EDGE BASED REGION GROWING

A thesis submitted in partial fulfillment of the Requirements for the award of the degree of

MASTER OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By:

Sanmeet Bawa

Roll No-8044123

Under the guidance of:

Mr. Kulbir Singh

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY

(DEEMED UNIVERSITY)

PATIALA-147004

INDIA

JUNE-2006

ABSTRACT

Image segmentation is a decomposition of scene into its components. It is a key step in image analysis. Edge, point, line, boundary, texture and region detection are the various forms of image segmentation. Two of the main image segmentation techniques edge detection and region growing are highly in use for image segmentation.

In human visual systems, edges are more sensitive than other picture elements. Edge detection technique when used alone for image segmentation results in small gaps in edge boundaries. It is sensitive to local variations intensity and the contours obtained are usually not closed. Region growing technique when used alone results in errors in region boundaries and the edge pixels might be joined to any of the neighboring pixels.

Edge based region growing corresponds to the optimum image segmentation technique in which the both edge detection approach and region growing approach is integrated. This technique is based on the fact that edge based and region based approaches are complementary to each other and use ancillary information to guide the segmentation procedure. This segmentation procedure separates the image in two segments namely background and foreground.

The algorithm described here is for integrating edges and regions. Firstly, the edge map of image is obtained by using canny edge operator. Then the edge region is grown. Very small regions are removed by merging. Thus the effect of noise is completely eliminated. The two types of seeds (pixels) hot and cold are obtained in the edge region and according to the type of data being analyzed and application area, the image is segmented into background and foreground objects.

It offers very precise segmentation in detecting objects of different sizes and also non-rigid targets. This approach is not sensitive to the parameters, such as the sizes of different operators and thresholds in the edge detection and edge region detection.

The algorithm is implemented in MATLAB and the result demonstrates that the algorithm is robust, satisfying and work well for images with non-uniform illumination.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE No.

CHAPTER	TITLE	PAGE No.

LIST OF FIGURES

FIGURE No.	NAME OF FIGURE	PAGE No.

FIGURE No.	NAME OF FIGURE	PAGE No.

FIGURE No.	NAME OF FIGURE	PAGE No.

CHAPTER 1

INTRODUCTION

Digital image processing refers to processing of a two-dimensional picture by digital computer. It implies digital processing of two dimensional data. A digital image is an array of real or complex numbers represented by a finite number of bits. Image segmentation is a key step in digital image processing. It was developed in 1960's for image analysis. It is the process of grouping together pixels which are semantically linked. Segmentation divides image into its constituent regions or objects. The level to which segmentation is carried out depends upon the problem being solved i.e. segmentation should stop when the objects of interest in an application have been isolated.

Segmentation accuracy determines the eventual success or failure of computerized analysis procedures. For this reason considerable care is taken to improve the probability of rugged segmentation . In some situations such as industrial inspection applications, at least some measure of control over the environment is possible at times. In others, as in remote sensing, user control over image acquisition is limited principally to the choice of image sensors.

1.1 OVERVIEW

Image segmentation is a tool used for precise image analysis. An object input image is taken and is preprocessed. Preprocessing is done to convert the image in more suitable form and to remove the noise. Image smoothing and binarizing are the two stages of preprocessing. Various filters such as median filter, spatial average filter, linear filter and Gaussian filters are used for image smoothing. In few cases noise is multiplicative. Noise smoothing filters are also designed for such images. Binarized image has only two levels i.e. black and white and is obtained by thresholding. The next step of image segmentation is feature extraction. Feature extraction generally refers to the extraction of discontinuities such as point, line and edge, and pixels forming homogeneous regions. Such features have difference in gray level when compared to the background area. Region growing is based on similarity criteria. Region growing is an iterative process by which regions are merged starting from individual pixels or initial segmentation and grow iteratively until every pixel is processed. Selection of edge or region depends upon the type of data being analyzed and on the application area.

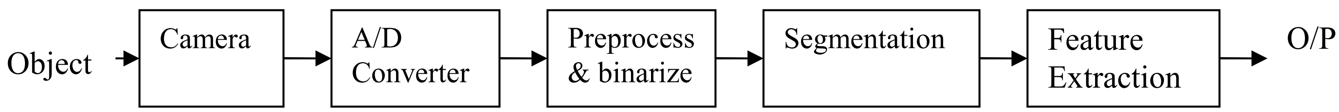


Figure-1.1: Image Analysis

Therefore, the final output image is a segmented image in which the features of the objects in foreground are extracted so precisely that they are separated from the background.

In human visual system, edges are more sensitive than other picture elements. As a result, if one uses either region-growing or edge detection technique alone, one may lose some information of interested objects. For example, if one uses region-growing technique alone, the lack of edge information would terminate region-growing process at wrong place. If the similarity criteria were too strict, many false edges would be generated. In other words, the region-growing process may not stop at the contour of object. In order to improve segmentation results, combination of region growing and edge detection techniques is a good research issue. The integrated method can exploit the edge information obtained by using edge detection techniques to help the region growing process determine where and when to stop the growing process. In this way, objects separated could have accurate contour on the true edges. Edge based and region based approaches are complementary to each other and use ancillary information to guide the image segmentation procedure. The early research used the edge information to check the boundary produced by performing region growing process on the raw input image. In our research, we would perform the region-growing process directly on the DIS (difference in strength) map.

Any region growing technique may produce false boundaries because the uniformity criterion may not be satisfied over a given area even if there is no clear line where a transition occurs. Furthermore, it is likely that such boundaries will reflect the data structures and traversal strategies used during region growing.

The application of any region growing process can lead to three kinds of errors:

a) A boundary is not an edge and there are no edges nearby.
b) A boundary corresponds to an edge but it does not coincide with it.
c) There exist edges with no boundaries near them.

The probability of third type errors mentioned above can be reduced significantly, if not eliminated altogether, by the proper selection of parameters. This results in an over segmented image because such parameter settings cause the errors of first type to increase.

This thesis takes the view that in order to achieve a meaningful segmentation, low level features must be extracted first and subsequently linked together using a series of opportunistic grouping algorithms. At the lowest level the only information is similarity.

The main goal of segmentation scheme presented is to combine edge and region information to achieve a stable segmentation. The segmentation scheme presented is designed to operate on general home and stock photographs. It returns comprehensive region based description of the visual content of an image. This segmentation scheme is designed to facilitate image retrieval and has been tested on several images and has been found to be robust, rapid and free of tuning parameters. The background noise is removed and reliability and accuracy of image segmentation is increased. It offers precise segmentation in detecting multiple objects of different sizes and non rigid targets. It improves static image segmentation and the computational load is low. Stable segmentation of satellite images is achieved by this process.

1.2 LITERATURE SURVEY

Image segmentation is the most widely used image processing techniques that emerged in early 1960's. Recent segmentation schemes in [1] are largely either texture based or edges based and are designed to operate on grey scale images. In early 1970's the techniques introduced were based on histogram thresholding and on detection of discontinuities.

In 1975, a cluster based image segmentation algorithms were introduced by R. M. Haralick and L. G. Shapiro [2]. The clustering techniques introduced in this paper were used for classification of raw data to establish classes and prototypes. Simultaneous consideration of smoothness and contrast was used first by Montanari and Ballard who applied dynamic programming for minimizing a weighted sum of a contrast measure and a smoothness measure in 1976 [3], [4]. Color edge detection and its use in scene segmentation was introduced by K.Niveta and G.S. Robinson in 1977 [5], [6]. In this paper the color edge detectors were described. In 1978, J.S. Weszka did survey on threshold selection techniques, and introduced the effects of using single threshold and multiple thresholds in image segmentation. Also picture segmentation using recursive splitting method was introduced [7], [8]. Further, a threshold selection method from gray-level histograms was introduced by N.Otsu in 1979 [9]. Then in 1980 theory of edge detection for image segmentation was proposed by D.Marr and E.Hildreth. In this paper detection of discontinuities such as point, line and edge were described. Also, Grinaker used edge detection in areas around the borders of regions found by region-based segmentation [10], [11].

In 1986, an optimal canny edge detection technique was introduced by J.Canny. In the same year Bajcsy *et al.* showed that both edge detection and region growing are aspects of the same processes under the assumption of step edges and approximately uniform brightness within the regions. It was described that both processes can be unified by making the decision whether the point is on boundary or on a homogeneous surface [12], [13].

In 1987, optimal edge detection using recursive filtering was introduced by Fua and Hanson. They used high-level domain knowledge and edge based technique to pick out the best segmentation from series of region based segmented images. Another integration technique was introduced in which the number of regions was approximately known and used it to estimate the corresponding parameters of an edge detection process. The result of the edge detection was then used to initialize and to assist a region growing process where a local similarity threshold t (which was used to judge whether two points belonged to the same region) was gradually increased until number of the region resulted. More recently Kass *et al.* solved explicitly a regularization problem that located contours by minimizing a cost function of three terms. In approach of this kind it needs either an initial guess for a contour, or strict assumptions about the contour. The boundaries found during segmentation by region growing are good initial guesses [14], [15], [16].

Seeded region growing method and region-based strategies for active contour models were introduced in 1994 by R. Adams and L. Bischof [17], [18]. Region based approaches introduced in this paper made an attempt to take both distance in space and similarity in properties into account in image segmentation and seeded region growing described was based on greedy algorithm of forming a set of pixels which satisfy the homogeneity criteria.

In 1998, a new region growing approach was introduced in which first region is grown and then the edges were defined so that boundaries obtained by region growing and edges detected collapse with each other [19].

The various image segmentation techniques described above and introduced are either edge based or region based alone. The drawback of using edge based techniques alone for image segmentation is that the resultant edge boundaries are left with small gaps which allow merging of dissimilar regions. Similarly, if region based techniques are used alone for image segmentation, then errors occur in region boundaries and edge pixels might be joined with the any of the neighbouring pixels. In this thesis a new segmentation technique is described in which the edge information and region growing information are merged together so that the drawbacks discussed above are removed while doing precise image analysis.

1.2 OBJECTIVE OF THESIS

The objectives of thesis are:

- To achieve stable segmentation by combining edge and region information of a particular image.
- To obtain optimum segmentation technique which is not sensitive to the parameters such as sizes of different operators and the thresholds in edge detection and region edge detection.

- To introduce a new technique for segmentation i.e. integration of optimum edge detection and region growing techniques.
- To obtain the closed contours of the objects lying in the foreground.
- To segment the objects lying in the foreground from the background area precisely.

1.3 ORGANIZATION OF THESIS

In chapter 2 over view of various image segmentation techniques i.e. cluster based, edge based, region based, watershed segmentation and histogram thresholding are discussed. Also the detection of discontinuities such as point and line detection techniques and their respective masks are discussed.

In chapter 3 thorough discussion of edge detection and several optimal edge detection techniques such as prewitt, sobel and canny are discussed. The respective edge detection masks and algorithms are also given in this chapter.

Chapter 4 describes region and how region growing is done. Different region growing techniques such as Local, global, split and merge and most optimum seeded region growing techniques are discussed.

Chapter 5 describes the algorithm for integrating edge detection and region growing technique, and provides with simulation results of three images i.e. block, vegetable and rice images.

Chapter 6 enlists the important conclusion of the thesis.

1.4 METHODOLOGY

PC configuration	Intel, Pentium III
	1.25 GHz
	256 Mb of RAM

Operating systems	Window XP Professional, Version (2002 Service Pack)
Software	MATLAB 6.1

CHAPTER 2

IMAGE SEGMENTATION

2.1 INTRODUCTION

Segmentation plays an important role in image analysis. The goal of segmentation is to isolate the regions of interest depending on the problem being solved. Many applications (e.g., OCR) of image analysis need to obtain the regions of interest before the analysis can start. Therefore, the need of an efficient segmentation method has always been there. A gray-level image consists of two main features, namely region and edge.

Segmentation algorithms for monochrome or gray images are generally based on of two basic properties of image intensity values: discontinuity and similarity. In the first category, the approach is to partition an image based on abrupt changes in intensity, such as edges in an image. The principle approaches in the second category are based on partitioning image into regions that similar according to a set of predefined criteria. Thresholding, region growing, and region splitting and merging are examples of methods in this category

2.2 IMAGE SEGMENTATION

Image segmentation refers to the major step in image processing in which the inputs are images and, outputs are the attributes extracted from those images. Segmentation divides image into its constituent regions or objects. The level to which segmentation is carried out depends upon the problem being solved i.e. segmentation should stop when the objects of interest in an application have been isolated. Image segmentation refers to the decomposition of a scene into its components. For example in the automated inspection of electronic assemblies, interest lies in the analyzing images of the products with the objective of determining the presence or absence of specific anomalies, such as missing components or broken connection paths. There is no point in carrying segmentation past the level of detail required to identify those elements. Segmentation of nontrivial images is one of the most difficult tasks in image processing.

Segmentation accuracy determines the eventual success or failure of computerized analysis procedures. For this reason considerable care is taken to improve the probability of rugged segmentation . In some situations such as industrial inspection applications, at least some measure of control over the environment is possible at times.

In others, as in remote sensing, user control over image acquisition is limited principally to the choice of image sensors.

2.3 DETECTION OF DISCONTINUTIES

There are three basic types of gray level discontinuities in a digital image: points, line and edge. The most common way of looking for discontinuities is to run a 3 X 3 mask as shown in figure 2.1 through the image [20].

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_7

Figure-2.1: A general 3 X 3 mask

This procedure involves computing sum of products of coefficients with the gray levels contained in the region encompassed by the mask.

The response of the mask at any point in the image is given by

$$R= w_1z_1 + w_2z_2 +\ldots\ldots + w_9z_9 \qquad - (2.1)$$

where, z is the gray level of pixel associated with mask coefficient w. The response of the mask is defined with respect to its centre location.

2.3.1 POINT DETECTION

The detection of isolated points in an image is straight forward in principle. Mask as shown in figure 2.2, when applied on an image detects a point at the location on which mask is centered if

$$|R| > T \qquad - (2.2)$$

where, T is non negative threshold and R is given by equation 2.1. Basically, this formulation measures the weighted differences between the centre point and its neighbours. The idea is that, an isolated point (a point whose gray level is significantly different from its background and which is located in a homogeneous or nearly homogeneous area) will be quite different from its surroundings and thus be easily detectable by this type of mask. The mask coefficients sum to zero, indicating that the mask response will be zero in areas of constant gray level [20].

-1	-1	-1
-1	-8	-1
-1	-1	-1

Figure-2.2: Point Detection Mask

2.3.2 LINE DETECTION

The next level of complexity in detection of discontinuities is line detection. Consider the masks shown in figure 2.3. If the first mask were moved around an image, it would respond more strongly to the lines (one pixel thick) oriented horizontally. With a constant back ground, the maximum response would result when the line passed through the middle row of the mask [20].

-1	-1	-1
2	2	2
-1	-1	-1

(a)

-1	-1	2
-1	2	-1
2	-1	-1

(b)

-1	2	-1
-1	2	-1
-1	2	-1

(c)

2	-1	-1
-1	2	-1
-1	-1	2

(d)

Figure-2.3(a): Horizontal Line Mask, (b): $+45^0$ Line Mask, (c): Vertical Line Mask, (d): -45^0 Line Mask

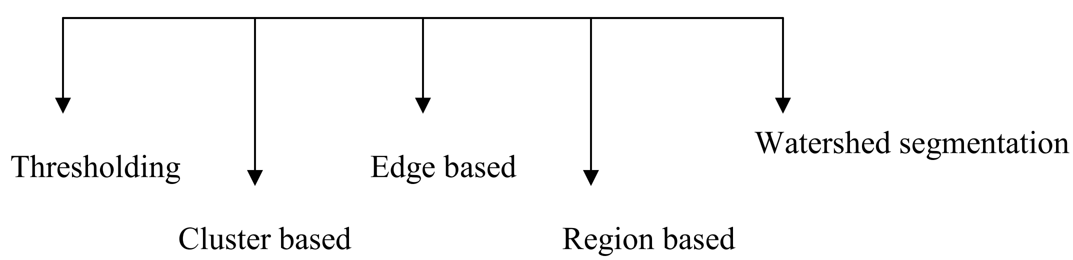
This is easily verified by sketching a simple array of ones with a line of different gray level running horizontally through the array. Second mask is oriented at $+45^0$, third mask detects vertical lines and fourth mask is -45^0 oriented. The coefficients in each mask sum to zero, indicating the zero response from the masks in areas of constant gray level. Simply the mask is run through the image and the absolute values of the result are thresholded.

