

A  
**DISSERTATION REPORT**

on  
*Automatic Bone Segmentation from X-ray images*

by  
**Somendra Meena**  
2017PEB5072

*Submitted in partial fulfillment for award of degree of*

**MASTER OF TECHNOLOGY**  
*in Embedded Systems*



**Internal Supervisor**  
**Sanjeev Agrawal**  
Associate Professor  
Dept. of ECE  
MNIT, Jaipur

**External Supervisor**  
**Mukesh Holani**  
Senior Software Engineer  
JR Robotics  
SGTC, Gurugram

**Department of Electronics and Communications  
Engineering**

© Malaviya National Institute of Technology Jaipur, 2019.  
All rights reserved

# Certificate



Department of Electronics and Communications Engineering  
Malviya National Institute of Technology, Jaipur

This is to certify that the Dissertation Report on “**Automatic Bone Segmentation from X-ray images**” by **Somendra Meena** 2017PEB5072 is bonafide work completed under my supervision and guidance, hence approved for submission in partial fulfillment for the **Master of Technology in Embedded Systems** to the Department of Electronics and Communication Engineering, Malaviya National Institute of Technology, Jaipur in the academic session 2018-2019 for the full time post graduation program of session 2017-2019. The work has been approved after plagiarism check as per institute rule.

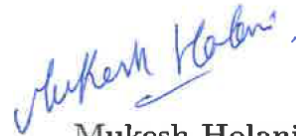
*Date:*

**Sanjeev Agrawal**  
Associate Professor  
Dept. of ECE, MNIT Jaipur

# Certificate

This is to certify that the Dissertation Report on “**Automatic Bone Segmentation from X-ray images**” by **Somendra Meena** 2017PEB5072 is completed under my supervision and guidance, hence approved for submission in partial fulfillment for the **Master of Technology in Embedded Systems** to the Department of Electronics and Communication Engineering, Malaviya National Institute of Technology, Jaipur in the academic session 2018-2019 for the full time post graduation program of session 2017-2019.

Date: 5 July 2019



**Mukesh Holani**  
Senior Software Engineer  
Joint Replacement Robotics, Stryker

# Declaration

I, hereby declare that the work which is being presented in this project entitled "**Automatic Bone Segmentation from X-ray images**" in partial fulfillment of degree of Master of Technology in Embedded Systems is an authentic record of my own work carried out under the supervision and guidance of **Sanjeev Agrawal**, Department of ECE, MNIT, Jaipur and **Mukesh Holani**, JR Robotics, Stryker Global Technology Center, Gurugram.

I am fully responsible for the matter embodied in this project in case of any discrepancy found in the project and the project has not been submitted for the award of any other degree. I also confirm that I have consulted the published work of others, the source is clearly attributed and I have acknowledged all main sources of help.

**Somendra Meena**  
2017PEB5072

# Acknowledgment

I am grateful to my supervisor **Sanjeev Agrawal**, Associate Professor for his constant guidance and encouragement and support to carry out this work. His excellent cooperation and suggestion provided me with an impetus to work and made the completion of work possible. He has been great source of inspiration to me, all through. I am very grateful to him for guiding me how to conduct research and how to clearly and effectively present the work done.

I would like to express my deepest sense of gratitude and humble regards to our Head of Department **Dr. D. Boolchandani** for providing necessary facility in the Department. I am very thankful to all the faculty members of ECE, MNIT for their valuable assistance and advice.

I am also thankful to **Mukesh Holani**, Senior Software Engineer for his valuable suggestion and guidance in my project during my internship in Stryker Global Technology Center. I would also like to thank my mentor **Aditya Chhabra** for his constant support throughout my internship in Stryker. I am grateful to him in guiding me in the successful completion of this project work.

I would also like to thank my friends for their support in discussions which proved valuable for me. I am indebted to my parents and family for their constant support, love, encouragement, and sacrifices made by them so that I could grow up in a learning environment.

Finally, I express my sincere thanks to all those who helped me directly or indirectly to complete this work successfully.

**Somendra Meena**  
2017PEB5072

# Abstract

Medical images have huge amount of information which can be utilized to diagnose a patient for various type of diseases. The most common forms of medical images includes X-ray, CT scans and MRI. MRI and CT scans captures the 3D model of the body part while the X-ray images have the 2D image. The X-ray images are widely used for examining bone structures. This paper is focused on extraction of bone contours from the x-ray images for the inferior section of human body. X-ray images are highly non-uniform in intensity and have high variation in homogeneity. This becomes a challenge for bone segmentation process. A method for the segmentation of bone contours in from x-ray images is proposed in this paper. It is based on detection of edge fragments and eliminating the discontinuities between them. The numerical parameters used for edge detection and joining may vary with each x-ray image. The algorithm gives the accurate edges of bones present in x-ray image. The results obtained using the proposed algorithm can be used for the diagnosis of diseases in patient body.

**Keywords:** Medical imaging, X-ray images, Bone segmentation, Contour extraction, Edge detection.

# Contents

<b>Certificate</b>	<b>i</b>
<b>Certificate</b>	<b>ii</b>
<b>Declaration</b>	<b>iii</b>
<b>Acknowledgement</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Tools Used . . . . .	2
1.3 Outline of Thesis . . . . .	2
<b>2 Literature Survey</b>	<b>4</b>
<b>3 Methodology</b>	<b>7</b>
3.1 Medical Terminologies . . . . .	7
3.1.1 Anatomical Planes . . . . .	7
3.1.2 Anatomical Locations . . . . .	8
3.1.3 Bone Anatomy . . . . .	10
3.2 Edge Detection . . . . .	13
3.2.1 Sobel Operator . . . . .	13
3.2.2 Kirsch Operator . . . . .	14
3.2.3 Elder Zucker Method . . . . .	15
3.2.4 Canny Edge Detection . . . . .	16
3.3 Elimination of discontinuity . . . . .	18
3.3.1 Removal of unwanted pixels . . . . .	18
3.3.2 Convex Hull Algorithm . . . . .	19
3.3.3 Bresenham's Line Drawing Algorithm . . . . .	20

<b>4</b>	<b>Implementation</b>	<b>22</b>
4.1	Pre-processing . . . . .	22
4.1.1	Change Color space . . . . .	23
4.1.2	Bilateral Filtration . . . . .	23
4.2	Edge Detection . . . . .	24
4.2.1	Canny Edge Detector . . . . .	24
4.2.2	Removal of small pixel areas . . . . .	25
4.2.3	Removal of closed areas . . . . .	25
4.3	Chaining . . . . .	26
4.3.1	Detection of endpoints . . . . .	26
4.3.2	Connecting the endpoints . . . . .	27
<b>5</b>	<b>Results</b>	<b>30</b>
<b>6</b>	<b>Conclusion &amp; Future Scope</b>	<b>32</b>
6.1	Conclusion . . . . .	32
6.2	Future Scope . . . . .	32
	<b>References</b>	<b>34</b>



# List of Figures

2.1	Result after (a) Initial segmentation (b) Region merging . . . . .	4
2.2	Incorrect Segmentation examples (a) Incorrect identification of initial position (b) one of the acceptable solutions but not accurate	5
2.3	(a) Candidate femoral shafts. (b) Candidate femoral heads . . . . .	6
3.1	Anatomical planes for a human body . . . . .	7
3.2	Medial and lateral locations in human body . . . . .	8
3.3	(a)Anterior and Posterior (b) Superior and Inferior locations in human body . . . . .	9
3.4	Proximal and distal locations in human body . . . . .	10
3.5	Hip joint . . . . .	11
3.6	Femur bone anatomy . . . . .	11
3.7	Tibia bone anatomy . . . . .	13
3.8	Non-maximum suppression . . . . .	17
3.9	Edge Connectivity for 2-D image . . . . .	18
3.10	Edge Connectivity for 3-D image . . . . .	19
3.11	(a) Original Image (b) Image showing holes with red color . . . . .	19
3.12	Bresenham's line drawing algorithm . . . . .	21
4.1	Main Block Diagram of system . . . . .	22
4.2	(a) Original Image (b) Filtered Image . . . . .	23
4.3	Edge Detection using (a) Sobel (b) Kirsch (c) Canny Operator . . . . .	24
4.4	(a) Output of Edge Detection (b) Removal of small pixel areas (c) Removal of closed areas . . . . .	25
4.5	(a) Correct (b) Incorrect detection of endpoints of edge fragments	27
4.6	Connecting broken edges with pixel distance of 0, 5 & 20 pixels . . . . .	28
4.7	Removal of false edges from binary image . . . . .	29
4.8	(a) Original Image (b) Segmented Image . . . . .	29
5.1	Examples of successful segmentation . . . . .	30
5.2	Examples of (a) Partial & (b) Incorrect segmentation . . . . .	31

# List of Tables

5.1	Time Complexity of the proposed method . . . . .	31
-----	--------------------------------------------------	----

# Chapter 1

## Introduction

### 1.1 Overview

In the current era of technologies, digital images have become a good source of information. They are widely used in the areas of photography, surveillance, space, medical and many others. In medical imaging, digital images are used in the field of instrumentation, diagnostics and therapeutic applications. Most of these applications are based on image processing techniques. Medical image processing has been a core field of innovation towards modern health care and is now used in many crucial applications such as diagnosis of diseases for a patient, manufacturing of various implants for the diseased bone or analysis of a patient in hospitals. The extraction of information from a medical image includes processes like enhancement, segmentation, registration, quantification, visualization and computer-aided detection.

Segmentation is an important process used in the field of medical image processing. It is a process of detecting the boundaries of an object from a 2D or 3D image. However, most of the research in segmentation of medical images are focused on CT and MRI images, X-ray images are one of the most widely spread and cost-effective non-invasive input for medical monitoring and image analysis.

In this paper we present the method developed and used for X-ray image analysis. This paper focuses on the problem of automatic extraction of bone contours in X-ray images. Low contrast in X-ray images and partial invisibility of bones are few major challenges in the segmentation of bone. Also, irregularity in texture and intensity makes the segmentation of bone more difficult. The method used here is based on detection of edge fragments and elimination of discontinuities between those fragments. The results obtained using this algorithm can be used for the diagnosis of diseases for the detected bone.

