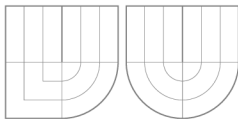
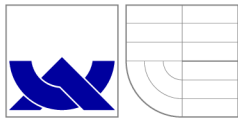




BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND
MULTIMEDIA



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

Delaunay-based Vector Segmentation of Volumetric Medical Images

PHD THESIS

DISERTAČNÍ PRÁCE

AUTHOR

AUTOR PRÁCE

Ing. Michal Španěl

SUPERVISOR

VEDOUCÍ PRÁCE

Doc. Ing. Přemysl Kršek, Ph.D.

BRNO 2010

© Copyright July 2010
Michal Španěl
All rights reserved

Declaration

I declare that this dissertation thesis is my original work and that I have written it under lead of Doc. Ing. Přemysl Kršek, Ph.D. All sources and literature that I have used during elaboration of the thesis are correctly cited with complete reference to the corresponding sources.

.....
Michal Španěl
July 2010

Acknowledgments

I would like to thank to Doc. Ing. Přemysl Kršek, Ph.D. for his lead and valuable suggestions. Moreover, special thanks to my family and all the closest friends for their patience and encouragement.

Abstract

Image segmentation plays an important role in medical image analysis. Many segmentation algorithms exist. Most of them produce data which are more or less not suitable for further surface extraction and anatomical modeling of human tissues. In this thesis, a novel segmentation technique based on the 3D Delaunay triangulation is proposed. A modified variational tetrahedral meshing approach is used to adapt a tetrahedral mesh to the underlying CT volumetric data, so that image edges are well approximated in the mesh. In order to classify tetrahedra into regions/tissues whose characteristics are similar, three different clustering schemes are presented. Finally, several methods for improving quality of the mesh and its adaptation to the image structure are also discussed.

Keywords

Medical imaging, computed tomography, volumetric data, image segmentation, surface reconstruction, surgery planning, custom-made implant, Delaunay triangulation, variational tetrahedral meshing, sliver elimination, feature extraction, clustering.

Bibliographic citation

Michal Španěl: *Delaunay-based Vector Segmentation of Volumetric Medical Images*, PhD thesis, Brno University of Technology, Faculty of Information Technology, Brno, 2010.

Contents

1	Introduction	3
2	Motivation	5
2.1	Anatomical Models	5
2.2	Medical Image Segmentation	8
2.3	Thesis Objectives	9
3	Background: Medical Image Processing	11
3.1	Computed Tomography	11
3.2	Magnetic Resonance	11
3.3	CT/MRI Data Preprocessing	12
3.4	Image Segmentation	20
3.5	Segmentation as Clustering	21
3.6	Difficulties of Medical Image Segmentation	21
4	Background: Delaunay Triangulation and Meshing	25
4.1	Delaunay Triangulation	26
4.2	Constrained Delaunay Triangulation	28
4.3	Mesh Quality	29
4.4	Isotropic Meshing	32
4.5	Variational Meshing	35
5	State of the Art in Anatomical Modeling	39
5.1	Medical Image Segmentation	39
5.2	Surface Reconstruction	45
5.3	Unstructured Meshing	47

6	Delaunay-based Vector Segmentation	51
6.1	Delaunay Triangulation for Image Segmentation	52
6.2	Delaunay-based Vector Segmentation	53
6.3	Data Preprocessing	54
6.4	3D Edge and Corner Detection	56
6.5	Initial Delaunay Triangulation	59
6.6	Iterative Adaptation	59
6.7	Mesh Segmentation	68
7	Experimental Results	77
7.1	Surface Accuracy	78
7.2	Mesh Quality	86
7.3	Mesh Segmentation	91
7.4	Runtime Statistics	98
7.5	Summary and Future Work	99
8	Conclusion	103
	Bibliography	105
	Sample Results	119

List of Figures

2.1	Three alternative volume rendering methods.	6
2.2	Surface models for plastic surgery.	6
2.3	Custom-made implant.	7
2.4	3D geometric modeling of human tissues.	7
2.5	Segmented CT slice.	8
2.6	Artifacts present in the CT image data.	9
3.1	Sample CT and MRI slice through the human head.	12
3.2	Adjusting density window of the CT image.	13
3.3	Result of the power-law contrast enhancement.	14
3.4	Result of the histogram equalization.	15
3.5	Multi-scale contrast enhancement of X-ray images.	16
3.6	Nonlinear anisotropic filtering of MRI data.	17
3.7	Important property of the anisotropic filtering.	18
3.8	Result of the bilateral filtering.	19
3.9	Segmented MRI slice.	20
3.10	Variability of the human anatomy.	22
3.11	Image artifacts caused by movement and metallic objects.	23
4.1	Advancing front method.	26
4.2	Two-dimensional Delaunay triangulation.	26
4.3	Delaunay criterion.	27
4.4	Relationship between Voronoi diagram and Delaunay triangulation.	27
4.5	Incremental construction of the DT.	28
4.6	Partition of constrained edges.	30
4.7	Examples of inappropriately shaped tetrahedra.	30
4.8	Delaunay meshes after sliver perturbation.	32

4.9	Isotropic meshing.	33
4.10	Triangular mesh after edge splitting.	34
4.11	Variational tetrahedral meshing by Alliez <i>et al.</i>	36
5.1	Scheme of the traditional way of 3D anatomical modeling.	40
5.2	Segmentation using a topologically adaptable snake.	41
5.3	Boundary surfaces extracted at different levels of volume image pyramid.	42
5.4	Level-sets of an embedding function.	43
5.5	Femur level set segmentation.	43
5.6	AAM of the metacarpals.	44
5.7	Three-dimensional AAM matching process.	44
5.8	Isosurfaces extracted for three different isovalues.	45
5.9	Surface reconstruction using the Marching Cubes method.	46
5.10	Direct surface extraction from volumetric data.	47
5.11	A tetrahedral mesh produced by isosurface stuffing.	48
5.12	Results obtained by the approach of Dardenne <i>et al.</i>	49
5.13	Results by Alliez <i>et al.</i> meshing the interior of the Stanford bunny.	50
6.1	Results of the proposed vector segmentation method.	51
6.2	Tessellation grid of the DT adapted to the underlying image structure.	52
6.3	Basic scheme of the Delaunay-based vector segmentation.	53
6.4	Neighboring nodes used to calculate the flow between voxels.	54
6.5	Result of the 3D anisotropic filtering of CT data.	55
6.6	Sampled initial set of vertices found by the edge and corner detection.	56
6.7	Tissue-selective edge detection.	57
6.8	Convolution kernels of the 3D Sobel operator in x and z axis.	57
6.9	Positive effect of the tissue-selective edge detection.	58
6.10	Results of the 3D SUSAN detector.	59
6.11	Incremental construction of the Delaunay triangulation.	60
6.12	Iterative mesh adaptation.	61
6.13	The effect of the isotropic edge splitting.	62
6.14	New vertices introduced by the edge splitting.	63
6.15	Pseudo-colored slices through the three-dimensional control space.	64
6.16	Variational tetrahedral meshing.	65
6.17	Illustration of the boundary refinement algorithm.	66

6.18	Slivers that occur close to domain boundaries.	67
6.19	Tetrahedra refinement inserting new vertices.	67
6.20	Result of the mesh segmentation phase.	68
6.21	Tetrahedra adjacency.	71
6.22	Result of the proposed agglomerative merging phase.	71
6.23	Surfaces extracted from a segmented mesh.	74
7.1	Error between two meshes calculated.	78
7.2	Alignment of surfaces.	80
7.3	The bunny colored by interpolating and mapping the surface approximation error.	81
7.4	Histograms of surface approximation error.	82
7.5	Surfaces extracted from meshes with different setting of minimal tetrahedron edge length.	83
7.6	Surface approximation error.	83
7.7	Overall surface approximation error.	84
7.8	Error distribution along sharp edges on the surface.	85
7.9	Influence of K and T_{avg} parameters on the mesh.	87
7.10	Quality of the meshing.	88
7.11	Slivers present in tetrahedral meshes.	89
7.12	Quality of reconstructed surfaces.	90
7.13	Illustration of the F-measure.	93
7.14	Surface reconstruction error when meshing already pre-segmented data.	94
7.15	Surfaces reconstructed from pre-segmented data.	95
7.16	Overall segmentation error of the VSeg method.	96
7.17	Result of the surface reconstruction.	97
7.18	Results of the VSeg method.	98
7.19	Runtime statistics of the VSeg method.	99
7.20	Influence of meshing parameters K and T_{avg} to final surfaces.	100

List of Abbreviations

AAM	Active Appearance Models
CT	Computed Tomography
DT	Delaunay triangulation
CDT	constrained Delaunay triangulation
EM	Expectation Maximization algorithm
FCM	Fuzzy C-means clustering technique
FEM	Finite Element Method
FWT	Fast Wavelet Transform
GMM	Gaussian Mixture Model
ICP	Iterative Closest Point algorithm
MC	Marching Cubes algorithm
MRI	Magnetic Resonance Imaging
RAG	Region Adjacency Graph
RMS	Root Mean Square
VSeg	vector segmentation approach proposed in the thesis
VTM	Variational Tetrahedral Meshing

Chapter 1

Introduction

Medical imaging devices like the Computed Tomography (CT) and the Magnetic Resonance (MRI) can be used to inspect patient body from the inside. These imaging devices produce image data detailing human anatomy within a scanned patient body part. The medical data obtained as planar image slices are mainly used for diagnostic purposes.

The most frequent way of medical diagnostics is investigation of such slices as grayscale images. However, the CT/MRI data make possible to explore other ways of medical diagnostics and treatment. Modern image data visualization and 3D modeling techniques can be used for design of custom-made implants, surgery planning, training, and navigation of surgeons.

Substantial step of many image understanding methods is the segmentation that separates objects (i.e. tissues) in the image. The segmentation plays an important role and provides crucial information for subsequent tasks such as tissue recognition, 3D modeling and visualization.

A novel *vector segmentation* algorithm based on the 3D Delaunay triangulation is proposed in this thesis. Tetrahedral mesh is used to divide a three-dimensional image data into several non-overlapping regions whose characteristics are similar. Methods for isotropic mesh construction and its adaptation to the underlying image structure are presented, so that the final mesh contains larger tetrahedra inside image regions while the size decreases close to the region boundaries.

Applying the vector segmentation a classified mesh whose tetrahedra are grouped into individual regions is obtained. Such mesh contains all information necessary to reconstruct geometry of any region (\sim human tissue). The polygonal surface model can be easily derived.

This thesis is organized as follows. First, a review of recent trends in medical treatment in conjunction with formulation of thesis objectives is given in Chap. 2. Then an introduction to medical image processing is given in Chap. 3, and prerequisites related to the Delaunay triangulation and meshing are summarized in Chap. 4. Further, a survey of existing segmentation techniques relevant to the thesis is given in Chap. 5. The proposed Delaunay-based vector segmentation is described in Chap. 6 including the discussion of some implementation details and specific optimization techniques. Finally, experimental results are shown in Chap. 7 followed by discussion and conclusions.

Chapter 2

Motivation

Modern computer graphics and techniques like the *volume rendering* were introduced to surgeons trying to develop novel methods of the medical treatment. Nowadays, recent research in this area is aimed at:

- direct 3D visualization of medical images (*volume rendering*) [12, 50],
- 3D anatomical modeling [59],
- surgery planning and training – so called *virtual surgery* [24, 27],
- computer aided surgery [11, 68],
- and implants design (Fig. 2.3).

Transparent visualization of the segmented CT/MRI data is significant to acquire accurate medical diagnosis. The term volume rendering [12, 50] is used to describe techniques which allow direct visualization of three-dimensional data.

Benefit of viewing the data as a three-dimensional rather than as individual planes is obvious. The segmented tissues can be observed from any view point, thus the patient anatomy can be investigated much precisely and more easily.

2.1 Anatomical Models

The medical image data can be used to create three-dimensional surface models of human anatomy. After a segmentation of discrete volumetric data, e.g. acquired by CT/MRI imaging, 3D surface models such as those in Fig. 2.2 can be derived.

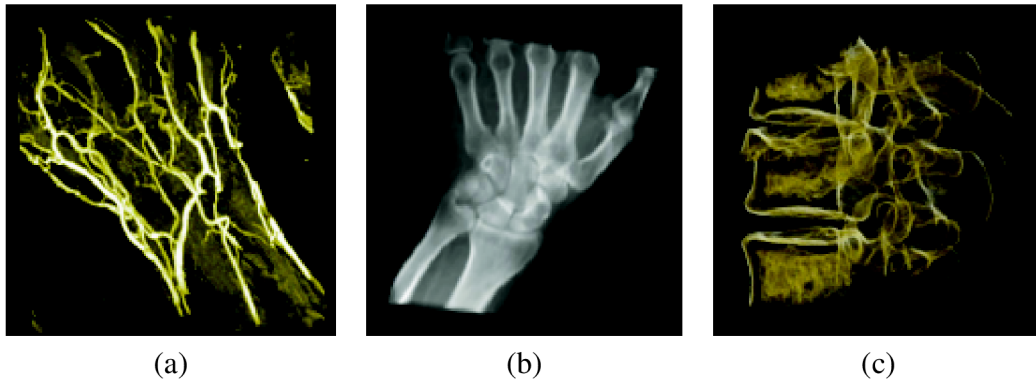


Figure 2.1: Three alternative volume rendering methods: (a) Maxima Intensity Projection (MIP) displays structures of maximal importance; (b) value integration results in X-ray like images; (c) non-photorealistic rendering enhancing contours. Published by Hauser *et al.* [50].

The advantage of a surface representation of human anatomy is that it gives a three-dimensional view from any angle, this is an improvement over the traditional investigation of two-dimensional grayscale images. Surface models can be extracted from segmented data using a number of algorithms, for example the *Marching Cubes* [59]. A detailed study of the most important surface reconstruction algorithms is given in Sec. 5.2.

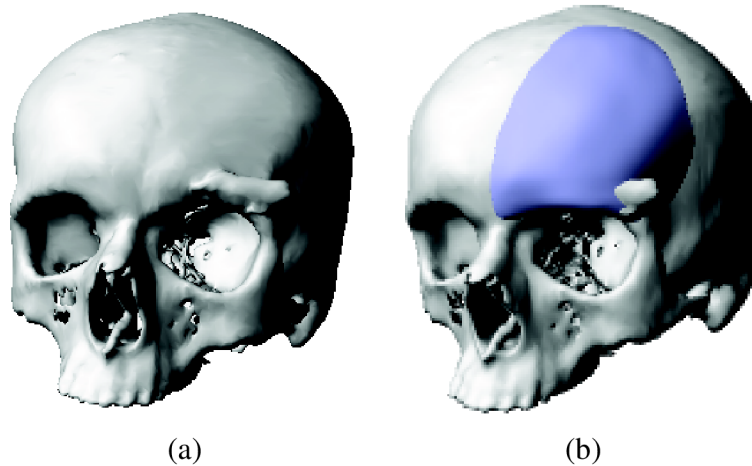


Figure 2.2: Surface model of a human skull (a) and custom made implant (b) for plastic surgery. Realized in cooperation with Faculty Hospital in Olomouc.

Many medical image segmentation algorithms can be found in the literature. This

thesis does not try to summarize them all. Only those techniques, related to the topic of the thesis, are discussed in Chap. 5.

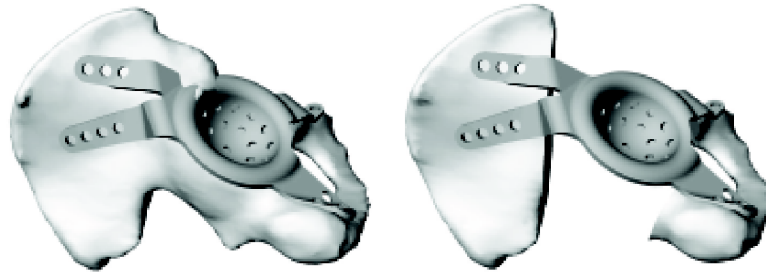


Figure 2.3: Custom-made implant realized by P. Krsek in cooperation with Czech company Beznoska a.s., the producer of orthopedic implants and instruments.

Research at the Faculty of Information Technology in Brno is aimed at 3D surface modeling of tissue geometry for implants design, surgery planning and simulation (see Fig. 2.3). In conjunction with St. Anne's University Hospital in Brno and Faculty Hospital Brno, clinical applications in aesthetic surgery, orthopaedics and dental surgery are investigated [60, 111].

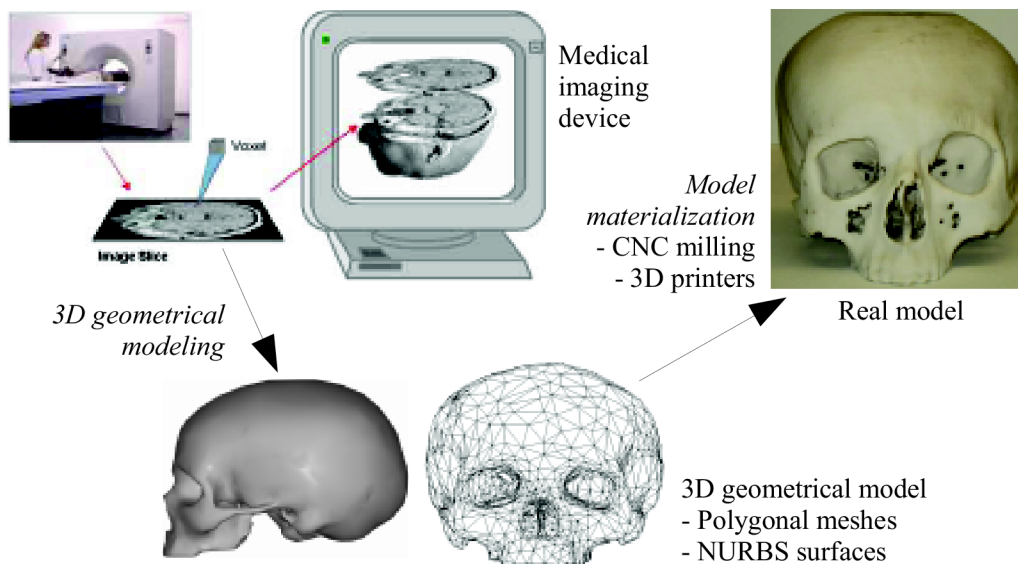


Figure 2.4: 3D geometric modeling of human tissues.

2.2 Medical Image Segmentation

The main goal of the segmentation process is to divide an image into parts that correspond to tissues of particular types. In other words, it is the process of labeling each pixel in a medical image to indicate the tissue type or anatomical structure. The segmentation is one of the most important steps in the analysis of the medical image data. The precise segmentation is crucial for 3D modeling of tissues and anatomical structures. Hence, it helps in diagnosis, surgery planning, surgery simulation, etc.

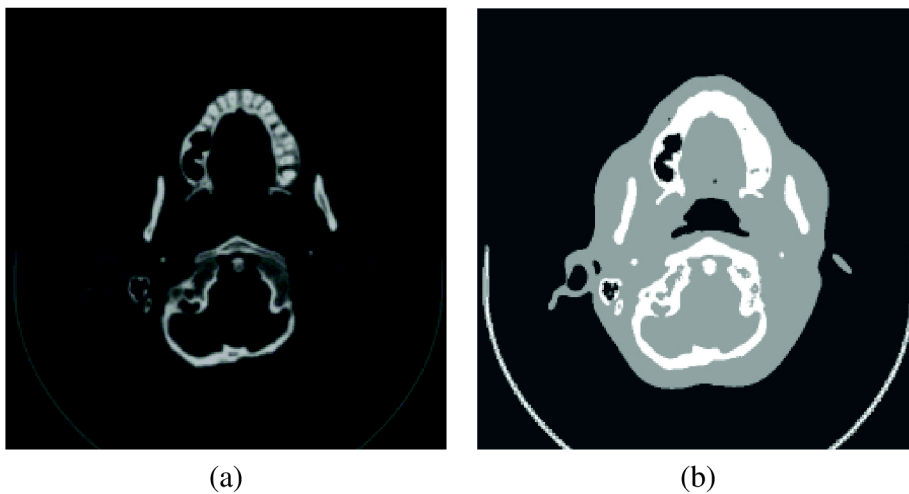


Figure 2.5: The segmented CT slice (b) and the original image (a).

In terms of the medical imaging, input to the segmentation process is a series of image slices, the result of a single CT or MRI scan. More information about medical image segmentation and its formal definition can be found in Chap. 3.4.

2.2.1 Difficulties of the Segmentation

There are many aspects that make general segmentation a difficult task. The first aspect is the imaging process itself. The chosen imaging method provides relevant information about the tissue of interest, but this does not mean that individual tissues will be separable. Strong edges may not be present around the borders. Such tissue is more detectable by the human eye than by even sophisticated computer algorithm.

The second aspect is the complexity and variability of the human anatomy. Due to the nature of the segmentation problem, most of the algorithms are specific to a particular

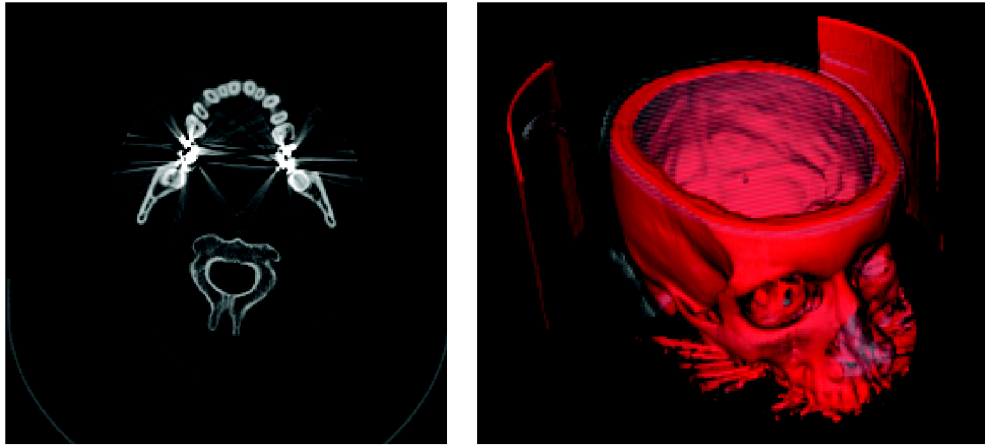


Figure 2.6: Artifacts caused by metallic objects present in the CT image data.

problem. Certain knowledge must be built into the algorithm. Therefore, such method is not suitable for other problems.

Moreover, the CT image data are sometimes damaged by artifacts when metallic objects are present in the patient body (Fig. 2.6).

Another important feature of the segmentation is automation. Performing automated segmentation still remains one of the most difficult problems. Although many researchers have shown success with automation in some cases, there is no generic algorithm which can perform automatic segmentation on any given data set.

2.3 Thesis Objectives

This thesis aims at the anatomical modeling of human tissues and techniques of medical image segmentation suitable for this kind of modeling. A 3D model of desired tissue is made on the basis of the segmented data. Most often, the segmentation step is done semi-automatically. Results of the segmentation are manually verified and corrected which may be very time consuming.

An inconvenience can be found in the assumption that a surgeon or assistant is skilled in the segmentation and its verification. In addition, it is necessary to do a lot of additional work different from traditional medical treatment. Even though the surgeons understand all the advantages that these techniques may bring into the medicine, only few of them are

willing to do such work. Therefore, it is very hard to establish these techniques in clinical practice.

Department of Computer Graphics and Multimedia at FIT BUT, namely *P. Kršek* and *M. Španěl*, cooperates on the research of the anatomical modeling in clinical applications [60, 111] for many years trying to establishing them in practice. It was important to keep in mind that the field of clinical applications is very wide in our case while objectives of this thesis were formulated:

Accurate surface approximation. In case of anatomical modeling, an error between reconstructed surfaces of human tissues and a "ground truth" must be minimal to guarantee correctness of a planned surgery. Therefore, more attention is given to surface reconstruction methods that work directly with volumetric data without any post-processing steps which may increase the surface error.

General algorithm. Because of the wide field of clinical applications, knowledge-based methods of tissue modeling which uses atlas of human anatomy are not suitable. Besides, in case of traumatic injury, most of the knowledge-based methods fail because such events are not present in training data. Unfortunately, traumatic injuries are typical incidents when the anatomical modeling helps in surgery planning. The goal is to propose a general algorithm, in a certain manner, that is not aimed at concrete treatment, tissue type, or situation.

Real data. Difficulty of the segmentation is the analysis of real CT/MRI data. It is important to deal with noise in the imaging process as well as inhomogeneity of the tissues. Some pre-processing algorithms (noise removal, MR inhomogeneity correction, etc.) as well as robust segmentation algorithms must be suggested.

High-quality surface meshes. Most frequently, anatomical models are used for surgery planning and custom-made implants design. However, mesh structure suitable for numerical simulations is necessary for some tasks. Hence, high-quality meshes should be produced by the modeling being able to describe interior structure of tissues as well.

High degree of automation. The goal is to develop segmentation algorithm which will work mostly automatically. Minimal manual corrections of the segmentation are required. Because manual corrections are always needed, it must be easy to modify the final segmentation.

Chapter 3

Background: Medical Image Processing

An increasing number of different diagnostic imaging techniques have been introduced in clinical applications in the last few years. Nowadays, medical images are obtained from different acquisition devices including *Computed Tomography (CT)*, *Magnetic Resonance Imaging (MRI)*, *Ultrasound*, etc. Each of them carries both the structural and the functional information on human tissues. A short overview of medical imaging and medical image processing techniques related to the topic of the thesis is given in this chapter.

3.1 Computed Tomography

Computed tomography [51,83] is a modern extension of the traditional X-ray examination. The X-ray beam scans a slice of the anatomy from multiple angles. Each slice pixel is then calculated combining the measurements from the multiple angles. The CT produces data in a planar 2D form as a series of slices through the examined part of the patient body.

The CT intensity is relative to that of water which is zero. In the same manner, the different tissues have a different predefined intensity values. This feature makes possible to use some low-level automated segmentation. The CT is very sensitive to differences in density and produces good anatomical images of organs and soft tissues [72].

3.2 Magnetic Resonance

The magnetic resonance [82,83] represents information of a chemical nature. The different intensities in the image reflect mainly the density of hydrogen atoms. The method has its

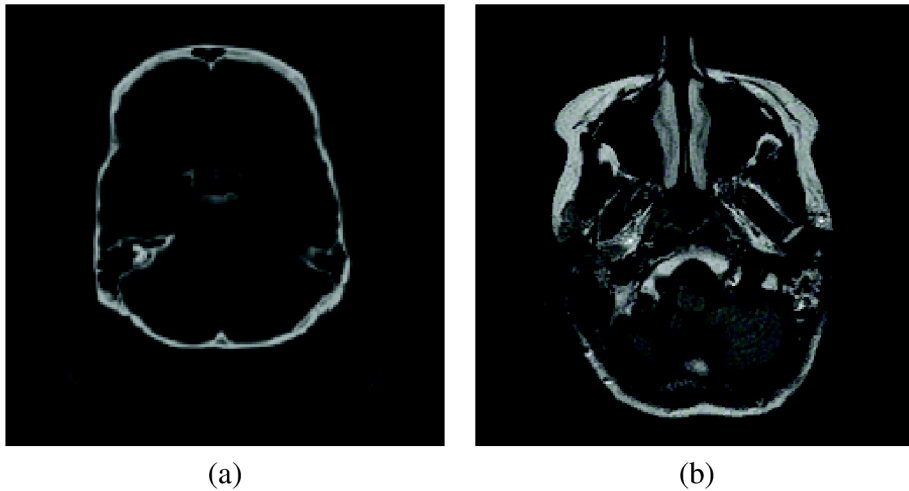


Figure 3.1: Sample CT (a) and MRI (b) slice through the human head. Notice that there are black areas in place of the bones in the MRI slice. The bones don't disturb the MR signal.

theoretical base in advanced nuclear physics. The main advantage of the MR imaging is that image intensity varies for different soft tissues. In addition, bones do not disturb the MR signal (see Fig. 3.1). Therefore, it is used to examine soft tissues and to find pathological changes like tumours.

3.3 CT/MRI Data Preprocessing

In medical imaging, different tissues can appear very similar, making it difficult to interpret the image. The visual examination of medical images is essential in the diagnosis. In order to visualize relevant anatomy, the image can be adjusted through a process known as *windowing* [119]. In fact, the windowing is a simple linear intensity scaling that increases contrast between tissues of interest. Intensity scaling allows one to focus on specific intensity bands in the image (see Fig. 3.2) by stretching the band of interest into the dynamic range of a display.

3.3.1 Enhancing Contrast

Furthermore, to obtain images with a greater detail of relevant anatomy, a contrast medium which highlights certain tissues is typically injected into a patient body [15]. However, for

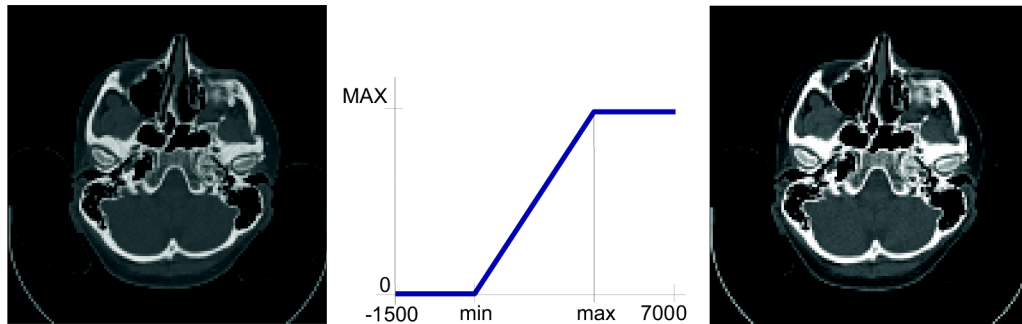


Figure 3.2: Adjusting density window of the CT image using linear intensity scaling.

many applications, there is no known contrast medium capable of differentiating between the relevant tissue types. For these reasons, it is helpful to enhance contrast by more computationally intensive image processing techniques.

A comprehensive survey of contrast enhancing methods applied in the medical imaging can be found in [4]. Here, a brief overview of selected methods is given with a reference to literature. Some techniques, the power-law technique, the anisotropic filtering and the bilateral filter, are described in more detail as they are further utilized in the thesis.

3.3.2 Enhancement in Spatial Domain

Most of the contrast enhancement techniques [4, 46] can be classified into two groups. Enhancement in *spatial domain* manipulates image pixels directly, while *frequency domain* approaches modify the Fourier transform of an image. Many spatial domain methods have been applied in the past:

- gray level transformations – linear, logarithmic, power-law and piecewise-linear transformation functions,
- histogram equalization and matching,
- linear and non-linear spatial smoothing filters – mean and median filtering,
- sharpening smoothing filters – unsharp masking, etc.

Power-law transformations [46] are useful for general-purpose contrast manipulation. The power-law transformation, also called *gamma correction*, is defined by the formula:

$$s = cr^\gamma \quad (3.1)$$

where c and γ are positive constants. When γ is less than 1, The transformation maps a narrow range of dark values into a wider range, and it does the opposite for the higher values. The inverse effect, compression of gray levels, is obtained when γ is greater than 1.

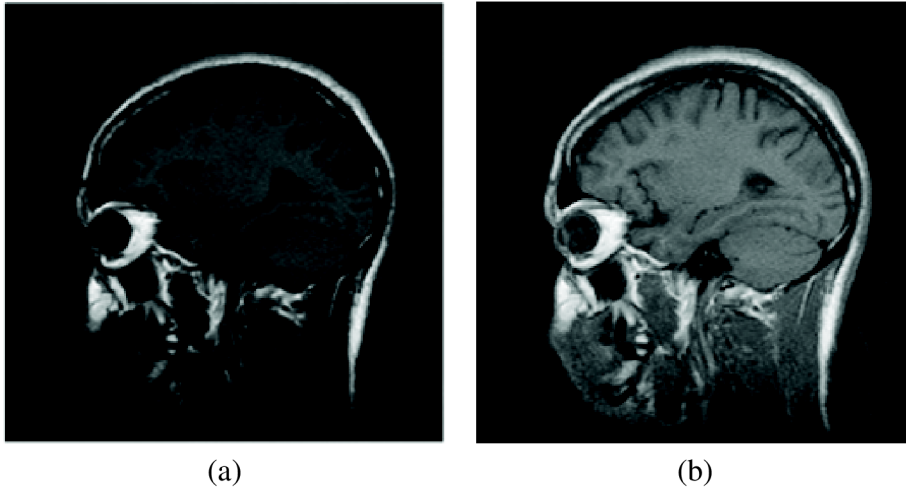


Figure 3.3: Result of the power-law contrast enhancement.

Histogram equalization is widely utilized for global image enhancement [16]. The image contrast is adjusted by mapping obtained from the integral of the image histogram. The histogram equalization assigns approximately equal number of pixels to each user-specified gray-scale levels \sim uniform distribution. Although this method is very simple, it does not take into account local details. In addition, global histogram equalization has the undesired effect of overemphasizing noise.

In diagnostic medical images, local details may be more important than global contrast. Therefore, a number of local adaptive histogram equalization and local adaptive contrast enhancement methods [16, 53, 88] have been proposed in the past. These algorithms map the gray values of pixels using the relationships obtained from the local histograms.

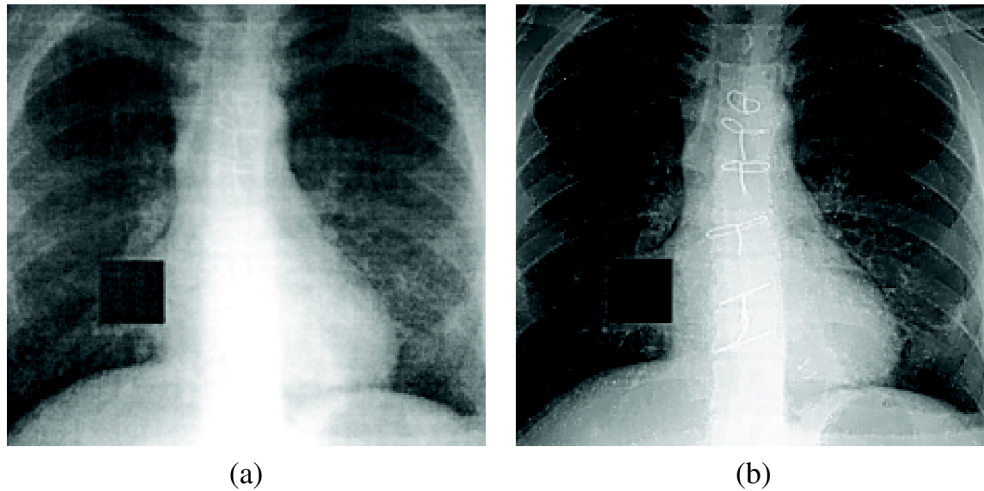


Figure 3.4: Result of the contrast enhancement based on the histogram equalization method proposed in [16] (a) original image (b) equalized image.

3.3.3 Frequency Domain and Multi-scale Techniques

Filtering can be also done in the frequency domain [39, 39, 118]. Low frequencies in the Fourier transform give the gray-level appearance of a smooth image. High frequencies show detail, such as edges and noise. A filter that reduces high frequencies while passing low frequencies is called a *low-pass filter* and provides noise suppression or image smoothing. A filter with the opposite characteristics is called a *high-pass filter*. The commonly used *Butterworth* high-pass and low-pass filters are presented in [4] as a good contrast enhancing filters.

Multi-scale methods can decompose an image into components, which can be used to improve contrast in the image. The *Laplacian Pyramid* [39] and the *Fast Wavelet Transform (FWT)* [4, 58] are both typical multi-scale methods. In general, enhancement by means of the Laplacian Pyramid was applied to X-ray images [101]. Wavelet-based methods were mainly used in the context of mammography [66]. Besides, there are also some applications to the CT/MRI images.

Performance comparison of both methods can be found in [30], where Dippel *et al.* stated that enhancement based on the FWT suffers from one serious drawback, the introduction of visible artifacts when large structures are enhanced strongly. The Laplacian Pyramid allows a smooth enhancement of large structures, such that visible artifacts can be avoided.

