X-ray Image Segmentation using Active Shape Models



Mayuresh Kulkarni

A dissertation submitted in partial fulfillment of the requirements for the degree of B.Sc. in Electrical Engineering at the University of Cape Town.

 $21 \ {\rm October} \ 2008$

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Bachelor of Science in Engineering at the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author

Cape Town 21 October 2008

Abstract

Image segmentation is an important stage in any image processing process. Regions of interest in the image are extracted from the image and are used to interpret the information in the image.

This study investigates the image segmentation of X-ray images and aims at separating the bone from the rest of the X-ray. Basic edge detection techniques and Active Shape Models are the image segmentation techniques analyzed. The performance of these methods is tested using error functions and best-fit curves. The process is modified to automate it to decrease human involvement. These are the initial steps of an automatic bone fracture detection algorithm.

Basic edge detection techniques consist of texture analysis and morphological operations on the image to find edges. In this thesis, the bone can be separated from the X-ray image using the bone boundary. The bone boundary can be found using edge detection techniques. The advantages of these techniques are examined and their drawbacks are explained. A study of these techniques shows their inadequacies in detecting the edges in certain bones, creating the need to find a robust and efficient way of finding bone boundaries.

Active Shape Models, presented in Cootes and Taylor [9, 10], is a method of finding a shape in an image. Active shape models are used to fit a shape, learnt from training images, to a test image. The algorithm is trained using X-ray images by manually selecting landmark points on the images. The shape of the bone is learnt using these images and then the model tried to fit the shape to a test image. Performance of these models is tested and variations on the model are studied. Interpretation of the results of experiments shows the best way to use Active Shape Models to segment X-ray images. It is proved that this technique extracts the bone in the X-ray effectively.

ETHICS FORM

Acknowledgements

I would like to thank the following people:

- Dr. Fred Nicolls for answering all my stupid questions patiently, and for always guiding me in the right direction. Also, thanks for cleaning up all the MATLAB code.
- My family for their encouragement and motivation during all my years of education. Without you, none of this would have been possible. Thanks for believing in me and helping me through the tough times.
- Mark and Nikki Ferris for being so much more than friends and guardians.
- Mark Pikker and Sello Phetla for all the fun we have had over the past 4 years. Also, thanks for all the group work and help in projects.
- Poovendren Govender, Clement Clementson, Nicholas Ip Cho and other inhabitants of the Blue Lab for the long nights of work and play.
- Kylie Hagan, Jason Suter and Ivan Serwada for introducing and helping me with Lyx and LateX.
- Ashish Bhatia, Nikhil Divekar, Joseph George and Ragesh Pillai for all the help and support over these years at UCT.
- All the friends who took the time to read my thesis, suggest changes and provide any help that I needed.

And most importantly, God for giving me the strength to tackle my problems, the perseverance to keep going and the character to enjoy the struggle.

Contents

D	eclar	ation			i
A	bstra	nct			ii
A	ckno	wledge	ements		iv
Ν	omer	nclatur	e		x
1	Intr	oducti	ion		1
	1.1	Introd	luction to Image Processing		1
	1.2	Introd	luction to Medical Imaging		1
	1.3	Descri	ption of the Problem		3
	1.4	X-ray	Data		3
	1.5	Struct	sure of the document		4
2	The	eory D	evelopment and Scope Definition		5
	2.1	Theor	y Development	•	5
		2.1.1	Image Segmentation		5
		2.1.2	Feature Extraction		6
		2.1.3	Classifiers and Pattern Recognition		7
		2.1.4	Thresholding and Error Classification		7
	2.2	Scope	Definition	•	9
3	Lite	erature	e Review and History		10
	3.1	Previo	ous Research	•	10
		3.1.1	Summary of Previous Research	•	10
		3.1.2	Common Limitations of the Previous Research		12
	3.2	Edge	Detection		12
		3.2.1	Sobel Edge Detector		13
		3.2.2	Prewitt Edge Detector		14

Con	clusio	ns	55	
\mathbf{Fut}	ure Re	search and Recommendations	53	
5.4	Overal	ll Interpretation and Optimization	52	
	5.3.2	Interpretation of results	51	
	5.3.1	Results	51	
5.3	Initiali	ization	51	
	5.2.2	Interpretation of results	50	
	5.2.1	Results	48	
5.2	Traini	ng Images	48	
	5.1.2	Interpretation of results	48	
	5.1.1	Results	46	
5.1	1 Landmark Points			
\mathbf{Res}	ults		45	
	4.5.3	Defining the error	43	
	4.5.2	Distance Transform	42	
	4.5.1	Manually landmarked images	42	
4.5	Model	Training and Evaluation	42	
4.4	Initiali	ization Problem	35	
4.3	Param	eters and Variations	33	
	4.2.3	Searching the test image	32	
	4.2.2	Generating shapes from the model	31	
	4.2.1	The ASM Model	30	
4.2	Active	Shape Models	29	
4.1	Shape	Models	26	
Act	ive Sha	ape Models	26	
3.5	Active	Shape Models	25	
3.4	Deform	nable Models	24	
<u> </u>	3.3.3	Filtering and Thresholding	21	
	3.3.2	Feature Recognition and Segmentation	19	
	3.3.1	Texture Analysis	18	
3.3	Image	Segmentation	18	
	3.2.4	Canny Edge Detector	16	
	3.2.3	Roberts' Edge Detector	16	
	3.3 3.4 3.5 Act 4.1 4.2 4.3 4.4 4.5 Fut 5.2 5.3 5.4 Fut	3.2.3 $3.2.4$ 3.3 3.4 $3.3.1$ $3.3.2$ $3.3.3$ 3.4 Deform 3.5 Active $Active Shape 4.1 Shape 4.2 Active 4.2.1 4.2.2 4.2.3 4.3 Param 4.4 Initial 4.5 Active 4.2.1 4.2.2 4.2.3 4.3 Param 4.4 Initial 4.5 Model 4.5.1 4.5.2 4.5.3 Results 5.1 Landn 5.1.1 5.2.1 5.1.1 5.1.2 5.1 Landn 5.1.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.1 5.2.2 5.3 5.3 5.3.1 5.3.1 5.3.1 5.3.2 5.4 5.3.1 5.3.1 5.3.2 5.4 5.3.1 5.3.2 5.4 5.3.1 5.3.2 5.4 5.3.1 5.3.2 5.4 5.3.1 5.3.2 5.4 5.3.1 5.3.2 5.4 5.3.1 5.3.2 5.4 5.3.1 5.3.1 5.3.2 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.3 5.$	3.2.3 Roberts' Edge Detector 3.2.4 Canny Edge Detector 3.3 Image Segmentation 3.3.1 Texture Analysis 3.3.2 Feature Recognition and Segmentation 3.3.3 Filtering and Thresholding 3.4 Deformable Models 3.5 Active Shape Models 4.1 Shape Models 4.2 Active Shape Models 4.1 Shape Models 4.2 Active Shape Models 4.1 The ASM Model 4.2.1 The ASM Model 4.2.2 Generating shapes from the model 4.2.3 Searching the test image 4.3 Parameters and Variations 4.4 Initialization Problem 4.5 Model Training and Evaluation 4.5.1 Manually landmarked images 4.5.2 Distance Transform 4.5.3 Defining the error 5.1 Landmark Points 5.1.1 Results 5.1 Interpretation of results 5.2 Interpretation of results 5.3 Initialization	

\mathbf{A}	A Detailed Results					
	A.1	Numerical Results	57			
	A.2	ASM output images	59			
Bi	Bibliography					

List of Figures

1.1	Medical Imaging $[17]$	2
1.2	X-ray data used for the thesis	4
2.1	Process flow diagram of a typical digital image processing system $~$	5
2.2	Basic Image segmentation [23]	6
2.3	Histogram of image $[23]$	8
2.4	Thresholding in images [23] \ldots \ldots \ldots \ldots \ldots \ldots	9
3.1	Basic Edge Detection [23]	13
3.2	Sobel Edge Detection	14
3.3	Prewitt Edge Detection	15
3.4	Roberts Edge Detector	16
3.5	Canny Edge Detector	17
3.6	Possible problems with edge detection techniques	18
3.7	Range and Standard Deviation filtering	19
3.8	Bone samples and standard deviation filtering	20
3.9	Flesh samples and standard deviation filtering $\ldots \ldots \ldots \ldots \ldots$	20
3.10	Plot of the means of the standard deviation filtering of 20 bone and	
	flesh samples	20
3.11	Feature Recognition and Segmentation method	21
3.12	Filtering and Thresholding method	22
3.13	Separation using filtering and thresholding $\ldots \ldots \ldots \ldots \ldots \ldots$	22
3.14	Plots of the sum of columns of images	23
3.15	Performance of ACM model [2]	24
4.1	Example of a shape	27
4.2	Aligning Shapes	29
4.3	Manually landmarked images	30
4.4	"whisker" profiles of training images	32
4.5	A sample image pyramid. (Picture taken by author)	33

4.6	Summary of the ASM	34
4.7	Poor initialization and Result	35
4.8	Improved initialization	36
4.9	Bone and its column sums (as shown in Section $3.3.3$)	38
4.10	Initialization methods	39
4.11	Poor initialization (example 1)	40
4.12	Poor initialization (example 2)	41
4.13	The solution images	42
4.14	The distance transform of an image	43
4.15	Sample Error Plots	44
5.1	Effects of landmark points on the error and computation time \ldots .	46
5.2	Final output of the ASM.	47
5.3	Effects of the number of training images on error and computation time	49
5.4	Final output of the ASM	50
5.5	Initialization results	52
A.1	ASM ouput images	60
A.2	ASM ouput images	61

Nomenclature

ROI — Region of Interest. The regions that need to be extracted from an image. The location of desired information in an image.

ASM — Active Shape Models. A statistical method used to segment images and in other applications.

ACM — Active Contour Models. A statistical method used to segment images and in other applications.

Training Images — The set of images used to train the Active Shape Model to learn the shape.

Test Image — The image that needs to be segmented using the shape learnt from training images.