## 1.2 Tools Used

For the purpose of image processing, there are mainly two tools that are mostly used: MATLAB and OpenCV. OpenCV is a C++ build library for performing the operations on image. It is available in many languages including Java and Python. OpenCV is mostly used with Python as it provides good functionalities with the language. MATLAB is another tool which came very earlier than OpenCV and has a lot of good functionalities over OpenCV in terms of image processing. The features of MATLAB are discussed in the next section:

### **MATLAB**

The term stands for MATrix LABoratory. It is a high end programming language used for technical computing. It includes toolboxes for signal & image processing, control systems, machine & deep learning, robotics, statistical operations etc. MATLAB also supports object-oriented programming and contains built-in functions for most of the algorithms of image processing.

For this project, we have used MATLAB for image acquisition, processing and displaying the output. Unlike OpenCV, MATLAB consists of various toolboxes that are helpful in image processing techniques. The MATLAB stores the image in the form of a matrix. This image matrix can be viewed using the variables toolbox. Also, the debugger in MATLAB helps in inspecting the value of each pixel at any point of time, even during the execution of code.

Another advantage of MATLAB is its documentation and community where we can find solutions to almost every problem. Based on these features and advantages of MATLAB over OpenCV, it is being preferred for this project.

## 1.3 Outline of Thesis

### **Chapter 1**

The thesis starts with Chapter 1 that briefs about the techniques of image processing and its applications in medical field. Overview section of this chapter consists of different approaches used in image segmentation. It also describes the challenges in the segmentation of bone from X-ray images. Tools used in the implementation of this project is also discussed in this chapter. Detailed discussion of the implementation will be discussed in later chapters.

### **Chapter 2**

This chapter describes the motivation for the work done and the challenges faces during the implementation of proposed method. The research papers referenced for this project are also discussed in this chapter. This chapter also throws light on the advantages and drawbacks of the methods proposed in those papers.

### **Chapter 3**

This chapter starts with the introduction of few medical terminologies. The planes and axes used while capturing the X-ray images are discussed in this chapter. Different techniques of edge detection are also described here. This chapter also includes the methods for elimination of discontinuities between different edge fragments. Algorithms for removal of noise and false edges is also mentioned here.

### **Chapter 4**

This chapter includes the implementation details of the proposed algorithm. It starts with the pre-processing of the acquired image. Different edge detection techniques are then compared and their merits & demerits are discussed here. The outputs from different edge detection methods are shown here. Next in this chapter are the algorithms used for removing the unwanted pixels. The results obtained with these algorithms are also shown here. A section in this chapter consists the steps and results from the chaining algorithm used in this project.

### **Chapter 5**

In this chapter, the final results obtained by using the proposed algorithm are shown here. It includes successful as well as unsuccessful results. Also, the time complexity of this algorithm is measured for different images of various resolutions and image intensity.

### **Chapter 6**

Chapter 6 covers the conclusion of the project and the future scopes and enhancements which can be further made to this.

## Chapter 2

# Literature Survey

### *Paper on An Interactive X-Ray Image Segmentation Technique for Bone Extraction*

In this paper[3], author proposed an user interactive method to segment the bones from the rest of the image. The proposed method is based on two separate image processing techniques. For the initial segmentation, a mean shift algorithm is used which produces a number of small regions. These small regions are then merged to form larger ones. Two regions are considered to be merged if they are similar or connected to each other. The statistical test used here to decide the merging of regions is Bhattacharyya coefficient. Bhattacharyya coefficient correlates images using their histograms and used as similarity measure between them. The proposed region merging method requires marking of the

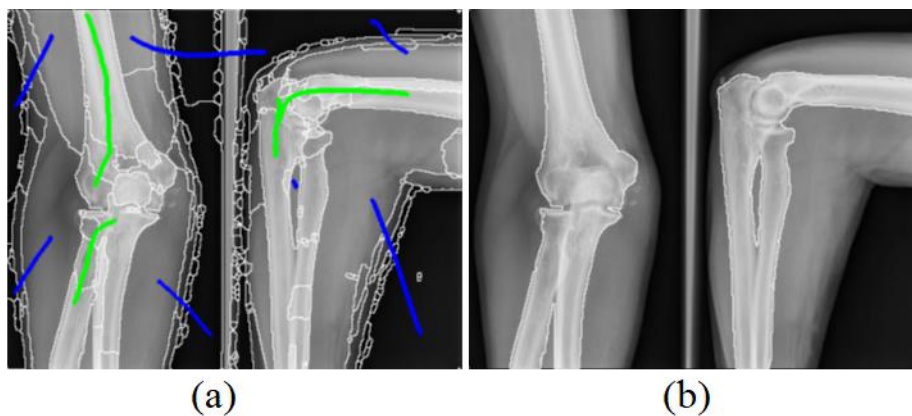


Figure 2.1: Result after (a) Initial segmentation (b) Region merging

object and the background which divides regions of image into object marker, background marker or non-marker region. The markers placed by the user and result after region merging algorithm are shown in Figure 2.1. The proposed

segmentation method is highly effective and produces accurate edges for the bone. But as this method requires user interaction, it is not application for automatic segmentation.

**Paper on *Automatic Detection of Bone Contours in X-Ray Images***

This paper[4] proposes an approach for automatic detection of bone contours. The proposed method does not require homogeneity of the regions. This method involves detection of edge fragments using Gradient Vector Flow (GVF) and eliminating the discontinuities between them using chaining algorithm. A criteria to calculate numerical estimate of the edge detection quality is defined in this method. The estimation of edge detection quality is defined as:

$$E = \frac{w(\bar{\Omega})}{v(I^{bin})} \left( 1 - \frac{v(\bar{G} * I^{bin})}{v(I^{bin})} \right) \left( 1 - \frac{|\bar{\Omega}| - w(\bar{\Omega})}{NM - v(I^{bin})} \right)$$

where,

$\Omega$  denotes the bounded region,

$w(\Omega)$  denotes the cardinality within  $\Omega$  region,

$I^{bin}$  denotes the binary version of image I,

$\bar{G}$  denotes binary image after applying chaining algorithm,

$N \times M$  denotes the dimensions of image matrix.

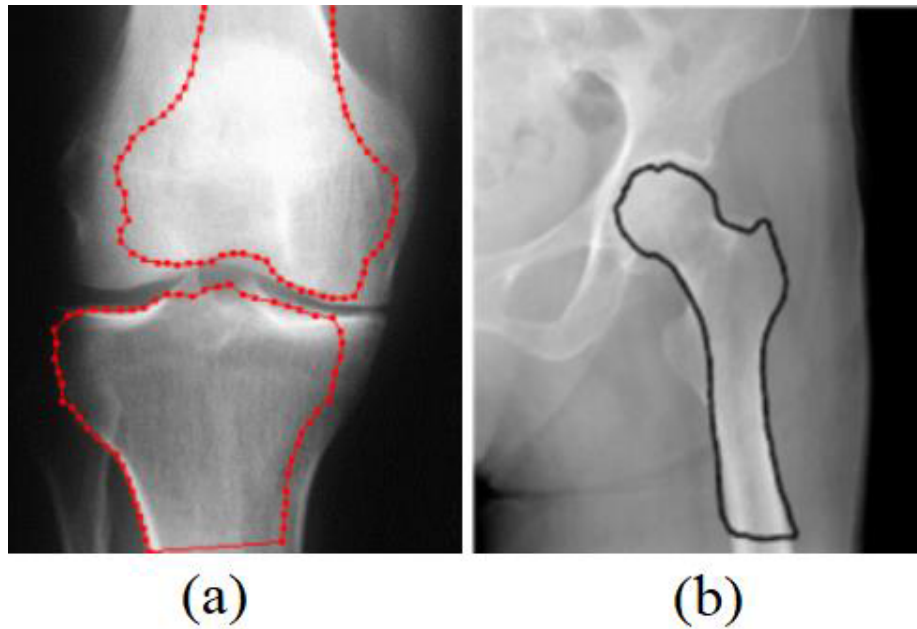


Figure 2.2: Incorrect Segmentation examples (a) Incorrect identification of initial position (b) one of the acceptable solutions but not accurate

**Paper on *Automatic Extraction of Femur Contours from Hip X-ray Images***

Some segmentation techniques are based on using a template for the desired bone and some prior knowledge about it. When using this method, an initial position of the template is identified on the image. Then the template is used to detect the identical region in the image. In paper [5], an incremental approach is proposed for the segmentation of femur bones from the hip X-ray images. In this approach various features in X-ray image are used for searching the initial position of template. These features, in case of Femur bones, includes parallel lines, circles, etc. These detected features of femur bone is shown in Figure 2.3.

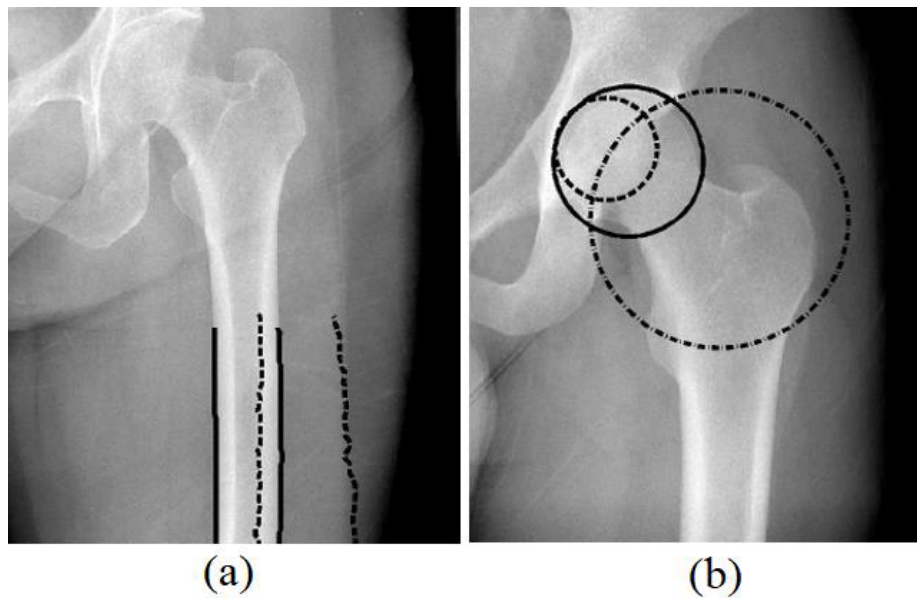


Figure 2.3: (a) Candidate femoral shafts. (b) Candidate femoral heads

However, the model-based approach for detecting the bone is not applicable for objects with strong shape distortion. The author of this paper demonstrates the example of incorrect contours extraction in cases of deformed or fractured bone.



