Mining, Metallurgy & Exploration (2022) 39:2011–2024
https://doi.org/10.1007/s42461-022-00662-5

A New Workflow of X-ray CT Image Processing and Data Analysis of Structural Features in Rock Using Open-Source Software

Neel Gupta[1] · Brijes Mishra[1] · Dustin M. Crandall[2]

Received: 18 July 2021 / Accepted: 26 July 2022 / Published online: 8 September 2022

Abstract
X-ray computed tomography (CT) images of rock specimens often contain artifacts which must be corrected before scientific analyses are performed. This paper presents a new workflow of automated image processing to utilize poor-quality X-ray CT scan images. The workflow runs on the open-source image analysis software and efficiently separates desired features from low-contrast scanned images. The new workflow is a two-step technique using contrast enhancement and automated feature segmentation to generate noise-free binary images. The results of binary images using the proposed workflow and using a conventional thresholding technique are analyzed to show the quality of the proposed method. The paper also presents a workflow of estimating the structural geometries of features in two and three dimensions. The results of the structural feature analyses and computational time were compared between the open-source (ImageJ) and commercial image analysis software (Bruker Computed Tomography Analyzer). The commercial software was more computationally efficient, but the task-specific macros in open-source software enabled the user-desired automation in image processing and data extraction of desired structural features of comparable quality.

Keywords Fracture geometry · X-ray CT scan · Image analysis · ImageJ · Bruker CTAn · GNU Octave

1 Introduction

In recent years, X-ray computed tomography (X-CT) proved useful in the three-dimensional imaging and characterization of internal rock properties, such as pore network, fracture geometry, and heterogeneous minerals. The quantitative characterization of fracture geometry is necessary, as fractures provide natural passage to fluid flow in low-permeability rock formations. The capability of X-CT equipment to scan large core specimens, non-destructively with minimum specimen preparation, is also valuable in the core flow experiments [1, 2]. Real-time X-CT scanning monitors the in situ physicochemical interaction between internal fractures and fluids like CO_2 or brine and can be used to determine hydrothermal alteration for rocks [3]. Research into geologic carbon storage, multi-phase flow through sub-surface rock formations, and petroleum exploitation which requires in-depth investigation of fracture network properties also benefit from X-CT scanning.

The industrial X-CT system usually consists of an X-ray source, specimen platform, radiation detector, and a computer system to analyze data [4]. During operation, X-ray tube emits the X-ray beam, and the detector captures the projection images (or radiographs) of specimen from different angles. Depending upon the configuration, either the specimen rotates on the platform or both the source and detector rotates around the specimen, radiographs are captured at desired rotational frequency. The computer system analyzes the radiographs and reconstruct the grayscale images, where different phases of rock composition are identified based on the density and atomic number [5]. In the grayscale image, low-density area such as voids or fractures appears dark, while high-density area such as minerals appears gray or brighter in color. The process of classifying grayscale image into multiple segments containing voxels of similar attributes, such as voids or minerals, is called image segmentation [6].

In this study, a NorthStar Imaging (NSI) M-5000 Industrial CT scanner, located at the National Energy Technology

✉ Brijes Mishra
brijes.mishra@mail.wvu.edu

[1] Mining Engineering, West Virginia University, Morgantown, WV 26505, USA

[2] US Department of Energy, Research & Innovation Center, National Energy Technology Laboratory, Morgantown, WV 26507, USA

Laboratory (NETL) facility in Morgantown, West Virginia, imaged the cylindrical shaped shale specimen. As shown in Fig. 1, a sandstone specimen with 4 inches in diameter and 12 inches in length is placed on the rotating platform between the X-ray source and detector, such that specimen rotates 360 degrees, and the detector collected each radiograph at 0.25-degree rotation of the specimen. Although the methods of collecting X-CT images vary with the manufacturer, the new workflow of image processing should work for any type of configuration.

Image segmentation can be subjective, and there is no one standard segmentation technique to provide suitable results for all imaging applications [7]. Global thresholding is the most common region-based thresholding technique, which works best for images with the objects of uniform intensities and distinctive gray value histograms. The desired feature in the image is extracted from background by comparing the individual pixel value with only one threshold value selected for the entire image [8, 9]. Maerz [10] and Reid and Harrison [11] applied thresholding algorithms to trace fractures in rocks and converted the grayscale images into binary images. Although global thresholding is a computationally fast and reliable technique, its application fails for the images with non-uniform background or low, fuzzy contrast between the object and the background. Researchers suggested an additional step of pre-processing before applying global thresholding technique, such as image smoothing, to segment the low-contrast grayscale images. The filters like mean or median smoothing sharpen the peaks of the unimodal histogram, make it more strongly bimodal, and allow selection of threshold for image segmentation [12, 13]. The mean filter is a sliding window spatial filter, which replaces the center value in the window with the average of all pixel values in the window [14]. However, in the median filter, the value of center pixel is replaced by the median value of all pixel values in the window. Here, the window is usually square of N-by-N size; $N=3, 5, 7$, etc. In particular, the median filter works well for the features like fractures in the image; unlike the mean filter, it does not blur the edges of fracture. Other segmentation techniques like region growing or edge-based segmentation also work well to segment low-contrast regions from the background. The technique of region growing compares the individual pixel value with a pixel or group of pixels (also called seeds) of object of interest and adds it to the growing region based on the uniformity test [15]. For example, for the comparison of pixel intensity with the mean intensity value of object of interest, if difference is less than a predefined value, the pixel is included in the region. The method works best for the spatially separated regions with similar properties. Gouze et al. [16] and Noiriel et al. [17] used region growing segmentation method to analyze the change in the 3D fracture volume and aperture in rocks during the core flow experiments. Wang and Hakami [18] suggested one-pass valley-edge detection technique for tracing thin fractures in less noisy rock images. In the case of thick fractures with high background noise, Wang and Hakami [18] suggested the application of multi-resolution technique during fracture tracing for improved results. Deng et al. [19] proposed an automated image segmentation technique, technique of iterative local thresholding to segment and quantify fracture geometry in low-contrast X-CT images. The program was based on MATLAB, which first improved the contrast among void and minerals, and Otsu's method of thresholding converted the grayscale image into binary images. Therefore, in the past, several filters and algorithms have been proposed to segment the fractures in rock matrix.

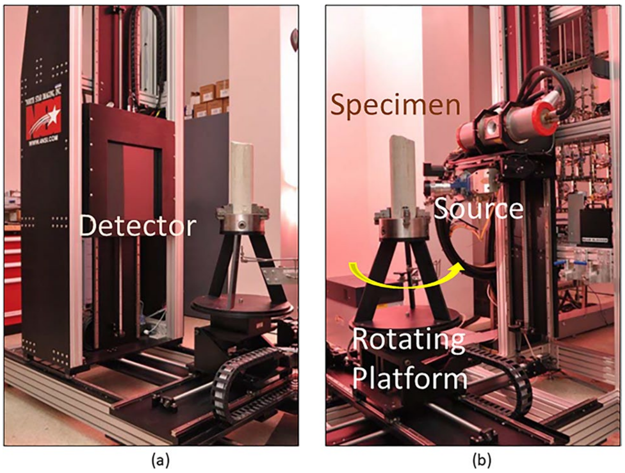


Fig. 1 NorthStar Imaging M-5000 Industrial CT scanner: **a** X-ray detector with vertical sandstone core; **b** X-ray source with vertical sandstone core [5]

In addition to the poor contrast among features of interest, X-ray CT image suffers with several artifacts that obscure the details of interest. The artifacts are caused by two primary reasons: problems with CT imaging setup, such as ring artifact, and sample dependent, such as beam-hardening artifact [20]. Ring artifacts appear as full or partial circles centered on the rotational axis, caused by shifts in output from individual detector or sets of detectors [21]. In the current experimental setup, the dead pixels unaccounted in detector corrections and calibrations caused the ring artifacts in CT image. The recent upgrade in the NSI software algorithm, named as ring reduction, compensates the irregular responses of the detector pixels during CT scan and efficiently removes ring artifacts without any additional time to a standard scan. However, the beam-hardening artifact is more problematic, as its effects can be reduced but not always eliminated. The non-uniform attenuation of polychromatic incident X-ray beam causes the brighter edges than the center of homogenous object, called as beam-hardening artifact [22]. The possible remedial methods range from sample and scanning preparation to data processing, for example, pre-harden, the X-ray beam by passing it through an attenuating filter, such as flat pieces of copper, brass, or aluminum, before or after it passes through scanned object [23]. The method is partially effective in removing beam-hardening artifact and introduce greater image noise. In current experimental setup, the beam-hardening artifact was corrected during CT image reconstruction. The NSI software filters out the artifacts due to lower attenuation of edges compared to the middle of the object.

In this study, a new image processing workflow is proposed to segment fractures from rock matrix in low-contrast, poor-quality X-CT images. The workflow is based on the open-source image analysis software, ImageJ [24]. A comparison of results of the new method with conventional, region-based thresholding techniques is presented. The quantitative results of fracture geometries (aperture and length) are also determined. The results of the geometric calculations of fractures were compared between ImageJ and a commercial image analysis software, Bruker Computed Tomography Analyzer (CTAn) in terms of accuracy and the time for computational analysis. Gupta [25] briefly explains the application of non-destructive X-ray CT imaging technique in determining the growth of microcracks due to constant stress for extended time and correlating it with the measured strains in the shale specimen.