Mean Shape — The average shape formulated using the shapes in the training images.

GVF — Gradient Vector Flow

MRF — Markov Random Fields

IGD — Intensity Gradient Density

GO — Gabor Orientation

Chapter 1

Introduction

1.1 Introduction to Image Processing

Image Processing is a form of signal processing where images and their properties can be used to gather and analyze information about the objects in the image. Digital image processing uses digital images and computer algorithms to enhance, manipulate or transform images to obtain the necessary information and make decisions accordingly.

Examples of digital image processing include improvements and analysis of the images of the Surveyor missions to the moon [15], magnetic resonance imaging scans of the brain and electronic face recognition packages. These techniques can be used to assist humans with complex tasks and make them easier. A detailed analysis of an X-ray can help a radiologist to decide whether a bone is fractured or not. Digital image processing can increase the credibility of the decisions made by humans.

1.2 Introduction to Medical Imaging

Image processing techniques have developed and are applied to various fields like space programs, aerial and satellite imagery and medicine [15]. Medical imaging is the set of digital image processing techniques that create and analyze images of the human body to assist doctors and medical scientists. In medicine, imaging is used for planning surgeries, X-ray imaging for bones, Magnetic resonance imaging, endoscopy and many other useful applications [31]. Digital X-ray imaging is used in this thesis project. Figure 1.1, shows the applications of digital imaging in medical imaging.



(a) MRI scanner





(c) Sagittal and

(d) transversal MRI brain slices

Figure 1.1: Medical Imaging [17]

Since Wilhelm Roentgen discovered X-rays in 1895 [14], X-ray technology has improved considerably. In medicine, X-rays help doctors to see inside a patient's body without surgery or any physical damage. X-rays can pass through solid objects without altering the physical state of the object because they have a small wavelength. So when this radiation is passed through a patient's body, objects of different density cast shadows of different intensities, resulting in black-and-white images. The bone, for example, will be shown in white as it is opaque and air will be shown in black. The other tissues in the body will be in gray. A detailed analysis of the bone structure can be performed using X-rays and any fractures can be detected. Conventionally, X-rays were taken using special photographic films using silver salts [28]. Digital X-rays can be taken using crystal photodiodes. Crystal photodiodes contain cadmium tungstate or bismuth germanate to capture light as electrical pulses. The signals are then converted from analogue to digital and can be viewed on computers. Digital X-rays, less expensive and are environmentally friendly [28].

A radiologist would look at the X-rays and determine if a bone was fractured or not. This system is time consuming and unreliable because the probability of a fractured bone is low. Some fractures are easy to detect and a system can be developed to automatically detect fractures. This will assist the doctors and radiologists in their work and will improve the accuracy of the results [28]. According to the observations of [27], only 11% of the femur X-rays were showing fractured bones. So the radiologist has to look at a lot of X-rays to find a fractured one. An algorithm to automatically detect bone fractures could help the radiologist to find the fractured

bones or at least confidently sort out the healthy ones. But no single algorithm can be used for the whole body because of the complexity of different bone structures. Even though a lot of research has been done in this field, there is no system that completely solves the problem [14]. This is because there are several complicated parts to this problem of fracture detection. Digital X-rays are very detailed and complicated to interpret. Bones have different sizes and can differ in characteristics from person to person. So finding a general method to locate the bone, and decide if its fractured or not, is a complex problem. Some of the main aspects to the problem of automatic bone fracture detection are bone orientation in the X-ray, extracting bone contour information, bone segmentation, extraction of relevant features.

1.3 Description of the Problem

This thesis investigates the different ways of separating a bone from an X-ray. Methods like edge detection and Active Shape Models are experimented with. The aim of this thesis is to find an efficient and reasonably fast way of separating the bone from the rest of the X-ray. The bone that was used for the analysis is the tibia bone. The tibia, also known as the shinbone or shankbone, is the larger and stronger of the two bones in the leg below the knee in vertebrates and connects the knee with the ankle bones. Details of the X-ray data used are provided in the next section.

1.4 X-ray Data

The X-ray data used in this thesis project consists of 12 digital X-ray images. These images were X-rays of the full body that were then segmented to get the X-ray of the tibia. The experiments discussed in the following chapters are conducted on these segmented images. The size of the images were reduced to increase the speed of the programs. All the images were resized to the same dimensions to ensure uniform processing on all of them. Figure 1.2, shows the full-body X-ray and the segmented and scaled tibia bone X-ray. The images shown in the thesis will be these scaled and segmented images of the tibia.



(a) Full-body X-ray

(b) Tibia X-ray

Figure 1.2: X-ray data used for the thesis.

1.5 Structure of the document

The thesis has been divided into six chapters. Chapter 1 introduces image processing and medical imaging. A detailed description of the problem is included in this chapter along with the X-ray data used. Chapter 2 explains the various stages involved in a generic image processing operation and defines the scope of this project. Background information about the topic and the methods used to solve the problem are compared in Chapter 3. In Chapter 4, Active Shape Models and the associated variables and parameters are explained. Chapter 5 reports the results of the experiments conducted on the X-ray data using the Active Shape Models described in the previous chapter. Conclusions and ideas for future research constitute Chapter 6. Appendices and Bibliography are included at the end. Nomenclature of all the important concepts is provided at the beginning of the thesis paper.

Chapter 2

Theory Development and Scope Definition

2.1 Theory Development

A typical digital image processing system consists of image segmentation, feature extraction, pattern recognition, thresholding and error classification. Image processing aims at extracting the necessary information from the image. The image needs to be reduced to certain defining characteristics and the analysis of these characteristics gives the relevant information. Figure 2.1 shows a process flow diagram of a typical digital image processing system, showing the sequence of the operations. Image segmentation is the main focus of this thesis. The other processes are briefly described for completeness and to inform the reader of the processes in the whole system.



Figure 2.1: Process flow diagram of a typical digital image processing system

2.1.1 Image Segmentation

Image segmentation is the process of extracting the regions of interest from an image. There are many operations to segment images and their usage depends on the nature of the region to be extracted. For example, if an image has strong edges, edge detection techniques can be used to partition the image into its components using those edges. Image segmentation is the central theme of this thesis and is done

using several techniques. Figure 2.2, shows how one of the coins can be separated from the image. Figure 2.2a. shows the original image and Figure 2.2b. highlights the boundary of one of the coins. These techniques are analyzed and the best technique to separate bones from X-rays is suggested.



(a) The original image

(b) Image segmentation

Figure 2.2: Basic Image segmentation [23]

When dealing with bone X-ray images, contour detection is an important step in image segmentation. According to [31], classical image segmentation and contour detection can be different. Contour detection algorithms extract the contour of objects whereas image segmentation separates homogeneous sections of the image. A detailed literature review and history of the image segmentation techniques used for different applications is given in Chapter 3.

2.1.2 Feature Extraction

Feature extraction is the process of reducing the segmented image into few numbers or sets of numbers that define the relevant features of the image. These features must be carefully chosen in such a way that they are a good representation of the image and encapsulate the necessary information. Some examples of features can be image properties like the mean, standard deviation, gradient and edges. Generally, a combination of features is used to generate a model for the images. Cross validation is done on the images to see which features represent the image well and those features are used. Features can sometimes be assigned weights to signify the importance of certain features. For example, the mean in a certain image may be given a weight of 0.9 because it is more important than the standard deviation which may have a weight of 0.3 assigned to it. Weights generally range from 0 to 1 and they define how important the features are. These features and their respective weights are then used on a test image to get the relevant information.

To classify the bone as fractured or not, [27] measures the neck-shaft angle from the

segmented femur contour as a feature. Texture features of the image such as Gabor orientation (GO), Markov Random Field (MRF) and intensity gradient direction (IGD) are used by [22] to generate a combination of classifiers to detect fractures in bones. These techniques are also used in [20] to look at femur fractures specifically. Best parameter values for the features can be found using various techniques.

2.1.3 Classifiers and Pattern Recognition

After the feature extraction stage, the features have to be analyzed and a pattern needs to be recognized. For example, the features mentioned above like the neck-shaft angle in a femur X-ray image need to be plotted. The patterns can be recognized if the neck-shaft angles of good femurs are different from those of fractured femurs. Classifiers like Bayesian classifiers and Support Vector Machines are used to classify features and find the best values for them. For example, [22] used a support vector machine called the Gini-SVM [22] and found the feature values for GO, MRF and IGD that gave the best performance overall. Clustering, nearest neighbour approaches can also be used for pattern recognition and classification of images. For example, the gradient vector of a healthy long bone X-ray may point in a certain direction that is very different to the gradient vector of a fractured long bone X-ray. So, by observing this fact, a bone in an unknown X-ray image can be classified as healthy or fractured using the gradient vector of the image.

2.1.4 Thresholding and Error Classification

Thresholding and Error Classification is the final stage in the digital image processing system. Thresholding an image is a simple technique and can be done at any stage in the process. It can be used at the start to reduce the noise in the image or it can be used to separate certain sections in an image that has distinct variations in pixel values. Thresholding is done by comparing the value of each pixel in an image and comparing it to a threshold. The image can be separated into regions or pixels that are greater or lesser than the threshold value. Multiple thresholds can be used to achieve thresholding with many levels. Otsu's method [21] is a way of automatically thresholding any image. Otsu's method uses the variance of the image to determine the threshold. Otsu's method finds this threshold value, t that minimizes the variance, $\sigma^2(t)$, within the two pixel set, one set with values greater than t and the other with values less than t. The variance $\sigma^2(t)$ can be calculated as

$$\sigma^{2}(t) = \eta_{1}(t) \,\sigma_{1}^{2}(t) + \eta_{2}(t) \,\sigma_{2}^{2}(t)$$

where $\sigma_1^2(t)$ and $\sigma_2^2(t)$ are the variances of the two pixel classes and $\eta_1(t)$ and $\eta_2(t)$ are the number of pixels in each class. Thresholding is used at different stages in this thesis. It is a simple and useful tool in image processing. The following figures show the effects of thresholding. Thresholding of an image can be done manually by using the histogram of the intensities in an image. It is difficult to threshold noisy images as the background intensity and the foreground intensity may not be distinctly separate. Figure 2.3, shows an example of an image and its histogram that has the pixel intensities on the horizontal axis and the number of pixels on the vertical axis.



