Comparison of Edge Detection Algorithms for Automatic Identification of Fractures in Hydrocarbon Reservoirs with Image Logs

Mina Shafiabadi*, and Abolghasem Kamkar-Rouhani

Faculty of Mining, Petroleum & Geophysics Engineering, Shahrood University of Technology, Shahrood, Iran

Article Info	Abstract
	Considering the effect of fractures in increasing hydrocarbon recovery, the study of reservoir rock fractures is of particular importance. Fractures are one of the most important fluid flow paths in carbonate reservoirs. Image logs provide the ability to detect fractures and other geological features and reservoir layers. In this study, two approaches were used to detect fractures using FMI image log in two wells A and B located in one of oilfields in southwest of Iran. In the first stage, the correction and processing of the FMI raw data were carried out to identify the number and position of fractures, as well as the dip, extension, classification, and density of fractures. In the second step, by considering that the fractures possess the edges in the FMI images, various edge detection filters such as Prewitt, Canny, Roberts, LOG, Zero-cross and Sobel were applied on the image data, and then, their performances for identification of fractures were compared. Finally, the automatic identification of fractures was done by applying the Hough transform algorithm and the results showed that Canny algorithm was the best option to perform Hough transformation. The comparison of the efficiency of the above-mentioned edge detection filters for identification of fractures, and more importantly, the automatic identification of fractures using the Hough transform algorithm can be considered as the novelty of this research work.

List of Acronyms

Formation Micro Imager	FMI
Charge-Coupled Device	CCD
Support Vector Machine	SVM
Pulse-Coupled Neural Networks	PCNN
Laplacian Of Gaussian operator	LOG
Hough Transform	HT
Red, Blue, and Green	RGB
Non-Maximum Suppression	NMS
Higher Threshold	T_H
Lower Threshold	T_L

1. Introduction

Fractures are one of the most important fluid flow paths in carbonate reservoirs [1]. Regarding the importance of fracture properties, their effective role in increasing porosity, permeability and consequently high oil production in these reservoirs can be mentioned [2]. There are several ways to identify reservoir fractures. One of the most important sources for studying the properties of the reservoir is the drilling core. Core study is a typical small-scale method for determining fractures in the well. In fractured reservoirs, the use of core has three limitations. High cost of core preparation, non-orientation and low recovery in fractured zones [3; 4]. Image logs do not have these limitations compared to cores. Image logs are a cylindrical, virtual image of a high-resolution well wall capable of displaying subtle wall phenomena [5]. FMI (Formation Micro Imager) is a new generation of imager logs that, by measuring the relative specific resistance, provides a cylindrical and virtual image of the well wall, which is able to show the subtle phenomena of the wall. Common applications of these images include: characterization of fracture properties, reservoir

✉ *Corresponding author: mina.shafiabadii@gmail.com (M. Shafiabadi)*

structure, bedding and dip detection, porosity and permeability, fault diagnosis and orientation, evaluation of stresses on wellbore [6; 7]. Many studies have been performed to identify fractures in different ways. Wang et al (2007) developed a new algorithm to detect fractures in rock using images taken from rock with advanced CCD (charge-coupled device) cameras. After removing noise, they segmented the image based on the edges and extracted eleven features using the support vector machine (SVM) to separate fractures from other phenomena [41]. Wang and Wang (2010) revealed the edges using an ultraviolet image taken from rocks alongside optical images. He used the Canny edge detection and the threshold to detect the edges. He then removed the noise and found the fracture curves. He then proceeded to attach the fragments to a fracture and fill the incisions [38]. He and Wang (2010) used a new type of neural network PCNN (pulse-coupled neural networks), to detect fractures using a new method for edge detection [39]. Seifallahi et al (2013) separated image fractures from wells using image processing and artificial intelligence techniques. They used the color feature as the parameter to determine the fracture points. They used the self-organized map (SOM) network algorithm to separate the pixels of the natural fracture points [40]. Assous et al (2014) were able to create a new algorithm for detecting fractures and separating features of sinusoidal planes with edge information [35]. Shafiabadi et al (2021) Identified reservoir fractures from an FMI imaging log using Canny and Sobel edge detection algorithms. The results of comparing the two methods showed that the use of Canny edge detection method helps the interpreter to identify

fractures [36]. Shafiabadi et al (2021) used the Canny Edge Algorithm to identify fractures and their dip by applying the Hough transform algorithm (HT) [37]. Zhang et al (2021) identified fractures using an improved ant colony method [42]. One of the goals and innovations of the current research is to use a variety of edge detection algorithms such as Canny, Sobel, Roberts, Prewitt, LOG and Zero-Cross edge detection algorithms for detection of fractures. These algorithms can be used to identify the fracture edges of oil reservoirs that appear as sinusoidal curves in the FMI image log. The next step after detecting fracture edges is to compare the performance of edge detection algorithms in order to choose the most appropriate algorithm, and then to apply the Hough transform algorithm for automatic detection of fractures in this research.

2. FMI imaging tool

The FMI tool was developed in 1991 by Schlumberger. The FMI tool is mounted on four main pads and four secondary pads called flaps. These eight pads/ flaps are composed of two rows of 12 buttons that result in 192 sensor electrodes [8]. The pads attach to the wall of the well by means of arms to allow good contact between the electrodes and the wall. The FMI tool has the ability to cover 80% of the inner wall of the well in 8.5 inch diameter wells. Figure 1 illustrates the FMI measurement principle and FMI tool configuration. Generally, in FMI logs, layers and fractures in the well appear as sinusoidal curves. From its sinusoidal curves, azimuth and dip of the layer can be obtained [9; 8].
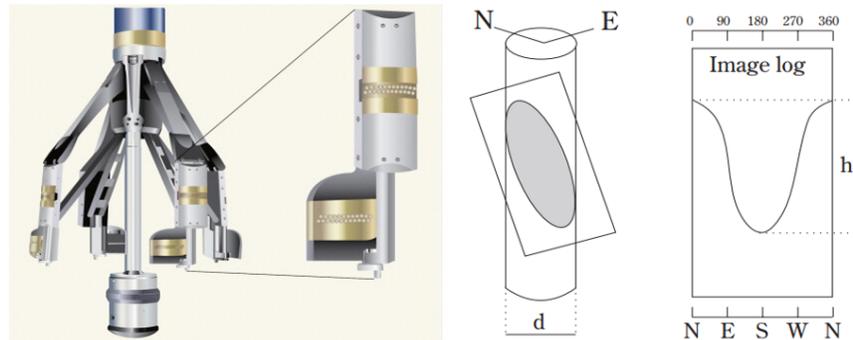
Figure 1. FMI tool configuration. Projection of a planar intersection with a cylindrical borehole. Dip direction of the planar feature is given by the orientation of the sinusoid minimum; dip angle = arc tan (h/d) where h = height of sinusoid and d = borehole diameter [10; 8].

