



A Survey on Edge Detection Methods

Technical Report: CES-506

Mohammadreza Asghari Oskoei and Huosheng Hu

29 February 2010

School of Computer Science & Electronic Engineering
University of Essex
Colchester CO4 3SQ, United Kingdom

Email: masgha@essex.ac.uk ; hhu@essex.ac.uk

ISSN 1744 - 8050

Abstract

This manuscript is a review over the published articles on edge detection. At first, it provides theoretical background, and then reviews wide range of methods of edge detection in different categorizes. The review also studies the relationship between categories, and presents evaluations regarding to their application, performance, and implementation. It was stated that the edge detection methods structurally are a combination of image smoothing and image differentiation plus a post-processing for edge labelling. The image smoothing involves filters that reduce the noise, regularize the numerical computation, and provide a parametric representation of the image that works as a mathematical microscope to analyze it in different scales and increase the accuracy and reliability of edge detection. The image differentiation provides information of intensity transition in the image that is necessary to represent the position and strength of the edges and their orientation. The edge labelling calls for post-processing to suppress the false edges, link the dispread ones, and produce a uniform contour of objects.

Table of Contents

1. Introduction.....	4
2. Definitions	5
2.1 Digital Image	5
2.2 Edge.....	5
3. Edge Detector	7
3.1 Image Differentiation.....	7
3.2 Discrete Differentiation	8
3.3 Convolution	10
3.4 Image Smoothing.....	13
3.5 Edge Labelling.....	14
3.6 Non-Maximum suppression	14
3.7 Hysteresis Algorithm	14
3.8 Sub-pixel Accuracy	15
4. Edge Detection Methods	17
4.1 Classical Methods.....	17
4.2 Gaussian Based Methods.....	18
4.3 Multi-Resolution Methods.....	20
4.4 Nonlinear Methods.....	23
4.5 Wavelet Based Methods.....	24
4.6 Statistical Methods	25
4.7 Machine Learning Based Methods	26
4.8 Contextual Methods	27
4.9 Line edge detectors.....	29
4.10 Coloured Edges' Methods	29
5. Discussion.....	31
5.1 Methods of Evaluation.....	32
5.2 Application: Contactless Paper Counting.....	32
6. Conclusion.....	34
References	35

1. Introduction

Interpretation of image contents is a significant objective in computer vision and image processing, and it has received much attention of researchers during the last three decades. An image contains different Information of scene, such as objects' shape, size, colour, and orientation, but discrimination of the objects from their background is the first essential task that should be performed before any interpretation. In order to extract the contour of an object, we must detect the edges forming that object, and this fact reveals the constitutional importance of edge detection in computer vision and image processing. Edge detection results benefit wide range of applications such as image enhancement, recognition, morphing, restoration, registration, compression, retrieval, watermarking, hiding, and etc.

Edge detection is a process that detects the presence and location of edges constituted by sharp changes in colour intensity (or brightness) of an image. Since, it can be proven that the discontinuities in image brightness are likely to correspond to: 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 curves that correspond to discontinuities in surface orientation. 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 (i.e. edge curves are not connected), missing edge segments, as well as false edges (i.e. not corresponding to interesting phenomena in the image), which all lead to complicating the subsequent task of image interpreting.

The edge representation of an image drastically reduces the amount of data to be processed, yet it retains important information about the shapes of objects in the scene. This description of an image is easy to integrate into a large number of object recognition algorithms used in computer vision and other image processing applications. An important property of the edge detection method is its ability to extract the accurate edge line with good orientation, and much literature on edge detection has been published in the past three decades. However, there is not yet any general performance index to judge the performance of the edge detection methods. The performance of an edge detection method is always judged subjectively and individually dependent to its application. But in general, it is mostly agreed that for a good edge detection, the edge line should be thin and with few speckles.

Generally, an edge detection method can be divided into three stages. In the first stage, a noise reduction process is performed. In order to gain better performance of edge detection, image noise should be reduced as much as possible. This noise reduction is usually achieved by performing a low-pass filter because the additive noise is normally a high-frequency signal. However, the edges can possibly be removed at the same time because they are also high-frequency signals. Hence, a parameter is commonly used to make the best trade-off between noise reduction and edges information preservation. In the second stage, a high-pass filter such as a differential operator is usually employed to find the edges. In the last stage, an edge localization process is performed to identify the genuine edges, which are distinguished from those similar responses caused by noise [15].

This report reviews some dominant literature published in recent two decades on edge detection, including background, significant works, categories and evaluation. Section I is the introduction. Section II introduces basic concepts and definitions that are mostly employed in the reviewed literature. Section III provides a comprehensive theoretical and mathematical background for edge detection, including its 3 main components: image differentiation, image smoothing and edge labelling. It helps the reader to understand the significance of each method. Section IV presents dominant works of edge detection methods in ten categories, and explains their advantages and disadvantages and the relationships among them. Section V contains a quick discussion about the reviewed works as well as the conclusion.

2. Definitions

2.1 Digital Image

A digital image is a binary representation of a two-dimensional (2D) image, and can be in form of a vector or raster type. Raster uses a finite set of digital values, called pixels, to present an image. It contains a fixed number of rows and columns of pixels. In general, the term "digital image" usually refers to raster image also called bitmap image. Vector image is comprised of geometrical primitives such as points, lines, curves, and shapes or polygon(s), which are all based on mathematical equations, to represent images.

In a raster image, pixel is the smallest individual element associated to a specific position, and has a value consisting of one or more quantities (e.g. brightness of the given colours) related to that position. Typically, the pixels are stored in a two-dimensional array of either integer or float numbers. These values are often transmitted or stored in different forms (e.g. compressed) that make different formats. Raster images can be created by a variety of input devices and techniques, such as digital cameras, scanners, coordinate-measuring machines. They can also be synthesized from arbitrary non-image data, such as mathematical functions or three-dimensional geometric models. The field of image processing is the study of algorithms for their transformation or interpretation.

Digital images can be classified according to the number and nature of those values. A binary image is a digital image that has only two possible values (i.e. 0 or 1) for each pixel. Typically, the two colours used for a binary image are black and white though any two colours can be used. A greyscale is an image, in which the value of each pixel is a single value that carries colour intensity information. Greyscale or so-called monochromatic images are composed exclusively of shades of gray, varying from black (lowest intensity) to white (highest intensity). Colour image contains colour information for each pixel. For visually acceptable results, it is necessary to provide three values (colour channels, typically, Red, Green, and Blue in RGB format) for each pixel. The RGB colour space is commonly used in computer displays, but other spaces such as HSV are often used in other contexts. A true-colour image of a subject is an image that appears to the human eye just like the original, while a false-colour image is an image that depicts a subject in colours that differ from reality.

Range image represents the depth in the value of each pixel. It can be produced by range finder devices, such as a laser scanner, and makes a 3D volume by inserting third dimension (i.e., depth) into the 2D array of pixels.

2.2 Edge

Edge is a part of an image that contains significant variation. The edges provide important visual information since they correspond to major physical, photometrical or geometrical variations in scene object. Physical edges are produced by variation in the reflectance, illumination, orientation, and depth of scene surfaces. Since image intensity is often proportional to scene radiance, physical edges are represented by changes in the intensity function of an image [2].

The most common edge types are steps, lines and junctions. The step edges are mainly produced by a physical edge, an object hiding another or a shadow on a surface. It generally occurs between two regions having almost constant, but different, grey levels. The step edges are the points at which the grey level discontinuity occurs, and localized at the inflection points. They can be detected by using the gradient of intensity function of the image. Step edges are localized as positive maxima or negative minima of the first-order derivative or as zero-crossings of the second-order derivative (Figure 1). It is more realistic to consider a step edge as a combination of several inflection points. The most commonly used edge model is the double step edge. There are two types of double edges: the pulse and the staircase (Figure 2).

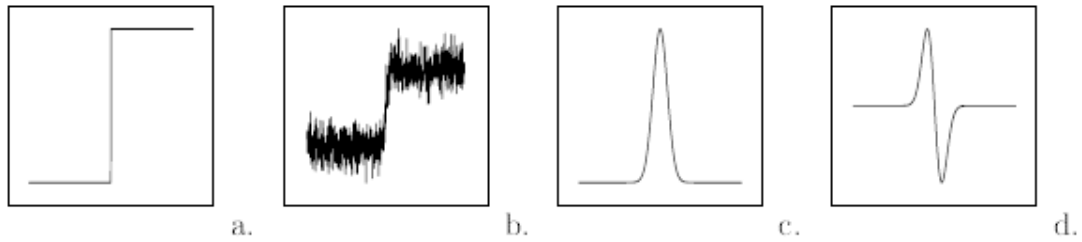


Figure 1 - profile of (a) ideal step edge (b) smoothed step edge corrupted by noise (c) first-order derivative (d) second-order derivative of the smoothed step edge corrupted by noise

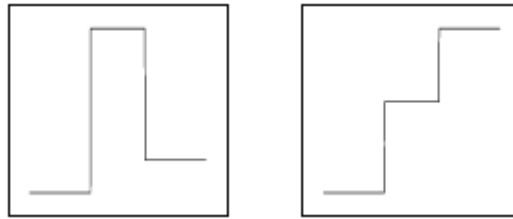


Figure 2 - profile of pulse (left) and staircase (right) step edges

The line edges are often created by either a mutual illumination between two objects that are in contact or a thin object placed over a background. Line edges correspond to local extremes in the intensity function. Lines correspond to local extrema of the image. They are localized as zero-crossings of the first derivative, or local maxima of the Laplacian, or local maxima of the grey level variance of the smoothed image. This type of edge is successfully used in remote sensing images for instance to detect roads and rivers [2]. Finally, the junction edge is formed where two or more edges meet together. A physical corner is formed at the junction of at least two physical edges. Illumination effects or occlusion, in which an edge occludes another, can produce a junction edge. Figure 3 depicts profiles of line and junction edges. The junction can be localized in various ways: e.g., a point with high curvature, or a point with great variation in gradient direction, or a zero-crossing of the Laplacian with high curvature or near an elliptic extremum. Though, the most of our studies encompass the all types of edges, but the majority of the reviewed literature is adapted to step edges, which are the most common.

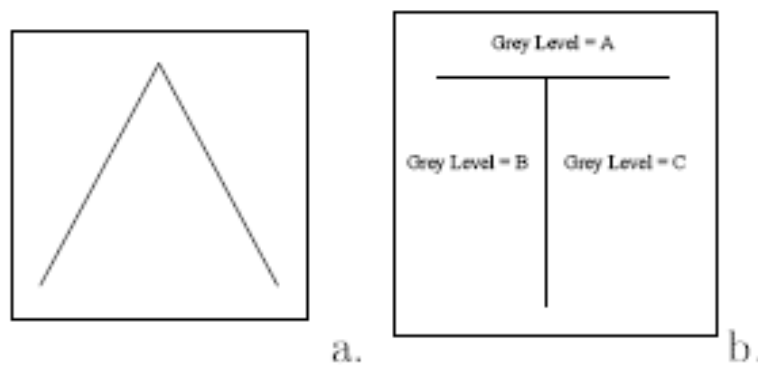


Figure 3 - profile of (a) line and (b) junction edges

The edges extracted from a 2D image of a 3D scene can be classified as either viewpoint dependent or viewpoint independent. A viewpoint independent edge typically reflects inherent properties of the 3D 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 [1].

3. Edge Detector

Edge detection is a terminology in image processing that refers to algorithms which aim at identifying edges in an image. It is encountered in the areas of feature selection and feature extraction in Computer Vision. An edge detector accepts a digital image as input and produces an edge map as output. The edge map of some detectors includes explicit information about the position and strength of the edges and their orientation.

