Edge Detection in Medical Images Using Morphological Operators

M.Sc. Thesis
in
Electrical and Electronics Engineering
University of Gaziantep

Supervisor
Asst. Prof. Dr. Nurdal WATSUJI

by
Taner İNCE
June 2006

Approval of the Graduate School of Natural and Applied Sciences

(Prof. Dr. M. Sadettin Özyazıcı)
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

(Assoc. Prof. Dr. Gülay Tohumoğlu)
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

(Asst. Prof. Dr. Nurdal Watsuji)
Supervisor

Examining Committee Members	signature
1. Prof. Dr. L. Canan DÜLGER	
2. Prof. Dr. Arif NACAROĞLU	
3. Prof. Dr. Rauf MİRZABABAYEV	
4. Assoc. Prof. Dr. Gülay TOHUMOĞLU	
5. Assist. Prof. Dr. Nurdal WATSUJI	

To my family, who have created and maintained a wonderful life for me and contributed to my life with their lovely supports and encouragements.

ABSTRACT

EDGE DETECTION IN MEDICAL IMAGES USING MORPHOLOGICAL OPERATORS

İNCE, Taner
M.Sc. in Electrical and Electronics Eng.
Supervisor: Asst. Prof Dr. Nurdal WATSUJI
June 2006, 92 pages

Edge detection is an important research area in digital image processing with several applications. Edges characterize boundaries and therefore a problem of fundamental importance in image processing. The aim of edge detection in an image is to reduce the amount of data and filter out useless information, while preserving the important structural properties in an image.

Mathematical morphology provides a systematic approach to analyze the geometric characteristics of signals or images, and has been used widely in many applications such as boundary detection, noise removal, image enhancement and image segmentation etc. The advantages of morphological approaches over linear approaches are 1) direct geometric interpretation, 2) simplicity, and 3) efficiency in hardware implementation.

The main purpose of this thesis is to provide an overview of mathematical morphology and review some edge detection algorithms based on mathematical morphology and also propose a method for detecting edges in medical images such as Computed Tomography and Magnetic Resonance images. Edge detection can be divided into two phases; the first is the noise removal, and the second is ideal edge detection. By using an iterative averaged closing-opening operation, impulse noise as well as Gaussian noise is eliminated from the image. Then, the resulting ideal edges can be extracted by using a simple morphologic operator.

Keywords: Edge detection, Mathematical Morphology, Alternating Sequential Filters,Thinning

ÖZET

MEDİKAL GÖRÜNTÜLERDE MATEMATİKSEL MORFOLOJİ KULLANARAK AYRIT SEZİMLEME

İNCE, Taner
Yüksek Lisans Tezi, Elektrik ve Elektronik Müh.
Tez Yöneticisi: Yrd. Doç. Dr. Nurdal WATSUJI
Haziran 2006, 92 sayfa

Ayrıt sezimi sayısal görüntü işleme alanında bir çok uygulaması olan önemli bir araştırma alanıdır. Ayrıtlar sınırları karakterize eder ve dolayısıyla görüntü işleme alanında önemli bir problemdir. Bir görüntüdeki ayrıt seziminin amacı, o görüntüdeki veri miktarını azaltmak ve gereksiz bilgiyi süzmektir, bununla birlikte o görüntüdeki önemli yapısal özellikleri korumaktır.

Matematiksel Morfoloji sinyallerin ve görüntülerin geometrik karakterlerini analiz etmek için sistematik bir yaklaşım sunar, ve bir çok alanda kullanılmaktadır örnek olarak sınır tespitleme, gürültü yoketme, görüntü iyileştirme ve görüntü bölütleme sayılabilir. Morfolojik yaklaşımın linear yaklaşımlar üzerine avantajları 1) doğrudan geometrik açıklama, 2) basitlik, ve 3) donanım uygulamalarının verimliliğidir

Bu tezin esas amacı Matematiksel Morfolojiyi tanıtma, bazı Matematiksel Morfoloji tabanlı ayrıt sezimleme algoritmalarını gözden geçirme ve Computed Tomografi ve Manyetik Rezonans gibi medikal görüntüler için bir ayrıt sezimleme yöntemi sunmaktır. Ayrıt sezimleme iki kısıma ayrılabilir; birincisi gürültü yoketme ve ikincisi ideal ayrıt sezimidir. Döngülü açma ve kapama işlemi kullanarak dürtü gürültü ve o kadar da Gauss gürültüsü görüntüden yok edilir. Sonra, basit bir morfolojik operator kullanarak sonuçlanan ideal ayrıtlar ortaya çıkarılabilir.

Anahtar Kelimeler: Ayrıt Sezimi, Matematiksel Morfoloji, Ardışık Değişen Filtreler, İnceltme

ACKNOWLEDGMENTS

It is my pleasure to express my gratitude to several people without whom this thesis wouldn't be possible.

First of all I am particularly very grateful to my supervisor Asst. Prof. Dr. Nurdal WATSUJI. She has helped me significantly throughout all stage of my study with his many valuable advices and encouragments.

My special thanks to our department head, Assoc. Prof. Dr. Gülay TOHUMOĞLU, who gave me the courage and inspiration to get my M.Sc. degree.

I would especially like to thank my colleague Cumali SABAH. I have been enormously profited in preparing of my thesis from his valuable comments, criticisms, and suggestions.

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

page

NOMENCLATURE

F	Image Matrix
(x,y)	Coordinates of discrete matrix F
f(x,y)	Value of matrix F at point (x,y)
A	Binary set
B	Structuring element
$a \in A$	a is a set member of A
$a \notin A$	a is not a set member of A
$(A)_z$	A is translated by an amount of z
$\hat{B}$	Reflection of set B
$\oplus$	Dilation
Θ	Erosion
A^c	Complement of A
$\bigcup$	Union
$\bigcap$	Intersection
$\subset$	Subset
$\varnothing$	Empty set
$\neq$	Not equal
$\circ, \gamma, \alpha$	Opening
$\bullet, \phi, \beta$	Closing
max	Maximum operator
min	Minimum operator
HoMT	Hit or miss transformation
$\otimes$	Thinning operation
Ψ	Filter
Δf	Gradient of f
mag	Magnitude
$\lvert\ \rvert$	Magnitude
a	angle
lim	Limit
$\Delta^2 f$	Laplacian
δ	Distance
D	Flat structuring element
N_4	Four connected neighborhood
N_8	Eight connected neighborhood
G	Morphological gradient
BMO	Blur Minimum operator
$*$	Convolution
ATM	Alpha trimmed mean
T	Threshold value
σ	Variance

p	Probability density function
FOM	Figure of merit

CHAPTER 1

INTRODUCTION

1.1 Image Definition

An image is a spatial representation of a two-dimensional or three-dimensional scene. In image processing, an image usually is digitized from a recorded image, such as a video image, camera or a picture. The digitization process includes sampling and quantization of continuous data. The sampling process samples the intensity of the continuous-tone image, such as monochrome, color or multi-spectrum image, at specific locations on a discrete grid. The grid defines the sampling resolution. The quantization process converts the continuous or analog values of intensity brightness into discrete data, which corresponds to the digital brightness value of each sample, ranging from black, through the grays, to white. A digitized sample is referred to as a picture element or pixel. An image is generally is sampled into a rectangular array of pixels, which, for a monochrome image is represented as a discrete matrix F:

$$F= \{f(x,y) \mid x = 0,\ldots..N-1,\ y = 0,\ldots\ldots M-1 \} \qquad (1.1)$$

where N is the total number of columns and M is the total number of rows. The value of the image at spatial coordinates corresponding to the sample index (x,y) is denoted by f(x,y), where x is the column number and y is the row number. The sample interval Δx and Δy are selected to match the smallest feature size. The physical coordinates are equal to $x \bullet \Delta x$ and $y \bullet \Delta y$. An example of an image matrix is shown in Figure 1.1.

Upper-left corner of image

x →

y	0	1	2	3	.....		N-1
0	f(0,0)	f(1,0)	f(2,0)		.....		f(N-1,0)
1	f(0,1)	f(1,1)	f(2,1)		.....		f(N-1,1)
2	f(0,2)	f(1,2)	f(2,2)		.....		
3							
4							
:							
:							
M-1	f(0,M-1)	f(1,M-1)	.....	.....			f(N-1,M-1)

Figure 1.1 Image Matrices

For optical or photographic sensors, f(x,y) is typically proportional to the radiant energy received in the electromagnetic band to which the sensor or detector is sensitive, and integrated over a small aperture around (x,y). This is interpreted as an intensity at point (x,y). A simple aperture can be defined as a small rectangular or circular function in data acquisition processing, and the continuous intensity sampled is the integral of the image over the aperture at position (x,y). Because of the integral effect, for example, the sharp edge in a real image will become a blurred edge in digital image. Normally digitized images are defined by levels of gray spanning from black to white, called gray-scale images. Figure 1.2 shows the process of digital conversion.

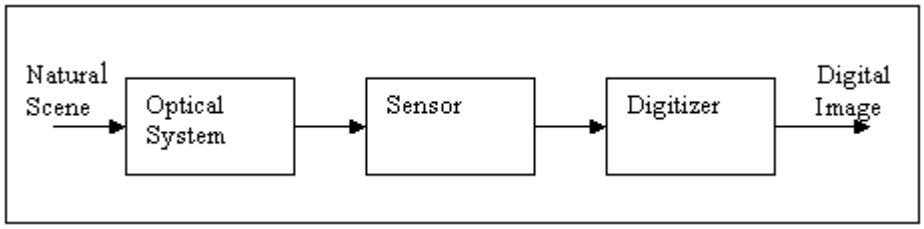


Figure 1.2 Model of a Digital Image Processing System

Actually, the image may come from different sources. For medical imaging applications the images can be digitized from x-rays, ultrasound waves, and Magnetic Resonance response. For astronomy and military applications, satellite sensors produce digital images directly from the measurements of reflected or emitted visible, infrared or microwave radiation. In auto vehicle navigation applications, a vision system may use a combination of passive and active spatial image data.

During the image digitization, the image may be degraded by several factors. One is the restriction of sampling frequency. Based on the sampling theorem [R], the image has to be sampled at a rate of at least twice the highest spatial frequency contained in original image. Otherwise the image will have spatial aliasing. In natural scenes there is no way to control the "highest spatial frequency", and this theorem is almost never satisfied. The second factor is the blurring effect of aperture. The third one is the quantization error. Because the quantization uses a finite number of values to represent the infinite value range of original images, the intensity of an image is not exact. The fourth one is additive and multiplicative noise produced by quantization devices, such as thermal effects in electronic components, which is often modeled as a Gaussian noise.

1.2 Edge Model Definition

An edge can be defined as the boundary between two regions separated by two relatively distinct gray-level properties [8]. It corresponds to local intensity discontinuities of an image. In the real world, the discontinuities reflect a rapid intensity change, such as the boundary between different regions, shadow boundaries, and abrupt changes in surface orientation and material properties. For example, edges represent the outline of a shape, the difference between the colors and pattern or texture. Therefore edges can be used for boundary estimation and segmentation in scene understanding. They can also be used to find corresponding points in multiple images of the same scene. For instance, the fingerprint, the human facial appearance and the body shape of an object are defined by the edges in images.

In a broad sense the term edge detection refers to the detection and localization of intensity discontinuities of these image properties. In a more restrictive sense, it only refers to locations of significant change of intensity. Points of these locations are called edges or edge elements. Figure 1.3 shows images in which edges define object size, shape, scene and human appearance.

a) Aerial Image b) Lena c) MRI image

Figure 1.3 Edges in image show object size, shape boundary, human facial appearance

The difference between boundaries and edges is that boundaries are the linked edges that characterize the shape of an object.

1.3 Need for Edge Detection

In general, edge detection is the process of locating the edge points. It can be used for region segmentation, feature extraction, and object or boundary description. Edges provide the topology and structure information of objects in an image. For example, different cars can be easily recognized from their body shape. The highway and river from aerial images can be detected in terms of their structure or distribution pattern, which all are described by the edges. By using edge detection techniques, machine vision and image processing systems can be built for a variety of applications. For example edge detection can be used for plate recognition of vehicles to identify them. Edge detection can also be used in medical applications such as detection of cancerous cells in mammograms, tumors in MR images e.g.

The human perceptual system emphasizes the importance of edges to human vision. It has been suggested that one of the most fundamental organizing principles of visual system is the detection and description of discontinuity. Physiologists have discovered the organization of the mammalian visual system in terms of the responses to spatially and temporally discontinues inputs, such as edges, bars and points of light turning on or off.

Edge detection, therefore, is consistent with human visual response, which will provide edge strength and orientation for feature extraction and object description. For reducing information being processed, after edge detection gray-scale edges are usually converted into binary images by thresholding. The transformation preserves a great deal of the primitive or intrinsic information from the original image, i.e., the outline of the shape.

In this thesis, an edge detection method based on mathematical morphology is investigated and they are compared with other edge detectors. Mathematical morphology is a branch of nonlinear signal processing and is a powerful tool for the geometrical shape description and analysis of images.

1.4 Difficulty with the Process of Edge Detection

Edge detection is a difficult issue. One difficulty comes from the complex contents of image itself. In real world applications, images contain object boundaries and object shadows and noise. The second cause of problems is degradation in image acquisition discussed in section 1.1. Sometimes it may be difficult to distinguish the exact edge from noise or trivial geometric features. For example, an image of the character “C” may be contaminated by noise during a fax transmission and it may look like a “G”. The cancerous cell in an MR image may be lost during acquisition and it may look like a normal cell.

Two level edge detection processes are often used since the difficulty of edge estimation can not be easily overcome from detection operators alone. The first level process, called low-level process, extracts pieces of raw edge segments and geometric features, called primitives. They may be incomplete and inaccurate. The second level process usually is called high-level process. It will interpret and combine raw edges based on the edge models or deduction rules from a broader image context and a knowledge database. Sometimes pattern matching and statistical analysis will occur at this level. The second level process tries to remove the uncertainty or make correct decisions using low-level inputs and context. The more accurate the low-level input is, the more accurate the high-level process result will be achieved. To measure the quality of low-level process, several criteria are proposed to help to improve the accuracy of edge detection.

1.5 Criteria for Edge Detection

The quality of edge detection can be measured from several criteria objectively. Some criteria are proposed in terms of mathematical measurement [1], some of them are based on application and implementation requirements. In all cases a quantitative evaluation of performance requires use of images where the true edges are known.

1. True Edge Detection

There should be minimum number of false edges or maximum Signal Noise Ratio (SNR). Usually, edges are detected after a threshold operation. The high threshold will lead to less false edges, but it also reduces the number of true edges detected.

2. Robustness to Noise

The robust algorithm can detect edges in certain acceptable noise (Gaussian, Uniform and impulsive noise) environments. Actually, an edge detector detects the edges and also amplifies noise simultaneously. Strategic filtering, consistency checking and post processing (such as non-maximum suppression) can be used to reduce noise sensitivity [2].

3. Edge Localization

The edge location must be reported as close as possible to the correct position, i.e. edge localization accuracy [2]

4. Orientation Sensitivity

The operator not only detects edge magnitude, but it also detects edge orientation correctly. Orientation can be used in post processing to connect edge segments, reject noise and suppress non-maximum edge magnitude.

5. Speed and efficiency

The algorithm should be fast enough to be usable in an image processing system. An algorithm that allows recursive implementation or separable processing can greatly improve efficiency.

CHAPTER 2

INTRODUCTION TO MATHEMATICAL MORPHOLOGY

The word morphology commonly denotes a branch of biology that deals with the form and structure of animals and plants. Also it can be interpreted as shape study using mathematical set theory. In image processing, mathematical morphology refers to a branch of nonlinear image processing and analysis developed initially by George Matheron and Jean Serra [3] that concentrate on the geometric structure within an image. The original theory developed by Matheron and Serra was limited to binary images, and later it was extended to gray scale morphology by Sternberg [4], Nakagawa and Rosenfeld [5] etc.

The language of mathematical morphology is the set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. These include enhancement, segmentation, restoration, edge detection [6], texture analysis, particle analysis and compression.

Some of the salient points regarding the morphological approach are as follows [7]

1. Morphological operations provide for the systematic alteration of the geometric content of an image while preserving the stability of important geometric characteristics.

2. There exists a well-developed morphological algebra that can be employed for representation and optimization.

3. It is possible to express digital algorithms in terms of a very small class of primitive morphological operations.

4. There exist certain representations theorems by means of which one can obtain the expression of morphological filters in terms of the primitive morphological operations.

In general, morphological operators transform the original image into another image through the interaction with the other image of a certain shape and size, which is known as the structuring element. Geometric features of the images that are similar in shape and size to the structuring element are preserved, while other features are suppressed. Therefore morphological operations can simplify the image data, preserving their shape characteristics and eliminate irrelevancies. In view of applications, morphological operations can be employed for many purposes, including edge detection, segmentation, and enhancement of images.

In Euclidean space, mathematical morphology is called Euclidean morphology and in digital setting it is called digital morphology. The actual implementation of morphological operators will be in digital setting, therefore in this thesis digital morphology setting will be considered.

There are different notations in the mathematical morphology literature. For consistent description, notations of the Gonzalez [8] will be used in this thesis. An image object F in mathematical morphology terminology is represented as a set inside an n-dimensional Euclidean space R^n. A binary image 2D image object A is denoted by a set in R^2 as:

$$A = \{a : f(a) = 1, a = (a_x, a_y) \in R^2\} \tag{2.1}$$

For convenience in the following sections, bold character **z**=(x,y) will be used to refer a point at the coordinate (x,y) of a set.

2.1 Binary Morphology

The theoretical foundation of binary mathematical morphology is the set theory [3]. In binary images, those points in the set are called the 'foreground' and those in the complement set are called the 'background'.

Besides dealing with the usual set operations of union and intersection, morphological operations are mainly dependent on translation operation. For convenience, '$\cup$' denotes the set-union, '$\cap$' denotes the set-intersection and '+' inside the set notation denotes the vector addition in the following equations. To introduce the basic concepts of

morphological operators, several basic definitions are discussed first. Let A and B be sets in R^2 where a=(a_x,a_y) and b=(b_x,b_y). The translation, reflection and complement are defined as follows:

a) Translation

The translation of set A by point $z = (z_1, z_2)$, denoted $(A)_z$, is defined as:

$$(A)_z = \{c \mid c = a + z, \text{ for } a \in A\} \tag{2.2}$$

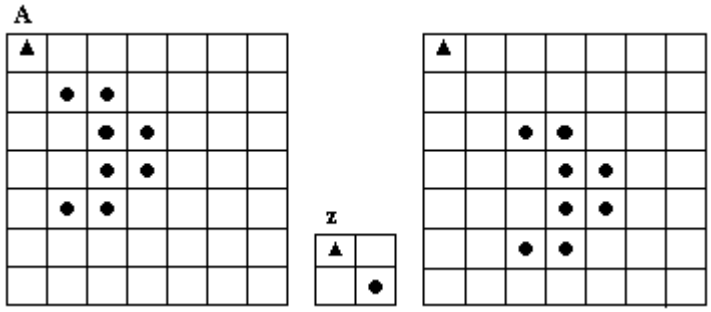


Figure 2.1 Illustration of translation operation

Figure 2.1 illustrates the definition of the translation operation; the symbol ▲ denotes the origin of each image set. The set **A** is an image in Z^2. The result is displayed in the right part of Figure 2.1.

b) Reflection

The reflection of set B, denoted $\hat{B}$, is defined as:

$$\hat{B} = \{w \mid w = -b, \text{ for } a \in A\} \tag{2.3}$$

c) Complement

The complement of a set A is the set of elements not contained in A:

$$A^c = \{w \mid w \notin A\} \tag{2.4}$$

2.1.1 Structuring Element

Before explaining the basic operations of mathematical morphology, definition of the structuring element must be given. A structuring element is a small image that is overlapped on input image to compute a certain definition. The basic operations of binary and also grayscale images depend on what structuring elements are used. In this section only binary morphology are considered. In the next sections grayscale morphology will be introduced.

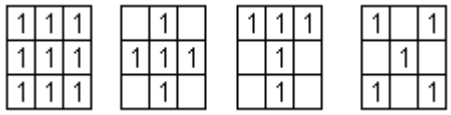


Figure 2.2 Examples of 3x3 structuring elements

Figure 2.2 shows structuring elements that are used commonly for binary images. The origins of the structuring elements are located on the center. The pixels marked '1' are the points that should be considered during any binary morphological operations. Set of coordinate points of the first structuring element shown in Figure 2.2 are given as {(-1,-1), (0,-1), (1, -1), (-1, 0), (0, 0), (1, 0), (-1, 1), (0, 1), (1, 1).