# Chapter 3

## Methodology

### 3.1 Medical Terminologies

#### 3.1.1 Anatomical Planes

In a 3D space, plane can be represented as a 2D slice which cuts the space into two regions. Similarly, an anatomical plane divides a body into two parts. There are three planes which are commonly used to divide the human body: sagittal, coronal, transverse. Figure 3.1 shows the three planes with respect to the human body.

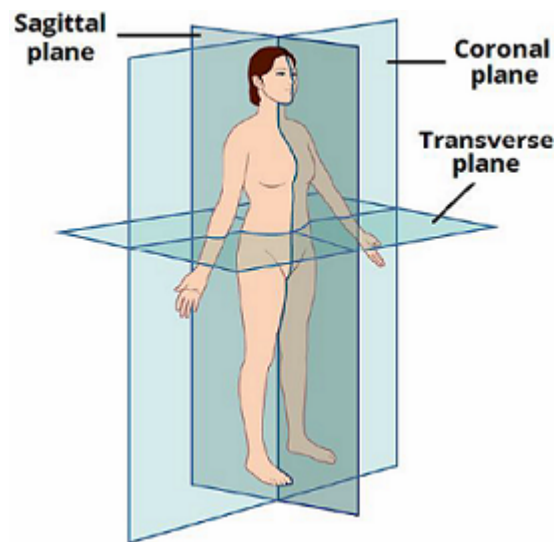


Figure 3.1: Anatomical planes for a human body

### Sagittal Plane

Sagittal plane is a vertical plane, parallel to the sagittal suture (a tissue joint in skull), which divides the body into left and right halves.

### Coronal Plane

Coronal plane is also a vertical plane which divides the body into anterior(front) and posterior sections.

### Transverse Plane

Transverse plane is a horizontal plane, parallel to the ground surface, which divides the body into superior(upper) and inferior(lower) parts.

## 3.1.2 Anatomical Locations

The location descriptive term[20] for a human body is referred to a body in a standing position. In this position, the arms should be at the sides and palms should face forward (with thumbs out).

### Medial and Lateral

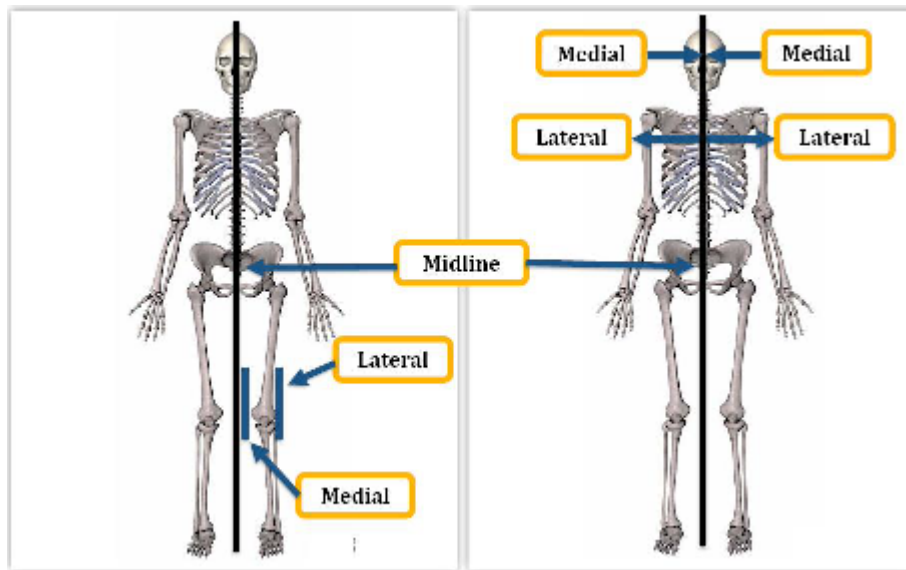


Figure 3.2: Medial and lateral locations in human body

The mediolateral axis divides the body into medial and lateral, which refers to the direction of body part from the midline. A side towards the midline is

referred to medial location and that away from the midline is referred to the lateral location. Figure 3.2 shows the mediolateral positions for the femur bone of a human body.

### Anterior and Posterior

Anteroposterior axis divides the human body into anterior and posterior parts. Anterior and posterior are the front and back parts of the human body respectively. Figure 3.3(a) shows the anterior and posterior views of the human body.

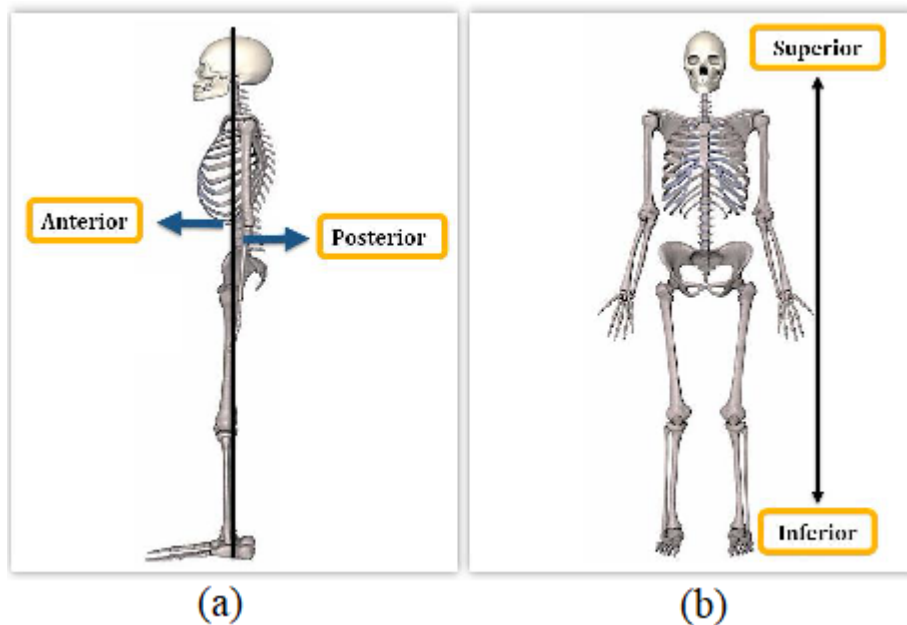


Figure 3.3: (a)Anterior and Posterior (b) Superior and Inferior locations in human body

### Longitudinal

The longitudinal axis divides the body in superior and inferior parts, which refers to the upper and lower parts of the human body. Figure 3.3(b) shows the superior and inferior views of the human body.

### Proximal/distal

The proximal location are those locations in the body at which appendage joins the body. Appendage are the body parts which are attached through a joint like shoulder or knee or hip. The distal locations are those which are at the extremity

of the appendage. Figure 3.3(a) shows the proximal and distal locations of the human body.

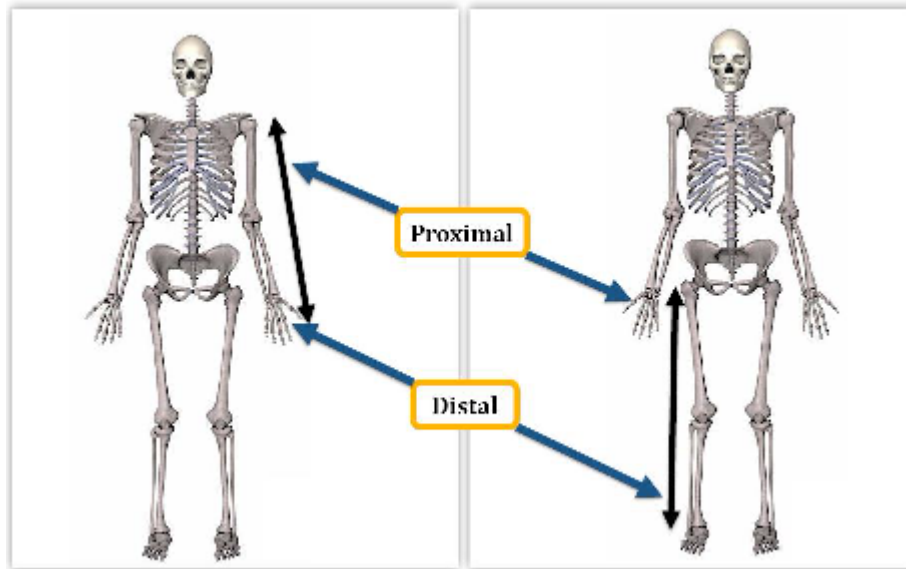


Figure 3.4: Proximal and distal locations in human body

### 3.1.3 Bone Anatomy

Bone anatomy tells us about the attachments, landmarks and clinical correlations of the bone. This section describes the bone anatomies of few bones in the inferior part of the body.

#### Hip Joint

Hip joint is the largest synovial joint in the human body. It is a ball-and-socket joint where the femoral head is the ball and the acetabulum is the socket. Figure 3.5(a) shows this ball and socket joint.

#### Hip Bone

Hip Bone is also known as Coxal or innominate bone. This bony structure is formed of three bones that come together to form the acetabulum, the socket of the hip joint. The bones are separate at birth, but have fused by the time the body reaches adulthood (Figure 3.5(b)). The three bones of the hip are:

- **Ilium** - largest bone of the hip, forming projections on either side
- **Ischium** - lower back part of pelvis bone
- **Pubis** - anterior portion of hip bone

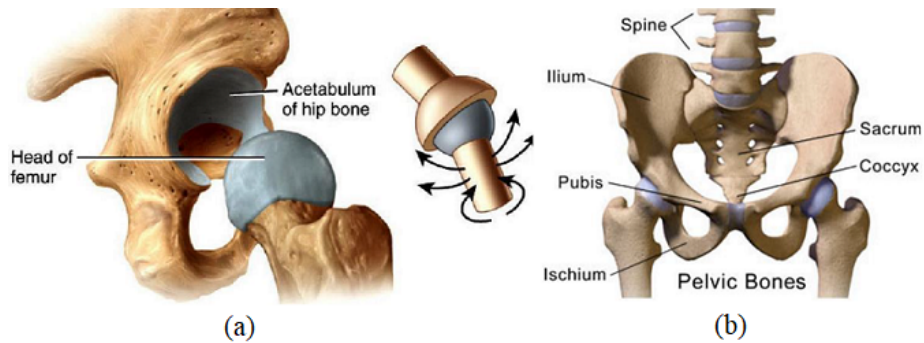


Figure 3.5: Hip joint

### Pelvis

It is the lower part of the trunk of the human body present between the abdomen and the thighs. The pelvic skeleton is formed in the area of the back, by the sacrum and the coccyx and anteriorly and to the left and right sides, by the pair of hip bones. The two hip bones connect the spine with the lower limbs. They are attached to the sacrum posteriorly, connected to each other anteriorly, and joined with the two femurs at the hip joints.