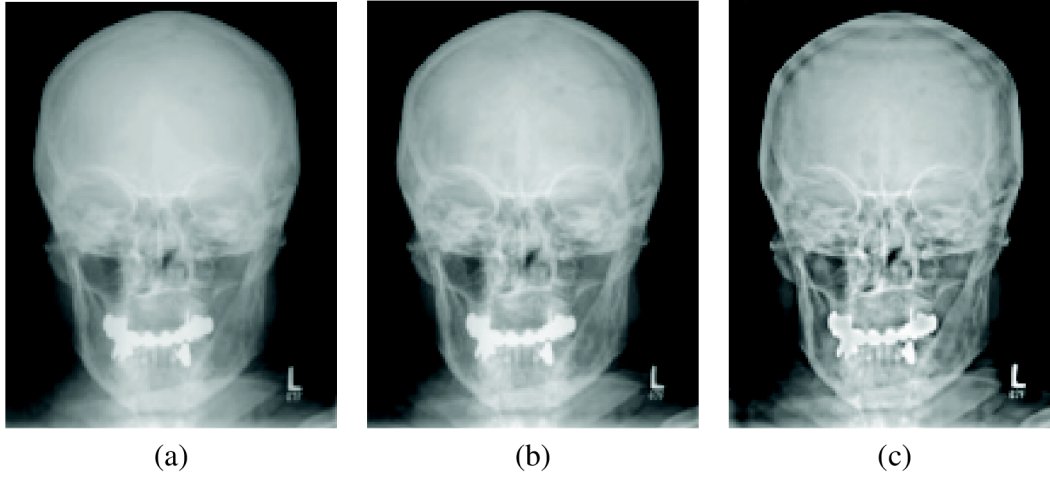


Figure 3.5: Multi-scale contrast enhancement of X-ray images: the original image (a); skull processed with Laplacian pyramid (b); and skull processed with wavelet pyramid (c) [30].

3.3.4 Anisotropic Filtering

Anisotropic filtering [92] performs piecewise smoothing of the original image. Its strength lies in the fact that it deals with local image structures which can be preserved and their positions will not be affected. The filtering process can be formulated as a diffusion. The filtering is suppressed at boundaries by locally adaptive diffusion strength. The diffusion process of anisotropic filtering can be expressed by the equation:

$$\frac{\partial}{\partial t} I(\bar{x}, t) = \text{div}(c(\bar{x}, t) \times \nabla I(\bar{x}, t)) \quad (3.2)$$

The function $I(\bar{x}, t)$ is the image intensity. The diffusion strength is controlled by $c(\bar{x}, t)$, where \bar{x} represents the spatial coordinates, and t is the iteration step. The diffusion function $I(\bar{x}, t)$ depends on the magnitude of the image intensity gradient and mainly diffuses within homogenous regions and does not affect edges and boundaries. The diffusion function 3.3 has been used frequently.

$$c(\bar{x}, t) = \exp\left(-\left(\frac{|\nabla I(\bar{x}, t)|}{\sqrt{2\kappa}}\right)^2\right) \quad (3.3)$$

The parameter κ is chosen according to the noise level and the edge strength. The relationship between the parameter κ and the gradient ∇I can be explained by the flux function

$\phi = c \times \nabla I$. For example, maximum flux is generated when the gradient equals to κ . Below κ , the flux reduces to zero, because only minimal flux takes place in homogeneous regions. Above κ , the flux again decreases to zero, stopping diffusion at locations of high gradients. A proper choice of the diffusion parameter not only preserves, but also enhances edges.

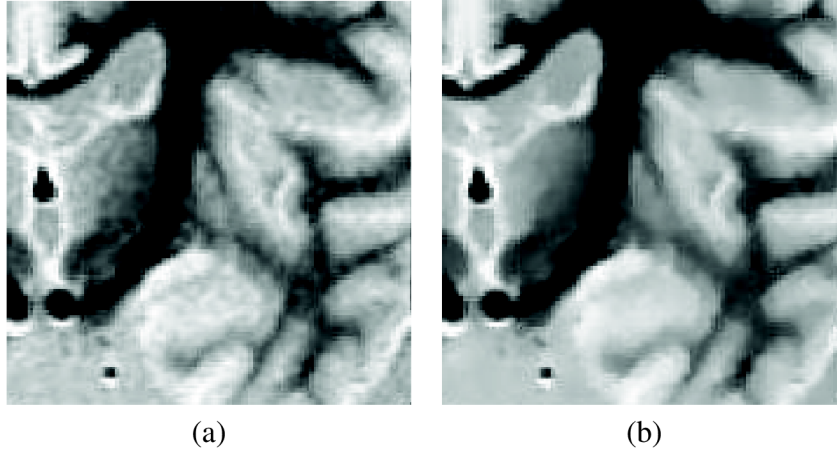


Figure 3.6: Nonlinear anisotropic filtering of MRI data: the original image (a); and result of the filtering (b) [43].

The filtering of discrete signals requires a reformulation of the method. To filter discrete 1D signal, it can be shown [92] that

$$I(t + \Delta t) \approx I(t) + \Delta t \times \frac{\partial}{\partial t} I \quad (3.4)$$

$$= I(t) + \Delta t \times (\phi_{right} - \phi_{left}) \quad (3.5)$$

where ϕ_{right} and ϕ_{left} are the flow contributions estimated by the flux function.

Local gradient estimates are calculated as differences between neighboring image pixels instead of differentiation. Stability of the iterated processing can be obtained by choosing a proper integration constant $\delta t = 1/5(1/7)$ while using 4-connected (8-connected) neighborhood structure.

Since an image usually consist of several objects with different contrasts, it is important to be adaptive to different areas with different gradients. The general idea is to apply adaptive filtering. The diffusion process depends critically on the value of κ . Therefore, the gradient in different areas of the image can be calculated and κ is chosen less than the edge gradient and larger than the average value of noise.

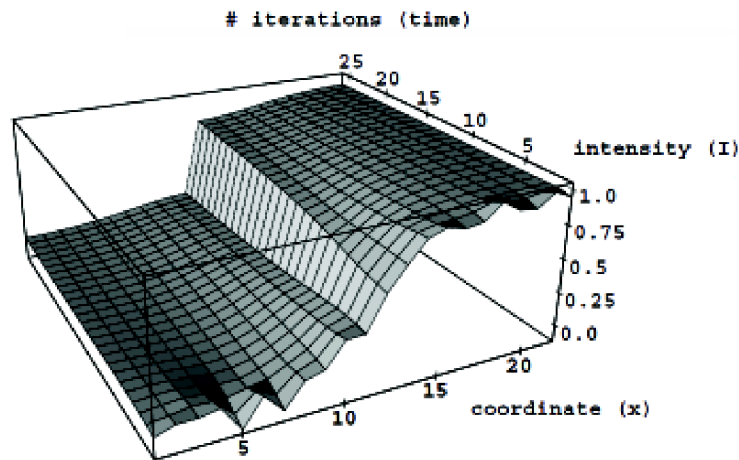


Figure 3.7: Illustration of the important property of the anisotropic filtering – iterative edge sharpening and noise suppression. Reprinted from [43].

Gerig *et al.* [43] presented an extension of the anisotropic filtering to 3D and multichannel data. Results of their MRI brain images filtering illustrate efficient noise reduction in homogeneous image regions, while object contours and boundaries are not only preserved, but even improved (Fig. 3.6).

3.3.5 Bilateral Filtering

A very popular image filtering technique is a *bilateral filter*. Similarly to the anisotropic filter, also the bilateral filter is able to remove noise while preserving important features like edges in the image.

A simple approach to the image smoothing is averaging of nearby pixels to compute an estimate of the original pixel value. The Gaussian low-pass filter performs an averaging using a set of weights defined over a normal distribution such that points nearby the sample point have greater weights and more distant points have the smaller weights. This type of filtering is independent of the underlying image structure, thus blurring of edges can be seen as a side effect of the filtering [7].

The idea of bilateral filtering, defined by Tomasi [104], is to smooth images while preserving edges by means of nonlinear combination of nearby pixel values. In order to achieve this, the influence of pixels is weighted by two different Gaussian functions g and c – one in the image space (the *geometric closeness*) and second in the signal space (the

photometric similarity). The idea of weighting in the signal space assumes there will be a large difference in the signal close to sharp edges, so the difference between the sample point and nearby pixels can be used to evaluate influence of each pixel within a local neighbourhood. The following equation is the discrete formulation of bilateral filtering:

$$I(\bar{x}) = \frac{1}{k(\bar{x})} \sum_{\bar{r} \in R} I(\bar{r}) g(\|\bar{r} - \bar{x}\|) c(|I(\bar{r}) - I(\bar{x})|) \quad (3.6)$$

where $k(\bar{x})$ is a normalization factor – the sum of all weights given by the g and c functions, and R is a local neighbourhood of the sample pixel \bar{x} .

The set of contributions from c varies within the image, as they depend on the actual set of differences observed across the neighbourhood. Therefore, in many papers [87, 123], distant optimization and approximation techniques have been presented to provide not only effective, but also fast filtering method.

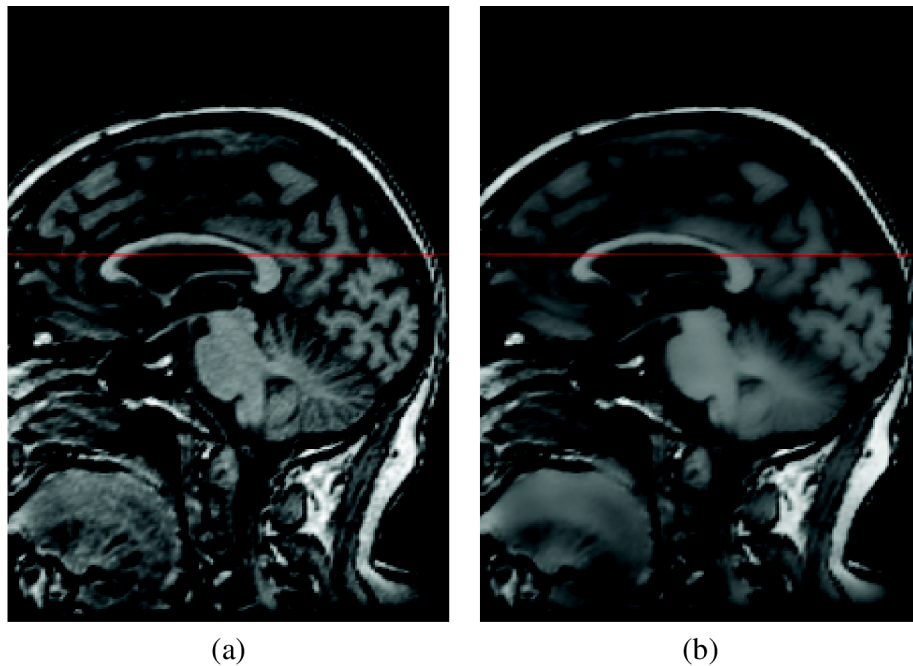


Figure 3.8: These images show results of the 3D bilateral filtering. Notice the bilateral filter performs smoothing while preserving important image features [7].

The bilateral filtering approach combining spatial and signal weights has shown to be robust and flexible to a variety of applications. For an example, a mesh smoothing algorithm based on bilateral filtering has been presented by Jones *et al.* [55]. Finally, an

extension of the original formulation of bilateral filtering for use on 3D volumetric data has been proposed by Bethel *et al.* [7].

3.4 Image Segmentation

The image segmentation can be formally defined [97] as the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. The image segmentation is typically used to locate objects and boundaries in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

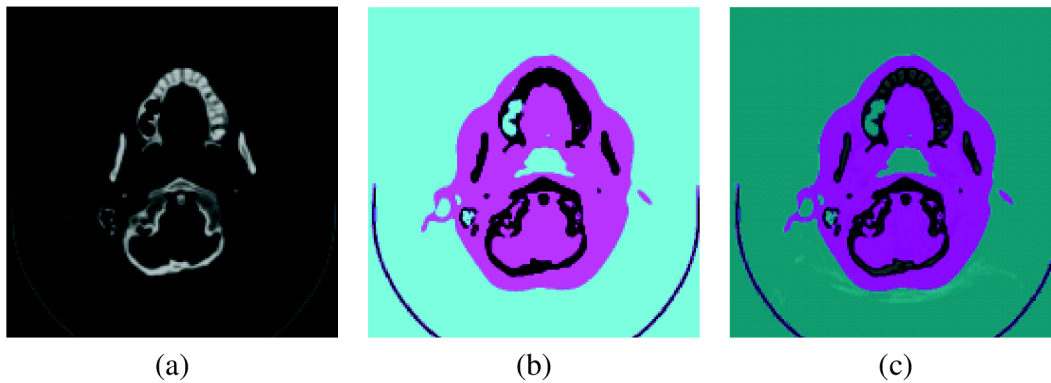


Figure 3.9: Result of a simple histogram thresholding. The segmented image (b) contains three different labels: hard tissues (i.e. bones), soft tissues and the bright background.

In medical image processing, the main goal of the segmentation is to divide an image into parts that correspond to tissues of particular types. In other words, label of each pixel in a medical image indicates the tissue type or anatomical structure. In terms of the CT/MRI medical imaging, input to the segmentation process is a series of grayscale slices, the result of a single CT or MRI scan.

The segmentation is one of the most important steps in the analysis of the medical image data. The precise segmentation is crucial for 3D modeling of tissues and anatomical structures. Hence, it helps in diagnosis, surgery planning, surgery simulation, etc.

3.5 Segmentation as Clustering

One natural view of the segmentation [39] is that we are attempting to determine which components of a data set naturally belong together. This problem is known as *clustering*. There is a wide literature [4, 39, 46] discussing the clustering for image segmentation.

In this thesis, three different clustering techniques will be proposed to partially solve the segmentation problem. First two techniques (the Fuzzy C-means (FCM) [95] algorithm and clustering based on the Gaussian Mixture Model (GMM) [81]) are built upon the idea of grouping. Data items that "*make sense*" are collected together according to some model. The third graph-based algorithm [8] is a partitioning technique. A large data set is decomposed into pieces that are "*good*" according to our model [39].

The new graph segmentation technique presented by Boykov *et al.* [8] belongs to the group of algorithms building search trees for detecting augmenting paths in the graph. The drawback of their approach is that the augmenting paths found are not necessarily shortest augmenting path. The algorithm iteratively repeats the following three stages:

- growth stage – search trees grow until they touch giving a shortest path,
- augmentation stage – the found path is augmented, search trees are broken into forests,
- adoption stage – trees are restored.

In most examples, their min-cut/max-flow algorithm proceeds faster than any other method, including the *push-relabel* [45] and *Dinic's algorithm* [29] which are known to outperform other min-cut/max-flow techniques.

Detail analysis of different clustering techniques is not the primary aim of the thesis. Please, follow given references to the literature for more details.

3.6 Difficulties of Medical Image Segmentation

Due to the nature of the segmentation problem, there is a lot of aspects that make the segmentation a difficult task [83, 121]:

- the imaging process itself,
- variability of the human anatomy,

- imaging a moving patient,
- artifacts appear in the data,
- automation.

3.6.1 Imaging Process

The first aspect is the imaging process itself. The imaging modality, for example MRI, CT, or ultrasound, is chosen so that its interactions with the tissues of interest will provide relevant information about the tissue in the resulting output image. But this does not mean that the anatomical feature will be separable from its background. Strong edges may not be present around borders of tissues. Due to noise in the imaging process as well as to inhomogeneity of the tissue, such region is more detectable by the human eye than by even sophisticated computer algorithms. Simple techniques, such as thresholding and pixel-based clustering, are not usually sufficient when applied to medical data.

3.6.2 Human Anatomy

The second fundamental aspect that makes segmentation difficult is the complexity and variability of the anatomy that is being analyzed. This makes general segmentation a difficult problem.



Figure 3.10: Illustration of variability of the human anatomy. Due to a serious damage of the hip joint, the precise segmentation is very difficult, even though a certain knowledge of the "common" anatomy is built into the segmentation process.

3.6.3 Artifacts

The CT scanners usually require the patient to remain extremely still for several minutes. This is often difficult, and in the case of involuntary movement (Fig. 3.11), such as the heart beating or breathing, becomes impossible.

The conventional algorithms used in CT sometimes produce artifacts (i.e. the impression of features which are not actually there). This is particularly true when metallic objects are present in the patient's body.

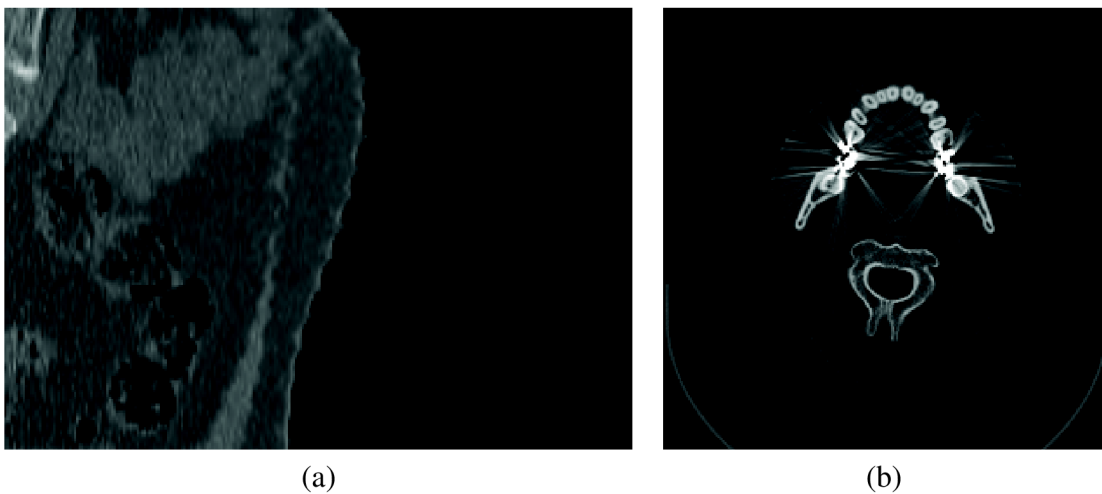


Figure 3.11: Image artifacts caused by breathing (a) and metallic objects (b).

3.6.4 Automation

An important desired feature of the segmentation is automation. Performing automated segmentation still remains one of the most difficult problems in the world of segmentation. Although researchers have shown success with automation in some cases, there is no generic algorithm which can perform automatic segmentation on any given data set.

Chapter 4

Background: Delaunay Triangulation and Meshing

A mesh generation aims at tessellation of a bounded 3D domain Ω with tetrahedra [42]. Algorithms for 3D mesh generation have been intensively studied over the last years. Basically, three main families of algorithms have been described in the literature:

- Octree methods [126, 127],
- Advancing front methods [54, 122],
- Delaunay-based methods [1, 14, 25, 61, 70].

The *octree technique* recursively subdivides the cube containing the geometric model until the desired resolution is reached. *Advancing front method* (Fig. 4.1) starts from a boundary and moves a front, adding new vertices, from the boundary towards empty space within the domain. Several heuristics are used to ensure that the generated tetrahedra have desired shape and size. Global optimization steps can also be performed to improve the mesh quality. A good survey of these methods can be found in [14, 86].

This chapter briefly summarizes basic principles and difficulties of a Delaunay triangulation which is one of the most popular triangulation and meshing method. The Delaunay triangulation and its construction methods are introduced, while those methods later used in the thesis are described in more detail. The following definitions are based on [42].

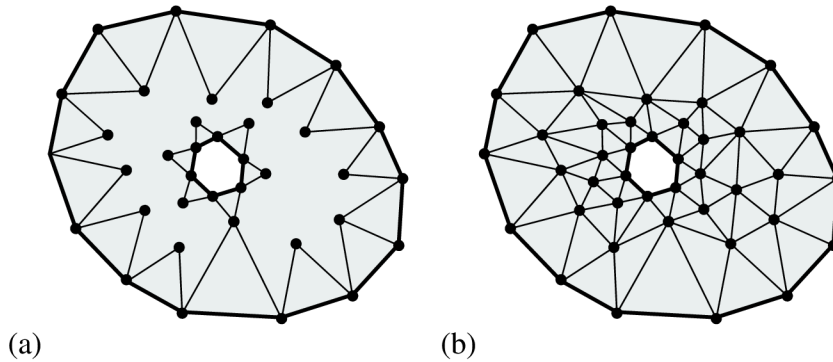


Figure 4.1: Advancing front method starts meshing from the boundary [59].

4.1 Delaunay Triangulation

Every triangle (tetrahedron in 3D space) of the *Delaunay triangulation* (DT for short) satisfies the *Delaunay criterion* shown in Fig. 4.3. This criterion, referred to as the *empty sphere criterion*, means that every circumcircle (circumsphere in 3D) associated with the mesh element e does not contain any vertices, except those of the element e . This criterion is a characterization of the Delaunay triangulation and it leads to several other characteristics.

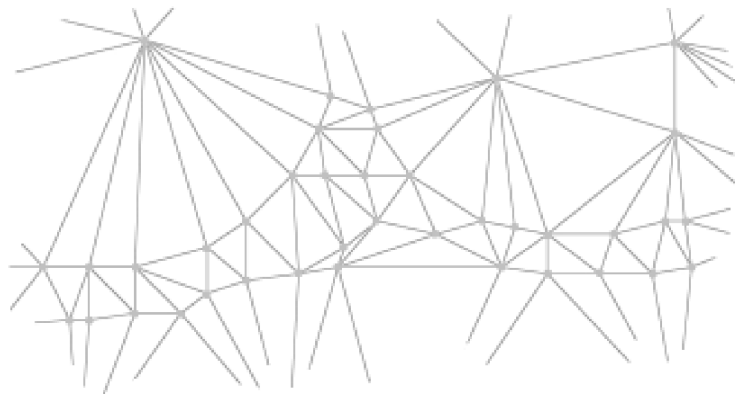


Figure 4.2: Two-dimensional Delaunay triangulation.

The DT maximizes the minimum angle, and minimizes the maximum smallest enclosing circle for each triangle. Therefore, the Delaunay triangulation of a set of points generates regularly shaped triangles and is preferred over alternative triangulations.

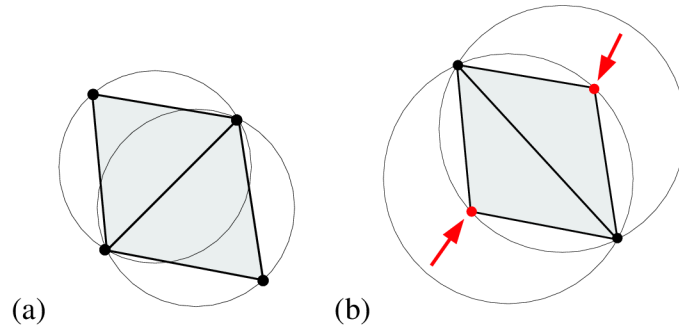


Figure 4.3: Delaunay criterion: satisfied empty sphere criterion (a); and violated criterion (b). Reprinted from [59].

Delaunay triangulations are also very attractive from a robustness point of view due to simplicity of the Delaunay criterion. In addition, various local transformations can be used to improve quality of the triangulation.

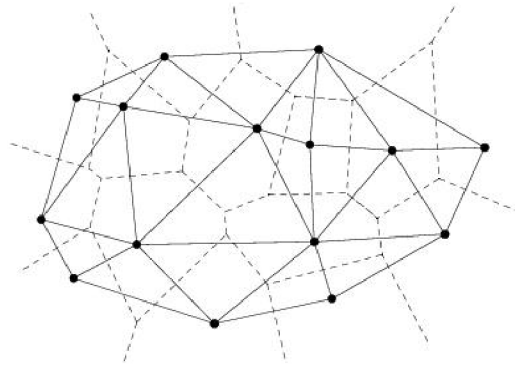


Figure 4.4: Figure shows the relationship between the *Voronoi diagram* [42] (dashed) and the Delaunay triangulation (solid).

The Delaunay triangulation can be constructed by using several methods. Most common is the *Incremental Method* which will be described first [42].

4.1.1 Incremental Construction Method

Be T^i the Delaunay triangulation of the first i points, we consider the $(i + 1)^{th}$ point of this set, denoted as P . The purpose of incremental method is to obtain T^{i+1} the Delaunay

triangulation including P as a vertex. The location of P , with respect to T^i , falls in two categories:

- P is enclosed in T^i (i.e. P lies inside the convex hull/envelope of all vertices in T^i),
- or P is outside of T^i .

In the first case (Fig. 4.5a) the set of elements in T^i whose circumcircle (circumsphere) contains P – the *cavity* of P – is removed from T^i , and the set of elements formed by joining P with external edges of the cavity is added to T^i .

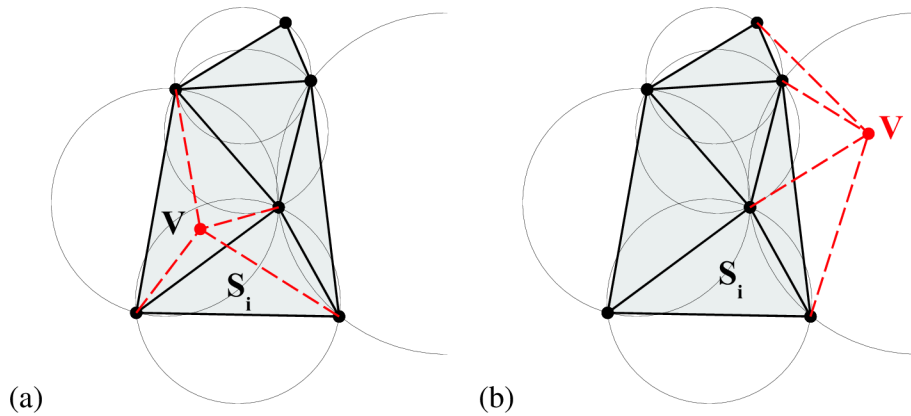


Figure 4.5: Incremental construction of the DT. Inserting point P ($P \in T^i$ on the left side and P outside of T^i on the right) [59].

In the second case, cavity is the same set enriched by the set of elements formed by joining P with the edges in T^i visible from P . Several proofs of the incremental method can be given. Detailed study can be found in [42].

4.2 Constrained Delaunay Triangulation

Given a set of *constraints* specified as a set of edges, or a set of edges and faces in 3D, *Constrained Delaunay triangulation (CDT)* is a triangulation where those constraints remain as entities of the resulting mesh [42].

The CDT leads to the problem of recovering the edge/face constraints from the initially constructed triangulation, or simply, the *problem of edge/face recovery*. Such problem has

been successfully solved in two-dimensional spaces, while it is still under active investigation in the 3D space.

There are two classes of methods depending on how the constraints must be satisfied. The first kind performs local modifications to enforce the given constraints, while the other kind tends to modify the constraints and creates an admissible set of constraints. A *constraint partitioning* method is a simple representative of the second class.

4.2.1 Constraint Partitioning Method

Some edges in constraints are not edges of the given triangulation. The key idea is to re-triangulate every triangle intersected by a constrained edge while ensuring that the created sub-edges are in the resulting triangulation. Each missing constrained edge is processed as follows

- Find intersection points of triangles and the constrained edge AB . Let P_1, P_2, \dots, P_n be these points.
- Introduce the edges $AP_1, P_1P_2, \dots, P_nB$ in the triangulation.
- Re-mesh the triangles while maintaining this list of edges.

Each missing edge, with endpoints A and B, can be then retrieved in the triangulation as the edges $AP_1, P_1P_2, \dots, P_nB$. In practice, a unique operator is required that re-meshes a given triangle with two sub-triangles having a specified point lying on one of its edges as vertex.

Advantage of the constraint partitioning method is that it can be easily extended to 3D. On the other side, it may be difficult to guarantee that the mesh remains Delaunay and no poorly shaped tetrahedra appear in the mesh.

4.3 Mesh Quality

An ideal tetrahedron, having the best quality, is equilateral. There are many measures of the quality regarding the ideal shape. The most general one is ratio of the longest tetrahedron

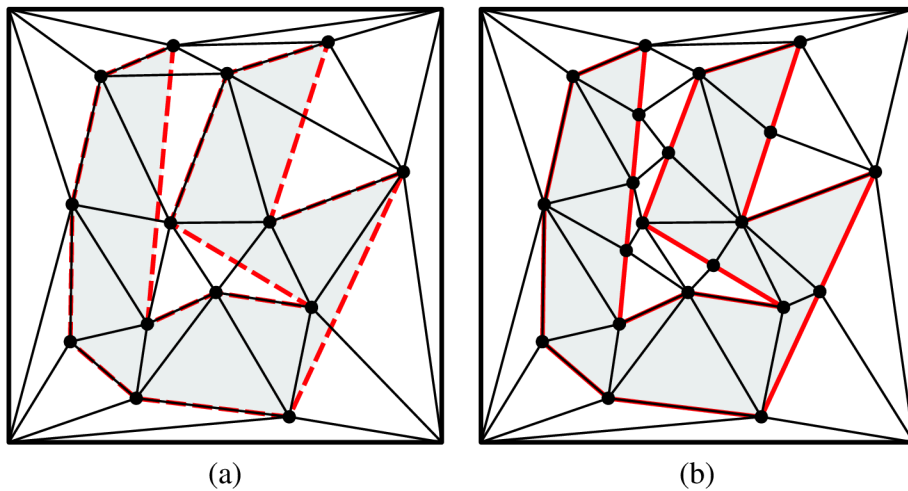


Figure 4.6: Partition of constrained edges not present in the mesh [59].

edge and the radius of its inscribed sphere [59]. The normalized form 4.1 is preferred in practice.

$$Q(t) = \frac{\sqrt{3} l_{max}(t)}{6 r_{ins}(t)} \geq 1 \quad (4.1)$$

In the equation, l_{max} is the length of the longest edge and r_{ins} is the radius of the inscribed circle. The ideal triangle has the normalized quality equal to 1. Any other triangle has the value greater.

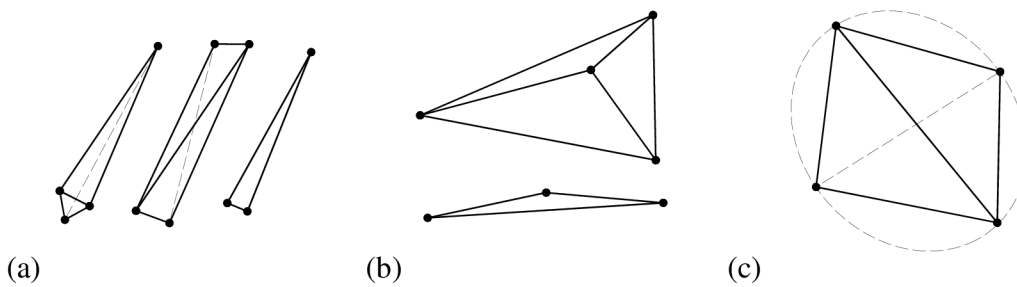


Figure 4.7: Examples of inappropriately shaped tetrahedra – the spear (a); the cap (b); and the sliver \sim almost flat tetrahedron (c).

The radius ratio [105], defined as the ratio of the radius of the inscribed sphere to the radius of the circumsphere, is another popular measure of tetrahedron quality. It is desired

to ensure that radius ratio of all tetrahedra are bounded from below by a constant.

A widely used criterion for the mesh quality is the *minimum dihedral angle*. This measure is more intuitive and geometrically meaningful than the edge-radius ratio or any other quality measure based on a ratio. Some valuable conclusions on different quality measures can also be found in [98].

4.3.1 Delaunay Refinement

Many *Delaunay refinement* methods [18, 19, 99] exist that improve tetrahedra locally by inserting new nodes to maintain the Delaunay criterion. However, most Delaunay refinement algorithms fail at removing all poorly shaped tetrahedra. A special class of almost flat tetrahedra, so called *slivers*, may remain in the triangulation. In the sliver, the minimal dihedral angle can be very close to zero (see Fig. 4.7). The presence of slivers in the mesh may cause troubles for many numerical methods and further processing.

Different approaches of removing slivers from a 3D Delaunay mesh were studied. Cheng *et al.* [17] provide a *sliver exudation* technique based on a weighted Delaunay triangulation is applied to a triangulation obtained by Delaunay refinement. The main strategy of the algorithm consists of assigning a weight to each vertex so that the weighted Delaunay triangulation is free of any slivers after connectivity updates, without any changes over the vertex locations.

The main disadvantage of the sliver exudation is that the process often ends with slivers near the boundary [105]. This is mainly due to the fact that sliver exudation is not allowed to modify the topology of the boundary of the mesh. Hence, weight assignments close to the boundary are constrained and do not remove the slivers.

4.3.2 Sliver Perturbation

Li *et al.* [70] proposed a sliver removal algorithm based on explicit random perturbation of vertices incident to slivers in an almost good mesh. The idea is based on the fact that for any triangle qrs , the region of locations of the vertex p such that the tetrahedron $pqrs$ is a sliver, is very small. Moving the point p out of this region ensures that the tetrahedron is not a sliver anymore, or has disappeared once the Delaunay connectivity is updated. This is achieved by moving the point p to a new location inside a small ball centered at p , whose radius is proportional to the distance from p to its nearest neighbor. Li *et al.* show that for

certain values of the involved parameters, there always exist some points in this ball which are outside all regions that form slivers with nearby triangles.

Tournois *et al.* [105] presented a more efficient algorithm for sliver elimination and improving the dihedral angles of a 3D Delaunay triangulation. The algorithm inspired by Li's random perturbation works in a more deterministic way by choosing a favored perturbation direction for each vertex incident to one or more slivers.

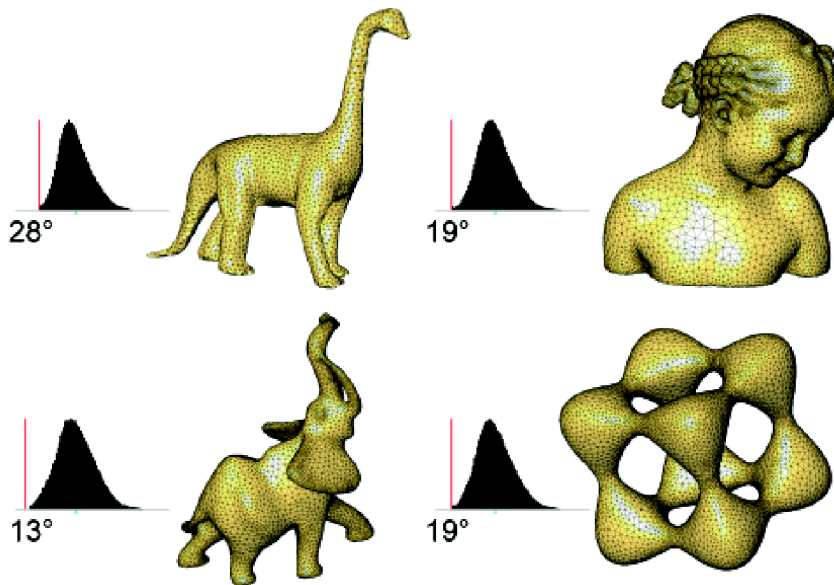


Figure 4.8: Delaunay meshes after the sliver perturbation algorithm proposed by Tournois *et al.* [105].

The key idea consists of performing a gradient ascent over the sliver circumsphere radius as well as a gradient descent over the sliver volume. However, in the cases where all vertices of a sliver are on the domain boundary, the perturbation can fail in removing a sliver as the boundary vertices are too constrained.

4.4 Isotropic Meshing

Most applications have specific requirements on the size and shape of elements in the mesh. Aim of the isotropic meshing is to locate vertices so that the resulting mesh consists of almost *regular tetrahedra* (\sim all faces are equilateral triangles). In addition, the element size is close to a predefined size constraint.

One of the existing methods to create the points in accordance with the size specifications contained, *creation of points along the edges* [42], will be discussed here.