2 X-ray Computed Tomography (X-CT) Image Processing

In the current study, a NorthStar Imaging M-5000 Industrial CT scanner imaged a cylindrical Marcellus shale specimen. The shale specimen was 2.1 inches in diameter and 4.2 inches in length. Specimen thickness, density, and X-ray attenuation influence the ability of X-rays to pass through a sample (i.e., the percentage of detector counts around and through the sample) [26]. Typically, the penetration values between 10 and 90% result good scan quality [26]. As the thickness of shale specimen was 2.1 inches, the operating condition of 185 kilovolts and 400 microampere current was found appropriate to enter the specimen. The X-CT scanner captured 1440 radiographs, which reconstructed the 3-D digital volume of the specimen. The voxel resolution of the X-CT images is 29.9 microns (μm). In this study, four possible methods of image segmentation in ImageJ are discussed. The nascent grayscale image from X-CT scanner is termed as *original image*, and the fractures are termed as *object of interest*, and the intact rock matrix is termed as *background*. In the current study the original X-CT image was converted from 16 to 8 bit. However, the workflow of image processing discussed in this paper is equally applicable to both types of X-CT images.

2.1 Case I: Global Thresholding Method on Original X-CT Image

In ImageJ, the plugin Auto Threshold performed the global thresholding of an 8 or 16-bit grayscale image. As shown in Fig. 2, global thresholding of original image generated a montage of binary images of four rows by four columns using 16 available algorithms. Unlike the ImageJ's thresholder applet accessed through ImageJ > Adjust > Threshold where different thresholding algorithms are manually selected, the Auto Threshold plugin automatically determined the single threshold value for all available algorithms. For example, Mean algorithm determined the mean of gray levels as the threshold value [27]. The results of global thresholding in Fig. 2b showed that only four algorithms, namely, Li, Mean, Otsu, and Percentile, showed few fractures in the binary image. The

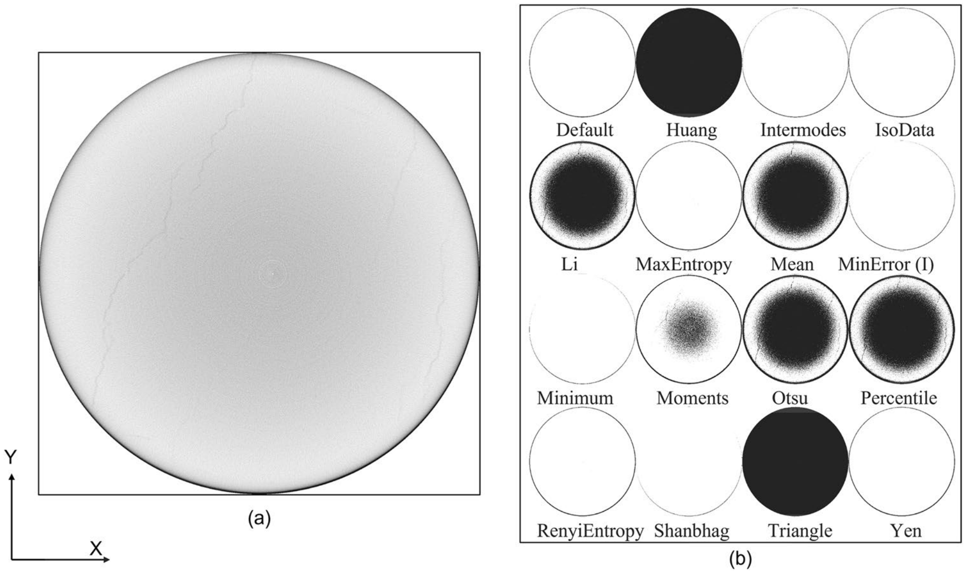


Fig. 2 Case I of image segmentation: **a** original X-CT image; **b** results of global thresholding through different algorithms

unusual darker region in the middle region of the original image in Fig. 2a indicated the presence of beam-hardening artifact in X-CT image, which caused the non-segmentation of fractures in the binary image.

2.2 Case II: Adaptive (Local) Thresholding on Original X-CT Image

The global thresholding method does not give good segmentation results in non-uniform background [7]. As the beam-hardening artifact caused the non-uniform background in original X-CT image, in case II, adaptive or local thresholding method was adopted to segment the fractures. The local thresholding technique partitions the entire image into subimages and determines a threshold for individual subimage [8]. In ImageJ, plugin Auto Local Threshold performed the adaptive thresholding of original image through nine different algorithms. The plugin computed the threshold for each pixel within the local domain, identified by the radius of the window. Figure 3 shows the result of Auto Local Threshold on the original X-CT image. Each of the seven algorithms excluding Phansalkar [28] and Sauvola [29] segmented the fractures and generated a binary image. Although the texture and morphology of fractures were visible, the binary images contained a lot of noise. In addition, the result of Contrast algorithm suffered with the beam-hardening artifact and the fractures in the middle region were still indistinguishable. Therefore, the corrections of beam-hardening artifact and low-contrast between the object of interest and the background were essential pre-processing steps for image segmentation.

2.3 Case III: Adaptive (Local) Thresholding on Pre-processed X-CT Image

In a homogenous specimen, the beam-hardening artifact causes brighter pixels at the edges and darker pixels in the middle of the image. The intensity of beam-hardening artifact is determined through the variation in the profile of gray-scale value (GSV) across the width of the image. Figure 4a shows the GSV of original X-CT image across the image width. The result showed that the profile of GSV

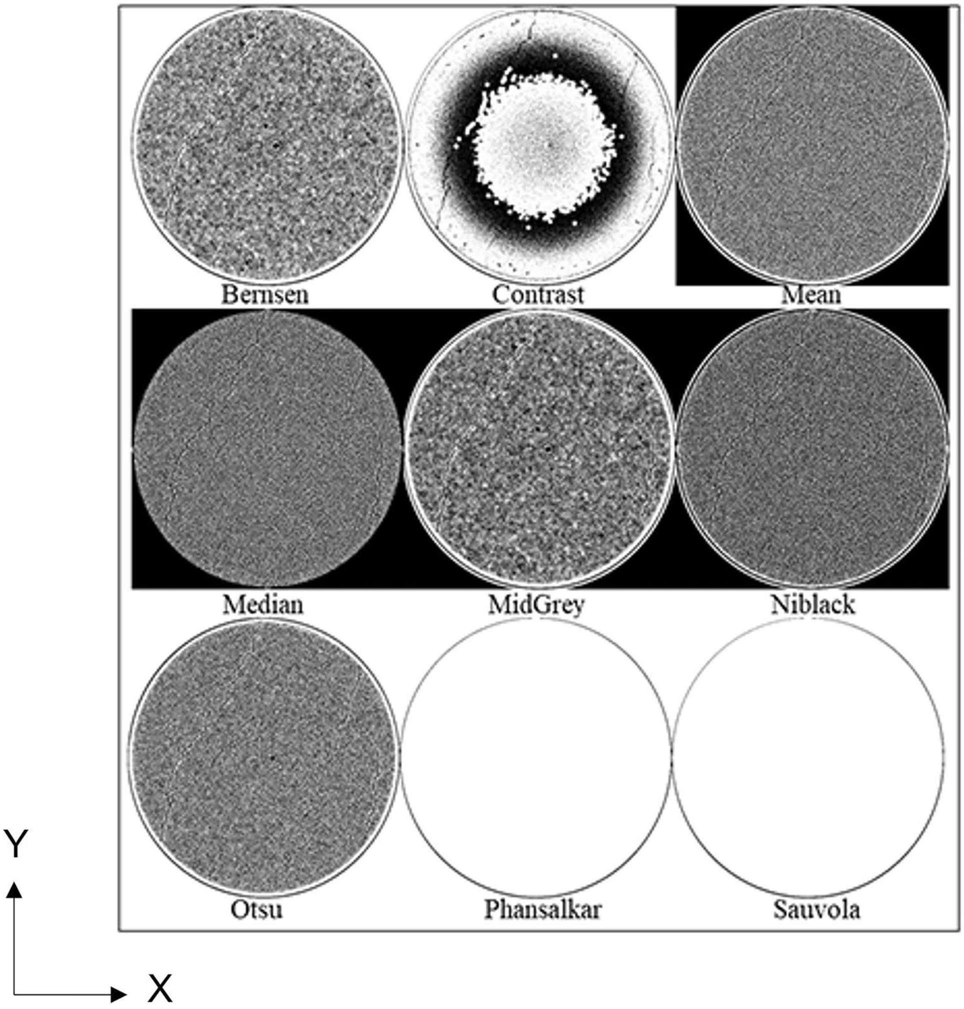


Fig. 3 Results of adaptive thresholding on original X-CT image through different algorithms

for original X-CT had a cupping artifact, initially decreasing and then increasing with minimum GSV at the middle pixel of the image [22]. Romano et al. [30] developed a plugin for the automated beam-hardening correction in X-CT images. The plugin runs on ImageJ software on both individual and stack of X-CT images. In the first step of pre-processing, the plugin corrected the artifact in original X-CT image and generated the beam-hardening corrected (B-Hc) X-CT image. In Fig. 4b, the profile of GSV for B-Hc X-CT image was approximately constant across the image width. The comparison showed that the plugin effectively removed the cupping artifact from the original X-CT image. In the next step, image processing with median filter improved the contrast between fractures and intact rock matrix of B-Hc X-CT image. Unlike Gaussian filter, the median filter did not blur the edges of fractures and improved their visualization.

