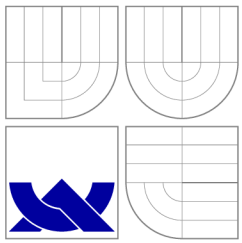
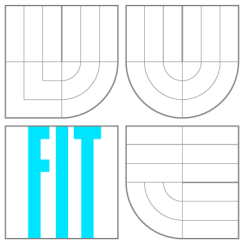


BRNO UNIVERSITY OF TECHNOLOGY

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 — Ing. Michal Španěl

AUTOR PRÁCE

SUPERVISOR — Doc. Ing. Přemysl Kršek, Ph.D.

VEDOUCÍ PRÁCE

BRNO 2010

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 proposed. 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.

Contents

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.

Chapter 2
Thesis Objectives

This thesis aims at the anatomical modeling of human tissues and techniques of medical image segmentation suitable for this kind of modeling. 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 [20, 39] 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 use 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 preprocessing 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

State of the Art in Anatomical Modeling

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.).

This classification is not very common, however, it makes a good sense in reference to the geometric modeling.

3.1 Raster-based Segmentation Techniques

Many 2D/3D segmentation algorithms can be found in the literature (Fuzzy C-means clustering [31], Hidden Markov Fields [26], Watershed transform [17], neural networks, etc.). Most of them produce segmented raster data, hence an algorithm such as *Marching Cubes* [25] is applied to reconstruct surfaces from the raster segmented data. Further, decimation and smoothing [34] of the model are required and may not be elementary. Applied smoothing and decimation methods may not shrink significant edges and corners and they must preserve volume of the original model.

3.2 Vector-based Segmentation

Most widely used vector segmentation methods are based on deformable models [42]. Deformable models include curves or solids deformed under influence of external and internal forces derived from image characteristics. Numerous researchers have explored application of deformable surface models to volumetric medical images [5, 24]. A deformable surface model capable of segmenting complex internal organs such as the cortex of the brain has been proposed [27].

There is also a second type of deformable models – *the geometric models*, best known is the *Level–Set* method. The level-set segmentation [**?**, **?**] 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.

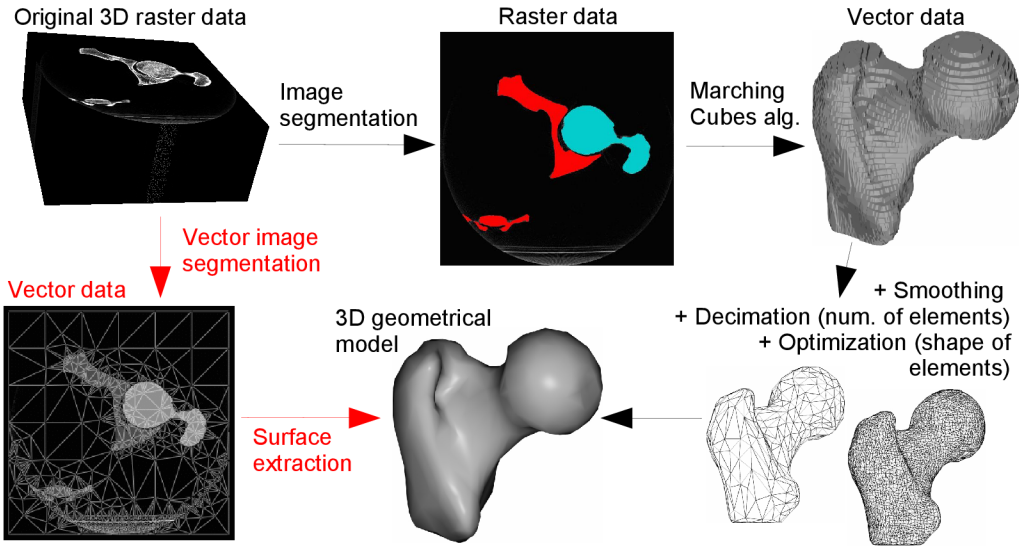


Figure 3.1: Comparison of the traditional raster-based segmentation (black labeling) and the proposed vector segmentation method (underlined).

In general, 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.

3.3 Unstructured Meshing

A mesh generation [14] aims at tessellation of a bounded 3D domain Ω with tetrahedra. Algorithms for 3D mesh generation have been intensively studied over the last years. Basically, three main families of algorithms have been described:

- octree methods [43,44],
- advancing front methods [19,41],
- and Delunay-based methods [1,7].

Zhang *et al.* [43] presented an algorithm to extract adaptive and quality 3D meshes directly from volumetric image 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.

The *isosurface stuffing* algorithm [23] was presented that fills an iso-surface with a uniformly sized tetrahedral mesh. 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

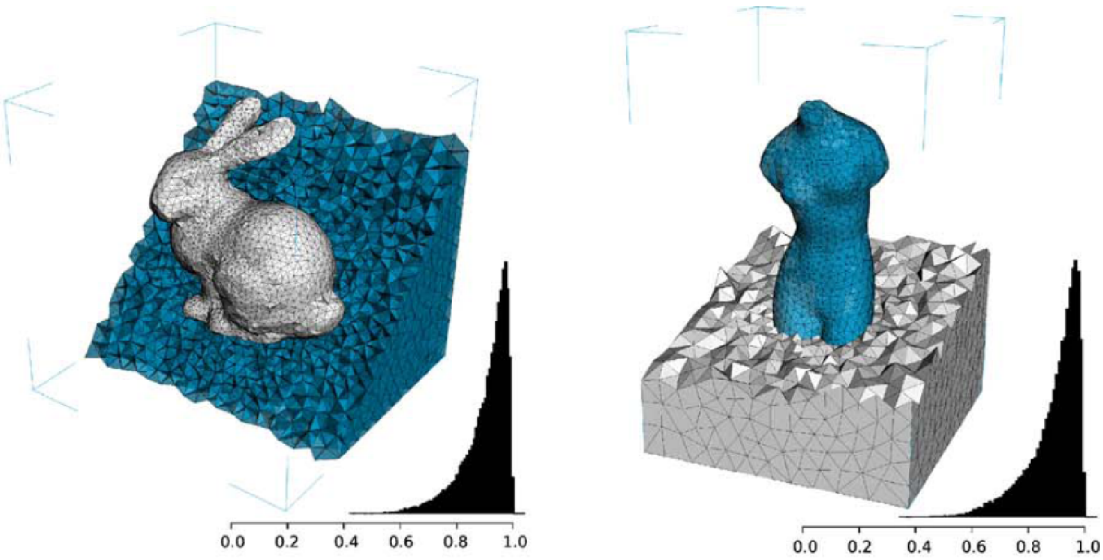


Figure 3.2: Results obtained by the approach of Dardenne *et al.* from artificial discrete data – rasterized input surface models. Histograms show the distributions of tetrahedron quality in each mesh [10].

of the algorithm creates a mesh with internal grading. However, the algorithm does not permit grading of both surface and interior tetrahedra and has a strong bound on the dihedral angles.