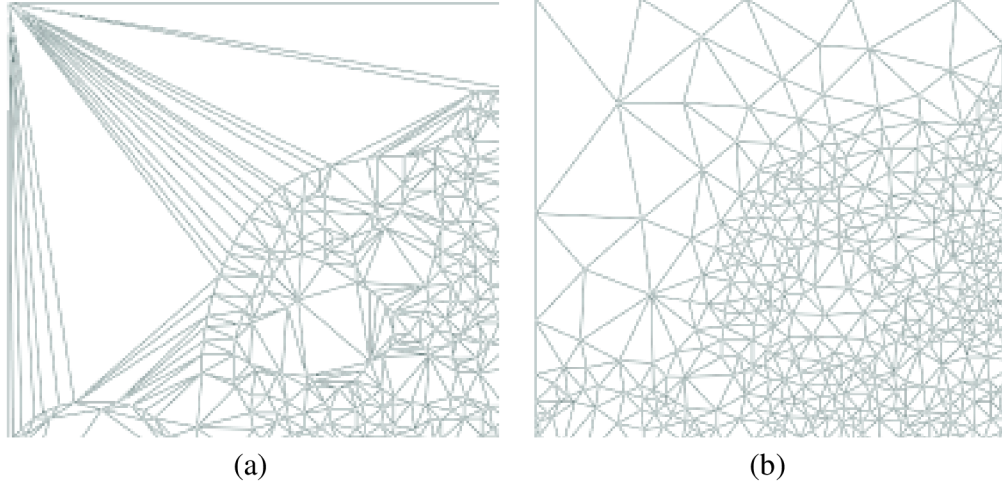


Figure 4.9: Triangular mesh constructed by the plain incremental method (a) and result of the isotropic meshing (b).

According to [42], control space $H(\Omega)$ (so called sizing field) is a function h_P defined at any point $P(x, y, z)$ of space. This function specifies the size of the elements in the mesh. The control space can be computed from the data, manually defined, or estimated with respect to the current mesh structure in an iterative process.

Let AB be an edge having endpoints A and B . Length of the edge in the control space metric can be calculated as follows:

$$l_H(AB) = \|AB\| \frac{\frac{1}{h(A)} + \frac{1}{h(B)}}{2}, \quad (4.2)$$

where $\|AB\|$ is the real distance between A and B . The size $h(P)$ is the desired length of all the edges originating from the point P defined by the control space.

An alternative definition 4.3 of the edge length exists. This modified definition has a positive influence on resulting meshes and provides better control of the control space gradation.

$$l_H(AB) = \|AB\| \frac{h_A - h_B}{h_A h_B \ln(\frac{h_A}{h_B})}, \quad (h_A > h_B) \quad (4.3)$$

The key idea of the algorithm is to create new points along existing edges in the triangulation and obtain nearly equilateral tetrahedra having edges of unit length in the control space (= length h in the real space).

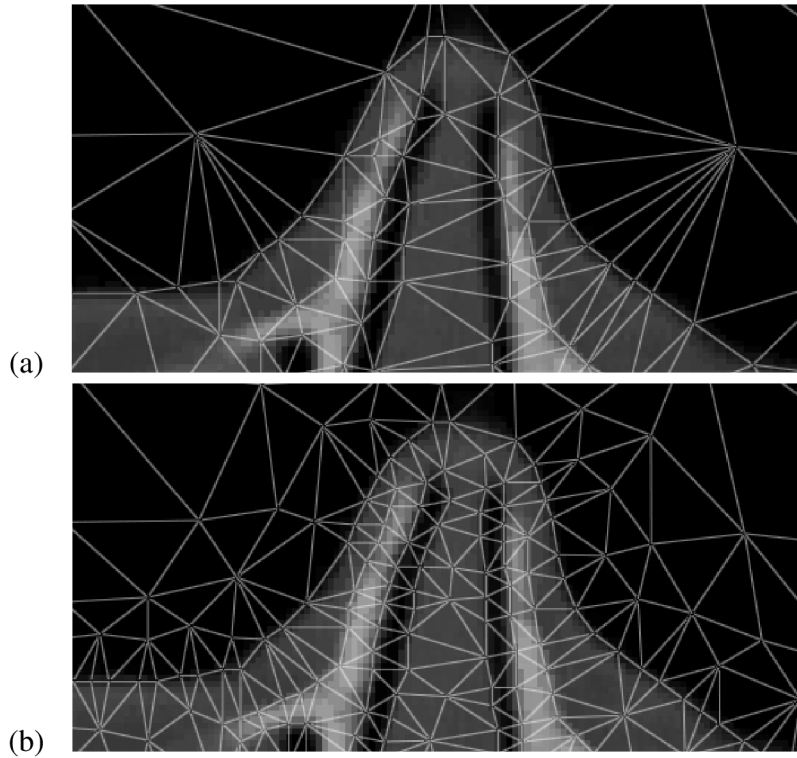


Figure 4.10: The original triangular mesh (a) and the mesh after edge splitting and subsequent optimization of poorly shaped triangles by adding points into the center of triangle's circumcircle (b).

Construction of the points along the edge AB. Let T be a threshold value < 1 , for instance 0.1. If $l_H(AB) < T$, the edge is not divided, otherwise a new point in the middle of the edge AB is introduced. Both obtained sub-edges are recursively tested and divided if necessary. Once we have a sequence of points $Q_0 \dots Q_n$ such that

$$l_H(Q_i, Q_{i+1}) < T \quad (4.4)$$

where $Q_0 = A$ and $Q_n = B$, the final set of points dividing the edge AB can be found.

Introducing a new point to the mesh. The smallest index i satisfying criterion 4.5 is found and the point Q_i is introduced to the mesh as new vertex. Iterating this process and comparing the sum to the increasing values $2, 3, \dots$ results in construction of several new points along the edge.

$$\sum_{j=0}^i l_H(Q_j, Q_{j+1}) > 1 \quad (4.5)$$

Applied to every edge in the current mesh, a large set of points is obtained. This set must be filtered to discard all points too close to any other before adding points to the mesh.

4.5 Variational Meshing

Many approaches based on energy minimization [1, 25, 32, 61] have been proposed as a powerful tool in meshing. In this thesis, a vector segmentation technique, built upon a *Variational Tetrahedral Meshing* (shortly VTM) approach [1], is presented. A simple minimization procedure alternates two steps:

- global 3D Delaunay triangulation optimizing connectivity,
- local vertex relocation,

to consistently and efficiently minimize a global energy (4.6) over the domain. It results in a robust meshing technique that generates high quality meshes in terms of radius ratios, as well as angles.

$$E_M = \frac{1}{4} \sum_i X_i^2 |\Omega_i| - \int_M X^2 dx \quad (4.6)$$

In this energy equation, $|\Omega_i|$ is the volume of the 1-ring neighborhood of vertex X_i , and the last term is constant for a given fixed mesh M .

As shown in [1], a derivation of the quadratic energy in X_i leads to a simple formula X_i^* representing optimal position of the interior vertex X_i in its 1-ring. In geometric terms, the formula can be expressed as:

$$X_i^* = \frac{1}{|\Omega_i|} \sum_{T_j \in \Omega_i} |T_j| c_j. \quad (4.7)$$

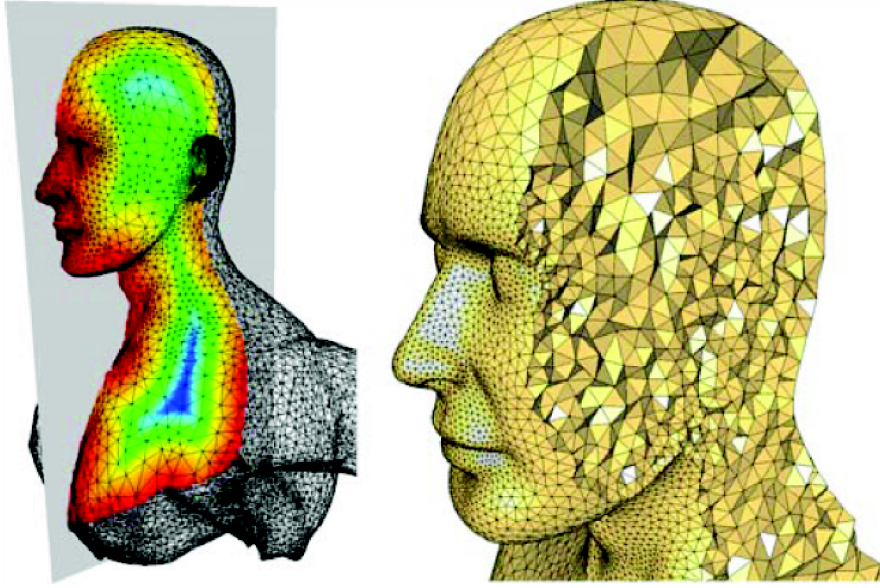


Figure 4.11: Variational Tetrahedral Meshing: Given the boundary of a domain, Alliez *et al.* compute the local feature size of this boundary as well as an interior sizing field (left), before constructing a mesh with a prescribed number of vertices and a smooth gradation conforming to the sizing field (right). The resulting tetrahedra are all well-shaped (i.e. nearly regular). The figure and the description were adopted from [1].

where c_j is the circumcenter of tetrahedra T_j . In other words, a vertex is moved at the *barycenter* of its neighboring circumshells.

4.5.1 Extension to Isotropic Meshes

The previous expressions apply to uniform meshing. To extend the equation to allow isotropic meshing, the sizing field H is introduced into the equation (4.7). A *mass density* in space can be defined and used in computation of the optimal vertex position. This density should agree with the sizing field. Alliez *et al.* [1] use a one-point approximation of the sizing field in a tetrahedron and defines the mass density as being $1/h^3$ since the local volume of a tetrahedra should be roughly the cube of the ideal edge size. Thus, the optimality condition is modified as follows:

$$X_i^* = \frac{1}{\sum_{T_k \in \Omega_i} \frac{|T_k|}{h^3(G_k)}} \sum_{T_j \in \Omega_i} \frac{|T_j|}{h^3(G_j)} c_j. \quad (4.8)$$

where G_k is the centroid of tetrahedron T_k . Finally, the variational tetrahedral meshing can be summarized by the iterative algorithm:

1. Generate initial vertices X_i .
2. Repeat the following alternating steps until convergence:
 - Construct Delaunay triangulation M .
 - Move vertices X_i to their optimal positions X_i^* .

4.5.2 Sizing Field

Alliez *et al.* presented a default sizing field robust for a large spectrum of mesh types. Definition of the sizing field is built on the notion of *local feature size* that corresponds to the combination of domain *boundary curvature* and *thickness* as well.

The local feature size $lfs(P)$ at a point P of domain boundary is defined as the distance $d(P, S_k(\Omega))$ to a *medial axis* $S_k(\Omega)$. The medial axis, or skeleton of the domain, is the locus of all centers of maximal balls inscribed in the boundary. Given the local feature size on the boundary, we need a controllable way to extrapolate this function to the interior. The function

$$h_P = \min_{S \in \delta\Omega} [Kd(S, P) + lfs(S)] \quad (4.9)$$

satisfies this criterion. The parameter K controls gradation of the resulting field, $K = 0$ being the uniform case.

Chapter 5

State of the Art in Anatomical Modeling

This thesis deals with the problem of 3D geometric modeling of human tissues. The most important phases of this complex task are medical image segmentation and subsequent surface reconstruction.

5.1 Medical Image Segmentation

In relation to the geometric modeling of human tissues, all medical image segmentation algorithms can be classified into two groups:

- techniques based on *raster segmentation* – a pixel value in the segmented image denotes label of an image region, or particular tissue type;
- and *vector-based segmentation* – region boundaries, and perhaps the internal structure, are represented as a set of vector graphic primitives (i.e. lines, curves, polygons, etc.) directly.

It is necessary to note that such classification is not very common, however, it makes a good sense in reference to the geometric modeling.

In this chapter, both categories are discussed in reference to the subsequent surface reconstruction. A short survey of *vector-based* segmentation techniques is given too. The presented methods form the basis of the most of the segmentation techniques nowadays.

5.1.1 Raster-based Image Segmentation

A lot of 2D/3D segmentation algorithms can be found in the literature (Fuzzy C-means clustering [91, 94], Hidden Markov Fields [74], Watershed transform [47, 48], neural networks [73], etc.).

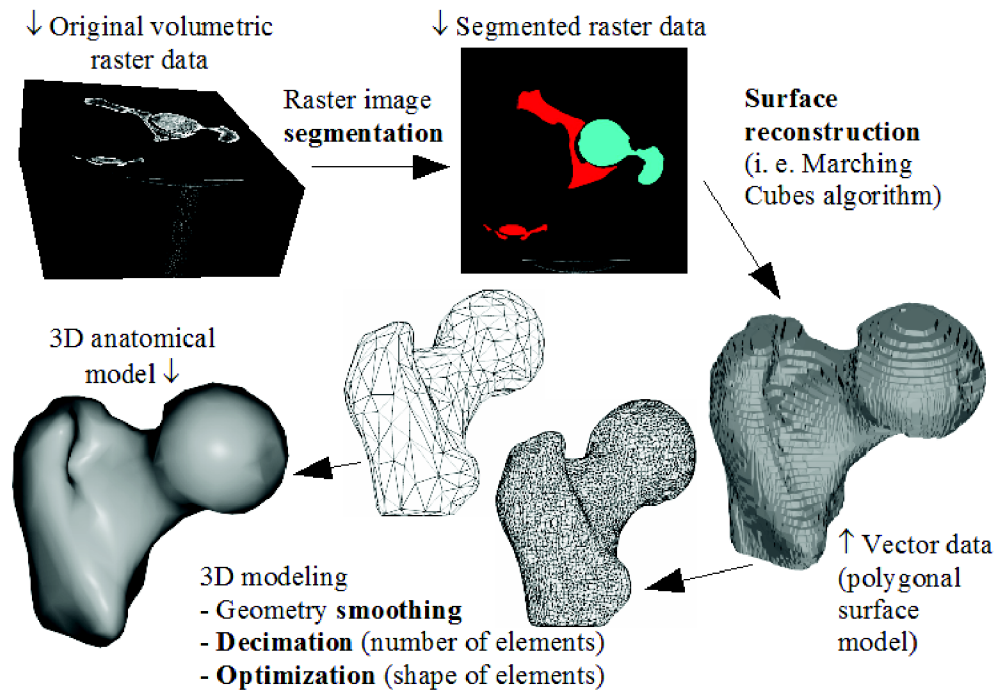


Figure 5.1: Scheme of the traditional way of the 3D anatomical modeling based on raster image segmentation.

Raster-based methods produce data which are not suitable for the geometric modeling (see Fig. 5.1) – each pixel value denotes label of an image region. Most often, an algorithm such as *Marching Cubes* [71] is applied to reconstruct surfaces from the raster data. Further, decimation and smoothing of the model are required and may not be elementary (Sec. 5.2). Applied smoothing and decimation methods may not shrink significant edges and corners and they must preserve volume of the original model.

5.1.2 Vector-based Image Segmentation

Vector-based segmentation techniques try to overcome previously described raster methods in efficiency and surface reconstruction simplicity. The most widely used vector segmen-

tation methods are based on deformable models [120, 124].

5.1.3 Deformable Contour Models

The deformable models, sometimes called *Active Contour Models* [76], are curves, surfaces, or solids defined within an image or volume domain, and they deform under the influence of external and internal forces derived from image characteristics. The internal forces regulate the ability of the contour to stretch at a specific point while preserving some degree of geometric smoothness. The external forces attract the contour to specific image features.

This type of active contour models is called *parametric models – Snakes* [28, 124]. There is also a second type of active contours – *the geometric models* [31, 69], best known is the *Level-Set* method.

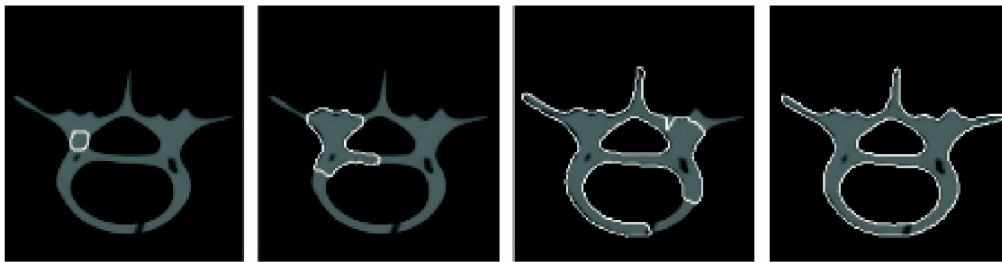


Figure 5.2: Segmentation of a cross sectional image of a human vertebra with a topologically adaptable snake. The snake begins as a single closed curve and becomes three closed curves. Images were published by McInerney and Terzopoulos in [76].

The deformable models are robust against noise and boundary gaps. These models are also capable of adjusting themselves to significant variability of human anatomy. Main disadvantage is that they require manual initialization and interaction during the segmentation. In more automatic methods, the initial model must usually be placed close to the region boundaries to guarantee good performance.

Extension of the deformable models to 3D space is not a trivial task. Numerous researchers have explored application of deformable surface models to volumetric medical images [10, 22, 65, 78]. A deformable surface model capable of segmenting complex internal organs such as the cortex of the brain has been proposed [75, 77]. The model is represented as a closed triangulated surface. This representation is more efficient, much less sensitive to initialization and spurious image features.

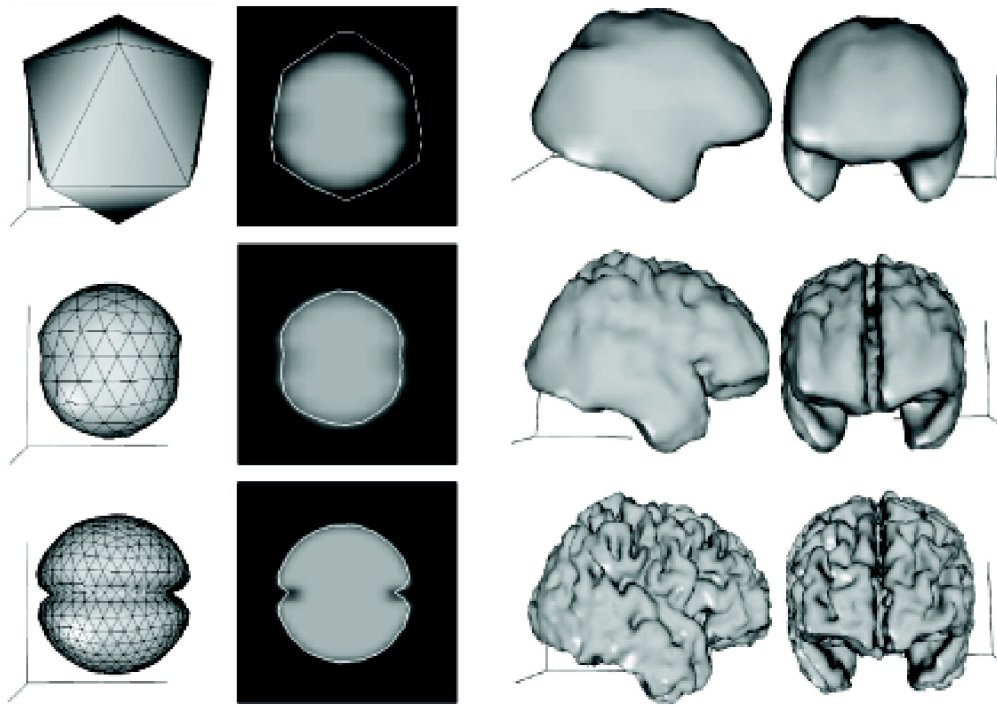


Figure 5.3: Artificial boundary surfaces extracted at different levels of volume image pyramid and brain cortical surfaces extracted from MR volume image pyramid [90].

5.1.4 Level-Sets

The level-set segmentation [31, 69] solves the energy based active contours minimization problem by the computation of minimal distance curves. In this approach, a curve is embedded as a *zero level set* of a higher dimensional surface, Fig. 5.4. The entire surface is evolved to minimize a metric defined by the curvature and image gradient.

Leventon *et al.* presented a level-set method that incorporates prior information about the intensity and curvature profile of the structure from a training set of images and boundaries [69]. The intensity distribution as a function of signed distance from the object boundary is modeled. A curvature profile acts as a boundary regularization term specific to the shape being extracted.

In general, level-set methods are used for highly convex shapes. These approaches achieve shape recognition requiring a little knowledge about the surface. In addition, initialization must be done close to the desired boundary, and it often requires user interaction for initial starting.

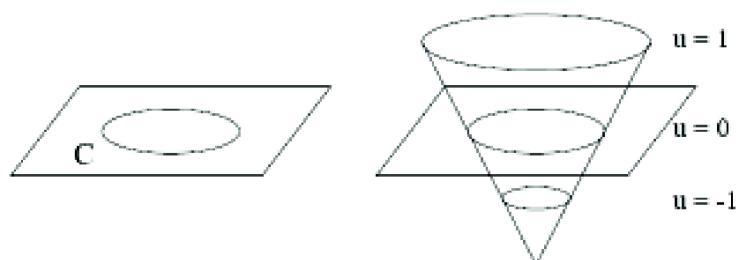


Figure 5.4: Level-sets of an embedding function u for a closed two-dimensional curve C . The curve is the zero level set of the higher dimensional surface u . The entire surface is evolved to minimize a metric defined by the curvature and image gradient [69].

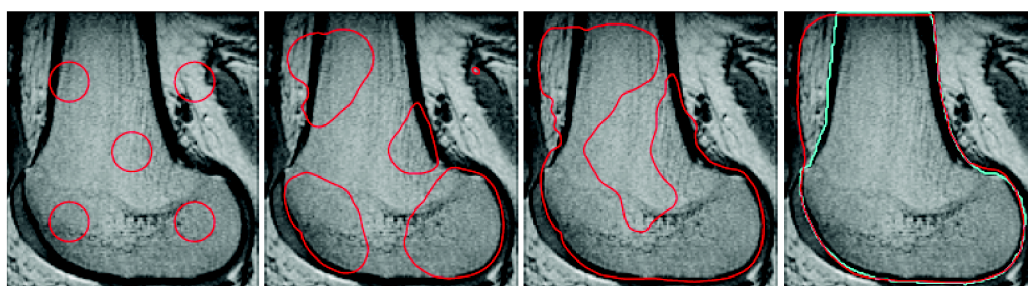


Figure 5.5: Initial, middle, and final steps in the 2D femur segmentation. The cyan curve in the last frame is manually segmented ground truth. Reprinted from [69].

5.1.5 Active Appearance Models

In recent years, the *Active Appearance Models* (AAMs) [23, 102] have achieved much success in medical applications. This knowledge-based method uses a prior model of what is expected in the image. It typically attempts to find the best match of the model to a new image. A statistical approach based on the *Principal Component Analysis* (PCA) [35] is used to build the model analyzing the appearance of a set of training samples, while the model parameters can be adjusted to fit unseen images and hence perform image registration.

The main drawback of AAMs, much like any knowledge-based method, is the anatomical variability. Accurate segmentation of complex structures is very difficult. Hence, these approaches are best suited for segmenting structures which are, in some way, stable over the population of study. Objects such as blood vessels are not suitable. Due to the design of the AAMs, occlusions may cause the model fitting to fail. Finally, AAMs are dependent on a good, mostly manual, initialization.

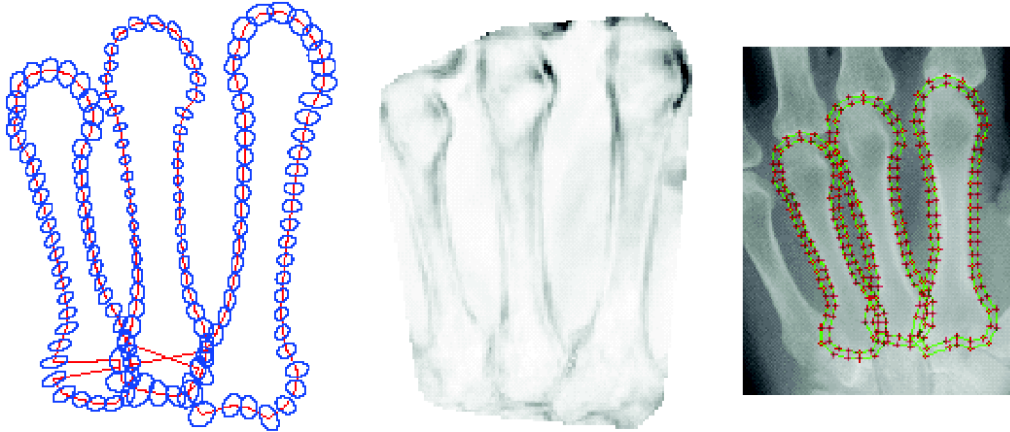


Figure 5.6: AAM of the metacarpals. From the left to right: independent analysis of each model point, texture variance and AAM optimized to fit the test image. Reprinted from [102].

5.1.6 3D AAM

Mitchell *et al.* [79] presented an extension of AAMs to 3D space for three-dimensional segmentation of cardiac MR and ultrasound images. In that paper, solution for several problems of the extension of AAMs to 3D space is given – point correspondence in 3D, model alignment, and 3D image warping.

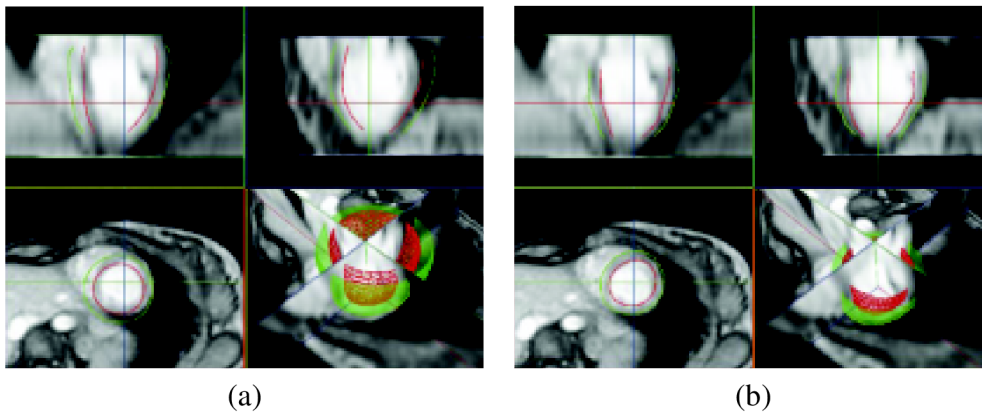


Figure 5.7: Three-dimensional AAM matching process. The initial position of the model (a); and the final match (b). Adopted from [79].

5.2 Surface Reconstruction

Many surface reconstruction algorithms able to create polygonal model of a desired region exist. All methods necessarily need to know which voxels form the region. Such definition of regions (i.e. separation of meaningful parts of an image) is the principal aim of the image segmentation.

5.2.1 Isosurfaces

For extracting boundary surfaces of three-dimensional regions directly from the given discrete volume data, the iso-surface algorithms [113] can be used. Isosurfaces are defined by connecting voxels with intensities equal to a given *iso-value* (intensity) in a 3D volume. Having the volume data, the isosurfaces may be extracted using an algorithm similar to the *Marching Cubes* [71].

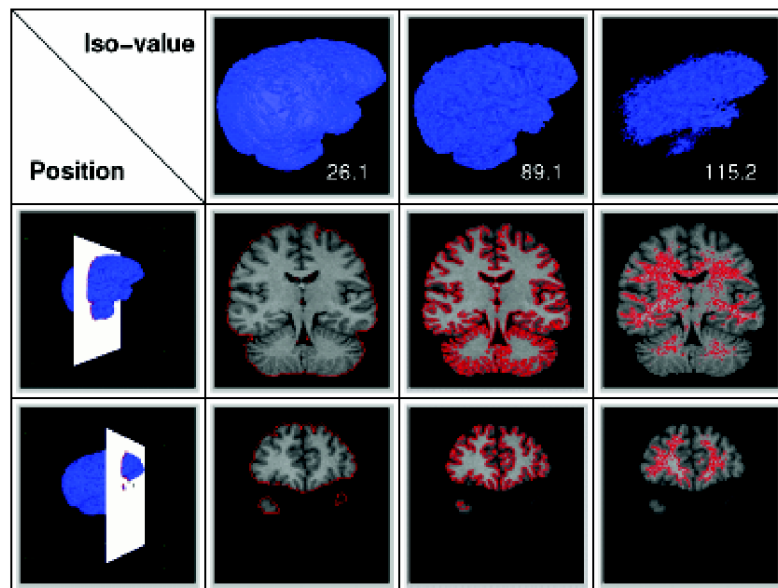


Figure 5.8: Isosurfaces extracted for three different isovalues [113].

5.2.2 Marching Cubes

Marching Cubes [71] (MC) algorithm creates a polygonal representation of predefined surfaces from a discrete volumetric data. It uses a divide-and-conquer approach to locate

the surface in a logical cube created from eight adjacent voxels. The algorithm determines how the surface intersects this cube, then moves (or marches) to the next cube.

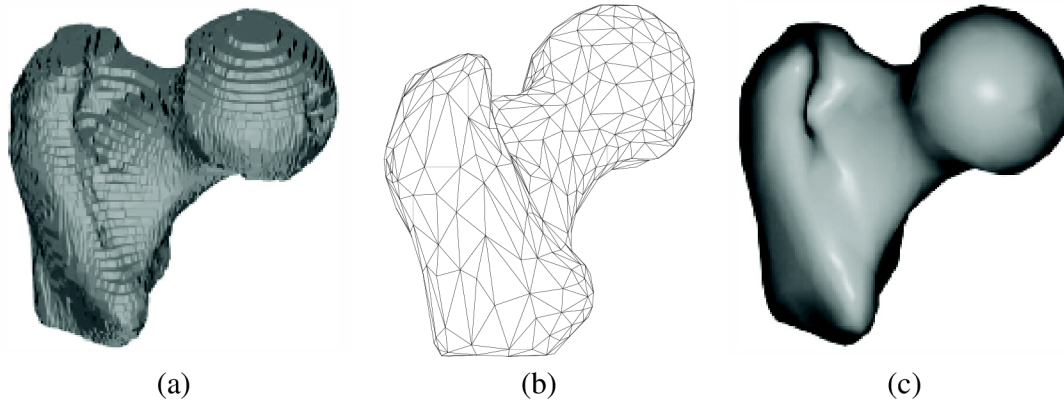


Figure 5.9: Final polygonal surface after the Marching Cubes (a); and the same model after smoothing and decimation (b,c) [59].

Unfortunately, a significant noise in the data causes that artifacts appear in the result. To deal with this problem a three-dimensional smoothing filter must be applied to the original data, or certain surface-smoothing algorithms were introduced.

The MC can not detect sharp features of the extracted isosurface [126]. The enhanced distance field representation and the extended MC algorithm [56] were introduced to extract feature sensitive isosurfaces from the volume data. The grid snapping method [80] reduces the number of elements in an approximated isocontour and also improves the aspect ratio of the elements.

5.2.3 Surface Smoothing

Different algorithms address the surface smoothing problem with varying success and there is no general algorithm, which works reliably in all cases [117].

One of the basic approaches to smoothing is the improved *laplacian operator* [114] which works by averaging position of the vertex with its neighbourhood. The main disadvantages of the original laplacian operator [41], shrinking of the volume, is reduced by moving the smoothed vertices back a bit. Another improvements were presented [103] which operate in alternating inward and outward diffusion of vertices in order to maintain the shape of the mesh.

The bilateral mesh denoising approach [38,67] has been quite successful in smoothing. It is essentially a bilateral filter applied on a mesh topology and works by filtering vertex positions in directions of their normals. By tuning parameters of the filter, the bilateral denoising is able to preserve sharp edges and corners in the mesh.

5.3 Unstructured Meshing

The three main families of algorithms for unstructured 3D mesh generation have been already described in Chap. 4:

- octree methods [126, 127],
- advancing front methods [54, 122],
- and Delunay-based methods [1, 14, 25, 61, 70].

Here, the recent work on surface approximation and optimal Delaunay triangulations is discussed in reference to medical applications and surface modeling.

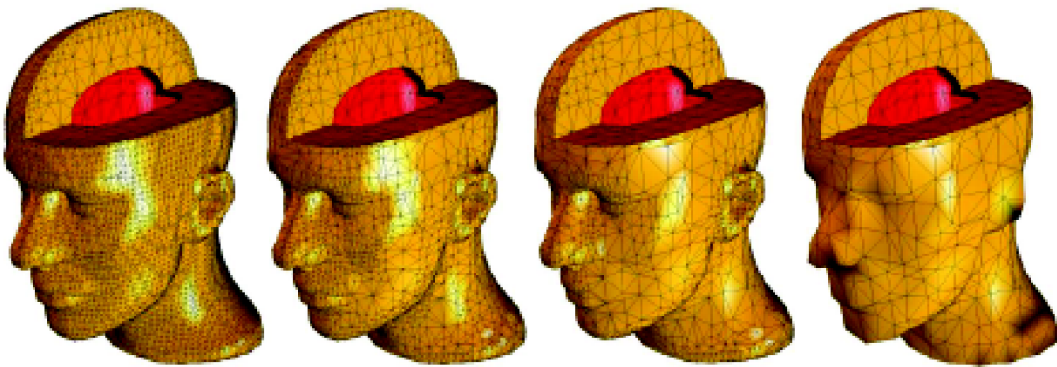


Figure 5.10: Direct surface extraction from volumetric data. Zhang *et al.* [127].

Zhang *et al.* [126] presented an algorithm to extract adaptive and quality 3D meshes directly from volumetric imaging data. In order to extract tetrahedral (or hexahedral) meshes, their approach combines bilateral and anisotropic diffusion filtering of the original data, with contour spectrum, iso-surface and interval volume selection. A top-down octree subdivision coupled with the dual contouring method is used to rapidly extract adaptive 3D

finite element meshes from volumetric imaging data (see Fig. 5.10). The main contributions of their approach is extension of the dual contouring method to crack-free interval volume tetrahedral/hexahedral meshing with feature sensitive adaptation [125].

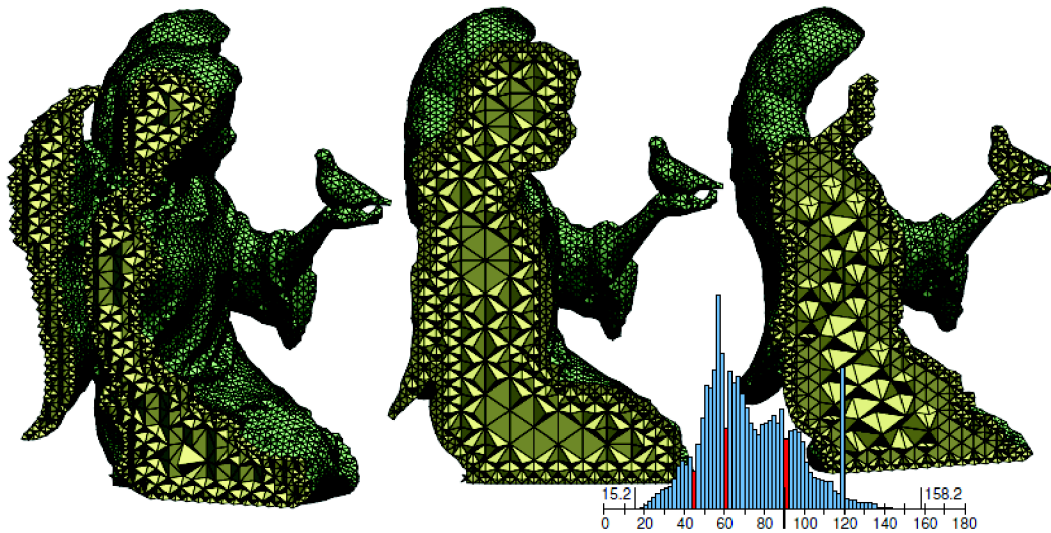


Figure 5.11: A tetrahedron mesh produced by isosurface stuffing. At the lower right is a histogram of tetrahedron dihedral angles. The extreme dihedral angles are 15.2 and 158.2. Labelle *et al.* [64].

In 2007, the *isosurface stuffing* algorithm [64] was presented that fills an iso-surface with a uniformly sized tetrahedral mesh whose dihedral angles are bounded. The algorithm is fast, numerically robust, and easy to implement because, like the Marching Cubes, it generates tetrahedra from a small set of pre-computed stencils. A variant of the algorithm creates a mesh with internal grading: on the boundary, where high resolution is generally desired, the elements are fine and uniformly sized, and in the interior they may be coarser and vary in size. However, the algorithm does not permit grading of both surface and interior tetrahedra and has a strong bound on the dihedral angles.

