UPPSALA
UNIVERSITET

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 1411

Interactive 3D Image Analysis for Cranio-Maxillofacial Surgery Planning and Orthopedic Applications

JOHAN NYSJÖ

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2016

ISSN 1651-6214
ISBN 978-91-554-9668-5
urn:nbn:se:uu:diva-301180

Dissertation presented at Uppsala University to be publicly examined in ITC 2446, Lägerhyddsvägen 2, Uppsala, Friday, 30 September 2016 at 10:15 for the degree of Doctor of Philosophy. The examination will be conducted in English. Faculty examiner: Jayaram K. Udupa (Medical Imaging Section, Department of Radiology, University of Pennsylvania).

Abstract

Nysjö, J. 2016. Interactive 3D Image Analysis for Cranio-Maxillofacial Surgery Planning and Orthopedic Applications. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 1411. 58 pp. Uppsala: Acta Universitatis Upsaliensis. ISBN 978-91-554-9668-5.

Modern medical imaging devices are able to generate highly detailed three-dimensional (3D) images of the skeleton. Computerized image processing and analysis methods, combined with real-time volume visualization techniques, can greatly facilitate the interpretation of such images and are increasingly used in surgical planning to aid reconstruction of the skeleton after trauma or disease. Two key challenges are to accurately separate (segment) bone structures or cavities of interest from the rest of the image and to interact with the 3D data in an efficient way. This thesis presents efficient and precise interactive methods for segmenting, visualizing, and analysing 3D computed tomography (CT) images of the skeleton. The methods are validated on real CT datasets and are primarily intended to support planning and evaluation of cranio-maxillofacial (CMF) and orthopedic surgery.

Two interactive methods for segmenting the orbit (eye-socket) are introduced. The first method implements a deformable model that is guided and fitted to the orbit via haptic 3D interaction, whereas the second method implements a user-steered volumetric brush that uses distance and gradient information to find exact object boundaries.

The thesis also presents a semi-automatic method for measuring 3D angulation changes in wrist fractures. The fractured bone is extracted with interactive mesh segmentation, and the angulation is determined with a technique based on surface registration and RANSAC.

Lastly, the thesis presents an interactive and intuitive tool for segmenting individual bones and bone fragments. This type of segmentation is essential for virtual surgery planning, but takes several hours to perform with conventional manual methods. The presented tool combines GPU-accelerated random walks segmentation with direct volume rendering and interactive 3D texture painting to enable quick marking and separation of bone structures. It enables the user to produce an accurate segmentation within a few minutes, thereby removing a major bottleneck in the planning procedure.

Keywords: medical image analysis, interactive segmentation, volume rendering, computed tomography

Johan Nysjö, Department of Information Technology, Division of Visual Information and Interaction, Box 337, Uppsala University, SE-751 05 Uppsala, Sweden. Department of Information Technology, Computerized Image Analysis and Human-Computer Interaction, Box 337, Uppsala University, SE-75105 Uppsala, Sweden.

ISSN 1651-6214
ISBN 978-91-554-9668-5
urn:nbn:se:uu:diva-301180 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-301180)

List of papers

This thesis is based on the following eight papers, which are referred to in the text by their Roman numerals.

I I. Nyström, **J. Nysjö**, F. Malmberg. *Visualization and Haptics for Interactive Medical Image Analysis: Image Segmentation in Cranio-Maxillofacial Surgery Planning.* In Proceedings of the 2nd International Visual Informatics Conference (IVIC), LNCS 7066, pp. 1–12, 2011.

II **J. Nysjö**, A. Christersson, F. Malmberg, I.-M. Sintorn, I. Nyström. *Towards User-Guided Quantitative Evaluation of Wrist Fractures in CT Images.* In Proceedings of International Conference on Computer Vision and Graphics (ICCVG), LNCS 7594, pp. 204–211, 2012.

III R.H. Khonsari, M. Friess, **J. Nysjö**, G. Odri, F. Malmberg, I. Nyström, E. Messo, J.M. Hirsch, E.A.M. Cabanis, K.H. Kunzelmann, J.M. Salagnac, P. Corre, A. Ohazama, P.T. Sharpe, P. Charlier and R. Olszewski. *Shape and Volume of Craniofacial Cavities in Intentional Skull Deformations.* American Journal of Physical Anthropology, Vol.151, No.1, pp. 110–119, 2013.