3. Geological setting

Zagros sedimentary basin is one of the most important oil basins in the world located in

southwestern Iran and northern Iraq. This belt was created by the closure of the young Tethys Ocean and the collision of the Arabian plate with Eurasia [11; 12]. The Dezful embayment is part of the

folded Zagros Belt located in the southern part of Khuzestan. It covers an area of about 60,000 kilometers and has 45 oilfields in the area [13]. The study area is located in the Dezful embayment in the Zagros Basin (Figure 2). In this research, 2 wells numbered 245 and 314 from the Gachsaran oil field located in the southwest of Iran were selected using GEOLOG software to identify and interpret the fractures. Gachsaran field is about 70 kilometers long and has a variable width of 6 to 15 kilometers [12]. Figure 3 shows the stratigraphic column of Zagros.
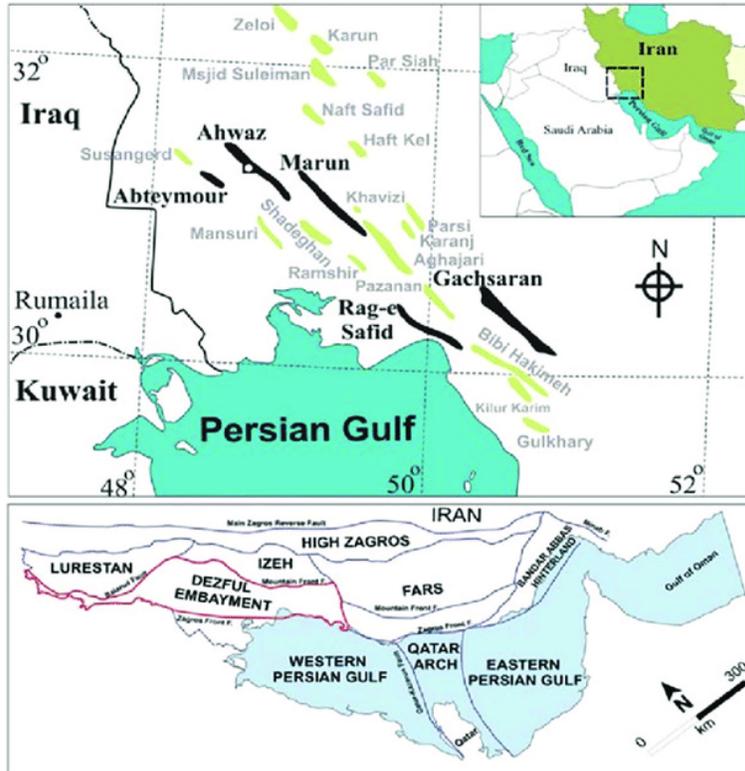
Figure 2. Location of Gachsaran oil field in Dezful embayment, southwest of Iran [14].
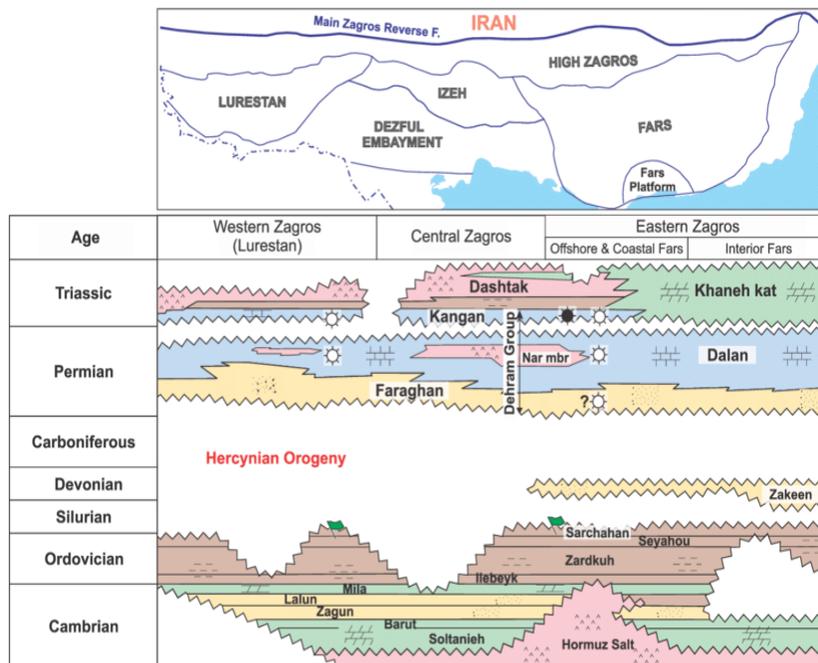
Figure 3. The stratigraphic column of the Zagros fold-thrust belt [15].

4. Fractures analysis from FMI image logs
4.1. Drilling induced fractures

These fractures are caused by drilling-related factors artificially and by the uneven distribution of tangential stresses applied to the well wall at the intersection of maximum horizontal stresses and minimum horizontal stresses. These fractures are semi-stable and over time the well begins to breakout in these areas, and it causes the well opening to become oval [16]. The pattern of these fractures on the FMI image logs is in the form of lines perpendicular to the layering (Figure 4a).

4.2. Closed fractures

Closed fractures are filled by cement and appear as white sine waves on FMI image logs. Cemented surfaces indicate permeable barriers to fluid flow (Figure 4b) [17; 10].

4.3. Open fractures

In open fractures or fractures filled by conductive materials, the structures appear dark in color. Because the matrix resistance is greater than the drilling mud, open fractures in the image logs appear to be continuous, dark or opaque sine waves (Figure 4c) [18; 10].
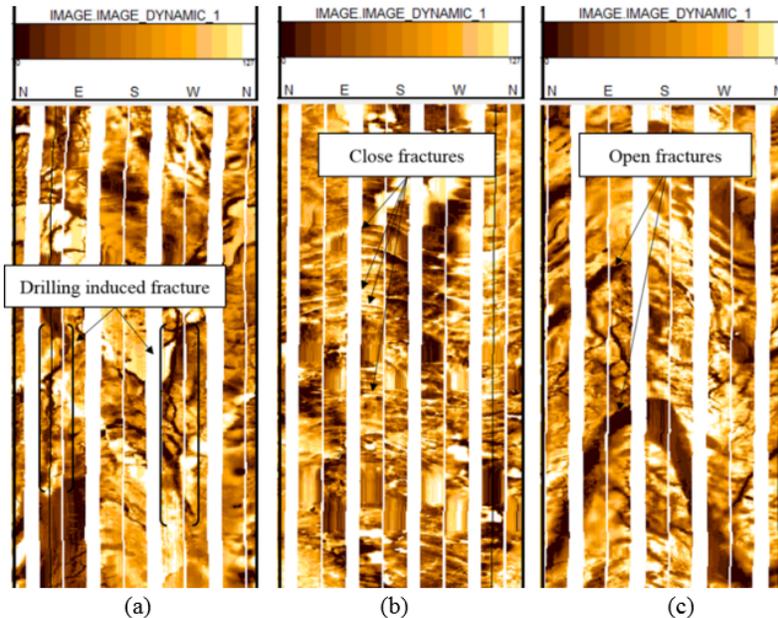
Figure 4. Example of fractures in wellbore image logs, a) Drilling induced fracture, b) Close fractures and c) Open fractures.

4.4. Fault

Faults are a type of fractures. The slight difference in depth between a similar images on either side of a bent surface may indicate a fault, so sudden changes in the dip between two sets of images may be a fault. The factors that help to identify the Fault are sudden change in dip and layering, Fractures around the fault and sudden change in the direction and angle of the well [19; 17].

5. Edge detection techniques

Image processing is the best tool for feature extraction and position analysis. The edge detection process is one of the most effective and useful techniques in image processing, especially in isolating and identifying the original image frame [20]. The edge can be defined as the discontinuity in the intensity of light from one pixel to another. The reason for creating an edge in an image is the difference in light intensity on both sides of the edge [21]. Many methods are used for edge detection, generally there are 2 general classification categories for edge detection: Laplacian-based methods and gradient-based methods. In these methods, the gradient and Laplacian stereotypes are matched by the convolution operator throughout the image points and reveal the amount of changes in the illuminance level in several limited directions and by applying a threshold value on the resulting image, edges related to stereotypes are extracted.