In point of technical view, the edge detection methods can be grouped into two categories: search-based and zero-crossing based. The search-based methods detect the edges by first computing a measure of ‘edge strength’, such as magnitude of gradient of the image intensity function, and then searching for local maxima in a direction that matches with the ‘edge profile’, such as the gradient direction. The first-order derivative is regularly used to express the gradient. The zero-crossing based methods search for zero crossings in a second-order derivative expression computed from the image in order to find edges, such as the Laplacian or a non-linear differential expression. [1]

In point of conceptual view, the edge detection methods are categorized into contextual and non-contextual approaches. The non-contextual methods work autonomously without any priori knowledge about the scene and the edges. They are flexible in the sense that they are not limited to specific images. However, they are based on local processing focused on the area of neighbouring pixels. The contextual methods are guided by a priori knowledge about the edges or the scene. They perform accurately only in a precise context. It is clear that autonomous detectors are appropriate for general-purpose applications. However, contextual detectors are adapted to specific applications that always include images with same scenes or objects.

Structurally, the edge detection methods incorporate three operations: differentiation, smoothing and labelling. Differentiation consists in evaluating the desired derivatives of the image. Smoothing lies in reducing noise and regularizing the numerical differentiation. Labelling involves localizing edges and increasing the signal-to-noise ratio (SNR) of the detected edges by suppressing false edges. Labelling is often the last stage, but the order in which differentiation and smoothing are run depends on their properties. Smoothing and differentiation of an image are realized only by filtering the image with the differentiation of the smoothing filter. In this regards, the terms filter and detector are often used synonymously [2].

3.1 Image Differentiation

The first and second orders of derivatives of an image are the most common in the edge detection methods. For instance, to detect step edges we can look for maxima of the absolute value of the first derivative or zero crossings of the second derivative of an image. This Section reviews the required mathematical background to compute the differentiation of an image [3]. Consider $g(x, y)$ as a function defined in $\mathbf{R} \times \mathbf{R} \rightarrow \mathbf{R}$ that represents the image values (intensity). The first-order derivative of g can be calculated along a direction of r , using the partial derivatives of g along the main axes:

$$g_x = \frac{\partial g}{\partial x}, \quad g_y = \frac{\partial g}{\partial y} \quad (1)$$

$$\frac{\partial g}{\partial r} = \frac{\partial g}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial g}{\partial y} \frac{\partial y}{\partial r} = g_x \cos \varphi + g_y \sin \varphi \quad (2)$$

where φ is the angle formed between r and x axis. The gradient of g is a vector with the same direction as the maximum directional derivative, and its magnitude and direction are defined as follows.

$$|\nabla g| = \sqrt{g_x^2 + g_y^2} \quad (3)$$

$$\varphi_v = \arctan\left(\frac{g_y}{g_x}\right) \quad (4)$$

According to the definition, we can say that edge points may be located by the maxima of the module of the gradient, and the direction of edge contour is orthogonal to the direction of the gradient. The edge detection methods based on the gradient are directional, since they give their maximum response when they are aligned with the orthogonal direction of the edges contour.

Edge detection based on second order derivatives is frequently performed using one of two operators: the second derivative along the direction of the gradient or the Laplacian. The second derivative of g along the gradient direction is related with the derivatives along the axes x and y in the following way:

$$\frac{\partial^2 g}{\partial r^2} = \frac{g_x^2 g_{xx} + 2g_x g_y g_{xy} + g_y^2 g_{yy}}{g_x^2 + g_y^2} \quad (5)$$

$$g_{xx} = \frac{\partial^2 g}{\partial x^2}, \quad g_{yy} = \frac{\partial^2 g}{\partial y^2}, \quad g_{xy} = \frac{\partial^2 g}{\partial x \partial y} \quad (6)$$

The Laplacian of g , defined in (7), is an estimation of the second order derivative along the gradient direction. In the context of edge detection, it is shown that the Laplacian is a good approximation to the second derivative along the gradient direction, providing that the curvature of the line of constant intensity that crosses the point under consideration is small. Moreover, the Laplacian is useless in the detection of junction edges (zones of high curvature) [3].

$$\nabla^2 g = g_{xx} + g_{yy} \quad (7)$$

There are, at least, three major advantages of using the Laplacian in relation to the second derivative along the gradient direction. First, it is simple to use, since it only requires the computation of two second order derivatives. Second, it is a linear operator, in opposite to the second derivative, which is non-linear. Finally, but not less important, the Laplacian is a non-directional operator. This characteristic avoids the necessity to determine the most appropriated direction to apply the operator.

3.2 Discrete Differentiation

As shown in Section 2, digital images are sets of quantified samples corresponding to 2D array of pixels; then we need to represent them by discrete 2D function, and determine discrete approximations of the differential operators. We define the intensity function of a digital image as mapping $Z_R \times Z_C \rightarrow Z_I$ where Z_R , Z_C , and Z_I are a subset of $Z = \{0, 1, 2, 3, \dots\}$, and present the row, column and intensity value of the pixels, respectively. The discrete function $g(r, c)$ represents the colour intensity of the pixel placed in row r and column c , and is an approximation obtained from sampling and quantization of analogue function $g(x, y)$. One of the simplest ways to approximate the first order derivatives g_x and g_y is through the calculation of the first differences along the main axes, i.e.:

$$\begin{aligned} g_c(c, r) &= g(c, r) - g(c+1, r) \\ g_r(c, r) &= g(c, r) - g(c, r+1) \end{aligned} \quad (8)$$

where $g_c(c, r)$ and $g_r(c, r)$ denote, respectively, the approximation of g_x and g_y around the pixel (c, r) . These operators can be represented as mask, such as:

$$g_c(c, r) = [1 \quad -1] \begin{bmatrix} g(c, r) \\ g(c+1, r) \end{bmatrix}, \quad g_r(c, r) = [g(c, r) \quad g(c, r+1)] \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (9)$$

$$H_c = [1 \quad -1], \quad H_r = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (10)$$

This representation has the disadvantage of not being symmetric in relation to the point of interest, i.e. (c, r) , which originates a bias in position. One of the ways to avoid this problem consists in using an odd number of mask elements as, for example,

$$g_c(c, r) = [-1 \quad 0 \quad +1] \begin{bmatrix} g(c-1, r) \\ g(c, r) \\ g(c+1, r) \end{bmatrix}, \quad g_r(c, r) = [g(c, r-1) \quad g(c, r) \quad g(c, r+1)] \begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix} \quad (11)$$

$$H_c = [-1 \quad 0 \quad +1], \quad H_r = \begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix} \quad (12)$$

Several other first order derivative approximations along two perpendicular axes have been proposed [3]-[4], and some of the most known of them are as follow:

Roberts:

$$H_1 = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}, \quad H_2 = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad (13)$$

Prewitt:

$$H_c = \frac{1}{3} \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}, \quad H_r = \frac{1}{3} \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (14)$$

Sobel:

$$H_c = \frac{1}{4} \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad H_r = \frac{1}{4} \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (15)$$

Frei-Chen (isotropic):

$$H_c = \frac{1}{2+\sqrt{2}} \begin{bmatrix} -1 & 0 & +1 \\ -\sqrt{2} & 0 & +\sqrt{2} \\ -1 & 0 & +1 \end{bmatrix}, \quad H_r = \frac{1}{2+\sqrt{2}} \begin{bmatrix} +1 & +\sqrt{2} & +1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad (16)$$

As seen, except the former, the proposed operators are all odd and based on the image column and row directions. The Roberts operator is calculated using a set of axes rotated 45 degrees in relation to the usual orientation of the column and row. To use these operators, we perform an internal product between the respective mask and the image, as follows

$$g_\alpha(c, r) = \sum_i \sum_j g(c+i, r+j) \cdot (H_\alpha)_{ij} \quad (17)$$

All of the above-mentioned approximations have the final objective of calculating the gradient using (3) and (4). Despite the fact that it is enough to compute two directional derivatives in order to calculate the gradient, some researchers, for noise suppression reasons, have used more than two directional derivatives [3]. In this case, the gradient would be

approximated by the directional derivative with the highest amplitude. One of the most known is probably the one proposed by Kirsch [3], which is formed by the following masks:

$$\begin{aligned} \mathbf{H}_E &= \frac{1}{15} \begin{bmatrix} -3 & -3 & 5 \\ -3 & \mathbf{0} & 5 \\ -3 & -3 & 5 \end{bmatrix} \rightarrow \mathbf{H}_{NE} = \frac{1}{15} \begin{bmatrix} -3 & 5 & 5 \\ -3 & \mathbf{0} & 5 \\ -3 & -3 & -3 \end{bmatrix} \nearrow \\ \mathbf{H}_N &= \frac{1}{15} \begin{bmatrix} 5 & 5 & 5 \\ -3 & \mathbf{0} & -3 \\ -3 & -3 & -3 \end{bmatrix} \uparrow \mathbf{H}_{NW} = \frac{1}{15} \begin{bmatrix} 5 & 5 & -3 \\ 5 & \mathbf{0} & -3 \\ -3 & -3 & -3 \end{bmatrix} \nwarrow \end{aligned} \quad (18)$$

The arrows show the directions of the derivatives approximated by the masks. As can be seen easily, these masks are generated by rotations of 45 degrees of the elements around the central element. Other sets of directional masks can be obtained using similar rotations of the orthogonal masks of Prewitt and Sobel. The angular resolution allowed by a 3x3 operator is, at most, of 45 degrees. This means that we are only able to distinguish four different directions. For larger angular resolutions we have to use masks with a larger spatial support.

The second order differences are the simplest approximation to the second order derivative. We define the second differences along the main axes as

$$\begin{aligned} g_{cc}(c, r) &= g_c(c-1, r) - g_c(c, r) \\ g_{rr}(c, r) &= g_r(c, r-1) - g_r(c, r) \end{aligned} \quad (19)$$

by substituting we obtain:

$$\begin{aligned} g_{cc}(c, r) &= g(c-1, r) - 2g(c, r) + g(c+1, r) \\ g_{rr}(c, r) &= g(c, r-1) - 2g(c, r) + g(c, r+1) \end{aligned} \quad (20)$$

that can be represented by the following mask:

$$H_{cc} = \begin{bmatrix} 0 & 0 & 0 \\ +1 & -2 & +1 \\ 0 & 0 & 0 \end{bmatrix}, H_{rr} = \begin{bmatrix} 0 & +1 & 0 \\ 0 & -2 & 0 \\ 0 & +1 & 0 \end{bmatrix} \quad (21)$$

Using the definition of Laplacian and (20) we obtain a discrete approximation to the Laplacian given by

$$H_{cc+rr} = H_{cc} + H_{rr} = \begin{bmatrix} 0 & +1 & 0 \\ +1 & -4 & +1 \\ 0 & +1 & 0 \end{bmatrix} \quad (22)$$

3.3 Convolution

Convolution, in mathematics, is an operation on two function producing third function that is typically viewed as a modified version of one of the original functions. The discrete convolution of 2D function f and g is given by

$$(f * g) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j) g(r-i, c-j) \quad (23)$$

In image processing, the convolution is a general purpose filter that allows producing a range of effects by specifying a set of convolution kernels. It works by determining new value for a pixel by adding weighted values of all its neighbouring pixels together. The applied weights are determined by a 2D array called convolution kernel or mask. Comparison of (17) and (23)

shows that the convolution can be adopted to compute an estimation of a discrete differentiation of an image, subject to selection of a proper kernel (or mask), i.e. $f(i,j)$.

Due existence of the noise in a real image, we often require to use a modified form of the image for edge detection. Let us to show this by a sample. As it is depicted in Figure 4, the first derivative of a function that is affected by the noise is not enough to localize a step within it. The Figure 5 shows that we can localize the step by applying convolution. It is shown that a local maximum in the first derivative of the modified form of the function can localize the step. The modified function is obtained by applying convolution, and the kernel is a Gaussian function. Since convolution is a linear operator, we can deploy the derivative of the kernel to simplify the computation. Regarding to (24) we can apply the derivative of the desired kernel for convolution, and then look for the local maxima to localize the steps.

$$\frac{d}{dx}(f * g) = \frac{df}{dx} * g, \quad \frac{d^2}{dx^2}(f * g) = \frac{d^2 f}{dx^2} * g \quad (24)$$

Moreover, we can use the second derivative of the kernel to localize the step. As it is depicted in Figure 6, zero-crossing in the modified function denotes the step. The second derivative of the Gaussian function is employed as the kernel of convolution. This example reveals that the differentiation and modification (we will call it hereafter filtering and discuss in detail in next sub-section) of an image is realized with convolution.