In ImageJ, the adaptive thresholding algorithm segmented the pre-processed X-CT image. The results of image segmentation are shown in Fig. 5. Comparing Figs. 3 and 5, pre-processing of original X-CT image improved the image segmentation. However, Phansalkar and Sauvola algorithm did not show any improvement in the image. For example, the results of algorithm contrast on original X-CT image suffered beam-hardening effect; however, similar algorithm showed all fractures in B-Hc X-CT image. Although the quality of image segmentation improved through pre-processing, the output binary images from adaptive thresholding were not directly useable for fracture analysis. The binary images required post-processing for background noise removal. In addition, binary images from each of seven algorithms contained noise in the form of rings, which also indicated the presence of ring artifacts in B-Hc X-CT images [31].

2.4 Case IV: Trainable Weka Segmentation on Contrast Enhanced X-CT Image

Case IV involved two steps in the image segmentation. The first step was the pre-processing of original X-CT image through contrast enhancement for clear visualization of fractures. The second step was image segmentation through Trainable Weka Segmentation (TWS) algorithm. In the current case, the contrast limited adaptive histogram equalization (CLAHE) was used for contrast enhancement in original X-CT image. The plugin was pre-installed in ImageJ. CLAHE is an improved version of adaptive histogram equalization (AHE) method and enhances the contrast in the local regions of the image [32].

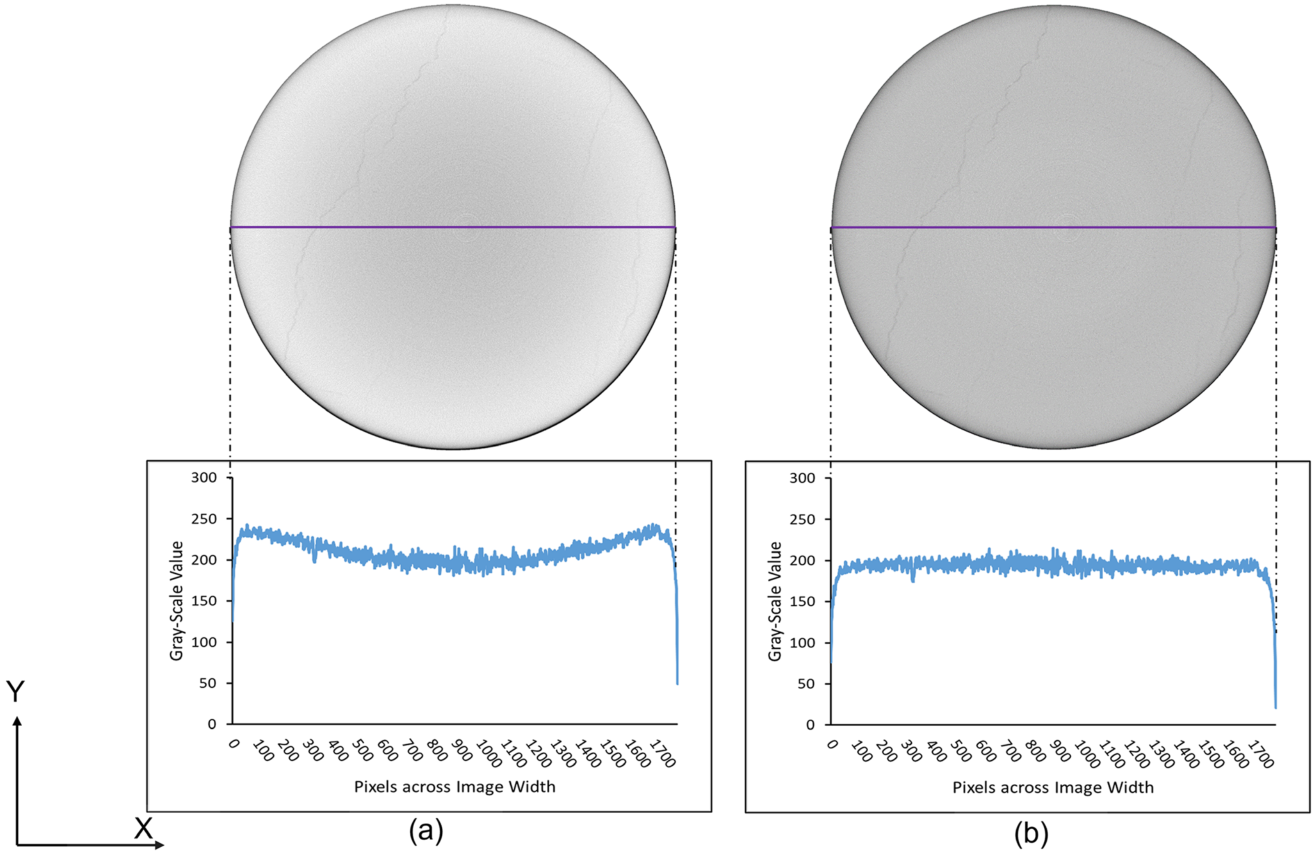


Fig. 4 Comparison of the GSV across the image width: **a** original X-CT image; **b** beam-hardening corrected X-CT image

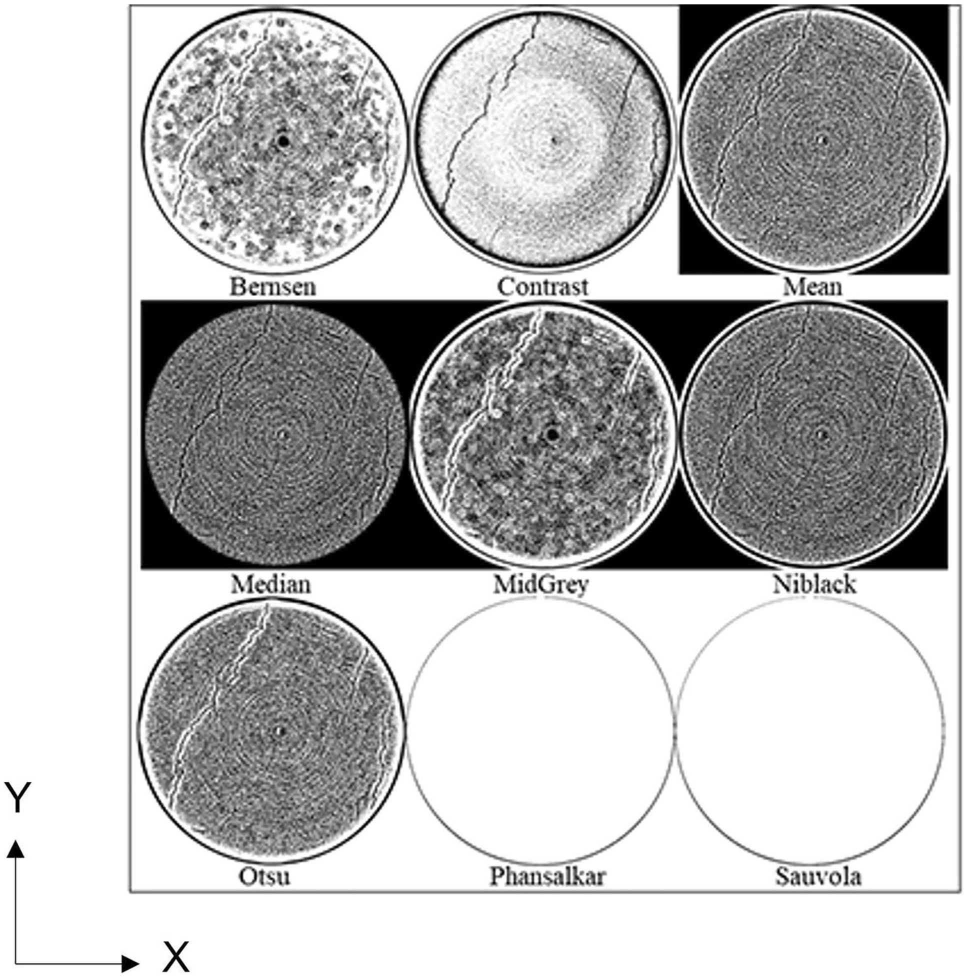


Fig. 5 Results of adaptive thresholding on B-Hc X-CT image through different algorithms

The improvement in the local contrast showed features, such as fractures and ring artifacts in the original X-CT image, as shown in Fig. 6. However, unlike other contrast enhancement methods such as linear unsharp masking or histogram equalization, CLAHE did not amplify the noise or beam-hardening artifact in the original image [33, 34].

Once the local contrast of original X-CT image was enhanced, the plugin TWS segmented the contrast enhanced X-CT image. TWS plugin is a combination of a machine learning algorithm and set of selected image features, which produces pixel-based segmentation [35]. First, the plugin trains an input image and generates an image classifier. The training of the input image meant differentiating and classifying representative voxels of different regions of image into the set of features, such as fractures and intact rock. Each set contained gray-level information of traced pixels. The biggest advantage of the method was that it allowed the user to finely discriminate different features into different classes by tracing a reasonable number of region of interests [36], which also improved the accuracy of image classifier. Once the image classifier was generated, unsupervised, completely automated image segmentation was performed on other input images of similar specimen. The result of binary image using TWS is shown in Fig. 6. Although the original X-CT image contained ring and beam-hardening artifacts, the image classifier efficiently segmented all fractures with least possible background noise. Therefore, the new workflow of image processing efficiently segmented fractures in the original X-CT images.