IV **J. Nysjö**, A. Christersson, I.-M. Sintorn, I. Nyström, S. Larsson, F. Malmberg. *Precise 3D Angle Measurements in CT Wrist Images.* In Proceedings of International Conference on Image Analysis and Processing (ICIAP), LNCS 8157, pp. 479–488, 2013.

V A. Hast, **J. Nysjö**, A. Marchetti. *Optimal RANSAC - Towards a Repeatable Algorithm for Finding the Optimal Set.* Journal of WSCG, Vol.21, No.1, pp. 21–30, 2013.

VI **J. Nysjö**, F. Malmberg, I.-M. Sintorn, I. Nyström. *BoneSplit - A 3D Texture Painting Tool for Interactive Bone Separation in CT Images*. Journal of WSCG, Vol.23, No.2, pp. 157–166, 2015.

VII A. Christersson, **J. Nysjö**, L. Berglund, F. Malmberg, I.-M. Sintorn, I. Nyström, S. Larsson. *Comparison of 2D radiography and a semi-automatic CT-based 3D method for measuring change in dorsal angulation over time in distal radius fractures.* Skeletal Radiology, Vol.45, No.6, pp. 763–769, 2016.

VIII **J. Nysjö**, J. Nilsson, F. Malmberg, A. Thor, I. Nyström. *Rapid and Precise Orbit Segmentation through Interactive 3D Painting.* Manuscript.

Reprints were made with permission from the publishers.

The author has substantially contributed to the work in all papers. In Papers I, II, IV, VI, and VIII, the author was the main contributor and had the main responsibility for method development, implementation, writing, experiments, and analysis of the results.

In Paper III, the author developed and implemented the segmentation method, prepared and performed the segmentation experiments together with the main author RK, contributed to the analysis, and wrote the segmentation-specific parts of the material and methods section.

In Paper V, the author prepared and performed the experiments on the medical datasets, contributed to the method development and implementation together with the main author AH, and wrote parts of results and discussion sections.

In Paper VII, the author developed and implemented the 3D measurement method, prepared and performed the experiments together with the main author AC, contributed to the analysis, and wrote parts of the paper.

Related work

In addition to the papers included in this thesis, the author has also written or contributed to the following publications.

Reviewed

i L. Svensson, **J. Nysjö**, A. Brun, I. Nyström, I.-M. Sintorn. *Rigid Template Registration in MET Images Using CUDA.* In Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP), pp. 418–422, 2012.

ii R. Khonsari, **J. Nysjö**, B. Way, T. Karunakaran, I. Nyström, D. Dunaway, R. Evans, R. Olszewski, J. Britto, G. Odri. *Orbital morphology in Crouzon-Pfeiffer and Apert syndromes before and after surgical correction: study by 3D cephalometry, semi-automatic segmentation and 3D shape comparison.* Proc. Computer Assisted Radiology and Surgery (CARS), Fukuoka, Japan, 2014. Abstract.

iii T. Smektala, **J. Nysjö**, A. Thor, A. Homik, K. Sporniak-Tutak, K. Safranow, K. Dowgierd, R. Olszewski. *Three-Dimensional Eyeball and Orbit Volume Modification After LeFort III Midface Distraction.* Journal of Craniofacial Surgery, Vol.26, No.5, pp. 1652–1655, 2015.

iv A. M. Clement, **J. Nysjö**, R. Strand, P. E. Ahlberg. *Brain - Endocast Relationship in the Australian Lungfish, Neoceratodus forsteri, Elucidated from Tomographic Data (Sarcopterygii: Dipnoi).* PLOS ONE, Vol.10, No.10, 2015.

v I. Nyström, P. Olsson, **J. Nysjö**, F. Nysjö, F. Malmberg, S. Seipel, J. Hirsch, I. B. Carlbom. *Virtual Cranio-Maxillofacial Surgery Planning with Stereo Graphics and Haptics.* Computer-Assisted Musculoskeletal Surgery - Thinking and Executing in 3D, 2016. Book chapter.

vi E. S. Linnér, M. Morén, K.-O. Smed, **J. Nysjö**, R. Strand. *LatticeLibrary and BccFccRaycaster: Software for processing and viewing 3D data on optimal sampling lattices.* SoftwareX, doi:10.1016/j.softx.2016.01.002, 2016.

vii L. Svensson, S. Svensson, I. Nyström, F. Nysjö, **J. Nysjö**, A. Laloeuf, L. Hollander, A. Brun, S. Masich, L. Sandblad, M. Sani, I.-M. Sintorn. *ProViz: a tool for explorative 3-D visualization and template matching in electron tomograms.* Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, doi:10.1080/21681163.2016.1154483, 2016.