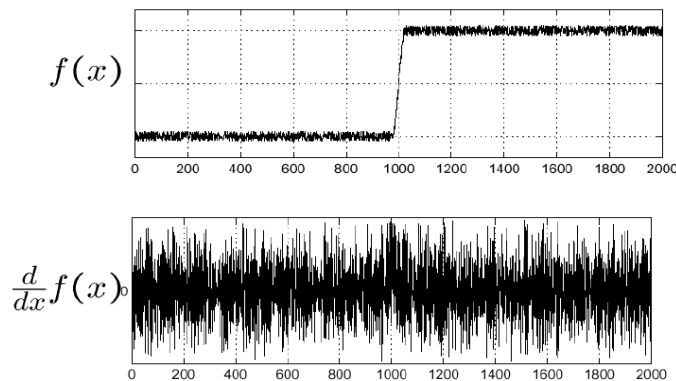
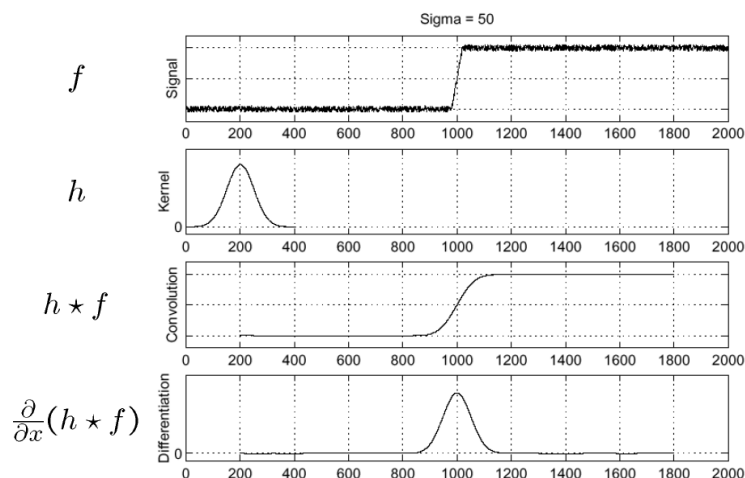


Figure 4 - first derivative of a function that is affected by the noise is not enough to localize a step within it



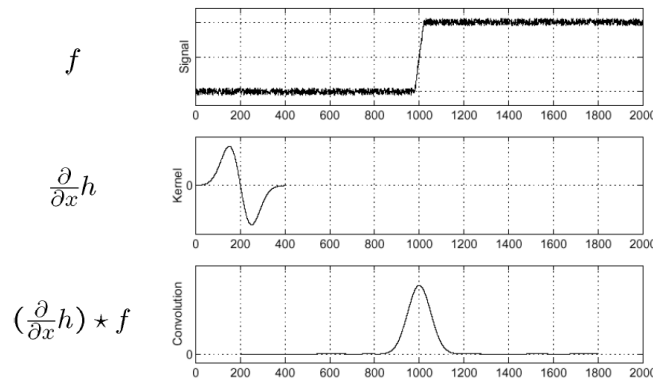


Figure 5 – (left) local maximum within the first derivative of the modified function by convolution represents the step (right) to reduce the operators we use the derivative of the kernel in convolution

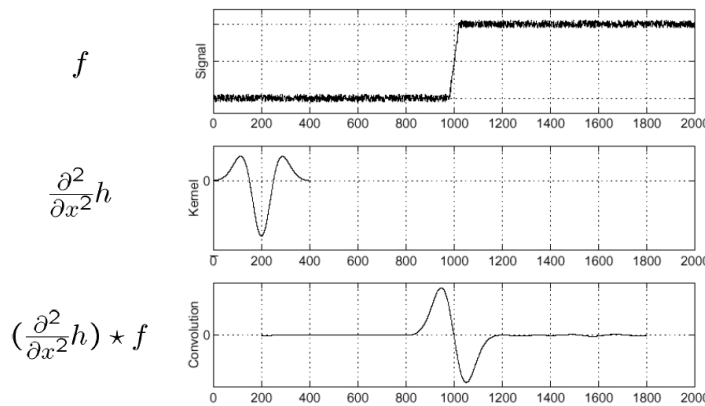


Figure 6 – zero-crossing in the modified function by a convolution with the second derivative of the kernel represents the step

Convolution allows us to study the effect of modifications on an image in both spatial and frequency domains. The use of a Fourier transform to convert images from the spatial to the frequency domain makes possible another class of filtering operations. Considering \mathfrak{F} as the Fourier transform operator, the formulation (25) help us to study the effect of applied filters on an image in the frequency domain. This property of the convolution is very helpful in the image smoothing which is going to be discussed in the next sub-section.

$$\mathfrak{F}(f * g) = \mathfrak{F}(f) \cdot \mathfrak{F}(g) \quad (25)$$

In practice, pixels located in the border of the image can raise problem, since convolution kernel extends beyond the borders. A common technique to cope with this problem, usually referred to zero boundary superposition, is simply to ignore the problematic pixels and to perform the convolution operation only on those pixels that are located at a sufficient distance from the borders. This method has the disadvantage of producing an output image that is smaller than the input image. Another option is performing the best job with the pixels near the boundary. For example, pixels on the corner only have about a quarter of the neighbours to use in the convolution that we have for pixels where the full filter can be used. The sum over those neighbouring pixels should be normalized by the actual number of pixels used in the sum. This normalization avoids overflow in the output pixels.

3.4 Image Smoothing

Noise reduction is the first objective in the smoothing operation. Noise in the image is inevitable, and it refers to points that do not match with the real scene. The noise results from both the image acquisition system and the nature of the scene under consideration (i.e., texture). Digital images include additional noise known as the sampling (or quantization) noise. It emerges due to transferring a real scene into limited pixels (i.e., resolution) and limited colours (i.e., depth). The noise in an image induces problems in edge detection and localization process. Image smoothing has a positive effect: noise reduction that ensures robust edge detection, but it also has a negative effect: lose of some information that degrades edge localization. Hence, there is a fundamental trade-off between loss of information and noise reduction. The ultimate goal is to find optimal detectors that ensure a favourable compromise between noise reduction and edge conservation.

Regularization of the numerical computation is another aim of smoothing. Due to applying the differentiation, the edge detection algorithms are often ill-posed in the sense that the existence, uniqueness, and stability of a solution cannot be guaranteed. One of the greatest problems is related to the instability over the high frequency noises. Let's see the following example that shows the differentiation amplifies high frequency noises [3]. Consider $g(x)$ is a function affected by a small amplitude noise $\varepsilon \cdot \text{Sin}(\omega t)$. The difference between $g(x)$ and $g(x) + \varepsilon \cdot \text{Sin}(\omega t)$ can be made arbitrarily small, if ε is made sufficiently small. However, the difference of their derivatives can be quite large if ω is made large. This means that the differentiation operation violates one of the principles of well-posed problems (i.e. stability). The other two principles are the existence of a solution and its uniqueness. It is shown (Poggio [86]) that by applying additional constraints, we can turn an ill-posed problem into a well-posed problem. This process is called regularization. Regularization is a formalization of the search for an optimal filter that builds the required additional constraints. The constraints should also concern the appropriate compromise between noise reduction and edge conservation.

Image smoothing is desired to provide an optimal filter that compromises between the elimination of noise and the preservation of image structure as well as keeping the edge detection algorithm computationally regular. Although the resultant filters can potentially ensure this compromise, there is an essential challenge: parameter adjustment. The optimal filters staged for image processing usually have free parameter known as the scale. Tuning appropriate scale for the filter is a real challenge that leads to new area of study in edge detection.

As illustrated in Figure 5 or Figure 6, the smoothing and differentiation are realized by filtering the image with the differentiation of the smoothing filter. Then, the terms of filter and detector are often used synonymously. In the context of filtering, the filters are required to be described in both spatial and frequency domains. The attributes of a smoothing filter that influence the performance of the edge detector are its linearity, the duration of its impulse response, and its invariance to rotation. Non-linear filters are proven to be more successful than linear filters, because they can remove certain kinds of noise better (e.g., impulse noise) while preserving edge information [84], but linear filters are more common in edge detection because of simplicity. The duration of the impulse response characterizes the support of the filter in the spatial or frequency domains. For instance, in edge detection, three kinds of linear low-pass filters have been used: band-limited filters, support-limited filters and filters with minimal uncertainty. The invariance to rotation property ensures that the effect of smoothing is the same regardless of edge orientation [3].

In the edge detection context, Poggio and Torre [85, 86] show that regularizing differentiation can be accomplished by convolution of the image with the cubic spline (or its derivatives), with area controlled by a regularization parameter. In addition to the cubic spline, two other regularization filters have been proposed in [86], the Green function and the Gaussian

function, which later because of its prominent advantages has received much attention in literatures. Another way of regularizing the operation of numerical differentiation is through the approximation or interpolation of the data using analytic functions. Although these filters ensure a compromise between noise elimination and the preservation of image structure, we are faced with the problem of choosing the regularization parameter. This is important since judicious selection of this parameter reduces information loss. This problem will be presented in following Sections.

3.5 Edge Labelling

Edge labelling process involves localizing the edges and increasing signal-to-noise ratio by suppressing the false edges. The localization procedure depends on the applied differentiation operator. In search-based detectors which use the gradient, the edges are localized by applying a threshold to the gradient magnitude. Since the resulted edges in these methods are scattered, they require a post-processing that produces uniform edges. Non-maximum suppression (NMS) algorithm is a post-processing that can improve the performance of threshold based edge detectors. The basic idea is to extract local maxima of the gradient magnitude along the direction of the gradient vector. That is, if we consider the image plane as real, then a given pixel is a local maximum if the gradient magnitude at this pixel is greater than the gradient of two neighbouring points situated at the same distance on either side of the given pixel along the gradient direction [2]. In zero-crossing methods, edge labelling is performed by comparing the output of a second-order operator at a given pixel with the neighbour pixels to the left and below it. If these three pixels do not have the same signs, there is a zero-crossing. It is shown that the use of more than the two principal directions (horizontal and vertical) improves the localization, especially for certain junction edges [2].

3.6 Non-Maximum suppression

Non-maximum suppression is often used along with edge detection algorithms. The image is scanned along the image gradient direction, and if pixels are not part of the local maxima they are set to zero. This has the effect of suppressing all image information that is not part of local maxima.

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 angle is zero degrees the point will be considered to be on the edge if its intensity is greater than the intensities in the north and south directions, if the rounded angle is 90 degrees the point will be considered to be on the edge if its intensity is greater than the intensities in the west and east directions, if the rounded angle is 135 degrees the point will be considered to be on the edge if its intensity is greater than the intensities in the north east and south west directions, if the rounded angle is 45 degrees the point will be considered to be on the edge if its intensity is greater than the intensities in the north west and south east directions. This is worked out by passing a 3x3 grid over the intensity map. 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".

3.7 Hysteresis Algorithm

The origin of false edges in the output of a detector is not only limited to the noise. There are other phenomena resulting in rising the false edges, even in edge detectors having robust differentiation and smoothing algorithms. One of the significant reasons in producing false edges is the limitation that emerges due to selection of a fixed single threshold. In the threshold based (search based) method, the rule commonly used to classify edges as true or false is that the plausibility value of true and false edges is above and below a given threshold, respectively. The threshold is the minimum acceptable plausibility value. Due to regular variation of the minimum plausibility measure in an image, edges resulting from such a

binary decision rule could not ever be valid. Hence, hysteresis algorithm is taken into account to improve the edge continuity. Two thresholds are used; a given edge (e.g., an ordered list of edge points) is true if the plausibility value of every edge point on the list is above a low threshold and at least one is above a high threshold. Otherwise, the edge is false [2].

