MODEL DRIVEN SEGMENTATION AND THE DETECTION OF BONE FRACTURES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE,
FACULTY OF SCIENCE
AT THE UNIVERSITY OF CAPE TOWN
IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Otto-Carl Marte
November 2004

Supervised by
Dr. P Marais

Abstract

The introduction of lower dosage image acquisition devices and the increase in computational power means that there is an increased focus on producing diagnostic aids for the medical trauma environment.

The focus of this research is to explore whether geometric criteria can be used to detect bone fractures from Computed Tomography data. Conventional image processing of CT data is aimed at the production of simple iso-surfaces for surgical planning or diagnosis — such methods are not suitable for the automated detection of fractures. Our hypothesis is that through a model-based technique a triangulated surface representing the bone can be speedily and accurately produced. And, that there is sufficient structural information present that by examining the geometric structure of this representation we can accurately detect bone fractures.

In this dissertation we describe the algorithms and framework that we built to facilitate the detection of bone fractures and evaluate the validity of our approach.

A coarse-to-fine approach is used to create a polygonal representation of the bone from the volumetric data. Firstly, some image processing is completed on the volumetric data. Then, a global registration is completed to bring the bone model and the bone data into a rough alignment. This is followed by a local alignment to achieve an even closer alignment. Finally, a local deformation is used to produce a final segmentation of the bone. Results indicate that not only the resolution but also the inter-scan distance between CT scans play an important role in the accuracy of the segmentation.

The fracture detection uses a metric based on curvature to detect geometric inconsistencies. Cluster analysis is employed to classify these geometric inconsistencies into fractures. The metric is employed at varying spatial resolutions to detect the whole gamut of fractures from slight fractures to large severe fractures. Results indicate that our approach is successful at detecting the majority of fractures. Results also indicate that the accuracy of the bone segmentation plays a significant role in the success of the bone fracture detection.

Acknowledgments

I would like to thank my supervisor Dr Patrick Marais for his guidance, support and patience. He provided many helpful suggestions, was always understanding and provided valuable advice. I am grateful to Prof S. Beningfield, Head of the Radiology Department of the Groote Schuur Hospital, for providing the digitized CT Scans. Thanks go to Dr. Hough for helping with the acquisition of the Lindsay dataset.

I would like to thank African Medical Imaging (AMI) Inc. for their generous financial support, as well as the MRC/UCT Medical Imaging Research Unit under which this project falls.

I am especially grateful to my family. My late mother, Olga, for always believing in me and helping to make my dreams possible. My brother, Walter, for his friendship and enthusiasm. I would also like to thank all my friends who helped me relax and kept me sane. I would like to thank my girlfriend, Magdeldt, for her constant support and love, and especially for her patience. Lastly, I would like to thank my colleagues at the CVC lab for many interesting discussions, their friendship and providing an enjoyable atmosphere in which to work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The field of medical imaging provides an excellent test-bed for new algorithms in computer vision and image processing. Medical data is information rich (many bits per pixel) and often degraded due to patient movement or inherent limitations imposed by the imaging technology.

Increases in computational power and technological innovations mean that there has been a marked increase in the production of diagnostic tools for medical professionals. These diagnostic tools are particularly effective at repetitive rule-based tasks and reducing diagnosis errors due to fatigue. The increase in computational power also means that techniques that were in the past unsuitable for the trauma setting due to performance constraints, can now be valuable tools in this critical environment.

The introduction of low dosage image acquisition devices, like Computed Tomography (CT) machines, means that scans can be administered that expose patients to less radiation and are so less detrimental to their health. The added effect of this development is that CT scans are used to diagnose a broader range of symptoms than they were traditionally used for.

1.1 Aims

Working with medical data produced from a CT scanner presents interesting challenges. Imaging artifacts introduced by the CT acquisition device and patient movement result in image degradation. Furthermore, CT scans represent a sampled version of continuous signal, and are thus an approximation. All these factors contribute to the fact that CT scans represent imperfect information. Challenges lie in devising methods by which valuable information can be gleaned from this approximated data.

Dealing with the large information sets represented by CT scans also represent their own demands. Manipulation of copious data places strain on computational resources and ways should be

devised to best utilise these limited resources.

This thesis investigates whether geometric criteria are a valid metric for diagnosing bone fractures from volumetric data. We argue that there is sufficient structural information present in a polygonal surface representation built from a Computed Tomography (CT) scan that we are able to detect inconsistencies in the integrity of the bone structure by geometric means.

As the detection of bone fractures can serve as a useful aid to radiologists in a trauma setting, these techniques must be able to complete their work speedily. The extraction of a surface model from CT data is usually a time consuming process. Computational intensive techniques that produce accurate surface representation from CT data are therefore not suitable for our needs. To fulfill our requirements a technique that trades some accuracy for a speed improvement is preferable. The challenge lies in developing a technique that is able to retain sufficient structural information so that break detection is possible, but is still sufficiently quick to be of use in a trauma environment.

Bone fractures range from so-called "hairline" fractures to large severe fractures. Developing a fracture detection framework that is able to detect this broad range of bone fractures represents a demanding challenge. Techniques must be developed that are able to detect geometric inconsistencies at different scales.

We argue that sufficient structural information can be speedily extracted from a CT scan and that this information can be used for the detection of bone fracture. We also believe that the surface normal of this extracted surface representation is a good approximation of the curvature and that this curvature approximation can be used to detect geometric inconsistencies on the surface of a polygonal representation of a bone extracted from volumetric data.

It is hoped that the algorithms developed can be of use as an aid to radiologists in a trauma setting.

1.2 Framework for Bone Fracture Detection

The framework for bone fracture detection consists of a two stage process. Firstly, a polygonal surface representation of a bone from volumetric data (in our case a Computed Tomography scan) is constructed. This process is termed the bone segmentation phase. Secondly, the surface normal of this generated polygonal representation is used to assess the fidelity of the bone. This is termed the fracture detection phase.

1.2.1 Bone Segmentation

The purpose of the bone segmentation phase is to speedily fashion an accurate polygonal representation of the volumetric data to serve as input to the fracture detection phase. The polygonal

representation should approximate the volumetric data as closely as possible. Not only should the segmentation be accurate, but it should be constructed rapidly as bone fracture detection can occur in a trauma setting.

We chose a model-based segmentation as the basic structure of a bone remains uniform across race, age and gender. This means that we built a template for a particular bone that is always used to help in the segmentation of that particular bone. The advantage of using such a template is that we are able to guide the segmentation process with foreknowledge of the expected shape of the bone. In an ambiguity we are able to refer to the model in order to correctly segment the bone.

We employ a coarse-to-fine approach to complete the segmentation of the bone from the volumetric data. Firstly, some image processing is completed on the volumetric data. Then, an alignment technique is used to bring the bone template and the bone data from the CT scan into a rough alignment. This is followed by a global alignment phase to achieve an even closer alignment. Finally, a local deformation was used to produce a final segmentation of the bone. The local deformation is necessary to account for the local structural differences between race, age and gender.

A number of existing segmentation techniques were combined to produce an accurate and speedy surface representation of a bone from CT data. New global alignment techniques were also developed and evaluated in the bone segmentation framework.

1.2.2 Fracture Detection

The purpose of the fracture detection phase is to use geometric criteria on the input produced from the bone segmentation phase to detect bone fractures. We developed a novel technique to detect bone fractures using structural information. We chose a metric based on curvature to measure the geometric at a certain location on the polygonal bone model. As we wish to detect the whole gamut of fractures from slight fractures to large severe fracture we need to measure geometry at different scales. We built a hierarchy of curvature measures to measure curvature at different spatial resolutions. We term this a *curvature hierarchy*.

A curvature hierarchy is built for the bone model template and this is compared to a curvature hierarchy that is built for the segmented bone model. Those locations where there is significant differences in curvature between the template curvature hierarchy and the segmented bone model curvature hierarchy are marked as geometric inconsistencies.

A cluster analysis algorithm is used to group the geometric inconsistencies into separate bone fractures. These detected bone fracture are then visually marked for easy identification.

1.3 Outline of this Dissertation

In this section we present an outline of what will be covered in this dissertation.

Chapter 2 In this chapter we describe the background work that is relevant to our research. We start out by giving a brief introduction on the different bone fractures that can occur. This is followed by a description of the Computed Tomography acquisition process. We then describe previous work that has been completed in the segmentation of volumetric data by means of deformable models. Hereafter, we provide background on some of the most important techniques that we have used in our work.

Chapter 3 This chapter describes the construction of a polygonal surface representation of the CT bone scan by means of a template. We explain in detail the steps involved in our coarse-to-fine approach which involves the synthesis of a number of techniques.

Chapter 4 Bone fracture detection is the subject of this chapter. We describe the construction of the curvature hierarchy and how this is used to detect geometric inconsistencies between a template and a polygonal bone model. We then describe how we used cluster analysis to classify these geometric inconsistencies into specific bone fractures.

Chapter 5 In this chapter we discuss the results obtained from our algorithms. Qualitative and quantitative techniques have been used to evaluate the successes and shortcomings of the bone segmentation and bone fracture detection phases of the framework. This analysis was performed against computer generated data as well as medical CT data.

Chapter 6 In the final chapter we describe the most important conclusions that can be drawn from this dissertation. We point out improvements that can be made to the algorithms developed and what work lies ahead that can be pursued in the future.

Chapter 2

Background

2.1 Introduction

The purpose of this chapter is to provide the reader with the background information necessary for an understanding of the work that is covered in this dissertation. It provides a broad overview of the terminology and important concepts that form the foundations upon which this thesis is built. The accompanying references should provide in depth coverage of the topics, if this is required.

As the aim of this dissertation is to detect bone fractures in Computed Tomography (CT) scans, we need to have some idea of the types of bone fractures that commonly occur. This information is provided in Section 2.2. Section 2.3 gives some background on Computed Tomography. This section describes X-Rays, CT, Limited Angle Tomography as well as some preliminary work on a CT simulator.

Previous work in the field of Deformable Models is covered in Section 2.4. This places in context the deformable model used in our research. We look at physics based, statistically based and geometrically based deformable models, as well as various other deformable models. In Section 2.5 we describe some of the techniques employed in this thesis. Finally, in Section 2.6 we describe the most important terminology used throughout this dissertation.

2.2 Fractures

Fractures, or bone-breaks as they are commonly called, occur as a result of stress to the human skeleton — See Figure 1. This results in damage to the structure of individual bones. The localized damage or *lesion* of the bone can take a variety of forms depending on the nature of the induced stress. As the aim of this research is to help in the detection of fractures, it is important that we become familiar with the various types of fractures that occur.

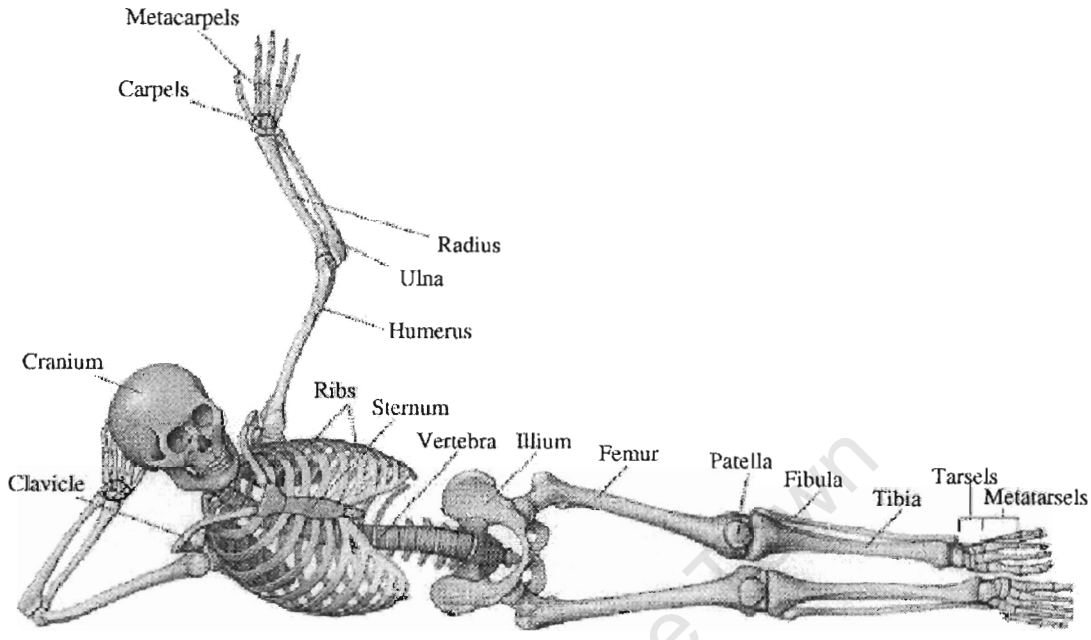


Figure 1: **The Human Skeleton:** The most important bones found in the human skeleton.

2.2.1 Complete Fracture

Complete fractures are those where the induced stress results in a bone shearing/breaking into one or more segments. The structure of the bone is severely altered and there is a significant change in the geometry of the bone. Complete fractures can be classified into several types:

1. **Closed/Simple Fracture:** The damaged bone does not penetrate the skin. The fracture is internal to the body. See Figure 2(a).
2. **Open/Compound Fracture:** The trauma to the bone results in the bone penetrating the skin. See Figure 2(b).
3. **Comminuted Fracture:** The induced stress results in the bone breaking into numerous shards or segments. See Figure 2(c).
4. **Noncomminuted Fracture:** The bone is separated into two pieces. See Figure 2(d).
5. **Chip/Corner Fracture:** A small chip or piece is separated from the bone. See Figure 2(e).
6. **Pathological Fracture:** A smooth break of the bone after the bone has been weakened by disease. See Figure 2(f).

2.2.2 Incomplete Fracture

In the case of incomplete fractures, the structure of the bone is predominantly maintained. However, there are deformations at specific local areas of the bone. The detection of these sort of fractures is far more complex than complete fractures, since the geometry of the bone may be only slightly altered. The accurate detection of these fractures is further hampered by the fact that bones that have healed from a fracture can resemble some of these skeletal lesions.

Incomplete fractures include:

1. **Impaction Fracture: Depressed Fracture** The surface of the bone is forced inwards due to the stress applied.
2. **Impaction Fracture: Compressed Fracture** The damage to the bone results in a decrease in the size of the bone.
3. **Avulsion Fracture:** A portion of the bone is pulled away forcefully by the muscle or ligament. See Figure 2(g).
4. **Greenstick/Hickory Stick Fracture:**Trauma results in a bending of one side of the bone. This fractures is common among children under the age of 10 and heals without complication. See Figure 2(h).
5. **Torus/Buckling Fracture:** Stress to the bone results in a buckling of the bone. The resulting damage is a bulging of the bone in the area affected. See Figure 2(i).
6. **Stress/Fatigue Fracture:** This fracture is a result of repetitive stress to the bone. This fracture commonly occurs to bones in the feet of long distance runners.

Impaction fractures result in a subtle white line being visible on the X-Ray in the area of bone damage. This makes the detection of impaction fractures extremely challenging.

An avulsion fracture results in no significant damage to the bone. In order to detect inconsistencies it is therefore important to examine the geometrical interdependence of bones in local areas of the skeleton.

2.3 Computed Tomography

Tomography is defined as the process of constructing a cross-section of an object from projections created by transmission (X-ray) or reflection data (Magnetic Resonance) taken at different angles. Tomography can be performed using a number of different modalities. These include Magnetic

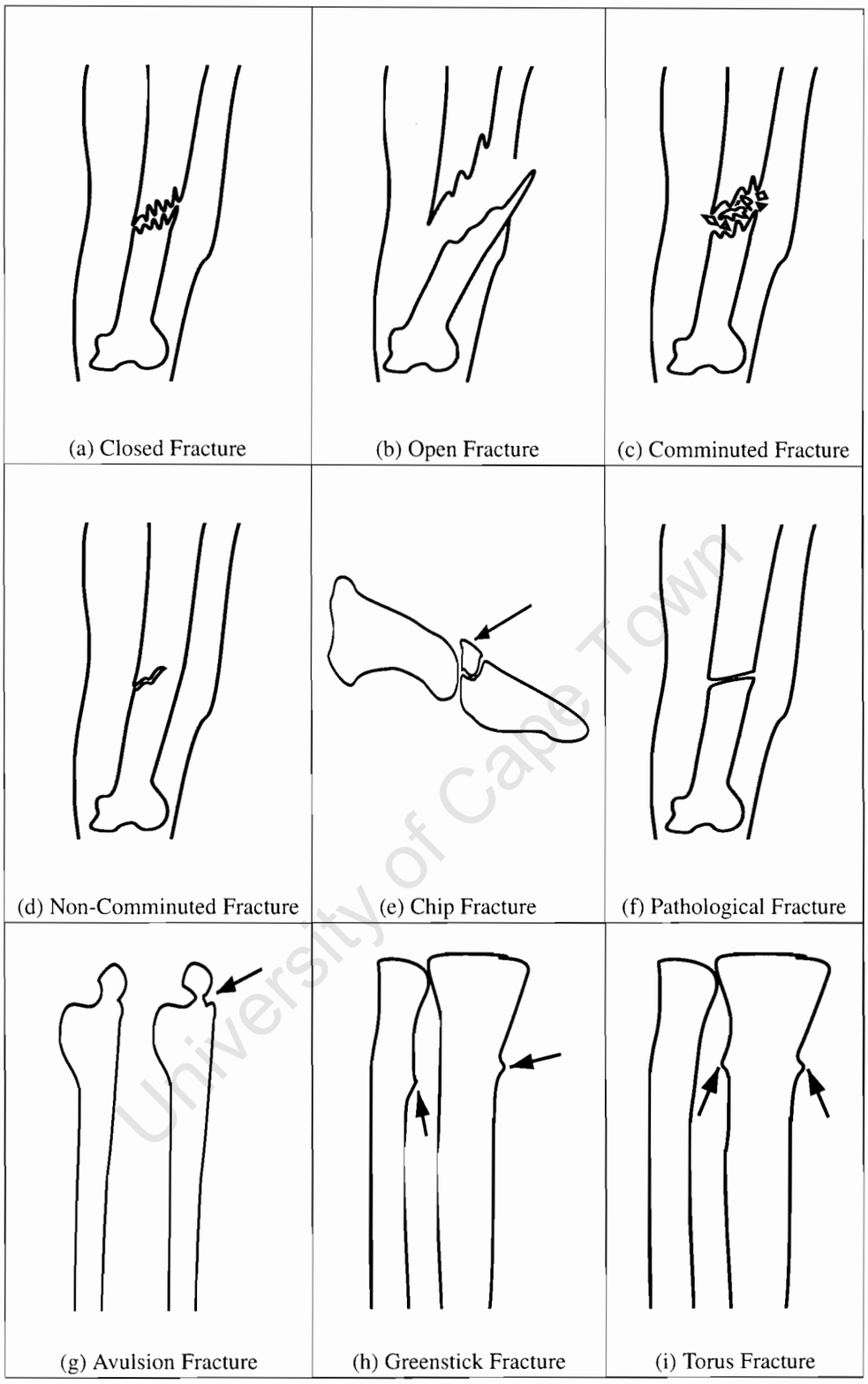


Figure 2: **Examples of Bone Fractures:** Pictorial examples of fractures that commonly occur.

Resonance Imaging (MRI), X-Ray Computed Tomography (CT) and Positron Emission Tomography (PET). Although tomography has helped solve problems in other disciplines one of which is engineering, where it has been used for the non-destructive testing of mechanical objects such as turbine blades, airplane wings, its chief use has been in the medical profession. Here, tomography has been used to great effect, as internal organs can be examined accurately, unobtrusively and with relative safety to the patient. We concentrate our attention on the medical use of X-Ray Computed Tomography.

An X-Ray is a high-energy photon. These high-energy light waves are produced by an X-ray tube and directed toward a set of Charge Coupled Devices (CCD's) in a parallel or fan-beam manner. The X-rays are attenuated by the materials that they pass through on their way towards the CCD. The density of the materials determine the amount of attenuation and scatter of the X-ray. Highly dense matter will result in a large attenuation and therefore a reduction in the amplitude of the X-ray. A CCD is a collection of light-sensitive diodes (photosites) that convert light (photons) into electrical charges (electrons). The higher the amplitude of the wave the greater is the electrical charge produced. A high electrical charge is measured as a low intensity value in image space i.e close to black, as the wave did not pass through dense material and was not attenuated. Collimators, which are thin tubes, ensure that a CCD measures mostly rays aimed at it, and not scattered rays.

In Computed Tomography (CT) the X-Ray Tube is rotated and translated about the object we are examining. X-rays are taken at different angles and positions surrounding the object (see Figure 3). Algorithms exist to reconstruct a volumetric object by combining the X-ray images taken from different positions in space.

CT volumetric data is exported in a common vendor independent format called DICOM. The header of the DICOM file provides detailed information regarding the acquisition process. This includes imaging information, physical information and machine specific information. For example, the header of the DICOM file tells us the spatial resolution of an image slice and the physical distance between successive image slices.

This research forms part of a project that is constructing a Low-Dosage Limited Angle Computed Tomography machine. The Limited Angle refers to:

- Either a restriction on the angular range of the CT. For example, traditional CT acquisition is over 180 degrees. A LAT acquisition might be limited to 100 degrees. or,
- A reduction in the angular sampling rate. For example, traditional CT acquires data every 0.5 degrees, whereas a LAT system might acquire data at 3 degree intervals.

The low-dosage and limited angle nature of the CT machine means that the image quality could well be degraded.

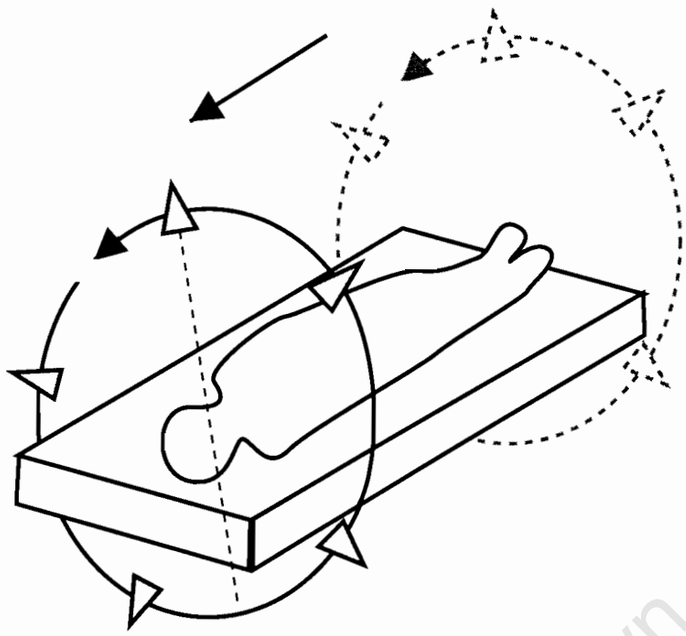


Figure 3: **The CT Process:** A CT Scan consists of X-Ray images taken at different positions and orientations.

As a preliminary to this research a software CT simulator was constructed to gain an understanding of the CT acquisition process (see Figure 5). It was also hoped that the CT Simulator could highlight some of the image artifacts that occur with Limited Angle Computed Tomography.

Filtered back projection was the algorithm used for reconstructing the volume. A number of different filters [48](Ram-Lak, Shepp-Logan,Cosine, Hamming and Hanning) were used. In order to simulate Limited Angle Computed Tomography the angular projection range and the number of projections used could be varied.

The reconstruction process consists of first constructing a sinogram (Radon transform) of a volumetric dataset. A single pixel in a sinogram represents the line integral of an X-ray at a particular position in space. Then, the sinogram is filtered using one of the above mentioned filters. Finally, the volumetric dataset is reconstructed by back projecting the filtered sinogram data.

Figure 4 shows various limited angle CT reconstructions produced by the software CT simulator using the Shepp-Logan filter. Note how a reduction in the angular range of the scan affects the quality of the reconstruction drastically, while a reduction in the angular increment has a less severe effect.

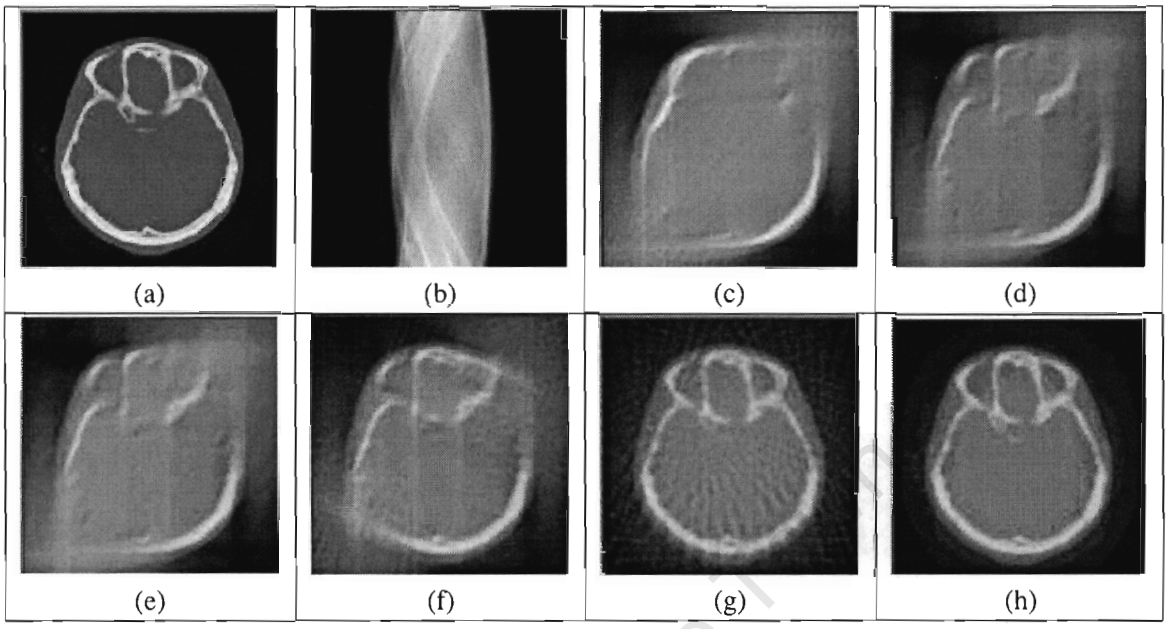


Figure 4: **CT Reconstructions produced by the CT Simulator:** (a) The input volumetric slice. (b) The sinogram produced from the input volume slice. Filtered back projection reconstructions produced by the simulator for CT Scans taken: (c) over a 90 degree range at a 3 degree angular interval (d) over a 90 degree range at a 1.5 degree angular interval (e) over a 90 degree range at a 1 degree angular interval (f) over a 120 degree range at a 4 degree angular interval (g) over a 180 degree range at a 6 degree angular interval (h) over a 180 degree range at a 1 degree angular interval

2.4 Deformable Models for Segmentation

Deformable models present a robust approach to the analysis and segmentation of medical images. A high-level model is constructed which encapsulates the shape of the biological structure under consideration. This model is a flexible representative that can identify all the varied members of a specific class of shapes.