### Femur

Femur bone is the longest and strongest bone in the human body. It is the only bone in thigh and is also called thigh bone. The femur forms the part of both hip and the knee joints. The femur bone consists of three main parts: head, neck and shaft[18].

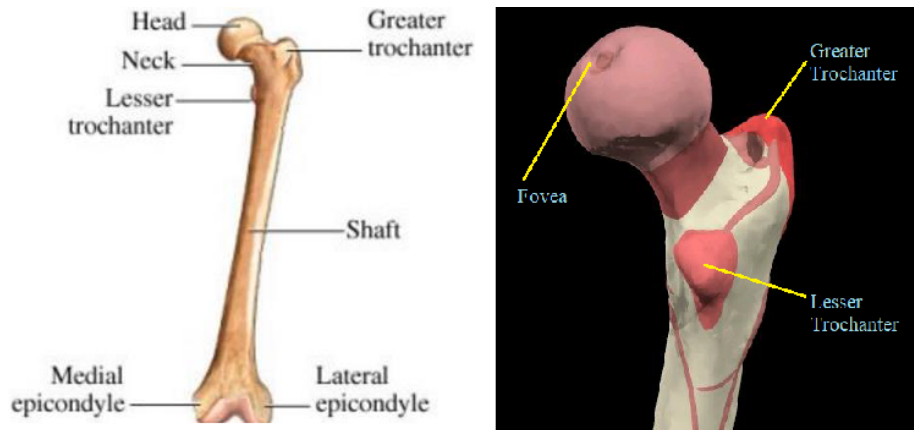


Figure 3.6: Femur bone anatomy

- **Femoral Head:**  
The femoral head is almost of spherical shape. Its diameter is variable and has no relation with person's age or height. It has a smooth surface that is covered with articular cartilage.
- **Femoral Neck:**  
The femoral neck is around 5cm long that connects femoral head to the trochanteric region. It has a neck angle of  $125^{\circ}$ - $135^{\circ}$  relative to the shaft
- **Femoral Shaft:**  
The femoral shaft is a cylindrical structure which goes slightly in the medial direction when moving from hip to knee joint. As this brings our knees towards center of gravity of body, it also increases stability. The cross-section of the femoral shaft is flattened and wide at proximal & distal locations while circular & narrow in the middle. The shape and size of the shaft varies with person to person and on deformities if any.
- **Trochanteric area:**  
This area is divided into two parts: Greater trochanter and lesser trochanter. Greater trochanter is a region lateral to the lesser trochanter. It is a bony prominence that marks the upper end of the shaft of the femur. Lesser trochanter is a blunt, conical projection located posterior to and medial to the shaft of the femur.

## Tibia

Tibia bone is the weight bearing bone present in the leg below the knee. It is the second longest bone in the human body after Femur. The tibia bone consists of three main parts: proximal, shaft and distal.[19]

- **Proximal part:**  
The proximal part of the tibia bone consists of two tibial condyles which are separated by intercondylar fossa. Intercondylar fossa is the depression between the condyles, in both anterior and posterior directions. The two condyles are at the medial and lateral positions just below the knee.
- **Shaft:**  
The shaft of tibia bone is a triangular in shape and therefore has three surfaces: posterior, medial and lateral and three borders: anterior, posterior and lateral. The anterior border is the most important one which begins from the proximal part and descends to the distal part of the bone.
- **Distal part:**  
The distal end of the tibia changes is wide and rectangular in shape. Increased cross-section at distal helps in weight-bearing. The lateral surface of tibia is bound to fibula. It forms the tibiofibular joint and is called fibular notch.

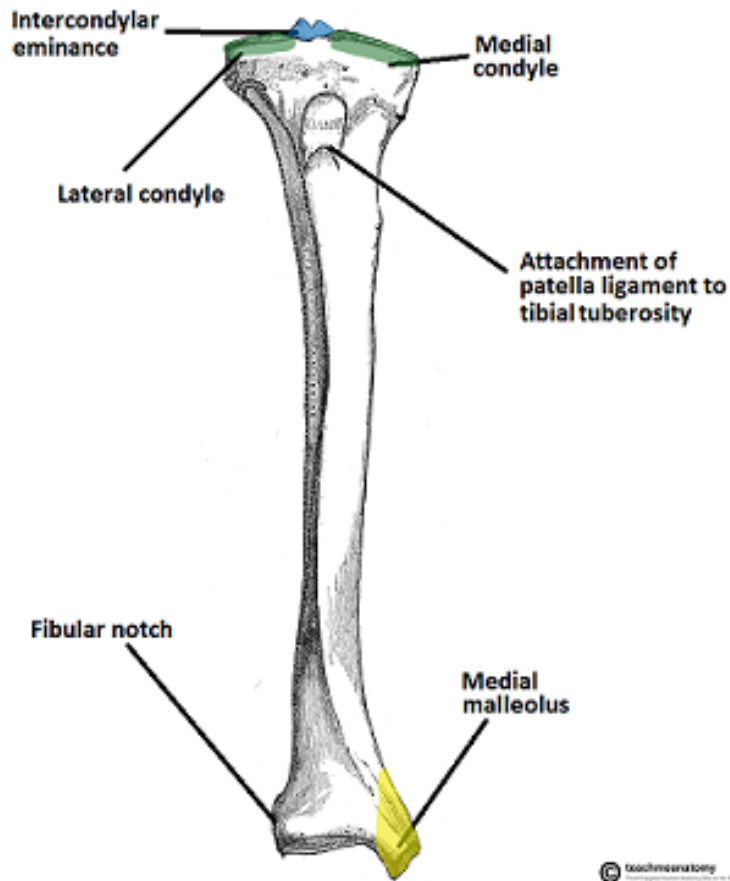


Figure 3.7: Tibia bone anatomy

## 3.2 Edge Detection

### 3.2.1 Sobel Operator

The Sobel operator is used in computer vision particularly for the purpose of edge detection in an image. This technique was given by Irwin Sobel and Gary Feldman in 1968 and sometimes called the Sobel-Feldman operator. The Sobel operator is based on approximating the gradient magnitude for a grayscale image at each point. The result of the Sobel-Feldman operator at any point in the image is either the corresponding gradient vector or the norm of this vector. It emphasizes the regions of high spatial frequency corresponding to the edges.

### WORKING:

This operator consists of two 3x3 kernels, where one kernel is simply the 90° rotation of another. These two kernels are convolved with the original image in order to calculate the derivative approximations, each for horizontal and vertical changes. These kernels are designed in such a way that they responds maximally to the edges in vertical and horizontal direction relative to the pixel. The derivatives can be computed as follows:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$
$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Now, the magnitude of the gradient can be calculated from  $G_x$  and  $G_y$  as:

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

And the gradient's direction as:

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

The Sobel operator is slower than Roberts Cross operator in computation of gradient but makes the operator less sensitive to noise by smoothing the image to a greater extent.

### 3.2.2 Kirsch Operator

Kirsch operator is named after the computer scientist Russell A. Kirsch. It is a non-linear edge detector which finds the maximum strength of edges in a predetermined directions.

The Kirsch operator takes a single kernel mask and rotates it in all the eight compass directions with increment of 45° at each rotation in order to get the edges in all eight directions. The magnitude of edge for Kirsch operator is calculated as the maximum magnitude in any direction. It can be written as:

$$h_{n,m} = \max_{z=1,\dots,8} \sum_{i=-1}^1 \sum_{j=-1}^1 g_{ij}^{(z)} \cdot f_{n+i,m+j}$$

where  $z$  represents the direction of kernel  $\mathbf{g}$ . The compass direction kernels[17] are taken as:



$$\begin{aligned}
g^{(1)} &= \begin{bmatrix} +5 & +5 & +5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & g^{(5)} &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ +5 & +5 & +5 \end{bmatrix} \\
g^{(2)} &= \begin{bmatrix} +5 & +5 & -3 \\ +5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & g^{(6)} &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & +5 \\ -3 & +5 & +5 \end{bmatrix} \\
g^{(3)} &= \begin{bmatrix} +5 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & -3 & -3 \end{bmatrix} & g^{(7)} &= \begin{bmatrix} -3 & -3 & +5 \\ -3 & 0 & +5 \\ -3 & -3 & +5 \end{bmatrix} \\
g^{(4)} &= \begin{bmatrix} -3 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & +5 & -3 \end{bmatrix} & g^{(8)} &= \begin{bmatrix} -3 & +5 & +5 \\ -3 & 0 & +5 \\ -3 & -3 & -3 \end{bmatrix}
\end{aligned}$$

The image is filtered with all the direction kernels and their magnitude in that direction is calculated. After filtering of the image with all eight directions, maximum gradient is calculated as the maximum magnitude in all the directions.

### 3.2.3 Elder Zucker Method

In 2000, Elder and Zucker proposed a method[1] for compressing the image by storing only small portion of the image, the areas near the edges, rather than storing the whole image. According to Elder, image is discontinuous at edges and is relatively smoother at other locations and hence much of that information may be redundant. So he proposed two necessary quantities of image: the intensity at edge locations and the estimate of the blur at those locations. As the edge density in image size is linear, larger images will have higher compression ratios and hence this compression technique is suitable for them.

#### Edge Detection

To find the edges in an image, we usually look for a sudden change in image intensity. The real images are usually corrupted with noise. So a good edge detector should be able to differentiate between real and random(due to noise) intensity transitions. Generally, noise can be filtered by using Gaussian filter. For images with little noise, a narrow filter is required while a broad filter will be required for blurry noise images. It computes a 1-D Laplacian across the edges locally and the zero-crossings in the minimum reliable Laplacian will indicate the edge locations.