In zero-crossing method, the elimination of false edges is getting more complex, because a zero-crossing can correspond to a weak gradient magnitude (i.e., saddle point). The false zero-crossings can be emerged by either noise or certain edge types (e.g. staircase edges). The false edges caused by the noise, due to their low gradient magnitude, can be discarded using hysteresis algorithm. The false edges caused due to the certain model of edges, which are called phantom edges, can be discriminated using their gradient sign. Intuitively, they are zero-crossings of the second derivative which correspond to either the positive minima or negative maxima of the first derivative of the staircase [2]. In practice, there may be phantom edges whose gradient magnitude is greater than that of some true edges. In addition, a phantom edge usually forms a continuous curve which extends a curved edge. Therefore, the use of edge continuity as a criterion to eliminate phantom edges can't be appropriate way, since it implies the non-suppression of phantom edges. Phantom edges as minima of the gradient magnitude are the only edge points which verify the following condition.

Another aspect of the elimination of false edges concerns threshold computation. Usually, a threshold value is found using a trial-and-error process and is used for all edges of an image. However, it is pointed out that the threshold is a function of edge characteristics, properties of the smoothing filter, and properties of the differentiation operator. Consequently, it is not easy to find a single threshold value for a given image. An automatic rule to compute the threshold for the Laplacian of Gaussian detector has been proposed in [18]. This rule is empirical, no justification has been given and it has been tested only on synthetic data. It is also proposed a cleaning rule for multi-scale edge detection based on the behaviour of the ideal step edge in scale space. The threshold is found at a high scale and propagated automatically to ideal step edges obtained at lower scales. Improvements of this algorithm are proposed in [128] for use with any smoothing filter, differentiation operator, and edge model.

3.8 Sub-pixel Accuracy

Sub-pixel rendering is an image processing technique to increase the apparent resolution of an image. It takes advantage of the fact that each pixel is actually composed of individual sub-pixels with greater detail. Sub-pixel approach is used to improve the accuracy of localization in edge detection. As it was mentioned, edges can be localized using either the local maxima or zero-crossing methods, however, by applying interpolation on the pixels we can localize them more accurate.

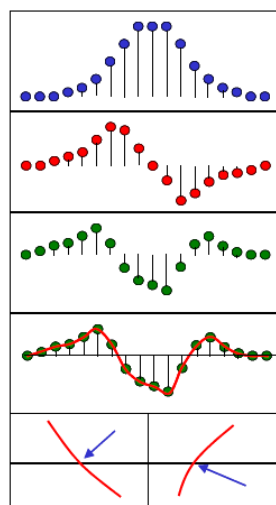


Figure 7 – interpolation on Sub-pixel methods

Figure 7 shows the different phases of the sub-pixel edge detection algorithm. The first figure (blue sample points) represents the intensity function corresponding to a line in an image. The aim is to determine the exact position of the edges of this line. The first step is to calculate the first order derivative of this function. This is shown in the second part of the figure (red sample points). The location of the edges is on the maximum and minimum of this derivative function. To determine these positions, the second order derivative is calculated (green sample points). The edges are now located on the zero-crossings of this function. To determine these positions very accurately, this function is interpolated, as shown in the fourth part of the drawing. Finally the blue arrows, point out the exact positions of the positive (=rising) and negative (=falling) edges.

4. Edge Detection Methods

This Section presents dominant works of edge detection methods in nine categories, and explains their advantages and disadvantages and the relationships among them. Some works, because of their importance and fundamental impacts, are described in more details, while some have been reviewed with just referring to their main contributions.

4.1 Classical Methods

Classical edge detectors have no smoothing filter, and they are only based on a discrete differential operator. The earliest popular works in this category include the algorithms developed by Sobel (1970), Prewitt (1970), Kirsch (1971), Robinson (1977), and Frei-Chen (1977). They compute an estimation of gradient for the pixels, and look for local maxima to localize step edges. Typically, they are simple in computation and capable to detect the edges and their orientation, but due to lack of smoothing stage, they are very sensitive to noise and inaccurate. To more simplicity, they often estimate the gradient magnitude through (26) rather than the popular definition in (3).

$$\left| \hat{\nabla} g(c, r) \right| = \left| \sum_{\alpha} g_{\alpha}(c, r) \right| \quad (26)$$

The Sobel operator is the most known among the classical methods. The Sobel edge detector applies 2D spatial gradient convolution operation on an image. It uses the convolution masks shown in (27) to compute the gradient in two directions (i.e. row and column orientations), and then works out the pixels' gradient through $g = /g_r + g_c/$. Finally, the gradient magnitude is thresholded. Sobel edge detector is a simple and effective approach, but sensitive to noise. Moreover, the detected edges are thick, which may not be suitable for applications that the detection of the outmost contour of an object is required.

$$H_c = \frac{1}{4} \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, H_r = \frac{1}{4} \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (27)$$

Though the classical methods were not in benefit of an independent smoothing module, they attempted to ease this drawback through the calculation of average over the image. This is because estimation of a derivative calculated using a relatively large set of neighbouring pixels is more robust to noise than using only two pixels. Following this idea, several operators have been proposed, some of them are an extended version of the 3x3 detectors mentioned earlier. For example, a mask adopted by the 7x7 Prewitt based operator to estimate the horizontal (column orientation) first order derivative of the image is presented as following.

$$H_c = \frac{1}{21} \begin{bmatrix} -1 & -1 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (28)$$

Right from the early stages of edge detection, it was recognized that we can use operators of several dimensions. Rosenfeld *et al.* [8]-[10] proposed an algorithm to detect edges, commonly known as “difference of boxes”, that relies on the use of pairs of neighbourhoods (one neighbourhood on each side of the point under analysis) of several dimensions and orientations. By convenience, they suggested that the neighbourhoods should have a square

shape and have sizes related to the powers of two. The output value of this operator is just the difference of the mean intensity values, calculated over the pair of neighbourhoods. Also, they indicated that one of the possible ways to find the “ideal” operator size is to look for the largest one that does not originate a significant decrease on the output value, when compared with the output value of the immediately smaller operator.

4.2 Gaussian Based Methods

Gaussian filters are the most widely used filters in image processing and extremely useful as detectors for edge detection. It is proven that they play a significant role in biological vision particularly in human vision system. Gaussian-based edge detectors are developed based on some physiological observations and important properties of the Gaussian function that enable to perform edge analysis in the scale space.

Marr and Hildreth [24][25] were the pioneers that proposed an edge detector based on Gaussian filter. Their method had been a very popular one, before Canny released his detector. They originally pointed out the fact that the variation of image intensity (i.e. edge) occurs at different levels. This implied the demand to smoothing filters with different scales, since a single filter cannot be optimal for all possible levels. They suggested the 2D Gaussian function, defined as following, as the smoothing operator.

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (29)$$

where σ is the standard deviation, and (x, y) are the Cartesian coordinates of the image pixels. They showed that by applying Gaussian filters of different scales (i.e. σ) to an image, a set of images with different levels of smoothing will be obtained.

$$\hat{g}(x, y, \sigma) = G_{\sigma}(x, y) * g(x, y) \quad (30)$$

Then, to detect the edges in these images they proposed to find the zero-crossings of their second derivatives. Marr and Hildreth achieved this by using Laplacian of Gaussian (LOG) function as filter. Since Laplacian is a scalar estimation of the second derivative, LOG is an orientation-independent filter (i.e. no information about the orientation) that breaks down at corners, curves, and at locations where image intensity function varies in a nonlinear manner along an edge. As a result, it can't detect edges at such positions. According to (24), the smoothing and differentiation operations can be implemented by a single operator consisting on the convolution of the image with the Laplacian of the Gaussian function. The final form of the filters, known as LOG with scale σ , which should be convolved with the image is as follows:

$$f_{\sigma}(x, y) = \nabla^2 G_{\sigma} = -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (31)$$

There are advantages for the Gaussian filter that make it unique and so important in edge detection. The first concerns to its output. It is proven that when an image is smoothed by a Gaussian filter, the existing zero-crossings (i.e. detected edges) disappear as moving from fine-to-coarse scale, but new ones are never created [26]. This unique property makes it possible to track zero-crossings (i.e. edges) over a range of scales, and also gives the ability to recover them at sufficiently small scales. Yuille and Poggio [26] proved that with the Laplacian, the Gaussian function is the only filter in a wide category that does not create zero-crossings as the scale increases. They also showed that for second derivatives, there is no filter that does not create zero-crossings as the scale increases. This implies the importance of combination made by Laplacian and Gaussian. Another issue concerns to the filters' conflicting goals of localization in spatial and frequency domains. The “optimal” smoothing filter should obey to two conditions: (1) the filter should be smooth in the frequency domain

and approximately frequency limited, to be able to reduce the scale range where intensity changes occur (well-posed); (2) the filter should be localized in the spatial domain, since the influence of a given point only makes sense in a relatively small neighbourhood. The LOG filter is a linear filter that is able to simultaneously minimize the (product of the) spread in space and frequency domains. At last, but not least, the 2D Gaussian filter is the only rotationally symmetric filter that is separable in Cartesian coordinates. This is important for computational efficiency when applying convolution in the spatial domain.

In spite of unique features of the Gaussian function, the filter proposed by Marr and Hildreth had some deficiencies related to using the zero-crossing approach. The zero-crossing approach is only reliable in locating edges if they are well separated and the signal-to-noise ratio (SNR) in the image is high. It is shown that for ideal step and ramp edges, the location of the zero-crossing is exactly at the location of the edge. However, the location shifts from the true edge location for the finite-width staircase steps. This shift is a function of the standard deviation of the Gaussian. The other problem is the detection of false edges. The reason is that zero-crossings correspond to local maxima and minima in the first derivative of an image function, whereas only local maxima indicate the presence of real edges. LOG filtered images also suffer from the problem of missing edges (i.e. edges in the original image may not have corresponding edges in a filtered image).

Moreover, combining LOG zero-crossings is a very difficult task, since a physically significant edge does not match a zero-crossing for more than a few and very limited number of scales, and zero-crossings in larger scales move very far away from the true edge position due to poor localization of the LOG operator, and finally there are too many zero-crossings in the small scales of a LOG filtered image, most of which is due to noise [5]. Relating image structures across scale is intrinsically problematic since, zero-crossings merge tangentially. Thus, one cannot rely upon zero-crossings to track edges. Edge-linking algorithms based on this approach will remain heuristic in nature. Furthermore, physiological experiments have found no evidence to support zero-crossings as a model for biological vision. Haralick [27] proposed the use of zero-crossing of the second directional derivative of the image intensity function. This is theoretically the same as using the maxima in the first directional derivatives, and in one dimension is the same as the LOG filter.

Later, in 80s, Canny [6][7] proposed a method that was widely considered to be the standard edge detection algorithm in the industry, and still it outperforms many of recent algorithms. In regard to regularization explained in image smoothing, Canny saw the edge detection as an optimization problem. He considered three criteria desired for any edge detector: good detection, good localization, and only one response to a single edge. Then he developed the optimal filter by maximizing the product of two expressions corresponding to two former criteria (i.e. good detection and localization) while keeping the expression corresponding to uniqueness of the response constant and equal to a pre-defined value. The solution (i.e. optimal filter) was a rather complex exponential function, which by variations it could be well approximated by first derivative of the Gaussian function. This implies the Gaussian function as the smoothing operator followed by the first derivative operator. Canny showed that for a 1D step edge the derived optimal filter can be approximated by the first derivative of a Gaussian function with variance σ as follow:

$$f_{\sigma}(x) = \frac{dG_{\sigma}(x)}{dx} = -k \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{\sigma^2}\right) \quad (32)$$

In 2D, Canny assumed the image affected by white noise, and proposed the use of two filters representing derivatives along the horizontal and vertical directions. In other word, the edge detection is performed through the calculation of derivative along two directions of the image filtered by Gaussian function. The separability feature of the 2D Gaussian function allows us to decompose it into two 1D filters.

$$f_{\sigma}(x, y) = [f_{\sigma}(x) * G_{\sigma}(y) \quad G_{\sigma}(x) * f_{\sigma}(y)] \quad (33)$$

where $G_{\sigma}(\cdot)$ and $f_{\sigma}(\cdot)$ denotes the 1D Gaussian function and its derivative, respectively, and $f_{\sigma}(\dots)$ denotes 2D optimal filter. The filter (33) shows that the filtering can be applied first to columns (rows) and then to rows (columns), reducing the computational burden. The optimal filter has rather an orientation perpendicular to the direction of the detected edge. The method proposed by Canny can be used for developing filters dedicated to a specific and arbitrary edge profile. For step edges, Canny's optimal filter is similar to the LOG operator because the maxima in the output of a first derivative operator correspond to the zero-crossings in the Laplacian operator used by Marr and Hildreth.