2.2 Basic Operations of Mathematical Morphology

There are two basic operations in mathematical morphology called namely; dilation and erosion. The other morphologic operators are derived from these basic operations.

2.2.1 Binary Dilation

Dilation is an operation that "grows" or "thickens" objects in a binary image. The specific manner and extent of this thickening is controlled by the structuring element.

Let A and B are sets in Z^2, the dilation of A by B denoted by $A \oplus B$, is defined as

$$A \oplus B = \left\{ z \in Z^2 \mid z = a + b, \text{for some } a \in A \text{ and } b \in B \right\} \tag{2.5}$$

where A usually is an image and B is the structuring element. It can be shown that dilation is equivalent to a union of translation of the original image with respect to the structuring element

$$A \oplus B = \bigcup_{b \in B} (A)_b \tag{2.6}$$

In other words, the dilation of *A* by *B* is the set consisting of all the structuring element origin locations where the reflected and translated *B* overlaps at least some portion of *A*.

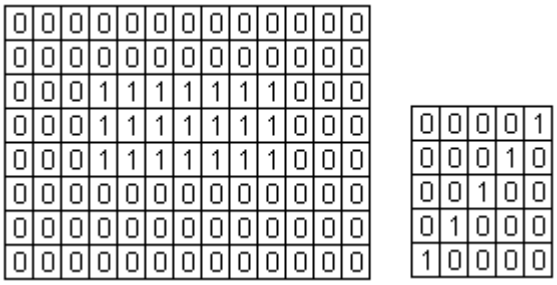


Figure 2.3 Example binary image and a structuring element

Figure 2.3 shows a simple binary image containing a rectangular object and a structuring element. The bold rectangle on the centre of the structuring element denotes the center of it. Application of dilation operation to the binary image with the structuring element yields the result given in Figure 2.4.

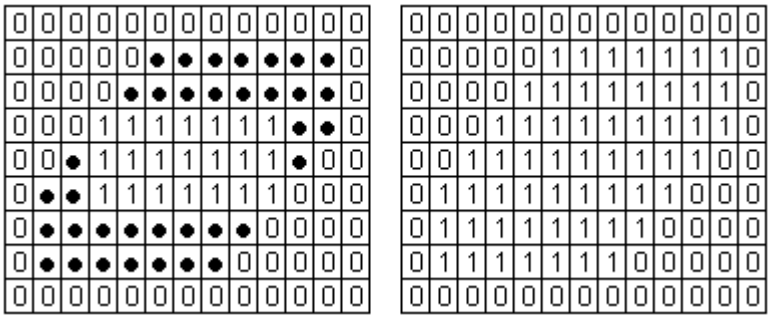


Figure 2.4 Dilated image

Dilation is found by placing the centre of template over each of the foreground pixels (1's) of the original image and then taking the union of all the resulting copies of the structuring element, produced by using the translation. In Figure 2.4 black dots shows the expanded points after dilation operation and the image on the right of the Figure 2.4 is the final resultant binary image. Dilation generally has an effect of expanding an image; so consequently, small holes inside foreground can be filled.

In another sense, dilation can be a morphological operation on a binary image defined as:

$$A \oplus B = \left\{ z \mid (\hat{B})_z \cap A \neq \varnothing \right\} \tag{2.7}$$

this equation is based on obtaining the reflection of B about its origin and shifting this reflection by z. The dilation of A by B then is the set of all displacements, z, such that $\hat{B}$ and A overlap by at least one element. Based on this interpretation the equation above may be written as

$$A \oplus B = \left\{ z \mid \left[(\hat{B})_z \cap A \right] \subseteq A \right\} \tag{2.8}$$

In this equation the structuring element can be thought as a convolution mask. Although dilation is based on set operations, whereas convolution is based on arithmetic operations, the basic process of "flipping" B about its origin and then successively displacing it so that it slides over set (image) A is analogous to the convolution process. Figure 2.5 illustrates the dilation operation using a binary image. The original image is dilated with a 6x6 'disk' type structuring element.

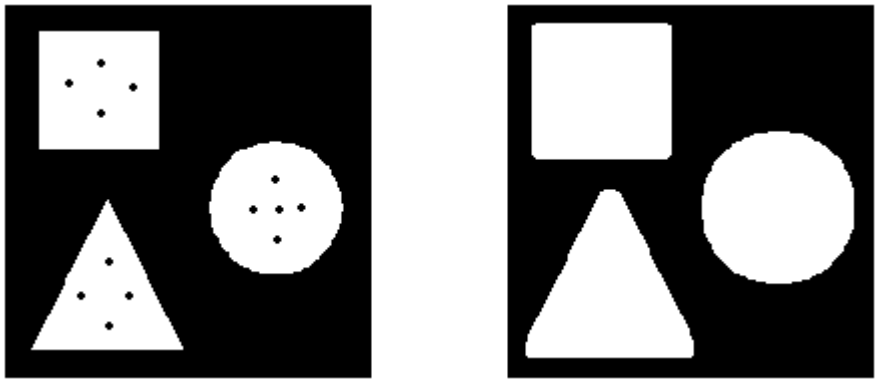


Figure 2.5 Binary dilation example

2.2.2 Properties of Binary Dilation

a) Dilation and erosion are duals of each other with respect to set complementation and reflection.

$$(A \Theta B)^c = A^c \oplus \hat{B} \tag{2.9}$$

Proof:

If set $(B)_z$ is contained in set A, then $(B)_z \cap A^c = \varnothing$, then the preceding equation becomes

$$(A \Theta B)^c = \left\{ z \,|\, (B)_z \cap A^c = \varnothing \right\}^c \tag{2.10}$$

But the complement of the set of z's that satisfy $(B)_z \cap A^c = \varnothing$ is the set of z's such that $(B)_z \cap A^c \neq \varnothing$. Thus

$$\begin{aligned} (A \Theta B)^c &= \left\{ z \,|\, (B)_z \cap A^c \neq \varnothing \right\} \\ &= A^c \oplus \hat{B} \end{aligned} \tag{2.11}$$

where the last step follows from the Eq (2.8). This concludes the proof.

b) Dilation is commutative

$$A \oplus B = B \oplus A \tag{2.12}$$

c) Dilation is associative

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C \tag{2.13}$$

d) Dilation is extensive

$$f \quad 0 \in B, A \subseteq A \oplus B \tag{2.14}$$

e) Dilation is increasing

$$A \subseteq B \ \textit{implies} \ \ A \oplus D \subseteq B \oplus D \tag{2.15}$$

2.2.3 Binary Erosion

Erosion is an operation that "shrinks" or "thins" objects in a binary image. As in dilation, the manner and extent of shrinking is controlled by a structuring element.

Let A and B are sets in Z^2, the erosion of A by B denoted by $A \Theta B$, is defined as

$$A \Theta B = \left\{ z \mid z + b \in A, \ \forall b \in B \right\} \tag{2.16}$$

Whereas dilation can be expressed as a union of translates, erosion can be represented as an intersection of the negative translates. So the definition of erosion can be refined as

$$A \Theta B = \bigcap_{b \in B} (A)_{-b} \tag{2.17}$$

where'–b' is the scalar multiple of the vector 'b' by -1.

Figure 2.6 shows the erosion operation on the binary image and structuring element given in Figure 2.3. Black dots are the shrinked points on the original binary image.

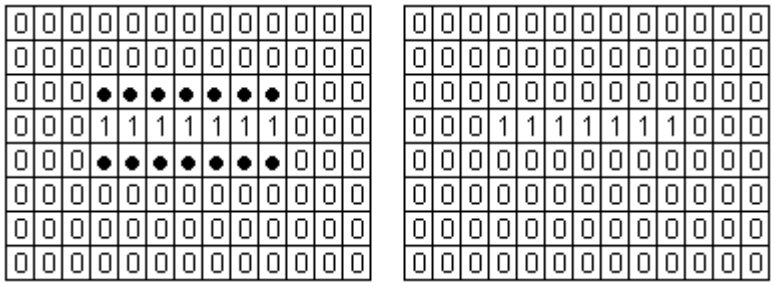


Figure 2.6 Eroded image

Figure 2.6 depicts erosion as a process of translating the structuring element throughout the domain of the image and checking to see where it fits entirely within the foreground of the image. The output image in the right part of the Figure 2.6 has a value of 1 at each location of the origin of the structuring element, such that the element overlaps only 1-valued pixels of the input image.

The mathematical definition of erosion is similar to that of dilation. The erosion of A by B, denoted $A\Theta B$, is defined as

$$A\Theta B = \left\{ z \mid (B)_z \cap A^c \neq \varnothing \right\} \tag{2.18}$$

In other words, erosion of A by B is the set of all structuring element origin locations where the translated B has no overlap with the background of A.

Figure 2.7 Binary erosion example

Figure 2.7 is an example of the erosion of a binary image. The image on the left is eroded with a 10x10 disk typed structuring element. As it is seen on the right of the Figure erosion shrinks the original image and eliminates the small enough peaks.

2.2.4 Properties of Binary Erosion

a) Erosion is not commutative as opposed to dilation

$$A\Theta B \neq B\Theta A \tag{2.19}$$

b) Extensivity

$$f \quad 0 \in B, A\Theta B \subseteq A \tag{2.20}$$

c) Erosion is increasing

$$A \subseteq C \text{ implies } A\Theta B \subseteq C\Theta B,\ B \supseteq C \text{ implies } A\Theta B \subseteq A\Theta C$$

d) Chain rule

$$A\Theta(B_1 \oplus ... \oplus B_k) = (...(A\Theta B_1)\Theta...\Theta B_k) \tag{2.21}$$

2.2.5 Binary Opening

The opening of a binary image *A* by the structuring element *B*, denoted by $A \circ B$ is defined as

$$A \circ B = \left(A\Theta B\right) \oplus B \tag{2.22}$$

Thus, the opening *A* by *B* is the erosion of *A* by *B*, followed by a dilation of the result by *B*.

The geometric interpretation of the opening process is shown in Figure 2.8. Structuring element *B* is viewed as a "rolling ball." The boundary of $A \circ B$ is then established by the points in *B* that reach the farthest into the boundary of *A* as *B* is rolled around the inside of this boundary. This geometric fitting property of the opening operation leads to a set-

theoretic formulation, which states that the opening of A by B is obtained by taking the union of all Translates of B that fit into A. That is, opening can be expressed as a fitting process such that B

$$A \circ B = \bigcup \{ (B)_z \mid (B)_z \subseteq A \} \tag{2.23}$$

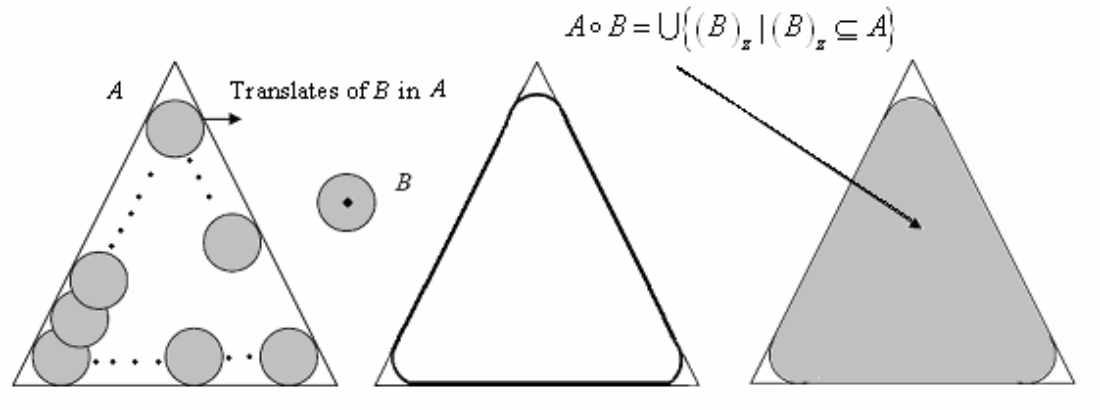


Figure 2.8 Illustration of binary opening process

The effects of opening process on the original image are smoothing, reducing noise from quantization or the sensor and pruning extraneous structures. These effects result from the fact that the structuring element can not fit into the regions. Therefore, it can be said that the result of the opening process heavily depends on the shape of structuring elements.

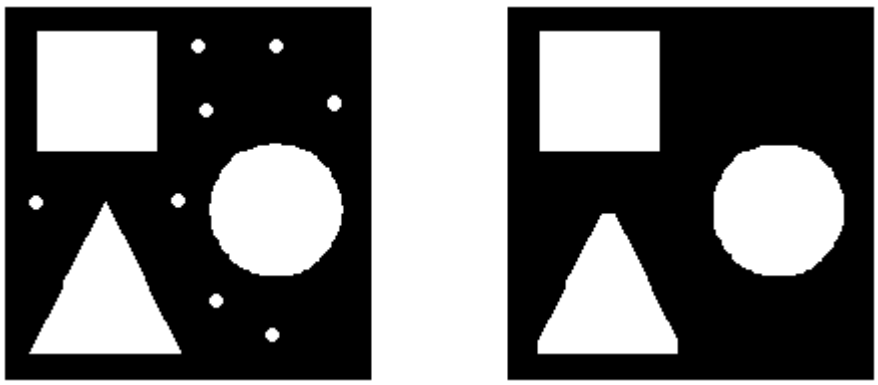


Figure 2.9 Binary opening process

The effects of the opening mentioned before are clearly shown in the Figure 2.9. The original image on the left of the Figure 2.9 are opened by a 7x7 square structuring element The vortices of the triangle foreground have been cut out because the original image is opened by a structuring element whereas the sides of the square are preserved due to the square structuring element.

2.2.6 Binary Closing

The closing of a binary image *A* by the structuring element *B*, denoted by $A \bullet B$ is defined as

$$A \bullet B = \left(A \oplus B\right) \Theta B \tag{2.24}$$

Thus, the closing *A* by *B* is the dilation of *A* by *B*, followed by an erosion of the result by *B*.

Closing has a similar geometric interpretation, except that in this case, the structuring element *B* is rolled on the outside of the boundary of the image object *A* as seen in Figure 2.10.

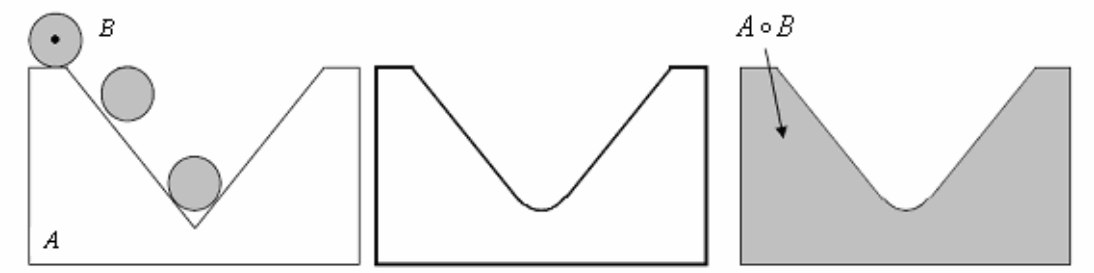


Figure 2.10 Illustration of binary closing process

In another aspect, the definition of closing can be given as; a point *w* is an element of $A \bullet B$ if and only if $\left(B\right)_z \bigcap A \neq \varnothing$ for any translate of $\left(B\right)_z$ that contains *w*.

Closing operation tends to smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and along thin gulfs, eliminates small holes, and fills gaps in the contour. Figure 2.11 shows how closing works

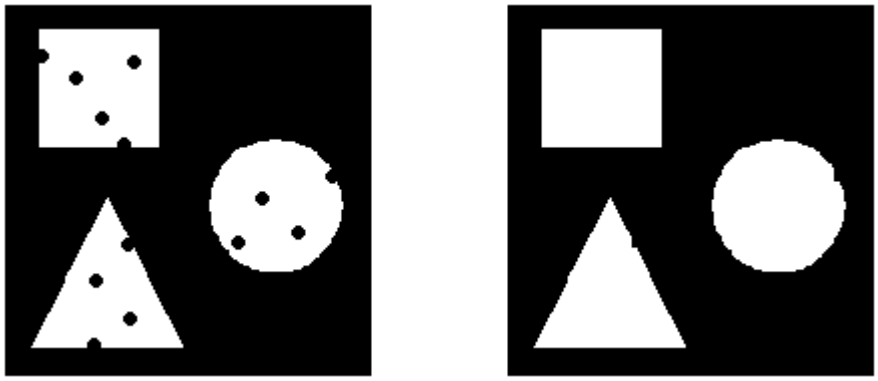


Figure 2.11 Binary closing process

2.2.7 Properties of Binary Opening and Closing

a) Opening and closing operations are translation invariant

$$A \circ (B)_x = A \circ B \qquad A \bullet (B)_x = A \bullet B \tag{2.25}$$

b) Opening is anti-extensive

$$A \circ B \subseteq A \tag{2.26}$$

c) Closing is extensive

$$A \subseteq A \bullet B \tag{2.27}$$

d) Opening and closing are duals of each other

$$(A \bullet B)^c = A^c \circ \hat{B} \tag{2.28}$$

2.3 Grayscale Morphology

A gray-scale image can be considered as a three-dimensional set where the first two elements are the *x* and *y* coordinates of a pixel and the third element is gray-scale value. In this section binary morphology is extended to gray-scale morphology. The key issue in grayscale morphology is to use 'Maximum' and 'Minimum' functions to define gray-scale morphological operators. Using these concepts, gray-scale morphology can be

easily extended from binary morphology. The differences between binary and gray-scale morphology results from the definitions of dilations and erosion because other operators basically depend on these. Except for these definitions, gray-scale morphology is fairly similar to the binary case. Hence in this section, definitions for gray-scale dilation and erosion as well as some of examples for gray-scale dilation and erosion are given.

Before giving the definitions of gray-scale dilation and erosion, structuring elements of the gray-scale morphological operations could have the same domains as those in binary morphology. However as will be seen in the definitions below, a gray-scale structuring element has certain values ('b') instead of having only position value '1' or '0' showing its domain. Therefore as mentioned before, grayscale image can be thought of as a three-dimensional (*x, y*) coordinates and grayscale value at that point. It can also be applied to the gray-scale structuring element. With this concept definition gray-scale definition of grayscale dilation and erosion are given below.

2.3.1 Gray-scale Dilation

Grayscale dilation of *f* by *b*, denoted $f \oplus b$,is defined as

$$(f \oplus b)(s,t) = \max\left\{ f(s-x, t-y) + b(x,y) \mid (s-x), (t-y) \in D_f ; (s,y) \in D_b \right\} \quad (2.29)$$

where D_f and D_b are the domains of f and *b*, respectively.

An example of gray-scale dilation is given below. F represent a 5×5 image, B represents a 3×3 non-flat structuring element as given in Figure 2.12

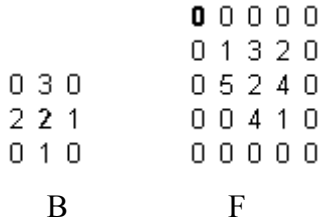


Figure 2.12 Calculation of grayscale dilation

The x, y plane is shown with the numbers representing the intensity at (x, y). The origin of B is shown by the bold number at the center. The origin of F is at the upper left corner. To start the algorithm, the origin of B is superimposed on the center element of F. Since B is a 3×3, the boundary elements of F is ignored. Those values are set to zero. For $F_d(1,1)$ calculation is as follows using equation 2.29.

$f_d(1,1) = \max\{f(1-i,1-j)+b(i,j) \mid i=-1,0,1;\ j=-1,0,1\}$

$= \max\{f(0,0)+b(1,1),\ f(0,1)+b(1,0),\ f(0,2)+b(1,-1)$

$f(1,0)+b(0,1),\ f(1,1)+b(0,0),\ f(1,2)+b(0,-1)$

$f(2,0)+b(-1,1),\ f(2,1)+b(-1,0),\ f(2,2)+b(-1,-1)\},$

$= \max\{0+0,\ 0+1,\ 0+0$

$0+1,\ 1+2,\ 5+3$

$0+0,\ 3+2,\ 2+0\}$

$= \max\{0,\ 1,\ 0$

$1,\ 3,\ 8$

$0,\ 5,\ 2\}$

$= 8$

If we continue this process we obtain the result shown in Figure 2.13 below. The grayscale value of f(1,1) is 8.

