



UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO



UNIVERSIDADE DE LISBOA
FACULDADE DE MEDICINA



Final Course Project for the
Biomedical Engineering Degree

Geometric Modeling of Human Structures Based on CT Data – a Software Pipeline

Author

Daniel Simões Lopes

Supervising Professors

João Arménio Correia Martins (IST)

Jorge Guedes Campos (FMUL)

December, 2006

Lisbon, Portugal

Abstract. *One can describe geometric modeling as a large systematic process, whose input is an ordered stack of medical images that is submitted to a cascade of digital image operations and finite element algorithms. The outputs are accurate 3D surface and volume meshes, digital equivalents of the human tissue being modeled. Despite the usage of several computational methods, modeling human structures requires keen visual capabilities and anatomy knowledge. In this work a pipeline is proposed for modeling the bones of the upper and lower extremities (namely the humerus, radius, ulna, femur, patella, tibia and fibula) based on computed tomography (CT) images. Computed tomography is a non-invasive image modality that reveals the inner parts of the human body. The CT based pipeline relies on anisotropic diffusion, Gaussian and adaptive filters for image enhancement and restoration, semi-automatic segmentation using 3D active contour methods, algorithms for surface mesh generation and Delaunay tetrahedralization for volume mesh generation. The resulting surface meshes are anatomically compared with real (dead) bones. This qualitative comparison demonstrates the realism of the result of the modeling process. The quality of the volume mesh is also analyzed, and it is quantitatively confirmed that it has good quality for subsequent finite element computations. The list of software employed for geometric modeling of anatomical structures based on CT images is presented. The majority of the software is non-commercial.*

Key Words: Computed Tomography, Bone tissue, Image Enhancement, Image Restoration, Image Segmentation, Mesh Generation, Mesh Adjustment, Freeware & Open Source.

ACKNOWLEDGMENTS

It is well known that scientific research and development is no *one man's job*. Several individuals contributed to make this work possible and each one of them helped in their own and special way. Family members, professors, colleagues, friends, technical staff and others, all together are part of this project, or in other words, the convergence of their roles (combined with mine) created this project.

But firstly and above all, I personally express my gratitude to my supervisors, Professor João Arménio Correia Martins and Professor Jorge Guedes Campos, M.D., for giving me the opportunity to do this project. Not only the opportunity but mostly their efforts in making things happen as soon as possible and granting me all the resources available to do a proper work. Both never judged my trustworthiness and provided me a cheerful research environment.

This project was developed in a computer with adequate memory and graphic capabilities that was purchased specifically for this purpose by the Group 2 (Structural Mechanics, Mathematics and Computational Methods) of the research institute ICIST (Instituto de Engenharia de Estruturas, Território e Construção) of the Department of Civil Engineering and Architecture (DECivil) of the Instituto Superior Técnico (IST). I thank Professors Eduardo Borges Pires, Fernando Simões and other members of the Group for their involvement and support to this project.

Although I developed all my work within the Group 2 of ICIST (DECivil, IST) this project wouldn't be feasible if the Imagiology Service at the Hospital of Santa Maria (HSM) hadn't provided me the computer tomographic data that was the raw material for the imaging pipeline. The technical staff of the Imagiology Service and representatives of Philips® Portugal offered their good will and demonstrated great technical capabilities. Their main preoccupation was to ensure that I had a set of high quality CT images.

Processing computer tomographic images can sometimes be very harsh due to the amount of noise and artifacts that such data presents. So, whenever I had serious doubts or issues related to digital image processing, Professors João Raposo Sanches and João Salvador Marques, at the Institute of Systems and Robotics (ISR, IST), gently advised me to follow certain paths and approaches so that I could achieve the desired results. In particular, when I was working with morphological filters, Professor João Sanches addressed me to Professor Pedro Pina, at the Center of Geo-Systems of the Mining Department of IST. His experience in the field was of great importance as he suggested key processing principles and enumerated possible booby traps, typically encountered when dealing with morphological filters. Their know-how proved to be of great use during the step of digital processing and I thank them very much for their concerned attention.

After the segmentation stage, the next step consists on mesh generation. Here, Professor José Moitinho de Almeida from the ICIST (DECivil) demonstrated valuable expertise in the finite element field. Thanks to him I obtained volume meshes capable for further computational modeling.

Loading the volume mesh to a finite element solver is one of the last stages of the pipeline. Professor João Folgado, from the Division of Mechanical Project of the Mechanical Engineering Department of IST, was extremely valuable for helping me to perform a simple simulation in ABAQUS®.

To Professor Jacinto Monteiro, M.D., at the Orthopedic Service (HSM), I express gratitude for allowing me to photograph dead bones. This was indeed an exciting experience.

Last but not least, to my family that always supported me and gave me strength to continue. To all that I have mentioned I hope good luck and best wishes.

ABBREVIATIONS

1D – One Dimensional

2D – Two Dimensional

3D – Three Dimensional

ACM – Active Contour Model

AMG – Automatic Mesh Generator

CAD – Computer Aided Design

CT – Computer Tomography

DIP – Digital Image Processing

FEA – Finite Element Analysis

MRI – Magnetic Resonance Imaging

NURBS – Non-Uniform Rational B-Splines

PDE – Partial Differential Equation

PET – Positron Emission Tomography

ROI – Region of Interest

CONTENTS

ACKNOWLEDGMENTS.....	i
ABBREVIATIONS.....	ii
1. INTRODUCTION.....	5
Geometric Modeling Pipeline	5
Modeling Human Structures	7
Contribution of the Thesis.....	8
2. GEOMETRIC MODELING IN BIOMECHANICS OF TISSUES AND STRUCTURES.....	11
Biomechanics of Tissues and Structures	11
Geometric Modeling in Biomechanics and Clinical Visualization.....	11
Bone Modeling.....	12
3. OSTEOLOGY OF THE UPPER AND LOWER LIMBS.....	13
4. FUNDAMENTALS OF DIGITAL IMAGE PROCESSING.....	19
Point Operations and Neighboring Processing.....	20
Image Histogram.....	22
Image Processing Pipeline	22
5. IMAGE ENHANCEMENT AND RESTORATION.....	25
Contrast Enhancement.....	25
Gaussian Filter.....	26
Median and Wiener Filters.....	27
Anisotropic Diffusion.....	29
6. IMAGE SEGMENTATION.....	32
Global Thresholding.....	33
Active Contour Methods	35
Manual Segmentation.....	36
7. SURFACE MESH GENERATION AND ADJUSTMENTS.....	37
Iso-Surfacing	38
Smoothing	41
Decimation	41
Quadric Clustering	42

8. VOLUME MESH GENERATION.....	44
Delaunay Tetrahedralization	44
Mesh Quality and Volume Visualization	47
9. BONE MODELING PIPELINE FOR THE UPPER AND LOWER LIMBS.....	49
Image Acquisition	49
3D Image Parameters	50
Computer Requirements.....	50
Software Specifications.....	51
Software Pipeline	51
Image Analysis and Image Navigation	53
Image Enhancement and Restoration	55
Contrast Adjustment and Image Segmentation	56
Surface Generation and Adjustments	61
Volume Generation and Mesh Quality.....	64
Computational Simulation.....	65
Time and Modeling	65
10. 3D MODEL VS. REAL BONE PICTORIAL COMPARISON.....	66
11. DISCUSSION	68
Image Acquisition	68
Image Enhancement and Restoration	68
Image Segmentation	70
Surface Generation and Adjustments	71
Volume Generation and Mesh Quality.....	72
Software Considerations	73
12. CONCLUSIONS.....	74
REFERENCES.....	75

1. INTRODUCTION

Geometric Modeling Pipeline

In our days, modern technology is continuously providing more and more possibilities for medical science to comprehend biological systems in a tendentious holistic manner. Various equipments have been developed in order to extract anatomical and physiological data that reflect the body status. One of the greatest advances in medical systems has occurred with medical imaging. Modalities such as MRI, PET and CT allow an accurate visualization of the inner body in a non-invasive fashion, mapping the anatomy of a subject. For a great number of pathologies, medical images are the starting point for the clinical evaluation of the patients and the establishment of their diagnoses. Images are then a medium containing relevant structural or functional information of the human body, dictating in many cases future treatment, surgical planning or palliative care.

With several contiguous tomographic images, one can extract the geometrical information of the anatomic tissues, from which it is possible to reconstruct anatomically accurate 3D geometric models. This process that extends from medical image acquisition to 3D model visualization incorporates several algorithm blocks that are assembled to form a combined image-mesh pipeline.

The imaging modality for tomographic image acquisition and the anatomical structures to be modeled provide the major guidelines for building a modeling pipeline. Despite the methods and parameters that distinguish several pipelines they all share in common the structure presented in **Figure 1.1** [1-4].

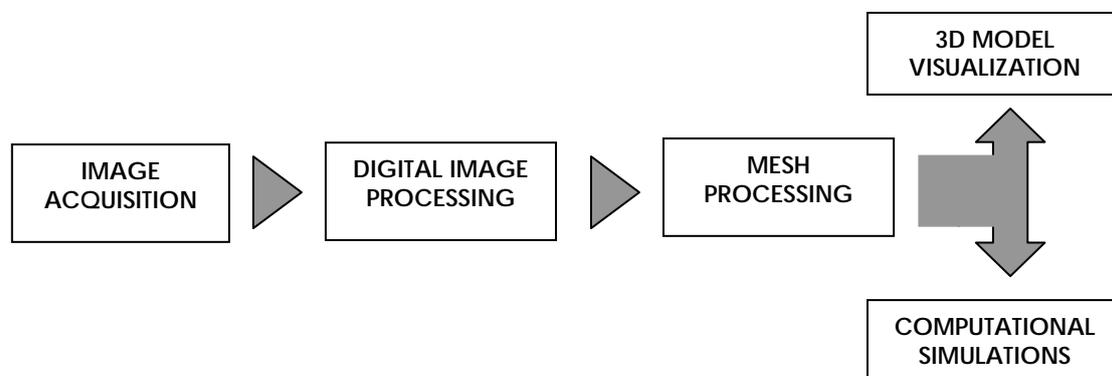


Figure 1.1. – Schema of a simplified pipeline for geometric modeling of human structures.

A typical geometric modeling pipeline based on medical images (**Figure 1.1**) is formed by three fundamental blocks connected in series: image acquisition, digital image processing and mesh processing. Each block of the proposed pipeline will be discussed in depth in forthcoming chapters, because they encompass several modules with specific functionalities. An overall introduction to the pipeline follows now.

Image Acquisition - obtaining images for modeling purposes can be quite challenging because several parameters have to be tuned, and sometimes the expected image quality is not satisfactory. Although it seems a simple task, acquiring data from medical imaging equipment

demands great acquaintance with the machine parameters; moreover, knowing how to place the patient relatively to the sensors saves time and grants a better “photography composition”.

Digital Image Processing - once acquired a stack of images, the next step to create a volume mesh of an anatomical structure is to attenuate signal noise and artifacts that may appear in the images. This operation is called filtering and it consists on altering the data so that certain features are improved while others are attenuated. After enhancing the desired image features with specific filters, the segmentation of anatomic tissues is performed. Segmentation is the most difficult stage of the entire pipeline. In short, the digital image processing block consists of filtering and segmentation procedures.

Mesh Processing - for each slice of the data volume the segmentation module outputs a closed surface. These segmented outlines are used to generate a surface boundary mesh of an anatomical structure. Due to the digital nature of the images, the surface mesh initially presents a ladder-like aspect. To eliminate this undesired effect, smoothing operations are applied. Reducing the number of nodes is also important, especially for numerical simulation purposes. With the surface mesh smoothed and simplified one can obtain the volume mesh. A volume mesh is nothing but a set of small finite volume elements, such as tetrahedral or hexahedral volumes, structurally organized to form a larger model. Volume mesh generation is the last of the pipeline tasks and involves time consuming computations.

The surface mesh produced by the pipeline has two major applications that depend on the mesh quality. To visualize a 3D model, a complex mesh with a large number of nodes is the best option to enhance highly detailed features. On the other hand, simplified meshes with much lesser nodes than those used for visualization, are more suitable for generating volume meshes used for subsequent computational modeling, so as to avoid excessively long computations.

Of all stages, the segmentation operation is the most time consuming as it implies user interaction to manually correct segmentation errors. Hence, segmentation is semi-automatic. Conversely, mesh processing and image filtering are practically automatic.

With this type of modeling system it is possible to create high-quality 3D meshes from a data volume of medical images, with large accuracy and, in some cases, remarkable mesh fidelity. Such systems provide the medical community important information to visualize and comprehend the patients anatomy and physiology through a virtual reality environment [2,3].

These meshes are necessary to solve bioengineering problems that involve the determination of stresses, strains, deformations, thermal or electromagnetic fields. The meshes must then be valid and numerically consistent models representing anatomical geometries suitable for finite element solvers and CAD environments. Quantitative data obtained from finite element analysis may be an important resource for increasing the probability of successful therapeutic treatments, and enable specific diagnosis, improving palliative care, surgical planning and injury prevention.

3D anatomical models are used in a wide spectrum of applications. But one of the most exciting domains of application is the biomechanical area. Simulation of the deformation of various soft tissues and muscles (like the muscles and tissues of the pelvic floor, the diaphragm, the intervertebral disks), simulation of the work of some organs or tissues under particularly stressing conditions (cough, vaginal delivery, car accidents) or surgical correction of some of their pathologies (stress urinary incontinence, idiopathic scoliosis) are just some of the applications of this vast biomedical field [5-8].

With this as motivation, the main goal of my project is to create a software modeling pipeline that generates subject-specific anatomical 3D models with high accuracy and suitable for

finite element solvers. There is one additional request: open source software or freeware should be preferably used. Commercial software also takes part of the software pipeline, such as MATLAB[®] and ABAQUS[®] that are common in most academic communities. An overall description of the utilized software and user-guide tips are provided in chapter 9.

But, as previously remarked, to design a modeling pipeline one must know which imaging modality is to be used and what anatomical structures are to be modeled. The input data of the present pipeline is a set of CT slices of the upper and lower limbs. The structures to be modeled are the long bones of these body segments and also the patella. Skin is another structure that can be extracted from CT data, even more easily than bone tissue. Soft tissue modeling, which includes muscle, fat, vasculature, fasciae, tendons and ligaments are more easily studied with MRI data.

The main reason for the correspondence between imaging modality and tissue type is due to image characteristics, namely signal contrast and noise structure. CT images are more suitable for bone modeling because hard tissue has a high contrast relatively to soft tissue. Thus, CT images are optimal hard tissue classifiers. The geometry of bones can also be reconstructed from MRI data, “but it is unclear what accuracy one can expect in this case” [9]. On the other hand, MRI images are appropriate for soft tissue modeling due to the greater tissue resolution in which it is possible to distinguish tendons from muscles and cartilage from bone. The small differences in signal intensity between these types of tissue do not allow a clear separation in a CT image, and therefore reasonable segmentation cannot be conducted. Even when vasculature is close to muscles, a CT image represents these different tissues as one only as they share the same signal intensity.

Modeling Human Structures

The huge amounts of data collected with various medical equipments are used to create, what many authors call [9], *in silico* human bodies. Such ‘humans’ are computational representations used to simulate, and partially recreate, human anatomy and physiology in a feasible way. Body, organ, tissue, cellular and molecular levels may have an artificial equivalent that is incorporated in a virtual environment. But as any other model, their purpose is to represent a certain reality. So these models are an approximation of an objective reality in this case a body, organ, tissue, cell or molecule of a human being.

The state of art of *in silico* humans is haunted by the current impossibility to create a whole-body model for an arbitrary human subject. If such subject-specific modeling were possible most of the practical biomedical questions would be answered because a simulation platform would be available for unlimited analysis and tests.

Due to the fact that human tissue is hierarchically organized, biophysical properties at one scale depend upon features that are present at other dimensional scales. Herein, a holistic understanding of a biological system demands multiple models that are coupled together sharing explicit relations and common parameters, in which each one of them models the phenomenon at a different dimensional scale.

However, two obvious computational problems arise: (i) in order to create a complete *in silico* human model, spanning multiple dimensional scales and that supports anatomy variability one must possess a humongous amount of information on the patient’s body; (ii) if all the models are coupled then they must be solved simultaneously, preferably in real-time. These two problems require tremendous computational capabilities and unlimited computer memory, which at the present time are virtually inaccessible.

Another major limitation for creating subject-specific *in silico* models, also related to data, is the impossibility to measure all of the parameters due to ethical restrictions or practical issues: some data can only be accessed invasively. Muscle-fiber and tendon length are two examples whose values are measured invasively. To surpass this limitation it is frequently assumed that the subject is not very different from the rest of the population; therefore invasive parameters have an average value based on anthropometric statistics.

The restrictive limitations for creating a whole-body *in silico* human are also shared with single scale modeling. For example, to model the femur a single computer with less than 1GB RAM and a 1.8 GHz CPU may crash down several times in response to huge amounts of data and over the limit processing requests. Many minutes of work are lost and consequently it is necessary to repeat the same action. All this is a dramatic waste of time and very annoying.

Geometrical modeling can be computationally expensive and time consuming, especially when the structures to be modeled are long bones. Nowadays, the minimum system requirements necessary for biomechanical geometric modeling are 1GB or more of RAM and 2GHz of CPU or more.

Contribution of the Thesis

The aim of this project is to reunite a set of software tools, that have digital image processing and computational geometry functionalities, establishing a modeling pipeline that transforms CT medical images of the body limbs into finite element meshes of bone structures in a timely fashion.

During the period of my project I studied and learned several digital image processing algorithms to enhance, restore and segment CT data [10,11]. The software used forced me to comprehend various methods so that I could create high quality meshes.

As I was building the pipeline, I became interested in certain image processing matters such as morphological filtering and edge detection with which I achieved satisfactory results:

- By applying a sequence of methods that use morphological operators, I was able to eliminate non relevant objects from the CT images such as the machine table and the all of the cloth present. This way it is possible to classify in the image which objects are human tissues and which are not (**Figure 1.2**).
- Based on the simple fact that an image is a 3D surface, detecting a contour in an image should be the same as finding regions of the surface whose unit normal vectors are almost parallel to the image coordinates, i.e., the N_z component must be close to zero (**Figure 1.3**).

So, besides the projects main objective, I tried to enhance this work by implementing these ideas that still need further development. Details and considerations behind these achievements will be presented in another document.

It is well known that edge detection procedures, such as the one I propose, are not entirely effective, especially for richly-detailed images. When a contour region does not have a high contrast the edges are not detected showing lacunae and boundary discontinuities. Thus, supposed closed contours are detected as a set of single opened contours, resembling a fragmented object.

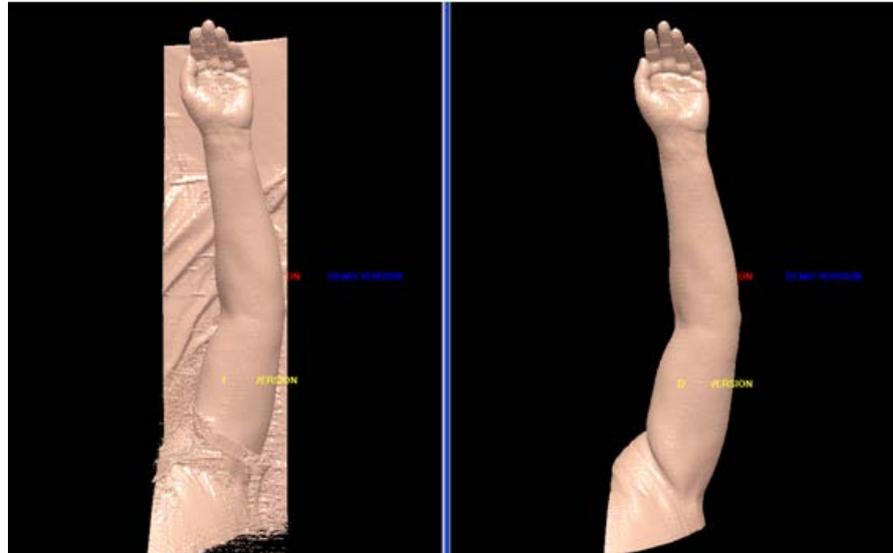


Figure 1.2. – Other project achievements besides volume mesh generation: non relevant object elimination. The CT table cloth was successfully removed from the images.

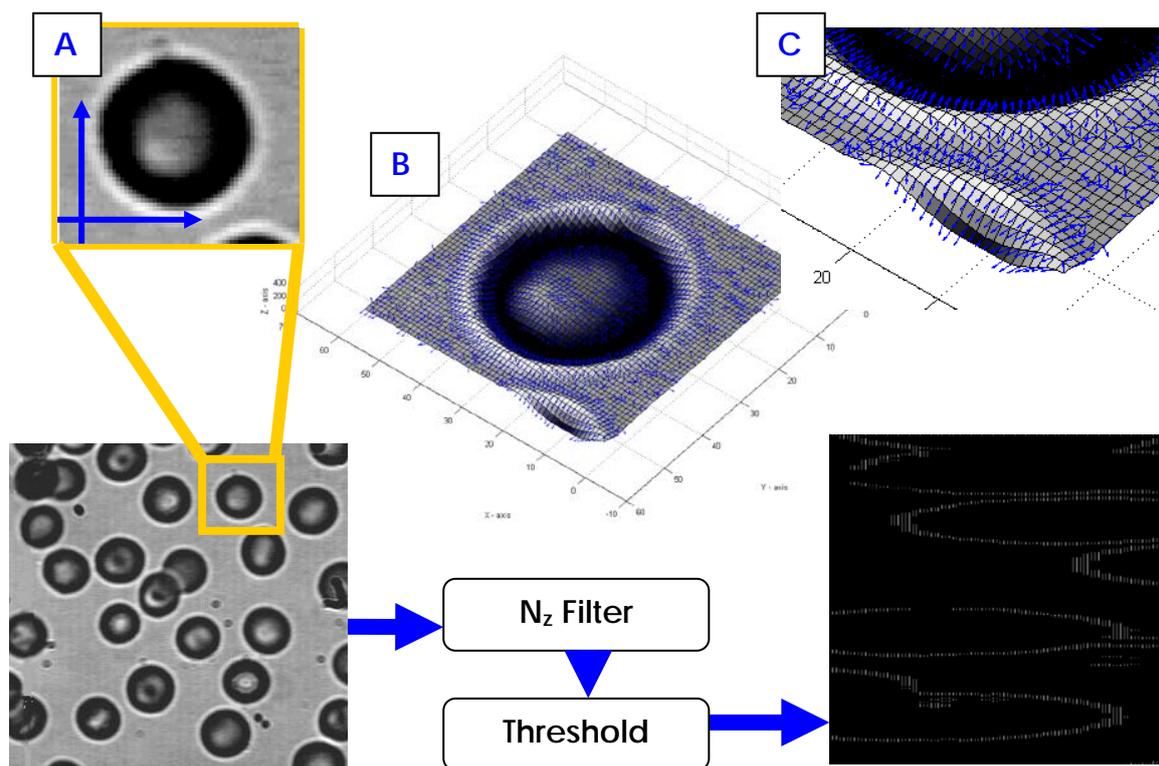


Figure 1.3. – Other project achievements besides volume mesh generation: edge detector based on the N_z component of the surface normal vectors. The N_z filter was applied to the blood cell image and then thresholded to show the pixels with a N_z value lesser or equal to 0.03. (A) – Image of a red blood cell with image coordinates. (B) – The image seen as a surface. (C) – A magnification of (B) shows the normal vectors.

Even the procedure for elimination of non relevant objects needs improvements. For instance, when noise or artifacts persist over the human/non-human tissue interface this method fails completely. In these cases manual segmentation is required.

At the end of the modeling pipeline I obtained volume meshes of the humerus, radius, ulna, femur, patella, tibia and fibula. These models can now be used for further computer simulations, especially in the field of biomechanics.

The whole-body model is a complex problem that can only be solved by a network of international research and investigation. Probably, it will take a significant amount of time and vast resources to achieve the ultimate *in silico* human: a complete virtual representation of the whole-body at every dimensional scale, or in other words, a virtual human clone [9].

It is my wish that the work I developed can help scientific community to get an infinitesimally bit closer to this idealistic scenario.

2. GEOMETRIC MODELING IN BIOMECHANICS OF TISSUES AND STRUCTURES

Biomechanics of Tissues and Structures

Biomechanics is the field of science that studies biological systems from the perspective of classical mechanics [1,2]. The investigation and research developed in this biomedical area provides an overwhelming understanding of how biophysical systems react and evolve when confronted with forces, or the dynamic response description of their movement. To do so a biomechanical analysis must incorporate realistic anatomical and physiological information in the models used in design of orthopedic implants and prosthesis, analysis of human movement, rehabilitation and ergonomic studies, geometric and computational modeling of tissues, among many other applications.

Nowadays, different but related areas, that are continuously expanding, can be distinguished in biomechanics. Biomechanics of tissues and structures is one of the disciplines that compose the overall biomechanics discipline mosaic. It regards the mechanical understanding of tissue, organ and structure behavior, both normal or pathological, aiming at realist model development and effective computational simulations.

As a biomedical engineering discipline, biomechanics of tissues and structures does not have a well established frontier as it frequently merges with other areas. Medical imaging, geometric modeling and finite element analysis are three engineering areas that form the groundwork for many biomechanical projects, including this one. Shall it be by sharing common issues or by complementing each other, this interrelation between (apparently) different scientific areas reinforces theoretical and experimental conclusions leading to accurate and precise models.

Geometric Modeling in Biomechanics and Clinical Visualization

Modeling anatomical structures consists of identifying a set of features including geometry, material properties, and boundary conditions, and establishing relations between each type of feature [1,2].

This work will focus on the creation of geometric models of human structures that are able to predict the biomechanics of the skeletal apparatus at the macroscopic scale (organ level).

Depending on the modality, certain anatomical structures can be accurately estimated from diagnostic imaging. For example, the skeletal anatomy can be accurately defined from computer tomography data, in which geometry accuracy is directly related to the radiation doses received and the spatial resolution required. So, the greater the doses and the larger the number of tomographic slices, the more accurate the model geometry becomes. But there is an antagonism between the amount of radiation and the required geometric accuracy: it is not ethically acceptable to deliver high doses of dangerous ionizing radiation to achieve better model accuracy. On the other hand, in clinical practice, X-ray radiation is administrated in much lower doses and with less tomographic images acquired, which is not quite acceptable for geometric modeling. Therefore a compromise between these two entities must be achieved so that one can obtain satisfactory results with the less damage possible to the subject.

Never let it be said that modeling the geometry of a human structure is a trivial task. Several challenges appear when attempting to accurately model such structures. The first barriers arise from anatomy itself. Organic structures can present very complicated curvatures and surfaces, which can be described by highly non-linear mathematical expressions (such as NURBS) [3]. Human anatomy is no exception. The anatomic variability is other difficulty which one must be aware of when modeling. Subject-specific models rely deeply on this feature. Medical imaging techniques also have limitations that bottleneck an entirely automatic extraction of information. Better resolutions and higher signal-to-noise ratios with less damage to the patient are expected in the near future.

To tackle all these limitations effectively one must compromise accuracy with a more coarse and practical solution.

When modeling, keeping an anatomy book near by, with fully illustrated or photographed images, is advisable as it helps the modeler to understand the results (s)he obtains.

Bone Modeling

Subject-specific bone geometry can be defined using CT imaging data. Currently the only information on the tissue that we can derive from CT images is the tissue density.

CT images present high contrast signal for bone tissue and skin, therefore CT data is suitable for modeling these structures. As bones are hard solids they greatly attenuate the high-energy radiation and therefore appear in the image with a high signal, while soft tissue, such as muscle and fat, are more transparent to X-rays. This way a high image contrast between soft and hard tissue can be visualized [4].

These systems are designed to produce image signals in which each picture element corresponds, univocally, to a certain type of tissue. This is of course an idealistic goal of medical image engineering, because ambiguous data is predominant on whatever system that has been or will ever be developed, and computer tomography is no exception to this rule.

3. OSTEOLOGY OF THE UPPER AND LOWER LIMBS

A priori knowledge concerning the anatomy of normal structures is of great importance for geometric modeling. If a modeler, strictly based on the available medical images, has few or no anatomic knowledge at all, then, the modeling task might be more prone to manual segmentation errors and non-optimal user-specified parameters. Thus, relying only on the image data less accurate anatomical models are created.

Studying the anatomy of the structures to be modeled [1-4] is then a primer to know how to perceive the acquired data. Depending on the image resolution, details can easily appear to the eyes of those who have prior knowledge of human anatomy.

The modeled bones are situated at the body extremities, also called limbs. From the upper limb, humerus, radius and ulna are considered; the femur, patella, tibia and fibula, belonging to the lower limb, are also described. The bone modeling will address the left arm and forearm and the right thigh and leg.