Variational approaches relying on energy minimization have been presented as a powerful and robust tool in meshing. These methods basically define energies that they minimize through vertex displacements and/or connectivity changes in the current mesh.

Du and Wang [32] propose to generate meshes that are dual to optimal Voronoi diagrams. The centroidal Voronoi tessellation [34] based Delaunay triangulation provides an optimal distribution of generating points with respect to a given density function and generates a high-quality mesh. By establishing an appropriate relationship between the

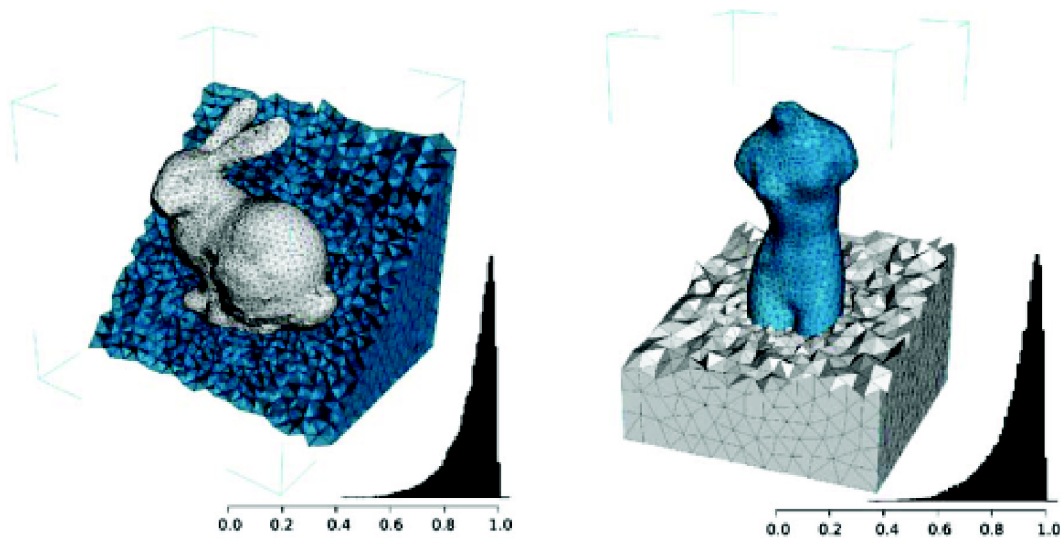


Figure 5.12: Results obtained by the approach of Dardenne *et al.* from artificial discrete data – rasterized input surface models. Histograms show the distributions of tetrahedron quality in each mesh [25].

density function and the specified sizing field and applying the Lloyds iteration, the constrained mesh is obtained as a natural global optimization of the initial mesh. Simple local operations such as edges–faces flipping are also used to further improve the mesh.

Following Du and Wang, another tetrahedral mesh generation algorithm based on centroidal Voronoi tessellation, which takes volumetric segmented data as an input, has been presented [25]. The algorithm performs clustering of the original voxels. A vertex replaces each cluster and the set of created vertices is triangulated in order to obtain a tetrahedral mesh, taking into account both the accuracy of the representation and the elements quality.

The medial axis of the original shape is used to generate a vertex density function in order to mesh more densely certain complex regions of the domain. The resulting meshes exhibit good element’s quality with respect to minimal dihedral angle.

Alliez *et al.* [1] presented a new variational tetrahedral meshing technique that use a simple quadratic energy and allow for global changes in mesh connectivity during energy minimization.

This meshing algorithm allows to create graded meshes, and defines a *sizing field* prescribing the desired tetrahedra sizes within the domain. A fast marching construction of the sizing field is proposed based on the notion of *local feature size* which corresponds to the combination of surface curvature and domain thickness. The sizing field estimation

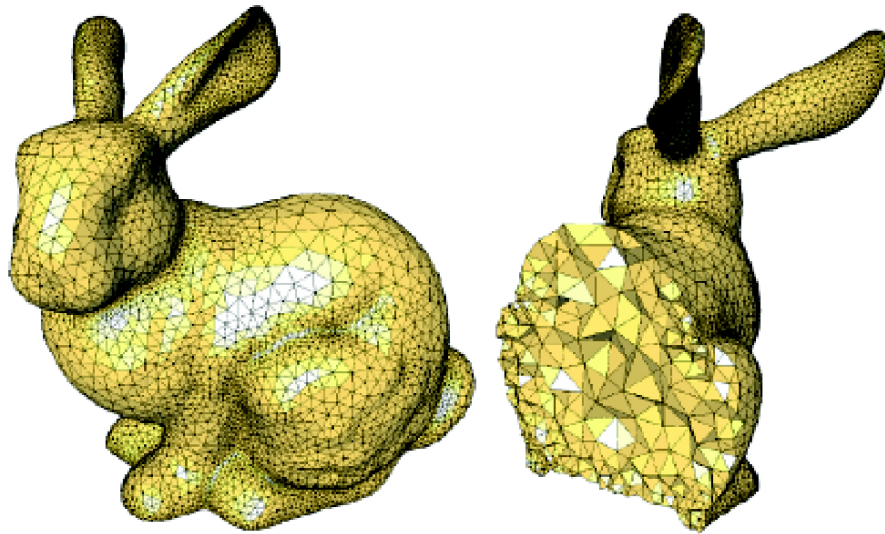


Figure 5.13: Results by Alliez *et al.* [1] meshing the interior of the Stanford bunny. The cutaway views show the well shaped elements inside the domain.

starts from discrete skeleton (or medial axis) of the domain.

This technique produces nicely shaped tetrahedra throughout the domain, however, slivers (i.e. degenerate elements) could appear near the domain boundary, as the boundary vertices are unaffected by the 3D optimization [105].

Chapter 6

Delaunay-based Vector Segmentation

Numerous researchers use triangulations and meshes as a spatial support in their scientific computations (e.g. solid and fluid mechanics). In the field of biomechanical simulation, reliable applications in dental surgery, for example the interaction of the human mandible with dental implants [36], can be found.

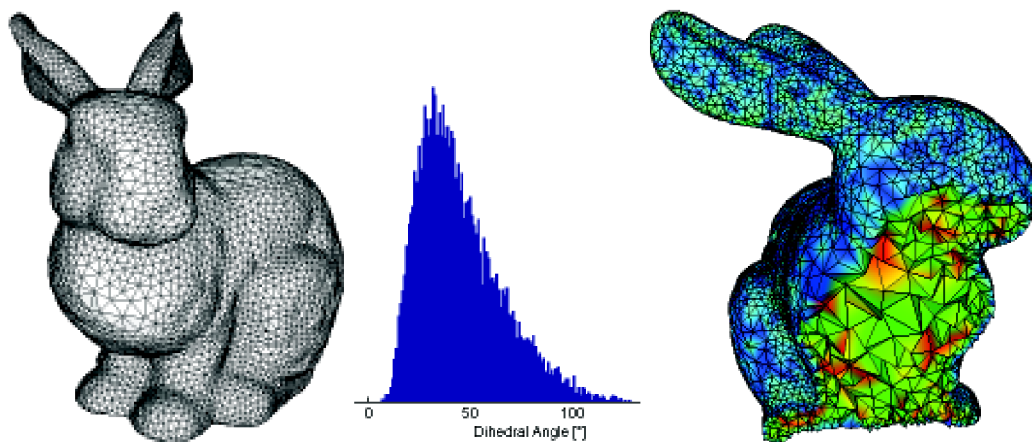


Figure 6.1: Results of the proposed vector segmentation method: surface extracted directly from the tetrahedral mesh (a); histogram of minimal dihedral angles on the surface (b); cut through the tetrahedral mesh (c).

In this thesis, a novel vector segmentation technique based on the *3D Delaunay Triangulation* is proposed. Tetrahedral mesh is used to partition volumetric image data into regions whose characteristics, such as intensity and texture, are similar. Process of the

mesh construction respects significant image edges. Therefore, surfaces of image regions are well approximated by the tessellation grid of the mesh and can be easily derived.

6.1 Delaunay Triangulation for Image Segmentation

A particular problem of the segmentation is image partitioning into a set of non-overlapping regions r_1, \dots, r_n so that the variation of some property (such as mean pixel value, variance, etc.) within each region r_k is either constant, or follows a simple model.

It has been shown [26, 44, 96] that Delaunay triangulation can be used to effectively partition an image and simultaneously, the tessellation grid of the Delaunay triangulation can be adapted to the structure of the image by combining region and edge information (see Fig. 6.2).

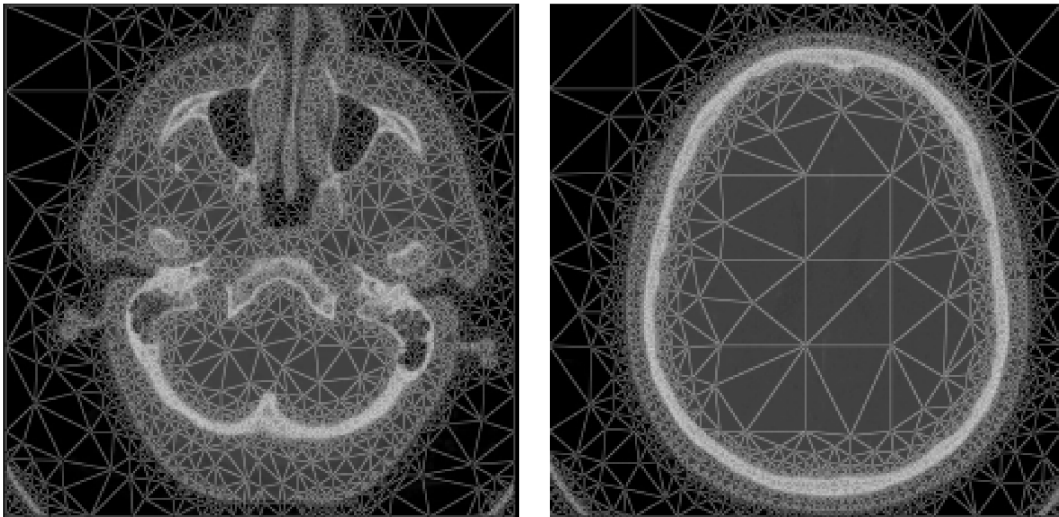


Figure 6.2: Tessellation grid of the 2D Delaunay triangulation adapted to the underlying image structure [115].

Constructing the DT, the image is divided into a number of non-overlapping vector primitives t_1, t_2, \dots, t_n – triangles in 2D space and tetrahedra in 3D. These primitives are not segments of the image by itself, but they belong to image regions r_k . Each region r_k is composed of a number of such elements.

This relationship can be expressed by a *region membership function*. *Hard assignment* means that this function assigns exactly one region to each tetrahedron. In practice,

membership function of the form

$$m(t_i, r_k) = p(r_k|t_i) \quad (6.1)$$

making a *soft assignment* of tetrahedra overcomes the hard one and leads to better results. The soft membership function is usually a *likelihood function* $p(r_k|t_i)$ assigning each tetrahedron into every image region with some certainty. The value is higher as the similarity of the tetrahedron and the region increases.

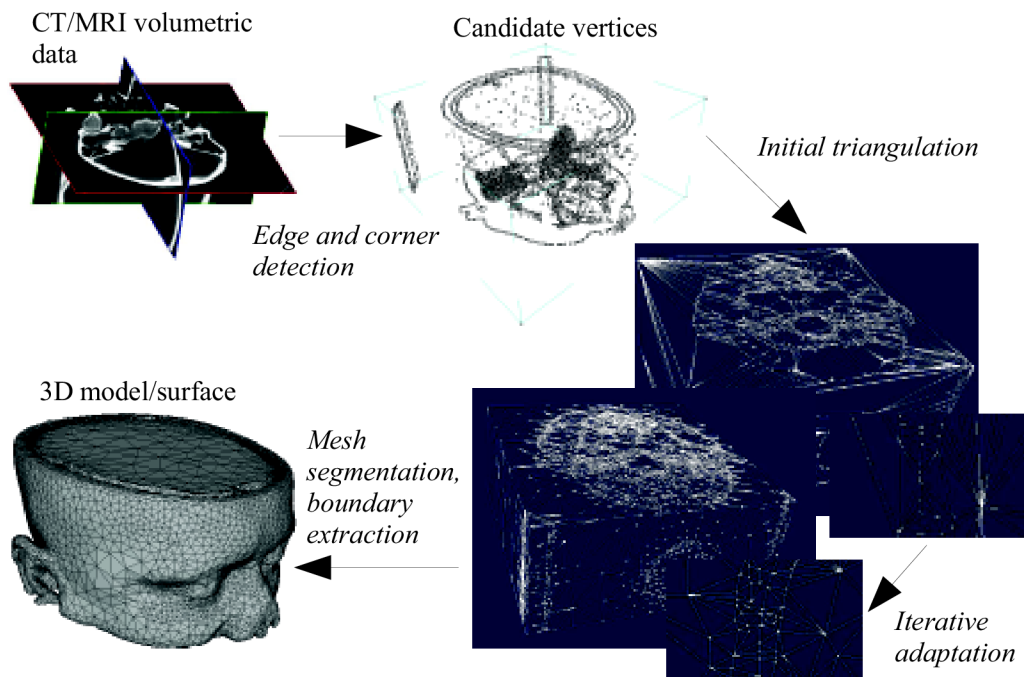


Figure 6.3: Basic scheme of the Delaunay-based vector segmentation.

6.2 Delaunay-based Vector Segmentation

Based on the introduced principles, the adaptive Delaunay-based vector segmentation is proposed as follows:

1. **Data preprocessing** – Noise reduction by means of the 3D anisotropic (or bilateral) filtering (Sec. 3.3.4).

2. **3D edge and corner detection** – Candidate vertices lying on region boundaries, meaningful edges and corners are located.
3. **Initial Delaunay triangulation** – Tetrahedral mesh is constructed from the pre-selected set of candidate vertices.
4. **Iterative adaptation** – The triangulation is adapted to the underlying image structure by means of isotropic edge splitting (introduces new vertices to the mesh), and variational meshing (optimizes vertex positions so that quality of tetrahedra increases).
5. **Mesh segmentation** – Final classification of tetrahedra into image regions according to results of some data clustering method.

The idea of the vector segmentation is also illustrated in Fig. 6.3, while details of all individual phases of the segmentation are discussed in next sections.

6.3 Data Preprocessing

An important part of the vector segmentation is adaptation of the tetrahedral mesh to an image structure which is derived from the found image edges. The adaptation process is strongly affected by quality of the edge detection. Therefore, it is highly recommended to filter the input data in order to deal with noise in the data.

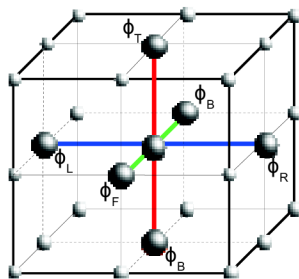


Figure 6.4: Anisotropic filter – neighboring nodes used to calculate the flow ϕ between voxels.

Anisotropic (or *bilateral*) filtering (Sec. 3.3.4) performs piecewise smoothing of the image, and its strength lies in the fact that object contours and boundaries are not only preserved, but even improved.

In the preprocessing step of the vector segmentation, a 3D anisotropic filtering is applied to the volumetric data. Gerig *et al.* [43] described such extension of the anisotropic filter into 3D space. The 1D discrete formulation (Eq. 3.4) of the filter can be easily reformulated as:

$$I(t + \Delta t) = I(t) + \Delta t \times (\phi_t - \phi_d + \phi_l - \phi_r + \phi_f - \phi_b) \quad (6.2)$$

The 3D discrete formulation results in simple, local operations over the image. In the first step, the flow ϕ is calculated between neighboring nodes (Fig. 6.4). In the second step, the node intensities are updated by the local sum of the flow contributions. The flow may be also calculated between diagonally neighboring voxels, resulting in better smoothing results. In that case, the integration constant Δt must be adjusted.

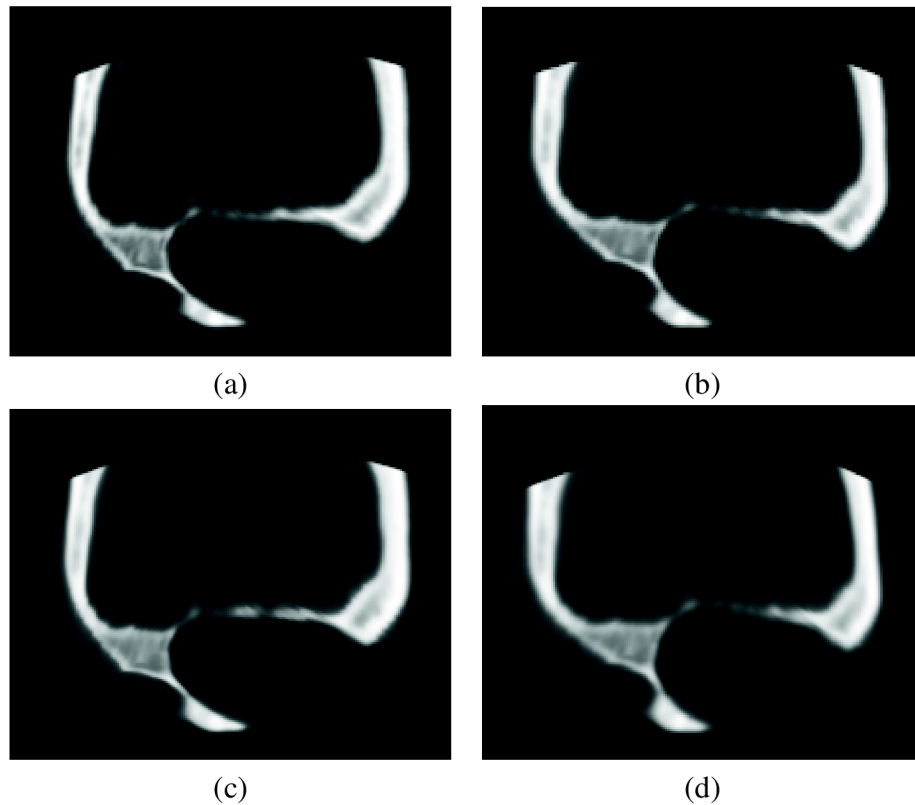


Figure 6.5: Result of the 3D anisotropic filtering of CT data: original slices (a); result for the value of $\kappa = 100$ (b); $\kappa = 200$ (c); and $\kappa = 400$ (c).

6.4 3D Edge and Corner Detection

The triangulation starts from a set of candidate vertices distributed over the entire image. These candidates can be found by various, more or less sophisticated, image edge detection algorithms [5, 13, 49, 89, 93, 100] extended to 3D space.

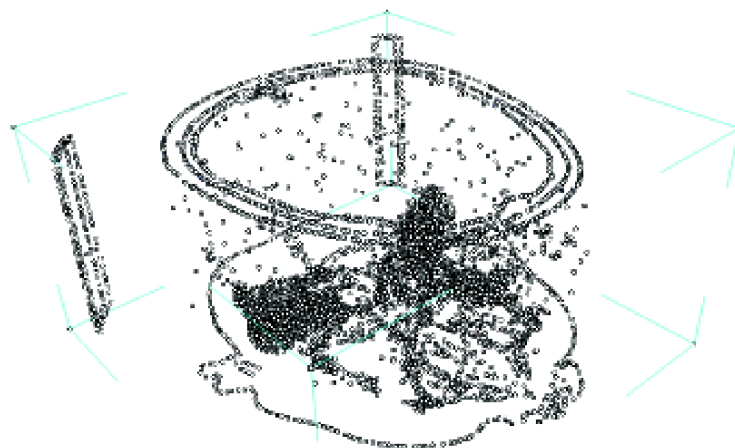


Figure 6.6: Sampled initial set of vertices found by the edge and corner detection.

Because of the complex nature of medical image data (Sec. 3.6), detection of meaningful edges that form boundary of desired tissues may be very problematic. Character and strength of edges differ between tissues. Moreover, extremely thin and weak edges may be present in the image data. Such edges must be also detected to approximate surface of tissues properly. In practice, this leads to highly sensitive setting of the edge detector that, unfortunately, results in many false detections of "noisy", less meaningful edges. In this thesis, a simple *tissue-selective edge detection* approach is proposed to partially reduce this undesired effect.

6.4.1 Tissue-Selective Edge Detection

The tissue-selective edge detection means that the edge detection is divided into separate steps (or parallel stages) per concrete tissue type. Before the detection starts, the image data are pre-processed using the *power-law contrast enhancement* technique (see Sec. 3.3.2) to increase contrast of the desired tissue against all others. Then edges of the highlighted tissue are detected. In the end, all found edges from all different tissues are merged together into a single image (see Fig. 6.7).

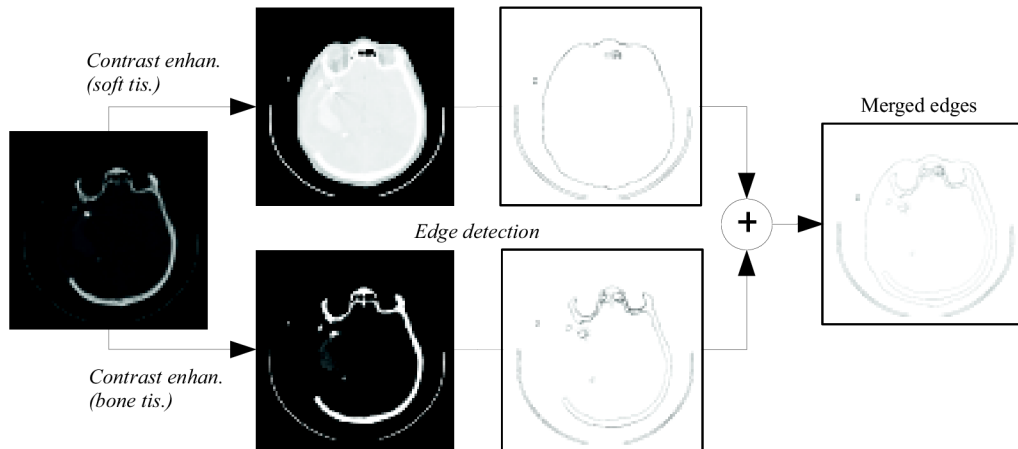


Figure 6.7: Scheme of the tissue-selective edge detection.

In our experiments, the well known *Canny edge detector* has been used in each step. Basic principle of the Canny detector remains unchanged in 3D space. The original image is filtered by the 3D Gaussian convolution filter. Afterwards, *3D Sobel operators* are applied to estimate image derivatives in x , y and z directions (Fig. 6.8).

$$S_x = \begin{bmatrix} -1 & 0 & 1 & & \\ -2 & -2 & 0 & 2 & \\ -1 & -4 & -1 & 0 & 1 \\ -2 & -2 & 0 & 2 & \\ -1 & 0 & 1 & & \end{bmatrix} \quad S_z = \begin{bmatrix} -1 & -2 & -1 & & \\ -2 & 0 & 0 & 0 & \\ -1 & 0 & 1 & 2 & 1 \\ 0 & 2 & 4 & 2 & \\ 1 & 2 & 1 & & \end{bmatrix}$$

Figure 6.8: Convolution kernels of the 3D Sobel operator in x and z axis.

Finally, edges are detected and refined by the help of image post-processing techniques like the *non-maxima suppression* and the *hysteresis* [13].

6.4.2 3D SUSAN Corner Detector

In order to respect significant features in the volumetric data during the meshing, we have modified the *Susan corner detector* [100] extending its functionality into 3D space. The Susan (*Smallest Univalve Segment Assimilating Nucleus*) detector was originally developed to locate feature points in 2D images.

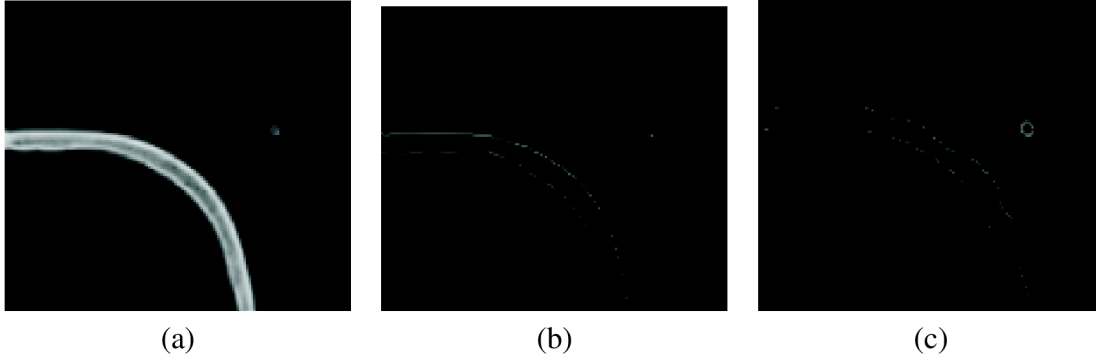


Figure 6.9: Positive effect of the tissue-selective edge detection: the original image (a); edges found by a single step detection (b); result of the tissue-selective edge detection (c).

Analogous to *Smith and Brady*, the modified 3D SUSAN places a spherical mask R over the voxel to be tested (the nucleus). The voxel in this mask is represented by $v \in R$. The nucleus is at v_0 . Every voxel is compared to the nucleus using the distance function:

$$c_v = \exp\left(-\left(\frac{I(v) - I(v_0)}{w}\right)^6\right), \quad (6.3)$$

where w is the brightness difference threshold. This function has the appearance of a smoothed rectangular function. The w parameter does affect the number of corners reported because it determines the allowed variation in brightness within the mask.

Response of the SUSAN detector [100] is defined as

$$SUSAN(R) = \begin{cases} T - n(R) & \text{if } n(R) < T \\ 0 & \text{otherwise,} \end{cases} \quad (6.4)$$

where $T \in \langle 0, 1 \rangle$ is geometric threshold, and $n(R)$ is area of the SUSAN given by:

$$n(R) = \frac{1}{N} \sum_{v_i \in R} c_{v_i}. \quad (6.5)$$

In Eq. 6.5, N is the number of voxels within a spherical mask R used as a normalization factor. If c_v is the rectangular function, then the previously defined area represents the number of voxels in the mask having brightness similar to the nucleus. This portion of the mask is called the *USAN* [100].

For successful corner detection, two further steps must be done. In a first step, the centroid of the USAN is found. A proper corner will have the centroid far from the nucleus.

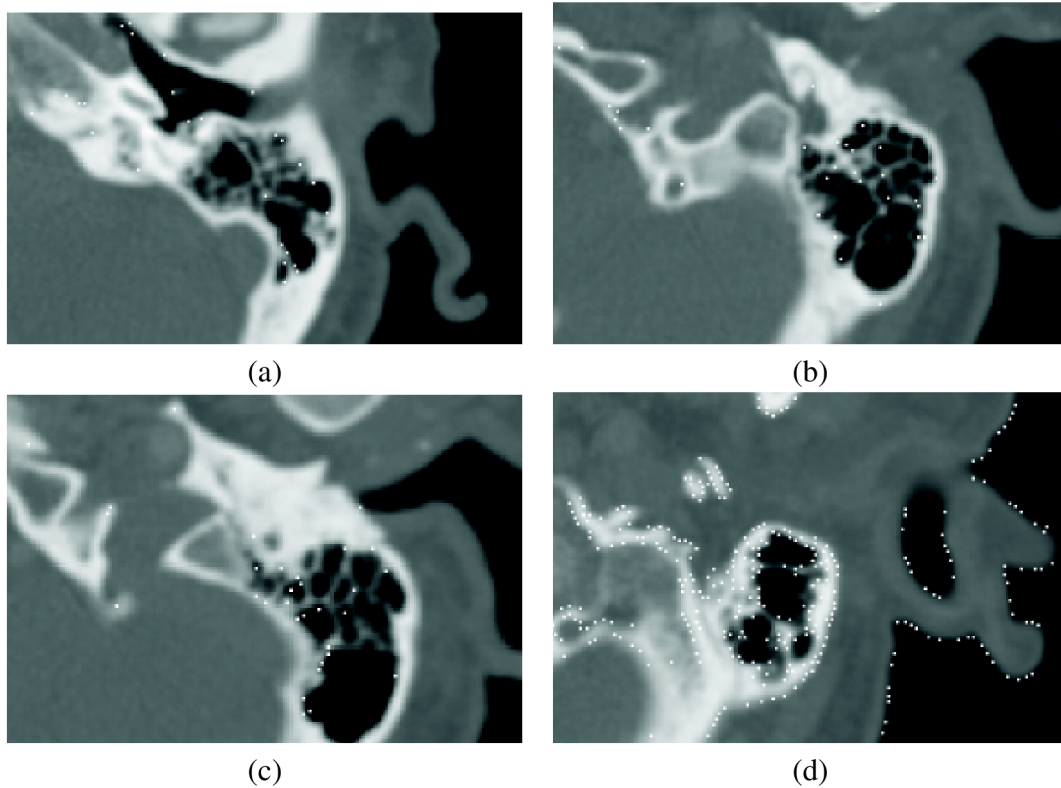


Figure 6.10: Results of the 3D SUSAN detector – corner points found in CT data.

The second step insists that all points on the line from the nucleus through the centroid out to the edge of the mask are in the USAN.

6.5 Initial Delaunay Triangulation

To construct the image partition, the edge points are sampled and together with all corner points ordered by their significance. While the initial DT is being constructed by the common *Incremental Method* (Sec.4.1.1), vertices located on strong edges are taken first (see Fig. 6.11).

6.6 Iterative Adaptation

Fundamental phase of the proposed segmentation method is adaptation of the tessellation mesh to cover the underlying image structure representing the anatomy of human tissues.

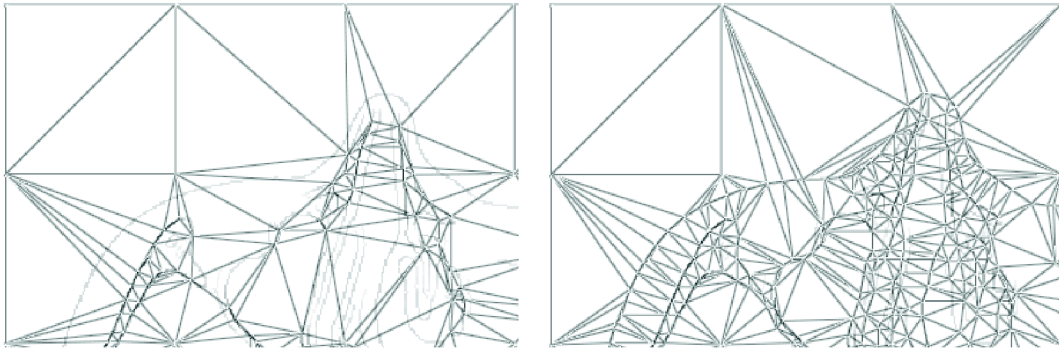


Figure 6.11: Incremental construction of the Delaunay triangulation (the 2D case). The number of vertices increases from left to right.

The following four main steps are repeated until the triangulation satisfies some convergence criterion (or just several times):

1. **Isotropic edge splitting** - creation of points along existing edges introduces new points to the mesh,
2. **Variational meshing** - optimization of the tessellation grid by means of vertex moving,
3. **Boundary refinement** - creation of new vertices along image edges to guarantee that all edges are well approximated by the tessellation grid.

During the iterative adaptation, only new vertices are gradually introduced to the mesh. The idea is to grow the mesh (in the sense of number of vertices) until a predefined limit is reached. An advantage of such progressive concept is that computational expensive operations like vertex removal and local re-meshing of the cavity are not necessary.

Before a new vertex is inserted to the mesh, several constraints are checked – minimal length of edges that will arise (L_{min}), minimal dihedral angle inside newly created tetrahedra (α_{min}), etc. In practice, these constraints guarantee that chosen parameters like minimal edge length will be satisfied in the final mesh. Moreover, it prevents corruption of the mesh and failures caused by a limited precision of math operations.

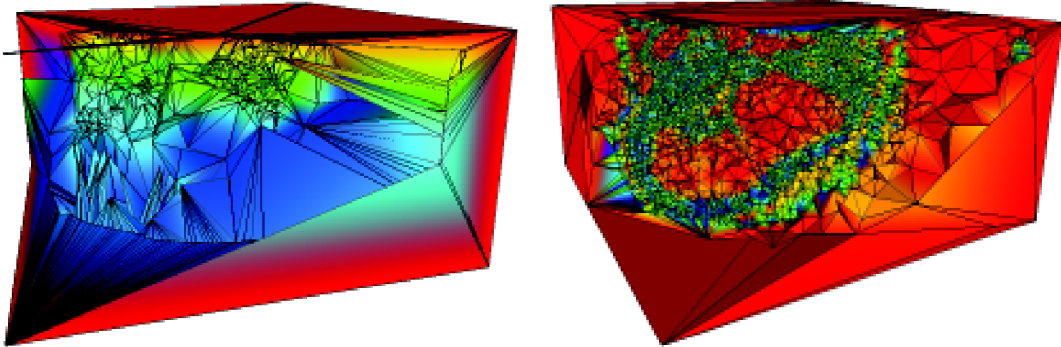


Figure 6.12: The initial Delaunay triangulation (a) and the final mesh after the iterative adaptation (b). Parameters of the sizing field were chosen to be $K = 0.8$ and $T_{avg} = 70mm$.

6.6.1 Isotropic Edge Splitting

In this phase, the isotropic meshing algorithm *creating new points along existing edges* and another well known technique of tetrahedral mesh optimization, *splitting of maximal/longest edge* [42], are combined together.

Instead of maximal edges, those edges crossing significant image edges are divided. A new vertex is inserted to the mesh at the point of intersection of both edges. This approach is partially similar to the constrained Delaunay triangulation. Unlike the previously described constraint partitioning method, the set of constraints is implicitly defined by all detected image edges in this case. The whole isotropic edge splitting process can be briefly formulated as follows:

1. Prepare the control space H .
2. Sequentially process every edge AB in the current triangulation T^i :
 - Find all intersection points P_i of the edge and image edges.
 - Introduce the sub-edges $AP_1, P_1P_2, \dots, P_nB$ in the triangulation.
 - Divide all sub-edges in the sense of isotropic meshing algorithm (Sec. 4.4).
3. Filter the set of newly created points to discard vertices too close to any other point respecting the control space metric.
4. Insert points to the mesh $T^i \rightarrow T^{i+1}$.

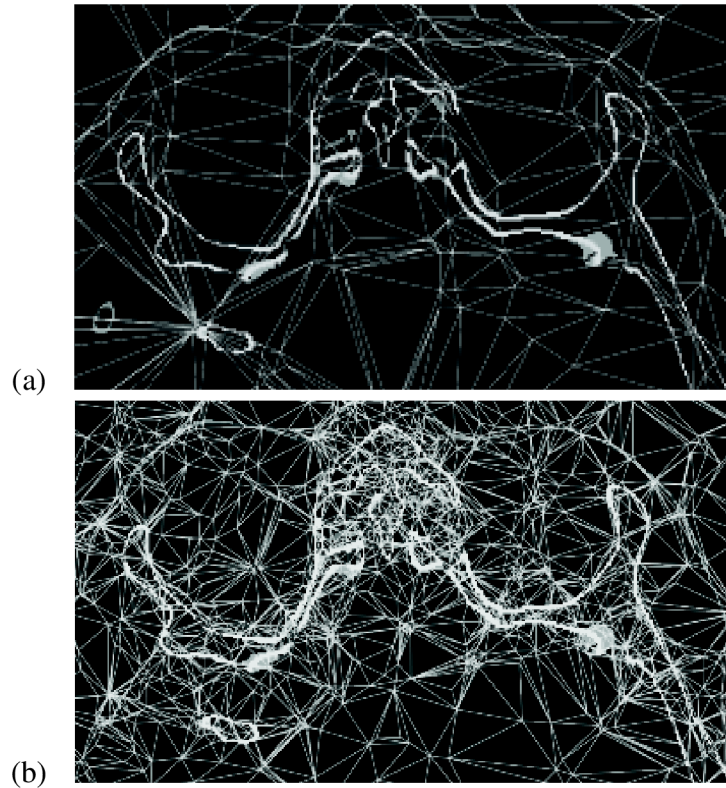


Figure 6.13: The effect of the isotropic edge splitting: cut through the initial tetrahedral mesh before the splitting (a) the same mesh after one iteration of the edge splitting algorithm (b).

