UNIVERSIDAD DE JAÉN

ESCUELA POLITÉCTICA SUPERIOR
DE JAÉN
DEPARTAMENTO DE INFORMÁTICA

TESIS DOCTORAL

ADVANCES IN IDENTIFYING OSSEOUS FRACTURED AREAS AND VIRTUALLY REDUCING BONE FRACTURES

PRESENTADA POR:
FÉLIX PAULANO GODINO

DIRIGIDA POR:
DR. D. JUAN JOSÉ JIMÉNEZ DELGADO

JAÉN, 19 DE SEPTIEMBRE DE 2016

ISBN 978-84-9159-020-0

Advances in identifying osseous fractured areas and virtually reducing bone fractures

by Félix Paulano-Godino

A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in the Engineering and Architecture doctoral program

GRAPHICS AND GEOMATICS GROUP

DEPARTMENT OF COMPUTER SCIENCE

University of Jaén

Advisor:
Dr. Juan José Jiménez Delgado

July 2016

Avances en identificación de zonas óseas fracturadas y reducción virtual de fracturas

TESIS DOCTORAL

para obtener el grado de

Doctor por la Universidad de Jaén

Programa de Doctorado en Ingeniería y Arquitectura

defendida por

Félix Paulano Godino

GRUPO DE INFORMÁTICA GRÁFICA Y GEOMÁTICA

TIC-144 del PAIDI

DEPARTAMENTO DE INFORMÁTICA

Universidad de Jaén

Director:
Dr. Juan José Jiménez Delgado

Julio 2016

El **Dr. Juan José Jiménez Delgado**, Profesor Titular de Universidad del Departamento de Informática de la Universidad de Jaén, España,

CERTIFICA:

Que la presente memoria, titulada "**Avances en Identificación de Zonas Óseas Fracturadas y Reducción Virtual de Fracturas**", ha sido realizada bajo su dirección. Y considerando que representa trabajo de Tesis, autoriza su presentación y defensa para optar al grado de Doctor por la Universidad de Jaén con mención de Doctor Internacional.

Dr. Juan José Jiménez Delgado
Dpto. de Informática
Universidad de Jaén

Memoria presentada para optar al grado de:
Doctor por la Universidad de Jaén, con mención de Doctor Internacional

Félix Paulano Godino
Ingeniero en Informática

Jaén, Julio de 2016

This PhD Thesis was funded by the FPI program of the *Ministerio de Economía y Competitividad* through the research project TIN2011-25259. This program also supported a short stay of 91 days as a visiting researcher in the Interactive Graphics and Simulation Group at the University of Innsbruck (Austria). Moreover, this PhD tesis was also sustained by the research project DPI2015-65123-R.

AGRADECIMIENTOS

Este documento que hoy culmino pone fin al trabajo realizado durante un periodo de retos, de esfuerzo, y de aprendizaje continuo que ha tenido lugar durante estos últimos cuatro años, y que no hubiera sido posible sin la ayuda de las personas que a continuación deseo mencionar.

Quiero expresar mi gratitud a mi director de tesis Juan José Jiménez por depositar su confianza en mí y darme la oportunidad de realizar este trabajo. Además, quiero acordarme de Rubén Pulido y Roberto Jiménez, que de forma activa han colaborado a que este trabajo salga adelante. Finalmente, quiero agradecer a Ángel Luis García, Antonio Rueda y Carlos Ogáyar sus consejos y revisiones en este intenso tramo final.

También deseo acordarme de mi familia y de las personas con las que convivo en el día a día. En especial, quiero dar las gracias a Mari Carmen por su apoyo diario y su ayuda constante, y a mis padres y hermanos, por el cariño, la educación y las oportunidades que me han dado. En este momento tampoco quiero olvidarme de mis abuelos y de sus siempre útiles consejos, ni de mis amigos que han conseguido hacerme desconectar en los momentos en los que no tocaba trabajar.

De igual manera quiero transmitir mi agradecimiento a los compañeros con los que he compartido el laboratorio A3-103 durante estos años. Quiero dar las gracias a Antonio Martínez por hacerme sentir uno más desde el primer día. También quiero acordarme de Bernar Domínguez, Jesús Jiménez y de la última incorporación, Alejandro Graciano, que han conseguido hacer amenos tantos días en el laboratorio.

Finalmente, me gustaría agradecer a las instituciones que han colaborado en el desarrollo de esta tesis: al programa FPI del Ministerio de Economía y Competitividad por financiar tanto esta tesis, como una estancia de investigación realizada en la Universidad de Innsbruck; a Matthias Harders y al grupo de investigación (*Interactive Graphics and Simulation Group*) que dirige por la cesión de datos de fracturas óseas.

ABSTRACT

The aim of this work is the development of computer-assisted techniques for helping specialists in the pre-operative planning of bone fracture reduction. The treatment of bone fractures is a complex task. In the case of simple fractures, those in which a bone is divided into two parts by a single fracture line, an X-ray image is usually sufficient to properly plan the surgery. Nevertheless, when the fracture is more complex, other scan techniques like Computed Tomography (CT) are necessary to obtain a 3D visualization of the osseous structures and to identify the number of bone fragments, their size and position, their order of placement and the most suitable fixation devices. Computer-assisted systems may aid in this process by enabling interaction with virtual models of bones and fragments, assisting in the planning of the surgical intervention, detecting lack of bone tissue or analysing different configurations of fixation devices. As a result, intervention time may be reduced and potential misinterpretations circumvented, with the consequent benefits in the treatment and recovery time of the patient. The computer-assisted planning of a bone fracture reduction may be divided into three main stages: (1) identification of bone fragments from medical images, (2) computation of the reduction and subsequent stabilization of the fracture, and (3) evaluation of the obtained results. The identification stage may include the generation of 3D models of bone fragments, with the purpose of obtaining useful models for the two subsequent stages.

The identification of bone fragments from CT scans requires the segmentation of the bone tissue and the labelling of each bone fragment. The review performed in this thesis allows extracting the main issues to be considered when identifying both healthy and fractured bone. The generation of 3D bone models is an artificial step required to generate valid virtual representations in order to be used in a simulation. These 3D models can require optimization techniques with the aim of obtaining adequate representations for the subsequent stages. The main goal of the computer-assisted fracture reduction stage is to position and align bone fragments in order to stabilize the whole bone structure. For that purpose, the inclusion of fixation devices is usually required. Finally, it is necessary to analyse the reduced fracture with the

goal of validating the obtained result. This analysis can be performed in different ways: obtaining the geometric accuracy of the fracture reduction, testing the mechanical stability of the osseous structure, or even evaluating the simulation process by comparing it with the real intervention.

This thesis deals with the identification of bone fragments from CT scans, the generation of 3D models of bone fragments and the computation of the fracture reduction, excluding the use of fixation devices. A novel method to identify bone fragments from CT scans has been developed in this thesis. Because of the resolution of the medical images, some bone fragments may appear joined after the segmentation procedure. The developed method is able to detect erroneously joined fragments and enables their separation with minimal user interaction. The developed procedure has been evaluated in clinical cases and compared with currently proposed approaches to segment fractured bone. Moreover, a new automatic method for calculating the contact zone between fragments has been developed. This method makes the application of algorithms to reduce bone fractures easier by calculating, matching and aligning the calculated contact areas between bone fragments. Furthermore, the method does not require the generation of 3D mesh models since it only uses the connectivity information of the contours. Regarding the analysis of the obtained results, some tests have been run to evaluate the geometric accuracy of the developed algorithms.

Nowadays, the Marching Cubes (MC) is a *de facto* standard for the generation of 3D models of bone fragments from the segmentation results. This work evaluates different alternatives to MC, describing their main benefits and drawbacks. Additionally, a preliminary study for the development of a new mesh generation method based on a spatial decomposition is presented. The use of a spatial decomposition allows addressing the problem with a divide-and-conquer approach. Finally, an application to enable a quality interaction with huge and noisy 3D bone fragment models has been developed. The application allows defining and relating models of bone fragments and computing a detailed collision detection between them, making the understanding of the fracture easier, and enabling the accomplishment of a coarse alignment of the fragments.

RESUMEN

Esta tesis pretende el desarrollo de técnicas asistidas por ordenador para ayudar a los especialistas durante la planificación preoperatoria de la reducción de una fractura ósea. En el caso de fracturas simples, en las cuales el hueso resulta dividido en dos partes por una única línea de fractura, una imagen de rayos X es suficiente en la mayoría de los casos para planificar la cirugía de manera correcta. No obstante, si la fractura es más compleja, se requieren otras técnicas de escaneado que permitan obtener una visualización 3D de las estructuras óseas, e identificar el número de fragmentos de hueso, su tamaño y posición, el orden en el que deben ser reducidos, así como los elementos de fijación más apropiados. Los sistemas asistidos por ordenador pueden ayudar en este proceso permitiendo interactuar con modelos virtuales de huesos y fragmentos, asistiendo en la planificación de la intervención quirúrgica, detectando la falta de tejido óseo, o analizando diferentes configuraciones de dispositivos de fijación. Como resultado, puede reducirse el tiempo de intervención y pueden evitarse errores de interpretación, con los consecuentes beneficios en el tratamiento y en el tiempo de recuperación del paciente. La planificación asistida por ordenador de una reducción de fractura ósea puede dividirse principalmente en tres grandes etapas: (1) identificación de fragmentos óseos a partir de imágenes médicas, (2) cálculo de la reducción y posterior estabilización de la fractura, y (3) evaluación de los resultados obtenidos. La etapa de identificación puede incluir también la generación de modelos 3D de los fragmentos óseos con el propósito de obtener modelos útiles para las dos etapas posteriores.

La identificación de fragmentos óseos a partir de imágenes médicas, generadas mediante Tomografía Computarizada (TC), requiere la segmentación del tejido óseo y el etiquetado de cada fragmento de hueso. La revisión llevada a cabo en esta tesis permite extraer los principales aspectos que deben ser considerados durante la identificación de hueso, tanto sano como fracturado. La generación de modelos óseos 3D es una etapa artificial requerida para obtener representaciones virtuales que resulten de utilidad en una simulación. Estos modelos 3D pueden requerir técnicas de optimización con el fin de obtener representaciones adecuadas para las siguientes etapas del proceso. El

objetivo principal de la reducción de fracturas óseas asistida por ordenador es posicionar y alinear fragmentos de hueso, con el fin de estabilizar la totalidad de la estructura ósea. Con ese propósito, en la mayoría de los casos se requiere la utilización de elementos de fijación. Finalmente, es necesario analizar la estructura ósea resultante con el objetivo de validar el resultado obtenido. Este análisis puede llevarse a cabo usando diferentes enfoques: calculando la precisión geométrica de la reducción, comprobando la estabilidad mecánica de la estructura ósea, o incluso evaluando el proceso simulado mediante su comparación con la intervención real.

Esta tesis aborda la identificación de fragmentos óseos a partir de imágenes médicas generadas por TC, la generación de modelos 3D de fragmentos óseos, y el cálculo de la reducción de fracturas, sin incluir el uso de elementos de fijación. Se ha desarrollado un método novedoso para identificar fragmentos óseos a partir de imágenes médicas generadas mediante TC. Debido a la resolución utilizada por el escáner, tras el proceso de segmentación pueden aparecer juntos distintos fragmentos óseos. El método desarrollado es capaz de detectar fragmentos conectados por error, y posibilita su separación con una intervención mínima por parte del especialista. Este procedimiento ha sido evaluado en casos clínicos y comparado con los enfoques actualmente utilizados para segmentar hueso fracturado. Por otra parte, se ha diseñado un nuevo método para calcular de manera automática las zonas de contacto entre fragmentos óseos. Este método facilita la aplicación de algoritmos para reducir fracturas óseas mediante el cálculo, la asociación y el alineamiento de las zonas de contacto calculadas entre fragmentos. Además, el método no requiere la generación de mallas 3D ya que tan sólo hace uso de la información de conectividad de los contornos. Con respecto al análisis de los resultados obtenidos, diversos tests han sido llevados a cabo para evaluar la precisión geométrica de los resultados obtenidos por los algoritmos desarrollados.

En la actualidad, el algoritmo Marching Cubes (MC) es un estándar de facto para la generación modelos 3D de fragmentos óseos a partir de los resultados de la segmentación. Esta tesis evalúa diferentes alternativas a MC, extrayendo sus principales beneficios y deficiencias. De manera adicional, se presenta un estudio preliminar para el desarrollo de un nuevo método de generación de mallas de triángulos basado en una descomposición del espacio. El uso de dicha estructura espacial permite abordar el problema con un enfoque basado en el paradigma divide y vencerás. Finalmente, se ha desarrollado una aplicación que permite llevar a cabo una interacción de calidad con grandes modelos de fragmentos de hueso que pueden contener ruido. La aplicación permite definir y relacionar modelos de fragmentos óseos y detectar colisiones entre ellos de forma detallada, facilitando la comprensión de la fractura y permitiendo llevar a cabo un alineamiento aproximado de los fragmentos.

Contributions related to this work

JCR-indexed journals

Jiménez-Delgado J.J., Paulano-Godino F., Pulido-Ramírez R. and Jiménez-Pérez J.R. 2016. Computer assisted preoperative planning of bone fracture reduction: simulation techniques and new trends. *Medical Image Analysis 30 pp. 30-45.*

Paulano-Godino F., Jiménez-Delgado J.J. and Pulido-Ramírez R. 2014. 3D segmentation and labeling of fractured bone from CT images. *The Visual Computer 30(6-8) pp. 939-948.*

Peer-reviewed journals

Paulano-Godino F., Jiménez-Delgado J.J. and Pulido-Ramírez R. 2014. Identification of fractured bone tissue from CT images. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization 4 (3-4) pp. 174-182.*

Book chapters

Paulano-Godino F., Jiménez-Delgado J.J. and Jiménez-Pérez J.R. 2015. Surface reconstruction of bone fragments: A comparative study. In *Computational Vision and Medical Image Processing V: VIPIMAGE 2015, CRC Press pp. 321-326.*

Paulano-Godino F., Jiménez-Delgado J.J. and Pulido-Ramírez R. 2015. Fractured bone identification from CT images, fragment separation and fracture zone detection. In *Developments in Medical Image Processing and Computational Vision, Springer International Publishing pp. 221-239.*

Paulano-Godino F., Jiménez-Delgado J.J. and Pulido-Ramírez R. 2013. Trends on identification of fractured bone tissue from CT images. In *Computational Vision and Medical Image Processing IV: VIPIMAGE 2013, CRC Press pp. 263-269.*

Peer-reviewed Conference Publications

Paulano-Godino F., Jiménez-Delgado J.J. and Pulido-Ramírez R. 2014. An application to interact with 3D models reconstructed from medical images. In *Proceendings of 9th International Conference on Computer Vision Theory and Applications (VISAPP), pp. 224-229.*

Paulano-Godino F., Jiménez-Delgado J.J., Pulido-Ramírez R. and Ogayar-Anguita C.J. 2012. A comparative study of implemented collision detection strategies. In *Proceendings of 7th International Conference on Computer Graphics Theory and Applications (GRAPP), pp. 485-490.*

Contents

Contenidos

List of Figures

List of Tables

List of Acronyms

AABB	Axis-Aligned Bounding Box
APSS	Algebraic Point Set Surface
BVTT	Bounding Volume Traversal Tree
CAD	Computer-Aided Design
CNC	Computer Numerical Control
CT	Computed Tomography
DICOM	Digital Imaging and Communication in Medicine
DLHs	Distal Locking Holes
FEA	Finite Element Analysis
GJK	Gilbert-Johnson-Keerthi
HU	Hounsfield Units
ICC	Intra-class Correlation Coefficient
ICP	Iterative Closest Point
IMN	Intra-Medullary Nail
ITK	Insight Segmentation and Registration Toolkit
LoD	Level of Detail
MC	Marching Cubes
MCMW	Maximum Cardinality Minimum Weight

MLS	Moving Least Squares
MRI	Magnetic Resonance Imaging
MS	Marching Squares
MSE	Mean Squared Error
MWGM	Maximum Weight Graph Matching
NURBS	Non-Uniform Rational B-Splines
OBB	Oriented Bounding Box
PCA	Principal Component Analysis
PCL	Point Cloud Library
PQP	Proximity Query Package
PWT	Probabilistic Watershed Transform
RG	Region Growing
RMSE	Root Mean Square Error
SWIFT++	Speedy Walking via Improved Feature Testing
VTK	Visualization Toolkit

CHAPTER 1

INTRODUCTION

In recent years, the use of computer techniques to assist surgical procedures has considerably increased. These techniques provide tools to train novel surgeons, improve information available during surgery, or even allow performing the pre-operative planning of a medical intervention. One of those surgical procedures is the reduction of a bone fracture, which can be defined as a medical condition where the continuity of the bone is broken.

The treatment of a bone fracture is a complex task. In the case of simple fractures, those in which a bone is divided into two parts by a single fracture line, an X-ray image usually provides enough information to properly plan the surgery. Nonetheless, the planning of the reduction of complex fractures commonly requires other scan techniques with the aim of obtaining 3D models of the osseous structures that enable the identification of the number of bone fragments, their position, their order of placement, and the most suitable fixation devices. The use of computer-assisted systems can aid in this process by identifying bone fragments, by enabling interaction with virtual models of bone fragments, computing the reduction of the fracture, or analysing different configurations of fixation devices. The use of computer-assisted techniques in the pre-operative planning allows specialists to save the time of surgical procedures and reduce potential misinterpretations, with the consequent benefits in terms of decreasing the recovery time [95, 93].

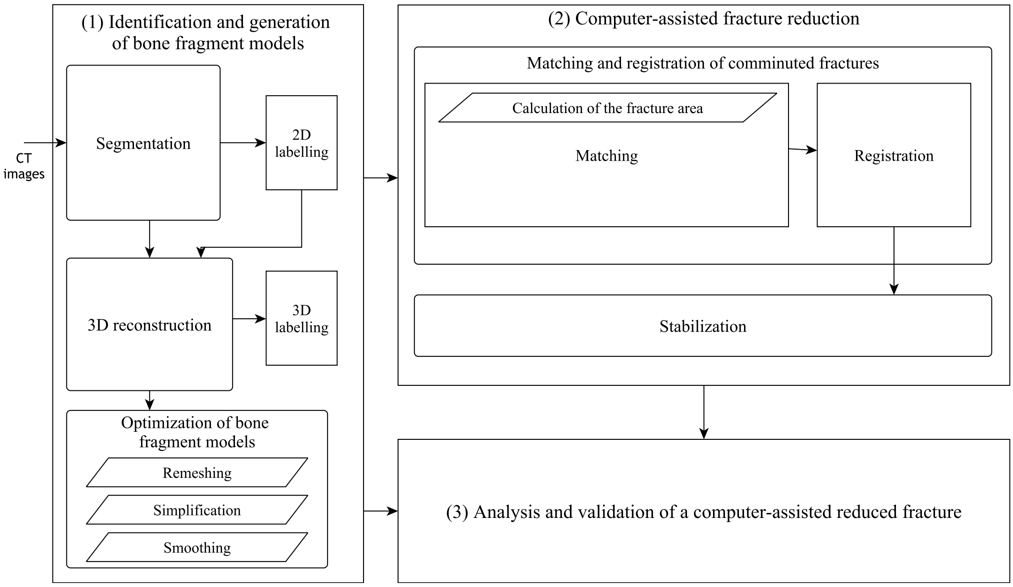
(1) Identification and generation of bone fragment models
CT images
Segmentation
2D labelling
3D reconstruction
3D labelling
Optimization of bone fragment models
Remeshing
Simplification
Smoothing
(2) Computer-assisted fracture reduction
Matching and registration of comminuted fractures
Calculation of the fracture area
Matching
Registration
Stabilization
(3) Analysis and validation of a computer-assisted reduced fracture

Figure 1.1: Detailed view of the stages of the computer-assisted pre-operative planning of bone fracture reduction.

1.1 Computer-assisted pre-operative planning of bone fracture reduction

Computer-assisted bone fracture reduction can be divided into three main stages: (1) identification of bone fragments from medical images and generation of a virtual representation of them, (2) calculation of the reduction and stabilization of the fracture, (3) and analysis of the obtained results from a geometrical and a mechanical point of view (Figure 1.1).

The generation of a virtual representation of the bones and fragments is an artificial step required in order to obtain helpful models to work with in a simulation. Different data representations may be necessary depending on the goals. These usually are: a volume for visualization, a point cloud for fast interaction, and a triangle mesh for geometric operations between models. This stage begins with the extraction of osseous tissue from medical images using segmentation and labelling techniques. Then, a 3D reconstruction is usually performed to produce 3D models of the osseous structures. These models can require the application of optimization techniques in order to get adequate representations for the subsequent stages.

The main purpose of computer-assisted fracture reduction is to position and align the bones and fragments in order to stabilize the whole bone structure. This procedure requires solving a variety of problems that depend on both the type of bone and fracture. In the case of a simple fracture, the reduction consists of aligning the two bone fragments in order to recover their original position. If the fracture generates more than two fragments, a previous procedure is required to solve the puzzle. Moreover, in the majority of cases it is necessary to stabilize the fracture by using fixation devices such as plates and screws.

The analysis of the reduced fracture enables the validation of the obtained result. This analysis can be carried out in different ways: by computing the geometric accuracy of the reduced fracture; by using Finite Element Analysis (FEA) or similar techniques for testing the mechanical stability of the fracture; or even by comparing the result of the simulation process with the actual surgery.

This work is mainly focused on the segmentation and labelling of bone fragments from CT scans, the generation of 3D models of bone fragments and the computation of the fracture reduction, excluding the use of fixation devices. Moreover, a geometrical analysis of the obtained results has also been carried out.

1.2 Aims

Considering the stages described in the previous section, the main aim of this thesis is to research and develop methods that help specialists to identify fractured bone tissue and to reduce bone fractures. Since interactive bone fracture reduction is a time-demanding procedure, the developed methods should aim at reducing user interaction in all stages of the process. Finally, evaluation methods are proposed in order to test the quality of the obtained results. The objectives of this work are:

- To research and develop methods that allow the segmentation and labelling of bone fragments from CT scans.

- To research and develop algorithms that make the automation of the fracture reduction process easier.
- To research and develop novel methods to produce geometric models representing bone fragments.
- To analyse and evaluate the results obtained by the proposed methods.
- To research and use spatial decompositions, collision detection and interaction techniques that might help specialists in the planning of a fracture reduction.

1.3 Organization of this thesis

This dissertation has been organized considering the computer-assisted fracture reduction process. First, a review of the current proposed approaches is performed. Then each chapter from 3 to 5 is focused on a different stage of the process. Finally, conclusions and future research are outlined in Chapter 6. Appendix A presents interactive tools developed with the purpose of facilitating the understanding of the fracture and enabling a coarse alignment of bone fragments. The following paragraphs depict the structure of the document.

Chapter 2 reviews the main proposed works in the literature related to pre-operative planning of computer-assisted bone fracture reduction. Firstly, currently proposed approaches to identify fractured bone are analysed and classified. The most recently proposed methods to segment healthy bone are also reviewed, in order to test whether techniques commonly used for this type of bone are also suitable for fractured bone. Methods to generate 3D models of bone fragments and to optimize them are also depicted. Then, the most relevant computer-assisted techniques to reduce bone fractures are also reviewed and classified. Finally, currently used methods to analyse and evaluate the computer-assisted pre-operative planning of a bone fracture reduction are reviewed.

Chapter 3 describes the main issues to be considered in order to identify bone tissue, as well as additional problems that arise if the bone is fractured. The identification of fractured bone includes not only the segmentation of bone tissue, but also the labelling of bone fragments and the detection of fracture regions. Labelling involves the identification of bone fragments separately. Moreover, this chapter presents a new method to segment and label bone fragments from CT images. The method is based on 2D Region Growing (RG) and requires minimal user interaction. The proposed procedure is able to separate wrongly joined fragments during the segmentation process. The method has been compared with currently used approaches in the literature to identify healthy and fractured bone, obtaining better results in most aspects. Moreover, the method has also been tested with different clinical cases, and it performs well in a large variety of bones and fracture types.

Chapter 4 introduces an automatic method to calculate the contact zone between two bone fragments. The method does not require previously generating 3D meshes of the fragments because it only needs point clouds as input. The developed method makes the application of puzzle-solving methods easier since it does not obtain the entire fracture zone but the contact area between each pair of fragments. Therefore it is not necessary to find correspondences between fracture zones and fragments may be aligned two by two. A matching algorithm is also presented, aiming at automatically reducing bone fractures using the calculated contact zones. The proposed method has been successfully applied to the reduction of different bone fractures. Firstly, the method has been applied to fractures in the ankle area. With the goal of quantitatively measuring the performance of the proposed method, the contact area and the overlapping between fragments have been calculated. Secondly, the method has also been applied to cadaver datasets containing fractures of humerus; hence the results have also been compared to the ground truth.

Chapter 5 presents a study that tests the performance and suitability of well-known mesh generation algorithms to generate triangle meshes representing bone fragments from medical images. The experiments performed allow us to find the benefits and drawbacks of each method when applied to different bone and fracture types. Moreover, the quality of the obtained results is analysed focusing on the special features of the data obtained from CT stacks representing bone fractures. Taking into account the results of the study, the conclusion is that Poisson reconstruction generates the best models for visualization purposes and the models generated by MC are the best suited to be used in computer-assisted bone fracture reduction. On the other hand, the initial results of a preliminary study for the development of a novel method to generate triangle meshes from medical images using a spatial decomposition are presented. In addition, a new approach to extract external contour points and to estimate normal vectors is used in a pre-processing step. In this preliminary study a spatial decomposition named tetra-tree has been used. However, any other spatial decomposition could have been used for this study. The novel method addresses the mesh generation problem using a divide-and-conquer approach in order to minimize problems arising when sewing consecutive contours. The main aim of this initial study is to determine if a spatial decomposition may help in the reconstruction of the mesh, allowing the analysis of the advantages and disadvantages of using it.

Appendix A presents an application that contains tools for performing a detailed interaction with 3D models of bone fragments obtained from CT stacks. This application aims at being the basis for integrating the algorithms developed in this work as well as new tools developed in the future. In order to generate 3D models from CT stacks, MC is used since experiments carried out in Chapter 5 demonstrated that it is the best option for computer-assisted procedures. Despite the fact that CT stacks are usually segmented before generating the 3D models, the developed application contains tools to clean noise and to manually segment and label models. The application calculates overlapping triangles, distances and nearest points between bone fragment models in real-time. For that purpose, a comparative study has been carried out in order to determine the collision detection strategy that best deals with the features of models obtained from the information available in CT stacks. Finally, the *Leonar3Do* system [58] has been incorporated in order to improve interaction and to provide stereo visualization. The developed application can be used in the pre-operative planning of a fracture reduction, since it makes easier the understanding of the fracture and allow carrying out a coarse alignment of the fragments.

CHAPTER 2

Recent advances in computer-assisted bone fracture reduction

This chapter presents a review of the available computer-assisted techniques to help specialists in the pre-operative planning of a fracture reduction intervention, including the analysis of its results. The chapter is structured according to the stages of the computer-assisted pre-operative planning of a bone fracture reduction (Figure 1.1). The process can be divided into three main stages: identification of bone fragments and generation of 3D geometric models representing them (Section 2.1), computation of the reduction and stabilization of the bone fracture (Section 2.2), and analysis and validation of the obtained results (Section 2.3). This review enables the identification of the main techniques proposed in order to solve each stage of the planning and shows the pros and cons of current approaches.

2.1 Identification and generation of bone fragment models

The generation of bone fragment models begins with the segmentation of bone tissue from CT scans. Bone fragments involved in the fracture have to be labelled during or after the segmentation process. Additionally, the fracture area is sometimes calculated with the aim of better knowing the fracture features. Finally, a 3D model of each bone fragment is generated from the segmentation results.

2.1.1 Identification of bone fragments from medical images

The identification of bone fragments from medical images requires segmenting and labelling the bone regions involved in the fracture. CT is the most appropriate and accessible tool for distinguishing bone from other tissues and thus, for performing the segmentation of bone fragments [21]. Most of the times, CT images are saved in Digital Imaging and Communication in Medicine (DICOM) format. A DICOM image stores at each pixel the radio-density obtained during the CT scan . These quantities are measured in Hounsfield Units (HU), which are calculated from a linear transformation of the original linear attenuation coefficient in which the radiodensity of distilled water is 0HU, and the radiodensity of air at standard temperature and pressure is -1000HU. In order to display these values in an image, intensity values are used. From this point onwards, we will refer to the values at the pixels of a CT image as intensity values.

2.1.1.1 Healthy bone

In the literature, various studies pretend to segment healthy bone from CT images. Table 2.2 summarizes the most recent ones.

Despite being simple and easy to implement, threshold-based approaches are able to segment bone tissue in some situations. In the literature, there are various works that use thresholding to segment bone tissue from CT. Neubauer et al. [73] propose to use a user-defined threshold in order to segment ulna, radius and carpus from CT images. Zhang et al.[110] utilize an adaptive threshold method for segmenting calcaneus and vertebrae.

Other authors propose to use region-based methods in order to segment bone tissue. In [70], 3D RG is used to segment the inferior maxillary bone from CT images . After the segmentation, holes are filled by using a morphological operation of closing. Then, 3D ray casting is applied to calculate the internal region of the bone by determining which points are inside of the outer shell. The segmented voxels are classified as cortical or trabecular bone using a fuzzy c-means algorithm. With the aim of improving the result, an adapted median filter allows removing outliers. Zhao et al. [111] also use a 3D RG method to segment bone tissue. Both the seeds and the threshold are calculated automatically. Since they use a unique threshold, some areas of bone are not segmented and they propose a method to fill them. This segmentation approach has been tested in the segmentation of skull and spine bones.

Edge-based methods pretend to calculate the contour, instead of the region of each osseous structure. Mastmeyer et al. [65] use a 3D deformable balloon model to segment the vertebral bodies semi-automatically. A novel active contour model has also been used to segment bone tissue in the knee region [99]. In [60], Willmore flow is integrated into the level set method in order to segment the spinal vertebrae.

Statistical methods are frequently used to segment bone tissue. Battiato et al. [5] use a generative model to classify pixels into cortical bone or another tissue. A learned model is constructed by modelling probability functions using Gaussian mixture models. Then the learned model allows assigning a probability to each pixel, and a maximum a-posteriori probability rule enables a crisp classification. An expectation maximization algorithm has been utilized to segment phalanx bones [81]. The method requires a previously generated CT atlas. The statistical texture method has also been proposed to segment mandible bones from CT images [72]. Janc et al. [44] use a genetic algorithm to search the better procedure to segment bone tissue and to separate cortical and trabecular tissue. For that aim, the genetic algorithm requires previous expert information.

Due to the heterogeneity of bone tissue in a CT scan, sometimes it is necessary to use a mix of processes. Sebastian et al. [87] combine RG, active contours and region competition to segment carpal bones. Statistical classifiers and graph cuts have been employed to segment the hip bone [64]. Graph cuts have also been used to segment vertebrae [3]; in this method, seeds are automatically placed using a matched filter and vertebrae are identified with a statistical method based on an adaptive threshold. Cortical and trabecular bone are then separated by using a local adaptive RG method.

2.1.1.2 Fractured bone

The methods applied to the segmentation of healthy bone could not be suitable for segmenting fractured bone . This is because fractured bone has different features: fragments are not completely surrounded by cortical tissue, they can have arbitrary shape and may belong to any bone in a nearby area, and different fragments can appear joined as one in the image. Therefore, the identification of fracture bone requires carrying out additional steps: labelling the fragments and splitting up wrongly joined fragments. All the currently proposed methods to identify fractured bone from CT images are classified in Table 2.3.

Region-based methods are used in most cases in order to identify bone tissue from CT images. Some methods achieve accuracy at the expense of requiring a lot of user intervention [98]. The area where bone tissue is located is detected using a threshold-based method . Then these approaches usually propose manual and semi-automatic tools to interactively segment bone fragments. These proposals include separation, merging and hole-filling tools for generating individually segmented fragments from the previous result . A global fixed threshold method has been utilized by [96] to detect the trabecular bone fracture zone. However, due to the variation of the intensity levels between slices, it is difficult to find a unique threshold that enables the segmentation of bone tissue in all the slices.

Other authors propose pre-processing the image in order to ease the segmentation of bone tissue [38, 29]. In these studies, a sheetness measure is used to enhance the cortical tissue in CT images. Then the resulting image is segmented using 3D RG . Fornaro et al. [29] apply a 3D connected component labelling algorithm to separate erroneously joined bone fragments in simple cases . To deal with the separation of erroneously joined fragments in complex fractures or in which the boundary of the bone is weak, an interactive graph cuts based segmentation is used . Lee et al. [57] propose a multi-region segmentation approach. Seeds are automatically placed by searching for pixels that have an intensity value higher than a threshold in the image. Then a RG algorithm propagates all the regions in turn. If this multi-region segmentation algorithm fails, the authors provide a manual region combination method that allows blending of the wrongly-segmented regions, and a region re-segmentation method that enables the splitting of wrongly-joined bone fragments . A modification of the classical watershed is proposed by Shadid and Willis [90]. They use the Probabilistic Watershed Transform (PWT) to segment bone fragments from CT images, encoding semantic information about the objects.

Other classical segmentation methods, based on edges or registration, are barely represented. Registration-based methods require templates of the fractured parts, thus they are limited to the specific fractures defined by the templates [78]. In addition, it is difficult to find two identical fractures, especially in the case of comminuted fractures. For this reason, registration-based methods are limited to simple fractures. On the other hand, edge-based methods try to obtain a closed contour associated with each bone fragment [105]. Nonetheless, bone fragments are partially surrounded by trabecular tissue due to the fracture. This type of tissue is very heterogeneous; hence it is complicated to obtain a closed contour that properly represents the bone fragments.

2.1.1.3 Fracture zone calculation

The fracture zone can be defined as the area in which the fracture occurs. The identification of this area is useful for the pre-operative planning of a bone fracture surgery. For instance, the simulation of a fracture reduction and the virtual analysis of the fracture may require a previous calculation of this area. Therefore, some approaches have been proposed aiming at computing the fracture area after the segmentation of bone fragments . Table 2.4 summarizes all the analysed works to calculate fracture zones.

Statistical based approaches have been proposed to identify fractured zones. Willis and Thomas [105] semi-automatically reconstruct highly fragmented bone fractures. Before performing the fracture reduction, they need to separate intact and fractured zones of each bone fragment. For that purpose, they propose to use a mixed model consisting of two Gaussian probability distributions to perform a binary classification. They choose a threshold that enables the classification of intact-surface intensities and minimize the type I classification errors. Thus, this threshold allows separating fractured and intact surfaces. After classifying all points, the fractured surface is the largest continuous region of fractured surface points. Zhou et al. [114] present an extension of the previous method that improves fragment alignment in highly fragmented bone fractures. In order to separate fractured and intact surfaces, they use a two-class Bayesian classifier based on the intensity values previously mapped on the surface vertices.

Other proposals take advantage of the specific shape of a particular type of bone . Winkelbach et al. [107] present an approach to semi-automatically perform the reduction of cylindrical bones. In order to identify vertices of the fractured area, they check the normal orientation of each vertex and compare them with the bone axis. This method does not work when fracture lines are almost parallel to the bone axis.

Curvature analysis has also been used to identify fractured surfaces. Okada et al. [75] present a procedure to virtually reduce proximal femoral fractures. Aiming at obtaining fracture lines in each slice, they use curvature analysis. For that purpose, a 3D curvature image is generated. Firstly, 0 or 1 values are assigned to each voxel depending on the voxel position: 1 is assigned if the voxel is inside and 0 is assigned if the voxel is outside the fragment region. After that, the surface voxels are defined as 1-value voxels adjacent to 0-value voxels. The 3D curvature image is generated by setting K_{abs} to each voxel belonging to the fracture surface and 0 to the rest of voxels, where $K_{abs} = |k_1| + |k_2|$. k_1 y k_2 are the maximum and the minimum curvature respectively, and are obtained from K and H

$$K = \frac{h_{xx}h_{yy} - h_{xy}^2}{(1 + h_x^2 + h_y^2)^2} \tag{2.1}$$

$$H = \frac{(1 + h_x^2)h_{yy} + (1 + h_y^2)h_{xx} - 2h_x h_y h_{xy}}{2(1 + h_x^2 + h_y^2)^{3/2}} \tag{2.2}$$

where $h(x, y)$ is a quadratic function fitted to 3D points generated from the surface voxels. Once the 3D curvature image is generated, an interactive line-tracking software allows extracting the fracture zone.

Tassani et al. [96] perform a comparison with healthy models with the purpose of identifying trabecular tissue in fractured zones. To that end, the authors compare the fractured region of interest in both pre-failure and post-failure slices. These regions are identified as disconnected trabecular tissue in the slice. If the regions of interest of both slices overlap less that a predefined threshold, the region is classified as broken. The threshold is determined by minimizing the Root Mean Square Error (RMSE) between resulted and manually calculated values

$$RMSE = \sqrt{\frac{\sum_i (a_{i(x)} - v_i)^2}{n}} \tag{2.3}$$

where $a_{i(x)}$ and v_i are the calculated and the visually obtained values respectively and

n is the number of analysed cases. Finally, they applied a median filter to remove the generated noise.

Interactive methods have also been proposed to identify fracture surfaces before being used in virtual craniofacial reconstruction [8, 17, 16]. In these works, fracture contours are extracted interactively from segmented bone fragments. With that aim, user has to select points belonging to the fractured area and then a contour tracing algorithm generates the rest of the points. Once the fracture contours are calculated, the 3D surface is generated by collating the contours extracted from each slice.

2.1.2 Generation of 3D models of bone fragments

The main aim of this stage is to provide useful models for simulating the fracture reduction . Depending on the objective of the simulation, these models can be volumes, points clouds or meshes. Volumes and points clouds can be directly extracted from segmented CT stacks; hence it is not necessary to detail the procedure used to generate them. In some cases, the models used are meshes, since simulation methods require geometric algorithms. In order to extract meshes from CT stacks, MC [63] based methods are the most widely used because of their simplicity and speed. The size of the models generated by these methods depends on the resolution of the medical images and the complexity of the segmented bone structures. With regard to the generation of fractured bone models, the MC algorithm has been used to generate meshes that represent acetabular fractures [27], fractured pelvic bones [29, 57] and proximal humerus fractures [30]. This algorithm has also been used to generate fragment models to represent highly fragmented bone fractures [105] and to provide models for a virtual orthopaedic surgery simulator [100]. Finally, Fornaro et al. [28] present an interactive surgical planning tool for reducing acetabular fractures in which models are generated by using a variant of the classical MC algorithm: the generalized marching cubes.

Considering the previous revision, the MC algorithm is used in most cases in the generation of bone fragments models from CT scans. Nevertheless, MC produces complex and noisy models because of the complexity of the medical images. Bone fragment models generated with MC contain a large amount of geometry and lacks topology in most situations. This can be a drawback for enabling proper visualization and real-time interaction, or even for processing the generated models during a simulation.

However, the use of optimization techniques may reduce the complexity of the problem, as will be shown in the following subsection.

2.1.3 Optimization of bone fragment models

The 3D representation of bone fragment models can be post-processed with the purpose of obtaining desirable properties for the subsequent simulation stage. Smoothing, simplification or re-meshing are the optimization techniques that are most commonly applied to improve the features of the models and thus enable interaction and enhance visualization and manageability. Re-meshing techniques can only be used with mesh-based models; the other techniques can also be applied to point-based and volume models. The models obtained from medical images are usually huge and complex; hence it is very difficult to obtain interactive response time in simulation. Moreover, a re-meshing procedure can be required to obtain a more homogeneous model in order to improve visualization. These optimizations may also be carried out during the generation of the bone fragment models. For example, the contours extracted during the segmentation process can be simplified to obtain a smoother mesh. In this subsection, the main goal of these techniques and different aspects to be considered when bone fragment models are post-processed are briefly described. Table 2.1 shows the benefits and drawbacks of each type of technique regarding models reconstructed from medical images.

The application of smoothing techniques to bone fragment models aims at denoising the area surrounding the bone and to extrapolating missing information between slices, therefore improving visualization. This step is especially necessary when the distance between slices is large or the resolution of the image is not good enough since the reconstructed model can present ripples. The main methods used to smooth 3D models are reviewed in [42, 24]. Regarding meshes, smoothing functions are able to process vertex positions and normals, considering their connected neighbour vertices, in order to obtain better-shaped models. As a result, the curvature and thus the visual aspect of the model are improved. This type of filters should be used with caution since they can remove not only noise, but also small features from the models.

With the aim of examining the entire fracture area in detail for diagnosis, a large number of high resolution images are required and, as a consequence, the models obtained are big. The required computational cost can easily overwhelm a processor of a common desktop PC and hinder interactivity. In contrast to re-meshing, the simplification of models aims at reducing the size of the models while minimizing the impact on their morphology and properties. Nonetheless, it is crucial to avoid oversimplification of the bone models in order to preserve their original features. Simplification techniques allow a faster processing and thus enable pre-visualization and interaction with bone fragment models. In contrast, important features can become blurry, especially in the fracture area and near joints. Several studies have been published in order to compare strategies to simplify 3D meshes [18, 67], point clouds [76] and volumes [24].

Re-meshing is a procedure intended to improve the quality of a geometric model. It is a useful technique in the case of models reconstructed from medical images, especially for visualization purposes. The application of this procedure may also involve a simplification of the model, but this is not its main goal. After applying a re-meshing, the quality of the model is enhanced by modifying properties such as sampling density, regularity, size, orientation, alignment or the shape of the models. A complete review of re-meshing techniques is presented in [2]. A common issue with medical images is the noticeable difference between the distance between two adjacent pixels with respect to the gap between consecutive slices. This is the cause of irregularities in the size and shape of the geometrical primitives. In this case, re-meshing techniques can be used to enhance the quality of the models and thus their visualization. Nevertheless, re-meshing is difficult to apply in the fracture area and near joints because of the heterogeneity of trabecular tissue.

2.2 Computer-assisted fracture reduction

Computer-assisted bone fracture reduction is the process by which bone fragments are relocated with the aim of recovering their original position. Moreover, in the majority of cases it is necessary to stabilize the fracture by using fixation devices such as plates and screws. This procedure requires resolving a variety of problems that depend on both the types of bone and fracture. In the case of a simple fracture, the reduction consists of aligning the two bone fragments in order to recover their original position. If the fracture generates more than two fragments, a previous procedure is required to solve the puzzle. For that purpose some approaches propose matching the fracture

Table 2.1: Benefits and drawbacks of optimization methods applied to bone fragment models.

Technique	Input data	Benefits	Drawbacks
Smoothing	Meshes, volumes	Denoise bone fragment models, extrapolate missing information between slices, improve visualization	Small features can be removed from the trabecular tissue, especially in the fracture area and near joints
Simplification	Meshes, point clouds, volumes	Bone fragment models can be processed faster, enables pre-visualization and interaction with huge bone fragment models	Important features can be removed from the trabecular tissue, especially in the fracture area and near joints
Re-meshing	Meshes	Better sampling density of bone fragment geometry, improves performance	Difficult to be applied due to the heterogeneity of trabecular tissue

zones, and therefore they present algorithms generated in order to calculate these zones. Subsection 2.2.1 describes the most recent works to compute the reduction of simple fractures. In Subsection 2.2.2, the proposed approaches to match and register bone fragments in complex fractures are analysed.

2.2.1 Simple bone fractures

The reduction of a simple bone fracture involves the alignment of the two bone fragments, so that the bone recovers its original shape. This procedure usually requires an initial alignment to approximately recover the position and orientation of the bone fragments, and then a registration process to accurately join them. Nevertheless, some works avoid performing the initial alignment, since their proposed registration method is able to fully relocate and reorientate the bone fragments. In spite of the fact that a simple fracture usually does not require a computer-assisted planning, computer-assisted fracture reduction is usually performed in order to make tele-operation processes easier. Table 2.5 summarizes all the reviewed studies on computing the reduction of simple bone fractures.

Ron et al. [82] present a method to reduce distal and proximal femur fragments by calculating the periaxial rotation of healthy and fractured femurs from pre-operative CT.

Before that, a previous manual alignment is required. The method takes into account that the desirable periaxial rotation value should be symmetrical to that of the healthy bone and is applicable in closed femoral fractures. Westphal et al. [103] perform the entire fracture reduction interactively . For that purpose, they present a telemanipulator system that allows surgeons to guide a robot by using a joystick which provides haptic feedback. In order to provide visual information, CCD-cameras simulate the X-ray device by taking images of the fractured bone from different angles. With the goal of testing the system, they attach receivers to both ends of a plastic bone before fracturing it. These receivers let them quantify how precise was the reconstruction. Kronman and Joskowicz [55] present an automatic algorithm for pairwise bone fragment fracture reduction. The fracture is reduced by identifying the contact surfaces of the bone fragments and aligning them with the purpose of maximizing the contact area between them. First, a coarse alignment is carried out using Principal Component Analysis (PCA) . Then, the contact surfaces are aligned by using Iterative Closest Point (ICP) rigid registration . In order to compare the results obtained by their method with the ground truth, they developed a novel approach for fracture simulation of healthy bones using realistic fracture patterns.

Registration algorithms have been used to reduce simple fractures. Bhandarkar et al. [8] use a hybrid DARCES-ICP algorithm for aligning and registering surfaces in the reduction of craniofacial fractures. Chowdhury et al. [16] present a modification of that work. Once fragment models are generated, they propose to use the ICP algorithm and a Maximum Cardinality Minimum Weight (MCMW) bipartite graph as registration algorithms. In this case, an initial alignment is not required. In these two works, fracture contours are extracted interactively from segmented bone fragments. With that aim, the user has to select some points belonging to the fracture area, and then a contour tracing algorithm generates the rest of points. Once the fracture contours are calculated, the 3D surface is generated by collating the contours extracted from each slice.