This chapter aims to be a concise “illustrated anatomy” focusing on the seven bony structures modeled, regarding essential osteologic concepts and descriptions.

A limb is an appendage of the body, connected to the trunk by one end, and free on the other end. The two types of extremities differ according to their functions: the upper extremities serve mainly for prehension while the lower extremities contribute to locomotion and weight support. Several bones assemble to compose the extremities, and each region of an extremity forms a body segment. The upper extremity consists of the shoulder girdle, arm, forearm and hand segments. The lower extremity comprises the pelvic girdle, thigh, leg and foot.

Bone tissue can be classified as dense (compact) or spongy (cancellous), according to how the cells and intercellular matrix materials are arranged. From a macroscopic point of view, the main difference between these two types of bone tissue resides on the density of the bone matrix. The bone matrix is the most X-ray attenuating substance within the human body. It contains abundant minerals composed by calcium carbonate and mostly of hydroxyapatite, a complex salt of calcium and phosphate. As the names suggest, compact bone tissue contains few spaces and more bone matrix than spongy bone tissue, which presents lots of gaps filled with bone marrow.

Bone surfaces are neither smooth nor flat. They are topologically intricate and texturally rich, populated with eminences and depressions which can distinguish a bone region. Eminences and depressions can be either articular or non-articular. The humerus and femur heads are evident examples of articular eminences. As is common in descriptive anatomy, the form of a structure is employed to designate non-articular eminences and depressions. The following table enumerates the different types and characteristics of non-articular eminences and depressions. The eminences and depressions listed in **Table 3-1** increase the surface area for ligament and muscle attachment. The enervation and irrigation paths that transmit neural signals and nourish the bone impress the bone surface with holes and canals conferring a characteristic texture.

Bones, themselves, can be classified according to their form. The modeled bones are of two types: long bones and flat bones.

In the human skeleton, the limbs are the segments that have the majority of the long bones. They assemble a system of levers used to support the weight of other tissues, such as muscles that confer the power of locomotion. Many produce blood at the medullary canal. Long bones

are characterized by their lengthy shaft and two articulated extremities, also known as diaphysis and epiphyses. Similar to a hollow cylinder, the shaft contains in its interior bone medulla. The narrowed form is suitable to allocate space for the muscles bellies. The extremities are generally large and expanded structures, which is convenient for bone connection and muscular attachment. Long bones are slightly curved structures. Histologically, the shaft is richer on compact tissue than the extremities which contain more cancellous bone tissue. The shaft walls present a density variation of cortical tissue throughout the bone main axis: greater thickness in the middle becoming thinner towards the extremities. Cancellous tissue is sparse but bone marrow fills the hollow interior. The extremities present a thin outside layer of compact bone that coats the cancellous tissue. Bone marrow fills all cancellous bone and the medullary canal. Most of the modeled bones are long bones, namely, humerus, radius, ulna, femur, tibia and fibula.

The only flat bone modeled was the patella. This unique bone acts as a shield towards the knee articulation. Protection and muscular attachment are the main functionalities of the broadly expanded flat plates of this type of bone. As in the extremities of long bones, a thin outside shell, formed by two layers, enfolds the prevailing cancellous tissue.

Eminences	Tuberosity / Protuberance / Process	Board, rough, uneven elevation.
	Tubercle	Small, rough prominence.
	Spine	Sharp, pointed eminence.
	Ridge / Crest / Line	Narrow, rough elevation, along the surface.
Depressions	Fossae	Anatomical pit.
	Groove	Long, narrow channel.
	Furrow	Deep, narrow depression.
	Fissure	Narrow opening or lengthy and deeply crack.
	Notch	V-shape indentation.

Table 3-1 – Non-articular eminences and depressions of bone surfaces.

The following figures illustrate the anatomy of the structures to be modeled. No deep anatomical description will be provided. Anatomy books are the indicated source for detailed information [1-3]. Furthermore in chapter 10, the 3D models obtained with the proposed pipeline will be compared with photos of authentic dead bones. The comparison will consist on visually evaluating the resemblance between these two entities.

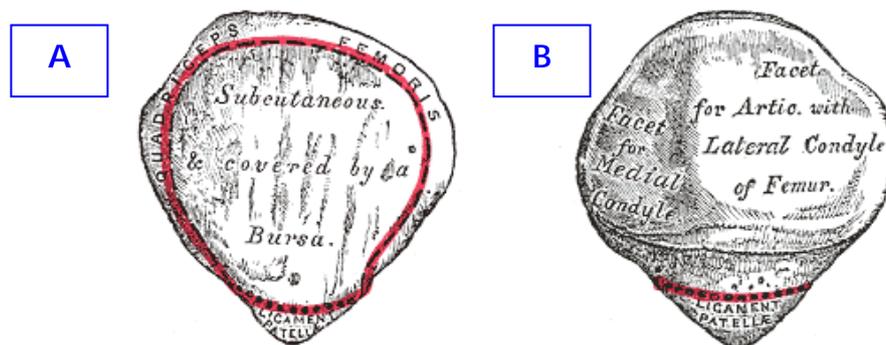


Figure 3.1 – (A) Anterior and (B) posterior view of the right patella [5].

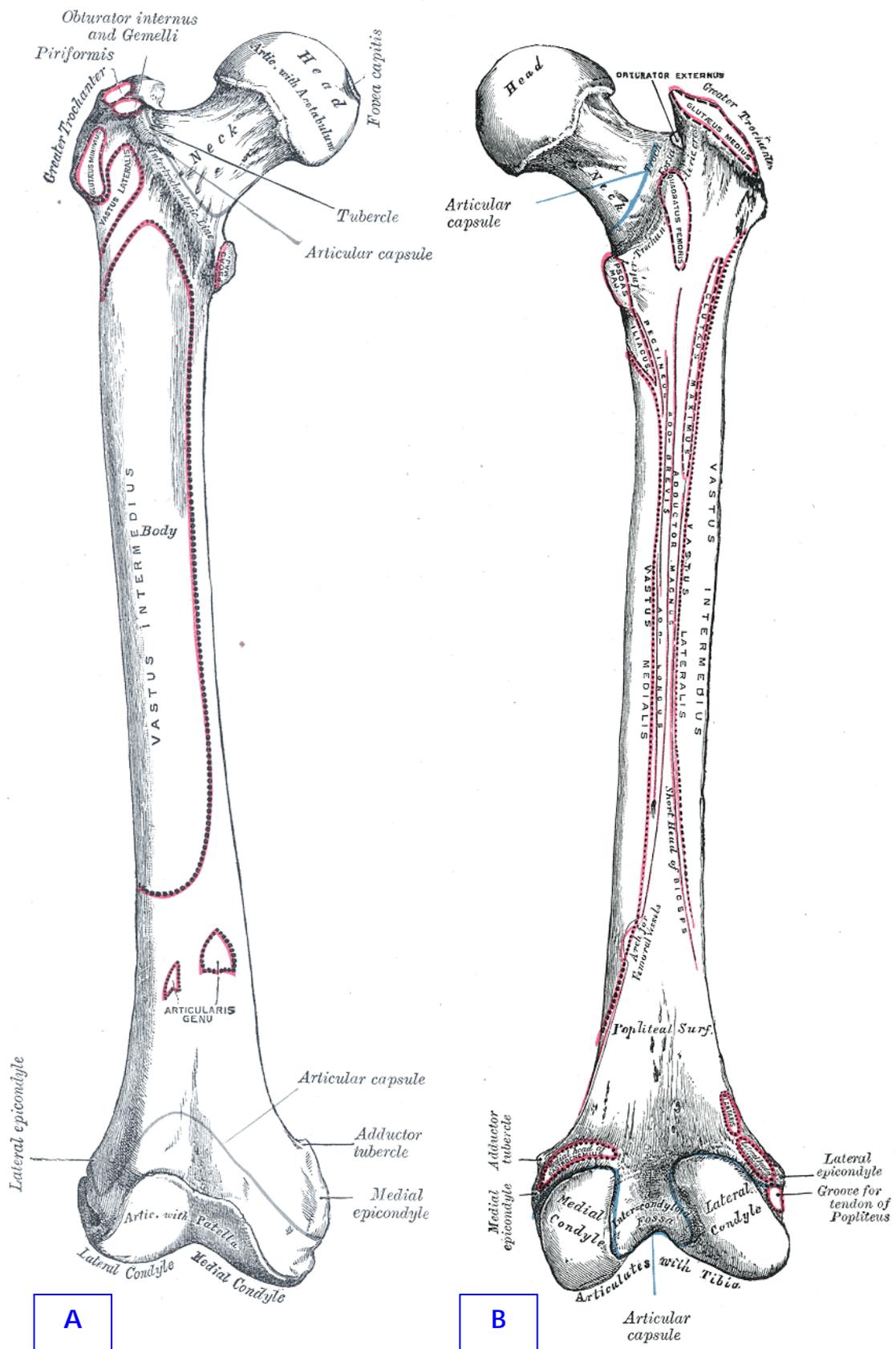


Figure 3.2 – (A) Anterior and (B) posterior view of the right femur [5].

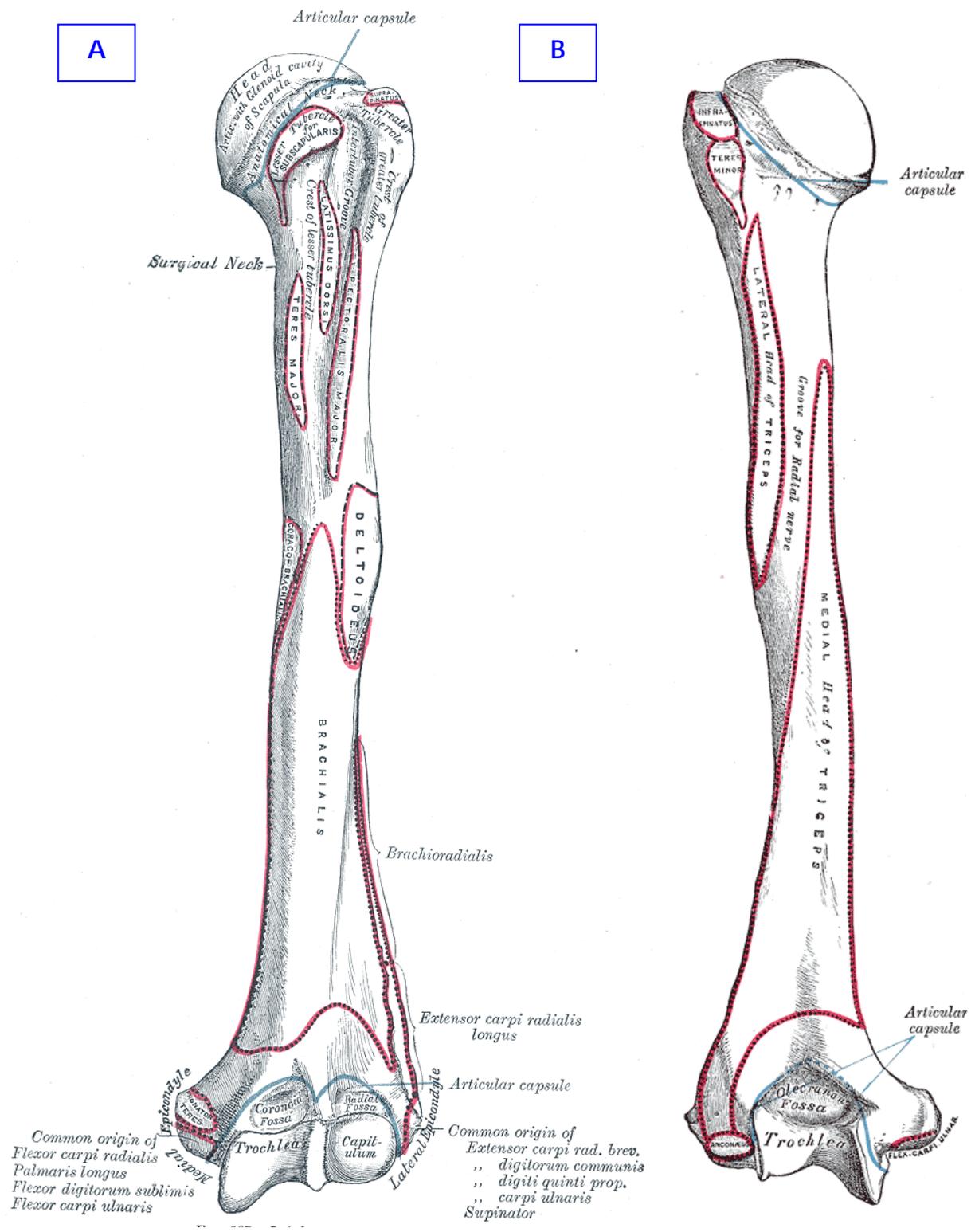


Figure 3.4 – (A) Anterior and (B) posterior view of the left humerus [5].

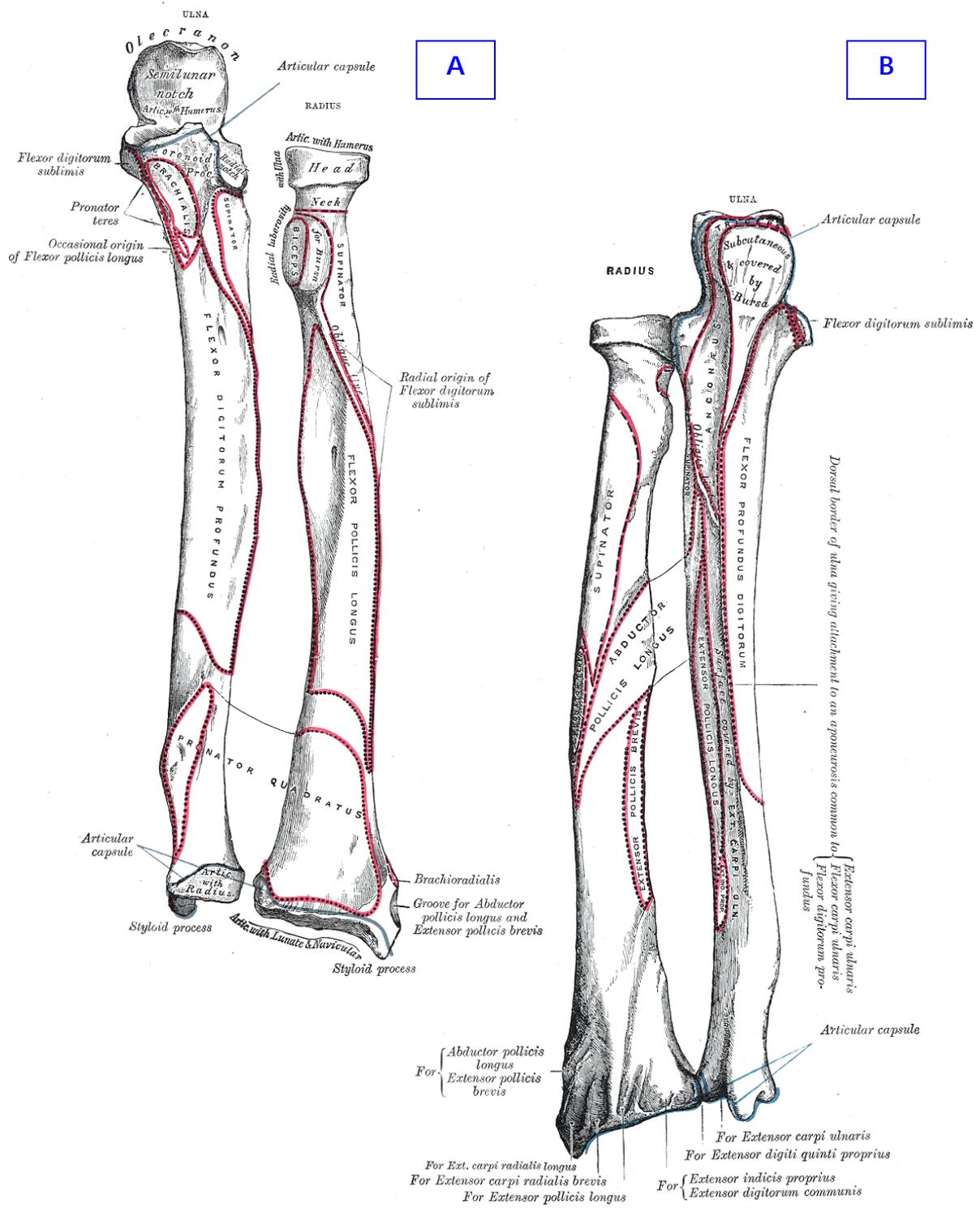


Figure 3.5 – (A) Anterior and (B) posterior view of the left ulna and radius [5].

4. FUNDAMENTALS OF DIGITAL IMAGE PROCESSING

There is an ancient Chinese proverb saying that an image is worth a thousand words. But, what is an image? Intuitively, an image is what we see. Mathematically, an image can be defined as a 2D function (or a set of 2D functions, in the case of color images) where at each point of the domain, the value of the function corresponds to the brightness or intensity of the image at that point [1,2]. An image defined as a 2D function, $z = f(x, y)$, can also be interpreted as a surface (**Figure 4.1**). Interpreting an image as a function is an important concept for the development and implementation of image processing algorithms.

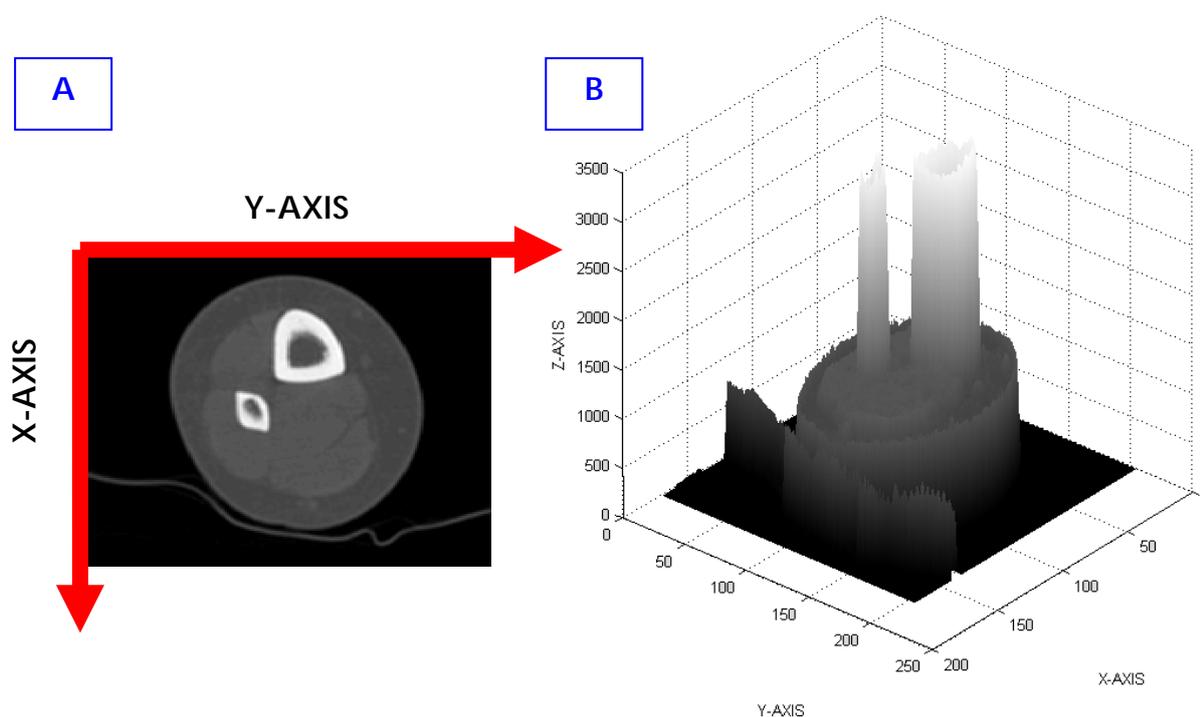


Figure 4.1 – A digital image (A) represented as a two-dimensional function (B).

Our visual system prefers sharp, clear and detailed images rather than blurred, noisy or vague images. With digital image processing (DIP) techniques it is possible to perform interesting transformations to enhance the visual information contained in an image, such as:

- edge enhancement, which produces a sharper image;
- attenuation or even elimination of several types of noise;
- improvement of image contrast;
- edge extraction, which delineates the objects.

The term digital in DIP refers to the discrete nature of the signal: the domain and the intensity values (or levels) of a digital image are finite and numerable. Depending on the type of numbers used to represent quantities (for example, integers, unsigned integers or double), an image can be binary, having only two intensity values, or a real color image, with over 16 million intensity values.

For most medical imaging applications, 256 (2^8) to 4096 (2^{12}) different intensities are used to represent anatomical or functional information. In particular, the acquired CT images are grayscale images with 4096 gray levels. Normally the value 0 is assigned to the black color and the highest value (ex.: 4096) corresponds to white. The values in between are different gray tones from dark to bright.

Due to its discrete nature, a digital image can be considered as a bi-dimensional matrix. Each element of the matrix is characterized by a pair of coordinates and a value equal to the image intensity at that point. This matrix element is called a *pixel*, the short designation for *picture element*.

The neighborhood of a given pixel is composed by the adjacent pixels that surround it. A neighborhood can have different shapes and sizes but usually it has a rectangular form, such as 5x5 or 7x9. A pixel's neighborhood can also be called a spatial mask.

The concept of an image as a matrix is more evident if a portion of an image is amplified, as seen in **Figure 4.2**.

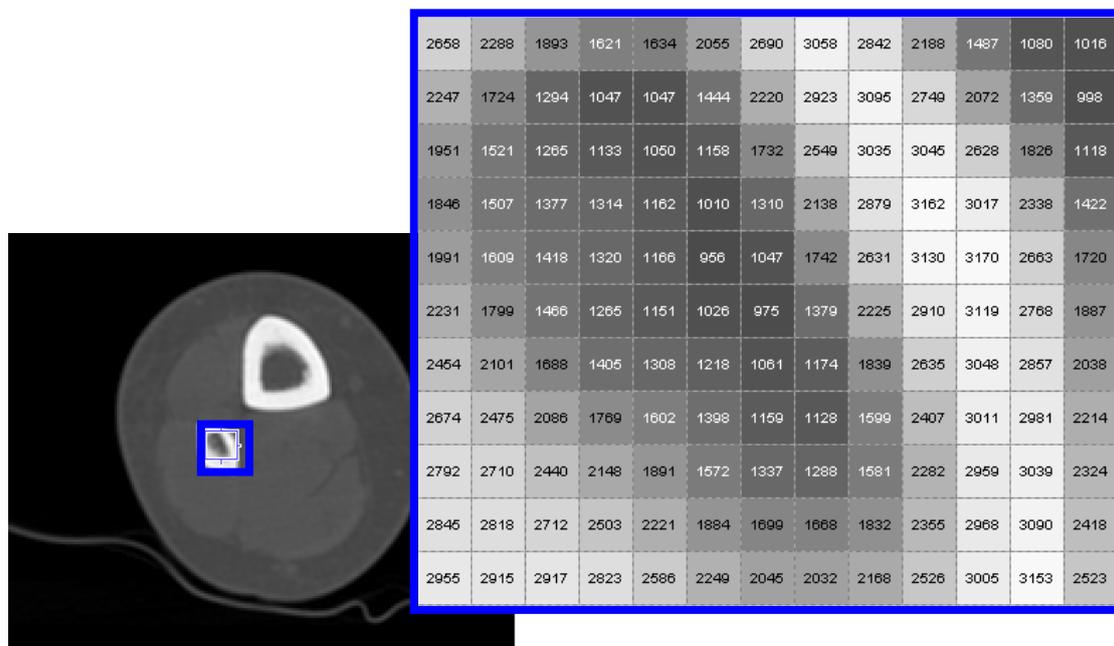


Figure 4.2 – Amplified portion of a digital image. The numbers represent the pixel's gray level.

Point Operations and Neighboring Processing

Based on the matrix concept, spatial image processing operations can alter the value of a certain pixel, individually or based on neighboring pixels. The former type of operations is called point operations; the latter is known as neighborhood processing.

Point operations are the simplest type of operations that can act upon an image's pixel. A certain pixel at a given location can be added, multiplied, divided, powered, or compared

quantitatively to another pixel of a different image at the same location. Besides arithmetic and relational operators, logical operators such as AND, OR and NOT can also be applied to images.

In neighborhood processing, the new value of a pixel $p(x, y)$ is calculated by the function $f(p, m)$ whose arguments are the neighboring pixels n and the mask coefficients m . The function output will be the new pixel value at that point $p'(x, y) = f(p, m)$. The combination of the mask and the function is called a filter.

The following figure illustrates how a pixel value can be computed by neighborhood processing. The example presented consists of filtering the image with a spatial Gaussian filter (standard deviation $\sigma = 3$, 5×5 mask).

Spatial filtering is done in three steps: (i) the mask is positioned over the pixel at coordinates (x, y) (the center of the mask coincides with the current pixel); (ii) each pixel within the neighborhood is multiplied by the corresponding mask coefficients; (iii) all the products are summed giving the new pixel value $p'(x, y)$.

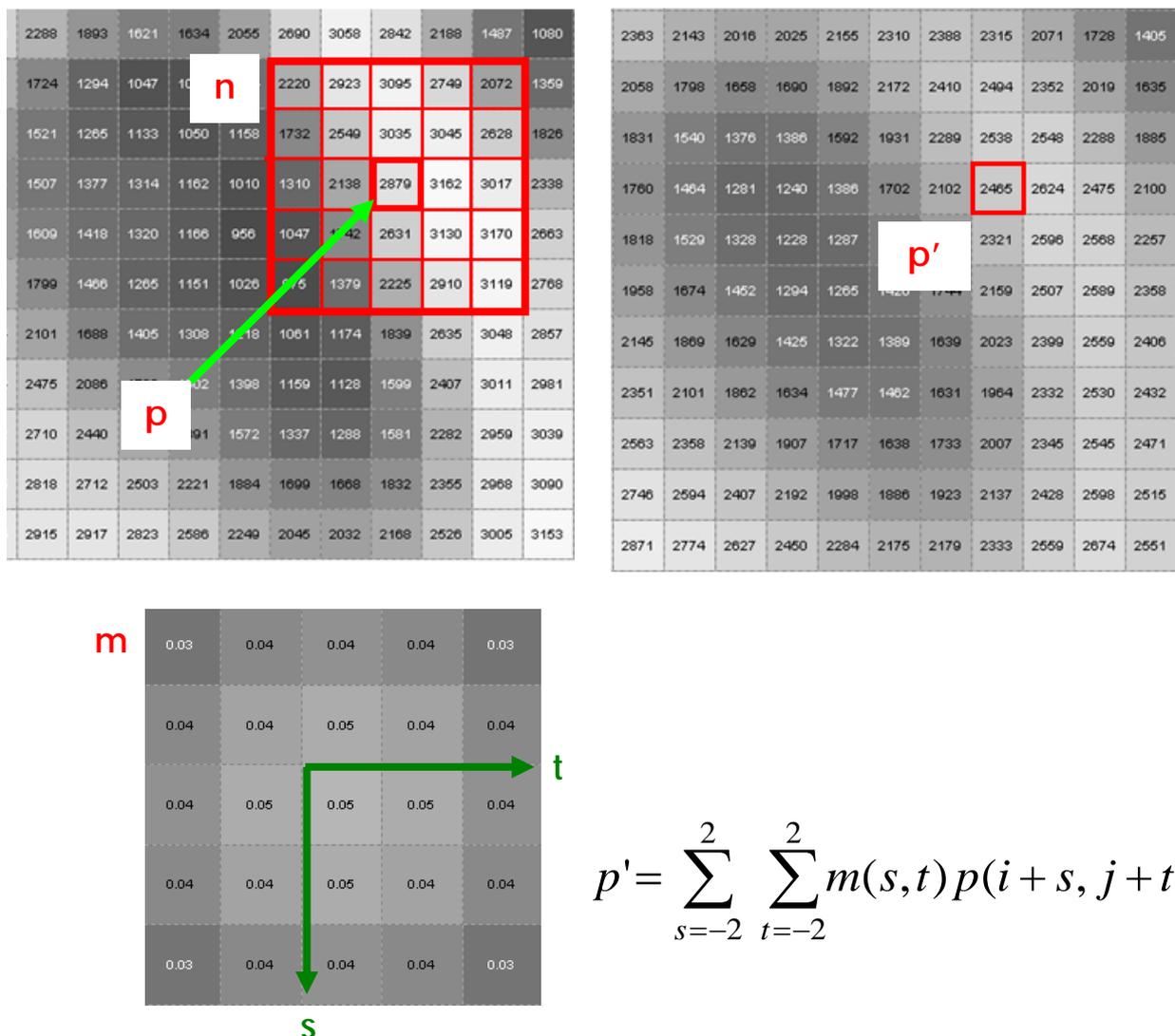


Figure 4.3 – Example of how neighborhood processing is performed upon a pixel.

n – pixel neighborhood; p – current pixel; m – mask; p' – new pixel value

Image Histogram

Several statistical parameters can be extracted from a grayscale image. One of the most important statistical functions of an image is the distribution of the number of occurrences of a specific gray level, i.e., the image histogram. The appearance of an image can be deduced just by analyzing its histogram. For example, a dark image would present higher density at the low levels of the histogram. On the other hand, the inverse would happen to a very bright image. A well contrasted image would present a uniformed histogram with no predominance of a specific gray level. By manipulating the histogram the image contrast can be enhanced.