Figure 2.3: Histogram of image [23]

Figure 2.3, shows the original image and the histogram of the image. Figure 2.4, shows thresholding done on the image displayed in 2.3. Figure 2.4a. is the original image, 2.4b. is the image using a threshold of 50. So the pixel values that are greater than 50 are set to 1 and those less than 50 are set to 0. As 0 and 1 are shown as black and white respectively, and as most of the image has pixel values greater than 50 (according to the histogram in 2.3b.), the image in 2.4b. is mostly white. The image in 2.4c. is the original image thresholded with a value of 150. Here the pixel values that are less than 150 are set to 1 and the ones greater than 150 are set to 0. This is the opposite of what is done in 2.4b. and is done to show that the grains of rice can be separated from the image in this way. The final image, Figure 2.4d., is a combination of two thresholds applied to the original image. So it is a combination of the images 2.4b. and 2.4c. This is multi-level thresholding as it has two threshold levels.



Figure 2.4: Thresholding in images [23]

2.2 Scope Definition

The previous section gave an overview of a typical digital image processing system and this section will define the scope of this thesis. This thesis focuses on the image segmentation of the X-ray of the tibia and aims at separating the bone from the X-ray. So the input image is the X-ray of the bone and the region of interest is the tibia. This thesis also aims at separating bones in general from X-rays, so techniques that work well for a particular kind of bone are mentioned but not given importance. A generic method that will work for all bones is investigated, and tested on the tibia.

Chapter 3

Literature Review and History

The first section in this chapter describes the work that is related to the topic. Many papers use the same image segmentation techniques for different problems. This section explains the methods, discussed in this thesis, used by researchers to solve similar problems. The subsequent section describes the workings of the common methods of image segmentation. These methods were investigated in this thesis and are also used in other papers. They include techniques like Active Shape Models, Active Contour/Snake Models, Texture analysis, edge detection and some methods that are only relevant for the X-ray data.

3.1 Previous Research

3.1.1 Summary of Previous Research

According to [14], compared to other areas in medical imaging, bone fracture detection is not well researched and published. Research has been done by the National University of Singapore to segment and detect fractures in femurs (the thigh bone). [27] uses modified Canny edge detector to detect the edges in femures to separate it from the X-ray. The X-rays were also segmented using Snakes or Active Contour Models (discussed in 3.4) and Gradient Vector Flow. According to the experiments done by [27], their algorithm achieves a classification with an accuracy of 94.5%. Canny edge detectors and Gradient Vector Flow is also used by [29] to find bones in X-rays. [31] proposes two methods to extract femur contours from X-rays. The first is a semi-automatic method which gives priority to reliability and accuracy. This method tries to fit a model of the femur contour to a femur in the X-ray. The second method is automatic and uses active contour models. This method breaks down the shape of the femur into a couple of parallel, or roughly parallel lines and a circle at the top representing the head of the femur. The method detects the strong edges in the circle and locates the turning point using the point of inflection in the second derivative of the image. Finally it optimizes the femur contour by applying shape constraints to the model.

Hough and Radon transforms are used by [14] to approximate the edges of long bones. [14] also uses clustering-based algorithms, also known as bi-level or localized thresholding methods and the global segmentation algorithms to segment X-rays. Clustering-based algorithms categorize each pixel of the image as either a part of the background or as a part of the object, hence the name bi-level thresholding, based on a specified threshold. Global segmentation algorithms take the whole image into consideration and sometimes work better than the clustering-based algorithms. Global segmentation algorithms include methods like edge detection, region extraction and deformable models (discussed in 3.4).

Active Contour Models, initially proposed by [19], fall under the class of deformable models and are used widely as an image segmentation tool. Active Contour Models are used to extract femur contours in X-ray images by [31], after doing edge detection on the image using a modified Canny filter. Gradient Vector Flow is also used by [31] to extract contours and the results are compared to that of the Active Contour Model. [3] uses an Active Contour Model with curvature constraints, to detect femur fractures, as the original Active Contour Model is susceptible to noise and other undesired edges. This method successfully extracts the femur contour with a small restriction on shape, size and orientation of the image.

Active Shape Models, introduced by Cootes and Taylor [9], is another widely used statistical model for image segmentation. Cootes and Taylor, and their colleagues [5, 6, 7, 11, 12, 10], released a series of papers that completed the definition of the original ASMs by modifying it, also called classical ASMs by [24]. These papers investigated the performance of the model with gray-level variation, different resolutions and made the model more flexible and adaptable. ASMs are used by [24] to detect facial features. Some modifications to the original model were suggested and experimented with. The relationships between landmark points, computing time and the number of images in the training data were observed for different sets of data. The results in this thesis are compared to the results in [24]. The work done in this thesis is similar to [24] as the same model is used for a different application. [18] and [1] analyzed the performance of ASMs using the aspects of the definition of the shape and the gray level analysis of grayscale images. The data used was facial data, from a face database and it was concluded that ASMs are an accurate way of modeling the shape and gray level appearance. It was observed that the model allows for flexibility while being constrained on the shape of the object to be segmented. This is relevant for the problem of bone segmentation as X-rays are grayscale and the structure and shape of bones can differ slightly. The flexibility of the model will be useful for separating bones from X-rays even though one tibia bone differs from another tibia bone.

The working mechanisms of the methods discussed above are explained in detail in

the following sections in this chapter. So this section gave an overview of the usage of the methods and the following sections will describe how these methods work.

3.1.2 Common Limitations of the Previous Research

As mentioned in previous chapters, bone segmentation and fracture detection are both complicated problems. There are many limitations and problems in the segmentation methods used. Some methods and models are too limited or constrained to match the bone accurately. Accuracy of results and computing time are conflicting variables.

It is observed in [14] that there is no automatic method of segmenting bones. [14] also recognizes the need for good initial conditions for Active Contour Models to produce a good segmentation of bones from X-rays. If the initial conditions are not good, the final results will be inaccurate. Manual definition of the initial conditions such as the scaling or orientation of the contour is needed, so the process is not automatic. [14] tries to detect fractures in long shaft bones using Computer Aided Design (CAD) techniques.

The tradeoff between automizing the algorithm and the accuracy of the results, using the Active Shape and Active Contour Models, is examined in [31]. If the model is made fully automatic, by estimating the initial conditions, the accuracy will be lower than when the initial conditions of the model are defined by user inputs. [31] implements both manual and automatic approaches and identifies that automatically segmenting bone structures from noisy X-ray images is a complex problem.

This thesis project tackles these limitations. The manual and automatic approaches are tried using Active Shape Models. The relationship between the size of the training set, computation time and error are studied.

3.2 Edge Detection

Edge detection falls under the category of feature detection of images which includes other methods like ridge detection, blob detection, interest point detection and scale space models. In digital imaging, edges are defined as a set of connected pixels that lie on the boundary between two regions in an image where the image intensity changes, formally known as discontinuities [15]. The pixels, or a set of pixels, that form the edge are generally of the same or close to the same intensities. Edge detection can be used to segment images with respect to these edges and display the edges separately [26][15]. Edge detection can be used in separating tibia bones from X-rays as bones have strong boundaries or edges. Figure 3.1, is an example of basic edge detection in images.



(a) The original image (b) Sobel edge detection (c) Canny edge detection

Figure 3.1: Basic Edge Detection [23]

3.2.1 Sobel Edge Detector

The Sobel operator, used to do the edge detection, calculates the gradient of the image intensity at each pixel. The gradient of a 2D image is a 2D vector with the partial horizontal and vertical derivatives as its components. The gradient vector can also be seen as a magnitude and an angle. If D_x and D_y are the derivatives in the x and y direction respectively, equations 3.1 and 3.2 show the magnitude and angle(direction) representation of the gradient vector, ∇D . It is a measure of the rate of change in an image, from light to dark pixel in case of grayscale images, at every point. At each point in the image, the direction of the gradient vector shows the direction of the largest increase in the intensity of the image while the magnitude of the gradient vector denotes the rate of change in that direction [15][26]. This implies that the result of the Sobel operator at an image point which is in a region of constant image intensity is a zero vector and at a point on an edge is a vector which points across the edge, from darker to brighter values.

$$\nabla D = \sqrt{D_x^2 + D_y^2} \tag{3.1}$$

$$\Theta = \arctan(D_y/D_x) \tag{3.2}$$

Mathematically, Sobel edge detection is implemented using two 3×3 convolution masks or kernels, one for horizontal direction and the other for vertical direction in an image, that approximate the derivative in the horizontal and vertical directions. The derivatives in the x and y directions are calculated by 2D convolution of the original image and the convolution masks. If A is the original image and D_x and D_y are the derivatives in the x and y direction respectively, equations 3.3 and 3.4 show how the directional derivatives are calculated [26]. The matrices are a representation of the convolution kernels that are used.

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A$$
(3.3)

$$D_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$
(3.4)

where * denotes a 2D convolution operation.

Figure 3.2, shows the original bone image and the Sobel filtered image. As seen in Figure 3.2b., the Sobel operator detects two edges, one of the flesh and the other of the bone. This is a problem that needs to be addressed and solved as only the bone boundary is needed.



Figure 3.2: Sobel Edge Detection

3.2.2 Prewitt Edge Detector

The Prewitt edge detector is similar to the Sobel detector because it also approximates the derivatives using convolution kernels to find the localized orientation of each pixel in an image. The convolution kernels used in Prewitt are different from those in Sobel. Prewitt is more prone to noise than Sobel as it does not give weighting to the current pixel while calculating the directional derivative at that point [15][26]. This is the reason why Sobel has a weight of 2 in the middle column and Prewitt has a 1 [26]. The equations 3.5 and 3.6 show the difference between the Prewitt and Sobel detectors by giving the kernels for Prewitt. The same variables, as in the Sobel case, are used. The kernels to calculate the directional derivatives are different.

