IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

OIL SPILLS DETECTION ON SEA SURFACE USING IMAGE PROCESSING

[1]S. Rama Devi, [2]S. Sai Harshitha, [3]S. Hemalatha, [4]K. Manasa Veena, [5]B. Jaya, [6]K. Priyankitha

[1]Project Guide, [2,3,4,5,6]Students
Electronics and Communication Engineering (ECE)
Andhra University College of Engineering for Women (AUCEW), Visakhapatnam, Andhra Pradesh, India

Abstract: Marine pollution is increasing at an alarming rate and causing detrimental effects on the health of underwater flora and fauna. Anthropogenic activities are contaminating the natural water bodies which have adverse effects on humans as well as the environment. One of the main reasons for marine pollution are oil spills.

An oil spill is the release of liquid petroleum hydrocarbons into the environment, especially the marine ecosystem. These oil spills are caused by accidents involving tankers, leaks from ships and natural disasters such as hurricanes, storm surge or high winds. The major threat to oceanic and costal environments is caused by the oil spills. Thus, oil spill detection is an important task that needs to be tackled.

The Synthetic Aperture Radar (SAR) is an effective tool in checking the oil spills over a wide area and is capable of monitoring the ocean surface day and night in all-weather conditions. However, a visual check performed by human operators on a great number of images would be too expensive and time taking process. Therefore, a tool is required to detect the oil spills automatically. This project describes a software application using MATLAB for oil spills detection on the sea surface using SAR images with the help of three different algorithms namely CFAR detector, Fuzzy C means clustering algorithm and Canny edge detection method and compares their performance.

IndexTerms - SAR, CFAR Detection, thresholding, Sobel operator, Fuzzy C-Means clustering, Speckle noise, Canny edge detector, MATLAB.

I. INTRODUCTION

Oils are chemical substances that are in a liquid state at room temperature and they are purely insoluble in water. There are many kinds of oils, but for our purpose, we are only concerned with the oil we commonly use as fuel, known as Petroleum or crude oil. Oil pollution has a wide variety of adverse effects on the environment, wildlife, and even humans too.
However, oceanic oil spills became a major environmental threat due to the liquid form of petroleum leaks into an ecosystem made by anthropogenic (human) causes. Mostly oil pollution occurs in the form of oil spills from the large oceans which results in a huge amount of crude oil into marine and coastal areas, Therefore the main cause of marine pollution is due to "OIL SPILLS".
Nowadays, Oil spills can affect our world at many different levels. So, in order to fully grasp the threat of oil spills which are severely ruining today's environment. Then it's important to understand what oil spills are. An oil spill refers to any uncontrolled release of crude oil and other gases released into environment, especially the marine ecosystem due to human actions and factories activity and is form of pollution.

A) How and where do the oil spills happen?
Generally, Oil spills occur due to various reasons. An oil spill is the accidental release of liquid petroleum hydrocarbons like crude oil from ships, offshore sea platforms, drilling rigs and wells into the environment during exploration activities. And most often is caused by accidents involving undersea petroleum pipelines, leaks at wells & deliberate acts by terrorists.

B) How do Oil spills harm or kill Ocean life?
Whenever oil accidentally spills into the ocean, it might cause the biggest problems or disasters. Eventually, Oil spills can harm sea creatures and make seafood unsafe to eat. In this way, oil spills can affect the physical, chemical & biological characteristics of both water and land and affect the survival of aquatic plants and animals. Oil spills can damage the marine food chain and local fishing business.