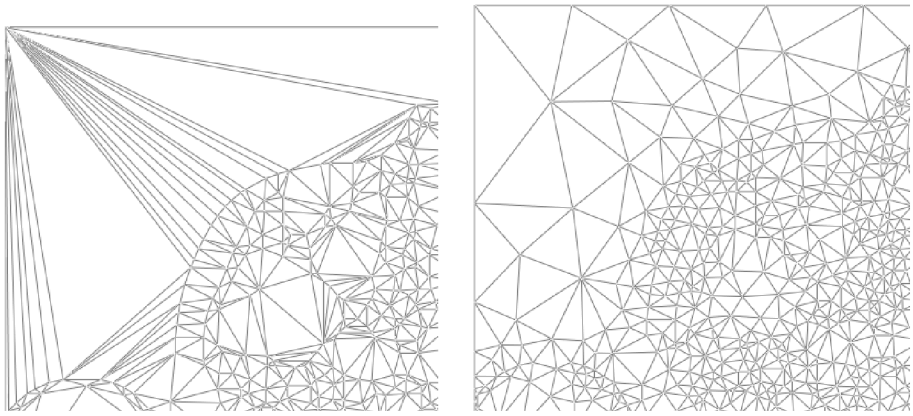
Variational approaches relying on energy minimization have been presented as a powerful and robust tool in meshing. These methods basically define energies that they minimize through vertex displacements and/or connectivity changes in the current mesh. Du and Wang [11] propose to generate meshes that are dual to optimal Voronoi diagrams. The centroidal Voronoi tessellation [12] based Delaunay triangulation provides an optimal distribution of generating points with respect to a given density function and generates a high-quality mesh. Following Du and Wang, another tetrahedral mesh generation algorithm based on centroidal Voronoi tesselation, which takes volumetric segmented data as an input, has been presented [10]. 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.

In this thesis, a vector segmentation technique based on the *variational tetrahedral meshing* (VTM) approach, proposed by Alliez *et al.* [1], is presented. The VTM approach uses a simple quadratic energy to optimize vertex positions within the mesh 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.

Chapter 4

Background: Tetrahedral Meshing

Many 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* [14], will be discussed here.



(a) (b)

Figure 4.1: Triangular mesh constructed by the plain incremental method [14] (a) and result of the isotropic meshing (b).

According to [14], 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:

$$l_H(AB) = \|AB\| \frac{\frac{1}{h(A)} + \frac{1}{h(B)}}{2}, \tag{4.1}$$

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

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, until we have a sequence of points $Q_0 \dots Q_n$ such that

$$l_H(Q_i, Q_{i+1}) < T, \tag{4.2}$$

where $Q_0 = A$ and $Q_n = B$. Afterwards, the final set of points dividing the edge AB can be found. The smallest index i satisfying the criterion (4.3) 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.

$$\sum_{j=0}^{i} l_H(Q_j, Q_{j+1}) > 1 \tag{4.3}$$

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.

4.1 Variational Meshing

Many approaches based on energy minimization [1, 11, 21] have been proposed as a powerful tool in meshing. In this thesis, a vector segmentation technique, built upon the VTM approach, is presented. A simple minimization procedure alternates two steps [1]:

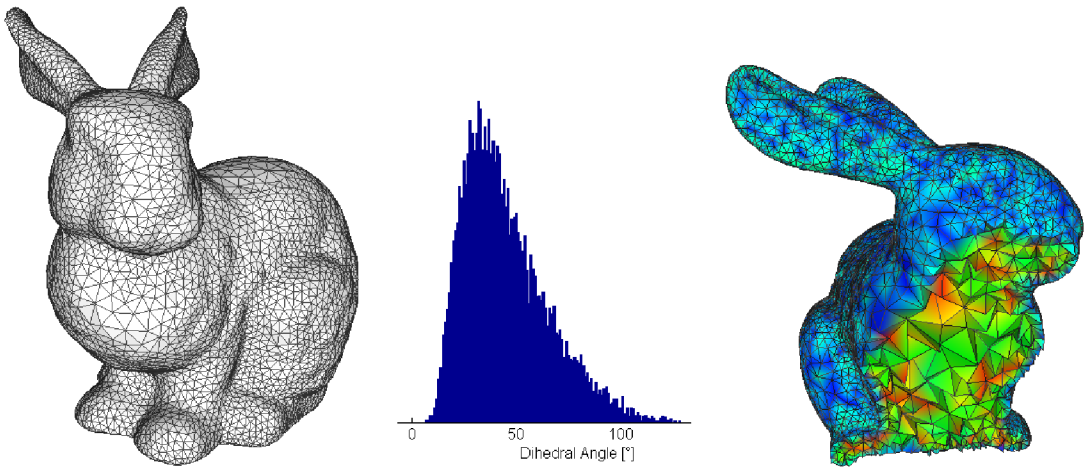
- global *3D Delaunay triangulation optimizing* connectivity,
- and *local vertex relocation*,

to consistently and efficiently minimize a global energy over the domain. It results in a robust meshing technique that generates high quality meshes in terms of minimal dihedral angles.

To extend the approach to allow isotropic meshing, the *sizing field H* is introduced. 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.* 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. In geometric terms, the optimal position of the interior vertex X_i in its 1–ring neighbourhood can be expressed as:

$$X_i^* = \frac{1}{\sum_{T_k \in \Omega_i} \frac{|T_k|}{h^3(G_k)}} \sum_{T_j \in \Omega_i} \frac{|T_j|}{h^3(G_j)} c_j. \tag{4.4}$$

where G_k is the centroid of tetrahedron T_k and c_j is the circumcenter of tetrahedra T_j. Alliez *et al.* presented a default sizing field robust for a large spectrum of mesh types. Definition of the siz-



(a) (b) (c)

Figure 4.2: Results of the proposed vector segmentation method which is based on variational tetrahedral meshing approach [1]: surface extracted directly from the tetrahedral mesh (a); histogram of minimal dihedral angles on the surface (b); cut through the tetrahedral mesh (c).

ing field is built on the notion of *local feature size* that corresponds to the combination of domain *boundary curvature* and *thickness* as well.

Sizing field. 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

$$h_P = \min_{S \in \delta\Omega} [Kd(S,P) + lfs(S)] \tag{4.5}$$

satisfies this criterion [1]. The parameter K controls gradation of the resulting field, $K = 0$ being the uniform case.

Chapter 5

Delaunay-based Vector Segmentation

Based on the introduced principles, the Delaunay-based vector segmentation (shortly *VSeg*) is proposed as follows:

1. **Data preprocessing** – Noise reduction by means of the *3D anisotropic filtering* [15].
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 sampled set of candidate vertices by the common *Incremental method* [14].
4. **Iterative adaptation** – The triangulation is adapted to the underlying image structure by means of isotropic edge splitting and variational meshing.
5. **Mesh segmentation** – Final classification of tetrahedra into image regions according to results of some data clustering method.

Details of all individual phases of the VSeg segmentation are discussed in next sections.

5.1 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 image edge detection algorithms [3, 6, 30, 33] extended to 3D space. Because of the complex nature of medical image data, 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. 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.

Tissue-selective edge detection. The edge detection is divided into separate steps per concrete tissue type. Before the detection starts, the image data are pre-processed using the *power-law contrast enhancement* [16] technique 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

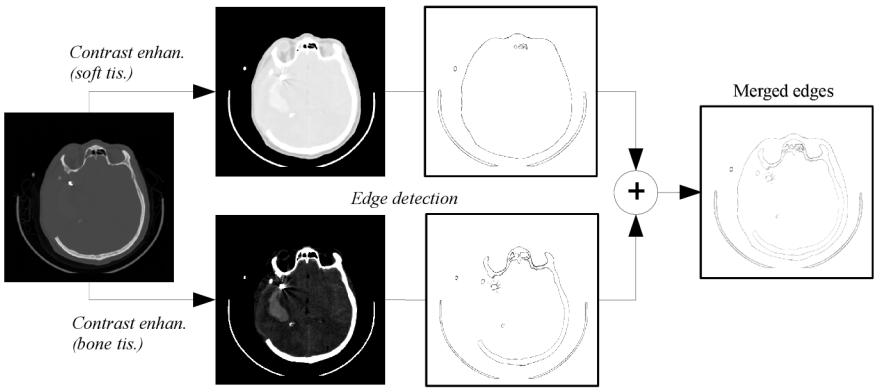


Figure 5.1: Scheme of the tissue-selective edge detection.

merged together into a single image (see Fig. 5.1). In our experiments, the well known *Canny edge detector* [6] extended to 3D space has been used in each step.