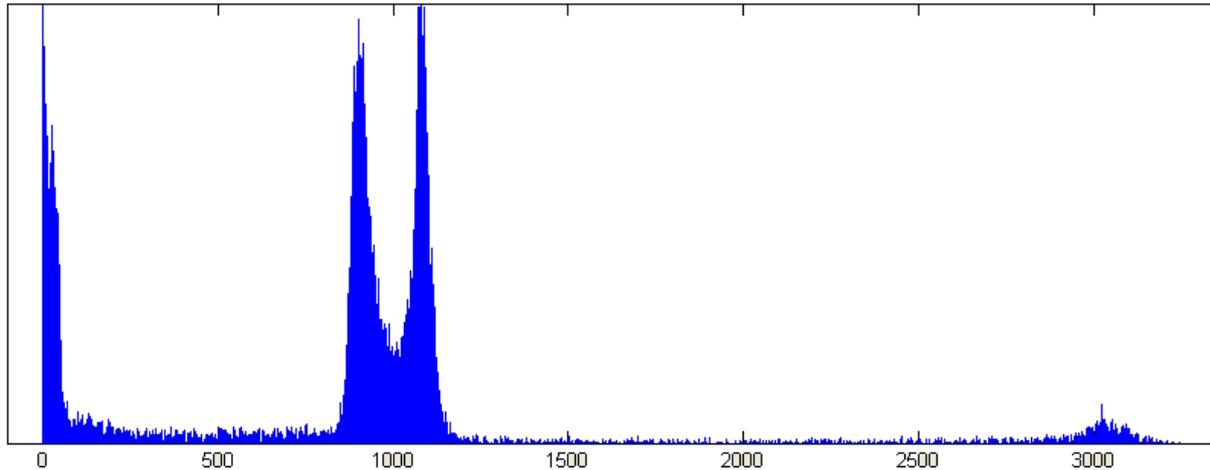


Figure 4.4 – The image histogram of an axial cross section of the forearm. The histogram is multi-modal.

Image Processing Pipeline

Depending of the problem under consideration, different image processing algorithms are applied to perform different tasks. These tasks are arranged in a systematic manner forming an image pipeline. Each stage executes a specific procedure or function according to the chosen parameters and the received inputs.

For instance, the proposed image pipeline for bone modeling is basically composed by three major stages: image enhancement, image restoration and image segmentation (**Figure 4.5**).

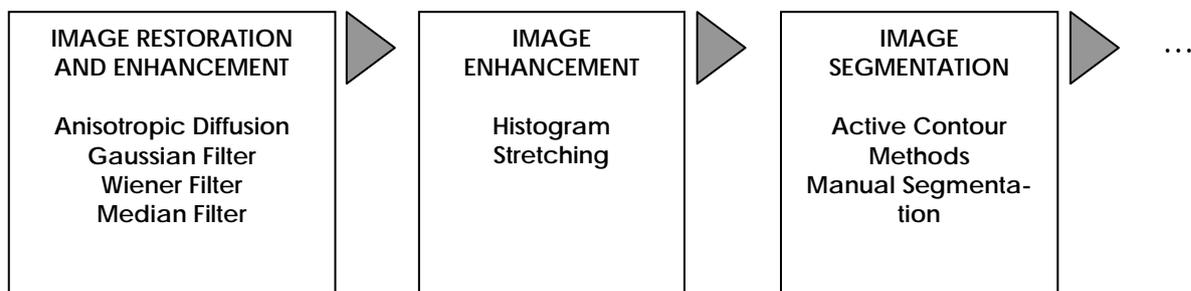


Figure 4.5 – Schematic of the image pipeline used for CT images of the upper and lower limbs. The operations of each stage are listed.

Image enhancement and restoration are tightly related because such methods are intended to increase image quality by removing noise, improving image contrast or highlighting edges. Four different filters were applied to the sagittal slices of the data volume. Gaussian, Wiener, median and anisotropic diffusion filters proved to be suitable as noise was attenuated and some image defects were repaired.

Modifying the histogram distribution directly influences the image contrast. By reallocating the number of observations of a gray level to another gray level, the contrast between soft tissues and hard tissues can be deeply accentuated.

After enhancing the contrast of the bone tissues a more suitable input for image segmentation was obtained. Any segmentation method has the objective of isolating certain aspects of an image, classifying each pixel as foreground or background, i.e., a pixel of interest or a non-interesting pixel. In this work thresholding combined with active contour methods are used to extract bone tissue, both cortical and cancellous tissue.

In general, bone tissue has a high contrast relatively to soft tissue due to the existence of a compact and dense cortical bone. But, wherever cancellous bone tissue is present and cortical bone is thin, namely at the epiphyses (**Figure 4.6**), the intensity signal is poorer and closer to soft tissue values. This feature is problematic for most segmentation methods, including the one applied here, because it consists in a boundary discontinuity. At these locations manual segmentation is frequently necessary.

Image processing is a multi parameter procedure in which each stage of the pipeline needs to be tuned. Adjusting the parameters of the pipeline is indeed the experimental side of any imaging process, demanding several iterations and evaluations before reaching the desired results. Many combinations of operations have to be developed, tested, tuned and altered before defining the final pipeline that gives the best results.

Even if the proposed image pipeline has some limitations on automation, it still provides accurate results (see chapter 10).

With the digital image fundamentals presented here and a brief introduction to each stage of the image pipeline, advanced explanations on image enhancement, image restoration and image segmentation will be more comprehensible.

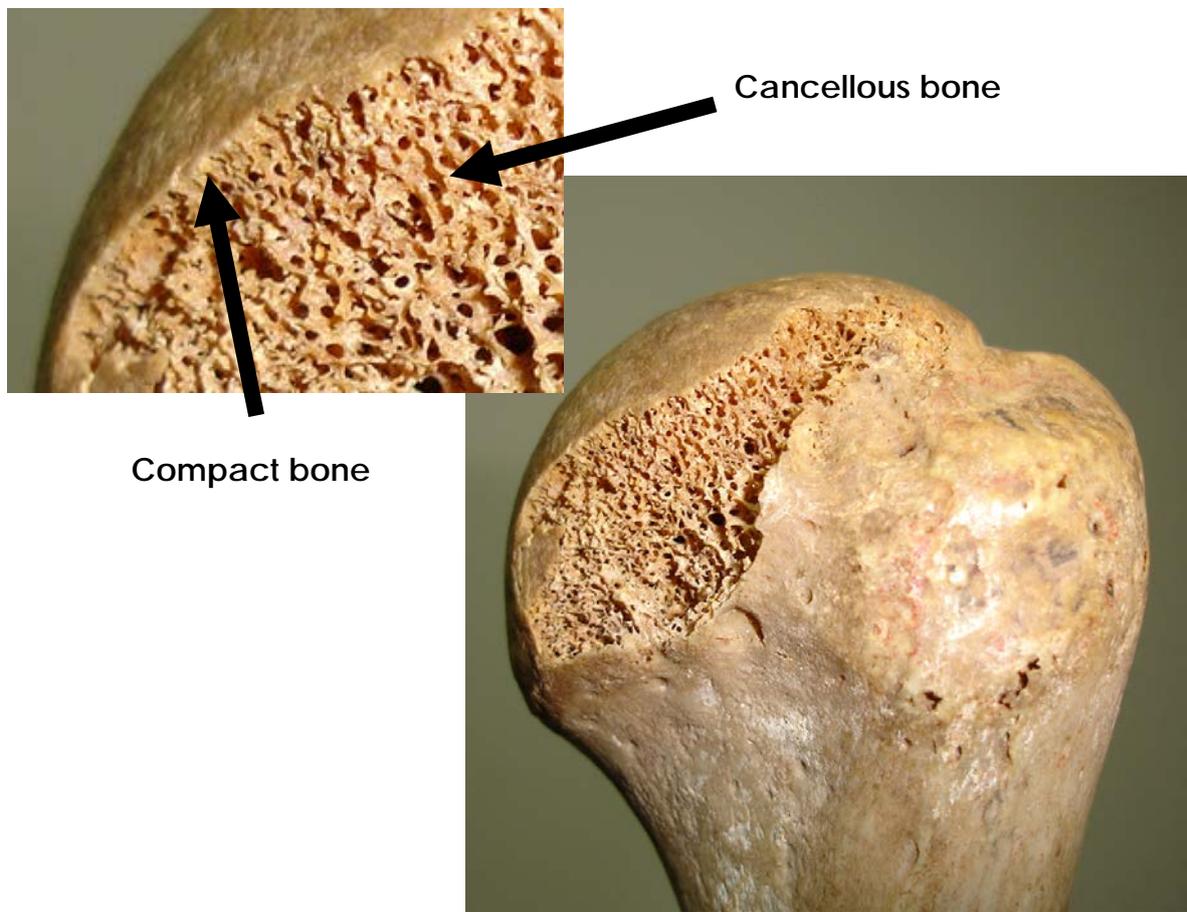


Figure 4.6 – Detail of the humerus head. At the epiphysis the cortical bone is very thin and the cancellous bone is abundant.

5. IMAGE ENHANCEMENT AND RESTORATION

The design of modern CT systems endeavors an achievement that involves opposing demands: improving image quality and increasing spatial resolution, with a reduction on patient radiation dose. The radiation dose/image quality relation is apparently inflexible, since only by increasing the radiation dose and X-ray power the image resolution will increase. But there is one way to partially solve this dilemma: filtering projection or image data. CT scanners use low-pass filters applied to the projection data reducing noise but also deteriorating the resolution of the image [1-5]. Advanced software platforms of CT scanners [6] incorporate multi-dimensional adaptive filters that prove to be very effective in increasing image quality and decreasing radiation dose. Filtering is therefore important for CT imaging.

Improving image quality is to repair the damage provoked by artifacts, and to eliminate noise without deteriorating the image itself. Image restoration and image enhancement procedures aim at producing images with optimal quality, thus providing improved diagnostic images to the physician.

A filtering algorithm can be classified as belonging to the enhancement or the restoration categories or even to both, depending on the application. In general, such classification is easily established since it relies on the task to be performed: image improvement corresponds to enhancement, while damage removal matches restoration.

As pointed earlier, the developed imaging pipeline (**Figure 4.5**) uses five different techniques for image enhancement and restoration. For image enhancement, contrast stretching was used. Gaussian, median and Wiener spatial filters were applied for image restoration. Now, anisotropic diffusion can be classified simultaneously as an image enhancement and a restoration procedure, given that the image quality is improved and artifacts are simultaneously removed.

The application of any restoration technique is based on a model of an image degradation process [7]. Such model stipulates that an image results from a degradation processes and noise addition. Here we will assume that there is no degradation function, and only noise is attenuated or eliminated in the CT images.

Filtering techniques may create high quality images from very noisy CT scans. Therefore, filters such as those applied may contribute for using less radiation upon a patient.

Contrast Enhancement

Manipulating the image or volume histogram is one of the first steps to be carried out in image analysis. The histogram summarizes the intensity information in a one-dimensional function. Contrast adjustment is a histogram processing procedure that reallocates the original gray levels to new gray intervals, in order to obtain the desired contrast enhancement. In other words, by “spreading” and dislocating the histogram the contrast will improve. This technique is known histogram stretching.

The histogram of the lower body extremity is presented in **Figure 5.1** (the histogram of the upper extremity is similar, **Figure 4.4**). At first glance, this graph indicates that the histogram is multi-modal. Each peak corresponds to a different type of tissue, from lowest to highest level: air, fat, muscle and bone (**Figure 5.1**).

Therefore, to achieve a good contrast between soft and hard tissue, one possible contrast stretching function can be a piecewise linear function [7] chosen with the purposes of (**Figure 5.1**):

- (i) broadening the intensities of the muscle interval (region B), because cancellous bone and muscle share the same intensity values;
- (ii) saturating the bone values (region C), since it is the tissue to be modeled and that is convenient for a better visualization;
- (iii) canceling the lower intensities to zero (region A) given that they are not of any interest.

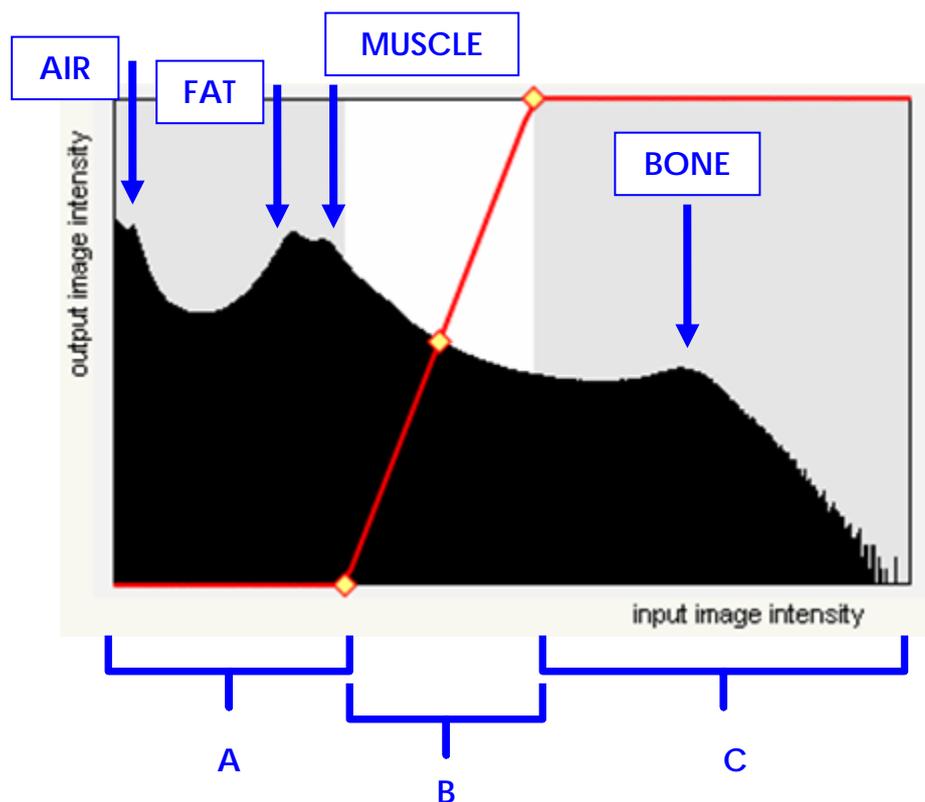


Figure 5.1 – Stretching function juxtaposed with the histogram of the lower extremity volume. The histogram observations are traced on a logarithm axis.

Gaussian Filter

Low-pass filtering is an essential step to attenuate high frequency noise that is present in most medical images.

The Gaussian spatial filter [8] can be seen as a weighted mean over the pixel's neighborhood, and the new pixel value will contain information of all the neighboring pixels. This operation smears the intensity values and the overall image appears blurrier. The image regions become more homogeneous.

This filter is based on the bi-dimensional Gaussian probability distribution function:

$$f(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (5.1)$$

where x and y are the pixel coordinates and σ the standard deviation. The greater the σ or the mask size the blurrier the image. **Figure 4.3** illustrates the application of the Gaussian filter to a portion of a CT image.

Median and Wiener Filters

The spatial filtering mechanism is a straightforward technique, as described in chapter 4 and illustrated in **Figure 4.3**. There are various types of filters designed for different purposes. For the CT images, nonlinear filters have been applied to restore image features by reducing certain degradations occurred during the data acquisition and transmission stages, namely streaks and noise mottle. Studies reveal that nonlinear filters might lead to better image quality than current linear filters, such as the Gaussian filter [3].

A nonlinear filter is a nonlinear function of the grayscale values within a mask. For image restoration, a nonlinear filter is aimed at reducing the artifact presence in an image, without reducing its resolution. The median and Wiener filters [8] are two examples of how a restoration technique can be successfully performed using neighborhood operations. Each of these filters is suitable for a specific type of noise or artifact.

When the image appears to have sharp and sudden disturbances, randomly scattered over the image as points, the median filter comes in handy since it eliminates outlier pixels very easily (**Figure 5.2**): the median filter takes the central value of an ordered list composed by the mask pixels. Therefore, a median filter is classified as a rank-order filter, meaning that the elements under the mask are sorted and a particular value is returned.

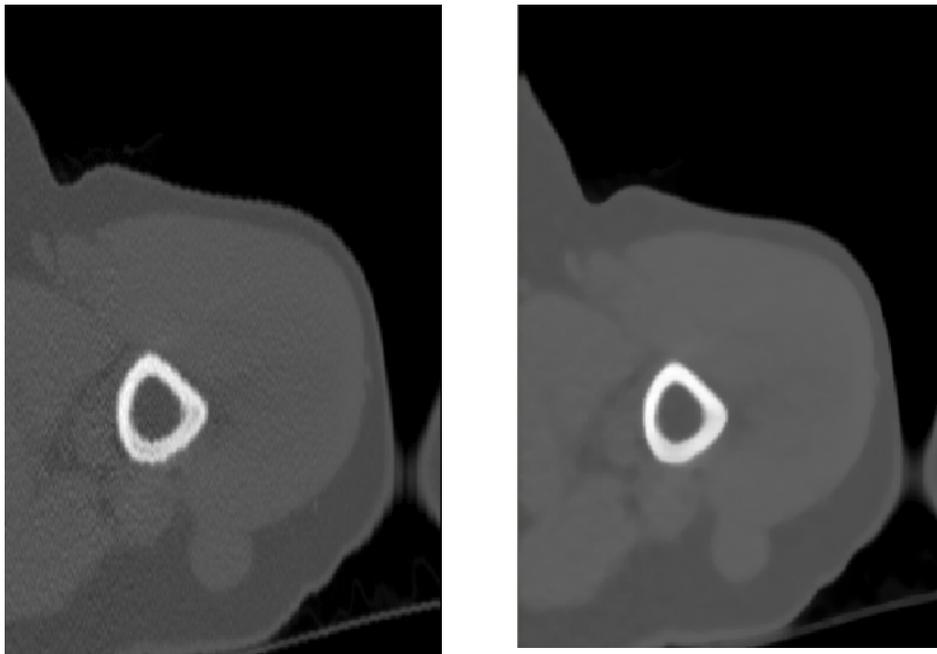


Figure 5.2 – Median filter as a spatial filtering technique applied to a sagittal cross section of the upper limb. The image at the left is the original image and at the right the filtered image.

Generally, spatial noise is a random variable whose statistical behavior can be modeled by a probability density function. In practice, it is frequently considered that random fluctuations that appear to be normally distributed over the image follow a Gaussian noise model.

The Wiener filter is a nonlinear adaptive filter used to clean Gaussian noise. This filter changes its characteristics, i.e., its parameters, according to local statistics of the grayscales under the mask. In the regions with outliers the Wiener filter acts as a median filter.

The nonlinear expression that defines the filter is given by

$$\hat{I}(x, y) = m_m(x, y) + \frac{\sigma_m^2(x, y)}{\sigma_m^2(x, y) + \sigma_I^2} (I(x, y) - m_m(x, y)), \quad (5.2)$$

where m_m and σ_m^2 are the mean and variance of the mask, respectively, and σ_I^2 is the variance of the overall noise of the image. The filter returns either $I(x, y)$ or $m_m(x, y)$ according to the local variance. By expression (5.2), if the local variance is high, then the fraction will be close to one and the output will be approximately $I(x, y)$. Note that high variance corresponds to details such as edges. On the contrary, low local variance will give a fraction close to zero and the returned value will be near $m_m(x, y)$. A region with a low variance matches constant regions.

Usually, the value of σ_I^2 needs to be estimated since the image noise model is only an assumption. An alternative expression to (5.2) that considers an estimate of σ_I^2 is as follows

$$\hat{I}(x, y) = m_m(x, y) + \frac{\max\{0, \sigma_m^2(x, y) - n\}}{\max\{\sigma_m^2(x, y), n\}} (I(x, y) - m_m(x, y)), \quad (5.3)$$

where n is the estimated noise variance calculated as the mean of all $\sigma_m^2(x, y)$.

Therefore, adaptive filters tend to smooth uniform regions and to preserve edges. The mechanism is straightforward: first, the region around a given pixel is evaluated to see if it is a uniform region. If that is the case smoothing is performed using only the pixels that belong to the same region.

By comparing **Figure 5.3** with **Figure 5.2** the differences between the two adaptive filters is evident: Wiener preserves edges better than the median filter and both are generally low-pass filters.

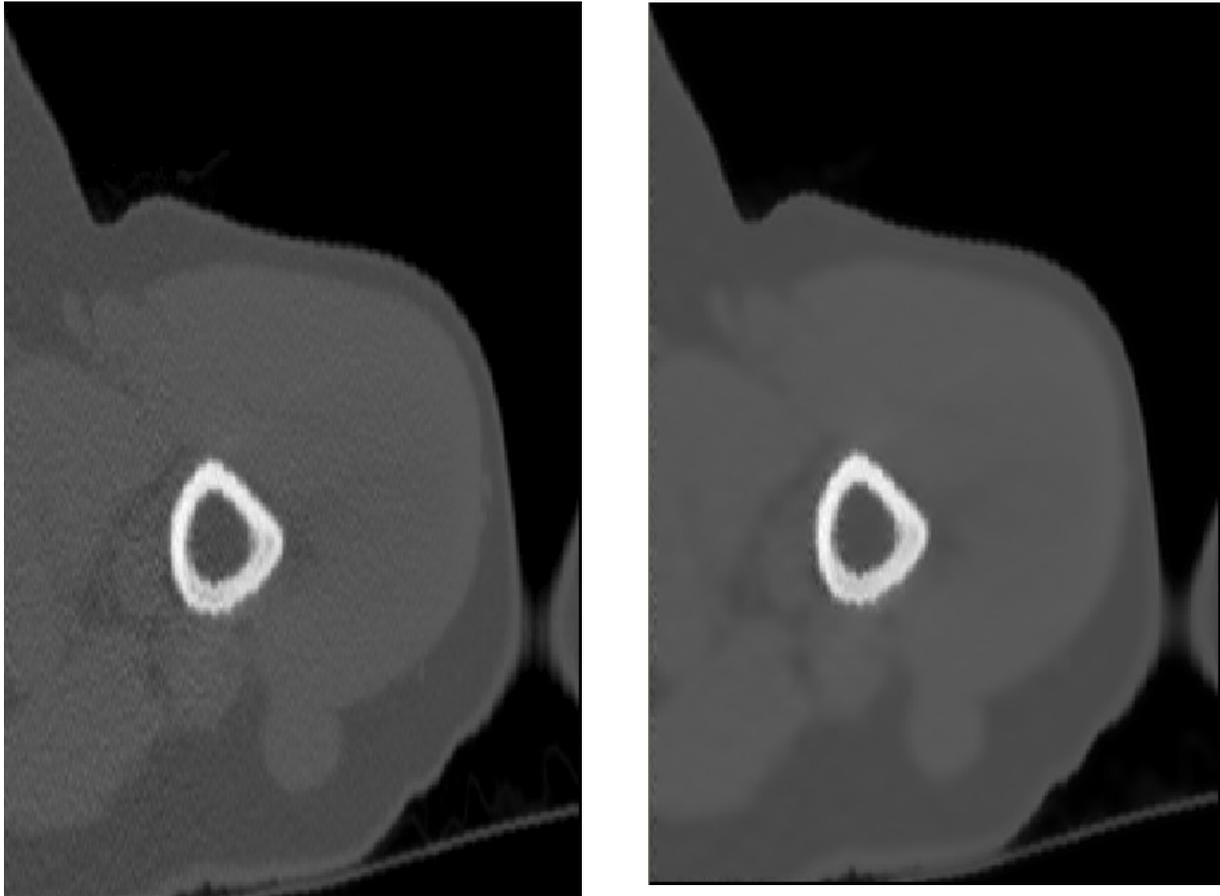


Figure 5.3 - Wiener filter as a spatial filtering technique applied to a sagittal cross section of the upper limb. The image at the left is the original image and at the right the filtered image.

Anisotropic Diffusion

Anisotropic diffusion is a widely used technique in MR brain images aimed at reducing noise and improving contrast [9]. Numerous authors have employed this technique to CT images where it has demonstrated good results [10].

This filter was designed to attain a dual purpose: smooth regions with small differences between pixels and preserve large differences between pixels. This way ubiquitous noise can be deeply attenuated while edges can be maintained.

Although, anisotropic diffusion [11] is a complex iterative filtering process it only requires the manipulation of two parameters, which are the number of iterations and a parameter that affects noise smoothing and edge preservation. By selecting the proper parameter values the resulting output can be quite satisfactory. **Figure 5.4** exhibits the filtering potential of anisotropic diffusion upon CT images.

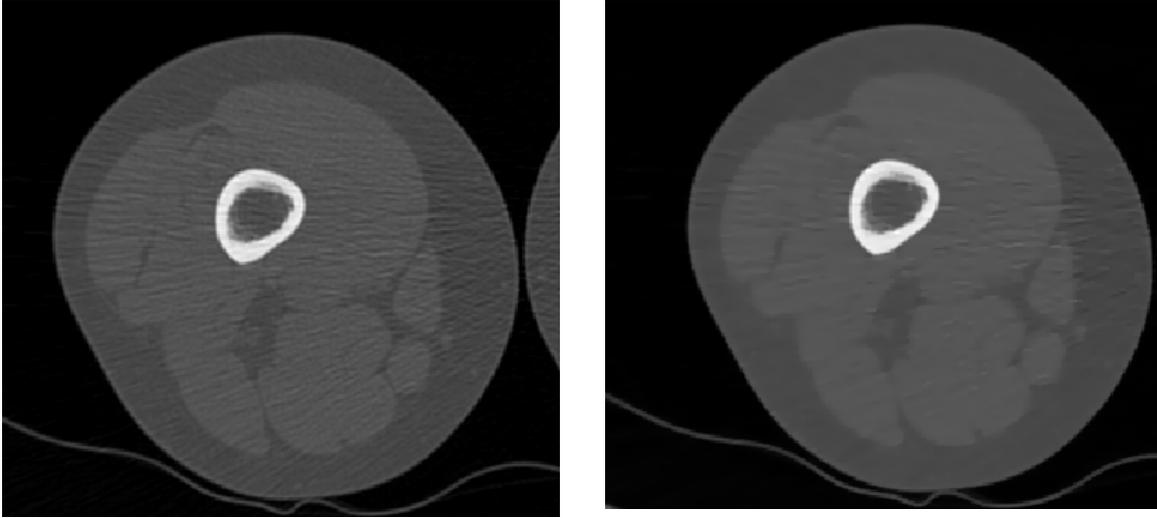


Figure 5.4 – Filtering CT images with the anisotropic diffusion technique demonstrates effective noise reduction and edge preservation.

The mechanism of anisotropic diffusion acts upon each pixel value, which is changed by an amount related to the divergence of the gradient of the image at that pixel multiplied by a diffusion function. Formally, the anisotropic diffusion filtering is expressed by the equation

$$I(x, y, t + \Delta t) = I(x, y, t) + \text{div}[c(x, y, t) \nabla I(x, y, t)], \quad (5.4)$$

where $I(x, y, t)$ is the image at iteration t , $\text{div}(\cdot)$ is the divergence operator, ∇ the gradient operator and $c(x, y, t)$ is the diffusion function. Anisotropic diffusion is a recursive function that generates a family of n related images $I(x, y, t)$, with $t = 0, \dots, n$. The initial condition, $I(x, y, t = 0)$, corresponds to the original image. As t gets higher more diffusion is performed and the image becomes smoother.

To understand the importance of the spatially varying function $c(x, y, t)$, an image must be visualized as a set of patches or regions and the boundary of each region as an edge. Hence, $c(x, y, t)$ is a function defined to execute intraregion smoothing rather than interregion smoothing, so that edge preservation is assured from iteration to iteration. The following equation expresses the most used diffusion function [11].

$$c(x, y, t) = e^{-\left(\frac{|\nabla I(x, y, t)|}{\kappa}\right)^2}. \quad (5.5)$$

The κ in expression (5.5) symbolizes the diffusion parameter. This parameter specifies which edge slopes are to be preserved and which are to be smoothed. As expression (5.5) demonstrates, $c(x, y, t)$ monotonically decreases as the magnitude of the image gradient increases. This way, pixels corresponding to edges have a high $|\nabla I(x, y, t)|$ and therefore $c(x, y, t)$ will

have a very small value, meaning that $I(x, y, t + \Delta t) \approx I(x, y, t)$; whereas for regions with small $|\nabla I(x, y, t)|$ the opposite occurs and smoothing is performed.