3	4	6	5	3
3	**8**	5	7	3
7	7	7	6	5
3	6	6	5	3
2	2	5	2	2

Figure 2.13 Gray-scale dilation example

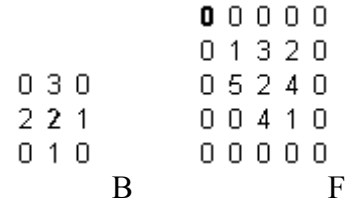
2.3.2 Gray-scale Erosion

Grayscale dilation of f by b, denoted $f \Theta b$,is defined as

$$(f \Theta b)(s,t) = \max\left\{ f(s+x, t+y) - b(x,y) \mid (s+x), (t+y) \in D_f ; (s,y) \in D_b \right\} \tag{2.30}$$

where D_f and D_b are the domains of each image function.

An example of the gray-scale erosion is given below. If the same F and the same structuring element B is used in Figure 2.12, then the calculation of gray-scale erosion using equation 2.30 is as follows,



$f_e(2,2) = \min\{f(2+i,2+j)-b(i,j)|\ i=-1,0,1;\ j=-1,0,1\}$

$= \min\{f(1,1)-b(-1,-1),\ f(1,2)-b(-1,0),\ f(1,3)-b(-1,1)$

$f(2,1)-b(0,-1),\ f(2,2)-b(0,0),\ f(2,3)-b(0,1)$

$f(3,1)-b(1,-1),\ f(3,1)-b(1,0),\ f(3,2)-b(1,1)\},$

$= \min\{1-0,\ 5-3,\ 0-0$

$1-2,\ 2-0,\ 4-1$

$0-0,\ 0-1,\ 4-0\}$

$= \min\{1\ ,2,\ 0$

$-1,\ 2,\ 3$

0, -1, 4}

= -1

-3 -3 -3 -3 -3
-3 -2 -1 -1 -3
-3 -2 **-1** -1 -3
-3 -2 -2 -2 -3
-3 -3 -3 -3 -3

Figure 2.14 Gray-scale erosion example

Gray-scale dilation and erosion are duals with respect to function completion and reflection. That is, the relation between these can be expressed as

$$\left(f \Theta b\right)^{c}\left(s,t\right) = \left(f^{c} \oplus \hat{b}\right)\left(s,t\right) \tag{2.31}$$

where

$$f^{c} = -f(x,y) \text{ and } \hat{b} = b(-x,-y) \tag{2.32}$$

The minimum operator will interrogate a neighborhood with a certain domain and select the smallest pixel value to become the output value. This has the effect of causing the bright areas of an image to shrink or erode. Similarly gray-scale dilation is performed by using the maximum operator to select the greatest value in neighborhood. Figure 2.15 shows a simple image and its dilation and erosion with a 'flat top' structuring element. The term 'flat top' refers to the fact that the values (b's) of the structuring element are all zero in a certain domain. In this example a 512 x 512 gray-scale Lena image and a disk-shaped structuring element is used. The radius of the structuring element is 4 pixels.

Figure 2.15 An example of gray-scale dilation and erosion

2.3.3 Gray-scale Opening and Closing

Gray-scale opening and closing are defined below in a similar manner as the binary case. The only difference is, when the operations are carried out, these opening and closing operations use gray-scale dilation and erosion described in the previous section.

The effect of gray-scale opening and closing is shown in Figure 2.16. As binary morphological operations do, gray-scale opening is anti-extensive and gray-scale closing is extensive. Both operations make an original image smooth along the nature of minimum and maximum functions. Also, both operations have 'increasing', 'idempotent' properties.

Figure 2.16 An example of gray-scale opening and closing

2.4 The Hit or Miss Transformation

The hit-and-miss transform is a basic tool for shape detection. This requires two structure elements. One element defines the foreground features to detect while the other defines the background features to detect.

The hit-and-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image. It is actually the basic operation of binary morphology since almost all the other binary morphological operators can be derived from it. As with other binary morphological operators it takes as input a binary image and a structuring element (or two structuring elements), and produces another binary image as output.

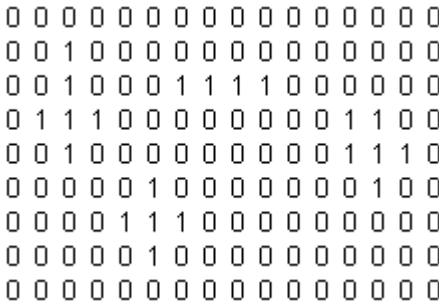
Hit-and-miss is defined as the intersection of the erosion of A (the object or objects in a binary image) by the first structure element and the erosion of the complement of A by the second structure element.

$$HoMT = \left(A \Theta B_1\right) \cap \left(A^c \Theta B_2\right) \tag{2.33}$$

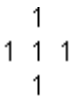
So, the output of a hit-or-miss transform is the set of points which match the erosion of A by SE_1 (the foreground) and those which match the erosion of the complement of A with SE_2 (the background). The result is the matching of certain edge features (exactly which features depends on the shapes within the two structure elements).

The hit-or-miss operation is performed in much the same way as other morphological operators, by translating the origin of the structuring element to all points in the image, and then comparing the structuring element with the underlying image pixels. If the foreground and background pixels in the structuring element exactly match foreground and background pixels in the image, then the pixel underneath of the origin of the structuring element is set to foreground color. If it does not match, then that pixel is set to the background color.

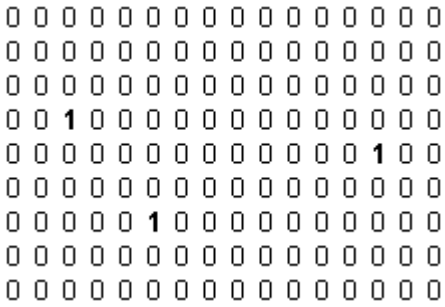
Figure below shows the hit-or-miss transformation to identify the locations of the cross-shaped pixel configuration



a) Original image

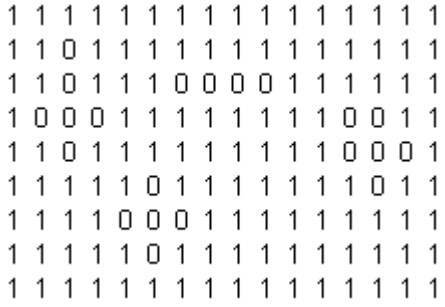


b) Structuring element B_1

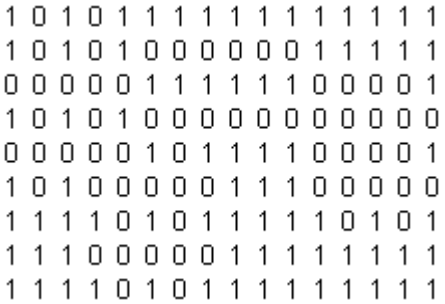


c) Erosion of A by B_1

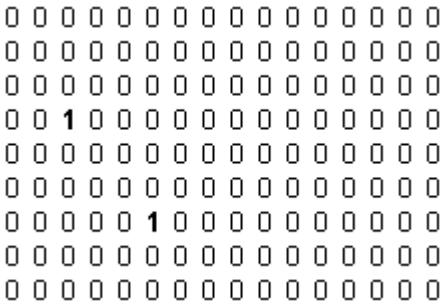
d) Structuring element B_2



e) Complement of the original image



f) Erosion of A^c by B_2



g) Output image

Figure 2.17 Hit or miss transformation

Figure 2.17 a) contains this configuration of pixels in two different locations. Erosion with structuring element B_1 determines the locations of foreground pixels that have north, east, south, and west foreground neighbors. Erosion of the complement with structuring element B_2 determines the locations of all the pixels whose northeast, southeast, southwest, and northwest neighbors belong to the background. Figure 2.17 g) shows the intersection of these two operations. Each foreground pixel of Figure 2.17 g) is the location of a set of pixels having the desired configuration.

2.5 Thinning

Thinning [22] means reducing binary objects or shapes in an image to strokes that are a single pixel wide. The thinning of a set *A* by structuring element *B*, denoted $A \otimes B$, can be defined in terms of the hit-or-miss transform:

$$A \otimes B = A - HoMT(A, B) \tag{2.34}$$

A more useful expression for thinning *A* symmetrically is based on a sequence of structuring elements:

$$\{B\} = \{B^1, B^2, B^3, ..., B^n\} \tag{2.35}$$

where B^i is a rotated version of B^{i-1}. Then definition for thinning is given as:

$$A \otimes \{B\} = \left(\left(... \left(\left(A \otimes B^1 \right) \otimes B^2 \right) ... \right) \otimes B^n \right) \tag{2.36}$$

The process is to thin A by one pass with B^1, then thin the result with one pass of B^2, and so on, until A is thinned with one pass of B^n. The entire process is repeated until no further changes occur. Each individual thinning pass is performed using Equation 2.34.

2.6 Structuring element Decomposition

When computing morphological operations on images, it is necessary to decompose, if possible, the structuring element into smaller ones.

It is shown that dilation is associative. That is

$$A \oplus \left(B \oplus C \right) = \left(A \oplus B \right) \oplus C \tag{2.37}$$

Suppose that a structuring element *B* can be represented as a dilation of two structuring elements B_1 and B_2:

$$B = B_1 \oplus B_2 \tag{2.38}$$

Then $A \oplus B = A \oplus \left(B_1 \oplus B_2 \right) = \left(A \oplus B_1 \right) \oplus B_2$. In other words, dilating *A* with *B* is the same as first dilating A with B_1, and then dilating the result with B_2 .Then it is concluded that B can be decomposed into the structuring elements B_1 and B_2.

The associative property is important because the time required computing dilation is proportional to the number of nonzero pixels in the structuring element.

For example, dilation with an 5×5 array of 1's:

1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1

This structuring element can be decomposed into a five-element row of 1s and a five-element column of 1s:

$$\begin{bmatrix} 1 & 1 & \boxed{1} & 1 & 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ \boxed{1} \\ 1 \\ 1 \end{bmatrix}$$

The number of elements in the original structuring element is 25, but the total number of elements in the row-column decomposition is only 10. This means that dilation with the row structuring element first, followed by dilation with the column element, can be performed 2.5 times faster than dilation with the 5×5 array of 1s.

CHAPTER 3

EDGE DETECTION USING CLASSICAL EDGE DETECTORS

Image edge detection is a basic tool in image segmentation since edges carry valuable features of the image. Edges in image are formed due to variations of illumination in the scene. Hence, the conventional approach to edge detection is composed of "gradient calculation" and "thresholding". For the first step, the original image is transformed into a gradient image which represents the edge strength of each pixel. A threshold is then applied to classify each pixel to the edge point or non-edge point. Traditionally, the gradient image can be obtained by means of first-order differential operators or a Laplacian operator which can enhance the spatial intensity changes in the image. Morphological edge detectors [10-12] have also been proposed for their robustness under noisy conditions and some of them are discussed in the following section. The classification of edge detectors discussed in this chapter is based on the behavioral study of these edges with respect to the following operators:

- Gradient based edge detectors
- Laplacian of Gaussian
- Gaussian edge detectors

3.2 Gradient Based Edge Detectors

Traditional methods to find the gradient of an image typically apply a first-order differential operation on the original image. This operation is similar to linear filtering which consists of scanning the original image with certain masks. The sizes and values of the mask are derived from standard mathematical expressions and approximated with ad

hoc procedures to digital settings. The most popular gradient-based edge detectors are Roberts cross gradient operator, Sobel operator and the Prewitt operator.

3.2.1 Derivative of a Digital Function

The derivatives of a digital function are defined in terms of differences. There are various ways to define these differences. However, any definition used for a first derivative 1) must be zero in flat segments (areas of constant gray-level values); 2) must be nonzero at the onset of a gray-level step or ramp; and 3) must be nonzero along ramps. Similarly, any definition of a second derivative 1) must be zero in flat areas; 2) must be nonzero at the onset and end of a gray-level step or ramp; and 3) must be zero along ramps of constant slope.

A basic definition of the first-order derivative of a one-dimensional function $f(x)$ is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \tag{3.1}$$

partial derivative is used in the above equation because an image function is composed of two variables, $f(x, y)$.

Similarly, definition of second order derivative as the difference is

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \tag{3.2}$$

3.3 Gradient Operators

First-order derivatives of a digital image are based on various approximations of the 2-D gradient. The gradient of an image $f(x, y)$ at location (x, y) is defined as the vector

$$\Delta \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{3.3}$$

it is well known from vector analysis that the gradient vector points in the direction of maximum rate of change of f at coordinates (x, y).

An important quantity in edge detection is the magnitude of this vector, denoted Δf, where

$$\begin{aligned}\Delta f &= \text{mag}(\Delta \text{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}\end{aligned} \tag{3.4}$$

The computational complexity of implementing Eq (3.4) over an entire image is not trivial, and it is common practice to approximate the magnitude of the gradient by using the absolute values instead of squares and square roots:

$$\Delta \text{f} \approx \left|G_x\right| + \left|G_y\right| \tag{3.5}$$

This quantity gives the maximum rate of increase of $f(x, y)$ per unit distance in the direction of Δf. It is a common (although not strictly correct) practice to refer to Δf also as the gradient.

The direction of the gradient vector is also an important property. Let $a(x, y)$ represent the direction angle of the vector Δf at (x, y). Then, from vector analysis,

$$a(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{3.6}$$

where the angle is measured with respect to the x-axis. The direction of an edge at (x, y) is perpendicular to the direction of the gradient vector at that point. The other method of calculating the gradient is given by estimating the finite differences.

$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x + h, y) - f(x, y)}{h} \tag{3.7}$$

$$\frac{\partial f}{\partial y} = \lim_{h \to 0} \frac{f(x, y+h) - f(x, y)}{h} \tag{3.8}$$

therefore this finite difference is approximated as

$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x+h, y) - f(x, y)}{h_x} = f(x+1, y) - f(x, y) \qquad (h_x = 1) \tag{3.9}$$

$$\frac{\partial f}{\partial y} = \frac{f(x, y+h) - f(x, y)}{h_y} = f(x, y+1) - f(x, y) \qquad (h_y = 1) \tag{3.10}$$

Computation of the gradient of an image is based on obtaining the partial derivatives $\partial f / \partial x$ and $\partial f / \partial y$ at every pixel location. Let the 3×3 area shown in Figure 3.1 represent the gray levels in a neighborhood of an image

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Figure 3.1 3×3 Neighborhood of an image

The center point, z_5, denotes $f(x, y)$, z_1 denotes $f(x-1, y-1)$, and so on.

3.3.1 Roberts Edge Detector

It is known that the simplest approximations to a first-order derivative that satisfy the conditions stated are $G_x = (z_8 - z_5)$ and $G_y = (z_6 - z_5)$. Two other definitions proposed by Roberts [8] are

$$G_x = (z_9 - z_5) \tag{3.11}$$

$$G_y = (z_8 - z_6) \tag{3.12}$$

If we used the equation (3.4), then gradient is computed as

$$\Delta f = \left[\left(z_9 - z_5 \right)^2 + \left(z_8 - z_6 \right)^2 \right]^{1/2} \tag{3.13}$$

If Eq (3.5) is used to calculate the approximated gradient values, then

$$\Delta f \approx \left| z_9 - z_5 \right| + \left| z_8 - z_6 \right| \tag{3.14}$$

this equation can be implemented using the masks given in Figure 3.2 and these masks are referred to as the Roberts cross-gradient operators. Masks of even size are hard to implement because they do not have a clear center.

The calculation of the gradient magnitude of an image is obtained by the partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ at every pixel location. The simplest way to implement the first order partial derivative is by using the Roberts cross gradient operator. Therefore

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \tag{3.15}$$

$$\frac{\partial f}{\partial y} = f(y+1) - f(y) \tag{3.16}$$

The operators given above can be implemented by approximating them two 2×2 masks. The Roberts operator masks are

-1	0
0	1

0	-1
1	0

Figure 3.2 Roberts operator

These filters have the shortest support, thus the position of the edges is more accurate, but the problem with the short support of the filters is its sensitivity to noise. It also produces very weak responses to genuine edges unless they are very sharp.

3.3.2 Prewitt Edge Detector

The Prewitt edge detector is a much better operator than the Roberts operator. This operator having a 3×3 masks deals better with the effect of noise. The partial derivatives of the Prewitt operator are calculated as

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \tag{3.17}$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \tag{3.18}$$

Therefore the Prewitt masks are as follows

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

Figure 3.3 Prewitt masks

These masks have longer support. They differentiate in one direction and average in the other direction, so the edge detector is less vulnerable to noise.

It is possible to modify the 3×3 masks in Figures 3.3 so that they have their strongest responses along the diagonal directions.

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

Figure 3.4 Modified Prewitt masks

3.3.3 Sobel Edge Detector

The Sobel edge detector is very much similar to the Prewitt edge detector. The difference between the both is that the weight of the center coefficient is 2 in the Sobel operator. The partial derivatives of the Sobel operator are calculated as

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \tag{3.19}$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \tag{3.20}$$

$$\Delta f = \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right| \tag{3.21}$$

and the Sobel masks are as follows

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

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

Figure 3.5 Sobel operator

The difference between the third and first rows of the 3×3 image region approximates the derivative in the *x*-direction, and the difference between the third and first columns approximates the derivative in the *y*-direction. The masks shown in Figure 3.5 are called Sobel operators.

Although the Prewitt masks are easier to implement than the Sobel masks, the later has better noise suppression characteristics.

It is possible to modify the 3×3 masks in Figure 3.4 so that they have their strongest responses along the diagonal directions. The two additional Sobel masks for detecting discontinuities in the diagonal directions are shown in Figure 3.6

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

Figure 3.6 Modified Sobel masks

3.3.4 The Laplacian

The principle used in Laplacian of Gaussian method is, the second derivative of a signal is zero when the magnitude of the derivative is maximum. The Laplacian for a function image $f(x, y)$ of two variables is defined as

$$\left(\Delta^2 f\right)(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{3.22}$$

In order to be useful for digital image processing, this equation needs to be expressed in discrete form. The partial second-order derivative in the *x*-direction

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \tag{3.23}$$

and similarly in the *y*-direction, as

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \tag{3.24}$$

The digital implementation of the two-dimensional Laplacian is obtained by summing these two components:

$$\Delta^2 \mathrm{f} = \left[f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \right] \tag{3.25}$$

and the other implementation of the two-dimensional Laplacian is

$$\Delta^2 f = \begin{bmatrix} f(x-1,y+1)+f(x,y+1)+f(x+1,y+1)+f(x-1,y)-8f(x,y)+ \\ f(x+1,y)+f(x-1,y-1)+f(x,y-1)+f(x+1,y-1) \end{bmatrix} \quad (3.26)$$

This equation can be implemented using the mask shown in Figure 3.7,

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1

Figure 3.7 Laplacian

The above partial derivative equations are isotropic for rotation increments of 90° and 45°, respectively. Edge detection is obtained by convolving an image with the Laplacian at a given scale and then mark the points where the result have zero value, which is called the zero –crossings. These points should be controlled to ensure that the gradient magnitude is large. The algorithm is developed by Marr Hildreth [13].

The Laplacian generally is not used in its original form for edge detection for several reasons: As a second-order derivative, the Laplacian typically is unacceptably sensitive to noise. The magnitude of the Laplacian produces double edges. Finally, the Laplacian is unable to detect edge direction.

3.4 Marr and Hildreth method

The Roberts and Sobel operators are the first derivative edge detectors. These filters give very little control over smoothing and edge localization. Marr proposed a Laplacian of Gaussian (LOG) method, which detects edges by checking zero-crossing of the second derivative of Gaussian filtered image.

The two-dimensional Gaussian function with zero mean and standard deviation σ is defined by the equation

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{x^2}{2\sigma^2}\right)} e^{\left(-\frac{y^2}{2\sigma^2}\right)} \tag{3.27}$$

For continuous variables Laplacian of Gaussian is defined as

$$\Delta^2(G(x,y)) = \frac{x^2+y^2-2\sigma^2}{2\pi\sigma^6} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} \tag{3.28}$$

The edge pixels in image are determined by a single convolution operation. The basic principle of this method is to find the position in an image where the second derivatives become zero. These positions correspond to edge positions. The Gaussian function firstly smoothes or blurs any edge. Blurring is advantageous here because Laplacian would be infinity at unsmoothed edge and therefore edge position is still preserved. LOG operator is still sensitive to noise, but by ignoring zero-crossings produced by small changes in image intensity can reduce the effects of noise.