C) Synthetic Aperture Radar (SAR)

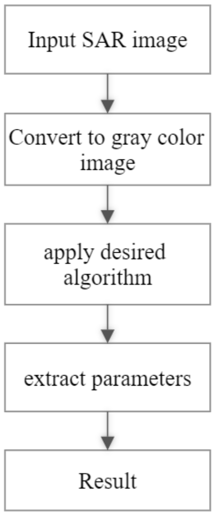
SAR (or) synthetic aperture radar is utilized for producing 2D images or 3D models of landscapes or objects. SAR is a side looking radar. It first transmits microwave signals and then receives back the signals that are backscattered from the earth's surface. These signals are processed and allows to create a higher resolution pixels. And the oil films decrease the backscattering of oceanic areas which results in a dark formation on SAR images. The size of the synthetic antenna aperture is determined by the SAR device's movement over the target scene while it is being illuminated.

D) WHY SAR?

- SAR has the ability to monitor the ocean from a broad range and under any meteorological setting, regardless of the time of day.
- It offers images with a very high level of detail regardless of any type of weather condition.
- SAR possesses diverse capacities such as the ability to acquire data from multiple perspectives, to gather various data sets, and provide images of different resolutions.

II. PROPOSED METHOD

SAR images contain speckle noise. In order to eliminate the noise and to identify the oil spill, different image processing algorithms like cfar detector, canny edge detection, fuzzy c means clustering are used.



III. METHODOLOGY

A) FUZZY C MEANS CLUSTERING

Fuzzy C-means clustering algorithm is an autonomous learning method. In this algorithm, if the probability of detecting can only be 1 or 0, then it is called **hard clustering**. The boundary to a cluster in hard clustering is visualized as a crisp boundary. But in **soft clustering** method, the probability of detecting can be taken any value between 1 and 0, such as 75%, for which boundaries can be visualized as a fuzzy boundary. Fuzzy is a **soft clustering** method. “C” is a hyper parameter which specifies the number of cluster centers.

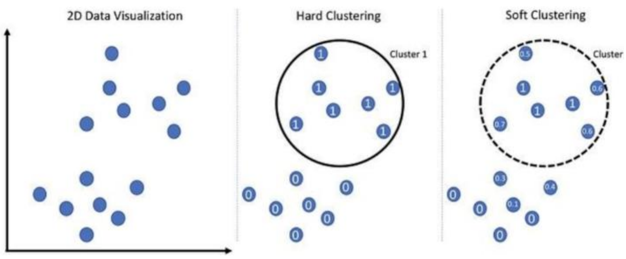


Figure 1: Clustering

The number inside each dot is the probability of that datapoint belonging to Cluster 1, and the black circle represents the cluster boundary of Cluster 1.

Steps in Fuzzy C Means Algorithm

1) It starts with random initial guess of ‘c’ cluster centres.
2) Each data point is assigned with a random membership grade for each cluster. Now calculate the fuzzy membership 'μ_{ij}' *using*

$$\mu_{ij} = 1 / \sum_{k=1}^{c} (d_{ij} / d_{ik})^{(2/f-1)} \qquad (1)$$

where

'μij' is the membership of i^{th} data to j^{th} cluster centre.

'f' is the fuzziness index f € [1, ∞].

'c' is total number of cluster centers.

'dij' is the Euclidean distance between i^{th} data to the j^{th} cluster center.

3) Compute the fuzzy centers 'v_j' using:

$$v_j = \left(\sum_{i=1}^{n} (\mu_{ij})^f \; x_i\right) / \left(\sum_{i=1}^{n} (\mu_{ij})^f\right), \forall j=1,2,\ldots c \qquad (2)$$

where
'vj' is the j^{th} cluster center.
'n' is the total number of data points.
'f' is the fuzziness index f € [1, ∞].
'c' is the total number of cluster centers.
'µij' is the membership of i^{th} data to j^{th} cluster center.

4) By repetitively updating the cluster centers and the membership grades for each datapoints, the cluster centres are moved to the correct location within a dataset and for each data point, degree of membership in each cluster is calculated.
5) This iteration decreases the value of an objective function that indicates the distance of any given datapoint to a cluster center weighted by the membership of that datapoint in the cluster.

