

DIFFERENT EDGE DETECTION TECHNIQUES FOR PROCESSING DIGITAL IMAGES: CANNY ALGORITHM AND GABOR METHOD

Mr. Yogesh B. Dongare*

Prof. Ruhi Kabra*

Abstract:

Edge detection is a primary function in image processing. It 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.

In this paper we are going to have a short introduction to edge detection basic concepts and continue with 2 popular methods: Canny edge detection and Gabor method.

Keywords: edge detection, Canny, Gabor

* Department of Computer Engineering, G.H.Raisoni College of Engineering & Management, Chas, Ahmednagar

I. 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 signals 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.

The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to:[1][2]

discontinuities in depth, discontinuities in surface orientation, changes in material properties and variations in scene illumination.

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 an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified. However, it is not always possible to obtain such ideal edges from real life images of moderate complexity.

Edges extracted from non-trivial images are often hampered by fragmentation, meaning that the edge curves are not connected, missing edge segments as well as false edges not corresponding to interesting phenomena in the image – thus complicating the subsequent task of interpreting the image data.[3]

Edge detection is one of the fundamental steps in image processing, image analysis, image pattern recognition, and computer vision techniques. During recent years, however, substantial (and successful) research has also been made on computer vision methods that do not explicitly rely on edge detection as a pre-processing step.

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 threedimensional 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 occluding one another.

A typical edge might for instance be the border between a block of red color and a block of yellow. In contrast a line (as can be extracted by a ridge detector) can be a small number of pixels of a different color on an otherwise unchanging background. For a line, there may therefore usually be one edge on each side of the line.

Although certain literature has considered the detection of ideal step edges, the edges obtained from natural images are usually not at all ideal step edges. Instead they are normally affected by one or several of the following effects:

- focal blur caused by a finite depth-of-field and finite point spread function.
- penumbral blur caused by shadows created by light sources of non-zero radius.
- shading at a smooth object

A number of researchers have used a Gaussian smoothed step edge (an error function) as the simplest extension of the ideal step edge model for modeling the effects of edge blur in practical applications [3][4]. Thus, a one-dimensional image f which has exactly one edge placed at $x=0$ may be modeled as:

$$f(x) = \frac{I_r - I_l}{2} \left(\operatorname{erf} \left(\frac{x}{\sqrt{2}\sigma} \right) + 1 \right) + I_l.$$

At the left side of the edge, the intensity is $I_l = \lim_{x \rightarrow -\infty} f(x)$, and right of the edge it is $I_r = \lim_{x \rightarrow +\infty} f(x)$.

The scale

$$x \rightarrow -\infty$$

$$x \rightarrow +\infty$$

parameter is called the blur scale of the edge [5].

II. CANNY EDGE DETECTION

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works.

Canny's aim was to discover the optimal edge detection algorithm. 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 the calculus of variations – a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

1) 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.

Here is an example of a 5x5 Gaussian filter, used to create the image to the right, with $\sigma=1.4$. (The asterisk denotes a convolution operation.)

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A.$$

2) Finding the intensity gradient of the image

An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (Roberts, Prewitt, Sobel for example) returns a value for the first derivative in the horizontal direction (G_x) and the vertical direction (G_y). From this the edge gradient and direction can be determined:

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

The edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals (0, 45, 90 and 135 degrees for example).

3) Non-maximum suppression

Given estimates of the image gradients, a search is then carried out to determine if the gradient magnitude assumes a local maximum in the gradient direction. So, for example,

- if the rounded gradient angle is zero degrees (i.e. the edge is in the north-south direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the **north and south** directions,
- if the rounded gradient angle is 90 degrees (i.e. the edge is in the east-west direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the **west and east** directions,
- if the rounded gradient angle is 135 degrees (i.e. the edge is in the north east-south west direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the **north east and south west** directions,

- if the rounded gradient angle is 45 degrees (i.e. the edge is in the north west-south east direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the **north west and south east** directions.

From this stage referred to as non-maximum suppression, a set of edge points, in the form of a binary image, is obtained. These are sometimes referred to as "thin edges".

4) *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.

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. This 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.

5) *Differential geometric formulation of the Canny edge detector*

A more refined approach to obtain edges with sub-pixel accuracy is by using the approach of differential edge detection, where the requirement of non-maximum suppression is formulated in terms of second- and third-order derivatives computed from a scale space representation (Lindeberg 1998) – see the article on edge detection for a detailed description.

6) *Variational-geometric formulation of the Haralick-Canny edge detector*

A variational explanation for the main ingredient of the Canny edge detector, that is, finding the zero crossings of the 2nd derivative along the gradient direction, was shown to be the result of minimizing a Kronrod-Minkowski functional while maximizing the integral over the alignment of the edge with the gradient field (Kimmel and Bruckstein 2003). See article on regularized Laplacian zero crossings and other optimal edge integrators for a detailed description.