The comparison of binary-segmented images (with fractures in the foreground) showed that case IV was the most useful workflow method of image processing among four different methods. As ImageJ contained CLAHE and TWS plugins, we automated the process of image segmentation through a macro (mentioned in Appendix 1). The macro saved contrast enhanced and binary-segmented images at different locations with the name similar to the original X-CT image. The macro segmented the fractures in one image of size 1779 × 1770 pixels in 90 s (workstation was equipped with RAM of 128 GB and processor of 1.83 GHz). The value of the parameters in CLAHE plugin and image classifier in TWS was entirely dependent on the rock type and geometry and was based on iterations to get the best result. For the purpose of consistency, the current study used

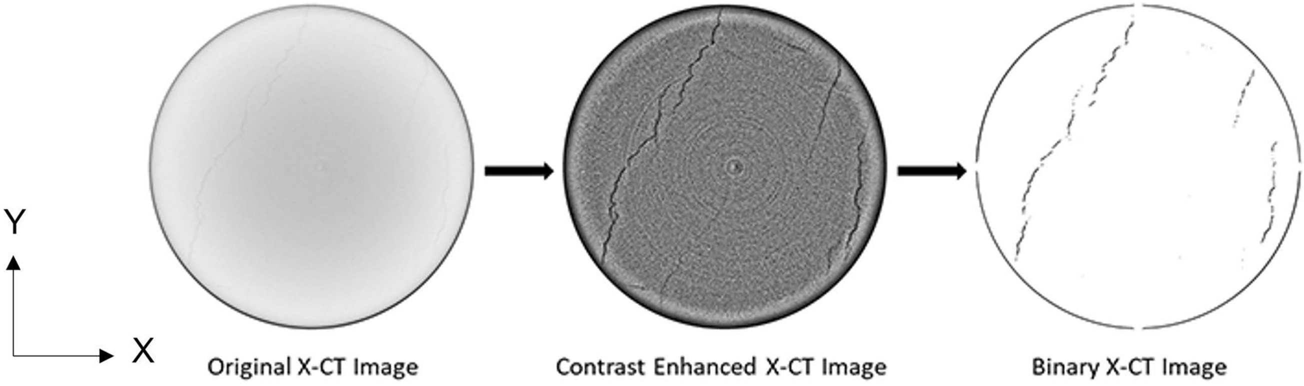


Fig. 6 Case IV: new workflow of image processing

similar values of parameters in both plugins for different image stacks of similar type of rock specimens.

3 Fracture Geometry from Binary X-CT Image

Based upon the length of the rock specimen and voxel resolution, the number of 2-D images in the 3D image stack was 3855. Figure 7a is an example of a montage of 2-D binary X-CT images spaced at 2.97 mm, representing the portion of the specimen. It showed the change in the geometry of fractures from the top left to the bottom right image. The sequential stack of 2-D binary X-CT images evenly spaced at 29.7 μm reconstructed the 3-D structure of the fractures in the shale specimen, as shown in Fig. 7b. As the X-CT image processing and reconstruction showed fracture distribution in both 2-D and 3-D coordinate system, the fracture geometry was determined separately in 2-D and 3-D coordinate system. The 2-D and 3-D geometry of fractures was separately determined from two image analysis software, i.e., ImageJ and Bruker CTAn. The comparison of fracture geometry and computational time established the time efficiency and accuracy of ImageJ against Bruker CTAn.

3.1 Two-Dimensional (2-D) Fracture Geometry in Shale

The two-dimensional fracture geometry included the area, aperture, and length. In ImageJ, the function *Analyze Particles* analyzed the binary images, counts, and measures objects and determined the area of randomly oriented fractures. Another plugin *BoneJ > Thickness* [37] in ImageJ, which operates on the principle of the Local Thickness plugin [38], determined

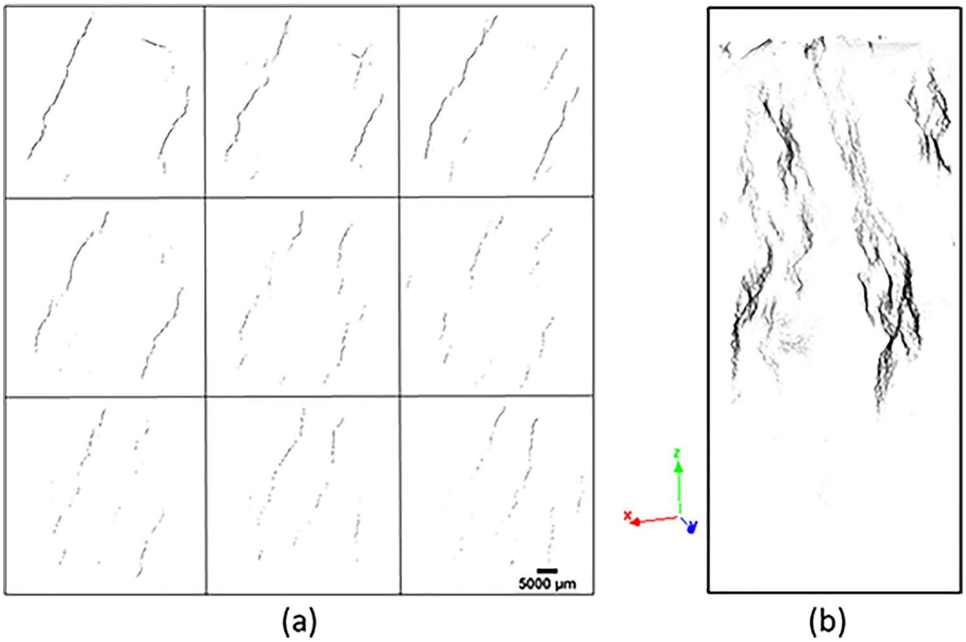


Fig. 7 **a** Montage of 2-D binary X-CT image spaced at 2.97 mm; **b** 3-D structure of fractures reconstructed from 2-D images spaced at 29.7 μm

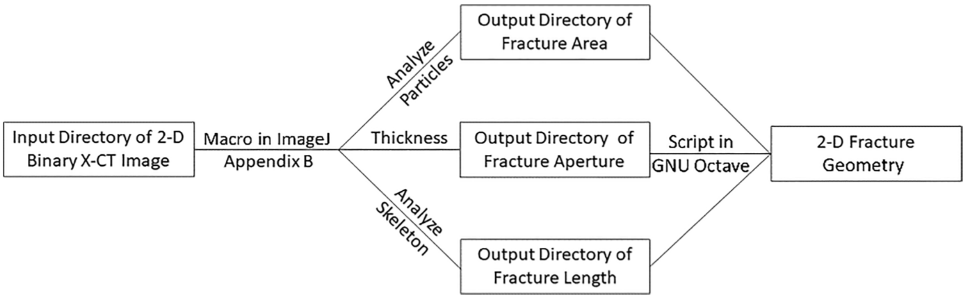

Input Directory of 2-D Binary X-CT Image
Macro in ImageJ Appendix B
Analyze Particles
Thickness
Analyze Skeleton
Output Directory of Fracture Area
Output Directory of Fracture Aperture
Output Directory of Fracture Length
Script in GNU Octave
2-D Fracture Geometry

Fig. 8 The workflow of data extraction and summarizing using scripts in ImageJ and GNU Octave

Table 1 Average value of 2-D geometry of fractures determined using ImageJ and Bruker CTAn

Geometrical parameter	ImageJ	Bruker CTAn	Difference (%)	Time (mins)	
				ImageJ	Bruker CTAn
Average cumulative area (mm^2)	4.27	4.36	−2.11	158.62	7.77
Average weighted mean aperture (mm)	0.10	0.08	20		
Average cumulative length (mm)	32.23	N/A	N/A		N/A

the weighted mean aperture (t_{wm}) of fractures. According to the principle of local thickness, thickness of a point is the diameter of the greatest sphere fits within the structure and contains the point. Further, plugin *Thickness* calculated the weighted mean and standard deviation of the local thickness of the binary object. In ImageJ, the plugin *BoneJ > Analyze Skeleton* calculated the length of fractures [39]. However, the operation included an extra step of skeletonizing binary X-CT image using plugin *Skeletonize 2D*. The plugin *Analyze Skeleton* analyzed the skeletonized binary X-CT images and