Several works propose the use of bone templates to perform the fracture reduction in simple cases. This template may be an intact bone, the symmetrical bone or even a bone artificially constructed based on statistical knowledge. These works do not require an initial alignment, since the templates determine the position and orientation of each bone fragment. Maubleu et al. [66] simulate the reduction of a fractured zygomatic bone. With the aim of defining the intact template, they use the midsagittal plane to mirror the healthy side of the bone. This midsagittal plane is defined by the foramen caecum, the posterior extremity of the sphenoid crest and the middle point between both clinoid apophysis. Authors indicate that these three points are defined by the surgeon. In the last step, after matching the mirrored side with the bone structures surrounding the fracture, a rigid registration is performed between each zygomatic fragment segmented and its target position. Gong et al. [33] use an anatomical atlas to register fragments of the distal radius. In this case, the atlas is constructed from human corpses and patient data. Firstly, each segmented volume, which is created from back-projections of 2D segmentations, is properly positioned on the atlas. Then the atlas is deformed to maximize the overlap area between its 2D projection and the individual fragments in the segmented regions.

2.2.2 Comminuted bone fractures

The reduction of a complex fracture requires finding the correspondence between different bone fragments. It is important to consider that, unlike other reconstruction problems, bone fracture surfaces may share matches with more than one fragment; hence the matching between fragments may not be one-to-one . In order to complete the fracture reduction, all bone fragments have to be translated to their original position. Some of these fragments can be discarded if they are too small to be used in the reduction (according to a surgical criterion). Comminuted fractures are usually produced by high energy traumas, thus some fragments can be deformed. These deformations notably increase the difficulty of recovering the original shape of the bone. After the matching procedure, some approaches propose performing a final registration between fragments in order to improve the final result.

All the reviewed studies of computer-assisted complex fracture reduction are summarized in Table 2.6. Some applications have been developed with the aim of performing a virtual reduction of a comminuted fracture interactively . Cimerman and Kristan [19] present an application to interactively compose pelvic and acetabular fractures. Acetabular fractures can also be reduced by using the software presented by Hu et al. [43]. Wilson et al. [106] developed an application to manipulate bone fragments reconstructed from CT scans. Due to the complexity of this type of fracture, manual alignment of the fragments becomes a difficult task. In order to make it easier, some authors propose the use of haptic devices [28]. The haptic feedback provided by these devices eases the correct alignment of the bone fragments. However, this remains a complicated procedure even for an expert.

In order to overcome the problems of the methods described above, other studies match the bone fragments manually or semi-automatically and then perform a final alignment. Some of these approaches propose matching fracture areas . Willis et al. [105] propose that the user interactively selects fracture surface patches in pairs that coarsely correspond. Zhou et al. [113] intend that the user manually specifies the matching surface regions between fragments. To that end, they provide an interactive system that allows the user to manipulate bone fragments; hence the user has to specify fragment surface matches and initiate pairwise and global alignments. The interactive method introduced in [105] is improved by Zhou et al. [114]. The algorithm provides a user-directed search in order to match the fragments. To separate fractured and intact surfaces, they use a two-class Bayesian classifier based on the intensity values previously mapped on the surface vertices. Winkelbach et al. [108] take advantage of the specific features of long bones. Using a voting system, they calculate the bone shaft axis and the circumference of the long bone fragments and separate the fracture surface. Then an initial solution is computed considering all these calculations. The correct position of the fragments of a broken cylinder structure is also estimated by Winkelbach et al. [107]. They use a two-step Hough-like voting mechanism to measure the orientation and position of the cylinder axes for each fragment. These methods cannot be applied to the reduction of other types of fracture, since they take advantage of the special features of long bones.

Other proposed studies focus on the reconstruction of complex craniofacial bone fractures. Chowdhury et al. [15] identify fracture surfaces using a Maximum Weight Graph Matching (MWGM) algorithm. In this graph, the nodes are the fracture surfaces and the edge weights are treated as elements of a *score matrix*. A given pair of fracture surfaces will have a high score if they are determined to be spatially proximal and to exhibit complementary fracture surface characteristics. A variation of this work is presented in [17]. In order to match the fracture fragments, the authors formulate a matrix score based on the appearance of mandibular fragments in the CT image sequence. As in the approach proposed in [15], a pair of fracture surfaces has a high matching score if they are determined to be proximal and they exhibit complementary fracture surface characteristics. Chowdhury et al. then propose to use the MWGM algorithm to identify the fracture surface pairs in polynomial time. To calculate the correspondence for a given pair of fracture surfaces, they use the MCMW bipartite graph matching algorithm.

Finally, the reduction of complex fractures can also be computed by using a healthy bone as a template . In these studies, the contralateral bone [75, 30], an artificially constructed bone [97] or a statistical bone [68, 1] are used to perform the initial alignment. These approaches are restricted to the bones defined by the templates and, in most cases, require prior work to generate the artificial bone. Moreover, recent studies show that the contralateral side should be use with caution as a template or even as a ground truth because differences in shape between symmetrical bones can exist [59]. Okada et al. [75] study and evaluate several methods for positioning bone fragments of proximal femur fractures. The proposed methods are based on the registration of the bone fragments using the contralateral bone shape, fracture lines or both simultaneously. Fürnstahl et al. [30] present an approach developed to compute the reduction of humerus fractures. For that purpose, they use the contralateral bone to perform a coarse initial alignment of the bone fragments by using ICP.

Thomas et al. [97] generated identical tibiae from blocks of high-density polyurethane foam and fractured them into 10-15 pieces by using an instrumented drop tower. These tibiae were scanned before and after being fractured in order to use them as a template and to test the accuracy of the reconstruction. After that, the fracture was composed by matching each native surface to the intact template using an iterative registration function built into the Geomagic Studio Software. Kato [52] proposes performing the fracture reduction by aligning the fragments to a template. For that purpose, they formulate the problem as an affine puzzle. Moghari and Abolmaesumi [68] generated an atlas by using PCA in a bone population generated from CT scans. Afterwards, a local registration process is performed for each bone fragment. Albrecht and Thomas [1] also propose using a statistical shape as intact template when the contralateral bone or the bone before being fractured is not available. In these situations, they use the ICP algorithm to perform a rigid alignment of the fragments, and then they adapt the statistical shape model to the bone fragments. Due to the fact that the adaptation is only possible if the fragment is already aligned, and the rigid alignment depends on the model adaptation, the authors propose solving these two steps simultaneously in an iterative scheme.

After the matching procedure, most of the proposed studies perform a final alignment in order to improve the final result . In many cases, the final alignment is performed by registering fracture zones to each other. These fracture zones can be surfaces [107, 105, 68, 114], lines [75, 30] or points [15, 17], depending on the models used to represent bone fragments and the techniques applied to calculate the fracture zones. In other cases, the final alignment is performed by registering each fragment to the template [97, 1]. The latter strategy is more robust for missing and deformed fragments, but the results obtained rely on the template used. In order to perform the registration, an ICP-based algorithm is utilized in most cases [107, 15, 105, 75, 114, 17, 97, 30, 1]. As an alternative, Moghari and Abolmaesumi [68] use an unscented Kalman filter-based registration .

2.2.3 Stabilization

After a fracture reduction, the next step can be to place fixation devices in order to stabilize the fracture. This procedure requires the user to decide which fixation devices will be used, select their location, and adapt them to the shape of the bone. Currently, certain authors propose to perform this procedure interactively like Cimerman and Kristan [19]. In that study, plates and screws are manually chosen by the surgeon. The contouring of the plate is performed automatically and the screws can be interactively inserted into the plate or across the fracture. In order to ease this process, the representation of the bones can be made more transparent. Hu et al. [43] propose using a commercial 3D modelling tool to generate plates and screws and place them across an acetabular fracture.

Other studies propose a semi-automatic procedure to determine the type of fixation needed and its position. Fornaro et al. [28] present a surgical interactive tool that enables the adaptation of appropriate osteo-synthesis implants onto reduced virtual pelvis fractures. In order to achieve this, the user first draws a sketch of the desired plate placement directly onto the bone surface using a haptic device. The system then automatically contours the tetrahedral model of a reconstruction plate of a user-selected type onto the virtual bone surface according to the sketch. Thereafter, the user can place screws of different lengths either through the plate holes at angles restricted by the type of implant or freely into the pelvic bone. Then, a report is generated including relevant measurements and type and size of osteo-synthesis implants, as well as bending and torsion angles of fixation plate segments in all three planes. Finally, the surgeon uses this information to manually contour osteo-synthesis implants pre-operatively according to the report. Results show a generally very good match between planning and final execution. However, some plates could not be placed exactly as planned because soft tissue interfered with the placement of the screws and hence the plate had to be tilted slightly.

Musuvathy et al. [71] propose a new method of customizing fixation plates to repair bone fractures. In order to avoid the manual adaptation of polygonal plate models to the bones of patients, they propose a semi-automatic adaptation of parametric Computer-Aided Design (CAD) models using Non-Uniform Rational B-Splines (NURBS) to generate customized plates. Thus, the plates conform to the desired region of the bone surface of patients. This enables an efficient and accurate approach that is also computationally suitable for interactive planning applications. Moreover, the patient-specific customized plates can then be produced directly from the adapted CAD models with a standard Computer Numerical Control (CNC) machine before surgery. This may dramatically reduce the time spent in the intervention, increase precision of the procedure and, as a result, improve the outcome for the patients.

A modelling and visualization system for simulating the pre-operative pre-bending of a steel plate is presented by Liu et al. [62]. The system allows obtaining the geometric parameters of the target fixation plate. The authors use an accurate NURBS Surface fitting method to construct the virtual plate. Chen and Huang [14] present an interactive tool to plan the stabilization of humeral shaft fractures using a semi-automatic fragment reconstruction approach. Fracture fragments are reduced anatomically by selecting three characteristic points through a manual operation. Finally, the appropriate plates are chosen from the internal fixation devices database of the system.

Intra-Medullary Nail (IMN) is a fixation device which is placed in the medullary cavity of the bone. In IMN surgical operations, one of the main tasks for surgeons is to know the positions and orientations of Distal Locking Holes (DLHs). This procedure is necessary for the insertion of distal transverse interlocking screws. The distal holes on an IMN, which are inside the intramedullary canal of the bone, can only be seen in a lateral X-ray view. For the standard surgical procedure, the localization of the distal hole axes is a trial-and-error process which results in a great use of surgical time and a large dose of

X-ray exposure. Zhu et al. [115] present an algorithm designed to obtain the 3D position and orientation of the distal hole axis. The algorithm first derives the nail axis through two X-ray images. Then the distal hole axis is calculated by projecting back the hole boundary onto the X-ray image from a lateral view to the 3D space. A least squares method is used to determine the centres of the front and the back hole iteratively. Zheng et al. [112] present an approach for solving this problem based on two calibrated and registered fluoroscopic images. The problem is formulated as a two-stage model-based optimal fitting process. The first stage, nail detection, automatically estimates the axis of the distal part of the IMN by iteratively fitting a cylindrical model to the images. The second stage, pose recovery, resolves the translations and the rotations of the DLHs around the estimated axis by iteratively fitting the geometrical models of the DLHs to the images. An iterative best matched projection point algorithm is combined with random sample strategies to effectively and robustly solve the fitting problem in both stages.

2.3 Analysis and validation of a computer-assisted reduced fracture

Once a bone fracture has been stabilized, it is useful to check how good the stabilization is. In order to check this goodness, an analysis of the reduced fracture can be performed. This analysis can be accomplished using different procedures. One of these may be to compare the result of the fracture reduction with a healthy bone, the bone of another patient or the symmetrical bone of the patient . On the other hand, the specialist may evaluate the quality of the computer-assisted pre-operative planning by comparing its results with the real intervention carried out afterwards.

2.3.1 Geometric accuracy analysis

In order to measure the geometric accuracy of the performed fracture reduction, most of the proposed methods compare the results obtained with the ground truth . Ron et al. [82] determine the usefulness of their method for reducing femur fractures based on the periaxial rotation. To that end, they compare the periaxial rotation value of the broken femur to the value of its mirror image. For that purpose, they use both real and cadaver data. If the reduction is successful, periaxial rotation values should be equal. Moghari and Abolmaesumi [68] also use bones from corpses, which are artificially fractured, to evaluate their femur fracture reduction method based on an atlas. Fürnstahl et al. [30] use cadaver bones and real clinical cases to measure the accuracy of their proposed method for reducing humerus fractures. Authors compute intersection and gap errors , surface smoothness, and rotation and translation errors.

A synthetic material with a mechanical behaviour very similar to cortical bone, with similar appearance in X-ray and prone to breaking into fragments in similar number and form is used by Zhou et al. [114, 113] to evaluate their method for reducing highly comminuted bone fractures by calculating the alignment error. Blocks of high-density polyetherurethane foam are also used to test the platform for solving comminuted articular fractures presented by Thomas et al. [97]. To test the reduction of cylindrical bone fractures, Winkelbach et al. [107] compare the obtained results with virtual manual repositions of the reconstructed bone fragment surfaces using synthetic bones by computing rotational and translational errors .

With the aim of validating the composition of zygomatic fractures, Maubleu et al. [66] remove virtually the bone from healthy patients virtually and then they apply their proposed fracture reduction method and compute distance errors. Gong et al. [33] generate random synthetic fractures in order to test their proposed method. For that purpose, they evaluate the error by calculating distance errors. Virtual fractures have also been randomly generated in order to test the method proposed in [1]. Other studies use not only virtual fractures but also real fractures to measure the quality of a fracture reduction method [75]. In this last work, the authors measure the femur reduction from the rotation error of a fragment and the distance between the points that correspond with the fracture area of two fragments.

In cases in which the alignment of the fragment is performed by registering the points belonging to the fracture surface, the error in approximating the points can be used as a measure of the quality of the reduction . Bhandarkar et al. [8] use the Mean Squared Error (MSE) to measure the goodness of the reduction carried out using ICP, DARCES and a hybrid approach. This measure is also used by Chowdhury et al. [16, 17] to determine the accuracy of a craniofacial reduction. Table 2.7 summarizes all the studies reviewed that analyse the geometric accuracy of virtually reduced fractures.

2.3.2 Planning evaluation

As mentioned in Chapter 1, the simulation of a fracture reduction is sometimes used for planning a later surgery . In these cases, the expert can evaluate the quality of the simulated process by comparing it with the actual intervention performed a posteriori.

In the literature, there are studies that evaluate the simulated process by testing the suitability of its utilization in planning a real fracture reduction. Cimerman and Kristan [19] have tested their interactive application by planning the reduction of pelvis fractures and checking whether the planned procedure is followed during the actual intervention. Moreover, the number and length of screws inserted during the operation are also analysed. Fornaro et al. [28] evaluate the usefulness of an interactive tool for planning acetabular fracture reduction. To that end, the tool was used to plan 7 real cases and the proposed plan was compared with the surgery performed later. In addition, the positions of the screws and plates were also compared. Acetabular fractures were also planned using the procedure described in [43]. The proposed procedure allows the user to interactively identify and relocate bone fragments. To evaluate the process, the authors used the proposed tools to plan 7 clinical cases in which surgery was subsequently performed. In the evaluation step, they checked whether the plan was followed during the actual intervention. In addition, the number of screws and the length of the plates used during the real intervention were compared with respect to those planned.

Tomazevic et al. [98] present a set of interactive tools for identifying articular fractures that can be useful for intervention planning. These tools are evaluated in the planning of real interventions by describing their utilization. Suero et al. [94] evaluate the use of their software for the reduction of tibial plateau fractures. For that purpose, they check in the planning phase the time required and whether the 3D reduction is successful. Lee et al. [57] present a planning system for reducing pelvis fractures and positioning fixation devices. Besides the tests performed with synthetic bones, the system is applied to plan a real surgery. During the evaluation, the features found by applying the method in specific cases are extracted, considering the time required to complete each step, the results obtained and the level of interaction required by the surgeon. Chen and Huang [14] evaluate their framework for interactively planning humeral shaft fractures in order to determine whether computer-assisted pre-operative planning improves the clinical outcomes of humeral shaft fractures. They used the Intra-class Correlation Coefficient

(ICC) to test for reliability.

2.4 Conclusions and future research

Due to the huge complexity of some fractures, computer-assisted techniques have been required to ease and provide technological support to the pre-operative planning of the fracture reduction process, reducing in this way the surgery risk and diminishing the recovery time of the patient. In this chapter, the techniques and approaches proposed in the literature to help medical specialists in this process have been reviewed, from the entry of a patient (generation of bones and fragments models) to the analysis of the final composition. The methods have been summarized and classified, and their main advantages and shortcomings have been highlighted and discussed. The review has revealed that none of the stages of the computer-assisted fracture reduction process is completely resolved, neither the identification and generation of fragments, nor the fracture reduction itself. As a consequence, this field of research is still open and faces important challenges in a mid-term period.

The identification of bone fragments requires manual user interaction and sometimes expert knowledge. Ideally, all the bone fragments would be segmented automatically and simple fractures would be identified without user intervention. For this purpose, new methods to automatically separate wrongly-joined fragments are needed. The use of more precise medical images, such as CT, could avoid fragments appearing together in most cases. However, these images are not always available and have an important radiative impact on the patient. After the identification process, the next step consists of placing all the bone fragments in their correct position to obtain the original shape of the bone. In the future, new methods need to be developed to relocate all the fragments properly in all types of bones and fractures without using any template. The virtual stabilization of a fracture is almost an unexplored research field. New systems that suggest the fixation devices to be used and their position would be very helpful in the pre-operative planning of complex fractures. Finally, the development of better image acquisition techniques is mandatory in order to test the suitability of the results obtained by the computer-assisted fracture reduction process. All these advances should allow the automation of the process as it is already happening in other medical disciplines, as well as the development of more advanced commercial simulators. Ideally, this automation could lead to the development of computer-guided surgery systems using robots.

Table 2.2: Summary of the reviewed studies on identifying healthy bone.

Method	Variant	Evaluation set	Advantages	References
Thresholding	User defined threshold	Ulna, radius and carpus	Semi-automatic bone fragments separation	[73]
	Adaptive threshold	Calcaneus and vertebra	Automatic segmentation	[110]
Region-based	3D RG and ray casting	Inferior maxilar	Bone tissue classification	[70]
	3D RG and background growing	Skull	Threshold and seeds are automatically selected	[111]
Edge-based	3D deformable balloon model	Vertebrae	Vertebra separation	[65]
	Active contours	Knee	Bone contour extraction from CT images	[99]
	Level set and Willmore flow	Vertebrae	Deals with missing information	[60]
Statistical	Gaussian mixture models	Knee	Cortical tissue pixels classification	[5]
	Expectation maximization	Phalanxes	Semi-automatic segmentation	[81]
	Statistical texture	Mandible	Identification of different bone tissues	[72]
	Genetic algorithms	Mandible, skull and knee	Cortical and trabecular tissues separation	[44]
Mixed approaches	RG, active contours and region competition	Carpus	Combines the advantages of all the methods used	[87]
	Statistical classifiers and graph cuts	Hip	Detailed tissue classification	[64]
	Gaussian classificators, graph cuts and RG	Vertebrae	Automatic cortical and trabecular tissue classification	[3]

Table 2.3: Summary of the reviewed studies on identifying fractured bone.

Method	Variant	Evaluation set	Advantages	Constraints	References
Thresholding	Interactive methods	Humerus, clavicle and scapula	Accuracy	Time consuming	[98]
	Global fixed threshold	Femur and tibia	Simple and fast	Intensity values can differs between slices	[96]
RG	3D region growing	3D CCL and graph cuts	labelling performed during segmentation	Interaction can be required to split bone regions in complex cases	[38, 29]
	Multi-region growing	Pelvis	Automatic definition of threshold and seeds	Interaction can be required to join or split bone regions in complex cases	[57]
Watershed	Probabilistic watershed transform	Tibia, fibula and talus	Incorporate semantic information	Require pre-processing the image; fragment separation is not specified	[90]
Registration	Morphon	Hip	Automatic segmentation	Require templates	[78]

Table 2.4: Summary of the reviewed studies on identifying fracture zones.

Methods	Requirements	Interaction	Evaluation set	Achievements	References
Gaussian mixture models and Bayesian classifiers	-	Set threshold and subdivide fractured zones	Tibia	Identification of fracture zones in comminuted fractures	[105, 114]
Comparison of normal vectors	Cylindrical bones	-	Femur	Automatic identification in cylindrical bones	[107]
Curvature analysis	-	Extract fracture lines	Femur	The 3D curvature image eases the interaction	[75]
Comparison with healthy models	A healthy model	Visually check values to set the threshold	Femur and tibia	Interaction is only required to define the threshold	[96]
Contour tracing algorithms	-	Select points belonging to the fractured zone	Mandible	The user only has to select the end points of the fracture contour in each slice	[8, 17, 16]

Table 2.5: Summary of the studies on computing the reduction of simple bone fractures.

Initial alignment	Registration	Evaluation set	Achievements	References
Manual	Align periaxial rotation	Femur	Yield predictable results	[82]
Interactive	Interactive	Femur	Haptic interaction	[103]
PCA	ICP	Femur	Does not require user interaction	[55]
DARCES	ICP	Craniofacial	Combine advantages of DARCES and ICP	[8]
Not required	Maximum Cardinality Minimum Weight and ICP	Craniofacial	Prior alignment and/or templates are not required	[16]
Not required	Mirroring and rigid registration	Zygomatic bone	Prior alignment is not required	[66]
Not required	Atlas-based registration	Distal radius	3D segmentation is not needed	[33]

Table 2.6: Summary of the studies on computing the reduction of complex bone fractures.

Method	Variant	Evaluation set	Advantages	Constraints	References
Interactive tools	Classic 3D interaction	Pelvis, acetabulum and phalanges	Versatility	Time consuming and low accuracy	[19, 106, 43]
	Haptic interaction	Acetabulum	Haptic feedback	Time consuming	[28]
Match fracture areas	Manual	Tibia and fibula	Automatic alignment of comminuted fractures	Matches need to be manually specified	[113, 105]
	User directed search	Tibia	Enables multiple potential reconstruction alternatives	Contact surfaces between fragments need to be manually specified	[114]
	A voting system	Femur	Automatic calculation of fracture areas	Limited to cylindrical bones	[107, 108]
	MWGM	Craniofacial	Automatic fragment matching	Focused on a specific fracture case	[15, 17]
Register templates	Contralateral bone	Humerus and femur	Accurate reduction by registering fracture lines	Fracture lines extraction requires user interaction	[75, 30]
	Intact template	Tibia and acetabulum	Results can be accurately tested	The intact template is not always available	[97, 52]
	Statistical models	Femur	Does not require the contralateral bone	The results can be influenced by the statistical model	[68, 1]

Table 2.7: Summary of the studies analysing the geometric accuracy of virtually reduced fractures.

Method	Variant	Evaluation set	Advantages	Constraints	References
Compare to the ground truth	Cadaver bones artificially fractured	Femur and humerus	Fractures can be simulated with great veracity	Hard to obtain bones of corpses	[82, 68, 30]
	Synthetic fractures	Femur and tibia	Easier to obtain than bones of corpses	Differences between the synthetic and the real bone may exist	[107, 114, 113, 97]
	Virtually generated fractures	Zygomatic bone, radius and femur	Different test cases can be generated using only one real bone	It is difficult to properly generate a real fracture	[66, 33, 75, 1]
	Contralateral bone	Femur	Contralateral bone model is obtained easily	Bones are not completely symmetrical	[82]
Compare to manually reduced fracture	Calculate translational and rotational errors	Femur	Tested fracture cases are real	Manual reduction may contain error	[107, 75]
Calculate parameters after reducing real cases	MSE of the registration and matching scores	Craniofacial	Tested cases are real	Results are not evaluated using the ground truth	[8, 16, 17]

CHAPTER 3

SEGMENTATION AND LABELING OF FRACTURED BONE FROM CT SCANS

The identification of fractured bone from CT images consists of segmenting and labelling all the bone fragments represented in the images. Labelling involves the identification of bone fragments separately (Figure 3.1). This procedure is very important in medical visualization and simulation. In visualization, it allows removing noise and undesirable parts from the image; hence it lets the specialist focus better on the area of interest. In simulation, the use of models reconstructed from CT scans of patients allows customizing the simulation. Manual fractured bone identification is a very time demanding task and, in most of the times, requires expert knowledge. Thus, its automation would be very profitable. Moreover, close fragments can be joined after the segmentation because of their proximity and the resolution of the CT image. Interactive techniques obtain good results, but they also require a lot of time. Therefore, it is important to develop methods to automate the procedure as much as possible and also get accurate results. In this chapter, a new method to segment and label bone fragments from CT images is presented. On the other hand, the main issues to be considered when identifying both healthy and fractured bone are described. The presented method is based on 2D RG and requires minimal user interaction. In addition, the method is able to separate wrongly joined fragments during the segmentation process.

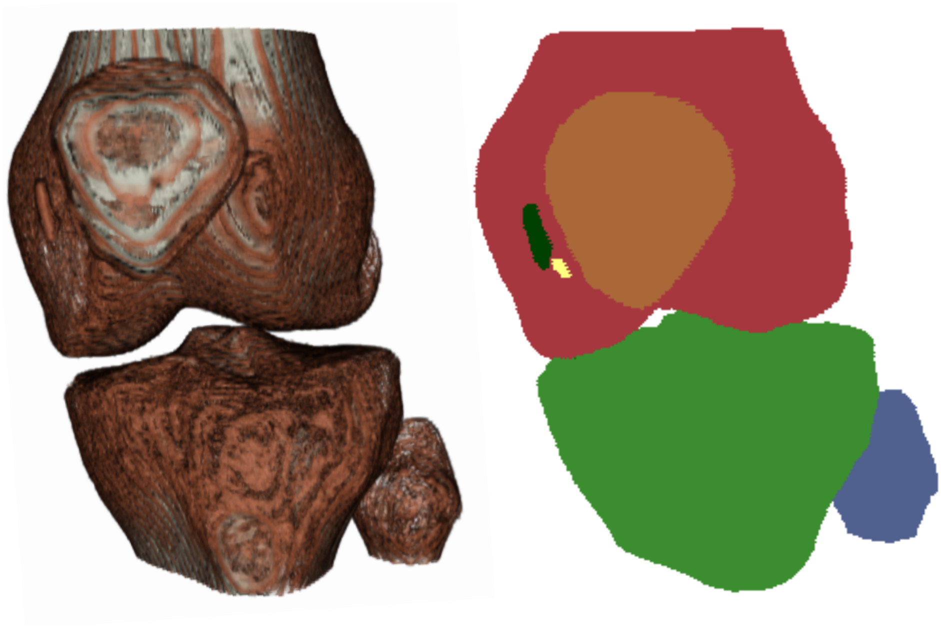

Figure 3.1: Volumetric visualization of a set of bone fragments generated from a CT stack using the presented approach (left). Since all the fragments are labelled, they can be displayed as individual fragments (right).

3.1 Issues for bone detection

3.1.1 Healthy bone

The segmentation of bone tissue from CT images is a complex process . It is difficult to find a solution that works in all cases. There are two very distinct zones in a bone: cortical and trabecular tissue. Cortical tissue is very dense and can be found in the outer part of the bone. Trabecular tissue is mainly in the inner part of the bone, it is more heterogeneous and has less intensity in a CT image. Furthermore, the intensity value for the same tissue differs between slices. This happens with both cortical and trabecular tissues. For instance, intensity values on the diaphysis and the epiphysis are different in a long bone (Figure 3.2). Near the joints, the cortical zone is very thin. This zone even disappears in the area closest to the join. Therefore, the transition of the intensity values near the joints generally appears to be fuzzy and some areas within the bone may have similar intensity than the soft tissue surrounding the bone. This may cause incomplete segmentation or overgrowing [57].

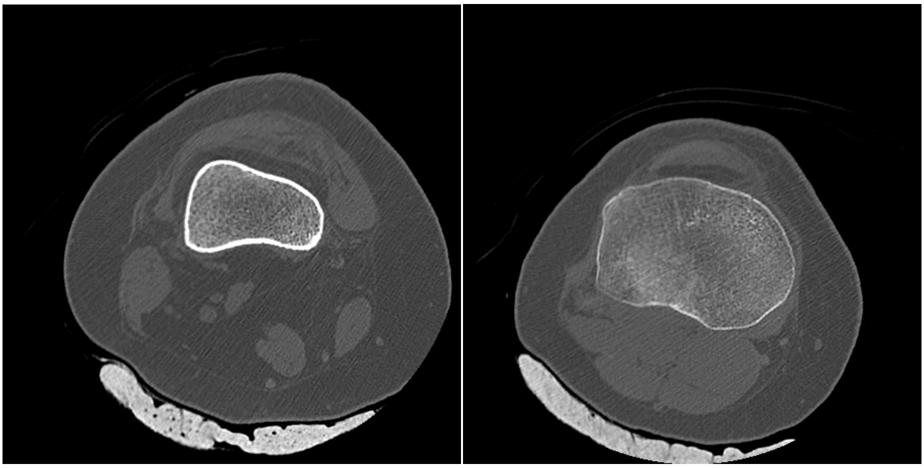

Figure 3.2: Two CT images belonging to the same patient dataset. The intensity values of the cortical zone are different in the diaphysis (left) and the epiphysis (right). The cortical area is much thinner in the epiphysis (right).

3.1.2 Fractured bone

Fractured bone tissue is more difficult to identify because it has some additional features to be considered. Due to the fact that bone fragments may have arbitrary shape and can belong to any bone in a nearby area, it is necessary to label all the fragments during or after the segmentation process. In some cases, this labelling requires expert knowledge. In addition, a priori knowledge cannot be easily used because it is uncommon to find two identical fractures and therefore it is difficult to predict the shape of the bone fragments, especially in comminuted fractures. On the other hand, bone fragments are not completely surrounded by cortical tissue, since they have areas on the edges without cortical tissue due to the fracture. Finally, proximity between fragments and the resolution of the CT image may cause that different fragments appear as one in the image. For this reason, smoothing filters should be used with caution. This type of filters can deform the shape of bone fragments and fracture zones, or even remove small bone fragments. In some cases, it is necessary to detect the fracture zone of each fragment after its segmentation. The fracture zone is the area of the bone where the fracture occurs and is composed of trabecular tissue (Figure 3.3). In situations in which bone fragments appear connected, it is difficult to accurately identify the fractured zone of each fragment. Therefore, post-processing can be necessary to delimit fracture zones in these situations .

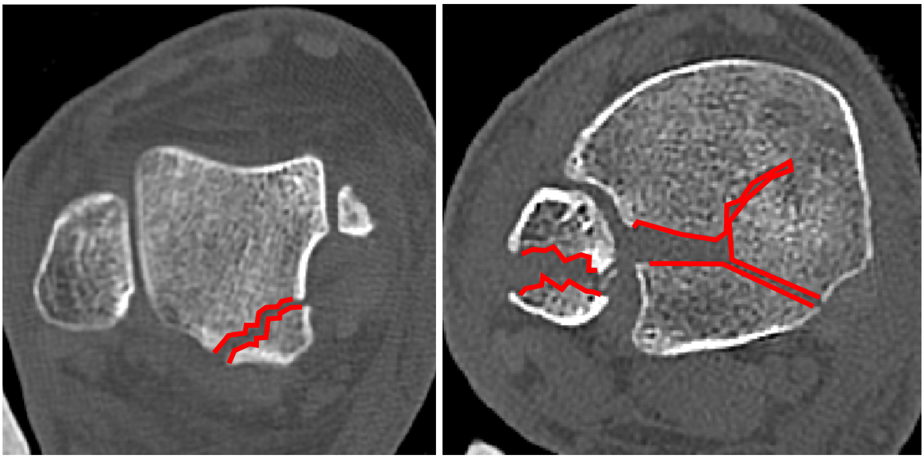

Figure 3.3: CT slices that represent some different simple bone fractures. Fracture lines are marked by hand in red.

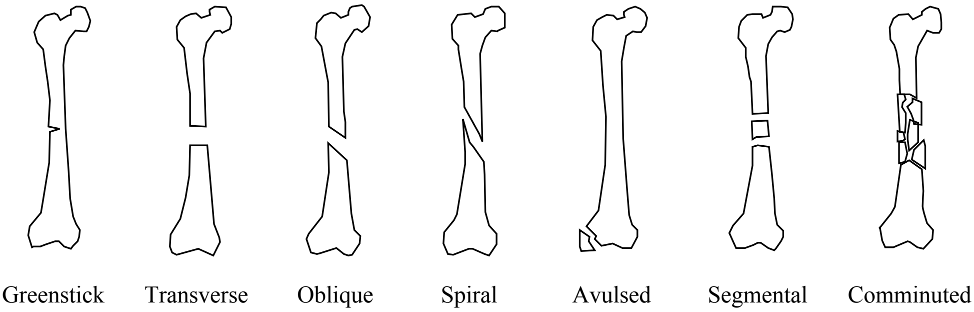

Figure 3.4: Fractured bones classified by their fracture lines.

The method applied in fractured bone identification depends on the fracture type. Based on the fracture line, a fracture can be classified as (Figure 3.4): greenstick, transverse, oblique, spiral, avulsed, segmental and comminuted [21]. In a greenstick fracture (Figure 3.5, a) there are no fragments because the bone is not completely broken. Thus, labelling is not necessary. Since the fracture barely changes the shape of the bone, segmentation methods that are based on previous knowledge are available. Nevertheless, the edges of the fracture zone, composed of trabecular tissue, may require special processing. The detection of the fracture zone is specially complicated since the bone is not completely broken and trabecular tissue is very heterogeneous. Therefore, the fracture zone can be fuzzy in the CT image.

Transverse, oblique and spiral fractures (Figure 3.5, b, c, and d) can be similarly treated during the segmentation. Despite having different fracture lines, these types of fracture generate two fragments with similar features. Labelling is necessary, but expert knowledge is not required. Segmentation methods that can be applied depend on the presence of absence of displacement. If there is no displacement (Figure 3.5, c, d, e, and f), they can be processed as a greenstick fracture, but considering that there are two fragments. These two fragments can be completely joined; hence an additional processing to separate them may be required. In order to detect fracture zones, the same issues applicable to greenstick fractures should be considered. In the case of having displacement (Figure 3.5, b), the probability that both fragments are jointly segmented decreases, and methods based on prior knowledge are almost discarded. In return, the fracture zone is easier to identify. Avulsed fractures normally occur near a joint, thus the fracture zone is composed almost exclusively by trabecular tissue and the boundaries of the fragments are weak. This makes the identification of the fracture zone difficult because practically the entire fragment is surrounded by trabecular tissue. Segmental fractures are simple fractures that generate three bone fragments. Therefore, they can be treated as transverse or oblique fractures, but considering that there are two distinct fracture regions.

Comminuted fractures (Figure 3.6) add some additional constraints; hence this is the type of fracture that is more complicated to be segmented. Comminuted fractures usually generate small fragments, and the bone may be deformed due to the fracture. This is because comminuted fractures are usually associated with crush injuries. In most cases, some fragments overlap in the CT image and require additional processing to be separated. Labelling is necessary, and expert knowledge is strongly required to identify fragments. The detection of fracture zones is difficult in this case. Due to the complexity of the fracture, several fracture zones are generated. Since the relationship between fragments in this type of fractures is many-to-many, it can be necessary not only to identify fracture zones, but also delimit which part of the fracture zone corresponds to each fragment. As mentioned before, some fragments can overlap due to the fracture and therefore post-processing and expert knowledge can be required to accurately identify fracture zones.

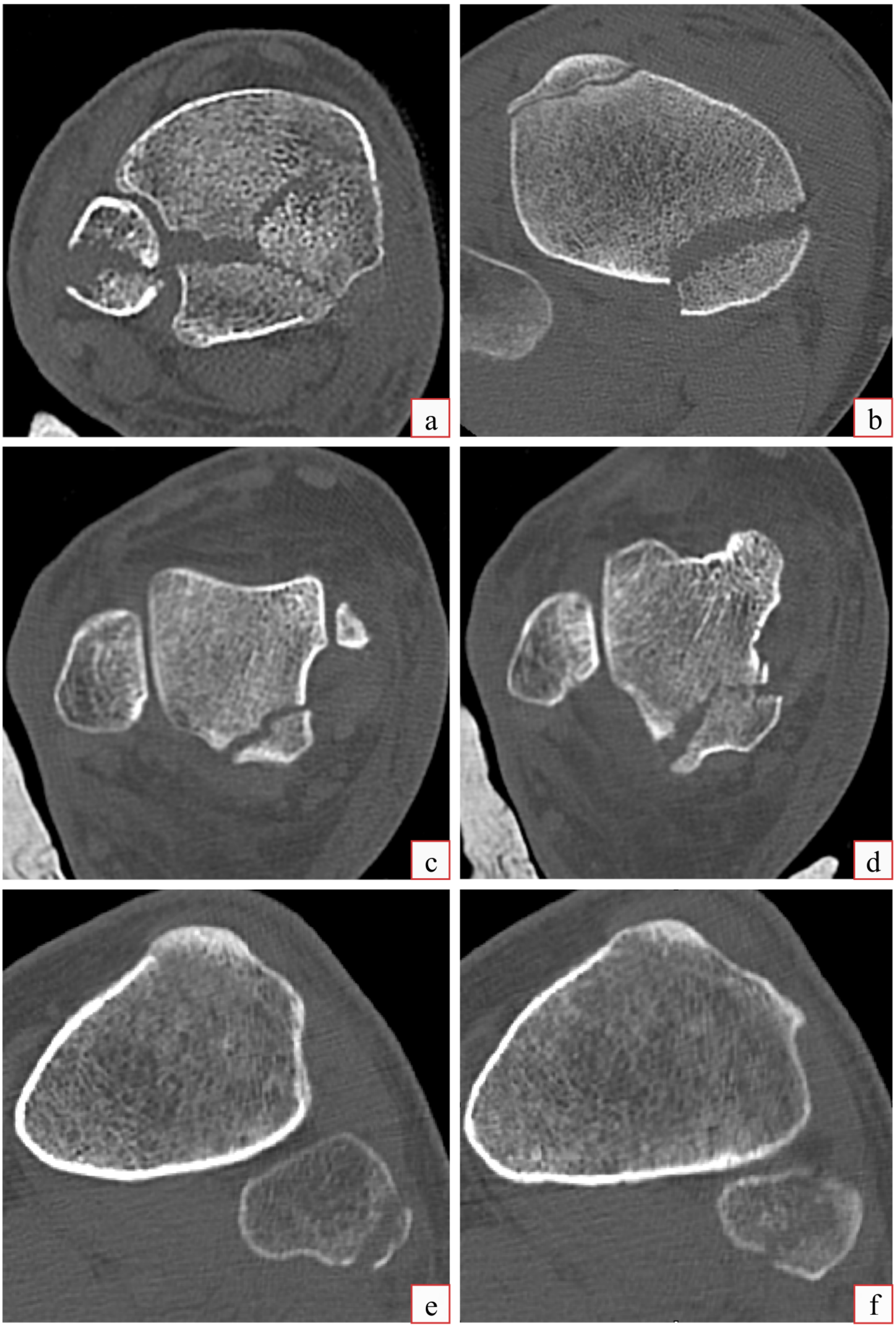

Figure 3.5: CT images that represent different simple fractures. (a) contains, among others, a greenstick fracture, since the bone is not completely broken. The remaining images contain simple fractures with (b) and without (c, d, e, f) bone displacement.

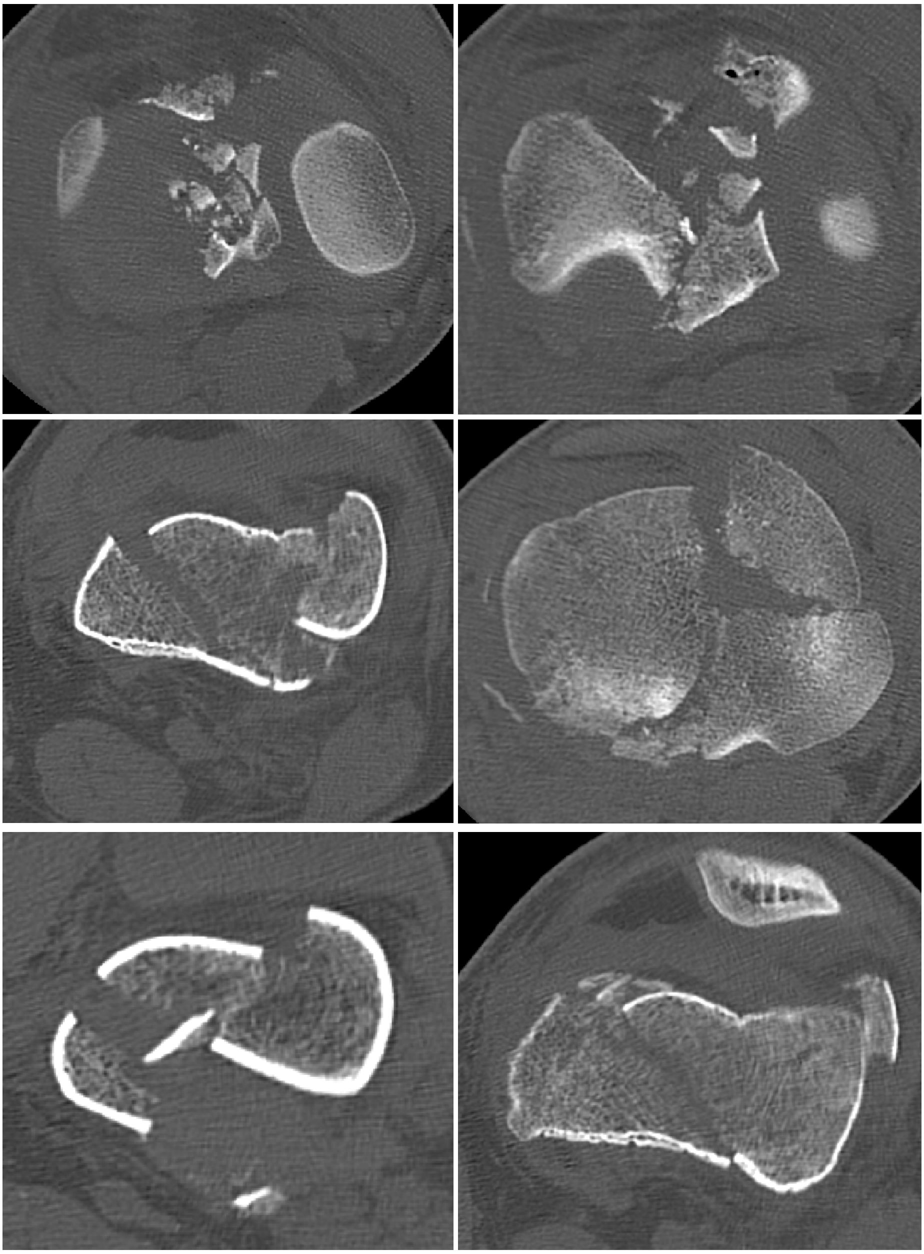

Figure 3.6: CT images representing highly comminuted bone fractures.

3.2 Fractured bone identification

As mentioned in Section 3.1.2, the identification of fractured bone requires not only segmenting bone tissue, but also labelling bone fragments. Moreover, erroneously joined fragments could need to be separated after the segmentation process. Therefore, expert knowledge is sometimes required. The implementation of an automatic procedure is not always possible. Nevertheless, user interaction must be reduced as much as possible because it enables time saving. With the aim of identifying bone fragments, an RG based method that only requires minimal user interaction has been developed.

For simplicity, the case of segmenting a single bone fragment will be considered. The procedure described in this section should be repeated to segment multiple bone fragments. The presented approach only requires that the user identifies bone fragments in the first slice where each one appears. For that purpose, the user has to place a seed inside each region of the target bone fragment. These seeds allow not only executing the segmentation method, but also labelling the bone fragments. After that, 2D RG is executed for each seed, and then all the seeds are propagated through the image stack. In order to discard noise data, regions with fewer pixels than a threshold, usually near 20, must be removed. This threshold value varies depending on the resolution of the image. Therefore, when a seed generates a region with a size that is smaller than the defined threshold, it is ignored. With the goal of smoothing the image and thus improving the segmentation result, a curvature flow filter is applied to each slice before each 2D segmentation process.

In contrast to other approaches based on 3D, the presented method easily allows detecting region overgrowing cases (Figure 3.7). Since the presented approach generates a 2D region for each slice, overgrowing can be detected and solved in the first slice that occurs. To achieve this, the proposed algorithm deals with some special cases that will be described in the following subsections.

3.2.1 Seed propagation

The segmentation algorithm starts at the slice in which seeds have been placed. The first step of the algorithm is to segment the slice using the first seed placed. For that purpose, 2D RG is used. After that, the obtained result is subtracted to the original image in order

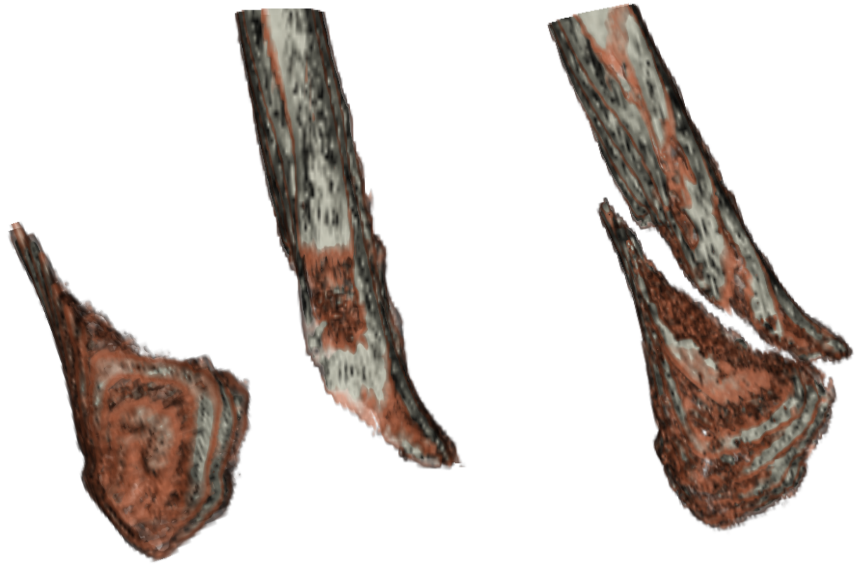

Figure 3.7: Single bone fragment segmentation results. Left, bone fragments segmented individually using the proposed method and later separated for viewing purposes. Right, classical 3D RG is not able to separately segment them.

to avoid that the subsequent obtained regions overlap and thus to discard seeds that do not segment anything (Figure 3.8). In addition, this subtraction procedure also prevents some over-segmentation cases. The entire process is repeated for each seed placed by the user. Upon completion of the segmentation with all the placed seeds, the results obtained with each seed are joined in a single image.