The parameter κ and the number of iterations n have a physical significance. The value of κ represents the magnitude of the gradient that is most modified, while constant regions and edges of greater magnitude are practically unaltered. The physical meaning of κ is, therefore, material conductivity. The number of iterations is simply the number of times diffusion occurs; it thus represents the time factor.

6. IMAGE SEGMENTATION

The classical definition of segmentation is quite simple: “image segmentation is defined as the partitioning of an image into non-overlapping, constituent regions that are homogeneous with respect to some characteristic such as intensity or texture” [1]. The quoted phrase is very understandable but difficult to implement for nontrivial images, such as CT images. The fact that there is no universal approach to solve all segmentation problems only demonstrates the difficulty of this subject.

The objective of medical image segmentation is to divide the 2D or 3D image into well defined anatomical structures or regions (**Figure 6.1**). However, when dealing with CT images, segmentation is one of the most difficult tasks to be performed within the entire pipeline. One should bear in mind that a CT image comprises the information relative to the tissue distribution in the human body, but also undesired signals. Artifacts, noise, blurred edges are some of the features common to any medical image.

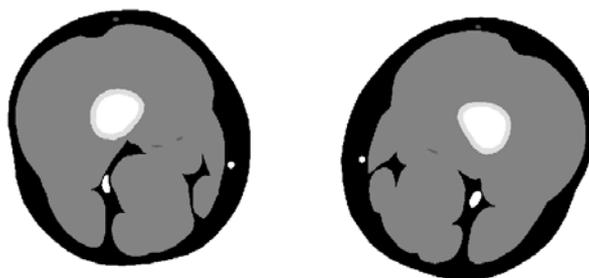


Figure 6.1 – Example of a segmented image of an axial cross section of the thigh. [2]

Segmentation can be considered a step forward in image processing because the input is an image but the output may be an attribute such as a geometric entity (area, volume, perimeter, etc). By extracting attributes from an image, new applications become possible, including 3D model reconstruction of anatomical structures from CT data. These models respond to several clinical needs. For example, by delineating anatomical structures and other regions it is possible to quantify tissue volume, locate abnormal tissue or pathologies, and study the form and structure of the relevant tissues. Consequently, segmentation of anatomical structures aids diagnosis, it is useful for planning treatment and makes computer-integrated surgery possible.

This way, segmentation plays the main role in the designed pipeline for anatomical modeling as it establishes the transition between image data and 3D mesh data.

So that a segmentation algorithm can be effective, it is necessary to tune the parameters according to the characteristics of the image modality used as input.

A myriad of different segmentation methods exist and can be selected according to the input image and classified according to their degree of automation. Manual, semi-automatic or automatic are the categories. The latter is only possible for images with well defined objects which, in general, is not the case of CT data or any other medical image. In CT, bone shaft and skin can be considered exceptions due to the very high contrast they present. Until now, there is no fully automatic algorithm capable of perfectly extracting region boundaries from medical images.

Since the dawns of image processing and computer vision, many semi-automatic segmentation algorithms have been developed and can be applied to different imaging modalities. In particular, to solve the problem of reconstructing 3D surfaces from medical images, several approaches have been taken [2-5].

In the present work, three techniques are used in conjunction to solve the bone segmentation problem: global thresholding, active contour methods and manual segmentation. The next sections describe each technique separately.

Global Thresholding

Global thresholding is based on the property of similarity of intensity values. Despite the inconsistencies present in CT images, an anatomical structure is practically characterized by its CT number. So, by partitioning an image into regions according to the voxel intensity value, a coarse but overall segmented object will result (**Figure 6.2**).

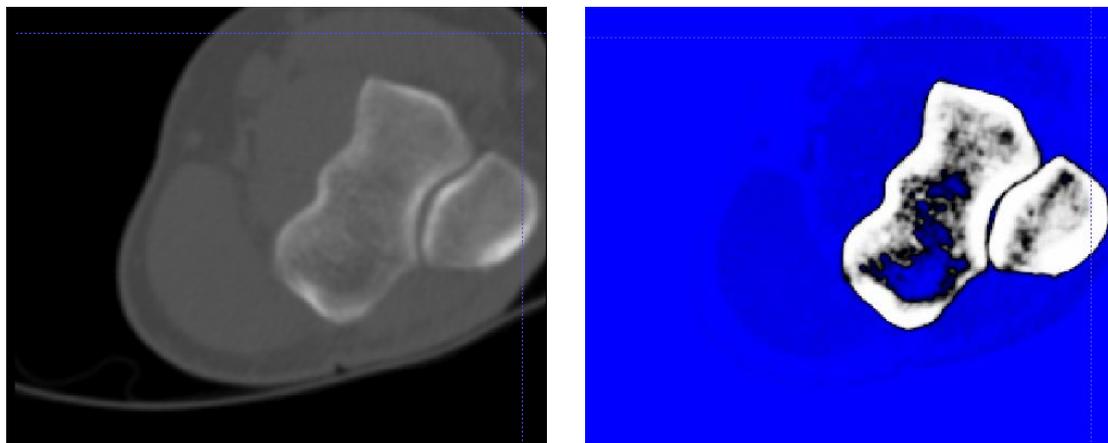


Figure 6.2 – Image segmented solely with global thresholding.

As expected, the mechanism of global thresholding is very simple. Considering the volume histogram, the intensities of the tissue of interest lie between two values, or thresholds. Hence, thresholding consists of selecting the intensities that identify a tissue (foreground) and cancel all the other values (background), estimating in this way the region occupied by the object. The values of a thresholded image range from -1 to 1, where the background and foreground pixels correspond, respectively, to the extremities. Any pixel with a value near zero indicates an edge between foreground and background.

This type of global threshold can be performed as a point processing operation changing each voxel value according to a function that maps the original values to new values. For instance, this mapping transforms the histogram domain to the $[-1, 1]$ interval described above. This mapping is also called intensity region filter and is based on the mathematical expression of a sigmoid. In **Figure 6.3** the intensity region filter used in **Figure 6.2** is presented.

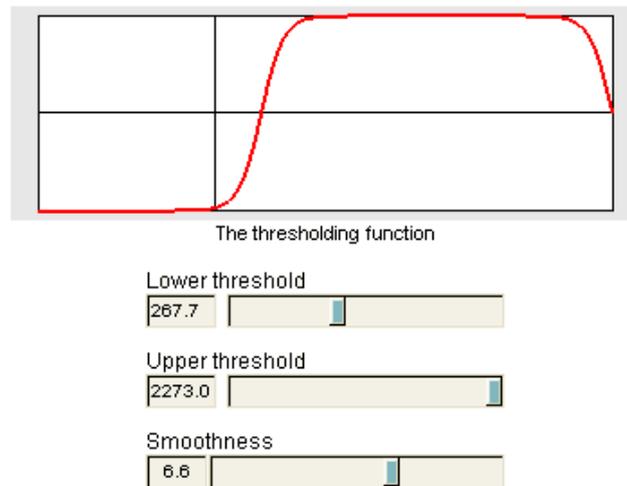


Figure 6.3 – Intensity region filter applied to **Figure 6.2**.

Due to the partial volume effect, the established intensity interval encloses not only bone tissue but also muscle, fasciae and bone marrow. Noise and other artifacts also have their share in this portion of the histogram. This histogram corruption only makes the threshold selection more difficult.

Since the thresholding technique generally does not take into account the spatial information, the segmentation will be very sensitive to noise and inconsistencies. **Figure 6.4** shows a detail of the 3D model of the elbow articulation. As we can see, the global threshold by itself produces a very defective and noisy 3D model.

For medical images, thresholding is very rapid and intuitive, but generally it produces a crude result. In this context another technique must be applied.

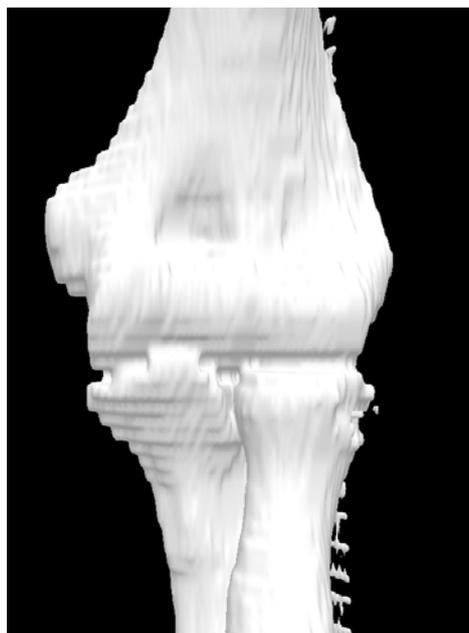


Figure 6.4 – 3D model of the elbow articulation obtained with a global threshold technique.

Active Contour Methods

The objective of segmentation is the determination of the (x,y,z) volume coordinates where bone is present. Nowadays, most of the employed segmentation algorithms for medical images are based on deformable models [6].

Simply put, a deformable model is a closed parametric curve or surface with physical properties that, under the influence of internal and external mechanical forces, deforms adapting to image characteristics. The model can also be referred to as a 2D or a 3D snake and, as a result, the deformation process is referred to as a snake evolution.

The modeling pipeline exploits the 3D active segmentation method known as region competition [7]. A tissue can be outlined by placing an initial set of closed surfaces, such as spherical surfaces, in the proximity of the region of interest. The contour initialization is nothing but a rough estimate of the anatomical structure of interest.

Each evolving snake $C(u, v; t)$ is a closed surface spatially parameterized by u, v and temporally parameterized by t . When submitted to forces, every (u, v) point moves in an iterative manner according to the following PDE:

$$\frac{\partial}{\partial t} C(u, v; t) = F \vec{N} . \quad (6.1)$$

\vec{N} is the unit vector normal to C at the point (u, v) and F is the sum of all forces that act on the snake in the normal direction.

For each step of the temporal parameter t , internal forces derived from the snake's geometry guarantee smooth surface variations, while external forces compel the snake toward the feature of interest, such as an edge. In other words, internal forces act as a smoothness constraint, assuring robustness to noise and spurious edges, external forces originate the deformation leading the model to adjust to the object boundaries.

The region competition method assigns the internal force as the mean curvature of $C(u, v; t)$ and the external forces originate from voxel probability maps that, in turn, derive from the previously described intensity region filter. The probability under consideration refers to how likely a certain voxel belongs to the structure of interest or foreground. The external force is calculated based on the difference between foreground and background probabilities.

Without entering the theoretical aspects, it is recorded here that the PDE (6.1) is solved by using the level set method [7], due to its numerical stability and the possibility for the contour to change its topology.

The process stops when the snakes enclosure all the voxels with greater probability value or until the user finds a suitable solution. The output of the segmentation is a binary volume where 1 represents foreground and 0 background.

The following figure illustrates the overall process taking the tibia as example.

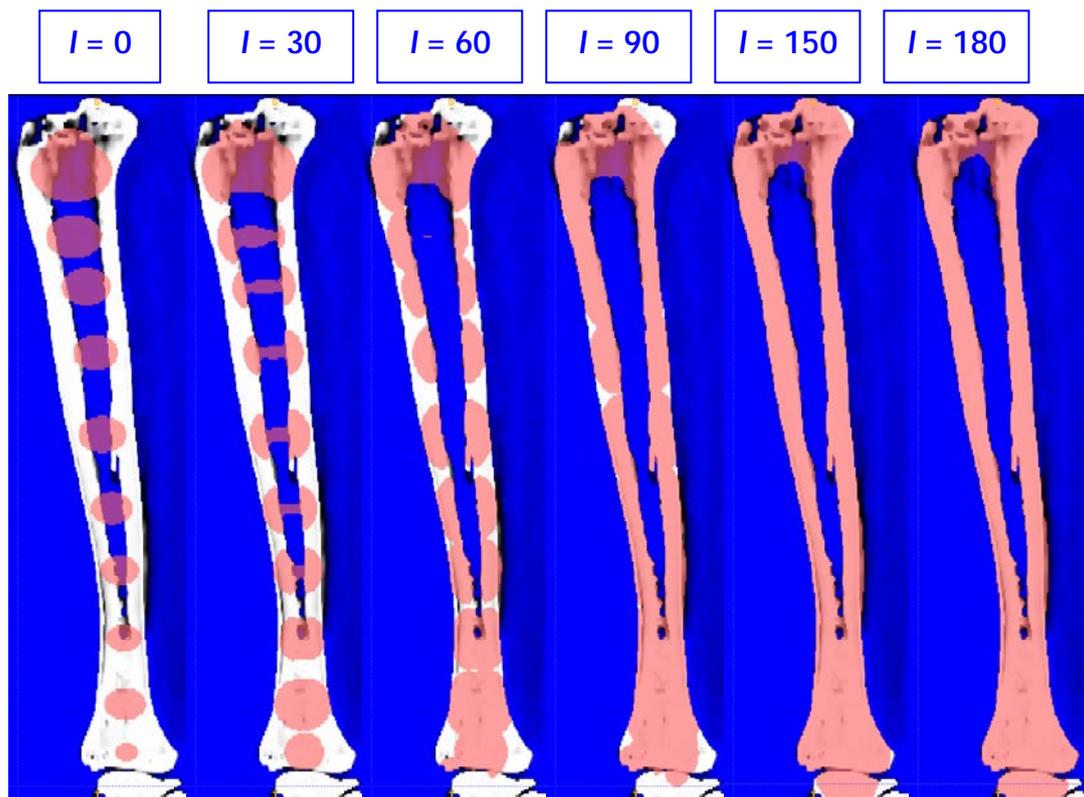


Figure 6.5 – Evolution of a snake from a very rough estimate of the anatomical structure of interest ($I = 0$) to a very close approximation of the structure ($I = 180$).

Manual Segmentation

Considering medical images, a fully automatic segmentation procedure does not exist. Manual and semi-automatic procedures must be combined in order to obtain accurate results.

Therefore, after applying the region competition method, manual segmentation was required. User interaction plays an important role on correcting the segmentation errors produced by the semi-automatic segmentation procedure. To improve the accuracy of manual interaction the user must have *a priori* anatomical knowledge of the structure to be modeled.

Manual segmentation alone guarantees a highly accurate result but the time, effort and training involved are impractical for large-population studies. Combining semi-automatic followed with manual segmentation provides a powerful and reliable instrument for 3D image segmentation.

7. SURFACE MESH GENERATION AND ADJUSTMENTS

After the segmentation stage, processing turns from image to mesh data. The modeling pipeline is now designed to generate and improve surface and volume meshes that are suitable for 3D visualization and for FEA.

A 3D mesh [1] is a discrete entity formed by geometrical elements such as nodes, lines, polygons or volumes. It can be interpreted as a 3D model tiled with polygons or polyhedrons over a three dimensional space with geometrical elements (triangles, quadrangles, tetrahedrons or prisms) arranged to intersect along a face, an edge or a node.

A mesh can be classified as structured (regular) or unstructured (irregular). In a structured mesh the inner nodes are attached to the same number of elements. If the interior nodes are attached to a variable number of nodes then a mesh is called unstructured.

An automatic mesh generator creates a 3D model without user intervention, by discretizing an arbitrary geometry which can exhibit concave, convex, or planar surfaces.

Mesh generation is a bottom-up procedure, i.e., points originate lines then the mesh of the lines is used to generate surface meshes and from them volume meshes.

The 3D models created were unstructured triangular surface meshes and unstructured tetrahedral volume meshes (**Figure 7.1**).

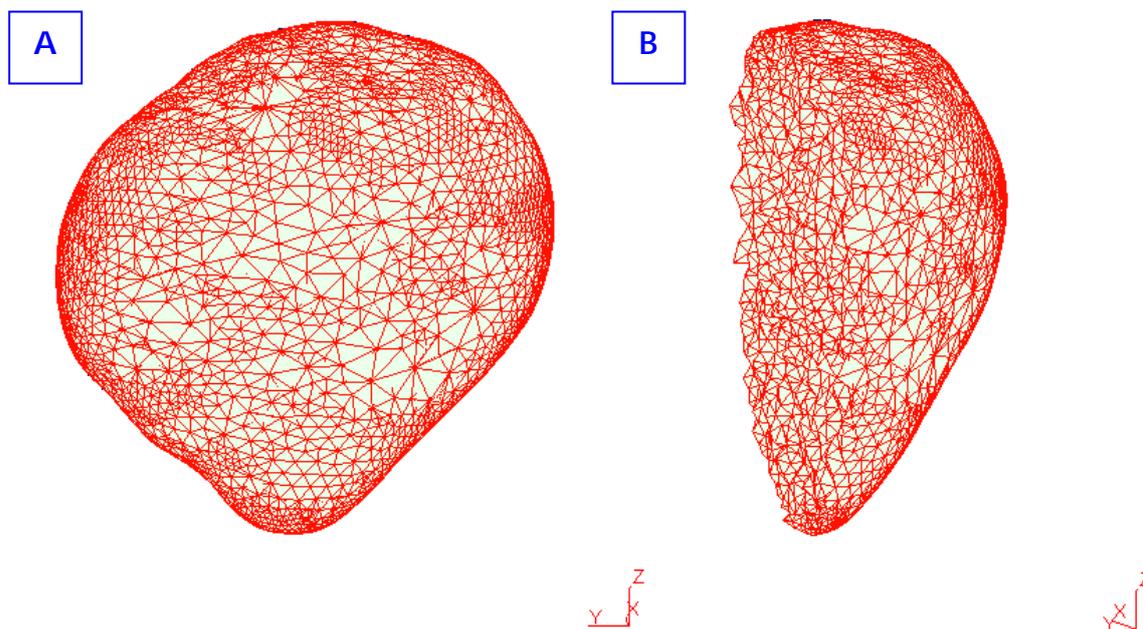


Figure 7.1 – Surface (A) and volume (B) meshes of the patella.

The output of the segmentation stage is a binary volume containing the anatomical structure of interest. To be exact, the anatomical structure is represented as a collection of white voxels surrounded by a dark background (**Figure 7.2**). In this manner, the structure of interest (the set of white voxels) can be interpreted as a volume mesh formed by parallelepiped elements. Although such mesh is composed by structured elements with regular geometries, it is not adequate for the proposed applications (chapter 1) because it is an excessively refined mesh.

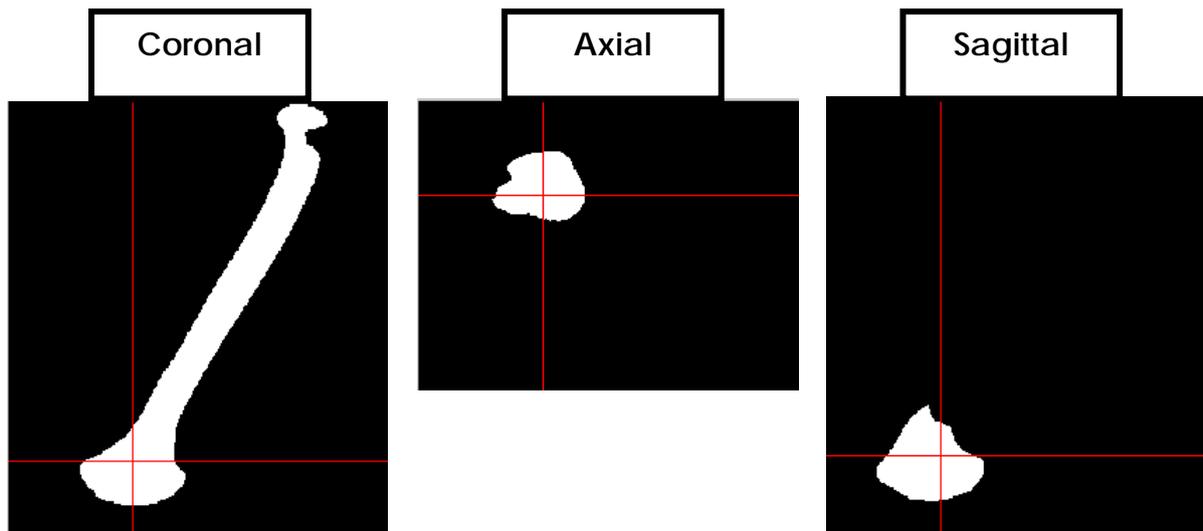


Figure 7.2 – Coronal, axial and sagittal planes of the segmented humerus volume.

In order to obtain the desired meshes from the segmented volume image, the geometry of the structure must be reconstructed by applying geometry modeling techniques such as iso-surfacing by the marching cube algorithm, smoothing, decimation and quadric clustering. Each of these techniques will be described next.

Iso-Surfacing

The marching cube algorithm is a technique that creates a triangular surface mesh from a volume scalar field, such as the previously obtained 3D binary image. The voxels of the 3D image are considered as three dimensional points with an associated intensity value. If the voxels of some cube have some intensity values lower and others above a user-specified value, also known as isovalue, then the voxels that form that cube contribute to the isosurface.

Eight neighboring voxels, from two adjacent segmented images, form a virtual parallelepiped, as shown in **Figure 7.3**. If any of the eight vertices has a different intensity value, then a boundary surface can be traced to separate different vertices (**Figure 7.4**). Considering a parallelepiped, 256 different intensity configurations are allowed, but due to symmetries and rotations only 15 cases are really distinct (**Figure 7.4**).

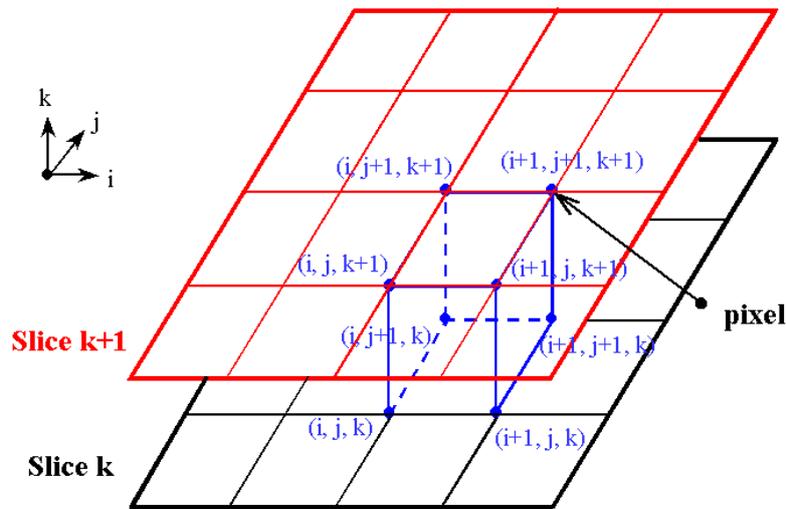


Figure 7.3 – Marching cube framework [2].

As seen in **Figure 7.4**, some intensity configurations may lead to topology defects of the surface mesh, such as holes or spikes. Several methods were developed to cope effectively with those ambiguities [2].

The algorithm marches through each and every parallelepiped and establishes a surface boundary composed by triangular elements. By connecting all the triangular facets an iso-surface will arise.

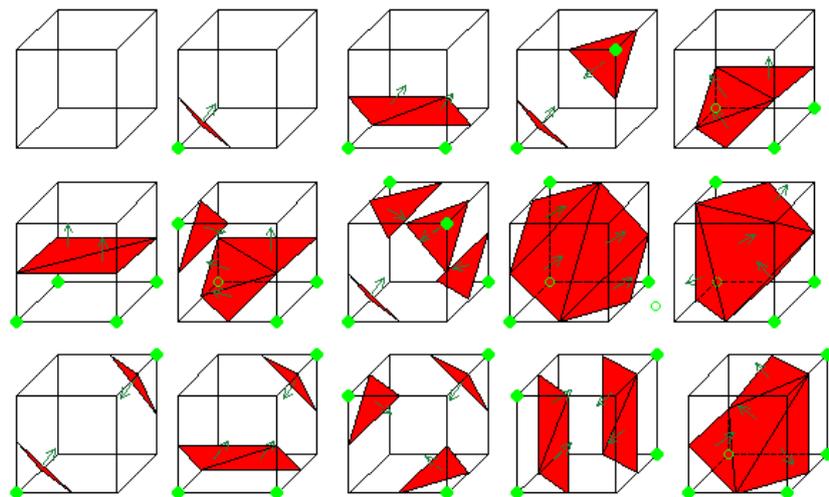


Figure 7.4 – The 15 possible intensity configurations and surface boundaries for the marching cube algorithm. The green nodes represent foreground voxels and the red triangles the facets [3].

The model created by the marching cube algorithm presents two major features that require a careful approach (**Figure 7.5 (A)**): (i) a characteristic stair-step shape surface, very jaggy in some cases, which obviously does not correspond to the natural surface curvature; (ii) an excess of nodes and facets that expresses irrelevant information and hampers further computational processes or simulations.

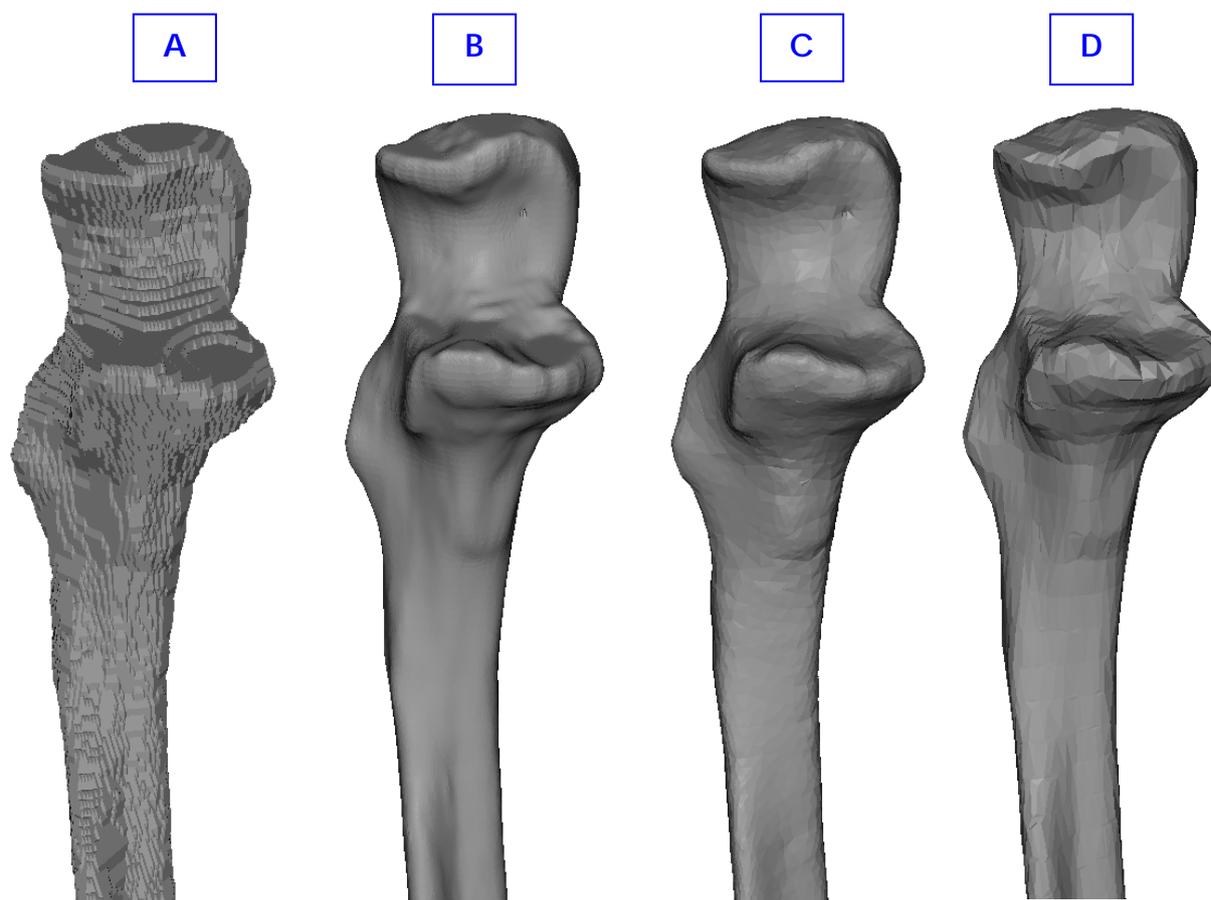


Figure 7.5 – Details of the (A) iso-surface, (B) smoothed, (C) decimated (SDS), and (D) quadric clustered (SQCS) meshes of the ulna.