2.3.3 EDGE DETECTION

Edge detection is the most common approach for detecting meaningful discontinuities in gray level. Edge is a set of connected pixels that lie on a boundary between the two regions. Edges are more closely modeled as having ramp like profile. The first derivative is positive at the points into and out of the ramp. It is constant for points on ramp, and is zero in areas of constant gray level. The second order derivative is positive at the transition associated with the dark side of the edge, negative with the transition associated with the light side of the edge, and zero along the ramp and in areas of constant gray level. Magnitude of the first derivative can be used to detect the presence of an edge at a point in an image. Similarly, sign of the second derivative can be used to determine that whether the edge pixel lies on the dark side or light side of an edge. Second derivative around an edge produces two values for edge in every image. Also an imaginary straight line joining the extreme positive and the negative values of the second derivative would cross zero near the mid point of the edge.

2.4 IMAGE SEGMENTATION TECHNIQUES

Image segmentation techniques can be broadly classified as into five main classes:



2.4.1 CLUSTER BASED METHOD

A cluster is a set of points in the feature space for which their local density is very large (relative maximum) as compared to the density of feature points in the surrounding region. Clustering techniques are useful for image segmentation and for classification of raw data to establish classes and prototypes. Clustering is useful vector quantization technique for compression of images.

Cluster based method uses thresholding, i.e. a threshold is estimated from grey level intensity histogram of the image. The valley in the histogram is taken as threshold. Problem arises in this method when the valley is not very distinct, when the object intensities overlap that of background.

2.4.2 EDGE BASED METHOD

Edge based methods try to find the places of rapid transition from one to the other region of different brightness or color value. The basic principle is to apply some of the gradient operators convolving with the image. High values of the gradient magnitude are possible places of rapid transition between two different regions and are called edges. After finding edges on the image they are linked to form closed boundaries of the regions. To go from image edges to the image boundaries is very difficult task. Edge based techniques utilize the discontinuity of boundary (e.g., edge detection) and can extract the contour of region of interest. On the other hand, edge detection is also a useful method for image segmentation, since an edge often represents a junction of two adjacent regions.

2.4.2.1 Edge

An edge is not a physical entity and it has no width. It is where the picture ends and the wall starts. It is where the vertical and the horizontal surfaces of an object meet. It is what happens between a bright window and the darkness of the night. If there were sensor with infinitely small footprints and zero-width point spread functions, an edge would be recorded between pixels within an image. The edge between a forest and a road in an aerial photo may not look like an edge in image taken on the ground. In the ground image, edges may be found around each individual tree. If looked a few inches away from a tree, edges may be found within the texture on the bark of the tree. Edges are scale dependent and an edge may contain other edges, but at a certain scale, an edge still has no width.

2.4.2.2 Edge detection techniques

Edge detection is by far most common approach for detecting meaningful discontinuities in intensity values. Such discontinuities are detected by first order and second order derivatives. The first order derivative of choice in image processing is the gradient. The laplacian is seldom used by itself for edge detection because as a second derivative, it is unacceptably sensitive to noise. Its magnitude produces double edges and it is unable to detect the edge direction. Following are the edge detection techniques:

- Gradient operators:

Before an image can be segmented, the objects in that image must be detected and roughly classified as to shape and boundary characteristics. Some techniques used to detect objects use a gradient operator to locate potential object boundaries or edges. The gradient operator applied to a continuous function produces a vector at each point whose direction gives the direction of the maximum change of the function at that point, and whose magnitude gives the magnitude of this maximum change. A digital gradient is often computed by convolving two windows with an image, one window giving the x component g_x of the gradient, and the other giving the y component g_y. This operation can be described by the expressions:

$$g(i,j) = mask * f(i,j) \qquad - (2.3)$$

$$g(i,j) = mask * f(i,j) \qquad - (2.4)$$

where, $f(i,j)$ is indicating some neighbourhood of pixel (i,j), and * denotes convolution.

- Sobel edge detector [21]:

The incoming image data, in this case, is convolved with four *3* X *3* masks (i.e., Sobel operators) weighted to measure the differences in intensity along the horizontal, vertical left and right directions. These four measurements E_{H}, E_{V}, E_{DL} and E_{DR} are then combined to estimate edge magnitude and direction. The inclusion of diagonal measurements allows a more accurate, yet computationally simple, magnitude approximation than could be attained with just the usual horizontal and vertical measurements and simplifies the directional computation. The gradient magnitude is given by

$$Mag = [E_H^2 + E_V^2]^{1/2} \qquad - (2.5)$$

$$\text{Mag} = \max [|E_H|, |E_V|, |E_{DL}|, |E_{DR}|] + k|\,E_T| \qquad - (2.6)$$

where,

$$E_H = \begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I \qquad - (2.7)$$

$$E_L = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \qquad - (2.8)$$

$$E_{DR} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} * I \qquad - (2.9)$$

$$E_{DL} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} * I \qquad - (2.10)$$

I is image and E_T is the measure in direction perpendicular to a selected maximum measure. The gradient direction is assigned from a template of eight angle directions based on the selected maximum measure and its corresponding sign.

$$\Theta = inv \tan[E / H] \qquad - (2.11)$$

k is determined as the value which minimizes the magnitude error, which is computed using error analysis.

➢ Canny edge detector:

The Canny operator works in a multi-stage process. First the image is smoothed by Gaussian convolution. Then a simple 2-D first derivative operator is applied to the smoothed image to highlight regions of the image with high first spatial derivatives.
Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non maximal suppression in [13], [22].
The ridge pixels are then thresholded using two thresholds, T_1 and T_2, with $T_1 < T_2$. Ridge pixels with values greater than T_2 are said to be "strong" edge pixels. Ridge pixels with values between T_1 and T_2 are said to be "weak" edge pixels. Finally the canny algorithm performs edge linking by incorporating the weak pixels that are 8-connected to the strong pixels. Following are the results for canny edge detector.

(a) **(b)**

Figure-2.4(a): Original Lina Image, (b): Result of canny edge detector

2.4.3 REGION BASED METHOD

Region growing is a procedure that group's pixels or sub regions into larger regions based on predefined criteria [23], [24]. Starting with a set of "seed" points, regions are grown from these by appending to each seed those neighbouring pixels that have properties similar to the seed. These properties could be grey level, or color etc. The seeds selection depends on the nature of the problem. When no a priori information about the location of the objects is available, the properties are computed that are chosen at every pixel. If the result of these computations leads to a cluster of pixels with similar values, a pixel near the centroid of this cluster is chosen, as a seed. For regions with high grey level the centers of these areas as seeds are used.

The region growing process stops when there are no more neighbouring pixels that satisfy the criteria, but additional information such as a restriction of the size of the region can be added. The criteria are then no longer purely local, but depend on the history of the region growth. For this procedure to make any sense, all the pixels in the region have to form a connected component. A connected component contains only pixels that can be reached by a path over pixels that are neighbours. There are 4-connected neighbours and 8-connected neighbours, defining each pixel as having four or eight neighbours.

In single linkage region growing, each pixel is regarded as a node in a graph. Two neighbouring pixels are joined with an arc if they are similar enough, for example if their grey levels differ by less than a certain number. These algorithms are attractive for their simplicity. Hybrid linkage region growing techniques assign a property vector to each pixel depending on the K X K neighbourhood of the pixel. Similarity is thus established as a function of neighbouring pixel values and the algorithm is less sensitive to noise.

If an edge detector is applied to an image and each pixel is labeled as edge or non edge, this information is used to grow regions where neighbours get connected as long as they are not edge pixels. The quality of this technique is highly dependent on the edge operator used. If there are gaps in the edges detected, regions will be linked and the segmentation will not be satisfying.

Region based methods rely on the similarity of localized features and can extract the region of interest directly, one example is the region growing technique [19]. Similarity measures can be the average contrast, peripheral contrast [19], textural features uniformity [25], or gray level uniformity [26]. The basic idea of region growing is that neighbouring pixels, which confirm to the similarity measure, would be clustered into the same region.

Region growing techniques perform quite well under non-uniform illumination in grey level segmentation. In novel edge based region growing method two kind of seeds (pixels)-hot and cold, are defined near edges (of objects) and both kind of regions are grown from seeds simultaneously.

The motivation behind region based approach is an attempt to take both distance in space and similarity of properties into account in segmentation. Some region growing methods use edge as a growth stopping condition, and the growing seeds are selected manually. The growth occurs in the homogeneous intensity regions and stop at the edges. If there are broken edges then the segmentation is incorrect.

The three basic achievements in edge based region growing method are:

(a) The growing seeds are selected at the edge region automatically. In the image, the great difference intensity regions will be segmented into hot and cold seeds. No seeds are defined in the homogeneous intensity regions.

(b) The region growing procedure is performed in the homogeneous regions, and stops automatically when either the hot or cold seeds stop growing. The hot and cold seeds prevent each other from growing into the opposite seed region.

(c) The detected edge boundary is correctly located at the real object boundary.

Quite often, in grey level image segmentation, the objects to be delineated are found in regions of non uniform illumination. In these cases, using thresholds derived from intensity based algorithms cannot yield good segmentation performance. Region growing techniques have been shown to perform well under these situations.

2.4.4 WATERSHED SEGMENTATION

Watershed segmentation [23] is an interesting method for separating objects that touch each other.

The concepts of watershed and catchments basins are well known in topography. Starting from a gradient image, image data is interpreted as a topographic surface, where the gradient values represent altitudes [23]. Thus, region edges represent high watersheds, and low gradient region interiors correspond to catchments basins. In watershed segmentation, all catchments basins of the topographic surface are homogeneous in the sense that all the pixels belonging to the same catchments basin are connected with the region of minimum altitude of the basin (grey level) by a simple path of pixels that have monotonically decreasing altitude along the path. Such catchments basins then represent the regions of the segmented image.

There are some different approaches on implementing the watershed idea. In 1990, Lindeberg *et al* [27] proposed an algorithm that was not too computationally expensive. The catchments basins were filled from the bottom. Each minimum represents one catchments basin, it was started from altitude minima. If a hole is imagined in each local minimum and immerses the topographic surface in water, the water starts filling all the catchments basins that have minima under water. If two catchments basins would merge, this represents a watershed line. Lindeberg *et al* sorted all pixels in the image according to their grey level and then they are flooded by a breadth first scanning of all pixels in the order of their grey levels. The watershed segmentation technique is illustrated in figure 2.5.

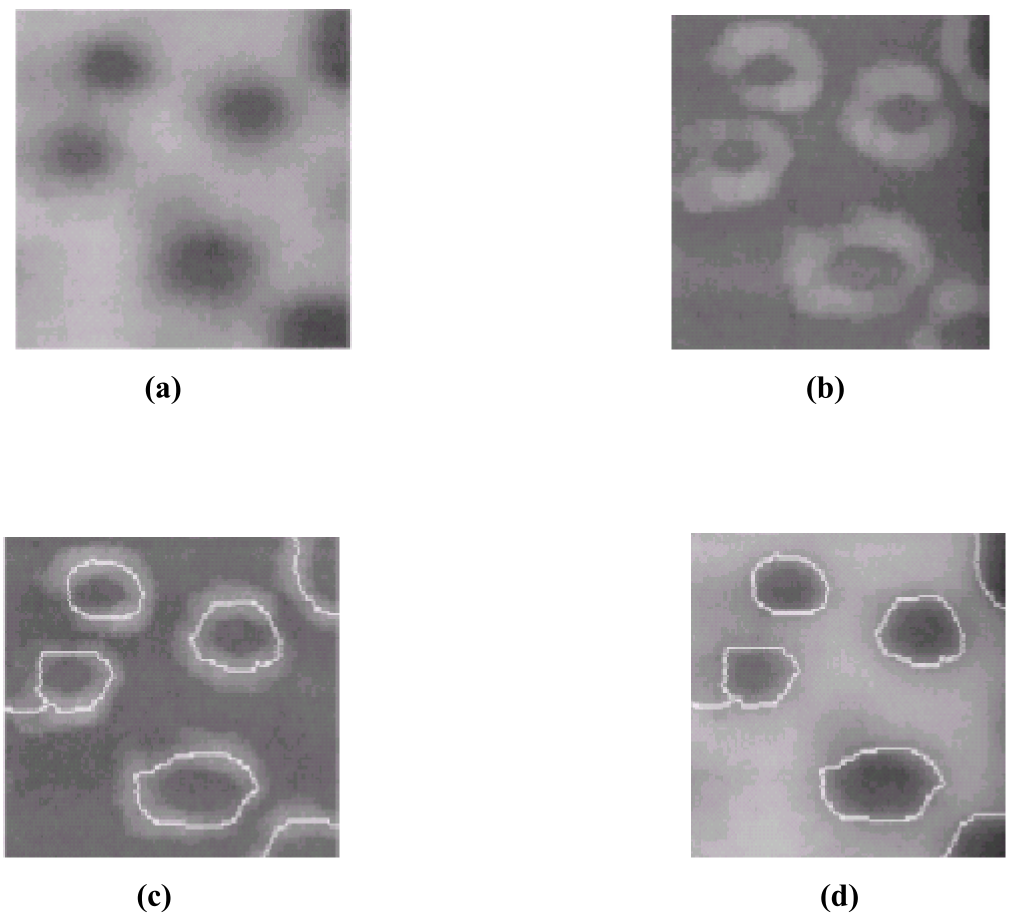


Figure-2.5(a): Original Image with blob shaped objects, (b): Image Gradient, (c): Watershed Lines, (d): Watershed Lines imposed on Original Image

Direct application of the watershed segmentation often leads to over segmentation due to noise and other local irregularities in the gradient magnitude map. An approach to control this is based on the concept of markers. A marker is a connected component belonging to an image. Internal markers are employed to associate objects of interest, and external markers, with the background. The criteria for an internal marker is, that it has to be surrounded by pixels with higher altitude. These markers are then made to be the only allowed minima in the watershed algorithm. The watershed lines created here are defined as external markers. The external markers efficiently partition the image into regions containing one single internal marker and some background. The problem is then reduced to partitioning each of these regions into two, which can be done by some simpler segmentation algorithm.

2.4.5 THRESHOLDING

Thresholding is an important tool in segmentation. For example an image of bright objects on a dark background, like microscope images are considered. When thresholding a grey level image of this form, with a threshold *T*, any point with intensity value larger than *T* is called an object point candidate, whereas all the other points are called background points [23]. A binary image in black and white is obtained. Because of its intuitive properties and simplicity of implementation, thresholding is a common operation in applications of image segmentation.

T does not have to be one single value for the entire image but can be a function.

$$T = T[x, y, p(x, y), f(x, y)] \qquad \text{- (2.12)}$$

where, $f(x, y)$ is the grey level of the point, $p(x, y)$ is some local property of the point and x and y are the spatial coordinates. When T depends only on $f(x, y)$, the same value for all pixels is obtained and this operator is called global thresholding. When T depends on both $f(x, y)$ and $p(x, y)$ it is called local. If T also depends on the spatial coordinates the thresholding is called adaptive or dynamic. If several thresholds are used to segment the image into more than two classes i.e. multilevel thresholding a classified image with more than two classes is obtained.

The success of global thresholding depends entirely on how well the histogram of the image can be partitioned. The ideal histogram should have distinctive peaks with gaps between them.

When using a single global threshold, the image is scanned and each pixel is labeled as an object or background depending on whether the grey level of that pixel is greater than or less than the threshold. *T* can be chosen interactively and subjectively, but usually threshold is selected automatically.