Once the first slice is segmented, all the seeds are propagated and the previously explained 2D segmentation algorithm is repeated for each slice. This approach eases the detection of wrongly joined fragments, since each seed is spread independently. To propagate a seed from a slice i to the next $i + 1$, first the seed is inherited from the previous slice. If this seed fails, all its neighbour pixels are considered as seed candidates. A seed fails when it is not able to segment a region that can be considered bone. If all its neighbours also fail, the seed is removed. The algorithm stops when all the seeds have been removed. The procedure is fully explained in Algorithm 1.

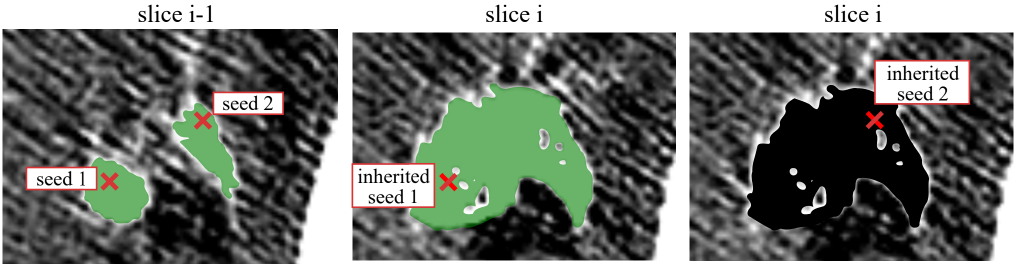

Figure 3.8: Left, two seeds segment two different regions that belong to the same bone in the slice $i-1$. Centre, the inherited seed 1 is able to segment the entire region in the slice i. Right, the region segmented by the inherited seed 1 is subtracted to the original image, hence there is no region to segment and the inherited seed 2 is discarded.

Algorithm 1 General seed spreading.

```
for all slices do
    for all seeds do
        count ← 2DRegionGrowing(seed)
        if count < MIN_BONE_SIZE then
            for all neighbourSeeds do
                count ← 2DRegionGrowing(neighbourSeed)
                if count ≥ MIN_BONE_SIZE then
                    seed ← neighbourSeed
                    subtractSegmentedRegion()
                    break
                end if
            end for
        else
            subtractSegmentedRegion()
        end if
        if count < MIN_BONE_SIZE then
            deleteSeed(seed)
        end if
    end for
end for
```

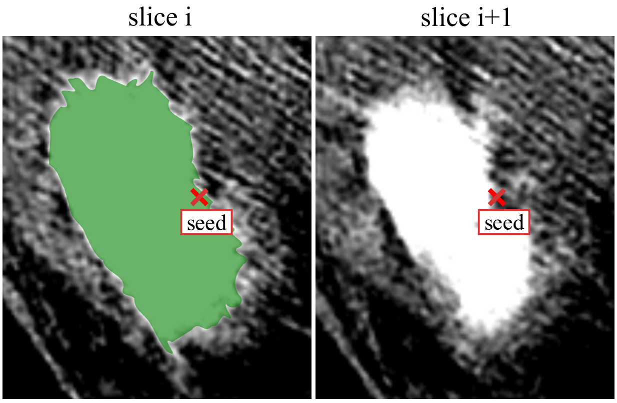

Figure 3.9: Special case 1. Two consecutive slices that show a considerable decrease in the size of the bone, hence the inherited seed cannot segment anything in the slice $i+1$.

3.2.2 Special cases

The previously explained seed spreading algorithm obtains good results with simple bone fragments. Nevertheless, the algorithm needs to consider some special cases to be able to segment more complex bone fragments. When a seed is spread from a slice to the next, two special cases need to be treated independently:

- The size of the target bone fragment region decreases significantly between two consecutive slices (Figure 3.9).
- The target bone fragment is divided into two or more regions between two consecutive slices (Figure 3.10).

If the size of a bone fragment decreases significantly between two consecutive slices, it is possible that neither the inherited seed nor its neighbours can generate a region (Figure 3.9). Especially if the inherited seed is located near the edge of the bone fragment. In this case, an alternative seed has to be selected by using other criteria. In order to detect this special case, the decrease of segmented pixels with respect to the previous slice is computed. If the decrease is significant, the case is treated independently as detailed below. The percentage of decrease considered as significant is established by experience. This percentage is set low enough to exclude false negatives, since the

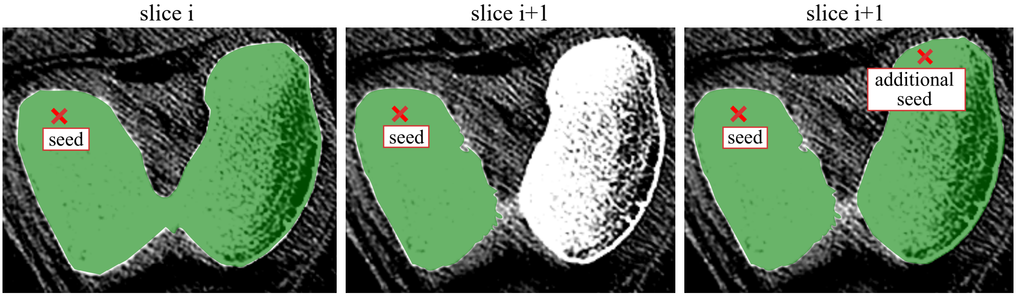

Figure 3.10: Special case 2. Two consecutive slices in which the bone is separated into two parts. Left, the seed is able to segment the entire region in the slice i. Centre, the region is subdivided in the slice $i+1$ and the seed can only segment one of the regions. Right, an additional seed allows to segment the other region.

proposed algorithm works well with false positive cases, although it takes a little longer. If this type of special case is detected, an alternative seed should be located. For that purpose, all the segmented pixels in the previous slice associated with the current seed are considered. To avoid testing all pixels and thus to optimize the method, a set of pixels is randomly selected and these pixels are considered as seed candidates. If one of the seed candidates works, it replaces the inherited seed. Otherwise, the inherited seed is discarded and its propagation ends.

The second special case occurs when a target bone fragment is divided into two or more regions when moving from one slice to another (Figure 3.10). In this case, the inherited seed can only segment one of them. With the aim of detecting this special case, the increment of segmented pixels between two consecutive slices is calculated. If there is a significant increment, it may be necessary to use an additional seed. The significant increase is determined by experience to avoid false negatives. As in the previous special case, false positives are accepted since they do not modify the result of the algorithm. To resolve this special case, first the image is segmented using the inherited seed. Then the obtained region is subtracted to the original image to avoid being re-segmented. In order to find an additional seed, all the segmented pixels in the previous slice associated with the inherited seed are considered again. As in the previous special case, a random sample of pixels is selected to avoid testing all pixels. After that, the resulting image from the subtraction is segmented using the randomly selected pixels as seeds. If an alternative seed is found, it is spread separately using a divide and conquer approach. This procedure is repeated until the increase ceases to be significant or none of the randomly selected pixels in the previous slice can be used as seed. Algorithm 2 shows the proposed method including the two special cases.

An additional special case appears when two different regions are joined when moving from a slice to the next. However, this case has not been specially treated because our method works correctly without modifications. This is because the region obtained with each seed is subtracted to the image to be segmented; hence in this case the second seed would have no region to segment and would be discarded.

3.2.3 Threshold definition

The intensity threshold for the RG algorithm is calculated automatically from the image data. On the one hand, intensity values differ between slices; hence global fixed thresholds do not obtain good results. Therefore, the average intensity of each slice must be considered when setting the threshold. On the other hand, the local variation in intensity depends on the area of the slice. Thus, the intensity threshold also depends on the value of neighbouring pixels of the seed.

In order to compute the intensity threshold for each slice, a linear regression model has been used. The threshold for a slice x is calculated as

$$T(x) = M(x)a + L(x)b + c$$

where $M(x)$ is the average intensity of the slice and $L(x)$ is the average intensity of the neighbouring pixels of the current seed. To construct the regression model, a sample of images has been randomly selected from the same stacks as those used to generate the results. Therefore, these images were segmented using the proposed method with a manually defined threshold in order to train the model. If additional stacks are used, the model should be trained again.

3.2.4 Fragment separation

In some cases, the proximity between fragments and the resolution of the CT images can cause that some fragments are erroneously segmented, as if they were only one. In this case, it is necessary to separate them during the segmentation procedure . To do that, an isolated connected region growing based method is used . This method requires

Algorithm 2 Seed spreading process including the two special cases.

```
for all slices do
    for all seeds do
        count ← 2DRegionGrowing(seed)
        if count < MIN_BONE_SIZE then
            for all neighbourSeeds do
                count ← 2DRegionGrowing(neighbourSeed)
                if count >= MIN_BONE_SIZE then
                    seed ← neighbourSeed
                    subtractSegmentedRegion()
                    break
                end if
            end for
        else
            subtractSegmentedRegion()
        end if
        if count < MIN_BONE_SIZE then
            newSeed ← SearchSeedInThePreviousSlice()
            if newSeed then
                count ← 2DRegionGrowing(newSeed)
            end if
            if count < MIN_BONE_SIZE or not newSeed then
                deleteSeed(seed)
            else
                subtractSegmentedRegion()
                seed ← newSeed
            end if
        end if
        if count >= MIN_BONE_SIZE then
            increment ← (count - preCount) / preCount
            while increment < MIN_INCREMENT do
                subtractAlreadySegmentedRegion()
                newSeed ← SearchSeedInThePreviousSlice()
                if newSeed then
                    localCount ← 2DRegionGrowing(newSeed)
                    if localCount ≥ MIN_BONE_SIZE then
                        addSeed(newSeed)
                        count ← count + localCount
                    end if
                    increment ← (count - preCount)/preCount
                end if
            end while
        end if
    end for
end for
```

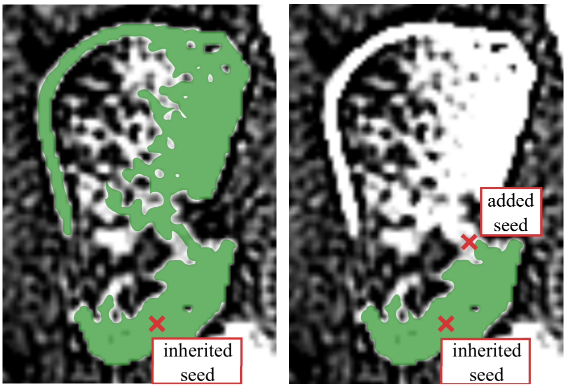

Figure 3.11: Left, two bone fragments are wrongly joined as a result of applying 2D RG. Result of the segmentation using the isolated region growing based method.

minimal user interaction, because the user only has to add an additional seed inside the fractured area (Figure 3.11). The other input of the algorithm is the seed inherited from the previous slice. Once both seeds are determined, the region grows to be connected to the inherited seed and not be connected to the seed added afterwards. For that purpose, the filter uses a binary search to find an intensity threshold that separates both seeds. As it is a semi-automatic procedure, it can be repeated until getting the desired result.

Not always there exists a threshold value that enables the separation of wrongly joined fragments. For example, when two fragments are actually connected by their cortical areas. For these situations, an alternative method has been developed. This method works in all cases at the cost of losing some information about fracture zones. To that end, the region segmented in the previous slice using the current seed is recovered, and its external contour is calculated (Figure 3.12, a). Then, pixels in the current image that are inside that contour and have a value above the intensity threshold form the provisionally segmented region (Figure 3.12, b). Due to this procedure, some pixels in the boundary of the fragment could not be segmented because they are outside the contour. In order to recover these pixels, re-segmentation is performed using 2D RG for each bone fragment, but before that, the result of the previous provisional segmentation is subtracted from the original image (Figure 3.12, c). This avoids over-segmentation and allows recovering lost pixels in the boundary of the fragment. The only drawback of this method is that it does not accurately determine the fracture line, but this line is difficult to be defined even manually.

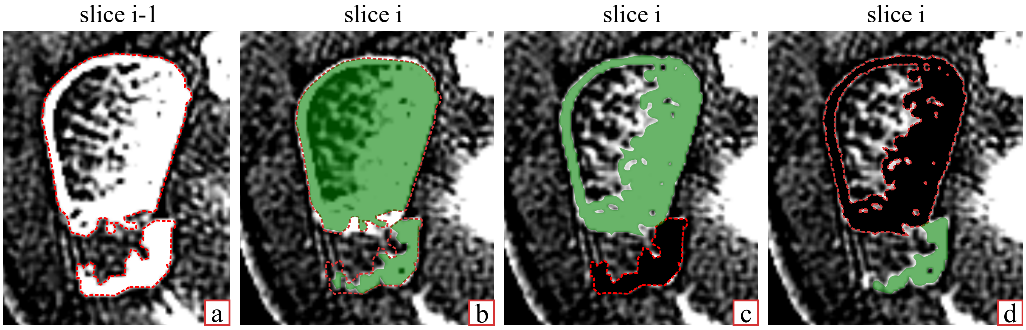

Figure 3.12: From left to right: contours extracted from the previous slice; provisional segmentation result obtained by calculating the inclusion in the contours; top fragment segmented using 2D RG after subtracting the other segmented fragment to the original slice; same procedure repeated with the bottom fragment.

3.3 Results

3.3.1 Comparison with other methods

The developed approach has been compared with some methods commonly used to segment both healthy and fractured bone from CT images. These methods are thresholding [89][110], 3D RG [51][25] and graph cuts [10][64]. The Insight Segmentation and Registration Toolkit (ITK) [109] has been used in order to apply thresholding and 3D RG to the test cases. The graph cuts segmentation has been carried out by using the system presented in [20]. In the case of the methods that require defining an intensity threshold (thresholding, 3D RG and our approach), it has been calculated using the method proposed in this paper. Thus, all the compared methods used the same intensity threshold.

First, all methods have been used to perform the segmentation of healthy bone. To test the robustness of the method, slices with different cortical and trabecular tissue distribution have been considered. That is, slices in which the cortical zone is much more intense that the trabecular area, and slices in which the cortex is thin and irregular (Figure 3.13). Although we focus on the results obtained by the tested methods in certain slices, methods were applied to the entire stack, because the final goal is to segment complete fragments.

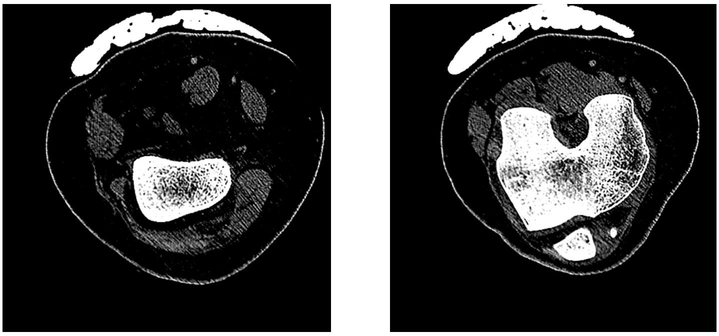

Figure 3.13: Original slices containing healthy bone used as input for tested methods. Left, a slice representing a femur diaphysis. Right, a slice representing a femur epiphysis.

As can be seen in Figure 3.14, all the compared methods obtained acceptable results in the segmentation of healthy bone . Thresholding was able to segment more trabecular tissue than the rest of the approaches and it did not require user interaction. However, this method generated a lot of noise that should be removed after the segmentation process. In addition, as it is not a region based method, it was not capable to segment a single region. This also caused that artefacts were also segmented, because they usually have an intensity value similar to cortical tissue. 3D RG obtained good results in healthy bone segmentation. As it is a region based method, it was able to segment only the regions defined by the placed seeds and it did not produce noise or artefacts. This method required that the user placed a seed in a slice in which the bone appeared. The method that required more user interaction was graph cuts . In order to segment a particular bone, it was necessary to make some scribbles to define the background and the foreground. This allowed segmenting an individual bone. However, graph cuts generated noise around the bone that should be removed after the segmentation process. The proposed approach obtained good results in the segmentation of healthy bone. It only required that the user placed a seed in the first slice where the bone appeared. As 3D RG and graph cuts, our method was able to only segment the target bone (Figure 3.14).

In a second stage, all the compared methods have been tested to segment fractured bone. To do that, different types of fracture cases have been used. These cases include fractures in which the fragments are completely separated and cases in which they are not (Figure 3.15).

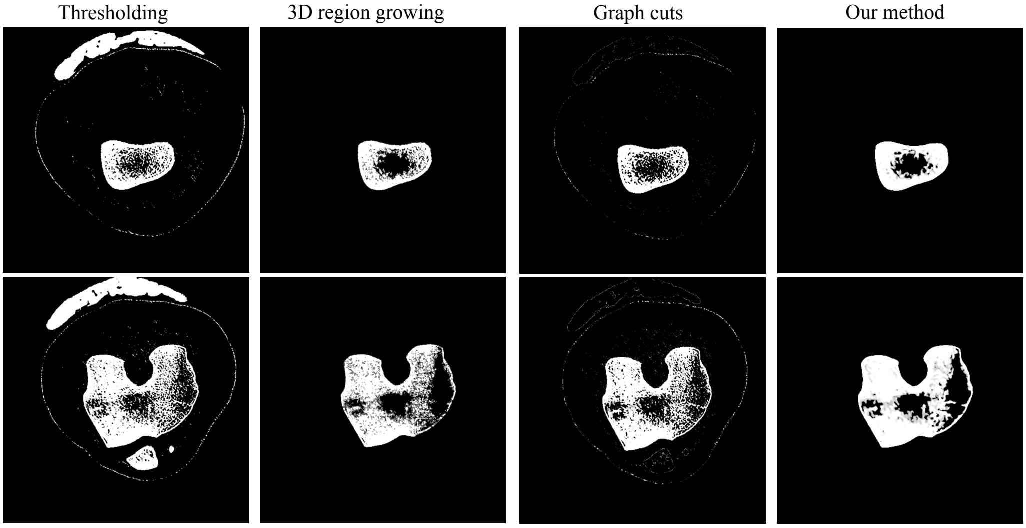

Figure 3.14: Results obtained by the tested methods when segmenting healthy bone. In the top row, a slice in which there is more cortical tissue is used as input. In the bottom row, the slice used as input has more trabecular tissue and therefore it is more heterogeneous.

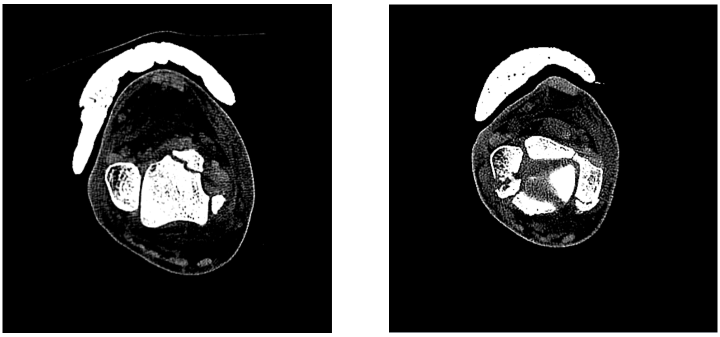

Figure 3.15: Original slices containing fractured bone used as input for testing methods. Left, a slice representing a fracture of talus. Right, a slice representing a fracture of fibula.

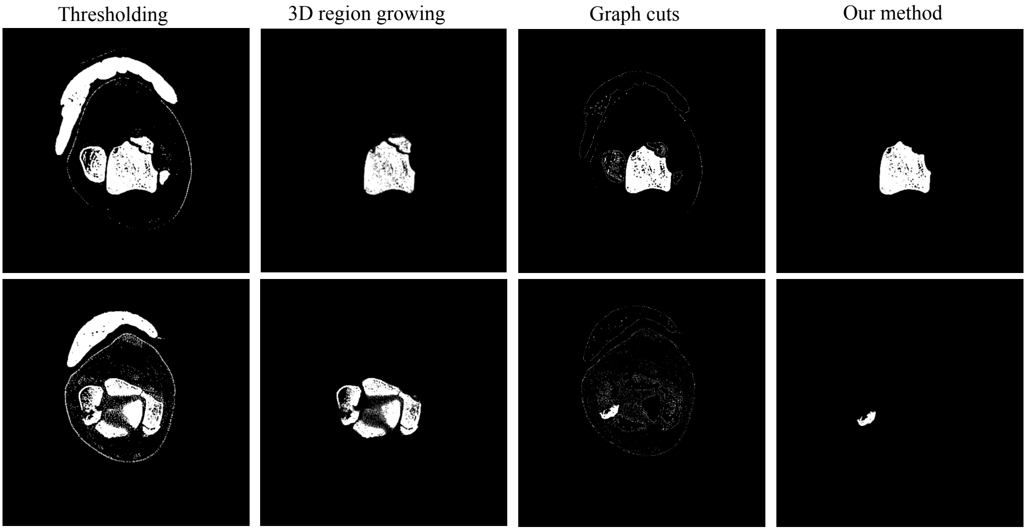

Figure 3.16: Results obtained by the tested methods when trying to segment a single bone fragment in two different situations: the target fragment is almost separated (top); the target fragment is connected to other fragments (bottom). Although our method could obtain separately all the fragments, only the successful segmentation of a single fragment is shown.

The tested approaches obtained different results when trying to segment fractured bone (Figure 3.16). Thresholding technique was not suitable to segment bone fragments. As in the case of healthy bone, this method generated a lot of noise and it was not able to discard artefacts. Moreover, it was not capable to split up neither close nor separated bone fragments. 3D RG did not correctly deal with bone fragments in most cases. If two bone fragments are connected by one segmented pixel in one of the slices, the method will cause overgrowing and will segment both fragments. Graph cuts was able to segment and separate close bone fragments, but it required a lot of user interaction in each slice to be separated. In addition, it required that the image was processed after the segmentation in order to remove noise data. Our approach allowed segmenting and separating bone fragments. For that purpose, it was only necessary to place an additional seed if bone fragments were joined. Summarizing, only graph cuts and the proposed approach segmented bone fragments successfully. Nevertheless, our approach required less user interaction and did not generate noise data, thus post-processing was not necessary.

3.3.2 Application to real pathologies

Additional tests were performed with image stacks containing different fracture types concluding that the proposed method obtained satisfactory results in all tested cases. Specifically, our method has been tested with 4 different bone or fracture types: simple fractures of large and separated bones, simple fractures of small and close bones, long bone fractures and comminuted fractures. In the following paragraphs, an example of each type of fracture to which the method has been applied is explained in more detail. In all cases, the dimensions of the slices are 512x512. The size of the slice and the separation between slices are different for each stack.

To identify and label bone fragments, one or two seeds were placed in the first slice in which each fragment appeared. Then the proposed algorithm was executed. The result of segmenting each slice was displayed in order to verify it and check whether it was necessary to separate joined bone fragments. In that case, a new seed was placed in the fractured zone and the isolated region growing algorithm was executed. This process could be repeated until the desired results were obtained. If this procedure did not perform well, the intersection based method was utilized. In order to visualize the segmentation results, a volume visualization was provided. To show the different labelled fragments, the volume was converted to a point cloud and a different colour was assigned to the points of each fragment. The average time needed to identify a bone fragment was about 5 minutes. No extensive tests were performed to measure time since it was not the goal of this study. Our aim is to reduce user interaction in the segmentation of fractured bones because manual interaction usually takes a long time. On the other hand, the obtained results have shown that the threshold calculation approach works correctly in approximately 95% of the cases.

Table 3.1 summarizes the parameters of all the CT image stacks used as input for experiments. The first image stack represents a simple patella fracture (Figure 3.17 a) and contains 157 slices. The dimensions of each slice are 156x156mm and the separation between slices is 0.625mm. Apart from the patella and its fragments, femur, tibia and fibula have also been segmented. Therefore, this is a simple fracture in which the bones are separated. In total, six different bone fragments were segmented and labelled. It was not necessary to separate wrongly joined bone fragments during the segmentation process. The second CT stack contains a broken thumb phalanx (Figure 3.17 b) and is

Table 3.1: Parameters of the CT stacks used as input for experiments.

	# slices	Size of each slice (mm)	Resolution of slices (px)	Spacing between slices (mm)
Patella	157	156x156	512x512	0.625
Thumb phalanx	193	162x162	512x512	0.625
Fibula	173	159x159	512x512	0.625
Radius	134	96x96	512x512	1.25

formed by 193 slices with dimensions 162x162mm and with 0.625mm spacing. In this case, the fracture generated two small fragments. This is a simple fracture in which the bones are close together. In total, 5 different bone regions were segmented and labelled. As in the previous test, it was not necessary to separate fragments during the segmentation process. The third example shows the segmentation of a fractured fibula (Figure 3.17 c) from 173 CT slices; hence its represents a long bone fracture. The dimensions of each slice are 159x159mm and the separation between slices is 0.625mm. According to its fracture line, the fracture type is spiral and the two generated fragments are nearly touching. Therefore, it was necessary to separate wrongly joined fragments in 4 slices. Isolated region growing based method was applied in 2 cases; hence two additional seeds were required. The other two slices were separated using the intersection based method. The proposed method was able to segment the two fragments individually without post-processing. The last image stack represents a comminuted fracture of radius (Figure 3.17 d) and contains 134 slices. The separation between slices is 1.25mm and the dimensions of each slice are 96x96mm. The fracture was near the joint, thus it made more difficult its segmentation. It was necessary to separate wrongly joined fragments in 7 slices. Isolated region growing was applied in 6 cases and the intersection based method in only one case. As in the previous test, post-processing was not necessary.

3.4 Conclusions

In this chapter, a new method to individually segment and label bone fragments from CT images has been described. Moreover, the main issues to be considered when identifying both healthy and fractured bone tissues have been described. The special cases that arise

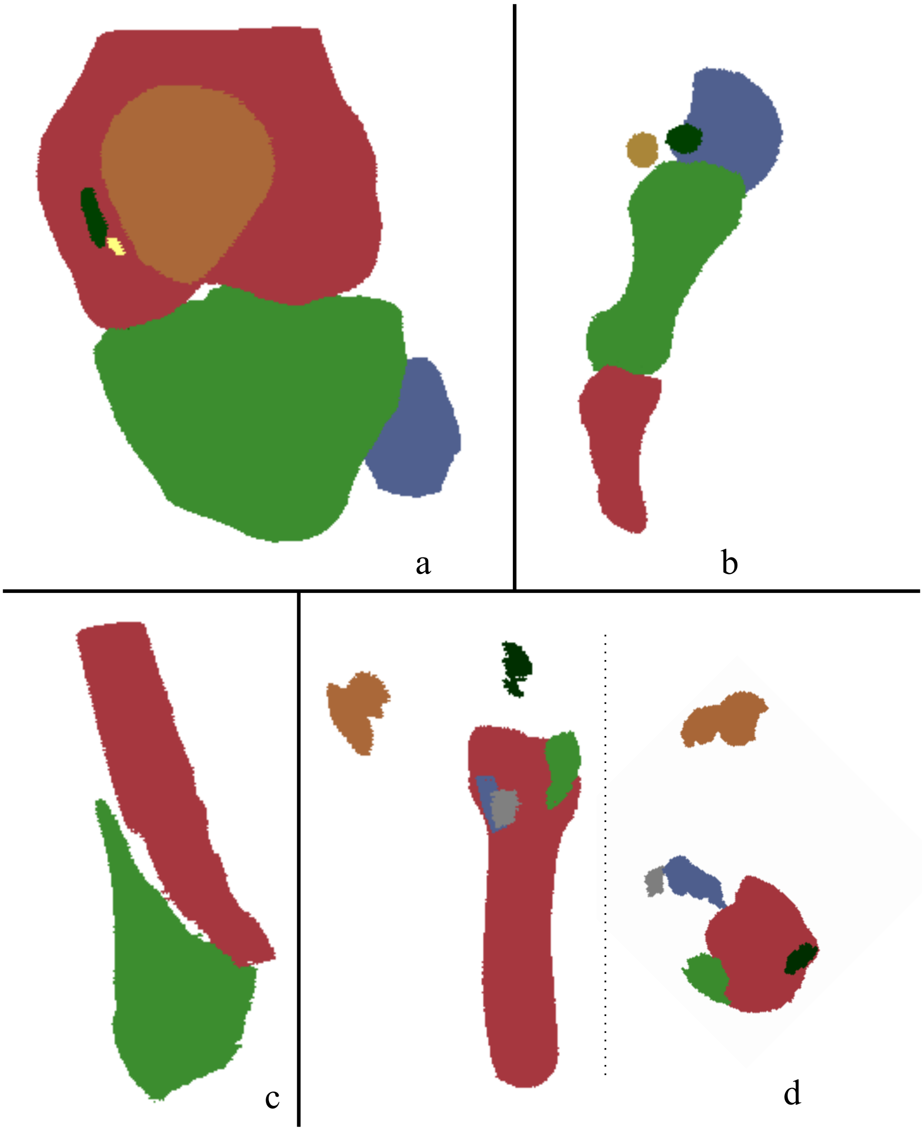

Figure 3.17: Point clouds generated from the result of applying the presented method to segment and label bone fragments from different CT stacks.

in the identification of each type of fracture have also been highlighted. This analysis has shown that the identification of fractured bone may require performing additional steps: labelling bone fragments and splitting wrongly joined fragments. The presented method allows identifying single bone fragments because it is able to separate wrongly joined fragments during the segmentation process. For that purpose, minimal user interaction is required. The method has been compared with currently used methods to identify healthy and fractured bone obtaining better results in most aspects. In addition, the method has also been tested with different clinical cases and it performs well in a variety of bones and fracture types. The work carried out in this chapter fulfils the objectives contemplated in this thesis about developing new methods for segmenting and labelling bone fragments from CT scans, as well as analysing and evaluating the obtained results. In the future, the method will be applied to identify bone fragments in more fracture cases, in order to identify all the bones and fracture types with which it produces good results. As will be shown in Chapter 4, the results of fractured bone identification can be used as input for computer-assisted fracture reduction.

CHAPTER 4

Identification of fracture zones and its application to automatic bone fracture reduction

The reduction of complex bone fractures requires solving a 3D puzzle in order to place each fragment into its proper position. Since the relationship of contact between bone fragments is usually MxN in comminuted fractures, additional processing is necessary to determine which part of the fracture zone is shared with each fragment. Depending on the complexity of the fracture, the surgery could last several hours. The previous identification of the bone fragments, their matching, and their correct positioning could help specialists to plan the surgery and thus to reduce the intervention time [95].

The main contribution of this chapter is an automatic algorithm to calculate the contact zone between two fragments. The calculation of the contact zone between every pair of fragments is useful for the pre-operative planning of a bone fracture reduction. On the one hand, the fracture reduction may be computed by matching and registering contact zones. On the other hand, contact zones provide additional information to better understand the fracture. The contact zone may be defined as the area of the fracture zone that is shared between two fragments. Contrary to what happens with complete fracture zones, the calculated contact zones ease the use of puzzle-solving methods to compute the fracture reduction, because the correspondence between fracture zones is also implicitly calculated. The process of calculating contact zones can be summarized as follows. Firstly, bone tissue is segmented from a CT stack, and each bone fragment is identified. In order to segment and label bone fragments from CT images, the method described in Chapter 3 is used. The results of the segmentation are processed to extract a point cloud for each fragment. Since the developed algorithm

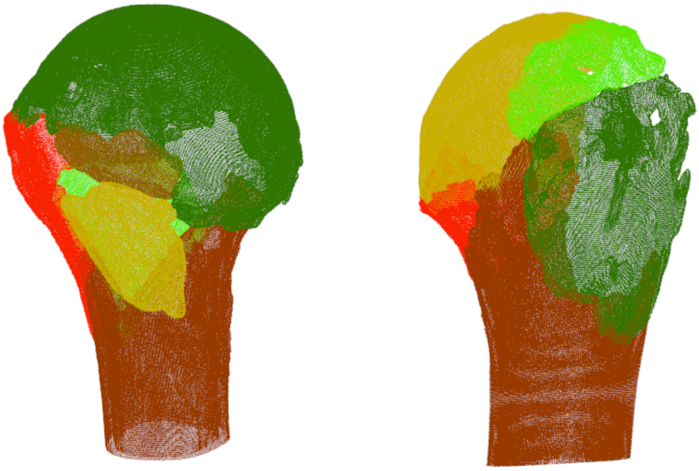

Figure 4.1: Results obtained by applying the proposed method to two of the cadaver cases provided by [30].

only requires the topological information of the contours, it is not necessary to generate 3D meshes representing bone fragments. Then the contact zone between fragments is calculated using the method presented in this chapter. The calculated contact zones can be matched and registered in order to reduce the fracture. For that purpose, a method to calculate the reduction of complex bone fractures using the obtained contact zones is also presented. The proposed procedure obtains good results in the case of moderate fracture displacement (Figure 4.1). If the fragments are too displaced from their original position, a previous coarse alignment of the fragments may be required. With that goal, the contralateral bone can be used as described in the literature [75, 30]. Otherwise, interactive methods can be utilized especially when the contralateral bone is not available [28, 43]. The proposed method has been successfully applied in the reduction of different bone fractures.

4.1 Calculation of the contact zone between fragments

With the aim of easing the application of puzzle-solving methods to reduce the fracture, the presented method does not calculate all the fracture zones of a fragment. Instead, it calculates the contact zone between each pair of fragments; hence the correspondences between fragments are implicitly calculated. The developed algorithm has two main steps. Firstly, a discretized sweep is carried out in order to obtain a set of candidate

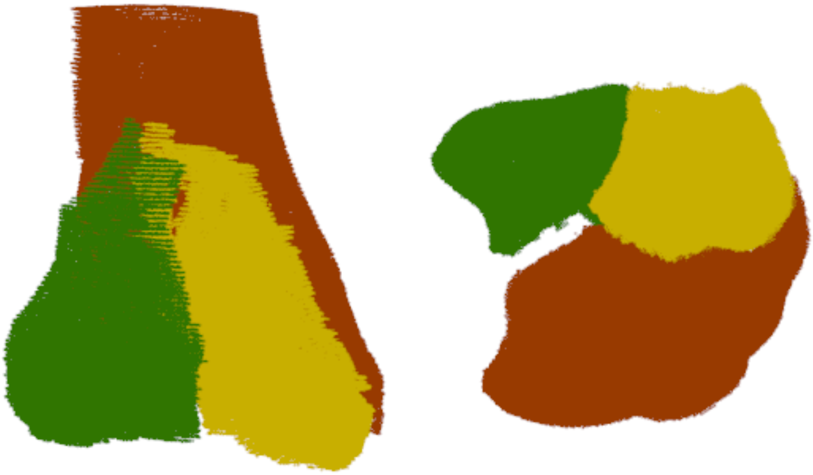

Figure 4.2: Point clouds representing bone fragments identified by the method presented in Chapter 3.

points for belonging to the fracture area. Since not all the candidate points belong to the fracture zone, the set of points is then filtered to get the actual contact zone. For filtering, parameters that have similar values at all points belonging to the fracture zone must be used. In this work, distance to the opposite fragment and curvature have been considered and tested as parameters.

For simplicity, let us consider the calculation of the contact zone between two fragments. Given two bone fragments A and B, the method presented in section 4.1.1 calculates the area of A that should be joined with the area of B to reduce the fracture and vice versa, namely the contact zones CZ_A and CZ_B between the two fragments. Then Section 4.2 shows how to use the calculated contact zones in order to compute the reduction of complex fractures. The input of the algorithm is a pair of point clouds representing each bone fragment (Figure 4.2). These point clouds are calculated from the CT image stack resulting of the identification of bone fragments explained in Chapter 3. Each point represents a voxel whose intensity value is greater than 0. The intensity value of the corresponding voxel in the CT image stack is stored for each point.

4.1.1 Calculating the set of candidate points

The set of candidate points contains most of the points belonging to the contact zone but not all the candidate points belong to the contact zone. Our method carries out a sweep with the purpose of calculating the candidate points of each fragment. The segment that joins the centroids of both fragments defines the direction of the sweep. Since a point cloud is used to represent each bone fragment, the space must be discretized before performing the sweep. Candidate points are those that are reached first by the sweep in each of the spatial divisions. The algorithm is explained in detail in the following paragraphs.

The proposed algorithm performs a sweep to calculate a number of points candidate for belonging to the contact zone between the two fragments. Therefore, the first step of the algorithm consists of the identification of the sweep direction. For that purpose, the centroid of each point cloud must be calculated. Both centroids will be named C_A and C_B respectively. The segment that joins the two centroids defines the direction of the sweep (Figure 4.3, left).

In order to perform the sweep, the volume of each point cloud is discretized using an Oriented Bounding Box (OBB) for each point cloud. For simplicity, only the process to calculate CZ_A is explained. Once the contact zone CZ_A is obtained, the process needs to be entirely repeated changing the sweep direction in order to obtain CZ_B. The only restriction of the OBBs is that one of their associated direction vectors is determined by the segment that joins the centroids of both point clouds. Thus the orientation of the OBB of fragment A will be defined by the basis $(\vec{x}, \vec{y}, \vec{z})$, where $\vec{x}$ is the normalized result of $C_A - C_B$ and $\vec{x}$, $\vec{y}$ and $\vec{z}$ form an orthonormal basis. (Figure 4.3, left). After being calculated, each OBB is homogeneously subdivided to build a grid, so that each voxel of the grid stores all the points that are located inside it. The size of the grid voxels is defined by the distance between slices in the original stack. This assures that a small number of points (no more than two or three) are classified in each voxel of the grid. Afterwards, a discretized sweep is performed using the grid. The direction of the sweep is from C_B to C_A. In each step of the sweep, the content of each grid voxel is studied. If the voxel contains at least a point, all the points classified in that voxel are considered as contact zone candidates and the remaining voxels in that column are not checked. This process is described in Figure 4.3, right. After finishing the sweep, a set of candidate points is obtained. Due to the size of subdivision chosen for the grid, the contact zone obtained practically represents a surface. This will allow the application of a registration algorithm to join the fragments.

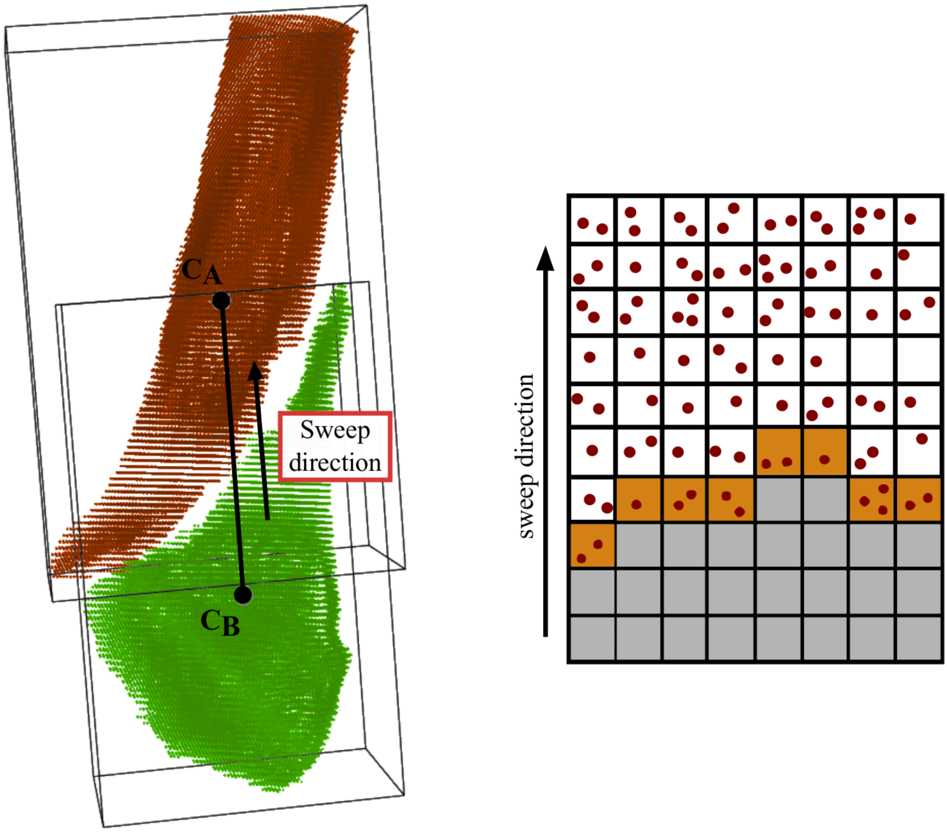

Figure 4.3: Left, OBBs and centroids associated with two fragments. Sweep direction used to calculate the fracture zone of fragment A. The sweep direction for calculating the fracture zone of fragment B is just the opposite. Right, 2D representation of the sweeping procedure. Checked cells that contain no points are displayed in grey. Checked cells that contain candidate points to belong to the contact area are displayed in orange. White cells are not checked.

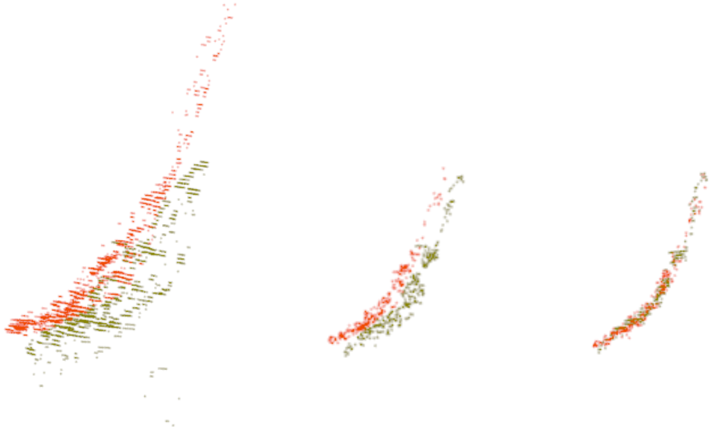

Figure 4.4: Left, set of candidate points of fragments A and B before applying any filter. Centre, fracture zone after the application of the distance-based filter. Right, fracture zones after performing the registration with the aim of joining the two bone fragments.

4.1.2 Filtering the candidate points

The set of candidate points needs to be filtered in order to obtain the actual contact zone of each fragment (Figure 4.4). The revision performed in subsection 2.1.1.3 showed that curvature-based methods are well-known and frequently used to identify the fracture zone of bone fragments. The small number of clinical cases of fractures at our disposal and their variety make impossible the utilization of statistics-based methods. Consequently, two different parameters are considered for filtering contact zones: the distance from each point to the opposite fragment and the estimated curvature at each point.

The first parameter is based on the distance from each point to the other fragment . This value is calculated as the minimal distance between the point and the candidate points of the opposite fragment. Since the contact zone must be close to the fragment to be joined, points whose distance to the other fragment is higher than a pre-defined threshold must be discarded. This threshold depends on each individual case and it is established by experience.

The second parameter is the estimated curvature at each point (Figure 4.5). Unlike the distance-based parameter, the estimation of the curvature requires the normal vector at

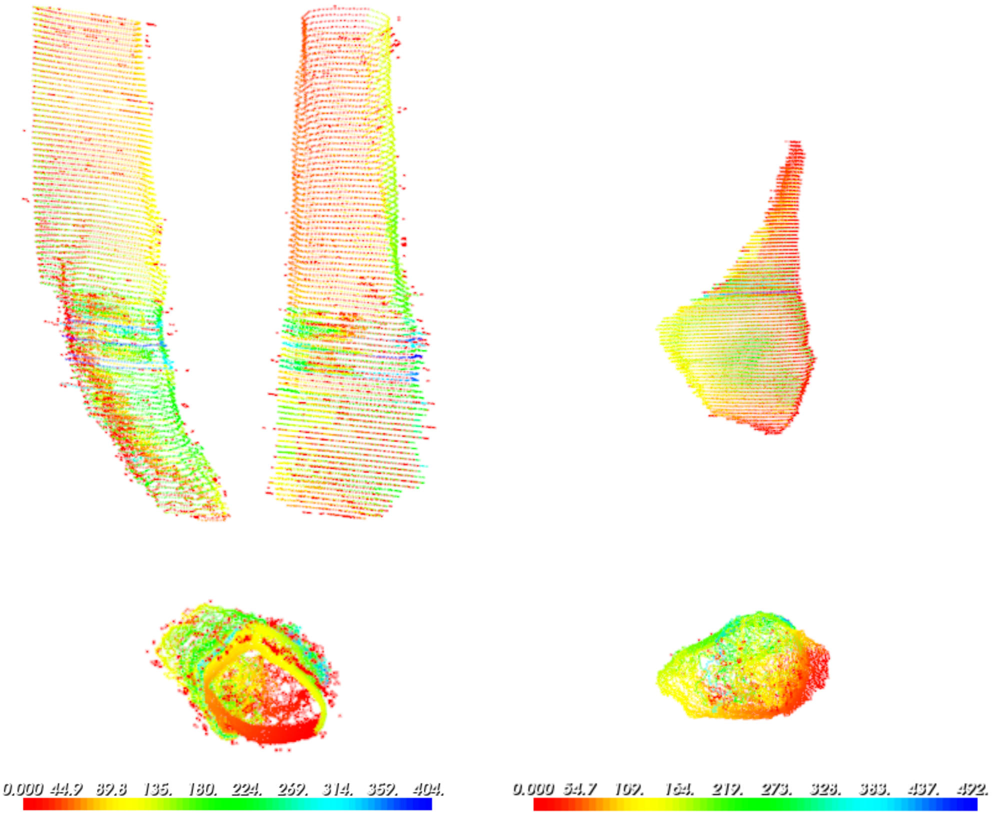

Figure 4.5: Estimated maximum principal curvature of two different point clouds representing bone fragments. High curvature values are shown in blue and low curvature values are shown in red.

each candidate point. After identifying bone fragments with the method described in Chapter 3, external contours are extracted from the segmented regions and the normal at each contour point is estimated. These normals are estimated in 2D for each point using topological information from contours . For each three consecutive points P_{i-1}, P_i and P_{i+1} in a contour, the normal associated to P_i is estimated as the sum of the normals of the two segments formed by $P_{i-1} - P_i$ and $P_i - P_{i+1}$. The third component of each normal is defined by the plane of the slice. Despite of this estimation, the performed tests showed that normals are accurate enough to estimate the surface curvature. Figure 4.6 shows the estimated normals of two different fragments. After the calculation of the normal at each point, the principal curvatures can be estimated. To that end, the implementation available in the Point Cloud Library (PCL) [83] is used. This implementation uses PCA

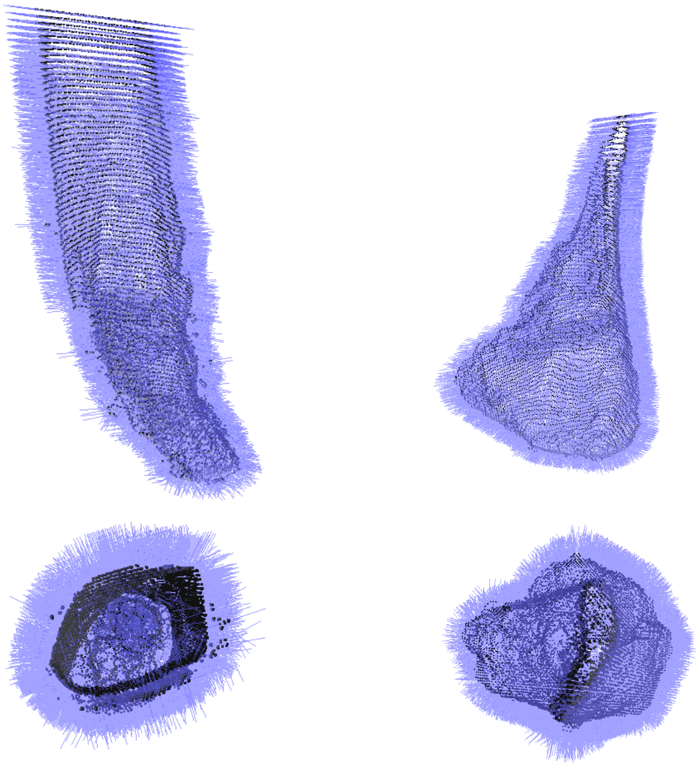

Figure 4.6: Estimated normals of two different point clouds representing bone fragments.

in order to estimate the principal curvatures at each point of the bone fragments. It is expected that contact zone points have a high curvature because the area is very irregular due to the fracture and it is mostly composed by trabecular tissue.

