BOUNDARY IDENTIFICATION IN A SATELLITE IMAGERY USING CANNY EDGE DETECTION TECHNIQUE

Abhishek Yadav , Piyush Trivedi

Department Of Civil Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh-208016, India
abhisheky21@iitk.ac.in, piyusht21@iitk.ac.in

KEY WORDS: Gaussian Blur, Edge Gradient, Non-Maximum Suppression, Double threshold, Hysteresis, canny edge

ABSTRACT:

Edges in an Image defines the features captured by it. Edges when continued define shape of physical features. Using edge as a parameter we can identify and differentiate between various physical features in an Image. The goal of edge detection is to extract information from a picture about its forms and reflectance or transmittance. Edge identification plays a crucial role in remote sensing. There are various methods of Edge detection namely Sobel Edge Detection, Prewitt edge detection, Kirsh edge detection, Robinson edge detection, Marr-Hildreth edge detection, LoG edge detection and Canny Edge Detection. In this project report, we emphasise on usage of "Canny Edge Detection" technique for boundary identification in a satellite imagery. The report presents insights into series of steps involved in the realization of the technique using MATLAB scripts.

1. INTRODUCTION

At the boundaries of the object, edges can be viewed as the projections of the processes which physical in nature as the changes in the reflectance or the changes in the illumination. These edges provides the information regarding the 3D structure of the world and due to this it becomes very important to detect the edges first. Edge detection is a very fundamental and the basic tool which is used in the recognition of objects, in the tracking of target, compression of data, reconstruction of an image etc. In this paper, the canny edge detection technique is used in finding the wide variety of edges in a satellite image to identify the boundaries of various objects.For the identification of edges, we wrote a script in MATLAB to convert the image into grayscale and then applyied various filters and operations.

2. CANNY EDGE DETECTION METHOD

The canny edge detector is an edge detection operator which uses a multi-stage algorithm to detect a wide range of the edges in an image. This edge detection process was developed by John F Canny in the year 1986. In this technique, useful structural information is extracted from the different vision objects in the image. The general criteria that canny follows include the detection of the edges with a very low error rate i.e. the recognition of edges as many as possible. The second criteria is that the detected edge point must localize on the center of the edge. The third criteria is that an edge should be counted only once to cancel the possibility of the false edges.
There have been various methods developed for the edge detection but canny edge detection algorithm provides a very good and reliable results in compare to the other methods due to the fact that it follows all the three criteria required for the edge detection. The process of the canny edge detection algorithm can be given in terms of various steps.

1. Application of the Gaussian filter for the smoothening of the image and the removal of the noise.
2. Determing the intensity gradients in the image.
3. Applying the Non maximum suppression to suppress those pixel to 0 which are not the edges.
4. Applying the double thresholding to determine the potential edges in the image.
5. Edge tracking by hysteresis i.e. to suppress all the weak edges which are not connected to the strong edges.

3. STEPS INVOLVED IN CANNY EDGE DETECTION ALGORITHM

The canny edge detection process includes a multi step algorithm involving various processes ranging from the grayscale conversion to the edge tracking. The various steps involved are described here.

3.1 Grayscale conversion from RGB imagery

In a grayscale image, every pixel is assigned a value between 0-255 for the representation of the intensity. A grayscale image is an image without the information of colour. Canny edge image is constructed from the grayscale image as this detection is based on grayscale. So we converted the RGB image into the grayscale first. The intensity values in the grayscale image are used in the calculation of the gradient of the image.