A simple algorithm [28] for thresholding is given below:

• Select an initial estimate for *T*.

• Segment the image using *T*, producing two groups of pixels *G1* and *G2*

• Compute the average grey-level values $\mu\,1$ and $\mu\,2$ of *G1* and *G2*.

• Compute a new threshold value as:

$$T = 1/2[\mu 1 + \mu 2] \qquad \text{- (2.13)}$$

Repeat until the difference in *T* between successive iterations is smaller than a predefined parameter.

.

CHAPTER 3

EDGE DETECTION

3.1 INTRODUCTION

In chapter 2 all the image segmentation techniques are discussed. Edge detection is one of the widely used methods in image segmentation. Edge based methods try to find the places of rapid transition from one to the other region of different brightness or color value. The basic principle is to apply some of the gradient operators convolving them with the image. High values of the gradient magnitude are possible places of rapid transition between two different regions, called edges. After this step of finding edges on the image they are linked to form closed boundaries of the regions.

3.2 EDGE

An edge is not a physical entity, just like a shadow. It is where the picture ends and the wall starts. It is where the vertical and the horizontal surfaces of an object meet. It is what happens between a bright window and the darkness of the night. If there were sensor with infinitely small footprints and zero width point spread functions, an edge would be recorded between pixels within in an image. The edge between a forest and a road in an aerial photo may not look like an edge any more in an image taken on the ground. In the ground image, edges may be found around each individual tree. If looked a few inches away from a tree, edges may be found within the texture on the bark of the tree. Edges are scale dependent in [29] and an edge may contain other edges, but at a certain scale, an edge has no width. Edges are loosely defined as pixel intensity discontinuities within an image. Edge detection is a subjective task. It is easy to detect those obvious edges, or those with high S/N ratio. Edge detector is tailored to take advantage of the domain knowledge. For example, a "straight edge" detector may be very effective in locating most buildings and objects such as tennis courts in an aerial photo.

Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrast i.e. a jump in intensity from one pixel to the next. Image edge detection significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories, gradient and laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one dimensional shape of a ramp and calculating the derivative of the image can highlight its location. Suppose the following signal is considered, with an edge shown by the jump in intensity below:

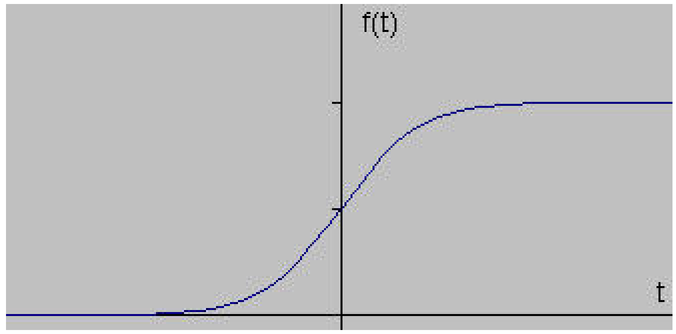


Figure-3.1: Edge

If gradient of this signal is calculated (which, in one dimension, is just the first derivative with respect to t) the following is obtained:

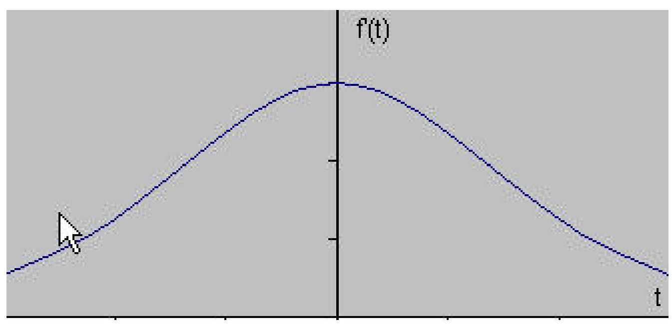


Figure-3.2: Gradient of an Edge

Clearly, the derivative shows a maximum located at the center of the edge in the original signal. This method of locating an edge is characteristic of the "gradient filter", family of edge detection filters and includes the Sobel method. A pixel location is declared an edge location if the value of the gradient exceeds some threshold. As mentioned before, edges will have higher pixel intensity values than those surrounding it. So once a threshold is set, gradient value can be compared with the threshold value and an edge can be detected whenever the threshold is exceeded. Furthermore, when the first derivative is at a maximum, the second derivative is zero. As a result, another alternative to finding the location of an edge is to locate the zeros in the second derivative. This method is known as the Laplacian and the second derivative of the signal is shown below:

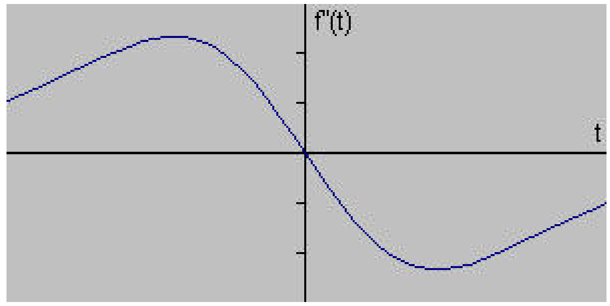


Figure-3.3: Laplacian of an Edge

Edges are often used in image analysis for finding region boundaries. Boundary and its parts (edges) are perpendicular to the direction of the gradient. In figure 3.4 few typical edge profiles are shown.

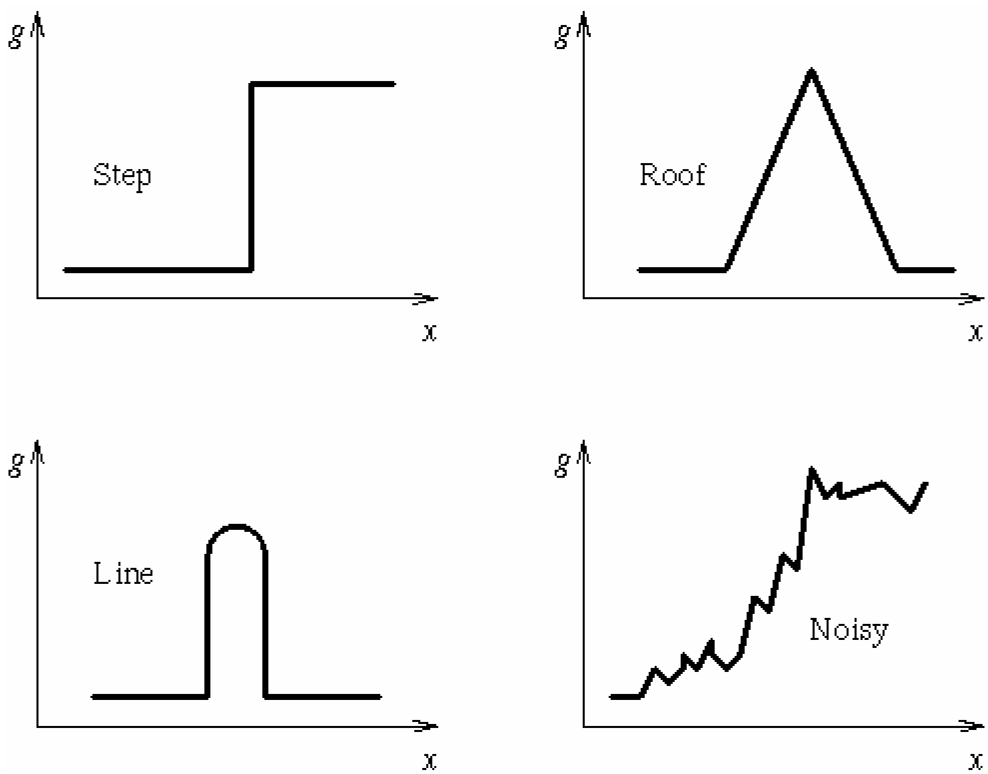


Figure-3.4: Typical Edge Profiles

3.3 EDGE DETECTORS

There are many ways to perform edge detection in [30]. However, the most may be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zerocrossings in the second derivative of the image to find edges.

3.3.1 LAPLACE EDGE DETECTOR

The Laplace operator [30] is a very popular operator approximating the second derivative which gives the gradient magnitude only. The Laplacian is approximated in digital images by a convolution sum. A 3 X 3 mask for 4-neighbourhood and 8-neighbourhood

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \qquad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(a) **(b)**

Figure-3.5(a): 4-neighbourhood 3 X 3 mask, (b): 8-neighbourhood 3 X 3 mask

A Laplacian operator with stressed significance of the central pixel or its neighbourhood is sometimes used. In this approximation it loses invariance to rotation. Its disadvantage is that it responds doubly to some edges in the image and its advantage is that image sharpening / edge detection can be interpreted in the frequency domain as well.

3.3.2 ROBERTS CROSS EDGE DETECTOR

The Roberts edge detector [30] is one of the oldest edge detectors in digital image processing. It uses the mask to approximate digitally the derivatives G_X and G_Y. This detector is used considerably less than the other detectors due to its limited functionality. It is not symmetric and cannot be generalized to detect the edges that are multiples of 45^0. However it is still used in hardware implementation where simplicity and speed are dominant factors.

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Figure-3.6: Roberts Cross Edge Operators

Magnitude of the edge is computed by the following equation

$$|g(i,j)-g(i+1,j+1)|+|g(i,j+1)-g(i+1,j)| \qquad \text{- (3.1)}$$

It is highly sensitive to noise, because very few pixels are used to approximate the gradient.

3.3.3 PREWITT EDGE DETECTOR

The Prewitt operator [30], similarly to the Sobel, Kirsch, Robinson (as discussed later) and some other operators, approximates the first derivative. Operators approximating first derivative of an image function are sometimes called compass operators because of the ability to determine gradient direction. The gradient is estimated in eight (for a 3 X 3 convolution mask) possible directions and the convolution result of greatest magnitude indicate the gradient direction. The direction of the gradient is given by the mask giving maximal response. This is valid for all following operators approximating the first derivative.

$$H_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \qquad H_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure-3.7: Prewitt Edge Operators

3.3.4 KIRSCH EDGE DETECTOR

It is based on convolution in very small neighbourhoods and work well for specific images only. Its parameters are dependent on the size of the objects and sensitivity to noise [30].

$$H_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & -3 \\ -5 & -5 & -5 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \qquad H_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}$$

Figure-3.8: Kirsch Edge Operators

3.3.5 SOBEL EDGE DETECTOR

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial gradient that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grey scale image. It is used as a simple detector of horizontality and verticality of edges in which case only masks H_1 and H_3 are used.

$$H_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \qquad H_3 = \begin{bmatrix} -1 & 0 & 1 \\ 2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure-3.9: Sobel Edge Operators

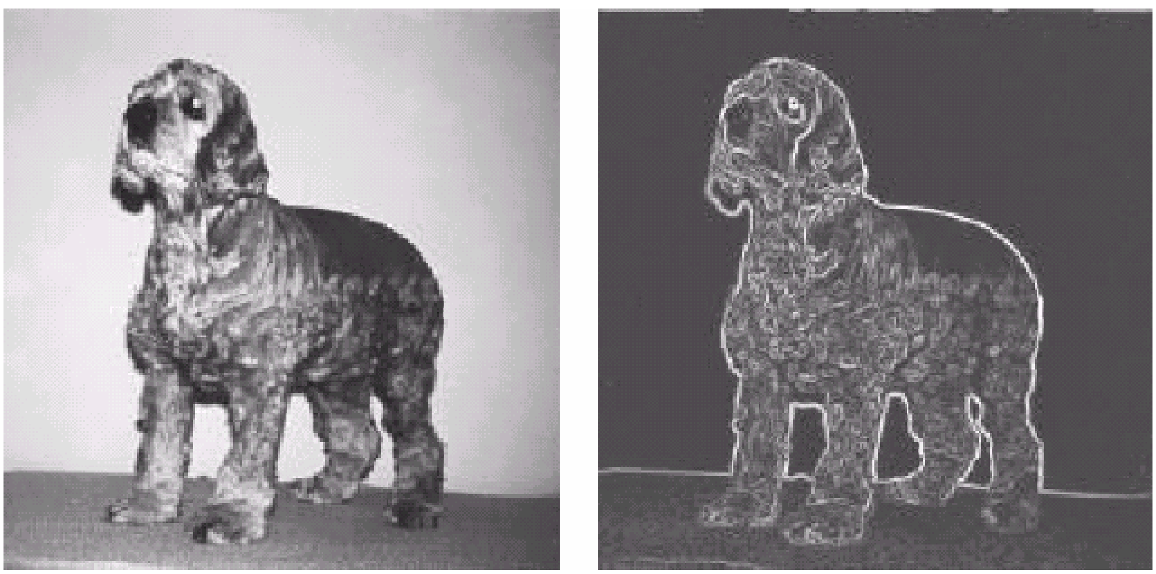
If the H_1 response is y and the H_3 response x, then the edge strength (magnitude) is

$$\text{Magnitude} = \sqrt{\left(x^2 + y^2\right)} \qquad \text{- (3.2)}$$

$$\text{Direction} = inv\tan y/x \qquad \text{- (3.3)}$$

The Sobel operator is slower to compute than the Roberts Cross operator, but its larger convolution mask smoothes the input image to a greater extent and so makes the operator less sensitive to noise. The operator also generally produces considerably higher output values for similar edges compared with the Roberts Cross.

As with the Roberts Cross operator, output values from the operator can easily overflow the maximum allowed pixel value for image types that only support small integer pixel values (e.g. 8-bit integer images). When this happens the standard practice is to simply set overflowing output pixels to the maximum allowed value. The problem can be avoided by using an image type that supports pixel values with a larger range. The Sobel edge detector can also be applied to range images.



(a) **(b)**

Figure-3.10(a): Original Annabel Image, (b): Effects of Sobel Operator

3.3.6 ZERO CROSSING EDGE DETECTOR

An edge detection technique, based on the zero crossings [30] of the second derivative explores the fact that a step edge corresponds to an abrupt change in the image function. The first derivative of the image function should have an extreme at the position corresponding to the edge in the image, and so the second derivative should be zero at the same position. It is much easier and more precise to find a zero crossing position than an extreme as shown in figure below

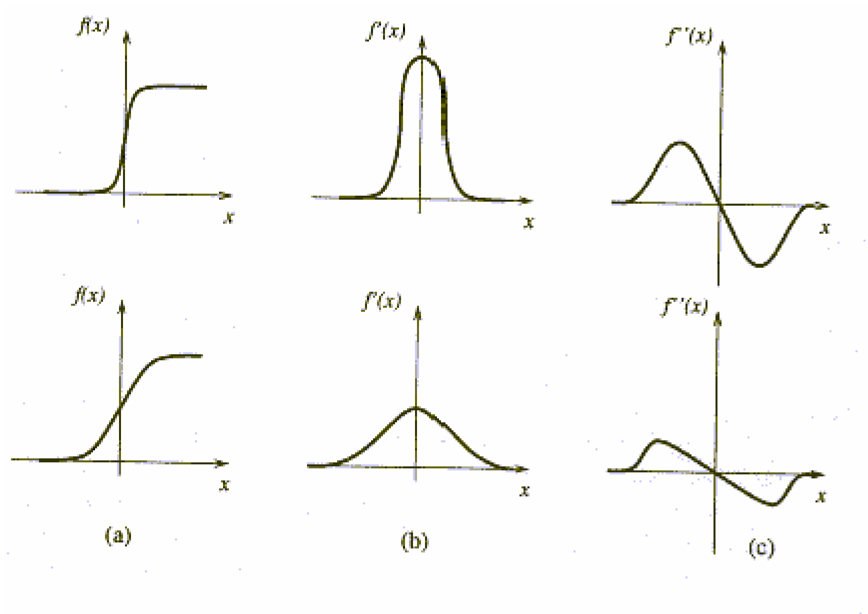


Figure-3.11: 1-D Edge Profile of the Zero Crossing

Its main disadvantages are that it smoothes the shape too much; for example sharp corners are lost and tends to create closed loops of edges.