5.1. Edge detection using the gradient method

The gradient works on the first derivative. Edge detection filters based on the first derivative include Prewitt, Roberts, Sobel and Canny edge detection filters [22; 20].

5.1.1. Roberts operator

This filter is one of the first methods of image edge recognition that uses two 2×2 matrices. The main purpose of this filter is to extract diagonal edges in the image and is based on the implementation of diagonal differences. The only drawback of this algorithm is its high sensitivity to noise due to the use of low points in derivative estimation [20]. The Roberts filter procedure is illustrated in Figure 5. This is very similar to the Sobel operator. The gradient magnitude is given by [23]:

$$|G| = \sqrt{G_x{}^2 + G_y{}^2} \qquad (1)$$

Finally, it becomes the following equation:

$$|G| = |G_x| + |G_y| \qquad (2)$$

The angle of orientation of the edge given by:

$$\theta = \text{Arctan}\frac{G_y}{G_x} - 3\pi/4 \qquad (3)$$

+1	0
0	-1

0	-1
+1	0

Gx   Gy
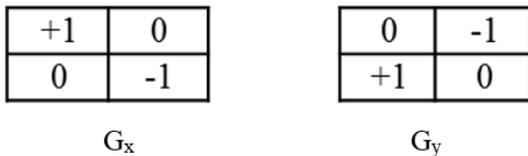
Figure 5. Masks used for gradient operations on Roberts operator [24].

5.1.2. Prewitt operator

As the simplest filter with a 3x3 mask, this filter is one of the filters that are symmetrical around the central point. The difference between the third and first row approximates the derivative in the x direction and the difference between the third and first column approximates the derivative in the y direction in the desired area (Figure 6). This operator is used to find vertical and horizontal edges [20; 25; 26]. The gradient magnitude is given by:

$$|G| = \sqrt{G_x{}^2 + G_y{}^2} \qquad (4)$$

Finally as:

$$|G| = |G_x| + |G_y| \qquad (5)$$

The angle of orientation of the edge given by the following equation:

$$\theta = \text{Arctan}\frac{G_y}{G_x} \qquad (6)$$

-1	-1	-1
0	0	0
+1	+1	+1

-1	0	+1
-1	0	+1
-1	0	+1

Gx   Gy
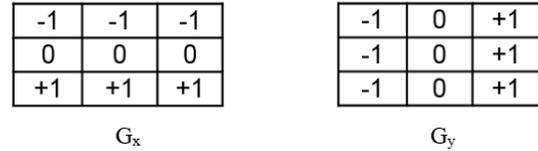
Figure 6. Masks used for gradient operations on Prewitt operator [24].

5.1.3. Sobel operator

The Sobel edge detection method created using the first-order derivative approximation. The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives – one for horizontal changes, and one for vertical [27; 20]. The G_y filter finds the horizontal edges and the G_x filter finds the vertical edges (Figure 7). The combination of these two filters finds all the horizontal and vertical edges of the image. The gradient magnitude is given by the following equation:

$$|G| = \sqrt{G_x{}^2 + G_y{}^2} \qquad (7)$$

Finally as:

$$|G| = |G_x| + |G_y| \qquad (8)$$

The angle of the edge direction in the Sobel algorithm is calculated by the following equation:

$$\theta = \text{Arctan}\frac{G_y}{G_x} \qquad (9)$$

-1	-2	-1
0	0	0
+1	+2	+1

-1	0	+1
-2	0	+2
-1	0	+1

Gx   Gy
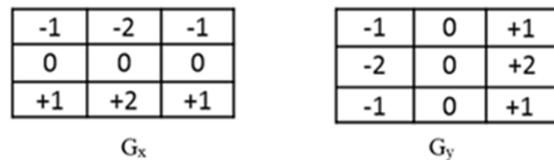
Figure 7. Masks used for gradient operations on Sobel operator [24]

5.1.4. Canny operator

The Canny edge detection technique was first developed by John Canny for his master's thesis at MIT in 1983 and this algorithm uses the first-order derivative of the image [28; 24]. Canny edge

detection technique is known for its ability to produce thin edges up to one pixel for continuous edges. The steps of Canny algorithm are as follows (Figure 8) [25; 24]:

1. Noise reduction: The input image may contain noise, and if the image noise is not reduced, many points in the image will be incorrectly identified as edges. Therefore, a Gaussian filter is applied to the image to reduce the image noise.

2. Gradient calculation; in the second step of the canny algorithm, the intensity and direction (It determines the direction of changes in brightness) of rotations are calculated.

3. Non-maximal suppression; in this section, non-maximal suppression is performed to thin the edges.

4. Double threshold; in this section, two threshold limits (upper threshold limit and lower threshold limit) are selected for the gradient image (the output image of the third step).

5. Hysteresis thresholding: In the previous step, the task of the strong pixels was determined, and only the weak pixels remain, and their task is also determined in this step. At this stage, a final decision must be made for weak pixels, either they should be turned into edges (strong pixels) or removed (irrelevant pixels).
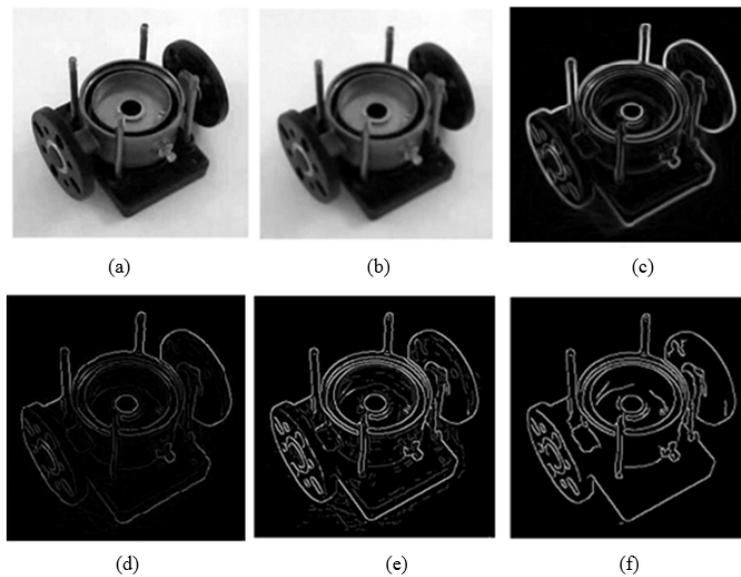
Figure 8. a) Original image, b) Smoothed image using Gaussian filter to remove noise, c) Calculated gradient amplitudes along with their directions, d) Removing non-maximum points to remove false edges, e) thresholding and f) The final image.

5.2. Edge detection using the Laplacian method

Obviously, wherever the first derivative is maximal, the second derivative will be zero. So another way to find the edge is to use a second derivative, which is the Laplacian method. The sensitivity of the Laplacian operator to noise is very high because it is correlated with zero crossing points, while in most cases the noise signal is small and near zero [29].