5. Continue from the step 2 until convergence.

To prevent degradation and over-partitioning of the mesh, the angle between the tetrahedron e_t edge and the image edge e_i is computed. The splitting operation is performed only if the angle is greater than a given threshold $T \in (0, 1)$:

$$n_i = (d_x, d_y, d_z) \quad (6.6)$$

$$\frac{|e_t \cdot n_i|}{\|e_t\| \cdot \|n_i\|} > T \quad (6.7)$$

Edges that are almost parallel with an image edge remain unchanged. Normal n_i of the image edge (i.e. derivatives d_x, d_y, d_z) can be found applying various local differential operators, e.g. the mentioned *3D Sobel operators*.

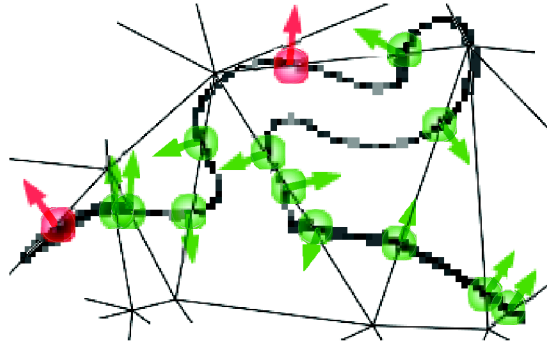


Figure 6.14: New vertices introduced by the edge splitting. Green crossings denote newly created vertices, while red ones represent vertices rejected due to the angle between image edge normal and the examined edge – both edges are almost parallel.

The splitting phase is similar to the one described in Sec. 4.4. The only difference is in the definition of the sizing field, so called *control space*. The control space prescribes length of edges in the mesh. In our case, the control space enforces creation of larger tetrahedra inside image regions and smaller ones along region boundaries (image edges). Apparently, definition of the sizing field strongly affects quality of the final mesh.

In this sense, the control space $H(\Omega)$ can be defined in the same way as the sizing field given by Eq. 4.9. This definition is robust and produces high quality meshes.

6.6.2 Preparing Control Space

Instead of the conventional domain boundary, we define the control space to respect found image edges. Thus, we generate the control space differently:

1. Estimate *distance transform* from all detected image edges first.
2. Find local maxima of the distance transform in order to identify medial axis.
3. Evaluate local feature size $lfs(P)$ on image edges using inverse distance transform propagating value from the medial axis.
4. Generate control space distributing $lfs(P)$ from edges using the formula (4.9).

This sizing field is relative. It describes the inhomogeneity of the required edge length. The real edge length is proportional to this relative value, and depending on the prescribed

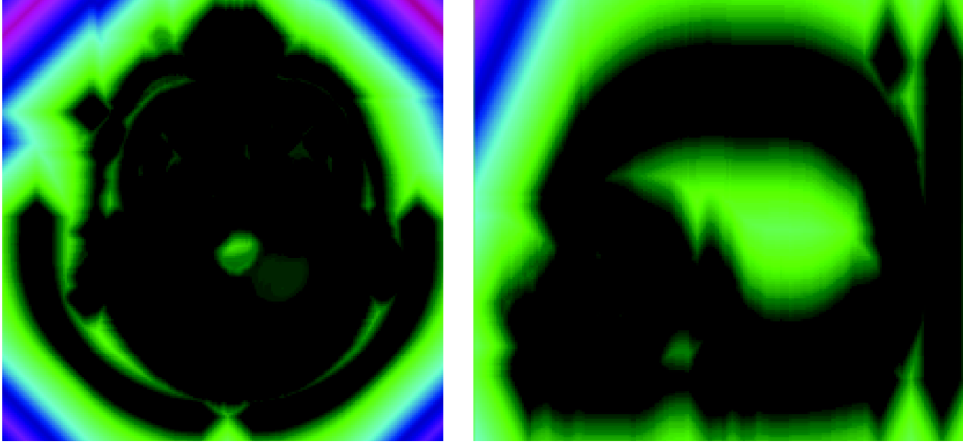


Figure 6.15: Pseudo-colored slices through the three-dimensional control space prescribing size of tetrahedra inside the mesh. Dark values stand for small tetrahedra close to image edges.

number of vertices. Such relative sizing is satisfactory for variational meshing, but it must be normalized for the isotropic edge splitting algorithm we use for introducing new vertices into the mesh. The normalization is simply given by:

$$h'_P = h_P \frac{T_{avg}}{\frac{1}{N} \sum_{v \in \Omega} h_v}, \quad (6.8)$$

where T_{avg} is the desired average tetrahedron edge size. If the point P lies exactly on an image edge, the control space value may be very small. Therefore, the *minimal edge length* L_{min} must be also specified in practice.

6.6.3 Variational Meshing

The variational meshing phase, alternating connectivity and geometry optimization, is an important part of the algorithm. The mesh energy is minimized by moving each *interior* vertex to its optimal position within its 1-ring neighborhood (Fig. 6.21). Further, the energy is minimized by computing the 3D Delaunay triangulation of these new sites optimizing the connectivity of vertices.

All boundary vertices are treated differently. In order to identify the current boundary vertices, each voxel V_i lying on an image edge is examined. Its nearest vertex S_j in the mesh is located, and the distance $d(V_i, S_j)$ as well as the coordinates of V_i (multiplied by

the distance d) are accumulated at that vertex. To deal with corner points, the distance d is weighted according to the point type. Corner points have the weight significantly greater than edge points, thus the closest vertex is attracted directly in place of the image corner.

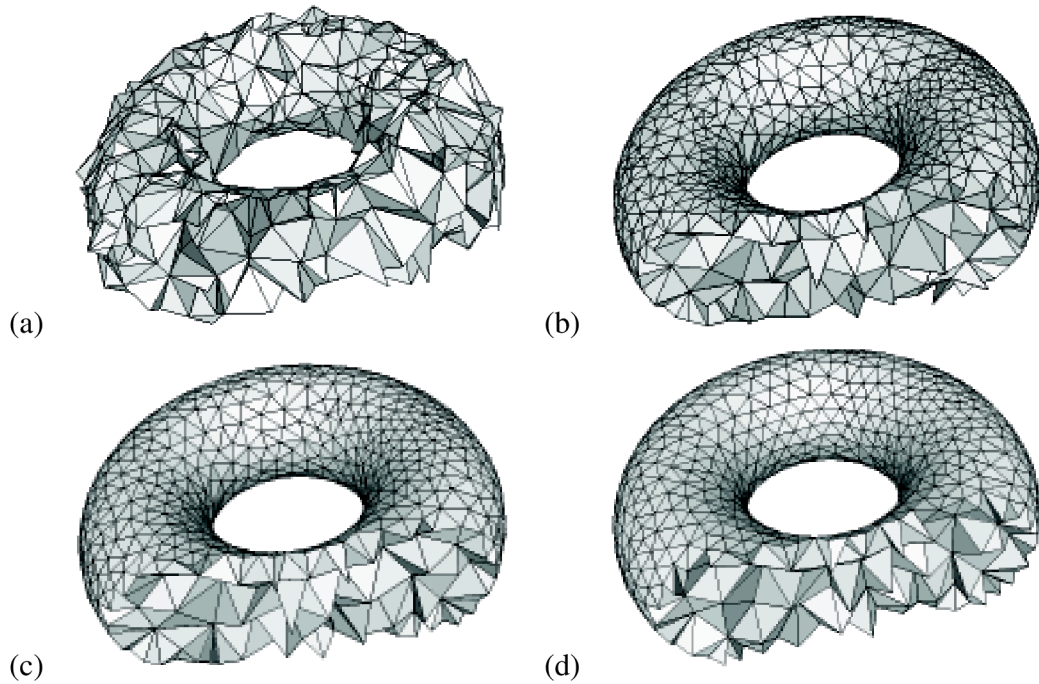


Figure 6.16: Variational tetrahedral meshing (VTM) – influence of the number of iterations on the mesh quality: the initial mesh (a); the same mesh after 3 iterations of the VTM algorithm (b); the mesh after 10 iterations (c); and 50 iterations (d).

Afterwards, vertices with a non-zero distance sum are those boundary vertices we are looking for. Focused on the boundary vertices that require a specific treatment, these vertices are moved to the average value they each have accumulated during the pass over all edge voxels. Such kind of iterative optimization is called *Lloyd's algorithm* [33], also known as *Voronoi iteration* or *relaxation*.

6.6.4 Boundary Refinement

The boundary refinement increases quality of the mesh in the sense of image edges approximation. Similarly to other Delaunay refinement methods, new vertices are added to the mesh to guarantee this criterion.

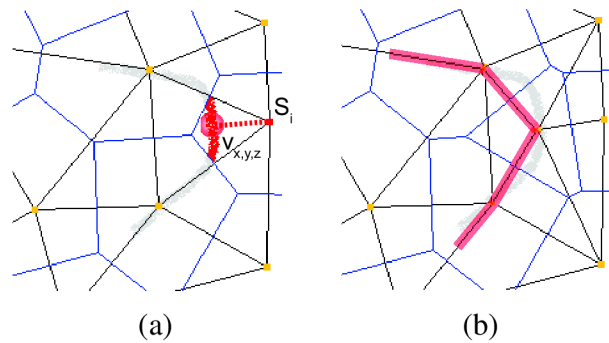


Figure 6.17: Illustration of the boundary refinement algorithm in two-dimensional space. Vertex S_i with an accumulated value (\sim distance from its closest edge point $V_{x,y,z}$) that is not itself located on an image edge exists in the mesh. Hence, a new vertex is added to the mesh in place of the point V .

In the first step, an algorithm similar to the identification of boundary vertices during the variational meshing is applied to locate proper places for new vertices:

1. Prepare an array of accumulators containing coordinates and distance of the edge point closest to each vertex. Initialize the distance to some large value d_{max} .
2. For each voxel V_i lying on an image edge:
 - (a) Locate its nearest vertex S_j .
 - (b) Compare the distance $d(V_i, S_j)$ with the value currently stored in the corresponding accumulator.
 - (c) If the distance is smaller, exchange the values in the accumulator.

In the second step, all accumulators that contain a distance lower than d_{max} are investigated. If there is a vertex with an accumulated value that is not itself located on an image edge, a new vertex is added to the mesh in place of the closest image edge point – the coordinates in the accumulator.

6.6.5 Dealing with Slivers

The variational meshing technique produces well shaped tetrahedra through the domain. Unfortunately, slivers could appear near the domain boundary, as the boundary vertices are guided by Lloyd relaxation, thus unaffected by the 3D optimization (Fig. 6.18).

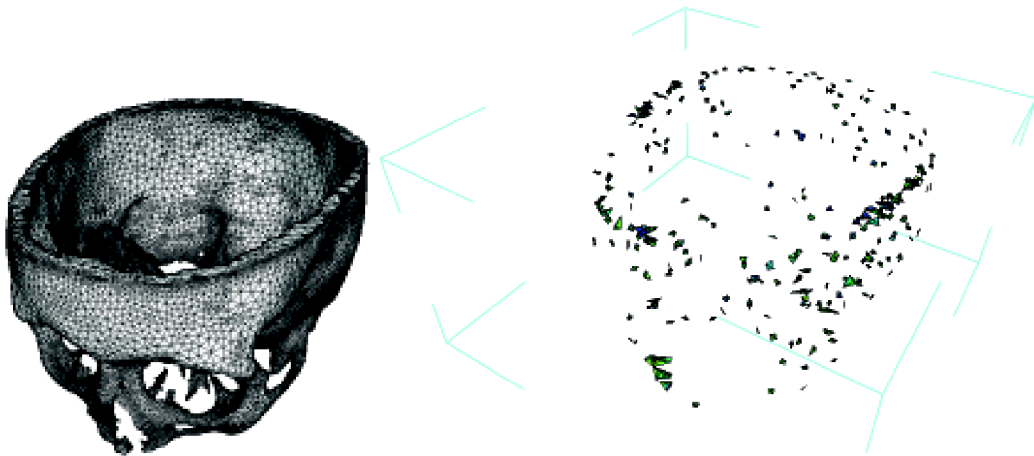


Figure 6.18: Slivers that occur in the final mesh close to domain boundaries.

Two powerful techniques [70, 105] capable of eliminating almost all slivers in the mesh were briefly described in Sec. 4.3.2. Both algorithms are based on explicit random perturbation of vertices incident to slivers. The region of locations of a vertex p such that the tetrahedron incident to p is a sliver, is very small. Moving the point out of this region ensures that the sliver disappears once the Delaunay connectivity is updated.

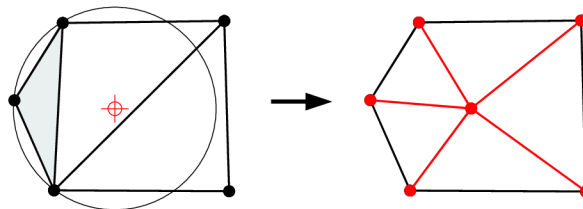


Figure 6.19: Tetrahedra refinement inserting a new vertex in the center of the circumsphere.

Towards creation of a sliver-free mesh, after each iteration of the adaptation scheme, the mesh is repeatedly tested for slivers, and new vertices lying in the center of sliver circumspheres are inserted to the mesh with a small random perturbation. If such addition does not eliminate the sliver, or generates new one, the vertex position is perturbed again. Such vertex perturbation continues until an optimal position is found, thus the sliver is successfully removed.

6.7 Mesh Segmentation

Within the mesh segmentation phase, all tetrahedra are classified into individual image regions which hopefully correspond to tissues of particular type.

Every tetrahedron t_i of the mesh is characterized by its feature vector. Individual features detail image structure of the tetrahedron, and perhaps its close neighborhood. Feature vectors may be grouped by the help of any conventional unsupervised clustering technique that classifies feature vectors into a certain number of classes.

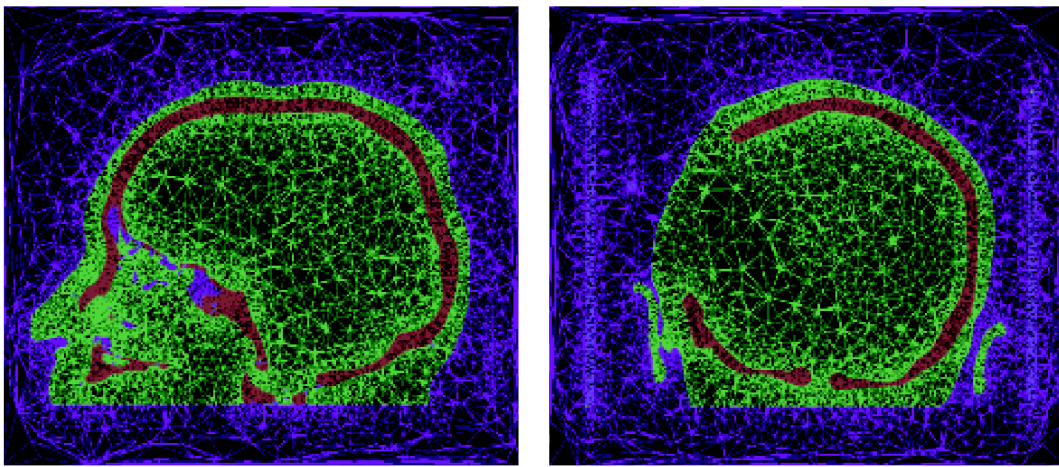


Figure 6.20: Result of the tetrahedral mesh segmentation phase – orthogonal cuts through the classified mesh.

6.7.1 Clustering Techniques

Three different algorithms are proposed for the unsupervised clustering of feature vectors (Sec. 3.5) into image regions:

- Fuzzy C-means (shortly *FCM*) algorithm [95],
- Gaussian Mixture Model optimized by the popular Expectation-Maximization (*EM-GMM*) algorithm [81].
- Min-Cut/Max-Flow graph-based algorithm [8].

First two techniques do not take into account any spatial/global information about the tetrahedra, since the classification is performed within local vicinity of processed tetrahedron.

Improvements can be made by incorporating global principles. Viewing the mesh as undirected graph, with edges weighted according to the similarity of feature vectors, would allow one to use graph algorithms (graph cuts, path-based clustering, etc.) for the segmentation. In this sense, the *Min-Cut/Max-Flow* [8] algorithm is used to cut a graph whose edges are evaluated according to a similarity of two adjacent tetrahedra.

$$m(P, Q) = \sum_{i=1}^N |p_i - q_i| \quad (6.9)$$

$$s(P, Q) = \frac{\sum p_i \cdot q_i}{\sqrt{\sum p_i^2 \cdot \sum q_i^2}} \quad (6.10)$$

The similarity of two adjacent tetrahedra (i.e. two feature vectors P and Q) can be defined as a distance function in the feature space. Most common choice is the *Manhattan* (Eq. 6.9) or the *Cosine* (Eq. 6.10) distance function. An alternative is the use of simple criterions described in Sec. 6.7.4.

6.7.2 Feature Extraction

In fact, the first two components of a tetrahedron's feature vector are mean pixel value $\mu(t_i)$ and intensity variance $\sigma(t_i)$ of voxels inside the tetrahedron. Others may cover image texture/shape properties:

- features derived from gray level co-occurrence matrices [62, 106],
- local moments of the image function [107],
- histogram of *Local Binary Patterns (LBP)* [37, 84, 85],
- wavelet features [2, 3, 109], etc.

and spatial configuration of adjacent tetrahedra.

LBP Texture Features. Local binary patterns [84] were presented by *T. Ojala* as a robust feature extraction technique capable to differentiate textures in an image. In the last few years, LBP have received a lot of attention from many researchers. The LBP feature was successfully applied to a wide range of different applications from texture analysis to face detection and recognition. See [85], for more details on LBP features, and their principles.

An advantage of the LBP feature is its simplicity. Extraction of LBP features is very fast for 2D images, so the LBP may also be a good choice in case of volumetric data. A novel method for the fast computation of fully rotation invariant local binary patterns on 3D volume data has been presented by *Fehr et al.* [37].

In this work, a simple extension of the LBP framework is used in all experiments with texture-based classification of tetrahedral meshes. Three individual rotation invariant, uniform LBP "image" features [85] are extracted in each possible direction (i.e. XY, XZ and YZ planes). The final feature vector contains three concatenated normalized histograms of these LBP features extracted from within the tetrahedron.

Any texture feature extraction is problematic if a tetrahedron is relatively small, just a few voxels are available. In that case, the texture analysis fails. Simple grouping of adjacent tetrahedra into larger units may reduce this problem. Another solution may be to reject classification of small tetrahedra, and use only relevant portion of the data. These non-classified tetrahedra, that appear mostly near to region boundary, will be assigned to particular regions in a next merging phase.

6.7.3 Agglomerative Merging

Topology of the tetrahedral mesh is suitable for image segmentation techniques such as region growing and merging. Instead of pixels and the traditional 4- and 8- pixel connectivity, tetrahedra adjacency is incorporated.

In the vector segmentation scheme, the agglomerative region merging [63] is used to assign non-classified, small tetrahedra into already known segments.

The agglomerative merging starts with a partition of the volumetric data into N regions (each region consists of one or more tetrahedra), and sequentially reduces the number of regions by merging the best pair of regions among all possible pairs in terms of a given criterion. This merging process is repeated until the required number of segments is obtained.

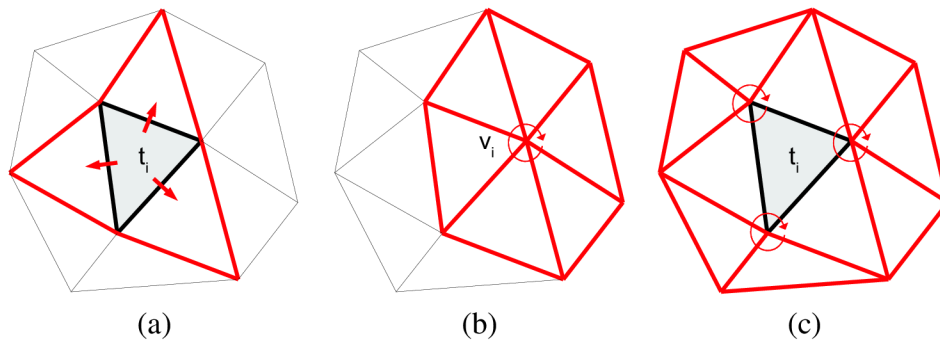


Figure 6.21: Figure illustrates tetrahedra adjacency (a); vertex 1-ring neighborhood (b); and tetrahedron 1-ring neighborhood.

Since only adjacent pairs of regions can be candidates for merging, the similarities can be represented as a *region adjacency graph (RAG)* in which nodes and edges denote regions and pairs of adjacent regions. To update the graph efficiently after merging a pair of regions, the graph is represented as a sparse matrix. In order to speedup the search for a pair of similar regions, an index of a matrix row containing the maximal similarity is stored and updated after the merging of two regions is performed.

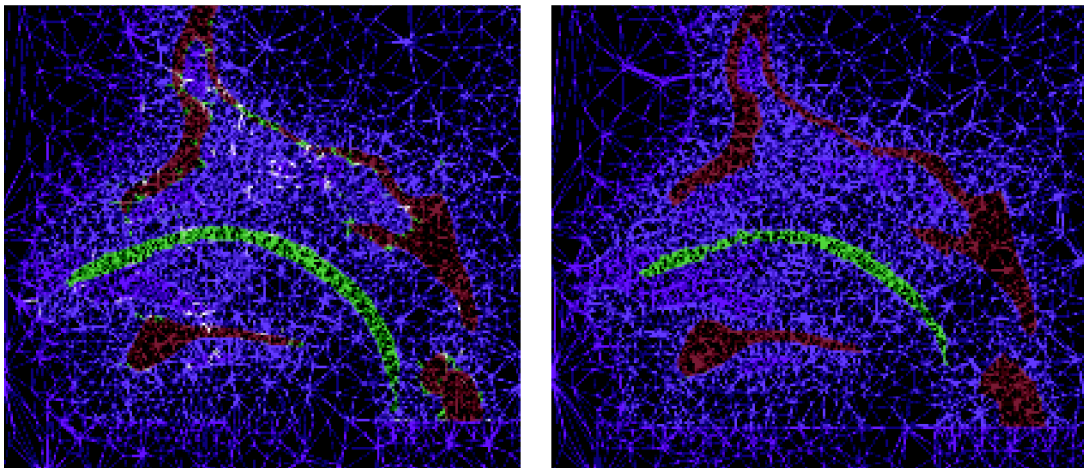


Figure 6.22: Result of the proposed agglomerative merging. The merging was applied as the post-processing step after the mesh segmentation via more sophisticated statistical clustering.

The general procedure of agglomerative merging can be summarized as follows:

1. Construct a graph which represents adjacency of regions and calculate similarity of all possible (i.e. adjacent) pairs of regions.
2. Search for the best pair of regions to merge.
3. Merge the pair of regions, recompute features related with the regions and update the graph structure.
4. Repeat the merging until no more regions can be merged, or the final number of regions was reached.

In practice, performance of this algorithm can be improved by a simple weighting of the similarity of two adjacent regions according to the number of voxels in both regions:

$$C(r_i, r_j) = \frac{N_i + N_j}{N_i N_j} S(r_i, r_j). \quad (6.11)$$

If the final number of regions is unknown, the stopping criterion for the merging should be a ratio between similarities $C_{t-1}(r_i, r_j)$ and $C_t(r_i, r_j)$ of last two merged pairs of regions. To prevent early termination of the algorithm, approximately first 10% of all possible merges are made without any checking of this termination criterion.

6.7.4 Similarity Measures

Let t_i and t_j be two feature vectors extracted for a group of adjacent tetrahedra, or a single tetrahedron. Similarity measure is a function whose value is greater as the difference between two feature vectors increases. Basic similarity measures are the mean intensity value and statistical test of the similarity based on voxel value variance:

$$S_\mu(r_j, r_i) = \exp\left(-\frac{1}{2\rho^2} |\mu_{r_i} - \mu_{r_j}|^2\right), \quad (6.12)$$

$$S_\sigma(r_j, r_i) = \frac{\sigma(r_i)\sigma(r_j)}{\sigma^2(r_{i,j})}, \quad (6.13)$$

where the parameter ρ affects sensitivity of the measure and $\sigma(r_i)$ is the variance of intensity in the region r_i and $\sigma(r_{i,j})$ is the variance of intensity in a joint region $r_i \cup r_j$. Both μ and σ are components of the feature vector.

These basic similarity measures can be evaluated for very small tetrahedra (\sim couple of voxels), therefore they serve as a merging criterion during the agglomerative merging when too small non-classified tetrahedra along boundaries are assigned into neighboring segments.

6.7.5 Noise Reduction

Due to noisy input data and classification errors, some isolated tetrahedra, classified to a region different from its neighbors, may appear in the segmented mesh. This kind of misclassification can be reduced using a simple filtering scheme similar to the *median filtering* [39] known from image processing. The noise reduction phase processes sequentially each tetrahedron t_j in the mesh in the following way:

1. Identify region of the tetrahedron t_j as well as all regions of tetrahedra adjacent to any edge of t_j (i.e. the 1-ring neighborhood).
2. Estimate histogram of occurrences of particular regions weighted by volume of tetrahedra.
3. If there is a significant maxima in the histogram different from the original region, re-classify the current tetrahedron t_j .

Repeating the filtering until no change may lead to large changes in the classification. So, the maximal number of iterations is limited. In practice, two or three iterations are enough.

6.7.6 Tissue Classification

Result of the segmentation phase is the classified mesh, all tetrahedra are classified into one of the regions. However, a relationship between these regions and human tissues is unknown. A unique tissue name or label should be assigned to every region in order to create a 3D model of the tissue. This can be accomplished manually, according to a density-tissue relationship, or by using an anatomical atlas. This step of the mesh segmentation is not addressed in the thesis, the manual selection of region(s) of interest is performed.

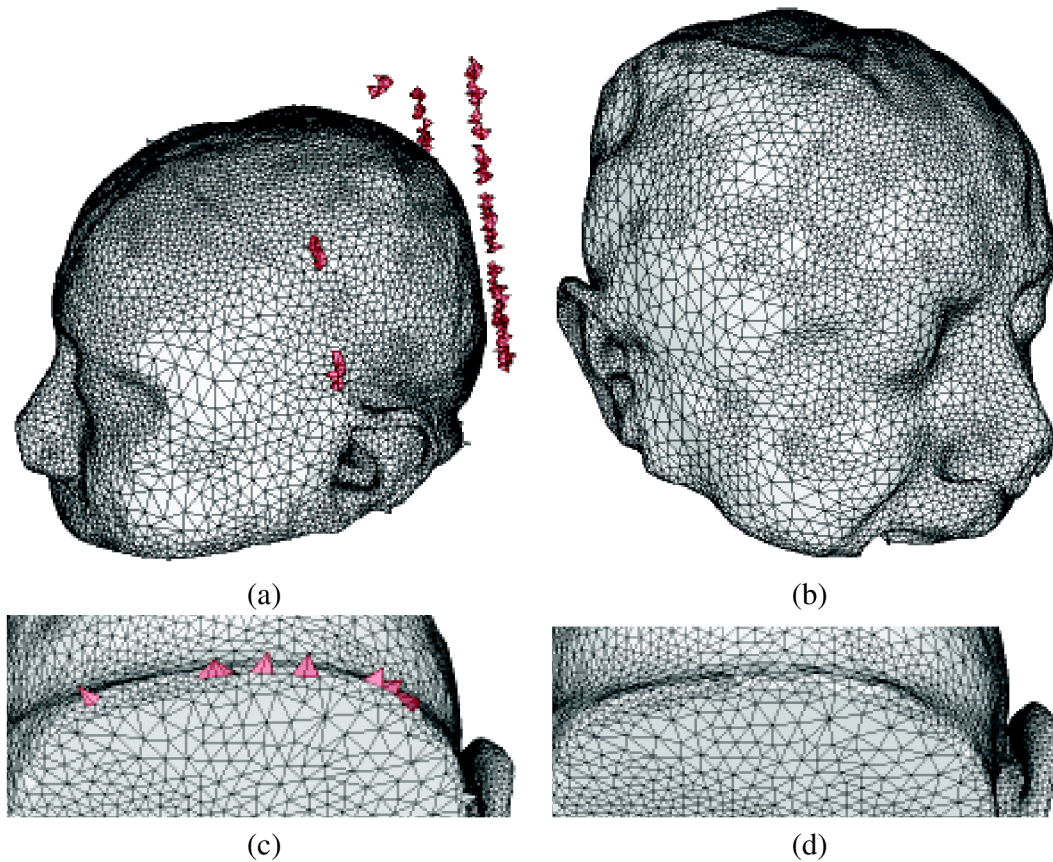


Figure 6.23: Surfaces extracted from a segmented mesh: an original surface without any filtering (a); the same surface after filtering of isolated small parts (b); artifacts that sometimes appear close to sharp edges (c); and the surface after a fine smoothing that removes only the artifacts, but preserves all other features (e.g. one iteration of the tuned Taubin’s smoothing algorithm [103]).

6.7.7 Surface Extraction

Once the mesh is properly segmented, surface of any region R_k can be easily extracted. All tetrahedra through the mesh are traversed looking for boundary faces that forms surface of the desired region. Boundary faces can be intuitively identified as faces between two different regions:

1. Clear the output set of faces S_k – closed surface of the region R_k .
2. Sequentially process every tetrahedron t_i in the classified mesh:

- If the tetrahedron t_i is classified into the region R_k
 $R_k = \arg_{R_i} \max m(t_i, R_i)$
 - Retrieve all tetrahedra t_j adjacent to t_i .
 - If t_j is classified into a region different from R_k , then add face f_{ij} incident to both tetrahedra to S_k .

3. Save the extracted surface S_k .

The extracted surface is closed and its mosaic conforms to the chosen parameters of the meshing – minimal required edge length L_{min} , coefficient K that controls gradation of the mesh, and the desired average tetrahedron size T_{avg} .

After the extraction, small isolated parts of the surface may be filtered to obtain a single closed surface if required. Moreover, to avoid artifacts that rarely appear on the surface, the final surface can be filtered for sharp spikes (Fig. 6.23).

Chapter 7

Experimental Results

In this chapter, results of the introduced vector segmentation method are discussed, compared to a traditional surface reconstruction technique based on the Marching Cubes method. The vector segmentation was mainly designed for segmentation of volumetric medical images towards anatomical modeling of fundamental tissues (i.e. soft and bone tissue) and their surfaces. In order to evaluate precision, advantages and disadvantages of this method, number of experiments on real medical CT data, as well as on artificial volumetric data, were carried out. However, due to the complexity and variability of medical images, the testing was divided into several separate tasks:

Surface accuracy. Evaluation of an accuracy of surfaces extracted from meshes, and comparison with the traditional Marching Cubes method followed by smoothing and decimation.

Mesh quality. Evaluation of produced tetrahedral meshes with respect to the quality of tetrahedra, distribution of dihedral angles through the mesh, the number of slivers, etc.

Mesh segmentation. Comparison of different unsupervised clustering techniques applied to meshes to classify tetrahedra into segments. Manually annotated medical CT data are used as a ground truth.

7.1 Surface Accuracy

In case of the anatomical modeling, an error between reconstructed surfaces of human tissues and a "ground truth" must be minimal to guarantee correctness of a planned surgery, or accuracy of a custom-made implant.

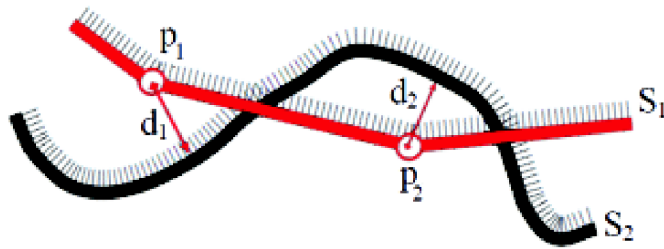


Figure 7.1: Error between two meshes seen as a distance between closest points sampled on the surface [20].

The following evaluation of the surface accuracy compares surfaces produced by the vector segmentation algorithm against ones made by the traditional *Marching Cubes* (MC) method followed by mesh smoothing and mesh decimation steps (for more details see Sec. 5.2.2). Since the smoothing is crucial for overall precision of the surface, two standard approaches were tested:

- Taubin's smoothing algorithm [103] that maintains the volume of the mesh,
- HC algorithm [114] that preserves sharp edges and corners in the mesh.

The MC algorithm produces very large meshes. Hence, after the smoothing, the *Quadric Edge Collapse* decimation algorithm, a variant of the well known edge collapse algorithm based on quadric error metric proposed by Michael Garland and Paul Heckbert [41], was used to reduce size of the mesh – the number of triangles. This re-meshing technique, as well as both the utilized smoothing methods, are implemented in the *Mesh-Lab* [21] tool which is de facto standard in the area of meshing.

7.1.1 Artificial Data

Artificial volumetric data of basic solids such as rectangular solid, cylinder, cone, semi-sphere, pyramid and the *Stanford bunny* (Fig. 7.2) were generated for the testing. An idea

of this measurement is to rasterize a solid into 3D raster, reconstruct surfaces from obtained volumetric data, and evaluate error between the reconstruction and the original surface of the solid.






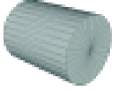
						
Model	bunny	cone	sphere	cuboid	toroid	cylinder
Num. of faces	69664	64	960	12	1024	182
Bbox diagonal [mm]	206	154	196	183	164	182

Table 7.1: Solids used for surface accuracy testing.

All solids (Tab. 7.1) were fitted into a volume having size $256 \times 256 \times 256$ voxels. The real voxel size was chosen to be 0.5mm in all directions.

7.1.2 Surface Approximation Error

An error between reconstructed surface and the original model is estimated using the *Metro* [20] tool. The *Metro* allows one to compare the difference between a pair of surfaces (e.g. a triangulated mesh and its simplified representation) by adopting a surface sampling approach and point-to-surface distance computation.

The approximation error between two meshes is defined as the distance between corresponding sections of the meshes (Fig. 7.1). Given a point p and a surface S , the distance $e(p, S)$ is defined as:

$$e(p, S) = \min_{v \in S} d(p, v), \quad (7.1)$$

where $d(p, v)$ is the Euclidean distance between two points p and v .

$$RMS = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{N}} \quad (7.2)$$

In the following testing, the *mean distance*, *maximum distance* and *quadratic mean* (i.e. root mean square – RMS) error between the two meshes are presented as measures of the surface accuracy.

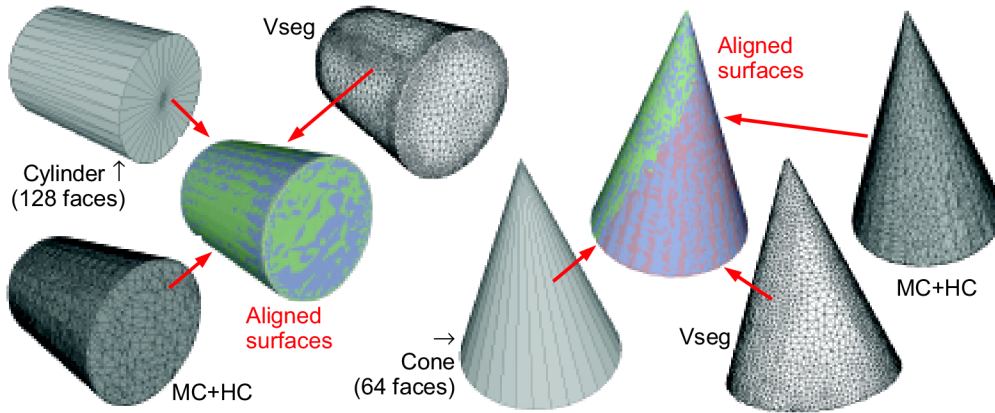


Figure 7.2: Alignment of surfaces performed before evaluation of the approximation error. The original solids serve as a reference for the error measurement.

7.1.3 Alignment of Surfaces

Before any comparison of two meshes, both meshes are precisely aligned using the *ICP* (*Iterative Closest Point*) [6] algorithm. Such alignment is very important. It eliminates errors caused by the differences in an internal implementation of different meshing methods. Each algorithm handles a regular grid of voxels in rather different way which may cause errors equal to approximately half the size of voxel.