viii A. M. Clement, R. Strand, **J. Nysjö**, J. A. Long, P. E. Ahlberg. *A new method for reconstructing brain morphology: applying the brain-neurocranial spatial relationship in an extant lungfish to a fossil endocast.* Royal Society Open Science, Vol.3, No.7, 2016.

ix R. Khonsari, B. Way, **J. Nysjö**, G. Odri, R. Olszewski, R. Evans, D. Dunaway, I. Nyström, J. Britto. *Frontofacial advancement and bipartition in Crouzon-Pfeiffer and Apert syndromes: impact of fronto-facial surgery upon orbital and airway parameters in FGFR2 syndromes.* Journal of Cranio-Maxillofacial Surgery, doi:10.1016/j.jcms.2016.08.015, 2016.

Non-reviewed

x **J. Nysjö**, A. Christersson, I.-M. Sintorn, I. Nyström, F. Malmberg. *Haptic-Enabled 3D Angle Measurements in CT Wrist Images.* In Proceedings of Swedish Symposium on Image Analysis (SSBA), 2012.

xi **J. Nysjö**, R. Khonsari, F. Malmberg, I.-M. Sintorn, I. Nyström. *Analyzing Orbital Size and Shape Before and After Surgical Correction: a Study by Semi-Automatic Segmentation and Statistical Shape Models.* In Proceedings of Swedish Symposium on Image Analysis (SSBA), 2014.

xii **J. Nysjö**, F. Malmberg, I.-M. Sintorn, I. Nyström. *BoneSplit - A 3D Painting Tool for Interactive Bone Segmentation in CT Images.* In Proceedings of Swedish Symposium on Image Analysis (SSBA), 2016.

Contents

Abbreviations

CBCT	Cone Beam Computed Tomography
CMF	Cranio-Maxillofacial
CPU	Central Processing Unit
CT	Computed Tomography
DVR	Direct Volume Rendering
FPS	Frames Per Second
GPU	Graphics Processing Unit
HU	Hounsfield Unit
ICP	Iterative Closest Point
MIP	Maximum Intensity Projection
MPR	Multi-Planar Reformatting
PVE	Partial Volume Effect
RANSAC	Random Sampling Consensus
ROI	Region Of Interest
TPS	Thin-Plate Spline

1. Introduction

Modern medical imaging devices are able to generate three-dimensional (3D) volumetric images of the skeleton that display fractures or other injuries in high level of detail. Computerized image processing and analysis methods, combined with real-time volume visualization techniques, can greatly facilitate the interpretation of such images and are increasingly used to virtually plan complex reconstructions of the skeleton after trauma or disease. Two difficulties, however, are to accurately separate (segment) bone structures or cavities of interest from the rest of the image and to interact with the 3D data in an efficient way.

Segmentation is required for creating 3D models of injured as well as intact bone structures that can be used as input for further analysis or virtual surgery planning. However, since bones can have high shape variability, and the contrast between bone and soft tissue sometimes is low, it is difficult for a computer to perform the segmentation automatically. Currently, the segmentation is thus often performed by human experts who manually trace the contours of the bone structures in individual slices of the 3D image. This approach tends to produce accurate results, but is very tedious and time-consuming due to the large number of slices to process. A compromise, which has been found to work well in many applications, is to instead use semi-automatic or interactive segmentation methods, where the user guides and corrects precise computer algorithms that perform most of the tedious and repetitive work.

1.1 Objectives

The main objective of this thesis has been to develop efficient and precise interactive methods and tools for segmenting, visualizing, and analysing 3D computed tomography (CT) images of the skeleton, to facilitate planning and evaluation of cranio-maxillofacial (CMF) and orthopedic surgery. The work can be divided into three main applications:

- Size and shape analysis of the orbit (eye-socket).
- Quantitative analysis of wrist fractures.
- Segmentation of bones and bone fragments for virtual surgery planning.

We emphasize that the focus of the thesis work is on the theory and implementation of the image analysis and visualization methods and their applications to medicine; the author has not been directly involved in the CT image

acquisition process and is not a medical expert. The work has, however, been carried out in close collaboration with researchers from the Department of Surgical Sciences, Orthopaedics and Oral & Maxillofacial Surgery, Uppsala University Hospital, Sweden, and from the Service de chirurgie maxillofaciale et plastique, Hôpital Necker Enfants-Malades, Paris.

The CT images shown in the thesis have been obtained from Uppsala University Hospital, the OsiriX DICOM repository, and the Musée de l'Homme in Paris.