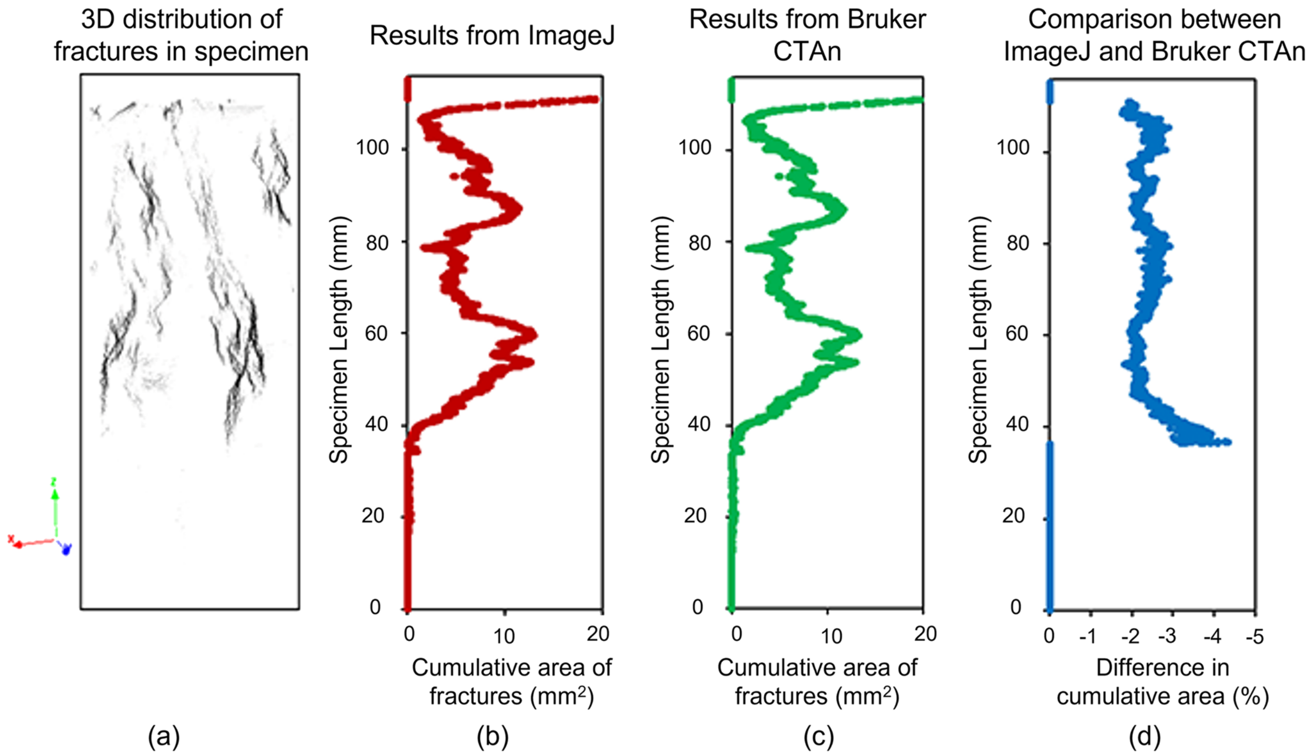


Fig. 9 **a** 3-D distribution of fractures reconstructed from 2-D binary X-CT images; **b** the graph between specimen length and cumulative area of fractures using ImageJ; **c** the graph between specimen length and cumulative area of fractures using Bruker CTAn; **d** the graph between specimen length and difference in cumulative area using ImageJ and Bruker CTAn

determined both Euclidean and tortuous length of fractures. Each of the three plugins generated a comma-delimited text file, which contained the area, weighted mean aperture and length of fractures in a binary X-CT image.

As the length of shale specimen was adequate to generate 3855 two-dimensional images, a macro in ImageJ determined the 2-D geometry of fractures. The macro (Appendix 2) read the binary X-CT image from input directory and saved the comma separated variable (*csv*) file of area, weighted mean aperture, and length of fracture in three different output directories. The macro created a *csv* data file numbered equivalent to three times of number of 2-D images, approximately 11,565 data files. Unlike the plugin *Thickness* that automatically calculated the weighted mean aperture of fractures in the image, the plugins, like *Analyze Skeleton* and *Analyze Particles*, calculated the length and area of individual fractures. Therefore, we required an extra step of estimating the sum of area and length of individual fractures in the image, called as cumulative area and cumulative length of fractures, respectively.

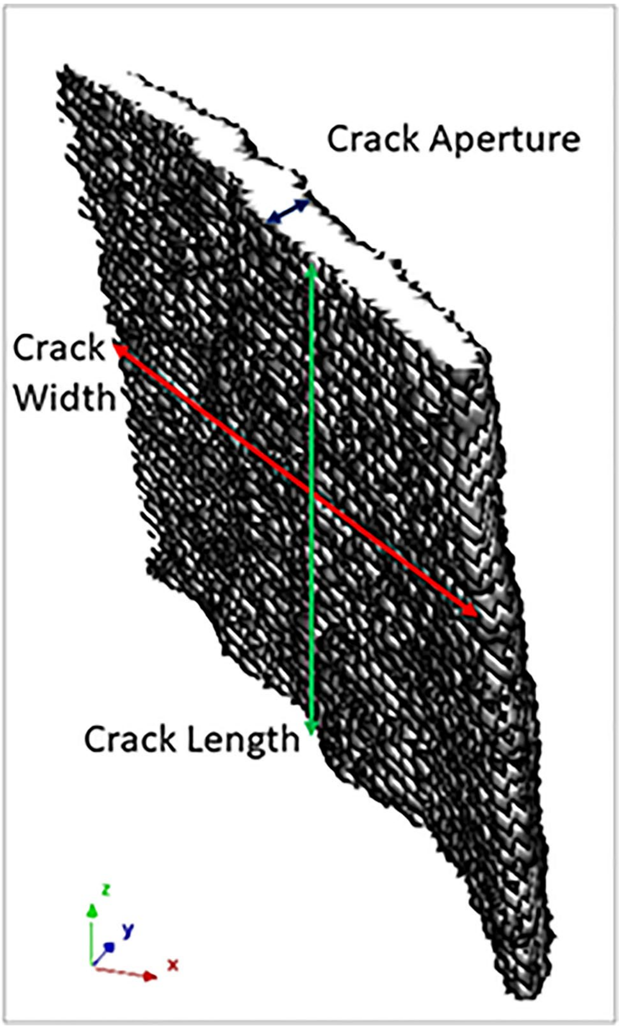


Fig. 10 Three-dimensional structure of fracture regenerated from 2-D binary X-CT images

Another script with.*m* extension, compatible to run in open-source software GNU Octave, was used to calculate the cumulative length (l_c) and area (a_c) of fractures in each image. The script also condensed the information of individual geometrical parameters from 11,565 files into a single file of 2-D fracture geometry. The file contained different columns, such as image name, image number, cumulative area, weighted mean aperture, and cumulative length of fractures. The complete workflow of data extraction using ImageJ and summarizing data using Octave is shown in Fig. 8 (Appendix 3). The use of macro in data extraction through different plugins in ImageJ and the script in GNU Octave to generate a single file of 2-D fracture geometry automated the process of data extraction using open-source software.

In Bruker CTAn, except for cumulative length of fractures, the plugin *2D Analysis* determined the cumulative area and weighted mean aperture of fractures. The output of the plugin contained many more parameters including area and aperture. Therefore, similar to the data extraction scheme of 2-D geometry of fractures in ImageJ, Bruker CTAn generated a single data file that contained columns, such as image name, image number, cumulative area, and weighted mean aperture of fractures. The advantage of Bruker CTAn over ImageJ was the automated calculation of the cumulative area of fractures in an image.

The 2-D geometry of fractures for a complete specimen was characterized through the average value of cumulative area, weighted mean aperture, and cumulative length (mentioned in Eq. 1). In Eq. 1, *n* represented the total number of images. The average values of different parameters from ImageJ and Bruker CTAn, the percentage difference between the values, and the computational time taken for analysis are summarized in Table 1.

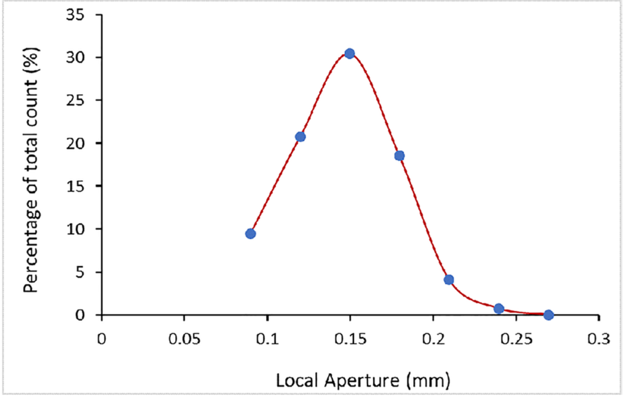


Fig. 11 Distribution of 3-D local aperture of fractures using ImageJ

Table 2 Results of 3-D geometry of fractures using ImageJ and Bruker CTAn

Geometrical parameter	ImageJ	Bruker CTAn	Difference (%)	Time (mins)	
				ImageJ	Bruker CTAn
3-D fracture volume (mm^3)	491.89	481.77	2.06	0.5	8.1
3-D aperture (mm)	0.123	0.125	− 1.63	37.8	
Surface area of fracture (mm^2)	13,656.13	13,727	−0.52	30	

$$\begin{aligned} Average\ a_c &= \frac{\sum_{i=1}^{n} a_{c_i}}{n} \\ Average\ t_{wm} &= \frac{\sum_{i=1}^{n} t_{wm_i}}{n} \\ Average\ l_c &= \frac{\sum_{i=1}^{n} l_{c_i}}{n} \end{aligned} \quad (1)$$

The scatter plot between the cumulative area of fractures and the length of specimen (Fig. 9) was advantageous in analyzing the distribution of fracture geometry in the whole specimen. The figure included the result of cumulative area of fractures from ImageJ and Bruker CTAn and the graph of the difference in the values along the length of specimen. Figure 9 shows that the cumulative area of fractures gradually varied along the specimen length, with comparatively higher values in the top half length. The reason for sudden decrease in the value of cumulative area of fractures at top edge was limitation of X-ray CT equipment in visualizing fractures at the specimen edge.

