Available Online at http://www.recentscientific.com

CODEN: IJRSFP (USA)

International Journal of Recent Scientific Research
Vol. 9, Issue, 5(H), pp. 27036-27038, May, 2018

ISSN: 0976-3031

International Journal of Recent Scientific Research

DOI: 10.24327/IJRSR

Research Article

THE CANNY EDGE DETECTOR

Khanjan K Pandya*

Department of Mathematics, Marwadi University, Rajkot (360009), Gujarat (India)

DOI: http://dx.doi.org/10.24327/ijrsr.2018.0905.2173

ARTICLE INFO

Article History:
Received 10th February, 2018
Received in revised form 6th March, 2018
Accepted 24th April, 2018
Published online 28th May, 2018

Key Words:

Canny edge detection, non maxima suppression, hysteresis, threshold values

ABSTRACT

Edge detection is the approach used most frequently for segmenting images based on abrupt changes in intensity. Edge models are classified according to their intensity profiles. A step edge involves a transition between two intensity levels occurring ideally over the distance of one pixel. Step edges occur, in images generated by a computer for use in areas such as solid modelling and animation. The Canny edge detection is derived using a step-edge model. Digital images have edges that are blurred and noisy, with the degree of blurring determined principally by limitations in the focusing mechanism and the noise level determined principally by the electronic components of the imaging system. In such cases edges are more closely modelled as having intensity ramp. The slope of the ramp is inversely proportional to the degree of blurring in the edge. A third model of an edge is the roof edge, are models of lines through a region, with the base of a roof edge being determined by the thickness and sharpness of the line.

INTRODUCTION

Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors already out at the time he started his work. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge. Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient.

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased. The Gaussian mask used in my implementation is shown below.

Let f(x, y) denote the input image and G(x, y) denote the Gaussian function:

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

We form a smoothed image $\boldsymbol{f_s(x,y)}$, by convolving G and f :

$$\boldsymbol{f_s(x,y) = G(x,y) * f(x,y)}$$

This operation is followed by computing the gradient magnitude and angle as:

Corresponding author:* **Khanjan K Pandya
Department of Mathematics, Marwadi University, Rajkot (360009), Gujarat (India)

$M(x,y) = \sqrt{g_x^2 + g_y^2}$ and $\alpha = tan^{-1}[\frac{g_y}{g_x}]$

Where, $g_x = \frac{\partial f_s}{\partial x}$, $g_y = \frac{\partial f_s}{\partial y}$ and M(x, y) and α(x, y) are arrays of the same size as the image from which they are computed. Because it is generated using the gradient , M(x, y) typically contains wide ridges around local maxima . The next step is to thin those ridges. One approach is to use non maxima suppression. The final operation is to threshold $g_N(x,y)$ to reduce false edge points. If the threshold is too low, there will be some false edges. /if the threshold is too high, then actual edges points will be eliminated. Canny's algorithm attempts to improve on the situation by using hysteresis thresholding which uses two threshold, a low threshold T_L and a high threshold, T_H. $g_{NH}(x,y) = g_N(x,y) \geq T_H$. and $g_{NL}(x,y) = g_N(x,y) \geq T_L$. Where initially both $g_{NH}(x,y)$ an$g_{NL}(x,y)$d are set to 0. After thresholding, $g_{NH}(x,y)$ will have fewer nonzero pixels than $g_{NL}(x,y)$, but all the nonzero pixels in $g_{NH}(x,y)$ will be contained in $g_{NL}(x,y)$because the latter image is formed with a lower threshold. We eliminate from $g_{NL}(x,y)$ all the nonzero pixels from $g_{NH}(x,y)$ by letting $g_{NL}(x,y) = g_{NL}(x,y) - g_{NH}(x,y)$. After the thresholding operations, all strong pixels in $g_{NH}(x,y)$ are assumed to be valid edge pixels and are so marked immediately. Depending on the value of T_H, the edges in $g_{NH}(x,y)$ typically have gaps. Non Maximal Suppression and Hysteresis are used in Canny edge detection.

Non Maximal suppression is used to suppress the edges that are not locally maximized in a particular direction and hysteresis is used to find the strong edge responses with the help of an upper and lower threshold, there by identifying strong responses. We understood the concepts on how to generate Gaussian kernels for different values of sigma. Also varying the sigma values to generate the kernels taught us about the filtering capabilities of the Gaussian kernel. Convolution with Gaussian helped us understand the techniques involved in convolution, which is a very crucial for this project. Edge detection by LOG, DOG and Zero Crossing was the foundation for this project. It helped us understand why there is a need for a good detector like Canny Edge detector. Feature Synthesis in Images for the Canny Edge detection. Fourier series and Fourier Transforms is the base for convolution and other important image processing operations. Image pre- processing techniques helped us understand how images can be processed before performing complex operations on them, like Gaussian blurring. All these topics and other Image processing and computer vision concepts helped us understand the concepts of images and image processing and computer vision and helped us do this project. Edge detection is one of the most important phases of an Image understanding or Image processing. It is the basis for further concepts like segmentation. This applies not only in Computer vision and Image processing applications but also in Human Image understanding. For this reason, Edge detection plays a very important role in the fundamentals of Images. So, there is a need for the implementation of an efficient edge detector. There are many types of Edge detections that in this course like Zero Crossing, LoG, DoG, which stands for Laplacian of Gaussian and Difference of Gaussian respectively which are quite natural and computationally easier ways of Edge detection. However, with these kinds of edge detection techniques, there are a few problems that we encounter. There are three important criteria in detecting the edges. They are

Detection

Localization

Single respons they are responsible for detecting the right edge, locating the right edge in its right place and getting a single response for each identified edge. The last factor, single response was where most of the above mentioned edge detectors fail. Some of the edge detectors previously also failed in the other criterion too. So, there was a need for an Edge detection scheme with a computational approach. That is where Canny Edge Detection comes from. It stressed on all these three features and gets the best edges possible. So, we implemented a Canny Edge Detector. The phases of our project are as follows.