1.2 Thesis outline

The thesis is structured as follows. Chapter 2 describes how digital volume images of the skeleton are represented and acquired. Chapters 3–6 present the underlying methods and pipeline for processing, visualizing, segmenting, and analysing bone structures in CT images. The main contributions of the thesis are presented and summarized in Chapter 7. Chapter 8 concludes the thesis and suggests directions for future work. Lastly, a summary in Swedish is provided in Chapter 9.

2. Digital volume images

This chapter provides a brief overview of how digital volume images of the skeleton are represented and acquired.

2.1 Image representation

A *grayscale image* captured with some form of imaging device can be expressed as a continuous function $f(\mathbf{x})$, where $\mathbf{x}$ is a spatial coordinate vector and $f(\mathbf{x})$ is the intensity value at $\mathbf{x}$. In order to be processed with a computer, the continuous image must first be *digitized* [19], that is, converted into digital form. Digitization involves two processes, *sampling* and *quantification*, where sampling is the process of taking samples of the continuous image at predefined grid points, and quantification is the process of converting the sampled intensity values into discrete quantities. In the resulting digital image, each image element is a grid point represented by a certain number of bits. An image with b bits per element has $L = 2^b$ possible intensity levels or *gray-levels*. Typical b-values for the images used in this thesis are 1, 8, or 16. Images with only two intensity levels ($b = 1$) are referred to as *binary images*.

Figure 2.1 illustrates a two-dimensional (2D) digital image $f(x,y)$ and a three-dimensional (3D) digital image $f(x,y,z)$. Elements of 2D images are called *pixels* (picture elements), whereas elements of 3D images are called *voxels* (volume elements). The size of a voxel can be *isotropic* (uniform) or *anisotropic* (non-uniform). Medical volume images typically have anisotropic voxels, but can be resampled to isotropic voxel size if required.

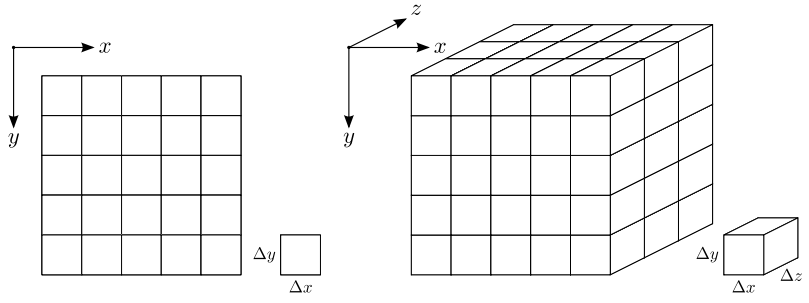


Figure 2.1. Left: a 2D digital image $f(x,y)$ and a pixel p of size $\Delta x \times \Delta y$. Right: a 3D digital volume image $f(x,y,z)$ and a voxel v of size $\Delta x \times \Delta y \times \Delta z$. Medical volume images tend to have anisotropic voxels, i.e., voxels where $\Delta z > \Delta x \wedge \Delta z > \Delta y$.

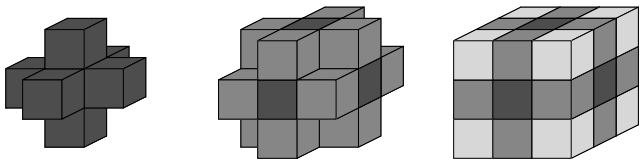


Figure 2.2. Different $3 \times 3 \times 3$ neighborhoods of a voxel $\mathbf{x}$ in a cubic grid. Left: the 6-neighborhood $N_6(\mathbf{x})$. Middle: the 18-neighborhood $N_{18}(\mathbf{x})$. Right: the 26-neighborhood $N_{26}(\mathbf{x})$.

In a 3D digital volume image sampled into a cubic grid, each voxel has six face neighbors, 12 edge neighbors, and eight vertex neighbors. Figure 2.2 shows the $3 \times 3 \times 3$ *neighborhoods* $N_6(\mathbf{x})$, $N_{18}(\mathbf{x})$, and $N_{26}(\mathbf{x})$ of a voxel $\mathbf{x}$. Two voxels $\mathbf{x}$ and $\mathbf{y}$ are said to be *m-connected* if $\mathbf{y} \in N_m(\mathbf{x})$ or vice versa.