3.2 Smoothening of Image by Gaussian filter

Since the edge detection results are highly sensitive to the image noise so it becomes important to remove the image noise. One method to get rid of it is by using the Gaussian blur to smooth it. We chose this size as the kernel size depends on the expected blur. Smaller the kernel, the less visible is the blur. The gaussian filter is mainly used to eliminate the gaussian noise among many filters. It is a linear filter. For applying gaussian filter we used inbuilt function in MATLAB providing σ = 1.7. Following equation defines the kernel of gaussian filter.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

3.3 Determining the Intensity Gradients

The edges in an image are because of the sudden changes in the brightness, assuming that each object consists of pixels that do not greatly vary in their brightness. One way to measure how big the change is to calculate the gradient magnitude at each pixel. The edge in an image may point in any direction, so the algorithm uses the filters to detect the horizontal, vertical and the diagonal edges in the smoothened image. The sobel operator of size 3x3 is passed upon the image and the gradients in X and Y directions are calculated. X and Y denote the horizontal and vertical direction respectively.

3.4 Determining the Edge Gradient Magnitude and Edge Direction

The edge gradient is calculated by taking the root of the summation of the squares of derivatives in X and Y direction. The edge direction angle is given in terms of one of the four angles representing the vertical, horizontal and the two diagonal directions. The range for the angles is given as:

1. If the angle is in between [0 - 22.5] and [157.5 - 180], it will be taken as 0 degree.
2. If the angle is in between [22.5 - 67.5] , it will be taken as 45 degree.
3. If the angle is in between [67.5 - 112.5] , it will be taken as 90 degree.
4. If the angle is in between [112.5 - 157.5.5] , it will be taken as 135 degree.

3.5 Non Maximum Suppression

The image magnitude produced results in the thick edges. The final output image must have thin edges. So we perform the non maximum suppression to thin out the edges. This method refers to the searching for the local maximum in the edge detection process. In this process the algorithm goes to all the points in the intensity gradient matrix and finds the pixels which have the maximum value in the edge directions. We can take an example as the pixel (i,j) is being processed and in the same direction the pixels are (i,j-1) and (i,j+1). If one of those pixels is more intense than the pixel which is being processed then the intense pixel will be kept and the value of (i,j) will be set to 0. And if there is no pixel intense than the processed pixel, then this pixel will be kept. So we can say that each pixel has the two main criteria for non maximum suppression as the edge direction and the pixel intensity. The non maximum suppression steps can be summarised as follows:

1. Creating a matrix having initial value as 0 and of the same size as that of the original intensity gradient matrix.
2. Identifying the edge direction based on the angle values from the angle matrix.
3. To check if the pixel in the same direction has the higher intensity value than the pixel which is currently being processed.
4. Return the image processed with the non maximum suppression.

3.6 Double Thresholding

Double thresholding takes care of the result obtained from the non maximum suppression as image may have some noise and some edges may not be the edges in actual. The double thresholding helps in identifying the three types of pixels given as Strong pixels, Weak pixels and the non-relevant pixels. Strong pixels are those pixels having very high intensity values and they contribute to the final edge. Weak pixels are those pixels which have intensity not enough to be considered as strong pixels. And it is not as weak as of the non relevant pixels. Other pixels except these two kinds are the non-relevant pixels. On the basis of these observations we can say that:

1. High threshold is used to identify the pixels having intensity higher than the high threshold.
2. Low threshold is used to identify the pixels having intensity lower than the low threshold.
3. All those pixels which have intensity between high threshold and low threshold are considered as weak and the hysteresis will help in identifying the pixels as which ones are weak and which ones are non relevant.

3.7 Edge Tracking by Hysteresis

From the above steps it is clear that the strong edge pixels will be included in the final output image because they are extracted from the true edges in the image. The weak edge pixels derived are either from the true edge or from the noise variations. In order to get the correct edges it is important to remove the weak edges obtained from the noises. Generally, a weak pixel which is obtained from the true edge will be connected with a strong edge pixel. In order to retain this pixel, the algorithm detects around the weak pixel if there is any strong edge pixel connected to it.

4. RESULTS AND DISCUSSION

Following are the output of various steps performed in Canny edge detection.

