, 2, "voxels");
	Dialog.show;
	root=Dialog.getString();
	doroifile=Dialog.getCheckbox();
	dolinesfile=Dialog.getCheckbox();
	ming=Dialog.getNumber();
	maxg=Dialog.getNumber();
	stepg=Dialog.getNumber();
}

if (endsWith(root, "\\")==0) {exit("Root folder filename is not correct! Does it
end with \\ ? Macro aborted")}
saveroifile=root+"ROI.ijm";
savelinesfile=root+"Lines.ijm";
inputfolder=root+"Input\\";
inputfiles=getFileList(inputfolder);
numfiles=inputfiles.length;


numtestfile=round(numfiles/2);                       //Take a slice halfway the
stack
testfile=inputfolder+inputfiles[numtestfile];

//ROI file generation
if (doroifile==true) {
```

```
run("Image Sequence...", "open=&inputfolder number=&numfiles starting=1 increment=1
scale=100 file=[] or=[] sort use");

setTool(1);
waitForUser("Create a circular/oval or rectangular ROI. Regard the complete stack!
Press OK when finished.");
getSelectionBounds(xroi,yroi,widthroi,heightroi);
seltype=selectionType();

if (widthroi==getWidth && heightroi==getHeight) {
	close();
	IJ.log("\\Close");
	exit("A rectangular selection of the WHOLE image was made. Macro
aborted.\nConflict with the main code, since the control lines must lie outside the
ROI.");
}

IJ.log("\\Close");
print('//Automatically generated file for ROI for multiscale Hessian fracture
filtering');

if (seltype==0) {
	print("makeRectangle(",xroi,",",yroi,",",widthroi,",",heightroi,")");
} else if (seltype==1) {
	print("makeOval(",xroi,",",yroi,",",widthroi,",",heightroi,")");
} else {
	close();
	IJ.log("\\Close");
	exit("No selection, or no rectangular or oval selection made! Macro
aborted.");
}

close();
selectWindow("Log");
run("Text...", "save=saveroifile");
}

//Control lines file generation
if (dolinesfile==false) {
	IJ.log("\\Close");
	if (doroifile==true) {
		exit("ROI file generated. Control lines file creation turned off.");
	} else exit("No files generated since all options are turned off!");
} else {
endok=false;

while (endok==false) {
linespacey=ming;				//Varies later
linewidth=ming*2;
linelength=0;
open(testfile);
if (File.exists(saveroifile)==false) {
	close();
	exit("No ROI file found! Generate this file first. Macro aborted!");
}
runMacro(saveroifile);
run("Divide...", "value=2");
setTool(0);

waitForUser("Create a rectangular selection for the position of the control lines,
outside of the ROI. Press OK when finished.\nThe chosen ROI is displayed for
comparison (darker area).");
getSelectionBounds(x1,y1,width1,height1);

//Starting file creation
IJ.log("\\Close");
print('//Automatically generated file for control lines for multiscale Hessian
analysis');
//Do NOT change the following first lines
print('arguments=getArgument();');
print('avgmat=substring(arguments,0,5);');
       //Substring positions are correct like this
print('fracthresh=substring(arguments,5,10);');
       //Counting starts at zero. Start: index1, End: index2-1
print('//____________________________________');
print("makeRectangle(", x1,",", y1,",",width1,",",height1,");");
print('run("Set...", "value=avgmat");');

//Generates control lines
for (i=(y1+4*linespacey); i<(y1+height1-linelength); i+=linespacey) {
       for (j=(x1+3*linewidth); j<(x1+width1-2*linewidth); j+=2*linewidth) {
              linelength=5*linewidth;
       //5:1 linelength:linewidth
              print("makeRectangle(", j,",", i,",",linewidth,",",linelength,");");
              print('run("Set...", "value=fracthresh");');
       //Single ' for printing text WITH "
              linewidth=linewidth+(2*stepg);
       //Max response at 2*Gaussian scale

              if (linewidth>(2*maxg)) {                                   //Breaks
out of loops when condition forfilled. Max response at 2*Gaussian scale
                     j=1e99;
                     i=1e99;
              }
       }
       linespacey=2*linewidth+linelength;
}

print("makeRectangle(", x1,",", y1,",",width1,",",height1,");
       //Repetition required to pass box to main code");           //Repetition
required for main code

selectWindow("Log");
run("Text...", "save=savelinesfile");



linearguments="65535"+"00000";
runMacro(savelinesfile, linearguments);

numlines=floor((maxg-ming)/stepg)+1;
numlinestext="Control lines OK? There should be "+numlines+" lines present";
Dialog.create("Control lines OK?");
Dialog.addMessage(numlinestext);
Dialog.addMessage("Press OK to finish.");
Dialog.addMessage("Uncheck and press OK to repeat procedure.");
Dialog.addCheckbox("Control lines OK?", true);
Dialog.show();

if (Dialog.getCheckbox()==true) {
       close();
       endok=true;
       IJ.log("\\Close");
} else {
       close();
}
}
}
```

X2.2. MSHFF Main code (FIJI Macro)

```
//MSHFF.ijm

//Multiscale Hessian fracture filtering: Main code
//Maarten Voorn, December 2012

requires("1.47b");
usedialog=1;                     //1 to use the GUI dialog, 0 to use parameters
defined below!!

if (usedialog==0) {
root='D:\\Rootfolder\\';          // Root folder with files. Note double backslash,
also at end!
avgmat=65535;                    // Average material greyscale. Defaults (not
specified): 65535 for 16bit, 255 for 8bit.
fracthresh=0;                    // Conservative threshold for clear fractures.
Default (not specified): 0.
maxmat=65535;                    // Maximum material greyscale to delete bright
spots. Defaults (not specified): 65535 for 16bit, 255 for 8bit.
padding=0;                       // Amount of padding at start and end of stack to
reduce deletion. 0-100% (over overlap). Recommended: 0 (no padding)!!
blocksize=100;                          // Size (number of slices) for blocks in 3D
Hessian analysis.
ming=2;                                 // Minimum Gaussian kernel to calculate.
maxg=6;                                 // Maximum Gaussian kernel to calculate.
Also controls overlap!!!
stepg=1;                         // Stepsize between consecutive Gaussian kernels
to calculate.
useming=2;                       // Eventually used min Gaussian kernel.
usemaxg=6;                       // Eventually used max Gaussian kernel.
usestepg=1;                      // Eventually used stepsize between consecutive
Gaussian kernels.
doprep=1;                        // Create preparation files (1=yes, 0=no)
dohess=1;                        // Perform Hessian calculation (1=yes, 0=no)
docombi=1;                       // Combine multiple scales outputted by Hessian
calculation (1=yes, 0=no)
}
else {