CHAPTER 4

EDGE DETECTION USING MORPHOLOGICAL OPERATORS

A simple method of performing gray-scale edge detection in a morphology-based vision system is to take the difference between an image and its erosion by a small rod shaped structuring element. The difference image is the image of edge strength. And if a threshold operation is applied to the image, a binary edge image can be obtained. The simplest morphological edge detectors are the dilation residue and erosion residue edge detectors [10].

4.1 Erosion Residue Edge Detector

The center of local neighborhood is $(0,0)$ and a point which is a distance δr from the center in row direction and a distance δc from the center in column direction is denoted by $(\delta r, \delta c)$.

The erosion of a gray-scale image $f(r,c)$ by the rod structuring element b of radius 1 is given by the formula.

$$e(r,c) = \min_{(i,j)\in D_{rod1}} [f(r+i, c+j) - b(r,c)] \tag{4.1}$$

where the domain of the structuring element is

$$D_{rod1} = \{(0,-1),(0,1),(-1,0),(1,0)\} \text{ and } b: D_{rod1} \to \{0,...,255\}$$

$$\begin{matrix} & 0 & \\ 0 & * & 0 \\ & 0 & \end{matrix}$$

Figure 4.1 Rod structuring element

Since a rod is flat on top, the gray-scale value of all the $b(r,c), (r,c) \in D_{rod1}$, is zero.

Then the formula given in (4.1) for the zero height structuring element becomes

$$e(r,c) = \min_{(i,j)\in D_{rod1}} [f(r+i,c+j)] \tag{4.2}$$

The erosion residue edge detector produces the edge strength image G_e defined by

$$\begin{aligned} G_e(r,c) &= f(r,c) - e(r,c) \\ &= f(r,c) - \min_{(i,j)\in D_{rod1}} [f(r+i,c+j)] \\ &= \max_{(i,j)\in D_{rod1}} [f(r,c) - f(r+i,c+j)] \end{aligned} \tag{4.3}$$

Since D_{rod1} includes exactly the four connected neighbors of position $(0,0)$. The edge strength image is

$$G_e(r,c) = \max_{(i,j)\in N_4(r,c)} \left[f(r,c) - e(r,c) \right] \tag{4.4}$$

where $N_4(r,c)$ is the set of four connected neighbors of position (r,c).

To compare the performance of the nonlinear morphological edge operator with the linear Laplacian operator, it is applied to four perfect digital step edge patterns of edge contrasts *E* running in directions 0°, 90°, 45°, and 135°, respectively.

$$\begin{matrix} E & E & E \\ E & E & E \\ 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} E & E & 0 \\ E & E & 0 \\ E & E & 0 \end{matrix} \qquad \begin{matrix} E & E & E \\ E & E & 0 \\ E & 0 & 0 \end{matrix} \qquad \begin{matrix} E & E & E \\ 0 & E & E \\ 0 & 0 & E \end{matrix}$$

$$E_{0^\circ} \qquad E_{90^\circ} \qquad E_{45^\circ} \qquad E_{135^\circ}$$

Figure 4.2 Digital Step Edge Patterns

The magnitude of the responses of G and $\Delta^2 f$ are as follows:

	E_{0°	E_{90°	E_{45°	E_{135°
G_e	E	E	E	E
$\Delta^2 f$	E	E	$2E$	$2E$

Figure 4.3 Erosion residue edge detector and Laplacian

Thus the responses of $\Delta^2 f$ to these edges are *E*, *E*, *2E* and *2E*. For diagonal edges Laplacian is two times the original value of the edge patterns. This problem can be solved if morphological operator G_e is used instead of $\Delta^2 f$. G_e produces values in the same range as the original gray-scale values, which is most convenient on any computer vision system.

Both operators are applied to a single noise pattern with noise height *h* given in Figure 4.4.

$$\begin{matrix} 0 & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & 0 \end{matrix}$$

Figure 4.4 Single noise pattern

The responses of G_e and $\Delta^2 f$ are *h* and *4h*, respectively. Thus although both G_e and $\Delta^2 f$ are noise sensitive, the noise response of $\Delta^2 f$ is four times the response of G_e and hence four times the response of $\Delta^2 f$ on a horizontal or vertical ideal step edge with edge contrast *h*.

If the neighborhood size of the morphologic edge detector is increased by increasing the size of the structuring element, for example; an eight-connected neighborhood edge operator by changing the structuring element to be flat on top and have domain in given in equation (4.5)

$$D_{8-connected} = \{(-1,-1),(0,-1),(1,-1),(-1,0),(1,0),(-1,1),(0,1),(1,1)\} \qquad (4.5)$$

then the edge strength then will be

$$G_{e8-connected}(r,c) = \max_{(i,j)\in N_8(r,c)} \left[f(r,c) - f(i,j) \right] \tag{4.6}$$

where $N_8(r,c)$ is the set of eight connected neighbors of image position (r,c).

The corresponding Laplacian operator for the eight connected neighborhood is

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Then application of this operator to the four perfect digital step edge patterns $E_{0^\circ}, E_{90^\circ}, E_{45^\circ}, E_{135^\circ}$. The magnitude of the responses of G_e and $\Delta^2 f$ are as follows:

	E_{0°	E_{90°	E_{45°	E_{135°
$G_{e8-connected}$	E	E	E	E
$\Delta^2 f_{8-onnected}$	$3E$	$3E$	$3E$	$3E$

Figure 4.5 Eight connected erosion residue and Laplacian

with increased neighborhood size, the $\Delta^2 f$ operator achieves uniform performance on these edges. The response of G_e and $\Delta^2 f$ on a single noise pattern given in Figure 4.4 is *h* and *8h*, respectively. Since both operators are noise sensitive but Laplacian being more sensitive. This shows that raw Laplacian operator is not a good edge detector in noisy images.

It is common to filter noisy images by a Gaussian filter and then apply a Laplacian operator. Edges are localized at zero crossings of the Laplacian (Marr and Hildreth). But in this case Laplacian can change the locations of the most of the edges in real images.

Since the erosion residue morphologic edge detector is noise sensitive; it can not be a good edge detector. Increasing the neighborhood size of the structuring element for

erosion residue edge detector fails in reducing the amount of noise. If erosion residue edge detector is applied to the image pattern shown in Figure 4.6

$$\begin{matrix} F & F & F & F & F \\ F & F & F & F & F \\ F & F & 0 & F & F \\ F & F & F & F & F \\ F & F & F & F & F \end{matrix}$$

Figure 4.6 Noise pattern

Then the pattern given in Figure 4.6 is a flat area with pixel intensity F and a noise spike at the center of this area with pixel intensity zero. The response of morphological edge operator is given in Figure 4.7.

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & F & F & F & 0 \\ 0 & F & 0 & F & 0 \\ 0 & F & F & F & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Figure 4.7 Response to noise pattern of morphological residue operator

It has the same value F for each of the eight-connected neighbors of the center point. Thus increasing the size of the neighborhood of the morphological operator results an increase in the pixel which is assigned to F. As a matter of fact, each pixel of the neighborhood support except the center point will be assigned to F. Therefore a larger neighborhood can result in worse results with this operator.

The erosion residue morphological edge detector is position biased. It only gives edge strength to border pixels on that side of the edge where the pixels have the higher value.

To solve this problem, the dilation residue edge detector is used conjunction with the erosion residue edge detector.

4.2 Dilation Residue Edge Detector

The dilation residue edge detector takes the difference between a dilated image and its original image. If the structuring element for the dilation is a rod of radius 1, then the dilation of the gray-scale image $f(r,c)$ is

$$d(r,c) = \max_{(i,j)\in D_{rod1}} = \left[f(r-i, c-j) \right] \tag{4.7}$$

and the edge strength image is

$$\begin{aligned} G_d(r,c) &= d(r,c) - f(r,c) \\ &= \max_{(i,j)\in N_4(x,y)} [f(i,j) - f(r,c)] \end{aligned} \tag{4.8}$$

This operator only gives the edge strength to that side of the edge which has the lower value.

A position independent edge operator is obtained by a combination of operators $G_e(r,c)$ and $G_d(r,c)$ using the pixel wise minimum, maximum, or sum. That is

$$\begin{aligned} E_{\max}(r,c) &= \max(G_e(r,c),\ G_d(r,c)) \\ &= \max_{(i,j)\in N_{(r,c)}} [f(r,c) - f(i,j)] \end{aligned} \tag{4.9}$$

$N(r,c)$ is the neighborhood support of the structuring element for both dilation and erosion operators.

To see the performance of the operator $E(r,c)$, it is applied on four perfect digital step edge patterns of edge contrast *E*, one ideal ramp edge pattern of edge contrast *E*, and one single noise pattern of amplitude *N*.

$$\begin{matrix} E & E & E & E & E \\ E & E & E & E & E \\ E & E & E & E & E \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad E_{0^\circ} \qquad \begin{matrix} E & E & E & 0 & 0 \\ E & E & E & 0 & 0 \\ E & E & E & 0 & 0 \\ E & E & E & 0 & 0 \\ E & E & E & 0 & 0 \end{matrix} \quad E_{90^\circ} \qquad \begin{matrix} E & E & E & E & E \\ E & E & E & E & 0 \\ E & E & E & 0 & 0 \\ E & E & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 \end{matrix} \quad E_{45^\circ}$$

$$\begin{matrix} E & E & E & E & E \\ 0 & E & E & E & E \\ 0 & 0 & E & E & E \\ 0 & 0 & 0 & E & E \\ 0 & 0 & 0 & 0 & E \end{matrix} \quad E_{135^\circ} \qquad \begin{matrix} 0 & 0 & E/2 & E & E \\ 0 & 0 & E/2 & E & E \\ 0 & 0 & E/2 & E & E \\ 0 & 0 & E/2 & E & E \\ 0 & 0 & E/2 & E & E \end{matrix} \quad E_{r0^\circ} \qquad \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & N & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad \text{noise}$$

Figure 4.8 Different directions step images and noise

The results of this operator are given in Figure 4.9

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ E & E & E & E & E \\ E & E & E & E & E \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad E_{0^\circ} \qquad \begin{matrix} 0 & 0 & E & E & 0 \\ 0 & 0 & E & E & 0 \\ 0 & 0 & E & E & 0 \\ 0 & 0 & E & E & 0 \\ 0 & 0 & E & E & 0 \end{matrix} \quad E_{90^\circ} \qquad \begin{matrix} 0 & 0 & 0 & 0 & E \\ 0 & 0 & 0 & E & E \\ 0 & 0 & E & E & 0 \\ 0 & E & E & 0 & 0 \\ E & E & 0 & 0 & 0 \end{matrix} \quad E_{45^\circ}$$

$$\begin{matrix} E & 0 & 0 & 0 & 0 \\ E & E & 0 & 0 & 0 \\ 0 & E & E & 0 & 0 \\ 0 & 0 & E & E & 0 \\ 0 & 0 & 0 & E & E \end{matrix} \quad E_{135^\circ} \qquad \begin{matrix} 0 & E/2 & E/2 & E/2 & 0 \\ 0 & E/2 & E/2 & E/2 & 0 \\ 0 & E/2 & E/2 & E/2 & 0 \\ 0 & E/2 & E/2 & E/2 & 0 \\ 0 & E/2 & E/2 & E/2 & 0 \end{matrix} \quad E_{r0^\circ} \qquad \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & N & 0 & 0 \\ 0 & N & N & N & 0 \\ 0 & 0 & N & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad \text{noise}$$

Figure 4.9 Result of dilation residue edge detector

It is understood from the results that the edge operator $E_{\max}(r,c)$ performs perfectly on ideal edge step patterns however it is not very good for the ramp edge pattern and sensitive to noise. It responds with five noise patterns to a single noise pattern.

In the second case summation version of the edge operator is investigated which is given by the formula

$$E_{sum}(r,c) = G_e(r,c) + G_d(r,c) \tag{4.10}$$

Similarly this operator is applied to the edge patterns given in Figure 4.8

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ E & E & E & E & E \\ E & E & E & E & E \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad E_{0^\circ} \qquad \begin{matrix} 0 & 0 & E & E & 0 \\ 0 & 0 & E & E & 0 \\ 0 & 0 & E & E & 0 \\ 0 & 0 & E & E & 0 \\ 0 & 0 & E & E & 0 \end{matrix} \quad E_{90^\circ} \qquad \begin{matrix} 0 & 0 & 0 & 0 & E \\ 0 & 0 & 0 & E & E \\ 0 & 0 & E & E & 0 \\ 0 & E & E & 0 & 0 \\ E & E & 0 & 0 & 0 \end{matrix} \quad E_{45^\circ}$$

$$\begin{matrix} E & 0 & 0 & 0 & 0 \\ E & E & 0 & 0 & 0 \\ 0 & E & E & 0 & 0 \\ 0 & 0 & E & E & 0 \\ 0 & 0 & 0 & E & E \end{matrix} \quad E_{135^\circ} \qquad \begin{matrix} 0 & E/2 & E & E/2 & 0 \\ 0 & E/2 & E & E/2 & 0 \\ 0 & E/2 & E & E/2 & 0 \\ 0 & E/2 & E & E/2 & 0 \\ 0 & E/2 & E & E/2 & 0 \end{matrix} \quad E_{r0^\circ} \qquad \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & N & 0 & 0 \\ 0 & N & N & N & 0 \\ 0 & 0 & N & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad \text{noise}$$

Figure 4.10 Summation version of dilation and erosion residue edge operator

It is seen from Figure 4.10 that the results are same for the summation version of the edge detector as in the maximum version except for the ramp edge pattern. For the ramp edge pattern, it detects an edge line whose edge strength equals edge contrast and two lines on both sides of the edge line whose edge strength equals half-edge contrast. Therefore if a threshold operation is applied to resultant ramp edge pattern greater than the half-edge contrast, it is possible to have perfect performance.

As a final case minimum version of the edge detector is investigated given by the formula

$$E_{\min}(r,c) = \min(G_e(r,c),\ G_d(r,c)) \tag{4.11}$$

The result of this operator to same edge patterns given in Figure 4.8 are shown below.

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad E_{0^\circ} \qquad \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad E_{90^\circ} \qquad \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad E_{45^\circ}$$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad E_{135^\circ} \qquad \begin{matrix} 0 & 0 & E/2 & 0 & 0 \\ 0 & 0 & E/2 & 0 & 0 \\ 0 & 0 & E/2 & 0 & 0 \\ 0 & 0 & E/2 & 0 & 0 \\ 0 & 0 & E/2 & 0 & 0 \end{matrix} \quad E_{r0^\circ} \qquad \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad \text{noise}$$

Figure 4.11 Minimum version of dilation and erosion residue operators

The result of this operator for the ramp edge pattern and noise are promising. Because it is noise insensitive and it detects a single edge line having an edge contrast $E/2$. But it is not good for the detection of step edge patterns. So if we blur the step edge patterns first and then apply the edge detector, result will be very good in performance.

4.3 Effective Morphologic Edge Detectors

From the results of the simplest edge detectors, they are either sensitive to noise or can not detect step edge patterns. In this section improved versions of the morphologic edge detectors are introduced.

Let D_1, D_2, D_3, D_4 and D be structuring elements which have domains

$$
\begin{aligned}
D_1 &= \{(-1,0),(0,0),(0,1)\} \\
D_2 &= \{(0,-1),(0,0),(1,0)\} \\
D_3 &= \{(-1,0),(0,0),(0,-1)\} \\
D_4 &= \{(0,1),(0,0),(1,0)\} \\
D &= \{(-1,-1),(-1,1),(0,0),(1,-1),(1,1)\}
\end{aligned} \tag{4.12}
$$

and this is given diagrammatically in Figure 4.12

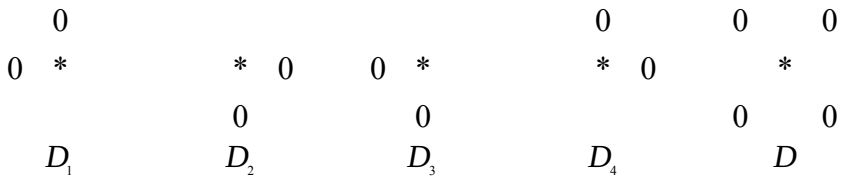


Figure 4.12 Structuring elements in different directions

Suppose dilation and erosion of $f(r,c)$ by the flat top structuring element whose domain is a is denoted by $dilation_a(r,c)$ and $erosion_a(r,c)$ respectively. Then the improved dilation residue operator is defined as

$$
G_d'(r,c) = \min\left\{dilation_{D_{rod1}}(r,c) - f(r,c),\ dilation_D(r,c) - f(r,c),\ G_d''(r,c)\right\} \tag{4.13}
$$

where $G_d''(r,c)$ is defined as

$$
G_d''(r,c) = \max\left\{\left|dilation_{D_1}(r,c) - dilation_{D_2}(r,c)\right|, \left|dilation_{D_3}(r,c) - dilation_{D_4}(r,c)\right|\right\} \tag{4.14}
$$

and the improved erosion residue operator is defined as

$$
G_e'(r,c) = \min\left\{f(r,c) - erosion_{D_{rod1}}(r,c),\ f(r,c) - erosion_D(r,c),\ G_e''(r,c)\right\} \tag{4.15}
$$

where $G_e^{"}(r,c)$ is defined as

$$G_e^{"}(r,c) = \max\left\{\left|erosion_{D_1}(r,c) - erosion_{D_2}(r,c)\right|, \left|erosion_{D_3}(r,c) - erosion_{D_4}(r,c)\right|\right\} \tag{4.16}$$

If we $G_d^{"}(r,c)$ and $G_e^{"}(r,c)$ is combined using pixel wise summation, a noise insensitive and no position biased edge detector will be

$$G_{sum}^{'}(r,c) = G_d^{'}(r,c) + G_e^{'}(r,c) \tag{4.17}$$

4.4 Blur and Minimum Operator

The blur minimum operator (BMO) or (MBMR) [10], as an improved morphological edge detector, blurs the image first in order to remove noise and detect ideal step edge, and takes the minimum difference from the residue of dilation and erosion of the blurred image. It is defined by

$$BMO = \min\left\{I_1 - erosion(I_1), dilation(I_1) - I_1\right\} \tag{4.18}$$

where I_1 is the blurred image of input image

The same size neighborhood is used for both dilation/erosion for simple calculation. For example, for a 3×3 window, the blurring operation can be obtained as using a mean filter with mask *h* as follows:

$$F_b = F * h \qquad h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

where *F* is the input image, F_b is the blurred image and *h* is the kernel of convolution.

Then blurred image will be F_b given in Equation (4.19)

$$F_b = \left[\begin{array}{l} f(x-1,y) + f(x+1,y) + f(x,y) + f(x-1,y-1) + f(x,y-1) + \\ f(x+1,y-1) + f(x-1,y+1) + f(x,y+1) + f(x+1,y+1) \end{array} \right] \quad (4.19)$$

Figure 4.13 Blurred Lena image

Resulting blurring operation on Lenna's image is shown in Figure 4.13

After obtaining the blurred image, BMO is applied to Lenna's image and obtained edge map is given in Figure 4.14

Figure 4.14 Blur minimum edge detector

The BMO is noise insensitive [10]. Due to blurring effect, it helps to reduce the Gaussian noise. Also the choice of minimum operator will suppress impulsive noise from a process window. The BMO thus is less sensitive to noise than dilation or erosion residue operators and basic morphological gradients.

The BMO produces a result which has nonzero edge strength on both the edge pixels for the ideal step edge due to the blurring effect. However, due to the effect of smoothing, the edge strength is weaker than minimum operator without using blurring operation. By thresholding the edge strength of BMO with a suitable threshold value, ideal ramp edges can be extracted.

4.5 Alpha Trimmed Morphologic Edge Detector

In alpha trimmed morphological edge detector proposed by [15] blurs the input image instead of mean filter as in blur minimum edge detector, using alpha trimmed mean filter and then calculates the edge strength using equation (4.18).

$$ATM = \min\left\{I_1 - erosion\left(I_1\right), dilation\left(I_1\right) - I_1\right\} \tag{4.20}$$

where I_1 is filtered using alpha trimmed mean filter.