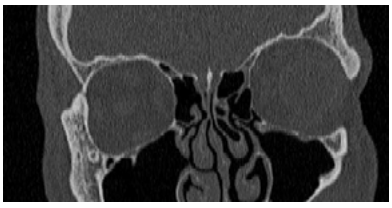
2.2 Computed tomography

Computed tomography (CT) is a medical imaging technique that uses X-rays to generate 2D cross-sectional images (*slices*) of the body [39]. CT scanners have two key components: an X-ray source, and a ring of sensors located on a gantry around the patient. To generate an image, the X-ray source is rotated around the patient while the sensors opposite the source measure the transmitted radiation that passes through the body. The output of the sensors is processed by mathematical reconstruction algorithms to transform the sensed data into a cross-sectional image. The scanning is repeated at even intervals while the patient is moved through the sensor ring by a motorized table, resulting in a stack of 2D images that together form a 3D volume image of the body. Different tissue types will absorb radiation to different degrees based on their *radio-density*, which makes it possible to distinguish between, for example, bone and soft tissue in the reconstructed image (Figure 2.3).

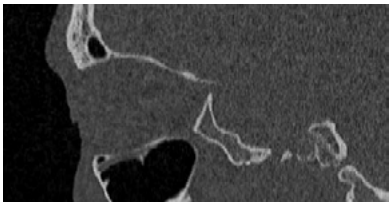
Most of the CT images encountered in this thesis are head or wrist images. These images typically have the dimensions $512 \times 512 \times N_z$ voxels, where N_z is the number of slices in the image and ranges from 100–500. The pixel spacing is usually between 0.3–0.5 mm and the slice thickness 0.5–2.0 mm.

The intensity of a voxel represents the radio-density of the contained tissue type and is given in *Hounsfield units* (HU) [39] ranging from -1024 to 3095. Table 2.1 shows approximate HU values for bone and other tissue types. Due to limited spatial resolution, a voxel in a CT image might cover more than one tissue type. Such voxels are assigned the average radio-density value of the contained tissue types, resulting in blurred boundaries between tissue regions. This artifact is referred to as the *partial volume effect* (PVE) [39]. Other artifacts commonly seen in CT images are intensity noise (visible in Figure 2.3), streak artifacts from metallic implants, and motion blur caused by patient movements.

(a) Axial (top) plane



(b) Coronal (front) plane



(c) Sagittal (side) plane

Figure 2.3. Cross-sectional views of a 3D CT image of the facial skeleton. The intensity of each voxel reflects the radio-density of the contained tissue type, making it possible to distinguish bone (bright regions) from soft tissues (gray) and air (black).

Table 2.1. *Approximate Hounsfield unit (HU) values for different tissue types.*

Tissue	HU
Air	-1000
Lung	-900 to -170
Water	0
Soft tissue	10 to 60
Soft tissue (contrast)	100 to 300
Bone	400 to 3000

Contrast agents can be injected before scanning to improve the visibility and contrast between soft-tissues. There are also cone-beam CT (CBCT) scanners, which use a cone-shaped X-ray beam to produce a 3D image in a single rotation around the scanned subject. CBCT images offer higher resolution at lower radiation dose than conventional CT images, but are more challenging to visualize and analyse since the grayscale values do not represent actual HU values and the image quality often is degraded by scatter artifacts.

3. Image processing tools

This chapter covers various pre- and post-processing tools that can be applied on CT images to remove, enhance, extract, or modify features of interest.

3.1 Region of interest selection

CT images are often large and might include structures that occlude the *region of interest* (ROI) during 3D visualization or analysis. To remove such structures and reduce the processing time, the image should be cropped around the ROI before further processing is performed. This can be accomplished in a simple but efficient manner by generating maximum intensity projections (Chapter 4.3) of the image in axial, coronal, and sagittal views and asking the user to select the ROI in each view with a simple rectangle selection tool, as shown in Figure 3.1. Another technique is to use a 3D box widget that can be resized and positioned around the ROI via mouse-controlled handles [39].

3.2 Spatial filtering

Various types of *spatial filters* [19] can be applied on CT images to enhance structures of interest or remove artifacts. The filtering is performed with a