7.1.4 Case Study – Stanford Bunny

First, an accuracy of surfaces produced by the iterative mesh adaptation scheme (Sec. 6.6) will be evaluated for a single model only – the Stanford bunny. This allows to explain differences between the vector segmentation technique (*VSeg* for short) and both the *MC+HC* and *MC+Taubin* combinations of the Marching Cubes method with one of the mesh smoothing filters.

As mentioned above, the original bunny was rasterized into a volume of 256^3 voxels, the synthetic volumetric data of the bunny was created. Afterwards, its surface was reconstructed using all compared techniques. Surfaces were aligned against the original model, and the approximation error was calculated.

Fig. 7.4 shows histograms of error distribution for surface models of different level of detail as returned by the Metro tool. Height of the histogram column denotes the fraction

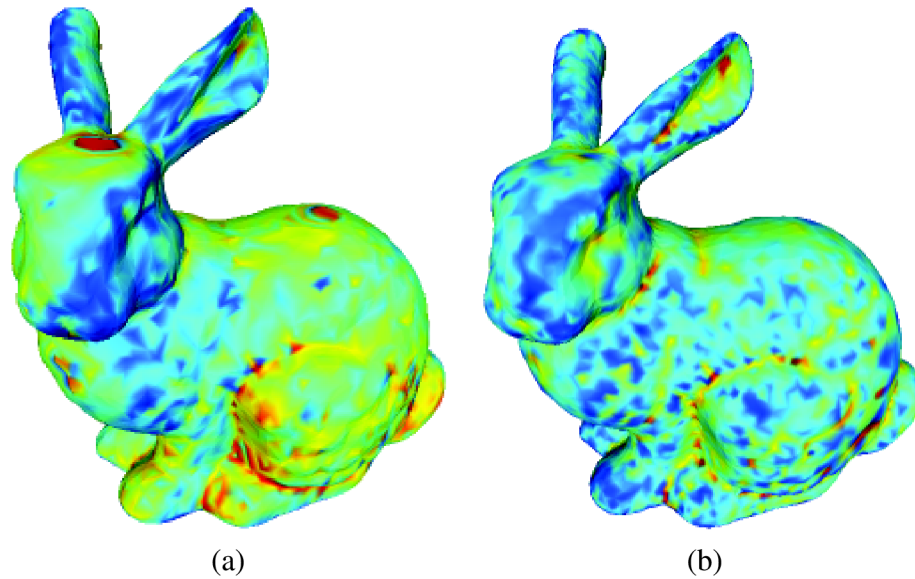


Figure 7.3: The bunny colored by interpolating and mapping the surface approximation error: the MC+HC method (a), and VSeg method (b). The error was computed in each vertex as the mean of the errors on the incident faces.

of the surface having an adequate error. All histograms are normalized, so that the sum of all columns should give 1.

In case of the MC-based methods, meshes were decimated to exactly 10k, 20k and 50k faces. Parameters of the vector segmentation (K that controls gradation of the mesh, average tetrahedron size T_{avg} and minimal allowed length of tetrahedra edges L_{min}) were chosen to produce almost the same number of faces.

Apparently, the VSeg method outperforms both the smoothing-based methods. However, the difference is more evident for smaller meshes. As the number of faces increases, the error distribution of the VSeg method moves towards the MC+HC method which seems to produce lower error than the MC+Taubin combination. Direct meshing of volumetric image data seems to be more accurate approach than post-processing methods smoothing reconstructed surfaces without any relationship to the original image data.

Similarly, graphs of the mean error (distance) and the RMS value shown in Fig. 7.6 maintain the assumption that for smaller meshes the VSeg method approximates surfaces significantly better than the MC+HC and MC+Taubin methods.

The question is why the performance decreases with the increasing number of faces? In fact, this behavior is opposite to the one of MC+HC and MC+Taubin methods. The

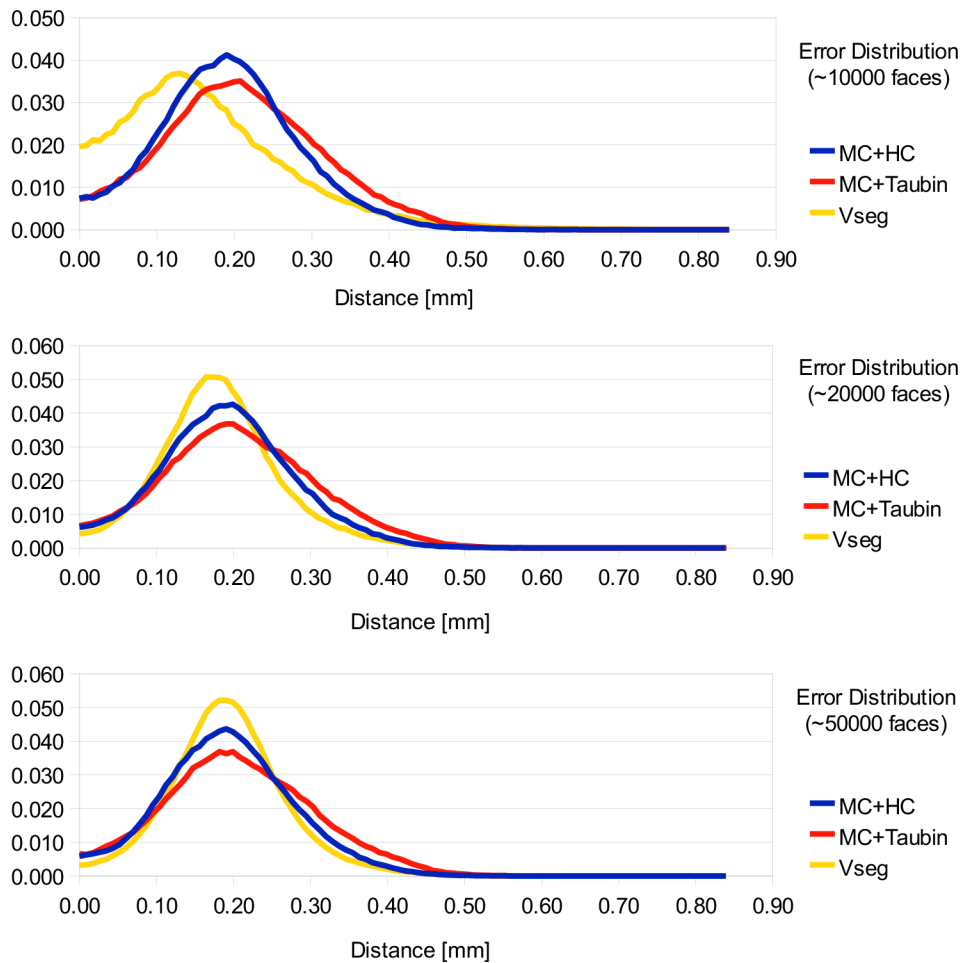


Figure 7.4: Histograms of the surface approximation error for three meshes with a different level of detail (number of faces) – the bunny model.

answer lies in the iterative adaptation of the mesh to the underlying image structure. To obtain a more detailed surface, the minimal allowed edge length L_{min} must be decreased. However, the resolution of the raster data is limited. Decreasing the L_{min} down to the real size of a single voxel causes the relocation of vertices along image edges to not perform optimally (Fig. 7.5). In the extreme case, the optimal vertex position is calculated as an average of a single "edge" voxel.

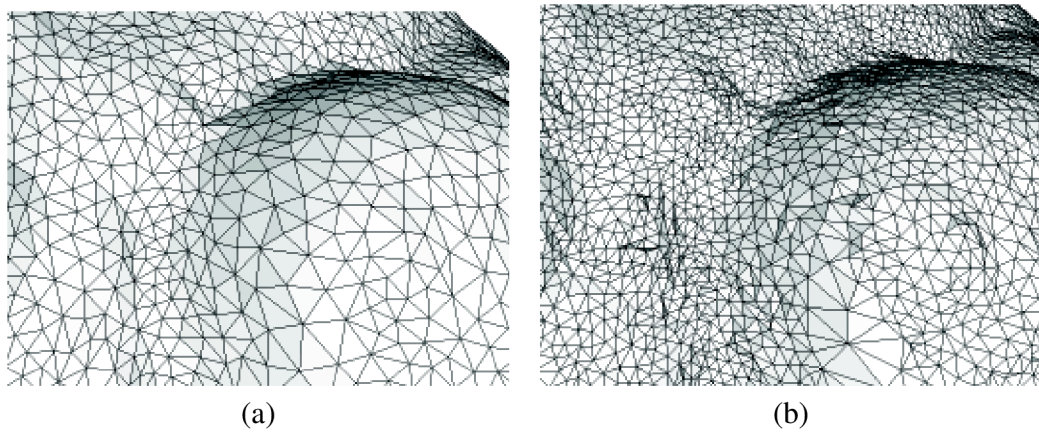


Figure 7.5: Surfaces extracted from meshes with different setting of the minimal tetrahedron edge length: $L_{min} = 1.5mm$ (a); and L_{min} equal to the voxel size $0.5mm$ (b). Too small L_{min} results in poorly smoothed surfaces.

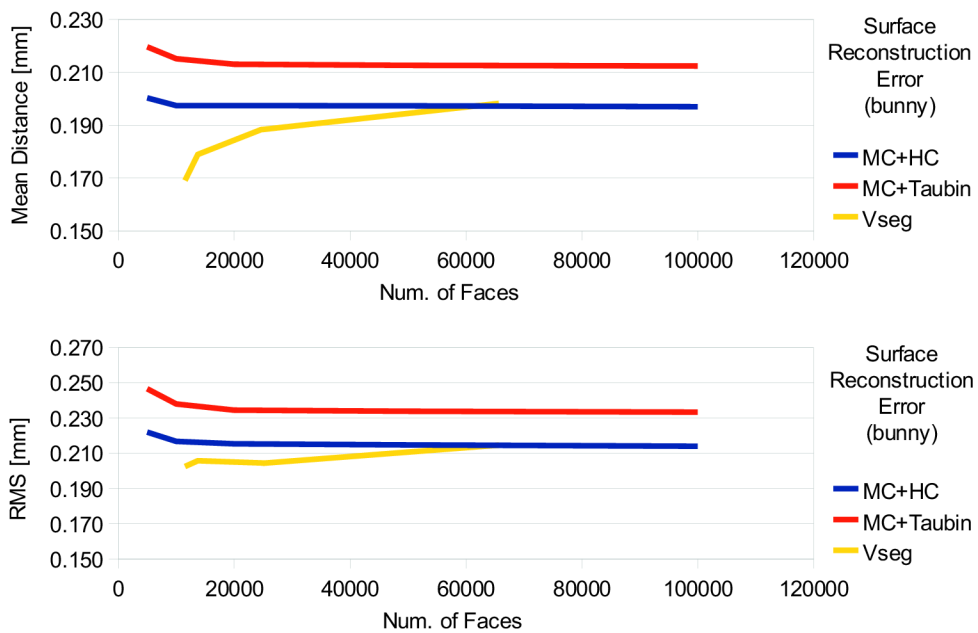


Figure 7.6: Surface approximation error – the bunny model at different level of details.

7.1.5 Overall Statistics

This close relation between the surface approximation error and the minimal allowed edge length L_{min} in the tetrahedra mesh was confirmed by evaluating the overall error for all

models in the test dataset. Fig. 7.7 illustrates the overall mean approximation error and the maximal error depending on the number of faces in the mesh. The same behaviour as with the bunny model can be seen. The VSeg method outperforms the smoothing-based methods for smaller meshes up to 20k faces. As the number of faces increases, the mean error of the VSeg method grows too. For meshes larger the 35k faces, the surface approximation error exceeds the error of the MC+HC method.

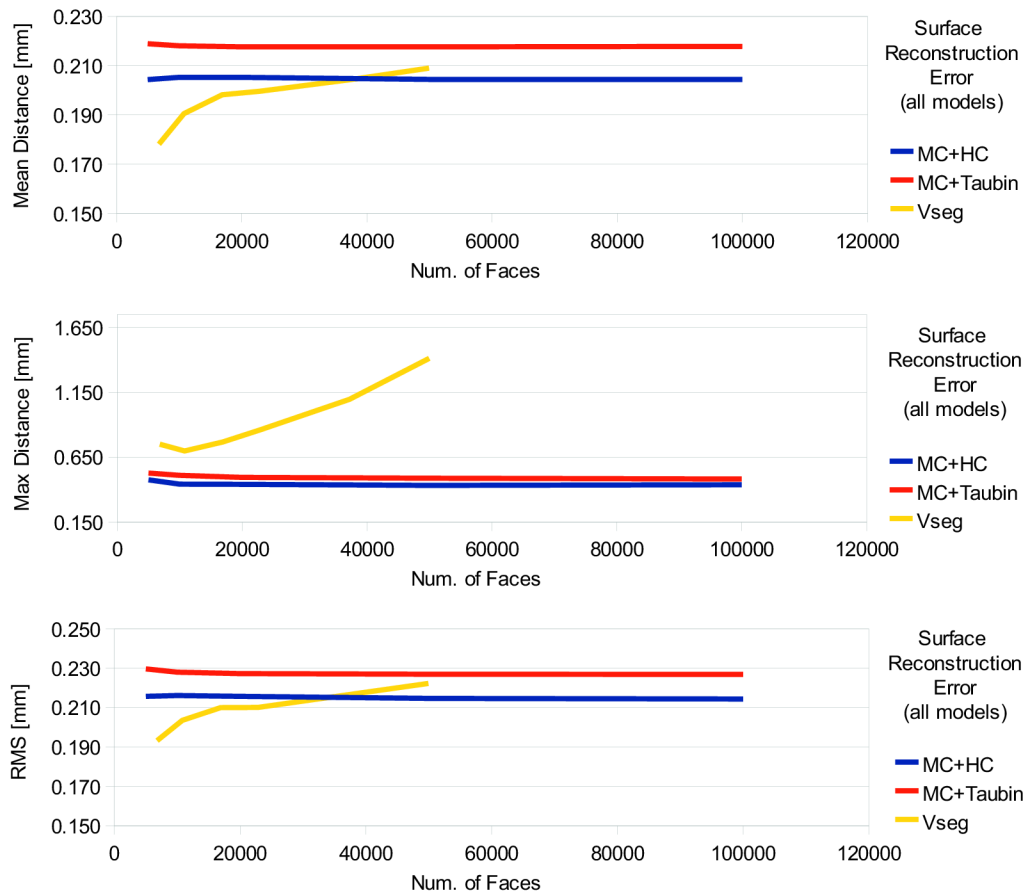


Figure 7.7: Overall surface approximation error – whole set of test models.

An interesting effect of the number of faces to the maximal error is notable in the same Fig. 7.7. In general, the maximal error, or maximal distance between sampled points on compared surfaces, is much greater for surfaces obtained by the VSeg method than the ones produced by the MC-based approaches.

Analogous to the previous discussion, explanation of the large maximal error in meshes

produced by the vector segmentation is the matter of the meshing process itself. The isotropic meshing generates high quality meshes with almost equilateral tetrahedra. Therefore, close to the sharp surface edges, the final mesh approximates the surface very roughly because of the limitation of tetrahedra shape and also the chosen minimal edge length. Illustration of this *problem* can be found in Fig. 7.8.

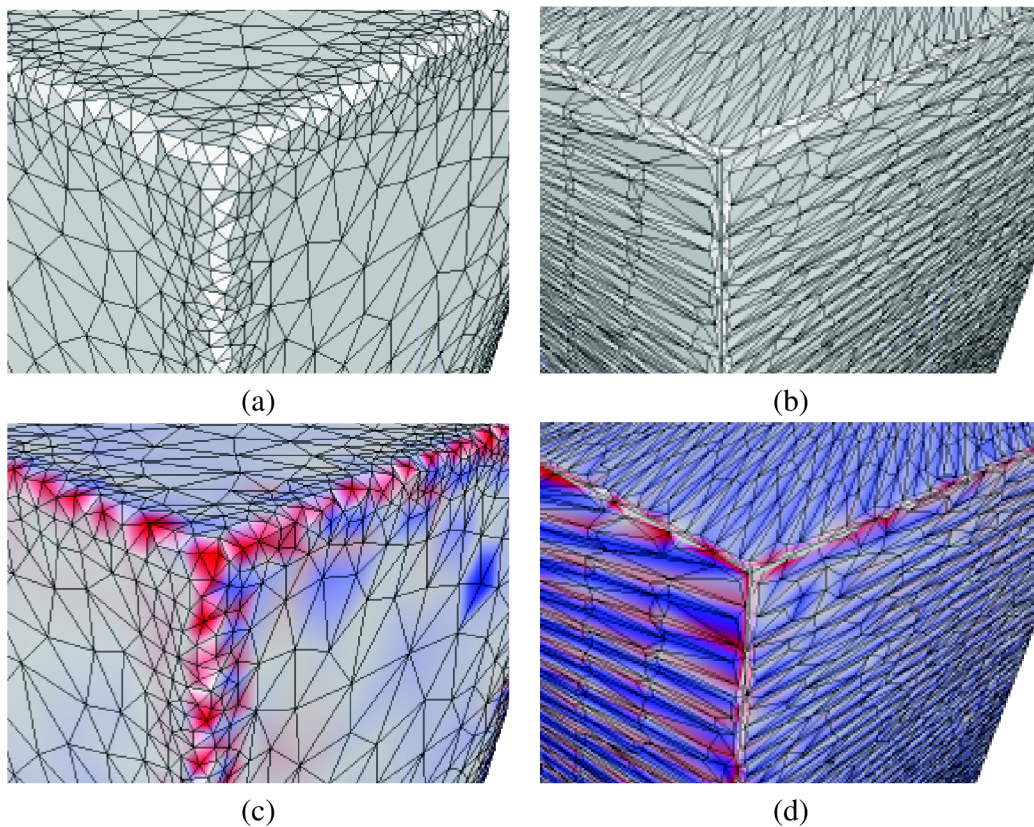


Figure 7.8: Error distribution on the reconstructed surface. The mean error of the VSeg method (a,c) over the entire surface is lower than the error of MC+HC method (b,d). On the other hand, the maximum error is larger along the sharp edges, approximately 1x for this model.

In Sec. 6.4.2, a modified 3D SUSAN corner detector has been proposed to detect sharp edges (corners) in the volumetric data. The idea of the corner detection is to refine the mesh close to the detected corner points, and moreover to attract vertices directly to the corner places. The corner detection, however, does not perform good while processing the artificial volumetric data. The reason way errors proportional to the minimal edge length are caused close to the sharp edges.

An exciting result of this experiment is that the mean error of all tested methods is lower than half of the voxel ($\Delta x = \Delta y = \Delta z = 0.5mm$).

7.2 Mesh Quality

The core of the vector segmentation algorithm is built upon the isotropic variational meshing approach (Sec. 6.6). The meshing phase is used for partitioning of the volumetric data, while the structure of the mesh is iteratively optimized to provide good distribution of dihedral angles in the mesh, and adapted to the underlying image data. Goal of the mesh adaptation is to generate meshes where faces of tetrahedra adjacent to image edges approximate boundaries of individual image regions. The proposed iterative mesh adaptation scheme was developed to produce high quality meshes suitable for many practical applications. Not only for the surface reconstruction of a desired tissue, but also for simulations that takes the internal structure of the tissue into account. The quality of produced meshes is discussed in this chapter.

7.2.1 Adjusting Parameters

There are several parameters of the meshing process that must be chosen initially. These parameters affect gradation of the mesh, average size of elements through the mesh, etc. For a detailed description of all capabilities of the meshing technique proposed in this thesis, few figures illustrating the effect of varying parameters can be found here.

To be more specific, there are three main parameters that control the isotropic meshing:

- $K \geq 0$... the K parameter controls gradation/isotropy of the mesh,
- T_{avg} (avg. tetrahedron size) ... the parameter represents the control space scaling factor, the T_{avg} prescribes size to tetrahedra through the mesh,
- L_{min} ... the minimal edge length.

Setting smaller K results in a visible gradation of the mesh not only inside the domain, but also on boundaries. The local feature size derived from medial axis (Sec. 6.6.2) takes more importance. As the K increases, the gradation depends more on the distance from the boundary, the gradation affects only interior parts of the mesh.

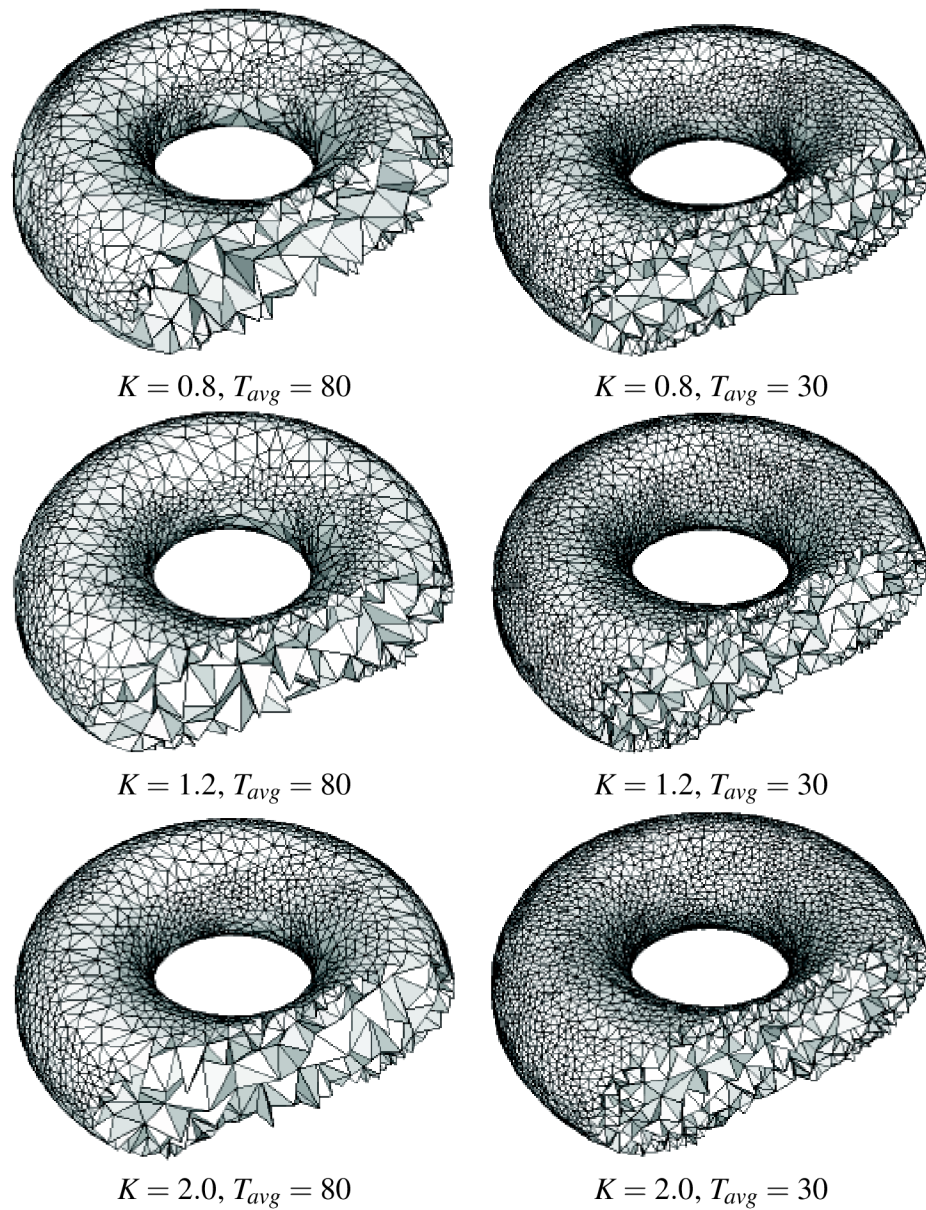


Figure 7.9: Influence of K and T_{avg} parameters on sizing of tetrahedra and gradation of the mesh.

The minimal edge length L_{min} must be chosen very carefully. The reason is the computation time which strongly depends on the minimal length. In time of the mesh adaptation, the L_{min} value is used for filtering of newly created vertices, so a properly chosen value may reduce runtime by early rejecting wrongly positioned vertices.

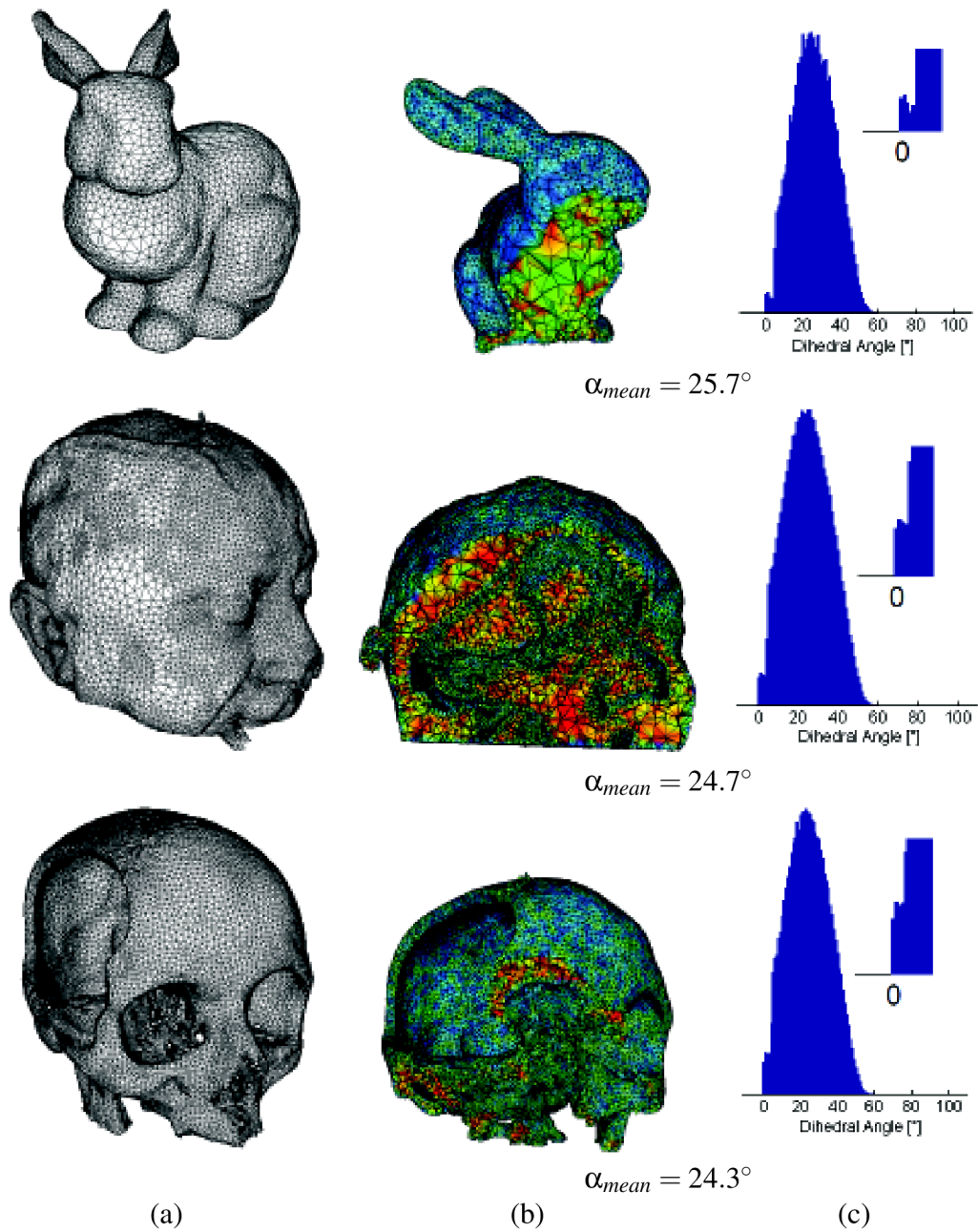


Figure 7.10: Example of surfaces extracted from tetrahedral meshes (a); cuts through the same meshes colored according to the tetrahedra quality; and histograms of minimal dihedral angles (c).

7.2.2 Minimal Dihedral Angles

A good measure of a mesh quality is the distribution of minimal dihedral angles through the mesh (Sec. 4.3). Fig. 7.10 shows histograms of dihedral angles for three different tetrahedral meshes made by the iterative meshing technique proposed in this thesis. The variational meshing approach introduced by Alliez *et al.* has been modified to work directly with volumetric data, and further extended by other techniques such as the boundary refinement and the sliver removal based on vertex perturbations (Sec. 6.6).

The original VTM approach produces well shaped tetrahedra inside the domain. However, poorly shaped tetrahedra and slivers may appear close to the boundary. Unfortunately, the same problem appears in case of the VSeg meshing method. Meshes in the middle of Fig. 7.10 are colored according to the quality of tetrahedra. Clearly, the quality of tetrahedra decreases as getting closer to the boundaries – the red shading moves towards blue. Even though the embedded sliver elimination algorithm removes a large number of poorly shaped tetrahedra, it does not ensure that all slivers will be successfully eliminated. This is the reason why tetrahedra of a low quality ($\alpha_{min} < 3^\circ$) are still present in all histograms.

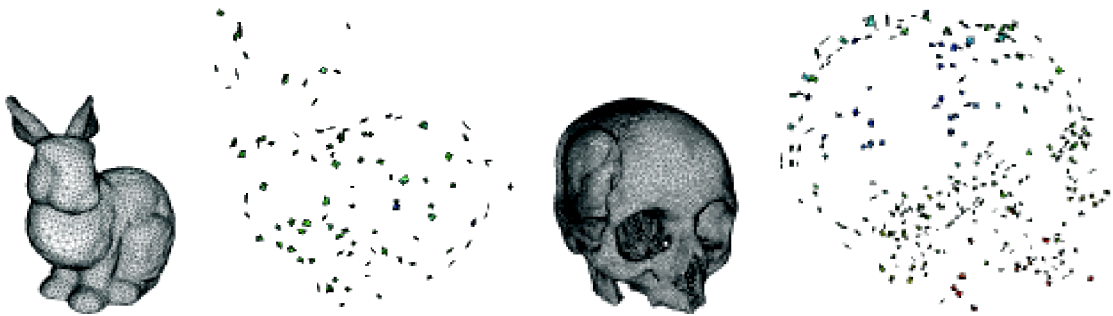


Figure 7.11: Illustration of slivers ($\alpha_{min} < 3^\circ$) that still remain in tetrahedral meshes after the mesh adaptation phase.

Because all boundary vertices are treated differently during the variational meshing, the quality of triangles approximating boundary of regions (i.e. tissues) is not affected by this knot. On the other hand, this ambiguity of the meshing process, when there are two types of vertices managed differently (interior vs. boundary vertices), is the source of the inappropriately shaped tetrahedra near the boundary. Recently, *J. Tournois* [105] has presented a new modification of the original VTM algorithm that particularly solves this problem and produces almost sliver free meshes.

One of the objectives of this thesis was to provide meshes suitable not only for surface modeling, but also for numerical simulations. The presence of slivers may cause instability of such simulations. Hence, this aspect of the presented VSeg technique remains open and will be addressed in the future. A smallest dihedral angle should be guaranteed.

7.2.3 Surface Quality

Surface of a desired region can be extracted directly from the mesh after the segmentation. In the extraction process, boundary faces are identified as faces between two distant regions. Quality of the extracted surface can be examined from two different sides. In the previous chapter, the approximation error of the surface has been studied. Here, the quality of surface with respect to the shape of triangles is briefly summarized.

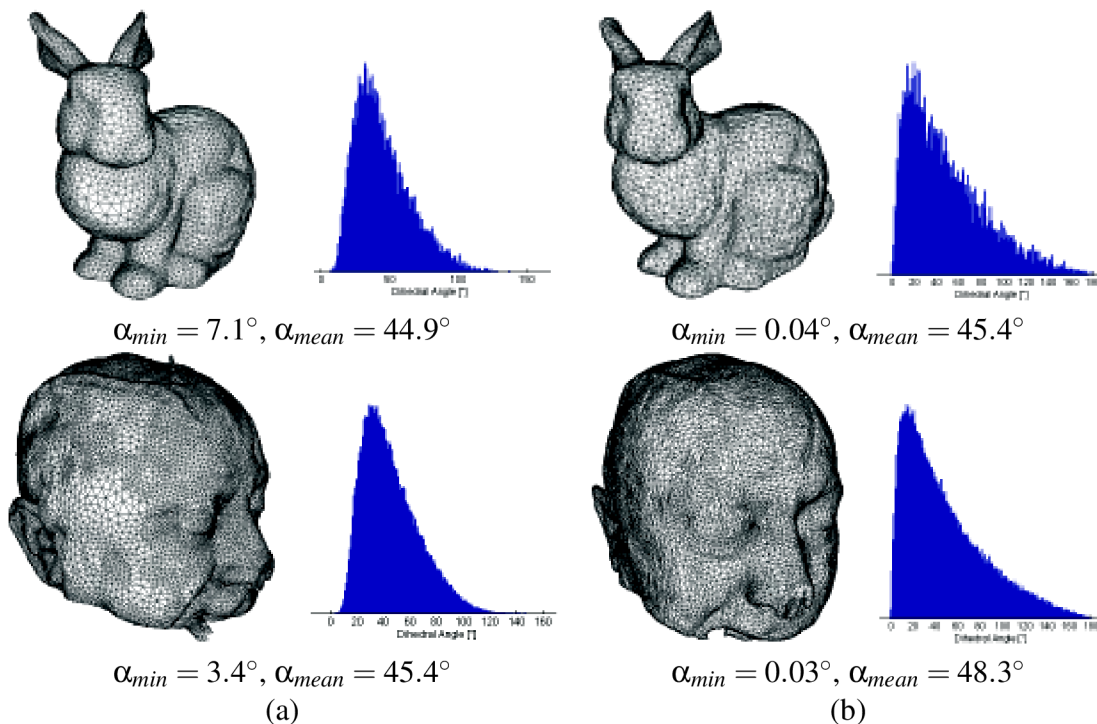


Figure 7.12: Quality of surfaces reconstructed using the VSeg method (a) and the MC+HC method. Histograms show distribution of dihedral angles.

In Fig. 7.12, surfaces extracted from tetrahedral meshes (the VSeg method) are compared against surfaces obtained from the MC+HC method. Contrast between both methods is evident. The VSeg approach itself produces well shaped triangles along the entire

surface, and moreover size of triangles is automatically adjusted according to a local complexity of the surface. Definitely, the MC meshes on the righthand side may be improved by existing re-meshing techniques, but such kind of post-processing increases the approximation error again.

7.3 Mesh Segmentation

During the mesh segmentation phase, all tetrahedra are classified into particular image regions with the aid of selected unsupervised clustering techniques. Every tetrahedron of the mesh is characterized by its feature vector that details image structure of the tetrahedron.

Accuracy of this last step of the vector segmentation approach is discussed in this section. As suggested in Sec. 6.7, three conventional clustering techniques were selected for testing with respect to our requirements (surface reconstruction of main tissues, i.e. bones):

- Fuzzy C-means (*FCM*) algorithm [95],
- Gaussian Mixture Model optimized by the Expectation-Maximization (*EM-GMM*) algorithm [81],
- Min-Cut/Max-Flow graph-based algorithm [8,9,40,57]. Many thanks to *O. Veksler* for his freely available implementation of the Min-Cut/Max-Flow algorithm [112].

These algorithms, in two different configurations of feature vectors – with or without LBP texture features, were tested on several CT data sets and their results were compared against manually annotated data – the ground truth.

In both the configurations, clustering techniques classify only relevant, sufficiently large tetrahedra. Very small tetrahedra (fewer than 10–30 voxels) remain unclassified. These non-classified tetrahedra, that appear mostly near to region boundaries, are assigned to particular regions in the subsequent merging phase (Sec. 6.7.3) which uses a linear combination of two simple similarity measures presented in Sec. 6.7.4.

7.3.1 Test CT Data

Medical CT imaging data having resolution mostly 512x512 pixels per slice were used in all the subsequent experiments. Concrete parameters of selected datasets are listed in

Tab. 7.2, highlighted data sets were manually annotated (segmented) by an expert. These data sets will be used for evaluation of segmentation error and comparison of methods in the following experiments. Here, I would like to thank to *3Dim-Laboratory s.r.o.* company for providing part of the test data.