Dialog.create("Multiscale Hessian Fracture Filtering settings");
Dialog.addMessage("Set the following parameters. Under every choice, several
numbers need to be entered.\nFull analyses (i.e. all steps taken at once) are
recommended to prevent conflicts, and to keep a single log with all information\nIf
a choice is however unchecked, make sure the starred (*) numbers DO still
correspond to previous analysis parts. Also note comments to padding!\nPerforming
steps later in the hierarchy is usually only possible if the previous steps have
been taken too!\n(e.g. the Hessian analysis is not possible without preparing input
files first).");
Dialog.addString("Path to root folder. Use \\ for subfolders and at end!",
"D:\\Rootfolder\\", 50);
Dialog.addCheckbox("Input files preparation", true);
Dialog.addNumber("Average material greyscale [avgmat]", 65535, 0, 5, "Default=not
specified: 65535 for 16bit, 255 for 8bit.");
Dialog.addNumber("Conservative threshold for clear fractures (greyscale value)
[consthresh]", 0, 0, 5, "Default=not specified: 0.");
Dialog.addNumber("Maximum material greyscale (to delete bright spots and patches)
[maxmat]", 65535, 0, 5, "Default=not specified: 65535 for 16bit, 255 for 8bit.");
Dialog.addMessage("");
Dialog.addMessage("Padding (performed in the folder with Hessian input
files).\nWhen Hessian input files have been prepared previously WITH padding
('Padding.txt' exists), set this to 0!!\nSeveral dialog boxes preventing conflicts
will appear in this case.\nNote that the overlap is related to the maximum Gaussian
kernel to calculate, so set that value in accordance!");
Dialog.addNumber("Padding of start and end of stack by percentage of overlap (0-
100%)", 0,0,3, "%. Recommended: 0% (no padding)");
Dialog.addMessage("");
Dialog.addCheckbox("Hessian calculations", true);
Dialog.addNumber("Blocksize: size (number of slices) per block of 3D analysis",
100, 0, 4, "Slices (optimum/best choice can be calculated seperately)");
Dialog.addNumber("Minimum Gaussian Kernel to calculate [s-min]", 2, 0, 2, "voxels
*");
Dialog.addNumber("Maximum Gaussian Kernel to calculate [s-max]", 6, 0, 2, "voxels
*");
Dialog.addNumber("Stepsize between Gaussian Kernel to calculate [s-step]", 1, 0, 2,
"voxels *");
Dialog.addMessage("");
Dialog.addCheckbox("Combination of outputted scales", true);
Dialog.addNumber("Minimum Gaussian Kernel to use in output [uses-min]", 2, 0, 2,
"voxels");
Dialog.addNumber("Maximum Gaussian Kernel to use in output [uses-max]", 6, 0, 2,
"voxels");
Dialog.addNumber("Stepsize between Gaussian Kernel to use in output [uses-step]",
1, 0, 2, "voxels");
Dialog.show();

root=Dialog.getString();
doprep=Dialog.getCheckbox();
avgmat=Dialog.getNumber();
fracthresh=Dialog.getNumber();
maxmat=Dialog.getNumber();
padding=Dialog.getNumber();
dohess=Dialog.getCheckbox();
blocksize=Dialog.getNumber();
ming=Dialog.getNumber();
maxg=Dialog.getNumber();
stepg=Dialog.getNumber();
docombi=Dialog.getCheckbox();
useming=Dialog.getNumber();
usemaxg=Dialog.getNumber();
usestepg=Dialog.getNumber();
}

//__________________________________________________________________________________
___________________________
//Testing settings. Defining locations and number of files. Starting logfile.
IJ.log("\\Close");                //Closes the log-window (if open)
if (endsWith(root, "\\")==0) {exit("Root folder filename is not correct! Does it
exist, and does it end with \\ ? Macro aborted")}
run("FeatureJ Options", "  progress");               //Setting FeatureJ to work
properly with macro
if (docombi==1) {                              //Tests required for combining.
       if (useming==usemaxg) {
              //Allows but warns for single Gaussian kernel.
              Dialog.create("Warning on selected Gaussian kernels");
              Dialog.addMessage("Only 1 Gaussian kernel to combine selected. The
analysis is hence not 'multiscale'.\nPress OK to continue anyway. Press Cancel to
abort macro.");
              Dialog.show();
       }
       if (useming<ming||useming>maxg) {exit("Gaussian kernels to combine [uses-min,
uses-max] are outside of range of calculated Gaussian kernels [s-min, s-max]. Macro
aborted.");}
       if (usemaxg>maxg||usemaxg<ming) {exit("Gaussian kernels to combine [uses-min,
uses-max] are outside of range of calculated Gaussian kernels [s-min, s-max]. Macro
aborted.");}

       selnum=((usemaxg-useming)/usestepg)+1;

       if (selnum/round(selnum)!=1) {
              //Check for integer
              exit("Range of Gaussian kernels to combine does not correspond to an
integer amount of scales. Revise!\nExample:\nSteps 2 to 5 with a stepsize of 2 is
wrong; gives [2,4] but leaves 'half a scale'.\nSteps 2 to 4 with a stepsize of 2 is
correct, and gives [2,4].\nMacro aborted.");
       }
```

```
	teststepg=usestepg/stepg;
	if (teststepg/round(teststepg)!=1) {
		exit("The stepsize in Gaussian kernels to combine [uses-step] is not a
multiple of the stepsize of Gaussian kernels to calculate [s-step]! Macro
aborted");
	}
}

if ((blocksize-4*maxg)<1) {exit("Blocksize is set too small with current maximum
Gaussian scale (required overlap is larger than blocksize). Macro aborted.");}
//Take overlap = 2*maxg
inputfolder=root+"Input\\";
inputfiles=getFileList(inputfolder);
numfiles=inputfiles.length;
if (blocksize>numfiles) {exit("Blocksize is larger than total number of files. Only
blocksize <= Number of files is supported. Macro aborted.");}
inputhessfolder=root+"InputHess\\";
checkempty=getFileList(inputhessfolder);
if (doprep==0 && dohess==0 && docombi==0) {
	exit("No analyses selected! Macro aborted.");
}
if (doprep==1) {
	if (checkempty.length>0) {
		Dialog.create("Folder not empty");
		Dialog.addMessage("The InputHess-folder is not empty. Press OK to
delete files. Press Cancel to abort macro.");
		Dialog.show();
		for (i=0; i<checkempty.length; i++)
{ok=File.delete(inputhessfolder+checkempty[i]);}
	}
} else if (doprep==0&&dohess==1) {
	if (checkempty.length==0) {
		exit("There are no Hessian input files (InputHess-folder is
empty)\nand the generation of input files is turned off! Macro aborted.");
	}
}
File.makeDirectory(inputhessfolder);
calcfolder=root+"Calc\\";
checkempty=getFileList(calcfolder);
if (dohess==1) {
	if (checkempty.length>0) {
		Dialog.create("Folder not empty");
		Dialog.addMessage("The Calc-folder is not empty. Press OK to delete
files. Press Cancel to abort macro.");
		Dialog.show();
		for (i=0; i<checkempty.length; i++)
{ok=File.delete(calcfolder+checkempty[i]);}
	}
} else if (dohess==0&&docombi==1) {
	if (checkempty.length==0) {
		exit("There are no files outputted by the Hessian analysis (Calc-
folder is empty)\nand the generation of these files is turned off! Macro
aborted.");
	}
}
File.makeDirectory(calcfolder);
outputfolder=root+"Output\\";
checkempty=getFileList(outputfolder);
if (docombi==1) {
	if (checkempty.length>0) {
		Dialog.create("Folder not empty");
		Dialog.addMessage("The Output-folder is not empty. Press OK to delete
files. Press Cancel to abort macro.");
		Dialog.show();
		for (i=0; i<checkempty.length; i++)
{ok=File.delete(outputfolder+checkempty[i]);}
	}
}
```