3D SUSAN corner detector. In order to respect significant features in the volumetric data during the meshing, we have modified the *Susan corner detector* [33] extending its functionality into 3D space. The Susan (*Smallest Univalue Segment Assimilating Nucleus*) detector was originally developed to locate feature points in 2D images. 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 Final response of the SUSAN detector [33] is proportional to $\frac{1}{N}\sum_{v_i \in R} c_{v_i}$, where 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 (see [33] for more details).

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

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

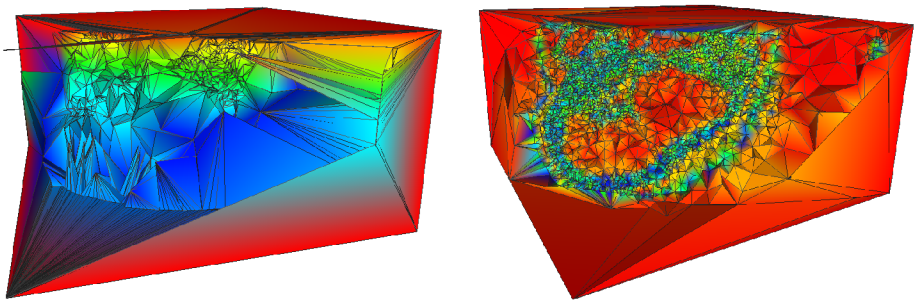


Figure 5.2: The initial Delaunay triangulation (a) and the final mesh after the iterative adaptation (b). Parameters of the sizing field were chosen to be $K = 0.8$ and $T_{avg} = 70mm$.

During the iterative adaptation, 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 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 failures caused by a limited precision of math operations.

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* [14], 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. The whole isotropic edge splitting process can be briefly formulated as follows:

1. 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 $AP_1, P_1P_2, \ldots, P_nB$ in the triangulation.
   - Divide all sub-edges in the sense of isotropic meshing algorithm (Sec. 4).
2. Filter the set of newly created points to discard vertices too close to any other point respecting the control space metric.
3. Insert points to the mesh $T^i \rightarrow T^{i+1}$.

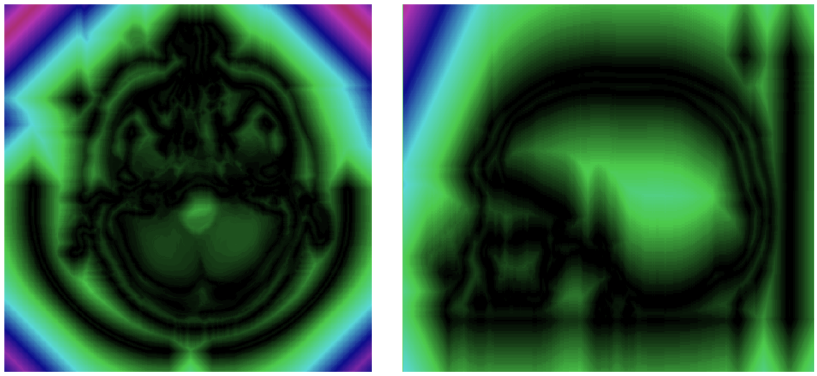


Figure 5.3: Pseudo-colored slices through the three-dimensional control space prescribing size of tetrahedra inside the mesh. Dark values stand for small tetrahedra close to image edges.

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 $\alpha > 10^\circ$. Edges that are almost parallel with an image edge remain unchanged.

Control space. The control space, so called *sizing field*, 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. The control space $H(\Omega)$ can be defined in the same way as the sizing field given by Eq. 4.5. However, 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 (4.5).

This sizing field (Fig. 5.3) 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 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:

$$h'_P = h_P \frac{T_{avg}}{\frac{1}{N}\sum_{v\in\Omega} h_v}, \tag{5.1}$$

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.

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. 5.5). 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. Afterwards, vertices with a non-zero distance sum are moved to the average value they each have accumulated during the pass over all edge voxels.

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. 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.

5.3 Mesh Segmentation

Within the mesh segmentation phase, all tetrahedra are classified into individual image regions which partially correspond to individual tissues. 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. Actually, three different algorithms were used for the unsupervised clustering of feature vectors into image regions:

- Fuzzy C-means (shortly *FCM*) algorithm [32],
- Gaussian Mixture Model optimized by the popular Expectation-Maximization (*EM-GMM*) algorithm [28].
- Min-Cut/Max-Flow graph-based algorithm [4].

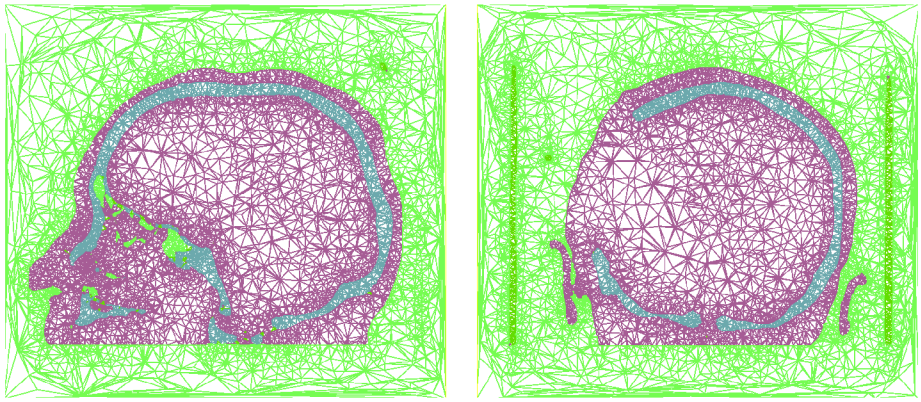


Figure 5.4: Result of the tetrahedral mesh segmentation phase – orthogonal cuts through the classified mesh.

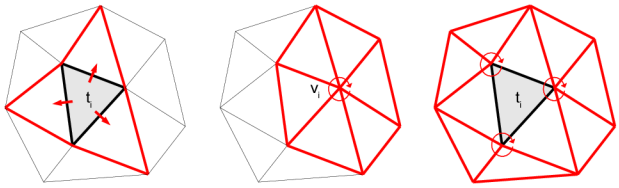
First two techniques do not take into account any spatial/global information about the tetrahedra. Improvements can be made by incorporating global principles. Viewing the mesh as undirected graph, with edges weighted according to the similarity of feature vectors, would allow one to use graph algorithms (graph cuts, path-based clustering, etc.) for the segmentation. In this sense, the *Min-Cut/Max-Flow* [4] algorithm is used to cut a graph whose edges are evaluated according to a similarity of two adjacent tetrahedra. 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* or the *Cosine* distance function. An alternative is the use of simple criterions described in Sec. 5.3.

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 and spatial configuration of adjacent tetrahedra:

- features derived from gray level co-occurrence matrices [36],
- histogram of *Local Binary Patterns (LBP)* [29],
- wavelet features [2, 37], etc.

The feature extraction is problematic if a tetrahedron is relatively small, just a few voxels are available.



(a) (b) (c)

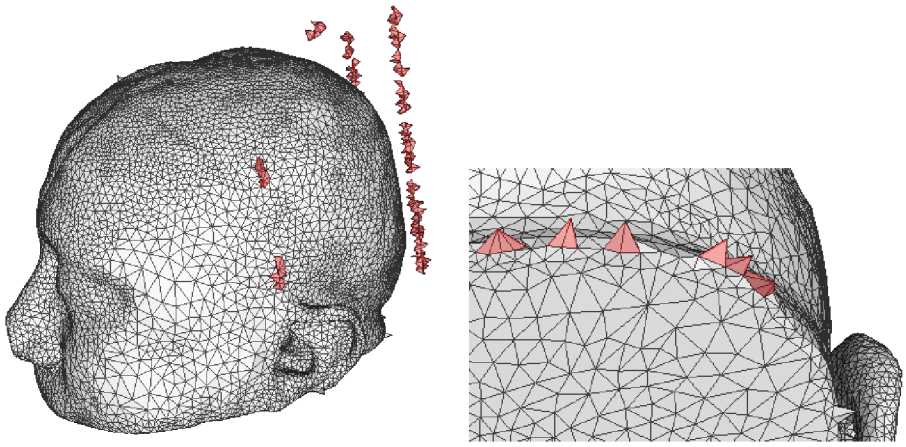
Figure 5.5: Tetrahedra adjacency (a); vertex 1–ring neighborhood (b); and tetrahedron 1–ring neighborhood.