The main goal of fuzzy c-means algorithm is to minimize the following objective function:

$$J(U,V) = \sum_{i=1}^{n} \sum_{j=1}^{c} (\mu_{ij})^f \| x_i - v_j \|^2 \qquad (3)$$

where ,
'$\|x_i - v_j\|$' is the Euclidean distance between i^{th} data and j^{th} cluster center.

6) Repeat the steps 2 and 3 until minimum 'J' value is achieved or

$$\|U^{(k+1)} - U^{(k)}\| < \beta. \qquad (4)$$

where,
'k' is the iteration step.
'β' is the termination criterion between [0, 1].
'$U = (\mu_{ij})_{n*c}$' is the fuzzy membership matrix.
'J' is the objective function.

B) CANNY EDGE DETECTION

The Canny edge detector is a multi-step algorithm to detect the edges for any image given as input. It involves the below steps to detect edges of an image. It was developed by John F. Canny in 1986.

1) removing of noise in given image using Gaussian filter.
2) Obtaining the derivative of Gaussian filter to calculate the gradient of image pixels to gain magnitude along x and y dimension.
3) Considering a group of neighbours for any curves in a direction vertical to the given edge in the image, suppress the non-max edge contributor pixel points.
4) By using the Hysteresis Thresholding to save the points greater than gradient magnitude and avoid ones lower than the low threshold value.

i. Noise Removal or Image Smoothing

In the noise presence, the pixel may not be close to being same as its neighboring pixels. This might affect detection of the edges in the given image. To avoid the same, we use the Gaussian filter, which is convolved with the image and removes the noise in output images.
Here we're convolving gaussian filter or kernel g(x, y) with an input image I. We have to ensure that given point must be similar to its neighbor points in output and so we take matrix (1 1 1) to maintain the similarity between pixels and reduce noise.

$$S = I * g(x,y) = g(x,y) * I \qquad (5)$$

$$g(x,y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2 - y^2}{2\sigma^2}}$$

ii. Derivative

Find the derivative of filter w.r.t X and Y dimensions and scroll it with I, input image to attain gradient magnitude along with the dimensions. Also, the orientation of the image can be found using the tangent of the angle between the above dimensions.

$$\nabla S = \nabla(g * I) = \nabla(g) * I \qquad (6)$$

Where, $\nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$

$$\nabla S = \begin{bmatrix} g_x \\ g_y \end{bmatrix} * I = \begin{bmatrix} g_x * I \\ g_y * I \end{bmatrix} \qquad (7)$$

(S_x , S_y) Gradient Vector

$$\text{magnitude} = \sqrt{S_x{}^2 + S_y{}^2} \qquad (8)$$

direction= $\theta = \tan^{-1}\frac{S_y}{S_x}$ (9)

iii. Non-maximum suppression

The given image is processed along the image gradient side, and if image points are not part of local maxima, then they can be zero. So, this has the advantage of suppressing the image information which is not the part of local maximum.

From the below, that has three edge points. Assume point (x, y) as the point having the highest gradient of edge. Check for the edge points in the direction vertical to the edge and validate if their gradient is lower than (x, y). If the values are less than (x,y) gradient, we can decrease those non-maxima points along the curve.

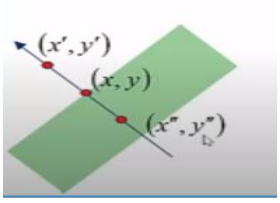


Figure 2: non maximum suppression

$$M(x,y) = \begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y) > |\Delta S|(x',y') \ \&\ |\Delta S|(x,y) > |\Delta S|(x'',y'') \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

iv. Hysterisis Thresholding

By tracing the edges, it eliminates weak edges and holds strong edges. If the gradient magnitude is high, then it will be considered as a edge pixel and it will be accepted. If the gradient magnitude is low, it will be considered as a non-edge pixel and it will be rejected. If the gradient magnitude lies between the two values, then they are accepted if they are connected to points which exhibit strong response.