The Canny algorithm contains a number of adjustable parameters, which can affect the computation time and effectiveness of the algorithm.

- The size of the Gaussian filter: the smoothing filter used in the first stage directly affects the results of the Canny algorithm. Smaller filters cause less blurring, and allow detection of small, sharp lines. A larger filter causes more blurring, smearing out the value of a given pixel over a larger area of the image. Larger blurring radii are more useful for detecting larger, smoother edges – for instance, the edge of a rainbow.
- Thresholds: the use of two thresholds with hysteresis allows more flexibility than in a single-threshold approach, but general problems of thresholding approaches still apply. A threshold set too high can miss important information. On the other hand, a threshold set too low will falsely identify irrelevant information (such as noise) as important. It is difficult to give a generic threshold that works well on all images. No tried and tested approach to this problem yet exists [6].

The algorithm can be summarized to following steps [7]:

1. Smoothing: Blurring of the image to remove noise.
2. Finding gradients: The edges should be marked where the gradients of the image has large magnitudes.
3. Non-maximum suppression: Only local maxima should be marked as edges.
4. Double thresholding: Potential edges are determined by thresholding.
5. Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

III.GABOR METHOD

In image processing, a Gabor filter, named after Dennis Gabor, is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. The Gabor filters are self-similar: all filters can be generated from one mother wavelet by dilation and rotation.

J. G. Daugman discovered that simple cells in the visual cortex of mammalian brains can be modeled by Gabor functions [8]. Thus, image analysis by the Gabor functions is similar to perception in the human visual system.

Its impulse response is defined by a harmonic function multiplied by a Gaussian function. Because of the multiplication convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function. The filter has a real and an imaginary component representing orthogonal directions [9]. The two components may be formed into a complex number or used individually.

Complex

$$g(x, y, \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{x'}{\lambda} + \psi\right)\right)$$

Real

$$g(x, y, \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right)$$

Imaginary

$$g(x, y, \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi \frac{x'}{\lambda} + \psi\right)$$

Where,

$$x' = x \cos \theta + y \sin \theta$$

and

$$y' = -x \sin \theta + y \cos \theta$$

In this equation, λ represents the wavelength of the sinusoidal factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the sigma of the Gaussian envelope and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function.

Gabor filters are directly related to Gabor wavelets, since they can be designed for a number of dilations and rotations. However, in general, expansion is not applied for Gabor wavelets, since this requires computation of bi-orthogonal wavelets, which may be very time-consuming. Therefore, usually, a filter bank consisting of Gabor filters with various scales and rotations is created. The filters are convolved with the signal, resulting in a so-called Gabor space. This process is closely related to processes in the primary visual cortex [10]. Jones and Palmer showed that the real part of the complex Gabor function is a good fit to the receptive field weight functions found in simple cells in a cat's striate cortex [11].

The Gabor space is very useful in image processing applications such as optical character recognition, iris recognition and fingerprint recognition. Relations between activations for a specific spatial location are very distinctive between objects in an image. Furthermore, important activations can be extracted from the Gabor space in order to create a sparse object representation [12].

IV. CONCLUSION

In this paper we talked about edge detection as a primary function in image processing. It 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. Due to it, we had a survey on canny edge detection method and Gabor filter. All the details and steps were described.

REFERENCES

- [1] H.G. Barrow and J.M. Tenenbaum (1981) "Interpreting line drawings as three-dimensional surfaces", *Artificial Intelligence*, vol 17, issues 1-3, pages 75-116.
- [2] Lindeberg, Tony (2001), "Edge detection", in Hazewinkel, Michiel, *Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4
- [3] T. Lindeberg (1998) "Edge detection and ridge detection with automatic scale selection", *International Journal of Computer Vision*, 30, 2, pages 117--154.
- [4] W. Zhang and F. Bergholm (1997) "Multi-scale blur estimation and edge type classification for scene analysis", *International Journal of Computer Vision*, vol 24, issue 3, Pages: 219 - 250.
- [5] Edge detection article. Available at: http://en.wikipedia.org/wiki/Edge_detection
- [6] Canny edge detector article. Available at: http://en.wikipedia.org/wiki/Canny_edge_detector
- [7] MasoudNosrati, RonakKarimi, Mehdi Hariri. Detecting circular shapes from areal images using median filter and CHT. *Global Journal of Computer Science and Technology* Volume XII Issue II Version I, January 2012.
- [8] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, July 1985.
- [9] 3D surface tracking and approximation using Gabor filters, JesperJuulHenriksen, South Denmark University, March 28, 2007
- [10] Daugman, J.G. (1980), "Two-dimensional spectral analysis of cortical receptive field profiles", *Vision Res.* 20 (10): 847–56, PMID 7467139
- [11] J.P. Jones and L.A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophysiol.*, 58(6):1233-1258, 1987
- [12] Gabor filter article. Available at: http://en.wikipedia.org/wiki/Gabor_filter