The application of these filters allows discarding points resulting from the sweep algorithm that do not belong to the contact zone. The algorithm proposed for calculating the contact zone between two fragments is depicted in Figure 4.7. The utilization of one parameter or the other for filtering could depend on the bone and the fracture type. In Section 4.4.1, both filters will be evaluated in the reduction of different bone fractures.

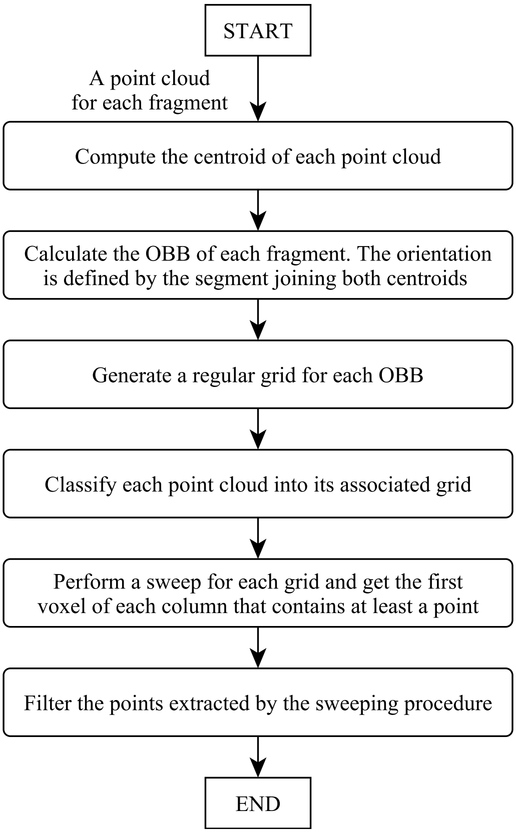
START
A point cloud for each fragment
Compute the centroid of each point cloud
Calculate the OBB of each fragment. The orientation is defined by the segment joining both centroids
Generate a regular grid for each OBB
Classify each point cloud into its associated grid
Perform a sweep for each grid and get the first voxel of each column that contains at least a point
Filter the points extracted by the sweeping procedure
END

Figure 4.7: Flowchart for the automatic calculation of the contact zone between two fragments.

4.2 Matching and aligning multiple bone fragments

The reduction of a bone fracture requires placing all the bone fragments at their original positions. In simple fractures, only two fragments are generated; hence the problem is reduced to align their fracture zones. The contact zones calculated by the method proposed in this chapter enable the application of the ICP algorithm [7] for registering the points belonging to the contact zones. The transformation matrix obtained by executing the ICP algorithm is applied to one of the fragments and as a result the fracture is reduced (Figure 4.4, right).

Fractures with more than two fragments need an additional matching procedure to establish the correspondence between fragments. Since the method described in Section

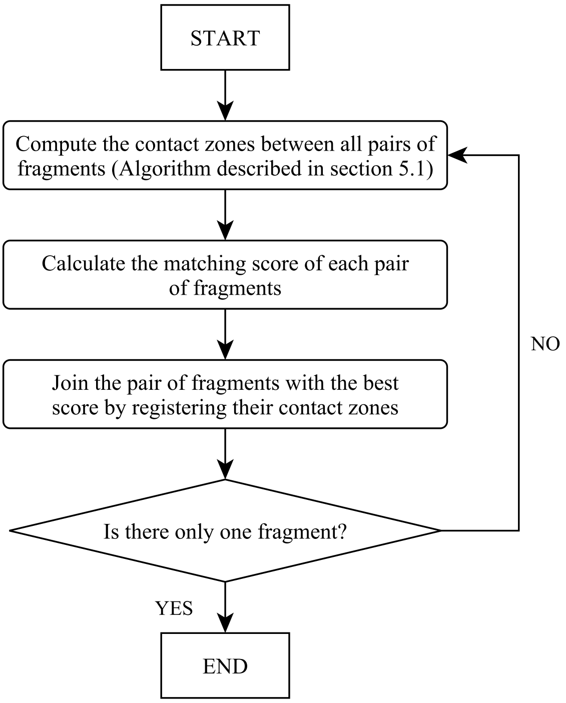
START
Compute the contact zones between all pairs of fragments (Algorithm described in section 5.1)
Calculate the matching score of each pair of fragments
Join the pair of fragments with the best score by registering their contact zones
Is there only one fragment?
NO
YES
END

Figure 4.8: Flowchart for reducing complex bone fractures using the proposed algorithm to calculate contact zones.

4.1 calculates the contact zone between each pair of fragments, bone fragments can be incrementally matched in pairs. In order to establish the order in which each bone fragment pair is joined in the fracture reduction process, a matching score is calculated for each pair of fragments. Thereby, the pair of fragments with the highest score has priority. As the goal is to ensure that the best joining is carried out in each cycle of the loop, the fitness measure obtained by the ICP algorithm is used as score .

Therefore, the first step consists of the identification of the contact area and the calculation of the fitness measure between each pair of fragments. Then the two fragments that obtain the best score are joined. To that end, the transformation matrix obtained by the ICP algorithm is used as in the case of simple fractures. After that, the contact zones between the resulting fragment and each one of the fragments that remain to be repositioned are calculated. The entire process is then repeated until all the fragments have been matched and joined and thus the bone fracture has been reduced. This algorithm is schematized in Figure 4.8 and fully described in Algorithm 3.

Algorithm 3 Fracture reduction with more than two fragments.

```
while numberOfFragments > 1 do
    for each boneFragmentA in boneFragments do
        for each boneFragmentB in boneFragments do
            CZA,CZB ← contactZone(boneFragmentA, boneFragmentB)
            score ← matchingScore(CZA, CZB)
            if score < minScore then
                firstFragment ← boneFragmentA
                secondFragment ← boneFragmentB
                firstCZ ← CZA
                secondCZ ← CZB
            end if
        end for
    end for
    registerContactZones(firstCZ, secondCZ)
    newBoneFragment ← joinBoneFragments(firstFragment, secondFragment)
end while
```

4.3 Dealing with the lack of trabecular tissue

In some cases, cortical tissue cannot be segmented from CT images. Sometimes the absence of trabecular tissue is due to anatomical properties of the bone. Other times the features or the configuration of the CT scanner preclude obtaining this type of tissue. In those cases, the method explained in Section 4.1 fails in the calculation of the contact zone when trabecular tissue is missing (Figure 4.9). In order to overcome this problem, the method has been adapted to deal with only cortical tissue. The required changes are detailed in the following subsections.

4.3.1 Calculating the contact zone

The absence of trabecular tissue makes the segmentation algorithm to practically obtain a surface for each bone fragment. In these situations, the segment defined by the centroid of two adjacent fragments could not establish an acceptable direction to perform the sweep (Figure 4.9, left). This occurs specially in locations close to joints, where the curvature of the bone surface tends to be high. With the purpose of obtaining a more accurate direction to perform the sweep, the presented algorithm only consider the points of each fragment that are close to the other. Thus the points of the fragments are filtered before calculating the centroids in order to obtain representative points. Given two fragments A and B, the representative points of fragment A are defined as $RP_A =$

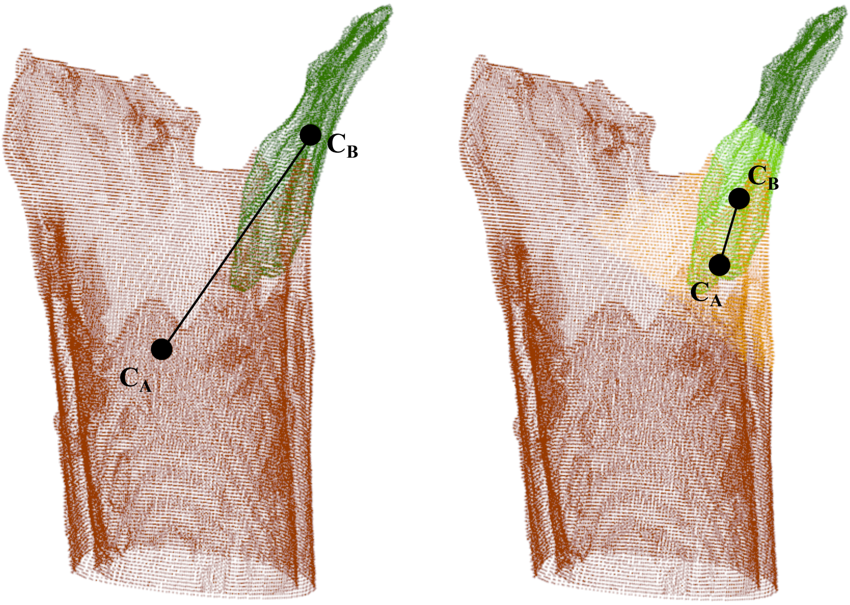

Figure 4.9: Left, centroids and sweep direction calculated using the general method explained in Section 4.1. Right, centroids and sweep direction calculated considering only representative points - These points are displayed lighter.

$\bigcup_{i=1}^{k} A_i \mid d(A_i, b) < \lambda$, k being the number of points in fragment A, d the Euclidean distance, b any point in B and λ a predefined threshold. The representative points of fragment B are similarly calculated. Therefore, only the representative points are used to calculate both centroids C_A and C_B. After that, the procedure explained in Section 4.1 for calculating contact zones is followed but only considering the representative points of both fragments (Figure 4.9, right).

4.3.2 Matching and alignment

Due to the fact that bone fragments are only composed of cortical tissue, the calculated fracture zones are almost lines. As the registration method loses accuracy when registering two lines in 3D, it needs to be adapted to improve accuracy. Once the contact zones between fragments have been calculated and the matching algorithm has determined which two fragments must be aligned , additional auxiliary points are extracted from the rest of fragments to ameliorate the registration. For that purpose, auxiliary points are obtained by using the method described in Section 4.3.1 for calculating candidate points between the fragment to be joined and all its adjacent fragments. Auxiliary points are composed of all these new candidate points whose distance to the fragment to be joined is smaller than a previously defined threshold. These auxiliary points are used as additional fracture zone points to improve the registration. The procedure is summarized in Algorithm 4.

Algorithm 4 Matching and registration of cortical fragments.

```
while numberOfFragments > 1 do
    for each boneFragmentA in boneFragments do
        for each boneFragmentB in boneFragments do
            CZA,CZB ← contactZone(boneFragmentA, boneFragmentB)
            score ← matchingScore(CZA, CZB)
            if score < minScore then
                firstFragment ← boneFragmentA
                secondFragment ← boneFragmentB
                firstCZ ← CZA
                secondCZ ← CZB
            end if
        end for
    end for
    for each boneFragment in boneFragments do
        CZA,CZB ← contactZone(secondFragment, boneFragment)
        score ← matchingScore(CZA, CZB)
        if score < THRESHOLD then
            firstCZ ← joinContactZone(firstCZ, CZB)
            secondCZ ← joinContactZone(secondCZ, CZA)
        end if
    end for
    registerContactZones(firstCZ, secondCZ)
    newBoneFragment ← joinBoneFragments(firstFragment, secondFragment)
end while
```

In the case of long bone fractures, the shaft of the bone can be used as a basis for reducing the fracture so that the remaining fragments are incrementally joined to the shaft [75, 30]. Therefore, the shaft can be considered as the first fragment and the rest of fragments can be incrementally registered to it. As a consequence, the algorithm is simplified and thus its performance is increased. Algorithm 5 shows the adaptation of the proposed matching method.

Algorithm 5 Matching and registration of cortical fragments using the shaft as base.

```
reducedFracture ← shaft;
while numberOfFragments > 1 do
    for each boneFragment in boneFragments do
        CZA,CZB ← contactZone(reducedFracture, boneFragment)
        score ← matchingScore(CZA, CZB)
        if score < minScore then
            nextFragment ← boneFragment
            reducedFractureCZ ← CZA
            nextCZ ← CZB
        end if
    end for
    for each boneFragment in boneFragments do
        CZA,CZB ← contactZone(nextFragment, boneFragment)
        score ← matchingScore(CZA, CZB)
        if score < THRESHOLD then
            reducedFractureCZ ← joinContactZone(firstCZ, CZB)
            nextCZ ← joinContactZone(secondCZ, CZA)
        end if
    end for
    registerContactZones(reducedFractureCZ, nextCZ)
    reducedFracture ← joinBoneFragments(reducedFracture, nextFragment)
end while
```

4.4 Results

The performance of the developed method has been initially evaluated with clinical cases. For that purpose, the contact and overlapping between fragments have been measured. Moreover, the results have been tested on cadaver cases .

4.4.1 Clinical cases

The algorithms described in Sections 4.1 and 4.2 have been successfully applied in the reduction of different bone fractures in the ankle area. Firstly, distance-based and curvature-based filters have been tested with the aim of determining which filter produces better results in the calculation of the contact zone for the available datasets. Subsequently, the performance of the proposed fracture reduction method has been measured. To that aim, the overlapping between fragments has been calculated and the runtime of the different parts of the method has been measured. In all cases, bone fragments have been successfully identified using the method described in Chapter 3.

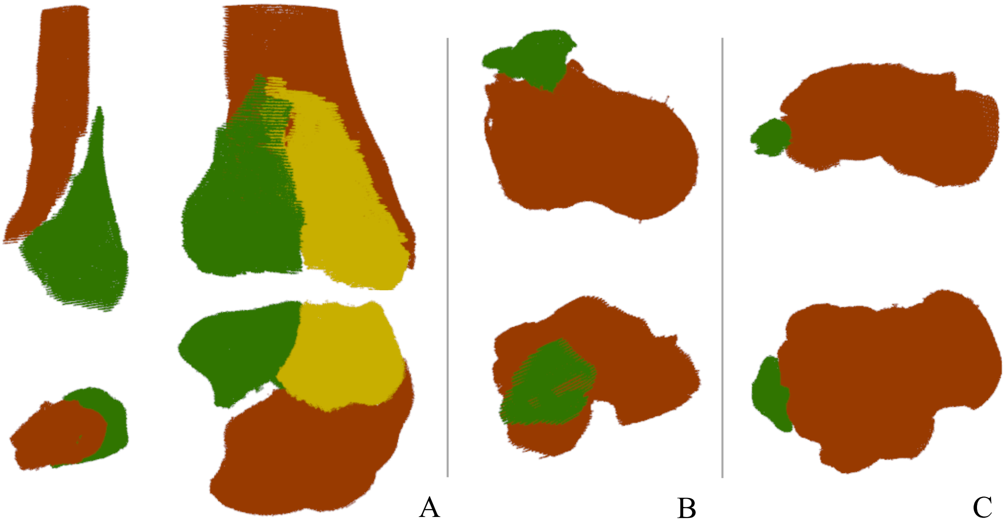

Figure 4.10: Front and top view of the different fracture cases used for experiments. (A) Complex fracture of fibula and tibia, (B) simple fracture of talus and (C) calcaneus. All the bone fragments were identified using the method described in Chapter 3.

The datasets contain three different fractures in the ankle area (Figure 4.10). The method described in Chapter 3 was able to segment and label bone fragments in all tested cases. The first CT stack contains a complex distal fracture of tibia and fibula (Figure 4.10, A), and it is formed by 173 slices with dimensions 159x159mm and with 0.625mm spacing. The fracture generated five bone fragments. The second case is an irregular extra-articular fracture of talus that generated two fragments (Figure 4.10, B). The CT stack contains 156 slices. The dimensions of each slice are 150x150mm and the separation between slices is 0.625mm. The third CT stack represents an extra-articular fracture of calcaneus (Figure 4.10, C) and contains 205 slices. The fracture produced two fragments. The separation between slices is 0.625mm and the dimensions of each slice are 173x173mm. The resolution of the images of the three stacks is 512x512. Table 4.1 summarizes the parameters of all the CT stacks used as input for experiments.

4.4.1.1 Testing different parameter configurations

The two proposed filters were tested for obtaining the contact zone between different bone fragments. The first test uses the curvature filter and the second test uses the distance-based filter . From this point, they will also be called curvature-based and distance-based method respectively. Curvature and distance thresholds were manually

Table 4.1: Parameters of the CT stacks used as input for experiments.

	# slices	Size of each slice (mm)	Resolution of slices (px)	Spacing between slices (mm)
Fibula and tibia	173	159x159	512x512	0.625
Talus	156	150x150	512x512	0.625
Calcaneus	205	173x173	512x512	0.625

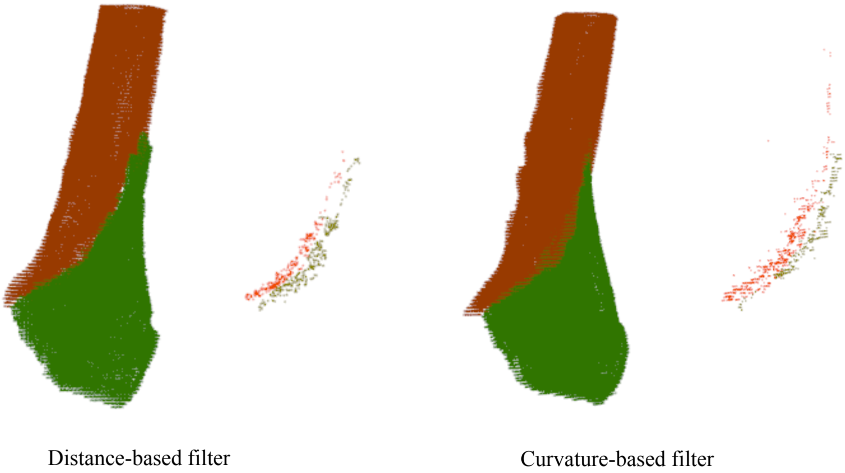

Figure 4.11: Results obtained for the fracture of fibula using the distance-based filter (left) and the curvature-based filter (right), including the calculated fracture zones.

established for each particular case in order to obtain the best result and thus to avoid compromising the comparison between both methods. In both cases, an additional radius outlier removal filter was applied in order to remove isolated points.

For the first fracture, the curvature-based method did not perform well because some points in the joint area have similar curvature values than the points in the fracture zone. By using the distance-based method, the contact zones were correctly calculated (Figure 4.12, A). Figure 4.11 (left) shows that the calculated contact zones enabled a proper reduction of the fracture of fibula using the distance-based method, but not using the curvature-based method 4.11 (right). The fracture of tibia required applying the matching algorithm in order to establish the two bone fragments to be joined in the first place.

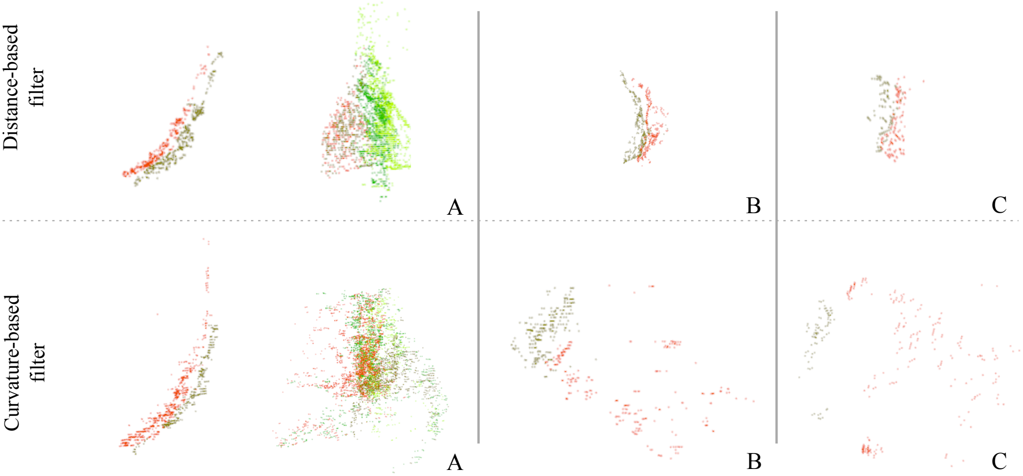

Figure 4.12: Fracture zones obtained by applying the distance-based filter (top) and the curvature-based filter (bottom). (A) Complex fracture of fibula and tibia, (B) simple fracture of talus and (C) calcaneus.

After joining the first pair of fragments, the contact zone of the resulting fragment was calculated with respect to the remaining fragments. Finally the fracture was completely reduced by registering these two new contact zones.

Due to the irregular shape of the bone in the talus fracture, the curvature filter performed even worse than in the previous case. Many points in the outer part of the bone have an estimated curvature similar to the points in the fracture zone; hence the curvature-based filter obtained a scattered point cloud. Nevertheless, the distance-based filter was able to automatically compute contact zones (Figure 4.12, B). Then the fracture was reduced by registering them.

The irregular shape of the bone in the calcaneus fracture caused that the curvature method also obtained a scattered point cloud. However, the distance-based filter was able to deal with the calculation of the contact zone automatically (Figure 4.12, C). Then the registration algorithm completed the fracture reduction.

Figure 4.13 shows the result of applying the distance-based method to reduce the three tested cases. The tests show that contact zones calculated by the distance-based method enable the application of the proposed matching method to reduce different fractures in

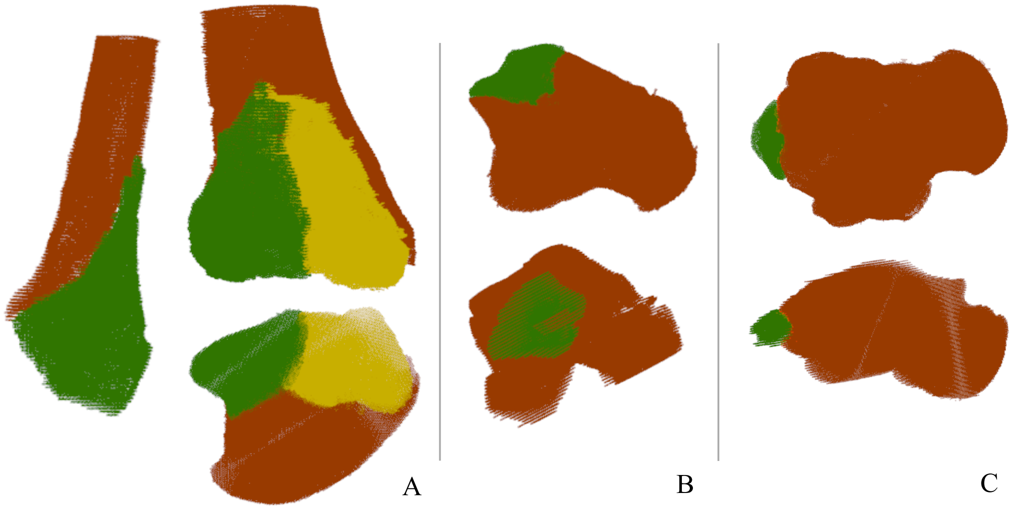

Figure 4.13: Fractures reduced by applying the distance-based method. (A) Complex fracture of fibula and tibia, (B) simple fracture of talus and (C) calcaneus.

the ankle area. On the other hand, the curvature does not seems to be a good parameter to filter fracture points of bone fragments belonging to irregular bones or near joints. In order to obtain acceptable results, the curvature-based method would need user interaction to discard points outside the fracture zone.

4.4.1.2 Performance of the proposed method

The results of the proposed method using the distance-based filter have been quantitatively measured. The quality of curvature-based filter has not been measured, since it obtained bad results in the qualitative tests. The ideal would be to compare the obtained results to the ground truth, but this is not available for our clinical cases. Alternatively, the overlapping between fragments has been considered , since it is a test parameter commonly used in the literature [75, 30]. After performing the fracture reduction process, bone fragments should be in touch. Figure 4.13 shows that there was no visible gap between fragments after the fracture reduction process. Nonetheless, tests have been performed to quantitatively measure that bone fragments were in touch but did not overlap. Since bone fragments were represented by point clouds, they were discretized before calculating the overlapping between fragments. For that purpose, first the Axis-Aligned Bounding Box (AABB) of the resulting point cloud was computed. Then the AABB was subdivided in voxels of the same size as the original voxels of the

Table 4.2: Efficacy of the distance-based fracture reduction process. The overlapping error is defined as the ratio between overlapping and touching voxels.

	#Touching voxels	#Overlapping voxels	Overlapping/Touching ratio (Overlapping error)
Fibula	218	8	0.0367
Tibia	1683	148	0.0879
Talus	210	12	0.0571
Calcaneus	105	4	0.0381

CT image stack. With the goal of enabling subsequent overlap calculations, each voxel referenced points that were located inside and each point stored the bone fragment to which it belonged.

In order to compute the overlapping, all the AABB voxels were processed. Touching voxels were defined as those containing points belonging to two different fragments. Then all the touching voxels were candidates to be overlapping voxels. The calculation of overlapping voxels between two fragments required studying the 26 neighbours of each touching voxel. If there is overlapping, touching voxels are surrounded by other touching voxels. For this calculation, empty voxels were not considered. Therefore, touching voxels were also overlapping voxels if the number of their touching neighbour voxels was greater than $2/3$ of the number of their non-empty neighbour voxels. In the case of the complex fracture of tibia, the process was repeated for each pair of fragments and the obtained overlapping voxels were accumulated. The overlapping error was defined as the ratio between overlapping voxels and touching voxels. Table 4.2 shows the results. The obtained overlapping error for the three tested cases was very small.

Additionally, the efficiency of each step of the algorithm was measured (Table 4.3). The tests were run on a PC with one first generation Intel Core i7 2.8Gz and 4GB RAM. The contact zone was calculated using the distance-based method described in Section 4.1. Once contact zones were calculated, the time to register them was also measured. The table includes all the times each stage was run during the fracture reduction process. In the case of simple fractures, each stage needs to be executed once. For the tibia fracture,

Table 4.3: Efficiency of the distance-based fracture reduction process. The tests were run on a PC with one first generation Intel Core i7 2.8Gz and 4GB RAM. Runtime is given in seconds.

	Calculate fracture zone (s)	#Fracture zone calculations	Register fracture zones (s)	#Fracture zone registrations	Fracture reduction (Total time) (s)	#points
Fibula	3.15	1	0.04	1	3.21	125599
Tibia	18.70	4	0.06	2	74.87	401416
Talus	8.37	1	0.09	1	8.49	519812
Calcaneus	11.39	1	0.08	1	11.51	346087

contact zones were calculated 4 times and the registration process was executed twice. Table 4.3 shows that the fracture reduction process requires a few seconds for simple cases and over a minute for the more complex case. These runtimes should not be an obstacle to its utilization in the pre-operative planning of a bone fracture.

4.4.2 Cadaver experiments

The results of the proposed method have also been compared to the ground truth. In this case, a priori knowledge of the correct final position of each bone fragment enables the measurement of displacement and rotation errors of each fragment . Four different fractures of humerus were provided by Fürnstahl et al. [30]. The four fracture cases contain up to 6 fragments (Table 4.4). Since the provided models only contain the cortical area, the version of the algorithm described in Section 4.3, which only considers cortical tissue, was used. Additionally, bone fragments were initially aligned using the contralateral bone as template in 2 of the 8 experiments: Experiments 1 and 4.

The models were generated using two different scan resolutions (0.67mm and 1mm). This allows testing the performance of the method with different point distributions. Despite the fact that the provided models are polygon meshes, only points were taken into consideration by our method (Figure 4.14). Moreover, the distance-based method was used to calculate the contact zones between fragments. With the goal of evaluating

Table 4.4: Cadaver experiments provided by Fürnstahl et al. [30].

	Dataset	Distance between slices (mm)	# fragments	# points (unfractured mesh)
Experiment 1	Fracture 1	0.675	6	290450
Experiment 2	Fracture 2	0.675	5	503400
Experiment 3	Fracture 3	0.675	3	367690
Experiment 4	Fracture 4	0.675	5	361370
Experiment 5	Fracture 1	1	5	290450
Experiment 6	Fracture 2	1	5	503400
Experiment 7	Fracture 3	1	3	367690
Experiment 8	Fracture 4	1	4	361370

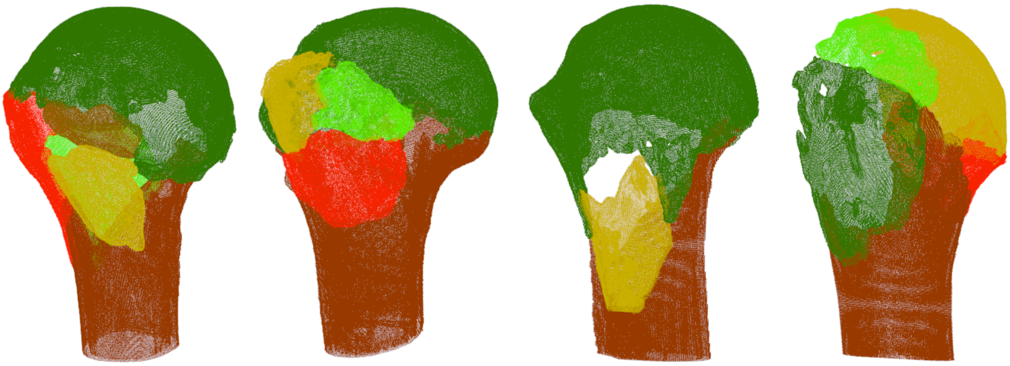

Figure 4.14: Results obtained by applying our method to the four cadaver cases provided by [30].

the accuracy of our developed method, the correct fragment positions estimated by [30] have been compared to the result obtained by our method. Exceptionally, experiment 5 could not be tested because the estimated correct position is unavailable.

With the aim of comparing the results obtained by our method, the tests were based on those already performed by Fürnstahl et al. in [30]. During experiments, translation (mm) and rotation (degrees) errors in the placement each fragment in its correct position

Table 4.5: Results of applying the proposed approach to the cadaver datasets.

	Translation error (mm)	α (degrees)	β (degrees)
Experiment 1	7.2968±2.4586	6.6984±2.5804	6.4492±4.1971
Experiment 2	1.6352±0.7555	7.7439±2.7796	6.6307±3.4122
Experiment 3	1.84±1.2051	6.7343±1.3085	6.782±1.2411
Experiment 4	3.2581±1.7875	6.7635±1.2983	6.5099±1.1554
Experiment 5	-	-	-
Experiment 6	2.4741±0.3095	8.9245±3.9489	8.0785±5.0821
Experiment 7	2.6155±1.0226	4.4828±0.1104	3.0207±2.3492
Experiment 8	3.3667±2.2657	5.6371±2.2053	5.9581±2.4849

were measured (Table 4.5). The translation error was defined as the Euclidean distance between the centroid of the fragment in the calculated position and the ground truth. The rotational errors α and β were calculated based on the fragment's three 2nd moment vectors in calculated and ground truth position. α was computed as the maximum rotational difference around the two largest moment vectors and has a stronger effect on the overall shape of the reconstructed bone. β was calculated as the rotational difference around the smallest 2nd moment vector, quantifying the in-plane rotational error of the fragment on the bone surface.

The obtained results show that the error committed by our method was slightly higher than the error made by the method of Fürnstahl et al. [30]. Nevertheless, our method calculated the fracture area without user interaction. The research of new approaches to improve the registration of fracture lines could reduce the error.

4.5 Conclusions

In this chapter, an algorithm to automatically compute the contact zone between bone fragments is presented. The calculated contact zones enable the automatic reduction of complex fractures. To that end, an automatic method to match bone fragments in complex fractures has also been proposed. In some cases, trabecular tissue cannot be extracted from CT scans because of the anatomical properties of the bone or the configuration of the CT scanner. Moreover, this tissue can be deformed due to the fracture or even small fragments can be discarded by specialists during the fracture reduction process. In order to deal with that situation, the procedure has been adapted to work with only cortical tissue. In the case of displaced fractures, the method could need a prior coarse alignment especially if bone fragments are rotated with respect to their correct position. Interactive tools or template-based approaches may be used to roughly align bone fragments before applying the developed algorithms.

The algorithms described in this chapter have been successfully applied in different fracture cases. The obtained results show that fracture reductions have been successfully computed for all the tested clinical cases, since there is no visible overlapping or gaps between fragments. The small overlapping error demonstrates the absence of visual overlapping in the figures. The calculation of the contact zone only takes a few seconds for the more complex cases. Nevertheless, this calculation could be easily parallelised in multi-core architectures due to the fact that the algorithm is based on the individual processing of each point of the bone fragments. In this way, overall time could be considerably reduced. Furthermore, the method has been evaluated with cadaver experiments. In this case, the correct alignment of the fragments was previously known, thus the obtained results have been compared to the ground truth. The developed method achieved results close to those obtained by currently proposed methods, but without requiring user interaction. Despite these results, a clinical study should be carried out to determine if the developed method would lead to an improvement in the pre-operative planning of bone fracture reduction.

The developed algorithms achieve the objectives contemplated in this thesis, related to facilitating the automation of the computer-assisted fracture reduction process and analysing and evaluating the obtained results. As future research, new automatic algorithms could be developed to perform the prior coarse alignment of bone fragment in complex fractures without the need of user interaction. Additionally, new registering approaches could be studied to improve the accuracy of the computer-assisted bone fracture reduction, especially when only cortical tissue is available.

Chapter 5

Generation of triangle meshes representing fractured bones

The generation of triangle meshes from medical images is a complex task. In addition, the morphology of bone fragments increases the complexity of the problem. As shown in Chapter 2, the shape the segmented regions is very irregular because trabecular tissue appears in the outer part of the bone fragment due to the fracture, making the process more complicated. These features influence the choice of a method for generating meshes.

In this chapter, a study has been carried out in order to test the suitability of mesh generation methods to model bone fragments. Some of the most well-known mesh generation methods have been tested and their main advantages and disadvantages are shown. Specifically, MC [63], Ball-pivoting [6], Poisson surface generation [53] and Algebraic Point Set Surface (APSS) [36, 35] methods have been considered (Figure 5.1). Although there are many methods to generate triangle meshes from point clouds, we chose these four alternatives because they are commonly implemented in libraries and applications to work with data from medical imaging. The obtained results show the performance of each method when applied to different bones and fracture types. In addition, the appropriateness and quality of the results are analysed focusing on the point distribution. Finally, the suitability of each method in the context of visualization and computer-assisted medical procedures is analysed and discussed.

The conducted study shows that current methods are not completely satisfactory in the generation of triangle meshes representing bone fragments to be used in computer-

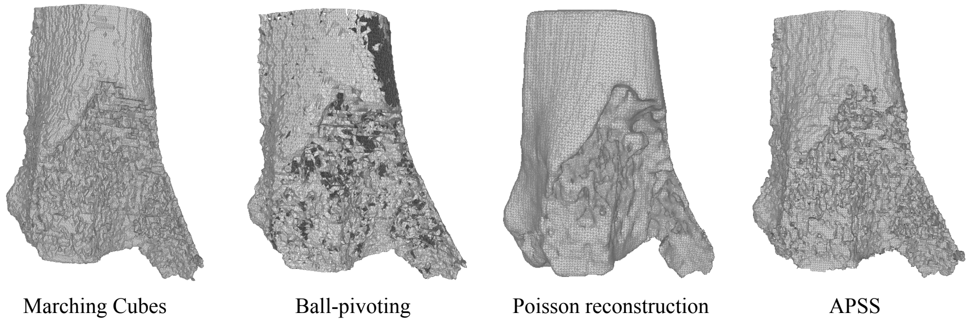

Figure 5.1: From left to right: triangle meshes representing a bone fragment generated using MC from regions, Ball-pivoting, Poisson reconstruction and APSS.

assisted medical procedures. The development of new methods to generate geometrically simple 3D models from CT image stacks that preserve the original information extracted from them would be of great interest. In order to achieve that, a preliminary study for the development of a new method to generate triangle meshes from segmented medical images is presented. The method does not modify the points extracted from CT images, and avoid generating triangles inside the bone. The aim of this initial study is to analyse if a spatial decomposition may help in the process of generating a triangle mesh. The proposed method uses a tetra-tree [47] to address the mesh generation using a divide-and-conquer approach. However, any other spatial decomposition could have been a valid alternative for this initial study. The tetra-tree has been chosen because conducted studies have demonstrated its possibilities in interactive environments [46]. The method is under development and therefore this chapter only presents some initial results and exposes the detected issues to be improved.

5.1 Evaluation of alternatives to generate triangle meshes representing bone fragments from medical images

As stated in Section 2.1.2, nowadays most of the published studies that work with triangle meshes representing bone fragments use MC to generate them [48]. Besides MC, the suitability of three alternative methods to generate triangle meshes to represent bone fragments has been tested in this study: Ball-pivoting, Poisson and APSS. These three methods have been widely used to generate triangle meshes for different applications [12, 104]. Nevertheless, their use is not widespread in the generation of meshes representing bone fragments.

The original MC is an algorithm to generate isosurfaces, which are represented by triangle meshes, from 3D discrete scalar information. The algorithm processes the scalar volume taking a cube composed by eight neighbour locations at each time. A case table is used in order to decide how to generate triangles depending on the value at the eight neighbour locations. The case table is generalized to 15 families by rotations and symmetries. The original version of the algorithm can lead to incorrect triangles because of ambiguous cases, and therefore different approaches have been proposed to solve this problem [74]. In this chapter, the modification proposed by [69] is used.

The models generated by MC are usually huge and contain noise due to the morphology of trabecular tissue. As an alternative, three additional methods have been considered for this study: Ball-pivoting [6], Poisson surface reconstruction [53] and APSS [36, 35].

Ball-pivoting is a geometric algorithm that triangulates a point cloud by traversing a ball of a specific size over the surface. The algorithm starts with an initial triangle and tests whether the ball is able to reach a point from each triangle edge. If so, the reached point is added to the model. This procedure is repeated until the ball is not able to reach any point from the current edges. This method produces good results if the distance between points is homogeneous. Otherwise some points could not be triangulated and the obtained model would be incomplete.

Poisson surface reconstruction requires as input a set of points and their associated normal vectors. This method formalizes the surface approximation as a Poisson problem. Both the points and normals are interpreted as samples of an indicator function that is 0 outside the model and 1 inside it. In order to approximate the indicator function, trivariant B-splines are used. This algorithm is robust against noise in the input data. As a last step, the MC algorithm is applied to extract a triangle mesh associated with the generated implicit surface.

APSS is based on Moving Least Squares (MLS) for the generation of surfaces from point clouds. The algorithm fits an algebraic sphere rather than a plane. By doing this, the algorithm improves stability in cases in which plane-based MLS fails. As in the previous case, MC is applied to generate explicit geometry.

The previous two surface-based algorithms do not preserve the original points, since they approximate implicit surfaces. In both cases, the amount of generated triangles will depend on the resolution of the discretization performed in order to apply MC. A higher resolution leads to better preservation of the shape of the implicit surface, but also to a larger amount of triangles. In the tests conducted, we prioritize the quality of the models.

5.1.1 Data preparation

The input of all the methods tested as alternative to MC are point clouds representing the outer part of bone fragments. These point clouds must be extracted from the information available in medical images. In this study, contours are generated after the segmentation process and then internal contours are removed in 2D. Finally, normals are estimated using the topological information of the contours. This process is explained in detail in the next paragraphs. Alternatively, MC could be applied to CT image stacks after segmentation, and then point clouds and normals could be extracted in 3D from the generated triangles. Nonetheless, we discarded this approach because MC generates many internal triangles due to trabecular tissue, and it modifies the original points extracted from medical images. The presence of noise inside the bone could be attenuated by using compression algorithms based on connectivity [77]. However, the inner noise is difficult to remove even by using manual tools, because trabecular tissue is connected with cortical tissue and it also appears in the outer part of the bone because of the fracture.

In a first step, all the bone fragments are segmented from CT scans. For that purpose, the segmentation algorithm described in Chapter 3 is used. The algorithm generates a region for each bone fragment in each slice. Then the external contour of each region is extracted. With that aim, an approach similar to the proposed in [80] is applied. First, the Marching Squares (MS) algorithm is utilized in order to generate contours from the segmented regions [40]. This algorithm can be considered as a 2D adaptation of MC. Using a divide-and-conquer approach, the algorithm studies each cell composed of four points in a slice to detect boundaries. The algorithm does not ensure that the generated contours are closed, but this is not relevant in this case since the final goal is to generate point clouds. Due to the noise present in the regions because of trabecular tissue, inner

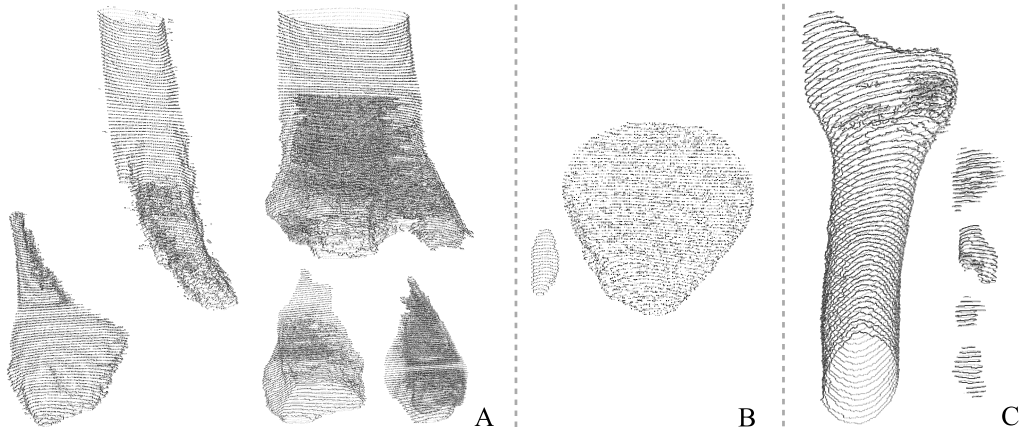

Figure 5.2: Point clouds extracted from CT scans. These points are used as input for the surface reconstruction algorithms. A - Complex distal fracture of tibia and fibula. B - Simple fracture of patella. C - Comminuted fracture of radius.

contours may be obtained. These inner contours are removed using the point-in-polygon algorithm by Feito and Torres [26]; hence only the outer contour of each fragment is extracted in each slice. This algorithm allows determining if a contour is inside other contours without the need to perform complex calculations, such as solving systems of equations. After removing internal contours, the vertices of each external contour are grouped in order to form the point cloud used as input by the mesh generation algorithms (Figure 5.2).

In addition to the outer contours, Poisson and APSS algorithms also require the surface normal vector at each point. In order to estimate normals, the procedure described in Section 4.1.2 is utilized. Although [36] proposed their own approach to estimate normal vectors, we used our own procedure for all the tested methods to avoid compromising the comparison.

Once point clouds have been obtained, mesh generation methods can be executed. The Visualization Toolkit (VTK) [85] has been used with the aim of applying MC to the CT stacks. On the other hand, the three alternative methods have been executed using the MeshLab software [102]. In order to conduct a more complete comparative, the MC algorithm has been applied to both regions and contours. For that purpose, a new image stack was built for each dataset from the geometrical information of the external contours. Once inner contours had been removed, external contours were mapped to an empty CT stack. By doing this, a new stack containing only information of the outer part of each fragment was generated.

5.1.2 Tests run with clinical cases

The methods were used to generate triangle meshes from 5 CT image stacks previously processed to extract the external contour of bone fragments in each slice. These CT stacks contain fragments from 3 different types of fractures: a complex fracture that produced big fragments, three simple fractures that leaded to a big and a small fragment, and a comminuted fracture in which small fragments were generated. Figure 5.2 shows a test case of each of the 3 fracture types. The resolution of all images is 512x512. The results show the performance of each method depending on the features of the CT scans and the type of fracture.

The first CT stack shows a complex distal fracture of fibula and tibia (Figure 5.2, A). The fracture produced five big fragments of irregular size: 2 fragments of fibula and 3 fragments of tibia. The stack has 173 slices with dimensions 159x159mm and 0.625mm spacing. The next three datasets contain simple fractures in which a small and a big fragment were generated. The second stack represents a simple fracture of patella (Figure 5.2, B) and contains 157 slices. The dimensions of each slice are 156x156mm and the separation between slices is 0.625mm. The third stack represents a simple fracture of calcaneus and contains 205 slices. The separation between slices is 0.625mm and the dimensions of each slice are 173x173mm. The fourth stack contains a simple fracture of talus and it is formed by 156 slices with dimensions 150x150mm and with 0.625mm spacing. The last stack shows a comminuted fracture of radius that produced five fragments (Figure 5.2, C). The stack has 134 slices with dimensions 96x96mm and 1.25mm spacing. These data are summarized in Table 5.1.