C) CONSTANT FALSE ALARM RATE (CFAR) DETECTOR

CFAR (constant false alarm rate) is a signal processing technique which is generally used to detect targets even in presence of noise. For any detection we use threshold value, signal which is below the threshold value is rejected whereas the signal above the target is detected.

In CFAR, a sliding window is used to process the signal. The window size determines the size of region over which CFAR algorithm estimates the background noise level. The window size should be large enough to estimate background noise accurately but not so large, which can lead to false alarms.

In cfar, the guard cells are used to avoid false alarms from clutter and other non-target objects. The guard size refers to width of the guard cells. The choice of guard size in cfar is tradeoff between detection sensitivity and false alarm rate. A larger guard size reduces false alarms but also reduces detection sensitivity, whereas lower guard size improve detection sensitivity, but increases false alarms.

In cfar, the cell (CUT) is tested for the presence of target by comparing signal level against noise estimate threshold.

The offset size is used to adjust threshold value, so that probability of false alarms remains constant across the range cells. The offset value is typically chosen based on noise statics and desired false alarm rate.

In CFAR, when detection is performed on CUT, noise power is estimated from neighbor cells. Then detection threshold T, is given by

$$T = \alpha \cdot P_n \quad (11)$$

where,

Pn is the noise power and is the scaling factor, called the threshold factor.
Process:
The cells on both sides of CELL UNDER TEST are ‘guard cells.’ ‘The training cells’ output are added and then multiplied by constant to establish a threshold value. Detection occurs when CUT output exceeds threshold value.

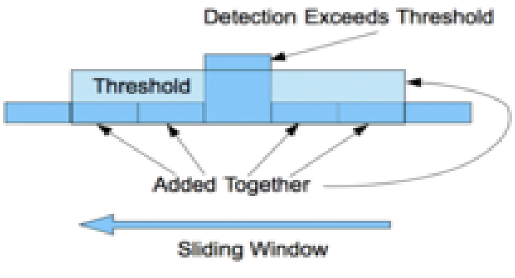


Figure 3: CFAR Detector

In CA - CFAR detection, threshold level is calculated by estimating noise level around CELL UNDER TEST. This can be set up by taking a block of cells around the CUT (guard cells - are ignored) and calculating the average power level.

A target is detected if the power in CELL UNDER TEST is larger than all its adjacent cells and larger than local average power. The CA-CFAR detector is most probably used detector. In a CA - CFAR detector, noise samples are extracted from training cells around CELL UNDER TEST. The noise estimate computed as

$$P_n = \frac{1}{N} \cdot \sum_{m=1}^{n} x_m \qquad (12)$$

where N is number of training cells and Xm is sample in each training cell.

$$\alpha = N \cdot (P_{fa}^{\frac{1}{N}} - 1) \qquad (13)$$

where, Pfa is the desired false alarm probability.

IV. RESULTS AND DISCUSSION

MATLAB software is used to implement the three proposed algorithms on SAR images. Figure 4(a) is the input SAR image containing oil spill. Figure 4(b) is the gray color image of SAR image. Figure 4(c) is the output image after applying CFAR detector. Figure 4(d) is the output image after applying fuzzy c means clustering. Figure 4(e) is the output image after applying canny edge detection which gives the edge of the oil spill that separates it with background clutter. Figure 4(f) is the output image using canny edge detection with hysteresis.