In order to suppress these undesired features smoothing, decimation, and quadric clustering are some of the techniques commonly used in geometric modeling. Usually, mesh adjustment techniques are not applied individually; instead they are sequentially combined to form a miniature pipeline for mesh quality improvement. According to the application in mind, the bone modeling proposed here took two possible approaches, as listed in **Table 7.1**.

Application	Mesh Adjustment Procedure
Structure Visualization	SMOOTH
Volume Mesh Generation	SMOOTH → DECIMATE → SMOOTH (SDS) or SMOOTH → QUADRIC CLUSTERING → SMOOTH (SQCS)

Table 7.1 – Mesh adjustment procedures according to the application. SDS and SQCS are acronyms for Smooth-Decimate-Smooth and Smooth-Quadric Clustering-Smooth.

Smoothing

As shown in **Figure 7.5**, the surface model is strongly corrupted by jagged and blister-like artifacts. Therefore, the surface requires a low-pass filtering, i.e., smoothing the node’s positions relatively to each other without modifying the mesh topology.

By adjusting the node coordinates the overall mesh appearance and the shape of the triangular elements will be modified. Consequently the mesh geometry will not be preserved but the number of nodes and triangular elements remains equal.

As in image spatial filtering, smoothing is a neighborhood processing operation meaning that the new coordinates of a given node will depend on the position of surrounding nodes. Laplacian smoothing is a very common and effective filter used to rectify the step-like artifacts produced by the reconstruction algorithm, improving the surface mesh appearance.

For a single node p_i at position \vec{x}_i the Laplacian smoothing will allocate p_i to a new position \vec{x}_{i+1} according to the following equation:

$$\vec{x}_{i+1} = \vec{x}_i + \lambda \sum_{j=1}^n (\vec{x}_j - \vec{x}_i), \quad (7.1)$$

where \vec{x}_j are the positions of the n neighboring nodes, p_j , connected to p_i , and λ a user-specified parameter that controls the amount of “smoothness” to be performed upon the surface mesh. The greater the value of λ the smoother the mesh will be.

Decimation

The smoothed reconstructed surface contains a significant amount of rendering elements that grant a realistic appearance to the model. For 3D visualization purposes a large number of nodes and surface elements are required to achieve high resolutions, especially for complicated curvatures.

Conversely, large data sets easily surpass computational capabilities and as a result this excess of data is superfluous for computer simulations. Several techniques are available to sim-

plify surface meshes by reducing the number of nodes and triangles before simulation computations.

The decimation operation [4] is a mesh simplification technique that reduces the total number of nodes and surface triangles. The reduction process, parameterized by the percentage of nodes to be eliminated, is user-specified. Usually, the reduction percentage is very high, in-between 75-90%. Such reduction can appear to be quit harsh but in practice the resulting mesh is a good approximation to the original geometry, although mesh topology may not be preserved.

The decimation algorithm is an iterative process that performs node removal at each pass. Every node is submitted to three steps in order to classify a node as a candidate or non-candidate for removal: at first, the node's local vertex geometry and topology are characterized; depending on the type of node characterization a certain decimation criterion is evaluated, if the node satisfies the criterion, the node and all triangles that use the node will be removed creating a hole in the surface mesh; finally, it is necessary to recoat the hole with triangles. Decimation iterates again over all the nodes until the reduction percentage is reached.

Quadric Clustering

As decimation, quadric clustering is used to reduce the number of nodes and triangles of a triangular surface [5]. The resulting mesh is close to the original geometry although the topology is not preserved.

The algorithm starts by partitioning the boundary domain in a user-specified number of uniform rectilinear cells. Within each cell, the nodes of the triangular elements are reduced to only one vertex. Thus, nodes are said to be clustered.

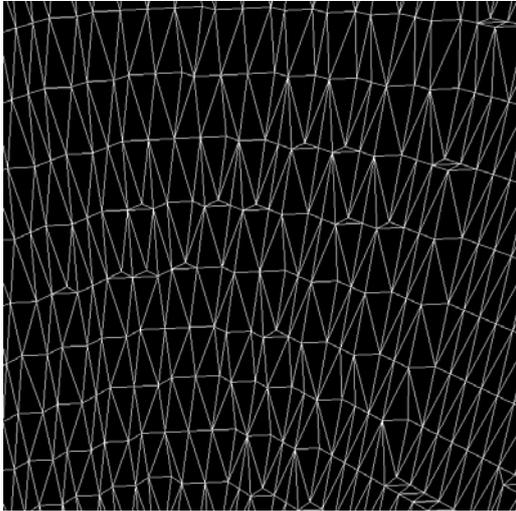
The clustering criterion determines the optimal position of a cell's new vertex by minimizing the sum of the squared volumes of all the tetrahedral elements formed by the new vertex and each of the surface triangles. The vertex that satisfies this criterion represents the new vertex of the simplified mesh at that cell.

With the new vertex determined the algorithm proceeds to re-triangulate the surface. Essentially, what happens is that all the nodes contained within a cell are replaced by the new vertex, and all the exterior nodes that connect to these interior nodes will establish an edge with the new vertex.

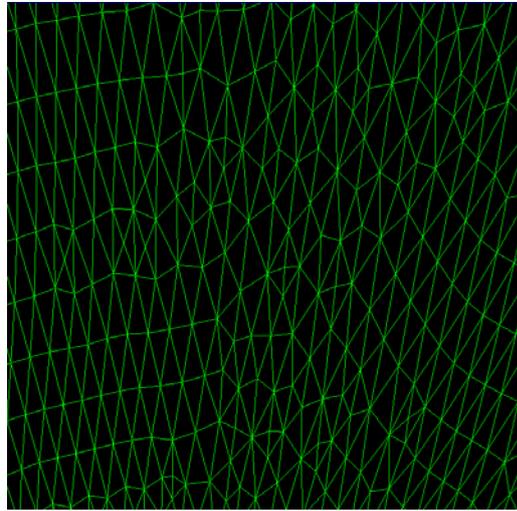
For instance, if a surface triangle has two nodes contained within the cell, these two nodes are replaced by the new vertex and an edge is established between the outer node and the cell's vertex. If all three nodes of a triangle are contained within the cell, they will be replaced by the cell's vertex. This way the number of surface triangles is reduced and the final mesh is much simpler than the original mesh.

The following figure presents the appearance of the triangular surface mesh for each of surface generation and adjustment algorithms described.

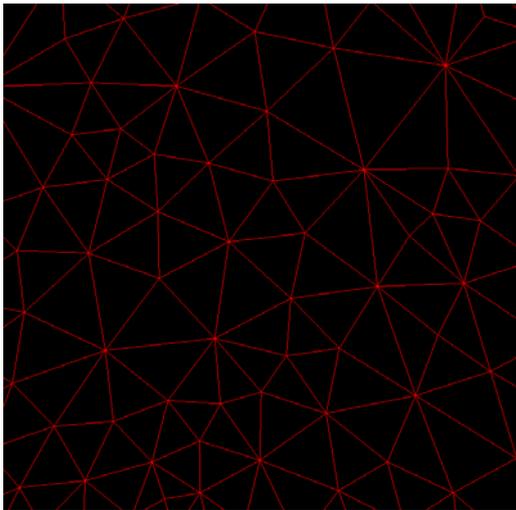
Iso-Surface



Smoothing



Decimate (SDS)



Quadric Clustering (SQCS)

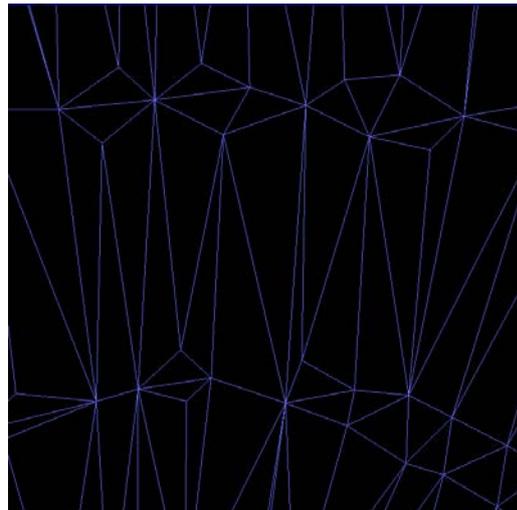


Figure 7.6 – Amplified portion of the iso-surface, smoothed, decimated (SDS), and quadric clustered (SQCS) surface meshes of the ulna.

8. VOLUME MESH GENERATION

Delaunay Tetrahedralization

Based on the segmented image data, the 3D reconstruction creates a triangular surface mesh that, after a few proper adjustments, represents the boundary of an anatomical structure. With the purpose of performing finite element simulations, volume meshes are generated from the boundary surface mesh. In order to do so, geometric elements are automatically built in the interior of the three dimensional domain, decomposing it into smaller volumes. The construction of these volume finite elements starts from a set of points and is constrained by the surface boundary.

The finite element models created are tetrahedral meshes. Therefore the type of mesh generator used is designated as an automatic mesh generator (AMG) of unstructured tetrahedra. Other types of mesh generators are capable of producing structured and/or hexahedral finite element meshes [1].

Some mesh generators, just as the AMG utilized [2], impose two constraints on the construction of a volume mesh: (i) the mesh of the volume space is entirely constrained by the mesh of its surface boundary, meaning that the triangles that discretize the boundary surface are facets of the outer tetrahedra in the final 3D mesh; (ii) each constructed volume element will respect a characteristic length. The characteristic length is a measure, necessary to establish the size of the finite elements. This parameter is chosen according with the geometric details of the structure.

One of the most common mesh generation algorithms is based on the Delaunay tetrahedralization [1]. This process of volume discretization has its foundation on the empty sphere criterion. **Figure 8.1** illustrates the mechanism behind the Delaunay criterion by exemplifying it with triangles. The 2D equivalent of tetrahedralization is called triangulation.

Tetrahedralization is the operation that transforms a set of points (more than 4) into tetrahedrons. So, given a set of nodes, the construction of tetrahedra considers a cluster of four candidate points for the vertices of a Delaunay tetrahedron. A sphere interpolates these four points and if no vertex is contained within the circumsphere, i.e., if the “sphere is empty”, the tetrahedron is a Delaunay. The empty sphere criterion is a necessary and sufficient condition to classify a tetrahedron as a Delaunay tetrahedron.

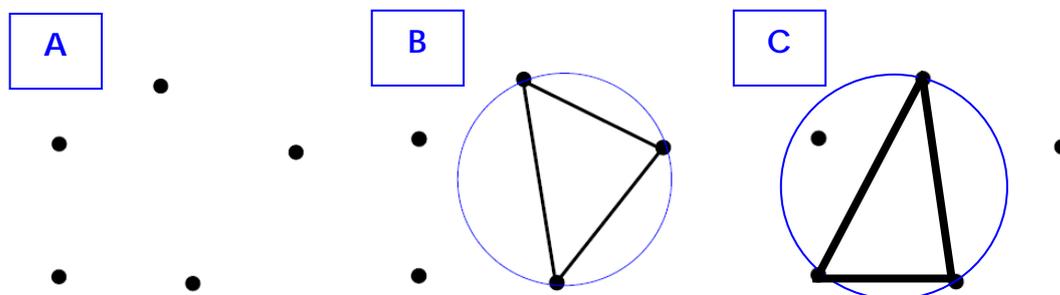


Figure 8.1 – (A) Set of initial points. (B) Delaunay empty circle (sphere) criterion applied to a set of points. (C) The triangle does not satisfy the Delaunay criterion.

As figure **Figure 8.1** exemplifies, the Delaunay criterion is a procedure that connects a cloud of points, but by itself it is not the mesh generation algorithm. Constructing a volume mesh involves more methods and techniques, such as, generating nodes on the surface boundary of the domain, inserting nodes within the volume of the domain, recovering the original surface boundary, and optimizing the mesh quality. In this way, the Delaunay criterion can be considered the kernel of this type of volume mesh generator. Depending on the engineering application, several Delaunay-based algorithms have been developed. They differ from each other on the methods employed for node insertion and boundary recovery.

Besides the Delaunay tetrahedralization method, the utilized AMG comprises an additional algorithm. The Bowyer-Watson algorithm establishes how nodes are inserted. In brief, given a Delaunay tetrahedralization (or triangulation) and a newly inserted node, the Bowyer-Watson algorithm locates the tetrahedron (triangle) that contains the inserted node and searches all tetrahedrons (triangles) whose circumsphere (circumcircle) contains the node and deletes tetrahedra (triangles), creating a hole in the mesh. Finally, new elements are formed from the node and the boundary of the hole by Delaunay tetrahedralization (or triangulation). **Figure 8.3** illustrates the Bowyer-Watson algorithm.

The AMG used for the anatomical bone structures has, essentially, five steps (to ease the visual interpretation, each step will be accompanied with a figure of an elementary bi-dimensional example, giving an idea on how the process evolves):

- 1 – At first, a simple mesh box is made (the red mesh in **Figure 8.2**) involving all the boundary surface nodes. This box is the initial volume mesh.

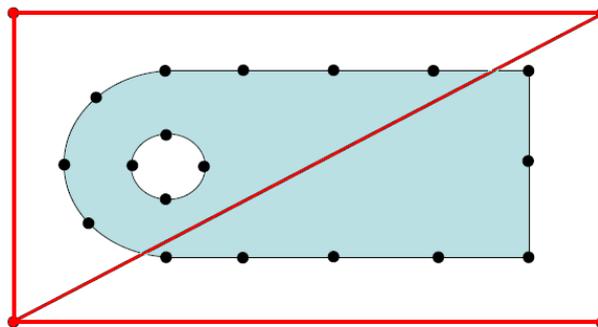


Figure 8.2 – Bounding mesh box composed of two larger triangles (red).

- 2 – The surface nodes, or external nodes, are inserted one at a time and the Bowyer-Watson method discretizes the mesh box at each pass. This way, vertices of the surface triangles are used as an initial cloud of points by the Delaunay tetrahedralization.

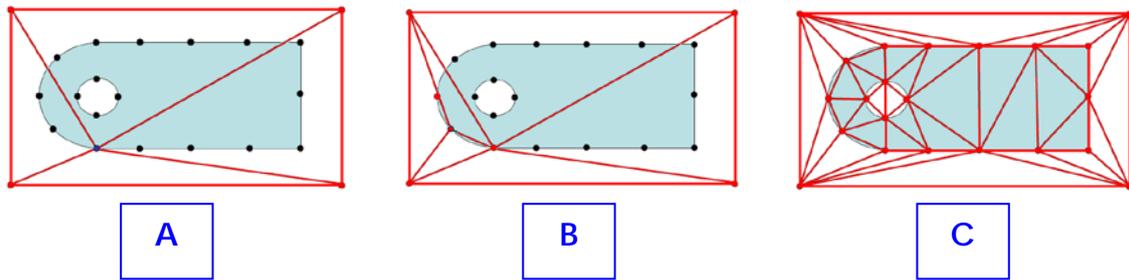


Figure 8.3 – Inserting boundary nodes using the Bowyer-Watson algorithm. (A) Initial iteration. (B) Second iteration. (C) Final iteration.

3 – By eliminating the outer finite elements, i.e., the elements that do not belong to the volume domain defined by the structure, the surface boundary is recovered forcing the triangular facets of the surface to be present on the volume mesh.

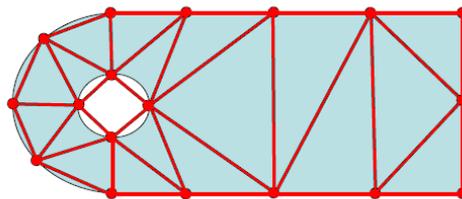


Figure 8.4 – Boundary recovery by eliminating all the finite elements containing nodes of the initial box.

4 – Internal nodes are then inserted in the volume domain. The nodes introduced are based on a regular or isotropic lattice of tetrahedral form. The outside nodes are ignored.

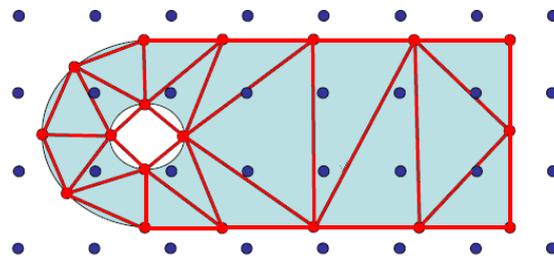


Figure 8.5 – Lattice based internal node insertion.

5 – With the internal nodes inserted, the Bowyer-Watson algorithm enters again in performance and this time tetrahedralization (triangulation) must respect the established characteristic length. The size of each generated finite element will be less or equal to

the characteristic length, which is evaluated at the center of its circumscribed sphere (circle).

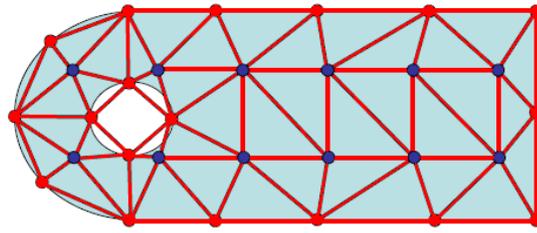


Figure 8.6 – Final mesh.

This type of volume mesh generator does not alter the initial surface mesh due to boundary recovery methods (step 3). Thus, the triangular surface elements must have the final desired sizes which imply that the surface mesh must not have gaps, holes or hanging nodes in order to be numerically consistent and well-conditioned. This must be guaranteed before step 1.

Even if the triangular facets of the boundary surface mesh are well-shaped this does not prevent the tetrahedrons from being very ill-conditioned. In some cases elements with a volume very close to zero, called slivers, may appear contributing to a worse mesh quality.

Whenever a newly inserted node (step 4) lies on the circumsphere surface of a tetrahedra the Delaunay kernel may create a geometrical degenerate case which expresses it self as a valid tetrahedralization, but no longer Delaunay, or an invalid tetrahedralization. Overlapping elements or elimination of existing vertices are possible situations when invalid tetrahedralization occurs. The reason why degenerated cases may appear is due to truncation errors that take place when calculations are performed. These errors can mislead the Delaunay criterion creating the degenerated cases. To surpass this problem, a slight numerical perturbation is introduced on the coordinates of the problematic points, relaxing the constraint of the Delaunay mesh.

The two aforementioned problems are intrinsically related to the implementation of the Delaunay kernel.

Mesh Quality and Volume Visualization

The generation of a tetrahedral mesh is constrained by the geometry of the volume domain, together with certain constraints for size and shape of the mesh elements, such as the characteristic length. To evaluate how suitable a volume mesh is for solving partial differential equations by finite element methods it is necessary to measure the quality of the mesh elements [3]. By associating a quality measure that depends on the geometric parameters of the finite element one can establish the ideal element and identify degenerate cases.

For the element type considered here, a regular geometry (all facets are equilateral) could correspond to the ideal tetrahedron shape. On the other hand, a sliver is the opposite of the perfect element geometry; hence, such degenerated element is not desired.

A quality measure outputs a value that expresses how regular a finite element is. For tetrahedral meshes, the quality measure considered here is quite simple but intuitive:

$$\rho = \frac{\min\{edge_length\}}{\max\{edge_length\}}. \quad (8.1)$$

ρ provides a good measure of the regularity of a tetrahedron. For instance, if the edges are all equal, then $\rho = 1$, and that corresponds to a regular tetrahedron. If $\rho \approx 0$, then at least one of the edges is very short comparatively to all others. Such a low value can only match a sliver-like element.

The overall mesh quality can be visualized by plotting the number of elements versus the quality measure. This graph is therefore the quality histogram and it allows a clear measure towards mesh quality.

A less quantitative approach to mesh quality relies on volume visualization. This can be done by selecting only the elements whose quality measure is above or below a certain threshold or by clipping planes (**Figure 9.12, B&C**).

Clipping planes are used to visualize the interior of a complex structure exposing the inner volume elements. A set of cross sections or planes are created to select a region of interest. Each clipping plane is defined by four coefficients A , B , C , and D of the explicit plane equation $Ax + By + Cz + D = 0$.

9. BONE MODELING PIPELINE FOR THE UPPER AND LOWER LIMBS

This chapter is written having in mind anyone wishing to use the present pipeline. Therefore, the objective of this chapter is to describe hardware and software specifications, putting together all the procedures that compose the pipeline as well as all the required parameter values. The parameter values given here are those specified for the 3D images studied in the present work.

Image Acquisition

Two volume images of the body extremities (left upper extremity and lower right extremity) were acquired from two different subjects.

The image acquisition was performed in accordance with clinical standards. Each scan was taken in just a few seconds, which is advantageous because fast scans improve image quality by decreasing the possibility of patient motion. The scanning mode was helical.

The image acquisition hardware was the high-speed, multi-slice CT scanner Mx8000 from Philips[®]. This system has one of the most advanced algorithms available today for image reconstruction: the cone beam reconstruction algorithm. More technical details concerning this equipment can be found in the site of Philips Medical Systems [1].

Some of the acquisition parameters are grouped in the following table.

UPPER EXTREMITY	LOWER EXTREMITY
<p style="text-align: center;">ACQUISITION PARAMETERS</p> Modality: 'CT' Manufacturer: 'Philips' ManufacturerModelName: 'Mx8000' KVP: 90 XrayTubeCurrent: 233 Exposure: 99 ScanOptions: 'HELIX' SpacingBetweenSlices: 1.3000 XVoxelSpacing: 0.666016 YVoxelSpacing: 0.666016 Width: 512 Height: 512	<p style="text-align: center;">ACQUISITION PARAMETERS</p> Modality: 'CT' Manufacturer: 'Philips' ManufacturerModelName: 'Mx8000' KVP: 90 XrayTubeCurrent: 245 Exposure: 70 ScanOptions: 'HELIX' SpacingBetweenSlices: 1.3000 XVoxelSpacing: 0.666016 YVoxelSpacing: 0.666016 Width: 512 Height: 512

Table 9.1. – CT acquisition and image parameters of each volume data (information extracted from the DICOM metafile).

3D Image Parameters

The input data of the pipeline is a set of DICOM image files that is ordered according to the longitudinal axis (or by the acquisition time). **Table 9.2** lists the parameters of the volume images.

UPPER EXTREMITY	LOWER EXTREMITY
3D IMAGE PARAMETERS	3D IMAGE PARAMETERS
BitDepth: 12 ColorType: 'grayscale' SliceThickness: 1.3000 SamplesPerPixel: 1 Rows: 512 Columns: 512 NumberOfSlices: 567 VolumeMemoryUsage: 294 MB	BitDepth: 12 ColorType: 'grayscale' SliceThickness: 1.3000 SamplesPerPixel: 1 Rows: 512 Columns: 512 NumberOfSlices: 896 VolumeMemoryUsage: 456 MB

Table 9.2 – 3D image parameters of each data set (information extracted from the DICOM metafile, except for the number of slices and memory usage).

Computer Requirements

Dealing with large data sets involves lots of computer memory and fast calculations capacities. For visualizing such data, the graphic cards must have a great performance to ensure an efficient interface with the user. Geometric modeling therefore requires large memory space, both RAM and ROM, GHz CPUs and advanced graphical cards.

The workstation for data processing and visualization was composed by three computers:

- PC computer with Windows XP (Service Pack 2) as operating system, Intel® Pentium® Dual-Core processors of 2.80 GHz and 2GB RAM. The graphical card: NVIDIA GeForce 7800 GTX with 256 MB of memory.
- PC computer with Suse Linux 9.1 as operating system, Dual Opteron 200 of 64 bits processor and 4 GB RAM. The graphical card: NVIDIA Quadro with 256 MB of memory.
- Apple PowerMac G4 with Mac OS X 10.4.5 as operating system, PowerPC G5 1.8GHz and 1 GB de RAM. The graphical card: ATI Radeon 9600 graphics chip with 128 MB of VRAM.

Software Specifications

Various software applications were used to model the anatomical structures. Most of the software is freeware and open source, entirely licensed for academic research and education. The commercial software is common in the academic community.

Several modeling and visualization applications are available in the internet. 3D Slicer, 3D Doctor (free demo) and OsiriX are three examples of highly functional and versatile medical image applications that are used in clinical practice (although 3D Slicer is not approved by the Food and Drug Administration).

Table 9.3 and **Table 9.4** summarize the main features of the software used [2-7].

Software Pipeline

To understand how the modeling was done it is necessary to track each step of the pipeline and to know which software to use. **Figure 9.1** presents the software pipeline together with the input/output file types. This figure is undoubtedly an important diagram as it gives a fore-taste of the mechanism behind geometric modeling.

Each software has a role in the pipeline. MATLAB® performs spatial filtering, surface generation and file conversion. ITK-SnAP is responsible for image segmentation. Blender, although being an advanced modeling software, is only used for file conversion. Surface adjustments and surface model visualization are assured by ParaView. Gmsh automatically generates the volume mesh and ABAQUS® is a powerful finite element solver.

The file types highlighted in **Figure 9.1** correspond to medical images (*.dcm, *.hdr, *.img), 3D models (*.obj, *.stl, *.ply) and native files (*.inp, *.msh). Except for the ABAQUS® input file (*.inp) and the GMS input file (*.msh), the rest of the files are common standards and usable in other software.

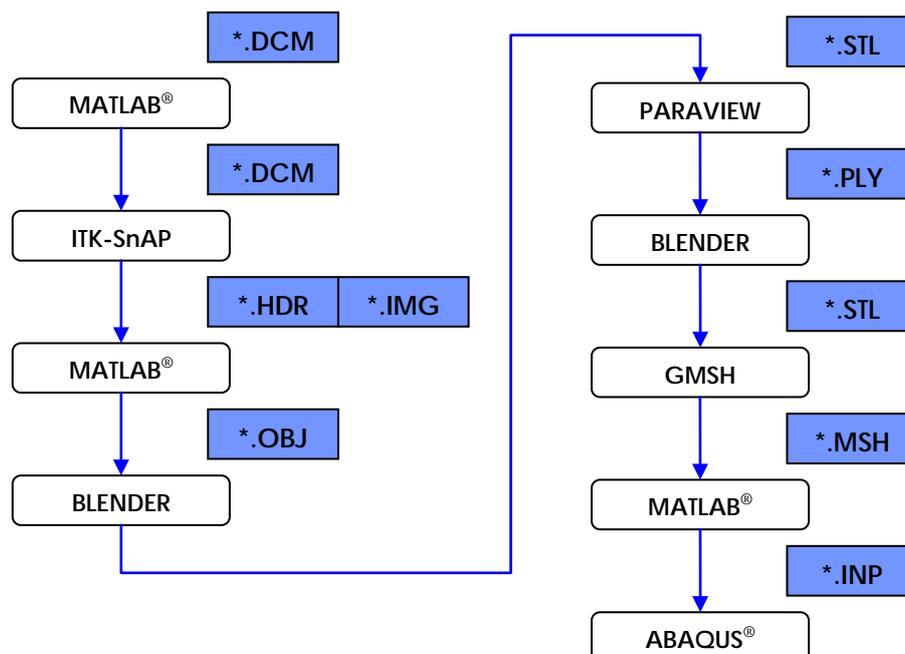


Figure 9.1– Diagram of the software pipeline. The white boxes present the software name and the blue boxes contain the data file extension.

Name	MATLAB <small>Mathworks</small>	ITK-SNAP <small>ITK-SNAP</small>	ParaView <small>Kitware</small>	GMSH <small>GMSH</small>	ABAQUS <small>ABAQUS</small>	blender <small>blender</small>
Synopsis	Technical computation and data visualization with a high-level language and interactive environment.	Segmentation of structures of 3D medical images by semi-automatic active contour methods and manual delineation. Provides image navigation.	Visualization of large data sets. Several algorithms take part of its repertoire. The user interface is very intuitive.	Automatic 3D finite element grid generator (primarily Delaunay) and post-processor.	Powerful finite element solver.	Software for 3D modeling, animation, rendering, post-production, active creation and playback.
Version	7.0.4	1.40	2.4, release 3	1.65	6.4	2.41
Author	Mathworks	Cognitica Corporation (under a contract from the National Library of Medicine)	Kitware	Christophe Geuzaine and Jean-Francois Remacle	ABAQUS Inc.	Stichting Blender Foundation
License	Commercial	Freeware Open source	Freeware Open source	Freeware Open source	Commercial	Freeware Open source
Platform	Windows/Linux/Macintosh	Windows/Linux/Macintosh	Windows/Linux/Macintosh	Windows/Linux/Macintosh	Windows/Linux/Macintosh/Solaris	Windows/Linux/Macintosh/Solaris
Language	-	ITK VTK FLTK	C++ class library, and several interpreted interface layers including Tcl/Tk, Java, and Python	-	-	C, C++ and Python
Interface	GUI / CLI	GUI	GUI	GUI / CLI	GUI / CLI	GUI
URL	http://www.mathworks.com	http://www.cognitica.com/ http://www.itksnap.org/	http://www.paraview.org http://www.kitware.com	http://www.geuz.org/gmsh	http://www.abaqus.com/	http://www.blender.org

Table 9.3 – Software of the geometric modeling pipeline. (The acronyms GUI and CLI are, respectively, for graphical user interface and command line interface).