In the tests conducted, firstly the MC algorithm was used to generate triangle meshes from segmented CT image stacks. The algorithm was applied to the regions obtained by the segmentation method. As before explained, MC was also applied to the external contours of the regions. The MC algorithm used an isovalue of 200 since it was the intensity threshold employed by the segmentation algorithm. Secondly, Ball-pivoting, Poisson reconstruction and APSS were applied to point clouds representing the external

Table 5.1: Point distribution for each CT image stack tested. Resolution of all slices is 512x512.

	Size of slice (mm)	Distance between points (mm)	Spacing between slices (mm)	Ratio
Fibula and tibia	159x159	0.3105	0.625	2.0126
Patella	156x156	0.3047	0.625	2.0513
Calcaneus	173x173	0.3379	0.625	1.8497
Talus	150x150	0.293	0.625	2.1333
Radius	96x96	0.1875	1.25	6.6667

contour of each bone fragment. The radius utilized by the Ball-pivoting algorithm was slightly higher than the distance between slices considering that points in consecutive slices should be triangulated together. The specific value for each case was empirically adjusted. In the case of Poisson reconstruction, an octree level 8 was used to extract the triangle mesh. The performed tests showed that a higher depth level barely modifies the result for the tested cases. The grid resolution used to extract the geometry from the implicit surface obtained by APSS was in the range 50-200 and depended on each test case. As previously mentioned in this section, we prioritized the quality of the meshes over the reduction of the amount of generated geometry when executing Poisson reconstruction and APSS.

Firstly, the size of the triangle meshes obtained by the different tested method has been compared. Figures 5.3 and 5.4 show the size of the obtained meshes. MC generated complex models with a lot of geometry inside the bone due to the trabecular tissue. Despite not including information about the inside of the bone, MC from contours only produced 26.6% less points and 25% less triangles. This is due to the fact that trabecular tissue appeared in the outer part of the bone because of the fracture. Moreover, incomplete models were created when applying MC to contours (Figure 5.5). Ball-pivoting does not modify the original points of the contours. The number of points of the meshes obtained by the Poisson reconstruction method was slightly lower, but the number of triangles was higher in comparison with Ball-pivoting. The number of geometric elements generated by APSS was significantly higher compared to Poisson

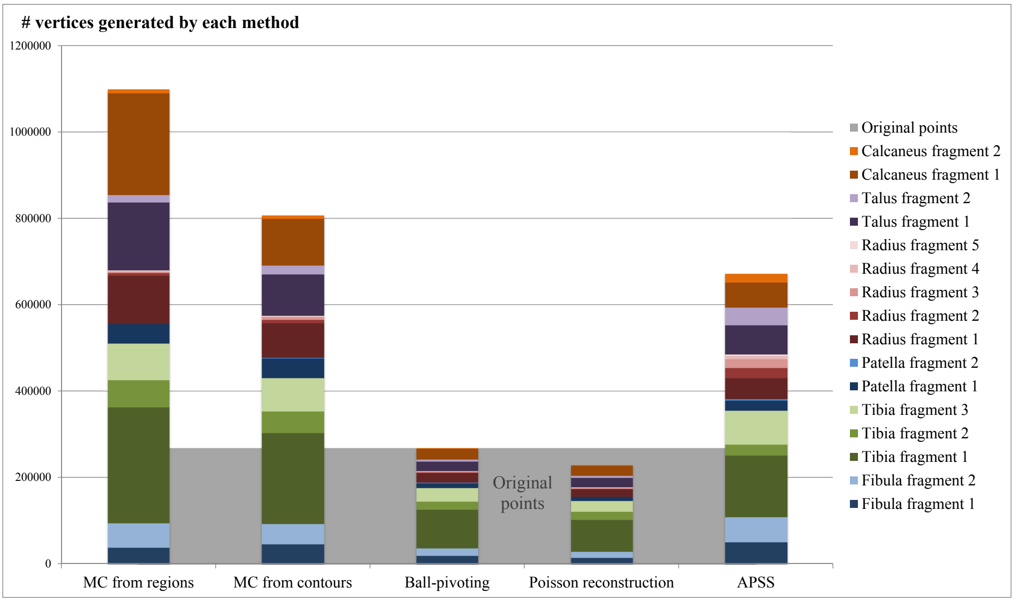

Figure 5.3: Number of vertices generated by applying the tested methods to each dataset. Fragments belonging to the same bone are displayed in the same colour tone. Original points are shown as a grey area.

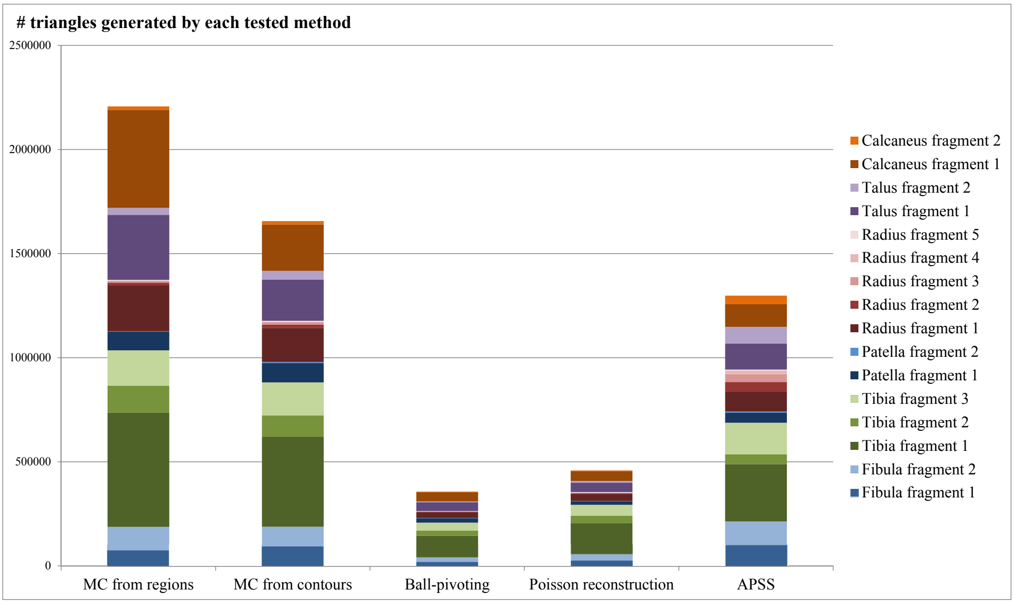

Figure 5.4: Number of triangles generated by applying the tested methods to each dataset. Fragments belonging to the same bone are displayed in the same colour tone.

reconstruction, because the implementation of APSS employs a grid, and Poisson makes use of an octree in order to generate the explicit geometry (Figure 5.6).

Table 5.2 shows the main features of the models generated by applying MC to regions and contours extracted from the test data. Specifically, unreferenced vertices, boundary edges and connected components have been calculated. Additionally, we checked whether the models are two-manifold. The two approaches created complete models since unreferenced vertices were not obtained. In addition, the 87.5% of the models generated by MC from regions and all the models obtained by MC from contours are two-manifold. Nonetheless, MC from regions produced holes and many connected components in some of the models because of the trabecular tissue inside the bone. The use of MC from contours leaded to more boundary edges in spite of not containing inner geometry. This is due to the fact that in some cases the algorithm was not able to connect points belonging to different contours. Moreover, models generated from contours contained several connected components because the fracture causes that trabecular tissue appears in the outer part of the bone and thus models are prone to contain noise in the fracture zone.

Tables 5.3, 5.4 and 5.5 show the characteristics of the models generated by applying Ball-pivoting, Poisson reconstruction and APSS to points clouds representing the outer part of each bone fragment. Ball-pivoting obtained incomplete models since it was not able to triangulate all the points. In addition, models created with Ball-pivoting were not two-manifold, contained many boundary edges, and there were a lot of error in the orientation of the triangles. Figure 5.6 shows mis-oriented triangles in all the meshes generated by Ball-pivoting. Since Ball-pivoting builds models by incrementally adding points to them, only a connected component was obtained. On the other hand, models generated by Poisson reconstruction and APSS do not contain unreferenced vertices, since they are based on implicit surfaces. Poisson reconstruction generated better models from a visual point of view (Figure 5.6). However, this method did not respect the shape of small bone fragments. In addition, the method did not preserve the shape of the bone at the top and the bottom of big fragments. Poisson reconstruction always produces closed models, thus it does not obtain boundary edges. Nonetheless, the method leaded to several connected components, because of the trabecular tissue that appears in the fracture zones. 75% of the models built by Poisson reconstruction are two-manifold.

Table 5.2: Features of the triangle meshes generated by applying MC to regions and contours extracted from each of the datasets.

Fragment	Method	Unreferenced vertices	Boundary edges	Connected components	2-manifold
Fibula 1	MCR	0	350	55	Yes
Fibula 1	MCC	0	280	23	Yes
Fibula 2	MCR	0	56	145	No
Fibula 2	MCC	0	56	48	Yes
Tibia 1	MCR	0	1006	570	Yes
Tibia 1	MCC	0	580	280	Yes
Tibia 2	MCR	0	0	152	Yes
Tibia 2	MCC	0	214	37	Yes
Tibia 3	MCR	0	0	258	Yes
Tibia 3	MCC	0	146	102	Yes
Patella 1	MCR	0	26	158	Yes
Patella 1	MCC	0	262	2	Yes
Patella 2	MCR	0	62	2	Yes
Patella 2	MCC	0	100	1	Yes
Calcaneus 1	MCR	0	322	1949	No
Calcaneus 1	MCC	0	252	12	Yes
Calcaneus 2	MCR	0	124	30	Yes
Calcaneus 2	MCC	0	204	2	Yes
Talus 1	MCR	0	178	751	Yes
Talus 1	MCC	0	332	21	Yes
Talus 2	MCR	0	142	16	Yes
Talus 2	MCC	0	226	2	Yes
Radius 1	MCR	0	824	137	Yes
Radius 1	MCC	0	792	19	Yes
Radius 2	MCR	0	0	8	Yes
Radius 2	MCC	0	72	7	Yes
Radius 3	MCR	0	0	3	Yes
Radius 3	MCC	0	164	1	Yes
Radius 4	MCR	0	0	1	Yes
Radius 4	MCC	0	138	1	Yes
Radius 5	MCR	0	0	3	Yes
Radius 5	MCC	0	72	3	Yes

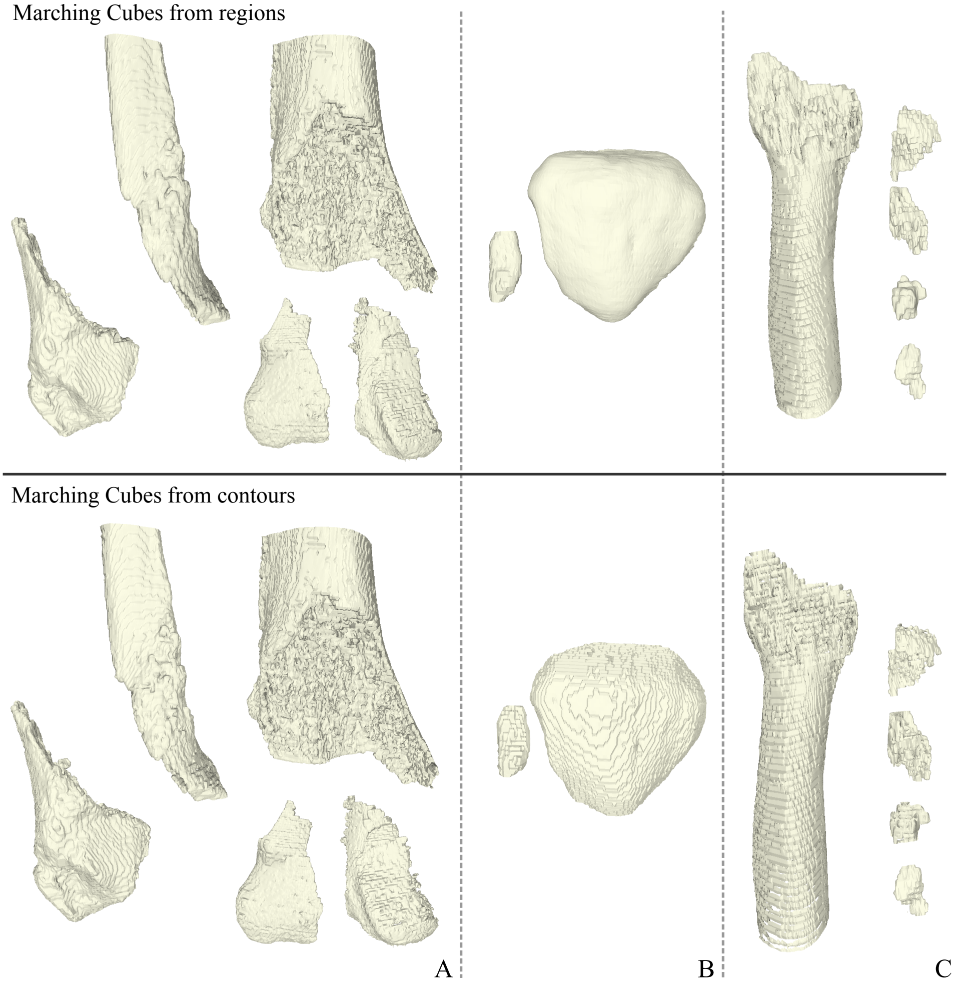

Figure 5.5: Meshes generated using MC from regions and contours. A - Complex distal fracture of tibia and fibula. B - Simple fracture of patella. C - Comminuted fracture of radius.

Table 5.3: Features of the triangle meshes generated by applying Ball-pivoting to contours extracted from each of the datasets.

Bone	# fragment	Unreferenced vertices	Boundary edges	Connected components	2-manifold
Fibula	1	7370	10746	1	No
Fibula	2	5720	8531	1	No
Tibia	1	34568	48247	1	No
Tibia	2	6426	6128	1	No
Tibia	3	10080	14613	1	No
Patella	1	612	2594	1	No
Patella	2	364	440	1	No
Calcaneus	1	1516	7322	1	No
Calcaneus	2	95	1088	1	No
Talus	1	567	10057	1	No
Talus	2	1978	1932	1	No
Radius	1	6821	11273	1	No
Radius	2	814	1064	1	No
Radius	3	635	948	1	No
Radius	4	174	313	1	No
Radius	5	246	375	1	No

APSS produced models similar to those created by MC, but it generated many boundary edges because the algorithm was more sensitive to noise than MC since it built additional triangles using points representing noise even outside the fracture zone. The method obtained only one connected component in all the models and 43.75% of the models generated by APSS were two-manifold.

Additionally, the adequacy and quality of the obtained results have been analysed depending on the features of the CT scans. The distribution of the points generated from a CT image stack depends on the resolution and the size of the 2D images, and the distance between slices. Since the resolution of all the test images is 512x512, the only varying data are the size of the images and the distance between slices. Table 5.1 shows the value of both variables for all the tested CT image stacks. For the first five CT stacks, the distance between slices approximately doubles the distance between points in the same slice. Despite the fact that all the tested algorithms perform better if the points are uniformly distributed, they obtained reasonable good results in most of these five cases.

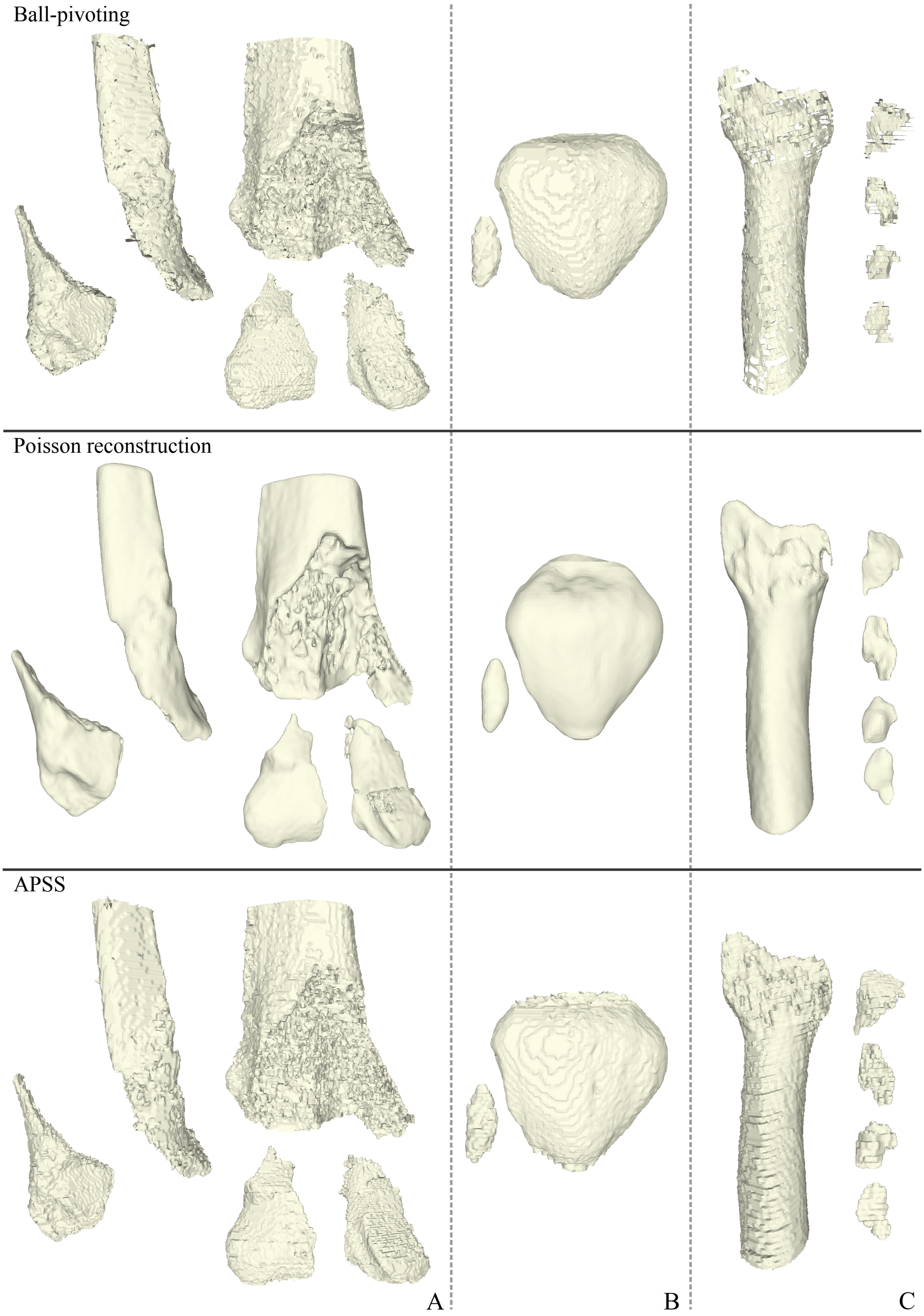

Figure 5.6: Meshes generated using Ball-pivoting, Poisson and APSS. A - Complex distal fracture of tibia and fibula. B - Simple fracture of patella. C - Comminuted fracture of radius.

Table 5.4: Features of the triangle meshes generated by applying Poisson reconstruction to contours extracted from each of the datasets.

Bone	# fragment	Unreferenced vertices	Boundary edges	Connected components	2-manifold
Fibula	1	0	0	3	Yes
Fibula	2	0	0	1	Yes
Tibia	1	0	0	4	No
Tibia	2	0	0	4	No
Tibia	3	0	0	17	No
Patella	1	0	0	1	Yes
Patella	2	0	0	1	Yes
Calcaneus	1	0	0	7	Yes
Calcaneus	2	0	0	1	Yes
Talus	1	0	0	7	Yes
Talus	2	0	0	4	Yes
Radius	1	0	0	3	Yes
Radius	2	0	0	1	No
Radius	3	0	0	1	Yes
Radius	4	0	8	1	Yes
Radius	5	0	0	1	Yes

The distribution of points extracted from the last CT image stack (fracture of radius) is more heterogeneous, because the distance between slices is more than 6 times higher than the distance between points in the same slice. In this situation, the triangle mesh generated by MC from regions showed more discontinuities between slices. Moreover, MC from contours generated more holes than in the other five cases. Ball-pivoting was not able to properly triangulate in most cases; hence the obtained mesh contains many holes and incorrect triangles. Poisson reconstruction performed better than Ball-pivoting. Nevertheless, the algorithm had difficulties in modelling the more detailed areas, such as the fracture zone. APSS generated better results in the fracture areas but the triangle mesh exhibits ripples because of the distance between slices.

5.1.3 Discussion an analysis of the obtained results

Bone fragment models generated from CT scans are expected to have certain features that influence the choice of the method to be used to generate them. These features have

Table 5.5: Features of the triangle meshes generated by applying APSS to contours extracted from each of the datasets.

Bone	# fragment	Unreferenced vertices	Boundary edges	Connected components	2-manifold
Fibula	1	0	2876	1	No
Fibula	2	0	3110	1	No
Tibia	1	0	19691	1	No
Tibia	2	0	5145	1	No
Tibia	3	0	8196	1	No
Patella	1	0	886	1	Yes
Patella	2	0	51	1	Yes
Calcaneus	1	0	7778	1	No
Calcaneus	2	0	2444	1	Yes
Talus	1	0	12415	1	No
Talus	2	0	3030	1	No
Radius	1	0	3538	1	No
Radius	2	0	1283	1	Yes
Radius	3	0	386	1	Yes
Radius	4	0	377	1	Yes
Radius	5	0	63	1	Yes

been translated into criteria so that the best suited method should meet as much criteria as possible.

On the one hand, the first set of criteria is related to the features that a model should have for using it in a computer-assisted medical procedure. One of the most important applications of the generation of bone fragment models from CT images is computer-assisted bone fracture reduction. As stated in Chapter 2, some of the methods proposed to compute a fracture reduction require the calculation of the fracture zone. These methods place bone fragments in their correct position by calculating, matching and registering fracture zones [113, 75, 30]. Therefore, it is important that the reconstruction method does not significantly modify the shape of the fracture zone. The accuracy of the obtained results is a key factor in computer-assisted medical procedures, thus mesh generation methods must avoid modifying the features of the fragments directly extracted from medical images. Because of the presence of trabecular tissue, fracture zones usually contain small details. These methods must keep the shape of the fragments in the more detailed areas. Moreover, they must preserve the shape of small fragments. The criteria associated with the use of the generated models in computer-assisted medical procedures have been summarized as follows:

- C1: Does not modify the original position of the points directly extracted from medical images. Preserves the original features of the fragments.
- C2: Does not have difficulties in modelling the more detailed areas in the fracture zone.
- C3: Keep the shape of small bone fragments.

The second group of criteria is based on the features that a bone fragment model should have in order to use it for visualization purposes. Sometimes, the generated models are not used in the calculations carried out in computer-assisted procedures, and the goal is to display them in the best possible way so that they look like real bone. In these, cases the accuracy with which the reconstruction is performed is not crucial and the main purpose of reconstruction methods is to obtain good-looking models; hence it is desirable to obtain models with smooth surfaces. This can be a challenge if the distance between slices is too big. Furthermore, the generated normal vectors must be well oriented for properly computing the visualization properties of the model. Additionally, the information inside the bone is usually not useful for visualization purposes. Therefore, it is advisable to remove it during the reconstruction process, in order to improve the performance of the visualization. The criteria related to visualization have been summarized as follows:

- C4: Generates correctly oriented normal vectors and well-oriented triangles.
- C5: Generates smooth models from a visual point of view.
- C6: Does not produce geometry inside the bone.

Finally, other criteria have been defined based on the general features that 3D models should have in any context, because these features are valuable regardless the

Table 5.6: Comparison between different mesh generation methods. For each method, we evaluated the degree to which it fulfils the described criteria. A higher number of asterisks means a better fulfilment. Refer to the text for a detailed description of each criterion.

	Computer-assisted procedures			Visualization			General			
Method	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
MC from regions	*	**	**	**			**	**	**	
MC from contours	*	**	*	**			*			
Ball-pivoting	**									**
Poisson				**	**	*	**	**	*	*
APSS	*	**	*	**			**		*	*

application. It is important that the generated models are complete and thus they do not contain holes. This can be a special challenge if the distance between slices is big. Trabecular tissue is prone to generate noise data during segmentation, and therefore, the reconstruction methods must be robust to noise and artefacts. Finally, the reconstructed models must have the simplest geometry without compromising their shape, with the aim of facilitating their use in visualization and computer-assisted procedures. These general criteria have been summarized as follows:

- C7: Does not produce incomplete models, nor models with holes.
- C8: Does not generate noise data or artefacts.
- C9: Properly triangulates when the distance between slices is big.
- C10: Generates models with a reduced amount of polygons.

Table 5.6 shows the evaluation of each method considering the previously defined criteria. We gave 0, 1 or 2 asterisks to each method depending on the grade to which it fulfils each criterion. A higher number of asterisks means a better fulfilment.

Table 5.7: Computer-assisted score (CAS) and visualization score (VS) obtained by each mesh generation method.

Method	CAS	VS
MC from regions	5.5	4
MC from contours	2.5	1.5
Ball-pivoting	2	1
Poisson	3	5.5
APSS	4	3

In order to calculate a score associated with each method, we assigned 0.5 for each asterisk, giving equal emphasis to all criteria. Then the total scores associated with visualization and computer-assisted medical procedures were calculated. Table 5.7 shows the scores obtained by each method. The computer-assisted score (CAS) was calculated as the sum of the scores obtained in the general criteria and in all the criteria related to computer-assisted medical procedures:

$$CAS = C1 + C2 + C3 + C7 + C8 + C9 + C10 \tag{5.1}$$

On the other hand the visualization score (VS) was calculated as the sum of the scores obtained in the general criteria and in all the criteria related to visualization:

$$VS = C4 + C5 + C6 + C7 + C8 + C9 + C10 \tag{5.2}$$

Considering the visualization criteria, Poisson reconstruction obtained the best results since it produced smooth models from a visual point of view and it barely generated geometry inside the models. However, the method significantly modified the original shape of the bone fragments (Figure 5.7). In addition, small details in bone fragments can be removed. This can be a drawback in computer-assisted processes that require accuracy, such as virtual fracture reduction.

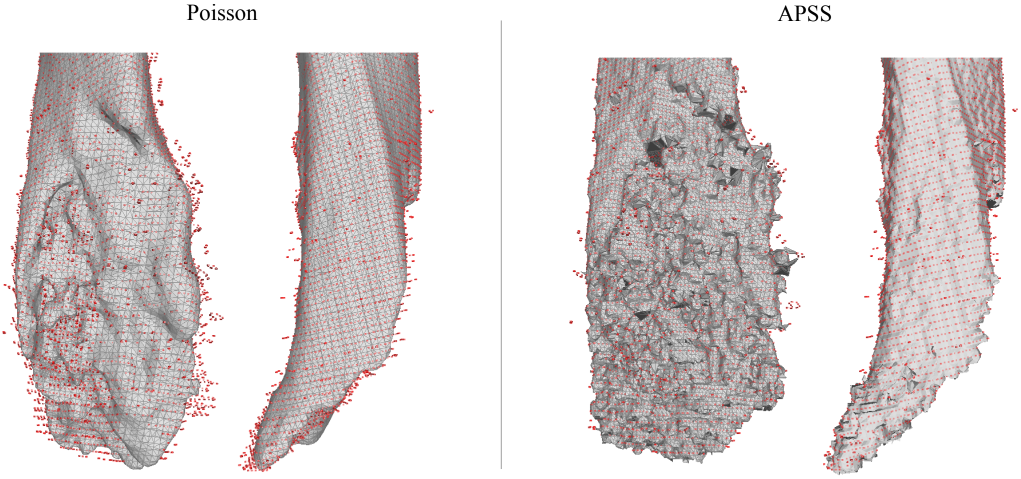

Figure 5.7: Two meshes generated by Poisson reconstruction and APSS representing the same fracture zone. In both cases, original points are in red. Differences in position can be appreciated between the original and the generated points.

Taking into account the criteria associated with computer-assisted medical procedures, MC from regions is the best option, since it slightly modifies the information directly extracted from CT scans and it does not have difficulties in modelling the more detailed areas and preserves the shape of small fragments. The main drawback of the MC algorithm is the large size of the obtained models. Nevertheless, this problem can be attenuated by using simplification methods. In that sense, algorithms based on vertex decimation [86] or progressive meshes [41] are commonly used in the literature. In this way, the total number of triangles can be reduced while preserving an acceptable approximation to the original geometry.

Compared to its application to regions, MC from contours obtains similar results in terms of preservation of the information and amount of geometry, but it produces holes and noise, especially if the distance between slices is big (Figure 5.5).

The main advantage of the Ball-pivoting algorithm is that it does not modify the original position of the points. Nevertheless, the method cannot assure that all the points are used to build the mesh because points usually are seldom uniformly distributed. Consequently, the method generated incomplete models in most of the tested cases (Figure 5.6).

Table 5.8: Advantages and disadvantages of the five approaches tested for the generation of triangle meshes representing bone fragments.

Method	Advantages	Disadvantages
MC from regions	Generates complete models and well-oriented triangles. Slightly modifies the original shape of the fragments.	Generates complex meshes with a lot of geometry inside because of trabecular tissue.
MC from contours	Compared to MC from regions, it reduces the amount of geometry inside the bone.	Generates complex and incomplete models.
Ball-pivoting	Does not modify the original position of the points, and thus preserves the original features of the fragments.	Produces mis-oriented triangles. Does not generate correct triangles in small fragments. Does not properly triangulate if the distance between slices is big.
Poisson	Generates smooth meshes from a visual point of view.	Does not respect the shape of small fragments. Has difficulties in modelling the more detailed areas.
APSS	Slightly modifies the original shape of the fragments.	It is susceptible to noise and thus generates artefacts in some cases. Generates ripples if the distance between slices is big.

APSS obtained visually good results and while slightly modifying the original shape of the bone fragments (Figure 5.7). However, it produced artefacts and ripples in some cases (Figure 5.6). The advantages and disadvantages of each method when generating 3D meshes of fractured bones are summarized in Table 5.8.

5.2 Generation of triangle meshes from medical images using a tetra-tree

The review presented in Section 2.1.2 and the tests described in Section 5.1.2 showed that triangle meshes generated by MC are the best suited to be used in computer-assisted medical procedures. Nevertheless, these models are complex, contain unconnected geometry and modify the data extracted from the CT image stack. Therefore, generated meshes need to be post-processed to improve their features.

In this section, an initial study for the development of a new method to generate triangle meshes from medical images is presented. The method allows generating 3D models using contours directly extracted from CT image stacks. The triangulation of consecutive contours involves the resolution of some complex problems, such us deciding which points should be sewn together in 3D, or managing sewing when the contours are subdivided in consecutive slices (Figure 5.8). The method presented in this section deals with these problems by using a divide-and-conquer approach. For that purpose, the proposed method uses a spatial decomposition named tetra-tree [47] . The choice of the spatial decomposition depends on how it fits the bone fragment; hence any other spatial decomposition might have been used in this preliminary study. The tetra-tree has been chosen because conducted studies demonstrated its suitability for being used in interactive environments [46]. The main purpose of this initial study is to determine if a spatial decomposition may help in the reconstruction of the mesh, allowing the analysis of the advantages and disadvantages of using it. In this section, only initial results are shown and issues to be improved are outlined.

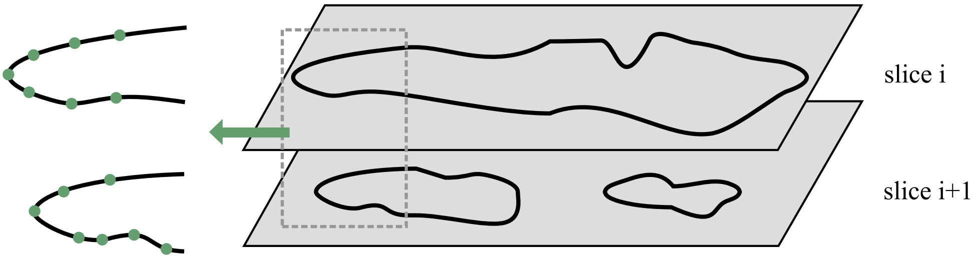

Figure 5.8: Simplified representation of the case in which contours from two consecutive slices do not have a one-to-one correspondence. Left, zoomed area showing the difficulty of deciding which points should be sewed together in 3D.

5.2.1 Classification in the tetra-tree

The generation of triangle meshes from point clouds is a complex problem that requires deciding which points should be triangulated together. The use of spatial decompositions [84] allows dividing the point cloud into small sets of points, reducing the mesh generation problem to smaller sub-problems. A tetra-tree [47] is a spatial decomposition technique that recursively divides the space into tetra-cones with a common origin, so

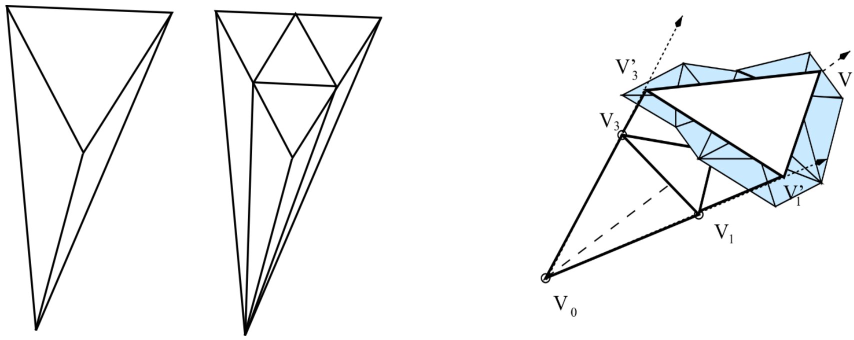

Figure 5.9: Left, a representation of the subdivision of a tetra-cone. Right, a schema that represents the bounding tetrahedra associated with a tetra-cone [49].

that these tetra-cones cover the entire space without overlapping among them. As the tetra-tree is built, the points are classified in the generated tetra-cones.

A tetra-tree is a hierarchical space decomposition defined in the whole space. At its first level, the tetra-tree divides the entire space into eight tetra-cones. A tetra-cone is a space region defined by three planes that intersect in one point. This intersection point usually is the centroid of the model but it could be any other point in 3D space. These eight tetra-cones cover the entire space without overlapping. In the next level, each tetra-cone is homogeneously divided into four new tetra-cones, as shown in Figure 5.9 (left). Each new tetra-cone is subdivided until reaching one of the following conditions:

- The maximum level of subdivisions is achieved. This level is previously defined.
- The tetra-cone contains fewer points than a previously set threshold.

In our proposal, the input of the mesh generation algorithm is a set of contour points generated using the algorithm explained in Section 5.1, and the origin of the tetra-tree is the centroid of these contour points. During the tetra-tree generation, all these points are classified in the hierarchy of tetra-cones. Then the tetra-tree is adapted in order to fit the classified geometric elements. For that purpose, the enveloping tetrahedron associated with each tetra-cone is calculated. The enveloping tetrahedron of a given tetra-cone is the smallest tetrahedron that shares the origin and the lateral faces with that tetra-cone, and contains all the points that are classified into it (see Figure 5.9, right).

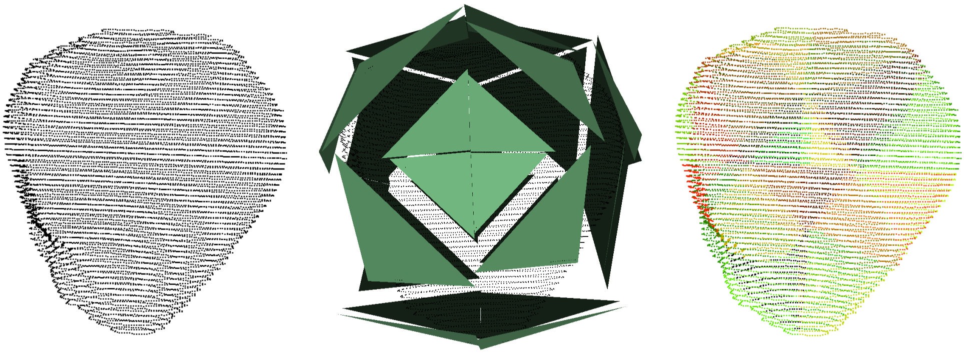

Figure 5.10: Left, external contours extracted from the segmentation results. Centre, a tetra-tree associated to the contour points. Right, points belonging to each tetra-cone are shown in a different colour.

Since a tetra-cone is defined by three planes that intersect in a point, the top cover of each tetrahedron is a triangle. This top cover is perpendicular to the segment that goes from the centroid of the model to its incircle. The distance from the centroid of the model to the top cover is established by the point classified in the tetra-cone which is furthest to the centroid. Figure 5.10 shows an adapted tetra-tree associated with the point cloud representing the patella.

In order to implement a divide-and-conquer approach, contours are classified in a tetra-tree [47] and triangle patches are generated in each of the tetra-cones. After that, the patches are hierarchically sewn in order to obtain the final mesh. In that stage of the algorithm, the patches are previously projected onto the planes defined by the top cover of each tetra-cone; hence the decision of which points should be sewn together becomes a 2D problem. Furthermore, the use of a tetra-tree provides additional properties to the generated model that encourage its utilization in interactive environments [46]. This spatial decomposition enables the visualization of the triangles that either take part or are close to taking part in an interaction, enabling the accessibility to far or non-visible parts of the generated model without having to use additional algorithms such as ray-triangle or ray-box intersection tests. Additionally, the tetra-tree provides a smooth transition when changing the Level of Detail (LoD) during interaction. In contrast to other widely used spatial decompositions like the octree [13], the tetra-tree is object-centred; hence the classification of the geometry is invariant to rotations. Moreover, the construction cost is lower and the interaction times are faster than using an octree [46].

With the aim of easing the understanding of the proposed algorithms, a simple model representing a patella (Figure 5.10, left) will be used to exemplify the process in the following subsections. Since the tetra-tree is centred on a point and is based on spherical subdivisions, this spatial decomposition fits better irregular bones like the patella, as well as isolated ends of large bones such as the distal femur.

5.2.2 Generation of patches

The classification of all the contour points in the tetra-tree enables the execution of a divide-and-conquer approach, so that the points classified in each tetra-cone are triangulated in order to generate patches. Nevertheless, not all the points classified in a tetra-cone must be triangulated together. In order to avoid generating incorrect triangles, the points are clustered before being triangulated. A Euclidean cluster extraction algorithm has been used to group the points that must be triangulated together. With this algorithm, the points are grouped whenever they can be connected by points closer than a pre-defined threshold. In the case of the patella that we use as example, only one cluster is generated for each tetra-cone.

Once the clusters have been generated, the points belonging to each cluster are used to generate patches. For this purpose, several triangulation algorithms were tested. Specifically, Ball-pivoting [6] and Delaunay triangulation [91] were considered. Nonetheless, both alternatives were discarded, because they did not produce acceptable results. Ball-pivoting achieves good results if the distance between slices is not too high. Otherwise, the algorithm generates holes. On the other hand, a 2D Delaunay triangulation was tested after projecting the points onto the top cover of their associated tetra-cone. However, the algorithm generates incorrect triangles in concave areas because of the projection of the points.

The proposed method triangulates points belonging to consecutive contours. In order to perform the triangulation, the connectivity information of the contours is used. First, the contour points are sorted and grouped by slices. Then the contours of each slice are divided into connected polylines. After that, the polylines belonging to consecutive slices are triangulated. Depending on the number of polylines in which each contour has been divided, three different triangulation cases are considered: one-to-one, one-to-two and two-to-two (Figure 5.11). Any other case is discarded and thus not triangulated during

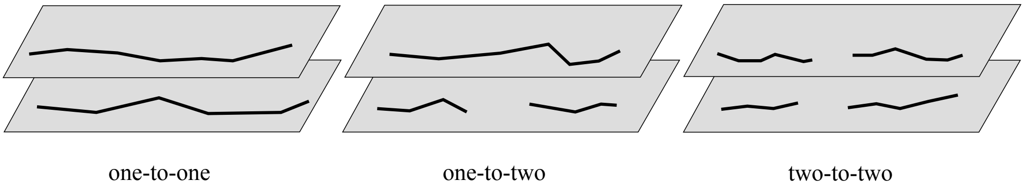

Figure 5.11: The tree different cases considered when sewing polylines between consecutive slices. Any other case is not triangulated in this stage of the algorithm.

this stage of the algorithm. Depending on each case, a different triangulation algorithm is applied.

One-to-one triangulation

The easiest case occurs when both contours are composed by only one polyline. In this case new triangles are generated, minimizing the number of triangles with large sides and small angles. Given the first two points of each polyline A_0, A_1, B_0, B_1, the algorithm first computes the distance from A_0 to B_1 and the distance from B_0 to A_1. The smaller of these two distances defines the segment which is used to generate the triangle. In Figure 5.12, the red coloured triangle $A_0B_0A_1$ is discarded since the distance between B_0 and A_1 is bigger than the distance from A_0 to B_1. This step is repeated until all the points belonging to one of the polylines have been triangulated. Finally, the last point of the already sewn polyline is used to generate triangles with the remaining points of the opposite polyline. Considering that all the points of B have been triangulated and A_j is the next point in A to be triangulated, the algorithm tries to generate new triangles until the distance from A_j to B_m is greater than twice the average distance between A and B. This constraint provides a better triangle distribution and avoids degenerated triangles. In Figure 5.12, the purple coloured triangle $A_{n-1}B_mA_n$ is discarded because the distance from B_m to A_n is bigger than twice the average distance between A and B.

One-to-two triangulation

If one of the contours is composed by two sub-polylines - A and B -, these must be sorted before being sewn. Then the sub-polylines are sequentially sewn to the polyline C of the opposite contour. Since the two sub-polylines are sewn using the same process, the algorithm to sew a single sub-polyline A to a polyline C is explained for simplicity. Firstly, the polyline point C_i closest to the first point of A is calculated. In Figure 5.13,

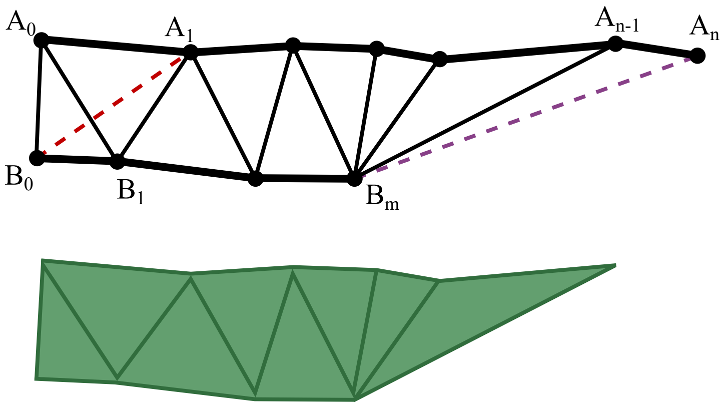

Figure 5.12: Top, example of a one-to-one triangulation of polylines using the proposed algorithm. Bottom, obtained triangles.

the closest point to A_0 is coloured in purple. After that, the one-to-one algorithm is applied but starting from the closest point previously calculated. Using that algorithm, new triangles are generated until all the points of A have been triangulated. Then the last point of A is used to generate new triangles. The main difficulty consists of deciding when to stop sewing a sub-polyline and to start the next one. Being C_j the next point of the polyline to be triangulated, the algorithm stops sewing the sub-polyline A when the distance from the last point of A to C_j is greater than twice the average distance between A and C and it is also bigger than the distance from the first point of B to C_j. As in the one-to-one case, these constraints provide a better triangle distribution and avoid degenerate triangles. Once the first sub-polyline A has been sewn, the same procedure is repeated for the second sub-polyline B. In Figure 5.13, the closest point to B_0 is coloured in blue. Being C_k the next point of C to be triangulated, in this case new triangles are generated until the distance from the last point of B to C_k is larger than twice the average distance between B and C or all the points have been triangulated. In Figure 5.13, the red coloured triangle $C_{l-1}B_mC_l$ is discarded because the distance from B_m to C_l is bigger than twice the average distance between B and C.

Two-to-two triangulation

The two-to-two case requires deciding which sub-polylines have to be sewn together. Two sub-polylines are sewn if the average distance between them is less than a predefined threshold. This value is manually set and mainly depends on the distance

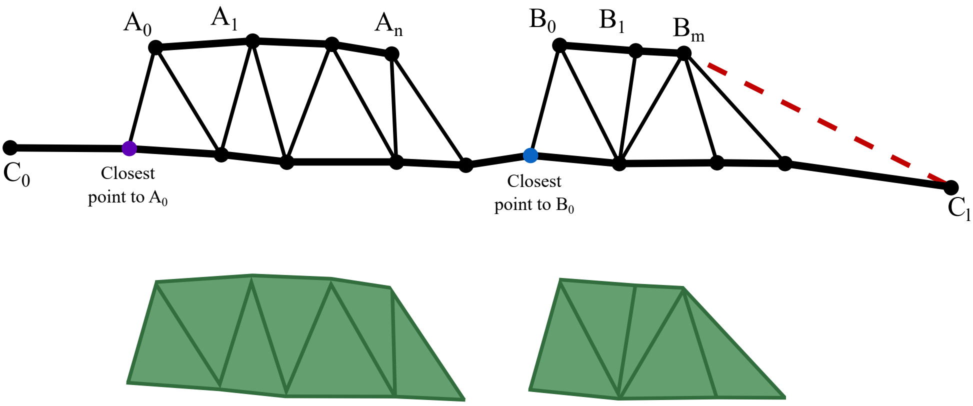

Figure 5.13: Top, example of a one-to-two triangulation of polylines using the proposed algorithm. Bottom, obtained triangles.