```
File.makeDirectory(outputfolder);

//Check padding file etc.
if (doprep==1) {
	ok=File.delete(root+"Padding.txt");	// When new Hessian input files are
generated, any previous padding file is deleted. ok= added to suppress output in
log
}
paddone=File.exists(root+"Padding.txt");
if (paddone==1) {
	padsettingprear=split(File.openAsString(root+"Padding.txt"),"\t\n");
	padsettingpre=padsettingprear[3];
	paddialogtext="Previously used padding of "+padsettingpre+" slices at both
start and end of stack detected in 'Padding.txt'.";
}
if (padding>0 && paddone==1) {
	exit("Padding conflict. Padding is turned on, but it has already been
performed previously ('Padding.txt' exists)!\nApply one of the following
options:\n1) To use the padding selected in a previous analysis, set the padding to
0.\n2) Delete 'Padding.txt' and the padded slices manually to perform the padding
with new settings.\n3) Restart the analysis with the generation of Hessian input
files turned ON (recommended, slowest but safest choice).\nMacro aborted.");
} else if (padding==0 && paddone==1) {
	Dialog.create("Previous padding detected");
	Dialog.addMessage("Possible padding conflict. Padding is turned off, but has
been applied in a previous stage ('Padding.txt' exists)!");
	Dialog.addMessage(paddialogtext);
	Dialog.addMessage("In current setup, the previous padding WILL affect the end
results.\nIf this is not desired, abort the macro and apply one of the following
options:\n1) Delete 'Padding.txt' and the padded slices manually to perform the
padding with new settings.\n2) Restart the analysis with the generation of Hessian
input files turned ON (recommended, slowest but safest choice).\nPress OK to
continue. Press Cancel to abort macro.");
	Dialog.show();
}
//End of padding checking

ROImacro=root+"ROI.ijm";
if (File.exists(ROImacro)==false) {exit("No ROI-file [ROI.ijm] found. Macro
aborted.");}
linesmacro=root+"Lines.ijm";
if (File.exists(linesmacro)==false) {exit("No lines-file [Lines.ijm] found. Macro
aborted.");}

getDateAndTime(year1,month1,dayofweek1,dayofmonth1,hour1,minute1,second1,msec1);
month1=month1+1;
if (month1<10) {month1b="0"+month1; } else {month1b=month1;}
if (dayofmonth1<10) {dayofmonth1b="0"+dayofmonth1; } else
{dayofmonth1b=dayofmonth1; }
if (hour1<10) {hour1b="0"+hour1; } else {hour1b=hour1; }
if (minute1<10) {minute1b="0"+minute1; } else {minute1b=minute1; }
if (second1<10) {second1b="0"+second1; } else {second1b=second1; }
timestart="Macro started: "+year1+"-"+month1b+"-"+dayofmonth1b+"
"+hour1b+":"+minute1b+":"+second1b;
print("3D Multiscale Hessian Fracture Filtering");
print(timestart);
print("___________________________________");
print(">Used settings<");
print("Input folder:", inputfolder);
print("Folder for input for Hessian calculation:", inputhessfolder);
print("Folder for Hessian calculation:", calcfolder);
print("Folder for final output:", outputfolder);
print("Number of files to be analysed:", numfiles);
if (doprep==1) {
	print("Input files for Hessian calculation will be created");
	print("**Average material greyscale (used for calibration of the Hessian
output) [avgmat]:", avgmat);
```

```
	print("**Conservative threshold for clear fractures [consthresh]:", 
fracthresh);
	print("**Maximum material greyscale; to exclude bright spots [maxmat]:", 
maxmat);
} else {
	print("Input files for Hessian calculation have already been created 
earlier");
}

if (padding>0) {
	print("Padding turned on with",padding,"% overlap");
} else if (padding==0 && paddone==1) {
	print("Padding turned off but done in a previous stage.");
	print("**",paddialogtext, "For more details, check earlier logs.");
} else {
	print("Padding turned off");
}

if (dohess==1) {
	print("The Hessian calculations will be performed");
	print("**Hessian calculations for Gaussian kernels from", ming, "to", maxg, 
", with stepsize", stepg);
	print("**Size of analysis blocks (Hessian):", blocksize);
} else if (dohess==0 && docombi==1) {
	print("The Hessian calculations have already been performed earlier");
	print("**Hessian calculations were for Gaussian kernels from", ming, "to", 
maxg, ", with stepsize", stepg, "Make sure this is correct!");
} else {
	print("The Hessian calculations will not be performed");
}
if (docombi==1) {
	print("The results of the Hessian calculations on various Gaussian kernels 
will be combined");
	print("**Hessian calculations to be included in final output: Gaussian 
kernels from", useming, "to", usemaxg, ", with stepsize", usestepg);
} else {
	print("The results of the Hessian calculations on various Gaussian kernels 
will not be combined");
}
print("___________________________________");
//_________________________________________________________________________________
___________________________________________________
//Converting avgmat & fracthresh to proper text strings for passing to other macros 
(must have 5 positions)
if (avgmat<10) {avgmatstr="0000"+avgmat;} else if (avgmat<100) 
{avgmatstr="000"+avgmat;} else if (avgmat<1000) {avgmatstr="00"+avgmat;} else if 
(avgmat<10000) {avgmatstr="0"+avgmat;} else {avgmatstr=toString(avgmat);};
if (fracthresh<10) {fracthreshstr="0000"+fracthresh;} else if (fracthresh<100) 
{fracthreshstr="000"+fracthresh;} else if (fracthresh<1000) 
{fracthreshstr="00"+fracthresh;} else if (fracthresh<10000) 
{fracthreshstr="0"+fracthresh;} else {fracthreshstr=toString(fracthresh);};
//_________________________________________________________________________________
___________________________________________________
//Checking the ROI and adding control lines for the Hessian analysis, checking 
conservative threshold
print("Check the ROI and control lines: Awaiting input...");
numtestfile=round(numfiles/2);                       //Take a slice halfway the 
stack
testfile=inputfolder+inputfiles[numtestfile];
open(testfile);
rename("TestROIandLines");
linearguments=avgmatstr+fracthreshstr;
runMacro(linesmacro, linearguments);
getSelectionBounds(lineboxx, lineboxy, lineboxw, lineboxh);
run("Select None");
runMacro(ROImacro);
Dialog.create("Check ROI and control lines position");
Dialog.addMessage("ROI and control lines OK?\nPress OK to continue running the
macro.\nPress Cancel to terminate (change the ROI and control lines macros and
restart).");
Dialog.show();
selectImage("TestROIandLines");
getSelectionBounds(x, y, r1, r2);
print("**ROI Bounds set:  x coordinates =", x, ", y coordinates =", y, ", width =",
r1, ", height =", r2);
print("**ROI and lines OK'd by user!");
print("Continuing analysis!");
print("___________________________________");