3.2 Three-Dimensional (3-D) Geometry of Fractures in Shale

In 3-D, the fracture regenerated from 2-D images appeared like a rectangular plate with finite length, width, and aperture in three mutually perpendicular directions, as shown in Fig. 10. The three-dimensional geometry of fractures included the volume, aperture, and surface area of the fracture plane. In ImageJ, the plugin *BoneJ > Volume Fraction* determined the 3-D volume of fractures. The plugin analyzed the sequential stack of 2-D binary X-CT images and calculated the volume of fractures through a voxel-based method. In ImageJ, the 3-D aperture of fractures was determined through the plugin *BoneJ > Thickness*. The plugin *Thickness* analyzed the complete stack of 2-D binary images, and similar to the 2-D aperture, it determined the local thickness of the fractures. Further, it calculated the overall 3-D aperture as the weighted mean value of local thickness. The distribution of local aperture of the fractures is shown in Fig. 11. The last 3-D geometrical parameter was surface area of fracture plane. In ImageJ, plugin *BoneJ > Isosurface* calculated the area of a 3-D object [40]. The algorithm of marching cubes in plugin *Isosurface* generated a triangular surface mesh around the binary object (fractures) and determined the surface area as sum of the areas of triangle making up the mesh. The results for 3-D geometrical parameters using ImageJ and time taken for the analysis are summarized in Table 2.

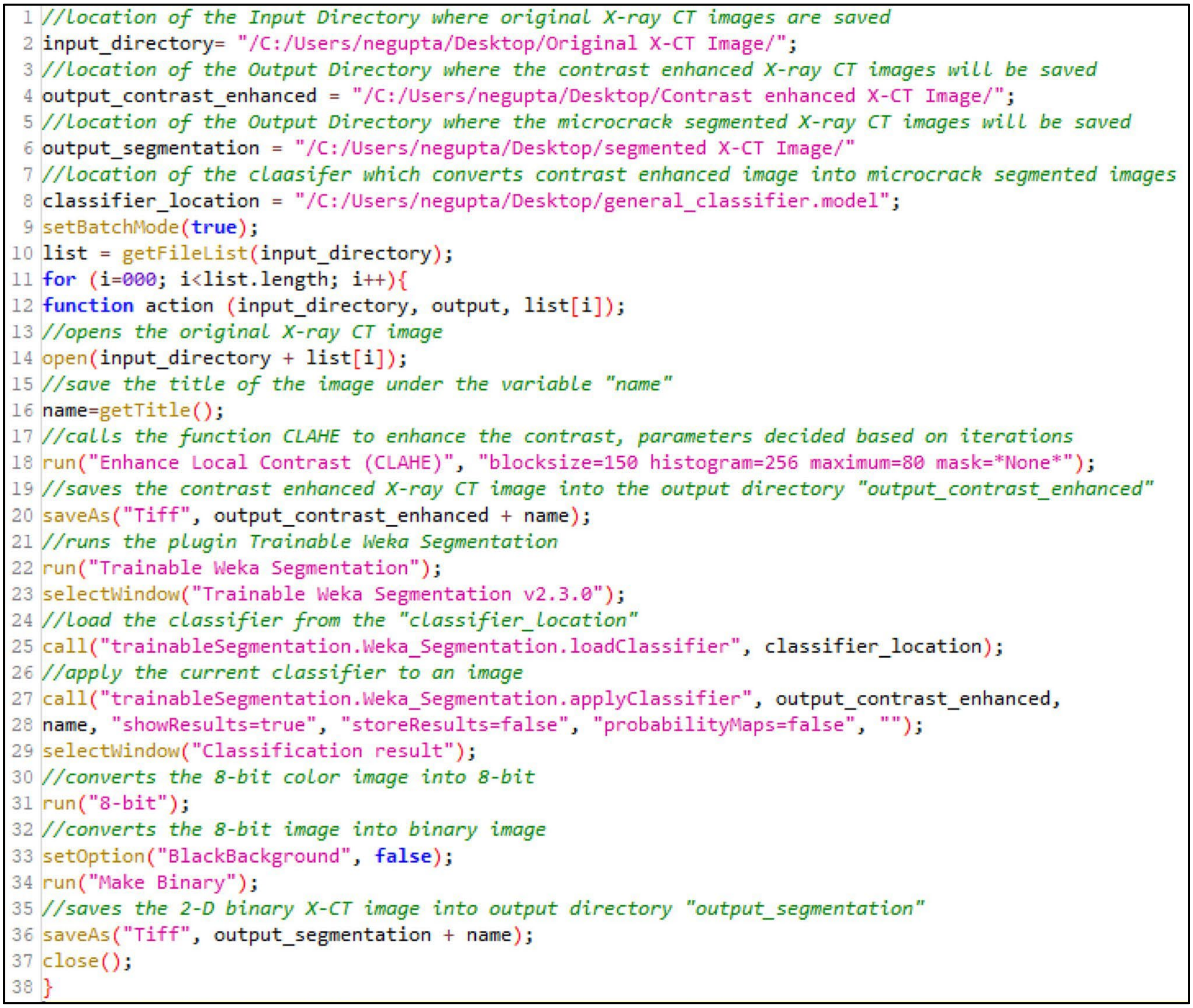
In Bruker CTAn, the plugin 3D Analysis determined the 3-D geometrical parameters of fractures. The analysis of plugin 3D Analysis on the stack of 2-D binary X-CT images determined the volume, weighted mean aperture, and surface area of fractures. The results of 3D geometry of fractures and time taken for analysis using Bruker CTAn are summarized in Table 2.

4 Discussion and Conclusions

In the current study, several methods of image processing in ImageJ were compared to segment fractures from low-contrast X-CT images, with beam-hardening and ring artifacts. Although the magnification of X-ray CT image wasn't sufficient to delineate the grain boundary or pores in the analyzed shale specimen, the low-density area such as fracture was distinguishable from minerals to the resolution of 29.9 µm. Higher magnification X-ray CT image was only possible if the scanned volume was smaller than the NX size core specimen (i.e., 2 inch in diameter and 4 inch in length). As the tested specimens of Marcellus shale were retrieved from the outcrop, the analyzed samples didn't contain the organic compound, such as kerogen, which is tightly bound with heterogenous mineral matrix [41], rather rich in calcite mineral [25]. The current study presented that, in the open-source software ImageJ, the process of local contrast enhancement using plugin CLAHE, followed by the automated fracture segmentation using plugin TWS, can provide efficient image segmentation on the X-ray CT image with inherent artifacts compared to the conventional thresholding techniques. The study also presented that the application of a macro in ImageJ expedited and automated the process of image segmentation. In addition, both ImageJ and Bruker CTAn software produced comparable results on the geometry of fractures in two and three dimensions. Unlike in Bruker CTAn, the process of data extraction was computationally intensive in ImageJ. However, ImageJ accurately determined each geometrical feature including the 2-D fracture length and surface area of fracture plane. Therefore, the current study presented that the open-source image analysis software can efficiently process and analyze X-ray CT images with user-desired automation. In addition, another open-source software GNU Octave efficiently summarized the 2-D fracture geometry in a user-friendly format.

Appendix 1

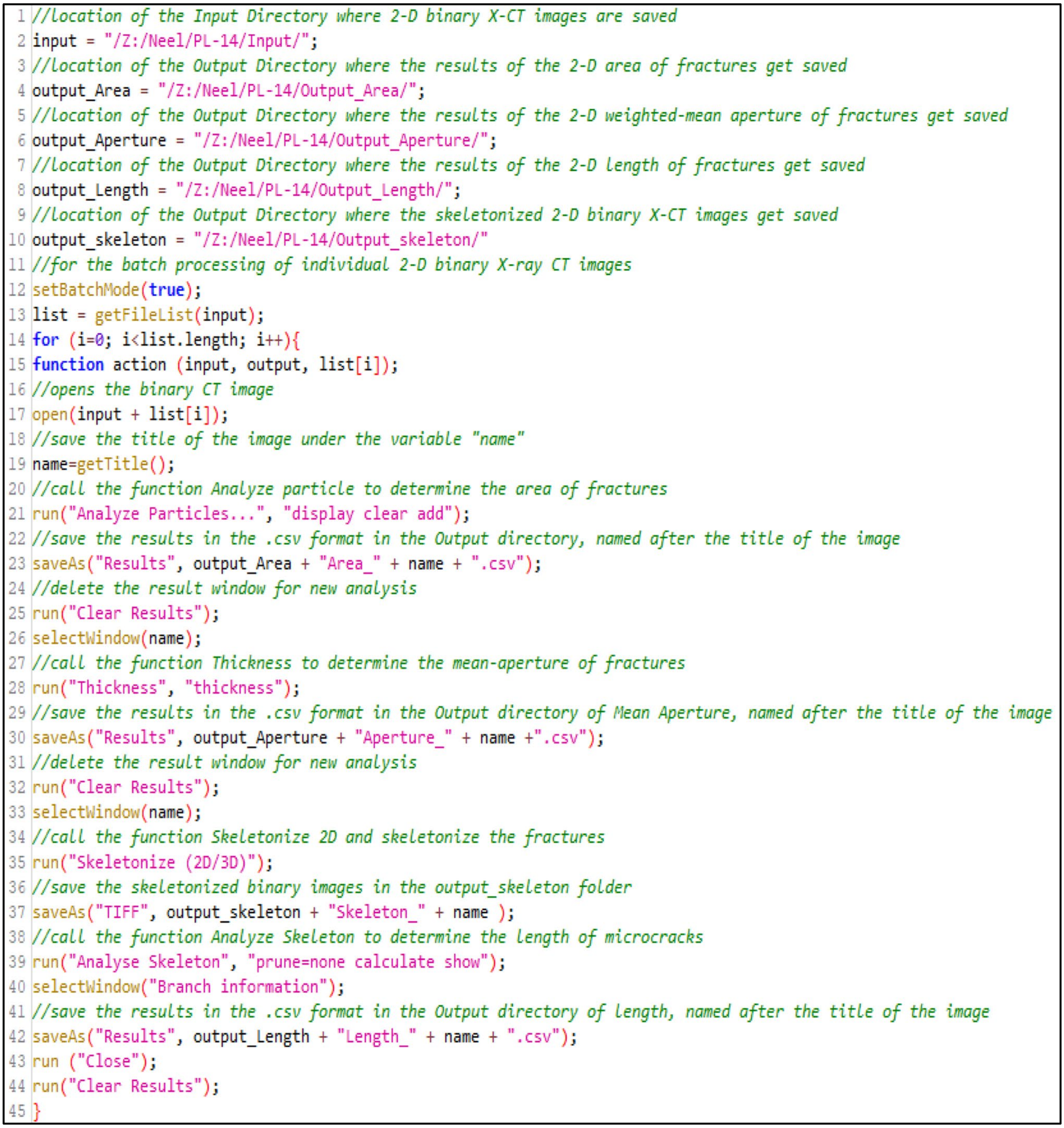
Macro to enhance contrast and segment fractures: case IV.



```
//location of the Input Directory where original X-ray CT images are saved
input_directory= "/C:/Users/negupta/Desktop/Original X-CT Image/";
//location of the Output Directory where the contrast enhanced X-ray CT images will be saved
output_contrast_enhanced = "/C:/Users/negupta/Desktop/Contrast enhanced X-CT Image/";
//location of the Output Directory where the microcrack segmented X-ray CT images will be saved
output_segmentation = "/C:/Users/negupta/Desktop/segmented X-CT Image/"
//location of the claasifer which converts contrast enhanced image into microcrack segmented images
classifier_location = "/C:/Users/negupta/Desktop/general_classifier.model";
setBatchMode(true);
list = getFileList(input_directory);
for (i=000; i<list.length; i++){
function action (input_directory, output, list[i]);
//opens the original X-ray CT image
open(input_directory + list[i]);
//save the title of the image under the variable "name"
name=getTitle();
//calls the function CLAHE to enhance the contrast, parameters decided based on iterations
run("Enhance Local Contrast (CLAHE)", "blocksize=150 histogram=256 maximum=80 mask=*None*");
//saves the contrast enhanced X-ray CT image into the output directory "output_contrast_enhanced"
saveAs("Tiff", output_contrast_enhanced + name);
//runs the plugin Trainable Weka Segmentation
run("Trainable Weka Segmentation");
selectWindow("Trainable Weka Segmentation v2.3.0");
//load the classifier from the "classifier_location"
call("trainableSegmentation.Weka_Segmentation.loadClassifier", classifier_location);
//apply the current classifier to an image
call("trainableSegmentation.Weka_Segmentation.applyClassifier", output_contrast_enhanced,
name, "showResults=true", "storeResults=false", "probabilityMaps=false", "");
selectWindow("Classification result");
//converts the 8-bit color image into 8-bit
run("8-bit");
//converts the 8-bit image into binary image
setOption("BlackBackground", false);
run("Make Binary");
//saves the 2-D binary X-CT image into output directory "output_segmentation"
saveAs("Tiff", output_segmentation + name);
close();
}
```