Dataset	Size	Num. of slices	Resolution [mm]
 head1	512 x 512	147	0.38 x 0.38 x 0.60
 head2	512 x 512	197	0.45 x 0.45 x 1.00
 head3	512 x 512	169	0.45 x 0.45 x 1.00
 pelvis1	512 x 512	119	0.83 x 0.83 x 2.00
 pelvis2	512 x 512	318	0.63 x 0.63 x 0.70
 pelvis3	512 x 512	169	0.75 x 0.75 x 1.59
 pelvis4	512 x 512	125	0.86 x 0.86 x 2.00
 knees	512 x 512	367	0.73 x 0.73 x 0.62
 arteries	512 x 512	200	0.78 x 0.78 x 1.60

Table 7.2: Medical CT data selected for testing.

Manual segmentation of medical images is a very complicated task (Sec. 3.6). Not unfrequently, the segmentation made by different people varies. Every expert has his own view of the data and the correct segmentation. In order to quantify this phenomenon, one of the datasets were segmented by four different experts. Tab. 7.3 summarizes the obtained results.

An important issue is that the average error between two manual segmentations of the same data is about 0.96, measured by the *F-measure* of goodness which is described below. Occasionally, the error grows up (the F-measure decreases under) 0.92.

Soft tis./Bone tis.	Man1	Man2	Man3	Man4
Man1	-	0.921465	0.965222	0.895532
Man2	0.974299	-	0.948906	0.971415
Man3	0.993253	0.978834	-	0.922118
Man3	0.977621	0.979286	0.982031	-

Table 7.3: Difference between manual segmentations of the same dataset provided by four experts. The F-measure of goodness was calculated for soft tissues and hard tissues (i.e. bones).

7.3.2 Measuring Segmentation Accuracy

A way to match a segmentation to the ground truth is needed in order to evaluate performance of a segmentation algorithm. Many sophisticated measures of segmentation accuracy can be found in the literature [52, 108]. An often used measure of segmentation *goodness*, sufficient for our task, is the *F-measure* [110] which is also very popular in information retrieval and natural language processing. The *F-measure* combines *recall* r and *precision* p with an equal weight in the equation of the form:

$$F_{measure} = \frac{2rp}{r+p}, \quad p = \frac{T_p}{T_p + F_p}, \quad r = \frac{T_p}{T_p + F_n} \quad (7.3)$$

where p is the number of correctly labeled voxels (so called *true positives* T_p) divided by the total number of voxels labeled as belonging to the same region. The recall r is defined as the number of true positives divided by the total number of elements that actually should belong to the positive class (see Fig. 7.13). A perfect score of the F-measure is 1, in the worst case the measure is equal to 0.

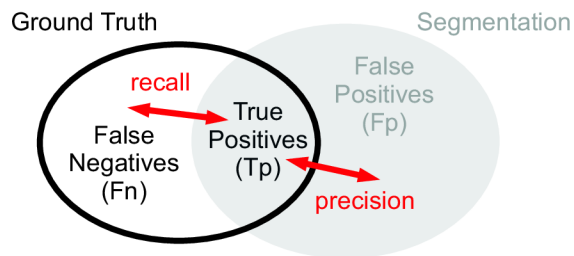


Figure 7.13: Illustration of the F-measure.

7.3.3 Meshing Segmented Data

Without too much effort, the Delaunay-based segmentation can be applied to already segmented data. Only the edge detection step and the mesh segmentation must be modified. Actually, both steps are simplified. Image edges are detected at those points where two adjacent regions touch each other. The assignment of a tetrahedron into a concrete region is made according to the labeling of voxels in the tetrahedron's interior.

In this case, the ability of the VSeg technique to mesh volumetric data is used for surface reconstruction. Fig. 7.14 shows an error between the obtained mesh and the segmented volumetric data. The F-measure rates how precisely the mesh approximates the original data. To compare this difference, all tetrahedra are rasterized into the volume data of the same size as the original one.

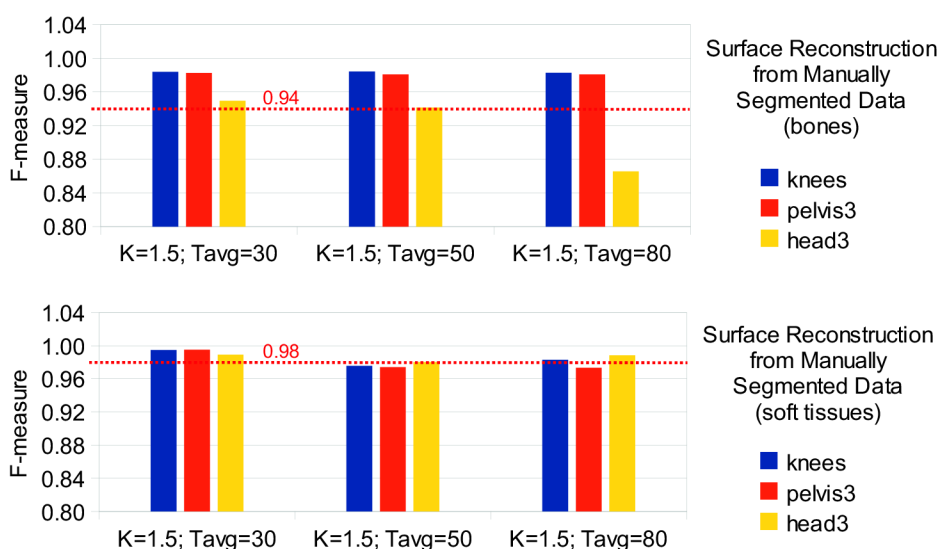


Figure 7.14: Surface reconstruction error when meshing already pre-segmented data. The red line implies the error observed when several people labeled a same CT dataset (see Tab. 7.3).

The measured average error of tested surfaces 0.96 represents approximately an upper limit (= best possible value) one can get when using the VSeg method. Results show that the error depends on initial setting of the meshing, it generally grows for meshes with larger tetrahedra. However, if adequate meshing parameters were chosen, the value is almost the same as the error, or variations, produced by different people when segmenting a same dataset (Tab. 7.3).

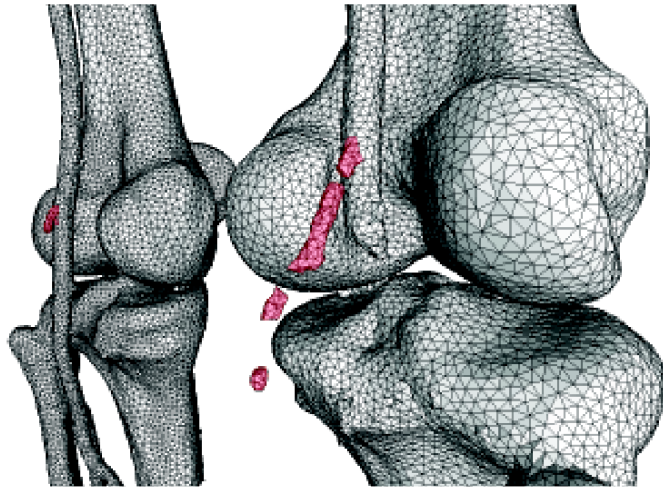


Figure 7.15: Surfaces reconstructed from pre-segmented data. In the red areas of the surface, small anatomical structures are weakly approximated because their size is relatively small compared to a prescribed minimal edge length.

In practice, large portion of this error is caused by limitations of the meshing process. All image structures smaller than the chosen minimal edge length L_{min} are lost. The mesh cannot approximate structures so small (Fig. 7.15).

7.3.4 Segmentation of Medical CT Data

In the last experiment, three different unsupervised clustering techniques (namely *FCM*, *GMM+EM* and *Min-Cut/Max-Flow*) were applied to meshes to classify tetrahedra into individual regions/segments. Analogous to the previous section, all meshes were rasterized into a volume of the same size as the original data. An error between the rasterized mesh and manually segmented data (the ground-truth) was rated by the F-measure (7.3). The calculated error implies how precisely the classified mesh approximates the manually segmented data.

Fig. 7.16 recapitulates results of the mesh segmentation. All parameters of the meshing phase were experimentally set to optimal values (most often $K = 1.5$, $T_{avg} = 50$ and $L_{min} = 1.5$). As a reference, the *FCM* clustering method was also applied directly to the volumetric data.

Results show that all clustering techniques are able to distinguish soft tissue from the bone tissue. When compared to the manual segmentation, the VSeg method provides

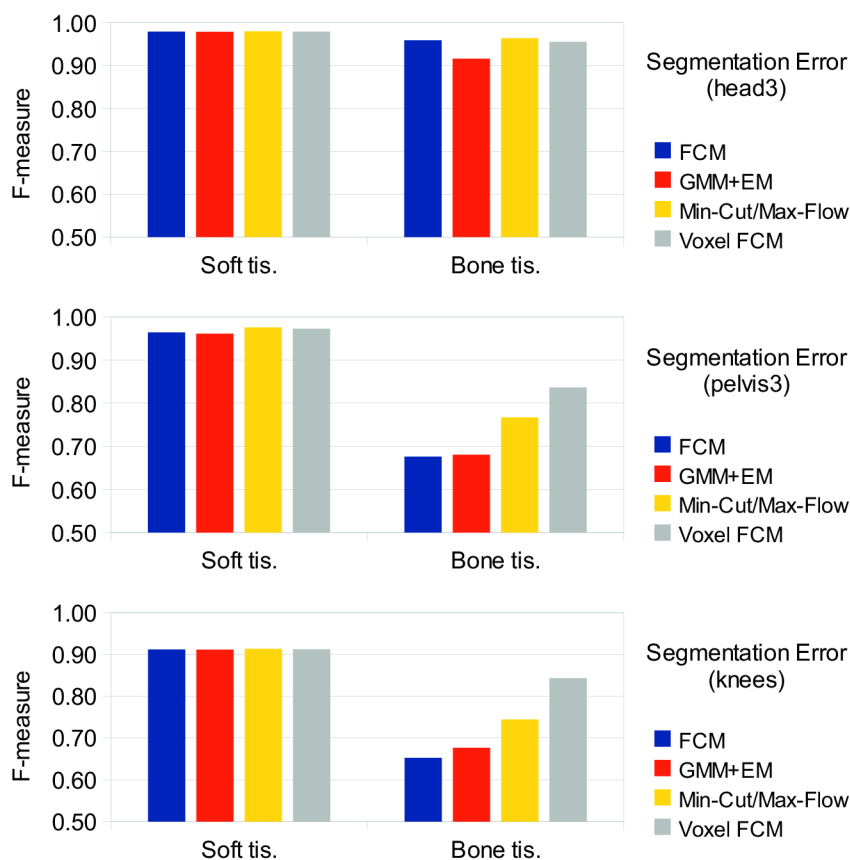


Figure 7.16: Overall segmentation error of the VSeg method. Three alternative clustering methods (*FCM*, *GMM+EM* and *Min-Cut/Max-Flow*) are compared to the straight *FCM* clustering of volumetric data (*voxel FCM*).

precise segmentation of the same quality as the voxel-based *FCM* clustering of the original image data. The segmentation error of soft tissues is comparable to the variation of manual segmentation of the same dataset by different individuals (Sec. 7.3.1).

Not the worse results occur in case of bone segmentation from the *head3* dataset. The VSeg method still produces quite good results. However, the measured error of the bone tissue segmentation significantly grows (i.e. value of the F-measure decreases) for the two remaining datasets. Only the graph-based *Min-Cut/Max-Flow* algorithm provides reasonable results. Why the VSeg mesh segmentation does not achieve good results for these two datasets?

Due to the thickness of the cortical bone and regarding resolution of CT data, very thin

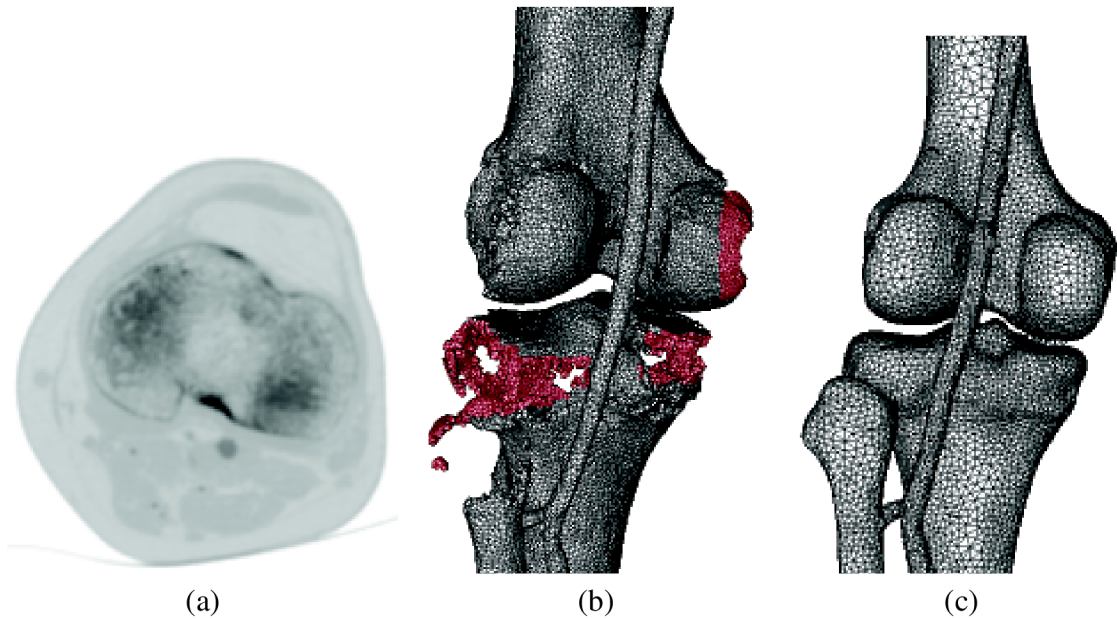


Figure 7.17: Result of the mesh segmentation if weak edges are present in the CT data: the original data – negative (a); surface derived from the classified mesh (b); result of the same method when applied to the manually segmented data (c).

edges are present in the image data which are practically undetectable by conventional edge detection techniques without more knowledge of the data (Fig. 7.17). Therefore, such kind of (non)edges is not well approximated during the meshing process which causes more errors in the final mesh segmentation.

Only the *Min-Cut/Max-Flow* clustering technique is able to partially overcome the problem of missing edges. Because the method takes spatial image structure more into account, results of this graph method overcomes other techniques. Unfortunately, once created the mesh structure cannot be changed, so the final surface is only a rough approximation of the bone tissue. This nature of some medical CT data is also one of the reasons to allow manual corrections of the mesh segmentation.

Several experiments with a texture-based clustering of tetrahedral meshes were also performed. Normalized histogram of simple *LBP* image features (Sec. 6.7.2) forms an extra part of a tetrahedron feature vector. Results of the texture-based mesh segmentation are not presented here. Actually, this extension does not improve results of the classification for the finally selected set of test CT data. Anyhow, for different kind of data (e.g. MRI), the texture-based segmentation may produce qualitatively better results.

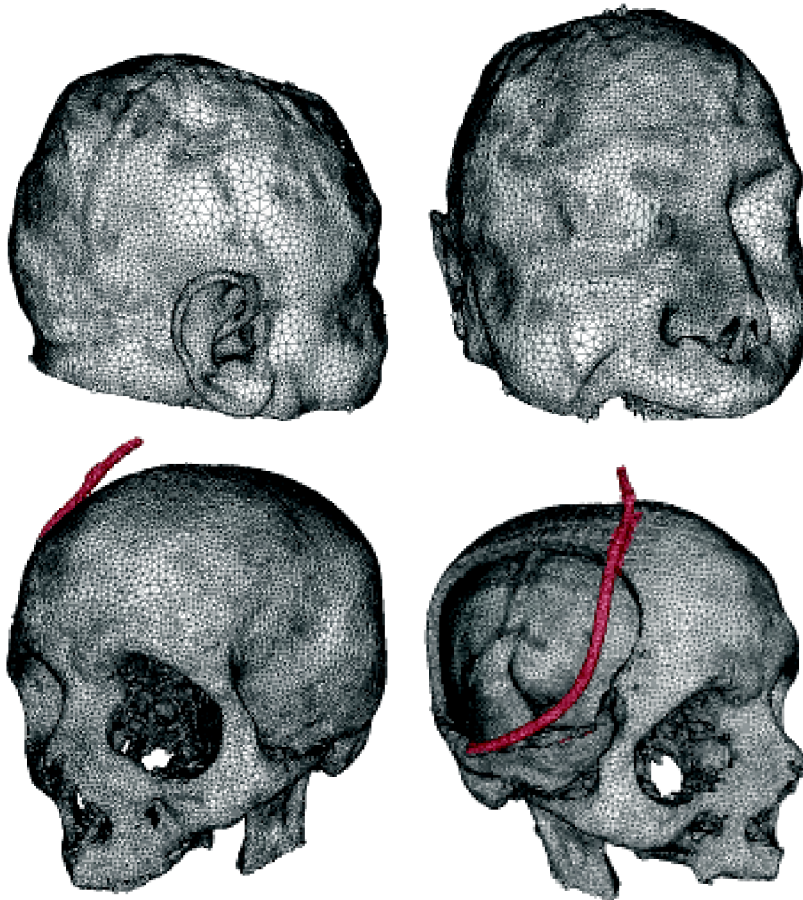


Figure 7.18: Surfaces extracted from the classified mesh – the colored plastic pipe was wrongly classified in to the same region as the bone tissue.

7.4 Runtime Statistics

Basic runtime statistics can be found in Fig. 7.19. The measurement was divided into four stages: *preprocessing* of input data (i.e. anisotropic filtering), *initialization* of the meshing (the edge and corner detection; generation of the control space), *iterative adaptation* of the mesh, and the mesh *segmentation*. All phases take approximately 25 – 50 minutes on a standard PC with Intel Core2Duo 2.54GHz processor depending on a concrete size of the data and specific parameters of the meshing algorithm.

In fact the runtime of the vector segmentation is not very impressive. In comparison with traditional surface reconstruction techniques like the MC algorithm (+ subsequent smoothing), the VSeg method loses. These techniques are able to reconstruct surfaces in

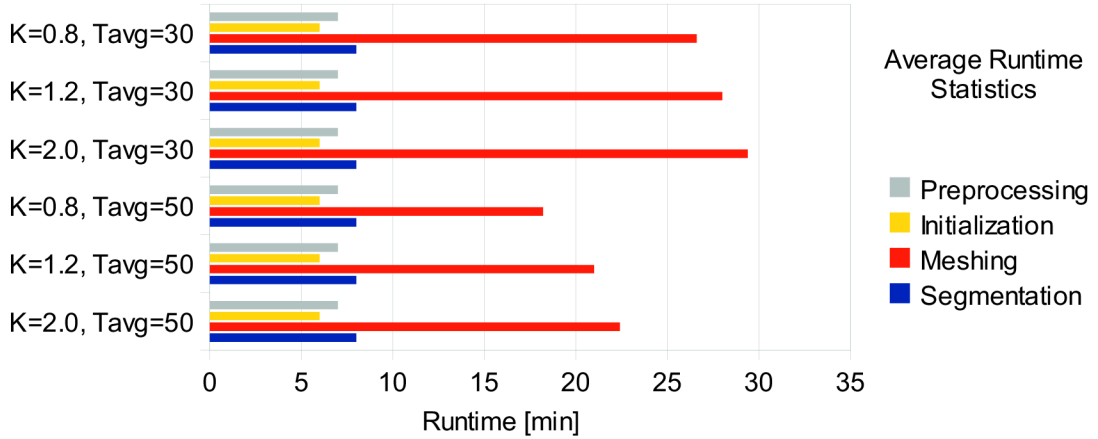


Figure 7.19: Runtime statistics of the VSeg method for different meshing setup. These statistics were measured for the *knees* dataset (512x512x367 voxels).

a much less time – just about minutes. However, such comparison is a bit unfair. Beside the surface, the VSeg method produces more comprehensive representation of the data – tetrahedral mesh – which may be useful for many other tasks.

In addition, surfaces of all desired tissues are reconstructed at once, and any further correction of the segmentation does not lead to a completely new rerun of the reconstruction. This is due to a close relationship between the obtained tetrahedral mesh and all extracted surfaces. Local modifications of the mesh, such as adding of new vertices and reclassification of tetrahedra, can be easily projected onto an already existing surface identifying a set of modified tetrahedron faces.

Aim of the thesis was to prove the concept of volumetric data segmentation based on Delaunay meshing. Not much attention was paid to the optimization of the implementation. The whole segmentation is divided into many blocks which is handy for making experiments, but not very effective due to unnecessary coping of data, repetitive allocations of large memory blocks, etc.

7.5 Summary and Future Work

Different experiments were carried out in order to proof qualities of the proposed vector segmentation approach. A short recapitulation of all results presented above is given here.

Surface accuracy. An error between reconstructed surfaces of human tissues and a *ground truth* must be minimal in order to guarantee the correctness of a surgery planning. In the first test, the accuracy of surface reconstruction was measured on synthetic models of solids. The VSeg method outperformed the both smoothing-based methods (MC+HC and MC+Taubin) for relatively small meshes – the smallest tetrahedra is larger than the voxel. In that case, direct meshing of volumetric image data is more accurate approach than the post-processing methods smoothing reconstructed surfaces without any relationship to the original image data. However, close to sharp surface edges, the final mesh approximates the surface very roughly because of the limitation of tetrahedra size – the minimal tetrahedron edge length is prescribed. Therefore, the maximal error is greater for surfaces obtained by the VSeg method than the ones produced by other approaches.

Mesh quality. The VSeg approach produces well shaped triangles along the entire surface, while the size of triangles is automatically adjusted according to the local complexity of the surface. However, the iterative mesh adaptation scheme was designed to produce high quality tetrahedral meshes suitable for many practical applications, not only the surface reconstruction of a desired tissue, but also for simulations that take an internal structure of tissues into account.

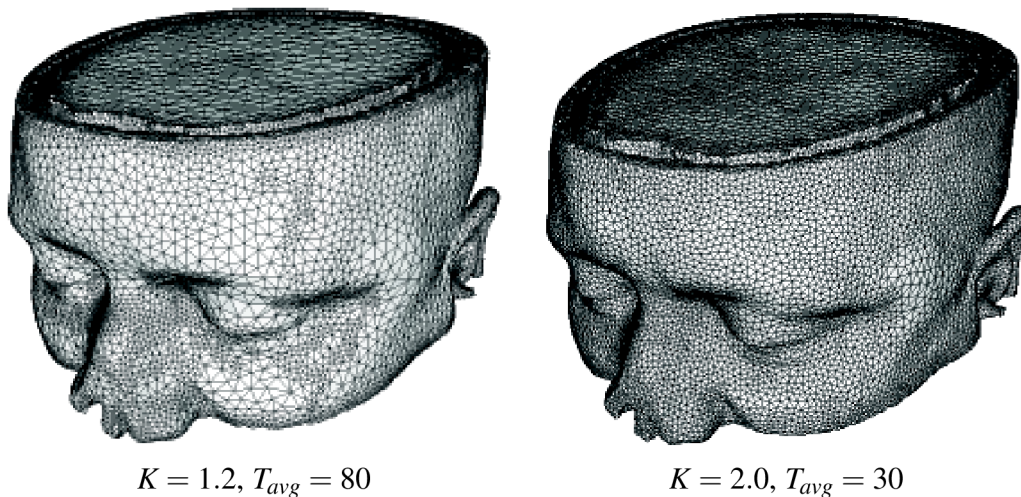


Figure 7.20: Influence of the meshing parameters K and T_{avg} to the structure of final surfaces.

An inconvenience of the described meshing technique is that the quality of tetrahedra decreases as getting closer to the boundaries and slivers appear close to the boundaries.

Hence, the sliver elimination algorithm has been incorporated into the iterative meshing phase. This modification successfully removes a large number of slivers, however it does not ensure that all slivers will be successfully eliminated. This inconvenience of the presented VSeg technique should be further improved.

Mesh segmentation. The overall VSeg approach including the mesh segmentation phase was applied on selected CT datasets, aimed at surface modeling of soft and bone tissues. Results of three different tetrahedra clustering techniques have been compared against manually segmented ground truth. Among the methods, the graph-based *Min-Cut/Max-Flow* clustering technique gives best results. In general, graph clustering techniques appear to be very promising as the mesh structure of clustered data suits well to the idea of represent volume data using a graph structure.

Only the *Min-Cut/Max-Flow* clustering technique is able to deal with the problem of weak edges in the image data (i.e. very thin cortical bone). The final surface is only a rough approximation of the bone tissue because the mesh structure does not approximate (missing) image edges well. In future, the problem of missing edges should be resolved in several ways. A set of manually annotated edges can be defined before the meshing process starts, or the mesh structure can be locally modified after the automatic meshing finishes. Another improvements can be made by incorporating more sophisticated edge detection techniques in the VSeg approach. In spite of that, manual corrections will be always necessary due to the unpredictable character of medical data in case of traumatic injury, which is unfortunately the case when anatomical modeling of tissues should help in a surgery planning.

In conclusion, the proposed vector segmentation can be successfully used for surface reconstruction of desired tissues, as well as for meshing of the interior structure of the tissues for the numerical simulation. Obtained results show that the current concept works very well for certain CT data and is applicable to anatomical modeling of a human skull or soft tissues (i.e. craniectomy in case of traumatic brain injury, or cranioplasty). For the purpose of plain surface reconstruction from an already pre-segmented data, the VSeg method produces surfaces of more than reasonable quality and can be used as is.

Anyhow, I believe that the concept of vector segmentation is quite universal, thus applicable to a wider variety of problems whenever volumetric data of some kind are processed. The only limitation of the present concept is that the meshing phase is based on edge detection.

During elaboration of this work, a new framework for volumetric data processing, tetrahedral meshing and Delaunay-based 2D/3D segmentation has been developed. This framework is freely available for research purposes [116].

Chapter 8

Conclusion

This dissertation thesis presents a novel technique for segmentation of volumetric medical images aimed at surface reconstruction of fundamental human tissues (i.e. bone and soft tissues). This technique of *vector segmentation* is based on the 3D Delaunay triangulation. Tetrahedral mesh is used to divide volumetric data into several disjoint regions whose characteristics are similar. All tetrahedra in the mesh are classified into individual image regions by means of clustering. Finally, certain methods for improving quality of the mesh and its adaptation to the underlying image structure have also been presented.

Such direct meshing of volumetric image data appears to be more accurate approach than traditional techniques which start with an iso-surface extraction followed by the surface decimation and smoothing without any relationship to the original image data. Nevertheless, this idea of the segmentation has several other advantages.

A more effective representation of the image structure is obtained which approximates the original raster data. The mesh representation decreases complexity of the subsequent segmentation because of processing a reduced number of tetrahedra instead of a large number of voxels. The obtained tetrahedral representation is also suitable for numerical simulations that take the internal structure of tissues into account.

If the original image data are affected by artifacts, or the data structure is too complicated due to traumatic injury, manual corrections of the segmentation can be directly applied to the reconstructed surface. It is not necessary to restart the whole meshing processes. There is a close relationship between the obtained tetrahedral mesh and all extracted surfaces. Actually, surface triangles are faces of tetrahedra in the mesh. Simple modifications of the mesh, such as adding new vertices, removing old ones, or manual reclassification of tetrahedra, affect the mesh locally, thus these changes can be directly

projected onto an already extracted surface of the desired tissue.

The concept of vector segmentation is quite general and presented experiments demonstrate its functionality for specific type of CT data. After several minor modifications, it is applicable to a wider variety of problems whenever volumetric data are processed.

However, several inconveniences can be still found in the method that are not very favourable from the practical point of view. Even though the quality of reconstructed surfaces is sufficient for many applications, the quality of produced tetrahedral meshes is not as good as it could be. Slivers still appear close to region boundaries, and the method does not guarantee minimum dihedral angle in the mesh.

Another disadvantage can be found in the edge detection step which is crucial for precise approximation of image boundaries. The proposed tissue-selective edge detection works well for selected CT data. Many parameters of the detection must be tuned to provide desirable results for other type of CT data. The edge detection limits potential application of the method in other research fields when different kind of volumetric data is used.

These aspects of the proposed vector segmentation technique should be addressed in the future work. In addition, many sophisticated image segmentation and classification techniques exist that may be modified to work with the mesh structure. We would like to correct the misclassification of the segmentation step by incorporating more sophisticated image features modeling spatial properties of particular image regions into the classification process.

Bibliography

- [1] ALLIEZ, P., COHEN-STEINER, D., YVINEC, M., AND DESBRUN, M. Variational tetrahedral meshing. *ACM Trans. Graph.* 24, 3 (2005), 617–625.
- [2] ARIVAZHAGAN, S., AND GANESAN, L. Texture segmentation using wavelet transform. *Pattern Recogn. Lett.* 24 (December 2003), 3197–3203.
- [3] ARIVAZHAGAN, S., GANESAN, L., AND PRIYAL, S. P. Texture classification using gabor wavelets based rotation invariant features. *Pattern Recognition Letters* 27, 16 (2006), 1976 – 1982.
- [4] BANKMAN, I. N., Ed. *Handbook of medical imaging: Processing and Analysis*. Academic Press, Inc., Orlando, FL, USA, 2000.
- [5] BASU, M. Gaussian-based edge-detection methods: A survey. *SMC-C* 32, 3 (August 2002), 252–260.
- [6] BESL, P. J., AND MCKAY, N. D. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 2 (1992), 239–256.
- [7] BETHEL, E. W. High performance, three-dimensional bilateral filtering. *LBNL Paper LBNL-1601E* (2009).
- [8] BOYKOV, Y., AND KOLMOGOROV, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on Pattern Analysis and Machine Intelligence* 26, 9 (September 2004), 1124–1137.
- [9] BOYKOV, Y., VEKSLER, O., AND ZABIH, R. Efficient approximate energy minimization via graph cuts. *IEEE transactions on Pattern Analysis and Machine Intelligence* 20, 12 (November 2001), 1222–1239.

- [10] BREDNO, J., LEHMANN, T. M., AND SPITZER, K. A general discrete contour model in two, three, and four dimensions for topology-adaptive multichannel segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 5 (2003), 550–563.
- [11] BUCHOLZ, R. D., AND HENDERSON, J. M. *Journal of computer aided surgery*. Willey InterScience, 2002.
- [12] BULLITT, E., AND AYLWARD, S. R. Volume rendering of segmented image objects. *IEEE Transactions on Medical Imaging* 21 (2002), 200–2.
- [13] CANNY, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 6 (1986), 679–698.
- [14] CAVALCANTI, P. R., AND MELLO, U. T. Three-dimensional constrained delaunay triangulation: A minimalist approach. In *Proceedings of the 8th International Meshing Roundtable* (Sandia National Laboratories, Lake Tahoe, California, USA, October 1999), pp. 119–129.
- [15] CENTRE, I. Medical and biological imaging, 2010. <http://www.bath.ac.uk/elec-eng/invert/medical.html>.
- [16] CHANG, D.-C., AND WU, W.-R. Image contrast enhancement based on a histogram transformation of local standard deviation. *IEEE transactions on medical imaging* 17, 4 (1998), 518–531.
- [17] CHENG, S. W., DEY, T. K., EDELSBRUNNER, H., FACELLO, M. A., AND TENG, S. H. Sliver exudation. In *In Proc. 15th ACM Symp. Comput. Geom.* (1999), pp. 1–13.
- [18] CHENG, S.-W., DEY, T. K., RAMOS, E. A., AND RAY, T. Quality meshing for polyhedra with small angles. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry* (New York, NY, USA, 2004), ACM, pp. 290–299.
- [19] CHENG, S.-W., DEY, T. K., AND RAY, T. Weighted delaunay refinement for polyhedra with small angles. In *Proceedings of the 14th International Meshing Roundtable* (2005), pp. 325–342.

- [20] CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. Metro: measuring error on simplified surfaces. In *Computer Graphics Forum* (1998), vol. 17, Blackwell Publishers, pp. 167–174. <http://vcg.sf.net/>.
- [21] CNR, V. C. L. I. Meshlab. <http://meshlab.sourceforge.net/>.
- [22] COHEN, L., AND COHEN, I. Finite element methods for active contour models and balloons for 2d and 3d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 11 (1993), 1131–1147.
- [23] COOTES, T. F., EDWARDS, G. J., AND TAYLOR, C. J. Active appearance models. *5th European Conference on Computer Vision 1407* (1998), 484–498.
- [24] COTIN, S., DELINGETTE, H., AND AYACHE, N. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics* 5, 1 (1999), 62–73.
- [25] DARDENNE, J., VALETTE, S., SIAUVE, N., BURAI, N., AND PROST, R. Variational tetrahedral mesh generation from discrete volume data. *Vis. Comput.* 25, 5-7 (2009), 401–410.
- [26] DAVOINE, F., AND CHASSERY, J.-M. Adaptive delaunay triangulation for attractor image coding. Tech. rep., Laboratoire TIMC-IMAG, Equipe Infodis, Institut Albert Bonniot, Faculté de médecine, Domaine de la Merci, France, 1994.
- [27] DELINGETTE, H., AND AYACHE, N. Hepatic surgery simulation. *Commun. ACM* 48, 2 (2005), 31–36.
- [28] DERRAZ, F., BELADGHAM, M., AND KHELIF, M. Application of active contour models in medical image segmentation. In *International Conference on Information Technology: Coding and Computing (ITCC'04)* (Los Alamitos, CA, USA, April 2004), vol. 02, IEEE Computer Society, p. 679.
- [29] DINITZ, Y. Dinitz' algorithm: The original version and even's version. 2006, pp. 218–240.
- [30] DIPPEL, S., STAHL, M., WIEMKER, R., AND BLAFFERT, T. Multiscale contrast enhancement for radiographies: Laplacian pyramid versus fast wavelet transform. *Medical Imaging, IEEE Transactions on* 21, 4 (april 2002), 343–353.

- [31] DROSKE, M., MEYER, B., RUMPF, M., AND SCHALLER, C. An adaptive level set method for medical image segmentation. In *17th International Conference on Information Processing in Medical Imaging* (London, UK, 2001), vol. 2082, Springer-Verlag, pp. 416–422.
- [32] DU, G., AND WANG, D. Tetrahedral mesh generation and optimization based on centroidal voronoi tessellations. *Inter. Journal on Numerical Methods in Engineering* 56, 9 (2003), 1355–1373.
- [33] DU, Q., EMELIANENKO, M., AND JU, L. Convergence of the lloyd algorithm for computing centroidal voronoi tessellations. *SIAM J. Numer. Anal.* 44, 1 (2006), 102–119.
- [34] DU, Q., FABER, V., AND GUNZBURGER, M. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review* 41, 4 (1999), 637–676.
- [35] DUDA, R. O., HART, P. E., AND STORK, D. G., Eds. *Pattern Classification, 2nd Edition*. Wiley-Interscience, New York, NY, USA, 2001.
- [36] ERDMANN, B., KOBER, C., LANG, J., DEUFLHARD, P., ZEILHOFER, H.-F., AND SADER, R. Efficient and reliable finite element methods for simulation of the human mandible. Tech. rep., Hartgewebe-Modellierung, Kloster Banz/Staffelstein, Konrad-Zuse-Zentrum, April 2001.
- [37] FEHR, J., AND BURKHARDT, H. 3d rotation invariant local binary patterns. In *Pattern Recognition* (2008), vol. 29, ICPR, pp. 1–4.
- [38] FLEISHMAN, S., DRORI, I., AND COHEN-OR, D. Bilateral mesh denoising. *ACM Trans. Graph.* 22, 3 (2003), 950–953.
- [39] FORSYTH, D. A., AND PONCE, J. *Computer Vision: A Modern Approach*. Prentice Hall, Pearson Education Inc., Upper Saddle River, USA, January 2003.
- [40] FULKERSON, B., VEDALDI, A., AND SOATTO, S. Class segmentation and object localization with superpixel neighborhoods. In *Proceedings of the International Conference on Computer Vision* (October 2009).
- [41] GARLAND, M., AND HECKBERT, P. S. Surface simplification using quadric error metrics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on*

- Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 209–216.
- [42] GEORGE, P.-L., AND BOROUCAKI, H., Eds. *Delaunay Triangulation and Meshing: Application to Finite Elements*. Hermes Science Publications, Orlando, FL, USA, November 1998.
- [43] GERIG, G., KUBLER, O., KIKINIS, R., AND JOLESZ, F. Nonlinear anisotropic filtering of mri data. *Medical Imaging, IEEE Transactions on* 11, 2 (jun 1992), 221–232.
- [44] GEVERS, T. Adaptive image segmentation by combining photometric invariant region and edge information. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 6 (2002), 848–852.
- [45] GOLDBERG, A. V., AND TARJAN, R. E. A new approach to the maximum flow problem. In *STOC '86: Proceedings of the eighteenth annual ACM symposium on Theory of computing* (New York, NY, USA, 1986), ACM, pp. 136–146.
- [46] GONZALEZ, R. C., AND WOODS, R. E. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
- [47] GRAU, V., MEWES, A. U. J., NIZ, M. A., KIKINIS, R., AND WARFIELD, S. K. Improved watershed transform for medical image segmentation using prior information. *IEEE Transactions on Medical Imaging* 23, 4 (April 2004), 447–458.
- [48] HARIS, K., EFSTRATIADIS, S. N., MAGLAVERAS, N., AND KATSAGGELOS, A. K. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing* 7, 12 (December 1998), 1684–1699.
- [49] HARRIS, C., AND STEPHENS, M. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conference* (1988), pp. 147–151.
- [50] HAUSER, H., MROZ, L., BISCHI, G. I., AND GRILLER, M. E. Two-level volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 7, 3 (July 2001).
- [51] HERMAN, G. T. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Springer Publishing Company, Incorporated, 2009.