if (doprep==1) {
print("Check the set conservative threshold / fracture greyscale and bright spot
removal: Awaiting input...");
rename("TestConsThresh");
run("Max...", "value=maxmat");
wait(100);                      // Without the wait statement, the threshold does
not work correctly
setThreshold(0,fracthresh);
Dialog.create("Check conservative threshold / fracture greyscale and bright spot
removal");
Dialog.addMessage("Conservative threshold / fracture greyscale OK (in ROI)?\nBright
spots correctly eliminated?\nPress OK to continue running the macro.\nPress Cancel
to terminate (change threshold setting in macro).");
Dialog.show();

print("**Conservative threshold / fracture greyscale of", fracthresh, "OK'd by
user!");
print("**Removal of greyscales above", maxmat, "(bright spots removal) OK'd by
user!");
print("Continuing analysis!");
print("___________________________________");
}                                                            // End of IF for
checking greyscales.
close();                                                     // Required to close
open image
//__________________________________________________________________________________
___________________________________________________
//Generating input files for Hessian analysis: applying the bright spot removal,
the ROI, and the addition of control lines

if (doprep==0)
       {print("Input files for Hessian analysis already created earlier.
Continuing.");
} else {
print("Generating input files for Hessian analysis: applying the bright spot
removal, the ROI, and the addition of control lines...");

setBatchMode(true);
for (i=0; i<numfiles; i++) {                           //File counting starts at 0
       showStatus("Generating input files...");
       showProgress(i/numfiles);
       inputfile = inputfolder+inputfiles[i];
       open(inputfile);
       run("Max...", "value=maxmat");
       run("32-bit");
       runMacro(ROImacro);
       run("Make Inverse");
       run("Set...", "value=NaN");
       runMacro(linesmacro, linearguments);
       run("Select None");
       inputhessfile=inputhessfolder+inputfiles[i];
       saveAs("tiff", inputhessfile);
       close();
}
setBatchMode(false);
print("Generating input files for Hessian analysis finished!");
```

```
}
print("___________________________________");
//__________________________________________________________________________________
____________________________________________________

overlap=2*maxg;

setBatchMode(true);
if (padding>0) {
	print("Padding files by",padding,"% overlap");
	padoverlap=floor((padding/100)*overlap);
	print("**Selected percentage corresponds to",padoverlap,"x copying of the 
first and last slices");
	inputpadfiles=getFileList(inputhessfolder);
	numpadfiles=inputpadfiles.length;

	testfile=inputhessfolder+inputpadfiles[numpadfiles-1];			// 
Last file is numpadfiles-1
	open(testfile);
	padfilename=inputpadfiles[numpadfiles-1];

	for (i=1; i<=padoverlap; i++) {
		padfilename="Z"+padfilename;
		padfile=inputhessfolder+padfilename;

		save(padfile);
	}
	close();
	testfile=inputhessfolder+inputpadfiles[0];
	open(testfile);
	padfilename=inputpadfiles[0];

	for (i=1; i<=padoverlap; i++) {
		padfilename="0"+padfilename;
		padfile=inputhessfolder+padfilename;

		save(padfile);
	}
	close();
	print("Padding finished!");
	setResult("Padding (no. of slices of overlap)",0,padoverlap);
	updateResults();
	selectWindow("Results");
	save(root+"Padding.txt");
	run("Close");
} else if (padding==0 && paddone==1) {
	print("Padding turned off here but done in a previous stage.");
} else {
	print("Padding turned off");
}
setBatchMode(false);
print("___________________________________");
//End of padding

//__________________________________________________________________________________
____________________________________________________
//Hessian analysis

//overlap=2*maxg;							//Removed here 
since already present higher up
inputhessfiles=getFileList(inputhessfolder);
numhessfiles=inputhessfiles.length;
if (dohess==0)
	{print("Hessian analysis and normalising turned off or already performed 
earlier. Continuing.");
} else {
print("Applying Hessian analysis...");
blockstep=blocksize-2*overlap;
```

```
delstart=overlap;                 //Slice number "delstart" is deleted itself as
well
delend=blocksize-overlap+1;             //Slice number "delend" is deleted itself
as well
laststart=numhessfiles-blocksize+1;     //Defines last block to be analysed. May
overlap severely with previous.

countblocktot=-floor(-(numhessfiles-(2*overlap))/blockstep);       //Total number of
analysed blocks [ -floor(-A)=ceil(A) ]
countblock=1;                                         //Start of counting.

setBatchMode(true);
for (i=1; i<laststart; i+=blockstep) {  //Image sequence counting starts at 1.
       namestart=i+delstart;
       nameend=namestart+blockstep-1;
       print("**Analysing block", countblock, "of", countblocktot, ". Non-
overlapping part of slices:", namestart, "-", nameend);
       run("Image Sequence...", "open=&inputhessfolder number=&blocksize starting=&i
increment=1 scale=100 file=tif or=[] sort");
       rename("Input");
       for (j=ming; j<=maxg; j+=stepg) {
              print("****Analysing Gaussian Kernel:", j);
              run("FeatureJ Hessian", "largest middle smallest smoothing=&j");
              selectImage("Input largest Hessian eigenvalues");
              run("Slice Remover", "first=&delend last=&blocksize increment=1");
              run("Slice Remover", "first=1 last=&delstart increment=1");
              selectImage("Input middle Hessian eigenvalues");
              run("Slice Remover", "first=&delend last=&blocksize increment=1");
              run("Slice Remover", "first=1 last=&delstart increment=1");
              run("Abs", "stack");
              selectImage("Input smallest Hessian eigenvalues");
              run("Slice Remover", "first=&delend last=&blocksize increment=1");
              run("Slice Remover", "first=1 last=&delstart increment=1");
              run("Abs", "stack");
              imageCalculator("Subtract stack", "Input largest Hessian
eigenvalues","Input middle Hessian eigenvalues");
              imageCalculator("Subtract stack", "Input largest Hessian
eigenvalues","Input smallest Hessian eigenvalues");
              selectImage("Input middle Hessian eigenvalues");
              close();
              selectImage("Input smallest Hessian eigenvalues");
              close();
              selectImage("Input largest Hessian eigenvalues");
              run("Min...", "value=0 stack");
              nameprefix=d2s(j,1)+"_";
              run("Image Sequence... ", "format=TIFF name=&nameprefix
start=&namestart digits=5 save=&calcfolder");
              close();
       }
       selectImage("Input");
       close();
       countblock=countblock+1;
}
setBatchMode(false);

