HYBRID EDGE DETECTION TECHNIQUE FOR DIGITAL IMAGES

Thesis submitted in partial fulfilment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

In

COMPUTER SCIENCE AND APPLICATIONS

Submitted by

Dinesh Vij

Roll No. 601003006

Under the supervision of

Mr. Singara Singh

Assistant Professor

SCHOOL OF MATHEMATICS AND COMPUTER APPLICATIONS

THAPAR UNIVERSITY

PATIALA - 147004

PUNJAB - INDIA

JUNE 2012

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Hybrid Edge Detection Technique for Digital Images*", in partial fulfilment of the requirements for the award of degree of Master of Technology in Computer Science and Applications submitted in School of Mathematics and Computer Applications, Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Mr. Singara Singh and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other university.

Dinesh Vij

(Dinesh Vij)
Roll No. 601003006
M. Tech. (CSA)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

29.6.12

(Singara Singh)
Assistant Professor
SMCA

Countersigned by

(Dr. S. S. Bhatia)
Professor and Head
SMCA, Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

Any fruitful effort in a new work needs a direction and guiding hands that show the way. The completion of this thesis has involved a lot of people to whom I would like to express my sincere thanks and gratitude for their help.

For most, I would like to express a deep sense of gratitude and thanks profusely to my thesis guide Mr. Singara Singh, Assistant Professor, School of Mathematics and Computer Applications, Thapar University, Patiala, for his sincere and invaluable guidance, suggestions and sympathetic attitude which inspired me in accomplishing this research work. Moreover, for mentoring me when I needed it the most. I am fortunate that I got an opportunity to work under his supervision.

I am thankful to other faculty members of School of Mathematics & Computer Applications, Thapar University, Patiala for their intellectual support.

Finally, I am thankful to all my family members and friends who constantly encouraged me to complete this study.

(Dinesh Vij)
Roll No. 601003006
M. Tech. (CSA)

Abstract

Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction, which aims at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities.

A kind of hybrid method using Sobel operator, Canny operator and our algorithm is proposed, which keeps their respective advantages. Experiments show that this technique improves the accuracy of edge detection. So, the final image contains a relatively complete edge profile. In practice, choosing a suitable method for image edge extraction is based on specific conditions. Different algorithms have their respective advantages and disadvantages. All kinds of algorithms exist flawed, so this hybrid method combines the advantages of different algorithms in order to obtain a better result.

Table of Contents

List of Figures

Chapter 1
Introduction

1.1 Definition of an Image

An image defined in the real world is considered to be a function of two real variables, for example, $a(x,y)$ with a as the amplitude (e.g. brightness) of the image at the real coordinate position (x,y). An image may be considered to contain sub-images sometimes referred to as regions of interest or simply regions.

An image is digitized to convert it to a form which can be stored in a computer's memory or on some form of storage media such as a hard disk. Once the image has been digitized, it can be operated upon by various image processing operations. The types of operations to transform an input image $a[m,n]$ into an output image $b[m,n]$ can be classified into three categories:

a) Point: the output value at a specific coordinate is dependent only on the input value at that same coordinate.

b) Local: the output value is dependent on the input values in the neighborhood of that coordinate.

c) Global: the output value is dependent on all the values in the input image.

We achieve different results when different types of operations are applied and the computational burden associated with a given operation is different.

A digital image $a[m,n]$ described in a two dimensional($2D$) discrete space is derived from an analog image $a(x,y)$ in a $2D$ continuous space through a sampling process that is frequently referred to as digitization. The $2D$ continuous image $a(x,y)$ is divided into N rows and M columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates $[m,n]$ with $\{m = 0,1,2,...,M-1\}$ and $\{n = 0,1,2,...,N-1\}$ is $a[m,n]$ [1]-[2].

1.2 Definition of an Edge

An edge is a set of connected pixels that lie on the boundary between two regions. Edges are pixels where image brightness changes abruptly. An edge is a property attached to an individual pixel and is calculated from the image function behavior in the neighborhood of the pixel. If a pixel's gray-level value is similar to those around it, there is probably not an edge at that point. If a pixel has neighbors with widely varying gray levels, it may present an edge point. Along the edge direction, pixel value changes more gently, while perpendicular to the edge direction, pixel value changes more dramatic [18]. 1st derivative tells us where an edge is, 2nd derivative can be used to show edge direction as shown in Figure 1.1.

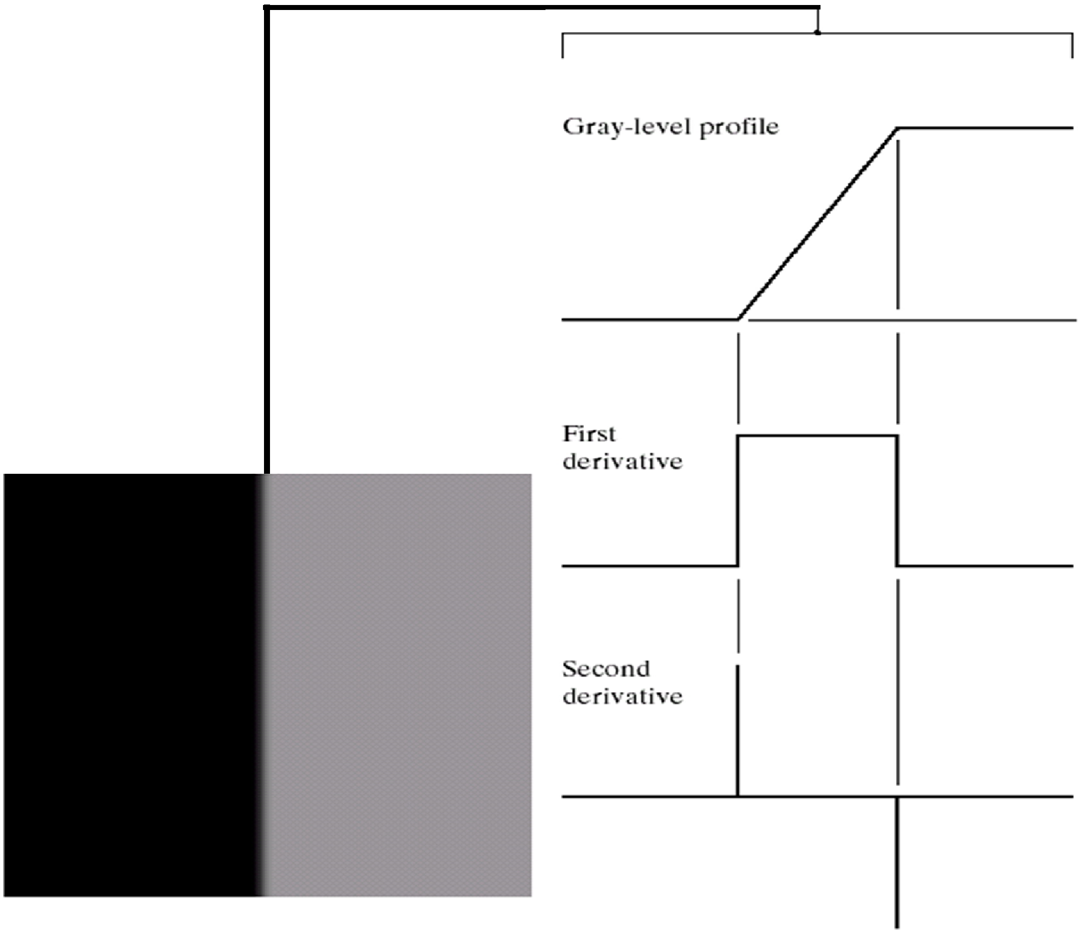


Figure 1.1 An image showing first derivative and second derivative.

1.3 Definition of Noise

Real world signals usually contain departures from the ideal signal. Such departures are referred to as noise. Noise arises as a result of unmodelled or unmodellable processes going on in the production and capture of the real signal.

A common form of noise is data drop-out noise (commonly referred to as intensity spikes, speckle or salt and pepper noise). Here, the noise is caused by errors in the data transmission. The corrupted pixels are either set to the maximum value (which looks like snow in the image) or in some cases, single pixels are set alternatively to zero or to the maximum value, giving the image a salt and pepper like appearance.