- [52] HUANG, Q., AND DOM, B. Quantitative methods of evaluating image segmentation. In *ICIP '95: Proceedings of the 1995 International Conference on Image Processing (Vol. 3)-Volume 3* (Washington, DC, USA, 1995), IEEE Computer Society, p. 3053.
- [53] IBRAHIM, H., AND KONG, N. Brightness preserving dynamic histogram equalization for image contrast enhancement. *Consumer Electronics, IEEE Transactions on* 53, 4 (nov. 2007), 1752–1758.
- [54] ITO, Y., SHIH, A. M., AND SONI, B. K. Reliable isotropic tetrahedral mesh generation based on an advancing front method. In *In Proceedings 13th International Meshing Roundtable, Williamsburg, VA, Sandia National Laboratories* (2004), pp. 95–105.
- [55] JONES, T. R., DURAND, F., AND DESBRUN, M. Non-iterative, feature-preserving mesh smoothing. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM, pp. 943–949.
- [56] KOBBELT, L. P., BOTSCH, M., SCHWANECKE, U., AND SEIDEL, H.-P. Feature sensitive surface extraction from volume data. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 57–66.
- [57] KOLMOGOROV, V., AND ZABIH, R. What energy functions can be minimized via graph cuts? *IEEE transactions on Pattern Analysis and Machine Intelligence* 26, 2 (February 2004), 147–159.
- [58] KOREN, I., LAINE, A., TAYLOR, F., AND LEWIS, M. Interactive wavelet processing and techniques applied to digital mammography. *Acoustics, Speech, and Signal Processing, IEEE International Conference on* 3 (1996), 1415–1418.
- [59] KRŠEK, P. *Direct creating of FEM models from CT/MR data for biomechanics applications*. PhD thesis, Brno University of Technology, Brno, ČR, 2001.
- [60] KRŠEK, P., ŠPANĚL, M., ČERNOCHOVÁ, P., KAŇOVSKÁ, K., KRUPA, P., STOKLAS, J., AND MOLITOR, M. 3d human tissues modeling in clinical applications. In *Medical Information Visualisation* (2006), IEEE Computer Society, p. 1.

- [61] KRYSL, P., AND ORTIZ, M. Variational delaunay approach to the generation of tetrahedral finite element meshes. *International Journal* 50 (1999), 1681–1700.
- [62] KURANI, A. S., HUI XU, D., FURST, J., AND RAICU, D. S. Raicu . co-occurrence matrices for volumetric data. the. In *7th IASTED International Conference on Computer Graphics and Imaging, Kauai* (2004).
- [63] KURITA, T. An efficient agglomerative clustering algorithm for region growing. In *Proc. of IAPR Workshop on Machine Vision Applications, Dec.1315* (1991), pp. 210–213.
- [64] LABELLE, F., AND SHEWCHUK, J. R. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Trans. Graph.* 26, 3 (2007), 57.
- [65] LACHAUD, J.-O., AND MONTANVERT, A. Volumic segmentation using hierarchical representation and triangulated surface. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I* (London, UK, 1996), Springer-Verlag, pp. 137–146.
- [66] LAINE, A., FAN, J., AND YANG, W. Wavelets for contrast enhancement of digital mammography. *Engineering in Medicine and Biology Magazine, IEEE* 14, 5 (sep/oct 1995), 536 –550.
- [67] LEE, K.-W., AND WANG, W.-P. Feature-preserving mesh denoising via bilateral normal filtering. In *CAD-CG '05: Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 275–280.
- [68] LEMKE, H., AND VANNIER, M. International journal of computer assisted radiology and surgery. Journal no. 11548, Springer Berlin Heidelberg.
- [69] LEVENTON, M. E., FAUGERAS, O., GRIMSON, W. E. L., AND WELLS, W. M. Level set based segmentation with intensity and curvature priors. In *Workshop on Mathematical Methods in Biomedical Image Analysis Proceedings* (June 2000), pp. 4–11.
- [70] LI, X. Sliver-free three dimensional delaunay mesh generation. Tech. rep., PH.D THESIS, UIUC, 2000.

- [71] LORENSEN, W. E., AND CLINE, H. E. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.* 21, 4 (1987), 163–169.
- [72] LUNDSTRÖM, C. Segmentation of medical image volumes. Master’s thesis, Linköping University, SE-581 83 Linköping, Sweden, November 1997. LiTH-ISY-EX-1864.
- [73] MA, F., WANG, W., TSANG, W. W., TANG, Z., XIA, S., AND TONG, X. Probabilistic segmentation of volume data for visualization using som-pnn classifier. In *VVS ’98: Proceedings of the 1998 IEEE symposium on Volume visualization* (New York, NY, USA, 1998), ACM Press, pp. 71–78.
- [74] MARROQUIN, J. L., SANTANA, E. A., AND BOTELLO, S. Hidden markov measure field models for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 11 (November 2003), 1380–1387.
- [75] MCINERNEY, T. *Topologically Adaptable Deformable Models for Medical Image Analysis*. PhD thesis, Dept. of Computer Science, University of Toronto, Toronto, On, 1997.
- [76] MCINERNEY, T., AND TERZOPOULOS, D. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1996.
- [77] MCINERNEY, T., AND TERZOPOULOS, D. Medical image segmentation using topologically adaptable surfaces. In *First Joint Conference of Computer Vision, Virtual Reality, and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery (CVRMed-MRCAS’97)* (Grenoble, France, 1997), vol. 1205, Springer-Verlag, pp. 23–32.
- [78] MILLER, J. V., BREEN, D. E., LORENSEN, W. E., O’BARA, R. M., AND WOZNY, M. J. Geometrically deformed models: a method for extracting closed geometric models from volume data. In *SIGGRAPH ’91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1991), ACM Press, pp. 217–226.
- [79] MITCHELL, S. C., BOSCH, J. G., LELIEVELDT, B. P. F., VAN DER GEEST, R. J., REIBER, J. H. C., AND SONKA, M. 3-d active appearance models: segmentation

- of cardiac mr and ultrasound images. *Medical Imaging, IEEE Transactions on* 21, 9 (2002), 1167–1178.
- [80] MOORE, D., AND WARREN, J. Compact isocontours from sampled data. 23–28.
- [81] NG, S.-K., AND MCLACHLAN, G. J. On some variants of the em algorithm for fitting mixture models. *Austrian Journal of Statistics* 23 (2003), 143–161.
- [82] NOVELLINE, R. A. *Squire's Fundamentals of Radiology*. Harvard University Press, 5th edition, Jan 1997.
- [83] O'DONNELL, L., ERIC, W., AND GRIMSON, L. Semi-automatic medical image segmentation. Master's thesis, Massachusetts Institute of Technology, 2001.
- [84] OJALA, T., PIETIKÄINEN, M., AND HARWOOD, D. A comparative study of texture measures with classification based on feature distributions. In *Pattern Recognition* (1996), vol. 29, pp. 51–59.
- [85] OJALA, T., PIETIKÄINEN, M., AND MÄENPÄÄ, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 7 (July 2002), 971–987.
- [86] OWEN, S. A survey of unstructured mesh generation technology. In *Proceedings of the Seventh International Meshing Roundtable* (Dearborn, Michigan, Oct 1998), Sandia National Laboratories.
- [87] PARIS, S., AND DURAND, F. A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vision* 81, 1 (2009), 24–52.
- [88] PARK, G.-H., CHO, H.-H., AND CHOI, M.-R. A contrast enhancement method using dynamic range separate histogram equalization. *Consumer Electronics, IEEE Transactions on* 54, 4 (november 2008), 1981–1987.
- [89] PARK, J., AND MURPHEY, Y. L. Edge detection in grayscale, color, and range images. In *Wiley Encyclopedia of Computer Science and Engineering*. 2008.
- [90] PARK, J. Y., MCINERNEY, T., TERZOPOULOS, D., AND KIM, K. H. A non-self-intersecting deformable surface for complex boundary extraction from volumetric images. *Computers and Graphics* 25, 3 (2001), 421–440.

- [91] PEDNEKAR, A., KAKADIARIS, I. A., AND KURKURE, U. Adaptive fuzzy connectedness-based medical image segmentation. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'02)* (Ahmedabad, India, December 2002), Space Applications Centre (ISRO).
- [92] PERONA, P., AND MALIK, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (1990), 629–639.
- [93] PERONA, P., AND MALIK, J. Detecting and localizing edges composed of steps, peaks and roofs. In *In Proc. 3rd Intl. Conf. Computer Vision* (1991), pp. 52–57.
- [94] PHAM, D. L. Robust fuzzy segmentation of magnetic resonance images. In *14th IEEE Symposium on Computer-Based Medical Systems, CBMS 2001* (Bethesda, MD, USA, July 2001), pp. 127–131.
- [95] PHAM, D. L., AND PRINCE, J. L. Adaptive fuzzy segmentation of magnetic resonance images. *IEEE Transactions on Medical Imaging* 18 (September 1999).
- [96] PRASAD, L., AND SKOURIKHINE, A. N. Vectorized image segmentation via trixel agglomeration. *Pattern Recogn.* 39, 4 (2006), 501–514.
- [97] SHAPIRO, L., AND STOCKMAN, G. *Computer Vision*. Prentice Hall, 2001.
- [98] SHEWCHUK, J. What is a good linear element? interpolation, conditioning, and quality measure. In *In Proc. of 11th Int. Meshing Roundtable* (2002), pp. 115–126.
- [99] SHEWCHUK, J. R. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications* 22 (2002), 21–74.
- [100] SMITH, S. M., AND BRADY, J. M. Susan - a new approach to low level image processing. *Inter. Journal of Computer Vision* (1996).
- [101] STAHL, M., AACH, T., BUZUG, T. M., DIPPEL, S., AND NEITZEL, U. Noise-resistant weak-structure enhancement for digital radiography. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (May 1999), K. M. Hanson, Ed., vol. 3661 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pp. 1406–1417.

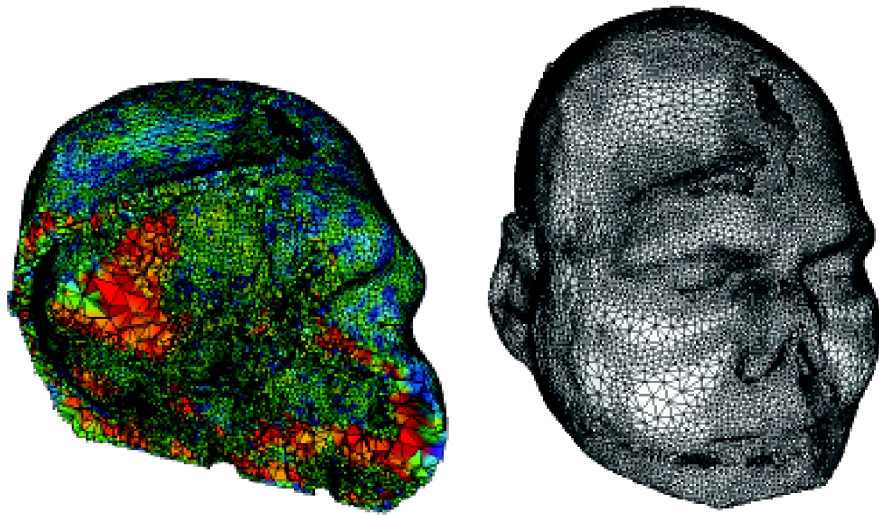
- [102] STEGMANN, M. B. Active appearance models: Theory, extensions and cases. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, August 2000.
- [103] TAUBIN, G., AND TAUBIN, G. Geometric signal processing on polygonal meshes, Sep 2000.
- [104] TOMASI, C., AND MANDUCHI, R. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision* (Washington, DC, USA, 1998), IEEE Computer Society, p. 839.
- [105] TOURNOIS, J., SRINIVASAN, R., AND ALLIEZ, P. Perturbing slivers in 3d delaunay meshes. In *Proceedings of the 18th International Meshing Roundtable* (November 2009), Springer Berlin Heidelberg, pp. 157–173.
- [106] TSAI, F., CHANG, C.-K., RAU, J.-Y., LIN, T.-H., AND LIU, G.-R. 3d computation of gray level co-occurrence in hyperspectral image cubes. In *EMMCVPR'07: Proceedings of the 6th international conference on Energy minimization methods in computer vision and pattern recognition* (Berlin, Heidelberg, 2007), Springer-Verlag, pp. 429–440.
- [107] TUCERYAN, M. Moment-based texture segmentation. *Pattern Recognition Letters* 15, 7 (1994), 659 – 668.
- [108] UNNIKRISHNAN, R., PANTOFARU, C., AND HEBERT, M. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 6 (2007), 929–944.
- [109] UNSER, M., ALDROUBI, A., AND LAINE, A. Guest Editorial: Wavelets in Medical Imaging. *IEEE Transactions on Medical Imaging* 22, 3 (2003), 285–288.
- [110] VAN RIJSBERGEN, C. J. *Information Retrieval 2nd Ed.* Butterworths, London, 1979.
- [111] ČERNOCHOVÁ, P., KAŇOVSKÁ, K., KRŠEK, P., AND KRUPA, P. Application of geometric biomodels for autotransplantation of impacted canines. In *World Journal of Orthodontics 2005* (2005), p. 1.

- [112] VEKSLER, O. Gcmex - matlab wrapper for graph cuts multi-label energy minimization, 2010. <http://vision.ucla.edu/brian/gcmex.html>.
- [113] VIVODTZEV, F., BONNEAU, G.-P., LINSEN, L., HAMANN, B., JOY, K. I., AND OLSHAUSEN, B. A. Hierarchical isosurface segmentation based on discrete curvature. In *VISSYM'03: Proceedings of the symposium on Data visualisation 2003* (Aire-la-Ville, Switzerland, 2003), Eurographics Association, pp. 249–258.
- [114] VOLLMER, J., MENCL, R., AND MLLER, H. Improved laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum* (Sep 1999), vol. 18, Blackwell Publishing, pp. 131–138.
- [115] ŠPANĚL, M., AND KRŠEK, P. Vector-based medical image segmentation using adaptive delaunay triangulation. In *Proceedings of the Sixth IASTED International Conference on Visualization, Imaging, and Image Processing* (2006), 541, ACTA Press, p. 6.
- [116] ŠPANĚL, M., KRŠEK, P., ŠVUB, M., AND ŠTANCL, V. Mdstk - medical data segmentation toolkit, 2010. <http://sourceforge.net/projects/mdstk/>.
- [117] ŠVUB, M., KRŠEK, P., ŠPANĚL, M., ŠTANCL, V., BARTOŇ, R., AND VAĐURA, J. Feature preserving mesh smoothing algorithm based on local normal covariance. In *Proceedings of WSCG'10* (2010), 17-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, University of West Bohemia in Pilsen, p. 6.
- [118] WATT, A., AND POLICARPO, F., Eds. *The Computer Image*. Addison Wesley, New York, USA, November 1998.
- [119] WIKIPEDIA. X-ray computed tomography, Jun 2010. http://en.wikipedia.org/wiki/X-ray_computed_tomography.
- [120] WILLIAMS, D. J., AND SHAH, M. A fast algorithm for active contours and curve estimation. *CVGIP: Image Understanding* 55 (January 1992), 14–26.
- [121] WORTH, A. J. Brain segmentation in mri, 2010. <http://www.cma.mgh.harvard.edu/seg/>.

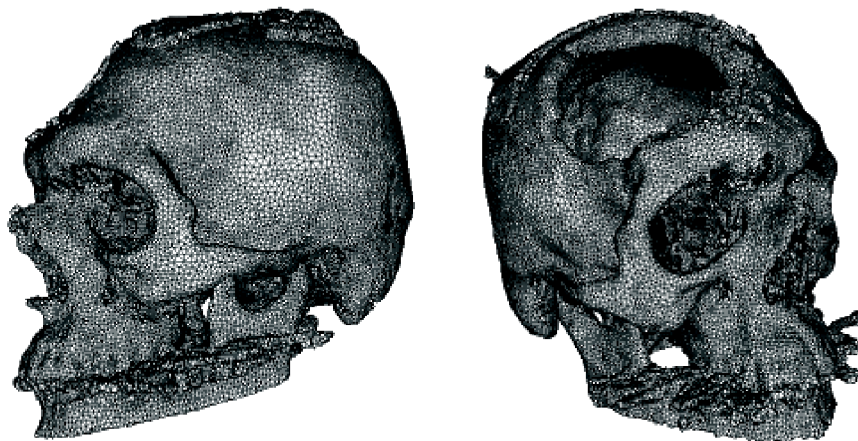
- [122] YANG LI, X., TENG, S.-H., NGR, A., AND HUA TENG ALPER, S. Biting: Advancing front meets sphere packing. In *Int. Jour. for Numerical Methods in Eng* (1999).
- [123] YOSHIZAWA, S., BELYAEV, A., AND YOKOTA, H. Fast gauss bilateral filtering. *Computer Graphics Forum* 29, 1 (2010), 24–52.
- [124] ZHANG, J. Reconstruction of geometry from cardiac mr images. *CVGIP: Image Understanding* 55 (January 1992), 14–26.
- [125] ZHANG, Y., AND BAJAJ, C. Adaptive and quality quadrilateral/hexahedral meshing from volumetric imaging data. In *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING* (2006).
- [126] ZHANG, Y., BAJAJ, C., AND SOHN, B.-S. Adaptive and quality 3d meshing from imaging data. In *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications* (New York, NY, USA, 2003), ACM, pp. 286–291.
- [127] ZHANG, Y., BAJAJ, C., AND SOHN, B.-S. 3d finite element meshing from imaging data. *Computer Methods in Applied Mechanics and Engineering* 194, 48-49 (2005), 5083 – 5106. Unstructured Mesh Generation.

Sample Results

Input dataset: *CT-head2*; meshing parameters: $K = 1.5$, $T_{avg} = 50mm$, and $L_{min} = 1.5mm$.

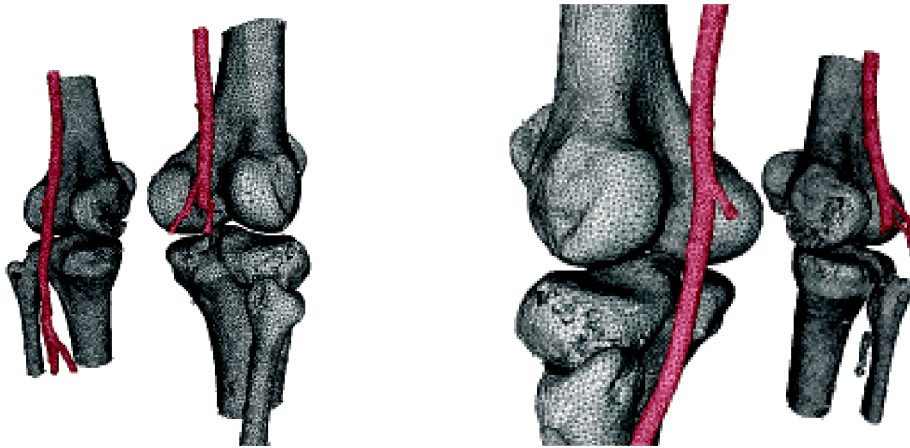


82885 vertices, 165348 faces

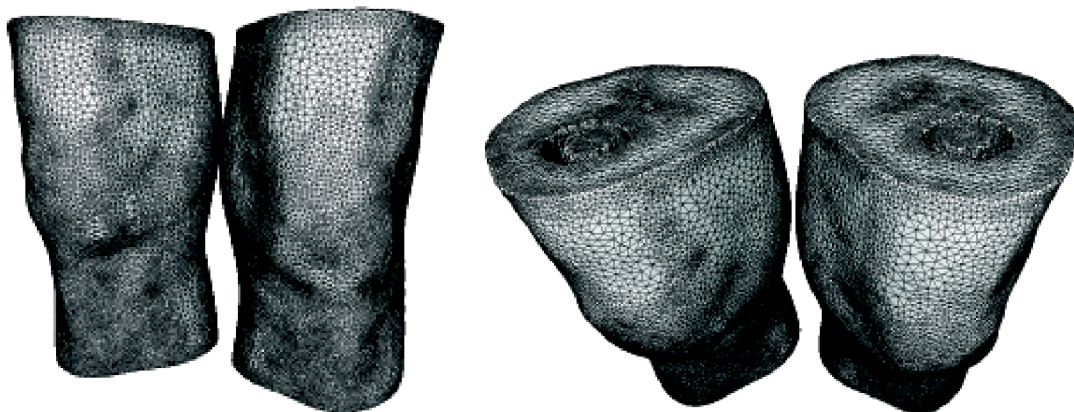


127528 vertices, 256060 faces

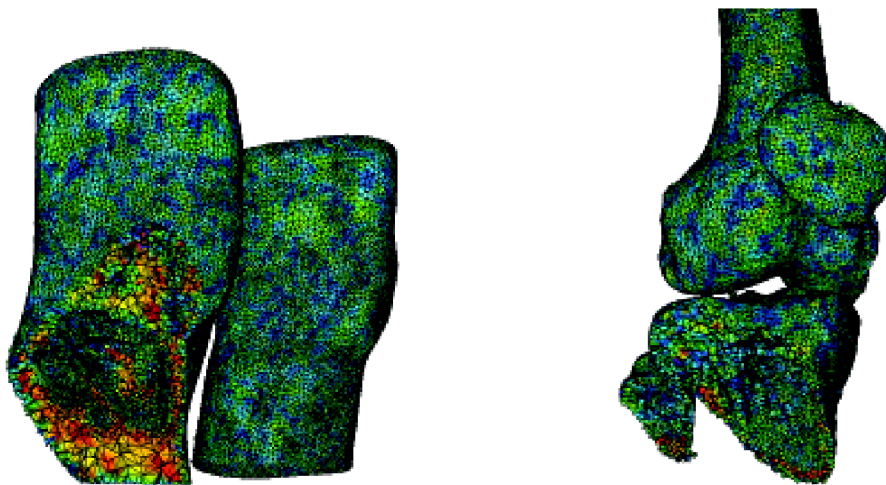
Input dataset: *CT-knees*; meshing parameters: $K = 1.5$, $T_{avg} = 30mm$, and $L_{min} = 1.0mm$.



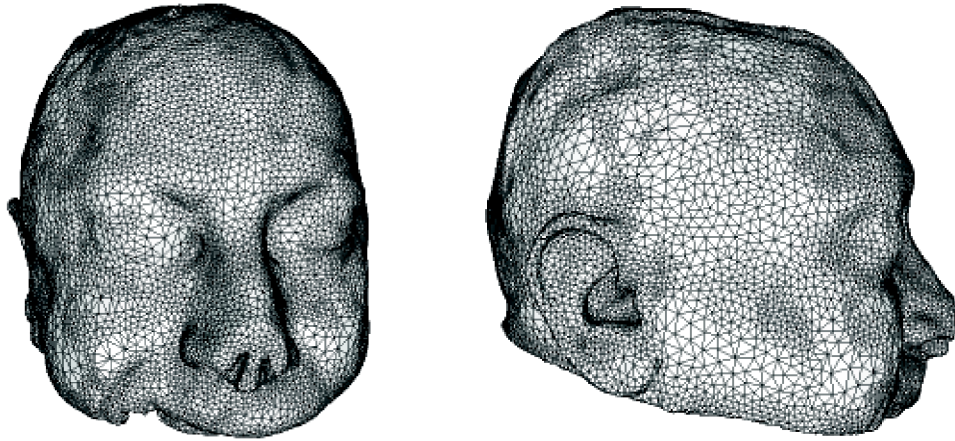
125444 vertices, 250518 faces



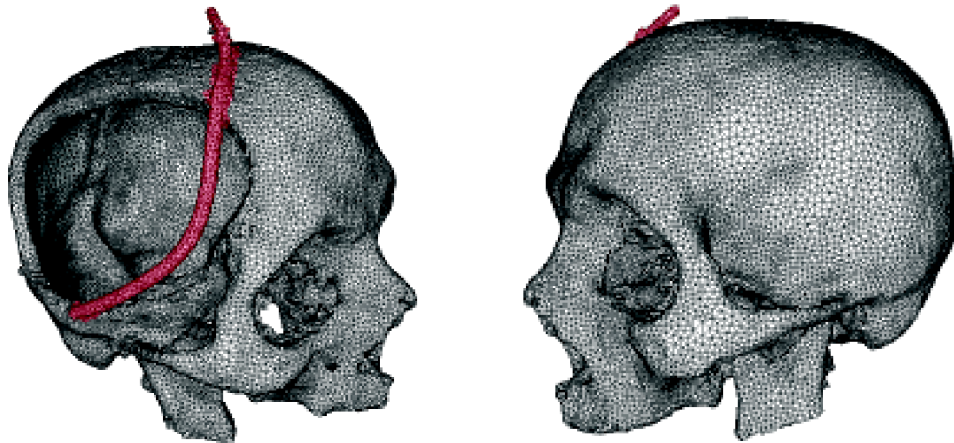
138883 vertices, 136266 faces



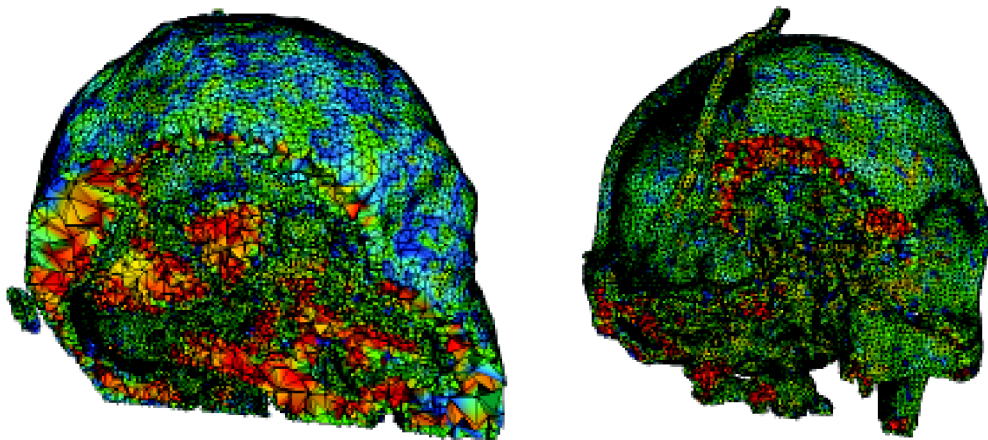
Input dataset: *CT-head3*; meshing parameters: $K = 1.5$, $T_{avg} = 50mm$, and $L_{min} = 1.5mm$.



89131 vertices, 178124 faces



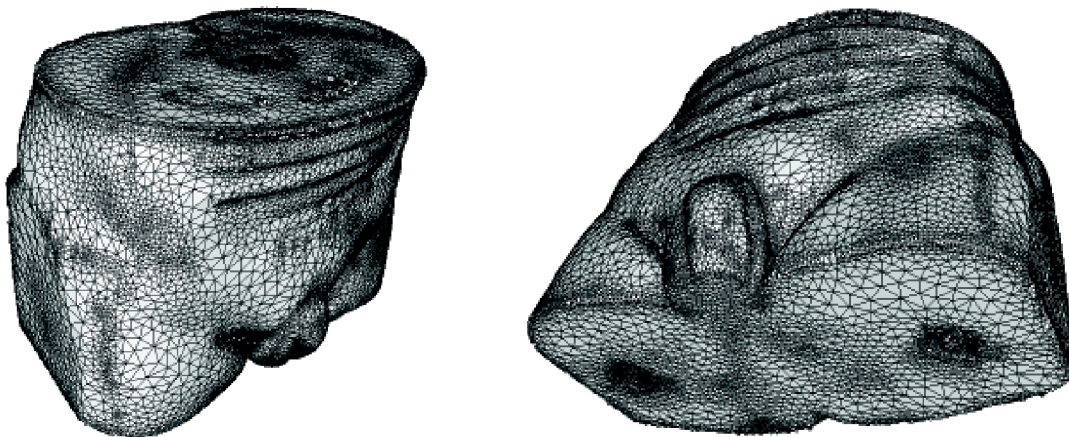
87833 vertices, 176318 faces



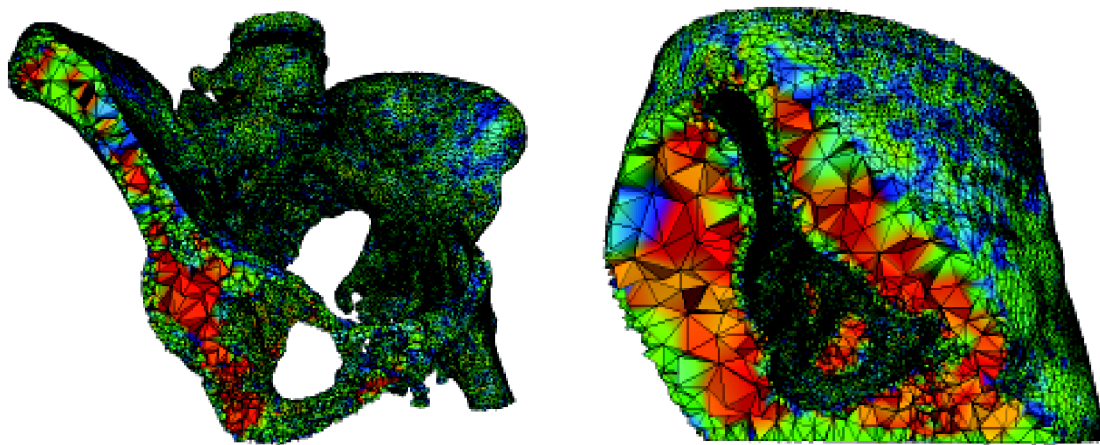
Input dataset: *CT-pelvis3*; meshing parameters: $K = 1.5$, $T_{avg} = 50mm$, and $L_{min} = 1.5mm$.



77033 vertices, 154356 faces



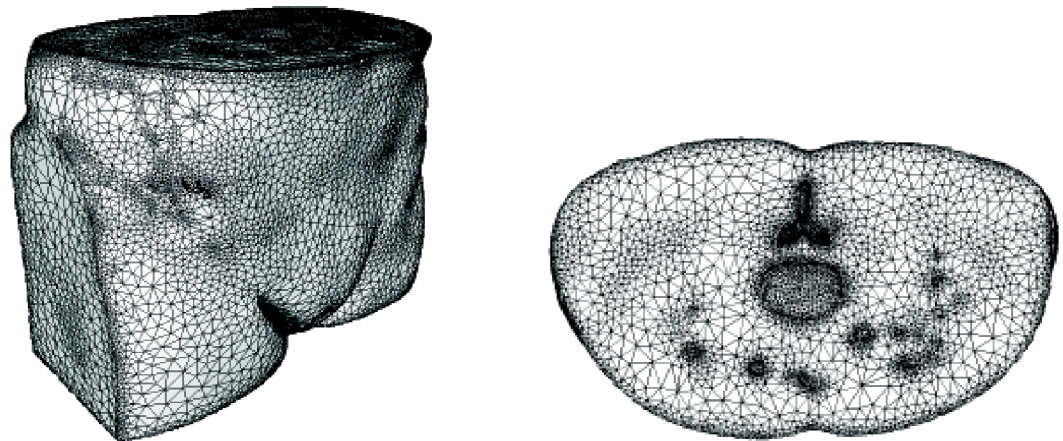
105758 vertices, 211320 faces



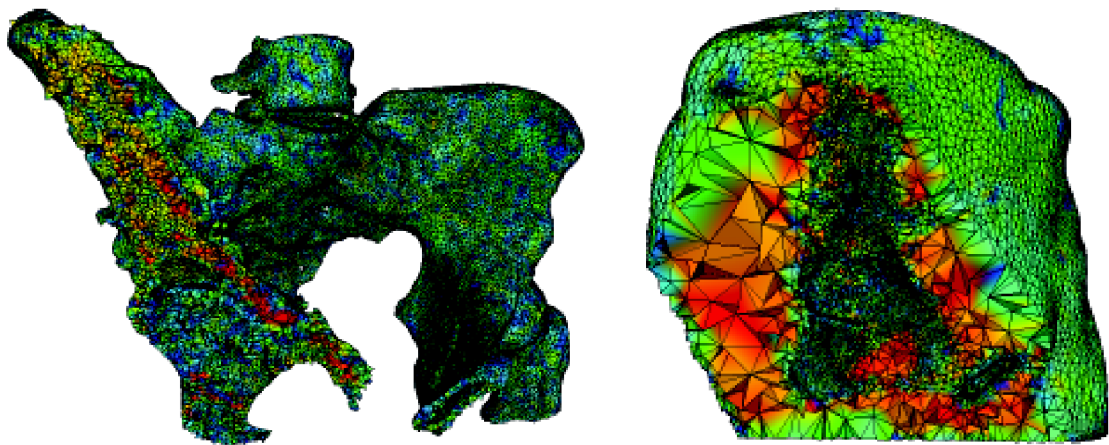
Input dataset: *CT-pelvis2*; meshing parameters: $K = 0.8$, $T_{avg} = 50mm$, and $L_{min} = 1.5mm$.



83938 vertices, 168308 faces



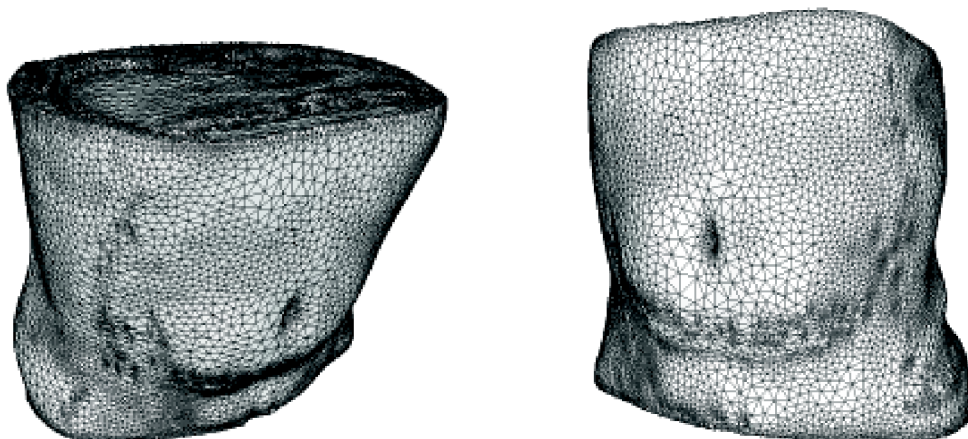
118850 vertices, 238164 faces



Input dataset: *CT-arteries*; meshing parameters: $K = 0.8$, $T_{avg} = 50mm$, and $L_{min} = 1.5mm$.



100613 vertices, 202084 faces



172915 vertices, 347336 faces