5.2.1. LOG operator

The Laplacian of Gaussian operator (LOG) uses the second derivative of the image and is independent of the direction of the boundaries, for this reason this operator is a scalar value and not a vector. The sensitivity of Laplacian operator to

noise is very high because it is related to zero crossing points and this is while in most cases the noise signal also has small values and close to zero. For this reason, first a low-pass filter is applied to the image to reduce the effect of noise, and then the Laplacian operator is applied. This is done using a 3×3 mask in Figure 9 and is defined as follows [20]:

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \qquad (10)$$

0	-1	0
-1	4	-1
0	-1	0

G_x

-1	-1	-1
-1	8	-1
-1	-1	-1

G_y

Figure 9. Masks used for gradient operations on LOG operator [24].

5.2.2. Zero-Cross method

One of the best methods of edge detection is the Zero-Crossing method. The starting point for the zero crossing detector is an image filtered using a Gaussian Laplacian filter. The output from the zero-crossing detector is usually a binary image with single thick lines indicating the positions of the Zero-Crossing points. Zero-Crossings also occur wherever the gradient of the image intensity begins to increase or decrease, and this may occur in places where edge sharpness does not occur [30; 31; 24].

6. Methodology

Converting the color image from the GEOLOG software to the gray level image is the first step of the research under study. An RGB image consists of three matrices, each of which holds the G, R, and B (red, blue, and green) values of the color image. The display of the image on the screen is done by combining the values of the corresponding levels in three matrices. A pixel will have a gray value when it's R, G, B components have the same values. In this step, we convert the RGB image to gray scale (Figure 10).

Figure 10. Results of edge detection on the FMI Image. a) Original image, b) Gray-scale image.

6.1. Preprocessing

In the preprocessing section, the size and direction of the gradient of the image is calculated. For this purpose, the image is first smoothed with a Gaussian filter, which reduces image noise. The value of the standard deviation of the Gaussian function is adjusted according to the amount of image noise. Then, the size and direction of the image gradient is calculated by applying Roberts, Sobel and the other filters in both vertical and horizontal directions.

6.1.1. Selection of initial edge point

In this section, edge candidate points are extracted based on the gradient and Laplace criteria of the image. In the proposed algorithm, the method of removing Non-maximum Suppression (NMS) and the Laplace zero crossing condition are used to determine edge candidate pixels. First, in order to select the initial points of the edge candidates, the gradient size image is thresholded.

6.1.2. Non-maximum suppression

The purpose of the canny algorithm is to find the center of the edges of the image. In the gradient intensity image, the places where there is an edge have a high thickness. If the same image is used directly, the edges obtained in the final image will have a high thickness, which is not desirable. To solve this problem, in the third step, the non-maximum points of the image are suppressed so that the thickness of the edges of the intensity image is reduced and suitable for the interpretation of the shape [24].

6.2. Postprocessing

The last step of the proposed algorithm is hysteresis thresholding on the edge image.

6.2.1. Hysteresis threshold

In the last step, the edges should be extracted by thresholding the images obtained from the previous step, which is done by the Canny algorithm with the hysteresis threshold method. In this method, there are two thresholds T_H and T_L, pixels that have a value greater than T_H are called strong edge pixels and pixels that have a value between T_L and T_H are called weak pixels [32; 24; 33]. The upper and lower thresholds obtained from this research are as follows (See the Appendix for more details):

$$Th = [0.3, 0.4], \sigma = \sqrt{10}$$

"The reason for using this threshold range in this study was that the lower the thresholds such as $Th = [0.1, 0.2]$ applied to the image, the more edges would appear other than the fractures. The higher the thresholds such as $Th = [0.7, 0.9]$ are applied to

the image, the edge of the sinusoidal curve of the fractures is removed. As a result, the most moderate Terscheld in this study (Th=[0.3, 0.4]) was selected so that neither the many edges of the

fractures were removed nor the many edges other than the fractures appeared". Figure11 shows an example of different thresholds, the wrong choice of threshold can lead us away from the target.

(a)   (b)   (c)
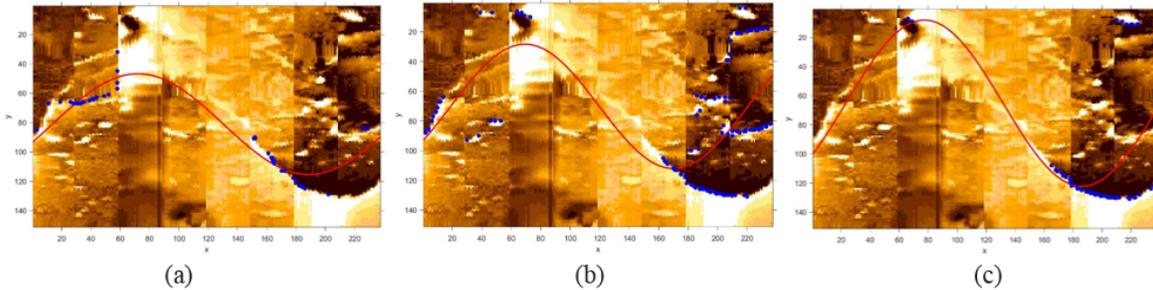
Figure 11. An example of incorrect thresholds that are not fitted to the fracture curve. a) Th=[0.1 0.6],Sigma=sqrt(15), b) Th=[0.4 0.6],Sigma=sqrt(1) and c) Th=[0.6 0.9],Sigma=sqrt(3)

7. Hough transform algorithm

Hough transform is a method to extract features in image analysis and digital image processing. The main idea of Hough's method is that we start from a point which is the pixels on the edge of the input image. We assume that (X, Y) is the coordinate of this point on the image, the equation of the line passing through this point is represented by a formula in the form of Y = aX+b and all (a, b) that

apply to this equation are stored in an accumulator array. For example, suppose one of the points is $(X, Y) = (1, 1)$. The form of the equation is $1 = a.1 + b$ and this form of the equation can also be converted to $b = -a + 1$. Therefore, the above equation includes all pairs of points associated with a single point (1, 1) which is shown in Figure 12 [34].
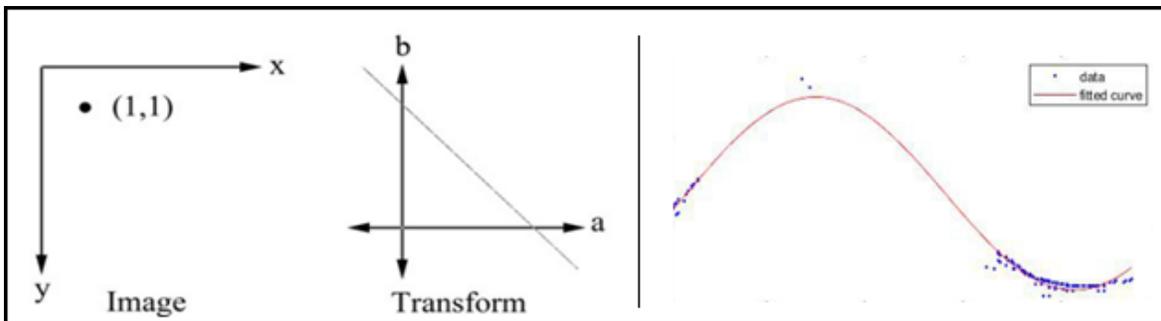
Figure 12. A schematic presentation of a line associated with the point $(1, 1)$.