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * A$$
(3.5)

$$D_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * A$$
(3.6)

where * denotes a 2D convolution operation.

Figure 3.3, shows the original bone image and the Prewitt filtered image. As seen in Figure 3.3b., the Prewitt filter gives a similar output as the Sobel operator by detecting two edges. Both techniques, Prewitt and Sobel, are not useful because the complete bone can't be separated from the X-ray as the edge boundaries are not continuous.



(a) The original image (b) Prewitt output

Figure 3.3: Prewitt Edge Detection

3.2.3 Roberts' Edge Detector

The Roberts edge detectors, also known as the Roberts' Cross operator, finds edges by calculating the sum of the squares of the differences between diagonally adjacent pixels [26][15]. So in simple terms, it calculates the magnitude between the pixel in question and its diagonally adjacent pixels. It is one of the oldest methods of edge detection and its performance decreases if the images are noisy. But this method is still used as it is simple, easy to implement and its faster than other methods. The implementation is done by convolving the input image with 2×2 kernels:

$$\left[\begin{array}{rrr}1&0\\0&-1\end{array}\right] \text{and} \left[\begin{array}{rrr}0&-1\\1&0\end{array}\right]$$

Figure 3.4, shows the original bone X-ray and the output of the Roberts' edge detector.



(a) The original image (b) Roberts' output

Figure 3.4: Roberts Edge Detector

3.2.4 Canny Edge Detector

Canny edge detector is considered as a very effective edge detecting technique as it detects faint edges even when the image is noisy. This is because in the beginning of the process, the data is convolved with a Gaussian filter. The Gaussian filtering results in a blurred image so the output of the filter does not depend on a single noisy pixel, also known as an outlier. Then the gradient of the image is calculated, same as in other filters like Sobel and Prewitt. Non-maximal suppression is applied after the gradient so that the pixels that are below a certain threshold are suppressed. A multi-level thresholding technique, same as the example in 2.4, involving two levels is then used on the data. If the pixel value is less than the lower threshold, then it is set to 0 and if its greater than the higher threshold then it is set to 1. If a pixel falls in between the two thresholds and is adjacent or diagonally adjacent to a high-value pixel, then it is set to 1. Otherwise it is set to 0 [26]. Figure 3.5, shows the X-ray image and the image after Canny edge detection.



Figure 3.5: Canny Edge Detector

The basic edge detection techniques are very basic and have some drawbacks. The first problem is that two edge boundaries are detected when only one of them is needed. The bone in the image needs to be separated, so only the boundary of the bone in required. Another level of edge detection and thresholding, something similar to 3.3.3, will be required to overcome this problem. As shown in the images, there are many other edges that are not relevant but are still detected. So it would be beneficial if the image were thresholded, before edge detection of any kind, to reduce noise. But noise is unpredictable and so a general solution may not be possible.

Another instance where these methods will not work is when there are other structures or elements in the image. For example, in Figure 3.6, the X-ray image has the fibula and the tibia bones and also a mark on the top left corner. The image in Figure 2.4b. clearly illustrates the possible problems with edge detection. Edge detection can't distinguish between the two bones or the mark so it detects all the edges in the image. This will get worse as the number of objects in the image grows. If these techniques don't work for a simple bone like the tibia, they will not be effective at all with complex bone structures like the rib-cage.

Due to these drawbacks, a more general, robust and effective solution needs to be experimented and established. The desired solution should be able to segment the bone out of the image even through noisy input images. It should also avoid detecting other components in the images like the skin boundaries or other bones.



(a) The original image (b) Edges in the image

Figure 3.6: Possible problems with edge detection techniques

3.3 Image Segmentation

3.3.1 Texture Analysis

Texture analysis attempts to use the texture of the image to analyze it. Texture analysis attempts to quantify the visual or other simple characteristics so that the image can be analyzed according to them [23]. For example, the visible properties of an image like the roughness or the smoothness can be converted into numbers that describe the pixel layout or brightness intensity in the region in question. In the bone segmentation problem, image processing using texture can be used as bones are expected to have more texture than the flesh.

Range filtering and standard deviation filtering were the texture analysis techniques used in this thesis. Range filtering calculates the local range of an image whereas standard deviation filtering calculates the local standard deviation [23]. In the case of range filtering, each pixel in the output image is the range, difference between the highest and the lowest pixels, of the corresponding input pixel and the pixels surrounding it. The standard deviation filter does a similar operation but it calculates the standard deviation of the pixels surrounding the corresponding pixel in the input image.

These operations should make the location of the bone clear as they operate on each pixel locally. Figure 3.7a. shows the original image and 3.7b. and 3.7c. show the output of the range and standard deviation filters respectively. As observed from the images, this technique works similar to the edge detection techniques because it detects the flesh and bone boundaries. A method that uses range and standard deviation filters and separates the bone and skin boundaries using thresholding is discussed in section 3.3.3.



(a) The original image (b) Range fitlered image (c) Std filtered image

Figure 3.7: Range and Standard Deviation filtering

3.3.2 Feature Recognition and Segmentation

The bone and the skin were expected to have different textures. This method is based on sampling the bone and the skin so that their properties can be studied and the bone can be separated from the X-ray using these properties. 20 samples of bone and skin each were taken from different places in different images and were studied. The bone had more texture than the skin and standard filtering displayed that. Figure 3.8a. and b. are examples of one of the bone samples and the standard deviation filtered output respectively. Figure 3.9a. and b. are examples of one of the flesh samples and the standard deviation filtered output respectively.



Figure 3.8: Bone samples and standard deviation filtering



Figure 3.9: Flesh samples and standard deviation filtering

The samples of bone and flesh were analyzed and a pattern that differentiated between them was observed. Figure 3.10, is a plot of the means of the standard deviation filtered outputs of the 20 samples. The bone samples are represented using the red plus signs and the flesh samples are represented using the blue circles. Even though there is no clear clustering the value of approximately 6000 can be used to separate bone and flesh.



Figure 3.10: Plot of the means of the standard deviation filtering of 20 bone and flesh samples

This method was tried as a bone separation technique and the original image and the output is shown in Figure 3.11. The performance of the method is dismal because it is not a robust method and is based on sampling and experimentation. The cutoff between bone and flesh properties is not well-defined in the graph as there is no

clear clustering of the two. So a small change in intensities in an image will result in that part of the image being wrongly classified as bone or flesh. The performance of this method is worse compared to that of the edge detection techniques and the standard deviation or range filtering.



Figure 3.11: Feature Recognition and Segmentation method

3.3.3 Filtering and Thresholding

This method is based on the data used in this thesis and was formulated while working with texture analysis techniques discussed above in section 3.3.1. It is observed that the X-ray images have two major boundaries, one on the skin and the other of the bone. In any ideal X-ray the bone boundary is brighter than the boundary of the skin. This can be used to separate the two and extract only the boundary of bone. Only the bone boundary is needed to isolate the bone from the image, so the other elements in the image are not needed. This method tries to overcome the problem presented by the edge detection techniques of detecting two edges. Figure 3.12, shows a process flow diagram of the process used to implement this technique.



Figure 3.12: Filtering and Thresholding method

The X-ray image is range filtered or standard deviation filtered. These filters are those mentioned in section 3.3.1 and both of them work well. Then the image is thresholded with a low threshold. This first threshold is low as it reduces the noise in the signal. The output of this stage is similar to the output of the edge detectors as it detects both the skin and the bone edges. This output is referred to as the mask as it is used to mask the original image. In the next stage of the process, the mask is multiplied point-by-point to the original image. The output of this stage shows the information in the original image at the edges in the mask, preserving the intensities of the pixels at the edges. A second thresholding stage is applied to separate the bone boundary as it is brighter than the skin boundary. This threshold is higher than the one in the first thresholding stage. Figure 3.13, shows the outputs at different stages of the process. The final image, to the extreme right, shows the separated bone in comparison to the input image to the extreme left.



Figure 3.13: Separation using filtering and thresholding

The working of this method can be explained with the graphs of the sum of columns in the different images. Figure 3.14b., shows the plot of the column-wise sum of the original image. This is expected as the image is mostly black on the sides and has a lot of bright content in the middle. So the plot is like a bell curve. The bone which is part of the bell curve needs to be separated. Figure 3.14c. shows the column-wise sum of the image after standard deviation filtering has been applied. This acts like a weighting function for the image. It shows the location of the relevant data on the image. The plot of the multiplication of the mask and the original image is shown in Figure 3.14d. The second threshold can be easily decided using this image. It clearly shows the location of the relevant data. The peaks in this plot are the boundaries of the bone, so the data at those locations in the original image is the bone boundary. Then the bone can be easily separated using the boundary to get the data on the boundary and between the two edges.



Figure 3.14: Plots of the sum of columns of images

Although this method will work well for most X-rays, it is very constrained. It assumes that the bone edges are distinctively brighter than the skin edge. This may not be the case if the bone and the skin edges have similar intensities. The technique works for tibia bones but it may not work for other bones. This is because the technique uses some properties like the orientation of the bone in the image that may be different for different bones. It also needs values of two thresholds which the user may have to estimate. These thresholds change depending on how noisy the image is and also the pixel distribution of the components of the image. So a technique that depends on the shape of the bone, and not on the properties of the

3.4 Deformable Models

The methods like edge detection, texture analysis and the filtering and thresholding method are all basic image segmentation methods. As observed before, these methods work well for ideal data under strict constrains. Deformable models are statistical modeling methods that try to lock on to a shape in the image, after the shape has been learnt from a set of training images.