Canny also proposed a scheme for combining the outputs from different scales. His strategy is fine-to-coarse and the method is called feature synthesis. It starts by marking all the edges detected by the smallest operators. It then takes the edges marked by the small operator in a specific direction and convolves them with a Gaussian normal to the edge direction of this operator so as to synthesize the large operator outputs. It then compares the actual operator outputs to the synthesized outputs. Additional edges are marked if the large operator detects a significantly greater number of edges than what is predicted by the synthesis. This process is then repeated to mark the edges from the second smallest scale that were not marked by the first, and then to mark the edges from the third scale that were not marked by either of the first two, and so on. In this way, it is possible to include edges that occur at different scales even if they do not spatially coincide [5].

Canny's edge-detector looks for local maxima over the first derivative of the filtered image. It uses adaptive thresholding with hysteresis to eliminate streaking of edge contours. Two thresholds are involved, with the lower threshold being used for edge elements belonging to edge segments already having points above the higher threshold. The thresholds are set according to the amount of noise in the image, which is determined by a noise estimation procedure.

The problem with Canny's edge detection is that his algorithm marks a point as an edge if its amplitude is larger than that of its neighbours without checking that the differences between this point and its neighbours are higher than what is expected for random noise. His technique causes the algorithm to be slightly more sensitive to weak edges, but it also makes it more susceptible to spurious and unstable boundaries wherever there is an insignificant change in intensity (e.g., on smoothly shaded objects and on blurred boundaries) [5].

There are many contributions in the last two decades that present edge detectors using either directly the Gaussian function or filters with high similarity to the Gaussian function and its derivatives. This leads us to believe that the "optimum" linear filter for the detection of step edges should not differ too much from the derivative of the Gaussian and, therefore, the smoothing filter should be based on the Gaussian function. This is not surprising since the Gaussian has been emerging as a very important function in several areas of image analysis and processing and, specially, in multi-resolution analysis. Our goal is not to give an inventory of algorithms and merely review significant works that attempt to achieve high performance edge detectors using multi-resolution analysis.

4.3 Multi-Resolution Methods

Multi-resolution methods incorporate repeating edge detection for several scales of the Gaussian filter to achieve a quality performance. The main challenges in these methods includes selection a proper range for the scales, combination of the outputs corresponding to different scales, and adaptation to level of noise in the image. There are plenty publications in this area, we just content few sample works in this sub-section.

In [28], Schunck introduces an algorithm for the detection of step edges using Gaussian filters at multiple scales. The initial steps of Schunck's algorithm are based on Canny's method. The

algorithm begins by convolving an image with a Gaussian function. The gradient magnitude and gradient angle are then computed for each point in the resulting smoothed data array. Next, the gradient ridges in the results of the convolution are thinned using non maxima suppression (NMS). Then, the thinned gradient magnitudes are thresholded to produce the edge map. The gradient magnitude data at the largest scale will contain large ridges which correspond to the major edges in the image. As the scale decreases, the gradient magnitude data will contain an increasing number of ridges, both large and small. Some of these correspond to major edges, some to weaker edges, and the rest are due to noise and unwanted details. The gradient magnitudes over the chosen range of scales are multiplied to produce a composite magnitude image. Ridges that appear at the smallest scale and correspond to major edges will be reinforced by the ridges at larger scales. Those that do not will be attenuated by the absence of ridges at larger scales. Therefore, in the combined magnitude image, the ridges that correspond to major edges are much higher than the ridges that do not. NMS is then performed using sectors obtained from the gradient angle of the largest filter. Schunck's algorithm chooses the width of the smallest Gaussian filter, and the filters that are used differ in width by a factor of two. However, he did not discuss how to determine the number of filters to use. In addition, by choosing such a large size for the smallest filter, Schunck's technique loses a lot of important details which may exist at smaller scales [5].

Witkin [29] studied the property of zero-crossings across scales for 1D signal. He marked the zero-crossings of second derivative of a signal smoothed by Gaussian function in a range of scale, and then presented them versus scales. This representation known as the scale-space representation of a signal contains the location of a zero-crossing at all scales starting from the smallest scale to the scale at which it disappears. This work initiated the study of edge detection as a function of scale, and led to algorithms that combine edges for better edge detection.

Bergholm [30] proposed an algorithm which uses the Gaussian filter and combines edge information moving from a coarse-to-fine scale. His method is called edge focusing, and uses a rule-based approach for detecting local features and for tracking and predicting a possible scale parameter. Both the Marr-Hildreth and Canny edge-detectors are possible schemes that can be used in edge focusing. The image is first smoothed with a large scale Gaussian filter and then the edge detection process is performed using adaptive thresholding. Assuming that edge contours rarely move by more than two pixels for a unit change in the scale parameter, the exact location of the edges is determined by tracking them over decreasing scales. Therefore, the results from one scale of the edge-detector are used to predict the locations of edges in the next, smaller, scale. The idea behind edge focusing is to reverse the effect of the blurring caused by the Gaussian operator. Blurring is not a desired feature as it results in poor edge localization. It is simply used as a means of removing the noise and other unnecessary features. The most obvious way of undoing the blurring process is to start with edges detected at the coarse scale and gradually track or focus these edges back to their original locations in the fine scale [5].

There are several problems associated with edge focusing, the foremost being how to determine the starting and ending scales of the Gaussian filter. Bergholm suggests the range between 3 and 6 for the maximum scale, but did not specify a minimum scale. He also did not discuss in detail how to choose the threshold which is used at the coarsest level, and this is a parameter which is critical in determining how well the algorithm performs. If it is too high, a number of true edge points will be eliminated right from the start. If it is too low, the output of the edge focusing could be very noisy. In addition, since edge focusing is obtained at a finer resolution, some edges (i.e., the blurred ones, such as shadows) present a juggling effect at small scales. This is due to the splitting of a coarse edge into several finer edges, and tends to give rise to broken, discontinuous edges [5].

Lacroix [31] avoids the problem of splitting edges by tracking edges from a fine-to-coarse resolution. His algorithm detects edges using the Canny method of NMS of the magnitude of

the gradient in the gradient direction. His method then considers three scales: σ_0 , σ_1 , and σ_2 . The smallest scale, σ_0 , is the detection scale, and is the finest resolution at which a group of edges appears. The largest scale, σ_2 , is the blurring scale, and is the coarsest resolution at which the first appeared edges still remains. The intermediate resolution is computed as

$$\sigma_1 = \sigma_0 + \frac{\sigma_2 - \sigma_0}{3} \quad (34)$$

The first appeared edges are validated as long as they are local maxima in the Gaussian gradient and the two recent regions are homogeneous and significantly different from one another. Only validated edges are then tracked through the scales. Although Lacroix avoids the problem of splitting edges, he introduces the problem of localization error as it is the coarsest resolution that is used to determine the location of the edges. He also provides no explanation as to how to decide which scales are to be used and under what conditions.

Williams and Shah [32] devised a scheme to find edge contours using multiple scales. They analyzed the movement of edge points smoothed with a Gaussian operator of different sizes, and used this information to determine how to link edge points detected at different scales. Their method, following the lead of Canny, uses a gradient of Gaussian operator to determine gradient magnitude and direction, followed by non-maxima suppression (NMS) to identify ridges in the gradient map. Since the resulting ridges are often more than one pixel wide, the gradient maxima points are thinned and then linked using an algorithm which assigns weights based on four measures: noisiness, curvature, contour length, and gradient magnitude. The set of points having the highest average weight is chosen.

The algorithm extends to multi-resolution by convolving the image with the Gaussian filter at three scales: σ , 1.4σ , and 2σ . First, the best partial edge contours are found using the largest scale. Then, the next smaller scale is used, and the regions around the end points of the contours are examined to determine if there are possible edge points at the smaller scale having similar directions to the end points of the contours. The original algorithm is then carried out for each of these possible edge points, and the best are chosen as an extension to the original edge contour. The scale is decreased to the smallest scale, and the process is repeated. Although Williams and Shah specify the number of scales to be used and the relationship between these scales, they did not suggest the best way to choose the value of σ and under what conditions [5].

Goshtasby [33] proposes an algorithm that works on a modified scale-space representation of an image. The author creates a representation of an image by recording the signs of pixels (instead of the zero-crossings) after filtering with LOG operator. The advantage of such a representation over that of regular scale-space is that the new representation does not contain any disconnected arches. The scale-step size is determined adaptively using the image structure in the following manner. Results of convolution of an image at scales σ_1 and σ_2 are overlaid one on top of another. If more than two regions of the same sign fall on top of each other, the complete information on behaviour of edges between these two scales is lacking. Therefore, one must consider an intermediate scale between σ_1 and σ_2 . Otherwise, there is no new edge information between these two scales. This procedure makes a decision on step sizes as one goes along rather than choosing step sizes before the process starts. This also avoids the use of too many or too few scales. This is the crucial part of the algorithm. Once the scale-space image is constructed using the correct values for σ , tracking of edge from lowest to highest resolution is possible since there are no disconnected arches. The major problem with Goshtasby's edge focusing algorithm is the need for a considerable amount memory to store the three-dimensional (3D) edge images [5].

To avoid the common problems associated with integrating edges detected at multiple scales, Jeong and Kim [34] proposed a scheme which automatically determines the optimal scales for each pixel before detecting the final edge map. To find the optimal scales for a Gaussian filter, they define an energy function that quantitatively determines the usefulness of the possible

edge map. They approach the edge detection problem as finding the scale of the Gaussian filter which minimizes a predefined energy function. The parameter is chosen so that: 1) it is large at uniform intensity areas, thereby smoothing out random noise; 2) it is small at locations where the intensity changes significantly, thus retrieving edges accurately; 3) it does not change sharply from pixel to pixel, therefore avoiding broken edges due to random noise. Since the Jeong-Kim algorithm is designed to adaptively find the optimal scale of the Gaussian filter for every location in the image function, it can be easily incorporated into any Gaussian-based edge-detection technique. However, this algorithm does result in reduced performance when it comes to detecting straight lines in vertical or horizontal directions. The algorithm also has the disadvantage of low-speed performance [5].

Deng and Cahill [35] also use an adaptive Gaussian filtering algorithm for edge detection. Their method is based on adapting the variance of the Gaussian filter to the noise characteristics and the local variance of the image data. Based on observations of how the human eye perceives edges in different images, they concluded that in areas with sharp edges, the filter variance should be small to preserve the sharp edges and keep the distortion small. In smooth areas, the variance should be large so as to filter out noise. They proposed that the variance of a 1D Gaussian filter as follows:

$$\sigma^2(x) = \frac{k\sigma_n^2}{\sigma_f^2(x) + \sigma_n^2} \quad (35)$$

where k is scaling factor, σ_n is the noise variance, and σ_f local variance of the signal. The major drawback of this algorithm is that it assumes the noise is Gaussian with known variance. In practical situations, however, the noise variance has to be estimated. The algorithm is also very computationally intensive [5].

In [36], Bennamoun *et al.* present a hybrid detector that divides the tasks of edge localization and noise suppression between two sub-detectors. This detector is the combination of the outputs from the Gradient of Gaussian and Laplacian of Gaussian detectors. The hybrid detector performs better than both the first-order and second-order detectors alone, in terms of localization and noise removal. The authors extended the work to automatically determine the optimal scale and threshold, of the hybrid detector. They do this by deriving a cost function which maximizes the probability of detecting an edge for a signal and simultaneously minimizes the probability of detecting an edge in noise [5].

4.4 Nonlinear Methods

This sub-Section looks into edge-detectors that leave the linear territory in search of better performance. Nonlinear methods based on the Gaussian filter evolved as researchers discovered the relationship between the solution to the heat equation (in physics) and images convolved with Gaussian filter for a smoothing purpose. Consider a set of derived images, $g(x, y, \sigma)$, by convolving the original image with a Gaussian filter $G_\sigma(x, y)$ of variance σ in (30). The parameter σ corresponds to time in the heat equation, whereas in the context of image it refers to the scale. This one parameter family of derived images can be viewed as the solution of the heat equation. However, in the case of linear heat equation as diffusion eradicates noise, it also blurs the edges isotropically (i.e. invariant with respect to direction). To overcome this problem, Perona and Malik [37] proposed a scale space representation of an image based on anisotropic diffusion. In the mathematical context, this calls for nonlinear partial differential equations rather than the linear heat equation.