8. Results analysis and discussion

Since color plays an important role in image analysis and color images have more information than gray and black and white images, in this study, the operation of the edge detection algorithm is implemented on FMI color images to obtain a more favorable result. Figure 13 shows the flowchart of the method of doing the research under study.
Figure 14 shows the results of implementation of the Canny, Sobel, Roberts, Prewitt, LOG and zero cross edge detection algorithms in MATLAB environment on a closed fracture at a depth of 2701 meters (True Dip Angle= 67.51, True Dip Azimuth= 301.93) in the FMI image log of well A

that is depicted as a sinusoidal curve in each Figure looking differently. Figure 15 shows the results of implementation of the Canny, Sobel, Roberts, Prewitt, LOG, and zero cross edge detection algorithms on the fault at a depth of 2173 meters on the FMI image log of well B that is demonstrated as a sinusoidal curve in each part of the Figure. Figure 16 shows an example of open fracture (Major open fracture) at a depth of 2560 meters (True Dip Angle=66.39, True Dip Azimuth= 330.32) from well A in Geolog software with the results of Edge detection algorithms, Sobel, Roberts, Prewitt, Zero-cross, LOG and Canny.
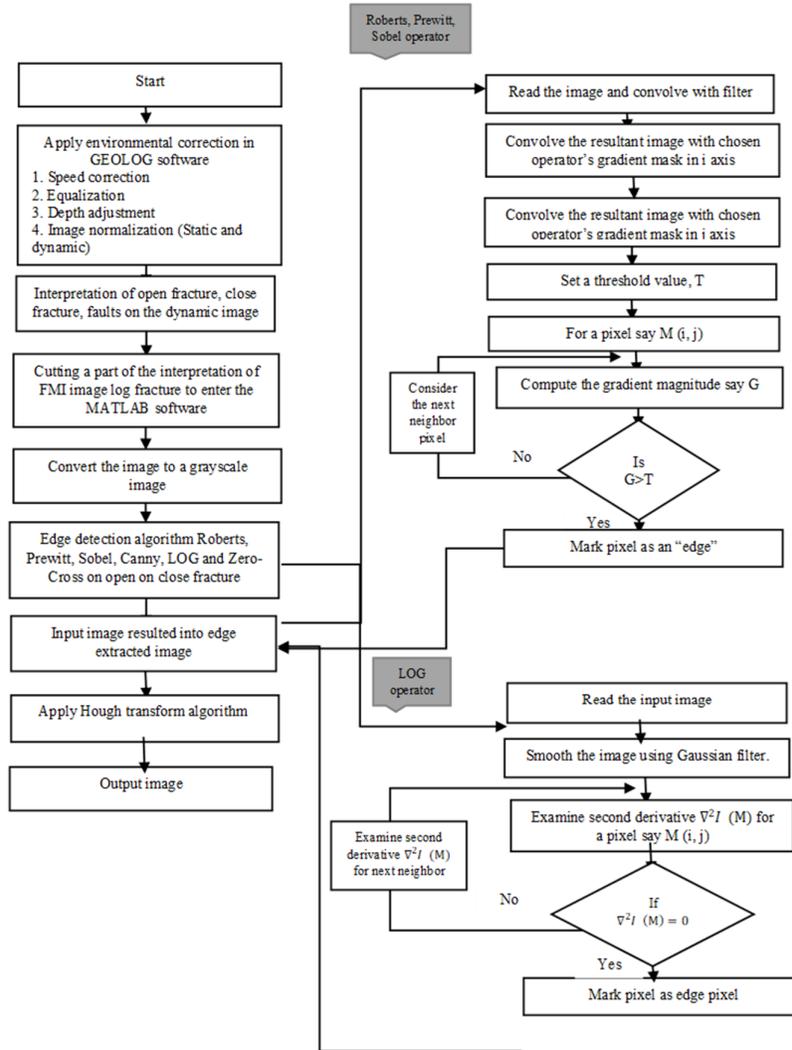
Figure 13. Flowchart of general algorithm for Roberts, Prewitt, Sobel and LOG operators

According to Figure 14, Figure 15 and Figure 16, it can be clearly seen that the Canny edge detection performance is better due to the continuous sinusoidal curves in the FMI image log versus the performance of Roberts, Prewitt, Sobel, LOG and Zero-cross edge detection. The Roberts, Prewitt and Sobel methods have curves with discontinuous edges. Thus, the Canny method acts better and in the meantime, continuity of edges in this method is strong. Table 1 compares different edge detection techniques.

Table 1 Comparison between edge detection techniques on fracture.

Method	Advantage	Disadvantage
Roberts	2*2 Mask, Used for image segmentation, less computation time, Simplicity	Highly Sensitive to noise
Sobel	3*3 Mask, Used for image segmentation, Simplicity	Computation time is high compared to Roberts operator, Less Sensitive to noise compared to Roberts operator
Prewitt	3*3 Mask, Used for image segmentation, Similar to Sobel operator, Smooth edge region, Simplicity	Suitable for noiseless image
Canny	3*3 Mask, Performance is good, Used for image enhancement, Smooth noise	High computation time, complex process
LOG	Covers wider area around the pixels, meanwhile finds correct places of the edges	High chances of finding false edges and localization errors on the curve edges
Zero-Cross	Detection of edges and their orientations have fixed characteristics among all the direction	Sensitive to image noise and re-responds to the some of the existing edges