3.3.7 CANNY EDGE DETECTOR

The Canny operator was designed to be an optimal edge detector (according to particular criteria there are other detectors around that also claim to be optimal with respect to slightly different criteria). It takes as input a gray scale image, and produces as output an image showing the positions of tracked intensity discontinuities.

The Canny operator works in a multi-stage process. First of all the image is smoothed by Gaussian convolution. Then a simple 2-D first derivative operator is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non maximal suppression. The tracking process exhibits hysteresis controlled by two thresholds: T_1 and T_2, with $T_1 > T_2$. Tracking can only begin at a point on a ridge higher than T_1. Tracking then continues in both directions out from that point until the height of the ridge falls below T_2. This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments.

The effect of the canny operator is determined by three parameters: the width of the Gaussian kernel used in the smoothing phase, and the upper and lower thresholds used by the tracker. Increasing the width of the Gaussian kernel reduces the detector's sensitivity to noise, at the expense of losing some of the finer detail in the image. The localization error in the detected edges also increases slightly as the Gaussian width is increased.

Usually, the upper tracking threshold can be set quite high and the lower threshold quite low for good results. Setting the lower threshold too high will cause noisy edges to break up. Setting the upper threshold too low increases the number of spurious and undesirable edge fragments appearing in the output.

One problem with the basic canny operator is to do with Y junctions i.e. place where three ridges meet in the gradient magnitude image. Such junctions can occur where an edge is partially occluded by another object. The tracker will treat two of the ridges as a single line segment, and the third one as a line that approaches, but doesn't quite connect to, that line segment.

The Gaussian smoothing in the canny edge detector fulfills two purposes. First it can be used to control the amount of detail that appears in the edge image and second, it can be used to suppress noise. Although the Canny edge detector can find the intensity discontinuities in an image, but it is not guaranteed that these discontinuities correspond to actual edges of the object.

CHAPTER 4

REGION GROWING

4.1 INTRODUCTION

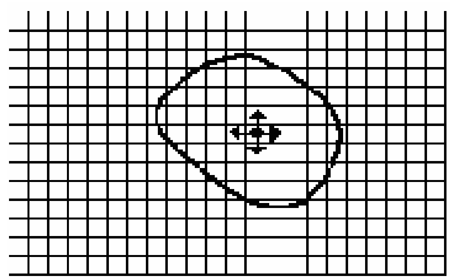
In chapter 3 various edge detection techniques are discussed. It is noted that the edge detection corresponds to the detection of discontinuities whereas, regions are based on detection of similarity in pixels. In this chapter region growing is discussed thoroughly. Region growing is a procedure that groups pixels or sub regions into larger regions based on the predefined criteria for growth. The basic approach is to start with seed points and from these grow regions by appending to each seed those neighboring pixels that have predefined properties similar to the seed such as specific ranges of grey level or color. At every pixel the same set of properties are used to assign pixels to regions during the growing process. When clusters of values occur, the pixels whose properties place them near the centroid of these clusters are used as seeds. Image edges act as hard barriers during region growing.

4.1.1 DEFINITION OF REGION GROWING

It is a method of segmenting regions from data volumes. The user selects a point and a region grows out from that seed point until some criteria for stopping growth are met. Region growing always gives just one area. It can be used to isolate single objects, for example a data volume could contain a number of similar objects, these could be isolated by threshold but all of them will be segmented, but region growing will just isolate the one object from its seed point.

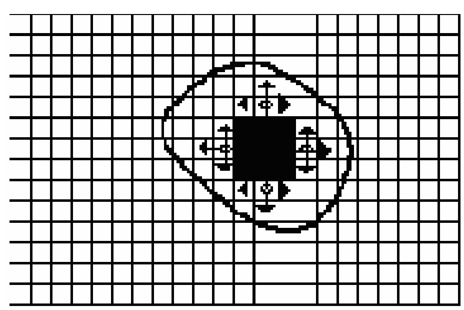
4.1.2 BASIC REGION GROWING APPROACH

Region growing approach is the opposite of the split and merge approach [23]. An initial set of small areas are iteratively merged according to similarity constraints. An arbitrary seed pixel is chosen and compared with neighbouring pixels as shown in figure 4.1. Region is grown from the seed pixel by adding in neighbouring pixels that are similar, increasing the size of the region. When the growth of one region stops another seed pixel is chosen which does not yet belong to any region and the whole procedure is followed again. This whole process is continued until all pixels belong to some region.



↑ direction of growth • seed pixel

Figure-4.1: Start of growing a region



■ grown pixels o pixels being considered

Figure-4.2: Growing process after few iterations

4.1.3 GEOMETRIC CHARACTERISTICS OF REGIONS

The goal of region growing is to use image characteristics to map individual pixels in an input image to set of pixels called regions. The geometric characteristics of region depends on the domain. Usually they are considered to be connected two dimensional areas. Whether regions can be disconnected, non simply connected (have holes), should have smooth boundary and so forth depend upon region growing technique and the goals of work. Ultimately, it is often the segmentation goal to partition the entire image into quasi disjoint regions, i.e. regions have no two dimensional overlaps and no pixel belongs to the interior of more than one region. However, there is no single definition of region, they may be allowed to overlap, the whole image may not be partitioned, and so forth.

4.2 REGION GROWING TECHNIQUES

The most primitive region growers [31] used only aggregate of properties of local group of pixels to determine regions. More sophisticated, grow regions by merging more primitive regions. To do this in a structured way requires sophisticated representations of regions and boundaries. Also, the merging decisions can be complex, and can depend upon descriptions of the boundary structures seperating regions in addition to the region semantics. Following are the early techniques of region growing:

1. Global Techniques: Pixels are grouped into regions on the basis of the properties of large numbers of pixels distributed throughout the image.
2. Splitting and Merging Techniques: The foregoing techniques are related to individual pixels or sets of pixels. State space techniques merge or split regions using graph structures to represent the regions and boundaries. Both local and global merging criteria can be used.

The effectiveness of region growing algorithms depends heavily on applications area and input image. If the image is sufficiently simple, then simple global techniques will be effective. However on very difficult images, even the most sophisticated techniques still may not produce a satisfactory segmentation. Hence, region growing is sometimes used conservatively to preprocess the image for more knowledgeable processes.

4.2.1 GLOBAL TECHNIQUES

Global techniques involve the region growing via thresholding and region growing via recursive splitting as discussed below.

4.2.1.1 Region Growing via Thresholding

This approach assumes an object background image and picks a threshold that divides the image pixels into either object or background.

X is part of the object if $f(X) > T$

otherwise it is part of background.

The best way to pick the threshold T is to search the histogram of grey levels, assuming it is bimodal, and find the minimum seperating the two peaks. Finding the right valley between the peaks of a histogram can be difficult when the histogram is not a smooth function. Smoothing the histogram can help but does not gurantee that the correct minimum can be found. The elegant method to treat the bimodal images assumes, that the histogram is the sum of two composite normal functions and determines the valley location from the normal parameters.

The single threshold method is useful in simple situations, but primitive. For example, the region pixels may not be connected, and further processing may be necessary to smooth region boundaries and remove noise. A common problem with this technique occurs when the image has a varying background of varying grey level, or for regions that vary smoothly in grey level by more than the threshold. Two modifications of the threshold approach to ameliorate are: (1) High pass filter the image to deemphasize the low frequency background variation and then try the original technique, and (2) Use a spatially varying threshold method.

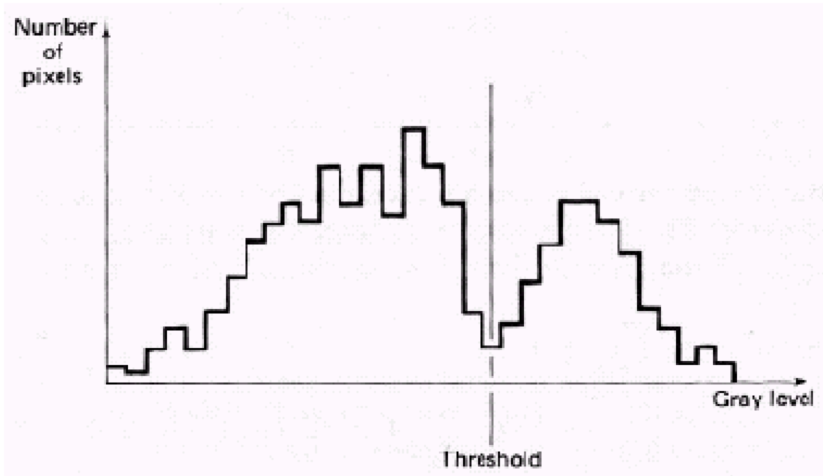


Figure-4.3: Threshold Determination from Gray Level Histogram

Spatially varying threshold method divides the image up into rectangular subimages and computes a threshold for each subimage. A subimage can fail to have a threshold if its gray level histogram is not bimodal. Such subimages receive interpolated thresholds from neighbouring subimages that are bimodal, and finally the entire picture is thresholded using the separate thresholds for each subimage.

4.2.1.2 Region Growing via Recursive Splitting

In this technique, the entire image is considered as a region and histograms are computed for each of the picture vector components. Peak finding test is applied to each histogram. If at least one component passes the test, the component with the most significant peak is picked and two thresholds are determined, on either side of the peak. These thresholds are used to divide the region into subregions. Each subregion may have a “noisy” boundary so the binary representation of the image achieved by thresholding is smoothed so that only a single connected subregion remains. This procedure is repeated for each subregion until no new subregions are created (no histograms have significant peaks). Multiple regions are often in the same histogram peak when a single measurement is used. The advantage of the multimeasurement histograms is that these different regions are often seperated into individual peaks, and hence the segmentation is improved.

4.2.2 SPLITTING AND MERGING TECHINQUE

The merge split algorithm due its use of criteria based on the difference between the maximum and minimum pixel values within the region tends to act like an edge detection algorithm. In smooth (no noise or textures) and low gradient images, edges are the only areas where large differences in pixel values tend to occur. As a result near the edges, merge and split algorithm tends to split blocks down to individual pixels. Large merged blocks appear in the interiors. So for this class of images, merge and splitting is an effective first stage in segmentation, and region growing can take place faster. For images with complex sub regions, fine detail, patterns, and gradients such as the plane, merge splitting with max-min criteria doesn't buy that much. Too low a merge split threshold creates too many small pixel size regions. Too high merge split threshold creates too many large blocky regions. Using merge splitting prior to region growing tends to result in sharper edges whereas, region growing without merge splitting generated images with blurry edges.

4.2.2.1 Region Growing via Split and Merge

In this technique any grid structure is chosen, and homogeneity property H. If for any region R in that structure $H(R)$ = false, split that region into four subregions. If for any four appropriate regions R_{K1}............R_{K4}, $H(R_{K1} \bigcup R_{K2} \bigcup R_{K3} \bigcup R_{K4})=$ true, merge them into a single region. When no regions can be further split or merged, stop. A significant simplification results merging of any two adjacent regions R_i and R_j is allowed and if each one satisfies predicate individually. This results in much simpler and faster algorithm because testing of the predicate is limited to individual quad regions. All the quad regions which satisfy the predicate are filled with one's and their connectivity can be easily examined. The quad regions that do not satisfy the predicate are filled with zero's to create a segmented image. Block diagram shown in figure 4.4 describes the complete split and merge region growing algorithm.

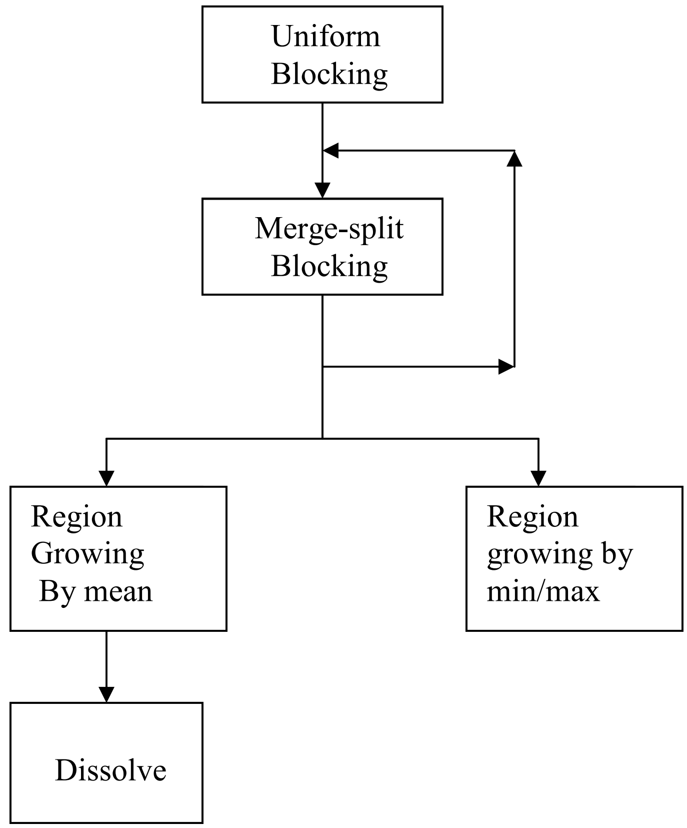


Figure-4.4: Block Diagram of Merge-Split Region Growing Algorithm

- **Uniform Blocking:** Uniform blocking is the first step in any of our algorithms. This step involves dividing the images into uniform blocks for processing. Typically 2 X 2 blocks are used if region growing was to be employed directly or 16 X 16 blocks if the merge-split algorithm was to be used. It shouldn't matter what block size is fed into the merge split routine, but picking an intermediate value enhances the speed for most images.

- **Merge Split Blocking:** The merge split routine is an optional stage of region growing based segmentation scheme. It requires a threshold as an input. This threshold determines which blocks can be merged into a single block and which blocks can be split into smaller blocks based on the difference between the maximum and minimum intensities in each block. If the max-min difference of a block is close to the max-min difference of its neighbors (i.e., difference between blocks is within the threshold), then the blocks are merged into a single block. A block is split in half if the max-min difference of the block exceeds the threshold. The merge split mechanism is a quad tree structure, meaning that the merging and splitting of blocks goes from 4 to 1 and 1 to 4 respectively. This process is done recursively until, no blocks satisfy the criteria to be split or merged. Thus a block whose max-min difference exceeds the threshold will continue to be split until the max-min difference of the subsequent block(s) are within the threshold or the block size reaches one pixel, in which case the max-min difference is zero. There is also a minimum block size argument which allows the user to specify the smallest block size that can be generated through splitting. This allows the user to force the segmenting algorithm to end up with a small number of regions by ensuring that the output of the merge split algorithm has blocks that are no smaller than a specified size. Without this feature there is a potential for the merge split routine to return many small blocks. If these blocks are not successfully merged by the region growing algorithm, undesirable results are likely.

- **Region Growing by Mean or Max-Min:** Region growing is done by examining properties of each block and merging them with adjacent blocks that satisfy some criteria. One of two criteria's is used. One criterion is to look at the max-min difference and combine adjacent regions whose max-min difference is within a tolerance of the seed blocks. The new region is now the seed and the process is repeated, examining adjacent regions, comparing max-min differences, and adding blocks that are within the tolerance specified by the user. This tolerance does not have to be the same as the threshold used in the merge split algorithm. Alternatively, the mean values of the blocks can be used to determine which blocks should be merged. The max-min algorithm did a better job of preserving edges and handled some textures better than the mean algorithm. The mean algorithm did better on images with speckle.