### Blur Estimation

Elder proposed the following formula to calculate the blur estimation at the edge:

$$blur = \sqrt{\left(\frac{d}{2}\right)^2 - s^2}$$

where:

d = distance in pixels between extrema in 2<sup>nd</sup> derivative map

s = minimum reliable scale for 2<sup>nd</sup> derivative operation

### Compression

To reconstruct the image, intensity values on either side of a detected edge is required along with a blur estimate along the edge. Either of these two schemes can be used to compress the data:

- Run length encoding for each map, followed by entropy encoding.
- Encode all relevant information as an order quadrapule.

### Reconstruction

The reconstruction of the image starts with the interpolation across all the edge information obtained from the original image so that we can fill the gaps between the edges. Also, the same technique is used to reconstruct the smoothly interpolated blur map from the estimated edge blur map. This technique is based on the Laplace equation:

$$\Delta u = u_{xx} + v_{xx} = 0 \tag{3.1}$$

### 3.2.4 Canny Edge Detection

This algorithm for edge detection was developed by John F. Canny in 1986. The operator used in this edge detection technique uses a multistage algorithm to detect wide range of edges. Edge detection by using Canny method involves following steps:

#### Gaussian Filter

Before undergoing the edge detection process, the image needs to undergo with a blurring process. By smoothing of the image, we reduce the noise effects in the edge detection process. The equation for a Gaussian filter is:

$$H_{ij} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}}$$

An example of a 5x5 Gaussian filter, with  $\sigma = 1.4$

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

### Edge Gradient and Direction

For each point of the image, gradient magnitude and directions are calculated. Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. Filtering the smoothed image with Roberts or Sobel kernel gives 1<sup>st</sup> derivative in horizontal and vertical directions as  $G_x$  and  $G_y$ . Edge gradient and direction can be calculated as:

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

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

### Non-maximum suppression

It is an edge thinning technique in which all the gradient values except the local maxima are suppressed (put to zero). This step removes any unwanted pixels which may not be an edge. Every pixel is checked if it is a local maximum in its neighborhood in the direction of gradient vector.

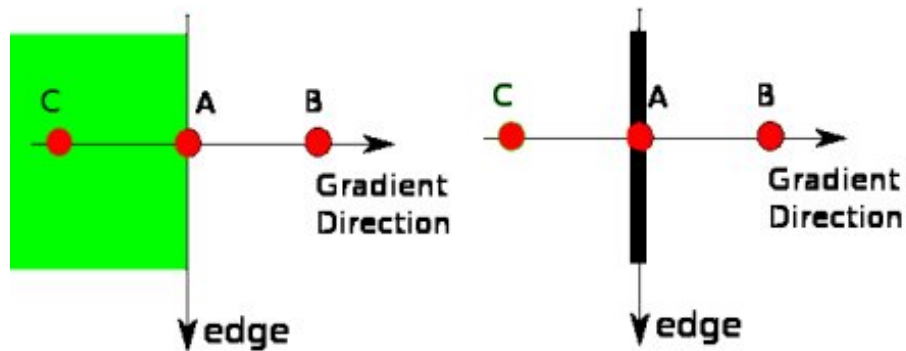


Figure 3.8: Non-maximum suppression

The Figure 3.8 shows the two cases where only one(right) of them is the local maxima in the gradient direction. If not, then it will be suppressed and will not be considered as an edge. This step results a binary image with thin edges.

### 3.3 Elimination of discontinuity

The image obtained from the edge detection step contains edge fragments which are part of the actual edge of the bone. To complete with the bone segmentation, these small fragments should be connected appropriately to form a complete edge. This algorithm is also known as chaining algorithm where we eliminate the discontinuity by joining the broken lines. The chaining process starts with finding the endpoints of the edge fragments, followed by connecting them using a line drawing algorithm. These processes are further explained in the next sections.

#### 3.3.1 Removal of unwanted pixels

During the detection of edges from any of the edge detector techniques, there are chances of getting some edge fragments which are not actually the part of the edges and needs to be removed from the image. The decision whether a pixel is noise or not depends upon the various factors which are considered while removing them. Some of the criteria for choosing noise signals and their removal are described below:

##### Removal of small pixel area

MATLAB provides a function *bwareaopen()* which removes all the connected components that consists of less than given pixel number. It takes two mandatory arguments: binary image, maximum number of pixels and one optional argument: connectivity. The connectivity arguments tells the function to specify the desired connectivity. For a 2-D image, connectivity can be of either of 4-connected or 8-connected, the default being 8-connected. For a 3-D image, default connectivity is 26-connected, while we can choose among 6-connected, 18-connected and 26-connected. The description of connectivity is shown in Figure 3.9 and 3.10.

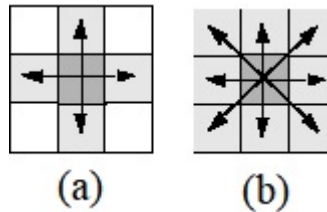


Figure 3.9: Edge Connectivity for 2-D image

##### Removal of closed areas

Sometimes unwanted figures or objects in front of the patient results in a noisy X-ray image. These objects are very likely to form a closed area in the binary

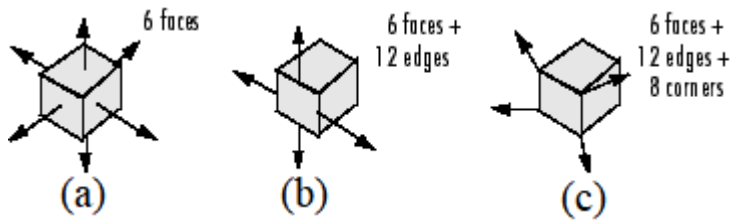


Figure 3.10: Edge Connectivity for 3-D image

edge detected output. On this basis, we can separate them from the binary image. In image processing, a closed loop can be easily detected by finding the Euler number for the binary image.

### Euler Number

Euler number of a binary image is a very important property of a digital image which can be used in image processing for pattern recognition, image reconstruction, finding closed loops in an image. Euler method can also be used to solve first order and first degree differential equations, provided an initial value is known. Euler number is the difference of total number of objects in the image and number of holes in those objects.

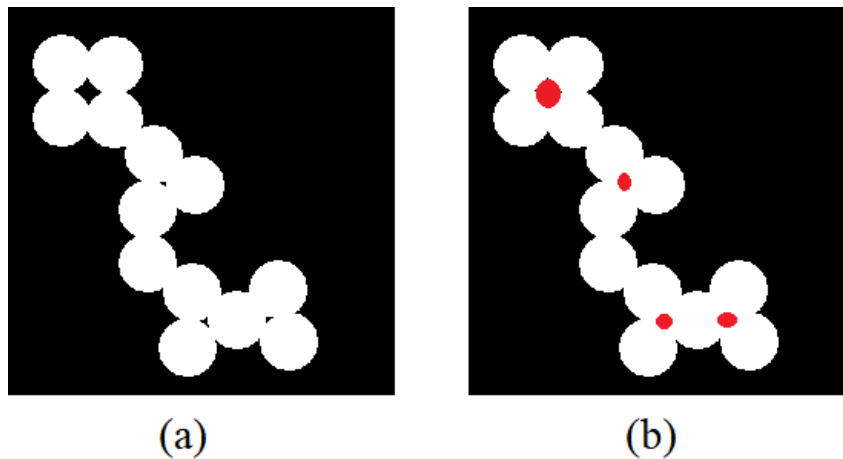


Figure 3.11: (a) Original Image (b) Image showing holes with red color

### 3.3.2 Convex Hull Algorithm

Convex Hull can be described as the smallest convex polygon which consists of a set of points such that each point in the set lies on or within the polygon.

There can be different approaches to find out the smallest polygon. Some of them are described in this section[12].

### **Graham's Scan**

The steps involved in Graham's Scan are:

1. Find  $p_0$ , the point with the minimum y coordinate
2. Sort all the remaining points in order of their polar angle from  $p_0$
3. Initialize a stack,  $S$
4. Push  $p_0$  to stack  $S$
5. Push  $p_1$  to stack  $S$
6. Push  $p_2$  to stack  $S$
7. For all other points  $p_i$ , pop  $S$  till the angle formed by second top  $S$ , top  $S$  and  $p_i$  makes a right turn.
8. Push  $p_i$  when the above condition fails
9. return  $S$

The complexity for this algorithm is  $O(n \log n)$

### **Jarvis's March**

The steps involved in Jarvis's March are:

1. Find the left-most and right-most points
2. Divide the points in upper and bottom half based on the position of point with respect to the line joining the above two points.
3. For the bottom half, start with the left most point and add the point which makes least angle to the Y-axis from the current point and continue to do so until the right-most point is reached.
4. Repeat the same in upper half part.

The complexity for this algorithm is  $O(nh)$ , where  $h$  is the number of vertices of convex hull.

### **3.3.3 Bresenham's Line Drawing Algorithm**

Bresenham's Line algorithm was developed in 1962 at IBM by Jack Elton Bresenham. This algorithm is used to determine the points in a 2-dimensional raster between two given points in order to draw a line between them. This algorithm is commonly used in computer graphics to draw line on a computer screen.

## WORKING

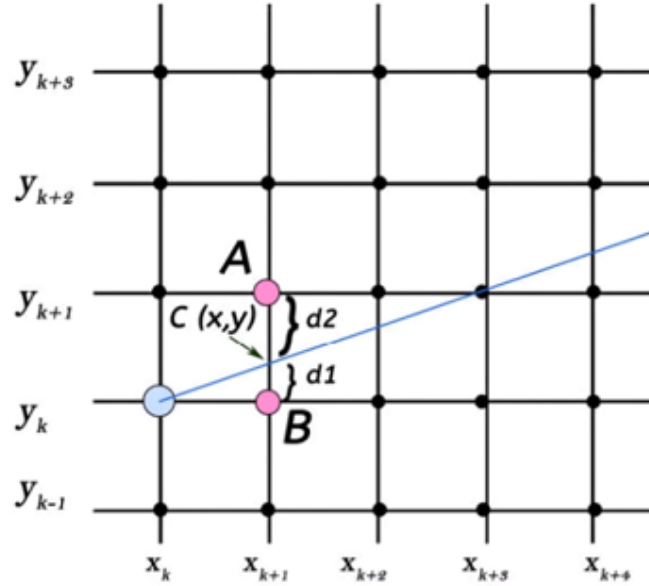


Figure 3.12: Bresenham's line drawing algorithm

Consider two points  $P1(x1, y1)$  and  $P2(x2, y2)$  (with  $x1 < x2$  and  $y1 < y2$ ) and we need to join these points by drawing a line between them. Before starting with the further steps, we need to first find out the slope of the line joining them.

$$m = \left| \frac{dy}{dx} \right| = \left| \frac{(y2 - y1)}{(x2 - x1)} \right|$$

So the equation of the line will be :  $y = mx + c$

Now based on the value of  $m$ , we will choose the base axis. We will then increment the value on base axis and will find the value of other axis point. If  $m=1$ , then the line will be angled at  $45^\circ$ , and the next point can be easily found by increment  $X$  and  $Y$  together. If  $m < 1$ , or  $dy > dx$ ,  $Y$ -axis will be taken as base axis otherwise if  $m > 1$ , or  $dx > dy$ ,  $X$ -axis will be taken as the base axis.