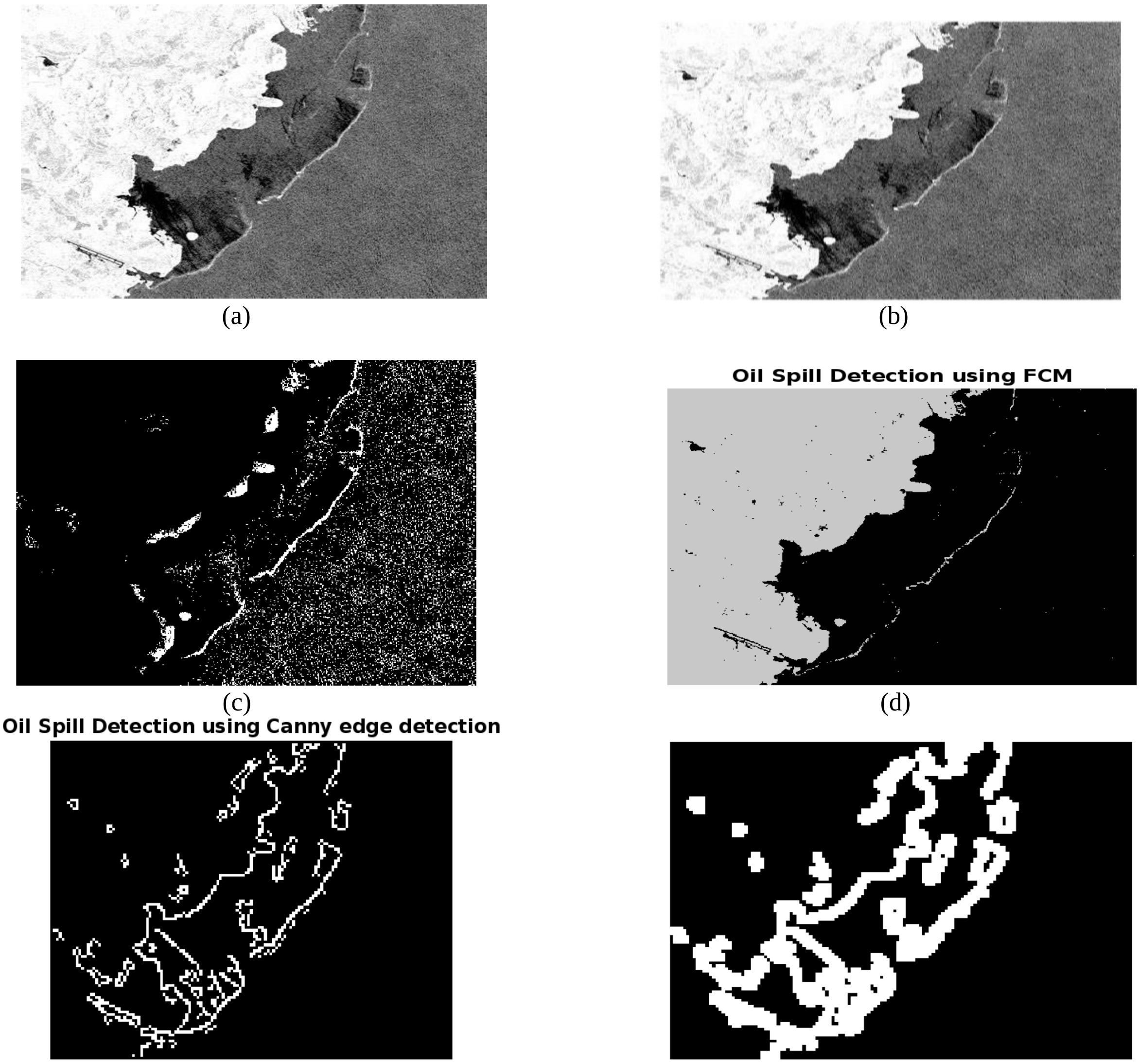


Figure 4: (a) Input SAR image; (b) Gray color image; (c) CFAR detector output; (d) FCM output; (e) canny edge detector output; (f) canny edge detector with hysteresis output.

Figure 5 shows a popup window that appears when oil spill is identified in the input image.

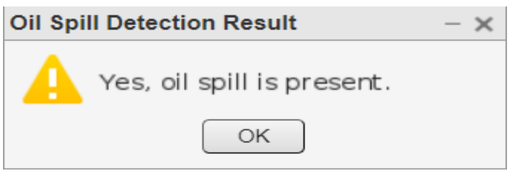


Figure 5: popup window

Table 1 shows the comparison between extracted parameters for the three proposed algorithms.