- **Dissolve:** The dissolve algorithm works in conjunction with the mean based region growing to merge regions that are less than a specified size into the adjacent region with the closest mean value. This process helps give a segmented image that corresponds more to the segmentation that a human would do by hand. The number of regions is reduced by eliminating the less significant regions, avoiding an excessive amount of segmentation. The results of splitting-merging algorithm are shown in figure 4.5

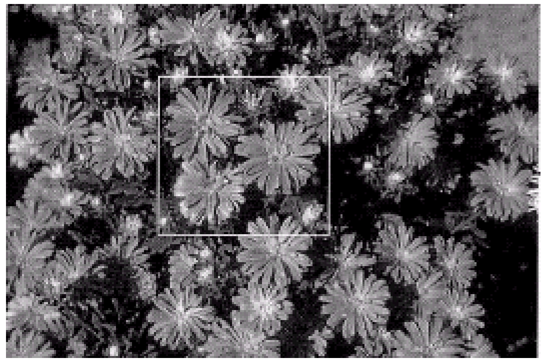


Figure-4.5(a): Original Image

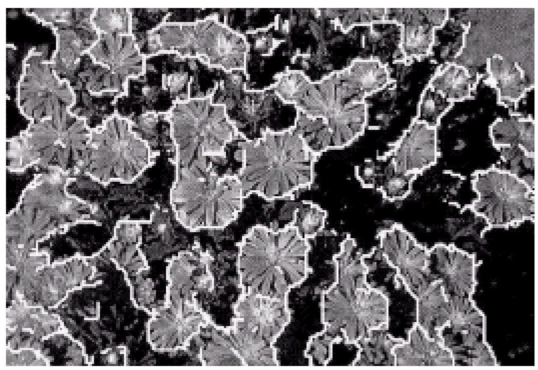


Figure-4.5(b): Splitting of Image

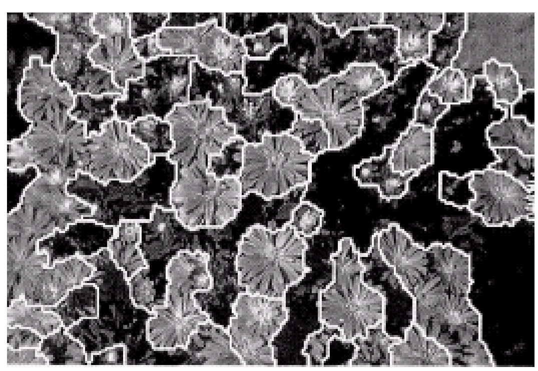


Figure-4.5(c): Merging of Sub Regions

4.2.2.2 Region Growing via Edge Detection

It is a method that combines region growing and edge detection for image segmentation. Split and merge algorithm is considered initially where the parameters have been set up so that an over segmented image results. Then region boundaries are eliminated or modified on the basis of criteria that integrate contrast with boundary smoothness, variation of the image gradient along the boundary, and a criterion that penalizes for the presence of artifacts reflecting the data structure used during segmentation.

4.2.3 SEEDED REGION GROWING

The seeded region growing (SRG) method described in this section is a greedy algorithm, closely related to the watershed transform, which assigns a label to every pixel in the image while satisfying a connectivity constraint. The technique begins with a set of seeds that mark the regions to be segmented and uses a priority based system to grow regions one pixel at a time. This means that a single seed will grow to fill the entire image if there are no other seeds to compete with it. Other approaches place a threshold and stop a region growing when the threshold is exceeded, so a single seed will not necessarily grow to fill the entire image.

A region growing method starts the growing process from a selected initial point or seed and evolves regions iteratively until a region of interest (ROI) is obtained. The nature of region growing leads to (i) where it starts, i.e., the search of seeds whether specified implicitly or explicitly, (ii) how it grows, i.e., the growing conditions and (iii) when it stops growing process. It is a region growing method which starts the growing process from selected initial points or seeds and evolves regions iteratively until maximal intra region homogeneity is reached. The image points that are not associated to any seed are related to the background. The nature of region growing leads to (i) the search of seeds whether specified implicitly or explicitly, and (ii) unbridled growing directions. The former implies the importance of seed selection, and thus scanning the image to determine seeds is prerequisite of the segmentation.

Regions can also be grown in an unseeded manner using split and merge or graph based approaches in [17]. Greedy approaches are parameter free, but require selection of seeds by some form of pre filtering or manual interaction.

4.2.3.1 Seed selection criteria

Seed selection basically depends on the nature of the problem. If target needs to be detected using infrared images, then the brightness pixels are chosen. Some region growing approaches exploit the important fact that the pixels which are close together have similar grey level values. So without a priori knowledge, the histogram is computed and the grey level values corresponding to the strongest peaks are chosen.

4.2.3.2 Similarity criteria (predicates)

The homogeneity predicate can be based on any characteristic of the region in the image such as

- Average intensity
- Variance
- Color
- Texture
- Motion
- Shape
- Size

4.2.3.3 Growing process

A set of seeds is considered, in which seeds are grown until each image pixel is allocated to one of the set of seeds. This growth is controlled by a priority queue mechanism which is dependent on the statistics of the image (or its gradient) covered by a region. The final shape of a region is not controlled by any higher level, problem dependent knowledge besides the marker selection. This can be both to an advantage and a disadvantage. For example it may be desirable to extract regions with smooth boundaries. Existing region growing algorithms cannot guarantee this. The seeded region growing (SRG) method described in this section is a greedy algorithm, closely related to the watershed transform, which assigns a label to every pixel in the image while satisfying a connectivity constraint. The technique begins with a set of seeds that mark the regions to be segmented and uses a priority based system to grow regions one pixel at a time. This means that a single seed will grow to fill the entire image if there are no other seeds to compete with it. Other approaches place a threshold on the value of δ and stop a region growing when the threshold is exceeded, so a single seed will not necessarily grow to fill the entire image. Regions can also be grown in an unseeded manner using split and merge or graph based approaches in [32]. Greedy approaches are parameter free, but require selection of seeds by some form of pre filtering or manual interaction. SRG takes two images as input: a control image I and a seed image S. I can be virtually single discrete grey level image, on an 8-connected grid. S contains a collection of labeled binary regions, $S_i \subset S, i \in 1,2,3...n$

Regions with the same label do not have to be connected. Let U be the set of pixels unassigned to any region but connected to atleast one of them:

$$U = x \in S, \forall i, x \notin Si, N(x) \bigcap \cup Si = 0 \qquad \text{- (4.1)}$$

where, $N(x)$ is the immediate neighbourhood of x. At each step of the algorithm, all the points $x \in U$ are examined in turn and a distance $\delta(x,Si)$ is computed between x and all the Si it is neighbouring, i.e: $Si, N(x) \cap Si \neq 0$. The pixels y that possesses the minimum distance is assigned to their neighboring region, and the process is repeated until all the are associated with a region.

Typically, the distance δ can be defined as follows:

$$\delta\,(x, Si) = |I(x)\ .\ mean_{y \in Si}\ (I(y))| \qquad \text{- (4.2)}$$

where, $I(x)$ is the value in I at x.

4.2.4 NOVEL REGION GROWING

A novel region growing method is used for ultrasound image segmentation. In this method, region growing is performed in a multi-feature vector space. It combines different texture features, gray scale and gradient together to construct a multi feature vector space in which tissue homogeneity is measured and difference between different classes are calculated.

Three novel criteria are developed for region growing control.

- Global comparison detection: In this instead of using the local information as do the conventional region growing methods. This method uses global information.
- Geography priority privilege: In this to overcome the effects of speckle noise and attenuation artifacts, a new idea termed, “geographic similarity”, is introduced.
- Equal opportunity competence: An equal opportunity competence criterion is employed to make results independent of processing order.

This segmentation results in vivo intra cardiac ultra sound images and the corresponding statistical analysis shows that this method is reliable and effective. If the features are specially selected this method can be used for other kinds of medical images such as CT, MRI.

4.2.5 LAYERED REGION GROWING

The segmentation of linear structures in a gray-scale image is widely used in a variety of image analysis and visualization tasks. In case of biomedical image analysis of volumetric data (such as CT or MRI scans), it is very common that a linear structure (blood vessel) is attached to a sheet-shaped (planar) and/or blob-shaped (ellipsoid) region (large tissue or fluid mass such as cistern basin in the brain) but where the image intensity of both structures is the same. Segmentation algorithms (threshold based or statistical approaches) are unable to clearly define and therefore segment the areas of such attachment. Automatic detection of the attachment and locating the start and end of linear structures is important for visualization and quantification in biomedical analysis. Layered region growing (LRG) algorithm in [34] is used to segment the linear structures from volumetric data. LRG algorithm for volumetric data presents a layer frame for extracting the region of interest, and introduces a width function on the layer. Width function is of constant order if LRG is used to extract a linear structure whatever the selection of seed, $O(n)$ for a sheet-shape, and $O(n^2)$ for a blob-shape, where n is the number of layers in the region. An automatic method to detect the change of the width function as a criterion is used to stop the region growth at the attachment areas where different shaped structures meet. This region growing method works well on simulated data as well as on cerebral vascular CT data and is used to segment vascular structures in biomedical images.

4.2.6 REGULARIZED SEEDED REGION GROWING

Region growing algorithms, including seeded region growing, are a widely employed technique used in image segmentation problems. It is common for region growing techniques to employ queue based techniques that sequentially add pixels to regions. These algorithms are very flexible, but it is sometimes desirable to be able to apply some constraints to the growing process that reject some higher level knowledge about the problem. Regularized seeded region growing [35] is a technique that constrains (regularizes) the border smoothness of a growing region using a polygonal representation and provides some results illustrating a possible application. The technique can be usefully combined with traditional, unregularized techniques.

Regularized region growing permits a combination of both regularized and unregularized regions. Seeded region growing is applied on unregularized regions. Whereas, regularized regions are more complex. The seed image S contains a collection of labeled binary regions $S_i \subset S, i \in 1,2,3....n$. Each regularized region S_i is represented by a closed contour consisting of a set of corner points, $P_i \; i \in 1,2,3....n$, connected by straight line segments i.e. a polygon. Point $P_{i,k}$ is connected to $P_{i,k-1}$ and $P_{i,k+1}$. P_{in} connects to P_{i1}. The maximum distance between adjacent corners is the regularization factor denied by the user. Each corner P_{ik} may move to a number of new locations in the 4 or 8 connected local neighborhood. A move will increase the number of pixels belonging to a region. The best move for a corner is the one with the minimum average pixel distance $\delta c = 1/m \sum_m \delta i$, where, δi is the distance measure used for pixels in unregularized region growing and m is the number of pixels labeled by a given move. The best move is computed for each corner and placed in a priority queue with priorities δc. This queue may also contain border pixels from unregularized regions. The region growing proceeds in the conventional fashion, with the highest priority elements being removed from the queue. If the element is a border pixel from an unregularized region it is processed in the conventional way (i.e. by adding neighbors). If it is a corner element from a regularized region then the relevant pixels are assigned to the region and the region mean is updated. The best new move is computed and the corner is re-queued.

If the distance to a neighboring corner is greater than the user denied value, then the edge is split by adding a new corner at the midpoint. The best move for this corner is computed and placed on the queue.

The regularized algorithm may be used on its own or in conjunction with the unregularized version. When an image is segmented using only regularized seeds there are likely to be many unassigned pixels in the area where the regularized regions meet. A common reason for using regularized regions is to prevent a region growing through a small gap. In these situations it may be useful to use regularized regions for "foreground" objects and unregularized regions for background ones.

4.2.7 SEED INVARIANT REGION GROWING

This method also known as symmetric region growing [36] is suitable for processing medical CT images. It relates not only to computation, but also memory usage in the midst of region growing segmentation - computational seperability. All the experimental results are shown in [37]. As shown in figure 4.6, operation flows of the conventional region growing process and the symmetric region growing (SymRG) process. *I* denote the input image; *S* represents the seed criteria for selecting the initial growing pints. The region growing criteria ψ include the inclusion criteria *I* that specify homogeneous properties of a region and the exclusion criteria *X* that rule out certain points in any of the regions of interest. The traditional region growing process [38], [39] begins with filtering the input image with the seed and exclusion criteria to enter the pre growing state. It then iteratively applies the inclusion criteria to give the final segmented result $S\ (I,\ RG\ (\psi),\ S)$. The SymRG process gives an equivalent segmentation result $S\ (I,\ RG\ (\psi),\ S)$, but has reorder the process quite a bit. It begins with filtering out those points that are excluded from any of the desired regions. The remaining points then evolve regions per inclusion criteria. Finally the seed criteria applied to determine the final segmentation.

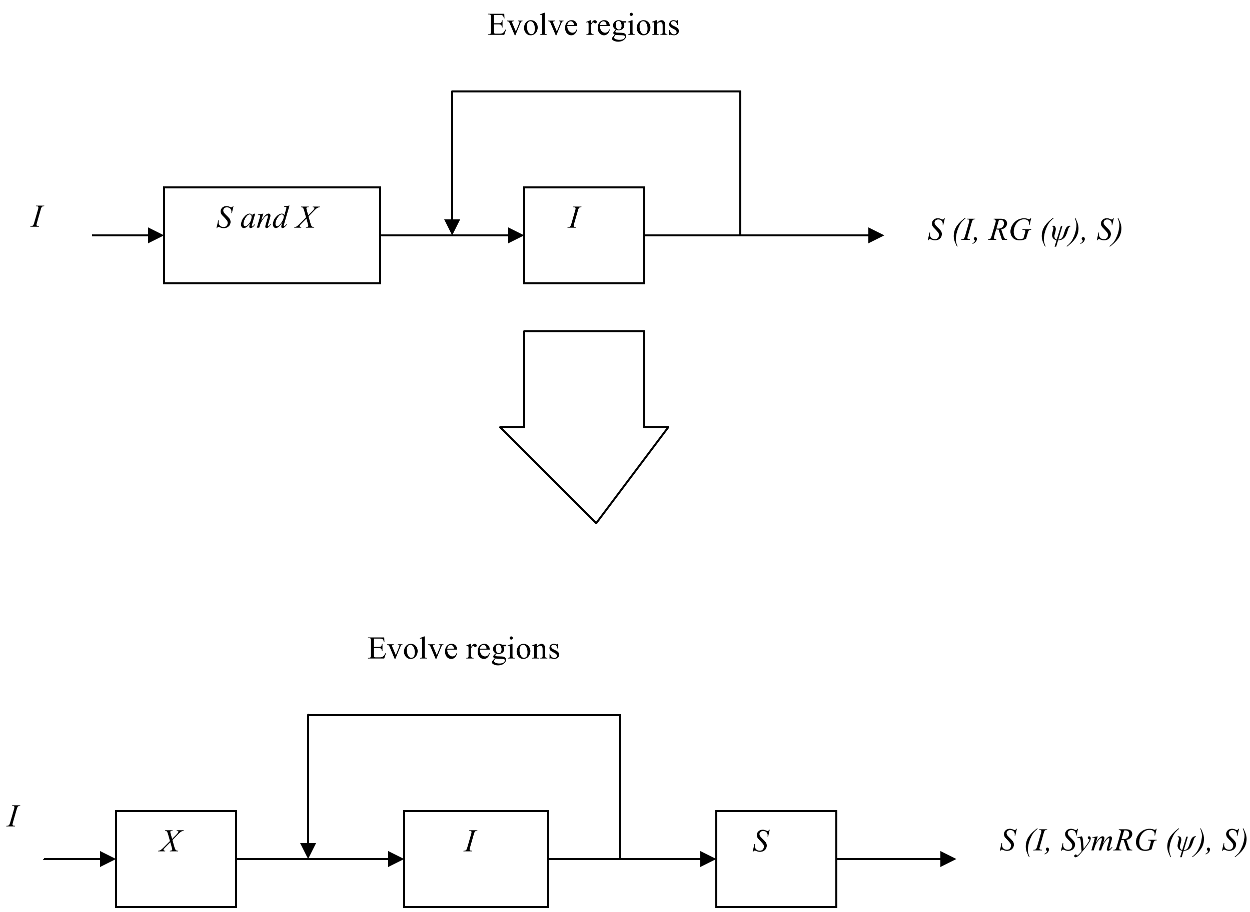


Figure-4.6: Conventional Region Growing Process

The conventional region growing process expands a region to any arbitrary direction iteratively or recursively. It implies that an image point can be visited at any time during the course of segmentation - the whole original image must be available at all time. Also, it is difficult to analyze the growing behavior. On the other hand, SymRG process can be carried on in one direction, and therefore the past image data can be discarded and it is possible to perform growing behavioral analysis.