The next step after choosing the best edge detection algorithm is to apply the Hough transform algorithm. Figure 17 shows the results of automatic fault detection by applying the Hough transform algorithm with different thresholding and the changes of each one. As shown in Figure 17, the result obtained from sigma=sqrt (20) shows a better fit than other thresholds.
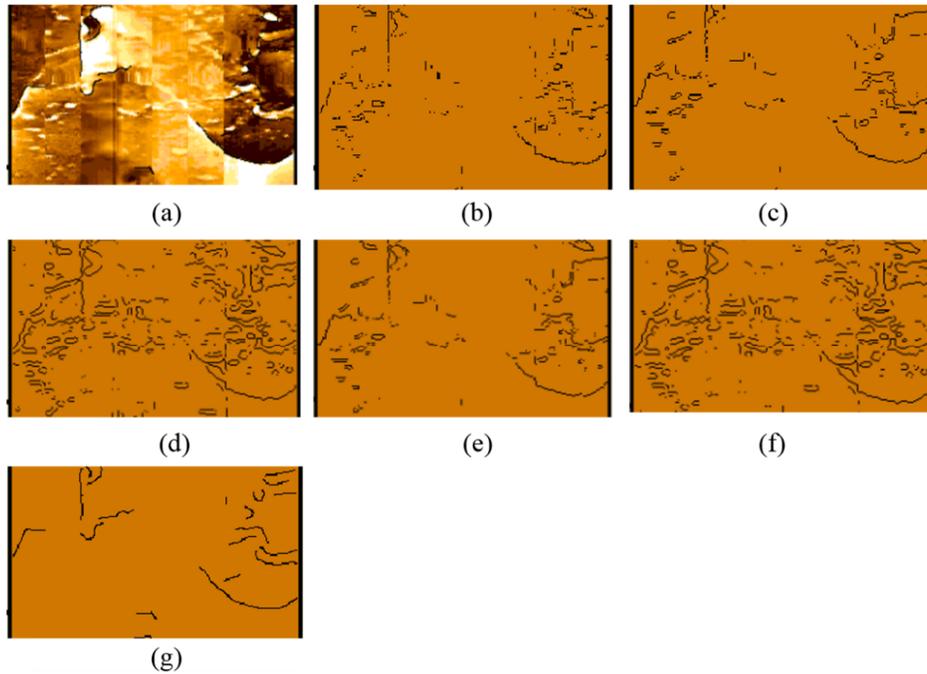
Figure 14. Comparison of the results of edge detection filters on closed fracture of well A indicated on the FMI image. The original image is shown in part (a), and the images obtained as a result of applying Roberts, Prewitt, zero cross, Sobel, LOG and Canny filters have been presented in parts (b), (c), (d), (e), (f) and (g), respectively.
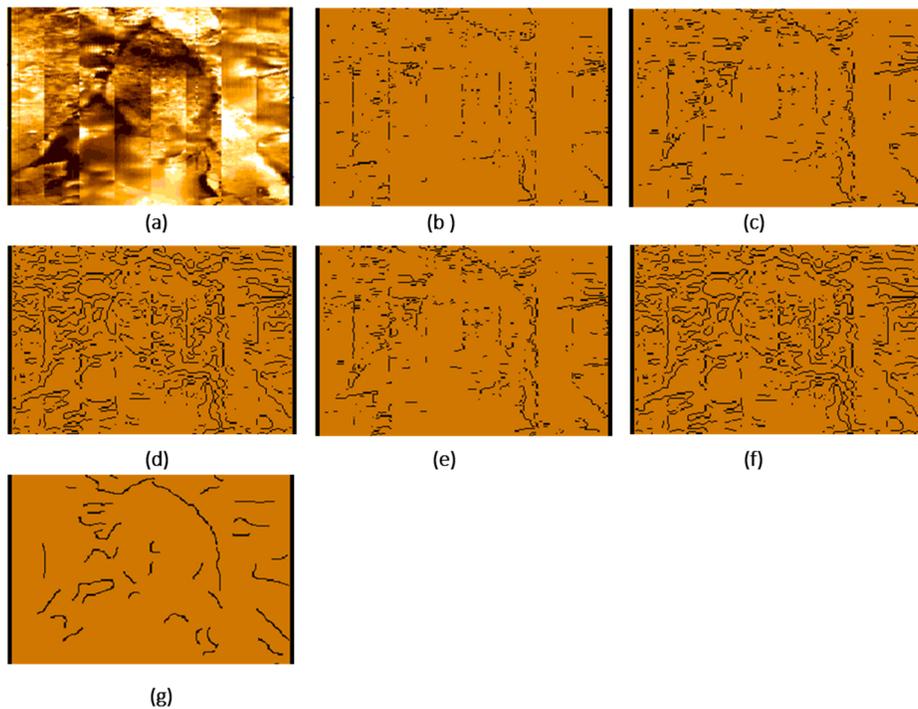
Figure 15. Comparison of the results of edge detection filters on the existing fault of the FMI image from well B. The original image is shown in part (a), and the images obtained as a result of applying Roberts, Prewitt, zero-cross, Sobel, LOG and Canny filters have been presented in parts (b), (c), (d), (e), (f) and (g), respectively.
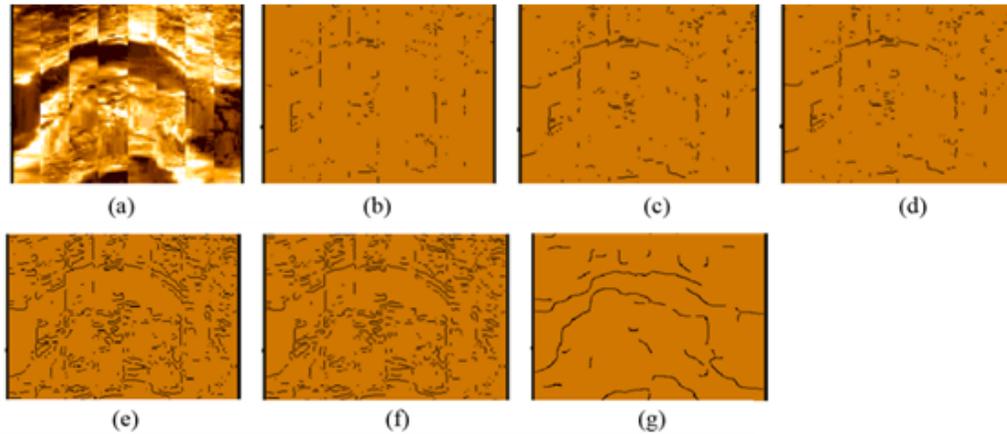
Figure 16. Comparison of the results of edge detection filters on open fracture (Major open fracture) of well A indicated on the FMI image. a) Original image in GEOLOG software, b) Apply the Roberts algorithm, c) Apply the Sobel algorithm, d) Apply the Prewitt algorithm, e) Apply the Zero-Cross algorithm, f) Apply the LOG algorithm and g) Apply the Canny algorithm in MATLAB.

Original image in GEOLOG software   Th=[0.1 0.6], Sigma=sqrt(16)   Th=[0.1 0.6], Sigma=sqrt(18)

Th=[0.1 0.6], Sigma=sqrt(20)   Th=[0.1 0.6], Sigma=sqrt(22)   Th=[0.1 0.6], Sigma=sqrt(24)
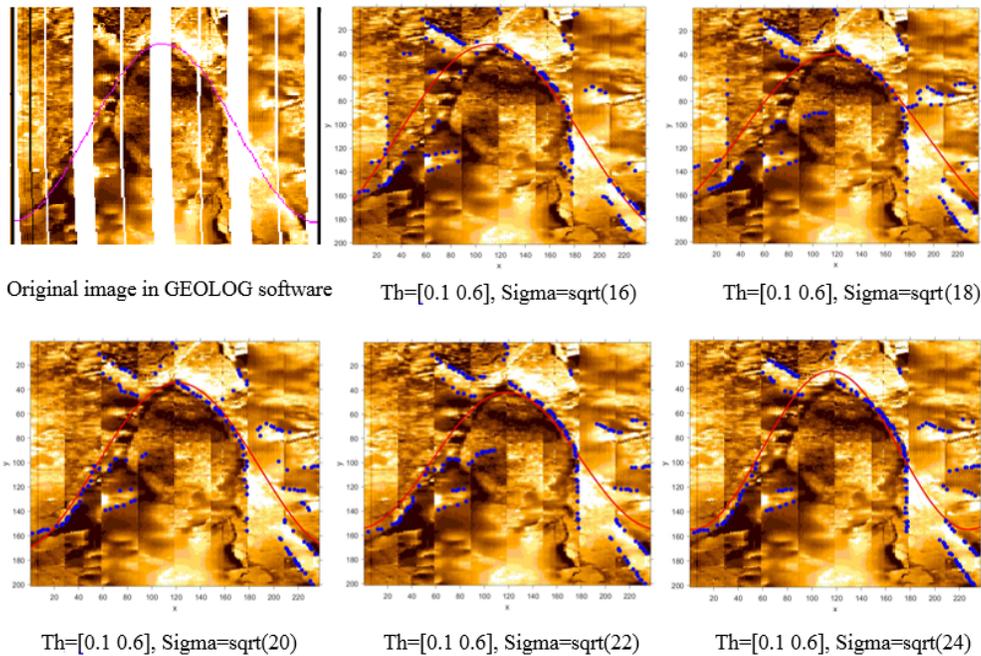
Figure 17. Display the results obtained with thresholding changes in the Hough transform algorithm in the observed fault.