1.4 Digital filters

A filter is a device or a process that removes some unwanted components or features from a signal. In image processing filters are mainly used to suppress either the high frequencies in the image, *i.e.* smoothing the image, or the low frequencies, *i.e.* enhancing or detecting edges in the image. An image can be filtered either in the frequency or in the spatial domain. The first involves transforming the image into the frequency domain, multiplying it with the frequency filter function and re-transforming the result into the spatial domain. The filter function is shaped so as to attenuate some frequencies and enhance others. The corresponding process in the spatial domain is to convolve the input image $f(i,j)$ with the filter function $h(i,j)$.

Filters are usually implemented as moving window operations. The operator usually affects one pixel of the image at a time, changing its value by some function of a local region of pixels covered by the window. The operator moves over the image to affect all the pixels in the image. Some common types of moving window operations are:

a) Neighbourhood-averaging filters: These replace the value of each pixel, $a[i,j]$ say, by a weighted-average of the pixels in some neighbourhood around it, *i.e.* a weighted sum of $a[i+p,j+q]$, with $p = -k$ *to* k, $q = -k$ *to* k for some positive k; the weights are non-negative with the highest weight on the $p = q = 0$ term.

b) Mean filters: If all the weights are equal then this is a *mean* filter.

c) Median filters: This replaces each pixel value by the median of its neighbours, *i.e.* the value such that 50% of the values in the neighbourhood are above, and 50% are below. This can be difficult and costly to implement due to the need for sorting of the values. However, this method is generally very good at preserving edges.

d) Mode filters: Each pixel value is replaced by its most common neighbour. This is a particularly useful filter for classification procedures where each pixel corresponds to an object which must be placed into a class; in remote sensing, for example, each class could be some type of terrain, crop type, water, etc.

The above filters are all space invariant in that the same operation is applied to each pixel location. A non-space invariant filtering, using the above filters can be obtained by changing the type of filter or the weightings used for the pixels for different parts of the image.

Filters may also be used to remove high spatial frequency noise from a digital image. The filtering employed for this purpose is known as low pass filtering or smoothing. There are two main types of low pass filtering:

a) Reconstruction filtering: Here an image is restored based on some knowledge of the type of degradation it has undergone. Filters that do this are often called optimal filters.
b) Enhancement filtering: It attempts to improve the subjectively measured quality of an image for human or machine interpretability. Enhancement filters are generally heuristic and problem oriented [1].

1.5 Convolution

The discrete convolution can be defined as a shift and multiply operation, where we shift the kernel over the image and multiply its value with the corresponding pixel values of the image. Convolution provides a way of multiplying together two arrays of numbers to produce a third array of numbers. In an image processing context, one of the input arrays is normally just a gray level image. The second array is usually much smaller, and is also two dimensional (although it may be just a single pixel thick), and is known as the kernel.
The convolution is performed by sliding the kernel over the image, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries of the image as shown in Figure 1.2.

J11	J12	J13	J14	J15	J16	J17	J18	J19
J21	J22	J23	J24	J25	J26	J27	J28	J29
J31	J32	J33	J34	J35	J36	J37	J38	J39
J41	J42	J43	J44	J45	J46	J47	J48	J49
J51	J52	J53	J54	J55	J56	J57	J58	J59
J61	J62	J63	J64	J65	J66	J67	J68	J69

K11	K12	K13
K21	K22	K23

Figure 1.2 A 6×9 image and a 2×3 filter

The value of the bottom right pixel in the output image will be given by:

$$O_{57} = J_{57}K_{11} + J_{58}K_{12} + J_{59}K_{13} + J_{67}K_{21} + J_{68}K_{22} + J_{69}K_{23}$$

1.6 Thresholding

In many vision applications, it is useful to be able to separate out the regions of the image corresponding to objects in which we are interested, from the regions of the image that correspond to background. Thresholding often provides an easy and convenient way to perform this segmentation on the basis of the different intensities or colors in the foreground and background regions of an image. In addition, it is often useful to be able to see what areas of an image consist of pixels whose values lie within a specified range, or band of intensities (or colors).

The input to a thresholding operation is typically a grayscale or color image. In the simplest implementation, the output is a binary image representing the segmentation. Black pixels correspond to background and white pixels correspond to foreground (or vice versa). In simple implementations, the segmentation is determined by a single parameter known as the intensity threshold. In a single pass, each pixel in the image is compared with this threshold. If the pixel's intensity is higher than the threshold, the pixel is set to, say, white in the output. If it is less than the threshold, it is set to black. Threshold directly determines the success of edge detection. How to automatically get the best threshold of edge has been one of the difficulties of edge detection. The thresholded image g(x, y) is defined as

$$g(x,y) = \left\{ \begin{matrix} 1 & if \quad f(x,y) \geq T \\ 0 & if \quad f(x,y) < T \end{matrix} \right\}$$

Pixels labelled 1 correspond to objects, whereas pixels labelled 0 correspond to the background. If the selected threshold is too low, it will not only generate false edges, but edges are very thick; these edges will need to be refined again and the locations of the reprocessed edges are often not precise enough. If the threshold is too high, then many of the edges may not be detected or the detected edges are too segmented [15].

Depending on the scope of applications, the threshold methods can be divided into these types:

a) Overall threshold method: Also known as global threshold method. Here, the threshold is calculated based on pixel values of the entire image.

b) Local threshold method: Image is divided into sub-images and threshold of sub-images is calculated independently. It is also known as adaptive threshold method because different thresholds are used for different regions in the image.

Chapter-2

Different Types of Edge Detectors

A variety of edge detectors are available for detecting the edges in digital images. However, each detector has its own advantages and disadvantages. The basic idea behind edge detection is to find places in an image where the intensity changes rapidly. Based on this idea, an edge detector may either be based on the technique of locating the places where the first derivative of the intensity is greater in magnitude than a specified threshold or it may be based on the criterion to find places where the second derivative of the intensity has a zero crossing. The popular edge detection operators are Roberts, Sobel, Prewitt and Canny. In certain situations where the edges are highly directional, some edge detector works especially well because their patterns fit the edges better. The edge detection techniques are discussed in detail in the next subsections:

2.1 First Order Derivative methods / Gradient methods

Differential methods use approximation of spatial gradient at each pixel location. Denote *f(x,y)* to be the function that maps gray scale value of a particular pixel a_0 to its cartesian co-ordinates. Let *G(x,y)* be the rate of change in gray scale value at pixel a_0. *G(x,y)* can be computed in terms of the derivatives along *x* and *y* directions, $G_x(x,y)$ and $G_y(x,y)$

$$G(x,y) = \left\{ \left[G_x(x,y) \right]^2 + \left[G_y(x,y) \right]^2 \right\}^{\frac{1}{2}}$$

Although typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

These approximations still behave as derivatives, that is they are zeroes in areas of constant intensity and their values are proportional to the degrees of intensity change in areas whose pixel values are variables. It is common practice to refer to the magnitude of the gradient which is much faster to compute.


(a) Original image

(b) Horizontal gradient component


(c) Vertical gradient component

(d) Combined edge image

Figure 2.1 An image showing horizontal and vertical gradient components.

The direction of the gradient vector also is an important quantity. Let $\alpha(x, y)$ represent the direction angle of the vector *G(x,y)* at *(x,y)*.

$$\alpha(x, y) = \tan^{-1} \frac{G_y(x, y)}{G_x(x, y)}$$

The direction of the edge at location *(x,y)* is perpendicular to the gradient vector at that point. Computation of the gradient of an image is based on obtaining the partial derivatives $G_x(x, y)$ and $G_y(x, y)$ at every pixel location. At the point of greatest slope, the first derivative has maximum value, e.g. for a continuous 1-dimensional function *f(t)* as shown in Figure 2.2:

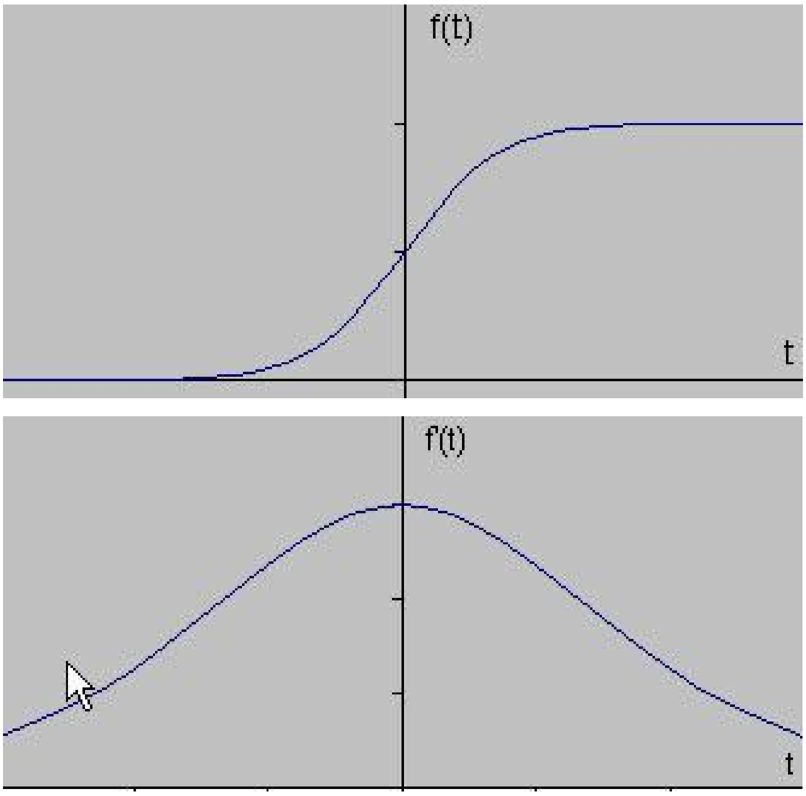


Figure 2.2 An image and its first derivative.

2.1.1 Roberts Operator

Provides an approximation to the gradient

$$G[f(i,j)] = |Gx| + |Gy| = |f(i,j) - f(i+1,j+1)| + |f(i+1,j) - f(i,j+1)|$$

The filters of this operator are small, thus the position of the edges is more accurate. On the other hand, the short support of the filters makes it very vulnerable to noise. Its effect is the best for the image with steep low noise. Works best with binary images. Roberts operator misses a few edges. Convolution masks used by Roberts operator are:

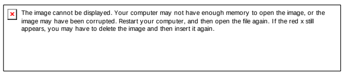
Gx=

1	0
0	-1

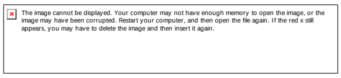
Gy=

0	1
-1	0

The kernels can be convolved separately with the input image, to produce separate measurements of the gradient component in each orientation. These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:



Although typically, an approximate magnitude is computed using:



These approximations still behave as derivatives, that is they are zeroes in areas of constant intensity and their values are proportional to the degrees of intensity change in areas whose pixel values are variables. It is common practice to refer to the magnitude of the gradient which is much faster to compute.

This detector is used considerably less than the others due to its limited functionality (e.g., it is not symmetric and cannot be generalized to detect edges that are multiple of 45^0).

2.1.2 Sobel Operator

Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. In order to suppress noise, a certain weight is correspondingly increased on the centre point. The 3×3 convolution mask smoothes the image by some amount, hence it is less susceptible to noise. But it produces thicker edges, so edge localization is poor. It is a method of combining the local average and partial differential [17]. The differences are calculated at the center pixel of the mask. It fails to detect true weak edges that are connected to strong edges. Convolution masks used by Sobel operator are:

Gx=

-1	0	1
-2	0	2
-1	0	1

Gy=

1	2	1
0	0	0
-1	-2	-1

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels are separately convolved with the input image, to produce separate measurements of the gradient component in each orientation. These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Typically, an approximate magnitude is computed using:

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

These approximations still behave as derivatives, that is they are zeroes in areas of constant intensity and their values are proportional to the degrees of intensity change in areas whose pixel values are variables. It is common practice to refer to the magnitude of the gradient which is much faster to compute. Then, we say that a pixel at location *(x,y)* is an edge pixel if *G>=T* at that location, where *T* is a specific threshold.

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \tan^{-1}(\frac{G_y}{G_x})$$

2.1.3 Prewitt Operator

It is similar to the Sobel operator but uses slightly different masks. Prewitt has a good processing result for the gray gradient images. Convolution masks used by Prewitt operator [3] are:

Gx=

-1	0	1
-1	0	1
-1	0	1

Gy=

1	1	1
0	0	0
-1	-1	-1

2.2 Second Order Derivative methods

Zero crossing of the second derivative of a function indicates the presence of a maxima as shown in Figure 2.3.

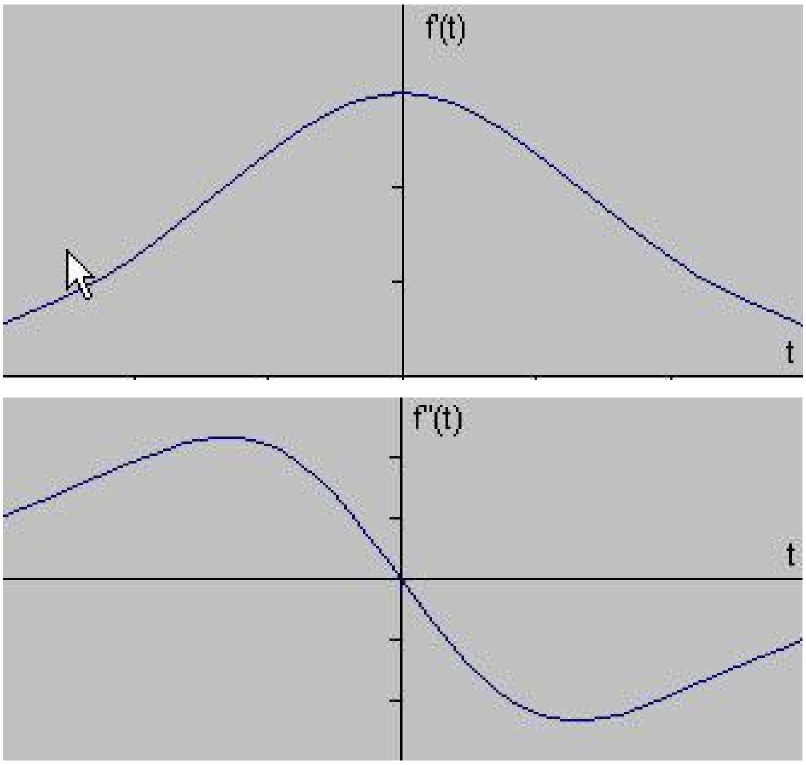
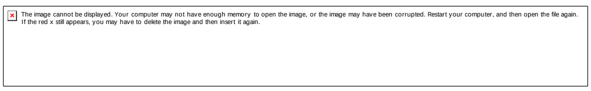


Figure 2.3 First derivative of an image and its corresponding second derivative

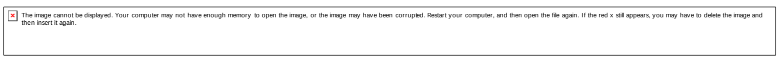
2.2.1 Laplacian Edge Detection

Gradient operation is an effective detector for sharp edges where the pixel gray levels change over space very rapidly. But when the gray levels change slowly from dark to bright, the gradient operation will produce a very wide edge. It is helpful in this case to consider using the Laplace operation. The Laplacian method searches for zero crossings in the second derivative of the image to find edges.



$$\frac{\partial^2 f}{\partial y^2} = f(i+1, j) - 2f(i, j) + f(i-1, j)$$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$



The Laplacian can be implemented using the mask shown below:

	0	-1	0
Gx=	-1	4	-1
	0	-1	0

	-1	-1	-1
Gy=	-1	8	-1
	-1	-1	-1

The Laplacian is typically not used by itself as it is too sensitive to noise. Sometimes we are interested only in edge magnitudes without regard to their orientations, the Laplacian may be used. The Laplacian has the same properties in all directions and is therefore invariant to rotation in the image. Edge magnitude is approximated in digital images by a convolution sum. The sign of the result (+ or -) from two adjacent pixels provide edge orientation and tells us which side of edge brighter.

2.2.2 Laplacian of Gaussian (LoG)

Also called Marr-Hildreth edge detector or Mexican hat operator. This uses the following steps:

a) Smooth the image using Gaussian filter.

b) Enhance the edges using Laplacian operator.

c) Zero crossings denote the edge location.

d) Use linear interpolation to determine the sub-pixel location of the edge.

Defined as $$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where σ is standard deviation, x and y are pixel co-ordinates.