CHAPTER 5

INTEGRATING REGION GROWING AND EDGE DETECTON

5.1 INTRODUCTION

In chapter 3 and chapter 4 two of the main segmentation techniques, edge detection and region growing are described briefly. In this chapter both edge detection and region growing techniques are merged together. The goal of this chapter is to show how combining two techniques for segmentation produces much better results than what would have been obtained by either technique alone. This is not surprising; the general principle of integration is well accepted in vision. But carrying out the general principle is quite challenging and this may explain the sparsity of the literature on integrated techniques. Thus the contribution of this is to show how such integration can be done. Besides integrating edge detection and region growing, different criteria for edge detection: contrast and smoothness of the resulting curve are also integrated.

5.2 INTEGRATING REGION GROWING AND EDGE DETECTION

The approach used for integrating edges and region is, first the edges of the image are detected using any optimum edge operator. Then the edge region is detected. Edge region is defined as the place where the region growing seeds will be selected. Therefore the edge region should surround the single pixel edges derived by an edge operator. Then a region size comparing is done. Very small regions are removed from edge region are removed instead of merging. Thus the effect of noise is completely eliminated. When this is done, the image is segmented in two kinds of areas, one is edge region and another is homogeneous region. In this work first edge detection is performed, then edge region detection and then seeded region growing.

Input Image

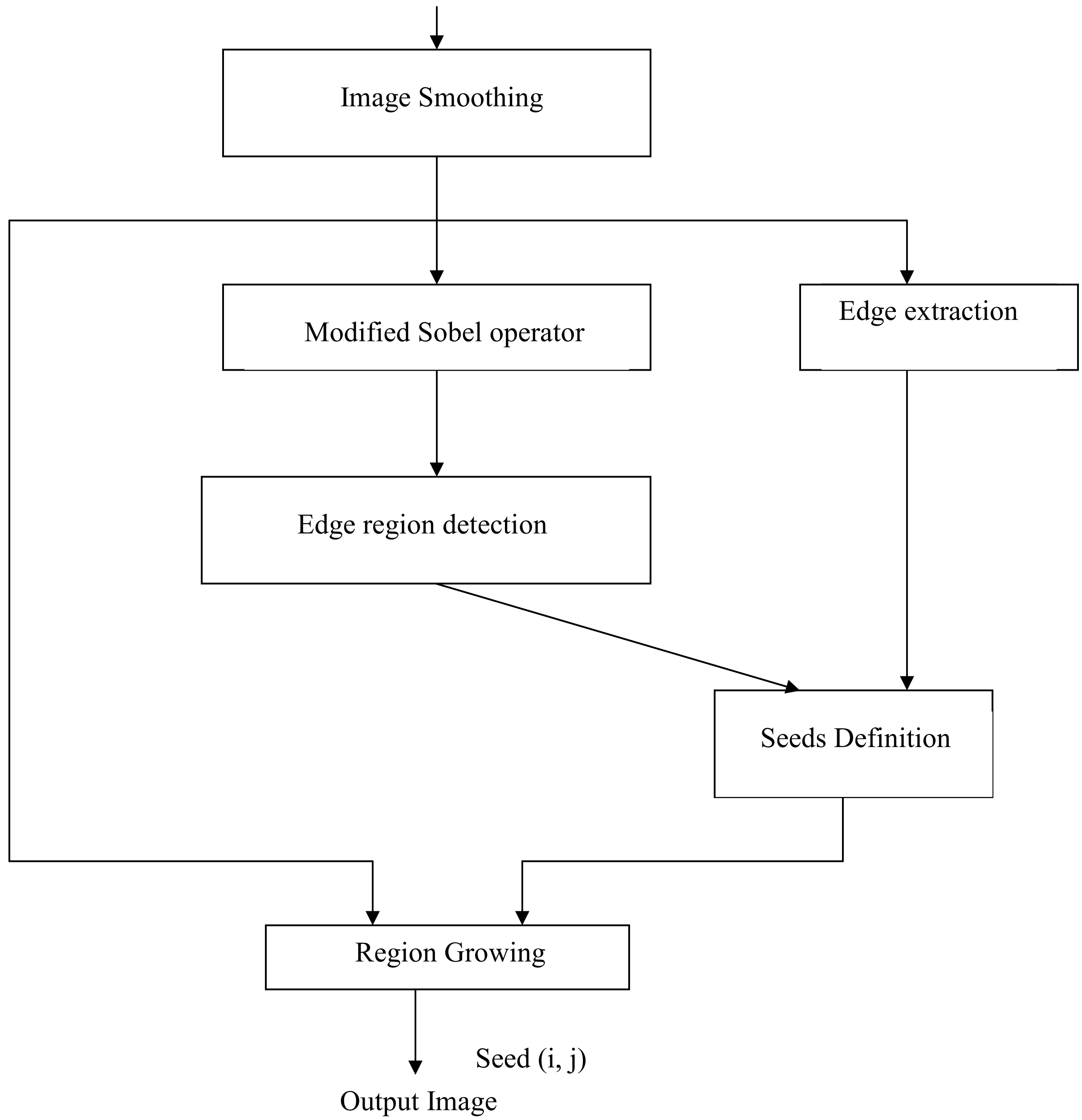


Figure-5.1: Flow chart for Integrating region growing and Edge Detection

As shown in figure 5.1 following steps are followed for integrating edge extraction and region growing for image segmentation.

The edge detector detects the edges and seed points for region growing are derived from the edge image. The region growing requires the input of a number of seeds, either individual pixels or regions, which will control the formation of regions into which the image will be segmented. Regions are grown from the current pixel relative to the region pixel and edge pixel. Image edges act as hard barriers during region growing. The growing seeds selected are based on the edge image and the edge region image. Two kinds of seeds (pixels) - hot and cold, are defined near the edges (of objects) and both kinds of regions are grown from these seeds simultaneously. The hotness degree of each pixel in the edge image is calculated with respect to its [5 X 5] neighborhood. If the current pixel is not an edge pixel or not located in edge region then it will remain zero. The rest of the pixels are queued in [5 X 5] buffer and the threshold T is computed. The threshold value corresponds to the median of all the pixels in [5 X 5] matrix. The pixel values greater than the threshold T are increased by one and the pixel values less than threshold T are decreased by one. This procedure is carried on until all the pixel values in the image are scanned.

5.2.1 PREPROCESSING

Before the region growing seeds can be selected, the image needs to be preprocessed to convert it into more suitable form. Preprocessing of an image introduces following three main stages:

5.2.1.1 Image smoothing

Image smoothing is the set of local pre-processing methods which have the aim of suppressing image noise – it uses redundancy in the image data. Smoothing poses the problem of blurring sharp edges in the image. To avoid blurring of sharp edges the edge preserving image smoothing techniques are introduced. They are based on the general idea that the average is computed only from those points in the neighborhood which have similar properties to the processed point. Local image smoothing can effectively eliminate impulsive noise or degradations appearing as thin stripes, but does not work if degradations are large blobs or thick stripes.

An averaging filter can be used to reduce the effect of high frequency noise in the image. At the same time a sharp edge is dilated into an edge region by the smoothing process. The Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased. The Gaussian mask used in my implementation is shown below.

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Figure-5.2: Gaussian mask applied on Image

This process ensures maximum noise removal and to make the edges more prominent to detect.

5.2.1.2 Edge Extraction

Edge detectors locate sharp changes in the intensity function .Edges are pixels where brightness changes abruptly. Calculus describes changes of continuous functions using derivatives; an image function depends on two variables – partial derivatives. A change of the image function can be described by a gradient that points in the direction of the largest growth of the image function. An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighbourhood of the pixel. A change of the image function can be described by a gradient that points in the direction of the largest growth of the image function. An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighbourhood of the pixel.

It is a vector variable

- magnitude of the gradient
- direction φ

The gradient direction gives the direction of maximal growth of the function, e.g., from black $(f(i,j)=0)$ to white $(f(i,j)=255)$. This is illustrated below; closed lines are lines of the same brightness. The orientation 0° points east.

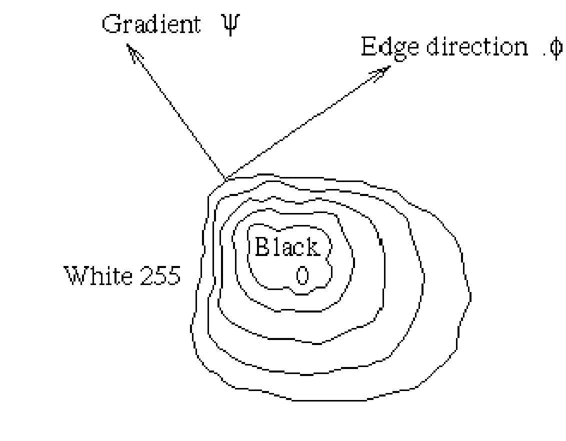


Figure-5.3: Gradient direction and Edge direction

Among all the edge detectors, canny edge detector is used because it is optimal in detecting edges corrupted by white noise. It is related to three principles:

- Detection: Important edges should not be missed and there should be no spurious responses.
- Localization: Distance between the actual and located position of the edge should be minimal.
- Single response: Minimizes multiple responses to a single edge (also partly covered by the first criterion since when there are two responses to a single edge one of them should be considered as false).

The canny filter is selected to perform the edge detection. The canny operator works in multistage process. Various steps followed for canny filtering to detect edges are:

Step1

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm.

Step2

After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found.

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \qquad \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Figure-5.4: Horizontal and Vertical Sobel Operators

The Sobel operator uses a pair of 3 X 3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). The magnitude, or edge strength, of the gradient is then approximated using the formula:

$$|G| = |Gx| + |Gy| \qquad - (5.1)$$

Step3

Finding the edge direction is trivial once the gradient in the x and y directions are known. However, an error is generated whenever sum is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If Gy has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is

$$Theta = inv \tan Gx / Gy \qquad -(5.2)$$

Step4

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

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

Figure-5.5: 5 X 5 Sub matrix including edge pixel "a"

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

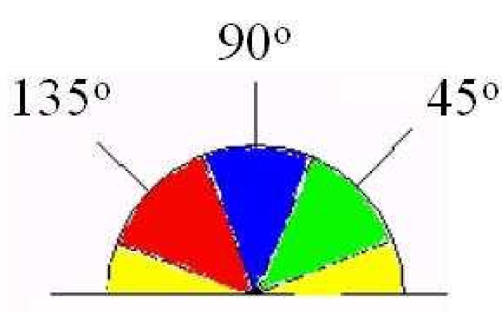


Figure-5.6: Semicircle representing pixel directions

Therefore, any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the red range (112.5 to 157.5 degrees) is set to 135 degrees.

Step5

After the edge directions are known, nonmaximum suppression is applied. Nonmaximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Step6

Hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, t_1 is applied to an image, and an edge has an average strength equal to t_1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than t_1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than t_2 are also selected as edge pixels. If you think of following an edge you need a gradient of t_2 to start but you don't stop till you hit a gradient below t_1.

Following are the results for canny edge detection algorithm.

(a) **(b)**

Figure-5.7(a): Original Nina Image, (b): Result of Canny Edge Detector

5.2.1.3 Edge Region Detection

The edge region is the place where the region growing seeds will be selected. Therefore the edge region should surround the single pixel edges derived by the canny operator. In this approach, it is assumed that the edge magnitude and direction are available at every pixel. They can be computed by using and edge operator such as sobel operator which can be efficiently implemented by a series of direction masks such as 6 masks in steps of 30^0 which assign each pixel the L_∞ norm of the edge magnitude outputs and the corresponding direction. The choice of an edge operator depends on the quality of the edges present in the image, their location, response, and susceptibility to noise. Based on this one-dimensional analysis, the theory can be carried over to two-dimensions as long as there is an accurate approximation to calculate the derivative of a two dimensional image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image. The Sobel edge detector uses a pair of 3 X 3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The actual Sobel masks are shown below:

1 2 1   2 1 **0**   3 3 1

$$M_1 = \begin{array}{ccc} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -1 & -2 & -1 \end{array} + \begin{array}{ccc} 1 & \mathbf{0} & -1 \\ \mathbf{0} & -1 & -2 \end{array} = \begin{array}{ccc} 1 & \mathbf{0} & -1 \\ -1 & -1 & -3 \end{array}$$

$$M_2 = \begin{array}{ccc} -1 & -2 & -1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & 2 & 1 \end{array} + \begin{array}{ccc} -2 & -1 & \mathbf{0} \\ -1 & \mathbf{0} & 1 \\ \mathbf{0} & 1 & 2 \end{array} = \begin{array}{ccc} -3 & -3 & -1 \\ -1 & \mathbf{0} & 1 \\ 1 & 3 & 3 \end{array}$$

$$M_3 = \begin{array}{ccc} 1 & \mathbf{0} & -1 \\ 2 & \mathbf{0} & -2 \\ 1 & \mathbf{0} & -1 \end{array} + \begin{array}{ccc} \mathbf{0} & -1 & -2 \\ 1 & \mathbf{0} & -1 \\ 2 & 1 & \mathbf{0} \end{array} = \begin{array}{ccc} 1 & -1 & -3 \\ 3 & \mathbf{0} & -3 \\ 3 & 1 & -1 \end{array}$$

$$M_4 = \begin{array}{ccc} -1 & \mathbf{0} & 1 \\ -2 & \mathbf{0} & 2 \\ -1 & \mathbf{0} & 1 \end{array} + \begin{array}{ccc} \mathbf{0} & 1 & 2 \\ -1 & \mathbf{0} & 1 \\ 2 & -1 & \mathbf{0} \end{array} = \begin{array}{ccc} -1 & 1 & 3 \\ -3 & \mathbf{0} & 3 \\ -3 & -1 & 1 \end{array}$$

M_1 and M_2 detect edges in the "--" and "/" directions respectively and M_3 and M_4 detect edges in "|" and "\" directions respectively. M_1, M_2, M_3, and M_4 are the four sobel masks which are applied on the image to detect edge region.

The mask is slid over an area of the input image, changes that pixel's value and then shifts one pixel to the right and continues to the right until it reaches the end of a row. It then starts at the beginning of the next row. The center of the mask is placed over the pixel you are manipulating in the image. It is important to notice that pixels in the first and last rows, as well as the first and last columns cannot be manipulated by a 3 X 3 mask. This is because when placing the center of the mask over a pixel in the first row (for example), the mask will be outside the image boundaries. So to avoid this problem zero padding is done i.e. two rows and two columns with zero padding are introduced on all the four sides of the image matrix on which the mask is being applied. As shown in the figure 5.

A_{11}	A_{12}	A_{13}	…..	A_{1k}
A_{21}	A_{22}	A_{23}	…..	A_{2k}
A_{31}	A_{32}	A_{33}	….	A_{3k}
:	:	:	:	:

Input Image

M_{11}	M_{12}	M_{13}
M_{21}	M_{22}	M_{23}
M_{31}	M_{32}	M_{33}

Mask

B_{11}	B_{12}	B_{13}	…..	B_{1k}
B_{21}	B_{22}	B_{23}	…..	B_{2k}
B_{31}	B_{32}	B_{33}	…..	B_{3k}
:	:	:	:	:

Output Image

$$B_{22}=(A_{11}*M_{11})+(A_{12}*M_{12})+(A_{13}*M_{13})+(A_{21}*M_{21})+(A_{22}*M_{22})+(A_{23}*M_{23})+(A_{31}*M_{31})+ (A_{32}*M_{32})+ (A_{33}*M_{33})$$

Figure-5.8: Mask Sliding Technique

This example shows the mask being slid over the top left portion of the input image. The formula shows how a particular pixel in the output image would be calculated. The center of the mask is placed over the pixel you are manipulating in the image. I & J values are used to move the file pointer multiplication can be done. For example, pixel (A_{22}) by the corresponding mask value (M_{22}). As shown below in figures 5.8(a), (b) and (c), the sobel operator amplifies the image twice.