4.5.1 Alpha trimmed mean filters

Alpha trimmed mean filter deletes the $d/2$ lowest and the $d/2$ highest gray-level values of corrupted image $g(x,y)$ in the area defined by S_{xy} . Let $g_r(s,t)$ represent the remaining $mn-d$ pixels. A filter formed by averaging these remaining pixels is called an alpha trimmed mean filter:

$$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(s,t)\in S_{xy}} g_r(s,t) \tag{4.21}$$

where the value of d can range from 0 to $mn-1$. When $d=0$, the alpha-trimmed filter reduces to the arithmetic mean filter and when $d=(mn-1)/2$, the filter becomes a median filter. For other values of d, the alpha trimmed filter is useful in situations involving multiple types of noise, such as a combination of salt-and-pepper and Gaussian noise.

Figure 4.15 Noisy image and filtered using alpha trimmed filter

Figure 4.16 ATM edge map

CHAPTER 5

DEVELOPMENT OF AN EDGE DETECTION ALGORITHM USING ALTERNATING SEQUENTIAL FILTERS

Edge detection plays a very important role in medical imaging. Pre-operative diagnosis of diseases such as cancer, tumors, bone fractures etc. depend on edge detection techniques to identify the malignant growths. In orthopedics, edge detection techniques are used in the reconstruction of the skeletal system from modalities such as MRI (Magnetic Resonance Imaging) and CT (Computed Tomography).

Medical images acquired from MR or CT scanners are generally low-contrast images. Before application of an edge detection method to a medical image, it must be preprocessed in order to increase contrast of it. Since all of the details in image must appear to extract edge points. To obtain a well contrast image some contrast stretching transformations must be applied to medical image.

Morphological filtering [3] is nonlinear image processing technique used widely in image processing. If the operation in mathematical morphology is composed of erosions and dilations then equal number of erosions and dilations constitutes a filter.

5.1 Alternating Sequential Filters

Alternating sequential filters (ASFs) [9,10] in morphology are a combination of iterative morphological filters with increasing size of structuring elements, which are composed of openings and closings.

Let X denotes a binary image and B a binary structuring element. The alternating filters are defined as

$$
\begin{aligned}
AF_B(X) &= (X \circ B) \bullet B \\
AF_B(X) &= (X \bullet B) \circ B \\
AF_B(X) &= ((X \circ B) \bullet B) \circ B \\
AF_B(X) &= ((X \bullet B) \circ B) \bullet B
\end{aligned}
\tag{5.1}
$$

where $\circ$ and $\bullet$ shows openings and closings respectively. Then alternating sequential filter (ASF) is an iterative application of $AF_B(X)$ with increasing size of structuring elements.

$$
ASF(X) = AF_{B_N} AF_{B_{N-1}} .....AF_{B_1}(X) \tag{5.2}
$$

where N is an integer and $B_N, B_{N-1}, \ldots, B_1$ are structuring elements with decreasing sizes. The B_N is constructed by

$$
B_N = B_{n-1} \oplus B_1 \quad \text{for } N \geq 2 \tag{5.3}
$$

In [9] the alternating sequential filters are applied on salt and pepper noised image (Input PSNR 13.06 db, probability of occurrence of noisy samples is 0.1) given in Figure 4.16. The noise suppression capability of these filters is compared with that of conventional filters. In all morphological operations 2×2 square structuring element is used.

To compare the performance of the morphological filters with that of conventional filters, PSNR (Peak signal to noise ratio) value is used. PSNR is defined by the equation

$$
\begin{aligned}
PSNR &= 20 \log \left\{ \frac{255}{RMSE} \right\} \; dB \\
MSE &= \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\left[f(i,j) - F(i,j) \right]^2}{N.M}
\end{aligned}
\tag{5.4}
$$

Here N and M denote the picture height and width, $f(i,j)$ and $F(i,j)$ are the pixel value at (i,j) of the source image and the reconstructed image respectively. *RMSE* is the root

mean squared error of the *MSE*. There are some other definitions of PSNR but it is not important because we are interested in relative comparison not absolute values.

Figure 5.1 Original and salt and pepper noised image

The results of ASF on noisy image are given in Table 5.1

Table 5.1. Output PSNR of the filters

Ψ	αβ	βα	αβα	βαβ	M	os	af
PSNR	20.7	21.3	21	21.1	17.8	16.1	14.4

The result with the greatest PSNR value is presented in the Figure 5.2.

Figure 5.2 Noisy image and reconstructed image

5.2 Contrast Enhancement in Mathematical Morphology

The notion of contrast enhancement can be defined in terms of the relative difference in intensity between an image structure and its background. The principal objective of contrast enhancement or sharpening is to emphasize fine details in a medical image.

The principle of contrast enhancement was introduced by Soille [3]. Contrast enhancement is done using top-hat and bottom-hat transformations.

The top-hat transformation in mathematical morphology is defined as the difference between the original image and its grayscale opening given by equation (5.5)

$$\gamma_{TH} = I_0 - \gamma_B \tag{5.5}$$

and bottom hat transformation is defined as the difference between the grayscale closing and the original image given by equation (5.6)

$$\phi_{TH} = \phi_B - I_0 \tag{5.6}$$

Then contrast enhanced image will be the summation of these transformations and the original image.

$$K = I_0 + \gamma_{TH} - \phi_{TH} \tag{5.7}$$

The top-hat transformation yields an image that contains all the residual features (peaks and ridges) removed by opening. Adding these residual filters features to the original image has the effect of high-intensity structures. The bottom hat transformation yields an image that is low intensity structures.

5.3 Proposed Method for Edge Detection in Medical Images

The flow diagram given in Figure 5.3 is proposed to detect edges in medical images. Firstly input image is gone through a transformation called top hat and bottom hat transformations. It is used for enhancement of the contrast of the input image. And then contrast enhanced image is filtered with alternating sequential filter given by 5.8 in order

to remove noise. Then simple erosion residue edge (4.26) detector is applied to the filtered image to extract edge map. After obtaining edge map a threshold operation is applied to edge map to get edge points. Then, edge image is thinned until all of the edges become one pixel thickness.

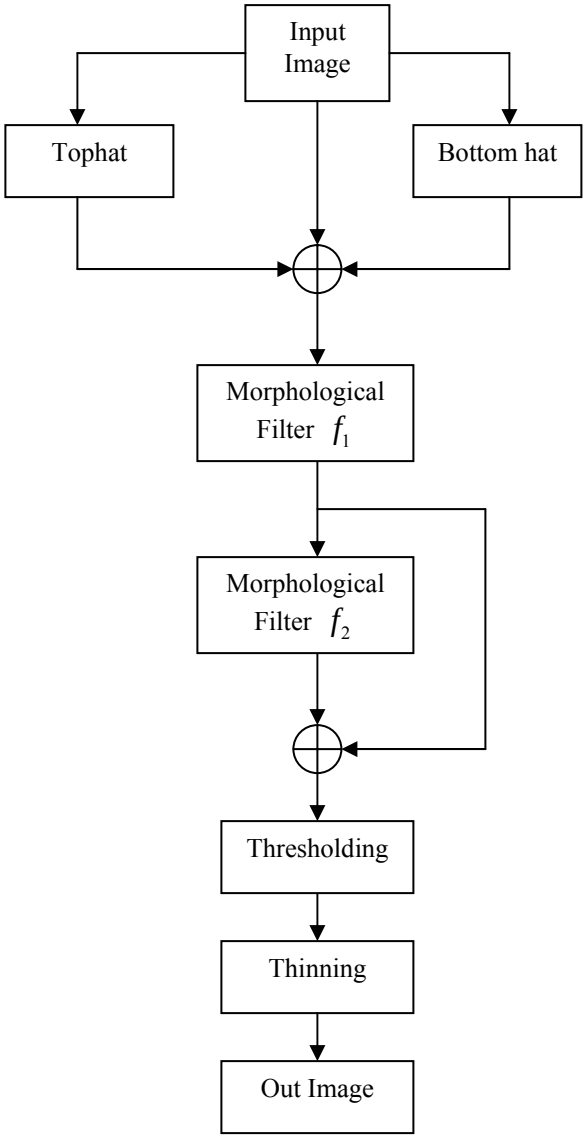


Figure 5.3 Flow diagram of new algorithm for edge detection

The flow diagram of the new edge detection method is given in Figure 5.3. The morphological filters f_1 used in the edge detector of Figure. 5.3 can be alternating sequential filters. Morphological filters f_1 and f_2 are given in equation (5.8)

$$\begin{aligned} f_1 &= \gamma_1\phi_1 \\ f_2 &= f_1 \Theta B \end{aligned} \tag{5.8}$$

where algebraic openings γ_1 are defined as:

$$\gamma_1 = Max(f_{(\alpha_1,l)}, f_{(\alpha_2,l)}, ... f_{(\alpha_n,l)}) \tag{5.9}$$

and the algebraic closings ϕ_1 are defined as:

$$\phi_1 = Min(f^{(\alpha_1,l)}, f^{(\alpha_2,l)}, ..., f^{(\alpha_n,l)}) \tag{5.10}$$

where (α, l) is the linear structuring element of length *l* and whose angle with the *x*-axis is α. $\alpha_1, ..., \alpha_n$ are the different directions chosen. Figure 5.4 gives different directions structuring elements of length 3 pixels.

0	0	0	0	0	1	0	1	0	1	0	0
1	1	1	0	1	0	0	1	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	1
$\alpha_1 = 0^\circ$			$\alpha_2 = 45^\circ$			$\alpha_3 = 90^\circ$			$\alpha_4 = 135^\circ$		

Figure 5.4 Different directions structuring elements of length 3 pixels.

Figure 5.5 Noisy image and filtered image using alternating sequential filter

Figure 5.6 ASF edge map

5.4 Thresholding

Thresholding is a way to get rid of the effect of noise and to improve the signal-noise ratio. That is, it is a way to keep the significant information of the image while get rid of the unimportant part. Using the general theory of thresholding, the grayscale image can be converted to a binary image by changing all pixels below a predetermined value, the threshold value, to zero while changing the remaining pixels to one. Mathematically for a function $f(x, y)$, the value of the new function $F(x, y)$ using the threshold value of T is

$$F(x,y)=\begin{Bmatrix}1 & \text{if } f(x,y)\geq T\\ 0 & \text{if } f(x,y)<T\end{Bmatrix} \tag{5.11}$$

The Otsu thresholding method allows for the automatic selection of this threshold value [14]. Clustering analysis is applied to the gray level data for the image. Two clusters are formed from the Gaussian distribution of the gray level pixels. For these clusters, one cluster represents the background and the other the foreground. The objective of the Otsu method is to achieve optimal threshold value by minimizing the weighted sum for the within-class variance for the two clusters. Application of Otsu's method is given in Figure 5.7

Figure 5.7 Original Lena image and thresholded image

CHAPTER 6

COMPARATIVE ANALYSIS

A number of researchers have considered the problem of measuring the edge detector performance. In fact, it is not a simple problem since it is not known what the underlying features are to be detected. However, it is assumed that they are step edges corrupted by Gaussian noise or impulse noise, and then some criteria can be set for evaluating performance. Such criteria are usually as follows:

- The probability of false edges;
- The probability of missing edges;
- The error in estimating the edge angle;
- The mean square distance of the edge estimate from the true edge;
- The algorithm's tolerance to distorted edges and other features such as corners and junctions.

The first two criteria relate to edge detection, the second two to edge localization, and the last to tolerance to departures from the ideal edge model. Pratt [1] introduced a *figure of merit* function FOM for measuring quantitatively the performance of various edge detectors. His measure is

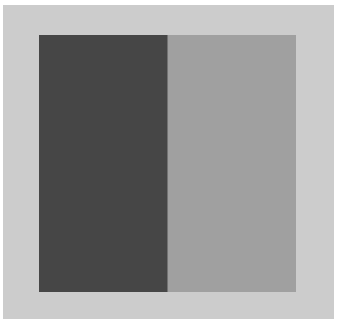
$$FOM = \frac{1}{\max(I_A, I_I)} \sum_{i=1}^{I_A} \frac{1}{1 + d_i \alpha^2} \tag{6.1}$$

where I_A , I_I, d, and α are respectively the detected edges, the ideal edges, the distance between the actual and the ideal edges, and a design constant to penalize displaced

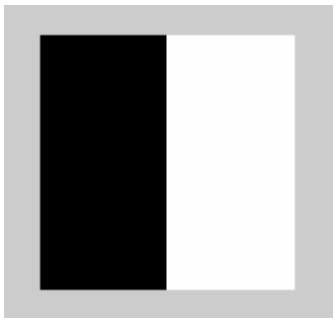
edges. The larger value of FOM corresponds to better performance, with 1 being a perfectresult.

6.1 Performance on Step Edge

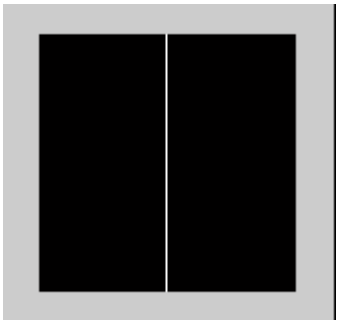
In the first experiment ideal step edge images are used since they allow objective performance measures. Two 128×128 synthetic images are composed of single vertical edge. First one is in binary form and the second is in gray scale form shown in Figure 6.1. The gray-scale value is chosen as 70 for left side and 160 for the right side of the image.



b) Gray-scale ideal step image



a) Binary ideal step image



c) Ideal edge image

Figure 6.1 Binary and gray level step images

Salt and pepper noise with different noise densities and Gaussian noise with zero mean and different variances are added to these two ideal step edges, respectively. Laplacian of Gaussian (Log), Sobel, Prewitt, Roberts and morphological edge detectors are applied to them. Figures 6.2 through 6.6 show the results of applying Log, Sobel, Prewitt, Roberts and morphological algorithms to ideal binary step edge under salt and pepper noise and Gaussian noise with different densities, respectively.

Tables 6.1and 6.2 list the results of these edge operators on binary and grayscale images corrupted with salt and pepper noise. The performance of ASF edge detector and the other edge morphologic edge detectors are approximately the same when noise is small; however, when noise is increased the performance of ASF edge detector algorithm is better than others. For instance when noise density is increased to 15 percent performance of ASF edge detector is 46 percent in binary case however MBMR and ATM edge detectors 16 percent and 11 percent respectively, for the same density. So in binary image corrupted with salt and pepper noise, the performance of ASF edge detector is superior to other conventional and morphologic edge detectors. This is because ASF edge detector suppresses noise using alternating sequential filter before detecting edges. Also Figure 6.2 shows the graphical performance of the three algorithms ATM, MBMR and ASF. It can be inferred from the graph that ASF edge detection algorithm is more robust to noise.

In grayscale case corrupted with salt and pepper noise, again the algorithm is very good in detecting edges as it is seen on Figure 6.3 an increase in noise density does not affect the performance of detecting edges.

A visual inspection also leaves the impression that the ASF edge detector suppresses more noise than the other conventional and morphologic edge detectors, also it leaves continuous edges. It is easily seen from Figure 6.4 and 6.7

Table 6.1 FOM values for binary image corrupted by salt and pepper noise

Edge Detector	NOISE DISTRUBUTION					
	No noise	SAP0.03	SAP0.06	SAP0.9	SAP0.12	SAP0.15
Log	0.88906	0.15905	0.12345	0.096191	0.091484	0.088771
Sobel	0.88594	0.15223	0.12147	0.28522	0.21919	0.1952
Prewitt	0.88594	0.15791	0.12136	0.26808	0.2176	0.19272
Roberts	0.98438	0.25374	0.18748	0.46669	0.36515	0.30786
MR	0.95	0.45281	0.29444	0.20105	0.16594	0.13881
MBMR	0.95	0.91883	0.36575	0.23669	0.18068	0.16505
ATM	0.99219	0.98281	0.81127	0.47498	0.35831	0.11141
ASF	0.9	0.89751	0.79937	0.64402	0.5508	0.46474

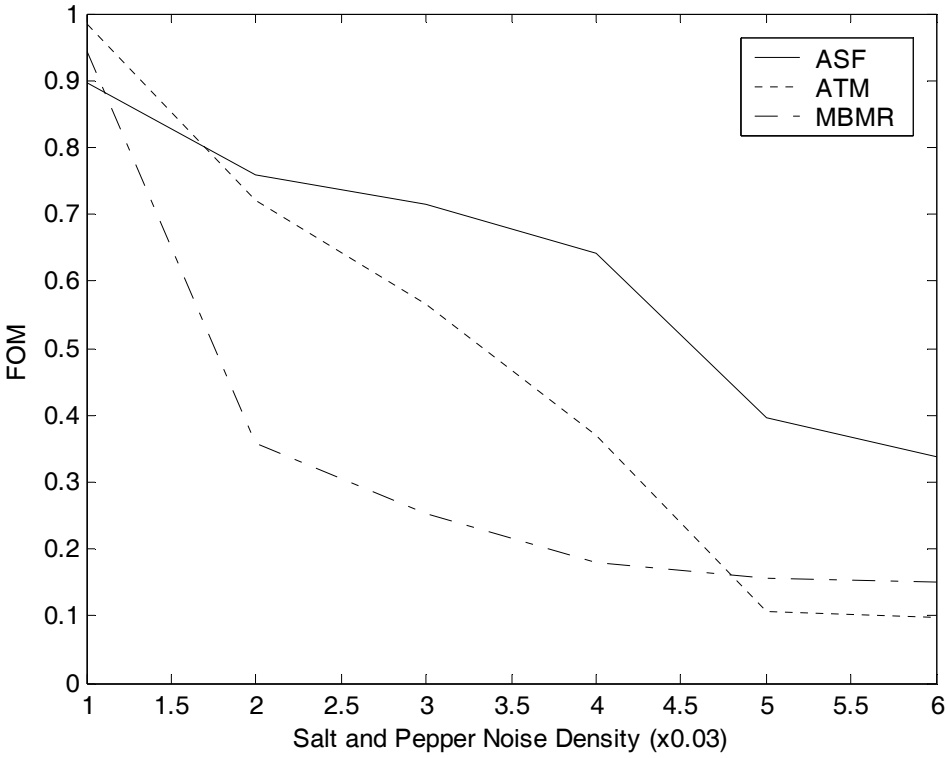


Figure 6.2 Performance on binary step image

Table 6.2 FOM values for gray level image corrupted by salt and pepper noise

Edge Detector	NOISE DISTRUBUTION					
	No noise	SAP0.03	SAP0.06	SAP0.09	SAP0.12	SAP0.15
Log	0.33937	0.149	0.10867	0.099425	0.084162	0.087672
Sobel	0.88594	0.17836	0.14986	0.13166	0.096992	0.15971
Prewitt	0.88594	0.15237	0.15061	0.12937	0.11167	0.16768
Roberts	0.98438	0.28124	0.081735	0.11774	0.073254	0.092549
MR	0.95	0.33551	0.20261	0.14557	0.12332	0.10947
MBMR	0.87308	0.65086	0.26488	0.1735	0.14815	0.13842
ATM	0.97656	0.81436	0.24598	0.13286	0.081166	0.089594
ASF	0.89297	0.86702	0.84105	0.84345	0.81033	0.61165