Low level image processing tools (edge detection, morphological operators) predominantly make use of local information (e.g the image gradient between neighbour pixels is used to detect the edges in an image) in order to infer the structure of shapes. Deformable models allow global information to guide the segmentation process. *A priori* knowledge of the object under inspection can then be used to guide the recovery of the geometry of the shape.

It is often the case that low level image processing tools fail in their task when there is significant noise present. Manual (user) intervention is then required to rectify the erroneous output.

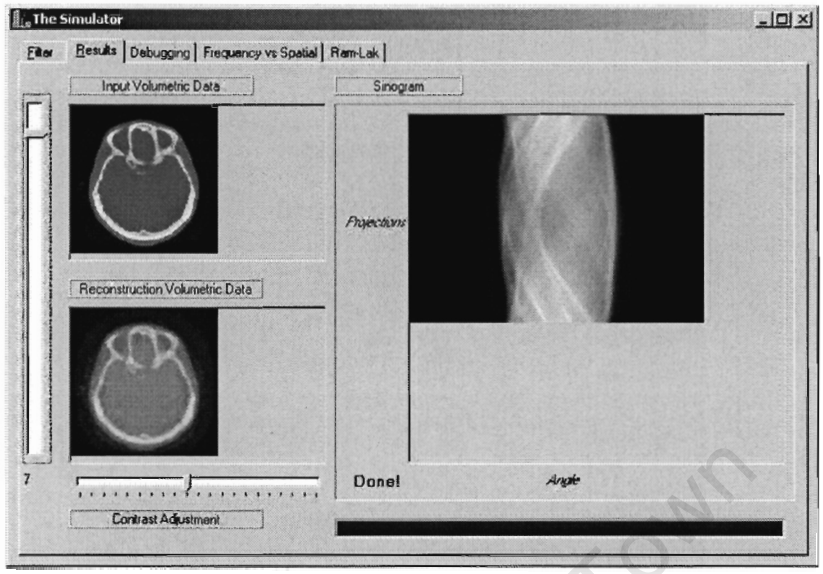


Figure 5: **CT Simulator:** A screenshot of the CT Simulator application

Global information concerning the geometry of the object is especially useful when there is noise present in the data. The uncertainty regarding the structure of the shape imposed by the presence of noise can be reduced by the global knowledge provided by deformable models. As there will be noise present in the LAT CT datasets that we will be working with, this feature of deformable models is especially attractive.

Deformable models combine the bottom up image cues provided by low-level image processing with top-down high level shape knowledge to facilitate the segmentation of anatomical structures. The deformation of an object may be modeled in terms of physics (forces acting on a shape), geometry (shape transforms), statistics (incorporating learned shape variation) and approximation theory (for matching/fitting of reference and target models).

The idea of using a shape template in order to guide the segmentation of an object from an image was first introduced by Widrow [91] and his rubber mask technique and Fischler and Elschlager [28] and their spring-loaded templates.

2.4.1 Physics Based Deformable Models

Physics based models attempt to explain the variation in shapes through forces and constraints acting on an elastic body.

Active contour models or *Snakes* [49] are the most famous example of a physics based deformable model. A parametric contour is defined in terms of an internal and external energy. The internal energy describes the constraints on the contour (For example, the flexibility, elasticity and tension of the contour). The external energy describes how the contour responds to the potential energy present in the image. This potential energy is derived from the object boundaries/image gradients present in the image. There is an equilibrium state where the internal and external energies reach a balance. The calculus of variations [18] tells us that this equilibrium is reached when the sum of the internal and external energies are minimised. A solution can be reached by solving the Euler-Lagrange equation [49, 10]. The snake will deform until an equilibrium state is reached.

Active contour models have been widely used in the segmentation of biological structures [49, 80, 10, 11, 74, 57, 43]. Snakes in their most basic form suffer from several weaknesses. The interactive nature of snakes means that they do not complete automatic tasks very well. Active contour models usually require active supervision.

Snakes are limited in that they must be initialized close to the shape they are attempting to segment. This means that a snake must be located in close proximity to the biological structure under consideration. This is necessary since they are strongly affected by local energy minima, which means that they sometimes fail to segment objects correctly.

There has been a significant amount of research to address this deficiency of active contour models. This research has mainly focused on different ways of describing how the contour reacts to the external energy or forces present in the image data.

Kass, Witkin and Terzopolous in their seminal paper [49] introduce interactive forces in order to guide the snake towards feature points in order to avoid local minima. *Springs* were used in order to pull the active contour towards image landmarks. While another interactive force *volcanoes* could be used to push the contour towards areas where the feature points were located. Interactive forces suffer from the disadvantage that they cannot be specified automatically, human intervention is required in order to identify where the interactive forces must be placed.

Terzopoulos et al. [84] proposed the concept of a multi-scale Gaussian potential force in order to deal with the problem of local energy minima. The effect of the Gaussian potential force was scaled depending on how far the snake was from the image target. Unfortunately, the different scales values had to be manually set and differed from target to target.

Cohen [10] introduced an external *balloon* pressure force in order to push the active contour beyond local energy minima. This technique provided the advantage that the performance of the active contour was less sensitive to the initialisation of the snake. The unit normal of the model active contour determined whether the balloon deflated or inflated. Work has been done [73, 71, 7] to incorporate region information to determine the magnitude of the inflation and deflation. Balloons suffer from self-intersection problems. In other words, the active contour model can cross itself and

form loops.

Cohen and Cohen [11] extended the attraction range of the external force with the use of distance maps. The reasoning behind this idea was that the extended attraction range would push the snake past local energy minima. Each point in a distance map stores the distance to the closest object boundary. The external energy is then defined from the distance map. The magnitude and effect of the external force at a particular position is based on a value at a corresponding position in the distance map. Unfortunately, this technique encounters difficulties with boundary concavities as the contour will be pulled in opposite directions.

Delingette [21] and MacDonald et al. [58] proposed a solution to the problem posed by concave boundaries. In their schemes the external energy was based on the signed distance between the model and target calculated along the model's normal direction. The distance was recomputed each time the model was deformed. This is a computationally expensive solution as a 1-D search must be conducted along the model's normal for each point considered. Although a maximum search distance can be defined to reduce computation, this can affect the quality of the resulting segmentation.

Another solution to the concave boundary problem has been suggested by Xu and Prince [93, 94]. They create a gradient vector flow field by using a vector diffusion equation to diffuse the gradient of an edge map that is distant from object boundaries. Care is taken to adapt the diffusion to edge strength so that object boundaries are not distorted. The external energy is defined from this gradient vector flow field. This solution is based on the principle that you want to attract points to the nearest edges.

It should be noted that one can combine several different approaches to define the external force of the snake. A hybrid approach combining a number of the discussed schemes can provide better results than a single technique.

A further limitation of snakes is that the geometric nature of active contours is restricted by internal energy constraints. It is difficult to encode additional prior knowledge regarding the expected shape of the target of the segmentation. This means that active contours struggle to segment certain geometric shapes (e.g tubular structures).

Snakes also have difficulty dealing with topology changes such as the splitting and merging of parts of a model. This is due to the fact that an active contour is a parameterised model. So, whenever there is a change in topology a new parameterization must be found for the snake. This re-parameterization is a computationally expensive and complicated process.

Sophisticated schemes have been suggested by McInerney [62], Durikovic et al. [22] and Sapiro et al. [76] to cater for topology adaptation in active contour models.

Although active contour models are formulated in a continuous domain, due to computational efficiency considerations they are usually implemented in a discrete fashion.

2.4.2 Statistically Based Deformable Models

This class of deformable models attempts to improve the segmentation of 3D structures by incorporating a statistical shape model in the segmentation process. The rationale is that by including global shape information in the model one is able to more accurately control the deformation process. Statistically based deformable models are tailored to a particular class of shapes and are not as general in representation as other deformable models.

In order to obtain this statistical information a training step is required. In the training step, a number of examples of the class of object to be segmented are analysed and the necessary statistical model is built. This statistical model provides a shape constraint which determines how the model can deform. Cootes and Taylor [14, 17] introduced a statistically based deformable model called an Active Shape Model (ASM). In their technique a training set is used to construct a Point Distribution Model (PDM). The PDM is built by sampling the surface along its boundary. A Principal Component Analysis (PCA) is conducted on the PDM to determine the principal modes in which the shape under analysis will vary.

A PCA will determine the n-principal vectors (or principal modes of variation) in an n-dimensional space, as well as conveying the extent to which a principal vector accounts for the variation within that space. The great advantage of the PCA is that we need keep only a subset of all the principal vectors and these will account for most of the variation in the space. In this way, an n-dimensional space can be reduced to a m-dimensional space, where m is much smaller than n. Clearly, this reduction in dimensional space has important efficiency advantages.

A shape can then be explained by a linear combination of the average shape and the modes of variation produced by the PCA. Shape parameters are also introduced into the statistical model to constrain the deformation of the shape so that it does not diverge from the learnt model.

Alignment of the statistically based deformable model and the target model is achieved through an iterative pose estimation scheme which follows a least squares approach. At each step in the iteration the shape parameters are updated, and this helps the model account for the local variation which is present between objects of the same class. Iteration ends when the update to the shape parameters is negligible.

ASMs suffer from the disadvantage that a large data set is required in order to construct a meaningful training set. ASMs are also not able to account for the variation in grey-levels that can occur across images. A further limitation is that the sampling for the Point Distribution Model must be done manually, i.e. an expert is required to mark feature points on the images. Although, some success has been achieved for automatic labeling by detecting feature or landmarks points [39, 54, 25].

In order to improve the reconstruction of the ASM, an outlier detection algorithm has been proposed by Duta [24]. This outlier detection occurs at each iteration of the segmentation process. In this way, the shape parameters are not updated with erroneous input and a more correct reconstruction is the result.

Wang and Staib [88] viewed the ASM as a Bayesian estimation problem. They introduce a smoothness covariance matrix in order to generate more flexible models. A solution to the Bayesian estimation is found by maximising an objective function (*a posteriori* probability).

Active Appearance Models (AAM) [27, 26, 12] are able to cater for grey-level variation between images and remedy this limitation of ASM. PCA is applied to build a grey-level profile for each sampled point in the PDM. The images are normalized in order to prevent the imaging artifacts specific to a particular image acquisition process having an effect. The shape model together with this grey-level model are then used to segment the object.

Staib and Duncan [81] use a Fourier representation to represent a deformable model. This approach has similarities to the ASM (where the dimensionality is reduced) in that a compact representation can be achieved by truncating terms in the Fourier series. Another advantage of this technique is that global shape properties can be derived by geometric descriptions. A Bayesian formulation is used to incorporate prior information. This prior function is built by labelling feature points on examples of the class of object to be segmented. These are then parameterised into their Fourier representation. A solution is found by maximising the *a posteriori* function that balances the difference between model and target models. This is achieved through the use of a gradient ascent algorithm. A genetic algorithm [20] has also been used to find a solution to maximizing the *a posteriori* function.

Székely et al. [82] have also used a Fourier representation to describe deformable models. Statistical analysis of training sets produces a mean shape model and its associated eigenmodes. These in conjunction with an elastic fitting method are used to segment target shapes. For computational efficiency the most important eigenmodes are used to achieve an initial fit. Hereafter, the fit is refined by incrementally adding more eigenmodes.

The major stumbling block presented by statistically deformable models is that a training dataset is required in order to obtain the relevant and meaningful statistics. Obtaining a large dataset is often problematic or impractical. Labelling of feature points, which is required for this class of deformable models, is often done manually and is a tedious and lengthy process. Even in automatic schemes, operator intervention is required to ensure that feature point detection is reliable. The fact that a training set is required for each class of object that needs to be segmented also creates an additional burden. However, the global shape control that these deformable models allow makes them very useful.

2.4.3 Geometrically Based Deformable Models

Geometric deformable models use geometric criteria in order to achieve their work. Caselles [4] and Malladi [59] first introduced this type of deformable model. These deformable models are classified into two groups: those based on the theory of curve evolution [78] and those based on the level set method [53, 52].

Curve evolution theory looks at how curves deform based on geometric criteria such as the unit normal and surface curvature. Particular attention is paid to the manner and the rate at which these curves evolve. Surface boundaries are extracted by combining the rate at which the curve deforms together with the information present in the image data. The rate at which a curve evolves is based on a *speed function*. The choice of a particular speed function plays a crucial role in determining how a particular surface is extracted.

In the level set method, a curve is represented as a level set of 2D scalar functions called the level set function. The points produced by a particular level set function value form part of that level set. The level set function therefore provides an implicit representation of a deforming curve. The initial level set function needs to be chosen so that that the zero level set corresponds to the position of the initial curve. A common choice is the signed distance function. Efficient methods have been discovered to compute this efficiently (e.g fast marching method [59]). The speed function chosen then needs to be extended to other values for the level set function. Care must be taken to ensure that the extended speed functions stay true to the original speed functions properties. Otherwise, inaccuracies will be introduced in the geometric calculations. If such inaccuracies occur, the level set function needs to be reinitialized.

In the initial models introduced by Malladi [59] and Caselles [4] a multiplicative term was used to couple the curve evolution with the image data. This worked well for images where there is a significant amount of contrast present i.e. the curve boundaries are distinct. However, where the image boundary is not well defined this method failed in its task. The curve will evolve passed the object boundary and will not converge to the object border.

Caselles [5] and Yezzi [95] introduced an additional term which helped to pull the curve back once it crossed an indistinct boundary. Unfortunately, their schemes were still not able to cater for gaps in object boundaries.

A geometric deformable model that is able to cater for small gaps in object boundaries has been presented by Siddiqi [79].

Guo and Vemuri [33] have incorporated global shape parameters in conjunction with a geometric deformable framework. This is a hybrid technique in which there is a global shape constraint present as well as retaining the advantages of the geometric deformable models i.e topologically adaptable.

Geometric deformable models are able to deal with topology alteration, the major limitation

of physics based deformable models, in an elegant fashion as the surfaces are deformed according to geometric measures and consequently the surface parameterization is not affected. There is therefore no need to re-parameterize a surface in the event of a topology alteration as this occurs implicitly.

In some cases, however, topology alteration is undesirable. As geometric deformable models can produce unpredictable topology alteration, especially in noisy imaging conditions, deformable models, which have stricter control of topology alteration, are required in these cases.

2.4.4 Other Deformable Models

Modal analysis has also been used to control the global shape of deformable models [68, 67]. This approach is similar to both the ASM and the Fourier based models and is based on the theory of finite elements. Each feature point is represented as a finite element called a node in a multidimensional space and by interpolation we are able to construct the object's shape. Segmentation is achieved iteratively and a nodal displacement vector controls the deformation at each step. This nodal displacement vector incorporates the effect of external forces such as mass, damping and stiffness. For efficiency reasons, only the most important modes are considered.

Deformable superquadrics were introduced by Terzopoulos and Metaxas [83]. This is a hybrid technique in which local and global fitting occur simultaneously. Global shape, which is defined by a superquadric surface, accounts for as much deformation as possible, while a local deformation accounts for the remaining local object variations. Superquadrics are an extension of standard quadric surfaces and can be described by a small number of parameters. For this reason, they can describe a wide variety of shapes.

Vemuri and Radisavljevic [87] have completed additional work on deformable superquadrics. In their scheme, they take advantage of the multi-resolution nature of a wavelet parameterization to smoothly combine the local fitting process and the global deformation process.

Bardinet et al. [1] make use of a deformable superquadric to achieve an initial fit to the target data. Hereafter, the segmentation is refined by free form deformation (FFD). In FFD the surface is embedded in a volume that behaves similarly to gelatin. Deformation of the volume will result in a rubber-like deformation of the embedded surface. The use of FFD allows additional global control as well as limiting the effects of local variation.

Ip and Shen [43] incorporate global shape information by using an affine transformation to align a shape template and the deformable model. The deformation is constrained so that it doesn't differ greatly from their original shape template.

Fritsch et al. [29] use information on the medial loci of the shape under consideration (which they call deformable shape loci) in order to guide the segmentation process. The use of the medial

loci as a global shape guide means that their technique deals well with noisy images that exhibit significant image artifacts. They use a prior probability to describe and control the relationship of feature points.

Jones and Poggio [47] construct a shape and texture vector for each element of template shape from a training set. A linear combination of shape and texture vectors are used to describe the shape. The matching between a target and reference shape is achieved through a sum of squares minimisation which is solved efficiently by a stochastic gradient ascent algorithm. A coarse to fine approach is employed in the matching through the use of a multi-resolution analysis pyramid to further improve computational efficiency.

2.5 Techniques

This section provides a brief and cursory introduction into a number of the techniques utilised in this dissertation. For an in-depth and thorough description of these topics, the references provided can be consulted.

2.5.1 Free-Form Deformation and Directly-Manipulated Free-Form Deformation

Free-Form Deformation (FFD) [77] is a spatial deformation technique in which local distortion can be achieved without affecting the global shape of an object.

In FFD, points that define an object are embedded in a deformable volume. This deformable volume is described by a hyper-patch. The hyper-patch in turn is described by a lattice of control points. Typically, the control points in the lattice are arranged in a regular grid. For example, in 3-dimensions, there would be a regular spatial interval between control points along a line segment on the xy plane. By warping the deformable volume/hyper-patch we are able to distort the corresponding embedded points. In effect, this means that by translating the control points that define the hyper-patch we are able to locally deform the embedded points. As in splines, a basis function is used to describe how the control points of the hyper-patch are related to one another.

This dissertation makes use of Directly Manipulated Free-Form Deformation (DMFFD) [42, 30] to perform the local deformations necessary to account for the local variation that is present in a bone from patient to patient. DMFFD is an extension to FFD in which constraints specify how deformation should occur. A constraint is defined by a point and a vector. The point specifies the origin of the deformation, while the vector specifies the intended direction and magnitude of the deformation. Based on these constraints the control points of the hyper-patch are deformed and consequently the points embedded in the hyperpatch.

We chose DMFFD for local deformation as it provides a more intuitive interface to specify a

required local deformation than FFD. In FFD, a spline must be specified to describe a deformation, while in DMFFD, a point and a vector that describe a constraint, are all that is required.

2.5.2 Cluster Analysis

Cluster analysis is used to identify distinct groups or clusters within data. Cluster analysis makes use of a dissimilarity metric in order to classify the relationships between data. In the case of spatial data, distance metrics are the dissimilarity criteria of choice.

A number of the more popular distance metrics (between data point x and y) are:

- **Euclidean distance:** $distance(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$
- **Euclidean square distance:** $distance(x, y) = \sum_i (x_i - y_i)^2$ Greater weight is placed on data points that are further away.
- **Manhattan or city-block distance:** $distance(x, y) = \sum_i |x_i - y_i|$ Measures the average distance across dimensions.
- **Chebychev distance:** $distance(x, y) = max|x_i - y_i|$ Useful at detecting differences in a dimension.

A linkage or amalgamation rule governs how data points are added to a cluster. A number of these joining rules have been developed. A couple of examples of these include:

- **Single linkage:** Nearest neighbour is the criteria used to join clusters. In other words, the two closest clusters (based on the particular distance metric used) are amalgamated.
- **Complete linkage:** The two clusters that are furthest apart are amalgamated.
- **Unweighted Pair-Group Method using Arithmetic Averages (UPGMA):** The average distance between all pairs of objects in the two different clusters is used to join clusters.
- **Weighted Pair-Group Method using Arithmetic Averages (WPGMA):** The average distance between all pairs of objects in the two different clusters is used to join clusters. The average distance is weighted by the number of elements in the cluster.
- **Unweighted Pair-Group Method using Centroid Averages (UPGMC):** The distance between the centroids of the clusters is used to to join the clusters.
- **Weighted Pair-Group Method using Centroid Averages (WPGMC):** The distance between the centroids of the clusters is used to to join the clusters. The distance is weighted according to the size of the cluster.

There are also several different approaches to cluster analysis. The most important of these are *Hierarchical Clustering* and *K-Means Clustering*.

In Hierarchical (Tree) Clustering, cluster sizes range from a cluster for each data point (at the leaves of the tree) to one large cluster encompassing all the data points (at the root of the tree). Typically in Hierarchical Clustering, a threshold is chosen to determine at which level of the hierarchy/tree we wish to examine the clusters formed.

In K-Means Clustering, the data is partitioned into K distinct/different clusters. This is achieved by minimizing the intra-cluster variance while maximizing the variance between clusters.

2.6 Explanation of Terminology

In order to clarify the understanding of the text, the following section will explicitly define in context the various technical terms used throughout this dissertation.

Segmentation refers to the grouping of areas with similar density or image intensity values. For example, the segmentation of the arm will result in separate areas for the tissue and bone. This is due to the fact that the imaging medium will affect objects with the same density in a consistent fashion. So, we are able to distinguish or group or separate areas with a common density or image intensity. On an image processing level, by the term segmentation, we mean that we are able to group the pixels that form part of a particular object that we wish to identify into a region.

In this dissertation, the *model* or the *reference model* refer to a reference three-dimensional polygonal model of a healthy bone model. This is the template that we use as guide in our segmentation process.

The *target* or *target model* refers to the voxel model created from the CT scan image data or a partially segmented polygonal model of that voxel model.

The term *registration* refers to the aligning or matching of two objects or shapes or images. In our case, it refers to the process of aligning the orientation of the target model and the reference model.

Especially in Chapter 4 we refer to vertices that are within a certain *neighbourhood*. Here, neighbourhood refers to the shortest number of edges away vertices are from a particular vertex. For example, all the vertices whose shortest path to a particular vertex is two edge lengths, are within 2-neighbourhoods of that point.

A *local patch* is defined as the set of connected polygons centered about a particular vertex.

Chapter 3

Segmentation

In this chapter, we present our algorithms and methodology for extracting a 3D polygonal model of a bone from a Computed Tomography Scan. In Chapter 4 we will use geometric criteria to evaluate the integrity of this bone mesh model.

We use a coarse-to-fine approach in order to extract the desired surface mesh. We begin by generating a rough approximation of the segmented 3D bone. This approximation is refined at each step of the algorithm. Throughout the segmentation process a reference mesh is used to guide and constrain the segmentation process.

We begin by describing, in Section 3.1, the initial image processing tasks performed on the CT scan data. The point set produced by this step serves as input to the global registration of the point set and reference surface mesh described in Section 3.4. We then discuss the two coarse alignment algorithms investigated to help bring the target point set and reference mesh into global alignment. Bounding Box Registration and Dual Line Segment Registration which are suited to compact, rounded bones are tackled in Sections 3.4.1 and 3.4.3, while Maximum Axis alignment which favours elongated bones is dealt with in Section 3.4.2. Section 3.5 describes the rigid alignment phase which completes the global registration process. The selection of corresponding points between target and reference models which plays a crucial role in rigid registration and local deformation is discussed in Section 3.6. Finally, in Section 3.8, we adjust for local variation in bone structure by applying a local deformation to our mesh.

3.1 Image Processing

Rather than dealing with the CT scan as a volumetric data set, we begin by considering the 3D volumetric data set as a stack of 2D components (see Figure 6), and operate independently on each of these image slices.

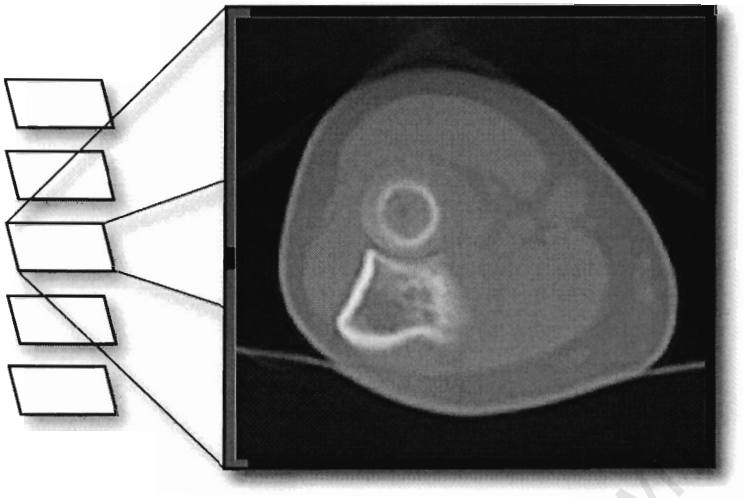


Figure 6: **CT Slice:** An example of an CT image from the stack of image slices that make up a CT scan of the lower forearm. Notice that there are distinct intensities that signify different densities. It is this contrast between densities that radiologists find so helpful in CT scans.

The rationale for such a simplification is as follows:

1. The spatial resolution between pixels in the image slices will in the vast majority of cases be greater than the voxel resolution between consecutive slices in the volumetric data. It therefore make sense to take as much advantage of this superior resolution as we can before moving to the coarser volumetric domain.

2. The output from a CT Scan is a stack of image slices. Without any data transformation we can make use of numerous well developed image processing techniques.

3. Working with volumetric data sets has high memory requirements. The simplification of the volumetric data set, after image processing on each image slice has been completed, will result in greater memory efficiency.

The major drawback of using a 2D approach is the possible loss of accuracy. This is important especially as accuracy can effect the correctness of a diagnosis in the medical field. However, we felt that such a sacrifice of accuracy was acceptable, as a parallel implementation, which would provide a solution to the high memory requirement, would add considerable complexity.

The aim of our image processing is to identify those pixels in each image slice that form part of the bone that we are interested in. This can be achieved in two logical steps. First, we need to

classify those pixels that we consider to be bone (see Figure 7a). Then, the collection of pixels that represent the target bone need to be isolated from the other bone segments that have been identified in the image slice (see Figure7b).

Studying the histogram of an image slice provides us with interesting insights. Peaks on the intensity histogram delimit the most important objects in the image. Objects can be classified according to these histogram peaks. Bone has a high density, and as a result of the CT acquisition process, in image space has a high pixel intensity. Flesh on the other hand, has a low density and therefore a correspondingly low pixel intensity.

Taking this into account, we can identify the pixel intensity values which are most likely to represent bone, as those that fall in a narrow band surrounding the bone intensity peak on the histogram. An intensity threshold on these values results in a classification of the bone segments contained in the image.

Simple heuristics based on knowledge of the anatomical area under inspection, combined with region growing and largest connected components [85] can be used to isolate the target bone.