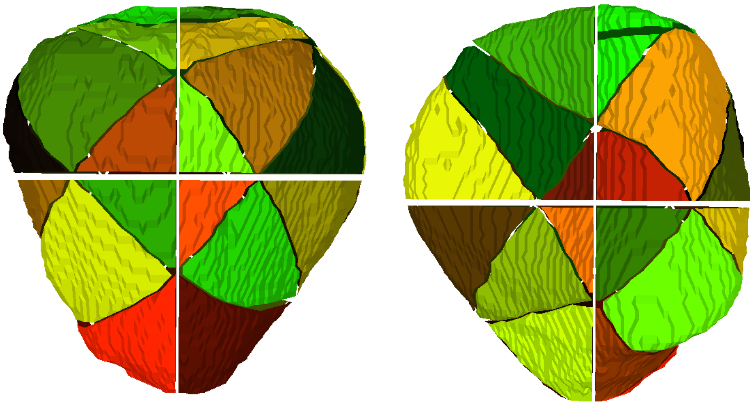

Figure 5.14: From left to right: front and back views of the result of applying the triangulation algorithm to each cluster in the leaf nodes of the hierarchy.

between slices in the source medical images. The order in which the sub-polylines are sewn is also determined by the distance between them. Firstly, the system tries to sew together the two closest sub-polylines, and then the two remaining sub-polylines. In both cases, the one-to-one algorithm (Figure 5.12) is used in order to sew the two sub-polylines.

The triangulation algorithm described above must be repeated in order to generate patches for each cluster. Figure 5.14 shows the patches generated by the triangulation algorithm in the case of the patella.

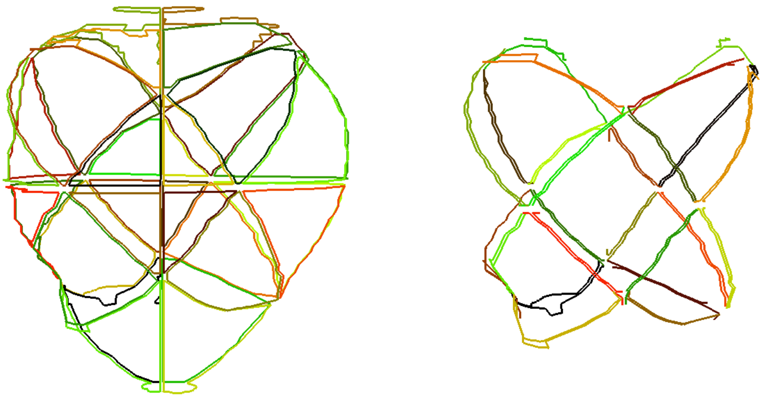

Figure 5.15: Left, external contours of the patches generated in each leaf tetra-cone. The external contour of each patch is displayed in a different colour. Right, pairs of polylines that are triangulated together in the last level of the tetra-tree. Each pair is displayed in the same colour.

5.2.3 Sewing triangle patches

The triangulation algorithm generates one or more triangle patches for each tetra-cone. In order to close the mesh, these patches need to be sewn together. For that purpose, the tetra-tree hierarchy is traversed in post-order. For each node of the hierarchy, the patches stored in its 4 children nodes that are closer than the previously defined threshold are sewn. As occurs in the patch generation algorithm, the threshold value is manually chosen and depends on the point distribution, and thus on the distance between slices and the resolution of the source medical images. The order in which the patches are sewn is defined by the distance between patches and closer patches have priority.

The algorithm proposed to sew two patch contours starts with the identification of the points to be sewn. For that purpose, the external edges of each patch are calculated as the edges that only belong to one triangle (Figure 5.15, left). Then these edges are projected onto the plane defined by the top cover of the tetra-cone containing them. In order to avoid overlapping of the projection of the two patches, the patches are separated by the displacing them the distance between the centroids of both patches, in the direction defined by both centroids. In 2D, the edges to be sewn are defined as those whose points can be connected to a point belonging to the other patch without intersecting any of the two patches (Figure 5.16). Figure 5.15 (right) shows the polylines to be sewn between

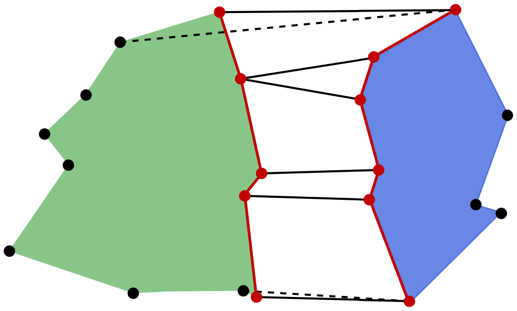

Figure 5.16: Schema representing the selection of the edges to be sewn between two patches. Solid lines connect points of edges that belong to the group of edges to be sewn group since they do not intersect with any patch. Dotted lines intersect with the patch, and therefore their associated edges are discarded.

every two patches. Finally, new triangles are generated using the one-to-one method to sew two polylines already described in Section 5.2.2.

This sewing procedure is repeated in each node of the hierarchy. Once all the nodes of a level have been processed, the generated patches are transferred to the parent node and the sewing algorithm is repeated at the upper level. The process is repeated until reaching the root node of the tetra-tree. The top image of Figure 5.17 shows the patches generated in each of the tetra-cones belonging to the top level of the tetra-tree.

At the top level, the parent node receives the patches generated in its eight children nodes. These patches have to be sewn together in order to close the mesh. In this step of the algorithm, patches are sewn two by two in a sequential process. Figure 5.17 summarizes the three stages of this process.

In the first two stages of the process (Figure 5.17, top and middle), the criteria used to select the points to be sewn is that each point must be sewn to its closest patch. Given a pair of patches, P_a and P_b, the points to be sewn in P_a are those whose closest patch is P_b. The points to be sewn in P_b are calculated using the same procedure. Then patches are sewn using the one-to-one algorithm previously described to sew polylines. Due to the holes associated with the first and the last slices of the stack, two sub-polylines may be obtained when calculating the edges to be sewn in some cases. To deal with the sewing in these cases, the two-to-two algorithm is applied.

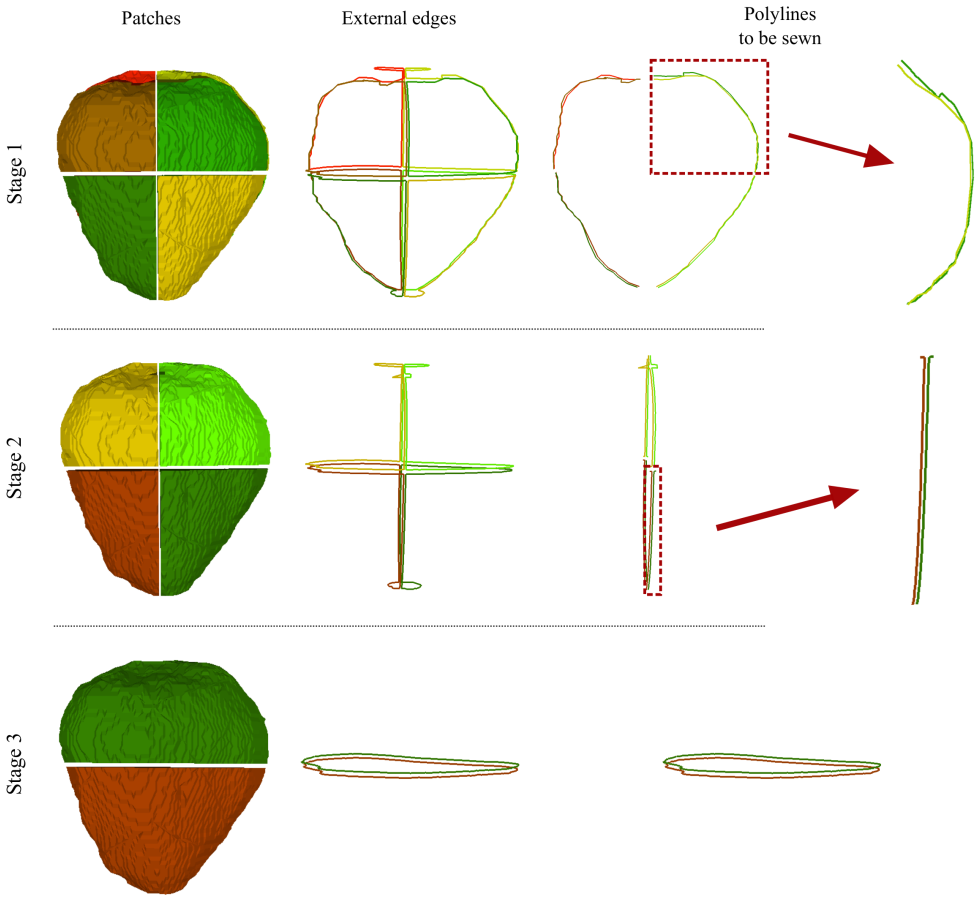

Figure 5.17: Left, patches received as input in each of the three stages carried out at the top level of the tetra-tree. Centre, external edges of those patches. Right, calculated polylines to be sewn in each of the three stages.

In the last stage of the process (Figure 5.17, bottom), only two patches are remaining. In this stage, all the external edges of both patches belong to the polylines to be sewn, and therefore, the algorithm to extract external contours is applied to get them. Since the polylines representing the external contours are closed, the one-to-one sewing algorithm requires being adapted to properly join the two patches. As mentioned above, the generated model contains two holes associated with the contours in the first and the last slices of the stack. Nevertheless, these two holes do not affect to the application of the algorithm, and can be filled at the end. Figure 5.18 shows the results obtained by applying the method in the case of the patella.

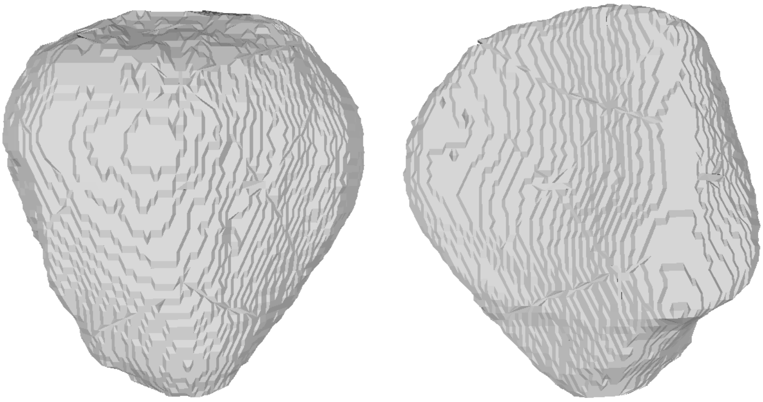

Figure 5.18: Front and back views of a mesh generated using the proposed procedure in the case of the patella.

5.2.4 Issues to be improved

Different tests have been carried out in order to measure the convenience of the developed method to generate 3D meshes representing bone fragments. Although the developed algorithm produces promising results in the generation of bone fragment models, some cases in which the algorithm does not perform well have been detected. The classification of points in the tetra-tree is not always optimal, and hinders the generation of patches in the more irregular areas, especially in the case of long bones. In addition, the classification of points often generates triangulation cases that cannot be solved using the algorithms described in Section 5.2.2. Finally, in the case of models with several concavities, the linear projection proposed in Section 5.2.3 in order to find the edges to be triangulated for sewing two patches may not be the best option. These cases should be fixed to enable the application of the method in the generation of more complex bone fragment models. This section identifies and describes all the detected cases.

Classification in the tetra-tree

Regarding the classification of the point cloud, the tetra-tree does not properly adjust to it in the case of long bone fractures. In these cases, the more detailed area, which is usually the fracture zone, is located at the ends of each bone fragment; hence the tetra-tree should be further subdivided in those areas. Nevertheless, the stop criteria defined in Section 5.2.1 does not produce good results, since it carries out a more detailed subdivision

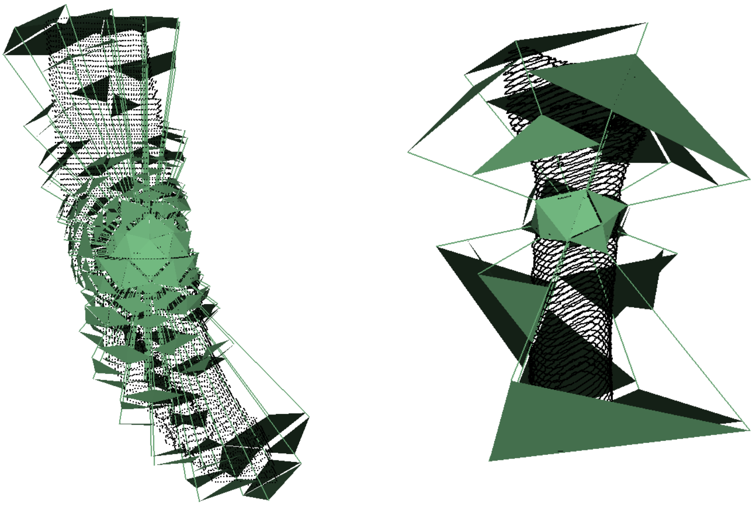

Figure 5.19: Tetra-tree applied to two different point clouds representing long bone fractures.

at the middle of the bone fragment (Figure 5.19). The research of new stop criteria is required in order to obtain a better adjustment of the tetra-tree. In this way, additional information such as the intensity value or the estimated curvature at each point could be utilized.

Generation of patches

During the generation of patches from contours, we detected that two special triangulation cases are not resolved. On the one hand, some isolated points can be classified into one tetra-cone because of the location of the centroid and the orientation of the tetra-tree (Figure 5.20, left). These points divide polylines into two sub-polylines, and thus a hole is obtained during triangulation. Furthermore, isolated points may cause that the triangulation algorithm described in Section 5.2.2 generates incorrect triangles (Figure 5.20, right). In order to solve this case, two different approaches could be used: developing a new triangulation algorithm that is not affected by isolated points, or reclassifying the points of the affected tetra-cone avoiding isolated points.

On the other hand, a closed small contour can be completely classified in a single tetra-cone. Therefore, that closed contour must be triangulated with the polyline extracted from the next slice (Figure 5.21). The one-to-one triangulation algorithm described in

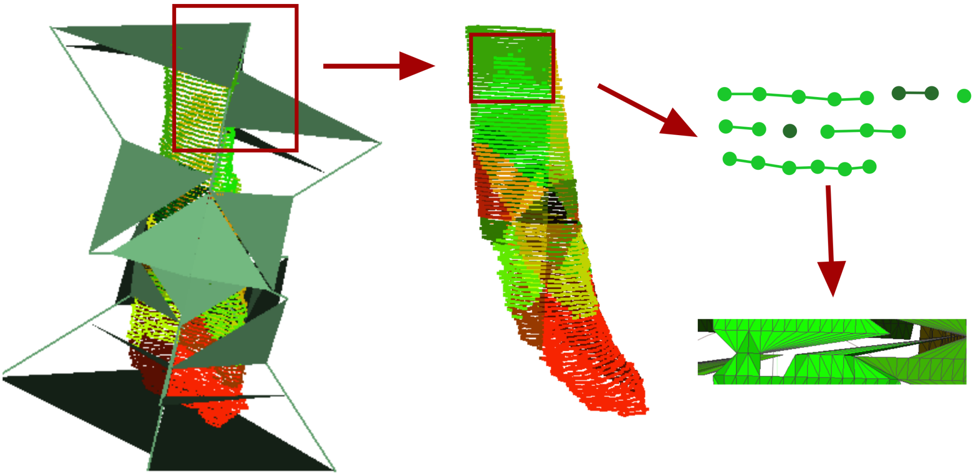

Figure 5.20: Left, tetra-tree applied to a point cloud. Centre, points classified in each tetra-cone are displayed in a different colour. Right, detailed view of the contours containing the isolated points and triangulation performed by using the method described in Section 5.2.2.

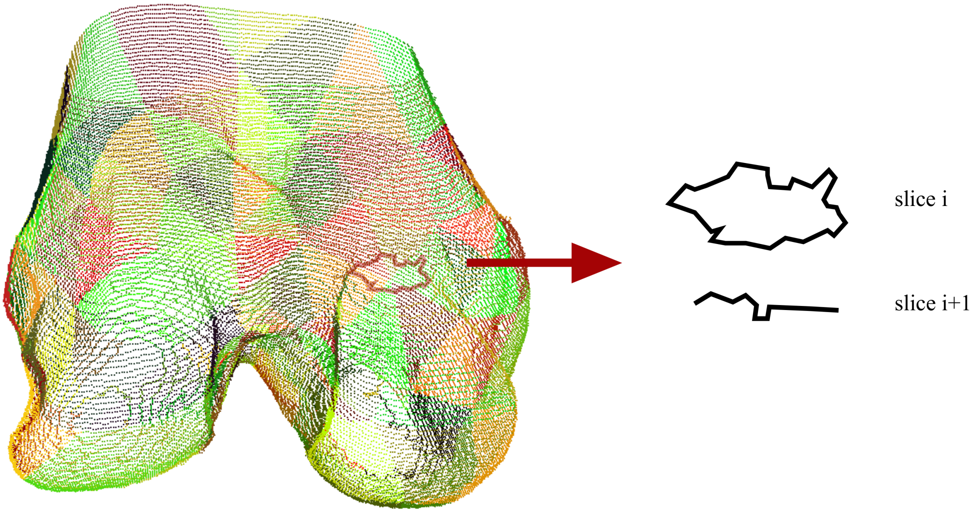

Figure 5.21: Left, point cloud representing a distal femur in which points classified into each tetra-cone are displayed in a different colour. The remarked small closed contour has been completely classified in a single tetra-cone. Right, the new triangulation case that results from that classification.

Section 5.2.2 is not able to properly generate triangles in that case. To address this case, two different strategies could be utilized: reclassifying the points in the tetra-tree avoiding the classification of entire contours in a single tetra-cone, or developing a new triangulation algorithm with the purpose of dealing with this case.

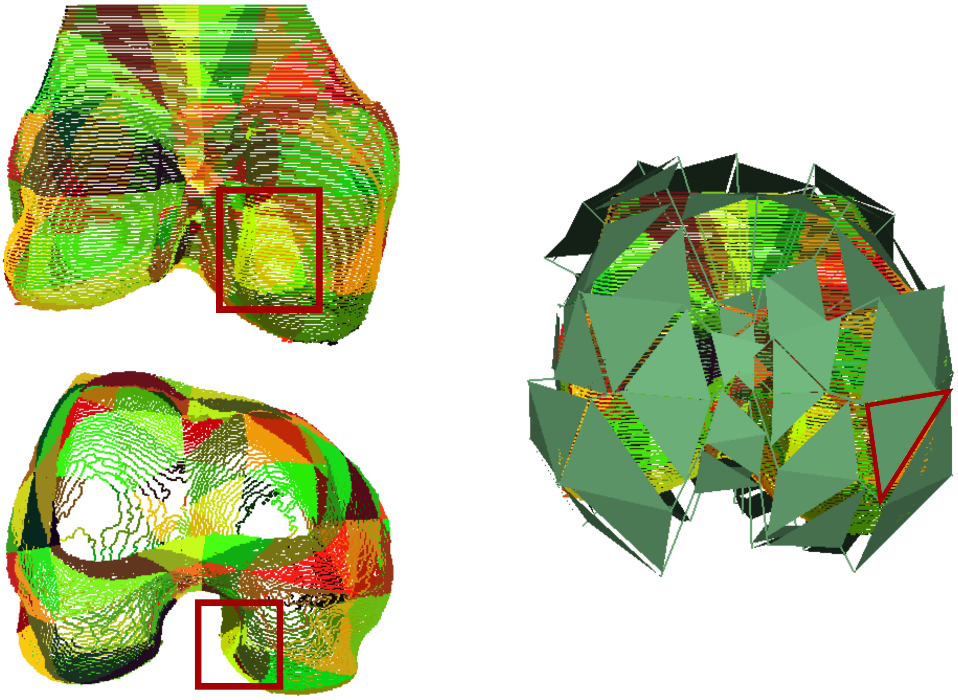

Figure 5.22: Left, front and top view of the point cloud representing the distal end of a femur. Points classified in each tetra-cone are displayed in a different colour. Red rectangles point out a concave zone in the point cloud. Right, front view of the tetra-tree associated with the point cloud. The top cover of the tetra-cone containing the concave area is marked in red.

Sewing triangle patches

Once patches have been generated in the leaf tetra-cones, these are hierarchically sewn as described in Section 5.2.3. In order to calculate the edges to be triangulated, patches are linearly projected onto the top plane of their associated tetra-cone. This projection obtains good results in some of the testes cases. However, the linear projection of the patches onto the top plane of the tetra-tree does not work well if the bone fragments have various concavities, such as the case of femoral condyles (Figure 5.22). In order to deal with these cases, a different type of projection should be used.

5.3 Conclusions

In this chapter, a comparison between well-known methods to generate triangle meshes to represent bone fragments has been presented. This study has previously required extracting point clouds representing the outer part of each bone fragment from CT scans. In order to segment bone fragments from CT image stacks, the method proposed in Chapter 3 has been used. Then the external contour of each segmented region has been extracted. The points of both the segmented regions and the extracted outer contours have been used as input for the tested algorithms. Additionally, surface normal vectors have been estimated using the connectivity information of the contours. Finally, some tests have been performed with the aim of knowing the benefits and shortcomings of each method when generating triangle meshes representing different bones and fracture types.

According to the obtained results, the performance of all the tested methods is heavily influenced by the quality of the CT stacks. However, Poisson reconstruction obtained best models for visualization and MC from regions was the best option for computer-assisted medical procedures. All the methods obtained better results if the points were homogeneously distributed. For that purpose, the distance between slices should be smaller, but this implies increasing the radiation to the patient. Therefore, the development of new data acquisition techniques that reduce radiation would greatly improve the results obtained by the reconstruction algorithms. On the other hand, segmentation results also influence the obtained meshes; due to the fracture, trabecular tissue is in the outer part of the bone, and the irregularity of this type of tissue and the low resolution of the CT image stack make difficult to correctly identify the boundary of bone fragments, especially when two fragments appear joined by their fracture zones in the CT image. This may lead to the generation of wrong surfaces in the fracture area.

In addition, a preliminary study for the development of a new method to generate triangle meshes representing bone fragments has been presented. This method uses a spatial decomposition named tetra-tree to implement a divide-and-conquer approach that allows simplifying the problems that arise when triangulating the contours extracted from consecutive slices. The use of a tetra-tree makes the models be more useful for interactive applications [46]. The presented method is under development, and thus only initial results have been shown as well as the detected issues to be improved. This work allows reaching the goal of researching and developing new methods to generate 3D models representing bone fragments. In the future, the method could be extended to work with alternative spatial decompositions that fit better the shape of any bone fragment. For that purpose, these spatial decompositions could use not only the position of the points, but also the intensity and the estimated curvature at each point, in order to adjust them better to the bone fragment in the more detailed areas like the fracture zone.

CHAPTER 6

Discussion, conclusions and future work

Computer-assisted techniques can supply technological support to the pre-operative planning of the bone fracture reduction process, reducing surgery risk and diminishing recovery time. In this work, we have reviewed the techniques and approaches proposed to help medical specialists in this process, from the generation of bone and fragment models to the analysis of the final composition. The methods have been summarized and classified, and their main advantages and shortcomings have been highlighted. The study of these methods has revealed that none of the stages of the computer-assisted reduction process is completely resolved, neither the identification and generation of fragments, nor the fracture reduction itself. As a consequence, this field of research is still open and faces important challenges in a mid-term period. In this work we have presented new methods to identify bone fragments from CT scans, to calculate the contact zone between fragments, to compute the fracture reduction and to generate triangle meshes representing bone fragments. Additionally, some tools have been developed to help specialists to manipulate bone fragment models generated from medical images (Appendix A).

6.1 Identification and generation of bone fragment models

The main difficulty in the segmentation step is to identify each one of the fragments deterministically. Currently, this process requires manual user interaction and sometimes expert knowledge. This issue is an open research line where previous knowledge would be advisable and more accurate segmentation and labelling methods are needed. With the aim of identifying each of the bone fragments, it is necessary to label them and to separate wrongly joined fragments. On the other hand, the key procedure in the segmentation step of most of the currently proposed methods is the selection of thresholds and seeds. Nowadays, the labelling step is in most cases performed manually. As a consequence, both segmentation and labelling can require a post-processing step. Thresholding-based approaches do not label bone fragments, thus fragments have to be labelled after the segmentation process. The approach presented in this work uses the placement of the seed required to segment each bone fragment to also label it. Other approaches in the literature also resolve this problem by using seeded based methods [38, 29]. In all these cases, seeds should be placed by an expert. Other studies propose methods to label bone regions automatically, but then an expert should evaluate the results and decide to which bone or fragment each region belongs. Ideally, all the bone fragments should be segmented automatically and simple bone fragments should be identified without user intervention. Then, in more complex cases, the expert could decide the bone to which each fragment belongs.

Because of the complexity of the trabecular tissue, the selection of threshold intensity values is a challenging procedure. Threshold values are difficult to determine, even manually, and each slice may require a different value. In addition, it is particularly difficult to set the threshold to segment bone tissue near joints. The method presented in this PhD dissertation uses a linear regression model to calculate the threshold value for each slice. Despite the fact that the calculated thresholds produce good results in most cases, the model needs to be built for each set of datasets. Most of the proposed methods in the literature require that the user to manually specify the intensity threshold. Ideally, threshold values would be selected automatically from the information available in the set of slices in all cases.

As mentioned in Section 3.1, it may be necessary to split up fragments during the identification process. The low image resolution may cause very close fragments to appear joined. This is especially common in fractures produced by high energy traumas. Different approaches have been proposed to deal with this problem: interactive tools [38, 98], 3D CCL and graph cuts [29], re-segmentation [57] or even comparison with healthy models [96]. Nevertheless, manual and semi-automatic fragment separation takes time; hence automatic methods would be important to enable time-saving. One solution would be to improve the segmentation method, thus joined fragments are avoided. The presented method separates wrongly joined bone fragments during segmentation by using additional seeds. A full automation of this process would be profitable, because then no additional methods after the segmentation process would be required. However, the usual low resolution of CT scans makes this automation very difficult. An alternative approach would be to directly implement a method to overcome the problem after the segmentation step. Therefore, the whole segmented solution would be available and additional information could be used from the already segmented adjacent slices.

The use of more precise data acquisition technology could prevent fragments from appearing together in most cases. CT is the most common method to acquire 3D images in order to distinguish bone tissue . This procedure has an important radiative impact on the patient. In recent years, there is a tendency to use μCT images to add extra precision to given regions. Nevertheless, this type of images increases radiation [88]. Thus, in spite of currently being the best option to distinguish bone tissue, CT scans possess this inescapable drawback. For that reason, finding alternative scan methods to improve medical images without increasing radiation is mandatory and has become a challenge for the research community. Although the main purpose of medical images is clinical diagnosis, they are also the input of computer-assisted techniques that can lead to a reduction of surgery time.

On the other hand, there exist studies that try to segment bone tissue from Magnetic Resonance Imaging (MRI) and X-ray . These techniques require extra information because osseous tissue cannot be identified as easily as in a CT scan. Proposed methods are based on an atlas previously constructed from CT images [39] or are applied to very specific bone areas [45, 37]. However, there are no proposed studies to identify fractured bone from these types of medical image.

With respect to the 3D reconstruction step, the MC [63] algorithm is a *de facto* standard due to its simplicity and advantageous trade-off for generating geometric models. However, this method has several disadvantages that represent a current challenge: the large amount of generated geometry and the noise obtained due to the complexity of trabecular tissue. In this PhD dissertation, some other mesh generation methods (Poisson reconstruction [53], Ball pivoting [6] and APSS [36, 35]) have been tested in the generation of triangle meshes representing bone fragments, extracting their main benefits and drawbacks. The conducted tests demonstrated that Poisson reconstruction obtained the best meshes for visualization and MC from regions was the best option for computer-assisted medical procedures. Furthermore, this work presents the initial results of a preliminary study for the development of an innovative method to generate 3D models of bone fragments by sewing the external contours of the segmented bone regions. In order to deal with special cases during the sewing procedure, the method implements a divide-and-conquer approach using a spatial decomposition named tetra-tree. Although the method produces promising results, some detected issues should be improved in order to generate more complex models. The development of an alternative spatial decomposition that fits any bone fragment better is an open research line. On the other hand, point clouds have been used as input for the proposed method to compute the fracture reduction because this data structure is easily generated and the proposed method does not require the geometrical information provided by meshes. Nevertheless, point clouds do not avoid the problems for visualizing, managing and interacting with these large models.

The main drawback of using geometry optimization methods is the difference between the processed models and the originally detected bone boundaries. This may cause some bone fragment features, such as fracture surfaces, to become difficult to detect. Paradoxically, smoothing is required to remove undesired ripples, whereas at the same time it can blur the boundaries and hinder the extraction, identification and ulterior reduction. However, these techniques are very useful in the visualization process where a LoD strategy can be used in order to adapt the computational requirements to the specialist's demand.

6.2 Computer-assisted fracture reduction

The revision presented in Section 2.2.2 allowed us to discuss about the existing approaches in the literature for computer-assisted fracture reduction regarding comminuted fractures. Most of the proposed works in the literature are focused on long bones (tibia, fibula, femur and humerus) [107, 105, 68, 114, 75, 97, 1, 30]. In a long bone, the diaphysis is cylindrical and is completely surrounded by cortical tissue. In contrast, the cortical zone in the epiphysis is very thin; hence trabecular tissue can even appear in the outer part of the bone. Therefore, this extra information has favoured the development of computer-assisted techniques to help specialists reduce long bone fractures. With the aim of reducing fractures of irregular bones, only just interactive tools have been proposed [19, 106, 28, 43]. As an exception, Chowdhury et al. [17] present an approach to reduce craniofacial fractures by calculating, matching and registering fracture zones and Kato [52] proposes to reduce acetabular fractures using a template.

Automatic methods require extra information in order to avoid user interaction. Most of the proposed approaches use templates in order to match bone fragments [68, 1, 52]. Others take advantage of differentiable features of a specific bone. Winkelbach et al. [107] propose a method to automatically reduce comminuted fractures of femur. For that purpose, the authors make use of the special features of long bones already mentioned in the previous paragraph. Chowdhury et al. [17] present an approach to automatically reduce craniofacial fractures. In this case, the authors take advantage of a previous classification of fragments as terminal or non-terminal, based on the presence or absence of condyles. Nowadays, there is not a method to automatically reduce comminuted fractures that could be applied to any bone. This PhD dissertation has presented a method to automatically calculate the contact zone between each two fragments. The calculated fracture zones enable the reduction of complex bone fractures. Regarding the final alignment, most of the proposed works use an adaptation of the ICP algorithm. The main drawback of this method is that local minima can be obtained. Therefore, a previous coarse alignment of the bone fragments is required. For that purpose, the application presented in Appendix A may be used to carry out a proper previous alignment of bone fragments. The development of more robust registration methods to reduce the probability of finding a local minimum would lead to the achievement of more automatic approaches.

The stabilization of a fracture reduction is a key step in the process of guaranteeing the fixation of the fragments in a long-term period. However, at the moment automatic stabilization is almost an unexplored field. Nonetheless, several user-interactive solutions have been proposed to help the specialist to plan the surgery and accelerate the whole process. This lack of general solutions is due to intrinsic technical difficulties, to the necessity of expert knowledge and to the requirement of a precise analysis of the fracture reduction. During the simulation, plates and screws have to be defined according to the fracture. If plates are virtually designed, they must be later specifically manufactured for the patient. Notwithstanding that some of the proposed approaches have been tested in clinical cases, implant manufacturing is still an expensive process. Thus, current solutions propose to contour standard implants in most cases.

6.3 Analysis and validation of a computer-assisted reduced fracture

As commented in Section 2.1.2, the computer-assisted reduction of complex bone fractures requires the generation of 3D models from CT images that represent the fractured bones. These virtual models are usually generated using a MC-based approach. Then noise is removed and models are simplified in most cases. Despite the fact that these models are complex, no method has been proposed to evaluate their goodness. On the other hand, there is no standard criterion for evaluating the accuracy of either the fracture reduction composition or the model quality. Different authors have proposed different methods to measure the quality of the obtained fracture reduction; hence the different proposed methods cannot be easily compared. In most cases authors used artificially generated bones to test their proposed techniques. In these cases, the results obtained cannot be compared, since they used different bones to perform the tests: corpse bones and artificial bones of different materials. The special features of long bones have also been used to check the accuracy of the reduction. These features enable the comparison between different methods, but this is restricted to long bone fractures. Registration-based methods can use MSE to test the accuracy of the final alignment of the fracture surfaces. This measurement can only evaluate this step of the process and it is only applicable in methods that use registration techniques. With the aim of facilitating comparatives, the fracture reduction method presented in this work has been evaluated using parameters commonly used in the literature. Specifically, overlapping and contact surface are calculated in clinical cases, and translation and rotation errors are used in cadaver cases since the ground truth is previously known.

6.4 Future work

The next steps in this research line should allow the automation of the surgery as has been introduced in other medical disciplines [50, 79]. This automation would enable the utilization of computer-guided surgery using robotic systems. Regarding this topic, methods to compute the reduction of bone fractures proposed in the literature have been applied in the planning of real surgery. Improvements of the currently proposed techniques would ease the development of robotic systems to assist in or even to perform a real fracture reduction. To the best of our knowledge, the commercial development is not widespread. The research of new methods to automate the fracture reduction process and the analysis of its results could contribute to the development and improvement of this type of commercial solutions.

Given the previous discussion, the computer-assisted bone fracture reduction process may be improved by researching the following topics:

- Identification of bone fragments from CT scans without needing user interaction. This includes the automatic calculation of threshold values and seed placement, if any. Expert aid should be reduced as much as possible.

- Development of new techniques that prevent fragments from appearing together after the segmentation process. This may be achieved by improving segmentation methods or by researching new image acquisition technologies that enhance image resolution without increasing patient radiation.

- Refinement and generalization of the proposed method to generate 3D models from medical images. Additionally, the method should be tested with more clinical cases, not necessarily bones.

- Development of new optimization techniques to improve the generated 3D bone fragment models without modifying the geometrical information originally extracted from the medical data.

- Development of algorithms to automatically perform a coarse alignment of bone fragments without the need of using templates or requiring extra information about the original shape of the bone.

- Automatic calculation of the fixation devices needed to stabilize a fracture as well as their position.

- Development of a new spatial decomposition that fits the shape of any bone fragment, and that is capable of being further subdivided in the more detailed areas.
- Evaluation of the goodness of 3D bone fragment models generated from medical images.
- Development of standard procedures in order to evaluate the goodness of a reduced fracture from both a geometrical and a mechanical point of view.
- Generation of realistic and complex virtual fractures of different bones in order to facilitate the testing of the developed methods.

Appendix A

Interactive tools to assist in pre-operative planning of bone fracture reduction

Once developed methods for identifying bone fragments from CT scans, obtaining 3D models representing bone fragments, identifying fracture zones and reducing bone fractures, they could be integrated into an application that would facilitate the work of medical specialists. This Appendix describes this application in its initial state in which known interactive techniques have been included to enable the generation and manipulation of bone models (Figure A.1). This application facilitates the proper visualization of geometric models representing bone structures and includes techniques to optimize the processes involved in fracture reduction. These techniques allow for better definition of the triangles involved in different processes, such as puzzle-solving problems, and provide visual aid to specialist using advanced collision detection algorithms.

The developed application contains tools to perform a detailed interaction with 3D models of bone fragments, and enable the interactive fracture reduction in cases in which it cannot be carried out automatically. The application uses MC to generate bone fragment models from CT image stacks, since the tests conducted in Chapter 5 revealed that it was the best option for computer-assisted procedures. Notwithstanding, the refinement of the method presented in Section 5.2.1 could enable its utilization for generating models used as input in this type of procedures. As commented in Chapter 5, the use of the tetra-tree provides additional properties to the generated models that improve the user experience in interactive applications. Although CT image stacks are usually segmented before generating the 3D models, the developed application contains

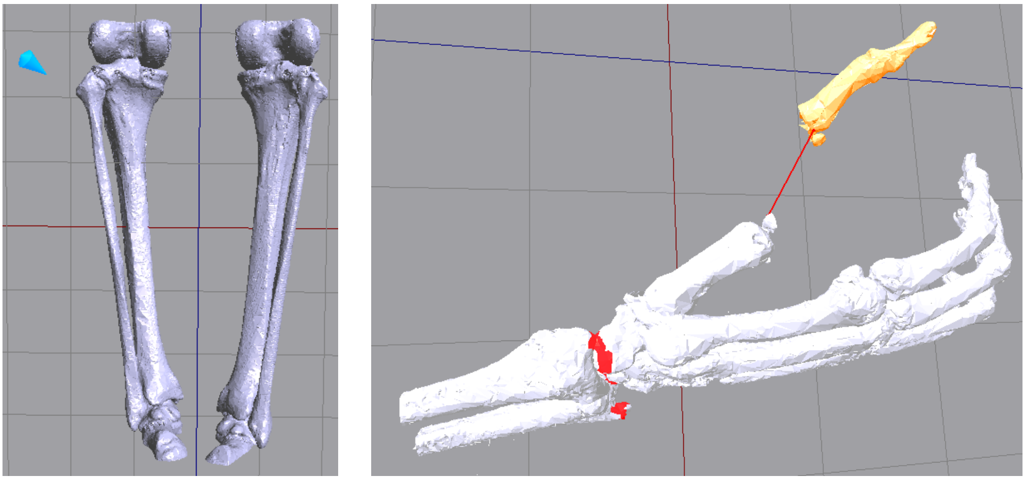

Figure A.1: Examples of using the developed application to interact with osseous models reconstructed from medical images.

tools to clean the image noise and to manually segment and label models. In addition, a comparative study has been carried out with the aim of choosing the collision detection strategy that suits best to the problem. The developed application uses swept sphere hierarchies [56] to calculate overlapping triangles, nearest points and distances between bone fragment models in real-time. Moreover, a stereoscopic motion-tracking device has been integrated into the application with the goal of enhancing interaction.

The developed application is the basis for integrating the tools proposed in this PhD dissertation. The application can be used in the pre-operative planning of a fracture reduction surgery. In that way, the application may be used to better understand bone fractures or even to carry out a manual alignment of the fracture. For that purpose, proximity queries and collision detection methods can help specialists to carry out a coarse alignment of the bone fragments if they are too displaced to apply automatic algorithms. In the future, the developed application will allow including new methods and techniques as well as to complete the fracture reduction process along with the analysis of the obtained solution.

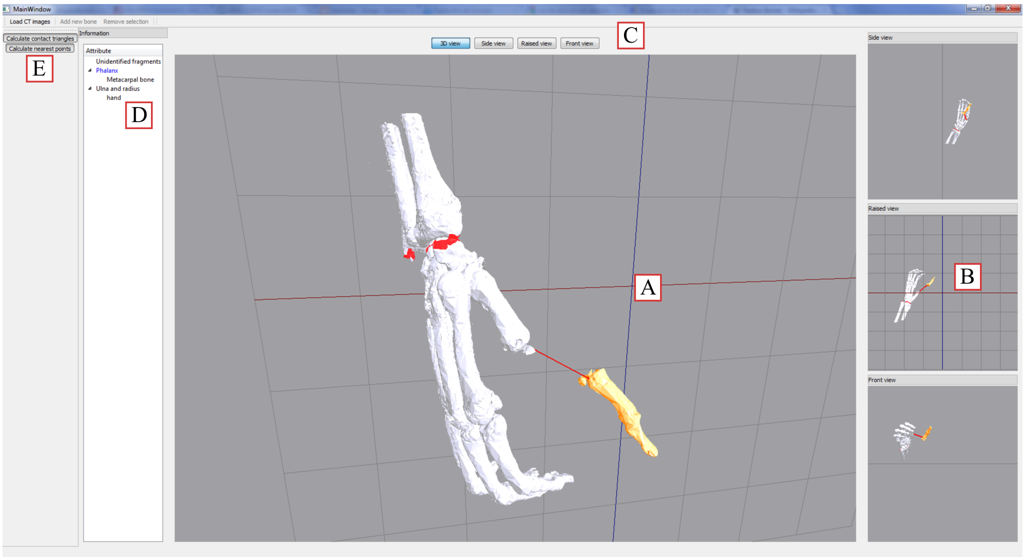

Figure A.2: Screenshot of the application. A - Main canvas. B - Secondary canvas. C - Buttons to switch canvas. D - Hierarchical tree of the scene. E - Toolbars.

A.1 Overview of an interactive tool for computer-assisted fracture reduction

The Graphic User Interface (GUI) is mainly composed of four canvases and a hierarchy tree. The main canvas (Figure A.2, A) allows moving the camera, and the other three show static views: side, front, and top (Figure A.2, B). All the canvas can switch their positions dynamically by clicking on any of the auxiliary canvas or by using the buttons on the top of the window (Figure A.2, C). This interface is similar to that used in current medical visualization applications; hence it is easy to use for professionals. On the left side of the application window, a hierarchy tree shows the relationships between the selected model and the other models in the scene (Figure A.2, D). At the top of the window there is a toolbar that allows loading medical images, defining models and their relationships, removing unnecessary parts, and enabling the calculation of nearest points and overlapping triangles (Figure A.2, E). The procedure for using the application can be summarized as follows. First of all, the user must specify the medical images to be loaded. Once the images are loaded, the application generates a triangle soup from them. Then models must be identified and defined using an area selector. Each time a new model is defined, the user can set the models to which it is related and the application updates the hierarchy of models. Moreover, unnecessary parts and outliers can be removed by using the area selector. The defined models can be selected, translated and rotated to place them correctly in the scene. With the aim of providing assistance during interaction, the application calculates automatically the nearest points and the overlapping triangles between models. Distances and collisions are computed only among the models in the same hierarchy. With the aim of enhancing the interaction, the Leonar3Do [58] input device may be used to manipulate the models.

A.2 Features of the interactive tool

Once a 3D geometric model has been reconstructed, our application allows interacting with it. With that aim, registration, collision detection, picking and multi-view methods have been integrated into the application. These methods let the user define 3D models from the reconstructed geometry and interact with each of them independently. In addition, the detailed collision detection features and the multi-view implementation provide visual aid to the user during interaction. Due to the size of the reconstructed models and the absence of topology, these methods must be fast and robust. The tests described in Section A.3 enabled the selection of a method suitable to work with this type of models.

In order to improve the interaction, support for the Leonar3Do [58] system has been included in the application. This is a virtual reality system that enables stereoscopic interaction. It mainly consists of a spatial input device (the *bird*), 3D glasses and monitor-mounted sensors (Figure A.3). Both the *bird* and the 3D glasses operate in six degrees of freedom, and the sensors can track both the bird and the glasses. To that end, the Leonar3Do system uses a technology based on infrared sensors. The bird has two buttons that can be programmed. In addition, one of the buttons is sensitive to pressure. On the other hand, the glasses enable stereoscopic vision. Leonar3Do can use either an active or a passive system for stereoscopic vision support. While the passive system uses commonly polarizing lenses, the active system uses liquid crystal shutters which are powered through an USB port. In our case, the active system has been utilized.

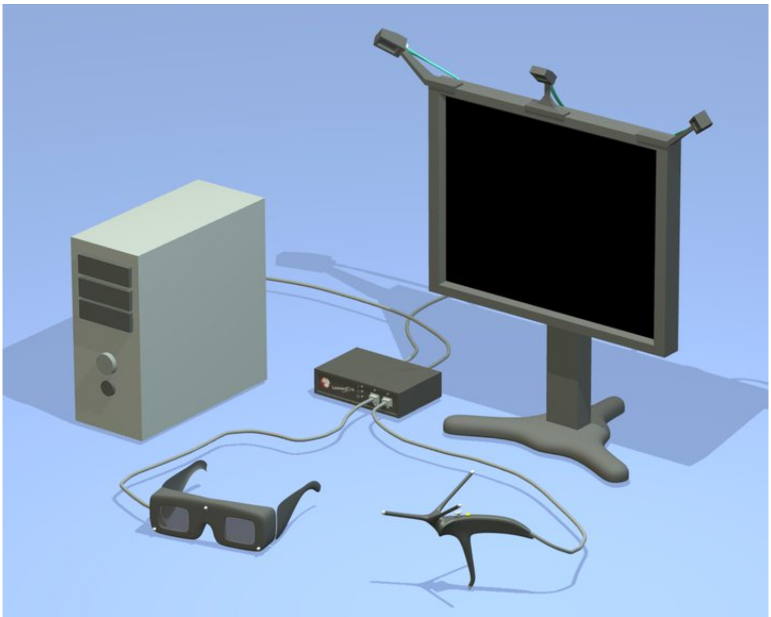

Figure A.3: Virtual representation of the Leonar3Do [58].

A.2.1 3D labelling of the models

In spite of CT image stacks being usually segmented before generating 3D models, the developed application contains a tool to clean noise and to manually segment and label models. This tool allows selecting a set of triangles and defining a model from them (Figure A.4, bottom). The area selector can only select triangles that do not belong to any model. When a model is defined by the area selector, the application allows relating it to another previously defined model. Thus, it is possible to establish a hierarchy between the defined models. Furthermore, the area selector can also be used to remove noise and unnecessary parts from the scene (Figure A.4, top).

In order to determine the triangles that have been selected, it is necessary to check which triangles have at least one vertex inside the selector. If the selector is a rectangle, each selection requires as many frustum-triangle intersection tests as triangles are in the scene. However, this task can be simplified by projecting the triangles in the plane determined by the selector. In this way, it is only necessary to resolve a triangle-rectangle test for each triangle in the scene. In addition, as the intersection is calculated in 2D, it is easy to implement new shapes for the selector.