Agglomerative Merging

In the vector segmentation scheme, the agglomerative region merging [22] 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. 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:

$$C(r_i, r_j) = \frac{N_i + N_j}{N_i N_j} S(r_i, r_j). \tag{5.2}$$

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.



(a) (b)

Figure 5.6: Surfaces extracted from the segmented mesh: an original surface without filtering of small isolated parts (a); artifacts that sometimes appear close to sharp edges and must be filtered (b).

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 decreases. Basic similarity measures are the mean intensity value and statistical test of the similarity based on voxel value variance:

$$S_\mu(r_j,r_i) = \exp(-\frac{1}{2\rho^2}|\mu_{r_i} - \mu_{r_j}|^2), \quad S_\sigma(r_j,r_i) = \frac{\sigma(r_i)\sigma(r_j)}{\sigma^2(r_{i,j})}, \tag{5.3}$$

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 $r_i \cup r_j$. These basic similarity measures serve as a merging criterion during the agglomerative merging when small non-classified tetrahedra along boundaries are assigned into neighboring segments.

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. The extracted surface is closed and its mosaic conforms to the chosen parameters of the meshing. 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. 5.6).

Chapter 6

Experimental Results

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.

6.1 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. 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. Since the smoothing is crucial for overall precision of the surface, two standard approaches were tested:

- Taubin's smoothing algorithm [34] that maintains the volume of the mesh (*MC+Taubin*),
- HC algorithm [40] that preserves sharp edges and corners in the mesh (*MC+HC*).

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 [13], 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 *MeshLab* [9] tool which is de facto standard in the area of meshing.

Artificial volumetric data of basic solids such as rectangular solid, cylinder, cone, semi-sphere, pyramid and the *Stanford bunny* 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.

An error between reconstructed surface and the original model is estimated using the *Metro* [8] tool. The approximation error between two meshes is defined as the distance between corresponding sections of the meshes. 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.

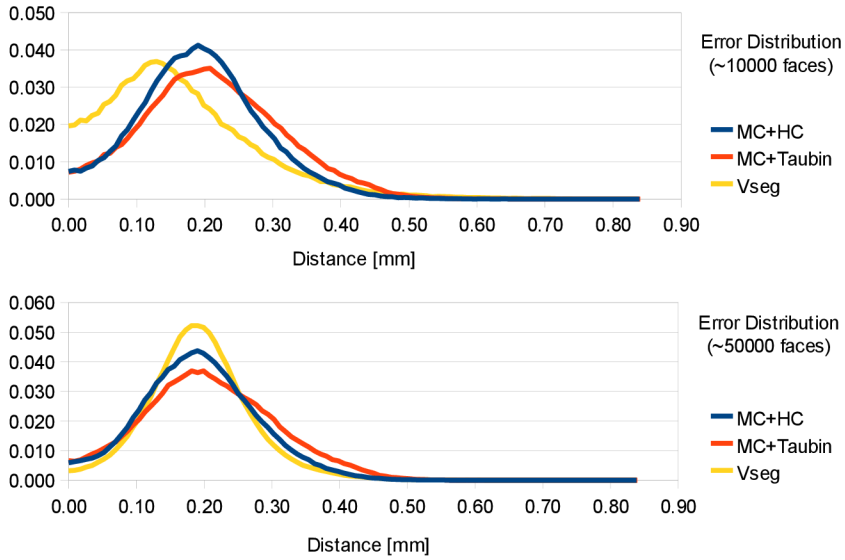


Figure 6.1: Histograms of the surface approximation error for two meshes with a different level of detail (number of faces) – the bunny model.

Fig. 6.1 shows histograms of error distribution for surface models of different level of detail as returned by the Metro tool. Apparently, the VSeg method outperforms both the smoothing-based methods. However, the difference is more evident for smaller meshes. Direct meshing of volumetric image data seems to be more accurate approach then post-processing methods smoothing reconstructed surfaces without any relationship to the original image data.

Fig. 6.2 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.

The question is why the performance decreases with the increasing number of faces? The 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.

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 regular 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 (Fig. 6.3).

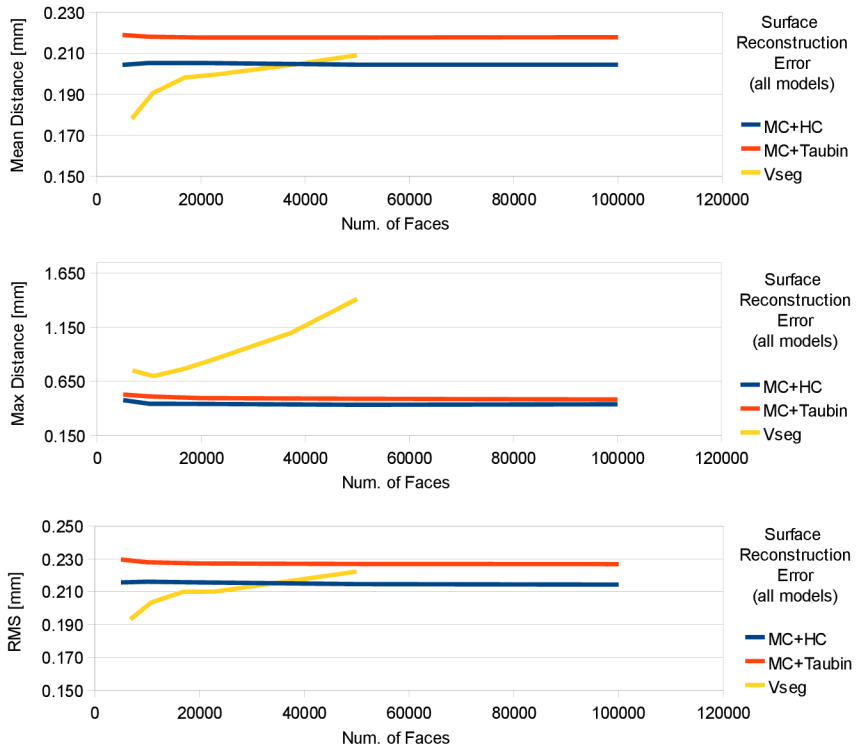
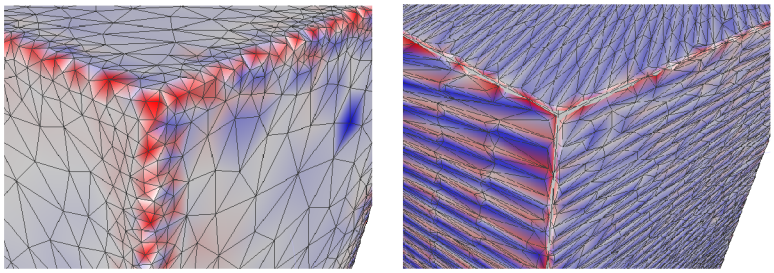


Figure 6.2: Overall surface approximation error – whole set of test models.