Snakes [19], also called Active Snakes or Active Contour Models, are an example of deformable models. These iterative algorithms try to minimize the sum of the "external energy" and the "internal energy" in an image. The external energy is also called the image energy [27]. The internal energy consists of two terms, one representing the tension along the snake and the other represents the bending of the curve on the image. The external energy is related to the accurate overlay of the model to the image as it is related to weighting the edges of the image. The algorithm attempts to minimize the sum of these two energies using a set of Euler equations [27]. The snake does not perform well with images that have curves or sharp bends in them [24]. There are many modifications suggested and implemented to improve the performance of the model. [2] suggests a way to improve the initial guess of the model by determining a global minimum. The accuracy of the model depends on the initial guess which depends on a local minimum in the active contour energy. The method discussed in [2] combines segmentation and denoising models and devises a fast way of getting a good initial guess. Figure 3.15, shows the performance of the active contour model.



Figure 3.15: Performance of ACM model [2]

3.5 Active Shape Models

Active Shape Models work on shapes learnt from training images and then try to locate the shape in a test image. These statistical models don't need handcrafted models, but they need the training of images. ASMs extract critical information of the training images at the landmark points given by the user [24]. ASMs were introduced and modified with a series of papers. Many other authors have extended the model, and have used it for different applications. ASMs are described in detail in the next chapter. A few modifications are listed below.

In the classical ASM, Cootes and Taylor [9] used a shape model that was derived from single multivariate Gaussian distribution, but in [8], Cootes et al. experiment with a shape derived from a mixture of multivariate Gaussians [24]. Cootes et al. [4] also compare the performance of ASMs and Active Appearance Models (AAMs) and observe that AAMs need fewer landmark points than ASMs [24]. But ASMs are faster to train and experiment than AAMs.

Rogers and Graham [25] make the ASM more powerful by using advanced leastsquare techniques to minimize the deviation between the actual shape and the suggested shape [24]. The classical ASM may not work under certain circumstances as it takes outliers into considerations.

Van Ginneken et al. [30] do a classification analysis and their method automatically selects the optimum set of good descriptors for image profiles. The standard image profile search is replaced with a k-nearest-neighbour classification [24].

There are also many implementations of the ASM algorithm. The code used in this thesis is the modified version of the code by Ghassan Hamarneh. It is based on the research on ASMs and the extensions made by Hamarneh et al. [18, 1, 16, 17].
Chapter 4

Active Shape Models

This chapter describes the workings of a typical ASM. Although there are many extensions and modifications made, the basic ASM model work the same way. Cootes and Taylor [9], gives a complete description of the classical ASM. Section 4.1 introduces shapes and shape models in general. Section 4.2 describes the workings and the components of the ASM. The parameters and variations that affect the performance of the ASM are explained in Section 4.3. The experiments that are performed in this thesis to improve the performance of the model are also described in this section. The problem of initialization of the model in a test image is tackled in Section 4.4. Section 4.5 elaborates on the training of the ASM and the definition of an error function. The performance of the ASM on bone X-rays will be judged according to this error function.

4.1 Shape Models

A shape is a collection of points. As shown in Figure 4.1, a shape can be represented by a diagram showing the points or as a $n \times 2$ array where the *n* rows represent the number of points and the two columns represent the *x* and *y* co-ordinates of the points respectively. In this thesis and in the code used, a shape will be defined as a $2n \times 1$ vector where the *y* co-ordinates are enlisted after the *x* co-ordinates as shown in 4.1c. A shape is the basic block of any ASM as it stays the same even if it is scaled, rotated or translated. The lines connecting the points are not part of the shape but they are shown to make the shape and order of the points more clear [24].



Figure 4.1: Example of a shape

The distance between two points is the Euclidean distance between them. Equation 4.1, gives the formula for Euclidean distance between two points (x_1, y_1) and x_2, y_2 . The distance between two shapes can be defined as the distance between their corresponding points [24]. There are other ways of defining distances between two points like the Procrustes distance but in this thesis, the distance means the Euclidean distance.

$$\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \tag{4.1}$$

The centroid \overline{x} of a shape x can be defined as the mean of the point positions [24]. The centroid can be useful while aligning shapes or finding an automatic initialization technique (discussed in 4.4). The size of the shape is the root mean distance between the points and the centroid. This can be used in measuring the size of the test image which will help with the automatic initialization (discussed in 4.4).

The first stage in the ASM comprises of training images to learn the shape that has to be found in the test image. In the training stage, aligning different shapes is important to get the mean shape. Aligning shapes is trying to get all the shapes in the same orientation. This means that all the shapes must have approximately the same position in the X-Y plane or translation, the same size or scaling and the same rotation. In the bone detection example, the bone can be anywhere in an X-ray image, so the shape of the bone has to be learned after aligning the training bone shapes. The bones can be anywhere in the image, can be of different sizes (although they are the same type of bones) and can be slightly at an angle. These are common variations in X-rays caused due to the different orientations of the patient's body when the X-ray is being taken.

Algorithm 1 was used by [24] to align images. This method is effective and can be used in bone segmentation.

The similarity transform is a transform that does scaling, translation and rotation but does not affect the image in other ways. Equation 4.2 shows the similarity transform T that rotates the point (x, y) by θ , scales it by s and translates it by (x_t, y_t) [24]

Algorithm 1 Aligning shapes

input set of unaligned shapes

- 1. Choose a reference shape (usually the 1^{st} shape)
- 2. Translate each shape so that it is centered on the origin
- 3. Scale the reference shape to unit size. Call this shape $\overline{x_0}$, the initial mean shape.
- 4. repeat
 - (a) Align all shapes to the mean shape
 - (b) Recalculate the mean shape from the aligned shapes
 - (c) Constrain the current mean shape (align to $\overline{x_0}$, scale to unit size)
- 5. *until* convergence (i.e. mean shape does not change much)

output set of aligned shapes, and mean shape

$$T\begin{pmatrix} x\\ y \end{pmatrix} = \begin{pmatrix} x_t\\ y_t \end{pmatrix} + \begin{pmatrix} s\cos(\theta) & s\sin(\theta)\\ -s\sin(\theta) & s\cos(\theta) \end{pmatrix} \begin{pmatrix} x\\ y \end{pmatrix}$$
(4.2)

A shape model defines the allowable set of shapes. This is learnt from the variations of the training images. Figure 4.2a. shows the unaligned shapes models of the tibia learnt from the training images. Figure 4.2b. aligns the shapes using algorithm 1.



Figure 4.2: Aligning Shapes

4.2 Active Shape Models

The ASM has to be trained using training images. In this project, the tibia bone was separated from a full-body X-ray (as shown in 1.2) and then those images were re-sized to the same dimensions. This ensured uniformity in the quality of data being used. The training on the images was done by manually selecting landmarks. Landmarks were placed at approximately equal intervals and were distributed uniformly over the bone boundary. Such images are called hand annotated or manually landmarked training images.

Figure 4.3 shows the original image and the manually landmarked image for training. While performing tests using different number of landmark points, a subset of these landmarks points is chosen.

After the training images have been landmarked, the ASM produces two types of sub-models [24]. These are the profile model and the shape model.

1. The *profile model* analyzes the landmark points and stores the behaviour of the image around the landmark points. So during training, the algorithm learns the characteristics of the area around the landmark points and builds a profile model for each landmark point accordingly. When searching for the shape in the test image, the area near the tentative landmarks is examined and the



(a) The original im- (b) Manually landage marked image

Figure 4.3: Manually landmarked images

model moves the shape to an area that fits closely to the profile model. The tentative location of the landmarks is obtained from the suggested shape.

2. The *shape model* defines the permissible relative positions of landmarks. This introduces a constraint on the shape. So as the profile model tries to find the area in the test image that tries to fit the model, the shape model ensures that the mean shape is not changed. The profile model acts on individual landmarks whereas the shape acts globally on the image. So both the models try to correct each other until no further improvements in matching are possible.

4.2.1 The ASM Model

The aim of the model is to try to convert the shape proposed by the individual profiles into an allowable shape. So it tries to find the area in the image that closely matches the profiles of the individual landmarks, while keeping the overall shape constant.

The shape is learnt from manually landmarked training images. These images are aligned and a mean shape is formulated with the permissible variations in it [24],

$$\hat{x} = \overline{x} + \Phi b \tag{4.3}$$

where

 \hat{x} is the generated shape vector by the model.

 \overline{x} is the mean shape, the average of the aligned training shapes x_i , defined as

$$\overline{x} = \frac{1}{\eta_{shapes}} \sum_{i=1}^{\eta_{shapes}} x_i \tag{4.4}$$

 Φ is the eigenvector matrix of the covariance matrix S_s of the training image shape points

$$S_s = \frac{1}{\eta_{shapes} - 1} \sum_{i=1}^{\eta_{shapes}} (x_i - \overline{x}) (x_i - \overline{x})^T$$

$$(4.5)$$

Using principal components approach, the eigenvalues of the matrix are sorted from largest to smallest. A certain number of eigenvalues are retained according to their magnitude. The eigenvectors corresponding to these eigenvalues are kept. The other values are ignored.

4.2.2 Generating shapes from the model

As seen in Equation 4.3, different shapes can be generated by changing the value of b. The model is varied in height and width, finding optimum values for landmarks. Figure 4.4 shows the mean shape and its whisker profiles superimposed on the bone X-ray image. The points that are perpendicular to the model are called "whiskers" and they help the profile model in analyzing the area around the landmark points. The shape created by the landmark points are used for the shape model and the whisker profiles around the landmark points are used for the profile model. A profile and a covariance matrix is built for each landmark. It is assumed that the profiles are distributed as a multivariate Gaussian and so they can be described by their mean profile \overline{g} and the covariance matrix S_g [24].



Figure 4.4: "whisker" profiles of training images

4.2.3 Searching the test image

After the training is over, the shape is searched in the test image. The mean shape calculated from the training images is imposed on the image and the profiles around the landmark points are search and examined. The profiles are offset ± 3 pixels along the whisker, which is perpendicular to the shape, to get the accurate area that closely resembles the mean shape [24]. The distance between the test profile g and the mean profile \overline{g} is calculated using the Mahalanobis distance given by

$$Distance = (g - \overline{g})^T S_a^{-1} (g - \overline{g})$$

$$(4.6)$$

If the model is initialized correctly (discussed in 4.4), one of the profiles will have the lowest distance. This procedure is done for every landmark point and then the shape model confirms that the shape is the same as the mean shape. The shape model assures that the profile model has not changed the shape. If the shape model were not employed, the profile model may give the best profile results but the resulting shape may be completely different. So, as mentioned before, the two models restrict each other. A multi-resolution search is done to make the model more robust. This enables the model to be more accurate as it can lock on to the shape from further away. So the model searches over a series of different resolutions of the same image, called an image pyramid. The resolutions of the images can be set and changed in the algorithm [17, 24]. Figure 4.5 shows a sample image pyramid. The sizes of the images are given, relative to the first image. A general picture, and not a bone X-ray, was chosen to clearly show how a typical image pyramid looks.