Greater the value of σ, broader is the Gaussian filter, more is the smoothing. Too much smoothing may make the detection of edges difficult. Detection improves with width; localization degrades with increase in width. Probability of false and missing edges remains. Localization is better than gradient operators. Thus, we see that the purpose of the Gaussian formulation is to smooth the image, and the purpose of the Laplacian operator is to provide an image with zero-crossings used to establish the location of edges. Smoothing the image reduces the effect of noise caused by the second derivative of the Laplacian.

2.2.3 Difference of Gaussian (DoG)

LoG requires large computation time for a large edge detector mask. To reduce computational requirements, approximate the *LoG* by the difference of two *LoG* - the *DoG*.

$$DoG(x,y) = \frac{e^{-(\frac{x2+y2}{2\pi\sigma_1^{\,2}})}}{2\pi\sigma_1^{\,2}} - \frac{e^{-(\frac{x2+y2}{2\pi\sigma_2^{\,2}})}}{2\pi\sigma_2^{\,2}}$$

Close approximation of *LoG*. Less computation effort. Width of edge can be adjusted by changing $\sigma 1$ and $\sigma 2$.

2.3 Optimal Edge Detector

Canny Edge Detector:

The Canny edge detector [5] is regarded as one of the best edge detectors currently in use. The basic algorithm can be outlined as follows:

a) Blur the image slightly. This removes the effects of random black or white pixels (*i.e.* noise pixels) that can adversely affect the edge detection procedure. Blurring is typically accomplished by using a Gaussian blur.

The size and the standard deviation are two important criteria that need to be taken into consideration when creating the Gaussian mask. Larger sizes of the Gaussian mask will mean that the edge detection will be less susceptible to noise; however, larger mask also leads to more calculations and therefore a slower performance. The mask only needs to represent the first three standard deviations of the Gaussian curve. If the standard deviation is too small, the three standard deviation range will not be accurately represented. If it is too big, extra calculations with little contribution will be performed.

b) Apply Sobel algorithm to find the gradient of the image in *x* and *y* directions. Then find magnitude of the gradient by using the formula:

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

c) Find the edge direction by using the formula:

$$\theta = \tan^{-1}(\frac{G_y}{G_x})$$

d) Edge thinning is performed by computing gradient magnitude in four possible directions- 0^0 (in the horizontal direction), 45^0 (along the positive diagonal), 90^0 (in the vertical direction), or 135^0 (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Therefore, any edge direction falling within the range (0 to 22.5 and 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the range (112.5 to 157.5 degrees) is set to 135 degrees.

e) Non-maximal suppression: Edges will occur at points where the gradient is at a maximum. Therefore, all points not at a maximum should be suppressed. In order to do this, the magnitude and direction of the gradient is computed at each pixel. Then for each pixel check if the magnitude of the gradient is greater at one pixel's distance away in either the positive or the negative direction perpendicular to the gradient. If the pixel is not greater than both, suppress it [12].

f) Once the edges have been thinned, the last step is to threshold the detected edges. Edge magnitudes above the upper threshold are preserved. Edge magnitudes below the upper threshold but above the lower threshold are preserved only if they connect to edges that are above the upper threshold. And edge magnitudes below the lower threshold are discarded.

Typical ratio of thresholds is roughly

$k_{high} / k_{low} = 2$

This process is known as hysteresis and allows edges to grow larger than they would by using a single threshold without introducing more noise into the resulting edge image.

Chapter - 3

Literature Review

Kuo *et al.* [6] proposed a Fuzzy Sobel method for detecting object edges. The fuzzy logic methodology is applied to extract feature value for an image and Sugeno-type fuzzy reasoning strategy is adopted for edge enhancement. Such a method can improve the drawbacks of conventional approaches, for instance, the Prewitt and Sobel methods. In the two traditional methods, they use fixed parameters for all kinds of images so that they might success in one image but fail in another one. Fuzzy Sobel method, on the other hand, can find four threshold values automatically and then use them to construct membership functions for each detected image. Therefore, Fuzzy Sobel method can determine the parameters adaptively for different images. The edges detected in Sobel method are often vague, but Fuzzy Sobel method can enhance edges and makes the resultant image clear.

Ahmad *et al.* [7] proposed a new method to detect edges present in an image. Firstly, the actual gray level image is locally thresholded using local mean value to make a binary image. This binary image is checked for edges by comparing with the known edge like patterns, utilizing the Boolean algebra. This approach recognizes nearly all, real edges and edges due to noise. For removing edges due to noise, the actual image is globally thresholded by variance value of the image, based on the noise level in the image and second binary image is produced which contains only true edges. To obtain the final edge map, logical AND operation is performed on the two resulting images.

El-Khamy *et al.* [8] proposed a modified Fuzzy Sobel edge detector which automatically obtains four threshold values, and then applies fuzzy reasoning for edge enhancement. The Prewitt and Sobel methods have two drawbacks. First, the edges extracted from the image are not very clear and tiny changes may not be detected. Second, a threshold must be used to obtain a result. The choice of the threshold value is left to experience or try-and-error iterations to get a better result. The proposed method overcomes the drawbacks of Prewitt and Sobel methods. In this method, a difference histogram is constructed from the input image and then threshold values are determined from this difference histogram. After determining threshold, fuzzy regions are constructed and Sobel gradient of the image is found. At last, fuzzy reasoning is applied and edge detected image is obtained.

Zhang *et al.* [9] proposed a ratio of gray levels between two successive image points to denote the variation of gray levels. Furthermore, this paper defines a ratio and an integer logarithm ratio of gray levels. Based on the integer logarithm ratio of gray levels, a new edge detection method is proposed. The edge detection methods based on difference operation are sensitive to noise. The experiment results have shown that the effectiveness of edge detection and the ability of noise rejection of the proposed edge detection method are better than that of the traditional ones based on the difference operation.

Wang *et al.* [10] proposed an improved template algorithm, which includes the first order partial finite differences of directions 45 and 135 degree in calculating the amplitude values. This improves the calculation accuracy of the amplitude values. In the non-maxima suppression process, the factor ratio of four quadrants of linear interpolation is improved to achieve better detection results. Experiments show that this improved Canny algorithm has better noise suppression and edge continuity. The proposed algorithm is an effective, real-time detection algorithm.

Cui *et al.* [11] discussed several digital image processing techniques applied in edge feature extraction. Firstly, wavelet transform is used to remove noises from the image. Secondly, some edge detection operators such as Differential edge detection, Log edge detection, Canny edge detection and Binary morphology are analyzed. And then according to the simulation results, the advantages and disadvantages of these edge detection operators are compared. It is shown that the Binary morphology operator can obtain better edge feature. Finally, in order to gain clear and integral image profile, the method of bordering closed is given.

Nadernejad *et al.* [12] compared several techniques for edge detection in image processing. The Boolean edge detector performs similar to the Canny edge detector even though they both use different approaches. Canny method is still preferred since it produces single pixel thick, continuous edges. The Boolean edge detector's edges are often spotty. Color edge detection seems like it should be able to outperform grayscale edge detectors since it has more information about the image. In the case of the Canny color edge detector, it usually finds more edges than the grayscale version. The Euclidian Distance/Vector Angle detector identifies the borders between image regions, but misses fine grained detail. Multi-flash edge detection strives to produce photographs that will be easy to edge detect, rather than running on an arbitrary image. One problem inherent to the Multi-flash edge detector is that it will have difficulty in finding edges between objects that are at almost the same depth or are at depths which are very far away.

Maini *et al.* [13] presented a comparative analysis of various image edge detection techniques. Analysis shows that the Canny's edge detection algorithm performs better than other operators under almost all scenarios. Evaluation of the images shows that under noisy conditions Canny, LoG, Robert, Prewitt, Sobel exhibit better performance, respectively. It has been observed that Canny's edge detection algorithm is computationally more expensive as compared to LoG, Sobel, Prewitt and Robert's operator. Gradient based algorithms such as the Prewitt filter have a major drawback of being very sensitive to noise. The size of the kernel filter and coefficients are fixed and cannot be adapted to a given image. An adaptive edge detection algorithm is necessary to provide a robust solution that is adaptable to the varying noise levels of these images to help distinguish valid image contents from visual artifacts introduced by noise. The performance of the Canny algorithm depends heavily on the adjustable parameters: σ, which is the standard deviation for the Gaussian filter, and the threshold values, *t1* and *t2*.