6.2 Mesh Quality

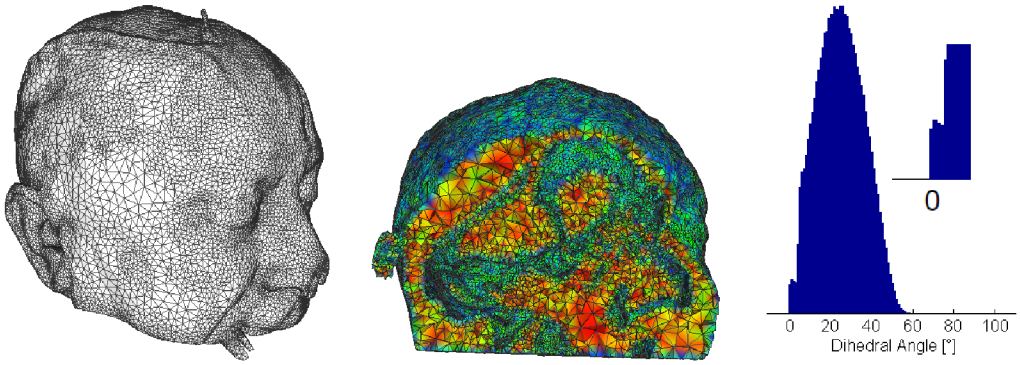
The original VTM approach produces well shaped tetrahedra inside the domain. However, poorly shaped tetrahedra and slivers may appear close to the boundary. Unfortunately, the same problem appears in case of the VSeg meshing method. Meshes in Fig. 6.4 are colored according to the quality of tetrahedra. Clearly, the quality of tetrahedra decreases as getting closer to the boundaries – the red shading moves towards blue. Even thought the embedded sliver elimination algorithm removes a large number of poorly shaped tetrahedra, it does not ensure that all slivers will be successfully eliminated. Hence, this aspect of the presented VSeg technique remains open and will be addressed in the future. A smallest dihedral angle should be guaranteed. Recently, *J. Tournois* [35] has presented a new modification of the original VTM algorithm that particularly solves this problem and produces almost sliver free meshes.

Fig. 6.5 compares surfaces extracted from tetrahedral meshes (the VSeg method) 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.



(a) (b)

Figure 6.3: Error distribution on the reconstructed surface. The mean error of the VSeg method (a) over the entire surface is lower then the error of MC+HC method (b). On the other hand, the maximum error is larger along the sharp edges.



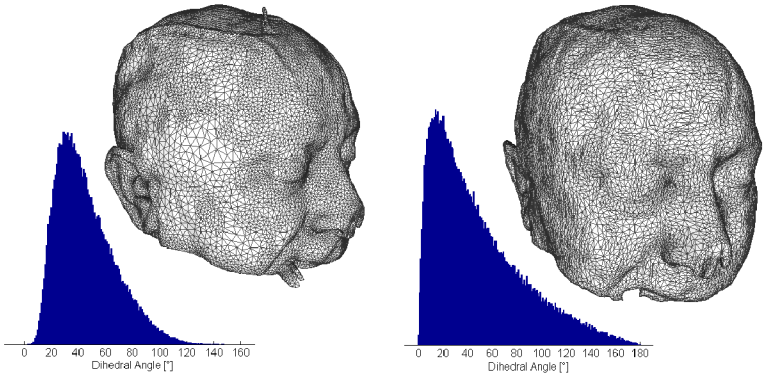
(a) (b) (c)

Figure 6.4: Surface extracted from the tetrahedral mesh (a); cut through the same mesh colored according to the tetrahedra quality; and histogram of minimal dihedral angles (c).

6.3 Mesh Segmentation

The mesh segmentation phase was tested on several CT data sets having resolution mostly 512x512 pixels per slice and its results were compared against manually annotated data – the ground truth. Here, I would like to thank to *3Dim–Laboratory s.r.o.* company for providing part of the test data.

Ground truth. Manual segmentation of medical images is a very complicated task. 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. 6.1 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.



(a) (b)

Figure 6.5: Quality of surfaces reconstructed using the VSeg method (a) and the MC+HC method. Histograms show distribution of dihedral angles.

Soft/Bone tis.	Man1	Man2	Man3	Man4
Man1	-	0.921	0.965	0.896
Man2	0.974	-	0.949	0.971
Man3	0.993	0.979	-	0.922
Man3	0.978	0.979	0.982	-

Table 6.1: 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).

F-measure. Many sophisticated measures of segmentation accuracy can be found in the literature [18]. An often used measure of segmentation *goodness* is the *F-measure* [38]. The *F-measure* combines *recall r* and *precision p* with an equal weight in the equation of the form:

$$F_{measure} = \frac{2rp}{r+p}, \quad p = \frac{T_p}{T_p + F_p}, \quad r = \frac{T_p}{T_p + F_n} \tag{6.1}$$

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. A perfect score of the F-measure is 1, in the worst case the measure is equal to 0.

Meshing Segmented Data

Without too much effort, the Delaunay-based segmentation can be applied to already segmented data. Fig. 6.6 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.

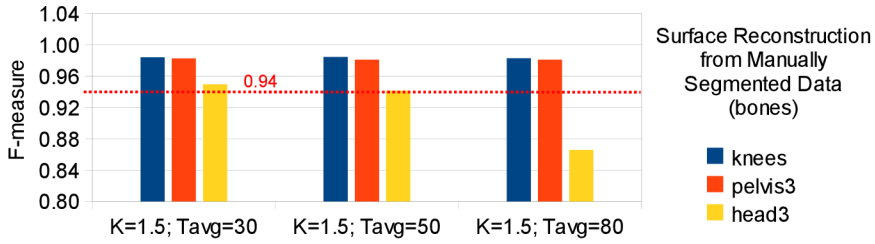


Figure 6.6: Surface reconstruction error when meshing already pre-segmented data. The red line implies the error observed when several people labeled a same CT dataset (see Tab. 6.1).

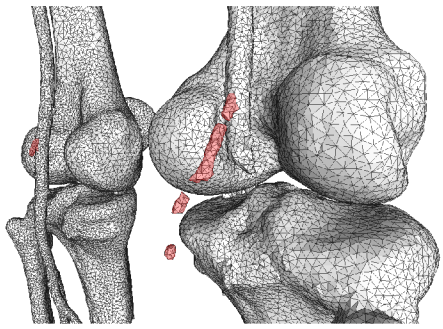


Figure 6.7: Surfaces reconstructed from pre-segmented data. In the red areas of the surface, small anatomical structures are weakly approximated because their size is relatively small compared to a prescribed minimal edge length.

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. 6.1). 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. 6.7).

Segmentation of Medical CT Data

In the last experiment, three different unsupervised clustering techniques (*FCM*, *GMM+EM* and *Min-Cut/Max-Flow*) were applied to meshes in order to classify tetrahedra into individual regions/segments. Fig. 6.8 recapitulates results of the mesh segmentation. All clustering techniques are able to distinguish soft tissue. When compared to the manual segmentation, the VSeg method provides 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.

Not the worse results occur in case of bone segmentation from the *head3* dataset. The VSeg

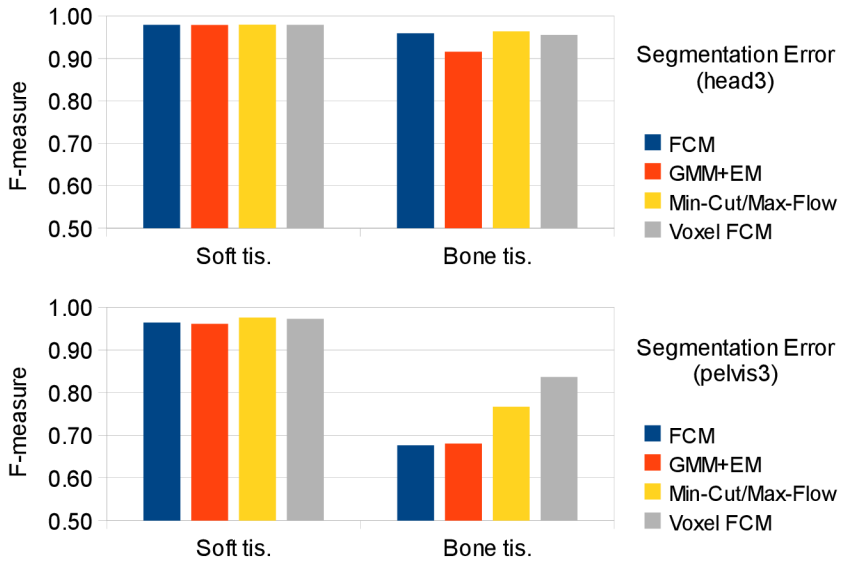


Figure 6.8: Overall segmentation error of the VSeg method. Three alternative clustering methods (*FCM*, *GMM+EM* and *Min-Cut/Max-Flow*) are compared to the straight *FCM* clustering of volumetric data (*voxel FCM*).