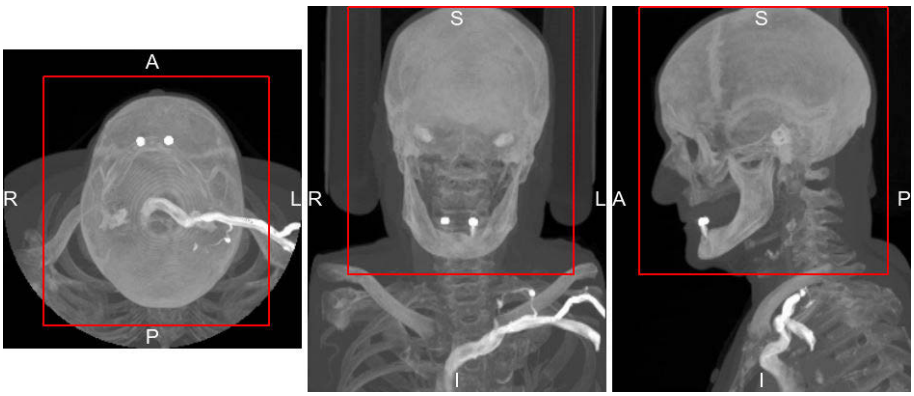
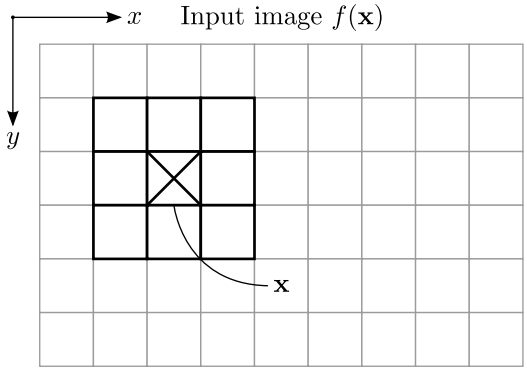


Figure 3.1. Interactive rectangle selection tool that allows the user to crop a CT image around the region of interest (ROI) in maximum intensity projection views.



Filter kernel h

$h(-1,-1)$	$h(0,-1)$	$h(1,-1)$
$h(-1,0)$	$h(0,0)$	$h(1,0)$
$h(-1,1)$	$h(0,1)$	$h(1,1)$

Figure 3.2. Spatial filtering of a 2D image $f(\mathbf{x})$ with a 3×3 filter kernel h.

mask or *kernel*, which covers a neighborhood N and defines a filtering operation that should be applied on all voxels in the neighborhood. To generate a filtered image g, the filter kernel h is moved over each voxel $\mathbf{x}$ in the input image f and the output value of the filtering operation assigned to $g(\mathbf{x})$. The process can be expressed with the *convolution* function

$$g(\mathbf{x}) = (f * h)(\mathbf{x}) = \sum_{\xi \in N} f(\xi) h(\mathbf{x} - \xi), \tag{3.1}$$

and is illustrated in Figure 3.2.

3.3 Noise reduction

Noise can suppressed by processing the CT image with a spatial smoothing filter, which will replace the intensity value of each voxel with a weighted average of its neighbors. This section reviews two common smoothing filtering techniques: Gaussian filtering; and bilateral filtering.

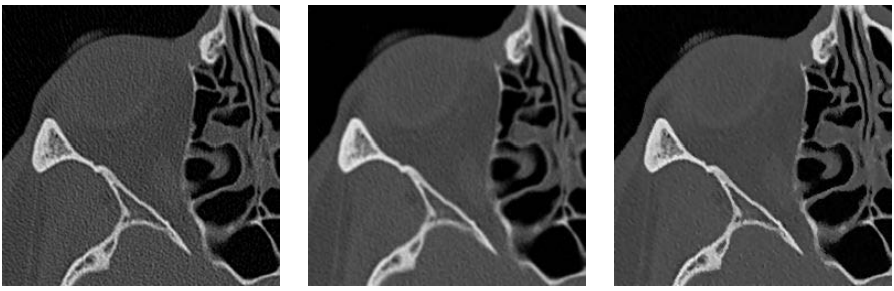
3.3.1 Gaussian filtering

The Gaussian filter kernel $G(\mathbf{x})$ is defined [44] as

$$G(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-||\mathbf{x}||^2/2\sigma^2}, \tag{3.2}$$

where σ denotes the standard deviation of the distribution. Given a noisy input image $f(\mathbf{x})$, a smoothed image $g(\mathbf{x})$ is obtained by convolving f with G as

$$g(\mathbf{x}) = \frac{\sum_{\xi \in N} f(\xi) G(\mathbf{x} - \xi)}{\sum_{\xi \in N} G(\xi)}. \tag{3.3}$$



(a) *(b)* *(c)*

Figure 3.3. Noise reduction: (a) original noisy CT image of the thin bone structures in the orbit; (b) Gaussian smoothing; (c) edge-preserving bilateral smoothing.