Appendix 2

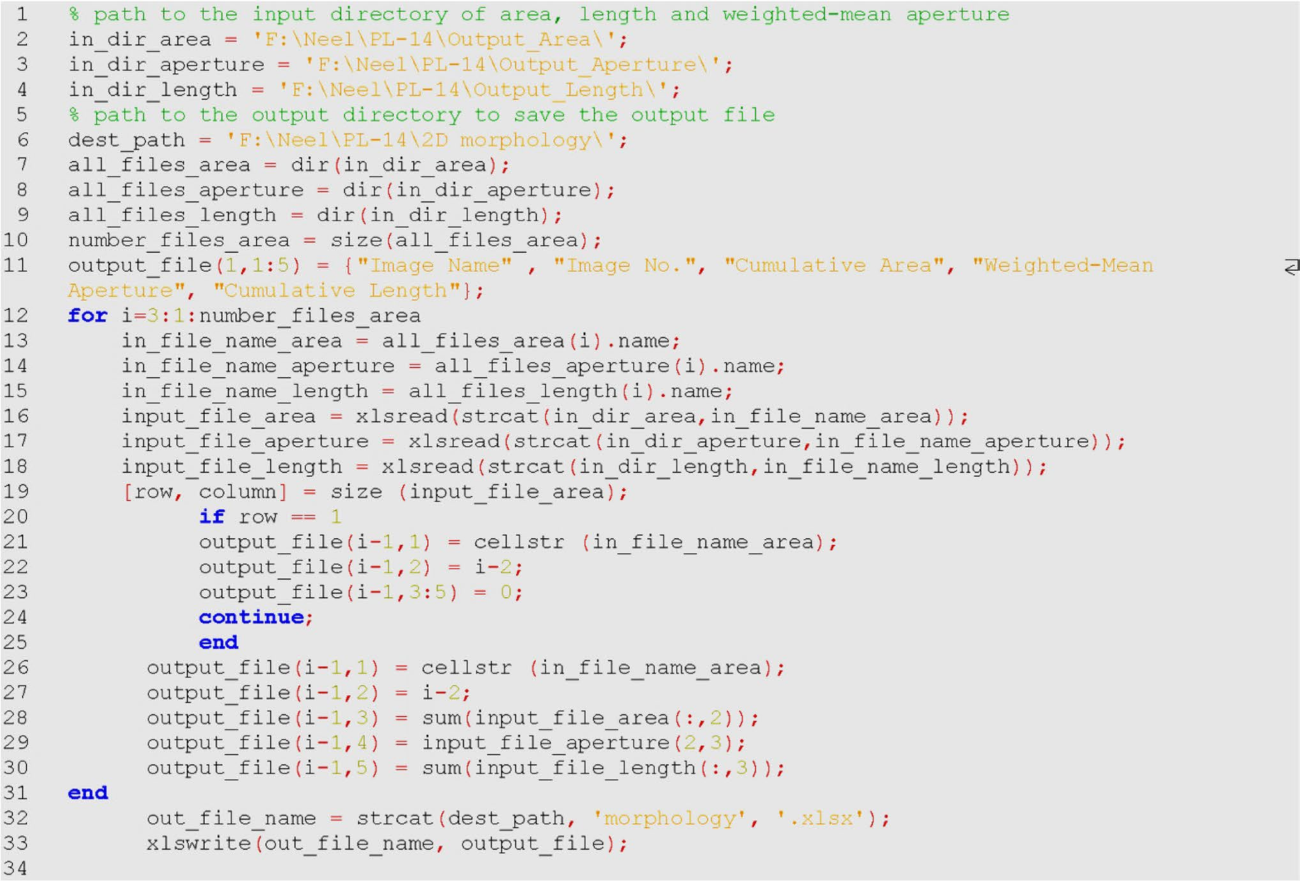
Macro for calculating area, weighted mean aperture, and length of fractures of individual 2-D image in ImageJ.



```
//location of the Input Directory where 2-D binary X-CT images are saved
input = "/Z:/Neel/PL-14/Input/";
//location of the Output Directory where the results of the 2-D area of fractures get saved
output_Area = "/Z:/Neel/PL-14/Output_Area/";
//location of the Output Directory where the results of the 2-D weighted-mean aperture of fractures get saved
output_Aperture = "/Z:/Neel/PL-14/Output_Aperture/";
//location of the Output Directory where the results of the 2-D length of fractures get saved
output_Length = "/Z:/Neel/PL-14/Output_Length/";
//location of the Output Directory where the skeletonized 2-D binary X-CT images get saved
output_skeleton = "/Z:/Neel/PL-14/Output_skeleton/"
//for the batch processing of individual 2-D binary X-ray CT images
setBatchMode(true);
list = getFileList(input);
for (i=0; i<list.length; i++){
function action (input, output, list[i]);
//opens the binary CT image
open(input + list[i]);
//save the title of the image under the variable "name"
name=getTitle();
//call the function Analyze particle to determine the area of fractures
run("Analyze Particles...", "display clear add");
//save the results in the .csv format in the Output directory, named after the title of the image
saveAs("Results", output_Area + "Area_" + name + ".csv");
//delete the result window for new analysis
run("Clear Results");
selectWindow(name);
//call the function Thickness to determine the mean-aperture of fractures
run("Thickness", "thickness");
//save the results in the .csv format in the Output directory of Mean Aperture, named after the title of the image
saveAs("Results", output_Aperture + "Aperture_" + name +".csv");
//delete the result window for new analysis
run("Clear Results");
selectWindow(name);
//call the function Skeletonize 2D and skeletonize the fractures
run("Skeletonize (2D/3D)");
//save the skeletonized binary images in the output_skeleton folder
saveAs("TIFF", output_skeleton + "Skeleton_" + name );
//call the function Analyze Skeleton to determine the length of microcracks
run("Analyse Skeleton", "prune=none calculate show");
selectWindow("Branch information");
//save the results in the .csv format in the Output directory of length, named after the title of the image
saveAs("Results", output_Length + "Length_" + name + ".csv");
run ("Close");
run("Clear Results");
}
```