Wang *et al.* [14] proposed an improved Canny algorithm. In this algorithm, self-adaptive filter is used to replace the Gaussian filter, morphological thinning is adopted to thin the edges and morphological operator is used to refine the edge points detected. The results of experiment prove that proposed approach gives better result than traditional Canny method. But, the improved algorithm has the problem of heavy calculation, which needs to be further improved.

Jin-Yu *et al.* [15] discussed Sobel edge detection operator and its improved algorithm using optimal thresholding. Then based on Genetic algorithms and improved Sobel operator, a new automatic threshold algorithm is proposed. Comparative experiment results show that the calculation speed and anti-noise capability of the new algorithm gets stronger.

Sim *et al.* [16] presented a new hybrid edge detector that combines the advantages of Prewitt, Sobel and optimized Canny edge detectors to perform edge detection while eliminating their limitations. Compared to the other three edge detection techniques, the hybrid edge detector has demonstrated its superiority by returning specific edges with less noise. When the Gaussian white noise in the original image increases, the Gaussian filter of the optimized Canny edge detector tends to increase the smoothing area of the Gaussian magnitude. As Gaussian standard deviation increases, the area of Gaussian gradient increases and it tends to overlap with another set of gradient image beside it. This proves that the optimized Canny edge detector has the possibility of being unable to detect some edges when image noise increases. In order to overcome this problem, a hybrid edge detector that is able to detect the optimum edges even in noisy images is formulated. The main component, which is the optimized Canny edge detector must be used to form the basic edge structure. Sobel and Prewitt edge detector components will detect edges that are close to each other.

Zheng *et al.* [17] reviewed mathematical morphology method and several classical edge detection operators. And finally this paper provides two methods: Canny operator and mathematical morphology, which summaries relatively good image edge detection methods.

Ma *et al.* [18] combined Sobel operator and median filtering to remove the salt and pepper noise in the image. The reason is that median filtering has a fine result when filtering out jump signal. The stochastic signal of salt and pepper noise is caused by the mutation of continuous signal, so the combine between this method and Sobel operator can better extract the edge of the image with salt and pepper noise signal. This new algorithm can effectively eliminate the peak-pulse and high frequency noise signal, and keep down the salt and pepper noise.

Deng *et al.* [19] proposed a modified Sobel edge detector, in which firstly the morphological filters are used to remove noise present in an image. Image is then sharpened by using Sobel operator. Then by using Otsu threshold method [4], improved Sobel operator is constructed. Then by making use of fusion technology, a kind of method combined with improved Sobel operator, wavelet transform, Canny algorithm and Prewitt operator is formulated, which keeps their respective advantages. In this way, edge extraction image contains a relatively complete profile and rich detailed information, effectively improves the accuracy of edge detection, and gets a quite ideal edge detection effect.

Jie *et al.* [20] described the conventional Canny in details, analyzes its drawbacks, and proposes the improved adaptive threshold Canny algorithm. This algorithm applies bilateral filtering to smooth the image, which suppresses noise in the image and preserves its edges. Then it uses Otsu, which is based on gradient magnitude to maximize the separability of the resultant classes, to adaptively determine the low and high thresholds of the Canny operator.

Chapter-4

Problem Statement

In real world machine vision problems, issues such as noise and variable scene illumination make edge and object detection difficult. There exists no universal edge detection method which works under all conditions. Effective edge detection is required for many important areas and is often used as the front end processing stage in object recognition and interpretation systems. Traditional image edge detectors commonly extract edges by adopting specific templates, or in combination with smoothing functions. However, traditional edge filtering methods often result in some drawbacks like broken edges, thick edges and false edges. In this thesis, we first proposed a new algorithm to detect edges. Experimental results show that the edges detected by this algorithm have a relatively complete edge profile than detected by traditional methods like Sobel, Roberts, Prewitt and Canny. Then using the fusion technique [19], a kind of hybrid algorithm is proposed using Sobel operator, Canny operator and our proposed new algorithm, which keeps their respective advantages. The visual comparison of the results of the proposed algorithms with the results of the already existing algorithms shows the effectiveness of the proposed algorithms. The proposed algorithms are explained in the next section:

Chapter-5
Proposed Solution

5.1 Proposed New Algorithm

The proposed new algorithm mainly consists of the steps explained below:

a) Make the 7×7 Gaussian filters phi_x and phi_y using the Gaussian equation:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels are convolved separately with the input image, to produce separate measurements of the gradient component in each orientation.

b) Apply above filters to find the gradient of the image in x and y directions. Then find magnitude of the gradient by using the formula:

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

c) The angle of orientation of the edge (relative to the pixel grid) is given by:

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

d) Non-maximal suppression: Edges will occur at points where the gradient is at a maximum. Therefore, all points not at a maximum should be suppressed. In order to do this, the magnitude and direction of the gradient is computed at each pixel. Then for each pixel check if the magnitude of the gradient is greater at one pixel's distance away in either the positive or the negative direction perpendicular to the gradient. If the pixel is not greater than both, suppress it [12].

e) Use Hysteresis based thresholding (uses two thresholds):

Take mean of the edge image and multiply it by 2 to get the high threshold.

Use k_{high} to find strong edges to start edge chain. Use k_{low} to find weak edges which continue edge chain. Typical ratio of thresholds is roughly:

$k_{high} / k_{low} = 2$

Edge magnitudes above the upper threshold are preserved. Edge magnitudes below the upper threshold but above the lower threshold are preserved only if they connect to edges that are above the upper threshold. And edge magnitudes below the lower threshold are discarded.

This process is known as hysteresis and allows edges to grow larger than they would by using a single threshold without introducing more noise into the resulting edge image.

5.2 Proposed Hybrid Algorithm

The proposed hybrid algorithm uses the technique used in [19]. It is described as follows:

a) Apply Sobel operator, Canny operator and proposed new algorithm to the input image to get edge detected images *u(x,y)*, *v(x,y)*, *w(x,y)*.

b) Make double two-dimensional wavelet decomposition of the images *u(x,y)*, *v(x,y)*, *w(x,y)* and respectively get three groups averages and details [*c1*,*s1*], [*c2*,*s2*], [*c3*,*s3*].

c) Take the averages of three groups: averages and details.

$$ca = \frac{(c1 + c2 + c3)}{3}$$

$$sa = \frac{(s1 + s2 + s3)}{3}$$

d) Make wavelet reconstruction using [*ca*,*sa*] and get the fusion image.(as shown in Figure 5.1).

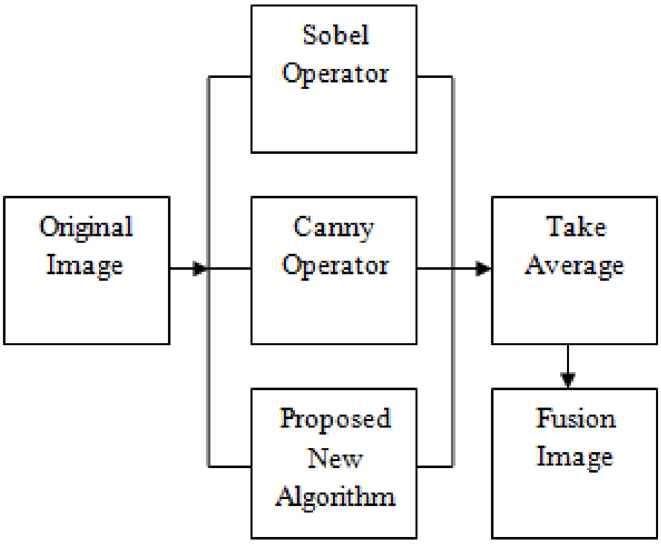


Figure 5.1 An image explaining fusion method

5.3 Visual Comparison of the Results

In order to test the effectiveness, we implemented the proposed algorithms in MATLAB 7.10.0 and the output of the proposed new and proposed hybrid algorithm is compared with the existing algorithms as shown in the following figures:

Figure 5.2(a) shows Blood image. Figure 5.2(b) shows edge detected image when Sobel operator is applied to original image. Figure 5.2(c) shows edge detected image when Prewitt operator is applied to original image. Figure 5.2(d) shows edge detected image when Roberts operator is applied to original image. Figure 5.2(e) shows edge detected image when Canny operator is applied to original image. Figure 5.2(f) shows edge detected image when proposed new algorithm is applied to original image. Figure 5.2(g) shows edge detected image when proposed hybrid algorithm is applied to original image.