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 second dataset. Only the graph-based *Min-Cut/Max-Flow* algorithm provides reasonable results. Because the method takes spatial image structure more into account, results of this graph method overcomes other techniques.

Due to the thickness of the cortical bone and regarding resolution of CT data, very thin edges are present in the image data which are practically undetectable by conventional edge detection techniques without more knowledge of the data. Therefore, such kind of (non)edges is not well approximated during the meshing process which causes more errors in the final mesh segmentation. This nature of some medical CT data is also one of the reasons to allow manual corrections of the mesh segmentation.

6.4 Runtime Statistics

Basic runtime statistics can be found in Fig. 6.9. 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.

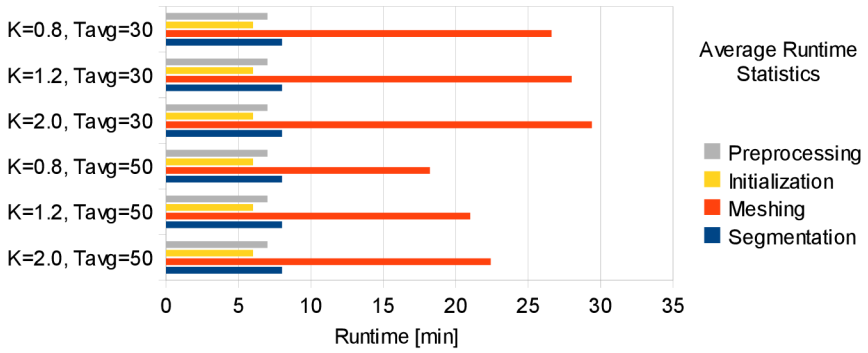


Figure 6.9: Runtime statistics of the VSeg method for different meshing setup.

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 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. 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.

Chapter 7
Conclusions

This paper presents a 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 partition the volumetric data. Such direct meshing of volumetric image data appears to be more accurate approach then traditional techniques which start with a surface extraction followed by the decimation and smoothing without any relationship to the original image data. Nevertheless, the idea of the vector 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 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). 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.

However, several inconveniences can be still found in the method that are not very favourable from the practical point of view. Even thought 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.

Another disadvantage is 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. However, 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. Theses aspects of the proposed vector segmentation technique should be addressed in the future work.

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., GANESAN, L., AND PRIYAL, S. P. Texture classification using gabor wavelets based rotation invariant features. *Pattern Recognition Letters 27*, 16 (2006), 1976 – 1982.

[3] BASU, M. Gaussian-based edge-detection methods: A survey. *SMC-C 32*, 3 (August 2002), 252–260.

[4] 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.

[5] 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.

[6] CANNY, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. 8*, 6 (1986), 679–698.

[7] 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.

[8] 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/.

[9] CNR, V. C. L. I. Meshlab. http://meshlab.sourceforge.net/.

[10] DARDENNE, J., VALETTE, S., SIAUVE, N., BURAIS, N., AND PROST, R. Variational tetrahedral mesh generation from discrete volume data. *Vis. Comput. 25*, 5-7 (2009), 401–410.

[11] 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.

[12] DU, Q., FABER, V., AND GUNZBURGER, M. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review 41*, 4 (1999), 637–676.

[13] 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.

[14] GEORGE, P.-L., AND BOROUCHAKI, H., Eds. *Delaunay Triangulation and Meshing: Application to Finite Elements*. Hermes Science Publications, Orlando, FL, USA, November 1998.

[15] 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.

[16] GONZALEZ, R. C., AND WOODS, R. E. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] KRYSL, P., AND ORTIZ, M. Variational delaunay approach to the generation of tetrahedral finite element meshes. *International Journal 50* (1999), 1681–1700.

[22] 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.

[23] LABELLE, F., AND SHEWCHUK, J. R. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Trans. Graph. 26*, 3 (2007), 57.

[24] 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.

[25] LORENSEN, W. E., AND CLINE, H. E. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph. 21*, 4 (1987), 163–169.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] PHAM, D. L., AND PRINCE, J. L. Adaptive fuzzy segmentation of magnetic resonance images. *IEEE Transactions on Medical Imaging 18* (September 1999).

[33] SMITH, S. M., AND BRADY, J. M. Susan - a new approach to low level image processing. *Inter. Journal of Computer Vision* (1996).

[34] TAUBIN, G., AND TAUBIN, G. Geometric signal processing on polygonal meshes, Sep 2000.

[35] 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.

[36] 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.

[37] UNSER, M., ALDROUBI, A., AND LAINE, A. Guest Editorial: Wavelets in Medical Imaging. *IEEE Transactions on Medical Imaging 22*, 3 (2003), 285–288.

[38] VAN RIJSBERGEN, C. J. *Information Retrieval 2nd Ed.* Butterworths, London, 1979.

[39] 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.

[40] 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.

[41] 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).

[42] ZHANG, J. Reconstruction of geometry from cardiac mr images. *CVGIP: Image Understanding 55* (January 1992), 14–26.

[43] 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.

[44] 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.

Selected Papers by the Author

M. Španěl, P. Kršek, M. Švub, V. Štancl. Vector Segmentation of Volumetric Image Data: Tetrahedral Meshing Constrained by Image Edges. In *Proceedings of the 3rd International Joint Converence on Computer Vision, Imaging and Computer Graphics Theaory and Applications – GRAPP 2010*, pp. 134-138, Portugal, 2010, INSTICC.

M. Švub, P. Kršek, M. Španěl, V. Štancl, R. Bartoň, J. Vaďura. Feature preserving mesh smoothing algorithm based on local normal covariance. In *Proceedings of WSCG'10*, pages 6, Plzeň, CZ, ZČU v Plzni, 2010.

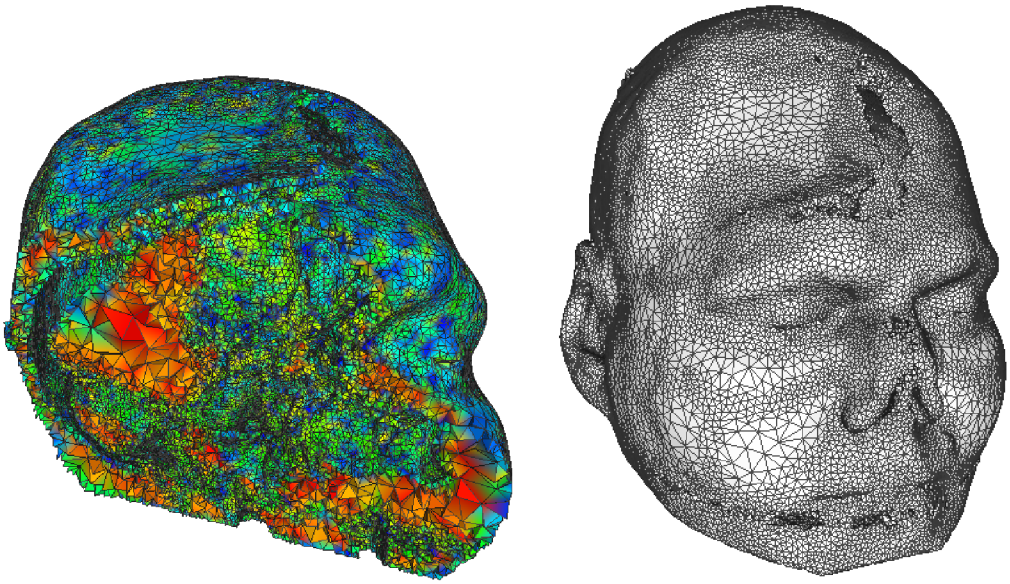
P. Kršek, M. Španěl, M. Švub, V. Štancl, R. Bartoň, O. Šiler. Network Collaborative Environment Supporting 3D Medicine. In *Proceedings of the 31st Annual International conference of the IEEE Engineering in Medicine and Biology*, Minneapolis, US, 2009, IEEE.