9. Conclusions

In this paper, first GEOLOG software has been used to identify and interpret fractures, including the number and type of fractures, their density, dip and azimuth. In the next step, fractures have been analyzed using a variety of edge detection algorithms. In this paper, we have reviewed and applied the edge detection methods such as Canny, Prewitt, Roberts, LOG, Zero-Cross and Sobel edge detection algorithms. Comparison of the results of applying these edge detection algorithms showed

better performance of Canny algorithm compared to the other edge detection algorithms in identifying sine curves. The Canny method had a better effect than Prewitt, Roberts and Sobel filters. The Roberts, Prewitt and Sobel methods had results with very weak and discontinuous edges and even false edges whereas the Canny filter gave clean, almost continuous and true edges. Thus, the Canny method acted better in addition to that the continuity of edges in this method was strong. As a general conclusion, we can say that using edge detection algorithms, the accuracy of identifying

image features will be improved and finally, choosing the right edge detection algorithm to use the Hough transform algorithm can provide better results. It is expected to have future research directions or areas for improvement, such as exploring advanced edge detection techniques or incorporating machine learning algorithms for fracture detection.

References

[1]. Roehl, P. O., & Choquette, P. W. (Eds.). (2012). *Carbonate petroleum reservoirs*. Springer Science & Business Media.

[2]. McQuillan, H. (1985). Fracture-controlled production from the Oligo-Miocene Asmari Formation in Gachsaran and Bibi Hakimeh fields, southwest Iran. In *Carbonate petroleum reservoirs* (pp. 511-523). New York, NY: Springer New York.

[3]. Martinez, L. P., Hughes, R. G., & Wiggins, M. L. (2002). Identification and characterization of naturally fractured reservoirs using conventional well logs. *The University of Oklahoma*, *23*.

[4]. Zohreh, M., Junin, R., Bakhtiary, H. A., Poor, S. T., Mohamadian, R., & Movahed, A. A. (2016). The evaluation of borehole imaging result comparing with cores in Sarvak fractured and non-fractured reservoir. *Arabian Journal of Geosciences*, *9*, 1-12.

[5]. Khoshbakht, F., Azizzadeh, M., Memarian, H., Nourozi, G. H., & Moallemi, S. A. (2012). Comparison of electrical image log with core in a fractured carbonate reservoir. *Journal of Petroleum Science and Engineering*, *86*, 289-296.

[6]. Kherroubi, J. (2008, December). Automatic extraction of natural fracture traces from borehole images. In *2008 19th International Conference on Pattern Recognition* (pp. 1-4). IEEE.

[7]. Movahed, Z., Junin, R., Amiri Bakhtiari, H., Safarkhanlou, Z., Movahed, A. A., & Alizadeh, M. (2015). Introduction of sealing fault in Asmari reservoir by using FMI and RFT in one of the Iranian naturally fractured oil fields. *Arabian Journal of Geosciences*, *8*, 10919-10936.

[8]. Schlumberger (1994) FMI fullbore formation microimager. Schlumberger educational services.

[9]. Rider, M. (1996). The geological interpretation of well logs (Second edi). *Caithness: Whittles Publishing*.

[10]. Serra O (1989) Formation micro scanner image interpretation. Schlumberger education services.

[11]. Alavi, M. (1994). Tectonics of the Zagros orogenic belt of Iran: new data and interpretations. *Tectonophysics*, *229*(3-4), 211-238.

[12]. Alavi, M. (2004). Regional stratigraphy of the Zagros fold-thrust belt of Iran and its proforeland evolution. *American journal of Science*, *304*(1), 1-20.

[13]. Mottie H, (2003) Geology of Iran (Zagros stratigraphy). Geological Survey of Iran.

[14]. Mehrabi, H., Rahimpour-Bonab, H., Enayati-Bidgoli, A. H., & Navidtalab, A. (2014). Depositional environment and sequence stratigraphy of the Upper Cretaceous Ilam Formation in central and southern parts of the Dezful Embayment, SW Iran. *Carbonates and Evaporites*, *29*, 263-278.

[15]. Esrafili-Dizaji, B., & Rahimpour-Bonab, H. (2013). A review of permo-triassic reservoir rocks in the zagros area, sw iran: influence of the qatar-fars arch. *Journal of Petroleum Geology*, *36*(3), 257-279.

[16]. Aadnoy, B. S. (1990). Inversion technique to determine the in-situ stress field from fracturing data. *Journal of Petroleum Science and Engineering*, *4*(2), 127-141.

[17]. Khoshbakht, F., Memarian, H., & Mohammadnia, M. (2009). Comparison of Asmari, Pabdeh and Gurpi formation's fractures, derived from image log. *Journal of Petroleum science and Engineering*, *67*(1-2), 65-74.

[18]. Schlumberger (2002) Borehole geology, geomechanics and 3D reservoir modeling (FMI). SMP-5822.

[19]. Fossen, H. (2010). Structural Geology, Cambridge University press, New York, first edition.

[20]. Gonzalez, R., woods, R., Steven, L. (2004) Eddins, Digital image processing using MATLAB, Pearson Prentice Hall.

[21]. Al-Amri, S. S., & Kalyankar, N. V. (2010). Image segmentation by using threshold techniques. *arXiv preprint arXiv:1005.4020*.

[22]. Li J (2003) A wavelet approach to edge detection (Doctoral dissertation, Sam Houston State University).

[23]. Bhadauria, H. S., Singh, A., & Kumar, A. (2013). Comparison between various edge detection methods on satellite image. *International Journal of Emerging Technology and Advanced Engineering*, *3*(6), 324-328.

[24]. Gonzalez, R. C., & Woods, R. E. (1992). Digit Image Processing. *ion, Bost on, MA: Addison-Wesley Longman P ublishing Company*.

[25]. Muthukrishnan, R., & Radha, M. (2011). Edge detection techniques for image segmentation. *International Journal of Computer Science & Information Technology*, *3*(6), 259.

[26]. Mathur, D., & Mathur, D. P. (2016). Edge Detection Techniques In Image Processing With Elaborative Approach Towards Canny. *Computer Science Department, Lachoo Memorial College Of Science & Technology*.

[27]. Sobel, I. E. (1970). *Camera models and machine perception*. stanford university.

[28]. Liu, X., Duan, Z., Wang, X., & Xu, W. (2016). An image edge detection algorithm based on improved wavelet transform. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *9*(4), 435-442.

[29]. Maini, R., & Aggarwal, H. (2009). Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, *3*(1), 1-11.

[30]. Haralick, R. M. (1987). Digital step edges from zero crossing of second directional derivatives. In *Readings in computer vision* (pp. 216-226). Morgan Kaufmann.

[31]. Pratt, W.K. (1991). Digital Image Processing, 2nd Ed. New York: John Wiley& Sons.

[32]. Bland, J. M., & Altman, D. G. (1996). Statistics notes: measurement error. *Bmj*, *312*(7047), 1654.

[33]. Huang, Y., & Wang, S. (2008, May). Multilevel thresholding methods for image segmentation with Otsu based on QPSO. In *2008 Congress on Image and Signal Processing* (Vol. 3, pp. 701-705). IEEE..

[34]. da Fontoura Costa, L., Ben-Tzvi, D., & Sandler, M. (1990, March). Performance improvements to the Hough transform. In *UK IT 1990 Conference* (pp. 98-103). IET.

[35]. Assous, S., Elkington, P., Clark, S., & Whetton, J. (2014). Automated detection of planar geologic features in borehole images. *Geophysics*, *79*(1), D11-D19.

[36]. Shafiabadi, M., Kamkar-Rouhani, A., Riabi, S. R. G., Kahoo, A. R., & Tokhmechi, B. (2021).