Lets say,  $dx < dy$ , so the slope will be less than  $45^\circ$  and  $X$ -axis is taken as our base axis. We will increment the value of  $X$  by 1 to find out the corresponding  $Y$  value. Now if the value of  $Y$  for  $X=x1+1$  is not an integer, we will find the distance of points  $A(x1 + 1, y1 + 1)$  and  $B(x1 + 1, y1)$  from point  $C(x, y)$  as  $d2$  and  $d1$  respectively. The point with lesser distance with point  $C$  will be selected for this value of  $X$ . This particular step for finding out the  $Y$  point corresponding to every value of  $X$  will be continued till we reach the point  $P2$ .

## Chapter 4

# Implementation



Figure 4.1: Main Block Diagram of system

Figure 4.1 shows the blocks of the system used for detecting the bone contours from X-ray images. The proposed method starts with the acquisition of the image, followed by pre-processing of the input image. This image is then applied to the edge detection technique which gives the edge fragments in the form of a binary image. Chaining algorithm is then applied to join the broken edges, which is further followed by removal of noise and some false detected edges. This chapter describes the implementation of the above blocks using the techniques discussed in chapter 3.

### Acquisition

An image in MATLAB is seen in the form of a matrix where every pixel value is a element of the image matrix. The dimensions of the image matrix depends on the resolution and color space of the image. For an image with resolution of 1280x720 where width and height of the image are 1280 & 720 respectively and color space of RGB, the dimension of a matrix will be  $720 \times 1280 \times 3$ .

### 4.1 Pre-processing

Before starting with the edge detection techniques, the image should be processed with some image enhancement techniques so that we can extract more information from it. The image pre-processing starts with conversion of color space for the input image to grayscale. Light blurring of image is then performed on this grayscale image in order to remove the spot noise. Out of many blurring techniques, the best results were obtained using bilateral filtration algorithm.



### 4.1.1 Change Color space

A X-ray image is always in grayscale color space. But while acquiring the image, it is being treated as an RGB image. So the RGB image needs to be converted to grayscale before going over to next steps. For this, MATLAB provides the function *rgb2gray()* which takes RGB image as input and returns the grayscale image as output.

### 4.1.2 Bilateral Filtration

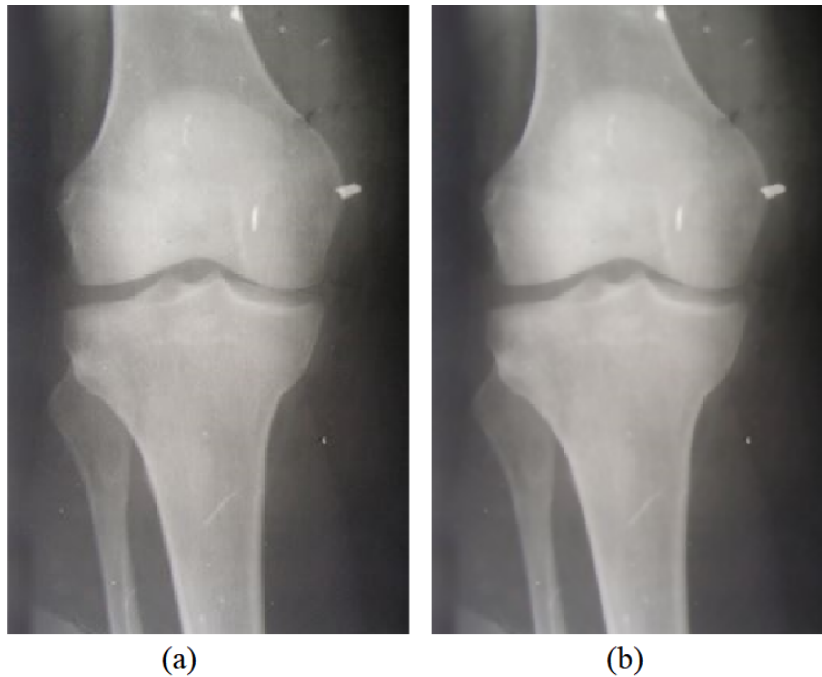


Figure 4.2: (a) Original Image (b) Filtered Image

A bilateral filter is an edge preserving filter which reduces the noise from the X-ray image. It replaces each pixel value with a weighted average of intensity values from nearby pixels. The number of pixels considered while calculating the average weight depends on the size of the Gaussian window. As the weight also depends on the radiometric differences, such as color intensity, depth distance, etc., the sharp edges are preserved. So this filter comes out to be the best filter for the edge detection. The original X-ray image along with its bilateral filtered output is shown in Figure 4.2.

## 4.2 Edge Detection

Edge Detection technique is used here to find out the boundary of the bone in the X-ray image. For this purpose, we have many options available but the Canny Edge Detector gives the best output in case of X-ray images. Sobel and Kirsch operators are also well known techniques for edge detection but in our case they have some limitations. Sobel operator is good in detecting edges in horizontal and vertical directions but if we need to detect edges in all the directions, Kirsch operator gives better results. Kirsch operator when used gives an closed shape which may detect edges incorrectly. The results obtained with these edge detection techniques is shown in Figure 4.3.

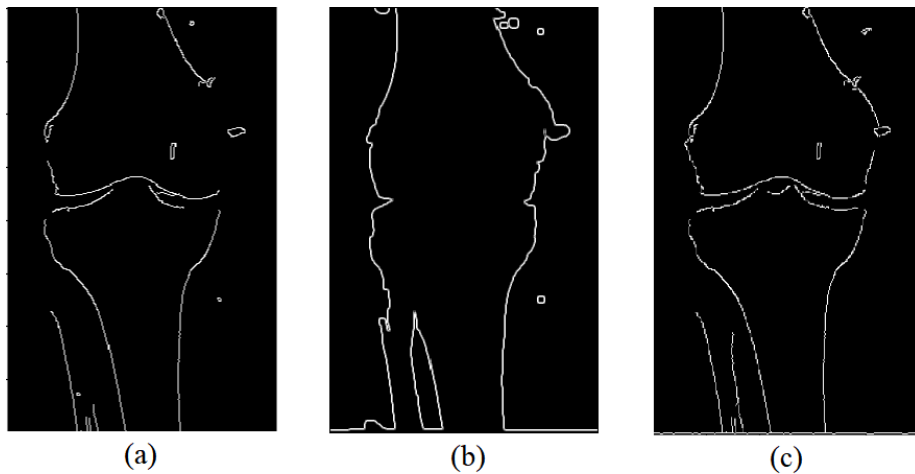


Figure 4.3: Edge Detection using (a) Sobel (b) Kirsch (c) Canny Operator

### 4.2.1 Canny Edge Detector

The most powerful among the edge-detection techniques is the Canny method. It differs from the other edge detection techniques as it uses two different thresholds (to detect strong and weak edges). It considers weak edges in the output only if they are connected to a strong edge. Therefore, the chances for being affected by noise are lesser in canny method, and is more likely to detect weak edges correctly.

The canny edge detector uses two threshold values (lower and upper) to decide whether a line is an edge or not. These values are determined by using the median of the image. Median is the average of the pixel values of the image which gives us the intensity information where pixel density is maximum. The optimal values of lower and upper threshold is calculated as below:

$$T_{low} = \max(0, (1 - \sigma) \times \text{median})$$

$$T_{high} = \min(255, (1 + \sigma) \times \text{median})$$

Here the value of  $\sigma$  is taken as 0.33 because in statistics along a distribution curve, values lying between 33% from the start and end of the curve are considered. Values lying beyond and below this curve as considered to be outliers.

### 4.2.2 Removal of small pixel areas

The output of edge detection may contain some very small white areas which are taken as an edge but are actually part of noise. A threshold value of area is taken here to decide whether the pixel area is part of noise or edge. All the connected pixels with the total area with less than this threshold value will be considered as noise and will be removed from the binary image. Binary image after removal of these small pixel areas is shown in Figure 4.4(b).

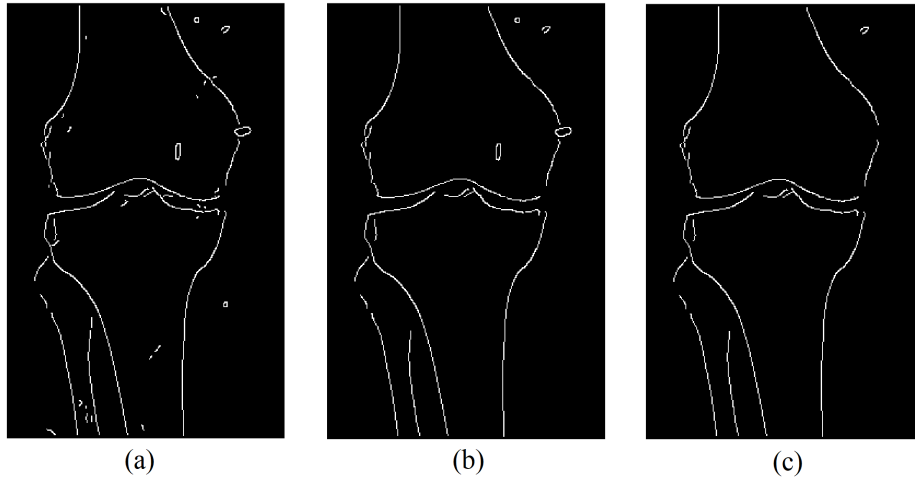


Figure 4.4: (a) Output of Edge Detection (b) Removal of small pixel areas (c) Removal of closed areas

### 4.2.3 Removal of closed areas

The image can also contains some objects with closed area. This may be a result of any unwanted object in between x-ray source and detector. These areas should also be removed from the image before starting with the chaining process. As discussed in section 3.3.1, closed areas can be detected by calculating the Euler number for that image. For every disconnected edge fragment, Euler

number is calculated and based on its value we find whether it is an closed shape or not. The shape is considered to be a closed loop only if the value of euler number is zero. Result after removal of closed loops is shown in Figure 4.4(c).