1. Original image was saved using QGIS in .jpg format.

Figure 1: Satellite imagery of an Area in Kanpur region

2. The image was converted in Grayscale format. The digital numbers of the 8 bit image ranged from 0 to 255.

Figure 2: Gray scaled image

3. Edge detection is sensitive to noise. So noise in the image was reduced using Gaussian filter. A sigma of 1.7 was used for smoothening of the image.

Figure 3: Image after gaussian filter smoothening

4. An edge corresponds to intensity change in an image. Sobel operator was applied to get intensity gradient in both horizontal and vertical direction of image.

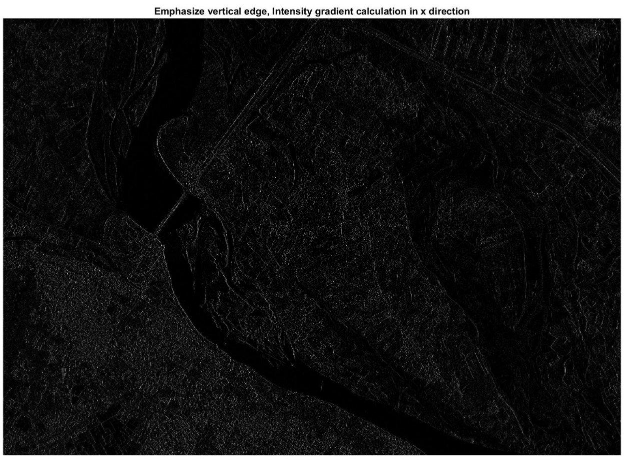

Figure 4: Intensity gradient along Horizontal direction

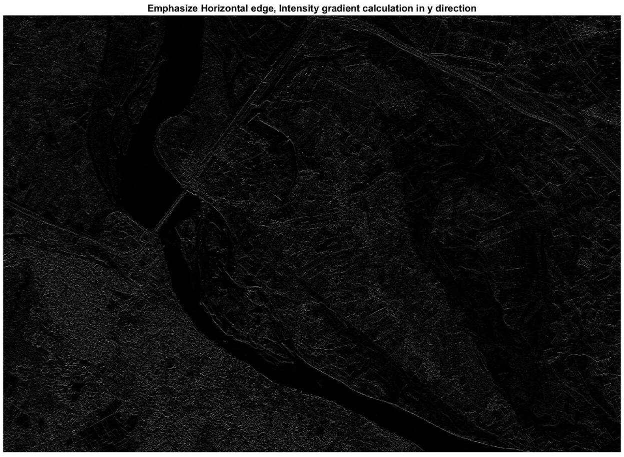

Figure 5: Intensity gradient along Vertical direction

5. Magnitude and angle of intensity gradient was calculated. Magnitude of the intensity resulted in the following image.

Figure 6: Gradient magnitude

6. Non maximum suppression was performed to reduce the thickness of edges. The step aims to keep only the points in a particular direction having maximum intensity. Following is the output of non maximum suppression.

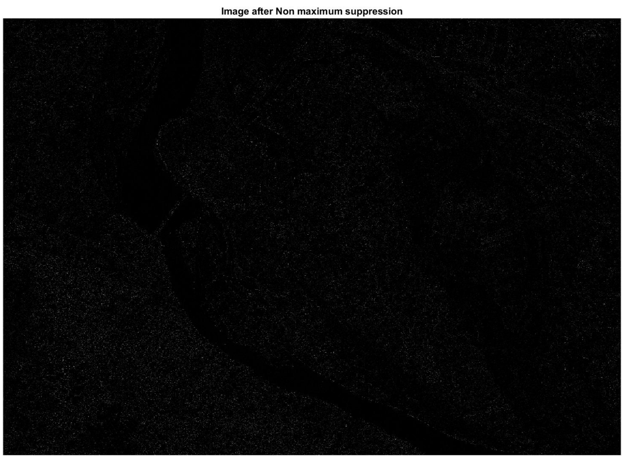

Figure 7: Image after non maximum suppression

7. Double thresholding is performed on the image we get after non maximum suppression. The step classifies the pixels into strong, weak and non-relevant category. Following is the output of double thresholding.