Figure 4.5: A sample image pyramid. (Picture taken by author)

4.3 Parameters and Variations

The performance of the ASM can be enhanced using optimizing the parameters that it depends on. Number of landmark points and number of training images are investigated in this thesis.

The number of landmark points is an important variable that affects the ASM. The profile model of the ASM works with these landmark points to create profiles. So the position of landmark points is as important as the number of landmark points. In the training images, landmark points are equally spaced along the boundary of the bone. Images are landmarked with 60 points and subsets of these points are chosen to conduct experiments. The impact of the number of landmark points on computing time and the mean error (defined in Section 4.5) is tested by running the algorithm with a different number of landmarks. As the number of landmark points is increased it is expected that the computing time increases and the error decreases. The results are explained in Chapter 5.

A training set of images is used to train the ASM. As the number of training images increases, the model becomes more robust and intelligent. The computing time is expected to increase as it will take time to train and create profile models for each image. However, as the number of training images increases the mean profile and the model performs better so the error is expected to decrease. The model in this thesis has 12 images, 11 are used to train the ASM and 1 is used as a test image.

Figure 4.6 gives an overview of the ASM. Figure 4.6a. shows the unaligned shape learnt from the training images. Figure 4.6b. displays the aligned shapes. The

whisker profiles, of one of the training images, calculated by the profile model are shown in Figure 4.6c. The mean shape calculated from the training images is shown in Figure 4.6d. The final output of the ASM is shown in Figure 4.6e.





Figure 4.6: Summary of the ASM

The results are interpreted in Chapter 5 and are compared with the results in [24]. The data used in [24] was face data but comparing them with the results in this thesis is relevant because [24] also uses the ASM model.

4.4 Initialization Problem

The Active Shape Model locks on to the shape learnt from the training images into the test image. It creates a mean shape profile from all the training images using landmark points. But the ASM starts off where the mean shape is located, but it may not be near the bone on a test image. So the model needs to be initialized or started somewhere close to the bone boundary in the test image.

Experiments were conducted to see the effect of initialization on the error and the tracking of the shape. It was observed that if the initialization is poor, which means that the mean shape starts away from the bone in test X-ray, the model does not lock on to the bone. The shape and profile models fail to perform as the profile model looks for regions similar to those of the training images in the regions away from the bone. So it is unable to find the bone as it is looking in a different region altogether. The error increases considerably if the mean shape is 40-50 pixels away from the bone in the test image. Figure 4.7a. shows the initialization. The pink contour is the mean shape and it starts away from the bone, so the result is a poor tracking of the bone.



Figure 4.7: Poor initialization and Result

Figure 4.8 a. shows the initialization that has been changed to make it better. The pink contour on the right is the original initialization and it is changed to a better position to the left by visual inspection. The result shown in 4.8 b. shows how the improved initial position can give a good result.



Figure 4.8: Improved initialization

Now that the importance of a good initialization has been established, the different aspects of initialization have to be inspected. The aim of initialization is to place the mean shape close to the bone in the test image. The same variables were used in the similarity transform to align shapes, namely translation, rotation and scaling, are the variables used to initialize the model.

The code used in the thesis gives an option to manually select an initialization so the user can place the mean shape according to the bone in the test image. To make the algorithm more useful, an automatic initialization can be built into it. In the data used in this project, the tibia was separated from the X-ray and all the images were scaled to the same dimensions. So the scaling of the mean shape and the test image was not a problem. Also, the images were carefully segmented so that the tibia is vertically in the center, approximately. Chapter 6 gives suggestions to tackle problems of initialization in the vertical (y) direction and scaling.

The initialization in the horizontal (x) direction is tested. The idea is to find the bone boundaries and start the ASM at that location. Getting the boundary of the bone was the problem for which ASM was used, so its not easily solvable. This time only an approximation of the bone boundary needs to be made as the model will correct itself.

A simple solution is to detect edges of the image (discussed in Section 3.2) and initialize the model at several locations and test the performance. This is not an intelligent way of addressing the problem and will also increase the computing time. The method of filtering and thresholding (discussed in Section 3.3.3) can be modified and used to get a method to initialize the model. Figure 4.9, shows a bone image and the sum of the columns of the image. The peaks in the plot of the column sums is the location of bone boundaries. The plot of the sums can be used to locate the bone boundary and the mean shape can be placed accordingly. The sum of the columns is divided into groups of 50 and the maximum values in each group are separated. This ensures that out of two peaks that are close together, only one peak is considered. The red line shows the mean of the two peaks, hence showing the middle of the bone. The mean of the mean shape can be calculated using the mean of the rows and columns and the model can be placed in the position denoted by the red line in Figure 4.9 b.



Figure 4.9: Bone and its column sums (as shown in Section 3.3.3)

Figure 4.10a. shows the model initialized using edge detection techniques. The model is placed at many different locations close to the bone boundary. Figure 4.10b., on the other hand, illustrates the initialization using the method discussed above. The approximation to the bone boundary is accurate and the result of the ASM will be better if the model is initialized on the red contour. The blue contour,

in both images, displays the position of the mean shape without any alterations.



Figure 4.10: Initialization methods

Although this method seems to work correctly for certain kinds of images, its performance is not accurate all the time. For example, for images where the bone is not straight, the sum of the columns will not have distinctive peaks and so the detection of the boundaries of the bone will be difficult. The graph of the sum of columns will have more than one peak. Figure 4.11 shows an image where the initialization method does not work well as its expected to, because the plot of the column sums does not have two distinctive peaks as seen in Figure 4.11b. So the red contour which is the initialization computed by the method is an insignificant improvement on the blue contour which is the mean shape.



Figure 4.11: Poor initialization (example 1)

Another example where the method does not perform well is shown in Figure 4.12. The bone image is shown in Figure 4.12a. The plot of the sum of all the columns is shown in Figure 4.12b. It shows three distinctive peaks instead of two so the initialization (red contour) is worse compared to the blue contour (mean shape). So

this method will work for bones that are vertical in the bone images. Suggestions to improve this method or the initialization of images in general are given in Chapter 6.



Figure 4.12: Poor initialization (example 2)

4.5 Model Training and Evaluation

4.5.1 Manually landmarked images

X-ray data does not have an error function or a way to measure the performance of the ASM. Hence an error function needs to be defined to measure the performance. So, the bone in the X-ray images is manually segmented and those images are considered to be the "ideal" solution to the image segmentation problem. Although this is not a good definition for an error function, the performance of the algorithm can be tested according to it. So the output obtained from the ASM is compared to these images, which will be called the "solution" images in this thesis. Figure 4.13, shows the original image and the solution image derived from the original. The solution images are binary images that display the bone boundary in white while the background is set to black.



(a) Original image (b) Solution image

Figure 4.13: The solution images

4.5.2 Distance Transform

As the solution images are binary images, the distance transform can be used to estimate the goodness of fit of ASM output. If the matrix X_{in} defines a binary image, then for each pixel in X_{in} the distance transform, also known as Euclidean distance transform, assigns a number that is the distance between that pixel and the closest non-zero pixel. As the solution images are binary, the distance transform will contain the distances from all points to the part of the model close to them. So the pixels that defined the shape in the solution image will have a value of 0 in the distance transform matrix, as the distance from them to the nearest white pixel is 0. As the distance transform shows the distance between each pixel and the nearest non-zero pixel, and because the output matrix has the same dimensions as the input matrix it can be used to define an error function for the ASM output.



Figure 4.14: The distance transform of an image

4.5.3 Defining the error

The distance transform is used to formulate an error function. The error function compares the distance between the solution image and the ASM output. This can be done easily by checking the value of the distance transform matrix at the landmark points of the ASM output. This is another reason why landmark points should be chosen carefully. The error is calculated at the landmark points and then the mean of the error is taken. Experiments were done and the relationship between different parameters and the error function were found. The errors at each point can be plotted and the mean error can be used as a numerical indicator of performance. Some error plots are shown in Figure 4.15.

The aim of this thesis is to accurately separate the bone from the X-ray image and so a visual inspection of the ASM output will also be a performance measure. The visual inspection gives a rough idea of performance of the model.

This error function will work if the ASM output shape is close to the solution image shape. So if the model has been initialized correctly, the error function will work properly. Otherwise, the wrong edges will be closer to each other and the error



Figure 4.15: Sample Error Plots. (the title of each figure shows the number of landmark points used)

function will lose its meaning. This is another reason why visual inspection of the segmentation is important. So error and visual inspection of the output are both important for measuring the performance of the ASM.

Chapter 5

Results

This chapter evaluates the performance of the Active Shape Model (section 4.2) on X-ray images. The experiments done are briefly described and the results obtained from there are interpreted. Other methods, other than the ASM, were not successful in separating the bone and their performances have been evaluated in Chapter 3.

Before displaying the results of the ASM model, an overview of the performance of other methods is discussed in this paragraph. The edge detection techniques did not perform well as they tracked both the bone and the flesh boundary. The same problem was encountered with the texture analysis methods and the feature recognition and segmentation techniques. The method of filtering and thresholding works well for certain X-ray images. This method is based on the assumption that the X-ray brightness will be ideal, but this may not be the case as shown in Chapter 3. After implementing these methods and observing their performance, the need for a generic, robust and intelligent method is realized. Active Shape Model is one such method and is tested and evaluated.