## 4.3 Chaining

After removal of small pixel areas from the image, there may be some edge fragments that may have discontinuities in between. These discontinuities must be eliminated in order to complete the edges. To join these fragments, we first need to detect the endpoints of the broken edges, followed by joining it to the nearest broken edge.

### 4.3.1 Detection of endpoints

To detect the endpoints of the edge fragments from the binary image, each edge fragment is first labeled in the image. MATLAB provides the function *bwlabel()* to label all the connected components separately. Now, for each of these labels we will find out two points which lie at end of the fragment. For doing so, *convhull()* is used to get the convex hull of the shape. Convex hull results all the minimum points by which a boundary can be created for the shape. We selects only those points out of all the points. In MATLAB, convex hull works only if the points are non-collinear. So before finding the convex hull for these points, we need to check if all the points lie on a line. If the points are collinear, all the points are used in the further step.

The convex hull generally gives 7-8 points for the shape. From these points, we need to find out the two, which are farthest in the shape. We initially calculated the distance between two points using the distance formula.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

It works fine only in the case of linear curves but detects incorrect endpoints if a curve is non-linear or is of a complex shape. The detection of incorrect endpoints for non-linear edge fragments has been shown in Figure 4.5(b).

The two endpoints of the curve should not be determined with the maximum coordinate distance but with the maximum number of white pixels which comes in between while reaching from one point to another. The distance from one white pixel to another through the white pixel area can be calculated by finding the distance map for each of the white pixel in the shape. Lets say for a point  $p_1$ , the maximum distance comes out to be  $d$ , with point  $p_2$ . We will assume this to be the maximum distance( $d_{max}$ ) for now. Now for the next point, distance map will be calculated again and the maximum distance( $d'_{max}$ ) will be determined. If the  $d'_{max}$  is greater than  $d_{max}$ , we will replace the maximum distance with the new one as:

$$d_{max} = d'_{max} (if d'_{max} > d_{max})$$

Along with the maximum distance, we also saves the two points which gave the

maximum distance. Once this is performed for all the points, the two points with the maximum distance are marked in the image as shown in Figure 4.5.

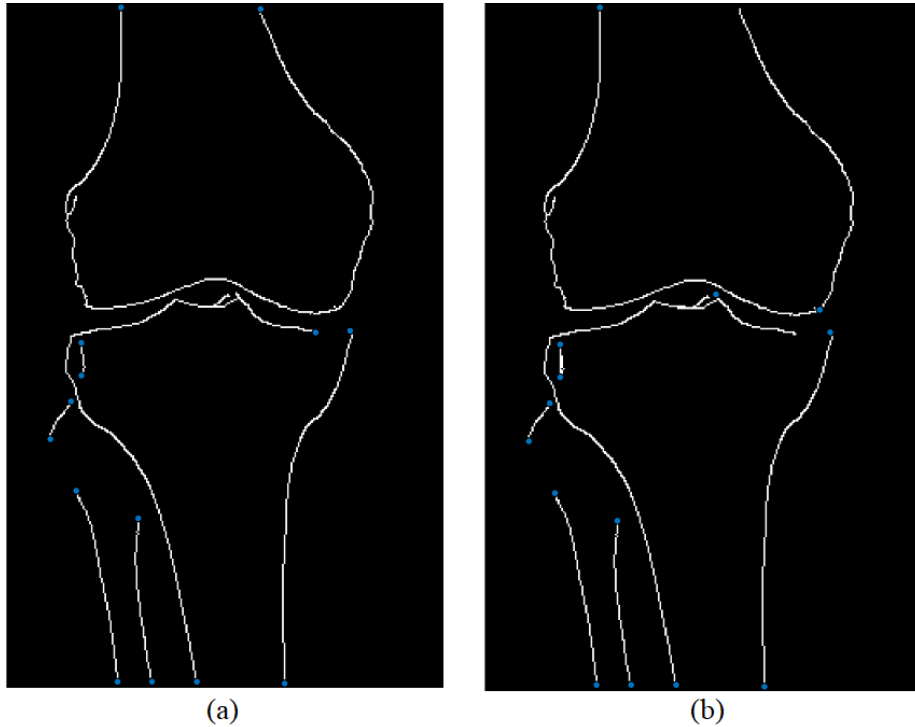


Figure 4.5: (a) Correct (b) Incorrect detection of endpoints of edge fragments

### 4.3.2 Connecting the endpoints

Once we have all the endpoints in the binary image, we can start connecting them with the other nearest point. Before doing this step, we select a maximum distance ( $r_{max}$ ) with which two endpoints could be connected. Lets say point  $P2(x2, y2)$  is the nearest endpoint for another point  $P1(x1, y1)$ . So before connecting them, we check if the distance between them is less than  $r_{max}$  or greater. The distance between the two points are calculated as:

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

If  $d < r_{max}$ , then we connect these two points using Bresenham's line drawing algorithm. Bresenham's algorithm is described in section 3.2.3. The value of  $r_{max}$  is increased iteratively until a good segmented output is received. The values taken here are 5,10,15,20,25 and the output with  $r = 0, 5, 20$  are shown in Figure 4.6.

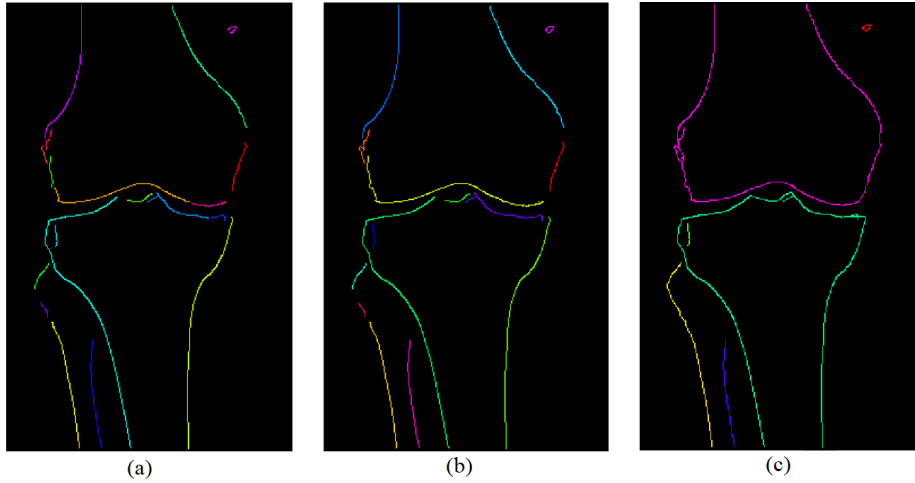


Figure 4.6: Connecting broken edges with pixel distance of 0, 5 & 20 pixels

In the figure, different edges are shown with separate colors. As the value of  $r$  is increased, more number of edge fragments are connected to each other and their numbers are reduced. At the end of this step, we get the segmented bone in the output along with some unwanted objects which again needs to be removed. These objects will be of smaller areas as compared to the bone. So we can again remove these pixels based on the total area. This time the area value will be greater than what we choose before as the pixel areas may have connected to nearby pixels, increasing the total area.

Figure 4.7 shows the connected components in the binary image before and after the removal of unwanted bone edges. Figure 4.7(a) shows the output of chaining process which is followed by removal of pixels with maximum area of 100 pixels shown in Figure 4.7(b), which is further followed by removing the pixels with maximum area of 200 pixels (Figure 4.7(c)).

Finally, we get the image with boundary of the targeted bone. The output of the segmentation along with the original image is shown in Figure ??.