The essential idea here is to allow space variant blurring. This is achieved by making the diffusion coefficient in the heat equation a function of space and scale. The goal is to smooth within a region and keep the boundaries sharp. A high value for the diffusion constant within the region and a very small value (possibly 0) on the boundary can produce the desired effect. Specifically, the heat diffusion coefficient is allowed to vary across the image plane and is

made dependent upon the image gradient. This effectively leads to a spatially adaptive smoothing which tends to preserve the location of edges throughout the scale hierarchy. When the Perona-Malik equation is decomposed into a process across the edge and one perpendicular to it, it can be understood how smoothing and sharpening can be carried out at the same time. The entire process is a combination of forward and backward diffusion processes. However, backward diffusion is well known to be an ill-posed process where the solution, if it exists at all, is highly sensitive to even the slightest perturbations of the initial data. In the context of image processing, the main observed instability is the so-called stair casing effect, where a smoothed step edge evolves into piecewise almost linear segments which are separated by jumps. The extent of this effect is dictated by the process of discretization. This effect is observable for fine spatial discretization and for slowly varying ramp edges. Fortunately, under practical situations, this phenomenon is hardly observed. It is an experimental fact that reasonable discretizations of the Perona-Malik equation are rarely unstable. Fontaine and Basu [38] suggest the use of wavelets to solve the anisotropic diffusion equation, which will be discussed in the next sub-Section.

4.5 Wavelet Based Methods

As it was mentioned, analysing an image at different scales increases the accuracy and reliability of edge detection. Focusing on localized signal structures, e.g., edges, with a zooming procedure enables simultaneous analysis from a rough to a fine shape. Progressing between scales also simplifies the discrimination of edges versus textures. Because of having this ability, wavelet transform is an advantageous option for edge detection in different applications. Wavelet-based multi-resolution expansions provide compact representations of images with regions of low contrast separated by high-contrast edges. Additionally, the use of wavelets provides a way to estimate contrast value for edges on a space-varying basis in a local or global manner as needed.

In the context of image processing, wavelet transform (WT) is defined as the sum over the entire of rows and columns (i.e. spatial domain) of the image intensity function multiplied by scaled and shifted versions of the mother wavelet function. It results in coefficients that are function of the scale and shifts. In other word, WT maps the image into a space with two variables: scale and shift. The scale represents the function by compressing or stretching it, and denotes its features in frequency domain, while the shift corresponds to the translation of the wavelet function in the spatial domain (i.e. row or column). There is a correspondence between scale and frequency: a low scale shows the rapidly changing details of the intensity function with a high frequency, and a high scale illustrates slowly changing coarse features, with a low frequency. Therefore, WT acts as a “mathematical microscope”, in which one can monitor different parts of an image by just adjusting focus on scale. An important property of WT is its ability to focus on localized structures, e.g. edges, with a zooming procedure that progressively reduces the scale parameter. In this way, coarse and fine signal structures are simultaneously analysed at different scales.

Heric and Zazula [13] presented an edge detection algorithm using Haar wavelet transform. They chose Haar wavelet as the mother wavelet function, because it was orthogonal, compact and without spatial shifting in the transform space. By applying WT, they presented the intensity magnitude variation between adjacent intervals on a time-scale plane. Positive or negative peaks in time-scale representations were called modulus maxima. Their values indicated the edge slope and width. A significant difference within a short interval was presented as large maximum value. They linked the marked maxima over the time-scale plane and then applied an adaptive threshold for each scale to detect the edge maxima lines. The position of modulus maximum at the lowest detected scale determines the edge position. Considering an extended model of step edge formed mathematically by a slope function affected by noise, they proved the wavelet transform within the edge region was constant and solely dependent on edge slope and scale. They observed that edges' modulus maxima are

larger than noise modulus maxima and the influence of noise decreases with progressing toward higher scales, because Haar wavelets perform averaging.

Furthermore, Heric and Zazula [13] centred on edge continuity by deploying a priori knowledge about edges and applying computational models rather than a local detector. They proposed edge linkage into a contour line with signal registration in order to close edge discontinuities and calculate a confidence index for contour linkages. Registration is a procedure of searching spatial transformation from a source image into a target image with the intention to find best alignment between the two images. The success of alignment depends on a similarity measure which measures locally or globally the degree of similarity between source and target images. They expected two row or column pixels taken from adjacent image rows or columns are very similar. If an edge influences one of them it probably does the same in the other one, and vice versa. When trying to register such two pixels, a low amount of change (adjustment) is expected. If the opposite happens, the registered pixels apparently reflect different characteristics, for example, it is not likely that they contain linked pieces of the same edge. They used the sum of squared differences (SSD) as a measure for uniformity.

Shih and Tseng [14] combined a gradient-based edge detection and a wavelet based multi-scale edge tracking to extract edges. The proposed contextual filter detects edges from the finest scale gradient images and then, the edge tracker refines the detected edges on the multi-scale gradient images.

4.6 Statistical Methods

Konishi et al. [16] formulated the edge detection as a statistical inference. This statistical edge detection is data driven, unlike standard methods for edge detection which are model based. For any set of edge detection filters, they used pre-segmented images to learn the probability distributions of filter responses conditioned on whether they are evaluated on or off an edge. Edge detection is formulated as a discrimination task specified by a likelihood ratio test on the filter responses. This approach emphasizes the necessity of modelling the image background (the off-edges). They represented the conditional probability distributions non-parametrically and illustrated them on two different data sets of images. Multiple edges cues including multiple scales were combined by using their joint distributions. Hence, this cue combination is optimal in the statistical sense. They evaluated the effectiveness of different visual cues and showed that their approach gives quantitatively better results than the Canny edge detector when the image background contains significant clutter. In addition, it is a measure to determine the effectiveness of different edge cues and provides quantitative measures for the advantages of multilevel processing for the relative effectiveness of different detectors. Furthermore, they showed that the method can learn these conditional distributions on one data set and adapt them to the other with only slight degradation of performance without knowing the ground truth on the second data set. This shows that the results are not purely domain specific. They applied the same approach to the spatial grouping of edge cues and obtained analogies to non-maximal suppression and hysteresis.

Bezdek et al. [17] described edge detection as a composition of four steps: conditioning, feature extraction, blending, and scaling. They examined the role of geometry in determining good features for edge detection and in setting parameters for functions to blend the features. They found that: 1) statistical features such as the range and standard deviation of window intensities can be as effective as more traditional features such as estimates of digital gradients; 2) blending functions that are roughly concave near the origin of feature space can provide visually better edge images than traditional choices such as the city-block and Euclidean norms; 3) geometric considerations can be used to specify the parameters of generalized logistic functions and Takagi–Sugeno input–output systems that yield a rich variety of edge images; and 4) understanding the geometry of the feature extraction and

blending functions is the key to using models based on computational learning algorithms such as neural networks and fuzzy systems for edge detection.

Santis and Sinisgalli [19] proposed a statistical edge detection algorithm using a linear stochastic signal model derived from a physical image description. The presence of an edge was modelled as a sharp local variation of the gray-level mean value. In any pixel, the statistical model parameters were estimated by means of a Bayesian procedure. Then a hypothesis test, based on the likelihood ratio statistics, was adopted to mark a pixel as an edge point. The advantage of this technique is that it exploits the estimated local signal characteristics and does not require any overall thresholding procedure.

4.7 Machine Learning Based Methods

Wu et al. [20] introduced a fast multilevel fuzzy edge detection algorithm that realizes the fast and accurate detection of the edges from the blurry images. The algorithm first enhances the image contrast by means of the fast multilevel fuzzy enhancement (FMFE) algorithm using the simple transformation function based on two image thresholds. Second, the edges are extracted from the enhanced image by the two-stage edge detection operator that identifies the edge candidates based on the local characteristics of the image and then determines the true edge pixels using the edge detection operator based on the extreme of the gradient values. They demonstrated that the algorithm can extract the thin edges and remove the false edges from the image, which leads to its better performance than the Sobel operator, Canny operator, traditional fuzzy edge detection algorithm, and other multilevel fuzzy edge detection algorithms.

Lu et al. [9] proposed a fuzzy neural network system for edge detection and enhancement by recovering missing edges and eliminating false edges caused by noise. The algorithm was comprised of three stages, namely, adaptive fuzzification by fuzzifying the input patterns, edge detection by a three-layer feed forward fuzzy neural network, and edge enhancement by a modified Hopfield neural network. The typical sample patterns were first fuzzified and applied to train a fuzzy neural network. The trained network was able to determine the edge elements with eight orientations. Pixels having high edge membership were traced for further processing. Based on constraint satisfaction and the competitive mechanism, interconnections among neurons were determined in the Hopfield neural network. A criterion was provided to find the final stable result that contains the enhanced edge measurement.

Zheng et al. [10] presented an edge detection algorithm that employs estimations of image intensity derivatives produced by least square support vector machine (LS-SVM). In SVM, the underlying intensity function $g(r, c)$ of a small neighbourhood in an image can be approximated by a combination of a set of support vectors. They deployed a neighbourhood with size of 5x5 pixels, employed Gaussian radial basis function as the kernel of SVM, and presented an estimation of intensity function as following.

$$\hat{g}(r, c) = \sum_k \alpha_k \exp \left\{ -\left(|r - r_k|^2 + |c - c_k|^2 \right) / \sigma^2 \right\} + b \quad (36)$$

where α_k and b are the solution of a quadratic problem (QP). They applied the estimation of derivatives into the both gradient and zero crossing methods to locate the edge positions. They stated a performance near to the Canny method, but faster computation.

Bhandarkar et al. [21] presented a genetic algorithm (GA) based optimization technique for edge detection. The problem of edge detection was formulated as the choosing a minimum cost edge configuration. The edge configurations were illustrated as 2D genome with fitness values inversely proportional to their costs, and meanwhile, the two basic GA operators (i.e. crossover and mutation operators) were described in the context of the 2D genomes. The mutation operator that exploits knowledge of the local edge structure was shown to result in rapid convergence. The incorporation of meta-level operators and strategies such as the

elitism strategy, the engineered conditioning operator, and adaptation of mutation and crossover rates in the context of edge detection were discussed and shown to improve the convergence rate. They examined various combinations of meta-level operators on synthetic and natural images, and compared the performance of the GA technique with local search-based and simulated annealing-based approaches. They stated that the GA performs very well in terms of robustness to noise, rate of convergence and quality of the final edge image.

4.8 Contextual Methods

Yu and Chang [15] suggested an adaptive edge detection approach based on context analysis. The proposed approach uses the information from predictive error values produced by the gradient-adjusted predictor (GAP) to detect edges. GAP uses a context, which is a combination of the intensity values of already processed neighbouring pixels defined by a template, to produce the predictive values. The context in the casual template of GAP is used to analyze whether the current pixel is an edge point or not.

The GAP is a nonlinear predictor adopted by the context based, adaptive, lossless image codec (CALIC). It can make itself adapt to the intensity gradients near the pixel to be predicted. Figure 8 shows the casual template employed in GAP predictor. The template involves two previous scan lines of coded pixels. The neighbouring pixels of the current pixel used in the prediction process are shaded. Here, the notations i, j, k, l, m, n, o , and x represent not only the pixel values, but their locations as well. GAP uses g_h and g_v to compute the gradient of intensity near the current pixel in the horizontal direction and in the vertical direction, respectively. The g_h and g_v are calculated as

$$\begin{aligned} g_h &= |i - m| + |j - k| + |j - l| \\ g_v &= |i - k| + |j - n| + |l - o| \end{aligned} \quad (37)$$

The detection of the magnitude and orientation for an edge across the casual template is based on the difference of g_h and g_v . Finally, the gradient-adjusted prediction procedure that produces the predictive values is shown below. The experimental results indicate that both the visual evaluations and objective performance evaluations of the detected image in the proposed approach are superior to the edge detection of Sobel and Canny.