//Last block:
setBatchMode(true);
       i=laststart;
       namestart=i+delstart;
       nameend=namestart+blockstep-1;
       print("**Analysing block", countblock, "of", countblocktot, "(LAST block).
Non-overlapping part of slices:", namestart, "-", nameend);
       run("Image Sequence...", "open=&inputhessfolder number=&blocksize starting=&i
increment=1 scale=100 file=tif or=[] sort");
       rename("Input");
       for (j=ming; j<=maxg; j+=stepg) {
              print("****Analysing Gaussian Kernel:", j);
              run("FeatureJ Hessian", "largest middle smallest smoothing=&j");
```

```
             selectImage("Input largest Hessian eigenvalues");
             run("Slice Remover", "first=&delend last=&blocksize increment=1");
             run("Slice Remover", "first=1 last=&delstart increment=1");
             selectImage("Input middle Hessian eigenvalues");
             run("Slice Remover", "first=&delend last=&blocksize increment=1");
             run("Slice Remover", "first=1 last=&delstart increment=1");
             run("Abs", "stack");
             selectImage("Input smallest Hessian eigenvalues");
             run("Slice Remover", "first=&delend last=&blocksize increment=1");
             run("Slice Remover", "first=1 last=&delstart increment=1");
             run("Abs", "stack");
             imageCalculator("Subtract stack", "Input largest Hessian
eigenvalues","Input middle Hessian eigenvalues");
             imageCalculator("Subtract stack", "Input largest Hessian
eigenvalues","Input smallest Hessian eigenvalues");
             selectImage("Input middle Hessian eigenvalues");
             close();
             selectImage("Input smallest Hessian eigenvalues");
             close();
             selectImage("Input largest Hessian eigenvalues");
             run("Min...", "value=0 stack");
             nameprefix=d2s(j,1)+"_";
             run("Image Sequence... ", "format=TIFF name=&nameprefix
start=&namestart digits=5 save=&calcfolder");
             close();
       }
       selectImage("Input");
       close();
setBatchMode(false);

print("Hessian analysis finished!");
print("___________________________________");
//_________________________________________________________________________________
___________________________________________________
//Normalising the slices using the control lines
print("Normalising the slices using the control lines...");
normfiles = getFileList(calcfolder);
setBatchMode(true);
for (i=0; i<normfiles.length; i++) {                 //File counting starts at 0
       showStatus("Normalising...");
       showProgress(i/normfiles);
       normfile = calcfolder+normfiles[i];
       open(normfile);
       makeRectangle(lineboxx, lineboxy, lineboxw, lineboxh);
       getStatistics(a,b,c,max);
       run("Select None");
       run("Divide...", "value=&max");               //Normalises stack by dividing
by maximum value of control lines
       run("Max...", "value=1");              //Sets maximum saturation to 1
       save(normfile);
       close();
}
setBatchMode(false);

print("Normalising finished!");
}
print("___________________________________");
//_________________________________________________________________________________
___________________________________________________
//Combining the selected range of normalised Hessian scales
if (docombi==0) {
       print("The results of the Hessian calculations on various Gaussian kernels
will not be combined. Continuing.");
} else {
print("Combining the selected range of normalised Hessian scales...");
print("**Gaussian kernel sizes combined:", useming, "to", usemaxg, ", with
stepsize", usestepg);
```

```
selnum=((usemaxg-useming)/usestepg)+1;                //"Sel" for "Select" NOTE
COUNTING STARTS AT 1
selstart=((useming-ming)/stepg)+1;
selstep=(usestepg/stepg);

hessfilestart=1+overlap;                       //First name of files to address.
Based on NAMES (namestart) series. Differs from original series. Explains taken
numbers.
hessfileend=numhessfiles-overlap;              //Last name of files to address

setBatchMode(true);                            //Loop to make names correspond to
earlier output.
for (i=hessfilestart; i<=hessfileend; i++) {
       showStatus("Combining Hessian outputs...");
       showProgress(i/hessfileend);
       if (i<10) {
       name="_0000"+i;
       }else if (i<100) {
       name="_000"+i;
       }else if (i<1000) {
       name="_00"+i;
       }else if (i<10000) {
       name="_0"+i;
       }else if (i<100000) {
       name="_"+i; }

       run("Image Sequence...", "open=&calcfolder number=&selnum starting=&selstart
increment=&selstep scale=100 file=&name or=[] sort");
       rename("Hessscales");
       if (selnum>1) {
       //IF required to also allow 1 Hessian scale to be renamed only
              run("Z Project...", "start=1 stop=100 projection=[Max Intensity]");
              //100 is fine, there will never this amount of different Gaussian
kernels at the same time.
              selectImage("Hessscales");
              close();
       } else {
              rename("MAX_Hessscales");
       //One scale only
       }
       //End of IF for one scale only.

       selectImage("MAX_Hessscales");
       j=i-1;                                         //Hessian files start counting
at 1. Filenames can be everything but internally start counting at 0
       outputfile=outputfolder+"Hessian_"+inputhessfiles[j];
       if (i==hessfilestart) {
              print("**First combined: Hessian analysis slice", i, "to file",
outputfile, "(Overlap is", overlap, ", Check if correct)");
       }
       saveAs("Tiff", outputfile);
       close();
}
setBatchMode(false);

print("Combining finished!");
}
print("___________________________________");
print("Selected analysis finished!");
print("___________________________________");

//_________________________________________________________________________________
___________________________________________________
//Displaying final time, duration, and memory usage (note: bytes / 1024^2 to get
MBs). Saving logfile.
getDateAndTime(year2,month2,dayofweek2,dayofmonth2,hour2,minute2,second2,msec2);
month2=month2+1;
if (month2<10) {month2b="0"+month2; } else {month2b=month2;}
if (dayofmonth2<10) {dayofmonth2b="0"+dayofmonth2; } else
{dayofmonth2b=dayofmonth2; }
if (hour2<10) {hour2b="0"+hour2; } else {hour2b=hour2; }
if (minute2<10) {minute2b="0"+minute2; } else {minute2b=minute2; }
if (second2<10) {second2b="0"+second2; } else {second2b=second2; }
timeend="Macro finished: "+year2+"-"+month2b+"-"+dayofmonth2b+"
"+hour2b+":"+minute2b+":"+second2b;
print(timeend);
if (dayofmonth2<dayofmonth1) {
       hours=((dayofmonth1+dayofmonth2*24)+hour2)-(dayofmonth1*24+hour1);
} else {hours=(dayofmonth2*24+hour2)-(dayofmonth1*24+hour1);}
minutes=minute2-minute1;
if (minutes<0) {
hours=hours-1;
minutes=minutes+60;
}
timetaken="Duration of analysis: "+hours+" Hours and "+minutes+" Minutes";
print(timetaken);
usedmemory=(IJ.currentMemory())/1048576;
maxmemory=(IJ.maxMemory())/1048576;
percmemory=(usedmemory/maxmemory)*100;
percmemorydisp="("+percmemory+" %)";
print("End memory usage:",usedmemory,"MB of",maxmemory,"MB",percmemorydisp, "NOTE:
Differs from Windows Task Manager memory usage!");