Name	 3DDoctor 3D-DOCTOR Software Able Software Corp.	 3D Slicer	 OSIRIX
Synopsis	Software for 3D modeling, image processing and measurement for MRI, CT, PET, microscopy, scientific, and industrial imaging applications.	Software for visualization, registration, segmentation, and quantification of medical data.	Works as a DICOM PACS workstation for medical imaging and an image processing software for medical research, functional imaging, 3D imaging, confocal microscopy and molecular imaging.
Version	4.0.060404	2.5	2.3.1
Author	Able Software Corp.	MIT Artificial Intelligence Lab and the Surgical Planning Lab at Brigham & Women's Hospital	Antoine Rosset
License	Commercial	Freeware Open source	Freeware Open source
Platform	Windows	Windows/Linux/Solaris	Macintosh
Language	3D Basic	C++ Tel VTK	VTK ITK Papyrus DICOM Offis PixelMED OpenGL X-Grid Cocoa
Interface	GUI / CLI	GUI / CLI	GUI
URL	http://www.ablesw.com/3d-doctor/	http://www.slicer.org/	http://homepage.mac.com/rossetantoine/osirix/

Table 9.4 – Other software also used for geometric modeling and data visualization.

Image Analysis and Image Navigation

Image analysis consisted on visualizing the image histogram, amplifying image regions by zooming in and out, analyzing the pixel value distribution and performing simple contrast adjustments. The Image Processing Toolbox of MATLAB has a function called `imtool(.)` that allows this type of data analysis (**Figure 9.2**).

Navigating through the volume data and visualizing simultaneously the three main reformatted planes is very useful, specially to start drafting the image enhancement and restoration pipeline. ITK-SnAP is the most suitable software for this task (**Figure 9.3**).

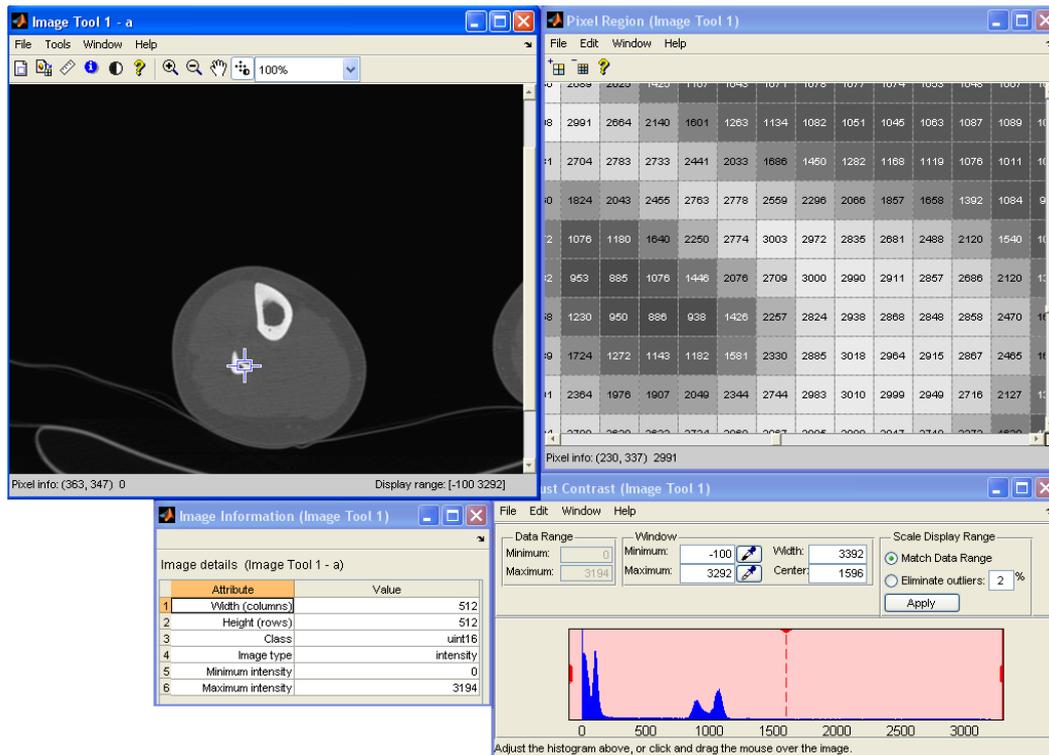


Figure 9.2 – Analyzing an image with the `imtool(.)` function.

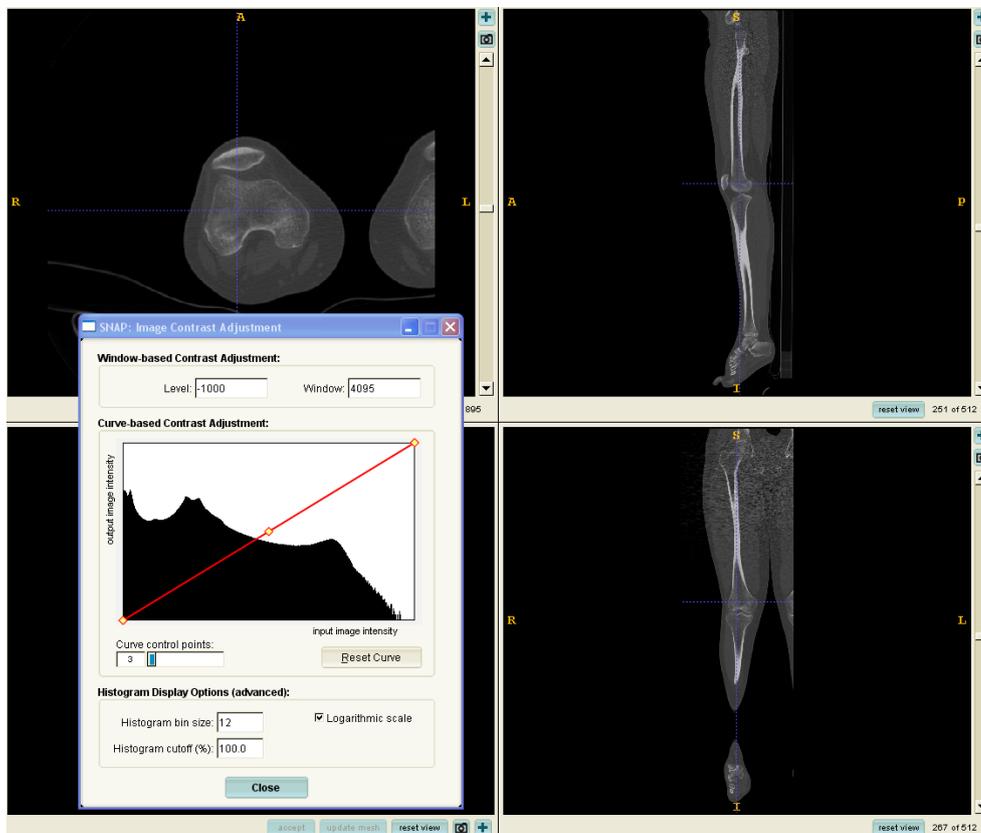


Figure 9.3 – Image navigation and volume histogram with ITK-SnAP.

Image Enhancement and Restoration

The anatomical structures modeled were the humerus, radius, and ulna, of the left upper extremity, and the femur, patella, tibia, and fibula, of the lower right extremity. To model each of these structures, the volume set was divided in sub-volumes. This division was necessary due to memory limitations and because different filtering approaches were applied. For instance, the humerus and femur were furthermore divided in two sub-volumes: the shoulder and pelvic articulations presented streak-like artifacts along the axial plane and adaptive filters (Wiener and median) were applied to minimize the artifact effect. All the other sub-volumes were subjected to anisotropic diffusion filtering.

Table 9.5 systematizes the filtering pipeline and the corresponding parameters for each sub-volume.

Sub-Volume	Filter Pipeline	Parameters			
Humerus	ANISO→GAUSSIAN→WIENER	Aniso	N = 10	$\kappa = 40$	$\lambda = 0.25$
		Gauss	Size = 5x5	$\sigma = 2$	-
		Wiener	Size = 7x7	-	-
Radius & Ulna	ANISO	Aniso	N = 10	$\kappa = 40$	$\lambda = 0.25$
Femur	ANISO→GASSIAN→MEDIAN	Aniso	N = 10	$\kappa = 40$	$\lambda = 0.25$
		Gauss	Size = 5x5	$\sigma = 2$	-
		Median	Size = 7x7	-	-
Patella	ANISO	Aniso	N = 10	$\kappa = 40$	$\lambda = 0.25$
Tibia & Fibula	ANISO	Aniso	N = 10	$\kappa = 40$	$\lambda = 0.25$

Table 9.5 – Filtering pipelines and parameters of each sub-volume.

N – number of iterations; **κ** – conduction coefficient; **λ** – diffusion speed; **σ** – Gaussian standard deviation; **Size** – mask dimension.

The MATLAB functions used were `imfilter(.)`, `medfilt2(.)`, and `wiener2(.)`. The `anisodiff(.)` function was written by Peter Kovesi and can be downloaded at [9].

The filtering was performed for each sagittal plane of the volume. Taking the humerus sub-volume as an example, a sagittal plane is selected and then `anisodiff(.)` is applied; then the resulting image is filtered by a Gaussian followed by a Wiener filter. This final image is then saved. The iteration continues to the next sagittal plane.

Filtering along the sagittal plane provides better results when compared to filtering along the axial plane. Comparing the results of these two filtering strategies with the original image demonstrates the preference for sagittal filtering (**Figure 9.4**).

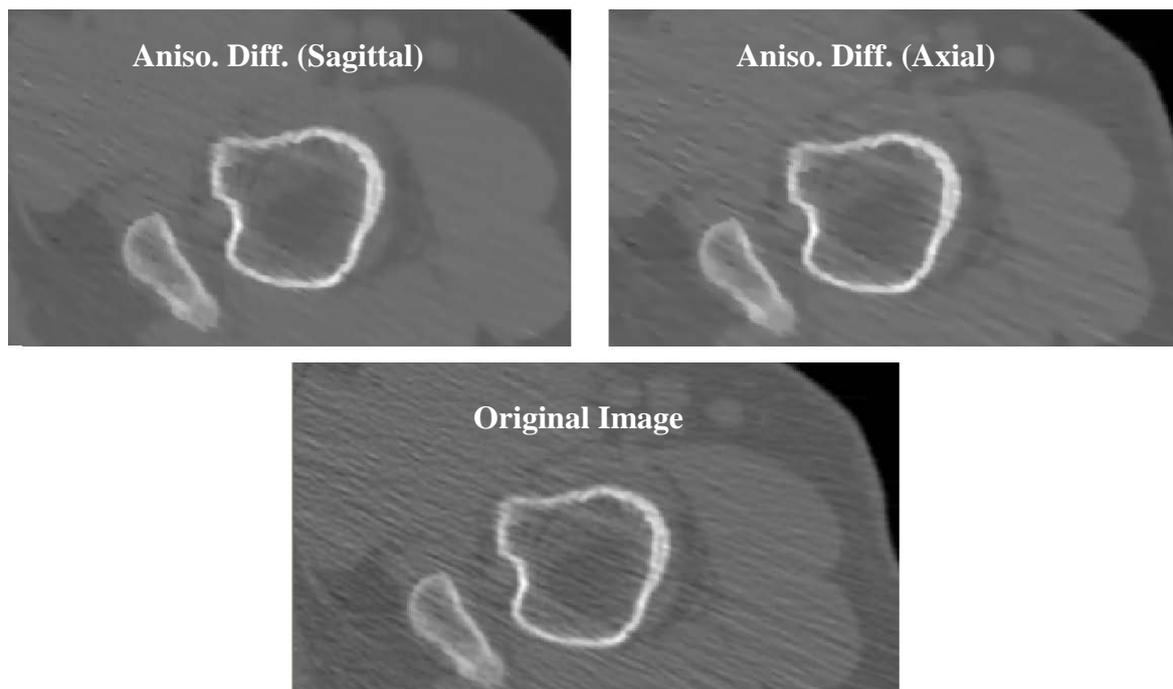


Figure 9.4 – Example of anisotropic diffusion filtering along the sagittal and the axial directions.

Contrast Adjustment and Image Segmentation

Each sub-volume is submitted to ITK-SnAP for 3D image segmentation.

The first step is to enhance the bone tissue by adjusting the contrast with a piece-wise function. This way the intensity transition between bone and soft tissue is greatly improved (**Figure 9.5**). Then a region of interest (ROI) is selected, which can be considered as a primary segmentation procedure. The ROI is a parallelepiped that contains the bone structure (**Figure 9.7**). According to the method implemented in ITK-SnAP [10], the intensity region filter is required to create the probability map of bone voxels. The greater the value (the more whiter the voxel) the greater the probability of being a bone voxel is (**Figure 9.6**).

The active contours, or 3D snakes, are initialized by placing several “bubbles” (spherical boundaries) throughout the bone structure (**Figure 9.9**). ITK-SnAP allows the snake parameters to be specified in an intuitive mode, a mathematical mode or an advanced mode. By choosing the intuitive mode, it is only required to set the balloon and curvature forces. The balloon force is responsible for the contraction or expansion of the snake and is proportional to the intensity probability map. The curvature force affects the smoothness of the 3D snake.

The segmentation procedure then iterates. In each iteration, the 3D snakes expand and adjust to the bone tissue according to the image characteristics until convergence or until the user finds a reasonable solution (**Figure 9.9**). Manual segmentation is required to correct the segmentation faults. **Figure 9.8** illustrates a manual procedure. The segmentation output is a binary volume where 1 represents the bone tissue and 0 the background. **Table 9.6** groups the contrast and threshold parameters and **Table 9.7** presents the snake parameters for each bone structure.

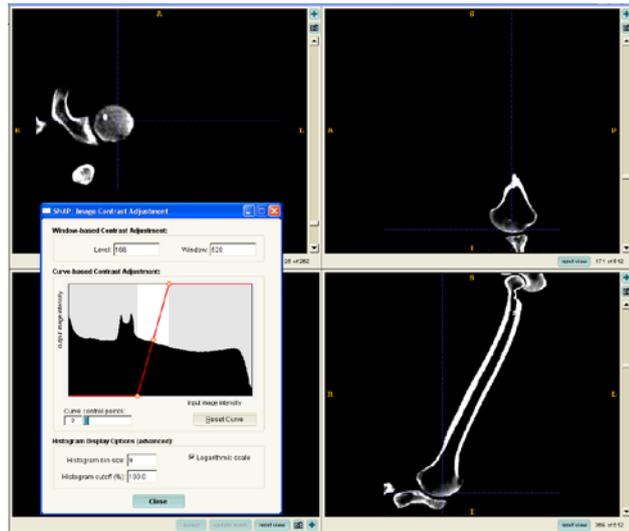
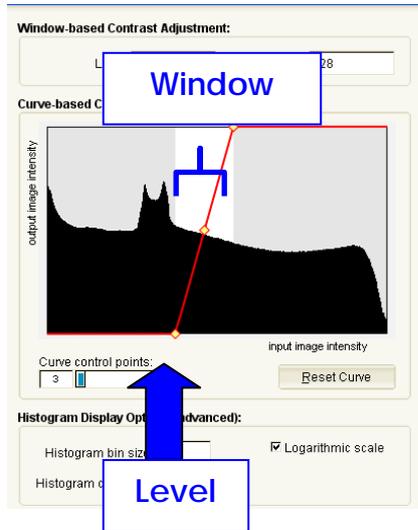


Figure 9.5– Contrast adjustment of the humerus sub-volume. The piece-wise linear function is juxtaposed with the histogram. The contrast adjustment parameters (CT number level and window) are presented in the amplified histogram.

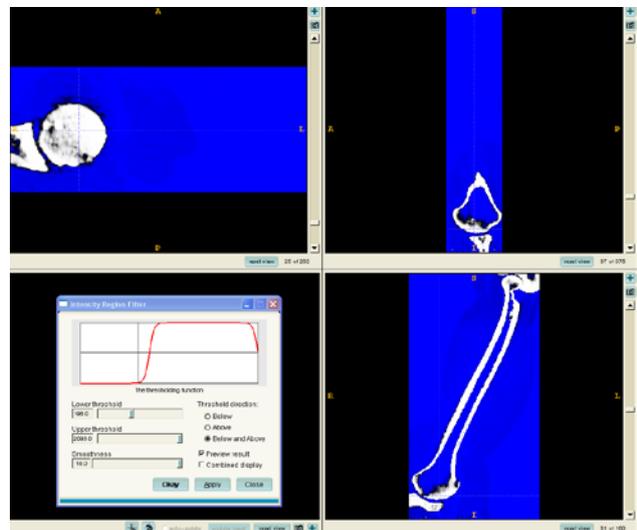
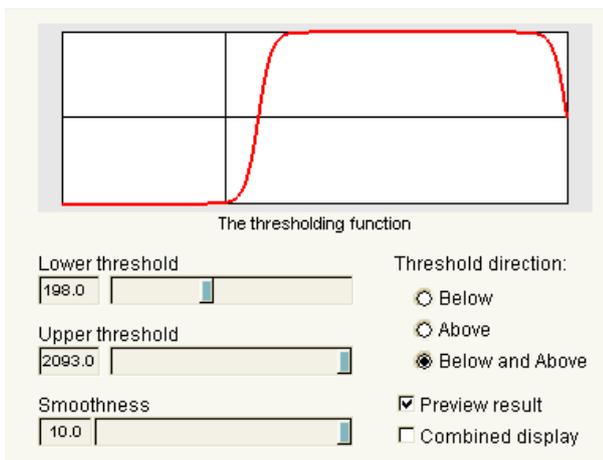


Figure 9.6– Intensity region filter or global threshold of the humerus sub-volume. The graph on the left represents the thresholding function.

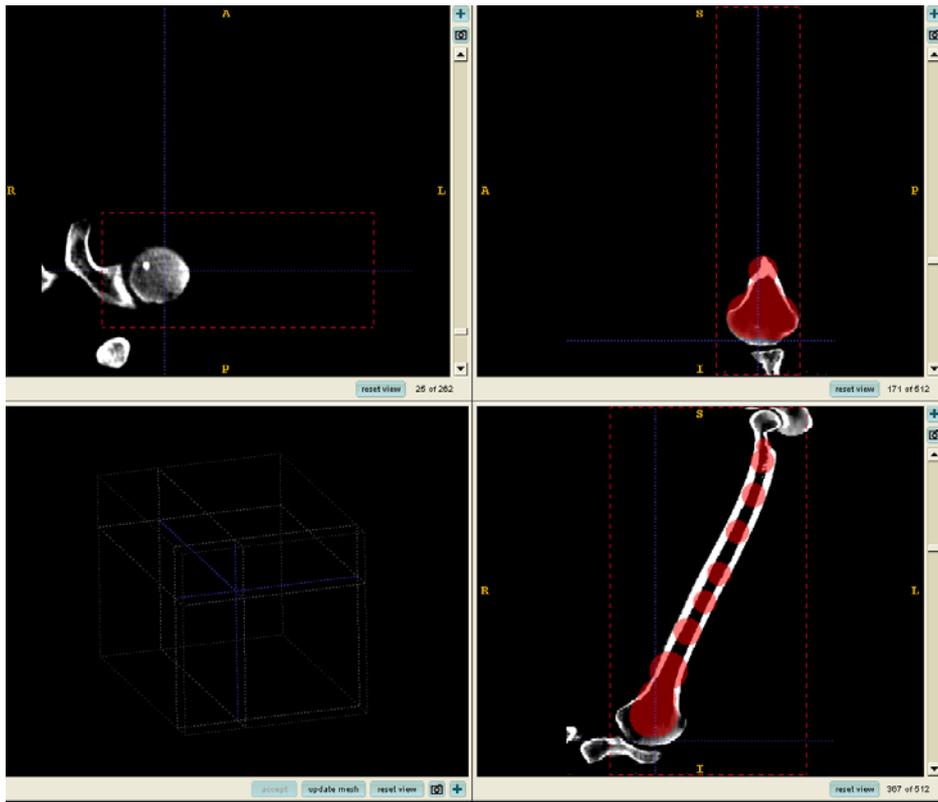


Figure 9.7– Selecting the region of interest of the humerus sub-volume after enhancing image contrast. The ROI is represented as the red dashed box surrounding the humerus.

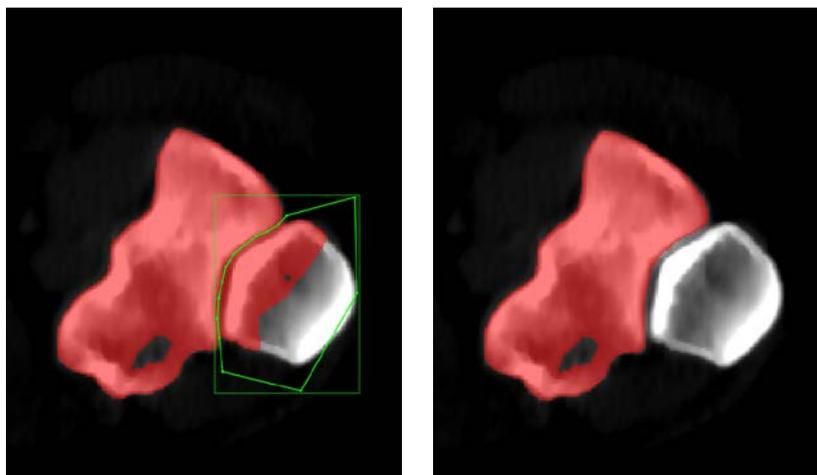


Figure 9.8– Over-segmentation is corrected manually with a polyline curve (green).

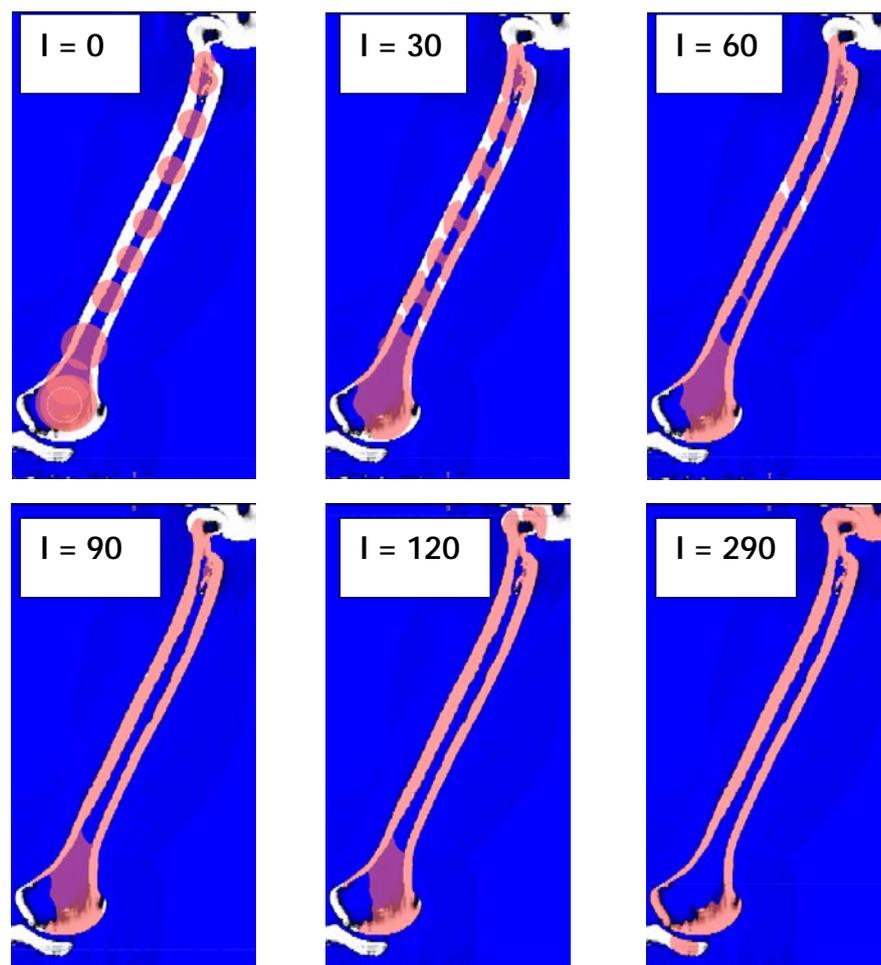


Figure 9.9– Iterations of the active contour method (iterations = 0, 30, 60, 90, 120, 290).

Bone Structure	Contrast Adjustment		Intensity Region Filter		
	Level	Window	Lower Thres.	Upper Thres.	Smoothness
Humerus	166	528	198.0	2093.0	10.0
Radius	113	763	218.3	2264.0	10.0
Ulna	113	763	225.1	2163.0	10.0
Femur	62	972	206.5	2115.0	10.0
Patella	109	193	217.4	1701.0	10.0
Tibia	108	297	170	2156.0	10.0
Fibula	118	316	200.2	2156.0	10.0

Table 9.6 – Contrast and threshold parameters.

Level – lower level of the histogram stretching; **Window** – size of the interval between the lower and upper levels; **Smoothness** – steepness of the threshold function.

Bone Structure	Humeru s	Radius	Ulna	Femur	Patella	Tibia	Fibula
Advection Speed Exponent	0	0	0	0	0	0	0
Advection Weight	0	0	0	0	0	0	0
Advection Weight	1	1	1	1	1	1	1
Clamp	1	1	1	1	1	1	1
Curvature Speed Exponent	-1	-1	-1	-1	-1	-1	-1
Curvature Weight	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Ground	5	5	5	5	5	5	5
Laplacian Speed Exponent	0	0	0	0	0	0	0
Laplacian Weigh	0	0	0	0	0	0	0
Propagation Weight	1	1	1	3	3	1	1
Snake Type	Region Competi- tion	Region Competi- tion	Region Com- petition	Region Competi- tion	Region Competi- tion	Region Competi- tion	Region Competi- tion
Solver Algo- rithm	Parallel Sparce Field						
TimeStepFacto r	1	1	1	1	1	1	1

Table 9.7– Advanced snake parameters [3,10].

Surface Generation and Adjustments

The segmentation volume files must be converted to a MATLAB file type in order to perform the following operations. Jimmy Shen developed a set of tools to read and load Analyze files (*.hdr and *.img) and are available at the open exchange MATLAB Central [11]

The segmentation process selected only what is bone tissue and therefore bone tissue marrow is not segmented. This introduces a hole within the segmented volume. To fill in the bone marrow void, the MATLAB function `imfill(.)` is used. `imfill(.)` is a morphological operator used to reconstruct binary images and is applied to each axial slice of the binary volume (**Figure 9.10**).

MATLAB performs the surface mesh generation with the `isosurface(.)` command. The output is a native variable that is converted to a *.obj mesh file with the tools developed by Anders Sandberg, at the Royal Institute of Technology in Stockholm [12].

To enter the ParaView environment, the surface mesh must be converted from *.obj to *.stl with the Blender application.

In ParaView, the surface meshes are adjusted with the smooth, decimate, and quadric clustering operations. **Figure 9.11** shows the effect of the operations listed in **Table 7.1** applied to the tibia.

The surface mesh that results from the first smoothing operation is suitable for 3D visualization. After the smoothing the result of decimation or quadric clustering, the surface mesh can be inputted to a volume mesh generator.