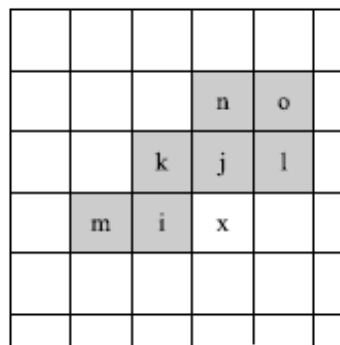


Figure 8 – casual template used by GAP [15]

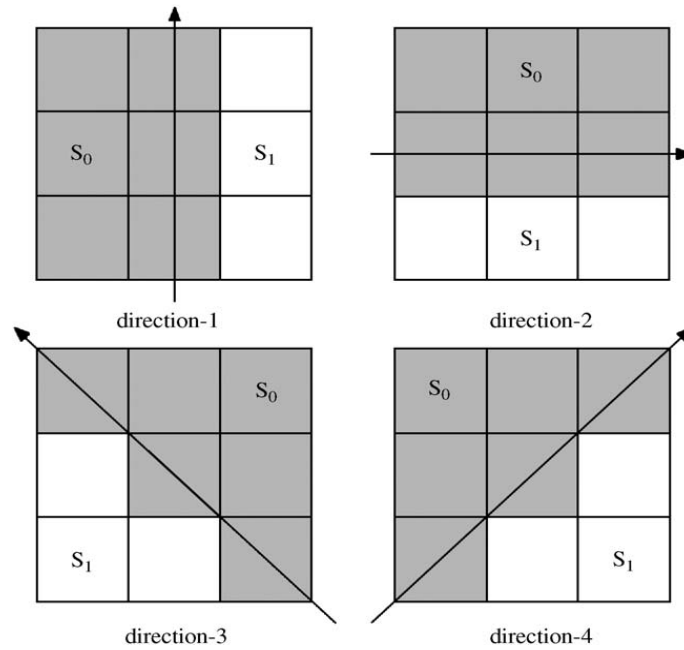


Figure 9 – four direction of two sets of pixels in 3x3 mask developed in [11]

Kang and Wang [11] developed an edge detection algorithm based on maximizing an object function. The values of the objective function corresponding to four directions determine the edge intensity and edge direction of each pixel in the mask. They normalized the image intensity function into certain levels, and used a 3x3 mask with two sets of pixels in four directions, as shown in Figure 9, to define an objective function. After all pixels in the image have been processed, the edge map and direction map are generated. Then, they applied the non-maxima suppression method to the edge map and the direction map to extract the edge points. The proposed method can detect the edge successfully, while double edges, thick edges, and speckles can be avoided.

Liang and Looney [12] introduced competitive fuzzy edge detection method. They adopted extended ellipsoidal Epanechnikov functions as a fuzzy set membership function, and a fuzzy classifier that differentiates image pixels into six classes consisting of background (no edge), speckle (noisy) edge, and four types of edges (in four directions) as shown in Figure 9.

Chang [22] employed a special design of neural networks for edge detection. He introduced a method called Contextual Hopfield Neural Network (CHNN) for finding the edges of medical CT and MRI images. Different from conventional 2D Hopfield neural networks, the CHNN maps the 2D Hopfield network at the original image plane. With the direct mapping, the network is capable of incorporating pixels' contextual information into a pixels' labelling procedure. As a result, the effect of tiny details or noises will be effectively removed by the CHNN and the drawback of disconnected fractions can be overcome. Furthermore, the problem of satisfying strong constraints can be alleviated and results in a fast converge. Our experimental results show that the CHNN can obtain more appropriate, more continued edge points than Laplacian-based, Marr-Hildreth's, Canny's, and wavelet-based methods.

Chao and Dhawan [23] presented an edge detection algorithm using Hopfield neural network. This algorithm brings up a concept which is different from those conventional differentiation operators, such as Sobel and Laplacian. In this algorithm, an image is considered a dynamic system which is completely depicted by an energy function. In other words, an image is described by a set of interconnected neurons. Every pixel in the image is represented by a neuron which is connected to all other neurons but not to itself. The weight of connection between two neurons is described as being a function of contrast of gray-level values and the

distance between pixels. The initial state of each neuron represents the normalized gray-level value of the corresponding pixel in the original image. As a result of Hopfield network analysis, output of neurons is modified until the convergence. Even though the outputs are analog, they are close to zero in all regions except edges where the corresponding neurons have near 1.0 output values. A robust threshold on the output level of the converged network can be easily set up at 0.5 level to extract edges. The experimental results are presented to show the effectiveness and capability of this algorithm.

In [39], the authors analyzed how the predictive distribution estimated using a set of context dependent nonlinear adaptive predictors can be used to localize edges in an images. Since the adaptive predictors have the potential of learning repetitive structure, as those characteristic to certain textures, the proposed predictive edge detection scheme can be a practical way to conceal the relative high contrast of certain texture regions.

Scargle and Quweider [40] proposed an edge detector based on finding major change points in a local 1D window of the image intensity values of the rows or columns. The approach amounts to separating the pixels in the window into sets or regions of constant intensities with the edge pixels providing transition points. The edge points are found based on partitioning the interval in an optimal way using dynamic programming with an appropriate cost function. Different cost functions are introduced for the algorithm with simulation results that show the detector's effectiveness even in the presence of noise.

4.9 Line edge detectors

As mentioned earlier, line edges correspond to local maxima of intensity function in the image and are of great use in the identification of image features, such as roads and rivers in remote sensing images, as well as the contactless paper counting. Most of line edge detectors are limited to thinning algorithms, and designed for binary images and a few for grey images. The main problem is that they usually yield edges which are not located accurately enough and they do not perform well in complex images such as remote sensing images.

Haralick [45] proposed an algorithm based on polynomial fitting. The image was fitted by a linear combination of discrete bases of Tchebychev's polynomial of order less than or equal to three. Lines occur at pixels having zero-crossings of the first directional derivative taken in the direction that maximizes the second directional derivative. Giraudon [46] proposed an algorithm for detecting a line at a negative local maximum of the second derivative of the image, rather than a zero-crossing of the first derivative. He estimated the second derivative by convolving the image with the difference of two Gaussians having close scales. The search for a negative maximum is performed along the gradient direction. The main problem with Giraudon's detector [46] comes from the use of the gradient since at the peak point, the gradient value is too small to be used. Using a 1D ideal roof model and Canny's criteria, Ziou [47] derived an optimal line detector. Koundinya and Chanda [48] proposed an algorithm based on combinatorial search. The basic idea behind this algorithm is to locate lines that maximize an ad-hoc confidence measure. The confidence measure of a candidate pixel is proportional to the number of pixels in its vicinity having a different grey intensity than the candidate pixel. They examined three strategies for combinatorial search: conventional tracking, best-first, and depth-first. According to the results, the best-first strategy seemed to provide more complete edges.

4.10 Coloured Edges' Methods

Koschan and Abidi [18] presented a review of techniques for the detection and classification of edges in colour images. While edge detection in gray-level images is a well-established area, edge detection in colour images has not received the same attention. The fundamental difference between colour images and gray-level images is that, in a colour image, a colour vector (which generally consists of three components) is assigned to a pixel. Thus, in colour image processing, vector-valued image functions are treated instead of scalar image functions.

The techniques used for this can be subdivided on the basis of their principle procedures into two classes: monochromatic-based techniques that treat information from the individual colour channels first separately then combine them together, and vector-valued techniques that treat the colour information as colour vectors in a vector space provided with a vector norm. They stated that the colour edge operators are able to detect more edges than gray-level edge operators. Thus, additional features can be obtained in colour images that may not be detected in gray-level images. However, it depends on the application whether these colour edge features are required. In addition to quantitative and qualitative advantages of colour edge detection, colour information allows for a classification of the edges. Such classification is not possible without the evaluation of colour information, and it can aid many image processing tasks which follow.

5. Discussion

So far, we have presented the key theoretical background of the edge detection, reviewed several methods, categorized them regarding to their paradigm and approach, studied their relationship, and stated evaluation regarding to their application, performance, and implementation. Given this relatively wide-ranging review, there is still a substantial question: “which method is the best?” It is obvious that according to each method’s optimization criteria and advantages each one shows better performance values than the others, but, that is not a fair way to compare them. Around this question there are some considerations that should be brought into discussion.

Except to Marr and Hildreth method, the filters are mostly selected in such way that provides optimum criteria in detection of edges with a certain model. The step edge model is, by far, the most used one, and basically imposes two conditions difficult to be met in real images: the transition is abrupt and the intensity is kept constant on both sides of the edge. Other types of edges, such as line, junction, staircase, and ramp edges, are required to be taken into account in edge detection and then final contours formed.

It seems that almost all the selected filters exhibit some similarity with the Gaussian function and its derivatives. This observation leads us to believe that the optimum smoothing filter should be based on the Gaussian function. This is not surprising since the Gaussian has been emerging as a very important function in several areas of image processing and, specially, in multi-resolution analysis [3].

The core of image differentiation is mainly based on discrete convolution that estimates image derivatives either by the gradient or Laplacian. The edges are localized either by local maxima on the gradient or zero-crossings on the Laplacian of image intensity function. Laplacian is a scalar estimation and has no information about the orientation of the edges. It breaks down at edges with the corners or curves profile and at locations with nonlinear intensity function. The localization by zero-crossing is problematic for edges with ramp or staircase profiles, because there is shift between the true and detected edge locations. Finally, zero-crossings suffer the false edges, since they correspond to local maxima and minima in the first derivative of an image function, whereas only local maxima indicate the presence of real edges.

The Gaussian filter has several desirable features which accounts for its wide use in many image processing applications. However, research clearly demonstrates that edge detection techniques involving this filter do not give satisfactory results. Linear methods suffer from problems associated with Gaussian filtering, namely, edge displacement, vanishing edges, and false edges. The introduction of multi-resolution analysis further complicates the issue by creating two major problems: 1) how to choose the size of the filters and 2) how to combine edge information from different resolutions. Adaptive approaches which avoid the multi-resolution problem all tend to be computationally intensive. Nonlinear approaches show significant improvement in edge detection and localization over linear methods. However, problems of computational speed, convergence, and difficulties associated with multi-resolution analysis remain. As it currently stands, use of the Gaussian filter requires making compromises when developing algorithms to give the best overall edge detection performance.

Edge detectors provide a set of edges potentially forming primitive objects, such as line, circles, arc and etc. This intermediate representation is crucial in the ultimate performance of a computer vision or image processing system, since if it performs well the objects in the scene can be recognised easily. Notwithstanding to huge research works, edge detectors still can not meet always the all requirements of many applications. They miss true edges, detect false edges, and localize the edge unsatisfying. These errors depend on image characteristics, detector properties, and implementation methods.

5.1 Methods of Evaluation

Evaluation of an edge detector, basically, requires criteria or references that describe the characteristics of the edges. The evaluation that uses certain model of edges is known as an objective evaluation, unless it would be subjective. The subjective evaluation consists of showing the detected edges to a human subject who rates the detector. While this technique seems easy, only a few characteristics (e.g., position, contrast, orientation) are visible to human. This evaluation is rough since it is difficult for human to distinguish between two close grey levels or two close orientations. Judgment by humans thoroughly depends on their experience, the image context, and properties of the detector. The human subject does not check whether the detector conforms to its initial specifications but rather whether perceived edges are detected. Subjective evaluations are vague and cannot be used to measure the performance of detectors but only to establish their failure.

Objective evaluation is performed using certain models of edges that should be detected. The goal of objective evaluation is to measure the performance of an edge detector, and it hardly can be applied into real life images. [41] has proposed a measure, which is a combination of three factors: non-detection of true edges, detection of false edges, and localization error. Using this measure it is difficult to determine the type of error committed by the detector. Kitchen and Rosenfeld's measure [42] combines errors that arise due to an edge's thickness and lack of continuity. Venkatesh and Kitchen [43] used four error types which reflect the major difficulties encountered in edge detection: non-detection of true edges, detection of false edges, detection of several edges instead of an edge one pixel wide, and edge localization error. The measures have been applied empirically to quantify the effect of edge characteristics such as contrast, noise, slope and width on various edge detectors. Subjective and objective evaluations can be used together to evaluate edge detectors. This combination inspired by psychological methods, is based on statistical analysis.