M. Španěl, P. Kršek, M. Švub, V. Štancl, O. Šiler. Delaunay-Based Vector Segmentation of Volumetric Medical Images. In *Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns – CAIP 2007*, LNCS 4673, pp. 261-269, Berlin Heidelberg, DE, 2007, Springer.

M. Španěl, P. Kršek. Vector-based Medical Image Segmentation using Adaptive Delaunay Triangulation. In *Proceedings of the Sixth IASTED International Conference on Visualization, Imaging, and Image Procesing*, Palma de Mallorca, ES, 2006, ACTA Press.

Sample Results

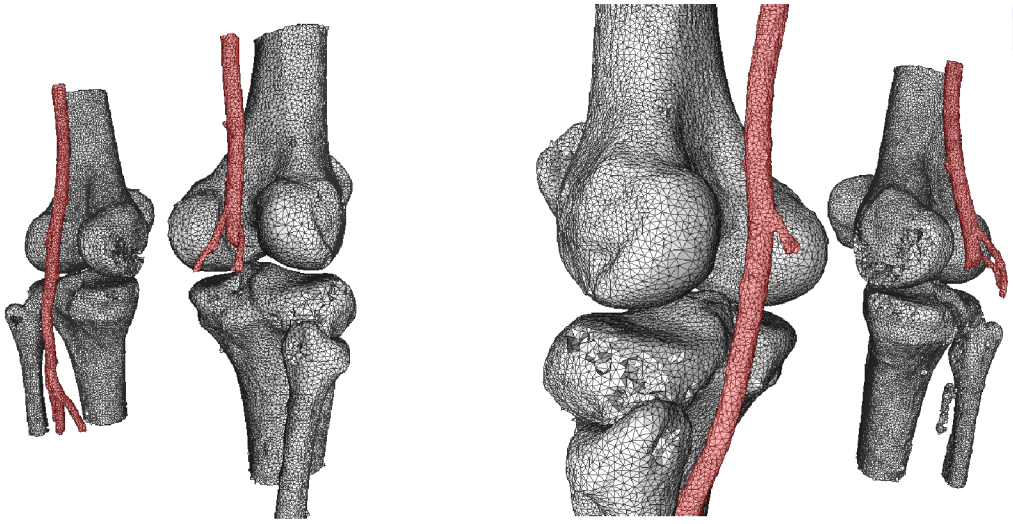
Input dataset: *CT–head2*; meshing parameters: $K = 1.5$, $T_{avg} = 50mm$, and $L_{min} = 1.5mm$.



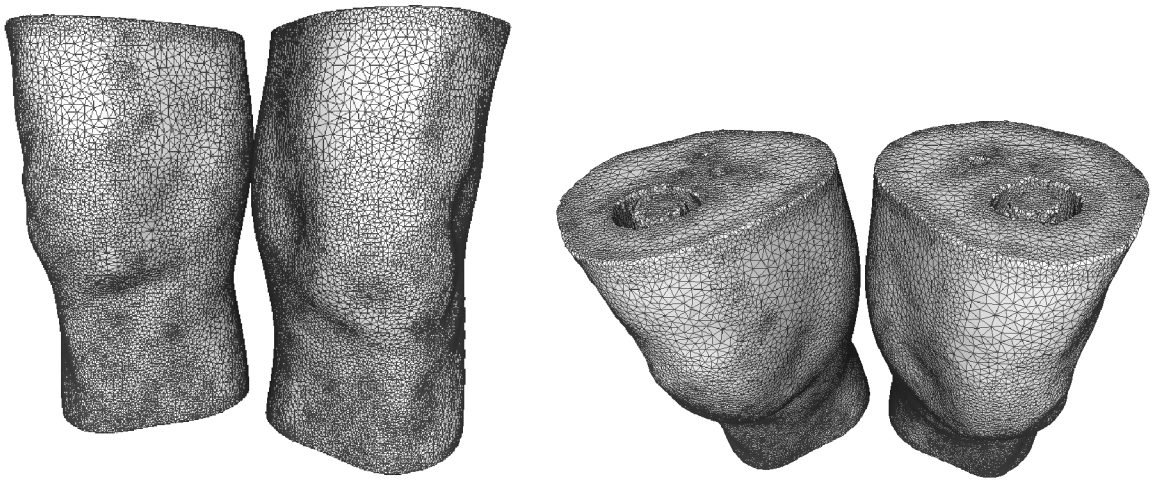
82885 vertices, 165348 faces

127528 vertices, 256060 faces

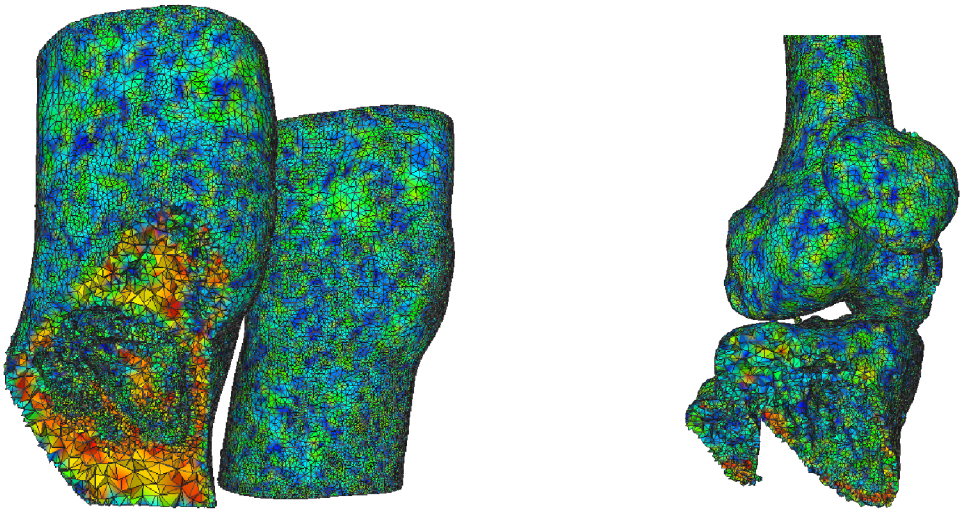
Input dataset: *CT–knees*; meshing parameters: $K = 1.5$, $T_{avg} = 30mm$, and $L_{min} = 1.0mm$.