Region growing is a technique which makes use of intensity differences between adjacent pixels in an image in order to group areas of similar intensity. A seed pixel is chosen from which adjacent pixel intensities are iteratively tested to determine if they form part of the seed pixel's intensity group. New additions to the intensity group serve as input to subsequent iterations of the algorithm. The effect is that connected regions appear to "grow" across the image.

Largest connected components is an algorithm that quantifies all the regions, produced for example by region growing, and identifies the largest of these.

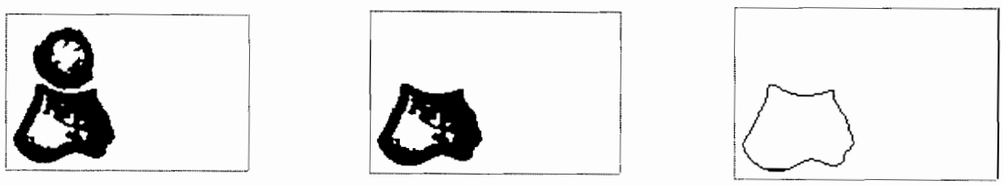
For example, consider a CT scan of the lower forearm. Two bones can be found in this area, the ulna and the radius. Knowledge of the human skeleton allows us to determine the relative spatial position of these two bones. Thus, if the ulna is the bone that we wish to inspect, we can restrict our search to a certain area of the image. Without this high-level knowledge identifying the bone we seek is a far more complicated task as we can only rely on the data inherent in the image.

Volumetric Reconstructions from Limited Angle Tomographic Scans will result in a certain amount of noise and imaging artifacts being present. A combination of region-growing and largest connected components is used to remove this spurious data.

What we are left with is a binary image of the target bone. A surface voxel representation is sought as we are generally interested only in the surface integrity of the bone. An *eroded* version of the binary image of the target bone is required as we are interested in the surface or outline of the bone. Applying traditional morphological operations of dilation and erosion [85] would have the desired effect, but this would result in the introduction of unacceptable error and approximation into the data.

We used a scan-line based algorithm [3] to extract the eroded target bone (see Figure 7c). Alternatively, edge-detection operators could be also used to extract an outline of the target bone.

Using an eroded version of the target bone has the added advantage of reducing computational costs and memory overhead in the forthcoming stages of the segmentation process.



(a) (b) (c)

Figure 7: **Image Processing:** In (a) the bone is isolated from the rest of the forearm. Then, in (b) using prior knowledge of the human anatomy the Ulna is separated from the Radius , and thinned (c) to reduce computational overhead.

3.2 The Reference Bone Model

A model for the bone we wish to extract can be acquired from a high resolution CT or MRI scan of 'normal" (non-pathological) anatomy. Several models may be required to accommodate sex and age differences. This is simply an additional *a priori* parameter. The algorithms remain unchanged. The bone can be extracted by manual or partially automated segmentation tools. For high quality image data there is usually a clear delineation between different tissue types within the data set and segmentation becomes significantly easier. Human intervention (via editing tools) is required to correct segmentation errors and to ensure that the model is anatomically correct. An iso-surface extractor can be used to generate a triangulated surface which can be re-meshed to any desired vertex density by simple triangle sub-division. The model can then be used to extract similar anatomy from poor resolution data or data with many artifacts.

3.3 Registration of Target Voxel Model and Reference Mesh Model

Once we have isolated the target bone from the flesh, fat, other bones and noise in the CT image slices, the voxel bone model must now be matched against a reference mesh surface bone model.

In order to use the geometric knowledge encoded in the bone model, we need to ensure that the salient geometric features of the voxel bone model and the reference surface bone model are in close proximity. This requires registration of the voxel model and surface models, so that the geometric information inherent in the reference model can be used to guide the segmentation process.

The alignment of the voxel model to the surface model is achieved through a coarse to fine approach. Firstly, the voxel and surface models are mapped into the same spatial location and a common orientation for both models is ensured. This is followed by global registration under a similarity transform (scaling, rotation and translation). Finally, a local deformation of the surface model, ensures that a smooth constrained approximation of the voxelised bone surface is obtained. The accuracy of this fit is limited by the resolution of the voxel data, and to a lesser extent, the vertex density of the mesh model. Using prior information, in this case the spatial resolution of the CT scanner, vertex density can be set to ensure sufficient flexibility in the surface model.

The target bone and reference bone could easily have been acquired with different manufacturers' CT-machines, each of which are calibrated with their own distinct co-ordinate systems. In order to make consistent comparisons it is important that we consider both 3D objects in a common coordinate system. To maintain the correct geometric proportions of the models, close attention must be paid to the CT acquisition information extracted from the DICOM header files. This includes Pixel Spacing, Patient Position, Spatial Resolution and Image Position.

Taking this information into account, we apply a transformation to map the voxel data into the reference mesh co-ordinate system.

3.4 Coarse Registration

Within the common co-ordinate system, the voxel bone model and the reference mesh model are both at arbitrary and most probably differing geometric orientations. One of the primary benefits of model-based segmentation lies in the introduction of a reference model to guide the segmentation of the voxel model. In this way, *a priori* knowledge concerning the anatomical structure of the bone (which is encoded in the reference model) can be used to eliminate uncertainty in the segmentation process.

For this to be possible, we require that the reference mesh model and voxel model are closely aligned in space as well as in orientation/pose. In other words, we require that the salient features of the particular bone under consideration in the voxel and mesh model are in close proximity.

This rough registration can be achieved in a number of different ways. We have explored three approaches:

1. **Bounding Box Registration:** This method involves registering the bounding cube enclosing

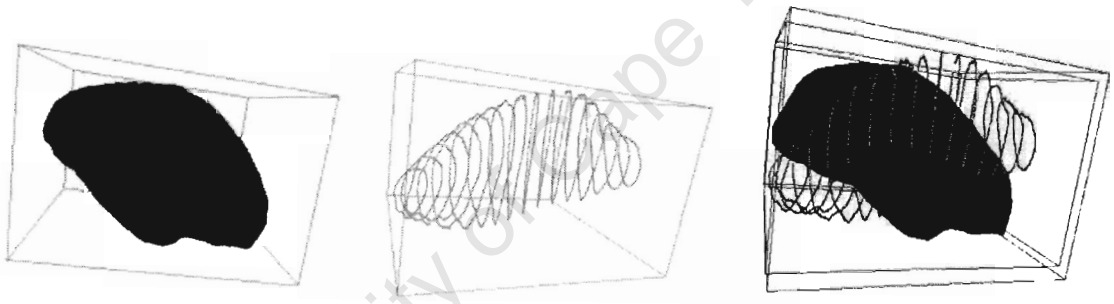
the target voxel model and the bounding cube enclosing the mesh reference model.

2. **Longest Line Segment Registration:** This method involves registration of the line segment representing the longest Euclidean distance through the target model with the line segment representing the longest Euclidean distance through the reference model.

3. **Dual Line Segment Registration:** This method is a two stage process which involves the alignment of two line segments that are calculated from the centroid of the voxel and reference models.

3.4.1 Bounding Box Registration

A bounding cube surrounding the target and reference model is constructed. The two bounding cubes are then registered using the ICP algorithm [96, 2], which is further explained in section 3.5. The bounding box registration is visually depicted in Figure 8.



(a) (b)

Figure 8: **Bounding Box Registration of the Patella:** In (a) bounding cubes are constructed surrounding the voxel and reference models, and (b) shows the result of the coarse registration when the boxes are aligned using ICP.

Construction of the bounding cubes is achieved as follows. The three primary axes through the models are determined by Principal Component Analysis [44].

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of a data set. In other words a multi-dimensional data set is reduced into a smaller number of variables called its *principal components*. The first principal component accounts for the majority of variability in the data set, and each successive principal component accounts for less of the variability in the data set. In this way, for example, 95% of the variability in a multi-dimensional data could be explained by its 10 most important principal components.

In our case, we are interested in the first three principal components that result from performing a PCA on the target/reference point set. The resulting three vectors form the x, y and z axes of the bounding box surrounding that particular point set. Projecting every point in the data set onto each of these vectors we are able to determine the extents of our bounding box. In fact, Bounding Box Registration involves aligning the corresponding three principal axes in the reference and voxel models.

Unfortunately, we have found Bounding Box Registration to be somewhat limited in practice. In many cases, the difference in point distribution of data in the voxel model and mesh model result in principal axes that are disparate. This results in a coarse alignment that is not favourable and thus causes a poor segmentation.

3.4.2 Longest Line Segment Registration

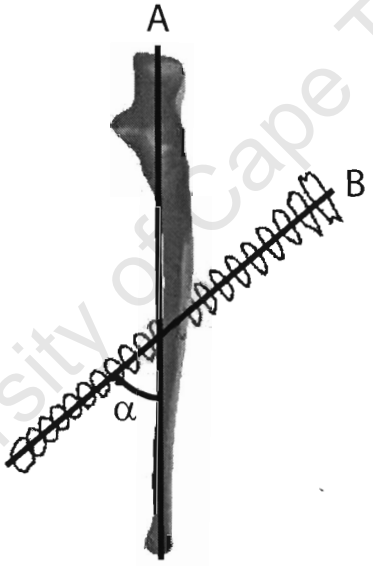


Figure 9: **Longest Line Segment Registration:** Coarse registration is achieved when the target point set is rotated about the line segment **A** by the angle α i.e the angle formed by the intersection of line segment **A** (the line segment which passes through the reference mesh) and **B** (the line segment which passes through the target point set)

Longest Line Segment Registration involves calculating the line segment with greatest Euclidean distance through the voxel model and the line segment with the greatest Euclidean distance through the reference mesh model. Then, calculating the transformation required to register these

two line segments [64], and applying this transformation to the voxel model. Figure 9 depicts an example of the longest line segments and the transformation required for registration.

The brute force calculation of the Longest Line Segment used for the voxel and mesh models is a $O(n^2)$ operation where n is the number of data points in the model. This is due to the fact that for every data point, the distance with every other data point in the model must be computed. It is of course possible to improve the algorithmic complexity of this calculation with the use of a spatial division of the data. But, considering that finding the Longest Line Segment is a once off preprocess, optimizing this computation was not considered critical.

This technique works particularly well for elongated bones, such as the femur, ulna and radius.

3.4.3 Dual Line Segment Registration

This technique of coarse alignment is a two stage process. Firstly, the 3D point that is the greatest Euclidean distance from the centroid, or center of gravity, of the respective model is identified. A line segment can be defined by this point and the centroid for both the mesh and voxel models (Figure 10(a)). The transformation to align these two line segments is calculated [64], and is applied to the voxel model.

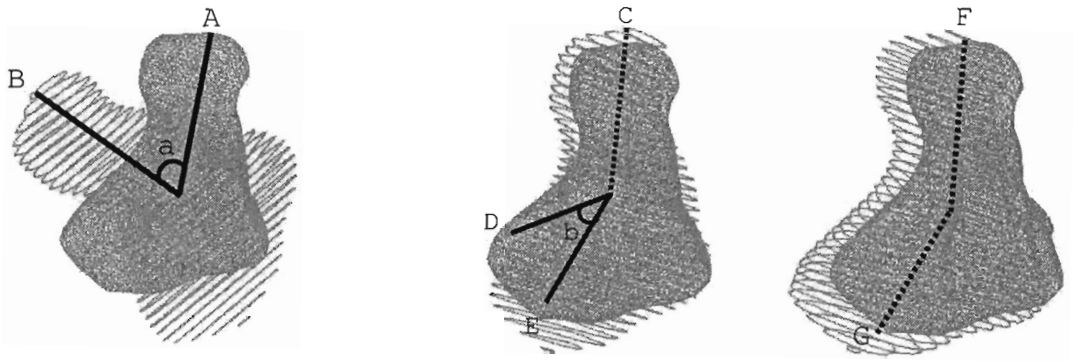
The reference and target model are effectively locked onto one axis by this process (Figure 10(b)). What this means is that we can reduce the remaining alignment from a 3D to a 2D problem. In each of the voxel and mesh models, the 3D data points are projected onto the plane defined by the previously calculated line segment and centroid. By applying this projection, all the 3D points in a model lie on this plane.

Lastly, we proceed, by identifying the 3D point on the plane that is the greatest Euclidean distance from the centroid. The transformation to align the line segments formed by these two points in the two data models is calculated and applied to the voxel model. The result is a coarse alignment of voxel and mesh data (Figure 10(c)).

Constructing the line segment based on the point that is the greatest Euclidean distance from the centroid is not always appropriate. In some cases, the line segment based on the point that is the shortest Euclidean distance from the centroid achieves superior alignment. This varies depending on the shape of a particular bone model. As we are using a model based approach where we know the bone that we are examining, this parameter can be built into the description of the reference model.

It should be pointed out that for efficiency purposes we have used the Euclidean square distance whenever we calculate the Euclidean distance.

In practice, we have found Dual Line Segment Registration to be most successful in the case of compact, rounded bones such as the capitate, patella and hamate.



(a) (b) (c)

Figure 10: **Dual Line Segment Registration of the Hamate:** In (a) the longest line segment from the centroid (**A** for the mesh model and **B** for the voxel model) are computed. The rotation by angle **a** of the voxel model results in (b) where the longest projected line segments (**D** and **E** respectively) are computed. Finally, the voxel model is rotated by angle **b** to achieve the coarse registration shown in (c).

3.4.4 Further Considerations

It is of course possible that the rough registration will result in an inconsistent orientation of target and reference models. The most common problem that can occur is an inversion of the orientation of the two models. In other words, the front-end of the target model is aligned with the back-end of the reference model, and vice versa. This could occur as a result of rotational symmetry. Fortunately, this ambiguity can be detected and corrected through the use of our knowledge concerning the human skeleton. In other words, we can ensure the correct orientation of specific bones by considering the position and orientation of the surrounding anatomy.

3.5 Rigid Alignment

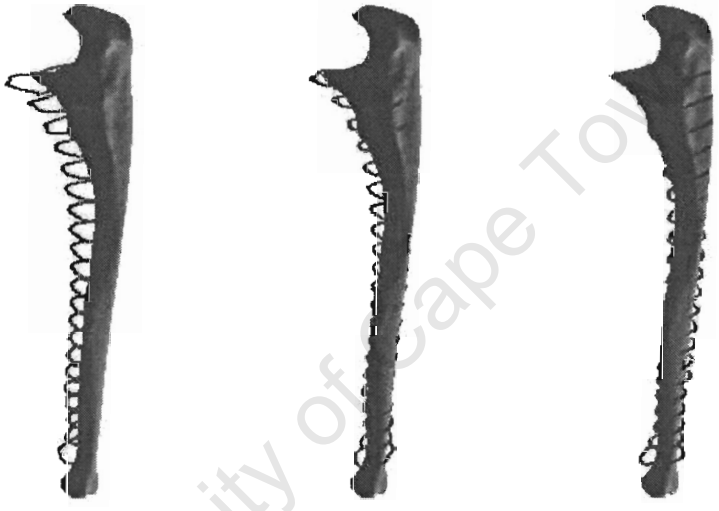
The voxel and reference models now share a common orientation and are spatially adjacent. A global alignment of the two models is now required. We used Iterated Closest Points (ICP) [2, 96] to achieve a rigid alignment between voxel and surface models.

ICP iteratively finds the rigid transformation (translation and rotation) that will map one set of points (reference points) onto another set of points (target points). For each iteration of the algorithm, the target and reference point sets are brought nearer to one another through the estimated transformation. Convergence is achieved when successive iterations are no longer able to bring the

two point sets any closer. At this point the ICP iterations halt. Figure 11 provides a visual example of the various stages of the ICP algorithm.

ICP is limited by the requirement that there be a one-to-one mapping between the two point sets and that this mapping is known. We do not know this mapping or correspondence between voxel and surface models. Section 3.6 explains how the matching between these two point sets is achieved.

Besl and McKay show in [2] that the ICP algorithm always converges towards a solution. This means that after a moderate number of iterations the ICP algorithm can ensure an accurate rigid alignment.



(a) (b) (c)

Figure 11: **Global Registration Results:** The dark rings represent the voxel model while the surface model represents the reference model. (a) After coarse registration, (b) 10 iterations of ICP, and (c) rigid registration of voxel and reference models.

3.6 Correspondences

Finding the matching points between the voxel and surface model plays an important role in the segmentation process. Inaccurate correspondences will result in a poor global registration of the two models. Feature points in the two model will not match. The high level knowledge of a model-based segmentation that is used to guide the segmentation process will be applied erroneously. This will lead to an incorrect segmentation of the target bone. It is therefore of crucial importance that

good correspondences between voxel and mesh models are established.

We have assumed that the rough registration of target and reference model brings the two models into close proximity. The search for a target correspondent can therefore be limited to a small subset of candidate points surrounding the reference surface model point.

By restricting our correspondent search to a limited area we seek to avoid mismatches in correspondence. Initially, this search range may be too conservative, but this range can be extended if insufficient correspondences are found. A conservative approach is justified since we wish to limit the number of mismatches, and we assume that correspondences are to be found in proximity to the reference model due to the global alignment of the two models.

A spatial subdivision of the voxel model is achieved through the use of a kd-tree [19]. Using this data structure, the above mentioned candidate points can be efficiently extracted from the voxel model with a rectangular range query. Kd-trees have the following properties for n data points:

- they can be constructed in $O(n \log n)$ time,
- require $O(n)$ storage, and
- a rectangular range query can be completed in $O(\sqrt{n} + k)$ time for k points found in the range.

We search along the surface normal of the reference model point and find the candidate point that is closest, in terms of Euclidean distance, to this ray. This is the point that we select as our correspondent. This process is repeated for each point of the reference model. The result is that each point on the reference model has a matching point in the target model.

It should be noted that we impose additional constraints on the selection of correspondences. This is necessary as an unconstrained selection of correspondences will result in a clustering of points around voxel slices. The resulting mesh will exhibit irregular surface artifacts around these voxel boundaries that do not represent the true nature of the underlying bone geometry. Our constraints ensure greater regularity in the distribution of vertices between voxel slice boundaries. Although vertices do still migrate towards voxel boundaries, surface curvature continuity is maintained and the resultant mesh is more representative of the underlying anatomy.

3.7 Virtual Slices

Very coarse CT scans, where the interval between voxel slices is large, cause difficulties for the correspondence selection stage and consequently for the entire segmentation process. This is due to the fact that the imposed constraints fail to ensure an even distribution of data points: points are

bunched around voxel boundaries. A "sagging" of the mesh between voxel slices can be observed in the resulting segmented polygonal model. Local maxima form where the voxel slices are located, with valleys forming in the intervals between slices.

In order to enable our system to deal with such coarse CT data we introduce additional voxels slices or *virtual slices* into the CT scan. The aim of these virtual slices is to ensure a more even distribution of points during correspondence selection between voxel and mesh data.

It is desirable that the virtual slices represent the best possible interpolation between true voxel slices. The literature reveals that a number of techniques to successfully interpolate between voxel slices have been investigated [56, 9, 31, 32, 55, 8]. We have followed closely the shape-based interpolation technique proposed by Herman et al. [35] to create virtual slices.

A distance map is built of each of the two voxel slices we wish to interpolate between. This is accomplished by applying a 2-pass 3×3 chamferring filter across the image. Then, we interpolate between the two newly created distance maps using a Catmull-Rom spline [6]. Linear interpolation or another form of spline could also have been used. A thresholding operation is applied to the interpolated distance map to create a binary image. Finally, a thinning operation is performed on this binary image to recover the interpolated virtual slice. Pictorial examples of the stages involved in creating a shape-based interpolated virtual slice can be viewed in Figure 12.

Using the above described technique we are able to easily add virtual voxel slices to our CT data. These virtual slices help ensure an even distribution of points between voxel slices. And so, the robustness and accuracy of the segmentation is enhanced.

3.8 Local Deformation

Following the global alignment of reference and target model, we need to account for the variation that occurs between bones of the same kind. To address this deviation we introduce a local alignment of target and reference models.

The *local patch*, the area of effect of the local registration, is described by the *curvature hierarchy* (see Section 4.1). The extent of a local patch is determined by a specified proportion of the total surface area that it covers. For example, each local patch on the surface model might take up 1% of the total surface area of the mesh.

This definition of a local patch has the advantage of incorporating both mesh connectivity and Euclidean distance. It also allows us the flexibility of easily being able to change the scale at which local alignments occur. The curvature hierarchy for each reference bone model is automatically constructed as a preprocess for subsequent use in bone fracture detection. So, in effect we gain this local patch definition for free.

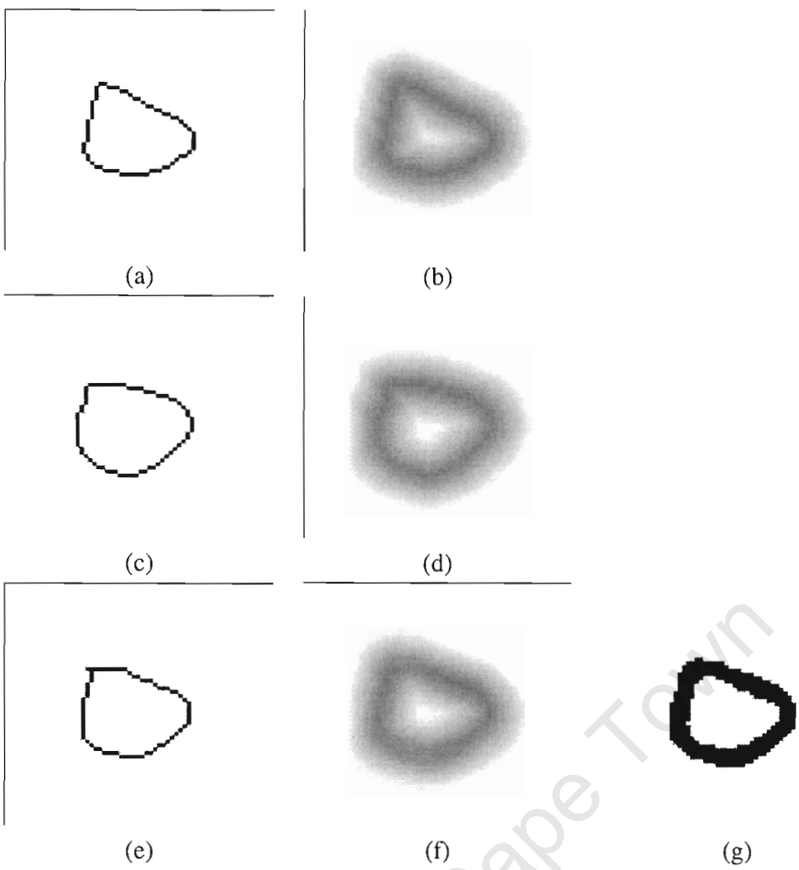


Figure 12: **Virtual Slice Creation:** (a) and (c) represent the true voxel slices. (b) and (d) are their respective distance maps created by a two pass chamferring process. (f) is the interpolated distance map, (g) represents the thresholded interpolated distance map, and (e) is the resulting virtual slice after binary thinning (f).

The magnitude of the local alignment is determined by a weighted, averaged distance between the local patch on the target and reference model.

The direction of deformation is calculated as the weighted average of the unit normals of the local patch on the reference model. What this signifies is that the local patch is *pulled* (see Figure 13) towards its target based on a measure of its curvature.

The weighting is determined in such a way that the centre of the local patch has the largest weighting and there is a fall-off in weighting values as we move towards the edges of the patch.

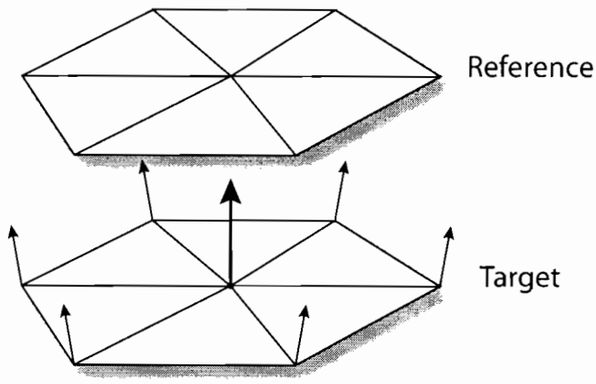


Figure 13: **Local Deformation:** A *local patch* in the target mesh is pulled in the direction of the reference mesh.

The central points of a patch therefore play the most influential role in the local deformation while peripheral points have far less influence.

The local alignment between target and reference model proceeds iteratively until the registration cannot be significantly improved. Iteration terminates when the Mean Squared Error between the two models has fallen below some model specific threshold or a preset maximum number of iterations has been reached. The choice of Mean Square Error and maximum number of iterations affect the time taken to achieve local alignment as well as the quality of local alignment. If enough iterations are executed, a point is reached in the alignment process where an additional iteration will not improve the alignment of target and reference models. This means that a Mean Square Error and optimal number of iterations can be chosen for the model under consideration.

It should be noted that the local deformation is constrained. There is a limit on the magnitude of the deformation that can occur at any iteration of the algorithm. This is to ensure that a local patch does not get wrenched away from the model. If the magnitude of a local deformation is large, it is preferable that this deformation is completed incrementally, in successive iterations, as this helps preserves the smoothness of the surface curvature.

3.9 Summary

In this chapter we presented the algorithms that we have used to construct 3-dimensional mesh models of bones from Computed Tomography scans. We make use of a model based segmentation process in which a reference model guides the segmentation.

The main contribution of this chapter is that we elaborate on our coarse to fine approach to building polygonal bone models of CT data. We describe how we conduct image processing tasks on each voxel image slice in order to reduce computational and memory overhead. We then achieve an approximate global registration of target and reference models by using three coarse registration methods: Longest Line Segment Registration which is suited to elongated bones, Bounding Box Registration, and Dual Line Segment Registration which is most effective when dealing with compact, rounded bones. The global registration is then improved by a rigid alignment of target voxel and reference models through the use of the Iterated Closest Point algorithm. Spatial subdivision enables us to efficiently select correspondences between target and reference models. These correspondences are required for the ICP algorithm as well as the local deformation. In the case of coarse CT scans, we introduce virtual slices between voxel slices to ensure an even distribution of points during correspondence selection. In the final stage of the segmentation, a local deformation is employed to account for the local variation in bones between different patients.

Throughout the segmentation process we use our *a priori* knowledge concerning the human anatomy as well as our reference model to choose the correct path when ambiguities arose.

The result is that we are able to extract a 3-dimensional polygonal model of a required bone from a CT scan. This mesh model serves as input to Chapter 4 where we use geometric criteria to find bone fractures.

Chapter 4

Fracture Detection

In this chapter, we describe the new framework that we have devised for the detection of bone fractures. The basis of this framework is that sufficient geometric information is encoded in the curvature of a bone so that fractures can be detected. The bone fracture detection framework takes the surface representation created by the bone segmentation system as input and produces as output, visual results that indicate the presence of bone fractures.