savelogname="HessianFilteringLog";
savelogfile=root+"\\"+savelogname+".txt";
i=1;
while (File.exists(savelogfile)==true) {                          //Prevents
overwriting of log-files
       savelognamealt=savelogname+i;
       savelogfile=root+"\\"+savelognamealt+".txt";
       i=i+1;
}
print("Logfile saved to", savelogfile);
selectWindow("Log");
run("Text...", "save=savelogfile");
```

X2.3. MSHFF Calibration (FIJI Macro)

```
//MSHFF_Cal.ijm

//Multiscale Hessian fracture filtering: Cosine calibration
//Maarten Voorn, December 2012

requires("1.47b");
usedialog=1;                    //1 to use the GUI dialog, 0 to use parameters
defined below!!

if (usedialog==0) {
root='D:\\Rootfolder\\';
mincal=0;
maxcal=1;
usemaxg=5;                //Maximum used Gaussian for analysis previously!! Used to
calculate reduced ROI size.
dothresh=1;               //1 = Apply conservative threshold. 0 = Don't.
fracthresh=0;
teston=0;                 //Amount of slices to test the calibration on. Fill in 0
for all slices.
starton=0;                //Offset in dataset for a testing series. If teston is 0,
this value is ignored.
} else {

Dialog.create("Cosine Calibration - Settings");
Dialog.addMessage("Set the following parameters.\nThe starred (*) numbers should
correspond to the Hessian analysis performed earlier (especially [uses-max])!");
Dialog.addString("Path to root folder. Use \\ for subfolders and at end!",
"D:\\Rootfolder\\", 50);
Dialog.addMessage("Set the following parameters for calibration");
Dialog.addNumber("mincal = ", 0, 5, 7, "Lower asymptote");
Dialog.addNumber("maxcal = ", 1, 5, 7, "Upper asymptote");
Dialog.addMessage("");
Dialog.addNumber("Used combined maximum Gaussian [uses-max]", 6, 0, 4, "voxels *.
As used in combining the files!");
Dialog.addCheckbox("Add a convervative threshold?", true);
Dialog.addNumber("Conservative threshold greyscale value", 0, 0, 5, "(Best: As
chosen previously [consthresh] *). Uncheck and ignore number when no conservative
threshold is needed.");
Dialog.addMessage(" ");
Dialog.addNumber("Number of slices to test the combination on", 0, 0, 4, "Set 0 for
ALL slices (full dataset)");
Dialog.addNumber("Starting file number of test range", 0, 0, 4, "Ignored when
number of slices set to 0/All");
Dialog.show();

root=Dialog.getString();
mincal=Dialog.getNumber();
maxcal=Dialog.getNumber();
usemaxg=Dialog.getNumber();
dothresh=Dialog.getCheckbox();
fracthresh=Dialog.getNumber();
teston=Dialog.getNumber();
starton=Dialog.getNumber();
}


IJ.log("\\Close");
if (endsWith(root, "\\")==0) {exit("Root folder filename is not correct! Does it
end with \\ ? Macro aborted")}
if (dothresh==true) {
       threshinputfolder=root+"InputHess\\";
}
outputfolder=root+"Output\\";
coscalfolder=root+"CosineCal\\";
checkempty=getFileList(coscalfolder);
if (checkempty.length>0) {
	Dialog.create("Folder not empty");
	Dialog.addMessage("The CosineCal-folder is not empty. Press OK to delete 
files. Press Cancel to abort macro.");
	Dialog.show();
	for (i=0; i<checkempty.length; i++) 
{ok=File.delete(coscalfolder+checkempty[i]);}
}
File.makeDirectory(coscalfolder);
ROImacro=root+"ROI.ijm";
if (File.exists(ROImacro)==false) {exit("No ROI-file (ROI.ijm) found. Macro 
aborted.");}

print("Calibration by cosine");
//_______________________________________________________
________________________
getDateAndTime(year1,month1,dayofweek1,dayofmonth1,hour1,minute1,second1,msec1);
month1=month1+1;
if (month1<10) {month1b="0"+month1; } else {month1b=month1;}
if (dayofmonth1<10) {dayofmonth1b="0"+dayofmonth1; } else 
{dayofmonth1b=dayofmonth1; }
if (hour1<10) {hour1b="0"+hour1; } else {hour1b=hour1; }
if (minute1<10) {minute1b="0"+minute1; } else {minute1b=minute1; }
if (second1<10) {second1b="0"+second1; } else {second1b=second1; }
timestart="Macro started: "+year1+"-"+month1b+"-"+dayofmonth1b+" 
"+hour1b+":"+minute1b+":"+second1b;
print(timestart);
//_______________________________________________________
________________________

outputfiles=getFileList(outputfolder);
if (teston==0) {
	print("Calibrating full dataset.");
	numfilesout=outputfiles.length;
	starton=0;
} else {
	print("Calibrating chosen test-dataset only (", starton, "files, starting 
from file number", teston, ")");
	numfilesout=starton+teston;
}

pi=d2s(PI,9);
setdivide=(maxcal-mincal)/pi;
mincalname=d2s(mincal,8);
maxcalname=d2s(maxcal,8);

print("_______________________");
print(">Used settings<");
print("Minimum="+mincalname);
print("Maximum="+maxcalname);
print("Largest combined Gaussian kernel:", usemaxg, "- used to remove edge effects 
of Hessian analysis.");
print("_______________________");
if (dothresh==true) {
	print("A conservative threshold with greyvalue", fracthresh, "will be added 
to the calibrated output");
} else print("No conservative threshold will be added to the calibrated output");

setBatchMode(true);
testfile=outputfolder+outputfiles[0];
open(testfile);
runMacro(ROImacro);                     // Opens ROI, in first run only
getSelectionBounds(x, y, r1, r2);       // Gets bounds of ROI
ROItype=selectionType();                // Allows rectangular (0) or oval (1) 
selection in a later stage
close();
smallx=x+4*usemaxg;                     // Shifts x
smally=y+4*usemaxg;                     // Shifts y
smallr1=r1-8*usemaxg;                       // Shrinks radius by overlap
```

```
smallr2=r2-8*usemaxg;                                 // Shrinks radius by overlap
print("Smaller ROI Bounds set (edge effect removal):  x coordinates =", smallx, ",
y coordinates =", smally, ", width =", smallr1, ", height =", smallr2);
setBatchMode(false);
print("___________________________________");