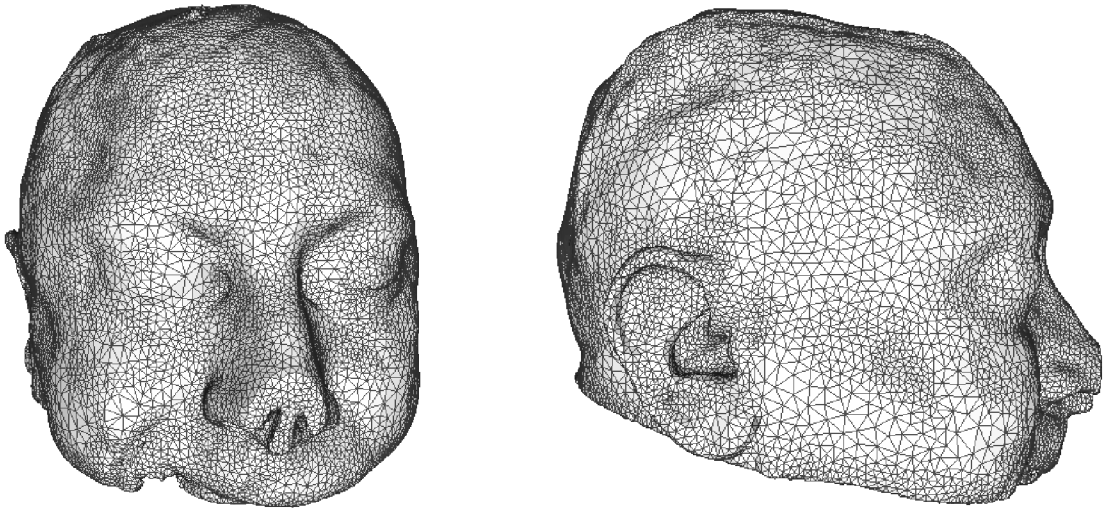
125444 vertices, 250518 faces



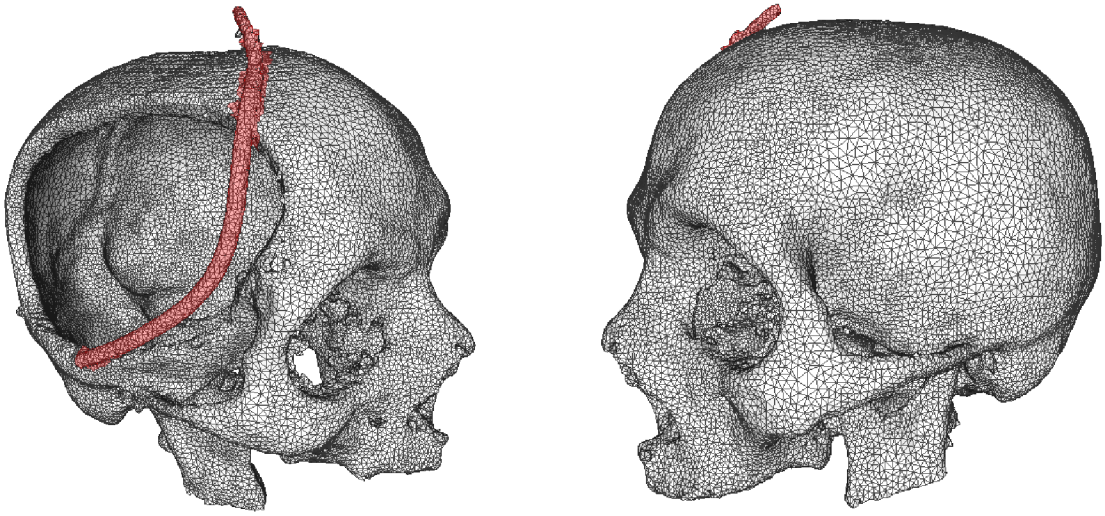
138883 vertices, 136266 faces



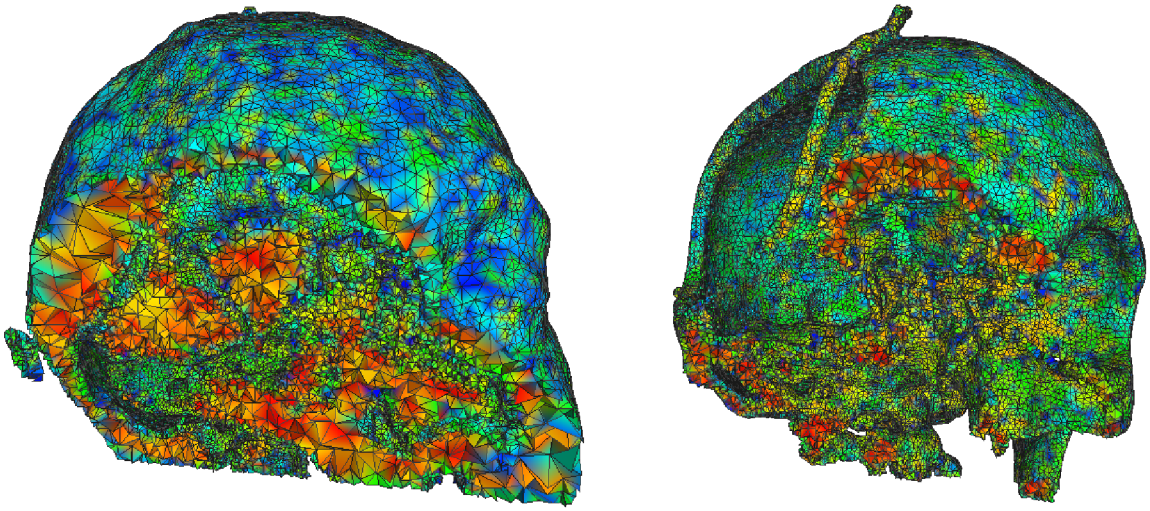
Input dataset: *CT–head3*; meshing parameters: $K = 1.5$, $T_{avg} = 50mm$, and $L_{min} = 1.5mm$.



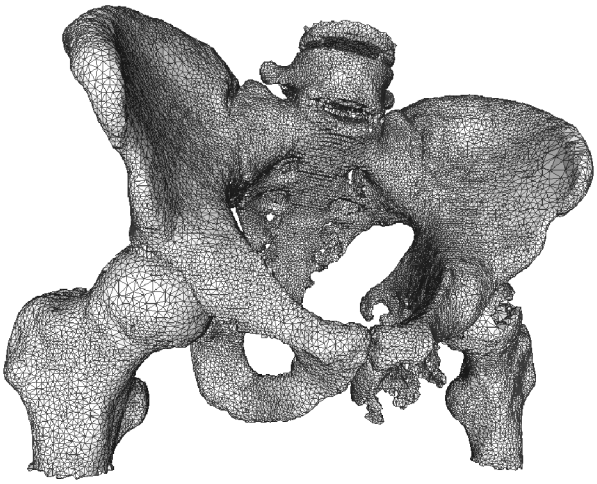
89131 vertices, 178124 faces



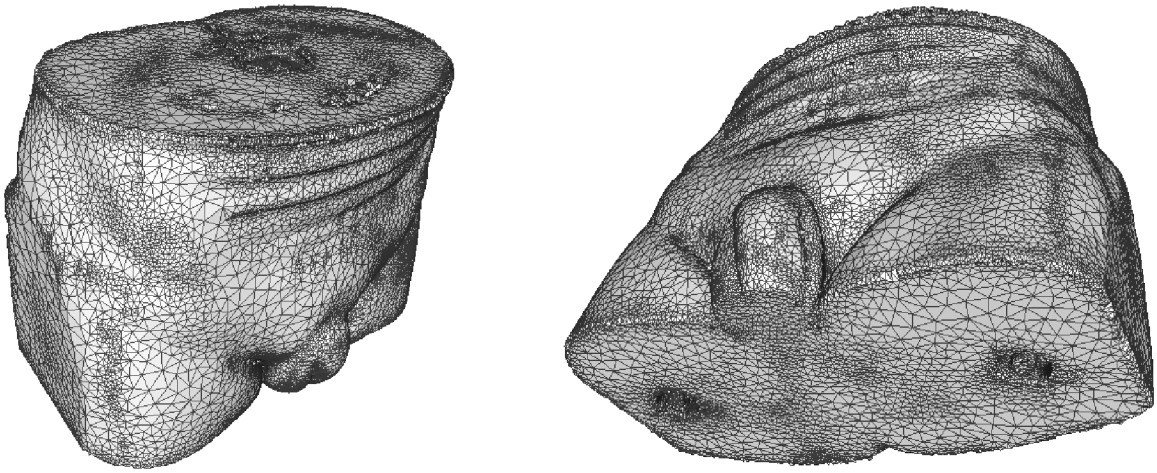
87833 vertices, 176318 faces



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