A Review on Edge detection Technique "Canny Edge Detection"

Ramnarayan
Assistant Professor
CSE Dept, Tula's Institute
The Engineering & Management College
Dehradun, Uttrakhand, India

Nikita Saklani
Research Scholar (M.Tech, CSE)
Tula's Institute
The Engineering & Management College
Dehradun, Uttrakhand, India

Vasundhara Verma
Research Scholar (M.Tech, CSE)
Tula's Institute
The Engineering & Management College
Dehradun, Uttrakhand, India

ABSTRACT
The traditional Canny edge detection method is widely used in gray image processing. However, this traditional algorithm is unable to deal with color images and the parameters in the algorithm are difficult to be determined adaptively. In this paper, an improved Canny algorithm is proposed to detect edges in color image. The proposed algorithm is composed of the following steps: quaternion weighted average filter, vector Sobel gradient computation, non-maxima suppression based on interpolation, edge detection and connection. Experimental results show that the proposed algorithm outperforms other color image edge detection methods and can be widely used in color image processing. This project we present a two dimensional edge detector which gives the edge position in an image with sub-pixel accuracy. The method presented here gives an excellent accuracy (the position bias mean is almost zero and the standard deviation is less than one tenth of a pixel) with a low computational cost' and its simple since it is derivated from the well known Non-Maxima Suppression method in Matlab[1].

Keywords
Canny algorithm, Matlab, Digital Image, Sobel Algorithm.

1. INTRODUCTION
Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*. The same problem of finding discontinuities in 1D signal is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.[2][3].

1.1 Canny Edge Detection
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**.** This was also stated in my Sobel and Laplace edge detection tutorial, but I just wanted reemphasize the point of why you would want to detect edges.

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. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "*A Computational Approach to Edge Detection*". In this project followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate[1]. 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.

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)[9].

Step 1
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.[4]

Step 2
After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows)[7]

Step 3
Finding the edge direction is trivial once the gradient in the x and y directions are known. However, you will generate an error whenever sum X is equal to zero. So in the code there has to be a restriction set whenever this takes place [8]. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If GY has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is just:Theta = invtan (Gy / Gx)

Step 4

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows[2].

x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions.

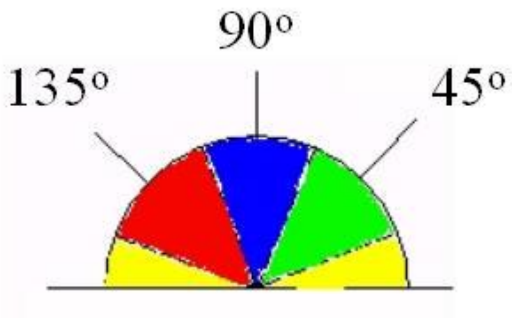


Figure 1.1

1.2 Background

This project we present a two dimensional edge detector which gives the edge position in an image with sub-pixel accuracy[10]. The method presented here gives an excellent accuracy (the position bias mean is almost zero and the standard deviation is less than one tenth of a pixel) with a low computational cost' and its simple since it is derivated from the well known Non-Maxima Suppression method in Matlab[6].

1.3 Motivation

A method that combines region growing and edge detection for image segmentation is presented. This method starts with a split-and merge algorithm wherein the parameters have been set up so that an over-segmented image results. Region boundaries are then eliminated or modified on the basis of criteria that integrate contrast with boundary smoothness, variation of the image gradient along the boundary, and a criterion that penalizes for the presence of artifacts reflecting the data structure used during segmentation (quadtree in this case).

1.4 Problem Statement

Canny edge detection method which has received much attention during the recent years due to its many applications in different fields. Edge detection is one of those challenging problems and up to date, there is no technique that provides a robust solution to all situations and different applications that edge detection method may encounter [7]

1.4.1 Technique description on which the project was implemented

The algorithms were implemented in the Matlab on workstation running under the Window XP operating system. Simple tool images and aerial photographs were used to test the algorithms. The impression of human observers is that the method is very successful on the tool images and less so on the aerial photograph images. It is thought that the success in the tool images is because the objects shown occupy areas of many pixels, making it is easy to select parameters to separate signal information from noise[5].

1.5 Experimental Framework

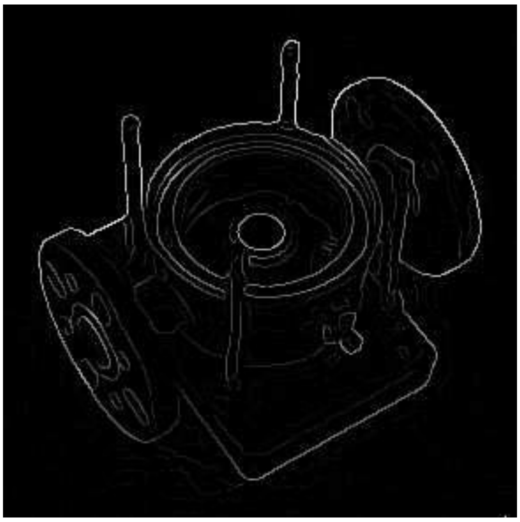
1. **Hardware Platform:** Intel Pentium PCs with 80 GB of Hard Disk and 1 GB of RAM.
2. **Software Platform:**

- Windows XP
- MATLAB

1.6 Result

(a) Original Image

(b) Smoothed



(c)

2. CONCLUSION

This paper provides the effective way to detect the edges of all type of images. Canny edge detection minimizes human effort for the detection of solution for algorithms and in many domains. This provides a affordable platform for problem solving. This edge detection improves the accuracy and easy to detect all possible edges according parameter.

3. REFERENCES

[1] LS. Davis, "A survey of edge detection techniques", Computer Graphics and Image Processing, vol 4, no. 3, pp: 248-260, 1975.

[2] R. O. Duda. and P. E. Hart, "Pattern Classification and Scene Analysis," Wiley, New York, 1973.

[3] Abdallah A. Alshennawy, And Ayman A. Aly, "Edge Detection in Digital Images Using Fuzzy Logic Technique". World Academy of Science Engineering and Technology 51 , pp:178-186, 2009.

[4] Begol, Moslem and Maghooli,Keivan "Improving Digital Image Edge Detection by Fuzzy Systems" World Academy of Science, Engineering and Technology 81, pp: 76-79, 2011.

[5] Lei Zhang, Mei Xiao, Jian Ma and Hongxun Song "Edge Detection by Adaptive Neuro-Fuzzy Inference System" 2nd International Congress on Image and Signal Processing, CISP '09.2009.

[6] Hamed Mehrara , Mohammad Zahedinejad and Ali Pourmohammad "Novel Edge Detection Using BP Neural Network Based on Threshold Binarization" Second International Conference on Computer and Electrical Engineering, pp:408-412, 2009.

[7] T. Law, H. Itoh, H. Seki, "Image filtering, edge detection, and edge tracing using fuzzy reasoning," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.18, pp.481-491, 1996.

[8] Jinbo Wu, Zhouping Yin, and Youlun Xiong, "The Fast Multilevel Fuzzy Edge Detection of Blurry Images," IEEE Signal Processing Letters, vol. 14, no. 5, 2007.

[9] Tamalika Chaira, A.K. Ray, "A new measure using intuitionistic fuzzy set theory and its application to edge detection," Applied Soft Computing, 8 pp.919–927, 2008.

[10] M. EminYüksel, "Edge detection in noisy images by neuro-fuzzy processing," Int. J. Electron. Commun. (AEü) 61 pp.82–89, 2007.