Figure-5.9(a): Original Nina Image

Figure-5.9(b): Noisy Sobel Edge Region Image

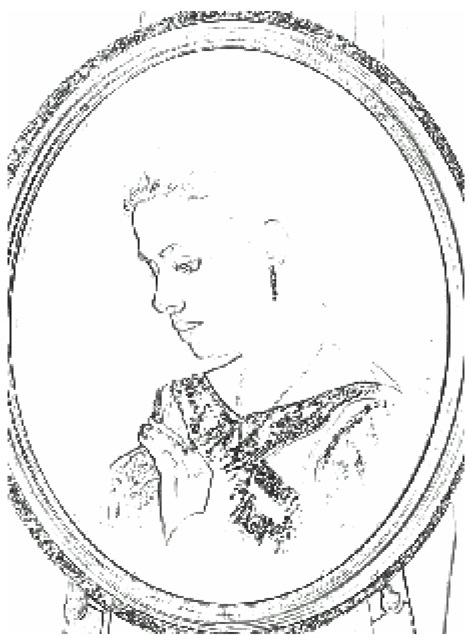


Figure-5.9(c): Sobel Edge Region Image divided by factor 2

5.2.2 DEFINITION OF REGION GROWING SEEDS

Basic definition of growing seeds is that there are two kinds of seeds (pixels)-hot and cold, are defined near edges (of objects) and both kinds of regions are grown from these seeds simultaneously. Selection of growing seeds based on the edge image and the edge region image. In the infrared image the objects to be segmented always appear red (hot) and the background appears black (cold). The same analogy is used in the segmentation method proposed here. The seed image has black for hot and grey for cold. Selection of seeds is based upon homogeneity criteria and can be done in semi automated way or manually.

5.2.3 REGION GROWING

Region growing is based on the fact that the grey levels of the hot seeds are lower than the pixels not far away from the edge region in the hot object and the grey levels of the cold seeds is higher than the pixels not far away from the edge region of the cold background. Thus the seeds grow into their respective regions to give a segmented binary image, which is the final output image. In all the region growing algorithms criteria of similarity of pixels is applied, but the mechanism of region growing is closer to the watershed algorithm. Instead of controlling region growing by tuning homogeneity parameters, is controlled by choosing a usually small number of pixels, known as seeds. Initially a single seed is chosen. Then its neighboring seeds are compared one by one with the seed chosen initially. If the homogeneity criteria match then same types of seeds are grouped together and hence region is grown.

Fig-5.10(a): Original Aircraft Image

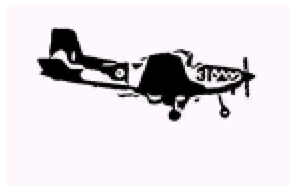


Fig-5.10(b): Seeded Region Grown Image

5.3 RESULTS AND DISCUSSIONS

The different image segmentation techniques are applied on five reference images i.e. Rice image (256 X 256), Block image (117 X 115), Barbara image (237 X 239), Lina image (256 X 256), and vegetable image (214 X 849) as shown in figures 5.11, 5.21, 5.31, 5.41 and 5.51.

Canny edge detector is one of the most powerful edge detector introduced by function edge in MATLAB. The basic principle behind using canny edge detector is to produce clean edge map by extracting the principle edge features of an original image. It will reduce the irrelevant detail, such as fine texture. The principle edges of interest are the shapes of objects in an image and building corners. As shown in the figures 5.12, 5.22, 5.32, 5.42 and 5.52 edges are found by applying canny edge operator on five reference original images with two thresholds. Following are the two threshold values applied, t2=0.10 and t_1=0.04, where t_2>t_1. The pixel values lying between these two thresholds are considered to be edges. Nonmaximum suppression technique is applied to generate ridges which further results in thin line of edges. The Gaussian smoothing in the canny edge detector fulfills two purposes: first it can be used to control the amount of detail that appears in the edge image and second, it can be used to suppress noise.

The edges detected without applying any threshold, shows that very fine detail of discontinuities in image is not detected as shown in figures 5.13, 5.23, 5.33, 5.43 and 5.53. The difference between the two is that the canny edge detector without threshold is able to detect the very fine discontinuities in an image as edges.

Similarly, when sobel edge detection technique is applied without threshold, the edges corresponding to the unevenness of the surface disappeared from the image, but some edges corresponding to changes in the surface orientation remain. Although these edges are weaker than the boundaries of the objects, the resulting pixel values are the same, due to the saturation of the image. Hence, when the image is scaled down before the edge detection, upper threshold of the edge tracker is used to remove the weaker edges. So the image pairs of 3 X 3 convolution masks are applied. One mask is simply the other rotated by 90^0. Natural edges in images often lead to lines in the output image that are several pixels wide due to the smoothing effect of the Sobel operator. As shown in figures 5.14, 5.24, 5.34, 5.44 and 5.54, unlikely canny edge detection technique using threshold, the sobel edge detection techniques failed to detect the minor discontinuities when no threshold is used. Also the edges detected are broken and does not form close boundaries. Whereas, when threshold is applied very minor discontinuities are also detected by a sobel operator and continuous contours are obtained as shown in figures 5.15, 5.25, 5.35, 5.45 and 5.55. Hence sobel operator is not able to detect the edges of the object while removing all the noise in the image.

Sobel edge region image is obtained when the four masks obtained by pairing the original four sobel convolution masks in the following four methods:

- Vertical mask + 45^0 mask
- Vertical mask + (-45^0) mask
- Horizontal mask + 45^0 mask
- Horizontal mask + (-45^0 mask)

When all these four masks are applied and absolute value of each pixel value is obtained the resulting images are as shown in figures 5.16, 5.26, 5.36, 5.46 and 5.56. Edges obtained form the thick lines of the width of few pixels. Hence these thick lines are known as edge regions. It is observed that the sobel edge region image contains noise. But on dividing each pixel value of these edge region images by amplification factor of 2, fine details of the image edges are obtained shown in figures 5.17, 5.27, 5.37, 5.47 and 5.57.

Sobel edge region images are then binarised to thicken the edge regions shown in figures 5.18, 5.28, 5.38, 5.48 and 5.58. The threshold for the binary image is chosen calculating mean of maximum and minimum pixel value. It is much easier to detect the hot and cold seeds from the binarised image.

In edge region hot degree and cold degree seeds are obtained. For this all the pixel values of canny edge detected image and binarised edge region image are compared with each other by applying certain set of conditions. As shown in figures 5.19, 5.29, 5.39, 5.49 and 5.59, it is observed that the seed image contain two types of regions, homogeneous and edge region. Homogeneous region contains no seeds, whereas, edge region contains two types of seeds namely hot and cold.

Region growing is based on the fact that the grey levels of the hot seeds are lower than the pixels not far away from the edge region in the hot object and the grey levels of the cold seeds is higher than the pixels not far away from the edge region of the cold background. Thus the seeds grow into their respective regions to give a segmented binary image, which is the final output image as shown in figures 5.20, 5.30, 5.40, 5.50 and 5.60.

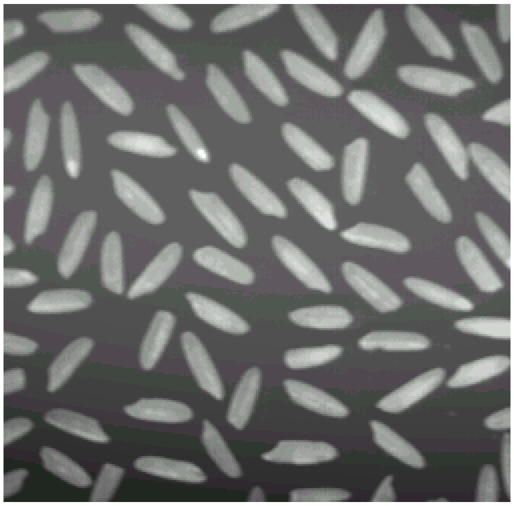


Figure-5.11: Original Rice Image (256 X 256)

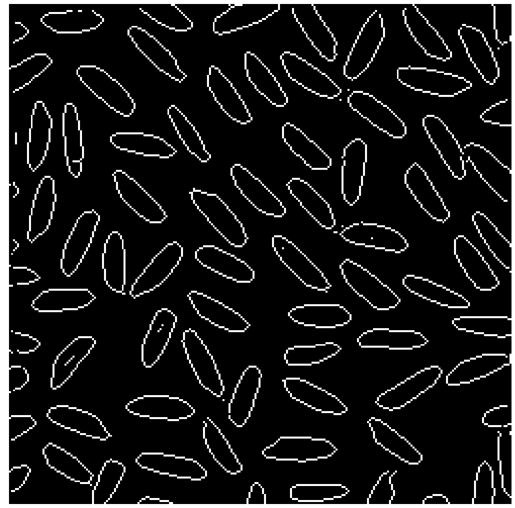


Figure-5.12: Canny Edge Detected Image, t_2=0.1, t_1=0.04, sigma=1.5

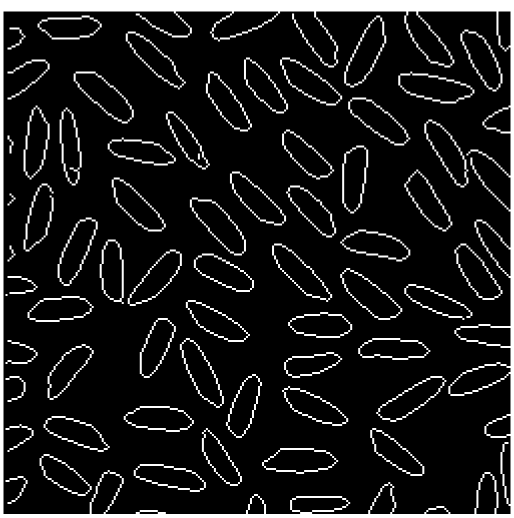


Figure-5.13: Canny Edge Detected Image

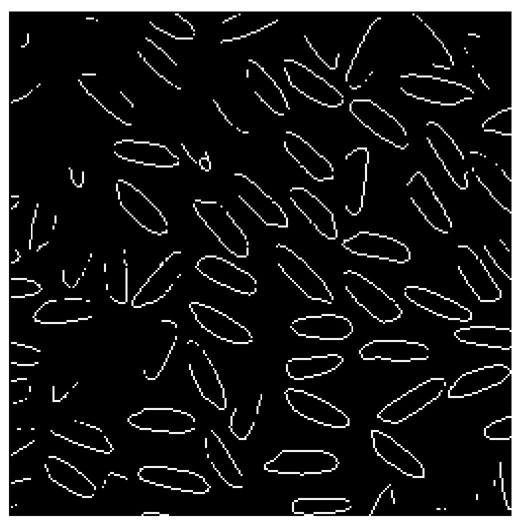


Figure-5.14: Sobel Edge Detected Image

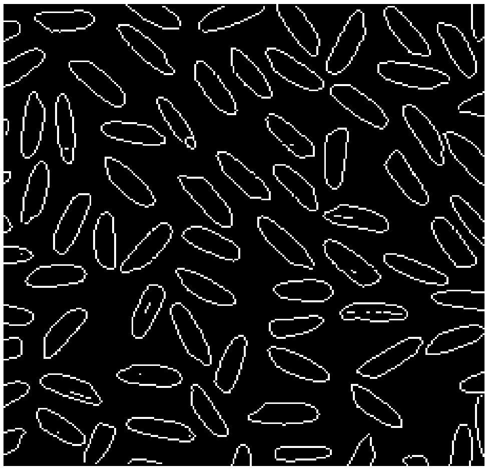


Figure-5.15: Sobel Edge Detected Image, t=0.05

Figure-5.16: Noisy Sobel Edge Region Image

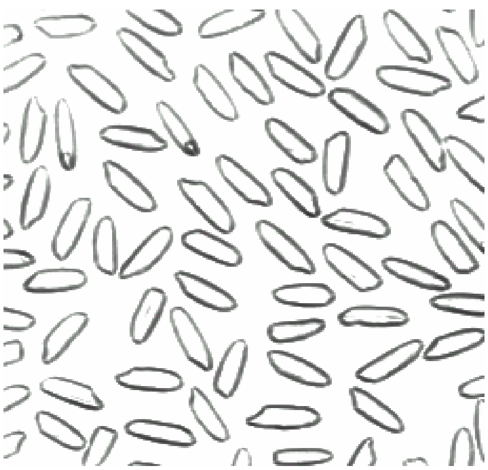


Figure-5.17: Amplified Sobel Edge Region Image

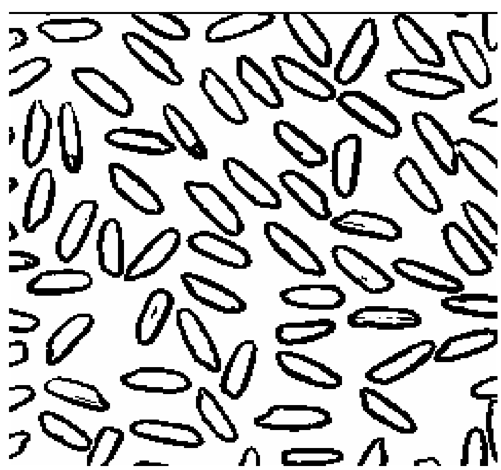


Figure-5.18: Binarised Sobel Edge Region Image

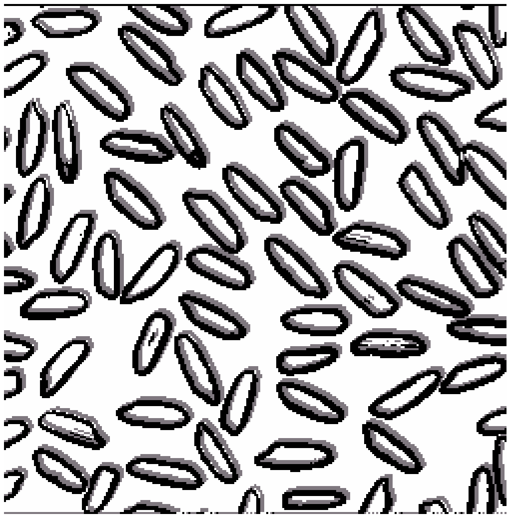


Figure-5.19: Seed Image

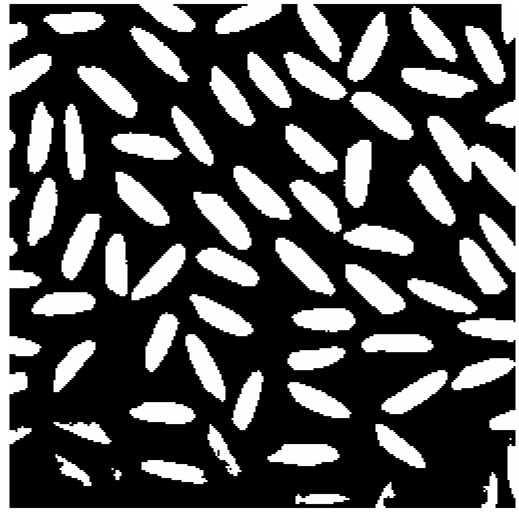
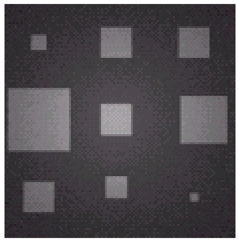


Figure-5.20: Final Segmented Image



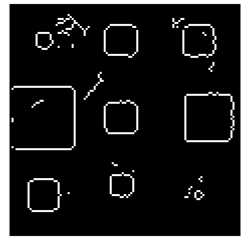
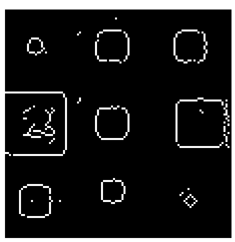


Figure-5.21: Block Image (117 X 115)