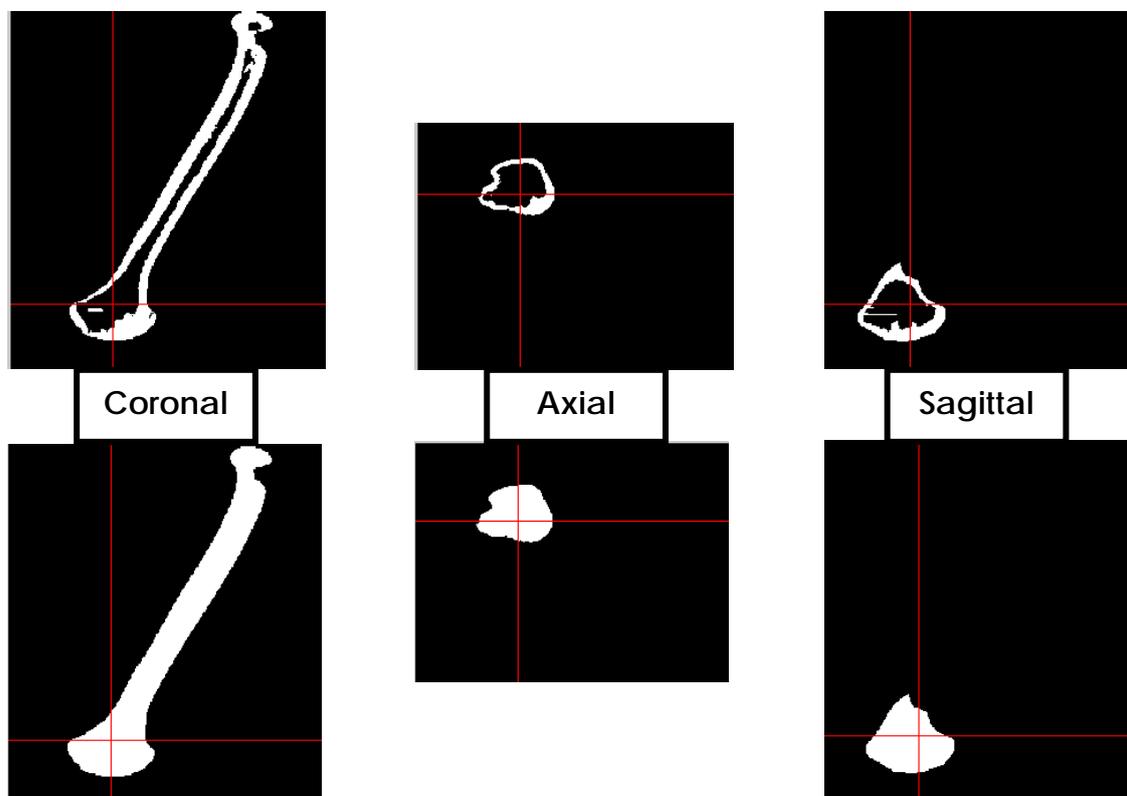


Figure 9.10 – Application of morphological operations to fill in the holes in the cross section images of the humerus.

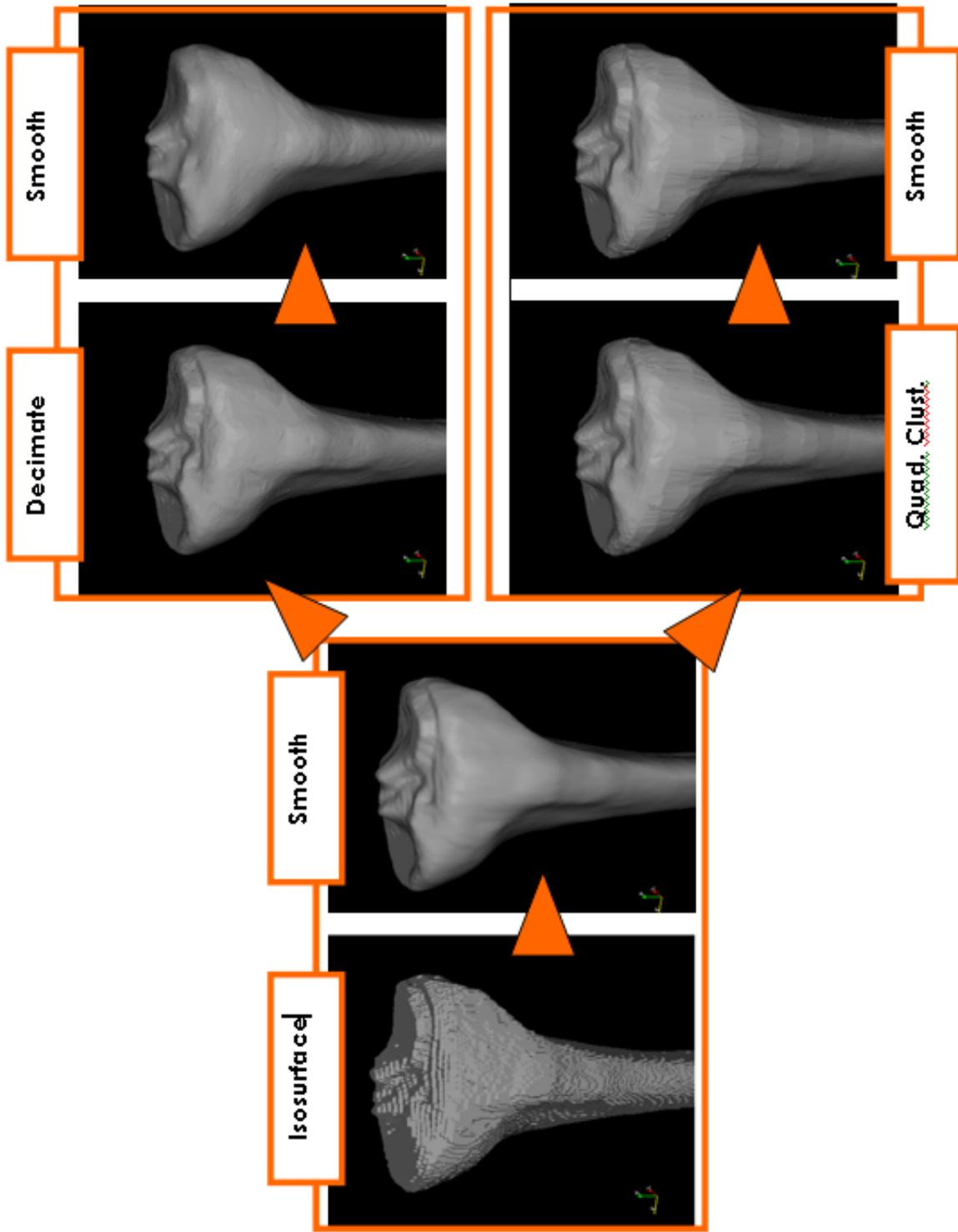


Figure 9.11 – Two types of surface adjustments performed upon the tibia.

The adjustment and simplification parameters are listed in **Table 9.6** and **Table 9.77**. The computational implementation of the algorithms described in chapter 7 involves other parameters. It is advisable to consult the ParaView user manual [13] and the ParaView mailing list for more details [5].

The results of mesh simplification are grouped in **Table 9.8**. Quadric clustering proves to be more efficient than decimation.

Bone Structure	Smooth		Decimate		Smooth	
	Num. Iterations	Convergence	Reduction Target	Feature Angle	Num. Iterations	Convergence
Humerus	500	0	0.75	15	500	0
Radius	500	0	0.75	15	500	0
Ulna	500	0	0.90	15	500	0
Patella	500	0	0.75	15	500	0
Tibia	500	0	0.75	15	500	0
Fibula	500	0	0.75	15	100	0

Table 9.6 – Parameters of the Smooth → Decimate → Smooth pipeline.

Bone Structure	Smooth		Quadric Clustering			Smooth	
	Num. Iterations	Convergence	Divisions			Num. Iterations	Convergence
Humerus	500	0	250	250	250	500	0
Radius	500	0	250	250	250	500	0
Ulna	500	0	250	250	250	500	0
Patella	500	0	50	50	50	500	0
Tibia	500	0	250	250	250	500	0
Fibula	-	-	-	-	-	-	-

Table 9.7 – Parameters of the Smooth → Quadric Clustering → Smooth pipeline.

Bone Structure	Isosurface			Decimate			Quadric Clustering		
	Num. Elements	Num. Nodes	Memory (MB)	Num. Elements	Num. Nodes	Memory (MB)	Num. Elements	Num. Nodes	Memory (MB)
Humerus	172848	86420	3.804	17284	8638	0.519	15830	7842	0.358
Radius	70816	35408	1.559	17704	8852	0.532	5336	2637	0.164
Ulna	77872	38937	1.714	19468	9735	0.585	7218	3567	0.175
Patella	13202	6603	0.292	3300	1652	0.1	4034	2008	0.091
Tibia	132854	66428	2.924	33212	16607	0.998	16376	8056	0.36
Fibula	49692	24848	1.095	12422	6213	0.374	-	-	-

Table 9.8 – Quantitative comparison of the mesh simplification with the isosurface, regarding number of elements, nodes and memory usage.

Volume Generation and Mesh Quality

Blender is once more applied for file conversion. A *.ply surface mesh is outputted from ParaView, and Blender converts it to a *.stl file, suitable for GMSH.

In GMSH it is necessary to define first the geometry, which is done by loading the surface mesh. After selecting the type of 3D mesh generator and other specific parameters like characteristic length factor and the random perturbation factor, volume elements will be generated filling in the domain delimited by the surface boundary. **Figure 9.12** shows the volume mesh of the patella by using clipping planes.

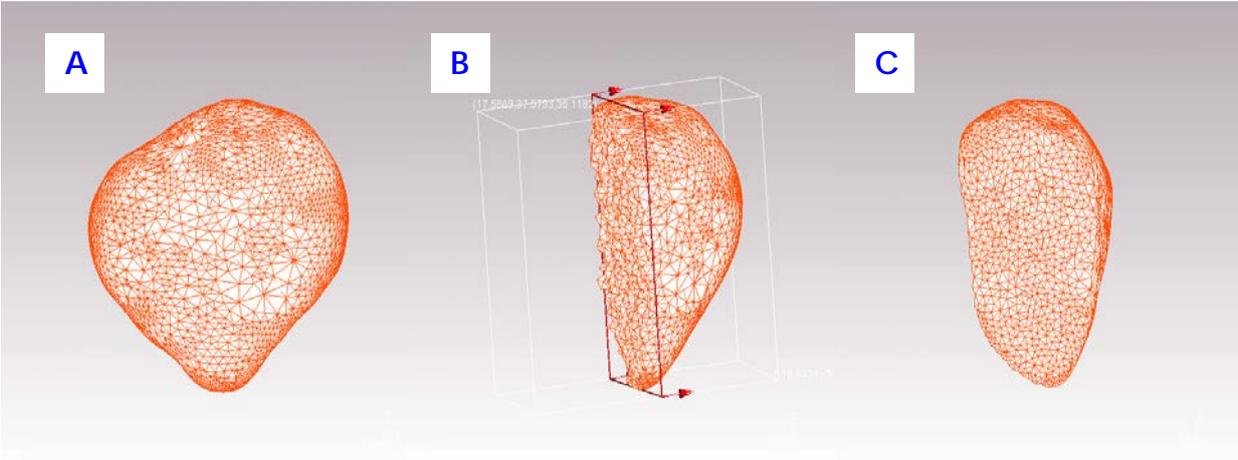


Figure 9.12 – Volume mesh of the patella. (A) Surface boundary; (B & C) Volume mesh clipped with the plane of equation $x - 10y = 0$.

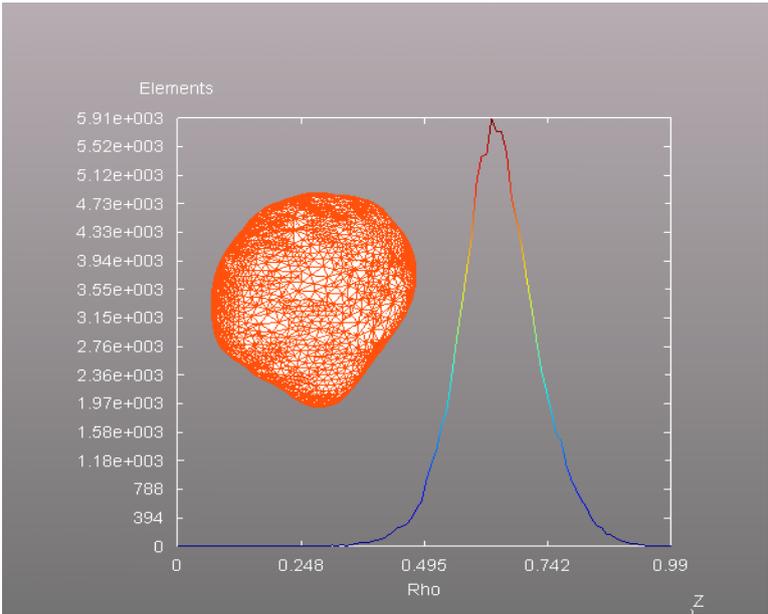


Figure 9.13 – Quality measure ρ of the patella.

The mesh quality can be analyzed by plotting the quality measure versus the number of elements with that measure. **Figure 9.13** shows the plot of the quality measure ρ .

Computational Simulation

The GMSH software package contains functions that convert the data contained in a *.msh file into a MATLAB[®] variable. This file contains the nodal coordinates and the node list of each finite element.

To load the volume mesh into ABAQUS[®] a script was written in MATLAB[®] that creates a *.inp file. This ABAQUS[®] file contains the commands for the finite element solver to perform a hypothetical static analysis. The analysis has no physical relevance and was done only to test the pipeline. The following figure shows the results of the Von Mises stress distribution in the ABAQUS[®] environment.

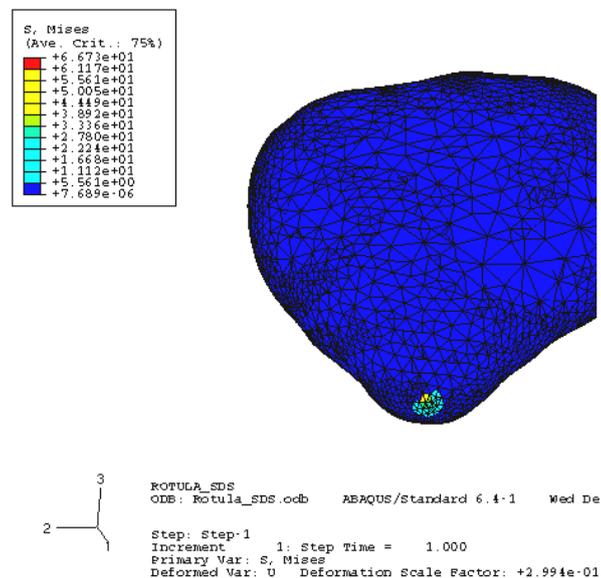


Figure 9.14 – Example of a final result of the software pipeline.

Time and Modeling

The total time required to complete the pipeline, from medical images to a numerically consistent volume mesh is of the order of a few hours. Modeling time depends on the quantity of data to be processed. The patella is obviously less time consuming than the humerus. Since the structures to be modeled are mostly long bones this implies large amounts of data and a few hours to process all different types of intermediate data.

Images with a higher SNR, or noise free, and more automation of the digital image processing stage, namely the segmentation module, would drastically reduce the total processing time.

10. 3D MODEL VS. REAL BONE PICTORIAL COMPARISON

The accuracy of the model will be evaluated by comparing the virtual 3D models with photographs of dead bones. This analysis is entirely qualitative and prone to subjective interpretation. **Figure 10.1** and **Figure 10.2** present osteologic details of the humerus and tibia models. The models were obtained with the SMOOTH and SMOOTH → QUADRIC CLUSTERING → SMOOTH strategies.

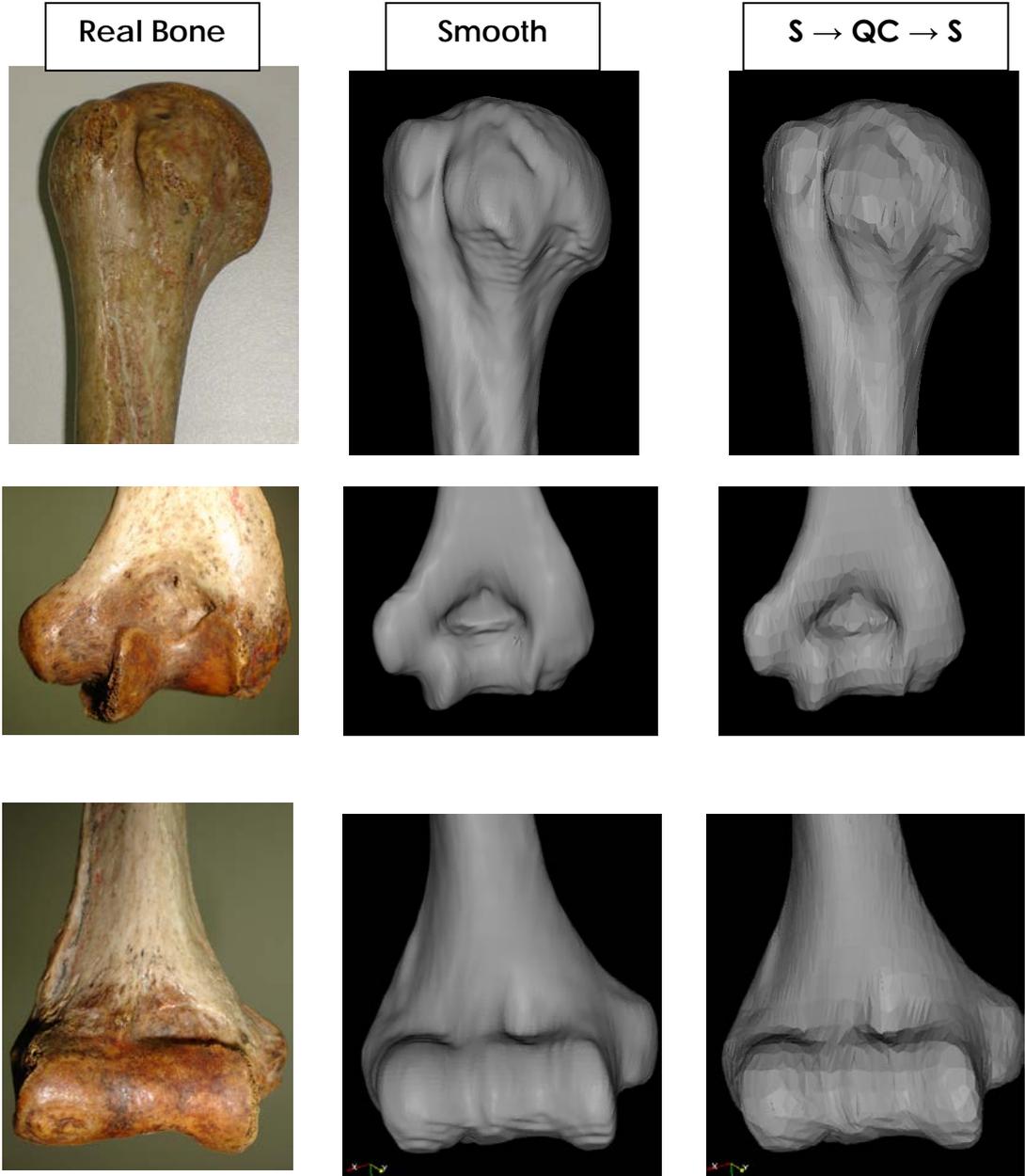


Figure 10.1 – Pictorial comparison of the humerus epiphysis.

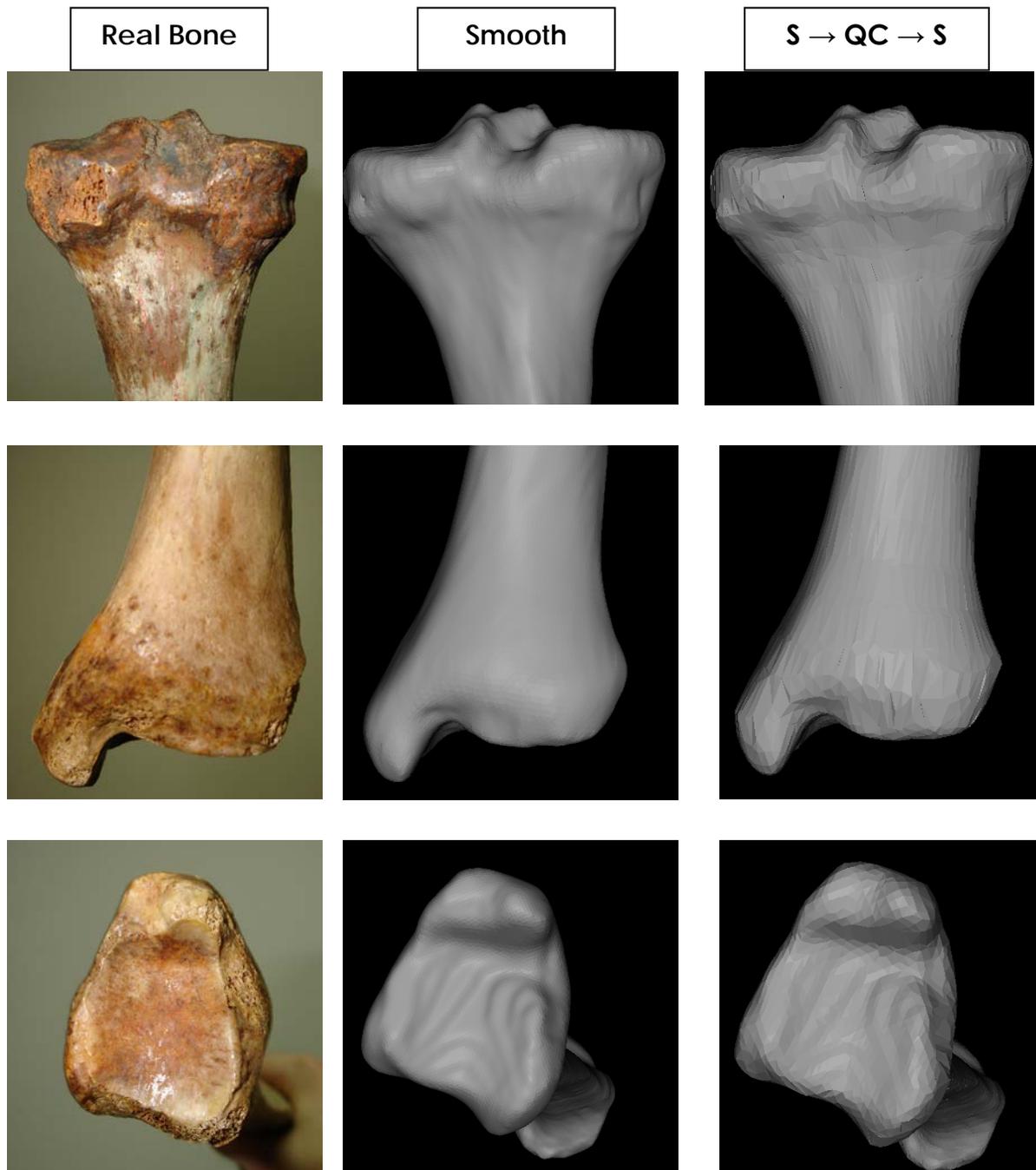


Figure 10.2 – Pictorial comparison of the tibia epiphysis.

The qualitative evaluation made possible by the comparisons above is believed to be *extremely positive*. A detailed evaluation of the complete models by experimented anatomists or orthopedists will be sought in the near future.

11. DISCUSSION

The geometric modeling of human structures, i.e., the production of an accurate 3D model from a 3D medical data set, can be solved by taking several approaches. Each approach has different DIP or AMG methods and techniques which characterize the production pipeline [1-3]. Robustness, accuracy, efficiency and effectiveness are the main characteristics that describe a pipeline and directly influence the solution's features. However, different pipelines provide different solutions. Despite possible dissimilarities, all solutions are approximations of the anatomical structure to be modeled. [4,5]

Therefore, the proposed pipeline is one among many different possible approaches to solve the problem of CT bone modeling. It may not be the most robust and efficient approach but the results are truthful and quite accurate (see **Figure 10.1** and **Figure 10.2**). The eminences and depressions of the bones are well defined and the resemblances are evident, even for the surface meshes intended for computational simulations.

Errors can occur at any stage of the pipeline, from image acquisition to volume mesh generation, degrading the fidelity of the meshes, and limiting the applicability at which they were aimed. Due to the sequential nature of the pipeline, this chapter will be divided into sections that correspond to the different stages of the pipeline. This chapter will end with some considerations on software.

Image Acquisition

Image acquisition was carried out with regular clinical standards. However, for modeling purposes, the only requirements made to the technical staff was that the images should have high resolution (preferably isotropic voxels) and optimum contrast. As a result of the high resolution request, the obtained data volumes had a large number of thin slices and the voxel dimensions, although not isotropic, were minute.

The bone/soft tissue contrast was high but the images displayed many grain-like artifacts. Because the X-ray beam was tightly collimated, the number of photons concentrated on the area of interest was reduced. By decreasing the number of photons that reach the detectors, the signal projections were noisy, due to fewer counts. To surpass this problem, the intensity of the X-ray beam should have been increased. This alternative involves higher radiation dose toward the subjects, therefore it was not carried out. By summing contiguous slices this type of noise can be attenuated, but resolution along the longitudinal axis becomes coarser. So, since the very beginning of the pipeline there is a compromise between image resolution and image quality.

Note that image resolution is the determinant factor of the accuracy of the reconstructed surface.

Image Enhancement and Restoration

The proposed combination of filters has proven that filtering in the sagittal plane is more effective than in the axial plane (**Figure 9.4**). The initial analysis of the data volume, as de-

scribed in chapter 9 (**Figure 9.2** and **Figure 9.3**), was determinant for the filtering procedure to be sagittal. By observing the axial, sagittal and coronal planes, a major difference was noticeable: the artifacts had different structures in each reformatted plane. For instance, in the axial plane the artifacts are predominantly streak-like, while in the sagittal and coronal planes the artifacts have a speck-like structure. Between these two planes, the sagittal plane exhibits less blotchy artifacts than those within the coronal plane. Note that, when filtering in the sagittal plane, the mask is spatially anisotropic, while, in the axial plane, the mask is isotropic.

In order to achieve the proposed combination of filters, various filter arrangements and associated parameter tunings have been made. This reveals that image filtering is an experimental procedure of trial and error, which is extremely necessary for knowing how a filter operates and to gain the numerical sensibility required for parameter tuning.

Basically, the reason why all the sub-volumes were filtered with anisotropic diffusion was its powerful capability for attenuating the majority of the noise, and to restore the homogeneity to tissue regions. Since the anisotropic diffusion smoothes the interior of the regions and preserves edges, this filter is not suitable to eliminate speck-like artifacts, which are dominant in the girdle regions. In contrast, the median and Wiener filters seem to be tailor-made for this type of artifact. The usability of the Gaussian filter is restricted to smoothen the image, which is useful for attenuating the noise that still persists after the anisotropic diffusion.

The only region to which the present filtering pipeline was not applicable was the pelvic girdle. There, due to photon starvation, the streak artifacts are present in a large amount and have a destructive influence. **Figure 11.1** shows how the artifacts can vanish away the thin cortical coating of the femur head. With this type of image information it is extremely difficult to restore the original characteristics. Studies developed by Kalender et al. [6,7] demonstrate that by filtering in the projection domain with multidimensional adaptive filters the streak artifacts can be remarkably removed. Such filter, together with the projection data, would have deeply improved the image quality for both pelvic and shoulder girdles.

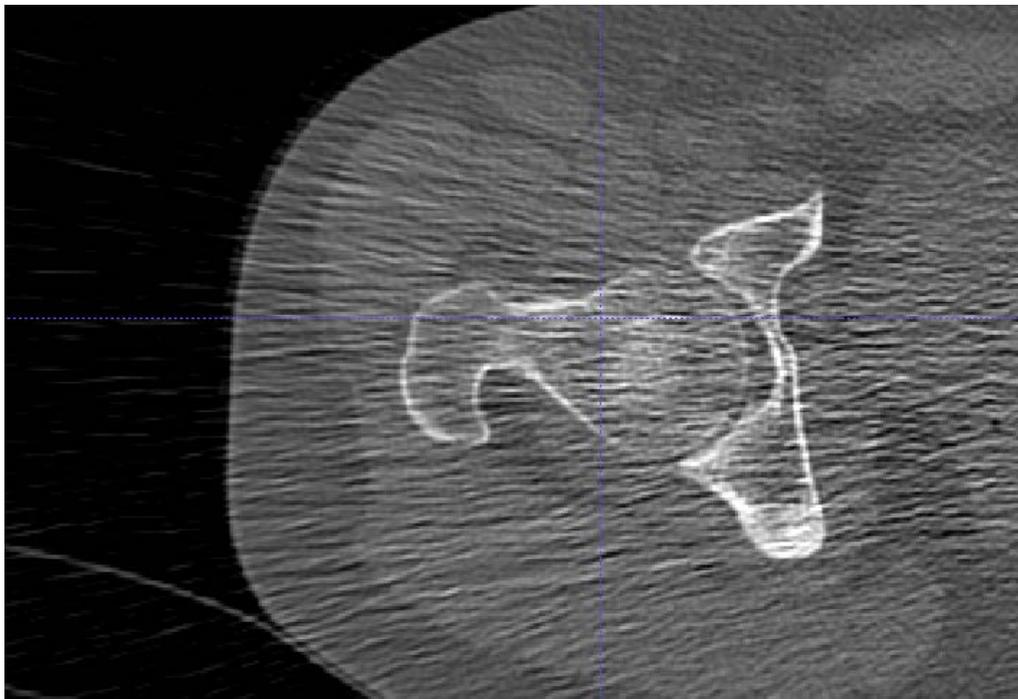


Figure 11.1 – Axial image of the pelvic girdle demonstrating how destructive streak-like artifacts can be.