The computation time is relative as it is measured using MATLAB on a 1.86GHz Core 2 Duo with 1GB RAM. There can be other implementations or languages that may be faster. But by comparing these relative times, a general idea about the correlation between the computation time and other variables can be studied.

5.1 Landmark Points

This section gives a description of the experiments conducted with landmark points and analyzes the results. Landmark points are the points that define the shape in the test and training images. The detailed results are logged into tables in the Appendix.

5.1.1 Results

Experiments were done using the same set of training images, but changing the number of landmark points defining the shape of the model. A large set of landmark points were manually assigned to images and a subset of these were chosen to do the tests to ensure uniformity of the tests. The effect of landmark points on the accuracy of the model and the computing time was recorded. The accuracy of the results is calculated using the error function defined in section 4.5.3 and also visual inspection. Figure 5.1 shows the relation between the landmark points, the time taken to run the ASM algorithm and the error. The results shown are for three different experiments and instances of the model being run.



Figure 5.1: Effects of landmark points on the error and computation time

Figure shows the final output of the ASM algorithm. These images show the tracking of the bone done by the program. These images show how the ASM managed to track the bone. So after getting a numerical measure of performance of the model in Figure 5.1, Figure 5.2 provides a visual indication of its performance.



(c) 70 points

Figure 5.2: Final output of the ASM. (The titles of individual images show the landmark points used)

5.1.2 Interpretation of results

It is expected that as the number of landmark points increases, the error will decrease and the computation time should increase. The experiments above reflect the expected results. In Figure 5.1a. the mean error decreases as the number of landmark points increase. According to Milborrow [24], who used ASM to detect facial features, the mean error decreases quadratically as the number of landmark points is increased. This is seen in the experiments conducted as the error goes down but it is not necessarily linear. This also may be because of the structure of the bone in the image. The tibia is a long bone without many curves and corners as compared to facial features like eyes or mouth. So the tibia may be easier to locate. But overall, the error decreases as the number of landmark points increases as the profile sub-model of the ASM has more data and becomes more robust.

The computation time, according to Milborrow [24], increases linearly as the number of landmark points increases. This is because as the number of landmark points increases the profile model, which calculates the whisker profiles for each landmark point, takes longer time to calculate the profiles. The greater the number of points the more the profiles. Figure 5.1b. reflects the fact that computation time increases linearly as the number of landmark images increases.

Even though increasing landmark points improves the accuracy of the model, as seen in Figure 5.2, it also increases the computation time. So for tibia X-ray images 30-35 landmark points was considered adequate for tracking the bone well. Figure 5.2b. shows the output of the ASM using 30 landmark points and it can be seen that the bone has been tracked well.

The results in the form of Figure 5.2 show that the tracking of the bone improves as the landmark points increase.

5.2 Training Images

The ASM model learns the shape using a set of training images. This section investigates the performance of the model with changing the number of training images.

5.2.1 Results

The model is run on the same test image and the number of landmark points is kept the same, but the number of training images is varied. The performance of the model is observed. Figure 5.3 shows the correlation between the number of training images, computation time and the mean error. Figure 5.4 gives an example of the improvement in the tracking of the bone with increasing the training set.



Figure 5.3: Effects of the number of training images on error and computation time



(a) 3 images

(b) 5 images



(c) 8 images

Figure 5.4: Final output of the ASM

5.2.2 Interpretation of results

It is expected that as the number of training images increases, the model becomes more "intelligent" as it has more data and so it performs well. It was observed in Figure 5.3a. that the graph of the mean error is not a smooth curve. This is because some of the images added to the model make the mean shape worse than it was before. This increases the error. But it is observed that the mean error is the lowest when the number of training images is the highest. This shows that the number of training images does affect the performance of the model. The error is low at the beginning when there are 2 or 3 training images, but the ASM model is not really a robust and intelligent statistical model at that stage as it has limited data. So even though Figure 5.3a. shows that the mean error is low with 2 training images, this result may not be true for other images.

The computation time curves, in Figure 5.3b., are approximately linear showing that the time increases as the number of training images increase. This is expected because, as the number of training images increases, the profile model expands as it has more images and more landmark points to generate profiles for. The number of terms in the calculations increase the computation time.

The images in Figure 5.4 show the improvement of ASM model with the increasing number of training images. In Chapter 6, a recommendation to have more training images is made to improve the performance of the model. So overall, the more the training images, the better the model will perform.

5.3 Initialization

Section 4.4 inspects the problem of initializing the model on a test image. The results in this section reflect the dependency of the performance on the initialization. The model was initialized at different locations and the performance is measured.

5.3.1 Results

Figure 5.5, shows the effects of initialization on the model. The y translation is not relevant as the images are of the tibia only and the changes in the y direction do not affect the model considerably. In the x direction, the model has been initialized relative to the mean shape and the performance is plotted in Figure 5.5.

5.3.2 Interpretation of results

The ASM model does not track the bone effectively if it is initialized away from the bone boundary. The purpose of these experiments was to find out an approximate value of how far away the initialization can be. Figure 5.5 shows that the error is minimum when the model is initialized close to the bone. The circle on each graph in the figure points to the minimum of the graph. This indicates the relative initializations where the error is the lowest. Hence it can be said that the distance



Figure 5.5: Initialization results

between bone boundary and the initial guess of the model should be between 20 and 40 pixels to get an effective tracking of the model. So any method that is used to improve the initialization or make it automatic should be within that range of the bone in the test image.

5.4 Overall Interpretation and Optimization

The results of the ASM are interpreted above and this section covers an overall interpretation and optimization of the results. So if this problem is investigated in the future, this section will provide a basic idea of how the parameter of the ASM interact with the performance.

The landmark points in the ASM are descriptors of the profile model. So increasing them will increase the accuracy of the ASM. More landmark points will lead to shorter interpoint distance and will result in a better profile model. So if one of the landmark points is not chosen correctly, others close to it will compensate for its inaccuracy. But as the number of landmark points is increased the computation time increases too. So an optimal value for the number of landmark points needs to be established. This value will ensure that the model is robust enough and also is fast. This value was found to be somewhere between 30 to 35 landmark points in this thesis.

The training images help the ASM to learn the shape it will track in the test image. The mean shape is calculated from these training images. So the model performs better when the number of training images increases. Also, the computation time increases as the training images increase because the landmark points increase too. This problem can be overcome by storing the trained images and using them on the test image, instead of training every time the algorithm is run.

So in general, the ASM model works better with more training images and landmark points. To run the algorithm faster, the number of training images and landmark points needs to be limited.

Chapter 6

Future Research and Recommendations

This thesis project covers many methods of separating the bone from the X-ray. These methods have varying degrees of accuracy. They have variables that affect each other and also certain images for which the performance of the method is inaccurate. The following are some recommendations for future research on this topic or related topics:

- Improving segmentation The images used in this thesis were segmented by hand and the tibia was separated from them. If this process is expected to be a part of an automatic bone fracture detection process, then the segmentation of the tibia (or any other bone in question) needs to be automatic. Some methods covered in [21] were impressive, but they were not implemented as they were not the main focus of this thesis.
- Larger dataset for training The performance of the ASM depends mainly on the amount of information it has. This defines how "intelligent" the model is and hence improves performance. The dataset used in this thesis was limited to 12 images. A larger dataset should be used to train the ASM model. This will ensure better performance as the mean shape generated by the model will be over a wide range of training images. So a dataset with more than 100 images is needed.
- Improving the error function The error function defined in this thesis can be improved. The error function was formulated by manually tracking the bone boundary. The error function can be made better by getting professional help in X-rays and understanding parameters that radiologists consider when they look for bone fractures or bones in general in X-rays. There are many manually landmarked image datasets for face images. A dataset like that for different types of bones can be very useful and will make the usage of the model easier.

- Initialization Problem The problem of initialization of the model is discussed in this thesis. This is a complex problem that can be simplified by estimating the distribution of data in an image. If a rough map of the data in an image can be derived, the model can then be started at that point. As mentioned in section 4.4, only an approximation of the bone boundary is needed to start the model. The x and y translation and rotation of the image data can be studied and various methods can be formulated to do these efficiently.
- Different kinds of profiles Milborrow [24] observes that there is more than one way of calculating profiles in the ASM. So the profile model can be formulated in different ways. Some of the ways are mentioned in 3. A detailed analysis of the performance, accuracy and speed of these methods can be performed and the best one can be chosen.
- *Different model(s)* Image segmentation is a complex problem and ASM is not the only solution. The bone segmentation problem can be tackled using other methods like Active Contour Models or other models.
- *Miscellaneous* Other improvements like the speed, implementation and userfriendliness can be improved. These were not looked into in this thesis because this project was just aimed at segmenting the bone from the X-ray. The performance of the methods was given priority over their optimization and improvement.

Chapter 7

Conclusions

The aim of this thesis is to separate the tibia bone from the rest of the X-ray. Many methods were used and different properties of X-ray images and the bone were utilized to solve the problem. The methods used were tested and their performance was analyzed with a set of 12 images. Even though this thesis used tibia images to test the methods, a broader aim of generalizing the separation process to any bone was kept in mind. Concluding remarks and an overview of the thesis are the contents of this chapter.

The basic image segmentation methods were used to separate the bone from the Xray. These methods included edge detection techniques, texture analysis, standard deviation filtering, range filtering and feature recognition. These methods worked to a certain degree, as they detected the bone boundary but they also detected the flesh boundary in the image. So another method based on texture analysis was experimented with to separate the bone and flesh. But, this method did not work well. The filtering and thresholding method was based on the assumption that the brightness of the X-rays is ideal, meaning the bone in the image is brighter than the flesh. But this did not work for certain images as the bone in the image was not vertical, so the brightness in some regions was different from other regions.

After these basic image segmentation techniques were tested, it was realized that a robust and generic method was needed to do the segmentation. Active Shape Models were chosen as they learn a shape from training images and track that shape in the test image. They ignore other structures in the image like other bones and noise, so they perform well and don't depend on specific properties of the bone or the X-ray. The ASM is a simple method that separated the bone from the X-ray. The parameters of the ASM were varied and the performance was evaluated.