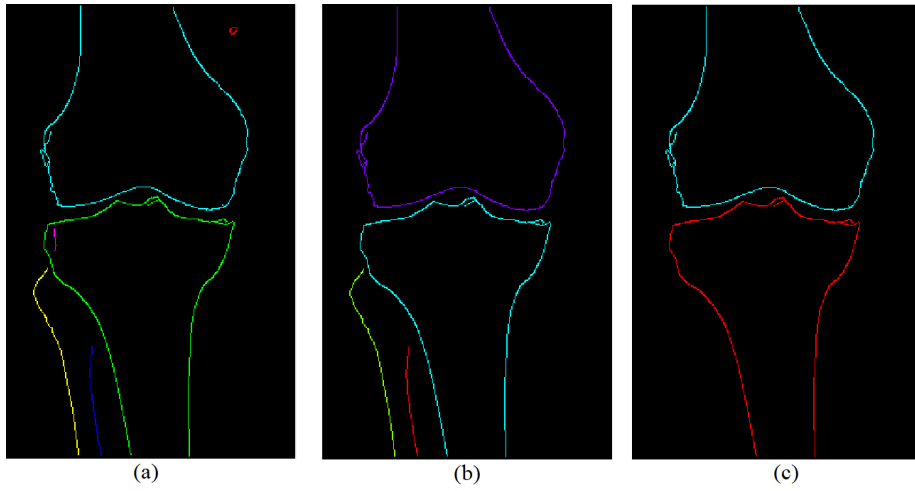


Figure 4.7: Removal of false edges from binary image

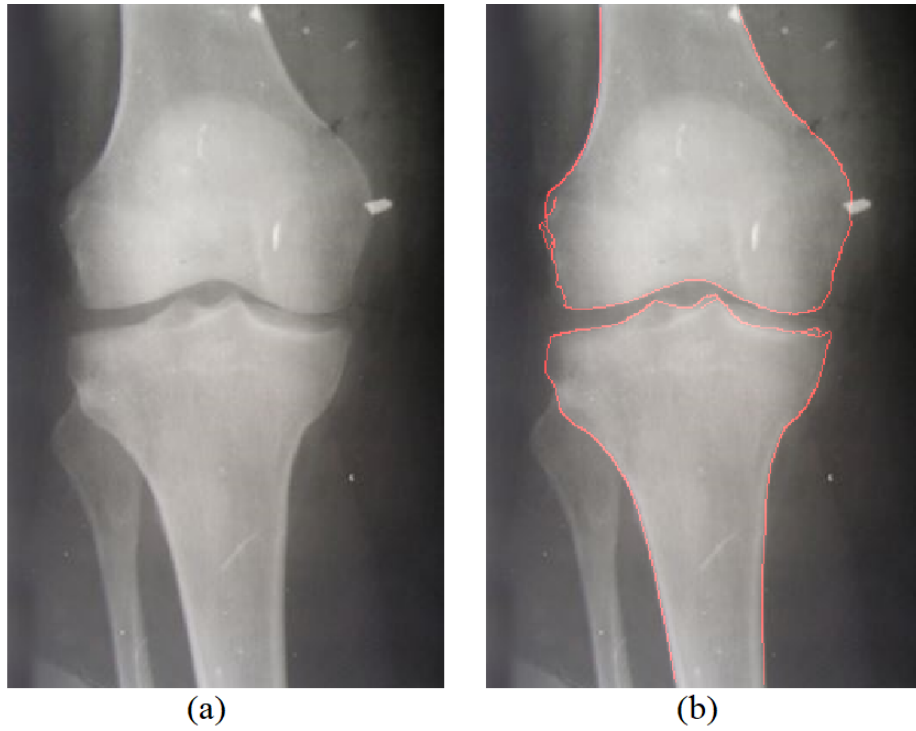


Figure 4.8: (a) Original Image (b) Segmented Image

# Chapter 5

## Results

We tested this segmentation algorithm for the X-ray images of lower part of the human body, in both lateral and frontal view. These images vary in quality, resolution, intensity and distortion of bone. We have tested In this section, we will show the results of segmented bones. The X-ray images were taken from the open databases on medical imaging [13], [14], [15] available online.

This segmentation algorithm worked on 75% of the images, while for 20% of the rest images, it was able to detect the bone partially. For rest of 5% images, the bones were detected incorrectly due to large intensity variation throughout the X-ray image.

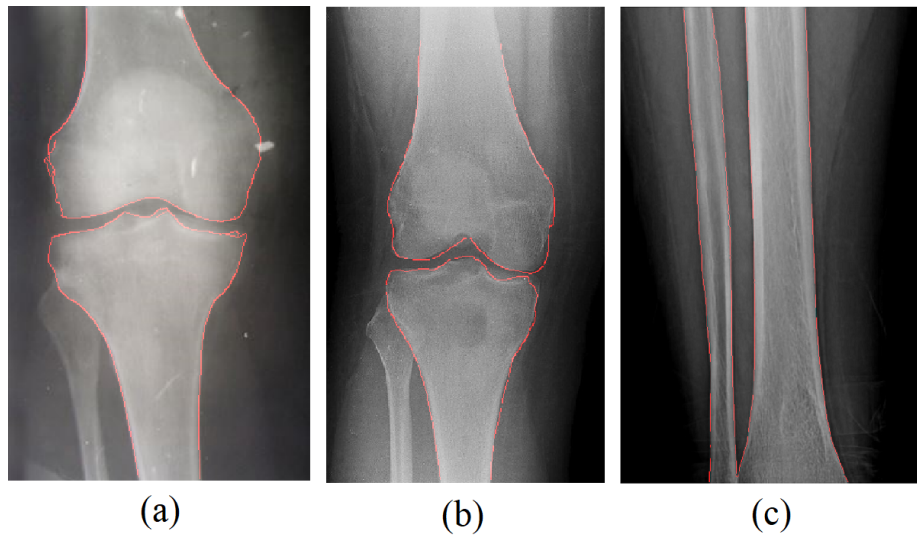


Figure 5.1: Examples of successful segmentation

Some of the successful sample results are shown in Figure 5.1. These tests were performed on a 2.40GHz Core i5 CPU. The time taken by this system for



segmentation of below X-ray images is presented in Table 5.1

Table 5.1: Time Complexity of the proposed method

Image	Resolution	Time Complexity(s)
Fig 5.1(a)	398×254	17.01
Fig 5.1(b)	512×512	29.99
Fig 5.2(d)	696×352	72.52
Fig 5.1(c)	698×447	156.73
Fig 5.2(e)	1024×408	539.82

For images with high intensity of white pixels, the detection of edges becomes difficult and may consider skin as the edge along with the bone. This may result to detection of partial or unsuccessful bone contour detection. Examples of these type of images are shown in Figure 5.2



Figure 5.2: Examples of (a) Partial & (b) Incorrect segmentation

## Chapter 6

# Conclusion & Future Scope

### 6.1 Conclusion

This paper proposes a method for the detection of bone contours from X-ray image by using edge detection technique followed by chaining. The edge detector used in this algorithm is Canny edge detector. The research paper[4] referenced here proposes Gradient Vector Flow (GVF) to detect the bone edges. GVF also gives similar result as of Canny edge detector, but needs to undergo through multiple stages before detection of edges like blurring, morphological dilation, gradient calculation, edge thinning etc. Results obtained by different edge detectors are shown in Figure 4.3. So Canny operator is preferred here over other edge detecting operators.

The detected edges are then applied to the chaining algorithm to eliminate the discontinuities between them. The chaining starts with detection of endpoints for each broken edges and mapping them to nearest endpoint of other edge. The chaining process completes when we have one single edge fragment for each bone. Also, some false detected edges are removed by filtering them by total area of connected edges. The connected edges with area less than a threshold value are removed from the final image. Finally, we have a closed boundary around the edges of the bone.

The description of algorithms for edge detection and chaining are described in chapter 2. Chapter 3 includes the implementation details of the above mentioned steps along with the output for each step. Sample outputs of successful as well as unsuccessful segmentation are shown in chapter 5 with their time complexity.

### 6.2 Future Scope

Segmentation of bone contours is the very initial stage for performing any surgery over the bone. The exact shape and size of the bone is determined using the bone contours. The results obtained using this algorithm can be used

as the input for other algorithms which needs the segmented bone as input. Few of the scopes of this project is described in this section.

As the x-ray images are highly nonuniform in intensity and texture, we must select different values for the parameters needed for edge detection technique. For canny edge detection, we need two threshold values which helps in deciding whether a pixel is edge or not based on its pixel value ranging from 0 to 255. After the detection of edge fragments, we select a maximum distance between two fragments of edges which can be connected to each other. As this parameter also varies based on the minimum and maximum distances between different edge fragments and their total numbers in the binary image. Finally, when we have connected our edges, we need to remove all those pixel areas which are lesser than a threshold value in order to remove the false edges from the image. Currently, all these values are given by the user and needs to be changed for each image until the best results are obtained. The manual input for these parameters can be avoided by training a model using Machine or Deep learning algorithms which will automatically select the parameter values based on the trained model and give the best result.

One approach can be made to detect the deformities and fractures in the bone. For detecting the deformity in the bone, we need to have a predefined model of the target bone. We can match the detected bone with this model to check the percentage of deformity in the bone. For the detection of fracture, we need to find the sharp curve in the bone contours. If any sharp curve is detected in the edges of bone, there is a possibility of fracture in it. The sharpness of the edge can be taken as an parameter to predict the severity of the fracture.

Another approach can be made for classifying the bone based on its shape and size. Every bone in human body has different anatomy which distinguishes itself from others. Similar to the previous one, if we have predefined bone models and the segmented bone, we can match the segmented bone with the defined models and predict which bone it is.

# Bibliography

- [1] Dion Monstavicus, *Elder Zucker Image Compression*, Stanford University, 1994.
- [2] Alexandru Telea, *An Image Inpainting Technique Based on the Fast-Marching Method*, ResearchGate, 2004.
- [3] Cristina Stolojescu-Crisan, Stefan Holban, *An Interactive X-Ray Image Segmentation Technique for Bone Extraction*, International Work-Conference on Bioinformatics and Biomedical Engineering, 2014.
- [4] Alexey Mikhaylichenko, Yana Demyanenko, and Elena Grushko, *Automatic Detection of Bone Contours in X-Ray Images*, AIST, 2016.
- [5] Ying Chen, Xianhe Ee1, Wee Kheng Leow, and Tet Sen Howe, *Automatic Extraction of Femur Contours from Hip X-ray Images*, Computer Vision for Biomedical Image Applications, 2005.
- [6] G. Behiels, D. Vandermeulen, F. Maes, P. Suetens, and P. Dewaele, *Active Shape Model-based Segmentation of Digital X-ray Images*, International Conference on Medical Image Computing and Computer-Assisted Intervention, 1999.
- [7] Xiao Dong, Miguel A. Gonzalez Ballester, Guoyan Zheng, *Automatic Extraction of Femur Contours from Calibrated X-Ray Images using Statistical Information*, Journal of Multimedia, 2007.
- [8] Moshe Aboud, Assaf B. Spanier, and Leo. Joskowicz, *Automatic Classification of Body Parts X-ray Images*, ResearchGate, 2018.
- [9] Shubhangi D.C, Raghavendra S.Chinchansoor, P.S Hiremath, *Edge Detection of Femur Bones in X-ray images – A comparative study of Edge Detectors*, ResearchGate, 2018.
- [10] Xicheng Liu, Jianzhong Xi, *An Algorithm for Improved Canny Adaptive Edge Detection in Image Processing*, Southwest Petroleum University, China, 2016

- [11] Walid Shahab, Hazem Al-Otum, and Farouq Al-Ghoul, *A Modified 2D Chain Code Algorithm for Object Segmentation and Contour Tracing*, International Arab Journal of Information Technology, 2009.
- [12] John Morris, *Finding the Convex Hull*, The University of Auckland, 2004.
- [13] Dr Mike Cadogan, *Radiology Image Database*, lifeinthefastlane.com, 2016
- [14] *Open Access Biomedical Image Search Engine*, OPENi.com
- [15] *Medical Imaging Challenges*, Radiopaedia.org
- [16] Rachmawan, *Canny Edge Detection*, Mathworks File Exchange
- [17] *Kirsch Operator*, Wikipedia.org
- [18] *Anatomical terms of location*, Wikipedia.org
- [19] *TeachMe Anatomy*, teachmeanatomy.info
- [20] Kunal Oberoi, *Hip Anatomy KSS Session*, Stryker Global Technology Center, 2019