Figure-5.22: Canny Edge Detected Image, t_2=0.10, t_1=0.04, sigma=1.5



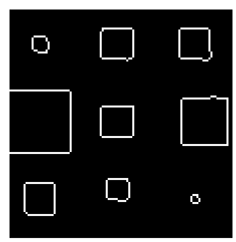
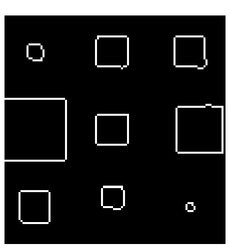


Figure-5.23: Canny Edge Detected Image

Figure-5.24: Sobel Edge Detected Image



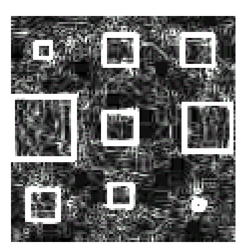


Figure-5.25: Sobel Edge Detected Image, t=0.05

Figure-5.26: Noisy Sobel Edge Region Image

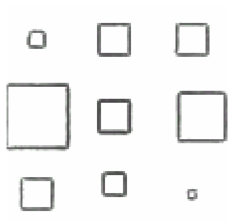


Figure-5.27: Amplified Sobel Edge Region Image

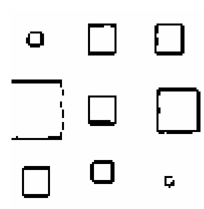


Figure-5.28: Binarised Sobel Edge Region Image

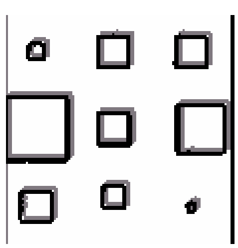


Figure-5.29: Seed Image

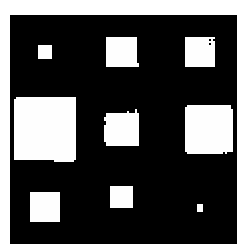


Figure-5.30: Final Segmented Image

Figure-5.31: Barbara Image

Figure-5.32: Canny Edge Detected

(237 X 239) **Image, t_2=0.10, t_1=0.04, sigma=1.5**

Figure-5.33: Canny Edge Detected Image

Figure-5.34: Sobel Edge Detected Image

Figure-5.35: Sobel Edge Detected Image, t=0.15

Figure-5.36: Noisy Sobel Edge Region Image

Figure-5.37: Amplified Sobel Edge Region Image

Figure-5.38: Binarised Sobel Edge Region Image

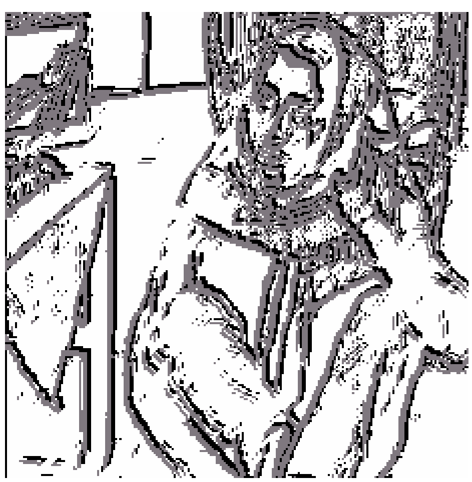


Figure-5.39: Seed Image

Figure-5.40: Final Segmented Image

Figure-5.41: Original Lina Image (256 X 256)

Figure-5.42: Canny Edge Detected Image, t_2=0.10, t_1=0.04, sigma=1.5

Figure-5.43: Canny Edge Detected Image

Figure-5.44: Sobel Edge Detected Image

Figure-5.45: Sobel Edge Detected Image, t=0.15

Figure-5.46: Noisy Sobel Edge Region Image

Figure5.47: Amplified Sobel Edge Region Image

Figure-5.48: Binarised Sobel Edge Region Image

Figure-5.49: Seed Image

Figure-5.50: Final Segmented Image

Figure-5.51: Original Vegetable Image (214 X 849)

Figure-5.52: Canny Edge Detected Image, t_2=0.10, t_1=0.04, sigma=1.5

Figure-5.53: Canny Edge Detected Image **Figure-5.54: Sobel Edge Detected Image**

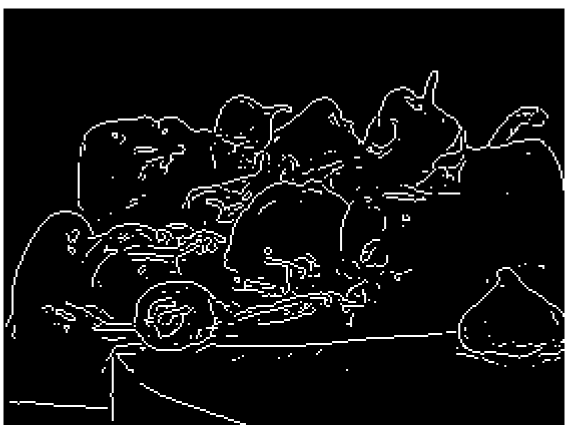


Figure-5.55: Sobel Edge Detected Image, t=0.05 **Figure-5.56: Noisy Sobel Edge Region Image**

Figure-5.57: Amplified Sobel Edge Region Image

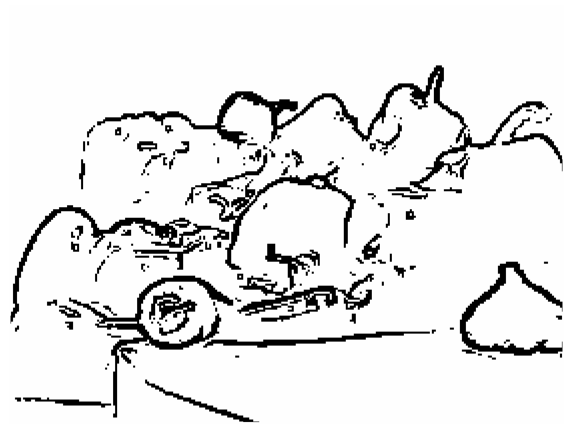


Figure-5.58: Binarised Sobel Edge Region Image

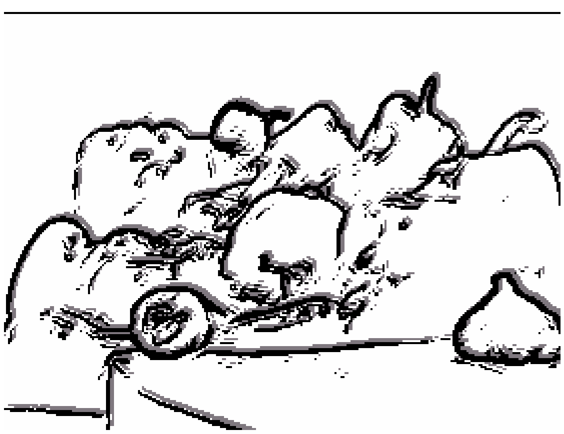


Figure-5.59: Seed Image

Figure-5.60: Final Segmented Image

CHAPTER 6

CONCLUSION

In this thesis, canny edge detector is used which is one of the most powerful edge detector as the edge points determined, give rise to ridges in the gradient magnitude image. The canny algorithm then tracks at the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give thin line as the output, a process known as non maximal suppression. Also due to the use of two thresholds, unlike other edge detectors it is able to detect small intensity variations in an image as edges.

Sobel masks are generally used for detecting the edges in all the four directions, horizontal, vertical, 45^0 and -45^0. They are used to find the approximate absolute gradient magnitude with particular direction in input gray scale image. A new approach used is that horizontal and vertical sobel masks are combined with the 45^0 and -45^0 diagonal masks to obtain the edge region, instead of applying the horizontal, vertical and diagonal masks alone. This method results in thick edge region, which is further used for growing seeds.

In the edge region detected by sobel operator the two types of region growing seeds (pixels) are grown, which are used to obtain the final segmented image. In the homogeneous region no seeds will be produced and seeds can grow in this region. If an area has both types of seeds then a competition will occur. In this case the kind of seed which is closer to that pixel will win and pixel will be merged into that seed set.

A method for stable segmentation is introduced, in which the two most important segmentation techniques i.e. edge detection and region growing are merged together. This method results in precise segmentation. Each and every object in the image to be segmented is defined in such a way that edge and region boundaries of an objects collapse with each other. Hence over segmentation, false impression of objects is eliminated completely.

This method works even for those images where the histogram shows an overlap between the object and the background. Edge based region growing segmentation combines the advantage of edge based segmentation and region growing segmentation. With seeds definition based on the edge image and the edge region image, the detected boundary is accurate. This method not only segments images with bad illumination, but also detects the object boundaries in such images correctly.

This method is very successful on tool images because objects shown occupy areas of many pixels, making it easy to select objects in foreground to separate from background and noise.

REFERENCES:

[1] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques", CVGIP, 29, pp. 100–132, 1985.

[2] J. Hartigan, "Clustering algorithms", New York: Wiley, 1975.

[3] U. Montaneri, "On the optimal detection of curves in noisy pictures," Commun. ACM, Vol. 14, No. 5, pp. 335-345, 1971.

[4] D. H. Ballard, "Hierarchic d etection of tumors in chest radiographs," Rep. ISR-16, 1976.

[5] G. S. Robinson, "Color Edge Detection", Opt. Eng, Vol. 16, No. 5, pp. 479-484, 1977.

[6] K. Nevita, "Color Edge Detection and its use in scene segmentation", IEEE Transaction System Man Cybernetics (SMC), Vol. 7, No.11, pp. 820-826, 1977.

[7] J.S Weszka, "A Survey of Threshold Selection Techniques", Computer Graphics Image Processing, Vol. 7, pp. 259-265, 1978.

[8] R. Ohlander, K. Price, D.R. Reddy, "Picture segmentation using recursive region Splitting method", Computer Image Processing, Vol. 8, pp. 313-333, 1978.

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

[10] S. Grinaker, "Edge Based Segmentation and Texture Separation," in Proc. 5th International Conference Pattern Recognition, Miami Beach, FL, pp. 554-557, 1980.

[11] D. Marr, E. Hildreth, "Theory of Edge Detection", Proc. Royal Soc. London, Vol. 207, pp. 187-217, 1980.

[12] M. Bajcsy, M. Mintz, and E. Liebman, "A Common Framework for Edge Detection and Region Growing," Univ. Pennsylvania GRASP Lab, Tech. Rep. 61, 1986.

[13] J. Canny, "A Computational Approach to Edge Detection", IEEE Transactions Pattern Analysis Intelligence, Vol. 8, pp. 679-714, 1986.

[14] P. Fua and T. Hanson, "Using Generic Geometric Models for Intelligent Shape Extraction," in Proc. DARPA Image Understanding Workshop, pp. 227-233, 1987.

[15] H. L. Anderson, R. Bajcsy, and M. Mintz, "A Modular Feedback System for Image Segmentation," Univ. Pennsylvania GRASP Lab, Tech. Rep. 110, 1987.

[16] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," in Proc. 1st International Conference & Computer Vision, pp. 259-267, 1987.

[17] R. Adams and L. Bischof, "Seeded Region Growing", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, pp. 641-647, 1994.

[18] E. Breen and D. Monro, "An Evaluation of Priority Queues for Mathematical Morphology", Mathematical Morphology and its Applications to Image Processing, pp. 249-256, 1994.

[19] SA. Hojjatoleslami, J. Kittler, "Region Growing: A new approach", IEEE Transactions of Image Processing, Vol. 7, No. 7, pp.1079-1084,2000.

[20] Gonzalez, R. C. and Wood, R.E, "Digital Image Processing", Welsey publishing

company, 2005.

[21] Nick Kanopoulos, Nagesh Vasantvadha, Robert. L. Baker, "Design of an Image Edge Detection Filter using the Sobel Operator", IEEE Journal of Solid State Circuits, Vol. 23, No. 2, 1988.

[22] V. Torre and T. Poggio, " On Edge Detection", IEEE Transactions Pattern Analysis Machine Intelligence, Vol. 8, No. 4, pp. 147-163, 1986.

[23] R. C.Gonzales, R. E. Woods, "Digital Image Processing", ISBN: 0-201-18075-8, 2nd Edition, Prentice Hall, 2002.

[24] R. M. Haralick, L. G. Shapiro, Computer and Robot Vision, ISBN: 0-201-10877-1, Addison Wesley Publishing Company, 1992.

[25] X. Yu, J. Yla-Jaaski, "A New Algorithm for Image Segmentation based on Region Growing and Edge Detection" Proc. International conference on Circuits and System, Vol.1, pp. 5 16-519, 1991.

[26] J. H. Xuan, T. Adali, and Y Wang, "Segmentation of Magnetic Resonance Brain Image: Integrating Region Growing and Edge Detection," Proc. International Conference on Image Processing, Vol. 3, pp. 544-547, 1995.

[27] T. Lindeberg, J. O. Eklundh, "Scale Detection and Region Extraction from a Scale-Space Primal Sketch", Proc. 3rd International Conference on Computer Vision, pp. 416-426, 1990.

[28] M. Sonka, V. Hlavac and R. Boyle, Image Processing, Analysis and Machine Vision, ISBN: 0-412-45570-6, 2nd edition, PWS Publishing, 1999.

[29] http://www.cse.unr.edu/~mircea/Courses/cs491E-791E/Lectures/1389,2,Edge Detection

[30] S.H Kim, D.O Kim, J.S. Kang ET. Al, "Detection of Edges", Proc. Spie Capturer Analysis, and Display of Image Sequences III", Vol. 4308, San Jose, A, 2001.

[31] R.D. Stewart, I. Fermin and M.Opper, "Region growing with pulse coupled neural Networks: An alternative to seeded region growing", IEEE Transactions on "Neural Networks", Vol. 13, pp. 1557-1562, 2002.

[32] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques", CVGIP, 29, pp. 100–132, 1985.

[33] Hao Xiaohui, C. Bruce, C. Pislaru, J.F. Greenleaf, "A novel region growing method for segmenting ultrasound images", IEEE Symposium Ultrasonics, Vol. 2, pp. 1717 –

1720, 2000.

[34] P.J. Yim, P.L. Choyke and R.M. Summers, "Gray scale Skeletonization of Small Vessels in magnetic resonance angiography", IEEE Transactions in Medical Imaging, Vol. 19, No.6, pp. 568 – 576, 2000.

[35] R. Beare, "Regularized Seeded Region Growing", CSIRO Mathematical and Information Sciences, pp. 91-99, 2002.

[36] Shu Yen Wan and William. E. Higgins, "Symmetric Region Growing", in IEEE International Conference on Image Processing, pp. 1196-1199, 2000.

[37] Shu-Yen Wan, Atilla P. Kiraly, Erik L. Ritman, and William E. Higgins, "Extraction of the hepatic vasculature in rats using 3D micro-CT images," IEEE Transactions on Medical Imaging, Vol. 19, No. 9, pp. 964-971, 2000.

[38] Yu-Bu Lee, Soo Min Song, Jae Sung Lee, Myoung Hee Kim, "Tumor segmentation from small animal PET using region growing based on gradient magnitude", Proceedings of 7th International Workshop on Image processing, pp. 243-247, 2005.

[39] Runzhen Huang, Kwan Liu Ma, "RGVis: Region Growing Based Techniques for Volume Visulization", 11th Pacific Conference on Computer Graphics and Applications, pp. 355-363, 2003.

List of Publications

1. Sanmeet Kaur, Mr. Kulbir Singh, "Edge Detection by Canny Algorithm", in proceedings of National Conference on Electronic circuits & Communication Systems (ECCS-06) held on 9th February, 2006 at Thapar Institute of Engineering and Technology, Patiala (Punjab).

2. Sanmeet Kaur, Mr. Kulbir Singh, "Analysis of Region Growing Techniques", communicated to National Conference on Wireless Networks and Embedded Systems

(WNES-2006) to be held on 28^{th} July, 2006 at Chitkara Institute of Engineering and Technology, Rajpura, Patiala (Punjab).