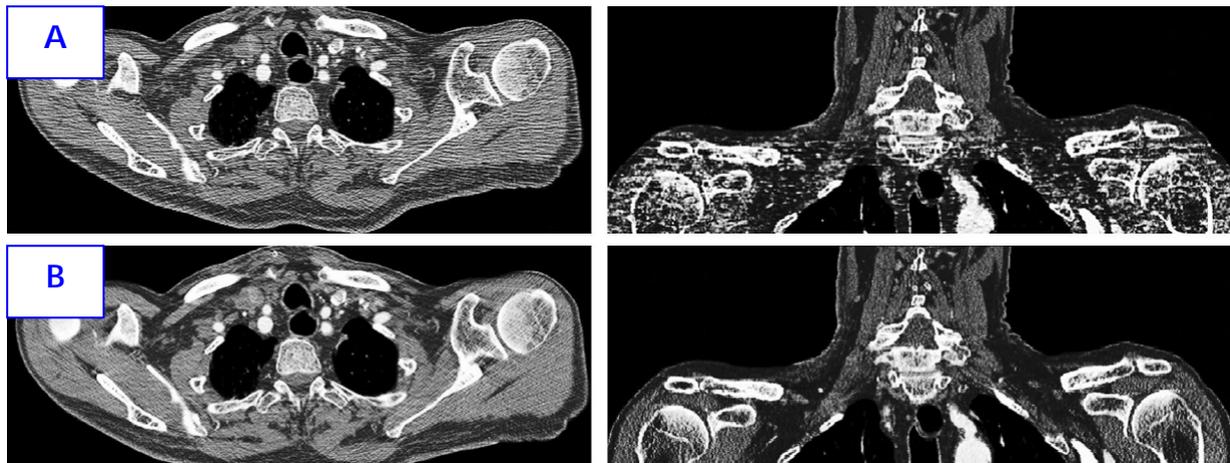


Figure 11.2 – The result of the multidimensional adaptive filter upon a shoulder girdle that is corrupted with streak-like artifacts. (A) – Axial and coronal view before filtering. (B) – Axial and coronal view after filtering. [7].

Image Segmentation

Even after filtering the images, the most prominent noise and artifacts, such as partial volume effect and streaks at the girdle regions, still persist. Thus, tissue boundaries can be ambiguous, which will induce the segmentation procedure to determine voxels that are outside the bone region (**Figure 11.3**). The opposite can also occur, as several bony regions may not even be classified (**Figure 11.4**). The former situation is called over-segmentation and the latter sub-segmentation. The main reasons for sub-segmentation to occur are low image contrast and the physical properties of the 3D snake.

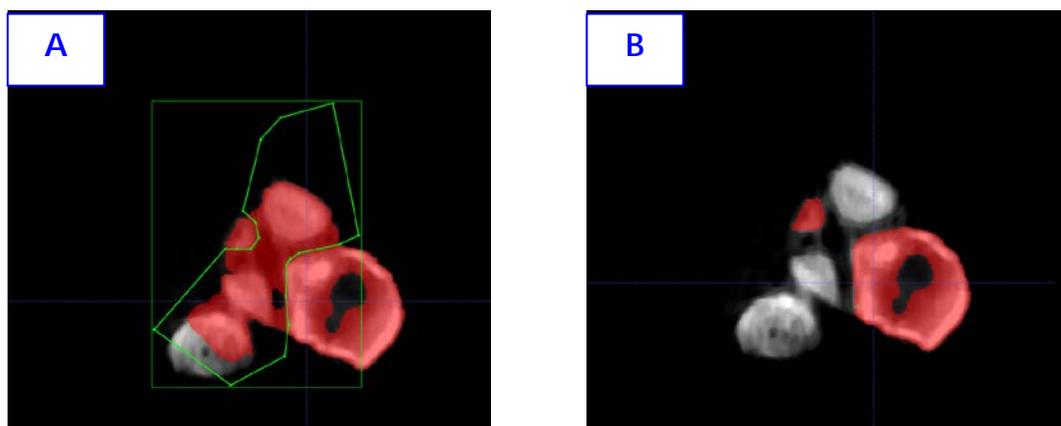


Figure 11.3 – Example of over-segmentation (where the humerus and ulna connect at the elbow articulation). (A) – Manual segmentation. (B) – Segmentation fault corrected.

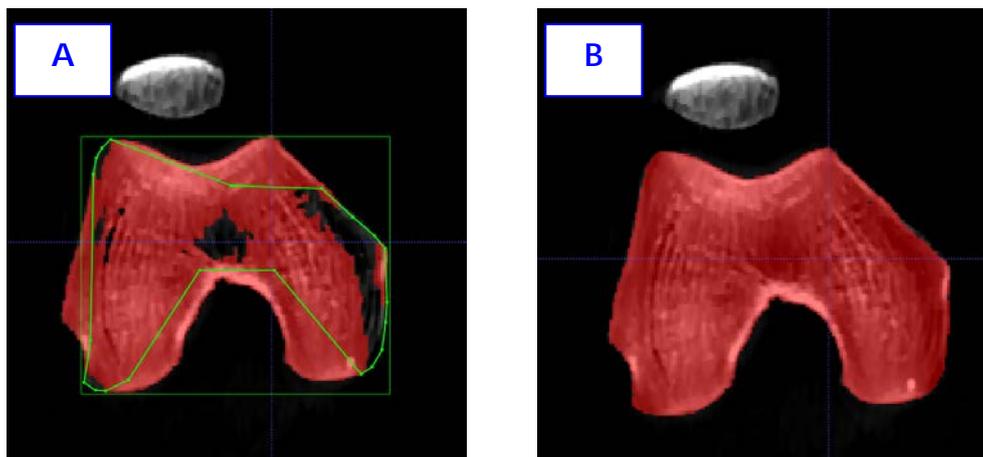


Figure 11.4 – Example of sub-segmentation (femur lower extremity). (A) – Manual segmentation. (B) – Segmentation fault corrected.

Segmentation faults such as those illustrated above were corrected manually. This can be time consuming but it is an accurate method to properly segment the bone tissue. The regions that require more user interaction are the epiphyses. At these regions the cortical bone is very thin and therefore it does not correspond to a highly contrasted intensity transition in the CT image, as it happens with the bone at the shaft.

Surface Generation and Adjustments

The stage of surface mesh adjustment induces the most important errors of the pipeline: smoothing and mesh simplification by decimation or quadric clustering.

Although the smoothing operation conserves the structure's volume and topology, the geometry is altered. Since the surface mesh generator inserts a node at half a voxel size, the error incurred by filtering the node geometry is of the order of a voxel. It may not seem much but, in association with an inappropriate λ (Equation 7.1) the surface mesh may cease to be representative due to excessive low-pass filtering.

Mesh simplification, which is extremely required for further computational simulations, may even have a harsher effect than smoothing, if done inappropriately. As the name implies, simplifying a mesh consists on eliminating nodes and elements. So, if data is deleted, then the resolution of the surface curvature will be at risk. Between the two operators, quadric clustering and decimation, the former is more effective than the latter, both in memory space and in mesh quality. By comparing two surfaces of the same anatomical structure the difference becomes evident (**Figure 11.5**).

In order to obtain a model that represents the anatomy there must be a compromise between quality and resolution. To improve the mesh quality it is necessary to smooth the initial jagged surface. The resulting mesh can be used for medical visualization. Computational simulation usually demands a coarser mesh than the one used for visualization. Simplifying a

mesh carries loss of resolution. A final smoothing eases the mesh artifacts generated by node elimination.

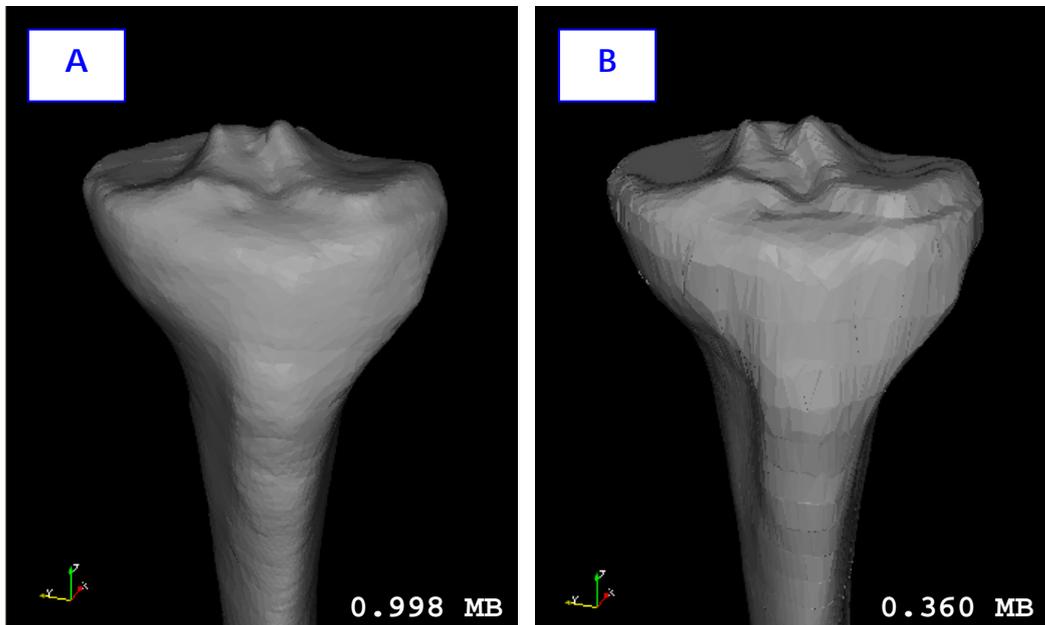


Figure 11.5 – Surface meshes obtained with (A) decimation and (B) quadric clustering. The corresponding memory usage is also indicated.

The pipeline could enhance its efficiency if a generalized triangular surface mesh optimizer were incorporated. With such optimizer the quality of the surface mesh would improve while maintaining its essential characteristics. The optimization is achieved by repositioning the mesh nodes according to local parametric spaces of the mesh facets without distorting the discrete surface too much, thus, improving the mesh quality [8,9].

The surface mesh quality established after the smoothing operation is determinate for further computer simulations, since the volume mesh generator preserves the surface boundary. No holes or degenerated surface elements must appear in the surface mesh, otherwise the volume mesh generator will not proceed with any operation due to input errors.

Volume Generation and Mesh Quality

As the volume mesh generator conserves the boundary surface mesh, the only concern regarding the volume mesh pipeline is the resulting mesh quality. For the example presented (**Figure 9.13**), the graph of the quality measure has a bell-like form indicating that there is a great number of elements that share a common form and very few sliver elements. This element diversity can be justified by the complexity of the anatomical surface. For example higher curvatures must be filled with smaller elements and therefore the mesh is more refined in these regions.

Software Considerations

The software arrangement is much decentralized. Each step of the pipeline requires different software. Even converting a file format to another implies using specific software. Integrating all these tools in a unique software would be more efficient but it is indeed a very demanding task.

The major difficulties in constructing a software pipeline are, firstly, discovering what type of software is available and, secondly, learning how to manage the software. This can take weeks searching Google[®] and some software have steep learning curves or are badly documented, which consumes even more time.

Besides the software listed in **Table 9.3** other software were learned and managed, such as OsiriX for Macintosh [10], the free demo of 3D Doctor[®] [11] and the (widely used in the academic community) 3D Slicer [12]. All of them provide excellent data visualization capabilities, as well as image and mesh processing capabilities that are used in clinical practice [13].

One of the greatest accomplishments of the proposed pipeline is that the software used is mostly freeware and open source. Besides MATLAB[®] and ABAQUS[®], that are common in the academic community, all the other software is entirely free.

Besides the aforementioned signal and processing errors, technical faults can also occur and they seem to strictly follow Murphy's laws. With the advantage of being at no cost, non-commercial software usually is full of bugs and errors that need to be corrected. Several times ITK-SnAP, ParaView, and GMSH presented fatal errors. As a result, all the work being done at the time of each software crash was deleted and impossible to recover.

For example, ParaView occasionally presents graphical breakdowns, that is, the software loads the data, processes the data but the data cannot be visualized. GMSH sometimes does not respond to certain commands such as performing the volume mesh generation or clipping the volume mesh.

But technical faults are not a unique characteristic of non-commercial software. Importing the volume mesh to a finite element solver, such as ABAQUS[®], is the final step of the modeling pipeline. To perform a simple computational simulation, such as a static analysis with only one load applied and all nodes constrained, the ABAQUS[®] input file reunites all the information and commands needed: mesh geometry (nodes and elements) element types, material properties, static loads and boundary conditions. For some unknown reason (Very large input file? Too low graphical memory? Unknown software bug?), the ABAQUS[®] software seems not to display all the element meshes and associated values. Only a subset of the mesh is shown in the graphical interface, both with ABAQUS[®] CAE or ABAQUS[®] Viewer.

12. CONCLUSIONS

The presented software pipeline combines various image and mesh processing techniques, and is capable of transforming a stack of CT medical images into representative 3D models of human anatomical structures. The pipeline was designed to model the geometries of bone structures belonging to the body extremities.

The models obtained are accurate and suitable for medical visualization and computer simulations. These models have a wide span of applications, including implant design, pre-operative planning, virtual surgery and biophysical simulations. Thus, by integrating medical imaging to well known engineering technologies, such as, computer aided design, finite element analysis and rapid prototyping, clinical diagnosis and therapeutics will be greatly benefited.

The pipeline incorporates several automatic procedures but user interaction plays an important role to achieve the final results. Every stage has a set of parameters that needs to be chosen. Currently, segmentation with active contours is still semi-automatic and manual correction is required. Therefore, the whole process, from image acquisition to mesh generation, takes several hours. Automation, principally on image segmentation, together with more efficient mesh generators, will reduce the total modeling time to the order of minutes.

The robustness of the pipeline could not be evaluated because only two 3D images, corresponding to both body extremities, were considered. A greater set of volume data would be required to test how robust the pipeline really is.

Further work can be developed in order to optimize the results. If the projection data is available, then the multidimensional adaptive filter could be implemented. This would improve the image restoration of artifacts that are difficult to attenuate with spatial image filters. The volume mesh quality can also be optimized with a method designated as Delaunay refinement. Such optimizations can increase the quality of the final results but also the overall modeling time. Triangular surface mesh optimizer will certainly be a future acquirement to the pipeline, which will improve mesh quality.

One major limitation of the pipeline are the errors and bugs that persist in the software used. These problems are very annoying and depend greatly on the software developers.

REFERENCES

INTRODUCTION

- [1] – Sullivan, Jr., J.M., Wu, Z., and Kulkarni, A. – “3D Volume Mesh Generation of Human Organs Using Surface Geometries Created from the Visible Human Data Set”, *The Third Visible Human Project Conference*, 2000.
- [2] – Cebral, J.R., and Löhner, R. – “From Medical Images to CFD Meshes”, *Proceedings of the 8th International Meshing Roundtable*, pp. 321–331, 1999.
- [3] – Cebral, J.R., and Löhner, R. – “From Medical Images to CFD to Anatomically Accurate Finite Element Grids”, *Int. J. Numer. Meth. Engng.* **51**, pp. 985-1008, 2001.
- [4] – Du, J., Yang, X., and Du, Y. – “From Medical Images to Finite Grids System”, *Engineering in Medicine and Biology 27th Annual Conference*, 2005.
- [5] – P.B. Dinis, J.A.C. Martins, E.B. Pires – "Finite Element Simulation of the Forearm Flexion", Fourth International Symposium on Computer Methods in Biomechanics and Biomedical Engineering, Lisbon, Portugal, Setember 1999. Published in: *Computer Methods in Biomechanics & Biomedical Engineering – 3*, ed. by J. Middleton et al., Gordon and Breach Science Publishers, The Netherlands, pp. 295-300, 2001.
- [6] – P. B. Dinis, E. B. Pires, J.A.C. Martins, S. Domingues, N. Perdigão – “Surgical correction of scoliosis by the Resina-Alves method: numerical simulation of a clinical case”, *International Congress of Computational Bioengineering*, ed. by M. Doblaré, M. Cerrolaza e H. Rodrigues, Vol.2, pp. 471-476, Saragoza, Spain, Setember 2003.
- [7] – D. d'Aulignac, J.A.C. Martins, E.B. Pires, T. Mascarenhas, R.M. Natal Jorge – “A Shell Finite Element Model of the Pelvic Floor Muscles”, *Computer Methods in Biomechanics and Biomedical Engineering*, Vol. 8, pp. 339-347, 2005.
- [8] – J.A.C. Martins, M.P.M. Pato, E.B. Pires, R.M. Natal Jorge, M. Parente and T. Mascarenhas – “Finite element studies of the deformation of the pelvic floor”, *Annals of the New York Academy of Sciences* (to appear), 2007.
- [9] – Viceconti, M., Testi, D., Tadei, F., Martelli, S., Clapworthy, G.J., and Jan, S.S – “Biomechanics Modeling of the Musculoskeletal Apparatus: Status and Key Issues”, *Proceedings of the IEEE*, vol. **94**, No. 4, pp. 725-739, 2006
- [10] – Gonzalez, R.C., and Woods, R.E., *Digital Image Processing*, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2002.
- [11] – Gonzalez, R.C., Woods, R.E., and Eddins, S.L., *Digital Image Processing using MATLAB[®]*, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2004.

GEOMETRIC MODELING IN BIOMECHANICS OF TISSUES AND STRUCTURES

- [1] – Fung, Y.C., *Biomechanics: Mechanical Properties of Living Tissue*, Springer-Verlag, New York, 1993.
- [2] – Freivalds, A., *Biomechanics of the Upper Limbs – Mechanics, Modeling, and Musculoskeletal Injuries*, CRC Presss, Boca Raton, Florida, 2004.
- [3] – Pereira, C., Mateus, A., Antunes, F.V., and Gaspar, M.C. – “Customizing Hip Joint Implants Based on Rapid Prototyping Techniques”, *Inter. Conf. Comp. Bioeng.*, Lisbon, Portugal, 2005.

- [4] – Webb, A., *Introduction to Biomedical Imaging*, John Wiley & Sons, Hoboken, New Jersey, 2003.

OSTEOLOGY OF THE UPPER AND LOWER LIMBS

- [1] – Gray, H., *The Complete Gray's Anatomy*, 16th edition, Senate, East Mosely, 2003.
[2] – Tortora, G.J., and Grabowski, S.R., *Introduction to the Human Body – The Essentials of Anatomy and Physiology*, 5th, John Wiley & Sons, New York, 2001.
[3] – Pina, J.A.E., *Anatomia Humana da Locomoção*, 2^a Edição, Lidel, Lisboa, 1999.
[4] – Düring, M.v., Poggesi, M. and Bambi, S., *Encyclopedia Anatomica*, Tashen, Cologne, 2006.
[5] – Anatomy of the Human Body – Henry Gray.
<http://www.bartleby.com/107/>

FUNDAMENTALS OF DIGITAL IMAGE PROCESSING

- [1] – Gonzalez, R.C., and Woods, R.E., *Digital Image Processing*, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2002.
[2] – McAndrew, A., *Introduction to Digital Image Processing with MATLAB*, Thomson, Boston, Massachusetts, 2004.

IMAGE ENHANCEMENT AND RESTORATION

- [1] – Webb, A., *Introduction to Biomedical Imaging*, John Wiley & Sons, Hoboken, New Jersey, 2003.
[2] – Barrett, J.F., and Keat, N. – “Artifacts in CT: Recognition and Avoidance”, *RadioGraphics*, **24**, pp.1679-1691, 2004.
[3] – Keselbrener, L., Shimoni, Y., and Akselrod, S. – “Nonlinear filters applied on computerized axial tomography: Theory and phantom images”, *Med. Phys.*, 19 (4), pp.1057-1064, 1992.
[4] – Hsieh, J. – “Adaptive streak artifact reduction in computed tomography resulting from excessive x-ray photon noise”, *Med. Phys.*, 25 (11), pp.2139-2147, 1998.
[5] – Kachelriess, .M, Watzke, O., and Kalender, W.A. – “Generalized multi-dimensional adaptive filtering for conventional and spiral single-slice, multi-slice, and cone-beam CT”, *Med. Phys.*, 28 (4), pp.475-490, 2001.
[6] - Sennst, D.A., Kachelriess, .M, Leidecker, C., Schmidt, B., Watzke, O., and Kalender, W.A. – “An Extensible Software based Platform for Reconstruction and Evaluation of CT Images”, *RadioGraphics*, **24**, pp.-601-613, 2004.
[7] – Gonzalez, R.C., and Woods, R.E., *Digital Image Processing*, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2002.
[8] - McAndrew, A., *Introduction to Digital Image Processing with MATLAB*, Thomson, Boston, Massachusetts, 2004.
[9] – Zingale, R. and Zingale, A. – “Detection of MRI Brain Contour using Isotropic and Anisotropic Diffusion Filter: a Comparative Study”, *J. Neurosurg. Sci.*, vol. **42**, n^o2, pp.111-114, 1998
[10] – Westin, C.F., Warfield, S., Bhalerao, A., Mui, L., Richolt, J., and R. Kikinis – “Tensor Controlled Local Structure Enhancement of CT Images for Bone Segmentation”, Reprint from *Proc. of First Int. Conf. on Medical Image Computing and Computer-*

Assisted Intervention (MICCAI'98), p. 1205-1212, Lecture Notes in Computer Science, Eds. W.M. Wells, A. Colchester, and S. Delp, Springer Verlag, October, 1998.

- [11] – Perona, P. and Malik, J. – “Scale-Space and Edge Detection using Anisotropic Diffusion”, *IEEE Transactions and Pattern Analysis and Machine Intelligence*, **Vol. 12**. No. 7, 1990.

IMAGE SEGMENTATION

- [1] – Pham, D.L., Xu, C. and Prince, J.L. – “Current Methods in Medical Image Segmentation”, *Annu. Rev. Biomed. Eng.*, **2**, pp.315-337, 2000.
- [2] - Sullivan, Jr., J.M., Wu, Z., and Kulkarni, A. – “3D Volume Mesh Generation of Human Organs Using Surface Geometries Created from the Visible Human Data Set”, *The Third Visible Human Project Conference*, 2000.
- [3] – Cebral, J.R., and Löhner, R. – “From Medical Images to CFD Meshes”, *Proceedings of the 8th International Meshing Roundtable*, pp. 321—331, 1999.
- [4] – Cebral, J.R., and Löhner, R. – “From Medical Images to CFD to Anatomically Accurate Finite Element Grids”, *Int. J. Numer. Meth. Engng.* **51**, pp. 985-1008, 2001.
- [5] – Du, J., Yang, X., and Du, Y. – “From Medical Images to Finite Grids System”, *Engineering in Medicine and Biology 27th Annual Conference*, 2005.
- [6] – McInerney, T. and Terzopoulos, D. – “Deformable Models in Medical Image Analysis: A Survey”, To appear in *Medical Image Analysis*, **1(2)**, 1996.
- [7] - Yushkevich, P.A, Piven, J., Hazlett, H.C., Smith, R.G., Ho, S., Gee, J.C., and Gerig, G.- “User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability”. *Neuroimage*, Article In Press, 2006.

SURFACE MESH GENERATION AND ADJUSTMENTS

- [1] – Viceconti, M. and Taddei, F. – “Automatic Generation of Finite Element Meshes form Computed Tomography Data”, *Critical Reviews in Biomedical Engineering*, **31(1)**, pp. 27-72 (2003).
- [2] – Sullivan, Jr., J.M., Wu, Z., and Kulkarni, A. – “3D Volume Mesh Generation of Human Organs Using Surface Geometries Created from the Visible Human Data Set”, *The Third Visible Human Project Conference*, 2000.
- [3] – The Marching Cubes.
<http://www.essi.fr/~lingrand/MarchingCubes/algo.html>
- [4] – Schroeder, W. J., Zarge, J. A., and Lorensen, W. E. – “Decimation of Triangle Meshes. “, *Computer Graphics (SIGGRAPH92 Proceedings)*, pp. 65–70, 1992
- [5] – Lindstrom. P. – “Out-of-Core Simplification of Large Polygonal Models”, In *Processdings of ACM SIGGRAPH 2000*

VOLUME MESH GENERATION

- [1] – Viceconti, M. and Taddei, F. – “Automatic Generation of Finite Element Meshes form Computed Tomography Data”, *Critical Reviews in Biomedical Engineering*, **31(1)**, pp. 27-72 (2003).
- [2] – GMSH.
<http://www.geuz.org/gmsh/doc/texinfo/gmsh.html>

BONE MODELING PIPELINE FOR THE UPPER AND LOWER LIMBS

- [1] – Philips[®] Medical Systems, Computed Tomography Mx8000 Dual.
<http://www.medical.philips.com/main/products/ct/products/brilliance/dual/>
- [2] – MATLAB[®] - The Language of Technical Computing.
<http://www.mathworks.com/products/matlab/>
- [3] – ITK-SnAP.
<http://www.itksnap.org/>
- [4] – Blender.
<http://www.blender.org/cms/Home.2.0.html>
- [5] – ParaView.
<http://www.ParaView.org/HTML/Index.html>
- [6] – GMSH.
<http://www.geuz.org/gmsh/>
- [7] – ABAQUS[®].
<http://www.hks.com/>
- [8] – Hanselman, D. and Littlefield, B., *Mastering MATLAB 7*, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2002.
- [9] – MATLAB and Octave Functions for Computer Vision and Image Processing.
<http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/Spatial/anisodiff.m>
- [10] – Yushkevich, P.A, Piven, J., Hazlett, H.C., Smith, R.G., Ho, S., Gee, J.C., and Gerig, G.-“User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability”. *Neuroimage*, Article In Press, 2006.
- [11] – Tools for NIfTI (ANALYZE) MR image.
<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=8797&objectType=file>
- [12] – Export meshes and isosurfaces from Matlab to the Alias/Wavefront OBJ format.
<http://www.nada.kth.se/~asa/Ray/matlabobj.html>
- [13] – ParaView User’s Guide (Version 1.6).
www.paraview.org/files/v1.6/ParaViewUsersGuide.PDF

DISCUSSION

- [1] – Sullivan, Jr., J.M., Wu, Z., and Kulkarni, A. – “3D Volume Mesh Generation of Human Oegans Using Surface Geometries Created from the Visible Human Data Set”, *The Third Visible Human Project Conference*, 2000.
- [2] – Du, J., Yang, X., and Du, Y. – “From Medical Images to Finite Grids System”, *Engineering in Medicine and Biology 27th Annual Conference*, 2005.
- [3] – Viceconti, M. and Taddei, F. – “Automatic Generation of Finite Element Meshes form Computed Tomography Data”, *Critical Reviews in Biomedical Engineering*, **31(1)**, pp. 27-72 (2003).
- [4] – Cebral, J.R., and Löhner, R. – “From Medical Images to CFD Meshes”, *Proceedings of the 8th International Meshing Roundtable*, pp. 321—331, 1999.
- [5] – Cebral, J.R., and Löhner, R. – “From Medical Images to CFD to Anatomically Accurate Finite Element Grids”, *Int. J. Numer. Meth. Engng.* **51**, pp. 985-1008, 2001.
- [6] – Kachelriess, .M, Watzke, O., and Kalender, W.A. – “Generalized multi-dimensional adaptive filtering for conventional and spiral single-slice, multi-slice, and cone-beam CT”, *Med.Phys.* 28 (**4**), April, 2001.

- [7] – Sennst, D.A., Kachelriess, .M, Leidecker, C., Schmidt, B., Watzke, O., and Kalender, W.A. – “An Extensible Software based Platform for Reconstruction and Evaluation of CT Images”, *RadioGraphics*, **24**, pp.-601-613, 2004.
- [8] – The Sylphis 3D Bible - Generalized Triangle Mesh Optimizer.
<http://devnet.sylphis3d.com/node/21>
- [9] – Garimella, R.V, Shashkov, M.J., and Knupp, P.M., - “Triangular and Quadrilateral Surface Mesh Quality Optimization Using Local Parametrization”, Preprint submitted to Elsevier Science, 2002
- [10] – OsiriX.
<http://homepage.mac.com/rossetantoine/osirix/>
- [11] – 3D Doctor.
<http://www.ablesw.com/3d-doctor/>
- [12] – 3D Slicer.
<http://www.slicer.org/>
- [13] – Luccichenti, G., Cademartiti, F., Pezzella, F.R., Runza, G., Belgrano, M., Midiri, M., Sabatini, U., Bastianello, S., and Krestin, G.P. – “3D reconstruction techniques made easy: know-how and pictures”, *Eur. Radiol.*, **15**, pp. 2146-2156, 2005.