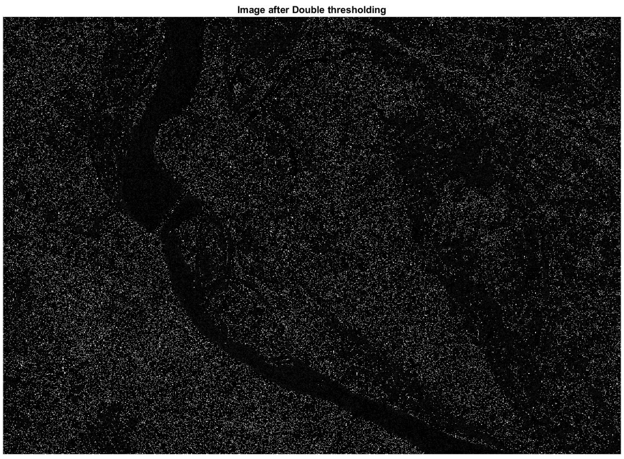

Figure 8: Image after double thresholding

8. The final step is edge tracking by hysteresis. The hysteresis transforms weak pixels into strong ones based on the threshold findings, if and only if at least one of the pixels around the one being processed is a strong one. Following is the output of Canny edge detection.

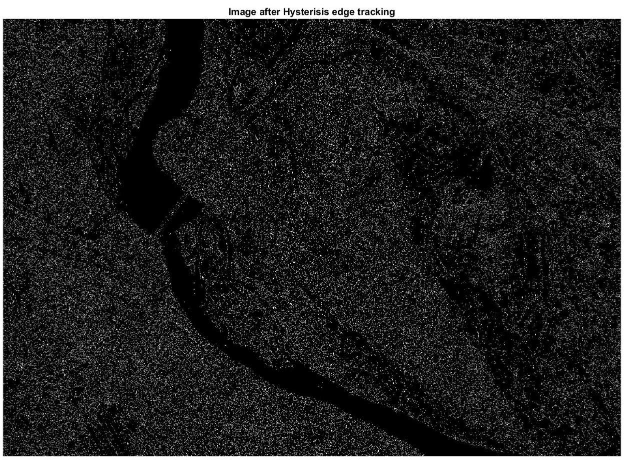

Figure 9: Final image after Hysteresis edge tracking

5. CONCLUSION

This paper provides a very effective method to detect the edges in all kinds of images using the canny edge detection technique. In this paper, we have used a satellite imagery for the identification of boundaries in it and we have obtained very promising results after applying canny edge detection. As it can be seen that the output image has clear separation among the boundaries of various objects. On zooming the output image, the boundaries of the objects are clear. The river path can be observed distinctly in the output image and this data can be used for the various purposes.

6. REFERENCES

William McIlhagga, The Canny Edge Detector Revisited, Int J Comput Vis (2011) 91: 251-261
DOI 10.1007/s11263-010-0392-0.

Edge Detection, (Trucco, Chapter 4 And Jain et al., Chapter 5)

A Review on Edge detection Technique "Canny Edge Detection", International Journal Of Computer applications (0975-8887), Volume 178- No. 10, May 2019

Dr. S. Vijayarani, M. Vinupriya, Performance Analysis of Canny and Sobel Edge detection Algorithms in Image Mining, International journal of innovative research in computer and communication engineering, Vol-1, Issue-8.

Canny Edge detection step by step in Python-Computer Vision, (towardsdatascience.com/canny-edge-detection-step-by-step-computer-vision-b49)

Canny edge Detection, (https://justin-liang.com/tutorials/canny/gradient)

Edge detection of agricultural products based on Morphologically improved canny algorithm,
(https://www.hindavi.com/journals/)

Canny edge dtector, (https://en.m.wikipedia.org/wiki/canny-edge-detector).