1. Convolution of the image with a suitable Gaussian filters.
2. Finding edge normal directions of the image using horizontal and vertical components.
3. Finding the locality and Magnitude of the edges.
4. Applying Non Maximal suppression
5. Apply two threshold values, upper and lower, through Hysteresis and eliminate weak responses. Then we could run the algorithms on MATLAB and we could get efficient edges from the given input images. The results are presented in the latter parts of this report. We followed all the steps suggested by Canny in his paper [1].

Convolution with a sigma(σ): The sigma chose initially was "1.4" (Based on the outputs observed, shown in the later part of this report). The reason being, if we use low value for sigma, we can see a sharp transition from edges to the non-edge image pixels. If we use a large sigma, one can see a gradual transition from the interior of the image to the edge.

1. Find the horizontal and vertical components of the image using Sobel Operator.
2. Then estimate normal edge directions using the function atan2 in MatLab.
3. Find the magnitude of the image using horizontal and vertical components.
4. Find out the location of every pixel by assigning each pixel to an angle among 0, 45, 90 and 135 depending upon the closest to their particular angle.
5. Then carry out non- maximal suppression to the direction assigned image. The pixels whose value is not maximum in the considered neighbourhood in its direction, they are removed or they remain as edges.
6. Then apply hysteresis using upper and lower thresholds. Mark all the pixels greater than that upper threshold as true and the pixels with value lesser than lower threshold are marked as false. And the pixels that lie in between upper and lower thresholds were assigned a dummy value. Then, these pixels were considered separately and if it contained any neighbouring pixel as an edge it was marked an edge or was marked false.
7. As Canny observed, the ratio of the two thresholds were put between 2 and 3 so that clear outputs were

observed. This image is the edge detector detected by Canny's algorithm. The following the above steps for increasing values of sigma (σ). Initially started with σ=1 and continued till σ=5 incrementing σ by 1 for each time. Then chose σ=1,1.4,2,3,4,5 . that in order to understand how the increment in the value of σ affected the sharpness or bluntness of the edge. All of the algorithms were applied and implemented in MATLAB. Also 1. Noise can be of many profiles and additive white noise (Gaussian is of one type). Canny is suitable for Gaussian only. Then try to give impulse noise to canny and observed that it did not provide expected results.

8. Learn that a great way to suppress impulse noisy profiles is by "MEDIAN FILTERING"3. Apply median filtering to the 'lena.jpg' corrupted by impulse noise and passed the output to canny edge detector. The results and observations are presented below.

Fig 1 Canny obtained by our algorithm sigma =1.4

Fig 2 Canny obained using edge with function of MATLAB with same sigma

Fig 3 Canny obtained by algorithm with sigma=2

Fig 4 Canny obtained by algorithm with sigma=3

Fig 5 Canny obtained by algorithm with sigma=4

Fig 6 Canny obtained by algorithm with sigma=5

Fig 7 Image corrupted by impulse noise

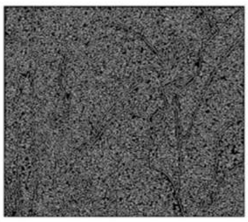

Fig 8 Canny obtained by with sigma=1.4

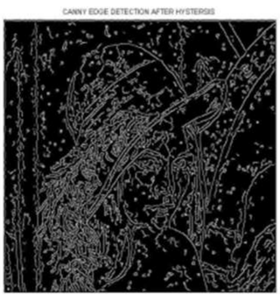

Fig 9 Canny obtained after median filtering with mask size 3 and sigma=1.4

Fig 10 Canny obtained by median filtering with mask size 5 and sigma=1.4

Observations

There are a few lessons that should be learnt while doing this:

1. Translation of algorithm to working code is a difficult task.
2. Canny's implementation is better compared to the other edge detection algorithms that are studied, it is an efficient edge detection algorithm but limited to correcting additive white noise.
3. Canny does not respond well to impulse noise and we needed to suppress the impulse noise using median filtering.
4. Median filtering performed with kernel size 5 performs well when given as input to canny edge detector.

References

1. Rafael C. Gonzalez and Richard E. Woods : Digital Image Processing (741, 746)
2. Artificial Intelligence and Computational Intelligence by Jingsheng Lei Fu Lee Wang Hepu Deng Duoqian Miao
3. K.V. Kale: Advances in computer Vision and Information Technology (860)
4. Jayaraman: Digital Image Processing
5. Anthony Reed Shipman :Detection of Tubes in Radiographs Using Canny Edge Detection and Progressive Hough
6. Mazlinda Ibrahim: Medical Imaging Through Histogram Equalization and Canny Edge Detection
7. Siti Nurul Balkhis Abdul Latif: Edge Detection Using Canny Edge Detector in Digital Image Processing
8. Srikanta Patnaik, Biojiang Zhong: Soft Computing Techniques in Engineering Application (183)
9. Dehuai Zeng: Advances in Computer Science and Engineering
10. Oge Marques: Practical Image and Video Processing Using MATLAB
11. Jingran Sun: Edge Detection, Image Segmentation and Their Application in Microarray Image Analysis
12. Wilhelm Burger, Mark J. Burge : Principles of Digital Image Processing : Fundamental Techniques of Edge Detecions

How to cite this article:

Khanjan K Pandya.2018, The Canny Edge Detector. *Int J Recent Sci Res.* 9(5), pp. 27036-27038.
DOI: http://dx.doi.org/10.24327/ijrsr.2018.0905.2173

****