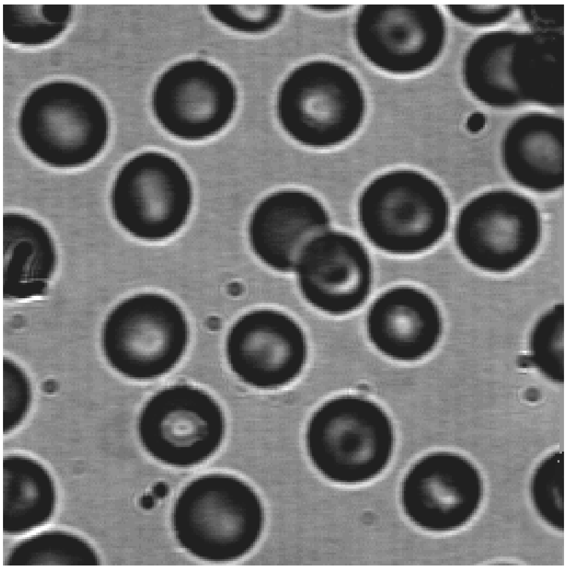


Figure 5.2(a) Blood image

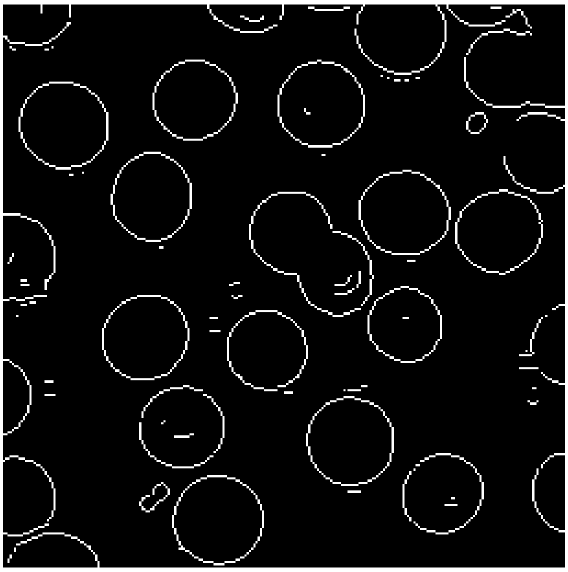


Figure 5.2(b) Sobel operator output

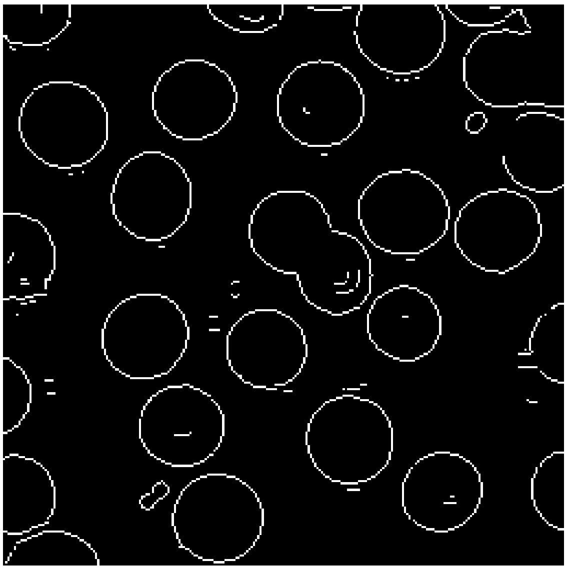


Figure 5.2(c) Prewitt operator output

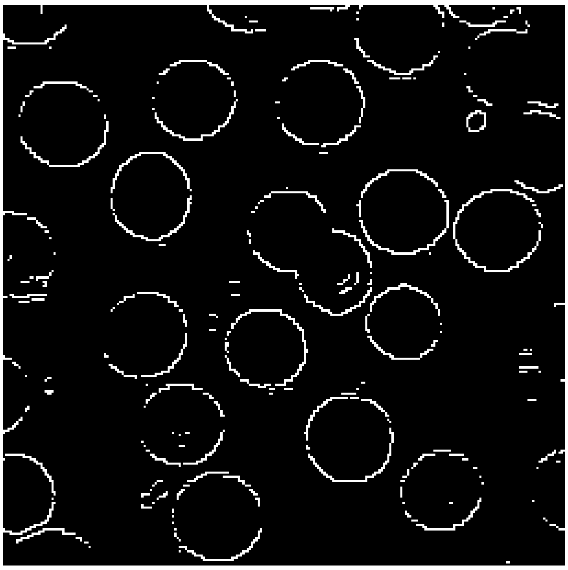


Figure 5.2(d) Roberts operator output

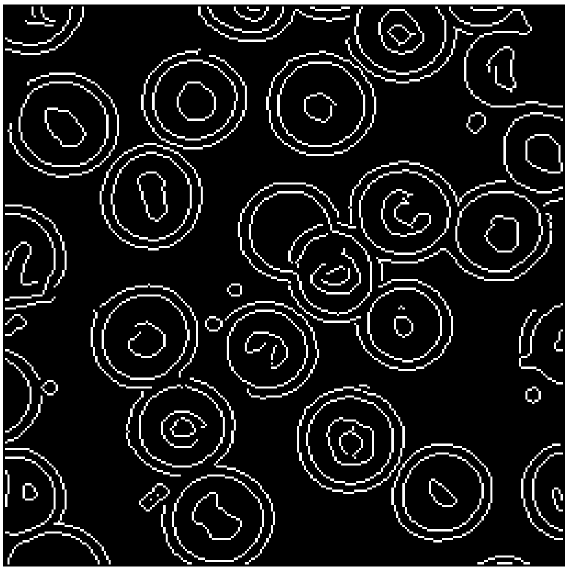


Figure 5.2(e) Canny operator output

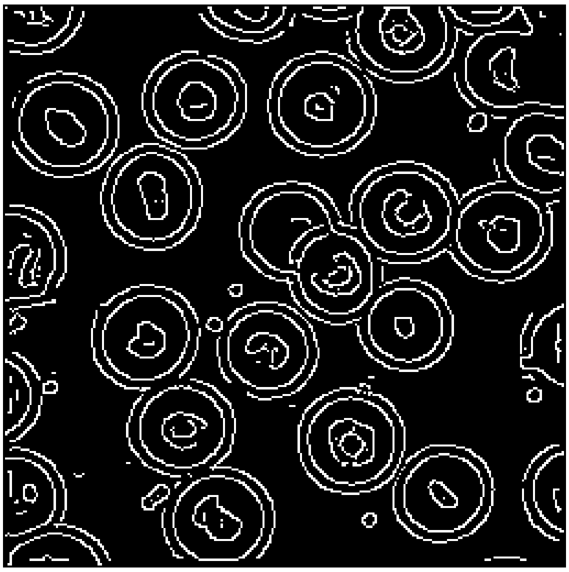


Figure 5.2(f) Proposed new algorithm output

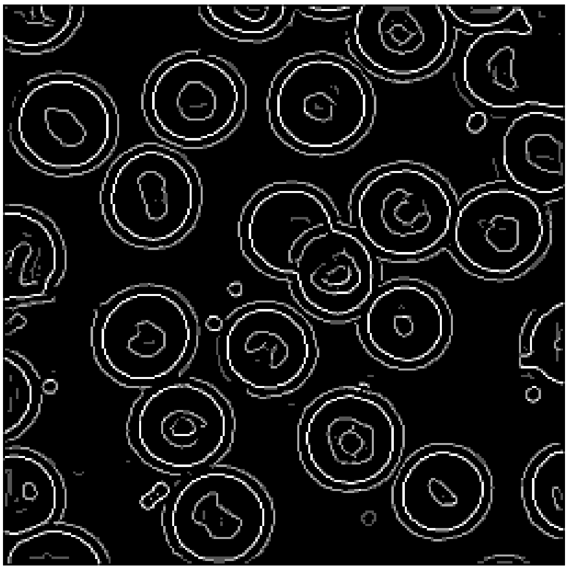


Figure 5.2(g) Proposed hybrid algorithm output

Figure 5.3(a) shows Koala image. Figure 5.3(b) shows edge detected image when Sobel operator is applied to original image. Figure 5.3(c) shows edge detected image when Prewitt operator is applied to original image. Figure 5.3(d) shows edge detected image when Roberts operator is applied to original image. Figure 5.3(e) shows edge detected image when Canny operator is applied to original image. Figure 5.3(f) shows edge detected image when proposed new algorithm is applied to original image. Figure 5.3(g) shows edge detected image when proposed hybrid algorithm is applied to original image.