Figure 3.3b shows an example of applying the Gaussian filter on a noisy CT image (Figure 3.3a) of the orbit. The filter will assign higher weights to voxels that are close to center of the kernel, and the amount of blurring can be increased by increasing σ.

A desirable property of the Gaussian filter is that it is *separable* and can be implemented as n consecutive 1D convolutions to speed up computations, where n is the number of dimensions in the input image. The main disadvantage of the Gaussian filter is that it blurs boundaries and homogeneous regions equally much, which reduces the contrast between bone and other tissue types and might lead to loss of fine details such as thin bones.

3.3.2 Bilateral filtering

To prevent loss of fine details, the Gaussian filter can be replaced with an *edge-preserving* smoothing filter. One such filter is the bilateral filter [47], which performs filtering in both the spatial domain and intensity range of the image to replace each voxel with a weighted average of *close* and *similar* (with respect to intensity) neighbor voxels. The filter takes a grayscale image $f(\mathbf{x})$ as input and produces a smoothed image $h(\mathbf{x})$, defined as

$$h(\mathbf{x}) = \frac{\sum_{\xi \in N} f(\xi) c(\xi, \mathbf{x}) s(f(\xi), f(\mathbf{x}))}{\sum_{\xi \in N} c(\xi, \mathbf{x}) s(f(\xi), f(\mathbf{x}))}, \tag{3.4}$$

where $c(\xi, \mathbf{x})$ is the Gaussian closeness function

$$c(\xi, \mathbf{x}) = \frac{1}{2\pi\sigma_d^2} e^{-||\mathbf{x}-\xi||^2/2\sigma_d^2}, \tag{3.5}$$

and $s(f(\xi), f(\mathbf{x}))$ is the Gaussian similarity function

$$s(f(\xi), f(\mathbf{x})) = \frac{1}{2\pi\sigma_r^2} e^{-||f(\mathbf{x})-f(\xi)||^2/2\sigma_r^2}. \tag{3.6}$$

(a) *(b)* *(c)*

Figure 3.4. Edge detection: (a) original grayscale CT image $f(\mathbf{x})$; (b) gradient magnitude $||\nabla f(\mathbf{x})||$ computed with the central difference operator; (c) gradient magnitude computed with the Gaussian gradient operator.

The main advantage of the bilateral filter is that it is capable of smoothing homogeneous regions while preserving sharp boundaries, as demonstrated in Figure 3.3c. Unlike the Gaussian filter, however, the bilateral filter is not separable, making it substantially slower to compute for larger filter sizes. A multi-threaded or GPU-accelerated implementation is thus to prefer for interactive applications.

Other examples of edge-preserving smoothing filters that are used in medical image analysis are anisotropic diffusion [38] and the sigma filter [29], which is implemented in Paper VIII. The sigma filter smooths the image selectively by averaging only the neighbor voxels that are within a specified range (sigma) of the center voxel intensity. It has a lower computational cost than the bilateral filter and can efficiently suppress noise while preserving thin bone structures.

3.4 Edge detection

An *edge* in a digital volume image is a voxel $\mathbf{x}$ where the intensity function $f(\mathbf{x})$ changes abruptly. The *gradient* $\nabla f(\mathbf{x})$ indicates the local orientation of the edge and is often approximated with the central difference formula

$$\nabla f(\mathbf{x}) = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \approx \begin{bmatrix} \frac{f(x+1,y,z)-f(x-1,y,z)}{2\triangle x} \\ \frac{f(x,y+1,z)-f(x,y-1,z)}{2\triangle y} \\ \frac{f(x,y,z+1)-f(x,y,z-1)}{2\triangle z} \end{bmatrix}. \tag{3.7}$$

The *gradient magnitude* $||\nabla f(\mathbf{x})||$ indicates the strength of the edge and is defined as

$$||\nabla f(\mathbf{x})|| = \sqrt{g_x^2 + g_y^2 + g_z^2}. \tag{3.8}$$

(a) *(b)* *(c)*

Figure 3.5. Fundamental binary mathematical morphology operations performed with a 26-connected 3D structuring element: (a) original binary image; (b) dilation; (c) erosion.

A disadvantage of the central difference operator is that it is sensitive to noise, as demonstrated in Figure 3.4b. More reliable gradient estimations can be obtained by considering a larger neighborhood or incorporating smoothing [44] in the filtering; see Figure 3.4c for an example. Gradient operators are used in several of the volume visualization and segmentation techniques presented in Chapters 4 and 5.