We begin this chapter by describing, in Section 4.1, the curvature hierarchy we have built to detect bone fractures. This curvature hierarchy forms the basis of the framework and allows the detection of bones fractures at different scales. A number of metrics for the construction of the curvature hierarchy are discussed. Then, in Section 4.2, we describe our approximation to measuring curvature. Finally, we present the algorithm we have developed to detect bone fractures, which is dealt with in Section 4.3. This algorithm consists of finding geometric inconsistencies and then grouping these inconsistencies into bone fractures.

4.1 Curvature Hierarchy

Bone fractures range from slight hair-line fractures to more severe compound comminuted fracture (See Section 2.2). A small local patch might fail to detect a severe fracture as the level of examination is too fine, while a large patch might fail to detect a slight fracture as the level of examination is too coarse. Consequently, we need to consider a range of local surface patches of the bone in order to detect all the classes of inconsistencies in the structure of the bone.

It is also important to take into account that damage to the integrity of the bone will probably persist over consecutive scales of local patches. Thus the use of a scaled range of local patches will help to confirm where a bone fracture occurs.

We build a new data structure called a *curvature hierarchy* that encompasses this scaled range

of local patches. Each level of the curvature hierarchy consists of progressively larger patch sizes. These levels are termed *resolution levels* of the curvature hierarchy. Membership of a particular resolution level of the hierarchy is determined through the use of a metric. These metrics are explained in detail in the next section. Triangle faces and vertices are the atomic members of a resolution level of the curvature hierarchy.

The curvature hierarchy is constructed in order to measure surface curvature at different resolution levels.

4.1.1 Metrics for Curvature Hierarchy Construction

Three metrics were considered in the construction of the curvature hierarchy.

1. **Distance Metric** Euclidean distance was the criteria used.
2. **Connectivity Metric** Mesh connectivity was the criteria used.
3. **Surface Area Metric** This metric was based on a percentage of total surface area.

Distance Metric

A simple Euclidean distance is used to determine the radius of a bounding sphere at a particular resolution level of the curvature hierarchy. Vertices that fall within the radius/sphere of a "focus vertex" form part of the local patch at that resolution. For each resolution level of the curvature hierarchy the radius of the sphere is increased.

The distance metric has the advantage that the area of effect of every patch at a particular resolution is the same size. Furthermore, the simplicity of this metric is also attractive.

Unfortunately, this metric does not take mesh connectivity into account at all. For this reason, the metric fails in the case of thin or concave bones. In these cases, a local patch would include segments of the bone that were not connected or related in any way. Calculation of curvature for these local patches is therefore meaningless.

Figure 14 illustrates some of the problems associated with constructing local patches for the curvature hierarchy based on the distance metric. As connectivity information is not included in the distance metric it is possible that triangles from a higher connectivity level could be included before all the triangles from a lower connectivity have been added to a local patch. In Figure 14 triangles in area D, which are 2-neighbourhoods away, have not been added to the local patch even though triangles from area C, which are 3-neighbourhoods away, have been included as they fall within radius circle R. This problem does not cause difficulty in the majority of cases but is critical when there is a sharply concave or convex area. This is the case in Figure 15. Then, triangles that

are completely unrelated to each other are included in the same local patch. This will result in a meaningless calculation of curvature for an area that is not a connected local patch.

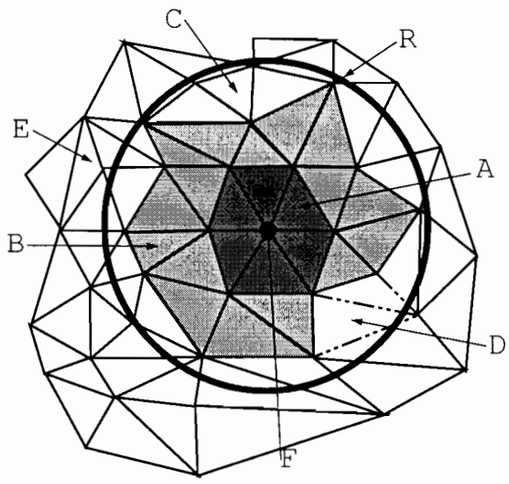


Figure 14: **Connectivity Problems with the Distance Metric:** Connectivity is not considered when constructing a local patch based on the distance metric. Triangles in area *D* (2-neighbourhoods away) are not included in the distance metric although triangles in area *C* (3-neighbourhoods away) are included.

Connectivity Metric

The connectivity metric uses the mesh connectivity of the surface model to assign vertices to different resolutions of the curvature hierarchy. Those vertices that are contained in a 1-neighbourhood of the focus vertex form part of the first resolution level. The fifth resolution level consists of vertices that are contained in the 5-neighbourhood of the focus vertex.

The advantage of the connectivity metric is that vertices in a local patch of a particular resolution are directly connected. In other words, they are part of the same local surface. Vertices in resolution level n of the curvature hierarchy are at most one edge away from vertices in resolution level $n-1$ of the curvature hierarchy. This means that connectivity information is maintained at every resolution level for every local patch.

Unfortunately, the size of local patches at a particular resolution level differs greatly depending on the area of the surface model being considered. Areas of high curvature on a bone have many small triangles. A local patch at a particular resolution in this area would be smaller than a patch in a predominantly planar area, where the triangles are larger.

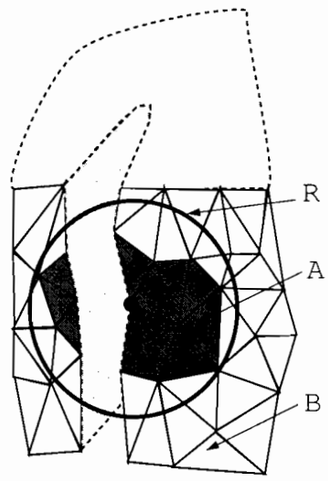


Figure 15: **Sharp Convexity Causes Problems for the Distance Metric:** Triangles that are not connected form part of the same local patch, as the distance metric does not consider connectivity.

Figure 16 gives an idea of how triangles of differing sizes could result in great differences in the area covered by local patches of the same resolution levels. This is problematic as we seek a consistent and stable measure of curvature for each local patch at a particular resolution level. The curvature measures between local patches is only meaningful if they measure the curvature of areas that are of a similar size.

This problem can be overcome if the surface model is a *regularised* mesh. A regularised mesh refers to a mesh where the area of each triangle in the mesh is the same size. Regularisation of surface model can be achieved through mesh re-parameterisation or simplification [86, 41]. Unfortunately, this can result in a poor approximation of the surface model which is unacceptable as the surface model is used in a medical context where accuracy is of great importance.

Surface Area Metric

We require a hybrid metric which takes into account mesh connectivity as well as the size of the local patch. Defining the size of a local patch as a percentage of the total surface area of the mesh and using this information in conjunction with the *n-neighbourhood* connectivity information satisfies this requirement.

For example, we could define the first resolution as local patches having surface areas the size of 1% of the total surface areas, the second resolution as local patches having surface areas of size

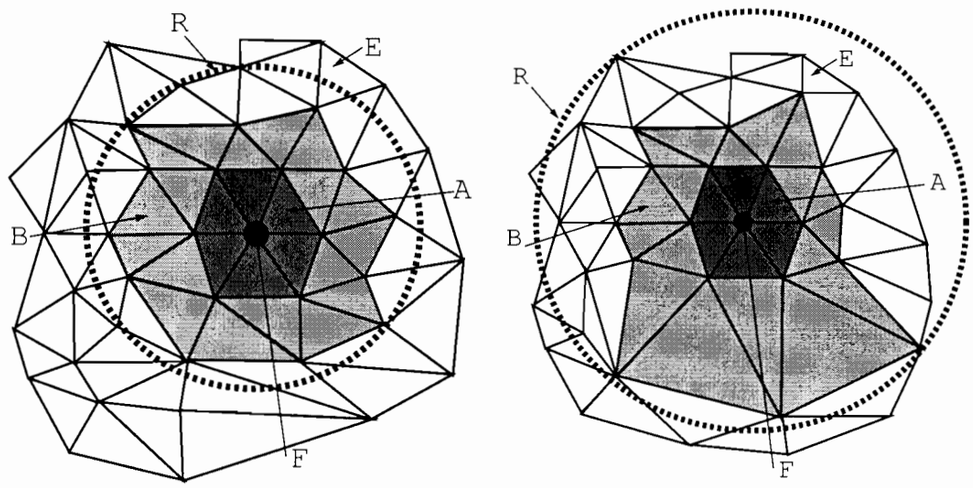


Figure 16: **Problems with the Connectivity Metric:** Local patches are produced with large disparities in area. The radius circle R in (a) is much smaller than the radius circle R in (b) even though they are local patches at the same resolution.

2% of total surface area, and so on. In practice, we have found a linear increase in the resolution level to be effective. So, if we construct a curvature hierarchy consisting of 5 resolution levels using the Surface Area Metric then the local patches of the five resolution levels will consist of 1%, 2%, 3%, 4% and 5% of surface area respectively.

This metric is the one we chose as it successfully combines distance and connectivity information to provide local patches that are of similar size at particular resolution levels.

4.1.2 Curvature Hierarchy Construction

The smallest element of the curvature hierarchy is a vertex or face of a polygonal mesh. A local patch is centered about each of these vertices or faces. We define this face or vertex as a *focus* face or vertex in our discussion of the curvature hierarchy. The size of this neighbourhood patch depends on the resolution of the curvature hierarchy we are inspecting.

The first step in building the curvature hierarchy involves finding the neighbouring faces and vertices for each face and vertex of the reference triangle mesh.

We have developed an efficient algorithm for finding the neighbouring faces for each face in a polygonal indexed face set. This technique involves hashing the squared distance of all edges of each triangle and sorting this edge list. The squared distance is chosen as it is a computationally cheap approximation for the edge gradient.

Those faces whose edges have the same hash value are candidate neighbours. As we cannot guarantee that each edge will produce a unique hash value we must explicitly examine the triangles associated with equivalent hash values to determine adjacent triangles. Finding neighbouring vertices is trivial once neighbouring faces have been discovered.

Once neighbouring faces and vertices have been found for each triangle in the polygonal mesh, we can build the local patch for each resolution level.

Without loss of generality we can assume that a patch at the lowest level of the curvature hierarchy covers 1% of the total surface area of the mesh we are examining. Therefore a patch at a focus vertex representing resolution 5 of the curvature hierarchy would cover approximately 5% of the total surface area of the mesh.

The surface area of a non-intersecting triangular mesh or triangular patch is calculated using the following formula [60]:

$$S = \sum_{i=1}^{N_T} \frac{\|(x_{i,1} - x_{i,0}) \times (x_{i,2} - x_{i,0})\|}{2}$$

where $x_{i,j}$ $j = 0, 1, 2$ is the jth vertex on the ith triangle and N_T is the number of triangles in the triangular mesh or patch.

Faces are added to a focus face's resolution level starting from its immediate neighbours and proceeding incrementally. The faces are sorted according to surface area for each focus vertex's neighbourhood and added incrementally to the local patch until the total surface area for that particular resolution is reached. This ensures that the patches at a specific resolution do not differ significantly in surface area and are therefore of a similar size.

Since we have precalculated the neighbouring faces for each triangle we are able to choose easily and efficiently the triangle to add to the resolution patch of a particular focus face. It is important to note that all the faces of a certain neighbourhood are first added to the resolution level before proceeding to those faces that are more distant neighbours. For example, all the faces that are *2-neighbourhoods* from the focus face are first added before considering those faces that are *3-neighbourhoods* away from the focus face.

The resulting curvature hierarchy consists of a range of triangle patches at different scales for each face of the triangle polygonal mesh. These different resolution levels have been chosen based on a surface area metric. Curvature hierarchy construction occurs as a preprocess and is based on the reference mesh model. However, the curvature hierarchy constructed also applies to the segmented mesh model. This is so that the geometry of target and reference models can be examined in a common framework.

4.2 Measuring Curvature

The geometric measure we have chosen in order to evaluate whether a bone is damaged or not is the curvature of the surface of the three dimensional bone model. A number of surface curvature measures exist [92].

Rather than using a computationally expensive curvature estimate, we use a measure based on the surface normal of a triangle. Since our work will be used in a trauma setting, the efficiency versus speed trade-off was justified as we felt that producing a diagnosis as quickly as possible was a valuable benefit. Of course, in a less time-critical setting, such as orthopedic surgery, an exact measure of curvature could be used. The above mentioned functionality could quite easily be added to the existing code base.

4.2.1 Basic Curvature Measure

The curvature approximation we use is the average difference between a focus face's surface normal and its neighbouring faces' surface normals:

$$crv_f = \frac{1}{N_T} \sum_{i=1}^{N_T} |s_f - s_i|$$

where N_T is the number of neighbouring faces, $i = 1...N_T$ is the index for the neighbouring face, f is the index for the focus face and s is the surface normal for a particular triangle.

This is a local curvature measure and, as discussed above, in order to detect bone fractures at different scales we need a curvature measure that is scale independent and can also give a global curvature measure. In other words, we require a curvature measure that will be useful across a range of differently scaled collections of triangle patches.

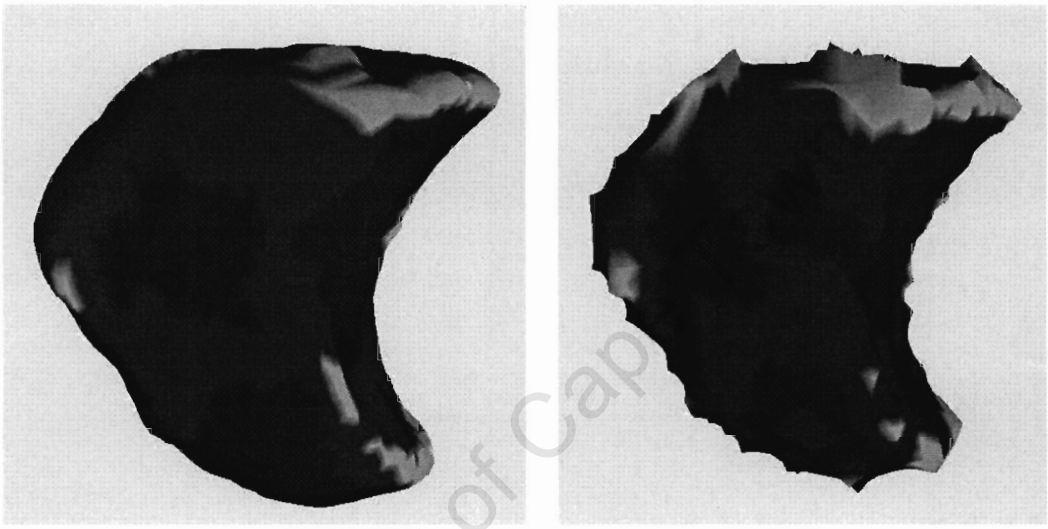
4.2.2 Resolution-Level Curvature Measure

Given our requirements for a curvature measure that quantifies curvature meaningfully across the levels of the curvature hierarchy, we define the *resolution level curvature measure*. This measure is a weighted sum of the difference in surface normal between neighbourhood and focus faces based on all the triangles of a certain patch in a resolution level of the curvature hierarchy. The weighting is based on the surface area that the patch covers, as well as the connectivity of the triangle mesh. The smaller the area a triangle covers the less it adds to the curvature measure. The closer a face is to the focus face the greater its contribution to the curvature measure.

$$crv_{r,f} = \sum_{i=0}^{N_T} a_i b_i |s_f - s_i|$$

where $crv_{r,j}$ is the curvature measure at resolution r for focus face f. N_T is the number of triangles in the resolution level patch, $i = 1...N_T$ is the index for a face in the resolution patch, s is the surface normal for a particular triangle and a and b are the distance and surface area weightings respectively for a certain triangle index.

The resolution-level curvature measure, is able to provide a reasonable measure of curvature across different scales of the curvature hierarchy, since it takes into account mesh connectivity as well as surface coverage.



(a) (b)

Figure 17: **Visualising the Curvature Measured:**

4.3 Fracture Detection

The bone fracture detection phase is a three step process. Firstly, the geometric inconsistencies on the target mesh are located by statistically analysing the curvature differences between the target and reference mesh. Then, geometric inconsistencies are grouped together in order to identify the number and extent of candidate fractures through the use of *cluster analysis*. Finally, these candidate fractures are compared across the different scales of the curvature hierarchy in order to produce a final bone fracture classification.

4.3.1 Detecting Geometric Inconsistencies

In order to find bone fractures we need to detect geometric inconsistencies between the target (segmented) and reference surface meshes. In our approach we use a combination of the resolution level curvature measure, curvature hierarchy and statistics to detect these anomalies.

The algorithm is as follows:

```
1. For each focus face
     • For every resolution level of target and reference mesh
       - Compute the resolution level curvature measure.
2. For each resolution level
     • For every focus face
       - Determine the difference in resolution level curvature
         measure between target mesh and reference mesh.
```

We have used both an L_1 norm (sum of absolute difference) and L_2 norm (sum of squared distance) for measuring the curvature difference.

What we now have is a difference measure for every face at every resolution level of the curvature hierarchy. We are interested in those faces where the difference measure is large. These will be the areas in which the curvature between target and reference model are significantly different and probably indicate parts of the bone surface where there could be geometric inconsistencies.

Statistics is employed to determine these large difference measures in curvature between reference and target models. We assume that the local variation in curvature between bones of the same class follows a Normal Distribution. This signifies that there is a natural curvature variation amongst bones of the same type that is centered about some average differing value. In other words, the distribution of difference measures follows a "bell curve". This assumption is justifiable as it has been shown that the variation in a large amount of biological structures does follow this distribution [72].

We classify the large differences in the curvature measure, which we are trying to identify, as outliers from the distribution function. This signifies that these large difference measures are anomalies that deviate from the expected curvature variation between bones of the same class and therefore require investigation. Those curvature difference values that fall within three standard deviations of the mean value account for 99.9% of the values within the Normal Distribution. We

can thus safely assume that those values that fall outside this range are outliers and hence represent geometric inconsistencies.

4.3.2 Grouping Geometric Inconsistencies

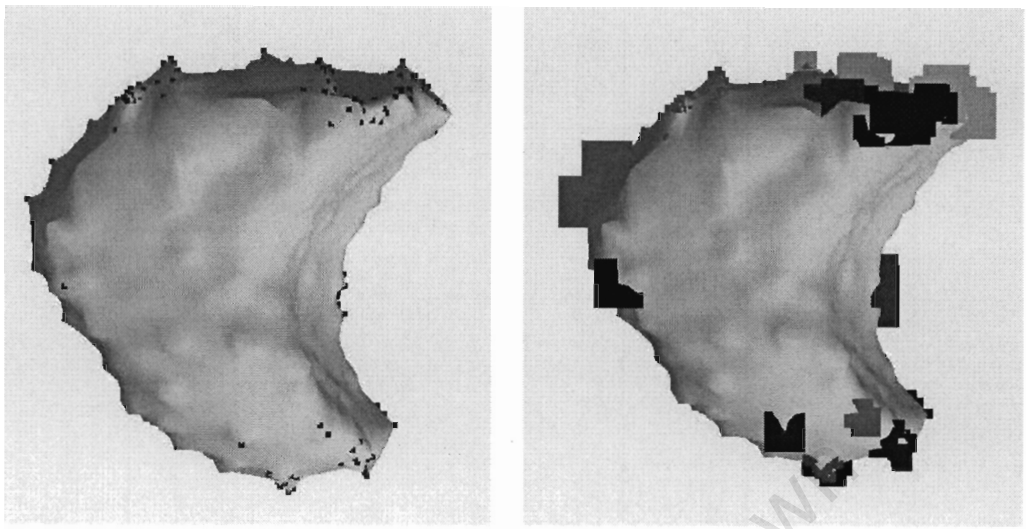
The algorithm described above produces a list of spatial locations of a number of 3D points where geometric inconsistencies occur (see Figure 18a). This calculation serves as a rough estimate of the location of fractures on the surface of the bone we are examining. This is unfortunately not adequate for our needs, as we are interested in a more exact classification of the bone fractures. We would like to identify the groups of geometric inconsistencies that form part of one particular fracture, as we envisage extending the current fracture detection system to be able to identify the type of fracture occurring. This fracture classification would be achieved with the help of an expert knowledge base.

Furthermore, we have discovered that noise in the input data as well as unresolved ambiguities in the segmentation process can result in geometric inconsistencies forming on the bone surface that are not fractures. These particular inconsistencies occur in isolated groups of mostly one or two surface points. If we are able to identify groups of geometric inconsistencies we could discard these small groups of one or two points and thus improve the reliability of the fracture detection system by removing such false positives from the fracture classification process.

We have used *cluster analysis* (See Section 2.5.2) in order to identify the groups of geometric inconsistencies. As the data under consideration, the location of geometric inconsistencies, is spatial data, we use Euclidean distance as our dissimilarity metric. We have chosen UPGMC (Unweighted Pair-Group Method using Centroid Averages) as the amalgamation rule. We make use of Hierarchical Clustering as we are unsure how many fracture there will be and we wish to be able to identify multiple fractures.

The distance based on the average edge length of triangles in the surface mesh is used as the threshold to determine when amalgamation of clusters terminates. The average edge length of triangles provides a unit distance measure on the surface of the polygonal mesh. The number of these units to use as the threshold is inversely proportional to the vertex density of the surface mesh.

The cluster analysis algorithm is as follows:



(a) (b)

Figure 18: **From Geometric Inconsistencies to Fractures:** In (a) the red squares represent the points where geometric inconsistencies have been identified. (b) Each group of colours shows a different bone fracture. This is obviously a pathological case.

1. Initiate every geometrically inconsistent point as a cluster.
2. Determine threshold by
   - Calculating average edge length of triangle mesh.
   - threshold = average edge length * k% of number of vertices
3. For each cluster
   - Calculate distance between own centroid and every other cluster's centroid
4. While distance between nearest 2 clusters centroid < threshold
   - Determine closest two clusters
   - Join these two clusters
   - Calculate centroid of new cluster
   - For each cluster
     - Calculate distance between own centroid and every other cluster's centroid

The cluster analysis algorithm produces distinct groupings of points across the surface of the target mesh. Those groupings that only have one or two members are marked as outliers or misclassifications and are discarded. The remaining clusters, whose members tally above a chosen minimal number, are considered bone fractures and are marked as such (see Figure 18b).

4.4 Summary

In this chapter we presented the new framework we developed for the detection of bone fractures. The basis of this framework is a curvature hierarchy that enables the detection of fractures at different scales/resolution. This enable the detection of both severe and slight fractures. A hybrid surface area metric, that combines both distance and connectivity information, provided the most useful metric with which to build the curvature hierarchy. An approximation of curvature based on the surface normals was chosen to represent the geometric information encoded in the surface model of the segmented bone. This curvature measure in conjunction with the curvature hierarchy were used to find geometric inconsistencies in the surface model. These geometric inconsistencies were then grouped into fractures by means of cluster analysis.

Chapter 5

Evaluation

In this chapter, we describe the processes employed to evaluate our work, as well as presenting the results of this evaluation.

In order to gauge the merits and limitations of the algorithms described in Chapters 3 and 4 we need to evaluate the results produced by these algorithms. For the bone segmentation, this consists of measuring how accurately and speedily we are able to produce a bone segmentation from a CT scan. For the bone fracture detection this consists of evaluating how successfully we are able to detect fractures.

We start this chapter by describing, in Section 5.1, the data sets we have used in our evaluation. We have made use of both computer generated or "synthetic" data sets as well as medical data sets produced by a CT scanner. We also outline how the computer generated data was produced. Then, in Section 5.2, we review the methodology we have used to evaluate our algorithms. We describe both the quantitative and qualitative measures employed to evaluate both the bone segmentation and bone fracture detection frameworks. In Section 5.3, we discuss the findings of the bone segmentation and fracture detection on computer generated data as well as medical data. Finally, in Section 5.4, we draw conclusions regarding the evaluation of our algorithms and discuss the strengths and weaknesses of our approach.

5.1 Data sets

The choice of data sets is an important decision in the evaluation process. Data sets must be chosen which will highlight not only the strengths but also the shortcomings of the process under investigation. We don't only want to know that the process works but also under which conditions it works best and which criteria it is not suited to. In this way, we are able to understand the value of the work produced.

5.1.1 Computer Generated or Simulated Data Sets

Due to the difficulty of obtaining medical data sets that represent a broad-scale of conditions, we have created simulated or computer generated data sets so that a large cross-section of different data sets can be evaluated. Although, computer generated data sets suffer from the shortcoming that they are, in the majority of cases, a simplified version of what occurs in reality, they do provide a basic benchmark of success. Computer generated data allows one to focus on a particular aspect one is trying to evaluate. This provides the opportunity to craft the data for a required specific scenario.

Bone Segmentation

For the bone segmentation we were primarily concerned with measuring the accuracy of the segmentation. In order to gauge the fidelity of the segmentation, a base benchmark is required to measure the segmentation against.

A prototype was constructed which takes a polygonal mesh representation of a bone as input and generates a volumetric representation as output. A ray-casting approach is used to construct the synthetic volumetric data. The approach is as follows: A bounding three dimensional cubic volume is constructed about the bone mesh. e.g a 200x200x200 block. Then, from a point in the centre of each voxel in the XY plane of the cubic volume, a ray is cast along the Z axis. Whenever this ray intersects with a triangle in the mesh, a voxel is marked as filled in the voxel volume. All the filled cells in the voxel volume form a shell which is a volumetric representation of the input polygonal mesh.

This prototype is able to produce voxel volumes at different spatial resolutions and is able to vary the distance between scans by sampling the voxel volume along the acquisition direction. This means that we are able to simulate data sets produced with a high resolution scanner i.e (512x512x512), with a large inter-scan distance. ie. only ten scans. As well as a low resolution scanner (128x128x128) with a small inter-scan distance (50 scans).

In total, computer generated data sets were constructed from 13 bone models. The selection of test models included bones that were long and elongated, such as the ulna and radius as well as compact, round bones such as the capitate and hamate. A wide spectrum of test models was chosen in order to show that the techniques developed were not specific to a particular bone shape but were generalisable to most bone shapes.

Four different spatial resolutions and inter-scan distances were chosen (see Table 1 and Figure 19) in order to evaluate the effect that these factors had on the accuracy of the bone segmentation. We were also interested in how the performance of the algorithms scaled with an increase in the dataset size.