//Main loop
print("Applying functions... ");
setBatchMode(true);
for (i=starton; i<numfilesout; i++) {
       showStatus("Applying functions...");
       showProgress((i-starton)/(numfilesout-starton));
       outputfile = outputfolder+outputfiles[i];
       open(outputfile);
       rename("Calibration");

       run("Max...", "value=&maxcal");
       run("Subtract...", "value=&mincal");
       run("Min...", "value=0");
       run("Divide...", "value=&setdivide");
       run("Add...", "value=&pi");
       run("Macro...", "code=v=0.5*(1+cos(v))");

       setMinAndMax(0, 2.55); //Scales from 0 to 100
       run("8-bit");
       run("Max...", "value=100");
       if (dothresh==true) {
              threshinputfilename=replace(outputfiles[i],"Hessian_","");
              threshinputfile=threshinputfolder+threshinputfilename;
              if (i==starton) {
                     print("Check if correct: file", threshinputfile, "used for
applying conservative threshold; combined with calibration in", outputfile);
              }
              open(threshinputfile);
              rename("ConsThresh");
              setThreshold(0,fracthresh);
              run("Convert to Mask", "  black");
              run("Divide...", "value=2.55");
              imageCalculator("Max", "Calibration", "ConsThresh");
              selectImage("ConsThresh");
              close();
       }
       selectImage("Calibration");
       if (ROItype==0) {
              makeRectangle(smallx, smally, smallr1, smallr2);
       } else if (ROItype==1) {
              makeOval(smallx, smally, smallr1, smallr2);
       }
       run("Make Inverse");
       run("Set...", "value=0");
       coscalfile = coscalfolder+outputfiles[i];
       save(coscalfile);
       close();
}
setBatchMode(false);

print("Calibration finished!");
print("___________________________________");
//__________________________________________________________________________________
___________________________________________________
getDateAndTime(year2,month2,dayofweek2,dayofmonth2,hour2,minute2,second2,msec2);
month2=month2+1;
if (month2<10) {month2b="0"+month2; } else {month2b=month2;}
if (dayofmonth2<10) {dayofmonth2b="0"+dayofmonth2; } else
{dayofmonth2b=dayofmonth2; }
if (hour2<10) {hour2b="0"+hour2; } else {hour2b=hour2; }
if (minute2<10) {minute2b="0"+minute2; } else {minute2b=minute2; }
if (second2<10) {second2b="0"+second2; } else {second2b=second2; }
```

```
timeend="Macro finished: "+year2+"-"+month2b+"-"+dayofmonth2b+"
"+hour2b+":"+minute2b+":"+second2b;
print(timeend);
if (dayofmonth2<dayofmonth1) {
       hours=((dayofmonth1+dayofmonth2*24)+hour2)-(dayofmonth1*24+hour1);
} else {hours=(dayofmonth2*24+hour2)-(dayofmonth1*24+hour1);}
minutes=minute2-minute1;
if (minutes<0) {
hours=hours-1;
minutes=minutes+60;
}
timetaken="Duration of analysis: "+hours+" Hours and "+minutes+" Minutes";
print(timetaken);
usedmemory=(IJ.currentMemory())/1048576;
maxmemory=(IJ.maxMemory())/1048576;
percmemory=(usedmemory/maxmemory)*100;
percmemorydisp="("+percmemory+" %)";
print("End memory usage:",usedmemory,"MB of",maxmemory,"MB",percmemorydisp, "NOTE:
Differs from Windows Task Manager memory usage!");
//______________________________________________________________________________
_________________________________________________
savelogcalname="CosineCalLog";
savelogcalfile=root+"\\"+savelogcalname+".txt";
i=1;
while (File.exists(savelogcalfile)==true) {                       //Prevents
overwriting of log-files
       savelogcalnamealt=savelogcalname+i;
       savelogcalfile=root+"\\"+savelogcalnamealt+".txt";
       i=i+1;
}
print("Logfile saved to", savelogcalfile);
selectWindow("Log");
run("Text...", "save=savelogcalfile");

if (teston>0) {
       run("Image Sequence...", "open=&coscalfolder number=&teston starting=1
increment=1 scale=100 file=[] or=[] sort use");
       setMinAndMax(0, 100);
       showMessage("Calibration of the test sequence shown. Restart the macro to
test other settings, or to perform on the whole stack.\nUsed settings are recorded
in the log-file. End of macro.");
}
```

X2.4. Connectivity filtering (MATLAB® script)

```
%MSHFF_Conn.m

clear;
tic;

%Provide the following two parameters:
rootfolder='D:/Rootfolder/';
threshold = 1;


%----------------------------------
%----------------------------------
pathDATA = 'CosineCal/';
typeDATA = '*.tif';
pathOUTPUT = 'Connected/';
nrcol = 255; % 255 for 8 bit images
%----------------------------------
cd(rootfolder);
fileDATA = dir([pathDATA typeDATA]);
mkdir(pathOUTPUT);
nrfiles=numel(fileDATA);

disp(['Loading ', num2str(nrfiles), ' input files...']);

h = waitbar(0,'Loading input files...');
for i=1:nrfiles
    perc=(i/nrfiles)*100;
    waitbar(i/nrfiles,h,sprintf('Loading input files %2.0f%%',perc));
    I(:,:,i) = uint8(imread([pathDATA fileDATA(i).name]));
end
pause(1);
delete(h);

disp(['Saving temporary MAT file with original data...']);
save('tempI.mat', 'I', '-v7.3'); % Save temp MAT file with original data

disp(['Setting a threshold of: ', num2str(threshold), ' for connectivity
analysis...']);
J=I.*0;                         %Preallocation
h = waitbar(0,'Setting threshold...');
for i=1:nrfiles
    perc=(i/nrfiles)*100;
    waitbar(i/nrfiles,h,sprintf('Setting threshold %2.0f%%',perc));
    J(:,:,i) = im2bw(I(:,:,i), (threshold/nrcol));
end
pause(1);
delete(h);

clear I;

disp(['Labeling connected voxels...']);
CC = bwconncomp(J, 26);

% CC.PixelIdxList is a list of clusternumbers, where for every number, an
% array of positions where this cluster occurs is saved.

imvolume=CC.ImageSize;
maxind=length(CC.PixelIdxList);         % Number of indices (=number of clusters)
maxvoxel=imvolume(1)*imvolume(2)*imvolume(3);
firstslice=imvolume(1)*imvolume(2);     % Max. value of first slice
lastslice=maxvoxel-firstslice+1;        % Min. value of last slice

disp('Checking for matching indices in first and last slice...');

% Adapted from http://www.mathworks.com.au/matlabcentral/answers/43833 :
indexok=false(1,maxind);                % Allocates an array of zeroes
```