Heath et al. [44] proposed an evaluation method in the context of object recognition. Edge detector results were presented to subject humans who compare different edge detectors. The results were interpreted using the analysis of variance technique to establish the statistical significance of observed differences. They conducted two experiments. The first was an automatic evaluation of parameters of the edge detector. The ratings were analyzed statistically and the best parameters selected corresponding to each edge detector and image. In the second experiment, original images and corresponding edges were presented to sixteen judges for rating. The results obtained by both empirical and analytical evaluation processes have clarified the mutual influence between edge characteristics and detector properties. Finally, it was suggested that the evaluation methods should take into account model of the edges, specification of the detector and characteristics of the image.

5.2 Application: Contactless Paper Counting

Contactless paper counting is based on computer vision techniques, in which we discriminate paper edges in an image taken from the lateral view of a paper batch, and then count them without any direct contact or mechanical process which could lead to waist or deformation. The most significant criterion in this application, particularly in security sections such as cash counting, is the “*accuracy*”. This urges a performance with very high accuracy in both paper edge detection and counting, and any deficiency can lead to dismiss this application. This feature, in context of edge detection, magnifies two criteria referring to errors of missing true edges and detecting false edges, and dismisses the criterion that refers to error of localization.

Edge detectors that employ optimal filters minimising two criteria of non-detection of true edges and detection of false edges as well as using edge models with line profile can be recommended for the paper counting application. According to advantages of search based approaches, we suggest a modified version of Canny edge detector (described in Section IV, sub-Section 2) based on line edge profile. This method is capable to secure a maximum level of performance in edge detection and be evaluated objectively. Applied models of line edges

should be parametric and adjustable corresponding to papers' properties (i.e. thickness, constructing materials, illumination, and orientation). This enables us to deploy optimum filters in the edge detector corresponding to various types of papers.

An intelligent agent could be associated to select the best setting (i.e. scale) for the optimum filter using multi-resolution methods (described in Section IV, sub-Section 3) regarding to the paper profile. A feedback mechanism as well as a machine learning scheme can provide a support to ensure improving state for the paper counting machine.

Contextual methods, in which the edge detection is guided by a priori knowledge about the edges, are also recommended to improve the performance of the edge detector applied to paper counting application. They can perform accurately on certain contexts (e.g. paper counting) and adapt to varying conditions, such as papers' thickness and materials. Fuzzy method, Hopfield Neural Network, and gradient-adjusted predictor (described in Section IV, sub-Section 8), in particular, are contextual approaches that have already been tested and approved for the edge detection.

Moreover, some recent researches suggest nonlinear, wavelet, and statistical approaches for edge detection. Although, comparing with other applications in computer vision, the paper counting, typically, is not a very difficult problem, and even it can be considered a 1D problem, because of the zero-tolerance in erroneous detections, we suggest a case study that compares the performance of nonlinear, wavelet, and statistical approaches (described in Section IV, sub-Sections 4, 5, and 6) with Canny based detectors.

In conclusion, regarding to the importance of accuracy in paper counting, and thank to advanced and fast processors, commercially available in the market, we can deploy certain methods of edge detection in parallel, and then fuse their individual results to work out the final result. The idea of parallel edge detection can be extended to more images taken from different sides of the paper batches to achieve to a high rate of performance.

6. Conclusion

This manuscript is a review over the published articles on edge detection. At first, it provides theoretical background, and then reviews wide range of methods of edge detection in different categorizes. The review also studies the relationship between categories, and presents evaluations regarding to their application, performance, and implementation. It was stated that the edge detection methods structurally are a combination of image smoothing and image differentiation plus a post-processing for edge labelling. The image smoothing involves filters that reduce the noise, regularize the numerical computation, and provide a parametric representation of the image that works as a mathematical microscope to analyze it in different scales and increase the accuracy and reliability of edge detection. The image differentiation provides information of intensity transition in the image that is necessary to represent the position and strength of the edges and their orientation. The edge labelling calls for post-processing to suppress the false edges, link the dispread ones, and produce a uniform contour of objects.

References

- [1] http://en.wikipedia.org/wiki/Edge_detection
- [2] D. Ziou, S. Tabbone, “Edge Detection Techniques – An Overview”
- [3] Armando J. Pinho and Luís B. Almeida, A review on edge detection based on filtering and differentiation, *REVISTA DO DETUA*, VOL. 2, No1, 1997, pp. 113-126
- [4] W. K. Pratt. *Digital image processing*. Wiley-Interscience, 2nd edition, 1991.
- [5] Mitra Basu, “Gaussian-Based Edge-Detection Methods—A Survey”, *IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews*, Vol. 32, No. 3, August 2002, pp. 252-260
- [6] J. Canny, “Finding Edges and Lines”, MIT Technical Report No. 720, 1983.
- [7] J. Canny, “A Computational Approach to Edge Detection”, *IEEE Transaction on Pattern Analysis and Machine intelligence*, No. 6, pp. 679-698, 1986.
- [8] T. Hermosilla, E. Bermejo, A. Balaguer, and L.A. Ruiz, “Non-linear fourth-order image interpolation for subpixel edge detection and localization”, *Elsevier journal on Image and Vision Computing* 26 (2008) 1240–1248
- [9] S. Lu, Z. Wang, and J. Shen, “Neuro-fuzzy synergism to the intelligent system for edge detection and enhancement”, *Elsevier Journal of Pattern Recognition* 36 (2003) 2395 – 2409
- [10] S. Zheng, J. Liu, and J. W. Tian, “A new efficient SVM-based edge detection method”, *Elsevier Journal of Pattern Recognition Letters* 25 (2004) 1143–1154
- [11] C. Kang, and W. Wang, “A novel edge detection method based on the maximizing objective function”, *Elsevier Journal of Pattern Recognition* 40 (2007) 609 – 618
- [12] L. R. Liang, and C. G. Looney, “Competitive fuzzy edge detection”, *Elsevier Journal of Applied Soft Computing* 3 (2003) 123–137
- [13] D. Heric, and D. Zazula, “Combined edge detection using wavelet transform and signal registration”, *Elsevier Journal of Image and Vision Computing* 25 (2007) 652–662
- [14] M. Y. Shih, D. C. Tseng, “A wavelet based multi resolution edge detection and tracking”, *Elsevier Journal of Image and Vision Computing* 23 (2005) 441–451
- [15] Y. Yu, C. Chang, “A new edge detection approach based on image context analysis”, *Elsevier Journal of Image and Vision Computing* 24 (2006) 1090–1102
- [16] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, “Statistical Edge Detection: Learning and Evaluating Edge Cues”, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 25, No. 1, pp 57-74, 2003
- [17] J. C. Bezdek, R. Chandrasekhar, and Y. Attikiouzel, “A Geometric Approach to Edge Detection”, *IEEE Transactions On Fuzzy Systems*, Vol. 6, No. 1, pp 52-75, 1998
- [18] A. Koschan and M. Abidi, “Detection and Classification of Edges in Color Images”, *IEEE Signal Processing Magazine*, pp 64-73 JANUARY 2005
- [19] A. D. Santis and C. Sinisgalli, “A Bayesian Approach to Edge Detection in Noisy Images”, *IEEE Transactions On Circuits And Systems—I: Fundamental Theory And Applications*, Vol. 46, No. 6, pp 686-699, 1999
- [20] J. Wu, Z. Yin, and Y. Xiong, “The Fast Multilevel Fuzzy Edge Detection of Blurry Images”, *IEEE Signal Processing Letters*, Vol. 14, No. 5, pp 344-347, 2007
- [21] S. M. Bhandarkar, Y. Zhang, and W. D. Potter, “An Edge Detection Technique Using Genetic Algorithm-Based Optimization”, *Pattern Recognition*, Vol. 27, No. 9, pp. 1159-1180, 1994
- [22] C. Chang, “A contextual-based Hopfield neural network for medical image edge detection”, *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, Vol. 2, pp. 1011 – 1014, 2004
- [23] C. Chao and A. P. Dhawan, “Edge detection using Hopfield neural network”, *Proc. SPIE*, Vol. 2243, 242 (1994); doi:10.1117/12.169971
- [24] D. Marr and E. Hildreth. *Theory of edge detection*. Proc. Royal Society of London, B, 1980, 207, pp. 187–217.

- [25] R. Kasturi and R. C. Jain, eds. *Computer vision: principles*. IEEE Computer Society Press, Los Alamitos, CA, 1991.
- [26] A. L. Yuille and T. A. Poggio, "Scaling theorems for zero-crossings," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 15–25, Jan. 1986.
- [27] R. M. Haralick, "Digital step edges from zero-crossing of second directional derivatives," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 58–68, Jan. 1984.
- [28] B. G. Schunck, "Edge detection with Gaussian filters at multiple scales," in *Proc. IEEE Comp. Soc. Work. Comp. Vis.*, 1987, pp. 208–210.
- [29] A. P. Witkin, "Scale-space filtering," in *Proc. Int. Joint. Conf. Artificial Intelligence*, vol. 2, 1983, pp. 1019–1022.
- [30] F. Bergholm, "Edge focusing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 726–741, June 1987.
- [31] V. Lacroix, "The primary raster: A multiresolution image description," in *Proc. 10th Int. Conf. Pattern Recognition*, 1990, pp. 903–907.
- [32] D. J. Williams and M. Shah, "Edge contours using multiple scales," *Comput. Vis. Graph Image Process.*, vol. 51, pp. 256–274, 1990.
- [33] A. Goshtasby, "On edge focusing," *Image Vis. Comput.*, vol. 12, pp. 247–256, 1994.
- [34] H. Jeong and C. I. Kim, "Adaptive determination of filter scales for edge-detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 579–585, May 1992.
- [35] G. Deng and L.W. Cahill, "An adaptive Gaussian filter for noise reduction and edge detection," in *Proc. IEEE Nucl. Sci. Symp. Med. Im. Conf.*, 1994, pp. 1615–1619.
- [36] M. Bennamoun, B. Boashash, and J. Koo, "Optimal parameters for edge detection," in *Proc. IEEE Int. Conf. SMC*, vol. 2, 1995, pp. 1482–1488.
- [37] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 629–639, July 1990.
- [38] F. L. Fontaine and S. Basu, "A wavelet-based solution to anisotropic diffusion equation for edge detection," *Int. J. Imaging Sci. Technol.: Special Issue on Image Sequence Processing, Motion Estimation, and Compression of Image Sequences*, vol. 9, pp. 356–368, 1998.
- [39] B. Cramariuc, I. Tabug and M. Gabbouj, "Use Of Predictive Coding Distribution For Edge Detection",
- [40] J. D. Scargle and M. K. Quweider, "Edge Detection Using Dynamic Optimal Partitioning",
- [41] W.K. Pratt. *Digital Image Processing*. Wiley-Interscience Publication, 1978.
- [42] L. Kitchen and A. Rosenfeld. *Edge Evaluation Using Local Edge Coherence*. *IEEE Transactions Systems, man, and cybernetics*, SMC-11(9), 597-605, 1981.
- [43] S. Venkatesh and L. J. Kitchen. *Edge Evaluation Using Necessary Comp onents*. *CVGIP: Graphical Models and Image Processing*, 54(1), 23-30, 1992.
- [44] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer. *Comparison of Edge Detectors: A Methodology and Initial Study*. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, 143-148, 1996.
- [45] R.M. Haralick. *Ridge and Valley on Digital Images*. *Computer Vision, Graphics and Image Processing*, 22, 28-38, 1983.
- [46] G. Giraudon. *Edge Detection from Local Negative Maximum of Second Derivative*. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, 643-645, 1985.
- [47] D. Ziou. *Line Detection Using an Optimal IIR Filter*. *Pattern Recognition*, 24(6), 465-478, 1991.
- [48] K. Koundinya and B. Chanda. *Detecting Lines in Gray Level Images Using Search Techniques*. *Signal Processing*, 37, 287-299, 1994.