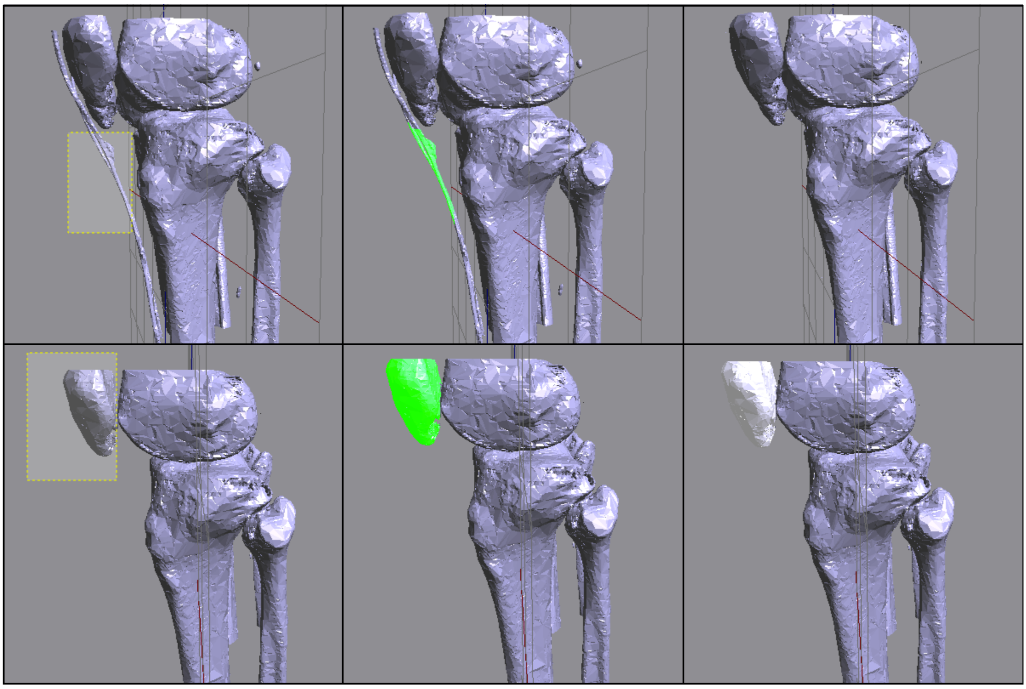

Figure A.4: The area selector may be used to remove unnecessary parts and to define models. Top, from left to right: selecting the part to remove; the selected part is displayed in green; unnecessary parts are removed from the scene. Bottom, from left to right: selecting the triangles that represent the patella; selected triangles are displayed in green; finally, the defined model is shown in white.

Each time the user makes a selection, the scene has to be repainted in order to draw the selector. This can be a problem when the scene is full of large models composed of millions of triangles. In order to handle this problem, repainting the entire scene is avoided by using a frame buffer. For that purpose, each time the scene changes, the frame buffer is updated with the new scene. Afterwards, in the rendering function, the content of the frame buffer is drawn in a 2D texture that is placed at the background of the scene. Therefore, when a 2D selector is drawn, the application only needs to repaint the background texture. To draw the selector, the Qt 2D drawing library is used.

A.2.2 Collision detection

With the aim of easing interaction between the models, the developed application not only calculates the collisions among them but also provides visual aid to the user. For that purpose, a detailed collision must be calculated . Due to the size of the reconstructed

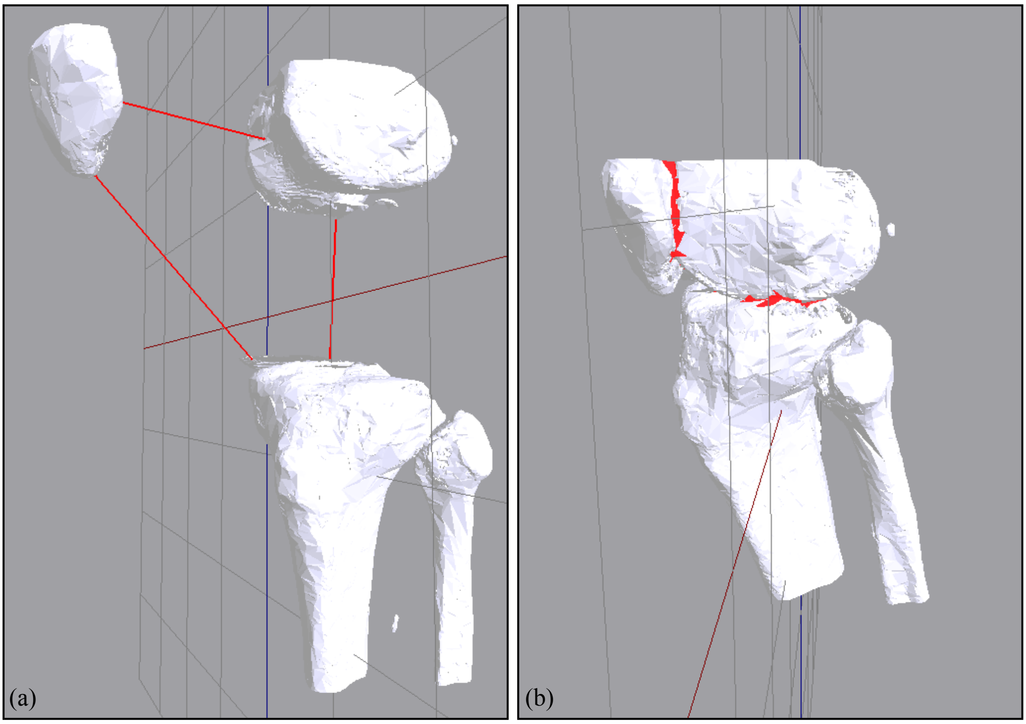

Figure A.5: a) Calculation of the nearest points between some models. b) Overlapping triangles are displayed in red.

models and the absence of topology, the methods used to compute the detailed collision detection must be fast and robust. The tests described in Section A.3 allowed selecting a method suited to work with that kind of models. The results showed that the Proximity Query Package (PQP) library [56] is quite fast and robust and thus it is a proper choice for working with these models. Besides detecting collisions, that library enables the calculation of distances, nearest points and overlapping triangles in real-time. These calculations are useful for avoiding that two models overlap or even letting the user know which parts of the models are colliding (Figure A.5, right). Moreover, the distances and the nearest points between models may help the user to place bone fragment models into their correct position (Figure A.5, left). Support for the Leonar3Do system has been implemented to improve the collision response. When a collision occurs, the spatial input device emits a small vibration.

A.2.3 3D Picking

Once the models are defined, the application allows selecting them by picking . To that end, the Leonar3Do device is used. By moving this device, a 3D cursor in the scene is manipulated. This cursor enables the selection of previously defined models. For that purpose, the small programmable button has been utilized. If the button is pressed when the 3D cursor is colliding with a model, that model becomes selected. If a model is selected, the spatial input device manipulates the selected model instead of the 3D cursor. Apart from this, each selected model can be rotated and translated by using the mouse. When a model is selected, the transformations are applied independently from the rest of the models in the scene.

There exist several strategies to implement picking in a 3D scene. Firstly, a picking method based on unique colour IDs [92] was tried. This is a very easy to implement method that has the following steps:

1. Disable lighting and textures.
2. Render each object using a unique colour.
3. Read colour data from the colour buffer at the current cursor position.
4. Determine the object to which the colour belongs.
5. Enable lightning and textures.
6. Render the scene.

The method is very fast and quite robust. However, the method was in conflict with the frame buffer used by the area selector. Moreover, due to the fact that it is necessary to assign a unique colour to each object, this method cannot work with many selectable objects. For these reasons, a method based on ray picking has been implemented. The method consists of throwing a ray from the observer that pass through the cursor position and calculating the collision between the ray and all the objects in the scene. In order to check the collisions, the PQP library has been used. In this way, the data structures previously created are reused.

Table A.1: Picking time using the ray picking method based on the PQP library. Run in a PC with an Intel i7 2,80GHz, 4GB of RAM, and an NVidia GeForce GTS 240.

# triangles in the scene	Picking time (s)
58206	0.0671
253696	0.2868
428411	0.3845
478462	0.4231
513658	0.4608
546308	0.4949
577827	0.5206

In order to measure the efficiency of the method, some tests have been performed. As shown in Table A.1, the method calculates the ray picking in a reasonable good time, even when there are several hundreds of thousands of triangles in the scene. Moreover, the method is also robust since it is based on the PQP library.

A.2.4 Multi-canvas

Despite the fact that the 3D view provides extra information to the user, orthopaedic surgeons and radiologists usually work with a 2D view of the injured area of the patient. For this reason, the application includes four canvas: a canvas with a free camera and three canvas with static cameras (Figure A.6). By default, the main canvas provides the 3D view, and the other three canvases represent top, side, and front views respectively. However, the four canvases can switch their positions dynamically.

Instead of implementing multi-canvas using several glViewports [92], four QGLWidgets have been used to implement the four canvas. In this way, the application takes advantage of all the functionality that is already implemented in the QtOpenGL module [9]. Compared to other approaches, the use of QGLWidget instances eases the implementation of mouse events for each canvas. Each QGLWidget has an associated QDockWidget that place it into the application window. As mentioned in Section A.1, the four canvas can switch their positions dynamically by clicking on them or by pushing the buttons located at the top of the window. In order to achieve that, an own camera class was implemented.

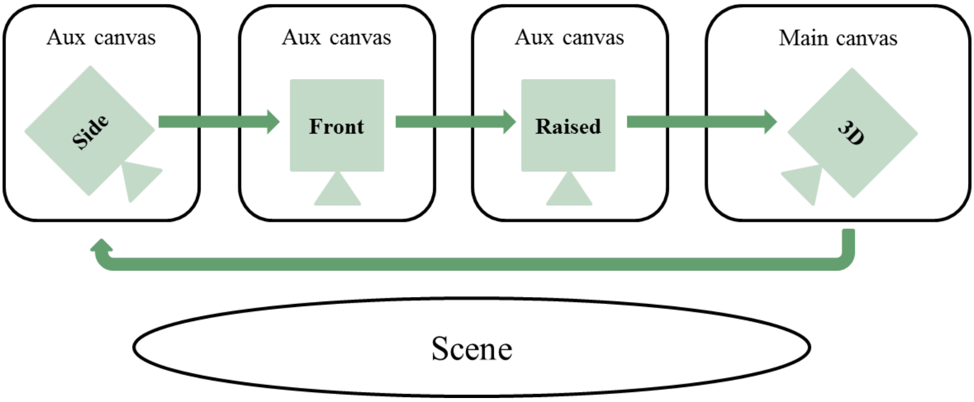

Figure A.6: Multiview architecture implemented in the application.

A.2.5 Stereo visualization and immersion

The Leonar3Do system has been used to provide stereo visualization . Specifically, active glasses have been used to that end. When the stereo mode is activated, the main canvas provides stereo visualization (Figure A.7). This visualization system allows the user to better understand the scene and makes selecting objects easier by using the spatial input device. In addition, the stereoscopic visualization increases the immersion feeling when using the application.

The Leonar3Do system needs to be configured before being used for the first time. This configuration depends on the display used. The height and the width of the monitor must be set manually in order to define the space where the sensors work. Once these parameters are set, the space in which the Leonar3Do works (LeoSpace) can be defined. The origin of the LeoSpace is the centre of the display. The left edge of the display has X coordinate -1 and the right edge has X coordinate +1. Considering the ratio = width/height previously established, the bottom edge has Y coordinate -ratio and the top edge has Y coordinate +ratio. Outside this range, both the spatial input device and the glasses cannot be positioned.

The Leonar3Do API (LeoAPI) is divided into two main parts: the tracking API and the stereoscopic API. The first one allows knowing where both the spatial input device and the glasses are situated. In the developed application, the spatial input device has been used to manage a 3D cursor that enables the selection of models in the scene. Moreover,

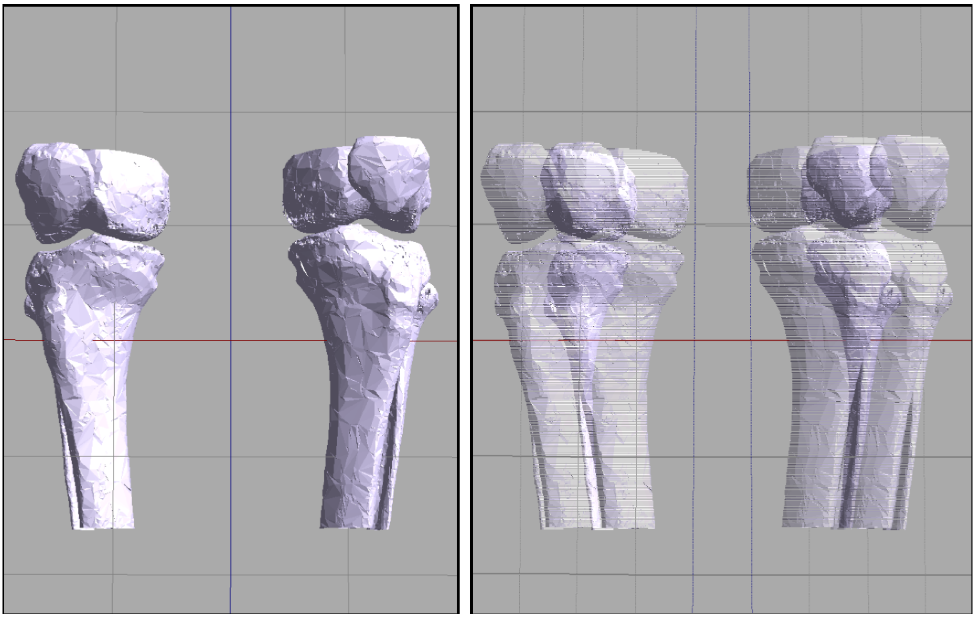

Figure A.7: Screenshot of the main canvas of the application. Left, non-stereo mode. Right, stereo view with image superimposition.

the device allows manipulating a model once it has been selected. On the other hand, the position of the glasses is used to manage the camera. For that purpose, the difference in position of the glasses with respect to the previous frame is calculated. This value represents the translation of the camera in each frame. Therefore, the camera moves accordingly when the user move the head. In order to achieve that, both the translation and the rotation of the user head are considered. This increases the feeling of immersion and makes the application easier to use. Before painting anything, the scene is translated and rotated in order to simulate the movement of the glasses. It is important to consider that the camera position is modified by the LeoAPI so that it is placed at the focal point.

The stereoscopic API enables the implementation of the stereo view and it is responsible for generating the final frame of the application, whether or not the stereo view mode is activated. In our application, only the main canvas has support for stereoscopic visualization. Because of that responsibility, it is necessary to disable the auto-buffer swapping because the LeoAPI already does it. Moreover, the LeoAPI fills a projection matrix that must be provided to OpenGL before rendering the main canvas. However, this projection matrix can be modified before introducing it into the OpenGL pipeline.

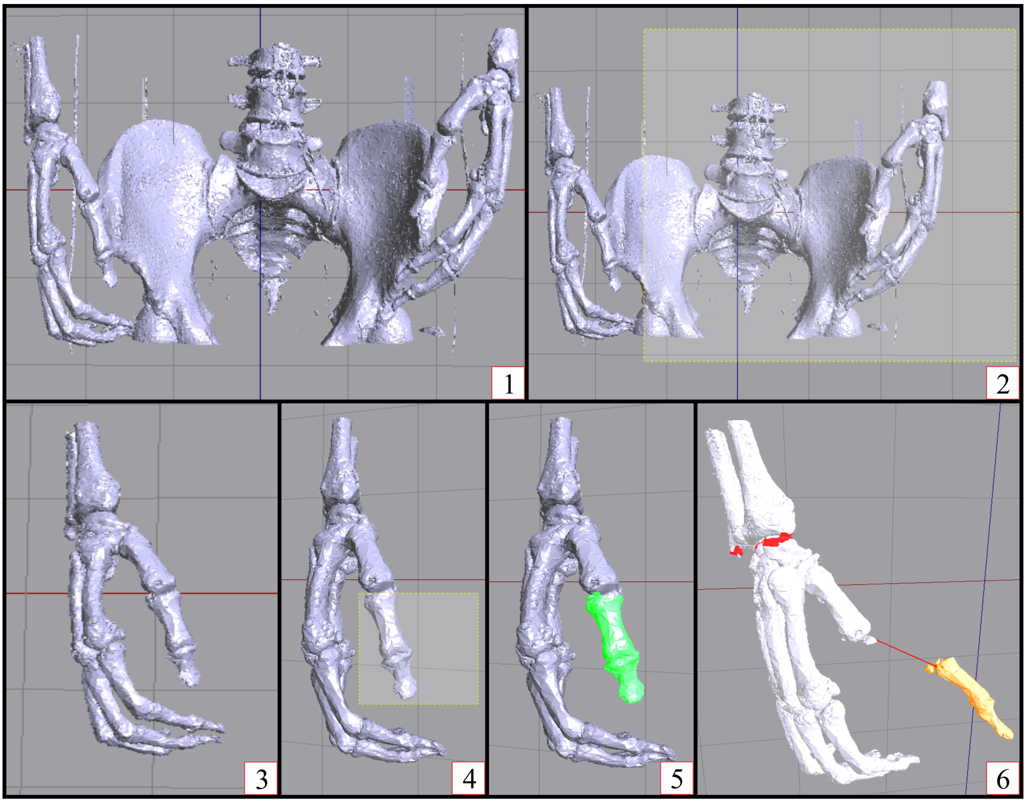

Figure A.8: Example of using the application. 1 - Triangle soup generated from CT scans. 2 - Selection of some unnecessary triangles. 3 - Cleaned scene. 4 - Selection of a model. 5 - Selected model. 6 - Interaction with defined models.

The Leonar3Do API specifies how many views have to be rendered (2 if the stereo mode is activated, 1 otherwise). In the case of the stereo view, the LeoAPI is responsible for generating the projection matrix for the two views. Before introducing the projection matrix into the OpenGL pipeline, the translation and the rotation calculated from the position of the glasses is added to these projection matrices. Inside the stereo mode, the 2D selector is deactivated by default, since it makes no sense. Therefore, all the models have to be defined before activating the stereo mode.

A.3 Strategies for enabling interaction

Interactive environments require methods that allow computing collision detection tests between geometrical models, usually triangle meshes [61, 101]. Since interaction must be in real time, it is necessary to use methods that resolve the collision detection quickly. In addition, the methods must be robust so that the interactive environment remains consistent.

Once the collision is detected, the interactive environment must give a response [31]. In order to do that, it is important not only to detect the collision, but also to calculate its details. This involves calculating some collision parameters, such us overlapping triangles, distances, and nearest points. These calculations must be also computed quickly in order to allow a real time interaction. In the literature, there are several approaches to carry out some of these calculations. Each approach has their advantages and disadvantages; hence the use of one method or the other depends on each situation. The aim of this section is to describe and compare different strategies to calculate a detailed collision between two triangle meshes. These strategies have been implemented in order to test and compare them. These tests allow extracting the main benefits and constraints of each strategy and thus selecting the approach that fits best to the developed interactive application.

A.3.1 Proposed strategies

In order to compute collision detection, the following strategies have been considered: spatial decompositions, convex hulls and bounding volume hierarchies. The main objective is to determine their positive and negative features regarding the interaction with a variety of models.

Spatial decompositions

Hierarchical spatial decompositions allow increasing the efficiency of the collision detection, as they permit to reduce the space where the collision detection is performed. In order to check the collision between two models, the spatial decompositions associated with each model are compared recursively. Each recursive step tests whether two nodes A and B, one from each hierarchy, collide. If A and B do not collide, the collision test ends. Otherwise, the collision test is performed recursively on their children. If A and B are both leaf nodes, the collision detection test is performed with the triangles that are inside them. Some different strategies can be chosen while traversing the hierarchy. For example, if A and B do not collide, A can be tested with each of the children of B, B with each of the children of A, or each of the children of A with each of

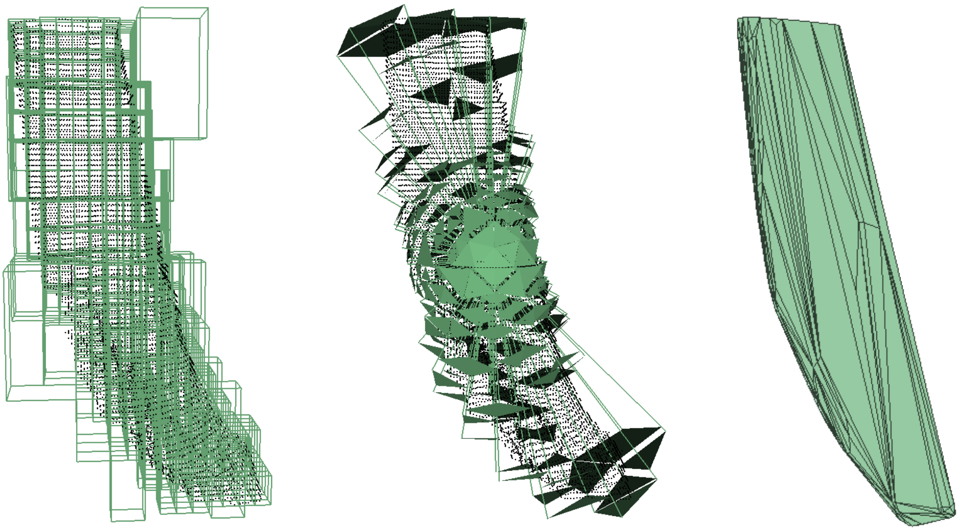

Figure A.9: Octree (left), tetra-tree (centre), and convex hull (right) associated to a point cloud.

the children of B. This choice modifies the efficiency of the method and depends on each specific case. To perform the triangle-triangle collision test, the point-in-solid algorithm by Feito-Torres [26] has been used. This method allows performing a point-in-solid test without any complex calculation. Furthermore, this method is robust because it can work with non-manifold polyhedra.

In this study, two spatial decompositions have been considered: octree [13] and tetra-tree [47]. The octree (Figure A.9, left) is one of the most used hierarchical data structures. This structure starts from a cuboid that contains all the triangles, usually a bounding box. In general, the root box is homogeneously divided into eight new boxes. The subdivision stops when a prefixed level is reached or the amount of triangles in a box is less than a given threshold. There are variations in the octree that can improve its performance. One of the most used ones proposes a change in the subdivision method of the boxes, so that instead of dividing homogeneously each box, the dividing point is selected so that it increases efficiency.

As mentioned in Chapter 5, the tetra-tree (Figure A.9, centre) is a hierarchical data structure that divides the entire space into eight equal parts named tetra-cones. Due to the adjustment obtained by computing the bounding tetrahedra associated with each tetra-cone [49], the tetra-tree fits the mesh better than other approaches. In addition, the tetra-tree allows classifying triangles quickly and robustly because it is based on barycentric coordinates.

Convex hull

The convex hull of a set of points is the minimum convex object that contains all the points (Figure A.9, right). Convex objects enable the application of specific algorithms in order to detect a collision. In general, these algorithms are more efficient, because they take advantage of the convex object features. Several collision detection algorithms are based on convex objects, such us the Gilbert-Johnson-Keerthi (GJK) algorithm [32] or some of its variants [11]. The quick hull method [4] was chosen to build the convex hull. Nevertheless, the main disadvantage of this technique is that convex hulls do not fit the mesh properly in most cases; hence the obtained results are not exact.

Bounding volume hierarchies

As an alternative to spatial decompositions and convex hulls, the use of bounding volume hierarchies [34, 54] was considered. Unlike spatial decompositions, bounding volumes do not fill all the space and their nodes can overlap between them. In this study, two different bounding volume hierarchies were considered: swept sphere hierarchies [56] and convex surface decomposition [23].

Larsen et al. [56] use swept sphere hierarchies to detect collisions, to identify overlapping triangles, to determine nearest points, and to calculate distances. They used three different types of swept volumes: a sphere, a cylinder with hemispherical caps, and a rounded box. These volumes depend on the swept primitive used: a point, a segment, and a rectangle. The efficiency of the algorithm depends on the volume used. The sphere allows building the data structure quickly but the rounded box usually fits the mesh better. Their algorithm utilizes a Bounding Volume Traversal Tree (BVTT) in order to perform a query. Each node of the BVTT represents a single collision test between two bounding volumes. Therefore, the BVTT is traversed to compute collision detection. In order to improve query performance, this method uses a priority directed search when the BVTT is being traversed. Moreover, temporal coherence is also taken into account. This method is implemented in the PQP library.

Table A.2: Features implemented by each method. Tetra-tree and octree use the Feito-Torres algorithm.

	Feito-Torres	GJK	PQP	SWIFT++
Collision	X	X	X	X
Overlapping triangles	X		X	
Distance		X	X	X
Tolerance			X	X
Contact features				X

Ehmann and Lin [23] present an approach to detect collisions and to calculate distances based on convex surface decomposition . This approach has three main steps. Firstly, the model is decomposed into convex parts. In order to achieve that, a method based on graph search is used. Secondly, a bounding volume hierarchy is constructed based on the previous decomposition. They propose a top-down approach in which each node of the hierarchy bounds all the geometry in its child nodes. For that reason, the primitives are divided recursively in order to obtain new convex patches. Lastly, queries can be executed once the hierarchy is constructed. In order to resolve queries between pairs of convex polyhedra, they use a distance minimization algorithm based on Voronoi marching [22]. Queries are accelerated using spatial and temporal coherence. This approach requires 2-manifold meshes whose triangles must be sorted counter-clockwise and it is implemented in the Speedy Walking via Improved Feature Testing (SWIFT++) library.

A.3.2 Implementation and results

In order to check their performance, the described strategies have been implemented under the same conditions. For that purpose, the same programming language (C++) and the same compiler (gcc) have been used. In addition, the qhull [4], PQP [56], and SWIFT++ [23] libraries have been used.

Spatial decompositions, as well as the Feito-Torres algorithm have been implemented without the use of any library. Moreover, implementations do not use any optimization and they do not consider spatial or temporal coherence. This means that the implementation of these two algorithms is less optimized than the other approaches, which are based on libraries. Therefore, the obtained results should not be as good as the implemented libraries. On the other hand, the collision detection based on the qhull implementation has not been tested because the results obtained are not as accurate as the rest of approaches; hence the obtained results should not be compared to them.

The PQP library [56], that is written in C++, can work with any model composed of triangles. Consequently, it is not necessary for the mesh to be closed or 2-manifold. This library allows detecting collisions and overlapping triangles, calculating distances and nearest points, and determining if two models are closer than a given threshold. The SWIFT++ library [23] is implemented in C++ and is divided into two parts. The main part allows computing intersections, calculating exact and approximate distances, and resolving contact determination queries between two or more objects. These objects must be convex and closed. The other part of the library enables the conversion of non-convex and open models into convex hierarchies that can be processed by the main part of the library. This library cannot work with non-2-manifold models or models that contain clockwise triangles. The features of all the tested algorithms are summarized in Table A.2.

Results

Some different tests were conducted to measure the efficiency of the proposed strategies. With that aim, six models with a polygonal complexity in the range from 7172 to 1085634 triangles were used (Table A.3). Some of the used models (Figure A.10) have holes and are not-2-manifold, thus the robustness of each strategy can be tested. The measured parameters were the pre-processing time, the size of the data structures, and the time to determine the collision detection and to calculate overlapping triangles. These tests were performed on a PC with a 2,8 GHz Intel i7 and 4 GB of RAM.

Firstly, some tests were carried out to measure the performance of the two tested spatial decompositions. Both data structures were built under the same conditions: same level and maximum threshold of triangles per node. Nevertheless, it should be noted that the octree is built by doing 8 subdivisions in each node, while the tetra-tree is built by doing only 4; hence the number of octree nodes is higher. The obtained results are shown in

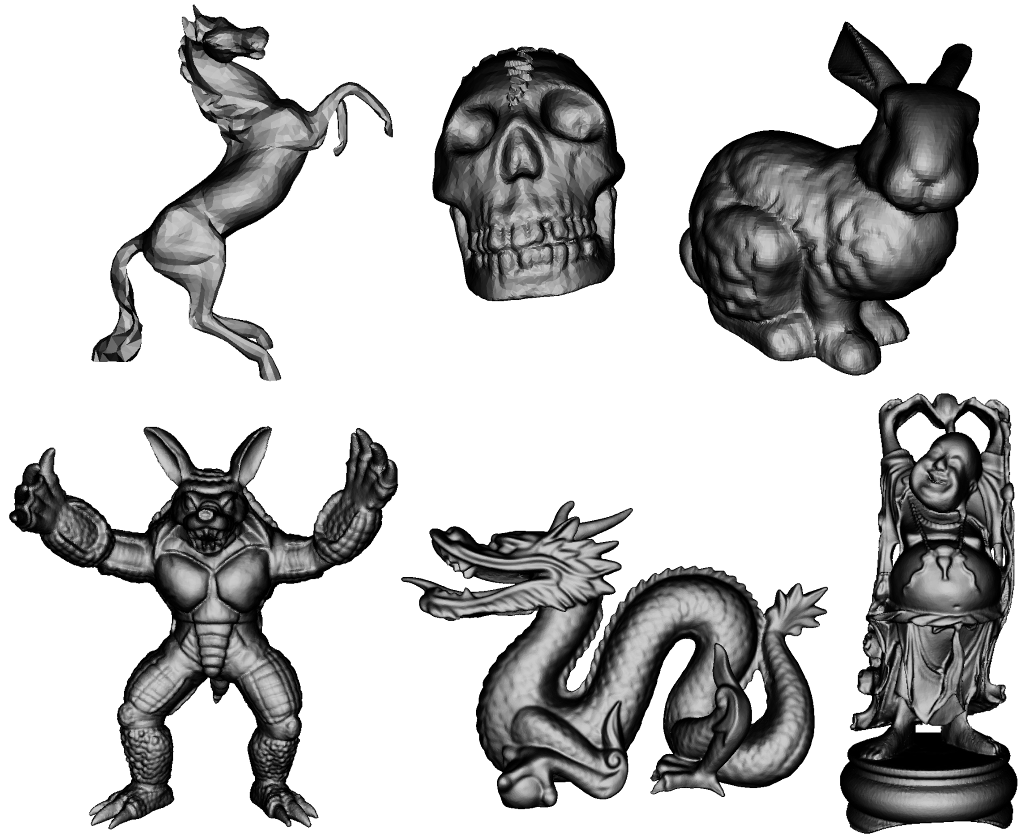

Figure A.10: Models used for testing. From left to right and from top to bottom: horse, skull, bunny, armadillo, dragon and buddha.

Table A.3. In general, the tetra-tree requires less pre-processing time, but needs more memory space.

The pre-processing time and size of the considered spatial decompositions have been measured (Table A.3). Moreover, the convex hull has also been tested (Table A.4). The size of each structure has been estimated theoretically because the real size depends on each machine and compiler. Therefore, the calculated size is approximate. In the case of spatial decompositions, the memory size has been calculated based on the theoretical size of each node. In order to achieve this, the number of nodes and the geometry associated to the spatial decomposition have been measured. It was considered that each node in the tetra-tree required 88 bytes: 16 bytes for children nodes (4 bytes for each pointer to a child node) and 72 bytes for tetra-cone vertices (24 bytes per vertex). Since the centroid

Table A.3: Pre-processing time (s) and size (MB) of the spatial decompositions.

			Tetra-tree		Octree	
Models	Vertices	Triangles	Pre-processing	Size	Pre-processing	Size
Horse	3582	7172	0,427	0,0351	0,496	0,0399
Skull	20002	40000	2,49	0,2608	4,329	0,261
Bunny	35947	69451	4,544	0,4211	8,744	0,3254
Armadillo	172974	345944	28,184	2,0322	44,579	1,6549
Dragon	437645	869928	70,25	4,8031	104,68	2,1975
Buddha	543652	1085634	84,704	5,5607	138,542	4,8717

Table A.4: Pre-processing time (s) and size (MB) of the convex hull.

	Convex Hull	
Models	Pre-processing	Size
Horse	0,202	0,0252
Skull	3,8	0,6182
Bunny	4,3	0,2916
Armadillo	15,7	0,3662
Dragon	42,5	0,4613
Buddha	130,4	1,1556

Table A.5: Pre-processing time (s) and size (MB) of the bounding volume hierarchies.

	PQP		SWIFT++	
Models	Pre-processing	Size	Pre-processing	Size
Horse	0,045	2,8726	0,599	2,9544
Skull	0,265	16,0216	8,094	20,6092
Bunny	0,47	27,818	8,148	31,6224
Armadillo	2,552	138,5654	380,121	162,3651
Dragon	6,678	348,9311	Not 2-manifold	Not 2-manifold
Buddha	8,327	435,5262	Not 2-manifold	Not 2-manifold

is a common vertex for all the tetra-cones, it is only necessary to store 3 vertices per tetra-cone. Additionally, the root node requires 340 bytes: 4 bytes for the origin pointer, 8 bytes for the pointers to the lists of vertices and triangles (4 bytes per pointer), 8 bytes for the number of vertices and triangles (4 bytes per number), 32 bytes for the pointer to the eight initial tetra-cones (4 bytes for each tetra-cone pointer), and 576 bytes for the vertices of the initial tetra-cones (3 vertices per tetra-cone and 24 bytes per vertex). The octree requires 80 bytes per node: 32 bytes for children nodes (4 bytes for each pointer to a child node) and 48 for the structure (2 vertices and 3 coordinates per vertex). In this case, the root node requires 72 bytes: 8 bytes for the pointers to the lists of vertices and triangles, 8 bytes for the number of vertices and triangles, 32 bytes for the pointers to the child nodes (4 bytes for each octant), and 24 bytes for the initial structure. In addition, both spatial decompositions need 4 bytes for each classified triangle.

PQP allows knowing the size of each structure at runtime; hence this procedure has been used to calculate the size of the structure associated with each model. It has not been possible to compute the size of the data structures used by SWIFT++. However, this library allows exporting the hierarchies to a file in order to facilitate reuse. The size of that file has been used to approximate the size of the hierarchical structure.

The results are shown in Table A.5. PQP is the option that requires less pre-processing time, but it needs a large amount of memory space. SWIFT++ also takes a long time and a lot of memory to build its structure. Moreover, it cannot be applied to all the tested models. The convex hull size is small because its structure is very simple.

Table A.6 shows the performance of each method to calculate one collision detection and the overlapping triangles. SWIFT++ is the fastest method that has been tested, but it cannot work with all the proposed models, because some of them are non-2-manifold. Moreover, it does not allow determining overlapping triangles. PQP has a reasonable good performance and its main advantage is that it can work with all the tested models. The methods based on spatial decompositions do not produce good results. The spatial decompositions have been built under the same conditions as in Table A.3.

Some tests to measure the performance of the tetra-tree in an interactive environment were carried out by Jiménez et al. [46]. Models of different sizes were used in order to run these tests. Their results showed that the tetra-tree required less pre-processing time than the octree, but it needed a little more space in memory. The tests showed that the tetra-tree enables the selection of inaccessible parts without the use of extra algorithms.

In summary, SWIFT++ is the fastest tested library, but it requires a lot of pre-processing time. On the other hand, PQP does not require much pre-processing time and it is quite efficient at calculating collisions. Therefore, the use of each library depends on the needs of each particular problem. Due to its robustness, PQP is a good option for systems that work with meshes that could not be topologically correct. This is the case of systems that work with meshes reconstructed from medical images, such us surgery simulators. Therefore, PQP is the best option to calculate detailed collisions in the developed application. Moreover, PQP could be a good solution for visualization applications, since meshes used in such applications do not need to be topologically correct. The SWIFT++ library is suitable for real time environments that require a good performance and can ensure that the topology of the models is correct. For that reason, this library could be used for working with solid models which have been generated by mathematical and boolean operations and thus conserve their topology. In all cases, the implemented systems must support a pre-processing step in which the data-structures are built.

A.4 Conclusions

In this Appendix, an application that contains tools to perform a detailed interaction with 3D models generated from CT image stacks is presented. This application is the basis for integrating the algorithms developed in this PhD dissertation with the goal of facilitating the work to medical specialists. Furthermore, a comparative study has been carried out with the goal of choosing the strategy that fits best to calculate a detailed collision detection between the reconstructed models in real-time.

Despite the fact that CT image stacks are commonly segmented before generating the 3D models, the developed application contains a tool to clean noise and to manually segment and label models. The application not only detects collisions between models, but also computes distances, nearest points and overlapping triangles. In order to enhance interaction, a multi-view interface has been developed and a stereoscopic spatial input device has been integrated into the application.

The developed application allows accomplishing the goal proposed in this PhD dissertation of researching and utilizing spatial decomposition and collision detection techniques to help specialists in the pre-operative planning of a fracture reduction surgery. In this context, the application could help specialists to better understand bone fractures. Moreover, the calculated collision detection parameters could ease the placing of the fragments in the correct position. As commented in Chapter 4, the contact zones calculated by the method described in this work enable the automatic reduction of bone fractures in the case of moderate fracture displacement. In that sense, the developed application allows carrying out a proper coarse alignment of the bone fragments in order to facilitate the application of the algorithm described in Chapter 4 for reducing displaced bone fractures.

Table A.6: Running time (s) of the computation of Collision detection (C.D.) and overlapping triangles (O.T.) between object A and B using the proposed strategies.

		Octree		Tetra-tree		PQP		SWIFT++
Model A	Model B	C.D.	O.T.	C.D.	O.T.	C.D.	O.T.	Collision
Buddha	Horse	0,5807	0,6236	0,2161	0,409	0,0034	0,0104	Not-2-Manifold
Buddha	Skull	1,8118	2,2317	0,2787	1,712	0,0076	0,017	Not-2-Manifold
Buddha	Bunny	1,2058	1,7777	0,2481	1,1735	0,0102	0,041	Not-2-Manifold
Buddha	Armadillo	2,026	4,5383	0,875	3,447	0,0106	0,4625	Not-2-Manifold
Dragon	Horse	0,4144	0,3593	0,1444	0,307	0,0014	0,0043	Not-2-Manifold
Dragon	Skull	0,8103	1,7643	0,2064	1,1583	0,0034	0,009	Not-2-Manifold
Dragon	Bunny	0,6852	1,4197	0,2195	0,8823	0,0045	0,0145	Not-2-Manifold
Dragon	Armadillo	1,1456	3,622	0,573	2,69	0,0087	0,0313	Not-2-Manifold
Armadillo	Horse	0,0923	0,32	0,0892	0,22	0,0013	0,004	≈ 0
Armadillo	Skull	0,2605	1,5928	0,1108	1,184	0,0026	0,0054	≈ 0
Armadillo	Bunny	0,234	1,483	0,1673	0,9248	0,0033	0,009	≈ 0
Armadillo	Armadillo	0,588	3,3435	0,4418	3,207	0,0037	0,016	≈ 0

Appendix B

Introducción

En los últimos años, el uso de técnicas computacionales para asistir procedimientos quirúrgicos se ha incrementado de manera considerable. Este tipo de técnicas provee herramientas para entrenar cirujanos nóveles, para mejorar la información disponible durante la cirugía, o incluso para ayudar en la planificación pre-operatoria de una intervención médica. Uno de esos procedimientos quirúrgicos es la reducción de una fractura ósea, que puede ser definida como una condición médica en la que se rompe la continuidad del hueso.

El tratamiento de una fractura ósea es una tarea compleja. En el caso de fracturas simples, que son aquellas en las que el hueso queda dividido en dos fragmentos por una única línea de fractura, una imagen de rayos X en la mayoría de los casos es suficiente para planificar la cirugía correctamente. Sin embargo, la planificación de una fractura compleja normalmente requiere utilizar otras técnicas de escaneado, con el fin de obtener modelos 3D de las estructuras óseas que posibiliten la identificación del número de fragmentos óseos, su posición, y su orden de colocación, así como la elección de los elementos de fijación más adecuados. El uso de sistemas asistidos por ordenador puede ayudar en el proceso mediante la identificación de los fragmentos óseos, la habilitación de la interacción con modelos virtuales de fragmentos, el cálculo de la reducción de la fractura, o el análisis de diferentes configuraciones de elementos de fijación. La utilización de técnicas asistidas por ordenador en la planificación pre-operatoria permite a los especialistas disminuir el tiempo de intervención y eludir potenciales interpretaciones erróneas, con la consecuente disminución del tiempo de recuperación del paciente [95, 93].

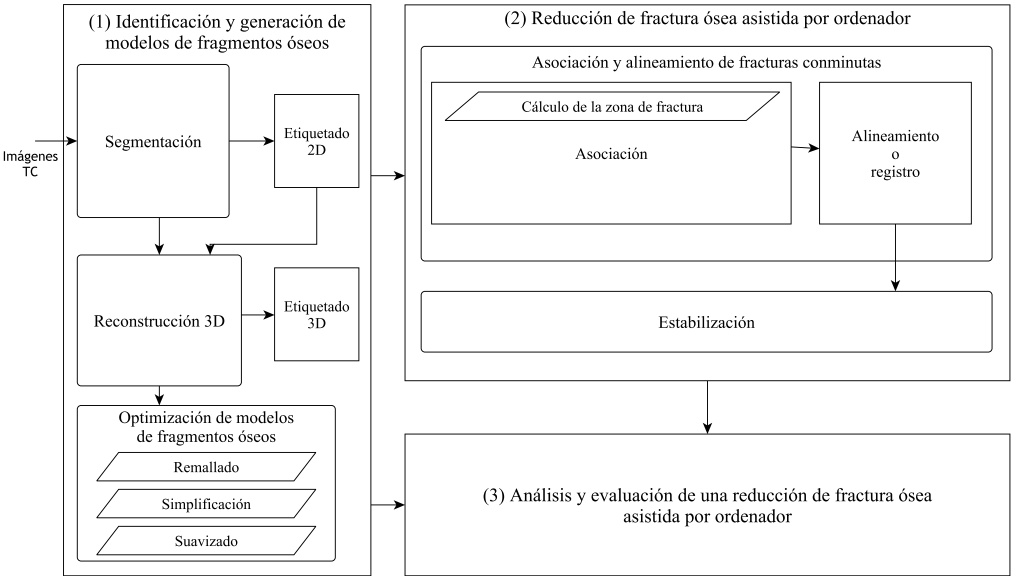
(1) Identificación y generación de modelos de fragmentos óseos
Imágenes TC
Segmentación
Etiquetado 2D
Reconstrucción 3D
Etiquetado 3D
Optimización de modelos de fragmentos óseos
Remallado
Simplificación
Suavizado
(2) Reducción de fractura ósea asistida por ordenador
Asociación y alineamiento de fracturas conminutas
Cálculo de la zona de fractura
Asociación
Alineamiento o registro
Estabilización
(3) Análisis y evaluación de una reducción de fractura ósea asistida por ordenador

Figure B.1: Vista detallada de las etapas de la planificación pre-operatoria asistida por ordenador de una reducción de fractura ósea.

Planificación pre-operatoria de una reducción de fractura ósea asistida por ordenador

La reducción de una fractura ósea asistida por ordenador puede dividirse principalmente en tres etapas: (1) identificación de fragmentos óseos a partir de imágenes médicas y generación de modelos virtuales de estos, (2) cálculo de la reducción y de la estabilización de la fractura, (3) y análisis de los resultados obtenidos desde un punto de vista tanto geométrico como mecánico (Figura B.1).

La generación de una representación virtual de los huesos y fragmentos es una etapa artificial necesaria para obtener modelos útiles para las siguientes fases del proceso. En función de los objetivos, pueden ser necesarias distintas representaciones; por ejemplo, un modelo volumétrico para visualización, una nube de puntos para interacción, o una malla de triángulos para llevar a cabo operaciones geométricas. Esta etapa comienza con la extracción de tejido óseo a partir de imágenes médicas utilizando técnicas de segmentación y etiquetado. Después, puede ser necesario llevar a cabo una reconstrucción 3D para obtener modelos de las estructuras óseas. Estos modelos pueden requerir la utilización de técnicas de optimización con el propósito de producir representaciones adecuadas para las siguientes etapas.

El objetivo principal de la reducción de fracturas asistida por ordenador es posicionar y alinear los fragmentos de hueso con el fin de estabilizar la estructura ósea. Este procedimiento requiere resolver una variedad de problemas que dependen tanto del tipo de hueso como de fractura. En el caso de una fractura simple, la reducción consiste en alinear los dos fragmentos para que recuperen su posición original. Si la fractura genera más de dos fragmentos, se requiere llevar a cabo un procedimiento previo al alineamiento para resolver el puzle. Además, en la mayoría de los casos es necesario estabilizar la fractura mediante el uso de elementos de fijación tales como placas y tornillos.

El análisis de la reducción posibilita la evaluación de los resultados obtenidos. Este análisis puede llevarse a cabo de diferentes maneras: obteniendo la precisión geométrica de la reducción, utilizando análisis de elementos finitos o técnicas similares para probar la estabilidad mecánica de la fractura, o incluso evaluando el proceso mediante su comparación con la intervención real.

Esta tesis está principalmente centrada en la segmentación y el etiquetado de fragmentos óseos a partir de imágenes médicas generadas mediante TC, en la generación de modelos 3D de fragmentos óseos, y en el cálculo de la reducción de fracturas, excluyendo la utilización de elementos de fijación. Además, se ha realizado un análisis geométrico de los resultados obtenidos.

Objetivos

Considerando las etapas descritas en la sección anterior, el objetivo principal de esta tesis es investigar y desarrollar métodos que ayuden a los especialistas médicos a identificar tejido óseo fracturado y a reducir fracturas óseas. Dado que la reducción interactiva de una fractura ósea es un proceso costoso en tiempo, los métodos desarrollados deberían tratar de reducir la actuación del especialista en todas las etapas del proceso. Finalmente, deben proponerse métodos de evaluación con el fin de medir la calidad de los resultados obtenidos. Los objetivos perseguidos en esta tesis son:

- Investigar y desarrollar métodos que posibiliten la segmentación y el etiquetado de fragmentos óseos a partir de imágenes médicas generadas mediante TC.
- Investigar y desarrollar algoritmos para facilitar la automatización del proceso de reducción de fracturas.
- Investigar y desarrollar nuevos métodos de generación de modelos geométricos que representen fragmentos óseos.
- Analizar y evaluar los resultados obtenidos por los métodos propuestos.
- Investigar y utilizar descomposiciones espaciales y técnicas de interacción y detección de colisiones que ayuden a los especialistas en la planificación de una reducción de fractura ósea.

Organización de esta tesis

Esta tesis está organizada en torno al proceso asistido por ordenador de reducción de una fractura ósea. En primer lugar, el capítulo 2 realiza una revisión de los trabajos propuestos actualmente. Después, cada capítulo desde el 3 al 5 está centrado en una etapa distinta del proceso. Finalmente, el capítulo 6 esboza las conclusiones y el trabajo futuro. El apéndice A presenta las herramientas interactivas desarrolladas con el propósito de facilitar el entendimiento de las fracturas, y posibilitar un alineamiento aproximado de los fragmentos óseos. Los siguientes párrafos describen la estructura del documento.