Figure 5.3(a) Koala image

Figure 5.3(b) Sobel operator output

Figure 5.3(c) Prewitt operator output

Figure 5.3(d) Roberts operator output

Figure 5.3(e) Canny operator output

Figure 5.3(f) Proposed new algorithm output

Figure 5.3(g) Proposed hybrid algorithm output

Figure 5.4(a) shows Lena image. Figure 5.4(b) shows edge detected image when Sobel operator is applied to original image. Figure 5.4(c) shows edge detected image when Prewitt operator is applied to original image. Figure 5.4(d) shows edge detected image when Roberts operator is applied to original image. Figure 5.4(e) shows edge detected image when Canny operator is applied to original image. Figure 5.4(f) shows edge detected image when proposed new algorithm is applied to original image. Figure 5.4(g) shows edge detected image when proposed hybrid algorithm is applied to original image.

Figure 5.4(a) Lena image

Figure 5.4(b) Sobel operator output

Figure 5.4(c) Prewitt operator output

Figure 5.4(d) Roberts operator output

Figure 5.4(e) Canny operator output

Figure 5.4(f) Proposed new algorithm output

Figure 5.4(g) Proposed hybrid algorithm output

Figure 5.5(a) shows Livingroom image. Figure 5.5(b) shows edge detected image when Sobel operator is applied to original image. Figure 5.5(c) shows edge detected image when Prewitt operator is applied to original image. Figure 5.5(d) shows edge detected image when Roberts operator is applied to original image. Figure 5.5(e) shows edge detected image when Canny operator is applied to original image. Figure 5.5(f) shows edge detected image when proposed new algorithm is applied to original image. Figure 5.5(g) shows edge detected image when proposed hybrid algorithm is applied to original image.

Figure 5.5(a) Livingroom image

Figure 5.5(b) Sobel operator output

Figure 5.5(c) Prewitt operator output

Figure 5.5(d) Roberts operator output

Figure 5.5(e) Canny operator output

Figure 5.5(f) Proposed new algorithm output

Figure 5.5(g) Proposed hybrid algorithm output

Figure 5.6(a) shows Pirate image. Figure 5.6(b) shows edge detected image when Sobel operator is applied to original image. Figure 5.6(c) shows edge detected image when Prewitt operator is applied to original image. Figure 5.6(d) shows edge detected image when Roberts operator is applied to original image. Figure 5.6(e) shows edge detected image when Canny operator is applied to original image. Figure 5.6(f) shows edge detected image when proposed new algorithm is applied to original image. Figure 5.6(g) shows edge detected image when proposed hybrid algorithm is applied to original image.

Figure 5.6(a) Pirate image

Figure 5.6(b) Sobel operator output

Figure 5.6(c) Prewitt operator output

Figure 5.6(d) Roberts operator output

Figure 5.6(e) Canny operator output

Figure 5.6(f) Proposed new algorithm output

Figure 5.6(g) Proposed hybrid algorithm output

Figure 5.7(a) shows Rice image. Figure 5.7(b) shows edge detected image when Sobel operator is applied to original image. Figure 5.7(c) shows edge detected image when Prewitt operator is applied to original image. Figure 5.7(d) shows edge detected image when Roberts operator is applied to original image. Figure 5.7(e) shows edge detected image when Canny operator is applied to original image. Figure 5.7(f) shows edge detected image when proposed new algorithm is applied to original image. Figure 5.7(g) shows edge detected image when proposed hybrid algorithm is applied to original image.