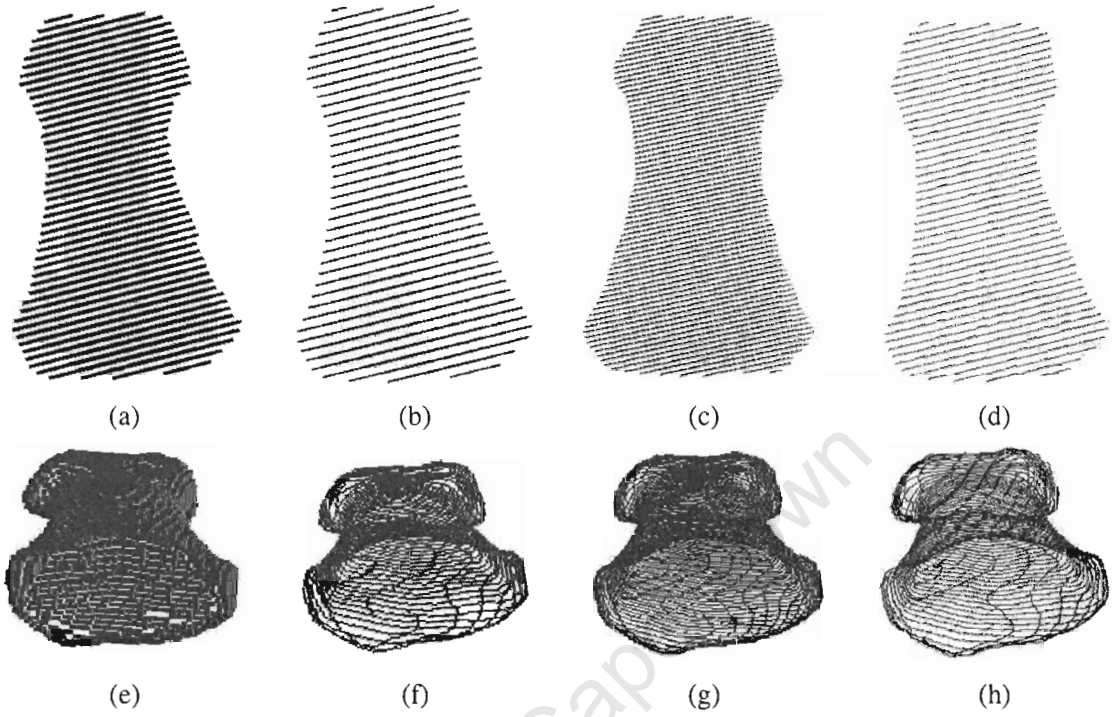


Figure 19: **The Four Groups of Computer Generated Data:** (a),(e) depict the 1 1 2 100 dataset, (b),(f) depict the 1 1 5 200 dataset, (c),(g) depict the 1 1 4 300 dataset and (d),(h) depict the 1 1 8 400 dataset.

Dataset Name	X	Y	Z	Inter-scan distance
1 1 2 100	100	100	100	2 voxels
1 1 5 200	200	200	200	5 voxels
1 1 4 300	300	300	300	4 voxels
1 1 8 400	400	400	400	8 voxels

Table 1: **Dimensions of the Computer Generated Data Groups:** The spatial resolution and interscan distances used to generate the four groups of synthetic bone models. For example, the 1 1 4 300 dataset consists of bone models generated in a 300x300x300 voxel volume with scans taken every 4 voxels

Bone Fracture Detection

We used 3D polygonal mesh editing software, 3D Studio Max, in order to introduce synthetic bone fractures into a triangular mesh. The prototype mentioned in Section 5.1.1 was then used to construct a voxel representation. This was segmented by the bone segmentation prototype to produce

a polygonal mesh that serves as input to the bone fracture detection system. Ten bone models with synthetic fractures were produced. A selection of fractures were introduced that range from single breaks to multiple fracture. Different bone models were also used.

5.1.2 Data from CT Scans

It is also important to evaluate one's work against data produced under real conditions. The data sets produced from image acquisition systems introduce their own set of challenges and complications. These include, amongst others, imaging artifacts produced by the particular type of data acquisition system used.

A full length CT scan of a healthy forearm was used to test the accuracy of the segmentation algorithms. This dataset was called Lindsay, and is a 512x512x21 CT Scan of the Ulna (lindsay_ulna) and Radius (lindsay_radius) acquired on a linear CT Scanner. There is a large inter-scan distance of 10mm between each scan.

Five digitized CT scans (410.10.fib, 41.10.fib, 41.19.fib, 41.1.fib, 412.12.fib) with bone fractures were provided by the Orthopedic Section of the Groote Schuur Hospital in Cape Town. These CT scans are all of fractures in the lower section of the fibula close to the ankle. Most of these fractures are severe as these are the types of fractures that require orthopedic attention. As these are not the raw CT scans, but digitized version of them, there are additional digital imaging artifacts presents. These CT scans were acquired on a linear CT scanner with a resolution of 512x512 in the image plan and interscan distance of 3mm.

Some preprocessing, especially of the digitized data, had to be performed before the datasets could serve as input to the bone segmentation framework. This included removing some of the imaging artifacts introduced by the digitization.

5.2 Evaluation Methodology

We have decided to employ two measures to judge the effectiveness of the bone segmentation and bone fracture detection processes. The first measure is a quantitative analysis, in which we use numerical and statistical analysis to gain an understanding of the effectiveness of our work. The second measure is a qualitative approach, in which we visually assess the results produced by our approach.

5.2.1 Bone Segmentation

For the bone segmentation, we are interested in how accurately we are able to produce a mesh representation of a volumetric volume data set.

Quantitative Analysis

We are able to use numerical means in order to measure the accuracy of the bone segmentation. One means of gauging the accuracy of the segmentation is by looking at the error, in terms of distance, between the input data (voxel data) and the output data (polygonal mesh). This distance error is measured by computing the Euclidean distance between each three-dimensional point in the voxel volume and the closest point on the polygonal representation produced by the segmentation. This evaluates to first, finding the closest triangle in the mesh to a three-dimensional point representing the voxel, and then, calculating the Euclidean distance between that point and the triangle.

This produces a distance error for each voxel in the volumetric data, from which we are able to derive the following useful statistics:

Average Error This indicates on average how accurately the segmentation was performed. A low average error indicates that the segmentation result produced was accurate. The larger the average error, the worse (in accuracy terms) the segmentation produced.

Minimum and Maximum Error Indicates the range of errors produced by the segmentation. A small minimum error and maximum error shows that all the error is contained within a thin band. This would be a positive sign of segmentation accuracy.

Error Variance This statistic measures the distribution of the error about the average error. A large error variance would indicate that the error fluctuates by a large amount from the average error. This would indicate volatility and would be a negative indicator of accuracy.

Coverage Percentage The percent of segmented result voxels within a specified (pixel) distance of the original data. This measure is a good indicator of the overall accuracy of a segmentation. The greater the coverage percentage, the greater the accuracy of the segmentation.

Qualitative Analysis

In terms of qualitative analysis, we will display the result of a segmentation in polygonal mesh form adjacent to the volumetric data that it was produced from, and then visually assess the correctness of the segmentation.

5.2.2 Bone Fracture Detection

For the bone fracture detection, we are interested in whether we are able to detect fractures if they are present in a bone.

Quantitative Analsysis

We have not made use of a quantitative method to evaluate the correctness of the bone fracture detection framework. The medical field makes use of a process known as Receiver Operator Characteristics (ROC) in order to evaluate the accuracy of computer aided tools. ROC is a process that involves a large number of participants and a significant financial and time investment which are beyond the requirements of this dissertation.

Qualitative Analysis

For this reason, we have made use of a purely qualitative method to evaluate the bone fracture detection framework. We display the results of the bone fracture detection and visually gauge the correctness of our approach. This involves pictorially displaying the clusters that are produced by the bone fracture detection framework that indicate the presence of a bone fracture. We have prior knowledge of the location of the fracture and are therefore able to verify visually the success of the bone fracture detection.

5.3 Results

In this section we present the results produced from the bone segmentation and bone fracture detection frameworks.

5.3.1 Segmentation - Computer Generated Data

We begin by examining the results produced by the bone segmentation system on computer generated data.

Quantitative Analysis - Accuracy

Selected segmentation results for the computer generated data are tabulated in Table 2. A full listing of accuracy results for 11 bone models at four resolution levels is available in Appendix A. The average error results indicate that the bone segmentation performed well in terms of accuracy. This is expected as the algorithm performed on data produced in laboratory conditions, and the input data was devoid of the noise and imaging artifacts that are produced by medical acquisition systems. The errors can be largely be attributed to sampling errors as the discrete voxel volume is converted into a surface representation. This is borne out by the fact that in the majority of cases the increase in spatial resolution results in a decrease in the average error, maximum error and variance.

Accuracy Results for the Segmentation of Computer Generated Data						
Bone	**Dist XY**	**Dist Z**	**Avg Error**	**Min Error**	**Max Error**	**Variance**
capitate						
1 1 2 100	0.19	0.38	0.02251633	0	0.12438426	0.001759078
1 1 5 200	0.095	0.475	0.021956305	0.00001482	0.06828771	0.001698006
1 1 4 300	0.063	0.252	0.018107775	0.000001008	0.055249425	0.002145853
1 1 8 400	0.0475	0.38	0.0044973	1.43E-07	0.026402353	0.000285159
hamate						
1 1 2 100	0.19	0.38	0.02920775	0.0000247	0.18720605	0.003628523
1 1 5 200	0.095	0.475	0.022943355	0.00000627	0.0974776	0.003342701
1 1 4 300	0.063	0.252	0.010131345	0.000002646	0.08323308	0.00226619
1 1 8 400	0.0475	0.38	0.00635683	1.57E-06	0.043326888	0.000893244
ulna						
1 1 2 100	2.6	5.2	2.4764194	0.0047034	28.3325224	5.936667207
1 1 5 200	1.3	6.5	1.2292436	0.0004758	16.3102758	4.067299855
1 1 4 300	0.8667	3.4667	0.41642868	0.000230533	7.101892804	0.610180438
1 1 8 400	0.65	5.2	0.3461198	0.000169	4.96721745	0.629838403

Table 2: **Accuracy Results for the Segmentation of Computer Generated Data:** The average, minimum, maximum and variance errors (all measured in millimetres) are displayed for the capitate, hamate and ulna. The pixel distance in the image plane and the interscan distance is displayed by columns Dist XY and Dist Z.

The effect that the inter-scan distance plays on the accuracy of the segmentation, is particularly interesting. A comparison of the 1 1 4 300 and 1 1 8 400 datasets and the 1 1 2 100 and 1 1 5 200 datasets provides interesting insights.

In the case of the 1 1 4 300 and 1 1 8 400 datasets, in many of the bone segmentations the accuracy of the 1 1 4 300 datasets are better than the 1 1 8 400 datasets, even though the spatial resolution in the XY plane is better for the 1 1 8 400 datasets. This is especially true for the long bones such as the fibula and radius.

Although the spatial resolution has been doubled between the 1 1 2 100 and 1 1 5 200 datasets, there is in many cases a small decrease in the average error and variance between the two datasets (e.g capitate, hamate, mip3, scaphoid). The increase in the inter-scan distance has offset some of the gains from increasing the spatial resolution. Although, one can observe that the increase in spatial resolution does reduce the maximum error.

Traditionally, radiologists have placed more emphasis on the spatial resolution in the image plane and less so on the inter-scan distance: the contrast between different matter is what makes CT scans so valuable to radiologists when they make their diagnoses. So, the greater the resolution in the image plane the greater the benefit the radiologist is able to gain. Another reason is to reduce

Coverage Results for the Segmentation of Computer Generated Data								
			Coverage Percentage in Pixels					
Bone	**Dist XY**	**Dist Z**	**1 Pix**	**1.5 Pix**	**2 Pix**	**3 Pix**	**5 Pix**	**10 Pix**
capitate								
1 1 2 100	0.19	0.38	80.5896	94.1036	97.9721	99.8296	100	100
1 1 5 200	0.095	0.475	17.8613	28.4071	40.1457	66.4098	95.7247	100
1 1 4 300	0.063	0.252	13.435	19.267	24.83	35.9673	54.304	96.3386
1 1 8 400	0.0475	0.38	32.0763	46.6068	59.3818	78.3345	94.6498	99.8488
hamate								
1 1 2 100	0.19	0.38	70.0641	84.7222	93.2051	98.3547	99.9573	100
1 1 5 200	0.095	0.475	25.2969	37.3054	48.2977	64.6477	86.9623	99.8944
1 1 4 300	0.063	0.252	37.4743	49.93	58.9076	71.2278	86.634	96.4379
1 1 8 400	0.0475	0.38	32.3397	44.2425	52.9181	65.4966	82.6016	96.3433
ulna								
1 1 2 100	2.6	5.2	94.2457	95.4899	96.7341	98.4448	100	100
1 1 5 200	1.3	6.5	88.0847	93.3804	94.6161	95.3222	97.6169	100
1 1 4 300	0.8667	3.4667	90.8346	95.5083	96.5706	97.4203	98.2701	100
1 1 8 400	0.65	5.2	82.3156	90.8242	93.9273	95.8625	97.2639	99.2993

Table 3: **Coverage Results for the Segmentation of Computer Generated Data:** The coverage percentage at 1, 1.5, 2, 3, 5 and 10 Pixels is displayed for the capitate, hamate and ulna. The pixel distance in the image plane and the interscan distance is displayed by columns Dist XY and Dist Z.

the radiation that the patient is exposed to. A reduction in the number of scans, hence an increase in the inter-scan distance is the method usually employed to reduce the radiation exposure.

The techniques employed in this dissertation focus on the surface of the bone and are less concerned with other details of the scan. For our needs, it is sufficient to distinguish the bone from other matter, and the contrast differences provided by CT provide ample information to complete this task. For this reason, the inter-scan distance plays an important role in a surface based segmentation technique. The introduction of low-dosage image acquisition systems means that patients will be exposed to less radiation per scan and inter-scan distances will be reduced.

The coverage analysis of the segmentation on the computer generated data is also encouraging – See Table 3 for selected results and Appendix A for a full listing. The reduction of coverage percentage as spatial resolution increases is misleading. As spatial resolution increases the pixel size is reduced. The coverage analysis does however bear out the finding that the inter-scan distance plays an important role in the accuracy of the bone segmentation.

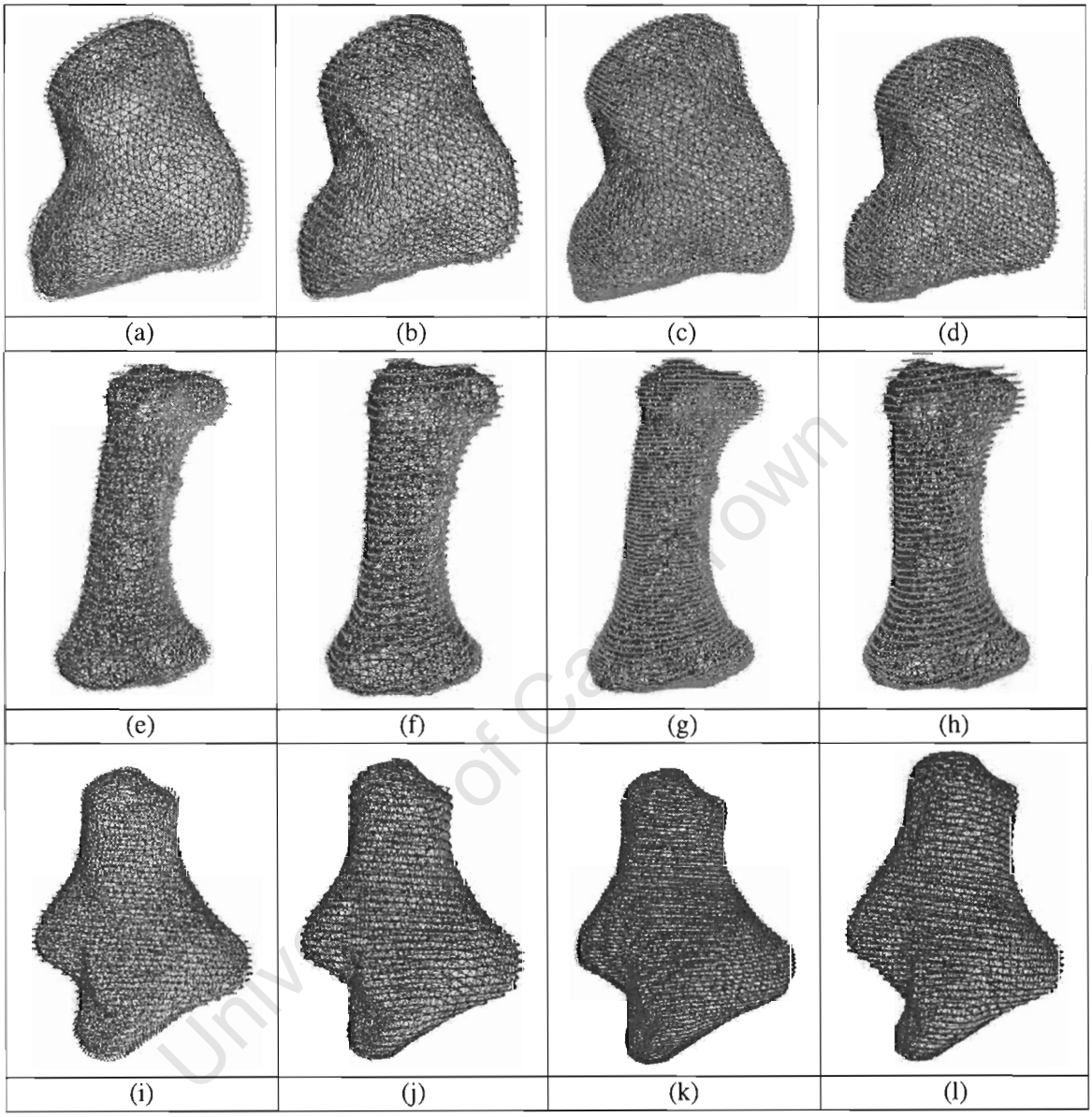


Figure 20: **Segmentation Results for Computer Generated Data:** Segmentations at the four resolution for the capitate (a)-(d), mpp3 (e)-(h) and trapezium (i)-(l) bones. The green represents the (input) voxel volume dataset, while the brown is a wireframe mesh representation of the segmentation achieved.

Quantitative Analysis - Speed

A table of the speed of the segmentation algorithm on the computer generated data can be viewed in Appendix A. Consult Table 4 for an abbreviated version. These results indicate the time taken in

Performance Results for the Segmentation of Computer Generated Data					
Data	**Setup**	**Rough Align**	**Global Alignment**	**Local Alignment**	**Total**
capitate 100 1 1 2	0.2	0	20.3	1.43	21.93
fibula 100 1 1 2	0.09	0	5.56	0.17	5.82
mpp3 100 1 1 2	0.1	0	8.24	0.66	9
capitate 200 1 1 5	0.71	0	38.42	2.22	41.35
fibula 200 1 1 5	0.11	0	8.15	0.27	8.53
mpp3 200 1 1 5	0.29	0	14.6	1.07	15.96
capitate 300 1 1 4	2.27	0.01	232.08	9.72	244.08
fibula 300 1 1 4	0.14	0	25.35	0.65	26.14
mpp3 300 1 1 4	0.85	0	53.69	2.95	57.49
capitate 400 1 1 8	4.91	0.01	174.3	8.47	187.69
fibula 400 1 1 8	0.2	0.01	22.73	0.62	23.56
mpp3 400 1 1 8	1.77	0.01	46.05	2.58	50.41

Table 4: **Performance Results for Computer Generated Data:** This table displays the time spent (measured in seconds) in each phase of the segmentation process.

seconds to complete the various steps in the segmentation process.

As can be observed, most of the processing time is spent in the global alignment phase of the segmentation process. This is due to the fact that the global alignment algorithms requires the matching of points in the two datasets, which is a computationally intensive process.

Qualitative Analysis

The visual results of the segmentation on synthetic data, which can be viewed in Figure 20 and Appendix A, confirm the success of the techniques employed.

The segmentation process managed to produce good results for all four groups of input data. The compact, round bones faired especially well. The long, elongated bones on the other hand performed less well. In particular, there were noticeable errors at the extremities of the long bones — see Figure 21. At these extremities, there is a great deal more structural detail compared to the rest of the bone. This problem could be due to the fact that there should be more sampling at these extremities compared to less detailed sections of the bone.

The patella was not recovered by our segmentation technique due to ambiguities in the rough alignment phase of the segmentation process — see Figure 22. Knowledge of the human skeleton and in particular the physical location of the patella in relation to surrounding bones can be used to resolve such ambiguities. Unfortunately, time constraints have meant that this check has not been included in the segmentation process.

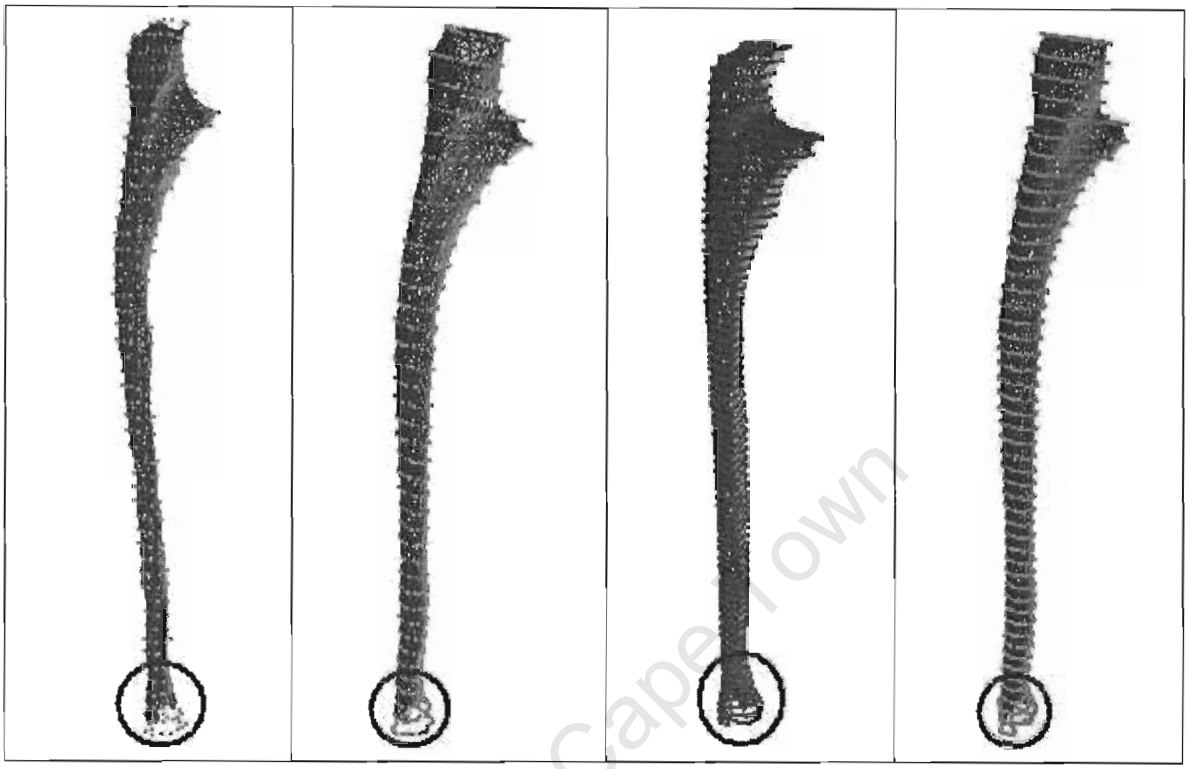


Figure 21: **Inaccuries in the Bone Segmentation:** The extremities of long, elongated bones can cause inaccuracies in the segmentation process.

5.3.2 Segmentation - Medical Data

In this section, we examine how well the bone segmentation framework performed on medical data.

Quantitative Analysis - Accuracy

The accuracy results for the bone segmentation on medical data is satisfactory — See Table 5. The average error for the digitised CT scans was +- 2 pixels which is acceptable when once consider the additional imaging artifacts introduced by the digitisation. The 410.10.fib and 41.19.fib datasets, in particular, were accurately segmented.

For the CT scans, the accuracy of the segmentation achieved for the lindsay_ulna dataset is very good. Whereas the result for the segmentation of the lindsay_radius dataset is less accurate. There was some ambiguity in the pose estimation of this dataset which accounted for the errors in the segmentation. As we have mentioned, prior knowledge of the human skeleton could help resolve

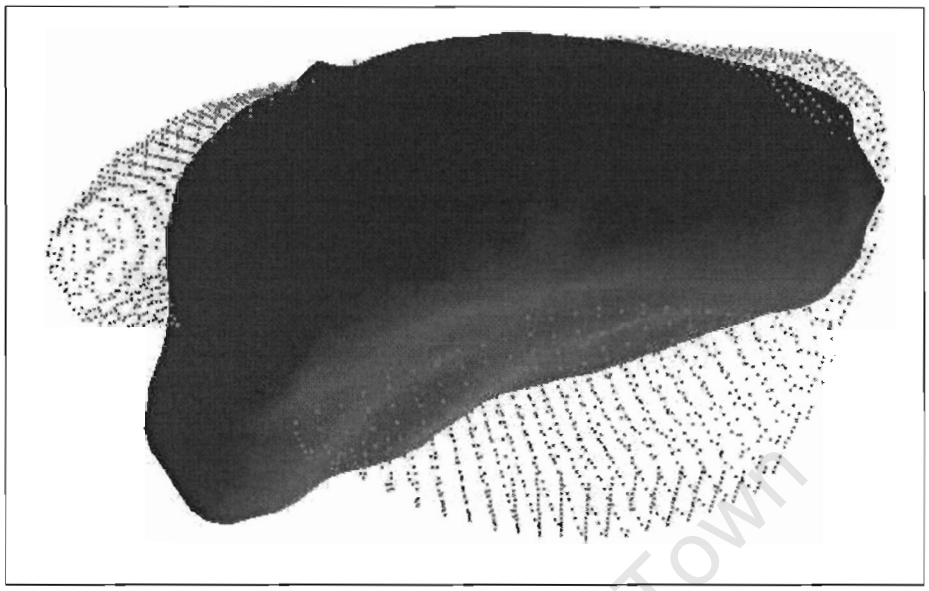


Figure 22: **Ambiguity in the Rough Alignment:** The patella failed to be segmented due to an ambiguity in the rough alignment phase of the segmentation process

Accuracy Results for the Segmentation of Medical Data						
Bone	**Dist XY**	**Dist Z**	**Avg Error**	**Min Error**	**Max Error**	**Variance**
410.10.fib	0.46875	3	0.475478906	0.000295	1.882443	0.309517
41.10.fib	0.46875	3	1.076587969	0.007053	2.926599	0.633197
41.19.fib	0.46875	3	0.574169063	0.003103	2.241264	0.461007
41.1.fib	0.46875	3	1.171682813	0.000202	4.099436	1.715978
412.12.fib	0.46875	3	0.815517656	0.002522	2.762238	0.761725
lindsay_radius	0.46875	10	1.430293125	0.000678	5.298423	4.276039
lindsay_ulna	0.46875	10	0.324226406	0.000439	1.828086	0.162956

Table 5: **Accuracy Results for the Segmentation of Medical Data:** The average, minimum, maximum and variance errors (all measured in millimetres) are displayed. The pixel distance in the image plane and the interscan distance is displayed by columns Dist XY and Dist Z.

such ambiguities.