	Oil spill Percentage (%)	Oil spill Area (pixels)	Elapsed time (seconds)
CFAR	7.8117	13552	18.8994
FCM	40.7536	70701	9.8049
Canny	18.5569	729.75	7.5675

Table 1: comparison table

V. CONCLUSION

SAR images are one of the best medium for oil spill detection as SAR is capable of monitoring the ocean surface day and night and in all weather conditions. The software application described in this paper uses different comparative approaches like CFAR detector, canny edge detection and fuzzy c means clustering to detect the oil spills over the sea surface. oil spill parameters like oil spill percentage, oil spill area, elapsed time are studied. By comparing these parameters, we can conclude that fuzzy c means clustering provides better reliability.

VI. ACKNOWLEDGEMENT

We would like to sincerely thank our Head of the Department, Dr. S. Aruna and Project guide, Mrs. S. Rama Devi for her support. We are pleased to take this opportunity to express our gratitude towards our project mentor, for all her guidance, inspiration, and insightful talks that have helped us finish our project.

Their advice proved to be the most valuable in overcoming all obstacles in the completion of this major project **"OIL SPILLS DETECTION ON SEA SURFACE USING IMAGE PROCESSING"**.

VII. REFERENCES

[1] Chavdar Alexandrov, Nikolay Kolev, Yordan Sivkov, Avgustin Hristov, Oil Spills Detection on Sea Surface by using Sentinel–1 SAR Images, ©2020 IEEE.

[2] Maio, A. D., Ricci, G., & Tesauro, M. (2001). On CFAR detection of oil slicks on the ocean surface by multifrequency and/or multipolarization SAR. In Radar conference, 2001. Proceedings of the 2001 IEEE (pp. 351– 356).

[3] A. Brahma Tejaswini, Dr. Y. S Rajasekhara Reddy, Dr. P. Siddaiah, Implementation of Oil Spill Detection from SAR Image through Improved Canny Edge Detection, ISSN: 2278-0181IJERTV10IS030151(This work is licensed under a Creative Commons Attribution 4.0 International License.) Published by: www.ijert.orgVol. 10 Issue 03, March-2021.

[4] John Canny, "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp 679-698, 1986

[5] YongSang Ryu, YoungSoo Park, JinSoo Kim, and SangHun Lee, Image Edge Detection using Fuzzy C-means and three Directions Image Shift Method,45:1, IJCS_45_1_01,10 February 2018.

[6] M. Migliaccio, A. Gambardella, and M. Tranfaglia, ''SAR polarimetry to observe oil spills,'' IEEE Trans. Geosci. Remote Sens., vol. 45, no. 2, pp. 506–511, Feb. 2007.

[7] Wenshuo Gao, Xiaoguang Zhang, Lei Tang, Huizhong Liu. "An Improved Sobel Edge Detection", Computer Science and Information Technology (ICCSIT) 2010 3rd IEEE International Conference on, China, pp 67-71, July 2010.

[8] S. K. Chaturvedi, S. Banerjee, and S. Lele, "An assessment of oil spill detection using Sentinel 1 SAR-C images," J. Ocean Eng. Sci., Sep. 2019, doi: 10.1016/j.joes.2019.09.004.

[9] C. Brekke and A. H. S. Solberg, "Oil spill detection by satellite remote sensing," Remote Sens. Environ., vol. 95, no. 1, pp. 1–13, 2005, doi: 10.1016/j.rse.2004.11.015

[10] ChuanLong Li, Ying Li, XueRui Wu, "Novel fuzzy c-means segmentation algorithm for image with the spatial neighborhoods", Remote Sensing, Environment and Transportation Engineering (RSETE), 2012 2nd International Conference on, China, pp 1-4, June 2012