Identification of reservoir fractures on FMI image logs using Canny and Sobel edge detection algorithms. *Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles*, *76*, 10.

[37]. Shafiabadi, M., Kamkar-Rouhani, A., & Sajadi, S. M. (2021). Identification of the fractures of carbonate reservoirs and determination of their dips from FMI image logs using Hough transform algorithm. *Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles*, *76*, 37.

[38]. Wang, W., & Wang, X. N. (2010, May). Micro rock fracture image acquisition and processing. In *2010 2nd International Workshop on Intelligent Systems and Applications* (pp. 1-4). IEEE.

[39]. He, C., & Wang, W. (2010, June). A PCNN-based edge detection algorithm for rock fracture images. In *2010 Symposium on Photonics and Optoelectronics* (pp. 1-4). IEEE.

[40]. Seifallahi, M., Tokhmechi, B., Soleimani, A., & Fard, A. A. (2013). A novel methodology for fracture extraction from borehole image logs. In *The First International Conference Oil, Gas, Petrochemical And Power Plant, Tehran: SID* (Vol. 7).

[41]. Wang, W., Liao, H., & Huang, Y. (2007, August). Rock fracture tracing based on image processing and SVM. In *Third International Conference on Natural Computation (ICNC 2007)* (Vol. 1, pp. 632-635). IEEE.

[42]. Zhang, W., Wu, T., Li, Z., Li, Y., Qiu, A., & Shi, Y. (2021). Automatic detection of fractures based on optimal path search in well logging images. *Journal of Sensors*, *2021*(1), 5577084.

Appendix

```
clear ; clc ; close all
%%    read image --- crop image --- user check
[I_base M] = imread('GS_245.tiff');
First_X_Pix = 381 ;                          % --- user check
Sec_X_Pix = 95931 ;                          % --- user check
First_Y_Pix = 118 ;                          % --- user check
Sec_Y_Pix = 354 ;                            % --- user check
I_FMI = I_base(First_X_Pix:Sec_X_Pix , First_Y_Pix:Sec_Y_Pix , :) ;
% imshow(I_FMI , rgb2gray(M)) ;


%%% Put Initial Value
TotalPix = Sec_X_Pix-First_X_Pix ;
Num_Sec = 100 ;                              % --- user choice
Th = [0.325 0.4] ; % Threshold                % --- user choice
Sig = sqrt(10) ; % Standard deviation of the filter % --- user choice
StepPix = ceil(TotalPix / Num_Sec) ;
FE = [] ;
FESG = [] ;


%%% Loop
for m = 1 : Num_Sec
   if m == Num_Sec
      Stepl = ((m-1)*StepPix)+(m) ;
      Steph = length(I_FMI) ;
   else
      Stepl = ((m-1)*StepPix)+(m) ;
      Steph = Stepl + StepPix ;
   end


   %%%   Denoise with Non Local Filter and Gaussian kernel
   I_FMI_sec = I_FMI(Stepl:Steph , : , :) ;
   Coef = 0.000005 ; % user ------- check 0.001  or 0.000005
   DoS = Coef*diff(getrangefromclass(I_FMI_sec)).^2 ; % degreeOfSmoothing
   Denoise_sec = imnlmfilt(I_FMI_sec,'DegreeOfSmoothing',DoS) ; % Non-local filtering of images with Gaussian
kernels
%    subplot(141) ; imshow(I_FMI_sec,rgb2gray(M)) ;
%    subplot(142) ; imshow(Denoise_sec,rgb2gray(M))


   %%%   Do canny algorithm
   G = ind2gray(Denoise_sec,M) ; % make grayscale for prepering to canny algorithm
   C_filt = ~edge(G,'canny',Th,Sig); %% ------------ Do canny algorithm
   FMI_Edge = uint8(C_filt) .* Denoise_sec ; % merge filter and image
   FE = [FE ; FMI_Edge] ;
%    subplot(143) ; imshow(FMI_Edge,rgb2gray(M)) ;


   %%%    Only Edge
   FMI_Edge_SemiGeolog = uint8(C_filt) .* uint8(ones(size(C_filt,1),size(C_filt,2))) ;
   FESG = [FESG ; FMI_Edge_SemiGeolog] ;
%    subplot(144) ; imshow(FMI_Edge_SemiGeolog,rgb2gray(M)) ;
```

```
    clc
    disp(['Section: ' num2str(m) 'done.'])
end


%%  Export
FEe = I_base ;
FEe(First_X_Pix:Sec_X_Pix , First_Y_Pix:Sec_Y_Pix , :) = FE ;
imwrite(FEe,rgb2gray(M),'GS_245_Edge.tiff') ;
% dos('GS_245_Edge.tiff') ;
FEsg = I_base ;
FEsg(First_X_Pix:Sec_X_Pix , First_Y_Pix:Sec_Y_Pix , :) = FESG ;
imwrite(FEsg,rgb2gray(M),'GS_245_SG.tiff') ;
% dos('GS_245_SG.tiff') ;
```

مقایسه الگوریتم‌های لبه‌یابی جهت شناسایی خودکار شکستگی در مخازن هیدروکربنی با استفاده از نگارهای تصویری

مینا شفیع آبادی* و ابوالقاسم کامکار روحانی

دانشکده معدن، نفت و ژئوفیزیک، دانشگاه صنعتی شاهرود، ایران

* نویسنده مسئول مکاتبات: mina.shafiabadii@gmail.com

چکیده:

با توجه به تاثیر شکستگی‌ها در افزایش بازیافت هیدروکربن، مطالعه شکستگی سنگ مخزن از اهمیت ویژه‌ای برخوردار است. شکستگی یکی از مهمترین مسیرهای جریان سیال در مخازن کربناته می‌باشد. نگارهای تصویری توانایی تشخیص شکستگی‌ها و سایر ویژگی‌های زمین‌شناسی و لایه‌های مخزن را فراهم می‌کنند. در این مطالعه از دو رویکرد برای تشخیص شکستگی‌ها با استفاده از نگار تصویری FMI در دو چاه A و B واقع در یکی از میادین نفتی در جنوب غرب ایران استفاده شد. در مرحله اول، تصحیح و پردازش داده‌های خام FMI برای شناسایی تعداد و موقعیت شکستگی‌ها و همچنین شیب، گسترش، طبقه‌بندی و تراکم شکستگی‌ها انجام شد. در مرحله دوم، با توجه به اینکه شکستگی‌ها دارای لبه‌هایی در تصاویر FMI می‌باشند، فیلترهای مختلف تشخیص لبه مانند Prewitt، Canny، Roberts، LOG، Zero-cross و Sobel بر روی داده‌های تصویری اعمال شد، سپس به مقایسه عملکرد آنها برای شناسایی شکستگی‌ها پرداخته شد. در نهایت شناسایی خودکار شکستگی‌ها با اعمال الگوریتم تبدیل هاف انجام شد و نتایج نشان داد که الگوریتم Canny بهترین گزینه برای انجام الگوریتم تبدیل هاف می‌باشد. مقایسه کارایی فیلترهای لبه‌یابی فوق‌الذکر برای شناسایی شکستگی‌ها و مهمتر از آن شناسایی خودکار شکستگی‌ها با استفاده از الگوریتم تبدیل هاف را می توان از نوآوری این کار پژوهشی دانست.

کلمات کلیدی: شکستگی، نگار تصویری FMI، فیلترهای تشخیص لبه، الگوریتم تبدیل هاف.