The coverage results for the segmentation on medical data (– See Table 6) indicate that the segmentation of medical data proceeded less accurately than for computer generated data. In particular, coverage results indicate that the 41.10.fib dataset and the 41.1.fib datasets only achieved a 50% coverage at a 5 pixel tolerance. On the other hand, the coverage results for lindsay_ulna, 410.10.fib and 41.19 datasets confirm the accuracy of their segmentations.

Coverage Results for the Segmentation of Medical Data								
			Coverage Percentage in Pixels					
Bone	**Dist XY**	**Z**	**1 Pix**	**1.5 Pix**	**2 Pix**	**3 Pix**	**5 Pix**	**10 Pix**
410.10.fib	0.46875	3	30.08982	46.70659	57.3353	72.6047	90.8682	100
41.10.fib	0.46875	3	6.505295	9.379728	14.8260	23.7518	50.0756	97.579
41.19.fib	0.46875	3	24.57003	39.47584	51.5151	65.2743	84.7665	99.91
41.1.fib	0.46875	3	12.69544	19.26204	23.76485	35.3345	54.4090	86.804
412.12.fib	0.46875	3	18.53526	25.67812	32.8209	48.2821	69.8010	97.830
lindsay_radius	0.46875	10	15.28926	24.07025	33.5227	48.3471	58.6776	72.469
lindsay_ulna	0.46875	10	44.90862	60.31332	71.6971	89.2950	98.1201	100

Table 6: **Coverage Results for the Segmentation of Medical Data:** The coverage percentage at 1, 1.5, 2, 3, 5 and 10 Pixels is displayed. The pixel distance in the image plane and the interscan distance is displayed by columns Dist XY and Z.

Performance Results for the Segmentation of Medical Data					
Data	**Setup**	**Rough Alignment**	**Global Alignment**	**Local Alignment**	**Total**
410.10.fib	0.01	0	5.72	0.04	5.77
41.1.fib	0.04	0	13.19	0.06	13.29
41.10.fib	0.02	0	5.67	0.03	5.72
41.19.fib	0.02	0	9.76	0.04	9.82
412.12.fib	0.03	0	7.52	0.05	7.6
lindsay_radius	0.58	0	5.28	0.44	6.3
lindsay_ulna	0.58	0	4.16	0.36	5.1

Table 7: **Performance Results for Medical Data:** This table displays the time spent (measured in seconds) in each phase of the segmentation process.

Quantitative Analysis - Speed

Table 7 tabulates the performance of the various steps in the segmentation algorithms. This table confirms the speed of the segmentation algorithm. The rigid alignment algorithm accounts for the majority of the processing time which corresponds to our findings for computer generated data – See Section 5.3.1.

Qualitative Analysis

Figure 23 displays the segmentation results for the digitized scans, while Figure 24 displays the segmentation results for the CT scan. From these figures we can clearly see that an acceptable segmentation was achieved for the datasets.

The errors on the extremities of the datasets are also evident from these graphics. This is caused

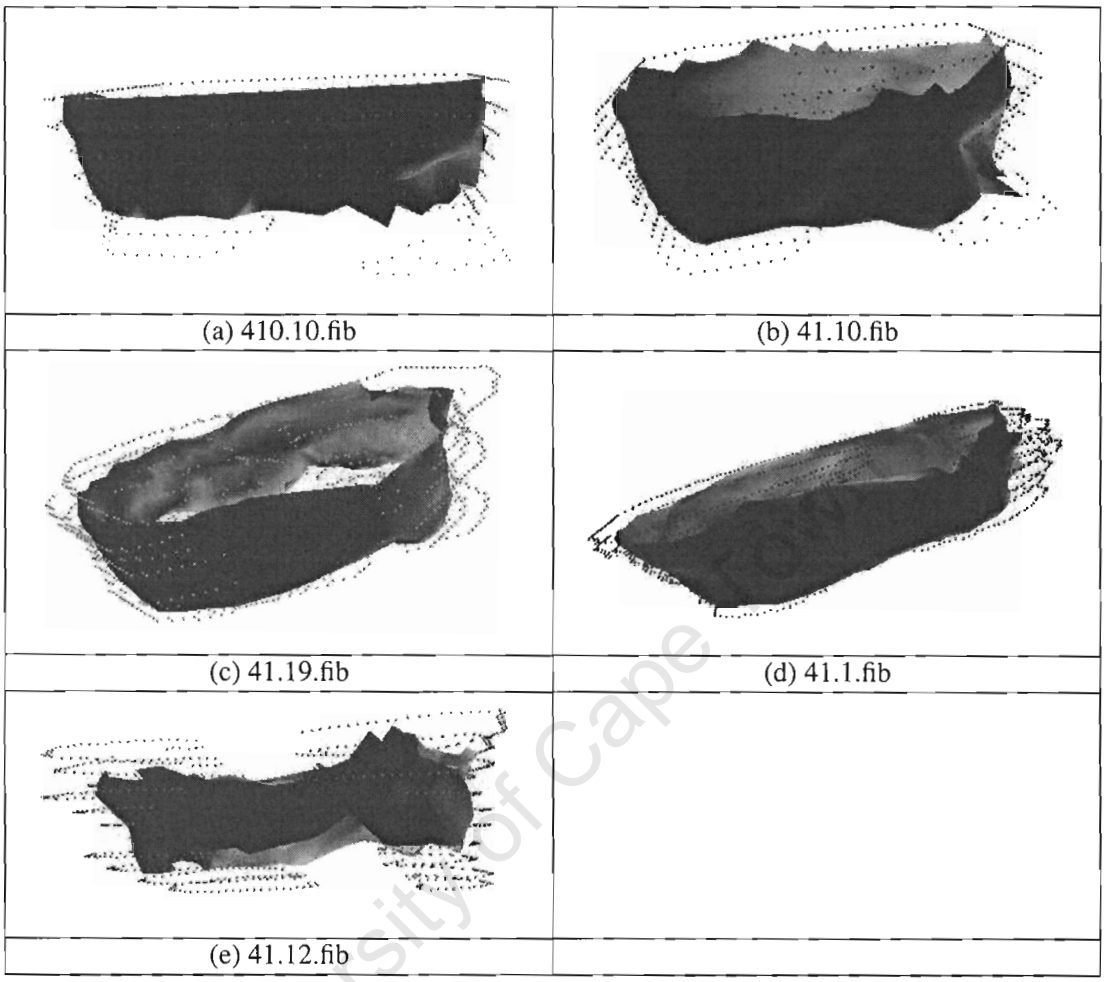


Figure 23: **Segmentation Results for Digitized CT Data:** Accurate segmentations were achieved for datasets (a) and (c). Acceptable segmentations were achieved for datasets (b), (d) and (e).

by a gravitation of mesh points towards the centre of the voxel volume. This results in the volume of the result mesh being less than the voxel volume. A constraint that helps ensure that the volumes of the input voxel volume and resulting mesh are kept roughly equal might help eliminate this problem. This shrinking does not adversely effect the bone fracture detection, as in our work, we are concerned with maintaining the geometric structure of the bone in the segmentation. A slight shrinking of the retrieved segmentation is thus acceptable for our needs, as long as the geometric integrity is not compromised.

What is also evident from Figure 23 is that some sort of topology alteration would improve the

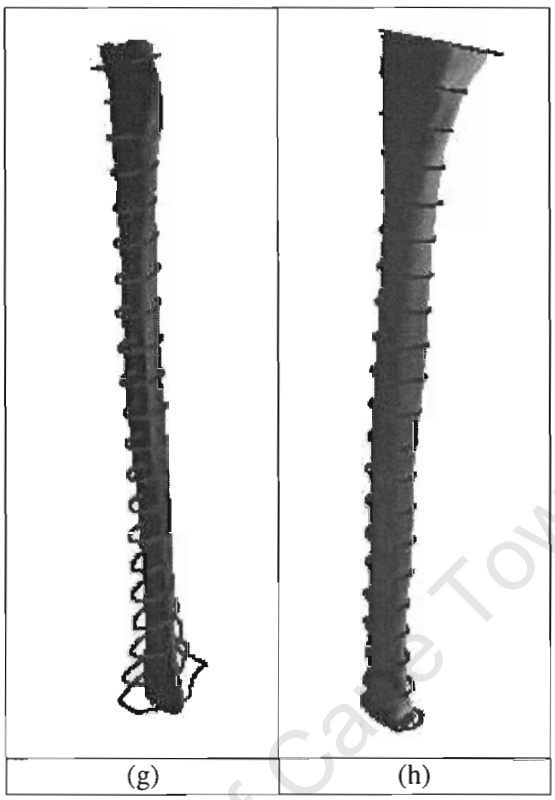


Figure 24: **Segmentation Results for CT Data:** The segmentations achieved on CT Data produced from a Linear CT scanner.

segmentation. This would greatly enhance the segmentation, especially for severe breaks such as the one depicted in Figure 23(e), and would most probably aid in ensuring accuracy in the bone fracture detection phase. This is however a non trivial task and beyond the scope of this work and could be pursued in future work.

5.3.3 Bone Fracture Detection - Computer Generated Data

In this section we examine the results produced by the bone fracture detection system on computer generated data.

Qualitative Analysis

Fracture detection was very accurate for the computer generated data as can be seen from Figure 25. This can be attributed to the high-quality segmentation produced by the bone segmentation

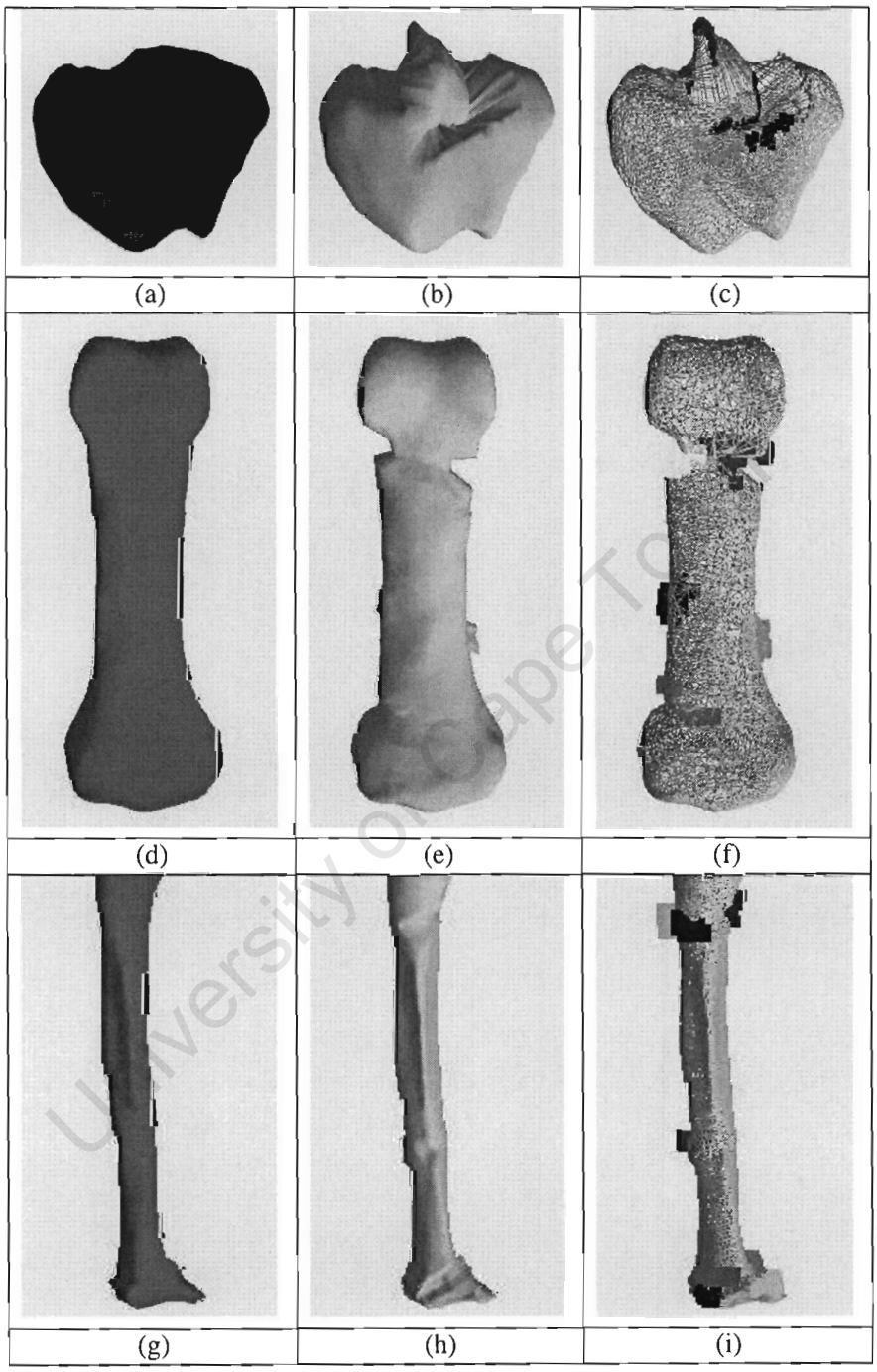


Figure 25: **Bone Fracture Detection Results for Computer Generated Data:** (a),(d),(g) the reference bone mesh, (b),(e),(h) the bone fractures in the computer generated data, (c),(f),(i) the bone fractures are highlighted by the bone fracture detection framework.

framework. This exact segmentation was possible due to the absence of image degradation in the input data as it was computer generated. As a large amount of structural information was retained in the segmentation, this eased the task of the fracture detection system.

From the figures we can ascertain that the bone fracture detection framework is able to detect multiple fractures. Also, it is evident that fractures of varying severity can be detected by our system.

5.3.4 Bone Fracture Detection - Medical Data

We present the results of the bone fracture detection system on medical data in this section.

Qualitative Analysis

The results produced for the bone fracture detection on the medical data can be viewed in Figure 26. From these pictures it is evident that the bone fracture detection algorithm is able to detect the fractures present.

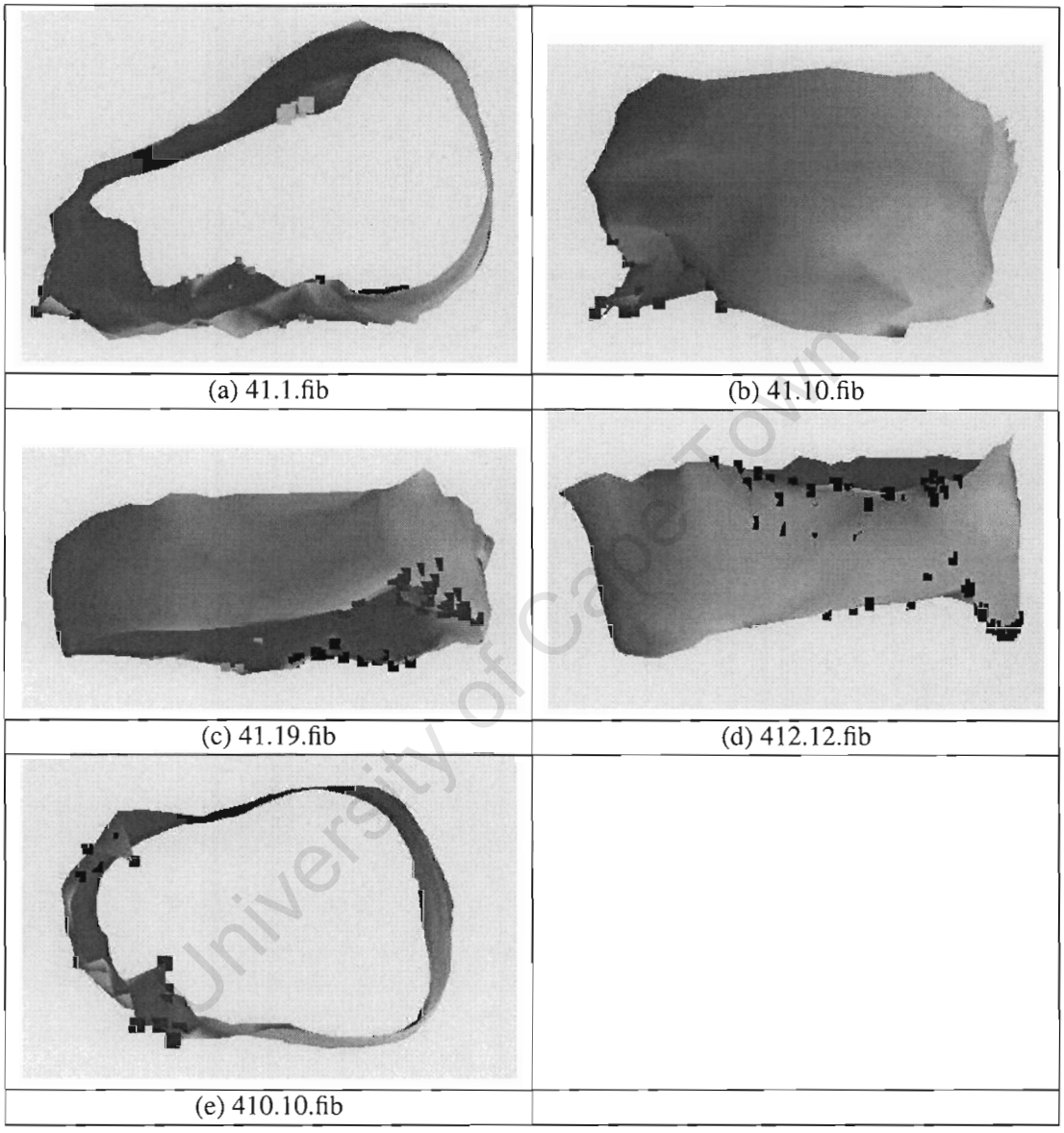
In Figure 26(a), the red blocks indicate a bone fracture, which is not present in the CT Data. This is due to the poor segmentation of this part of the bone. This result helps to enforce the fact that the correctness of the bone fracture detection is tied to the accuracy of the segmentation achieved.

5.4 Conclusion

In this chapter we have presented the results of the bone segmentation and bone fracture detection algorithms. Results were generated for both synthetic computer generated data as well as medical data from a CT scanner. Quantitative and qualitative means were used to evaluate the algorithms.

The bone segmentation algorithms were evaluated in terms of speed and accuracy. The segmentation results were also visually assessed for correctness. The results indicated that the segmentation algorithms performed speedily for both medical and synthetic data. The accuracy results for computer generated data was excellent, while the results for medical data were acceptable. A number of limitations in the segmentation algorithms were also highlighted.

The fracture detection results for computer generated data were very good. This indicates that the novel algorithms developed are effective at detecting bone fractures. And in this, lies the greatest contribution of this dissertation. The results for medical data were less pleasing, this was largely due to the accuracy of the segmentation used, rather than the failure of the bone fracture detection framework.



(a) 41.1.fib (b) 41.10.fib

(c) 41.19.fib (d) 412.12.fib

(e) 410.10.fib

Figure 26: **Bone Fracture Detection Results for Medical Data:** The bone fractures are indicated by groupings of coloured rectangles.

Chapter 6

Conclusion

In this dissertation we presented a framework to detect bone fractures from CT data. We have shown that there is sufficient structural information present in a polygonal surface representation built from a Computed Tomography (CT) scan that we are able to detect inconsistencies in the integrity of the bone structure by geometric means.

We adopted a two phase approach to this problem, a segmentation phase and bone fracture detection phase. The segmentation phase consisted of extracting a polygonal representation of a bone from CT data. This was followed by the bone fracture detection phase, in which we constructed a framework through which we were able to examine the geometric structure of the polygonal representation in order to assess the fidelity of the bones.

6.1 Bone Segmentation

In order to deal with the artefacts present in imaging data we made us of a model based segmentation process in which a reference model guided and constrained the segmentation process to produce an accurate segmentation. A coarse to fine approach was used to extract a polygonal representation from the CT data. What this entailed was that an initial rough approximation was produced and that this was refined at each step of the segmentation algorithm. After initial image processing to reduce computational and memory overheads, a coarse alignment step was used to bring reference and target models into a global alignment. Three coarse alignment algorithms were developed to complete this task. An improved global registration was then achieved through the use of the ICP algorithm [2]. A local deformation was applied to account for the local variation in the bones of different patients. The contribution of the segmentation phase is the synthesis of a number of algorithms to produce a polygonal representation of a bone from CT data speedily.

We have also shown that this process can be speedily completed and is thus appropriate in a

trauma setting.

6.2 Fracture Detection Framework

We developed a new fracture detection framework. The foundation of this framework is a curvature hierachy which enables the detection of different degrees of fractures by examining the geometry of a polygonal mesh at different scales. A hybrid metric, the surface area metric, that combined both distance and connectivity information provided the most useful metric with which to build the curvature hierarchy. We chose the surface normal as an approximation of curvature to represent the geometric information encoded in the polygonal representation of the bone. This curvature measure in conjunction with the curvature hierachy were used to find geometric inconsistencies in the surface model. These geometric inconsistencies were then grouped into fractures by means of cluster analysis.

6.3 Evaluation

We evaluated the bone segmentation and bone fracture detection algorithms we developed. Results were generated for both synthetic computer generated data as well as medical data from a CT scanner. Synthetic data was used to evaluate our algorithms against a known test-bed, while the medical data was used to evaluate the effectiveness of our algorithms against real-world data. Quantitative and qualitative means were used to evaluate the algorithms.

The bone segmentation algorithms were evaluated in terms of speed and accuracy (quantitative analysis). The segmentation results were also visually assessed for correctness (qualitative analysis). The results indicated that the segmentation algorithms performed speedily for both medical and synthetic data. The accuracy results for computer generated data were excellent, while the results for medical data were acceptable. A number of limitations in the segmentation algorithms were also highlighted.

The fracture detection results for computer generated data were very good. This indicates that the novel algorithms developed are effective at detecting bone fracture. And in this, lies the greatest contribution of this dissertation. The results for medical data were less pleasing, this was largely due to the accuracy of the segmentation used, rather than the failure of the bone fracture detection framework. The following section expounds on a number of improvements that can be made to alleviate these short-comings.

6.4 Future Work

The system developed is able to produce a surface representation from CT Data, and detect the bone fractures from the geometric information encoded in its structure. However, the results produced have highlighted a number of areas that could be addressed which would lead to an improvement in the algorithms.

6.4.1 Bone Segmentation

The bone segmentation could be greatly improved if the topology of the surface mesh could be altered. This is especially true when there is a severe bone fracture present in the CT scan, as this generally results in the bone fracturing into a number of seperate pieces. A topology alteration would be especially useful in improving the accuracy of the segmentation.

Presently, segmentation process is memory intensive. This limits the size of datasets that can be segmented. An on demand memory loading scheme, when data is only loaded when it is required, could alleviate this problem and enable the segmentation of large datasets.

At the moment, knowledge of the human skeleton is not used to resolve ambiguities during the segmentation. The physical location of bones in relation to neighbouring bones could provide valuable information which could significantly improve the accuracy of the segmentation.

Constraints that prevent a gravitation of points towards the center of the mesh would help reduce the errors present at the extremeties of the segmentations. A constraint based on volume could help prevent shrinkage in the surface model produced by the segmentation.

Performance results (See Sections 5.3.1 and 5.3.2) indicate that a large proportion of the computation time for the segmentation is spent in the local alignment. A parallellised version of the ICP [2], which exists but is not implemented, would improve the time taken to achieve a segmentation.

There is scope for an improvement in the local deformation phase of the segmentation algorithm. At present, only one algorithm based on Free Form Deformation has been examined. Although this algorithm has produced acceptable results, it is felt that improvements could be made if alternative algorithms were investigated.

6.4.2 Bone Fracture Detection

The bone fracture detection is unable to classify bone fractures. Bone fracture classification could prove helpful in suggesting an appropriate treatment. An expert knowledge base of bone fracture in conjunction with geometric criteria could make this attainable.

Appendix A

Results

In this appendix, the full set of results are displayed.