The main variables of the ASM are the number of training images, landmark points and the initial condition. Having many training images makes the model more effective. The landmark points should be evenly spread along the shape as the profile model in the ASM uses them to calculate whisker profiles. The initial condition or the initialization of the model makes a difference to the performance of the model. Finding a good initialization automatically is a complex problem in itself. This was evaluated using the methods that were used to segment the bone before.

In conclusion, it can be said that the ASM is a simple way of separating the bone from the X-ray image. It is a complex model that works on the shape of the image to be tracked, but it is an effective and accurate way of image segmentation. This thesis has shown that ASMs can be used to track the bone boundary effectively and hence be used to segment the bone.

Appendix A

Detailed Results

A.1 Numerical Results

The appendix gives all the results. These also include the results of the model on the test images. Plots of these results may have been shown in the thesis paper, but these are the results. All the numbers have been rounded off to 2 decimal places.

Table A.1, displays the results of three tests. These tests examine the effect of different number of landmark points on the ASM output. The error and the computation time are recorded. The model was initialized appropriately and 2 training images were used.

	Test 1		Test 2		Test 3	
Landmarks	Mean Error	Time	Mean Error	Time	Mean Error	Time
20	35.52	86.46	31.10	86.60	44.94	87.82
25	31.48	92.75	28.79	90.50		
30	33.20	107.63	25.18	93.07	36.26	107.55
35	42.81	116.60	21.60	101.53		
40	35.17	124.04				
45	33.85	134.34				
50	30.70	136.75			21.31	167.81
55	29.68	144.54				

Table A.1: Landmark points in ASM model

Table A.2 shows the effect of the number of training images on the computation time and mean error. The model was initialized appropriately and 30 landmark points were used.

30 landmarks (test 1)			30 landmarks (test 2)		
Images	Mean error	Time	Images	Mean error	Time
2	45.53	99.04	2	20.03	79.80
3	29.22	117.70	3	28.17	109.34
4	39.46	154.06	4	27.68	133.56
5	30.48	162.63	5	35.57	188.81
6	26.10	180.94	6	32.31	198.17
7	30.74	198.83	7	27.74	214.012
8	27.03	221.92	8	29.77	228.57
9	26.62	229.99	9	28.01	258.16
10	29.24	250.22	10	23.52	258.11
11	25.55	280.77	11	18.95	284.48

35 landmarks (test 3)					
Images	Mean error	Time			
2	22.17	106.54			
3	34.78	130.62	$\boxed{\begin{array}{c c} 30 \text{ landmarks (test 4)} \end{array}}$		
4	30.66	148.21	Images	Mean error	Time
5	48.51	180.94	2	33.20	107.64
6	29.02	213.75	3	34.46	131.13
7	28.29	220.89	4	34.06	143.02
8	43.99	233.16	5	30.95	165.07
9	29.03	257.41			
10	22.17	262.31			
11	21.24	274.23			

Table A.2: Training images in ASM model

The effect of translation of the initial mean shape is shown in the following table. Table A.3 displays the results of the x translation and Table A.4 displays the results of the y translation. The experiments are done with different number of landmark points to see if there is any change in the output.

20 landmarks		30 land	marks	40 landmarks	
x-translation	Mean error	<i>x</i> -translation	Mean error	<i>x</i> -translation	Mean error
-40	32.13	-40	22.44	-40	34.55
-20	28.15	-20	20.68	-20	31.45
0	28.26	0	20.03	0	30.59
20	28.33	20	20.10	20	29.33
40	39.54	40	24.67	40	32.35

Table A.3: x translation in ASM model

y-translation	Mean error
-20	32.56
0	30.99
20	42.41

Table A.4: y translation in ASM model

A.2 ASM output images

Figure A.1 shows the results of the ASM model on the test images. Some of these images may be used in the main document, but they are displayed here for completeness. The title of each image shows the parameters used (lp is the number of landmark points, trimg is the number of training images used and initx and inity are the initializations used in the x and y directions). The initialization is done relative to the mean shape and 'right' is the positive x direction and 'down' is the positive y direction. All of the images are shown on the next page to make it easy to compare them.



(a) lp=30, trimg=2, initx = -40, (b) lp=70, trimg=2, initx = -40, inity = 0 inity = 0



(c) lp=10, trimg=2, initx = -40, (d) lp=25, trimg=5, initx = -40, inity = 5 inity = 0

Figure A.1: ASM ouput images



(a) lp=20, trimg=2, initx = -40, (b) lp=25, trimg=8, initx = -40, inity = 0 inity = 10



(c) lp=25, trimg=3, initx = -40, (d) lp=40, trimg=2, initx = -40, inity = 0 inity = 5

Figure A.2: ASM ouput images
Bibliography

- R. Abu-Gharbieh, G. Hamarneh, and T. Gustavsson. Review Active Shape Models - Part II: Image Search and Classification. Swedish Symposium on Image Analysis, pages 129–132, 1998.
- [2] X. Bresson. Fast Global Minimization of the Active Contour/Snake Model. Journal of Mathematical Imaging and Vision, 28(2):151–167, June 2007.
- [3] Y. Chen, X. Ee, W. Leow, and T. Howe. Automatic Extraction of Femur Contours from Hip X-Ray Images. Computer Vision for Biomedical Image Applications, Volume 3765/2005:200-209, November 2005.
- [4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Comparing Active Shape Models with Active Appearance Models. Proc. British Machine Vision Conference (ed T.Pridmore, D.Elliman), 1, 1999.
- [5] T. F. Cootes and C. J.Taylor. Active Shape Models. 3rd British Machine Vision Conference, pages 266–275, 1992.
- [6] T. F. Cootes, C. J.Taylor, D. Cooper, and J. Graham. Training Models of Shape from Sets of Examples. 3rd British Machine Vision Conference, pages 9-18, 1992.
- [7] T. F. Cootes and C. J. Taylor. Active Shape Model Search using Local Grey-Level Models: A Quantitative Evaluation. 4th British Machine Vision Conference, pages 639-648, 1993.
- [8] T. F. Cootes and C. J. Taylor. A Mixture Model for Representing Shape Variation. Image and Vision Computing, 17(8):567–574, 1999.
- [9] T. F. Cootes and C. J. Taylor. Technical Report: Statistical Models of Appearance for Computer Vision. Technical report, The University of Manchester School of Medicine, 2004.
- [10] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. A Trainable Method of Parametric Shape Description. 2nd British Machine Vision Conference, pages 54-61, 1991.

- [11] T. F. Cootes, C. J. Taylor, and A. Lanitis. Active Shape Models: Evaluation of a Multi-resolution Method for Improving Image Search. 5th British Machine Vision Conference, York:327–336, 1994.
- [12] T. F. Cootes, C. J. Taylor, A. Lanitis, D. H. Cooper, and J. Graham. Building and Using Flexible Models Incorporating Grey-level Information. 4th International Conference on Computer Vision, IEEE Computer Society Press:355-365, 1993.
- [13] E. Davies. Machine Vision, Theory Algorithms Practicalities. 1997.
- [14] M. Donnelley. Computer aided Long-bone Segmentation and Fracture Detection. PhD thesis, Flinders University of South Australia, January 2008.
- [15] R. Gonzalez and R. E. Woods. Digital Image Processing. Prentice Hall, New Jersey, ISBN 0201180758, 2 edition, 2002.
- [16] G. Hamarneh. Deformable Spatio-Temporal Shape Modeling. Technical Report 311L, Chalmers University of Technology, 1999.
- [17] G. Hamarneh. Towards Intelligent Deformable Models for Medical Image Analysis. Ph.d. thesis, Department of Signals and Systems, School of Electrical and Computer Engineering, Chalmers University of Technology, ISBN 91-7291-082-8, 2001.
- [18] G. Hamarneh, R. Abu-Gharbieh, and T. Gustavsson. Review Active Shape Models - Part I: Modeling Shape and Gray Level Variation. Swedish Symposium on Image Analysis, pages 125–128, 1998.
- [19] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. International Journal of Computer Vision, 1:321–331, 1987.
- [20] S. E. Lim, Y. Xing, Y. Chen, W. K. Leow, T. S. Howe, and M. A. Png. Detection of Femur and Radius Fractures in X-Ray Images. 2nd International Conference on Advances in Medical Signal and Information, pages 249–256, 2004.
- [21] M. Long. Segmentation of Body Regions and Airways in Full-Body Digital Paediatric X-Ray Images. Master's thesis, University of Cape Town, August 2008.
- [22] V. L. F. Lum, W. K. Leow, Y. Chen, T. S. Howe, and M. A. Png. Combining classifiers for bone fracture detection in X-ray Images.
- [23] MathWorks. MATLAB Help, 2008.
- [24] S. Milborrow. Locating Facial Features with Active Shape Models. Master's thesis, University of Cape Town, November 2007.

- [25] M.Rogers and J.Graham. Robust Active Shape Model Search. 7th European Conference on Computer Vision, 4(517-530), 2002.
- [26] A. Mushabe. Edge Detection Techniques for the Segmentation of Diamonds to Silhouettes. Master's thesis, University of Cape Town, October 2006.
- [27] T. T. Peng. Detection of Femur Fractures in X-ray images. PhD thesis, National University of Singapore, 2002.
- [28] D. Sullivan. Digital X Ray. Available : http://www.encyclopedia.com/doc/1G2-3400900158.html, 2004. [Accessed 27 September 2008].
- [29] T. P. Tian, Y. Chen, W. K. Leow, W. Hsu, T. S. Howe, and M. A. Png. Computing neck-shaft angle of femur for x-ray fracture detection. *International Conference on Computer Analysis of Images and Patterns*, 2003.
- [30] B. van Ginneken, A. F. Frangi, J. J. Stall, and B. ter Haar Romeny. Active Shape Model Segmentation with Optimal Features. *IEEE-TMI*, 21:924–933, 2002.
- [31] C. Ying. Model-Based Approach for Extracting Femur Contours in X-ray Images. Master's thesis, National University of Singapore, 2005.