International Journal of Scientific Engineering and Science
Volume 1, Issue 10, pp. 58-59, 2017. ISSN (Online): 2456-7361

Edge Detection Techniques in Processing Digital Images: Investigation of Canny Algorithm

K. Shankar[1], Dr. S. Srinivasan[2], Dr. T. S. Sivakumaran[3], K. Madhavi Priya[4]
[1,2]Department of E&I Engg., Annamalai University, Chidhambaram, Tamil Nadu, India
[3]Sasurie Academy of Engg., Coimbatore, Tamil Nadu, India
[4]Department of ECE, SKP Engg. College Tiruvannamalai, Tamil Nadu, India

Abstract**—*Edge detection is a primary function in image processing. It is the name for a set of mathematical methods which aims at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. In this paper we are going to have a short introduction to edge detection basic concepts and continue with popular methods: Canny edge detection.*

Keywords**—** *Edge detection, canny algorithm.*

I. Introduction

The images captured by an image processor consist of low pixel values. They are not suitable for the processing of an image, so the conversion of an image is needed. The original image should be processed in three ways, first the image enhancement, which processes a given image and it accentuates or sharpens image features such as edges, boundaries, or contrast to make a graphic display, which is more helpful for display and analysis.

The analysis of grey level transformation to perform three independent transformations to the grey level of any input pixel. This method produces a composite image whose colour content is modulated by the nature of the transformation functions.

Edge detection is one of the fundamental steps in image processing, image analysis, image pattern recognition, and computer vision techniques. In the ideal case, the result of applying an edge detector to an image may lead to a set of connected curves, that indicate the boundaries of objects, the boundaries of surface markings, as well as curves that correspond to discontinuities in surface orientation.

Thus, applying a canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

Further, to discuss about the edge detection and canny edge detector

II. Edge Detection

Edge detection to detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction [2].

The edges extracted from a two-dimensional image of a three-dimensional scene can be classified as either viewpoint dependent or viewpoint independent.

A viewpoint independent edge typically reflects inherent properties of the three- dimensional objects, such as surface markings and surface shape.

A viewpoint dependent edge may change as the viewpoint changes, and typically reflects the geometry of the scene, such as objects including one another.

III. Canny Edge Detection

The Canny Edge Detector is one of the most commonly used image processing tools, detecting edges in a very robust manner. It is a multi-step process, which can be implemented on the GPU as a sequence of filters [3]. In this situation, an "optimal" edge detector means:

Good detection: - the algorithm should mark as many real edges in the image as possible.

Good localization: - edges marked should be as close as possible to the edge in the real image.

Minimal response: - a given edge in the image should only be marked once, and where possible, image noise should not create false edges.

To satisfy these requirements Canny used a various technique

Noise Reduction

Because the Canny edge detector is susceptible to noise present in raw unprocessed image data, it uses a filter based on a Gaussian (bell curve), where the raw image is convolved with a Gaussian filter.

The result is a slightly blurred version of the original which is not affected by a single noisy pixel to any significant degree.

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

$$G(x,y)=e^{-(x^2 + y^2/2 a^2)}$$

We form a smoothed image fs(x,y) by convolving G and f:

$$fs(x,y) = G(x,y) * f(x,y)$$

Tracing Edges through the Image and Hysteresis Thresholding

Large intensity gradients are more likely to correspond to edges than small intensity gradients. It is in most cases impossible to specify a threshold at which a given intensity gradient switches from corresponding to an edge into not doing so. Therefore Canny uses Thresholding with hysteresis.

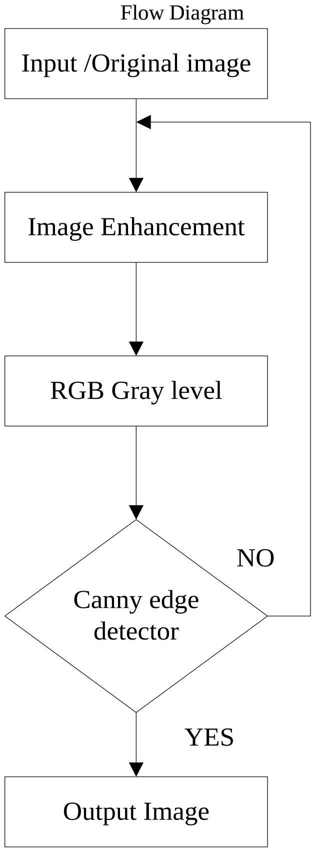
Volume 1, Issue 10, pp. 58-59, 2017.
International Journal of Scientific Engineering and Science
ISSN (Online): 2456-7361

Thresholding with hysteresis requires two thresholds – high and low. Making the assumption that important edges should be along continuous curves in the image allows us to follow a faint section of a given line and to discard a few noisy Pixels that do not constitute a line but have produced large gradients. Therefore we begin by applying a high threshold.

These marks out the edges we can be fairly sure are genuine. Starting from these, using the directional information derived earlier, edges can be traced through the image. While tracing an edge, we apply the lower threshold, allowing us to trace faint sections of edges as long as we find a starting point.

Once this process is complete we have a binary image where each pixel is marked as either an edge pixel or a non-edge pixel. From complementary output from the edge tracing step, the binary edge map obtained in this way can also be treated as a set of edge curves, which after further processing can be represented as polygons in the image domain.

Flow Diagram



IV. Simulation & Environments

Fig. (a). Input image.

To capture the high pixels and resolutions of an image is given to input in Fig. (a).

Fig. (b). Image enhancement.

The given input image to sharpening and edge detection in second derivative of an image Fig. (b).

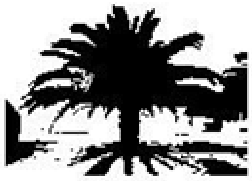


Fig. (c). RGB gray level.

Color transforms involve conversion of color information from one color space to another, conversions from color images to grayscale, and representing grayscale images with false color Fig. (c).

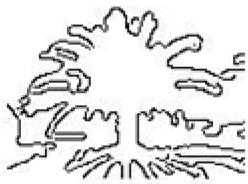


Fig. (d). Canny edge detector.

The three images of the calculated edges demonstrate the reduction of noise with the increase in the size of the convolution kernel. That disappeared when we used the canny detector Fig. (d).

V. Conclusions and Future Work

It is a basic processing of the Digital Images convert into canny edge detector. In the future, To study the interaction between appearance and boundary information for image segmentation using probabilistic approaches. Extending our Framework to video-based object extraction of an image.

References

[1] E. Argyle, “Techniques for edge detection,” *Proc. IEEE*, vol. 59, pp. 285-286, 1971.

[2] M. Juneja and P. Singh Sandhu, “Performance evaluation of edge detection techniques for images in spatial domain,” *International Journal of Computer Theory and Engineering*, vol. 1, no. 5, pp. 614-621, December, 2009.

[3] M. Nosrati, R. Karimi, M. Hariri, and K. Malekian, “Edge detection techniques in processing digital images: Investigation of canny algorithm and gabor method,” *World Applied Programming*, vol. (3), issue (3), pp. 116-121, March 2013.

[4] D. R. Martin, C. C. Fowlkes, and J. Malik, “Learning to detect natural image boundaries using local brightness, color, and texture cues,” *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 26, no. 5, pp. 530-549, May 2004.

[5] http://en.wikipedia.org/wiki/Canny edge detector