Accuracy Results for the Segmentation of Computer Generated Data						
Bone	**Dist XY**	**Dist Z**	**Avg Error**	**Min Error**	**Max Error**	**Variance**
capitate						
1 1 2 100	0.19	0.38	0.02251633	0	0.12438426	0.001759078
1 1 5 200	0.095	0.475	0.021956305	0.00001482	0.06828771	0.001698006
1 1 4 300	0.063	0.252	0.018107775	0.000001008	0.055249425	0.002145853
1 1 8 400	0.0475	0.38	0.0044973	1.43E-07	0.026402353	0.000285159
fibula						
1 1 2 100	2.9	5.8	1.4800614	0.0000319	23.1876663	3.165483312
1 1 5 200	1.45	7.25	0.8859761	0.00001015	11.6146189	1.343735495
1 1 4 300	0.9667	3.8668	0.429410073	5.90E-05	3.713474613	0.205244034
1 1 8 400	0.725	5.8	0.564774275	0.000416875	4.465497575	0.480896586
hamate						
1 1 2 100	0.19	0.38	0.02920775	0.0000247	0.18720605	0.003628523
1 1 5 200	0.095	0.475	0.022943355	0.00000627	0.0974776	0.003342701
1 1 4 300	0.063	0.252	0.010131345	0.000002646	0.08323308	0.00226619
1 1 8 400	0.0475	0.38	0.00635683	1.57E-06	0.043326888	0.000893244
lunate						
1 1 2 100	0.22	0.44	0.02995696	0.00000814	0.23814934	0.003784881
1 1 5 200	0.11	0.55	0.01525579	0.0000011	0.11664818	0.002152002
1 1 4 300	0.073	0.292	0.009428023	0.000001095	0.07342851	0.00138615
1 1 8 400	0.055	0.44	0.00689799	0.000000825	0.0595683	0.001024699
mip3						
1 1 2 100	0.22	0.44	0.0334345	0.00000286	0.15446024	0.003056621
1 1 5 200	0.11	0.55	0.03080715	0.00000913	0.13611741	0.006088246
1 1 4 300	0.073	0.292	0.010581934	0.000001022	0.067607855	0.001316302
1 1 8 400	0.055	0.44	0.00803715	0.000005775	0.042654315	0.000889912
mpp3						
1 1 2 100	0.22	0.44	0.03140852	0.00000528	0.16683788	0.002587279
1 1 5 200	0.11	0.55	0.02164459	0.00000704	0.11979594	0.00280483
1 1 4 300	0.073	0.292	0.008667436	0.000001679	0.038532247	0.000483849
1 1 8 400	0.055	0.44	0.00834031	0.00000253	0.07235349	0.001063599
radius						
1 1 2 100	1.9	3.8	1.5802433	0.0006403	12.1033534	2.435623828
1 1 5 200	0.95	4.75	0.53639375	0.00008835	7.69471785	1.388260527
1 1 4 300	0.63	2.52	0.21042567	0.00000567	2.05880409	0.076582588
1 1 8 400	0.475	3.8	0.190459325	0.000103075	2.9075035	0.301169915

Accuracy Results for the Segmentation of Computer Generated Data (continued)						
Bone	**Dist XY**	**Dist Z**	**Avg Error**	**Min Error**	**Max Error**	**Variance**
scaphoid						
1 1 2 100	0.25	0.5	0.02321175	0.00001075	0.11808525	0.001244293
1 1 5 200	0.125	0.625	0.01793975	0.000014625	0.07201725	0.001082136
1 1 4 300	0.0833	0.3333	0.002048918	0	0.019959035	5.40E-05
1 1 8 400	0.0625	0.5	0.005218188	2.06E-06	0.029425	0.000292578
trapezium						
1 1 2 100	0.22	0.44	0.02227786	0	0.0912252	0.001279327
1 1 5 200	0.11	0.55	0.01960321	0.00000572	0.06171198	0.001478362
1 1 4 300	0.073	0.292	0.0017739	0	0.012748866	3.90E-05
1 1 8 400	0.055	0.44	0.004633695	0.000001485	0.023084435	0.00026428
trapezoid						
1 1 2 100	0.17	0.34	0.01694339	0.00000969	0.10174415	0.001367197
1 1 5 200	0.085	0.425	0.00836162	0.00000085	0.060561225	0.000754218
1 1 4 300	0.0567	0.2268	0.00475197	2.27E-07	0.0374732	0.000399794
1 1 8 400	0.0425	0.34	0.00370532	2.98E-07	0.03184457	0.000340624
triquetral						
1 1 2 100	0.15	0.3	0.0139476	0.00000405	0.0631785	0.000704686
1 1 5 200	0.075	0.375	0.0024624	0	0.0154218	4.72E-05
1 1 4 300	0.05	0.2	0.0075748	0.0000014	0.04680275	0.000673042
1 1 8 400	0.0375	0.3	0.003169388	0.000001125	0.014707875	0.000151309
ulna						
1 1 2 100	2.6	5.2	2.4764194	0.0047034	28.3325224	5.936667207
1 1 5 200	1.3	6.5	1.2292436	0.0004758	16.3102758	4.067299855
1 1 4 300	0.8667	3.4667	0.41642868	0.000230533	7.101892804	0.610180438
1 1 8 400	0.65	5.2	0.3461198	0.000169	4.96721745	0.629838403

Accuracy Results for the Segmentation of Medical Data						
Bone	**Dist XY**	**Dist Z**	**Avg Error**	**Min Error**	**Max Error**	**Variance**
410.10.fib	0.46875	3	0.475478906	0.000295	1.882443	0.309517
41.10.fib	0.46875	3	1.076587969	0.007053	2.926599	0.633197
41.19.fib	0.46875	3	0.574169063	0.003103	2.241264	0.461007
41.1.fib	0.46875	3	1.171682813	0.000202	4.099436	1.715978
412.12.fib	0.46875	3	0.815517656	0.002522	2.762238	0.761725
lindsay_radius	0.46875	10	1.430293125	0.000678	5.298423	4.276039
lindsay_ulna	0.46875	10	0.324226406	0.000439	1.828086	0.162956

Coverage Analysis for the Segmentation of Computer Generated Data								
			Coverage Percentage in Pixels					
Bone	**Dist XY**	**Dist Z**	**1 Pix**	**1.5 Pix**	**2 Pix**	**3 Pix**	**5 Pix**	**10 Pix**
capitate								
1 1 2 100	0.19	0.38	80.5896	94.1036	97.9721	99.8296	100	100
1 1 5 200	0.095	0.475	17.8613	28.4071	40.1457	66.4098	95.7247	100
1 1 4 300	0.063	0.252	13.435	19.267	24.83	35.9673	54.304	96.3386
1 1 8 400	0.0475	0.38	32.0763	46.6068	59.3818	78.3345	94.6498	99.8488
fibula								
1 1 2 100	2.9	5.8	96.875	97.6563	98.4375	100	100	100
1 1 5 200	1.45	7.25	93.48711	94.979647	96.3365	97.829	99.7286	100
1 1 4 300	0.9667	3.8668	92.4347	97.249	98.3494	99.0371	100	100
1 1 8 400	0.725	5.8	60.404	80.5051	90.303	95.5051	97.2727	100
hamate								
1 1 2 100	0.19	0.38	70.0641	84.7222	93.2051	98.3547	99.9573	100
1 1 5 200	0.095	0.475	25.2969	37.3054	48.2977	64.6477	86.9623	99.8944
1 1 4 300	0.063	0.252	37.4743	49.93	58.9076	71.2278	86.634	96.4379
1 1 8 400	0.0475	0.38	32.3397	44.2425	52.9181	65.4966	82.6016	96.3433
lunate								
1 1 2 100	0.22	0.44	82.0678	92.3077	96.311	99.1232	100	100
1 1 5 200	0.11	0.55	56.5418	72.2627	81.1083	90.5593	97.845	100
1 1 4 300	0.073	0.292	46.4789	61.4623	70.4808	81.5482	92.6448	99.4395
1 1 8 400	0.055	0.44	35.5583	52.1513	63.4724	76.184	87.82	97.9346
mip3								
1 1 2 100	0.22	0.44	76.6287	91.8838	97.2041	99.8914	100	100
1 1 5 200	0.11	0.55	27.359	41.4046	51.4395	67.0661	85.9211	99.6339
1 1 4 300	0.073	0.292	38.093	52.1947	63.2658	77.7811	91.9478	99.7105
1 1 8 400	0.055	0.44	26.4811	38.8204	49.8807	67.104	85.2419	98.8535
mpp3								
1 1 2 100	0.22	0.44	76.6253	94.4006	99.0605	99.8497	100	100
1 1 5 200	0.11	0.55	35.5591	51.3835	64.0521	82.9179	95.9982	100
1 1 4 300	0.073	0.292	32.8501	50.7168	68.1704	88.6045	98.9309	100
1 1 8 400	0.055	0.44	20.9653	32.7406	45.015	66.1163	88.4423	97.8227
radius								
1 1 2 100	1.9	3.8	93.1034	94.8276	95.8333	97.8448	100	100
1 1 5 200	0.95	4.75	92.3077	94.7064	95.6989	96.6915	97.3532	100
1 1 4 300	0.63	2.52	88.7848	96.025	97.7286	99.0062	99.9716	100
1 1 8 400	0.475	3.8	80.4147	92.7273	95.4386	97.0016	97.799	98.2456

Coverage Analysis for the Segmentation of Computer Generated Data (continued)								
			Coverage Percentage in Pixels					
Bone	**Dist XY**	**Dist Z**	**1 Pix**	**1.5 Pix**	**2 Pix**	**3 Pix**	**5 Pix**	**10 Pix**
scaphoid								
1 1 2 100	0.25	0.5	96.862	99.7714	100	100	100	100
1 1 5 200	0.125	0.625	47.0732	70.3975	85.3081	98.9965	100	100
1 1 4 300	0.0833	0.3333	97.2144	99.4803	99.9134	100	100	100
1 1 8 400	0.0625	0.5	46.1511	65.0783	78.9385	92.6656	98.5004	100
talus								
1 1 2 100	0.64	1.28						
1 1 5 200	0.32	1.6	39.3805	56.4583	68.2546	86.2832	99.8211	100
1 1 4 300	0.2133	0.8533	47.9259	63.9513	75.7751	88.57	98.308	100
1 1 8 400	0.16	1.28	40.1782	54.6731	67.5728	82.8728	96.0009	99.9663
trapezium								
1 1 2 100	0.22	0.44	91.4991	99.0829	100	100	100	100
1 1 5 200	0.11	0.55	33.2354	51.0668	66.1224	88.0361	99.9782	100
1 1 4 300	0.073	0.292	96.5401	99.4748	99.9266	100	100	100
1 1 8 400	0.055	0.44	43.1675	58.4457	70.324	86.4695	98.5125	100
trapezoid								
1 1 2 100	0.17	0.34	82.8512	93.073	97.332	99.7428	100	100
1 1 5 200	0.085	0.425	57.4208	72.0143	82.4283	92.4749	99.0271	100
1 1 4 300	0.0567	0.2268	49.8716	63.8839	74.5787	86.6702	96.0633	99.9683
1 1 8 400	0.0425	0.34	40.3325	52.6869	62.4235	77.2022	90.7293	99.1077
triquetral								
1 1 2 100	0.15	0.3	79.9969	95.2166	99.3167	100	100	100
1 1 5 200	0.075	0.375	92.982	99.2224	99.9232	100	100	100
1 1 4 300	0.05	0.2	21.8581	31.7753	40.6705	56.3384	80.2746	99.0188
1 1 8 400	0.0375	0.3	28.3484	40.1577	51.0409	70.4194	92.6221	99.9847
ulna								
1 1 2 100	2.6	5.2	94.2457	95.4899	96.7341	98.4448	100	100
1 1 5 200	1.3	6.5	88.0847	93.3804	94.6161	95.3222	97.6169	100
1 1 4 300	0.8667	3.4667	90.8346	95.5083	96.5706	97.4203	98.2701	100
1 1 8 400	0.65	5.2	82.3156	90.8242	93.9273	95.8625	97.2639	99.2993

Coverage Analysis for the Segmentation of Medical Data								
			Coverage Percentage in Pixels					
Bone	**Dist XY**	**Dist Z**	**1 Pix**	**1.5 Pix**	**2 Pix**	**3 Pix**	**5 Pix**	**10 Pix**
410.10.fib	0.46875	3	30.08982	46.70659	57.33533	72.60479	90.86826	100
41.10.fib	0.46875	3	6.505295	9.379728	14.82602	23.75189	50.07564	97.57943
41.19.fib	0.46875	3	24.57003	39.47584	51.51515	65.27437	84.76659	99.9181
41.1.fib	0.46875	3	12.69544	19.26204	23.76485	35.33458	54.40901	86.80425
412.12.fib	0.46875	3	18.53526	25.67812	32.82098	48.2821	69.80109	97.83002
lindsay_radius	0.46875	10	15.28926	24.07025	33.52273	48.34711	58.67769	72.46901
lindsay_ulna	0.46875	10	44.90862	60.31332	71.69713	89.29504	98.1201	100

Timing Results for the Segmentation of Computer Generated Data					
Data	**Setup**	**Rough Align**	**Global Align**	**Local Align**	**Total**
capitate 100 1 1 2	0.2	0	20.3	1.43	21.93
fibula 100 1 1 2	0.09	0	5.56	0.17	5.82
hamate 100 1 1 2	0.18	0	17.93	1.3	19.41
lunate 100 1 1 2	0.2	0.01	17.99	1.14	19.34
mip3 100 1 1 2	0.08	0	14.5	0.51	15.09
mpp3 100 1 1 2	0.1	0	8.24	0.66	9
radius 100 1 1 2	0.12	0	6.38	0.33	6.83
scaphoid 100 1 1 2	0.14	0	32.97	1.15	34.26
talus 100 1 1 2	0.23	0	26.39	1.78	28.4
trapezium 100 1 1 2	0.17	0	18.2	1.22	19.59
trapezoid 100 1 1 2	0.19	0	18.33	1.13	19.65
triquetral 100 1 1 2	0.19	0	28.31	1.06	29.56
ulna 100 1 1 2	0.1	0	4.44	0.24	4.78
capitate 200 1 1 5	0.71	0	38.42	2.22	41.35
fibula 200 1 1 5	0.11	0	8.15	0.27	8.53
hamate 200 1 1 5	0.66	0	31.95	2.05	34.66
lunate 200 1 1 5	0.91	0	34.08	1.99	36.98
mip3 200 1 1 5	0.37	0	13.65	0.82	14.84
mpp3 200 1 1 5	0.29	0	14.6	1.07	15.96
radius 200 1 1 5	0.15	0	11.3	0.51	11.96
scaphoid 200 1 1 5	0.54	0.01	57.95	1.86	60.36
talus 200 1 1 5	0.83	0	48.35	2.89	52.07
trapezium 200 1 1 5	0.73	0	34.1	1.91	36.74
trapezoid 200 1 1 5	0.81	0.01	35.31	1.91	38.04
triquetral 200 1 1 5	0.74	0	27.46	1.78	29.98
ulna 200 1 1 5	0.11	0	7.95	0.36	8.42

Timing Results of the Segmentation of Computer Generated Data (continued)					
Data	**Setup**	**Rough Align**	**Global Align**	**Local Align**	**Total**
capitate 300 1 1 4	2.27	0.01	232.08	9.72	244.08
fibula 300 1 1 4	0.14	0	25.35	0.65	26.14
hamate 300 1 1 4	2.1	0.01	151.95	8.36	162.42
lunate 300 1 1 4	3.05	0.01	211.89	7.93	222.88
mip3 300 1 1 4	1.18	0	132.62	2.99	136.79
mpp3 300 1 1 4	0.85	0	53.69	2.95	57.49
radius 300 1 1 4	0.24	0	34.87	1.31	36.42
scaphoid 300 1 1 4	1.76	0.01	280.68	7.68	290.13
talus 300 1 1 4	2.66	0.01	318.75	13.18	334.6
trapezium 300 1 1 4	2.41	0.01	199.86	8.33	210.61
trapezoid 300 1 1 4	2.72	0.01	224.72	7.96	235.41
triquetral 300 1 1 4	2.49	0.01	27.91	7.48	37.89
ulna 300 1 1 4	0.2	0	25.21	0.94	26.35
capitate 400 1 1 8	4.91	0.01	174.3	8.47	187.69
fibula 400 1 1 8	0.2	0.01	22.73	0.62	23.56
hamate 400 1 1 8	4.67	0.01	123.27	7.29	135.24
lunate 400 1 1 8	6.78	0.01	327.18	7.01	340.98
mip3 400 1 1 8	2.53	0.01	105.85	2.59	110.98
mpp3 400 1 1 8	1.77	0.01	46.05	2.58	50.41
radius 400 1 1 8	0.39	0	15.25	1.18	16.82
scaphoid 400 1 1 8	3.79	0.01	130.98	6.61	141.39
talus 400 1 1 8	5.85	0	253.72	11.17	270.74
trapezium 400 1 1 8	5.42	0.01	307.42	7.48	320.33
trapezoid 400 1 1 8	6.05	0	345.03	7.11	358.19
triquetral 400 1 1 8	5.5	0	26.95	6.39	38.84
ulna 400 1 1 8	0.33	0	11.33	0.87	12.53

Timing Results for the Segmentation of Medical Data					
Data	**Setup**	**Rough Align**	**Global Align**	**Local Align**	**Total**
410.10.fib	0.01	0	5.72	0.04	5.77
41.1.fib	0.04	0	13.19	0.06	13.29
41.10.fib	0.02	0	5.67	0.03	5.72
41.19.fib	0.02	0	9.76	0.04	9.82
412.12.fib	0.03	0	7.52	0.05	7.6
lindsay_radius	0.58	0	5.28	0.44	6.3
lindsay_ulna	0.58	0	4.16	0.36	5.1

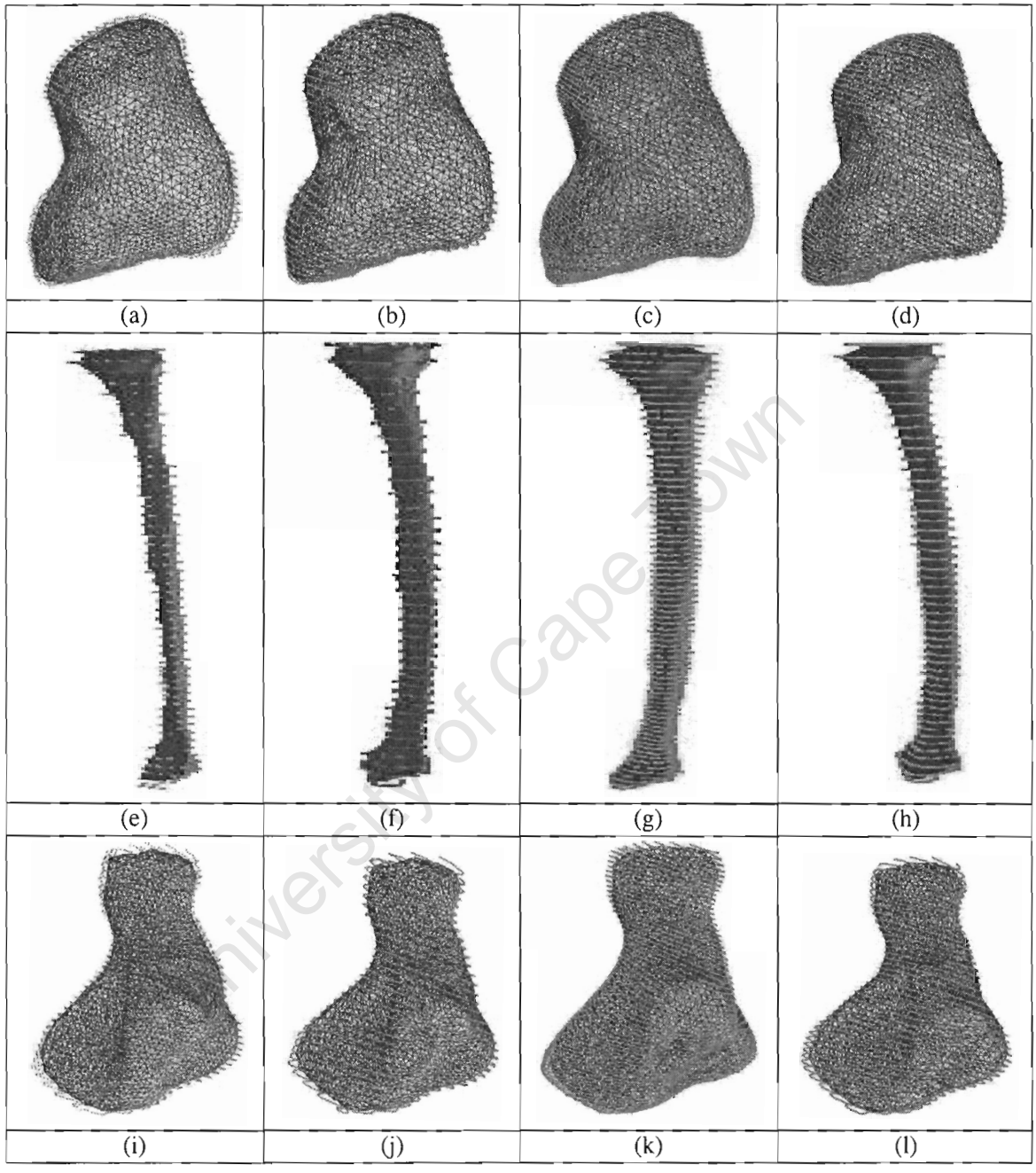


Figure 27: **Segmentation Results for Computer Generated Data (1):** Segmentations at the four resolution for the capitate (a)-(d), fibula (e)-(h) and hamate (i)-(l) bones. The green represents the (input) voxel volume dataset, while the brown is a wireframe mesh representation of the segmentation achieved.

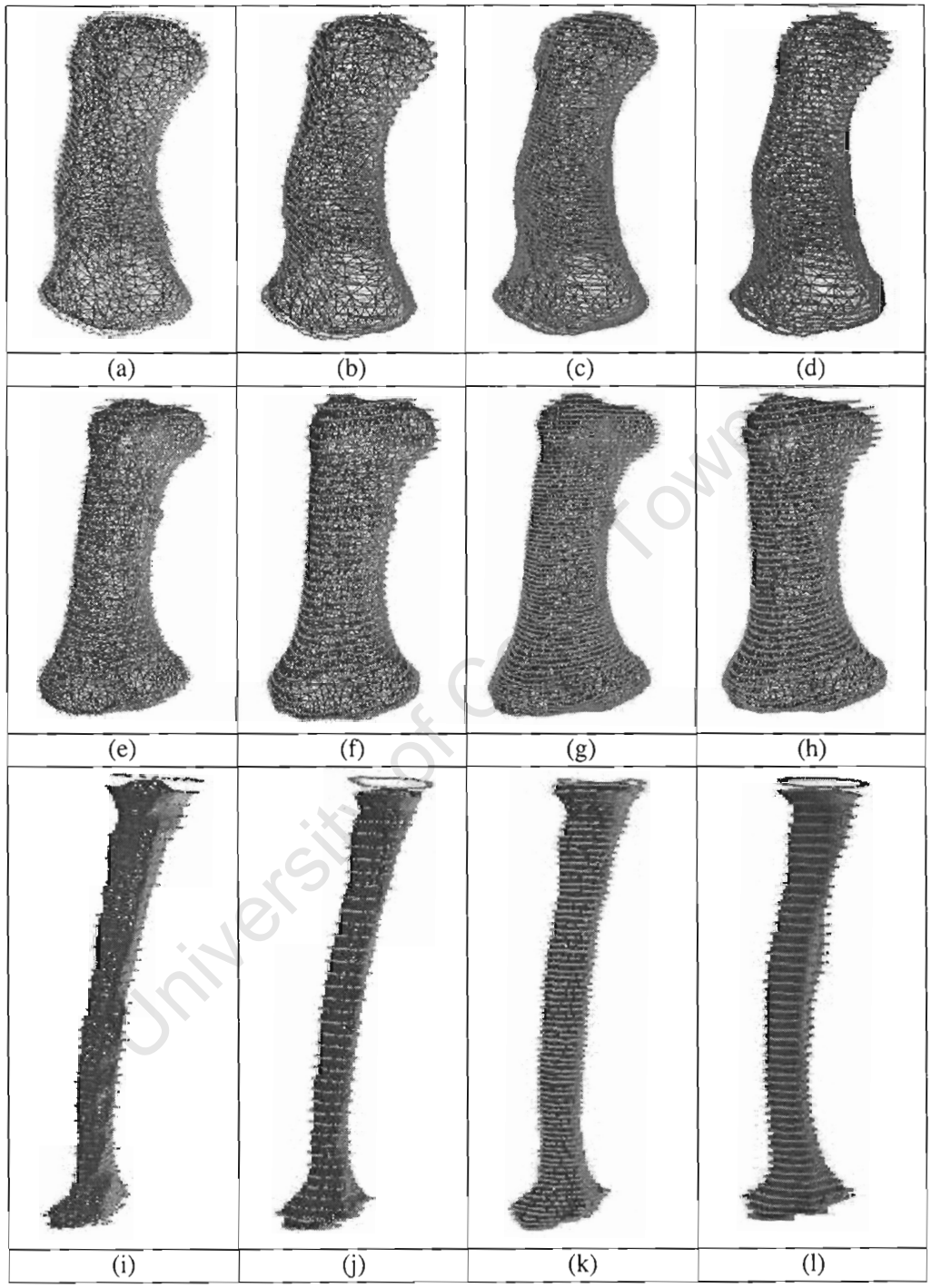


Figure 28: **Segmentation Results for Computer Generated Data (2):** Segmentations at the four resolution for the mip3 (a)-(d), mpp3 (e)-(h) and radius (i)-(l) bones. The green represents the (input) voxel volume dataset, while the brown is a wireframe mesh representation of the segmentation achieved.

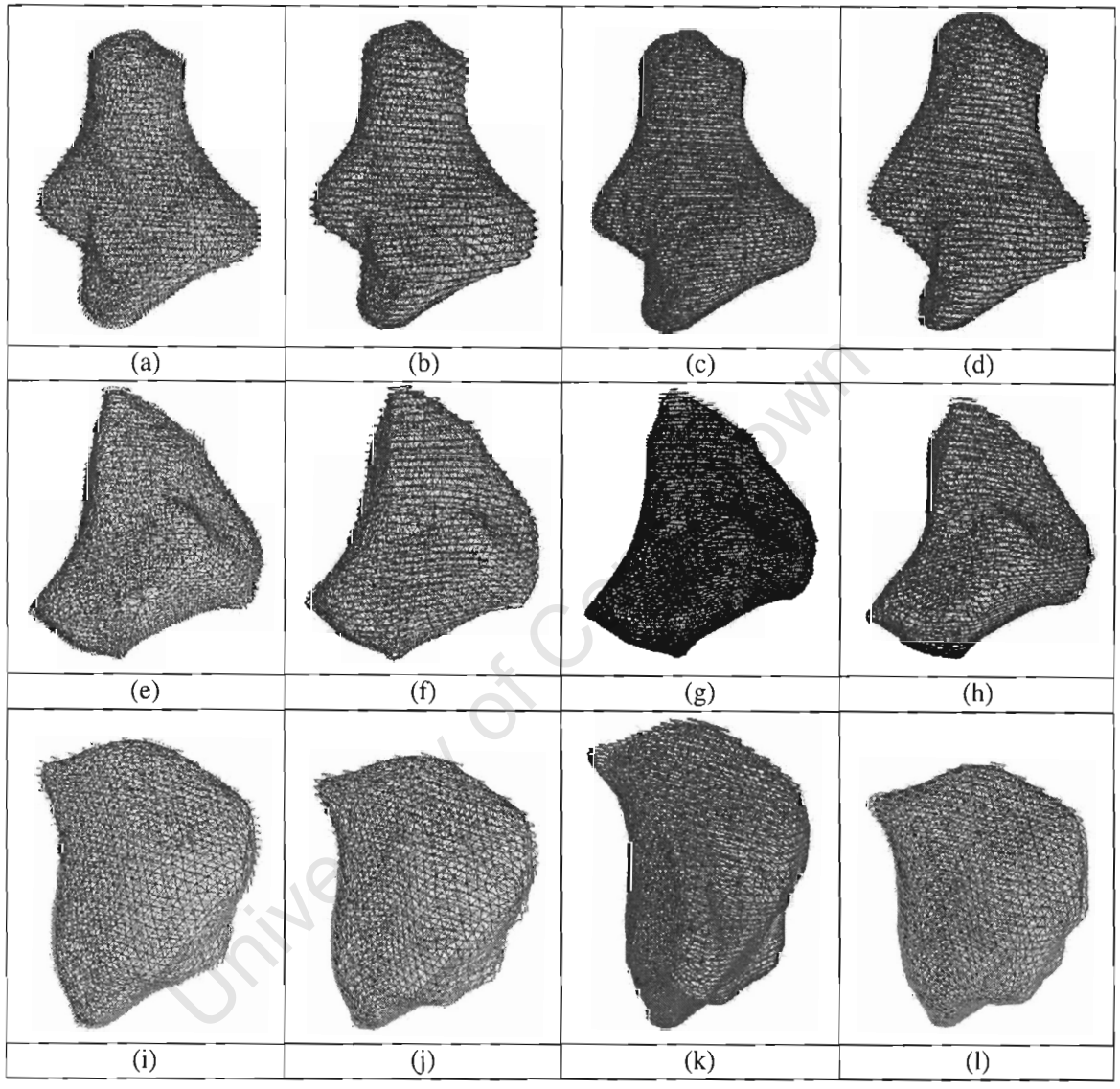


Figure 29: **Segmentation Results for Computer Generated Data (3):** Segmentations at the four resolution for the trapezium (a)-(d), trapezoid (e)-(hand (i)-(l) triquetral bones. The green represents the (input) voxel volume dataset, while the brown is a wireframe mesh representation of the segmentation achieved.