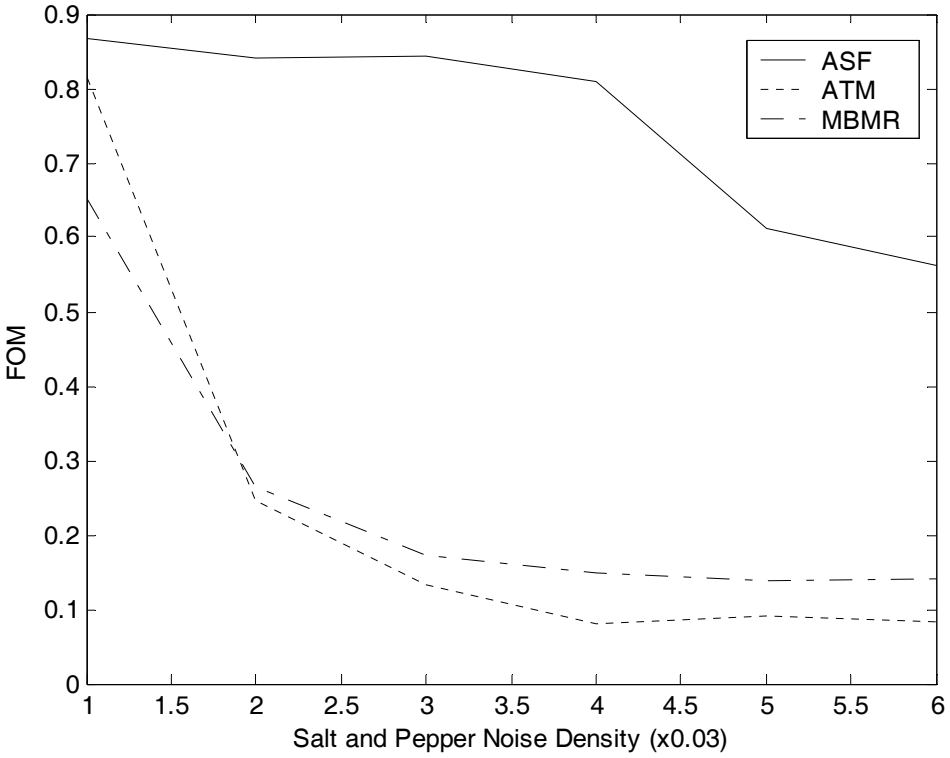


Figure 6.3 Performance on gray level step image corrupted by salt and pepper noise

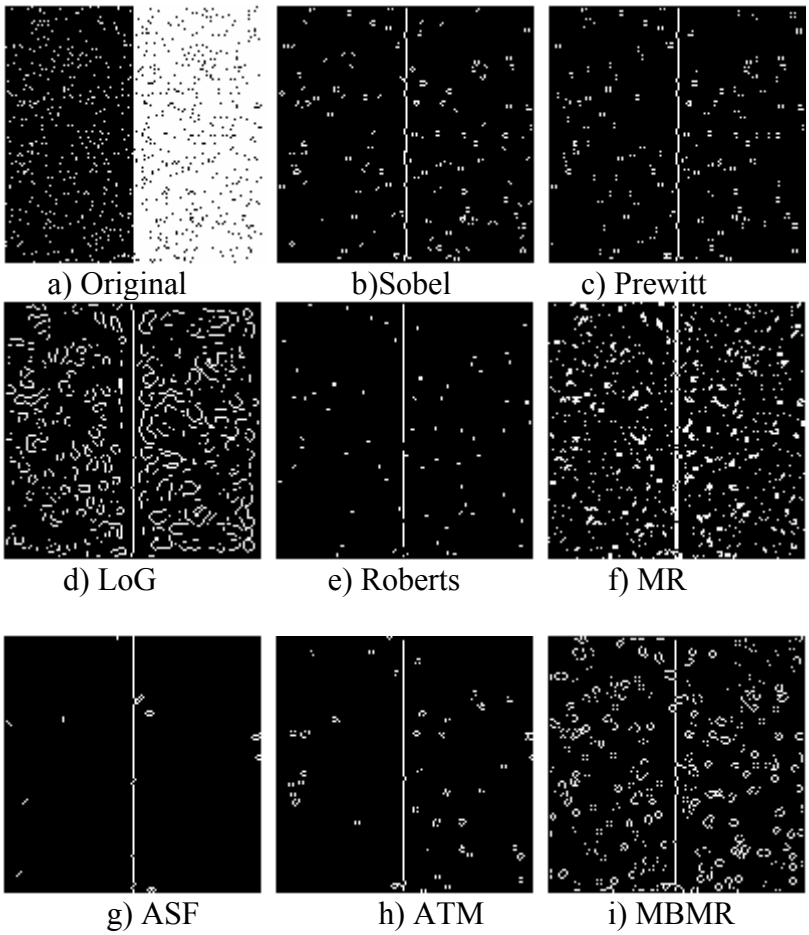


Figure 6.4 Results of Edge operators on binary image corrupted with 10 percent salt and pepper noise

Table 6.3 and 6.4 lists the results of same edge operators to binary and gray scale images corrupted with zero mean Gaussian noise. From Figure 6.5, it is seen that when Gaussian noise is present in a gray scale image, performance of new algorithm using alternating sequential filters is not very good for gray scale case. Even it is worse than the other morphology based algorithms When variance of the Gaussian noise is increased a little bit, performance of the algorithm using alternating sequential filter declines suddenly.

In binary case MBMR and ATM edge detection algorithms are better than the new algorithm. But up to some point they have equal performance on noisy image.

Table 6.3 FOM values for gray level image corrupted by Gaussian noise

Edge Detector	NOISE DISTRUBUTION					
	No noise	G0.03	G0.06	G0.09	G0.12	G0.15
Log	0.33937	0.12535	0.092464	0.090709	0.084169	0.083047
Sobel	0.88594	0.48598	0.36479	0.32871	0.26998	0.24217
Prewitt	0.88594	0.4743	0.37864	0.35176	0.30281	0.29039
Roberts	0.98438	0.63943	0.30494	0.17558	0.08312	0.021116
MR	0.95	0.094259	0.079733	0.078744	0.079161	0.074071
MBMR	0.87308	0.89116	0.78551	0.20222	0.16902	0.15474
ATM	0.97656	0.96785	0.70906	0.093648	0.090434	0.091792
ASF	0.89297	0.82932	0.090091	0.090904	0.10061	0.094085

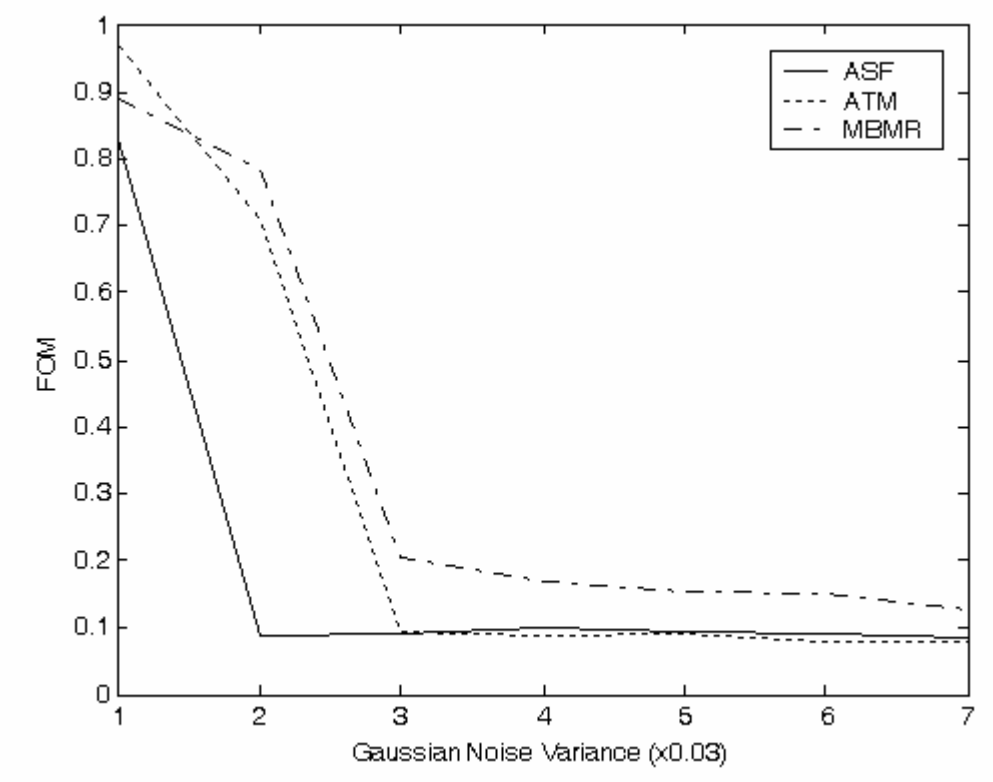


Figure 6.5 Performance on gray level step image corrupted by Gaussian noise

Table 6.4 FOM values for binary image corrupted by Gaussian noise

Edge Detector	NOISE DISTRUBUTION					
	No noise	G0.03	G0.06	G0.09	G0.12	G0.15
Log	0.88906	0.19644	0.12064	0.10024	0.093991	0.085248
Sobel	0.88594	0.71146	0.45655	0.35768	0.33896	0.31521
Prewitt	0.88594	0.75306	0.48082	0.38823	0.34325	0.32892
Roberts	0.98438	0.78244	0.64804	0.58811	0.56025	0.42382
MR	0.95	0.13718	0.10366	0.095865	0.093684	0.091042
MBMR	0.95	0.87537	0.88503	0.8896	0.896	0.84606
ATM	0.99219	0.96478	0.9375	0.94688	0.93516	0.87514
ASF	0.9	0.88648	0.87674	0.83097	0.66864	0.16008

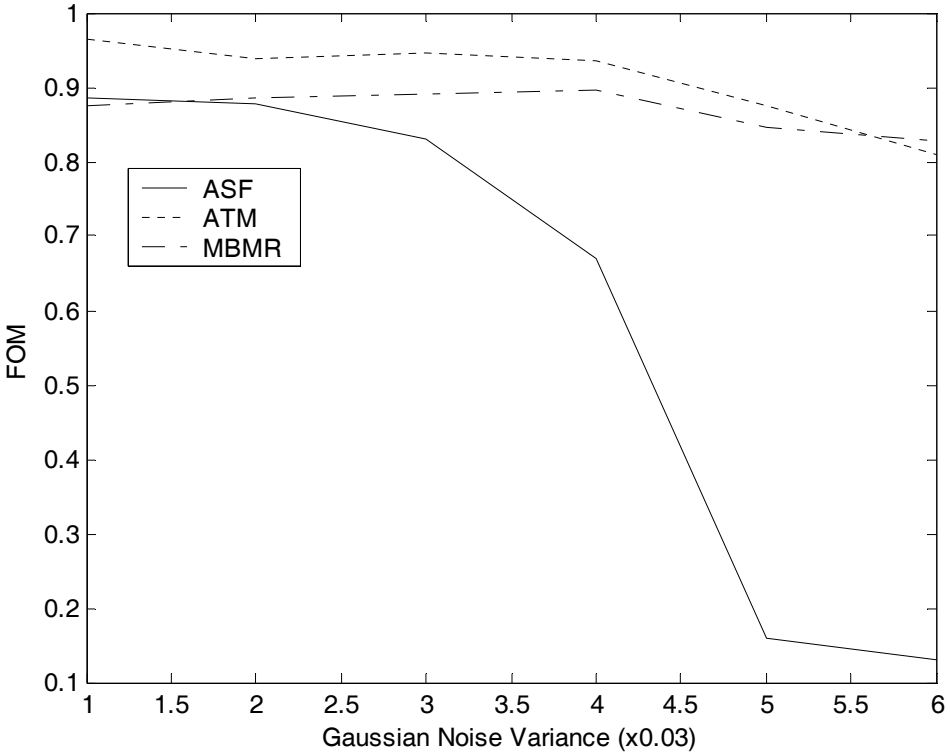


Figure 6.6 Performance on binary step image corrupted by Gaussian noise

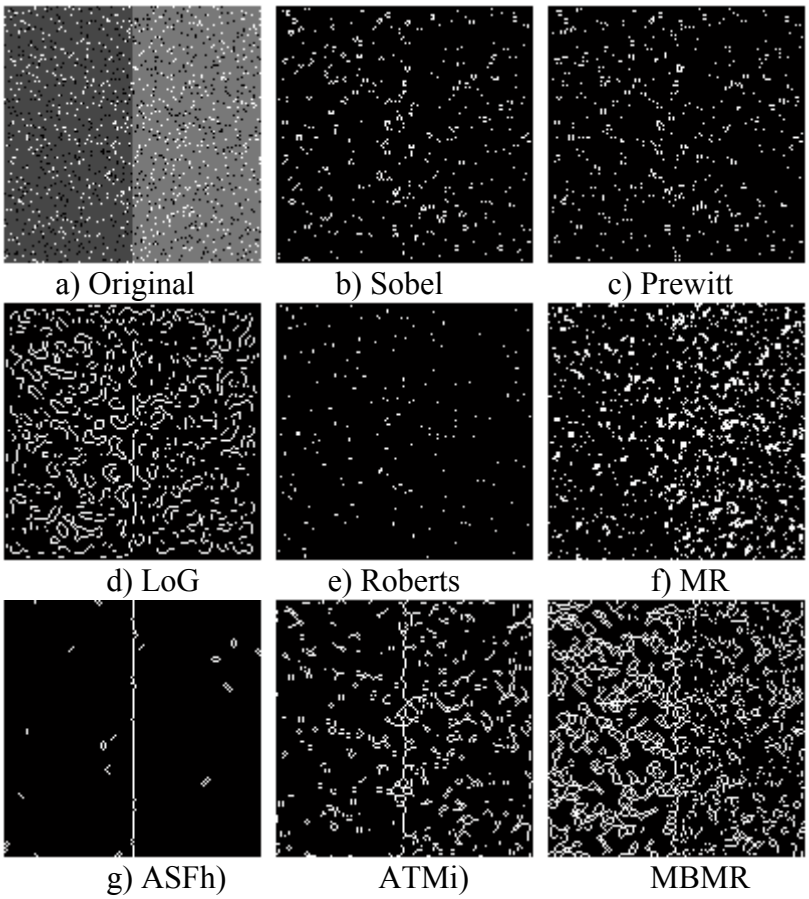


Figure 6.7 Results of edge operators on gray level step image corrupted with 10 percent salt and pepper noise

6.2 Image Formats

Images in medical imaging are stored in many formats. Image formats being dealt with in this thesis are

- DICOM (Digital Imaging and Communications in Medicine)
- ANALYZE FORMAT

6.2.1 Dicom

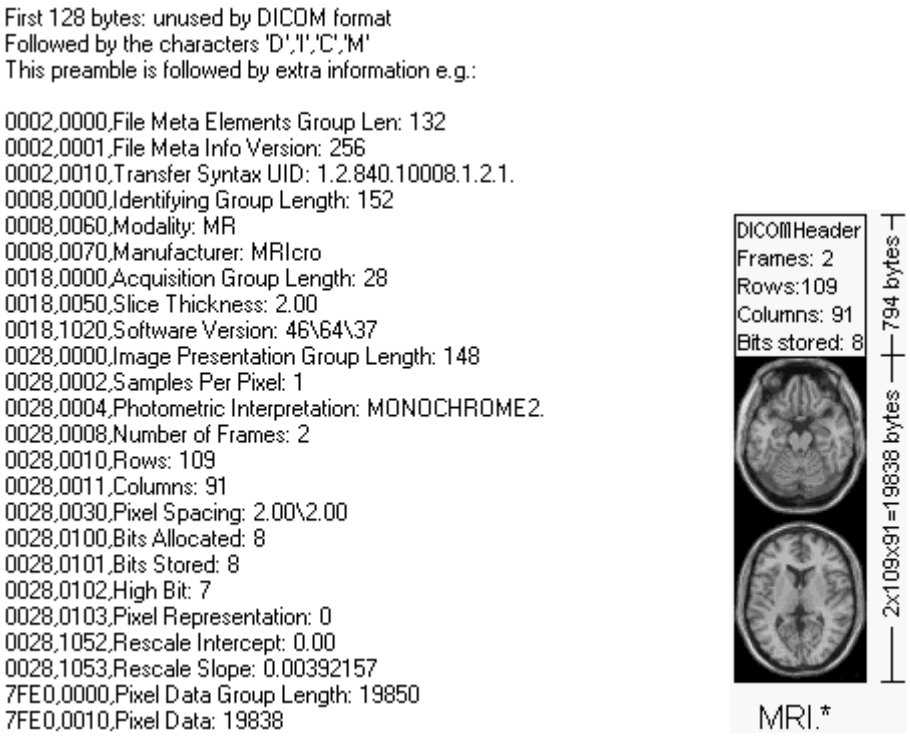
DICOM (Digital Imaging and Communications in Medicine) standard was created by the National Electrical Manufacturers Association (NEMA) to aid the distribution and viewing of medical images, such as CT scans, MRIs, and ultrasound.

A single DICOM file contains both a header (which stores information about the patient's name, the type of scan, image dimensions, etc), as well as all of the image data (which can contain information in 3Ds). This is different from the popular Analyze format, which stores the image data in one file (*.img) and the header data in another file (*.hdr). DICOM image data can be compressed (encapsulated) to reduce the image size. Files can be compressed using lossy or lossless variants of the JPEG format, as well as a lossless Run-Length Encoding format (which is identical to the packed-bits compression found in some Tiff format images).

DICOM is the most common standard for receiving scans from a hospital. Figure 6.8 b) shows a typical DICOM file with the DICOM image. In this example, the first 794 bytes are used for a DICOM format, which describes the image dimensions and retains other text information about the scan. The size of this header varies depending on how much header information is stored. Here, the header defines an image which has the dimensions 109x91x2 voxels (three dimensional pixels), with a data resolution of 1 byte per voxel (so the total image size will be 19838). The image data follows the header information (the header and the image data are stored in the same file).

In Figure 6.8 a), a more detailed list of the DICOM header as displayed. Note that DICOM requires a 128-byte preamble (these 128 bytes are usually all set to zero),

followed by the letters ‘D’, ‘I’, ‘C’, ‘M’. This is followed by the header information, which is organized in ‘groups’. For example, the group 0002hex is the file meta information group, and (Figure 6.8 a**)** contains 3 elements: one defines the group length, another stores the file version and a third stores the transfer syntax. For example, this image modality is ‘MR’ (see element 0008, 0060), so it should have elements to describe the MRI echo time. The absence of this information in this image is a violation of the DICOM standard. In practice, most DICOM format viewers do not check for the presence of most of these elements, extracting only the header information, which describes the image size.



a) b)

Figure 6.8 Example DICOM header

6.2.2 Analyze Format

An Analyze (7.5) format image consists of two files, and image and a header file. If the image is for example named "brain", then the files for that image will be called "brain.img" and "brain.hdr". The *.img file contains the numbers that make up the information in the image. The *.hdr file contains information about the *.img file, such as the volume represented by each number in the image (voxel size) and the number of pixels in the X, Y and Z directions. This header contains fields of text, floating point, integer and other information.

6.3 Application of the New Edge Detection Algorithm to Medical Images

In this section we applied the algorithm to real world clinical images. In first experiment proposed algorithm in Figure 5.3 is applied to several medical images. The dimensions of each image set is 512×512. Some of them are given in appendix A given in Figure 6.9 and 6.14. Format of these images are either DICOM or Analyze format. Then it is compared with that of other conventional and morphological edge detectors. Results are given at appendix in Figures 6.10-6.14 and 6.16-6.19. Experimental results were done on a computer having P4 2.8 GHz processor and 512 RAM.

Visual evaluation of results shows that algorithm used for edge detection gives more realistic results than that of other algorithms. First our algorithm extracts more points than others. Second edge points are more continuous and linked smoothly.

In Figure 6.9 we used an MR of an human spine. The reason to choose this image is that the image is noisy and the content of the image is very complex. When we look at the result of all algorithms from Figures 6.10 through 6.13 algorithm is extracting more points than the other algorithms.

In Figure 6.14 we used a CT of a chest in order to extract the edge points of lungs of a human. The detected edges using alternating sequential filters in Figure 6.15 are continuous, well localized and most of the basic edge features are extracted comparing with that of other algorithms. Extracting of edge points efficiently and correctly is very

important in medical imaging, because lung cancer detection or tumor detection is all done using edge detection techniques. Any false edge point may lead to misdiagnosis of the case.

In Figure 6.19 original MR brain is shown and the result of the algorithm is given in Figure 6.20.

CHAPTER 7
CONCLUSION AND FUTURE WORKS

This thesis is focused on comparison of an edge detection algorithm using alternating sequential filter with previously known edge detection algorithms including mathematical morphology based edge detection algorithms. The comparison was made on the extraction of the edge points in different images, one is synthetic and the other is medical image of modalities CT and MR. For synthetic images, a step image in binary and grayscale case were used, then we investigated the performance of the edge detection algorithms on this step edge images under noisy environment. When noise is small, the performances of all algorithms are approximately the same. However, when noise density is increased gradient based algorithms performs very badly. The experimental results show that proposed algorithm using alternating sequential filters for edge detection are better than that of other morphologic edge detectors and conventional edge detectors at some points in synthetic images. In medical images our algorithm also gave satisfactory results.