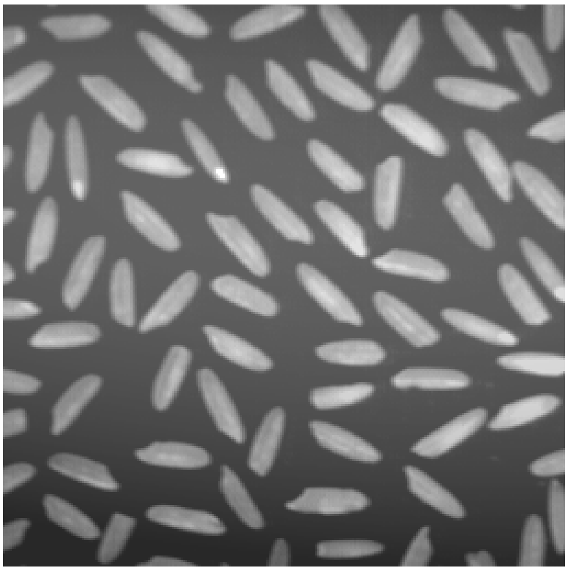


Figure 5.7(a) Rice image

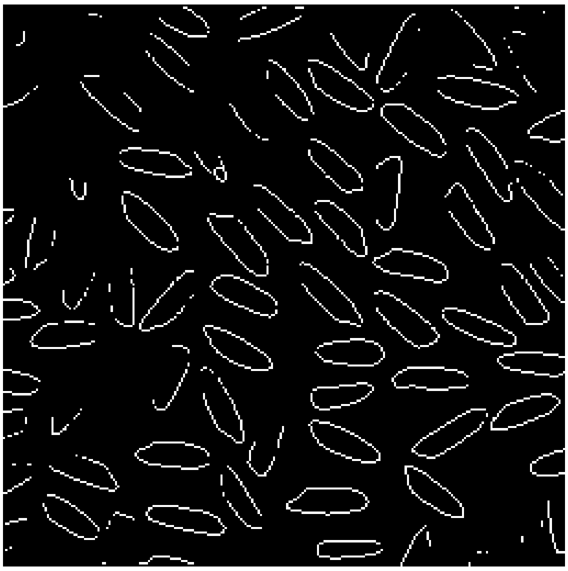


Figure 5.7(b) Sobel operator output

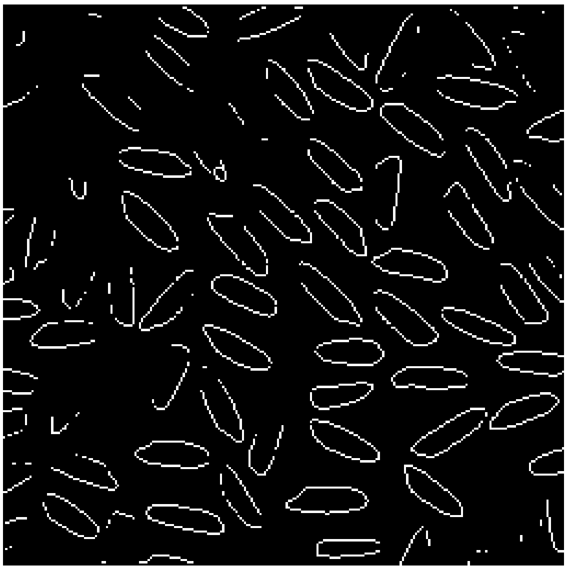


Figure 5.7(c) Prewitt operator output

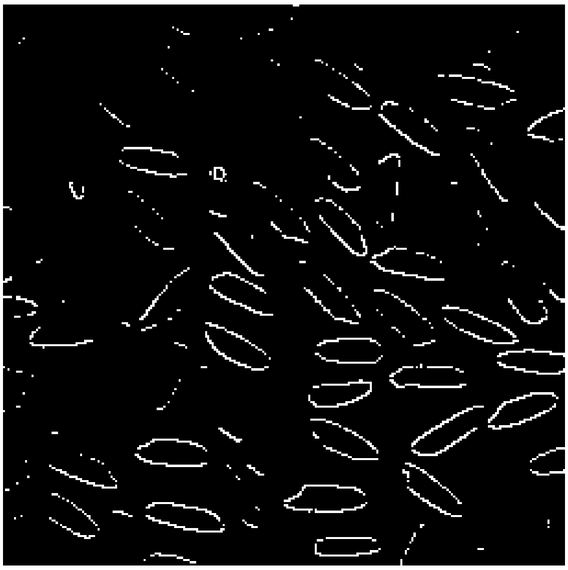


Figure 5.7(d) Roberts operator output

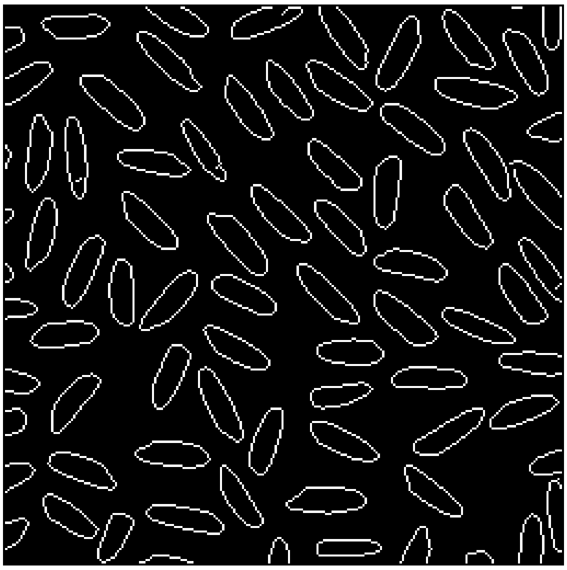


Figure 5.7(e) Canny operator output

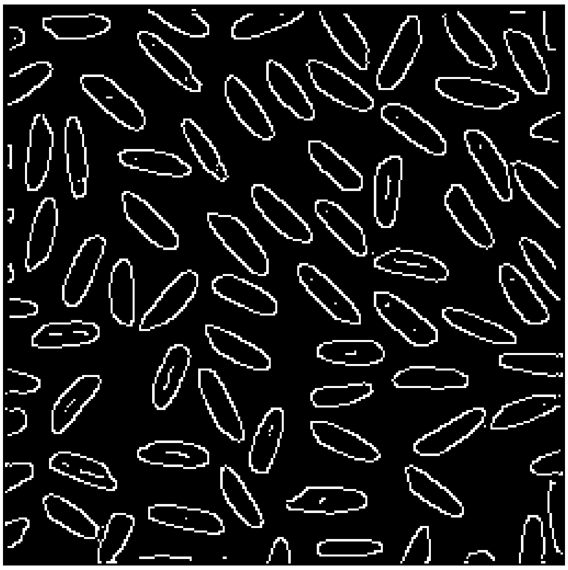


Figure 5.7(f) Proposed new algorithm output

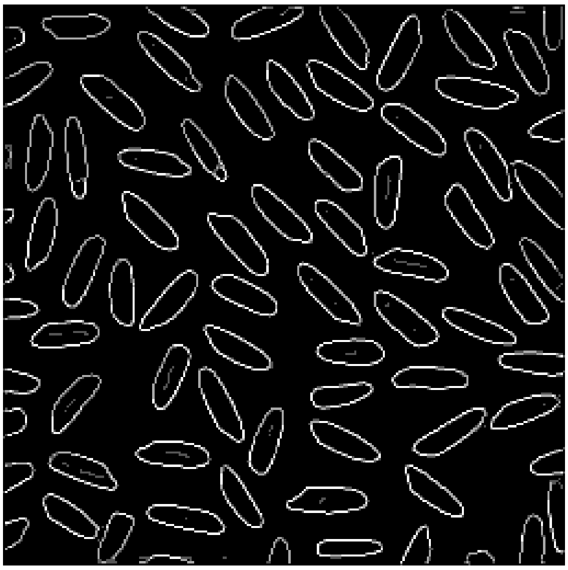


Figure 5.7(g) Proposed hybrid algorithm output

Chapter 6

Conclusion and Future Scope

This thesis presents an improved edge detection method. Experiments show that this hybrid technique improves the accuracy of edge detection and the final image contains a relatively complete edge profile. In practice, choosing a suitable method for image edge extraction is based on specific conditions. Different algorithms have their respective advantages and disadvantages. All kinds of algorithms exist flawed, so this hybrid method combines the advantages of different algorithms in order to obtain a better result.

Canny produces single pixel thick, continuous edges. But the performance of the Canny algorithm depends heavily on the adjustable parameters: σ, which is the standard deviation for the Gaussian filter, and the threshold values, $t1$ and $t2$. The output of the proposed new algorithm clearly shows that it automatically obtains a relatively complete edge profile as compared to the traditional methods like Sobel, Prewitt, Roberts and Canny. The output of proposed hybrid algorithm shows that it combines the edge detected images of all of the methods. It enlightens those portions of the final edge detected image where all of the fused edge detected images have edges and other parts are not highlighted well. In this process, it leaves some of the true edges as dim which should have been highlighted otherwise. So, the future scope will be to study the reasons for this in detail and improve this hybrid method so that it combines the advantages of all of these methods without affecting the highlighting of true edges.

References

[1] W. Niblack, “An Introduction to Digital Image Processing”, Prentice Hall International, 1986.

[2] R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, Second Ed., Prentice Hall, 2002.

[3] J. M. S. Prewitt, "Object enhancement and extraction," Academic Press, B. Lipkin, A. Rosenfeld, (Eds.): Picture Processing and Psychopictorics, New York, pp. 75-149, 1970.

[4] N. A. Otsu, “A threshold selection method from gray-level histograms,” IEEE Transactions on Systems, Man and Cybernetics, Vol. 9, No. 1, pp. 62-66, 1979.

[5] J. Canny “A Computational Approach to Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 679-698, 1986.

[6] Y. H. Kuo, C. S. Lee and C. C. Liu, “A New Fuzzy Edge Detection Method for Image Enhancement”, IEEE Int. Conference on Fuzzy Systems, pp. 1069-1074, 1997.

[7] M. B. Ahmad and T. S. Choi, “Local Threshold and Boolean Function Based Edge Detection”, IEEE Transactions on Consumer Electronics, Vol. 45, No. 3, pp. 332-333, 1999.

[8] S. E. El-Khamy, M. Lotfy and N. El-Yamany, “A Modified Fuzzy Sobel Edge Detector”, Seventeenth National Radio Science Conference, Minufiya University, Egypt, C32, pp. 1-9, 2000.

[9] R. Zhang, G. Zhao and Li Su, “A New Edge Detection Method in Image Processing”, Proceedings of ISCIT 2005, IEEE, pp. 430-433, 2005.

[10] X. Wang and Jian-Qiu Jin, “An Edge Detection Algorithm Based on Improved Canny Operator”, Seventh International Conference on Intelligent Systems Design and Applications, IEEE, pp. 623-628, 2007.

[11] Feng-ying Cui, Li-jun Zou and B. Song, “Edge Feature Extraction Based on Digital Image Processing Techniques”, International Conference on Automation and Logistics, IEEE, pp. 2320-2324, 2008.

[12] E. Nadernejad, S. Sharifzadeh and H. Hassanpour, “Edge Detection Techniques: Evaluations and Comparisons”, Applied Mathematical Sciences, Vol. 2, No. 31, pp. 1507-1520, 2008.

[13] R. Maini and H. Aggarwal, “Study and Comparison of Various Image Edge

Detection Techniques", International Journal of Image Processing (IJIP), Vol. 3, No. 1, pp.1-12, 2009.

[14] B. Wang and S. S. Fan, "An improved CANNY edge detection algorithm", Second International Workshop on Computer Science and Engineering, IEEE, pp. 497-500, 2009.

[15] Z. Jin-Yu, C. Yan and H. Xian-Xiang, "Edge Detection of Images Based on Improved Sobel Operator and Genetic Algorithms", IEEE, 2009.

[16] K. S. Sim, L. W. Thong, M. A. Lai and C. P. Tso, "Enhancement of Optical Images using Hybrid Edge Detection Technique", Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA 2009), IEEE, pp.186-191, 2009.

[17] You-yi Zheng, Ji-lai Rao and L. Wu, "Edge Detection Methods in Digital Image Processing", The 5th International Conference on Computer Science & Education, IEEE, pp. 471-473, 2010.

[18] C. Ma, L. Yang, W. Gao and Z. Liu, "An Improved Sobel Algorithm Based on Median Filter", 2nd International Conference on Mechanical and Electronics Engineering (ICMEE 2010), IEEE, Vol. 1, pp. 88-92, 2010.

[19] C. Deng, W. Ma and Yin Yin, "An Edge Detection Approach of Image Fusion Based on Improved Sobel Operator", 4th International Congress on Image and Signal Processing, IEEE, pp. 1189-1193, 2011.

[20] G. Jie and L. Ning, "An improved adaptive threshold canny edge detection algorithm", International Conference on Computer Science and Electronics Engineering, IEEE, pp. 164-168, 2012.