El capítulo 2 revisa los trabajos actualmente propuestos en la literatura relacionados con la planificación pre-operatoria asistida por ordenador de la reducción de una fractura ósea. En primer lugar, se analizan y clasifican los enfoques propuestos para identificar hueso fracturado. Los trabajos más recientes para segmentar hueso sano también han sido revisados con el fin de evaluar si las técnicas normalmente usadas para este tipo de hueso son también aplicables a hueso fracturado. Además, se han descrito los métodos comúnmente usados para generar y optimizar modelos 3D de fragmentos óseos. A continuación, se han revisado y clasificado las técnicas asistidas por ordenador actualmente propuestas para reducir fracturas óseas. Finalmente, se han analizado los métodos a día de hoy utilizados para analizar y evaluar la planificación pre-operatoria asistida por ordenador de la reducción de una fractura ósea.

El capítulo 3 describe los principales aspectos que deben considerarse en la identificación de tejido óseo, y los problemas que adicionalmente surgen cuando el hueso está fracturado. La identificación de hueso fracturado incluye, no sólo la segmentación de tejido óseo, sino también el etiquetado de los fragmentos de hueso y la detección de las regiones fracturadas. El etiquetado de los fragmentos óseos implica la identificación de cada fragmento de manera separada. Además, este capítulo presenta un nuevo método para segmentar y etiquetar fragmentos óseos a partir de imágenes médicas generadas mediante TC. El método está basado en crecimiento de regiones 2D y requiere una interacción mínima por parte del especialista. El procedimiento propuesto permite, durante el proceso de segmentación, separar fragmentos erróneamente conectados. El método ha sido comparado con los enfoques actualmente utilizados en la literatura para segmentar hueso sano y fracturado, obteniendo mejores resultados en la mayoría de aspectos. Por otra parte, el método también ha sido evaluado usando varios casos clínicos de diferentes huesos y tipos de fractura con resultados satisfactorios.

El capítulo 4 introduce un método automático para calcular la zona de contacto entre dos fragmentos óseos. El método no requiere la generación previa de mallas 3D para representar los fragmentos ya que tan sólo necesita nubes de puntos como entrada. El método desarrollado facilita la aplicación de algoritmos de resolución de puzles debido a que no obtiene la totalidad de la zona de fractura, sino el área de contacto entre cada par de fragmentos. Por lo tanto, no es necesario encontrar correspondencias entre zonas de fractura y los fragmentos pueden ser alineados de dos en dos. Por otra parte, también se presenta un algoritmo de alineamiento con el propósito de reducir fracturas óseas de manera automática utilizando las zonas de contacto calculadas. El método propuesto ha sido aplicado con éxito en la reducción de distintas fracturas óseas. En primer lugar, el método ha sido probado en fracturas de la zona del tobillo. Con el objetivo de medir el rendimiento del método propuesto, se han calculado tanto el área de contacto como el solapamiento entre fragmentos. En segundo lugar, el método ha sido aplicado a fracturas de húmero en cadáveres que han permitido comparar los resultados con la realidad observada.

El capítulo 5 presenta un estudio que evalúa el rendimiento y la idoneidad de destacados algoritmos de reconstrucción para la generación, a partir de imágenes médicas, de mallas de triángulos que representan fragmentos óseos. Los experimentos llevados a cabo han permitido extraer los beneficios y limitaciones de cada método al aplicarlo en diferentes huesos y tipos de fractura. Además, se ha analizado la calidad de los resultados obtenidos, poniendo el foco en las características especiales de los datos de fracturas óseas obtenidos mediante TC. Teniendo en cuenta los resultados del estudio, la reconstrucción de Poisson genera los mejores modelos para ser usados en aplicaciones de visualización, y los modelos generados con MC son los más adecuados para ser utilizados en la reducción de una fractura ósea asistida por ordenador. Por otro lado, se presentan los resultados iniciales de un estudio preliminar para el desarrollo de un novedoso método de generación de mallas de triángulos a partir de imágenes médicas utilizando una descomposición del espacio. Además, como pre-procesamiento se utiliza un nuevo enfoque para extraer los puntos de los contornos exteriores de las regiones segmentadas, y estimar los vectores normales en cada punto. En este estudio preliminar se ha utilizado una descomposición del espacio denominada tetra-tree. No obstante, se podría haber utilizado cualquier otra estructura espacial para este estudio. El nuevo método aborda la generación de la malla usando un enfoque basado en el paradigma divide y vencerás, con el fin de minimizar los problemas que surgen al coser contornos consecutivos. El objetivo principal de este estudio inicial es determinar si una descomposición del espacio puede ayudar en la reconstrucción de la malla, permitiendo el análisis de las ventajas e inconvenientes derivados de su utilización.

El apéndice A presenta una aplicación que contiene herramientas para llevar a cabo una interacción detallada con modelos 3D de fragmentos óseos obtenidos a partir de TC. Esta aplicación tiene como objetivo ser la base para la integración, tanto de los algoritmos desarrollados en esta tesis, como de las nuevas herramientas que se desarrollarán en el futuro. Con el propósito de generar modelos 3D a partir de las imágenes médicas obtenidas mediante TC, la aplicación utiliza MC ya que los experimentos que se llevaron a cabo en el capítulo 5 demostraron que obtiene los mejores modelos para ser aplicados en procedimientos médicos asistidos por ordenador. A pesar de que las imágenes médicas normalmente son segmentadas antes de generar los modelos 3D, la aplicación desarrollada contiene herramientas para eliminar ruido, y para segmentar y etiquetar manualmente los modelos. La aplicación calcula en tiempo real triángulos solapados, distancias, y puntos más cercanos entre modelos. Con ese fin, se realizó un estudio comparativo previo para determinar la mejor estrategia de detección de colisiones que permite tratar con las características de los modelos obtenidos a partir de imágenes de TC. Finalmente, el sistema *Leonar3Do* ha sido incorporado en la aplicación para mejorar la interacción y proveer visualización estéreo. La aplicación desarrollada puede ser utilizada en la planificación pre-operatoria de una reducción de fractura ósea, ya que facilita el entendimiento de la fractura, y permite realizar un alineamiento aproximado de los fragmentos óseos.

Appendix C

Conclusiones

Las técnicas asistidas por ordenador pueden proveer soporte tecnológico en la planificación pre-operatoria del proceso de reducción de fracturas óseas, reduciendo riesgos durante la cirugía, y disminuyendo el tiempo de recuperación del paciente. En esta tesis se han revisado las técnicas y enfoques propuestos para ayudar a los especialistas médicos en este proceso, desde la generación de modelos de huesos y fragmentos, hasta el análisis de la composición final. Estos trabajos han sido resumidos y clasificados, destacando sus principales ventajas y carencias. La revisión llevada a cabo ha puesto de manifiesto que ninguna de las etapas del proceso de reducción de fracturas óseas asistido por ordenador está completamente resuelta, ni la identificación y generación de fragmentos, ni la propia reducción de la fractura. Como consecuencia, este campo de investigación sigue abierto y afronta importantes retos a medio plazo. Esta tesis ha presentado nuevos métodos para identificar fragmentos óseos a partir de imágenes TC, para calcular la zona de contacto entre fragmentos, para computar la reducción de la fractura, y para generar mallas de triángulos que representen fragmentos óseos. Además, se han desarrollado algunas herramientas para ayudar a los especialistas a manipular modelos de fragmentos óseos construidos a partir de imágenes médicas.

Identificación y generación de modelos de fragmentos óseos

La principal dificultad que surge en esta etapa es la identificación de cada uno de los fragmentos óseos de manera determinística. En la actualidad, este proceso requiere la interacción manual del especialista y en ocasiones conocimiento experto. Este problema constituye una línea de investigación abierta que requiere métodos de segmentación y etiquetado más precisos, y en la cual sería recomendable el uso de conocimiento previo. Con el fin de identificar cada uno de los fragmentos óseos que componen una fractura, se requiere tanto su etiquetado, como la separación de fragmentos erróneamente conectados. Por otro lado, el procedimiento clave en la etapa de segmentación de la mayoría de los métodos actuales es la selección de umbrales y semillas. Actualmente, la fase de etiquetado se lleva a cabo de forma manual en la mayoría de los casos. Como resultado, tanto la segmentación como el etiquetado pueden requerir un procesamiento posterior. Los enfoques basados en umbralización no clasifican los fragmentos óseos, por lo que estos tienen que ser etiquetados tras el proceso de segmentación. El enfoque presentado en esta tesis aprovecha la colocación de las semillas, requeridas para segmentar cada fragmento, para también etiquetarlo. Otros enfoques en la literatura también resuelven este problema utilizando métodos basados en semillas [38, 29]. En todos estos casos, las semillas deberían ser colocadas por un experto. Otros estudios proponen métodos para etiquetar las regiones óseas automáticamente, pero después un experto debe evaluar los resultados y decidir a qué hueso o fragmento pertenece cada región. Idealmente, todos los fragmentos óseos deberían segmentarse automáticamente, y los fragmentos más sencillos deberían ser identificados sin intervención del usuario. Después, en los casos más complejos, el experto podría decidir el hueso al que pertenece cada fragmento.

Dada la complejidad del tejido trabecular, la selección de umbrales de intensidad es un procedimiento difícil, ya que es complicado determinar estos valores incluso de forma manual, y cada loncha puede requerir un umbral diferente. Además, resulta particularmente difícil establecer el umbral para segmentar tejido óseo cerca de las articulaciones. El método presentado en esta tesis utiliza un modelo de regresión lineal para calcular el umbral en cada loncha. A pesar de que los umbrales calculados obtienen buenos resultados en la mayoría de los casos, el modelo necesita ser reconstruido para cada conjunto de datos. La mayor parte de los métodos propuestos requieren que el usuario especifique manualmente el umbral de intensidad. Lo ideal sería que, en todos los casos, los valores umbrales fuesen seleccionados automáticamente a partir de la información disponible en el conjunto de imágenes.

Tal y como se mencionó en la sección 3.1, durante el proceso de identificación puede ser necesario separar fragmentos óseos. La baja resolución de la imagen puede causar que fragmentos muy cercanos aparezcan unidos. Esto es especialmente común en fracturas producidas por fuertes impactos. Para abordar este problema se han propuesto diferentes enfoques: herramientas interactivas [38, 98], etiquetado 3D de componentes conectados y cortes de grafos [29], re-segmentación [57], o incluso comparaciones con modelos sanos [96]. No obstante, la separación manual o semi-automática es costosa, por lo que sería importante su automatización para posibilitar el ahorro de tiempo. Una posible solución sería mejorar el método de segmentación de modo que no obtenga fragmentos conectados. El método presentado en esta tesis separa fragmentos durante la segmentación utilizando semillas adicionales. La automatización completa de este proceso sería provechosa ya que evitaría el uso de métodos adicionales tras el proceso de segmentación. Sin embargo, la usual baja resolución de las imágenes médicas generadas mediante TC hace muy complicada dicha automatización. Un enfoque alternativo consistiría en implementar un método para abordar este problema tras la etapa de segmentación. De este modo, el resultado completo de la segmentación estaría disponible y, por tanto, se podría utilizar información adicional extraída de las lonchas adyacentes ya segmentadas.

La utilización de una tecnología de adquisición de datos más precisa podría evitar, en la mayoría de los casos, que fragmentos óseos aparezcan unidos. La manera más común de generar imágenes 3D en las que se pueda distinguir tejido óseo, es utilizar TC. Este procedimiento conlleva un importante impacto radiactivo en el paciente. Además, en los últimos años existe una tendencia a utilizar imágenes de microtomografías computarizadas para añadir precisión extra a las regiones óseas. Este tipo de imágenes incrementa la radiación [88]. Por lo tanto, a pesar de actualmente ser la mejor opción para distinguir tejido óseo, las imágenes generadas mediante TC poseen este inconveniente ineludible. De ahí que el desarrollo de métodos alternativos de escaneado, que mejoren las imágenes médicas sin incrementar la radiación, se ha convertido en un reto para la comunidad científica. Aunque el principal propósito de las imágenes médicas es el diagnóstico clínico, estas son también la entrada de técnicas asistidas por ordenador que pueden conducir a la reducción del tiempo de cirugía.

Por otro lado, algunos trabajos tratan de segmentar tejido óseo a partir de imágenes de resonancia magnética o rayos X. Estas técnicas requieren información extra porque, en esas imágenes, el tejido óseo no puede identificarse con la misma facilidad que utilizando imágenes médicas generadas mediante TC. Los métodos propuestos para identificar tejido óseo en ese tipo de imágenes están basados en atlas previamente construidos a partir de imágenes de TC [39], o están aplicados a áreas óseas muy específicas [45, 37]. Sin embargo, no se han propuesto estudios para identificar hueso fracturado a partir de esos tipos de imagen médica.

Con respecto a la etapa de reconstrucción 3D, el algoritmo de MC [63] es un estándar de facto debido a su simplicidad para generar modelos geométricos. No obstante, este método tiene algunas desventajas que representan un reto en la actualidad: la gran cantidad de geometría generada, y el ruido obtenido a causa de la complejidad del tejido trabecular. En esta tesis se han probado algunos otros algoritmos de generación de mallas (reconstrucción de Poisson [53], Ball-pivoting [6], y Algebraic Point Set Surface (APSS) [36, 35]) para tratar de generar modelos de fragmentos óseos, extrayendo sus principales beneficios y limitaciones. En las pruebas realizadas, la reconstrucción de Poisson obtuvo las mejores mallas para visualización y MC a partir de regiones fue la mejor opción para generar mallas que puedan ser usadas en procedimientos médicos asistidos por ordenador. Además, esta tesis presenta los resultados iniciales de un estudio preliminar para el desarrollo de un método innovador de generación de modelos 3D de fragmentos óseos mediante el cosido de los contornos externos de las regiones segmentadas. Con el fin de abordar los casos especiales que surgen durante el cosido, el método utiliza un enfoque basado en el paradigma divide y vencerás utilizando una descomposición del espacio denominada tetra-tree. Aunque el método obtiene resultados prometedores, se han detectado algunos aspectos que deberían ser mejorados para poder generar modelos más complejos. El desarrollo de una descomposición del espacio alternativa, que se ajuste mejor a los modelos de fragmentos óseos, constituye una línea de investigación abierta. Por otro lado, el método propuesto para calcular la reducción de fracturas óseas utiliza nubes de puntos como entrada, ya que esta estructura de datos se genera fácilmente, y dicho método no requiere la información geométrica que proveen las mallas de triángulos. No obstante, las nubes de puntos no evitan los problemas para visualizar, gestionar, e interactuar con estos modelos de gran tamaño.

El principal inconveniente de utilizar métodos de optimización es la diferencia entre los modelos procesados y los contornos de las regiones originalmente segmentadas. Esto puede causar que las características de algunos fragmentos óseos, como las zonas de fractura, lleguen a ser difíciles de detectar. Paradójicamente, los métodos de suavizado son necesarios para eliminar artefactos no deseados en los modelos y, al mismo tiempo, estos métodos pueden difuminar las fronteras y por tanto dificultar la extracción, identificación, y posterior reducción de la fractura. Sin embargo, estas técnicas son muy útiles en el proceso de visualización donde se puede seguir una estrategia de nivel de detalle con el fin de adaptar los requisitos computacionales a la demanda del especialista.

Reducción de fractura ósea asistida por ordenador

La revisión llevada a cabo en la sección 2.2.2 nos permite discutir acerca de los enfoques existentes en la literatura para la reducción de fracturas óseas conminutas asistida por ordenador. Tal y como se muestra en la tabla 2.6, la mayor parte de los métodos propuestos está centrado en huesos largos (tibia, peroné, fémur y húmero) [107, 105, 68, 114, 75, 97, 1, 30]. En un hueso largo, la diáfisis es cilíndrica y está completamente rodeada de tejido cortical. Por otro lado, en la epífisis la zona cortical es muy fina e incluso el tejido trabecular puede aparecer en la zona exterior del hueso. Esta información adicional ha favorecido el desarrollo de técnicas asistidas por ordenador para ayudar al especialista en la reducción de fracturas de hueso largo. Por el contrario, con el propósito de reducir fracturas de huesos irregulares tan sólo han sido propuestas herramientas interactivas [19, 106, 28, 43]. Como excepción, Chowdhury et al. [17] presentaron un enfoque para reducir fracturas craneofaciales mediante el cálculo, la asociación y el alineamiento de las zonas de fractura, y Kato [52] propuso reducir fracturas de acetábulo utilizando una plantilla.

Los métodos automáticos requieren información extra con el fin de evitar la interacción del usuario. La mayor parte de los enfoques propuestos utilizan plantillas para recolocar fragmentos óseos [68, 1, 52]. Otros enfoques aprovechan las características diferenciables de un hueso específico. Winkelbach et al. [107] propusieron un método para reducir fracturas conminutas de fémur de manera automática. Con ese propósito, los autores hacen uso de las características especiales del hueso largo mencionadas en el párrafo anterior. Chowdhury et al. [17] presentaron un enfoque para reducir automáticamente fracturas craneofaciales. En este caso, los autores aprovechan una clasificación previa de los fragmentos como terminales o no terminales, basada en la presencia o ausencia de cóndilos. En la actualidad, no existe un método automático de reducción de fracturas que pueda ser aplicado a cualquier hueso. Esta tesis ha presentado un método para calcular la zona de contacto entre cada par de fragmentos de forma automática. Las zonas de contacto calculadas posibilitan la reducción de fracturas óseas complejas. En cuanto al alineamiento final, la mayoría de los métodos propuestos utilizan una adaptación del algoritmo iterativo del punto más cercano (ICP). La principal deficiencia de este método es la posible obtención de soluciones locales. Por lo tanto, el método requiere un alineamiento previo aproximado de los fragmentos óseos para evitar ese problema. Con ese fin, se puede utilizar la aplicación presentada en el apéndice A para realizar una alineamiento previo adecuado de los fragmentos. El desarrollo de métodos de registro más robustos, que reduzcan la obtención de soluciones locales, llevaría a la consecución de enfoques más automáticos.

La estabilización de una reducción de fractura es una etapa clave en el proceso de garantizar la fijación de los fragmentos a largo plazo. En la actualidad, la estabilización automática es un campo inexplorado. Sin embargo, se han propuesto algunas soluciones interactivas para ayudar al especialista a planificar la cirugía y acelerar todo el proceso. Esta falta de soluciones generales es debida a las dificultades técnicas intrínsecas, a la necesidad de conocimiento experto, y a la exigencia de un análisis preciso de la reducción. Durante la simulación, el material de osteosíntesis tiene que definirse de acuerdo a la fractura. Si las placas se diseñan de manera virtual, luego deben ser específicamente fabricadas para el paciente. A pesar de que algunos de los enfoques propuestos han sido probados en casos clínicos, la fabricación de implantes es todavía un proceso costoso. Por lo tanto, en la mayoría de los casos las soluciones actuales proponen adaptar implantes estándar.

Análisis y validación del resultado obtenido al realizar una reducción de fractura ósea asistida por ordenador

Tal y como se comentó en la sección 2.1.2, la reducción asistida por ordenador de fracturas óseas complejas requiere, a partir de imágenes médicas generadas mediante TC, la generación de modelos 3D de fragmentos óseos. Estos modelos virtuales se generan normalmente utilizando un enfoque basado en MC. Después, en la mayoría de los casos se elimina el ruido y se simplifican los modelos obtenidos. A pesar de que estos modelos son complejos, no se han propuesto estudios para evaluar su calidad. Por otro lado, no existe un criterio estándar para evaluar, ni la precisión de la composición llevada a cabo en la reducción, ni la calidad de los modelos. Algunos autores han propuesto enfoques diferentes para medir la calidad de la reducción obtenida, por lo que los métodos propuestos no pueden ser comparados fácilmente. En la mayoría de los casos, los autores utilizan huesos generados de forma artificial para evaluar sus métodos de reducción propuestos. En estos casos, los resultados obtenidos por distintos autores no pueden compararse ya que utilizan distintos huesos para llevar a cabo los tests: huesos de cadáveres y huesos artificiales de distintos materiales. Las características especiales de los huesos largos también han sido utilizadas para comprobar la precisión de la reducción. Estas características posibilitan la comparación de diferentes métodos pero la restringen a fracturas de huesos largos. Los métodos basados en registro pueden utilizar el error cuadrático medio para evaluar la precisión del alineamiento final de las superficies de fractura. Esta medida sólo evalúa esta etapa del proceso y únicamente es aplicable en métodos que utilizan técnicas de registro. Con el fin de facilitar las comparativas, el método de reducción de fracturas propuesto en esta tesis ha sido evaluado utilizando parámetros normalmente usados en la literatura. En concreto, se han calculado el solapamiento y las zonas de contacto en los casos clínicos, y los errores de traslación y rotación en los casos de cadáveres, ya que en estos se conocía previamente la realidad observada.

Investigación futura

Los siguientes pasos en esta línea de investigación deberían permitir una automatización de la cirugía tal y como se ha introducido en otras disciplinas médicas [50, 79]. Esta automatización posibilitaría la cirugía guiada por ordenador mediante la utilización de robots. A este respecto, los métodos propuestos para calcular la reducción de fracturas óseas han sido aplicados en la planificación de la cirugía. La mejora de las técnicas actualmente propuestas facilitaría el desarrollo de sistemas robóticos para asistir, o incluso para llevar a cabo la intervención real. Hasta donde tenemos conocimiento, el desarrollo comercial de estas tecnologías no está muy extendido. La investigación de nuevos métodos para automatizar el proceso de reducción de fracturas y el análisis de los resultados obtenidos, podría contribuir al desarrollo y mejora de este tipo de soluciones comerciales.

Teniendo en cuenta la discusión anterior, el proceso de reducción de fracturas asistido por ordenador puede mejorarse mediante la investigación en los siguientes temas:

- Identificar fragmentos óseos a partir de imágenes generadas mediante TC sin necesidad de la intervención del usuario. Esto incluye el cálculo automático de umbrales y la colocación de semillas, si es que se requieren. La ayuda del experto debería reducirse tanto como fuese posible.
- Desarrollar nuevas técnicas para evitar que distintos fragmentos óseos aparezcan unidos tras el proceso de segmentación. Esto puede lograrse mejorando los algoritmos de segmentación, e investigando nuevas tecnologías de adquisición que mejoren la resolución de la imagen sin incrementar la radiación en el paciente.
- Completar y generalizar el método propuesto para producir modelos 3D de fragmentos óseos. Adicionalmente, el método puede ser evaluado con más casos clínicos, no necesariamente de fracturas óseas.
- Investigar nuevas técnicas de optimización para mejorar los modelos 3D generados sin modificar la información geométrica originalmente extraída de los datos médicos.
- Desarrollar algoritmos para llevar a cabo un alineamiento aproximado de los fragmentos de manera automática, y sin la necesidad de utilizar plantillas o requerir información adicional de la forma del hueso.
- Calcular los dispositivos de fijación necesarios para estabilizar una fractura, así como posicionarlos automáticamente.
- Desarrollo de una nueva descomposición del espacio que se ajuste mejor a la forma de cualquier fragmento óseo, y que sea capaz de subdividirse más en las zonas más detalladas.
- Evaluar la calidad de los modelos 3D de fragmentos óseos generados a partir de imágenes médicas.
- Desarrollar procedimientos estándar con el fin de evaluar la calidad de una reducción de fractura, tanto desde un punto de vista geométrico, como mecánico.

- Generar de manera virtual fracturas complejas y realistas de diferentes huesos, con el fin de facilitar la evaluación de los métodos desarrollados.

BIBLIOGRAPHY

[1] T. Albrecht and T. Vetter. Automatic fracture reduction. In J. A. Levine, R. R. Paulsen, and Y. Zhang, editors, *Mesh Processing in Medical Image Analysis 2012*, volume 7599 of *Lecture Notes in Computer Science*, pages 22–29. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[2] P. Alliez, G. Ucelli, C. Gostman, and M. Attene. Recent advances in remeshing of surfaces. In L. Floriani and M. Spagnuolo, editors, *Shape Analysis and Structuring*, Mathematics and Visualization, pages 53–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[3] M. S. Aslan, A. Ali, H. Rara, and A. A. Farag. An automated vertebra identification and segmentation in CT images. In *2010 IEEE International Conference on Image Processing*, pages 233–236. IEEE, 2010.

[4] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, dec 1996.

[5] S. Battiato, G. M. Farinella, G. Impoco, O. Garretto, and C. Privitera. Cortical bone classification by local context analysis. *Computer Vision/Computer Graphics Collaboration Techniques*, 4418:567–578, 2007.

[6] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

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

[8] S. M. Bhandarkar, A. S. Chowdhury, Y. Tang, J. C. Yu, and E. W. Tollner. Computer vision guided virtual craniofacial reconstruction. *Computerized Medical Imaging and Graphics : the Official Journal of the Computerized Medical Imaging Society*, 31(6):418–427, 2007.

[9] J. Blanchette and M. Summerfield. *C++ GUI Programming with Qt 4 (2nd Edition) (Prentice Hall Open Source Software Development Series)*. Prentice Hall, 2008.

[10] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[11] S. Cameron. Enhancing GJK: computing minimum and penetration distances between convex polyhedra. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 3112–3117. IEEE, 1997.

[12] R. Campos, R. Garcia, and T. Nicosevici. Surface reconstruction methods for the recovery of 3D models from underwater interest areas. In *OCEANS 2011 IEEE - Spain*, pages 1–10. IEEE, 2011.

[13] H. H. Chen and T. S. Huang. A survey of construction and manipulation of octrees. *Computer Vision, Graphics, and Image Processing*, 43(1):112–113, 1988.

[14] Y. Chen, M. Qiang, K. Zhang, H. Li, and H. Dai. Novel computer-assisted preoperative planning system for humeral shaft fractures: report of 43 cases. *The International Journal of Medical Robotics + Computer Assisted Surgery : MRCAS*, 2014.

[15] A. S. Chowdhury, S. M. Bhandarkar, R. W. Robinson, and J. C. Yu. Virtual craniofacial reconstruction from computed tomography image sequences exhibiting multiple fractures. In *2006 International Conference on Image Processing*, pages 1173–1176. IEEE, 2006.

[16] A. S. Chowdhury, S. M. Bhandarkar, R. W. Robinson, and J. C. Yu. Virtual craniofacial reconstruction using computer vision, graph theory and geometric constraints. *Pattern Recognition Letters*, 30(10):931–938, 2009.

[17] A. S. Chowdhury, S. M. Bhandarkar, R. W. Robinson, and J. C. Yu. Virtual multi-fracture craniofacial reconstruction using computer vision and graph matching. *Computerized Medical Imaging and Graphics : the Official Journal of the Computerized Medical Imaging Society*, 33(5):333–342, 2009.

[18] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computers & Graphics*, 22(1):37–54, 1998.

[19] M. Cimerman and A. Kristan. Preoperative planning in pelvic and acetabular surgery: the value of advanced computerised planning modules. *Injury*, 38(4):442–9, 2007.

[20] D. Doria and D. Chen. Interactive image graph cut segmentation. *The Insight Journal*, July-December, 2010.

[21] K. A. Egol, K. J. Koval, and J. D. Zuckerman. *Handbook of fractures*. Lippincott Williams & Wilkins, 2010.

[22] S. A. Ehmann and M. C. Lin. Accelerated proximity queries between convex polyhedra by multi-level Voronoi marching. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, volume 3, pages 2101–2106. IEEE, 2000.

[23] S. A. Ehmann and M. C. Lin. Accurate and fast proximity queries between polyhedra using convex surface decomposition. *Computer Graphics Forum*, 20(3):500–511, 2001.

[24] K. Engel, J. M. Kniss, C. Rezk-salama, D. Weiskopf, and M. Hadwiger. *Real-time volume graphics*. A. K. Peters, Ltd., Natick, MA, USA, 2006.

[25] J. Fan, G. Zeng, M. Body, and M.-S. Hacid. Seeded region growing: an extensive and comparative study. *Pattern Recognition Letters*, 26(8):1139–1156, 2005.

[26] F. R. Feito and J. C. Torres. Inclusion test for general polyhedra. *Computers & Graphics*, 21(1):23–30, 1997.

[27] J. Fornaro, M. Harders, M. Keel, B. Marincek, O. Trentz, G. Szekely, and T. Frauenfelder. Interactive visuo-haptic surgical planning tool for pelvic and acetabular fractures. *Studies in Health Technology and Informatics*, 132:123–5, 2008.

[28] J. Fornaro, M. Keel, M. Harders, B. Marincek, G. Székely, and T. Frauenfelder. An interactive surgical planning tool for acetabular fractures: initial results. *Journal of Orthopaedic Surgery and Research*, 5(50):1–8, 2010.

[29] J. Fornaro, G. Székely, and M. Harders. Semi-automatic segmentation of fractured pelvic bones for surgical planning. *Biomedical Simulation*, 5958:82–89, 2010.

[30] P. Fürnstahl, G. Székely, C. Gerber, J. Hodler, J. G. Snedeker, and M. Harders. Computer assisted reconstruction of complex proximal humerus fractures for preoperative planning. *Medical Image Analysis*, 16(3):704–20, 2012.

[31] B. Geiger. Real-time collision detection and response for complex environments. In *Proceedings Computer Graphics International 2000*, pages 105–113. IEEE Comput. Soc, 2000.

[32] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation*, 4(2):193–203, 1988.

[33] R. H. Gong, J. Stewart, and P. Abolmaesumi. Reduction of multi-fragment fractures of the distal radius using atlas-based 2D/3D registration. In *SPIE Medical Imaging*, pages 726137–726137–9, 2009.

[34] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: a hierarchical structure for rapid interference detection. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 171–180, 1996.

[35] G. Guennebaud, M. Germann, and M. Gross. Dynamic sampling and rendering of algebraic point set surfaces. *Computer Graphics Forum*, 27(2):653–662, 2008.

[36] G. Guennebaud and M. Gross. Algebraic point set surfaces. *ACM Transactions on Graphics*, 26(3):23, 2007.

[37] S. Hao, Y. Han, J. Zhang, and Z. Ji. Automatic isolation of carpal-bone in hand X-ray medical image. In *Informatics and Management Science I*, volume 204 of *Lecture Notes in Electrical Engineering*, pages 657–662. Springer London, London, 2013.

[38] M. Harders, A. Barlit, C. Gerber, J. Hodler, and G. Székely. An optimized surgical planning environment for complex proximal humerus fractures. In *MICCAI Workshop on Interaction in Medical Image Analysis and Visualization*, 2007.

[39] G. Hermosillo, V. C. Raykar, and X. Zhou. Learning to locate cortical bone in MRI. *Machine Learning in Medical Imaging*, 7588:168–175, 2012.

[40] C.-c. Ho, W. Fu-che, C. Bing-yu, and M. Ouhyoung. Cubical marching squares: adaptive feature preserving surface extraction from volume data. *Computer Graphics Forum*, 24:2005, 2005.

[41] H. Hoppe. Progressive meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '96*, pages 99–108, New York, USA, 1996. ACM Press.

[42] G.-F. Hu, Q.-S. Peng, and A. R. Forrest. Robust mesh smoothing. *Journal of Computer Science and Technology*, 19(4):521–528, 2004.

[43] Y. Hu, H. Li, G. Qiao, H. Liu, A. Ji, and F. Ye. Computer-assisted virtual surgical procedure for acetabular fractures based on real CT data. *Injury*, 42(10):1121–1124, 2011.

[44] K. Janc, J. Tarasiuk, A. S. Bonnet, and P. Lipinski. Semi-automated algorithm for cortical and trabecular bone separation from CT scans. *Computer Methods in Biomechanics and Biomedical Engineering*, 14(sup1):217–218, 2011.

[45] D. X. Ji, K. W. C. Foong, and S. H. Ong. A two-stage rule-constrained seedless region growing approach for mandibular body segmentation in MRI. *International Journal of Computer Assisted Radiology and Surgery*, 2013.

[46] J. J. Jiménez, F. R. Feito, and R. J. Segura. Tetra-trees properties in graphic interaction. *Graphical Models*, 73(5):182–201, 2011.

[47] J. J. Jiménez, F. R. Feito, R. J. Segura, and C. J. Ogáyar. Particle oriented collision detection using simplicial coverings and Tetra-Trees. *Computer Graphics Forum*, 25(1):53–68, 2006.

[48] J. J. Jiménez, F. Paulano, R. Pulido, and J. Jiménez. Computer assisted preoperative planning of bone fracture reduction: simulation techniques and new trends. *Medical Image Analysis*, 30:30–45, 2016.

[49] J. J. Jiménez and R. J. Segura. Collision detection between complex polyhedra. *Computers & Graphics*, 32(4):402–411, 2008.

[50] Y. Jun, L. Hao, L. Demin, D. Guohua, J. Hua, and S. Yi. Da Vinci robot-assisted system for thymectomy: experience of 55 patients in China. *The International Journal of Medical Robotics + Computer Assisted Surgery : MRCAS*, 2014.

[51] R. K. Justice, E. M. Stokely, J. S. Strobel, R. E. Ideker, and W. M. Smith. Medical image segmentation using 3D seeded region growing. In K. M. Hanson, editor, *Medical Imaging 1997*, pages 900–910. International Society for Optics and Photonics, 1997.

[52] Z. Kato. A unifying framework for correspondence-less shape alignment and its medical applications. *Communications in Computer and Information Science*, 276:40–52, 2013.

[53] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, pages 61–70, Cagliari, Sardinia, Italy, 2006. Eurographics Association.

[54] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36, 1998.

[55] A. Kronman and L. Joskowicz. Automatic bone fracture reduction by fracture contact surface identification and registration. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pages 246–249. IEEE, 2013.

[56] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha. Fast Proximity Queries with Swept Sphere Volumes. Technical report, Department of Computer Science, UNC Chapel Hill, 1999.

[57] P.-Y. Lee, J.-Y. Lai, Y.-S. Hu, C.-Y. Huang, Y.-C. Tsai, and W.-D. Ueng. Virtual 3D planning of pelvic fracture reduction and implant placement. *Biomedical Engineering: Applications, Basis and Communications*, 24(03):245–262, 2012.

[58] Leonar3Do. Leonar3do virtual reality hardware kit. http://www.leonar3do.com. Last accessed on June 25, 2016.

[59] C. Letta, A. Schweizer, and P. Fürnstahl. Quantification of contralateral differences of the scaphoid: a comparison of bone geometry in three dimensions. *Anatomy Research International*, 2014:5, 2014.

[60] P. H. Lim, U. Bagci, and L. Bai. Introducing Willmore flow into level set segmentation of spinal vertebrae. *IEEE Transactions on Bio-Medical Engineering*, 60(1):115–22, 2013.

[61] M. C. Lin and S. Gottschalk. Collision detection between geometric models: a survey. In *Proceedings of IMA Conference on Mathematics of Surfaces*, pages 1–20, 1998.

[62] B. Liu, X. Luo, R. Huang, C. Wan, B. Zhang, W. Hu, and Z. Yue. Virtual plate pre-bending for the long bone fracture based on axis pre-alignment. *Computerized Medical Imaging and Graphics : the Official Journal of the Computerized Medical Imaging Society*, 38(4):233–44, 2014.

[63] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169, 1987.

[64] D. F. Malan, C. P. Botha, and E. R. Valstar. Voxel classification and graph cuts for automated segmentation of pathological periprosthetic hip anatomy. *International Journal of Computer Assisted Radiology and Surgery*, 8(1):63–74, 2013.

[65] A. Mastmeyer, K. Engelke, C. Fuchs, and W. A. Kalender. A hierarchical 3D segmentation method and the definition of vertebral body coordinate systems for QCT of the lumbar spine. *Medical Image Analysis*, 10(4):560–77, 2006.

[66] S. Maubleu, C. Marecaux, and M. Chabanas. Computer-aided planning for zygomatic bone reconstruction in maxillofacial traumatology. In *Proceedings of the 2nd International Conference Surgetica*, pages 405–411, 2006.

[67] B. Mocanu, R. Tapu, T. Petrescu, and E. Tapu. An experimental evaluation of 3D mesh decimation techniques. In *ISSCS 2011 - International Symposium on Signals, Circuits and Systems*, pages 1–4. IEEE, 2011.

[68] M. H. Moghari and P. Abolmaesumi. Global registration of multiple bone fragments using statistical atlas models: feasibility experiments. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2008, pages 5374–7, 2008.

[69] C. Montani, R. Scateni, and R. Scopigno. A modified look-up table for implicit disambiguation of Marching Cubes. *The Visual Computer*, 10(6):353–355, 1994.

[70] S. Moreno, S. L. Caicedo, T. Strulovic, J. C. Briceño, F. Briceño, S. Gómez, and M. Hernández. Inferior maxillary bone tissue classification in 3D CT images. *Computer Vision and Graphics*, 6375:142–149, 2010.

[71] S. Musuvathy, S. Azernikov, and T. Fang. Semi-automatic customization of internal fracture fixation plates. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, volume 2011, pages 595–8, 2011.

[72] T. M. Nassef, N. H. Solouma, M. Alkhodary, M. K. Marei, and Y. M. Kadah. Extraction of human mandible bones from multi-slice computed tomographic data. In *2011 1st Middle East Conference on Biomedical Engineering*, pages 260–263. IEEE, 2011.

[73] A. Neubauer, K. Bühler, R. Wegenkittl, A. Rauchberger, and M. Rieger. Advanced virtual corrective osteotomy. *International Congress Series*, 1281:684–689, 2005.

[74] T. S. Newman and H. Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854–879, 2006.

[75] T. Okada, Y. Iwasaki, T. Koyama, N. Sugano, Y.-W. Chen, K. Yonenobu, and Y. Sato. Computer-assisted preoperative planning for reduction of proximal femoral fracture using 3-D-CT data. *IEEE Transactions on Bio-Medical Engineering*, 56(3):749–59, 2009.

[76] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization, 2002. VIS 2002.*, pages 163–170. IEEE, 2002.

[77] J. Peng, C.-S. Kim, and C.-C. Jay Kuo. Technologies for 3D mesh compression: A survey. *Journal of Visual Communication and Image Representation*, 16(6):688–733, dec 2005.

[78] J. Pettersson, H. Knutsson, and M. Borga. Non-rigid registration for automatic fracture segmentation. In *IEEE International Conference on Image Processing*, pages 1185–1188, 2006.

[79] J. Pile and N. Simaan. Modeling, design, and evaluation of a parallel robot for cochlear implant surgery. *IEEE/ASME Transactions on Mechatronics*, PP(99):1–10, 2014.

[80] R. Pulido, F. Paulano, and J. J. Jiménez. Reconstruction & interaction with 3D simplified bone models. In *WSCG 2014 Communication Papers Proceedings*, pages 321–327, 2014.

[81] A. J. Ramme, N. DeVries, N. A. Kallemyn, V. A. Magnotta, and N. M. Grosland. Semi-automated phalanx bone segmentation using the expectation maximization algorithm. *Journal of Digital Imaging*, 22(5):483–91, 2009.

[82] O. Ron, L. Joskowicz, C. Milgrom, and A. Simkin. Computer-based periaxial rotation measurement for aligning fractured femur fragments from CT: a feasibility study. *Computer Aided Surgery : Official Journal of the International Society for Computer Aided Surgery*, 7(6):332–341, 2002.

[83] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4. IEEE, 2011.

[84] H. Samet. Sorting in space: multidimensional, spatial, and metric data structures for computer graphics applications. In *SA '10 ACM SIGGRAPH ASIA 2010 Courses*, pages 1–52, Seoul, Republic of Korea, 2010. ACM.

[85] W. Schroeder, K. Martin, and B. Lorensen. *The visualization toolkit, third edition*. Kitware Inc., 2004.

[86] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *ACM SIGGRAPH Computer Graphics*, 26(2):65–70, 1992.

[87] T. B. Sebastian, H. Tek, J. J. Crisco, and B. B. Kimia. Segmentation of carpal bones from CT images using skeletally coupled deformable models. *Medical Image Analysis*, 7(1):21–45, 2003.

[88] R. C. Semelka, D. M. Armao, J. Elias, and W. Huda. Imaging strategies to reduce the risk of radiation in CT studies, including selective substitution with MRI. *Journal of Magnetic Resonance Imaging : JMRI*, 25(5):900–9, 2007.

[89] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146—-168, 2004.

[90] W. Shadid and A. Willis. Bone fragment segmentation from 3D CT imagery using the Probabilistic Watershed Transform. In *2013 Proceedings of IEEE Southeastcon*, pages 1–8. IEEE, 2013.

[91] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22(1-3):21–74, 2002.

[92] D. Shreiner. *OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1.* Addison-Wesley Professional, 7th edition, 2010.

[93] J. M. Sikorski and S. Chauhan. Computer-assisted orthopaedic surgery: do we need CAOS? *Journal of Bone & Joint Surgery, British Volume*, 85-B(3):319–323, 2003.

[94] E. M. Suero, T. Hüfner, T. Stübig, C. Krettek, and M. Citak. Use of a virtual 3D software for planning of tibial plateau fracture reconstruction. *Injury*, 41(6):589–91, 2010.

[95] N. Sugano. Computer-assisted orthopedic surgery. *Journal of Orthopaedic Science : Official Journal of the Japanese Orthopaedic Association*, 8(3):442–8, 2003.

[96] S. Tassani, G. K. Matsopoulos, and F. Baruffaldi. 3D identification of trabecular bone fracture zone using an automatic image registration scheme: a validation study. *Journal of Biomechanics*, 45(11):2035–40, 2012.

[97] T. P. Thomas, D. D. Anderson, A. R. Willis, P. Liu, M. C. Frank, J. L. Marsh, and T. D. Brown. A computational/experimental platform for investigating three-dimensional puzzle solving of comminuted articular fractures. *Computer Methods in Biomechanics and Biomedical Engineering*, 14(3):263–70, 2011.

[98] M. Tomazevic, D. Kreuh, A. Kristan, V. Puketa, and M. Cimerman. Preoperative planning program tool in treatment of articular fractures: process of segmentation procedure. In P. D. Bamidis and N. Pallikarakis, editors, *XII Mediterranean Conference on Medical and Biological Engineering and Computing 2010*, volume 29 of *IFMBE Proceedings*, pages 430–433, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[99] P. T. H. Truc, T.-S. Kim, S. Lee, and Y.-K. Lee. Homogeneity- and density distance-driven active contours for medical image segmentation. *Computers in Biology and Medicine*, 41(5):292–301, 2011.

[100] M.-D. Tsai, M.-S. Hsieh, and S.-B. Jou. Virtual reality orthopedic surgery simulator. *Computers in Biology and Medicine*, 31(5):333–351, 2001.

[101] G. Van der Bergen. *Collision detection in interactive 3D environments*. Elsevier, 2003.

[102] Visual Computing Lab - ISTI - CNR. Meshlab, 2014.

[103] R. Westphal, J. Faulstich, T. Gösling, S. Winkelbach, T. Hüfner, C. Krettek, and F. Wahl. Fracture reduction using a telemanipulator with haptical feedback. In *Computer Assisted Radiology and Surgery*, page 1369, 2003.

[104] T. Wiemann, H. Annuth, K. Lingemann, and J. Hertzberg. An extended evaluation of open source surface reconstruction software for robotic applications. *Journal of Intelligent & Robotic Systems*, 77(1):149–170, 2014.

[105] A. Willis, D. Anderson, T. Thomas, T. Brown, and J. L. Marsh. 3D reconstruction of highly fragmented bone fractures. In *Medical Imaging 2007: Image Processing. Proceedings of the SPIE*, page 65121, 2007.

[106] A. F. Wilson, P. B. Musgrove, K. A. Buckley, G. Pearce, J. Geoghegan, and D. R. Infirmary. Developing an application to provide interactive three-dimensional visualisation of bone fractures. In W. Tang and J. Collomosse, editors, *Theory and Practice of Computer Graphics*, pages 10–13. The Eurographics Association, 2009.

[107] S. Winkelbach, R. Westphal, and T. Goesling. Pose estimation of cylindrical fragments for semi-automatic bone fracture reduction. In B. Michaelis and G. Krell, editors, *Pattern Recognition, LNCS 2781*, volume 2781 of *Lecture Notes in Computer Science*, pages 566–573. Springer Berlin Heidelberg, 2003.

[108] S. Winkelbach, R. Westphala, and T. Gösling. Automatic computation of reposition parameters of fractured long bones based on CT-analysis. In *Computer Assisted Radiology and Surgery*, volume 1256, 2003.

[109] T. S. Yoo, M. J. Ackerman, W. E. Lorensen, W. Schroeder, V. Chalana, S. Aylward, D. Metaxas, and R. Whitaker. Engineering and algorithm design for an image processing Api: a technical report on ITK–the Insight Toolkit. *Studies in health technology and informatics*, 85:586–92, 2002.

[110] J. Zhang, C.-H. Yan, C.-K. Chui, and S.-H. Ong. Fast segmentation of bone in CT images using 3D adaptive thresholding. *Computers in Biology and Medicine*, 40(2):231–6, 2010.

[111] K. Zhao, B. Kang, Y. Kang, and H. Zhao. Auto-threshold bone segmentation based on CT image and its application on CTA bone-subtraction. In *2010 Symposium on Photonics and Optoelectronics*, pages 1–5. IEEE, 2010.

[112] G. Zheng, X. Zhang, D. Haschtmann, P. Gedet, X. Dong, and L.-P. Nolte. A robust and accurate two-stage approach for automatic recovery of distal locking holes in computer-assisted intramedullary nailing of femoral shaft fractures. *IEEE Transactions on Medical Imaging*, 27(2):171–87, 2008.

[113] B. Zhou, A. Willis, Y. Sui, D. Anderson, T. Brown, and T. Thomas. Virtual 3D bone fracture reconstruction via inter-fragmentary surface alignment. In *IEEE Workshop on 3D Digital Imaging and Modeling (3DIM)*, pages 1809–1816. Ieee, 2009.

[114] B. Zhou, A. Willis, Y. Sui, D. Anderson, T. P. Thomas, and T. Brown. Improving inter-fragmentary alignment for virtual 3D reconstruction of highly fragmented bone fractures. In *SPIE Medical Imaging*, pages 725934.1–725934.9, 2009.

[115] Y. Zhu, R. Phillips, J. G. Griffiths, W. Viant, A. Mohsen, and M. Bielby. Recovery of distal hole axis in intramedullary nail trajectory planning. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 216(5):323–332, 2002.

Alphabetical index