The experimental results show that each algorithm has advantageous properties as well as some specific drawbacks. First, implementation of an algorithm in mathematical morphology is very easy compared to other algorithms. The reason for simplicity is that any complex operation in mathematical morphology can be done using primitive operations of mathematical morphology. Second, at some noise types, noise removing capability of morphology based approaches is better than that of differential based operators.

We propose some problems related to edge detection using alternating sequential filters which may worth future research works.

1. Computation time for edge detection using morphological operators is longer than that of other differential based operators. This computation is directly related with the neighborhood of the structuring element. When the neighborhood of the structuring element is increased, the computation time is also increased. To overcome this problem, some fast algorithms were developed [25]. Using these algorithms may further reduce the computation time of primitive operations of mathematical morphology which are dilation and erosion.

2. When Gaussian noise is presented in an image, noise removing capability of morphological operators is worse than that of other conventional differential based operators. But this problem can be solved, if we apply a preprocessing operation like Gaussian filtering or other suitable type of filters or non-flat structuring elements with center weighted may further reduce the effect of noise.

3. Selection of structuring element is a key point in mathematical morphology. When using a large structuring element, it is good for noise removing but in this case some details are lost due to the dimension of the structuring element. To overcome this problem morphological reconstruction algorithms [27] can be used to reduce the effect of distortion in the image. Besides this, shape of the structuring element is also important because if the image contains circular objects, it is not good to choose a square structuring element and vice versa.

REFERENCES

[1] Pratt, W. (1991). *Digital Image Processing*, John Wiley & Sons Inc., Second edition.

[2] Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE PAMI, Vol.**8**, No.6, Novermber.*

[3] J. Serra, (1982). *Image Analysis and Mathematical Morphology*, Academic Press.

[4] Sternberg, S. R. (1986). Grayscale morphology. *Comp. Vis. Graph. Im. Proc.* **35**, 333—355.

[5] Nakagawa Y. and Rosenfeld A. (1978). A Note On The Use of Local Min and Max Operators In Digital Image Processing, I*EEE Trans. Syst. Man Cybern.*, ***8***, 632-635.

[6] Pitas I. and Venetsanopoulos. A.N. (1990) *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic Publishers.

[7] Giardina Charles R. and Dougherty Edward R. (1988). *Morphological Methods in Image and Signal Processing*, Prentice-Hall Inc.

[8] Gonzalez, Rafael C. and Woods, Richard E (2002). *Digital Image Processing 2nd*, Prentice-Hall Inc.

[9] Heijmans, H. J. A. M. Morphological Filters, pp. 1-16,

[10] Heijmans, H. J. A. M. (1997). Composing Morphological Filters, *IEEE*, **Vol.6**, No.5, 713-723,

[10] Lee,S.J. Haralick, R.M and Shapiro L. G. (1987). Morphologic edge detection *IEEE J.Robot. Automat.*, **3(2)** 142-155,.

[11] Song X. and Neuvo Y. (1993). Robust edge detector based on morphological filters *Pattern Recognition* **14.**

[12] Kundu, Malay K. Chanda,Bhabatosh and Padmaja, Y. Vani (1998). A multiscale morphologic edge detector, *Pattern Recognition*, **31(10)**: 1469-1478.

[13] Marr D., Hildreth E. C. (1980). *Theory of edge detection.* London, U.K. Proc. R. Soc

[14] Otsu, N. (1979). A thresholding selection method from graylevel histogram, *IEEE Transactions on Systems, Man, and Cybernetics,* **9** ,62–66

[15] Feehs, R. J. and Arce, G. R. (1987). Multidimensional Morphologic Edge Detection *SPIE Vol 845 Visual Communication and Image Processing*, 285-292.

[16] Serra Jean (1982). *Image Analysis and Mathematical Morphology Volume2: Theoretical Advances*. London, Academic Press

[17] Maragos,Petros Schafer, Ronald W., Butt, Muhammad Akmal, (1996). *Mathematical Morphology and Its Applications to Image and signal Processing*. Boston, Dordrecht, London,Kluwer Academic Publishers

[18] Parker J. R., (1997) *Algorithms for ımage processing and computer Vision,* Newyork Wiley Computer Publishing,

[19] Chen T., Wu Q.H., Rahmani-Torkaman R., Hughes J.,(2000). A Pseudo top-hat mathematical morphological approach to edge detection in dark regions, **35**, 199-210, *Pattern Recognition*, *The Journal of the Pattern Recognition Society.*

[20] Zhuang Hanqi, Humano Fumio (1988). A new type of morphologic edge detectors, 304-311, *IEEE*

[21] *Podaru E., Stanomir D.* (2003). Noise Suppression Using Morphological Filters,61- 64

[22] Jang Ben Kwei, Chin Roland T. Analysis of thinning algorithms using mathematical morphology

[23] Maragos P., Schafer R. W., (2003) Morphological Filters Part I: Their Set-Theoretic Analysis and Relations to Linear Shift Invariant Filters, *IEEE Transactions on Acoustics, Speech and signal Procesing*, 1153-1169

[24] Maragos petros, Schafer Ronald W., (2003) Morphological Filters Part II: Their Relations to Median, Order-Statistics, and Stack filters, , *IEEE Transactions on Acoustics, Speech and signal Procesing.* 1170-1184

[25] Junior, Barrera Hirata, JR Roberto, (1997). Fast Algortihms to compute the elementary operators of Mathematical Morphology.,*IEEE*

[26] Vincent Luc, (1993), Morphological Grayscale Reconsruction in Image Analysis: Applications and Efficient Algorithms, 176-201, *IEEE Transactions on ımage Processing, **Vol.2** No2*