Appendix 3

Script to summarize 2-D fracture geometry in GNU Octave.



```
% path to the input directory of area, length and weighted-mean aperture
in_dir_area = 'F:\Neel\PL-14\Output_Area\';
in_dir_aperture = 'F:\Neel\PL-14\Output_Aperture\';
in_dir_length = 'F:\Neel\PL-14\Output_Length\';
% path to the output directory to save the output file
dest_path = 'F:\Neel\PL-14\2D morphology\';
all_files_area = dir(in_dir_area);
all_files_aperture = dir(in_dir_aperture);
all_files_length = dir(in_dir_length);
number_files_area = size(all_files_area);
output_file(1,1:5) = {"Image Name" , "Image No.", "Cumulative Area", "Weighted-Mean Aperture", "Cumulative Length"};
for i=3:1:number_files_area
    in_file_name_area = all_files_area(i).name;
    in_file_name_aperture = all_files_aperture(i).name;
    in_file_name_length = all_files_length(i).name;
    input_file_area = xlsread(strcat(in_dir_area,in_file_name_area));
    input_file_aperture = xlsread(strcat(in_dir_aperture,in_file_name_aperture));
    input_file_length = xlsread(strcat(in_dir_length,in_file_name_length));
    [row, column] = size (input_file_area);
          if row == 1
          output_file(i-1,1) = cellstr (in_file_name_area);
          output_file(i-1,2) = i-2;
          output_file(i-1,3:5) = 0;
          continue;
          end
      output_file(i-1,1) = cellstr (in_file_name_area);
      output_file(i-1,2) = i-2;
      output_file(i-1,3) = sum(input_file_area(:,2));
      output_file(i-1,4) = input_file_aperture(2,3);
      output_file(i-1,5) = sum(input_file_length(:,3));
end
      out_file_name = strcat(dest_path, 'morphology', '.xlsx');
      xlswrite(out_file_name, output_file);
```

Acknowledgements The authors would like to thank Sarah Brown, Johnathan E. Moore, and Bryan Tennant at the National Energy Technology Laboratory in Morgantown, WV, for X-ray CT scan of shale specimen. The authors would also like to thank Dr. Karen Martin and Sarah McLaughlin from Animal Models and Imaging Facility (AMIF) of West Virginia University (WVU) for providing the access of Bruker CTAn software.

Funding Financial support for this work was provided by the Centers for Disease Control and Prevention–National Institute of Occupational Safety and Health (No. 200–2016-92214).

Data Availability All data used to support the findings of the current study are available from the corresponding author upon request.

Code Availability Attached in the manuscript.

Declarations

Conflict of Interest There is no conflict of interest between authors regarding the publication of this paper.

References

1. Baraka-Lokmane S, Main IG, Ngwenya BT, Elphick SC (2009) Application of complementary methods for most robust characterization of sandstone cores. Mar Pet Geol 26(1):39–56
2. Cnudde V, Boone MN (2013) High-resolution X-ray computed tomography in geosciences: a review of the current technology and applications. Earth Sci Rev 123:1–17
3. Nehler M, Andolfsson T, Renner J, Steeb H, Bracke R (2015) Laboratory setups for core flooding and CT scanning experiments at in-situ HP/HT conditions. Proceedings World Geothermal Congress, (p. 7). Melbourne, Australia
4. Sun W, Brown SB, Leach RK (2012) An overview of industrial X-ray computed tomography. National Physical Laboratory, Engineering Measurement Division. Teddington, Middlesex
5. Rodriguez R, Crandall D, Song X, Verba C, Soeder D (2014) Imaging techniques for analyzing shale pores and minerals. U.S. Department of Energy, National Energy Technology Laboratory
6. Efford N (2000) Digital image processing: a practical introduction using Java. Harlow, Pearson Education Ltd.

7. Rogowska J (2008) Overview and fundamentals of medical image segmentation. In: Bankman, I. N. (eds.) Handbook of medical image processing and analysis, pp. 73–90. Academic Press
8. Sahoo PK, Soltani S, Wong A, Chen YC (1988) A survey of thresholding techniques. Computer Vision, Graphics and Image Processing 41:233–260
9. Sonka M, Hlavac V, Boyle R (1999) Image processing, analysis, and machine vision. PWS Publishing, Pacific Grove, CA
10. Maerz, N (1990) Photoanalysis of rock fabric. PhD Thesis, University of Waterloo, Canada
11. Reid TR, Harrison JP (2000) A semi-automated methodology for discontinuity trace detection in digital images of rock mass exposures. Int J Rock Mech Min Sci 37:1073–1089
12. Lester JM, Brenner JF, Selles WD (1980) Local transforms for biomedical image analysis. Comp Graph Imag Proc 13:17–30
13. Schmid P (1999) Segmentation of digitized dermatoscopic images by two-dimensional color clustering. IEEE Trans on Med Imag 18(2):164–171
14. Sakthivel N, Prabhu L (2014) Mean-median filtering for impulsive noise removal. International Journal of Basics and Applied Sciences 2(4):47–57
15. Adams R, Bischof L (1994) Seeded region growing. IEEE Trans Pattern Recogn Mach Intell 16(6):641–647
16. Gouze P, Noiriel C, Bruderer C, Loggia D, Leprovost R (2003) X-ray tomography characterization of fracture surfaces during dissolution. Geophysical Research Letters 30(No. 5)
17. Noiriel C, Made B, Gouze P (2007) Impact of coating development on the hydraulic and transport properties in argillaceous limestone fracture. Water Resour Res 43(W09406):16
18. Wang W, Hakami E (2005) A segmentation algorithm for rock fracture detection. Third International Conference on Advances in Pattern Recognition, pp. 580–588. Bath, UK: Springer
19. Deng H, Fitts JP, Peters CA (2016) Quantifying fracture geometry with X-ray tomography: technique of iterative local thresholding (TILT) for 3-D image segmentation. Comput Geosci 20:231–244
20. Hsieh J (2015) Image artifacts: appearances, causes, and corrections. Computed tomography: principles, design, artifacts, and recent advances. Third Ed. Jiang Hsieh. Belligham: SPIE, 245–338
21. Axelsson M, Svensson S, Borgefors G (2006) Reduction of ring artifacts in high resolution X-ray microtomography images. In: Franke, K., Muller, K.R., Nickolay, B., Schafer, R. (eds) Pattern Recognition. DAGM, 61–70
22. Mohapatra S (2012) Development and quantitative assessment of a beam hardening correction model for preclinical micro-CT. Master's Thesis, The University of Iowa, p. 91
23. Meganck JA, Kozloff KM, Thornton MM, Broski SM, Goldstein SA (2009) Beam hardening artifacts in micro-computed tomography scanning can be reduced by X-ray beam filtration and the resulting images can be used to accurately measure BMD. Bone 45(6):1104–1116
24. Schneider CA, Rasband WS, Eliceiri KW (2012) NIH image to ImageJ: 25 years of image analysis. Nat Methods 9(7):671–675
25. Gupta N (2019) Fundamental mechanism of time dependent failure in shale. Dissertation, West Virginia University
26. Plessis AD, Broeckhoven C, Guelpa A, Roux SI (2017) Laboratory X-ray micro-computed tomography: a user guideline for biological samples. Gigascience 6(6):1–11
27. Glasbey CA (1993) An analysis of histogram-based thresholding algorithms. CVGIP: Graphical Models and Image Processing 55, 532–537
28. Phansalskar N, More S, Sabale A, et al (2011) Adaptive local thresholding for detection of nuclei in diversity stained cytology images. International Conference on Communications and Signal Processing, pp. 218–220. Kerala, India
29. Sauvola J, Pietaksinen M (2012) Adaptive document image binarization. Pattern Recogn 33(2):225–236
30. Romano C, Minto J, Shipton ZK, Lunn R (2018) Automated high accuracy, rapid beam hardening correction in X-ray computed tomography of multi-mineral, heterogenous core samples. InterPore 2018. New Orleans
31. Boas FE, Fleischmann D (2012) Computed tomography artifacts: causes and reduction techniques. Imaging in Medicine 4(2):229–240
32. Zuidervald K (1994) Contrast limited adaptive histogram equalization. In: Heckbert PS (ed) Graphic Gems IV. Academic Press Professional, Inc., San Deigo, CA, USA, pp 474–485
33. Nithya K, Karthik K, Ramasamy K (2014) Comparison of contrast enhancement technique with partitioned iterated function system. Int J Sci Eng Res 5(5):167–172
34. Yadav G, Maheshwari S, Agarwal A (2014) Contrast limited adaptive histogram equalization based enhancement for real time video system. International Conference on Advances in Computing, Communications and Informatics pp. 2392–2397. Delhi, India
35. Arganda-Carreras I, Kaynig V, Rueden C, Eliceiri KW, Schindelin J, Cardona A, Seung HS (2017) Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. Bioinformatics 33(15):2424–2426
36. Arena A, Piane CD, Sarout J (2014) A new computational approach to cracks quantification from 2D image analysis: application to micro-cracks description in rocks. Comput Geosci 66:106–120
37. Doube M, Klosowski MM, Arganda-Carreras I, Cordelieres F, Dougherty RP, Jackson J, Shefelbine SJ (2010) BoneJ: free and extensible bone image analysis in ImageJ. Bone 47(6):1076–1079
38. Dougherty R, Kunzelmann KH (2007) Computing local thickness of 3D structures with ImageJ. Microsc Microanal 13(S02):1678–1679
39. Darma IS, Sugiyama T, Promentilla MAB (2013) Application of X-ray CT to study diffusivity in cracked concrete through the observation of tracer transport. J Adv Concr Technol 11:266–281
40. Lorensen WE, Cline HE (1987) Marching cubes: a high resolution 3D surface construction algorithm. 14th Annual Conference on Computer Graphics and interactive techniques, pp. 163–169. https://doi.org/10.1145/37401.374220808
41. Tiwari P, Deo MD, Lin CL, Miller JD (2013) Characterization of oil shale pore structure before and after pyrolysis by using X-ray micro CT. Fuel 107:547–554

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.