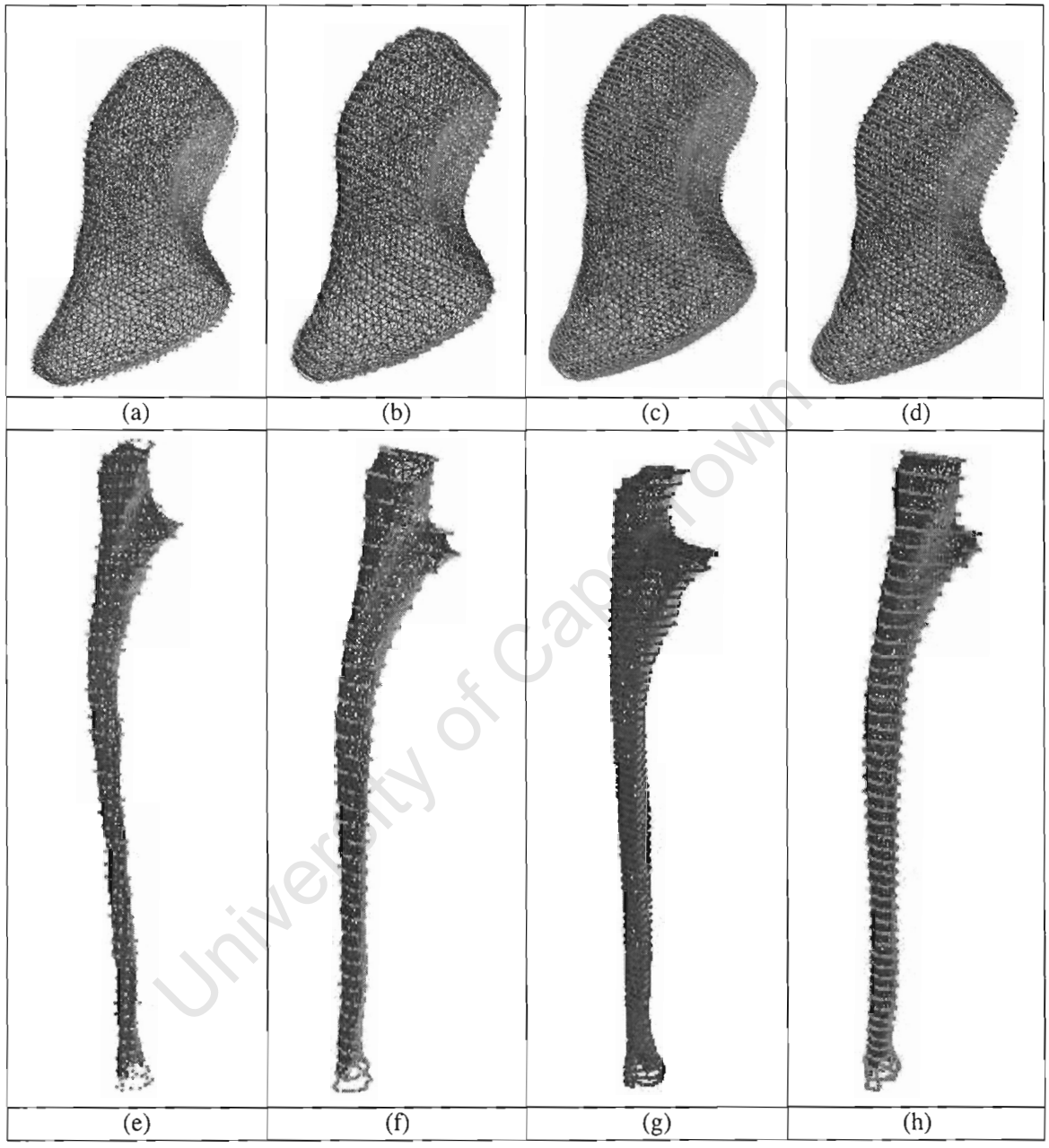


Figure 30: **Segmentation Results for Computer Generated Data (4):** Segmentations at the four resolution for the scaphoid (a)-(d) and the ulna (e)-(h) bones. The green represents the (input) voxel volume dataset, while the brown is a wireframe mesh representation of the segmentation achieved.

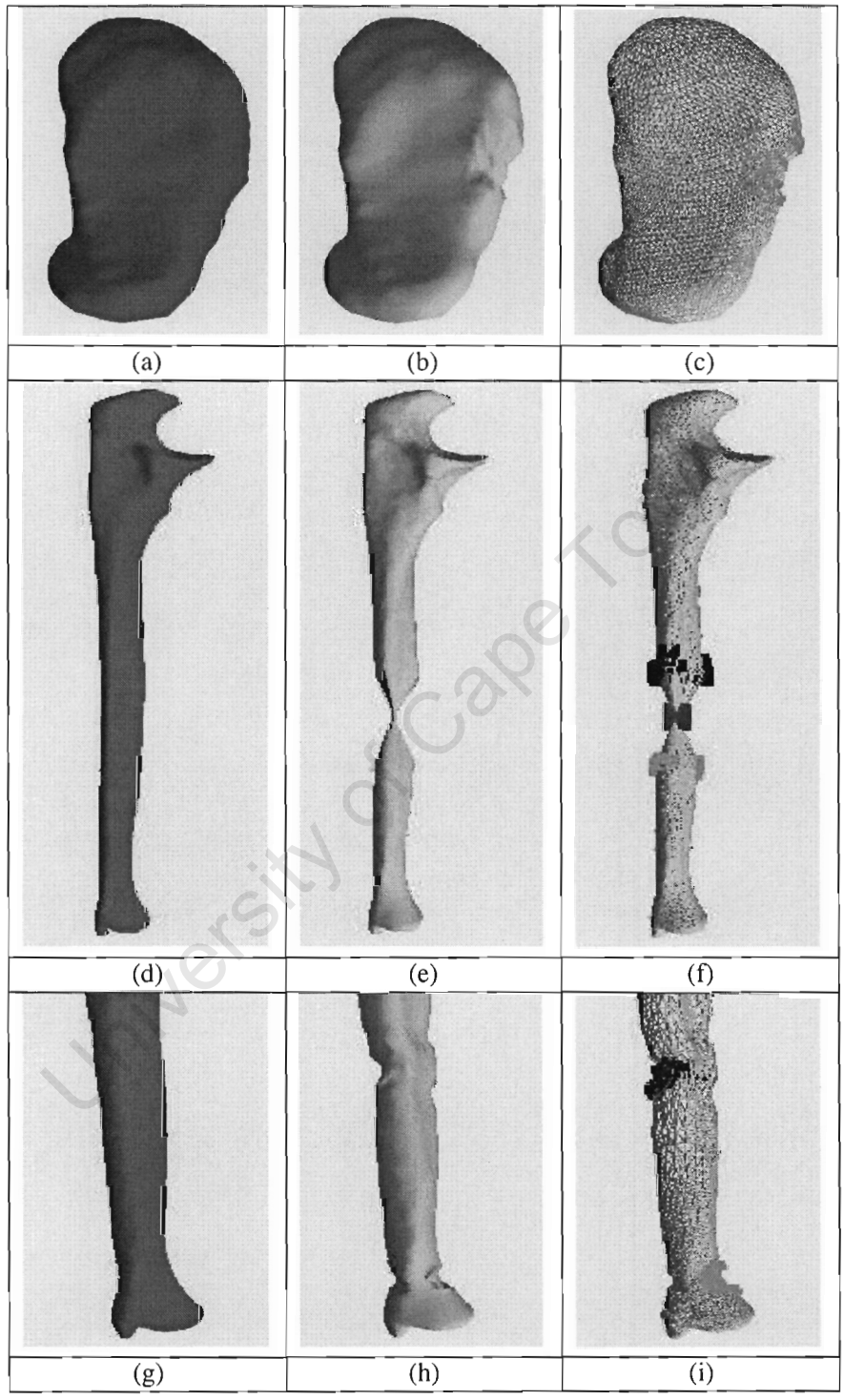


Figure 31: **Bone Fracture Detection Results for Computer Generated Data (1):** (a),(d),(g) the reference bone mesh, (b),(e),(h) the bone fractures in the computer generated data, (c),(f),(i) the bone fractures are highlighted by the bone fracture detection framework.

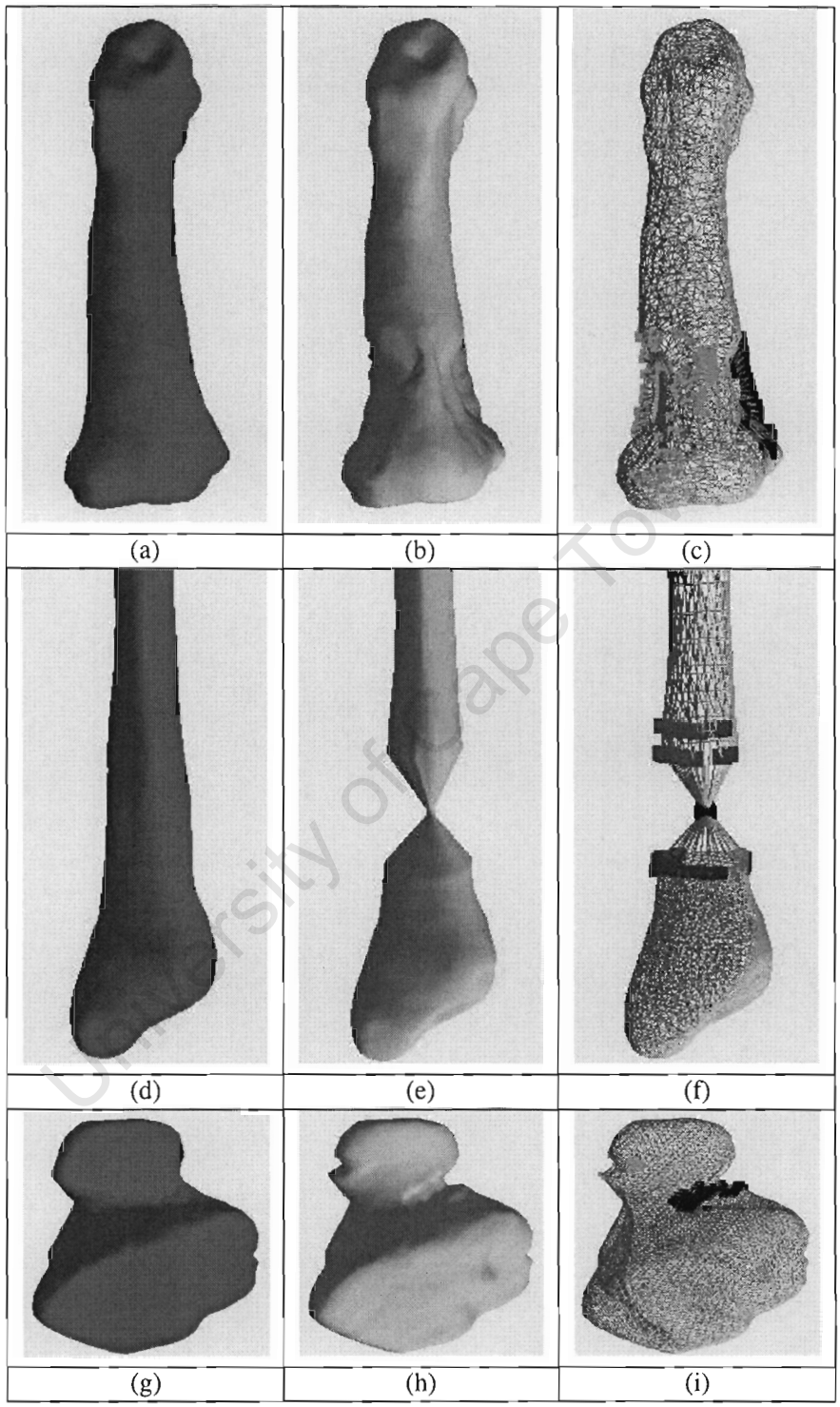


Figure 32: **Bone Fracture Detection Results for Computer Generated Data (2):** (a),(d),(g) the reference bone mesh, (b),(e),(h) the bone fractures in the computer generated data, (c),(f),(i) the bone fractures are highlighted by the bone fracture detection framework.

Bibliography

[1] E. Bardinet, L. Cohen, and N. Ayache. Superquadrics and free-form deformations: A global model to fit and track 3d medical data. In *First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine*, pages 319–326. Springer-Verlag, April 1995.

[2] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE. Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[3] J. Bresenham. Algorithm for computer control of digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[4] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1–31, 1993.

[5] V. Casseles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 134(275-301), 1997.

[6] E. Catmull and R. Rom. A class of local interpolating splines. *Computer Aided Geometric Design*, pages 317–324, 1974.

[7] A. Chakraborty, L. H. Staib, and J. S. Duncan. Deformable boundary finding influenced by region homogeneity. In *Conference on Computer Vision and Pattern Recognition*, pages 624–627. Los Alamitos, CA: IEEE Computer Society Press, June 1994.

[8] V. Chatzis and I. Pitas. Shape-based interpolation of binary 3-d images using morphological skeletonization. In *ICMCS*, volume 2, pages 939–943, 1999.

[9] S. Chen, W. Lin, C. Liang, and C. Chen. Improvement on dynamic elastic interpolation for 3d medical image reconstruction from serial cross sections. *IEEE Transactions on Medical Imaging*, 9(1):71–83, March 1990.

[10] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, 1991.

[11] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.

[12] T. Cootes, C. Beeston, G. Edwards, and C. Taylor. A unified framework for atlas matching using active appearance models. In *Information Processing in Medical Imaging*, pages 323–333, 1999.

[13] T. Cootes, A. Hill, C. Taylor, and J. Haslam. The use of active shape models for locating structure in medical images. *Image and Vision Computing*, 12(6):276–285, July 1994.

[14] T. Cootes and C. Taylor. Active shape models - 'smart snakes'. In *Proc. British Machine Vision Conference*, pages 266–275. Springer-Verlag, 1992.

[15] T. Cootes and C. Taylor. Active shape model search using local grey-level models: A quantitative evaluation. In J. Illingworth, editor, *British Machine Vision Conference*, pages 639–648. BMVA Press, 1993.

[16] T. Cootes and C. Taylor. Modelling object appearance using the grey-level surface. In E. Hancock, editor, *5th British Machine Vision Conference*, pages 479–488. BMVA Press, September 1994.

[17] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, pages 9–18. Springer-Verlag, 1992.

[18] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. New York: Interscience, 1953.

[19] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer Verlag, 2nd edition, 2000.

[20] K. Delibasis, P. Undrill, and G. Cameron. Designing fourier descriptor-based geometric models for object interpretation in medical images using genetic algorithms. *Computer Vision and Image Understanding*, 66:286–300, 1997.

[21] H. Delingette. Simplex meshes: A general representation for 3-d shape reconstruction. Technical report, INRIA, Sophia-Antipolis, France, 1994.

[22] R. Durikovic, K. Kaneda, and H. Yamashita. Dynamic contour: A texture approach and contour operations. *The Visual Computer*, 11:277–289, 1995.

[23] N. Duta and M. Sonka. Segmentation and interpretation of mr brain images using an improved knowledge-based active shape models. In *Information Processing in Medical Imaging*, pages 375–380. Springer-Verlag, 1997.

[24] N. Duta and M. Sonka. Segmentation and interpretation of mr brain images: An improved active shape model. *IEEE Trans. Medical Imaging*, 1998.

[25] N. Duta, M. Sonka, and A. Jain. Learning shape models from examples using automatic shape clustering and procrustes analysis. In *Information Processing in Medical Imaging*, pages 370–375, 1999.

[26] G. Edwards, T. Cootes, and C. Taylor. Advances in active appearance models. In *7th International Conference on Computer Vision*, 1999.

[27] G. Edwards, C. Taylor, and T. Cootes. Interpreting face images using active appearance models. In *International Conference on Automatic Face, Gesture Recognition*, pages 300–305, 1998.

[28] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[29] D. Fritsch, S. Pizer, L. Yu, V. Johnson, and E. Chaney. Segmentation of medical image objects using deformable shape loci. In *Information Processing in Medical Imaging*, pages 127–140, 1997.

[30] J. Gain. *Enhanced Spatial Deformation for Virtual Sculpting*. PhD thesis, Cambridge, 2000.

[31] A. Goshtasby, D. Turner, and L. Ackerman. Matching of tomographic slices for interpolation. *IEEE Transactions on Medical Imaging*, 11(4):507–516, December 1992.

[32] G. Grevera and J. Udupa. Shape-based interpolation of multidimensional grey-level images. *IEEE Transactions on Medical Imaging*, 15(6):881–892, December 1996.

[33] Y. Guo and B. Vemuri. Hybrid geometric active models for shape recovery in medical images. In *Information Processing in Medical Imaging*, pages 112–125, 1999.

[34] O. Henricsson and F. Heitger. The role of key-points in finding contours. In J. Eklundh, editor, *European Conference on Computer Vision*, pages 371–383. Springer-Verlag, 1994.

[35] G. Herman, J. Zheng, and C. Bucholtz. Shape-based interpolation. *IEEE Computer Graphics and Applications*, 12(3):69–79, May 1992.

[36] A. Hill, T. Cootes, and C. Taylor. Active shape models and the shape approximation problem. In *6th British Machine Vision Conference*. BMVA Press, 1995.

[37] A. Hill and C. Taylor. A generic system for image interpretation using flexible templates. In *Proc. British Machine Vision Conference*, pages 276–285. Springer-Verlag, September 1992.

[38] A. Hill and C. Taylor. Model-based image interpretation using genetic algorithms. *Image and Vision Computing*, 10(5):295–300, 1992.

[39] A. Hill and C. Taylor. Automatic landmark generation for point distribution models. In *Proc. of the British Machine Vision Conference*, pages 429–438, 1994.

[40] A. Hill, A. Thornham, and C. Taylor. Model-based interpretation of 3d medical images. In *Proc. British Machine Vision Conference*, pages 339–348, 1993.

[41] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. Technical report, University of Washington, Seattle, 1993.

[42] W. Hsu, J. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. In *SIGGRAPH*, pages 177–184, 1992.

[43] H. Ip and D. Shen. An affine-invariant active contour model (ai snake) for model-based segmentation. *Image, Vision Computing Journal*, 16:135–146, 1998.

[44] I. T. Jolliffe. Principal component analysis. In *Principal Component Analysis*. Springer Verlag, New York, 1986.

[45] M. Jones and T. Poggio. Model-based matching by linear combinations of prototypes. MIT AI memo, 1997.

[46] M. Jones and T. Poggio. Multidimensional morphable models. In *6th International Conference on Computer Vision*, pages 683–688, 1998.

[47] M. Jones and T. Poggio. Multidimensionsal morphable models: A framework for representing and matching object classes. *International Journal of Computer Vision*, 29(2):107–131, August 1998.

[48] A. Kak and M. Slaney. *Principles of Computerised Tomographic Imaging*. IEEE Computer Society Press, 1988.

[49] M. Kass, A. Witkin, and D. Terzopolous. Snakes: Active contour models. In *Proc. First International Conference on Computer Vision*, pages 259–268. IEEE Computer Society Press, 1987.

[50] A. Kaufman. An algorithm for 3d scan-conversion of polygons. *EUROGRAPHICS*, August 1987.

[51] A. Kelemen, G. Szekely, and G. Gerig. Three-dimensional model-based segmentation. In *Proc. of International Conference on Computer Vision*, January 1998.

[52] B. Kimia, A. Tannenbaum, and S. Zucker. Shapes,shocks, and deformations i: the components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1995.

[53] R. Kimmel, A. Amir, and A. Bruckenstein. Finding shortest paths on surfaces using level set propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):635–640, 1995.

[54] A. Kotcheff and C. Taylor. Automatic construction of eigenshape models by genetic algorithm. In *Information Processing and Medical Imaging*, pages 1–14. Springer-Verlag, 1997.

[55] C. Lee, K.-Y. C. I.-N. Chang, and L.-S. Chen. Multi-tuple interpolation using fourier descriptors. In *12th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP '99)*, pages 551–558, August 1999.

[56] W. Lin, C. Liang, and C. Chen. Dynamic elastic interpolation for 3-d medical image reconstruction from serial cross sections. *IEEE Transactions on Medical Imaging*, 7(3):225–232, September 1988.

[57] S. Lobregt and M. Viergever. A discrete dynamic contour model. *IEEE Transactions on Medical Imaging*, 14(1):12–24, 1995.

[58] D. MacDonald, D. Avis, and A. C. Evans. Multiple surface identification and matching in magnetic resonance images. In *SPIE Proceedings on Visualization in Biomedical Computing*, volume 2359, pages 160–169, 1994.

[59] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[60] P. Marais. *The Segmentation of Sparse MR Images*. PhD thesis, Oxford University, 1998.

[61] T. McInerney and D. Terzopolous. Deformable models in medical image analysis:a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[62] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *International Conference on Computer Vision*, pages 840–845. Los Alamitos, CA: IEEE Computer Society Press, June 1995.

[63] D. Metaxas and D. Terzopoulos. Dynamic deformation of solid primitives with constraints. *Computer Graphics*, 21(2):309–312, July 1992.

[64] T. Möller and E. Haines. *Real-Time Rendering*. A. K. Peters Limited, 1999.

[65] J. Montagnat and H. Delingette. A hybrid framework for surface registration and deformable models. In *Computer Vision and Pattern Recognition*, June 1997.

[66] Y. Moses and S. Ullman. Limitations of non model-based recognition. In G. Sandini, editor, *Second European Conference on Computer Vision Proceedings*, pages 820–828. Springer-Verlag, May 1992.

[67] C. Nastar and N. Ayache. Frequency-based nonrigid motion analysis: application to four dimensional medical images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1067–1079, 1996.

[68] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 13:730–742, 1991.

[69] A. Pentland and S. Sclaroff. Closed-form solutions for physically based modelling and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[70] R. Pichumani. *Construction of 3D Geometric Models for Segmentation and Visu alization of Cervical Spine Images*. PhD thesis, Stanford University, November 1997.

[71] C. S. Poon and M. Braun. Image segmentation by a deformable contour model incorporating region analysis. *Phys. Med. Biol.*, 42:1833–1841, 1997.

[72] J. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, 1995.

[73] R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, 1994.

[74] N. Rougon and F. Preteux. Directional adaptive deformable models for segmentation with application to 2d and 3d medical images. *Medical Imaging 93: Image Processing*, pages 193–207, 1993.

[75] S. Rowe and A. Blake. Statistical feature modelling for active contours. In *European Conference on Computer Vision*, pages 561–569, 1996.

[76] G. Sapiro, G. Kimmel, and V. Caselles. Object detection and measurements in medical images via geodesic deformable contours. *Vision Geometry IV*, 2573:366–378, 1995.

[77] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *SIGGRAPH*, pages 151–160, 1986.

[78] J. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry,Fluid Mechanics,Computer Vision and Material Science*. Cambridge University Press, 2nd edition, 1999.

[79] K. Siddiqi, Y. Lauziere, A. Tannenbaum, and S. Zucker. Area and length minimizing flows for shape segmentation. *IEEE Transactions on Medical Imaging*, 7(433-443), 1998.

[80] A. Singh, L. von Kurowski, and M. Chiu. Cardiac mr image segmentation using deformable models. *Biomedical Image Processing and Biomedical Visualization*, 1993.

[81] L. Staib and J. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, 1992.

[82] G. Szekely, A. Kelemen, A. Brechbuhler, and G. Gerig. Segmentation of 2-d and 3-d objects from mri volume data using constrained elastic deformations of flexible fourier surface models. *Medical Image Analysis*, 1(1):19–34, 1996.

[83] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:703–714, 1991.

[84] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[85] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley, Reading, Massachusetts, U.S.A., 1974.

[86] G. Turk. Re-tiling polygonal surfaces. In *Proceedings of SIGGRAPH*, 1992.

[87] B. Vemuri and A. Radisavljevic. Multiresolution stochastic hybrid shape models with fractal priors. *ACM Transaction on Graphics*, 13:177–207, 1994.

[88] Y. Wang and L. Staib. Boundary finding with correspondence using statistical shape models. *Computer Vision and Pattern Recognition*, pages 338–345, June 1998.

[89] C.-F. Westin, A.Bhalerao, H. Knutsonn, and R. Kikinis. Using local 3d structure for segmentation of bone from computer tomography images. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 794–800, June 1997.

[90] C.-F. Westin, S.Warfield, A. Bhalerao, L. Mui, J.Richolt, and R. ikinis. Tensor controlled local structure enhancement of ct images for bone segmen tation. In W. Wells, A. Colchester, and S. Delp, editors, *Proc. of First International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'98)*, pages 1205–1212. Springer Verlag, October 1998.

[91] B. Widrow. The "rubber-mask" technique. *Pattern Recognition*, 5:175–211, 1973.

[92] D. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP:Image Understanding*, 55(1):14–26, 1992.

[93] C. Xu and J. L. Prince. Generalized gradient vector flow external forces for active contours. *Signal Processing – An International Journal*, 71(2):131–139, 1998.

[94] C. Xu and J. L. Prince. Snake, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(2):359–369, 1998.

[95] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tennenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16:199–209, 1997.

[96] Z. Zhang. Iterative point matching for registration of free-form curves. Technical Report 1658, INRIA, March 1992.