APPENDIX A

FİGURES FROM 6.9 THROUGH 6.20

Figure 6.9 Original Spine MR Image

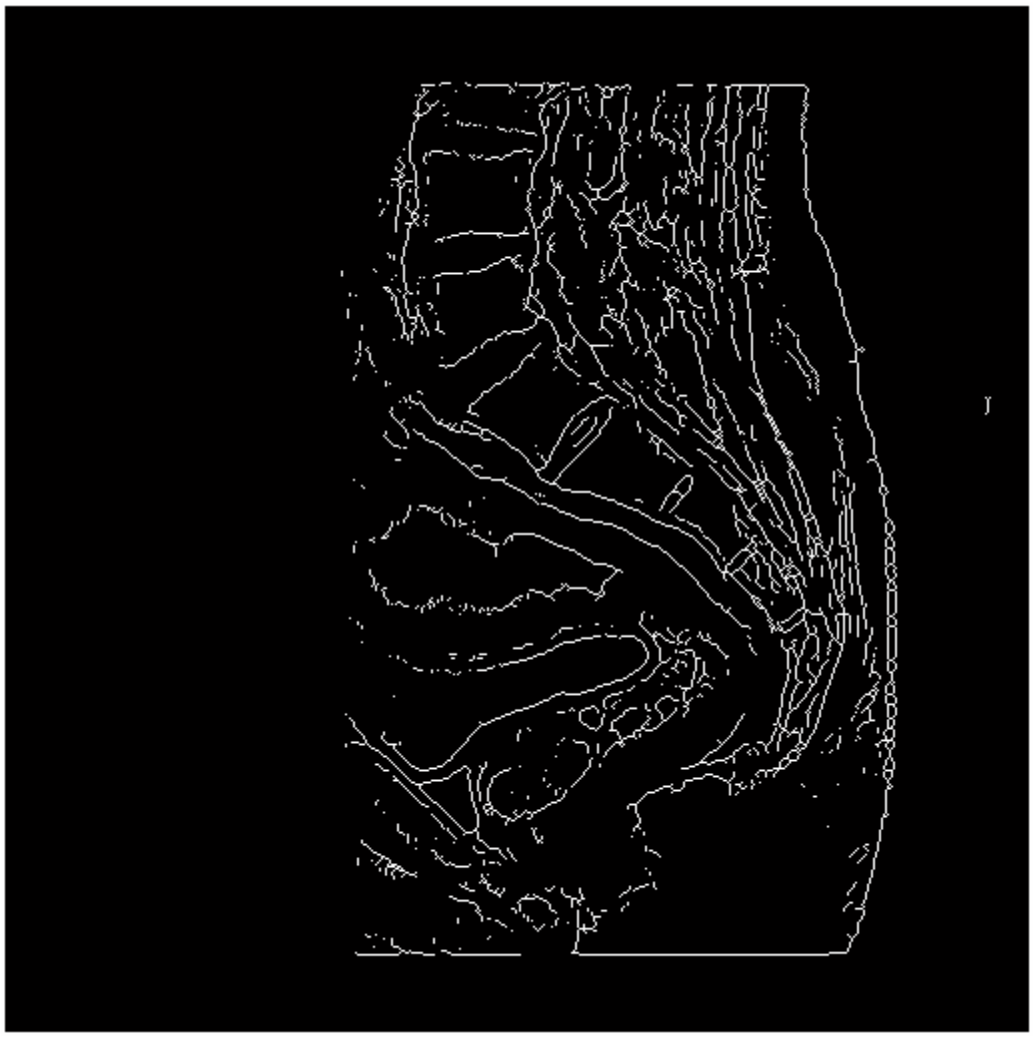


Figure 6.10 New algorithm for edge detection

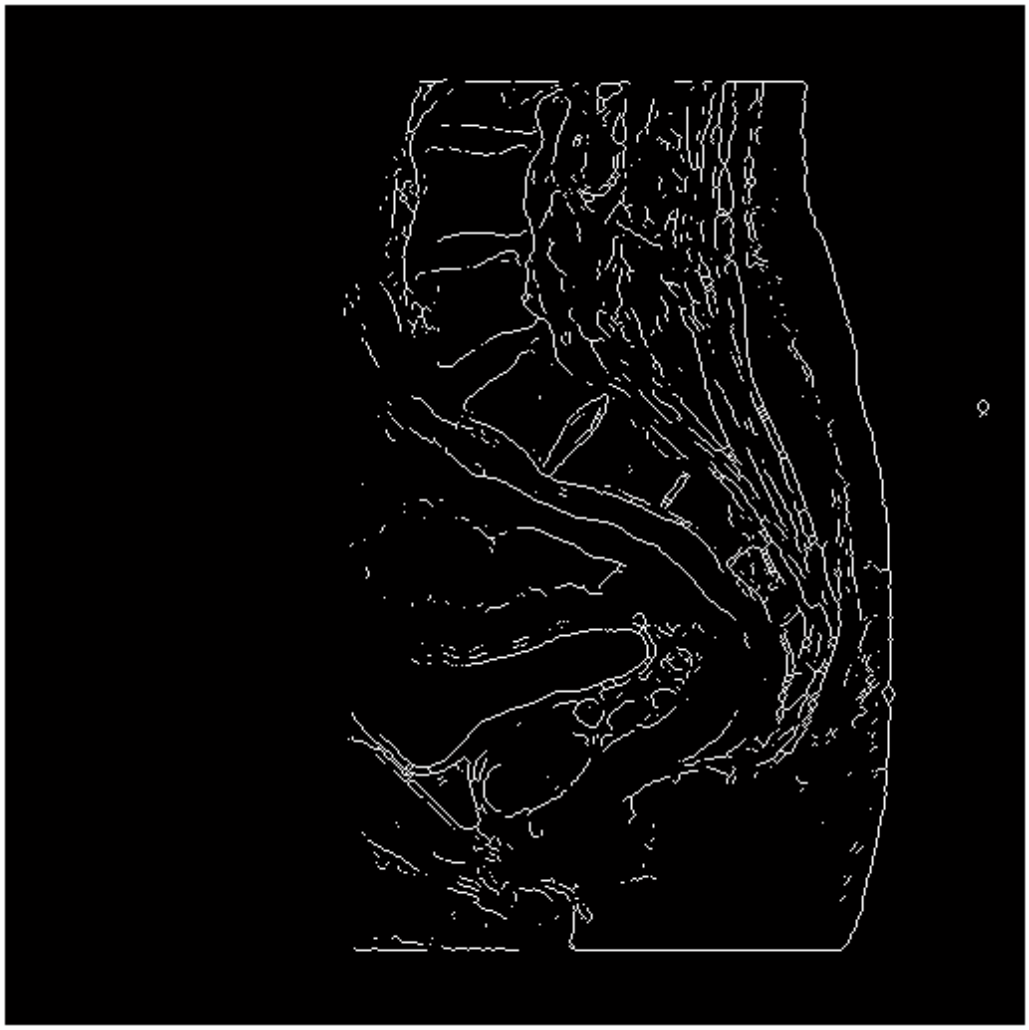


Figure 6.11 ATM edge detector

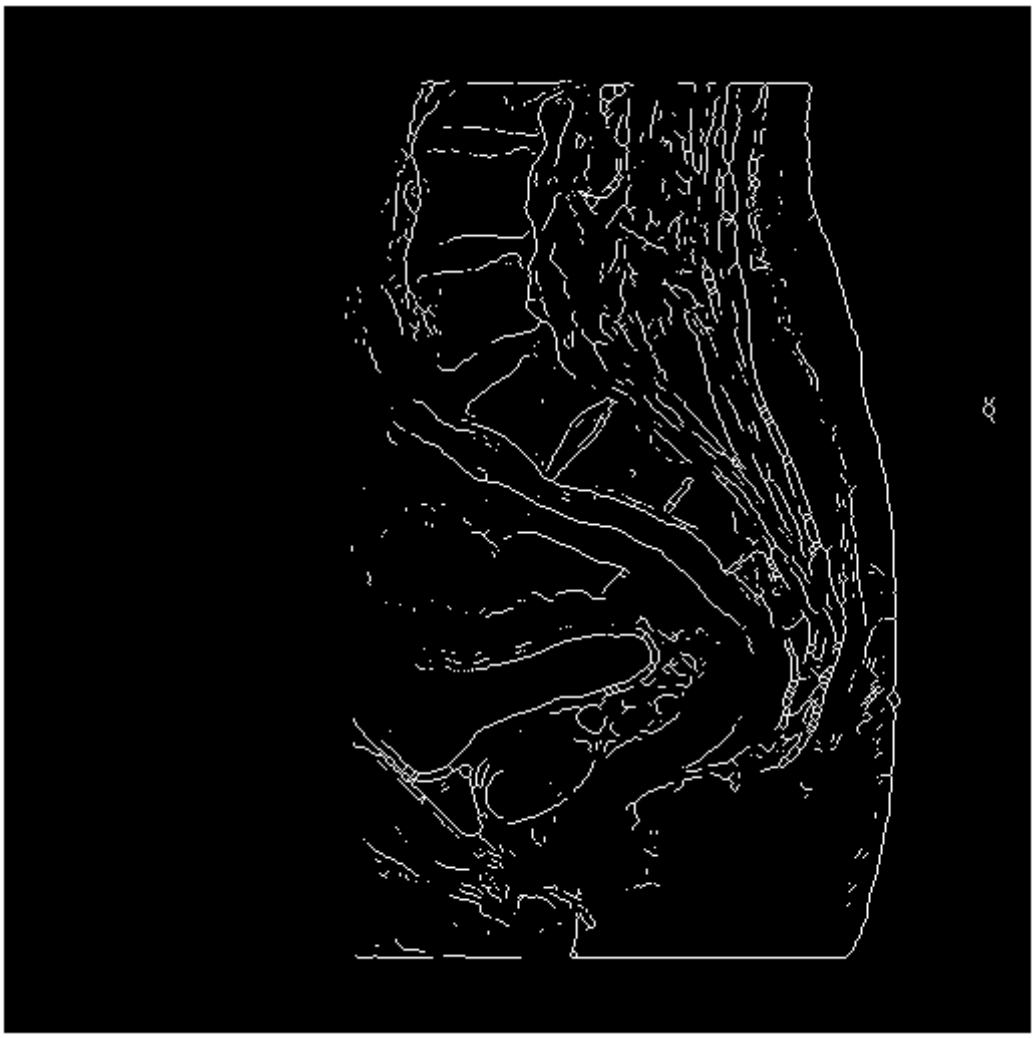


Figure 6.12 BMO operator

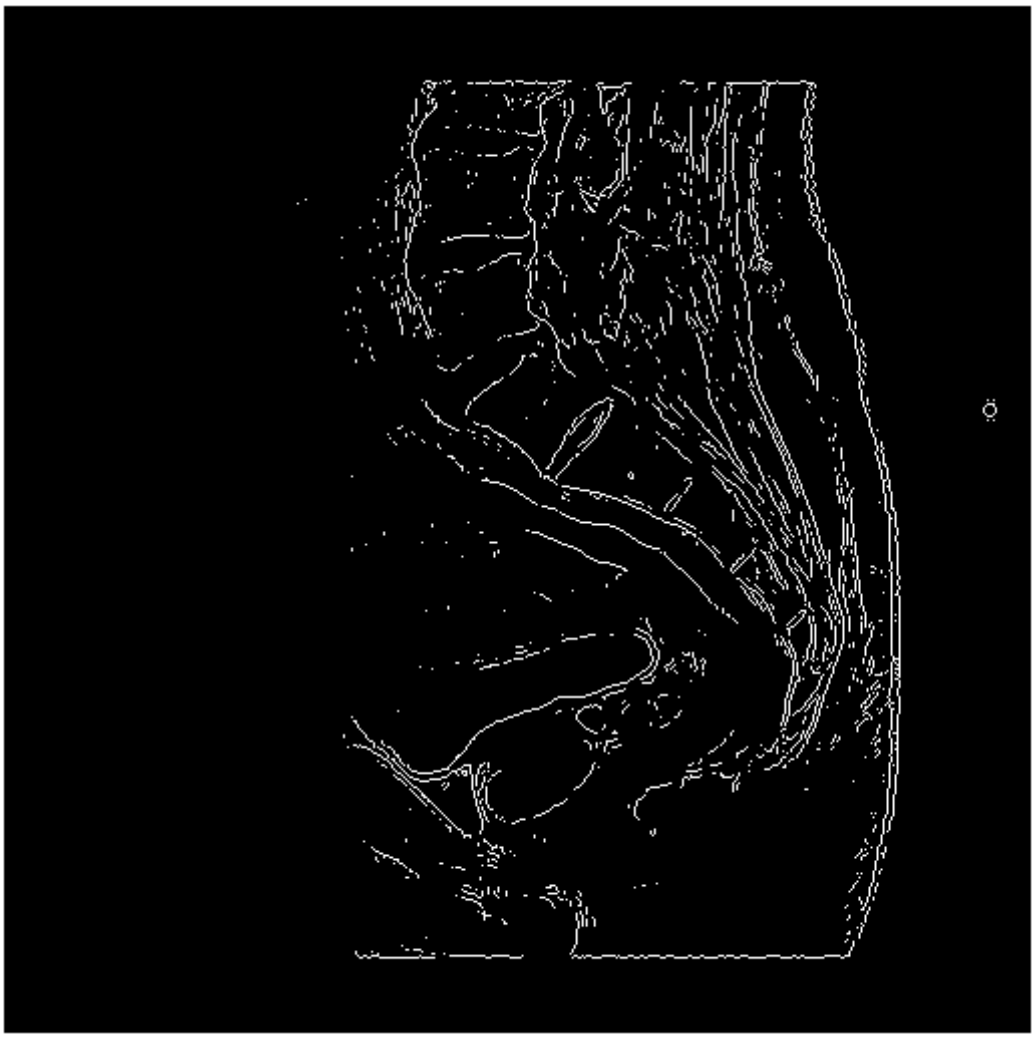


Figure 6.13 Sobel edge detector

Figure 6.14 Original CT of chest

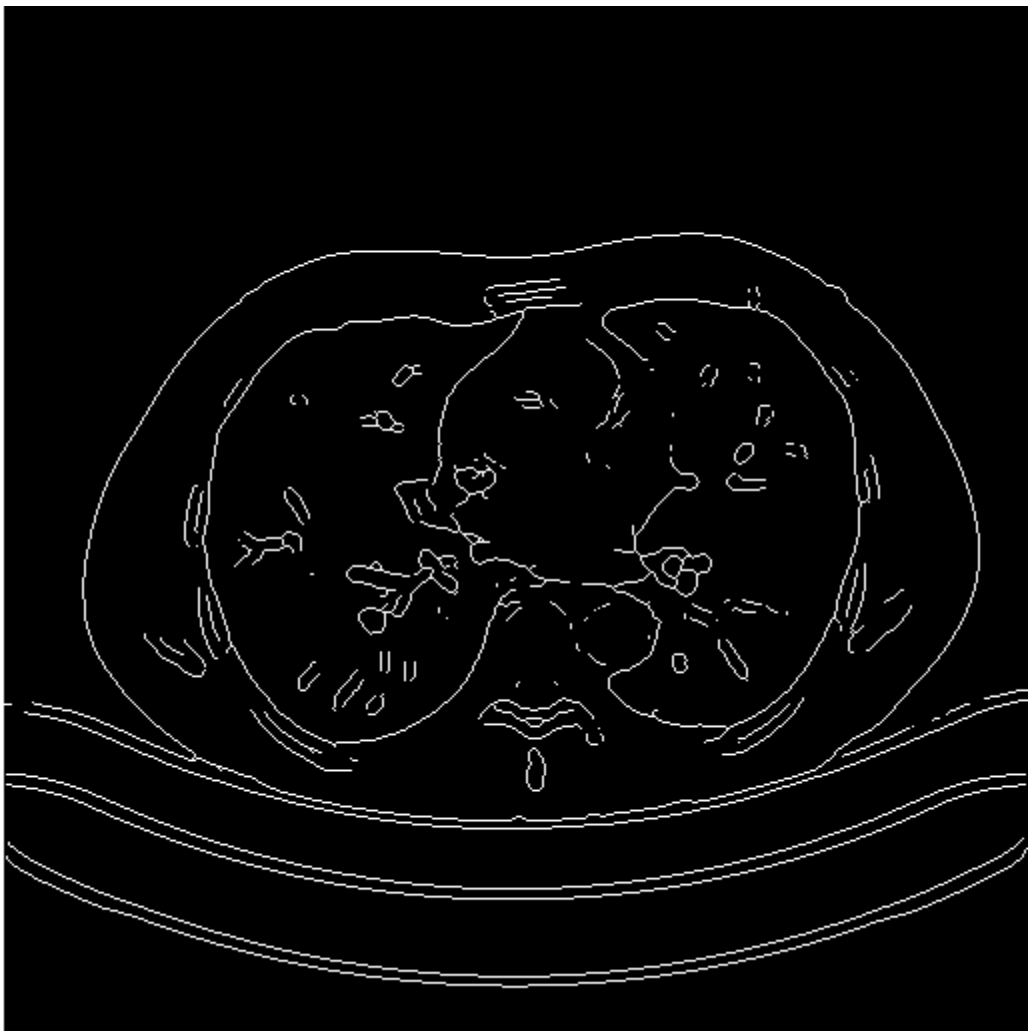


Figure 6.15 New algorithm for edge detection

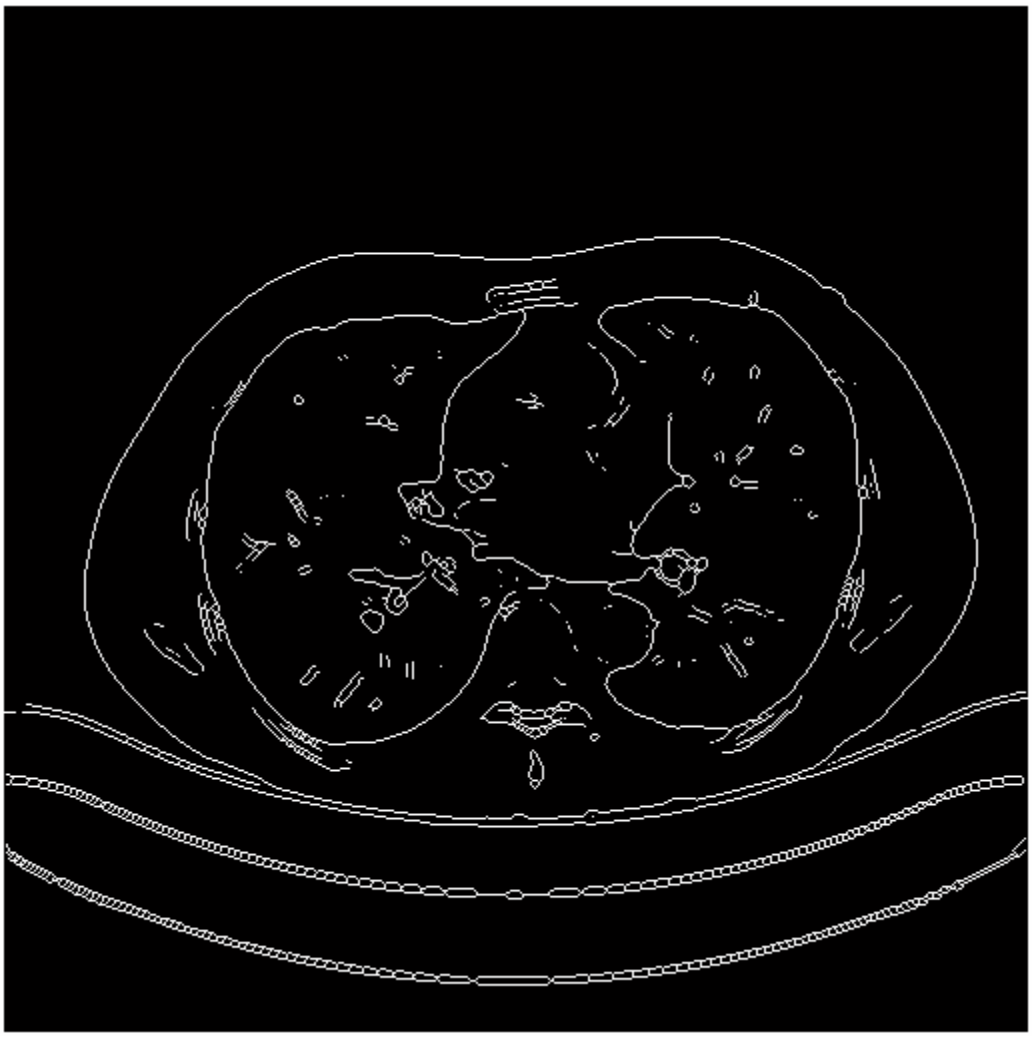


Figure 6.16 ATM edge detector

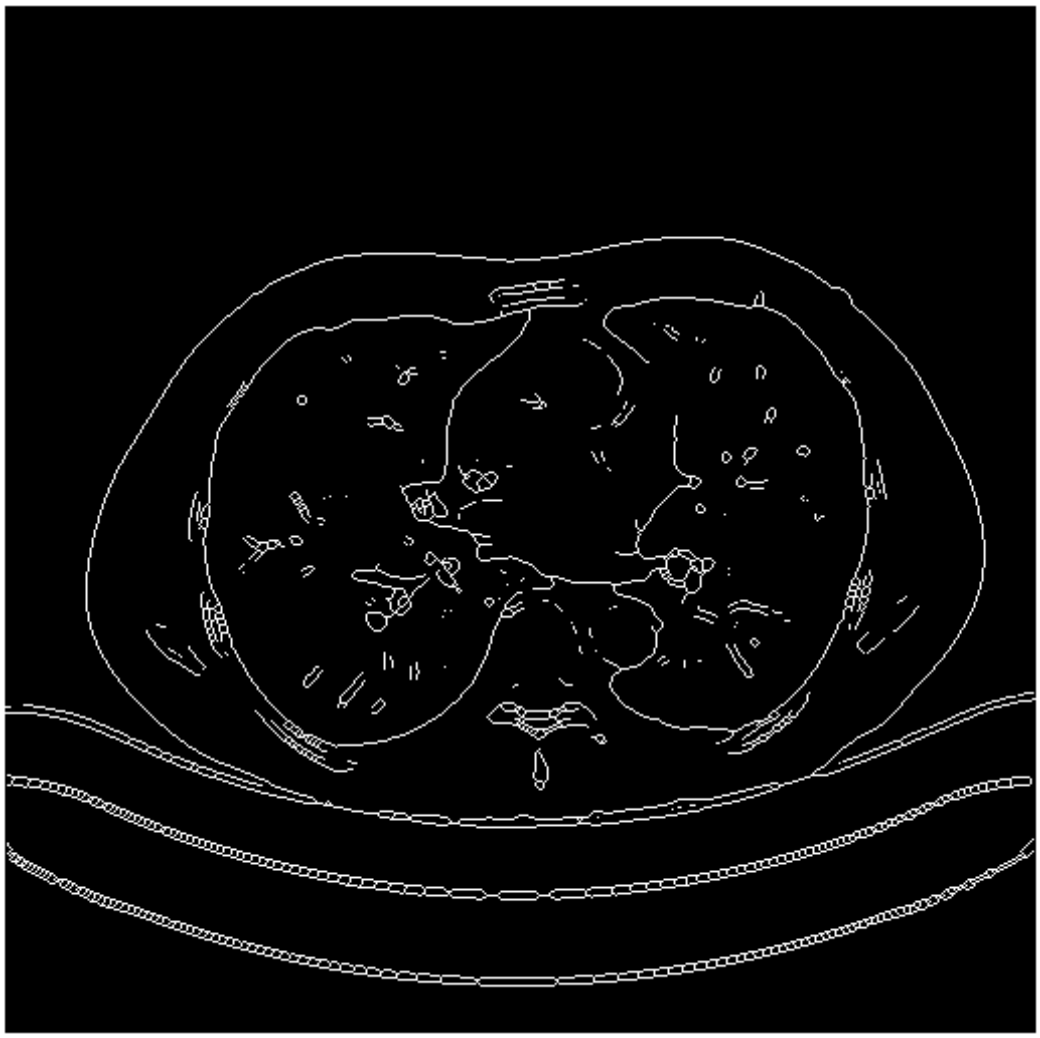


Figure 6.17 BMO operator

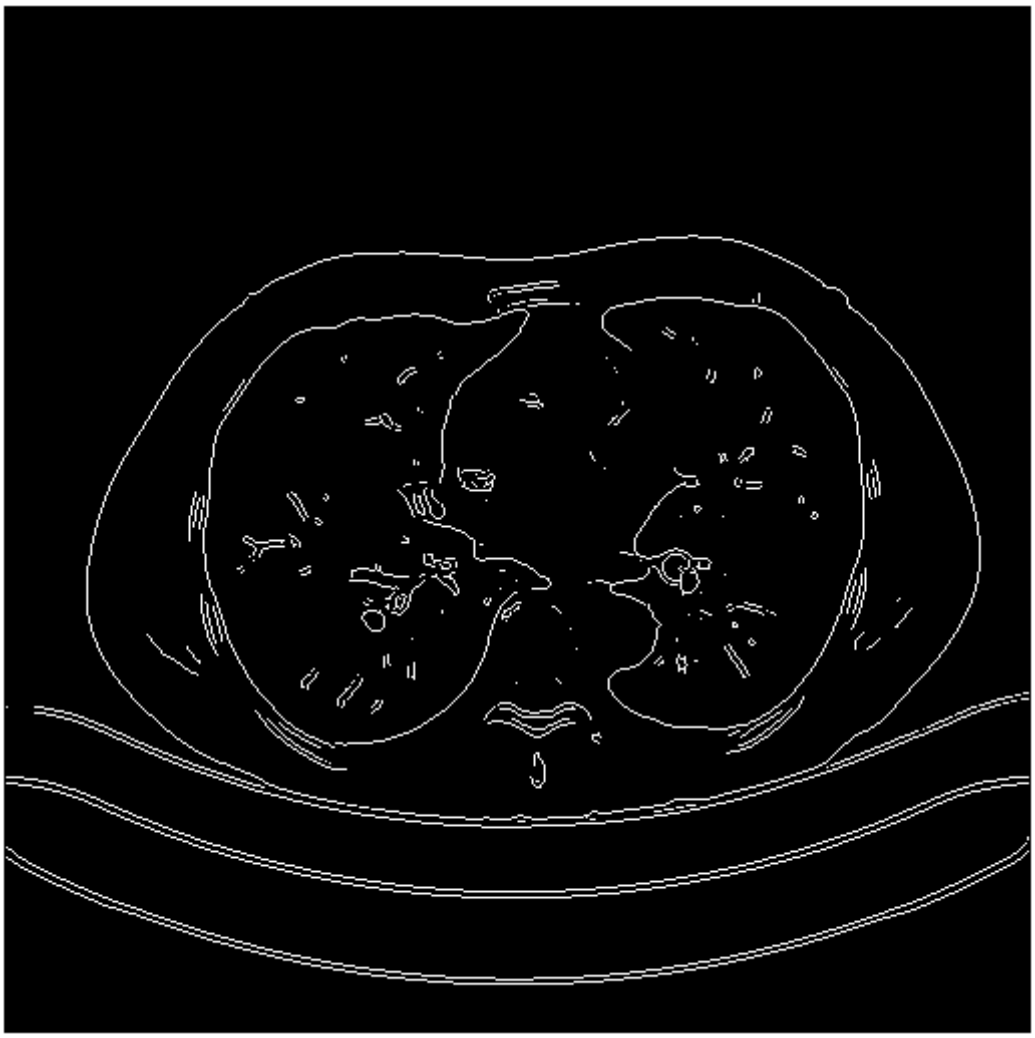


Figure 6.18 Sobel edge detector

APPENDIX B

ORIGINAL BRAIN MR and EDGE IMAGE

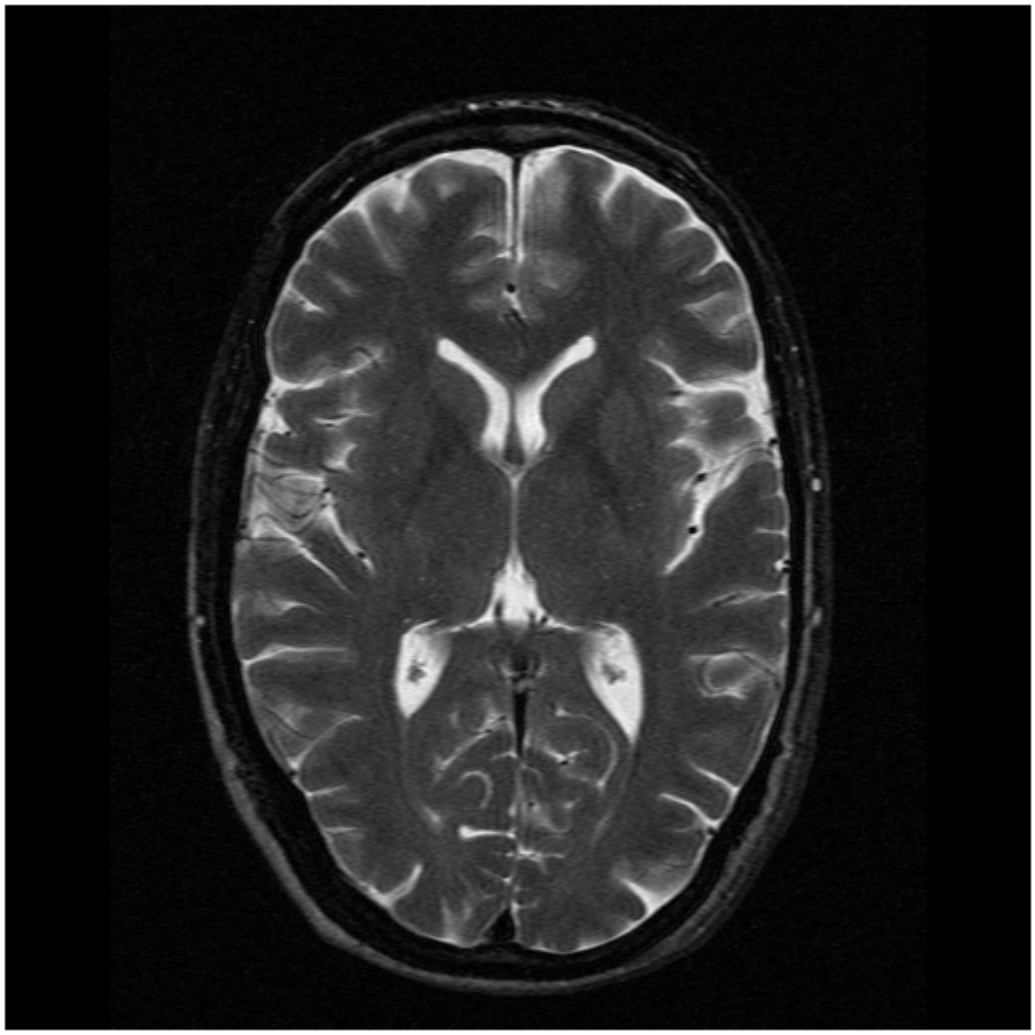


Figure 6.19 Original brain MR

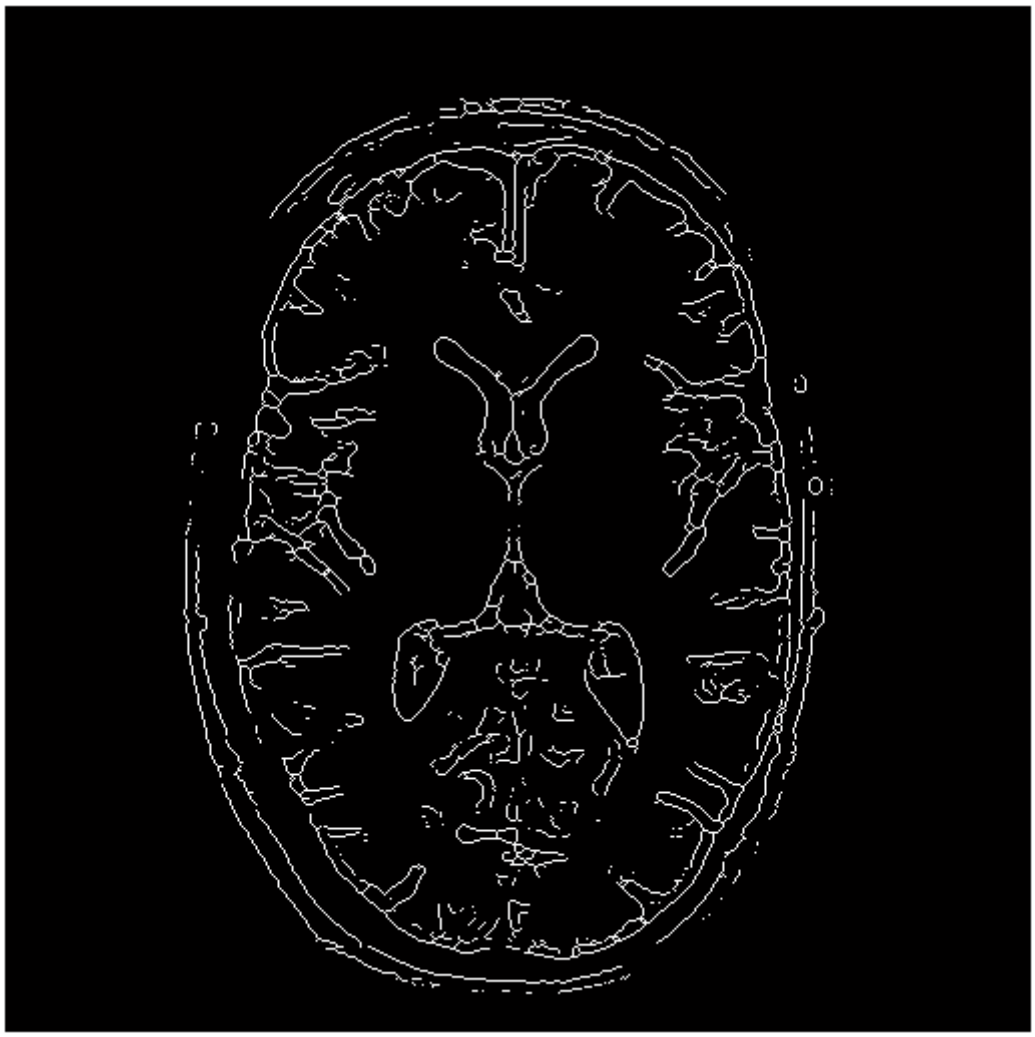


Figure 6.20 Edge image of the brain MR using new algorithm (Contrast enhanced)

APPENDIX C

SOME IMPORTANT NOİSE PROBABİLİTY DENSİTY FUNCTİONS

Gaussian noise

The PDF of a Gaussian random variable, z is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

where z represents gray level, μ is the mean of average value of z, and σ is its standard deviation. The standard deviation squared, σ^2 is called the variance of z.

Uniform Noise

The PDF of a uniform noise is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$

Impulse (salt-and-pepper) noise

The PDF of impulse noise is given by

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$