```
for j=1:maxind;
    if length(CC.PixelIdxList{j})<imvolume(3)
        indexok(j)=true;                % Sets index number to 1 if it's not 
throughconnected
    elseif any(CC.PixelIdxList{j}<=firstslice)==0 || 
any(CC.PixelIdxList{j}>=lastslice)==0;
        indexok(j)=true;                % Sets index number to 1 if it's not 
throughconnected
    end
end
CC.PixelIdxList(indexok)=[];            % Deletes non-connected parts!!!

pause(1);

disp('Extracting all connected image locations...');
A=vertcat(CC.PixelIdxList{1:end});      % Combines ALL elements (=pixel locations) 
left in CC.PixelIdxList into 1 array A.

clear CC;
pause(1);

%Copying original dataset
J=J.*0;

pause(1);
disp(['Loading original dataset...']);
load('TempI.mat');

pause(1);

disp(['Filtering original data by connected components...']);
A=sort(A);                              % Sort A (not sure if really necessary)
%h = waitbar(0,'Filtering data...');    % Waitbar just doesn't work here...
nrA=numel(A);
for j=1:nrA;
    J(A(j))=I(A(j));
end
pause(1);
%delete(h);

pause(1);
disp(['Writing output files...']);
h = waitbar(0,'Writing output files...');
for i=1:nrfiles
    [useless, filename, useless]=fileparts(fileDATA(i).name);
    clear useless;
    perc=(i/nrfiles)*100;
    waitbar(i/nrfiles,h,sprintf('Writing output files %2.0f%%',perc));
    imwrite(J(:,:,i),[pathOUTPUT, 'Conn_', filename, '.tif']);
end
pause(1);
delete(h);
pause(1);
disp(['Finished!']);
elapsedtime=toc;
disp(['Ellapsed time: ', num2str(elapsedtime)]);

delete('tempI.mat');
```

Curriculum Vitae

Maarten Voorn

Personal data

Date and place of birth: 13 May 1987, Sassenheim, The Netherlands. Nationality: Dutch

Education

2011 - 2014 PhD study, Department of Geodynamics and Sedimentology, University of Vienna, Austria
Industry-funded by OMV E&P (in close collaboration and with regular reporting)

Topic: *Fracture and porosity analysis of (drill) core samples using 3D µCT, FIB-SEM, 2D microscopy techniques and laboratory experiments*
Supervisors: Ulrike Exner & Bernhard Grasemann

2008 - 2010 MSc study, Geology, Utrecht University, The Netherlands
Grade Point Average 4.0 / Cum Laude (NL) degree

Thesis: *A new way to confirm meteorite impact produced planar features in quartz: Combining Universal Stage and Electron Backscatter Diffraction techniques*
Supervisors: Maartje Hamers & Martyn Drury

2005 - 2008 BSc study, Earth Sciences, Utrecht University, The Netherlands
Grade Point Average 4.0 / Cum Laude (NL) degree

Thesis: *Characteristics and emplacement of orogenic peridotites: A comparison between the Betic Cordillera (Ronda) and Pyrenean ultramafic bodies*
Supervisor: Hans de Bresser

1999 - 2005 Gymnasium (Dutch secondary education), Rijnlands Lyceum, Oegstgeest, The Netherlands

Work experience

2011 - 2014 PhD Study under a paid contract (see *education* and *PhD project description*)

Past Various side jobs (not geology related)

Relevant skills

Computer skills
- Image processing and analysis (using e.g. ImageJ, MATLAB, Amira, Photoshop, CorelDraw)
- Programming of image processing and analysis routines (mainly ImageJ and MATLAB)
- Visualisation of, processing of, and modelling on 3D datasets
- Various stereographic plotting and other geology-specific software
- General other skills in a wide range of other software (e.g. Office, Acrobat)

Imaging Techniques
- Knowledge and (supervised) use of X-ray Micro Computed Tomography (µCT)
- Knowledge and use of SEM
- Knowledge and (supervised) use of SEM-EBSD (MSc) and -CL, and FIB-SEM (PhD)
- Optical light microscopy throughout BSc, MSc and PhD studies
- Optical light microscope and U-stage experience, MSc Thesis.

Languages
- Dutch (native)
- English (fluent)
- German (fluent, CEFR Level C1.1 course completed)
- French (beginner to intermediate level, secondary school classes and holiday practice)

Publications

March 2013	Voorn, M., Exner, U. & Rath, A., 2013. *Multiscale Hessian fracture filtering for the enhancement and segmentation of narrow fractures in 3D image data*. Computers and Geosciences 57, pp44-53. DOI: 10.1016/j.cageo.2013.03.006
2014	Voorn, M., Exner, U., Barnhoorn, A., Baud, P., Reuschlé, T., 2014. *Porosity, permeability and 3D fracture network characterisation of dolomite reservoir rock samples.* Submitted to Journal of Petroleum Science and Engineering.

Presentations, conferences and courses

Oral presentations at international conferences

- European Geosciences Union (EGU) General Assembly 2011, Vienna, Austria (Abstract)
- European Geosciences Union (EGU) General Assembly 2012, Vienna, Austria (Abstract)
- Conference on Industrial Computed Tomography (iCT) 2012, Wels, Austria (Paper)
- European Geosciences Union (EGU) General Assembly 2013, Vienna, Austria (Abstract)
- AAPG Exploring and Producing Fractured Reservoirs in the Middle East 2013, Dead Sea, Jordan (+ Poster, Abstract)

Other oral presentations

- OMV “Employer of Choice” Day 2011, Vienna, Austria
- Contribution to seminar “Imagination and imaging in scientific practice” 2012, Vienna, Austria
- Midterm (2012) and final report (2013) for OMV E&P (PhD project Sponsor), Vienna, Austria
- Various other formal and informal presentations (during PhD, MSc and BSc)

Poster presentations

- Tectonomechanics Colloquium (TecMec) 2011, Salzburg, Austria (Abstract)
- Deformation mechanics, Rheology and Tectonics (DRT) meeting 2011, Oviedo, Spain (Abstract)
- 1st International Conference on Tomography of Materials and Structures (ICTMS) 2013, Ghent, Belgium (Extended abstract)
- Deformation mechanics, Rheology and Tectonics (DRT) meeting 2013, Leuven, Belgium (Abstract)

Other conference attendances

- International Association for Mathematical Geosciences (IAMG) Conference 2011, Salzburg, Austria
- International Conference on Flows and Mechanics in Natural Porous Media from Pore to Field Scale (Pore2Field) 2011, Rueil-Malmaison, France

Other courses (apart from BSc, MSc and PhD curricula)

- Image Analysis for Porous Media (IAPM) Short Course & Workshop 2011, Austin, USA (+ Poster, Abstract)
- Image Analysis in Geosciences workshop 2012, Basel, Switzerland
- “Der Rote Faden” in English Academic Writing and how to spin it, 2011 (part 1) & 2013 (part 2), Vienna, Austria