3.5 Binary mathematical morphology

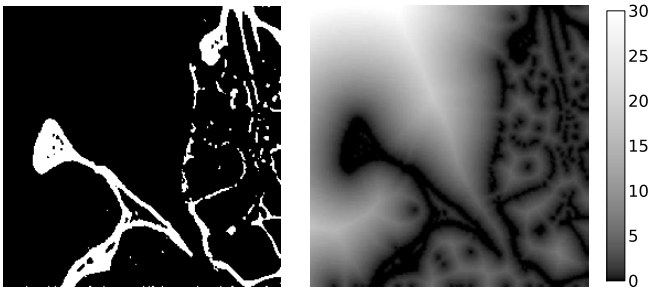
Binary mathematical morphology is a tool for analysing and processing objects in binary images [19]. The basic idea is to use small binary point sets, so-called *structuring elements*, to modify or probe a binary image for properties of interest. Two fundamental morpholgical operations are *dilation* and *erosion*. Given a binary image A and a binary structuring element B, which are subsets of the n-dimensional integer space Z^n, dilation is defined as

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\}, \tag{3.9}$$

and erosion as

$$A \ominus B = \{z \mid (B)_z \subseteq A\}. \tag{3.10}$$

Dilation expands structures (Figure 3.5b), whereas erosion shrinks them (Figure 3.5c). From dilation and erosion, more advanced morphological operations can be constructed, for example, closing, opening, and boundary extraction [19]. Morphological operations are used in some of the pre-processing steps in Papers I and VIII.



(a) *(b)*

Figure 3.6. Distance transform: (a) binary image to transform; (b) computed distance map showing the Euclidean distance (in mm) from every background voxel to the closest foreground (bone) voxel in the image.

3.6 Distance transforms

Distance transforms are used to compute the distance from every background voxel $\mathbf{x}$ in a binary image f to the closest foreground voxel in the same image, or vice versa, according to some distance metric. The result is stored in a grayscale image D_f, referred to as a *distance map* or *distance field*. See Figure 3.6 for an example. Distance fields can be *signed* or *unsigned*; in signed distance fields, the sign of a voxel indicates if the voxel is inside or outside the foreground, whereas unsigned distance fields only provide the distance to the foreground or foreground border.

The commonly used Chamfer 3-4-5 distance transform [8] offers a good trade-off between speed and accuracy, but assumes that voxels are isotropic and is not trivial to extend to the anisotropic case. Paper VIII uses, instead, the fast separable Euclidean distance transform proposed by Felzenszwalb and Huttenlocher [17], which is simple to parallelize and can be adapted to take voxel anisotropy into account, making it suitable for processing high-resolution CT datasets in interactive applications.

4. Volume visualization

To enable users to view and interact with the information in CT images, the 3D scalar data need to be transformed, at real-time speed, into a 2D projection that can be displayed on a computer screen or other type of display device. This can be accomplished by combining efficient volume visualization techniques [39] with the parallel processing power of modern graphics processing units (GPUs). A selection of volume visualization techniques that have been implemented and used in this work are described in the following sections.

4.1 Multi-planar reformatting

Multi-planar reformatting (MPR) is a simple yet effective visualization technique that uses arbitrarily oriented 2D planes to probe and display cross-sections (slices) of a volume image [39]. Medical volume images such as CT images are typically reformatted into three orthogonal planes, each of which is orthogonal to one of the image axes and can be translated along that axis by the user exploring the image. See Figures 2.3 and 4.1 for examples. By scrolling through the MPR slices, the user can inspect the volume data and, with some training, construct a mental image of the anatomical 3D structures.

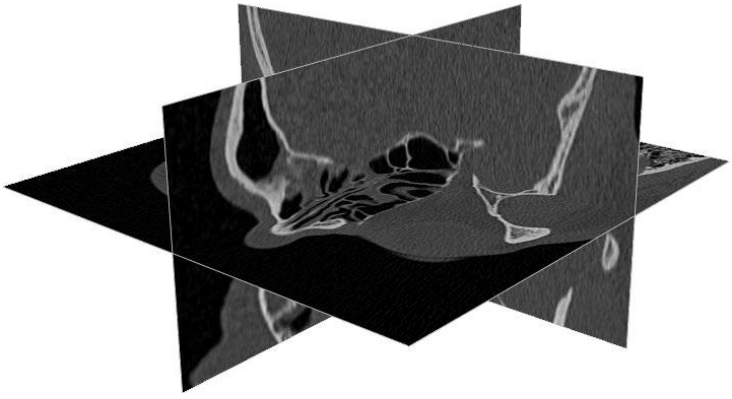
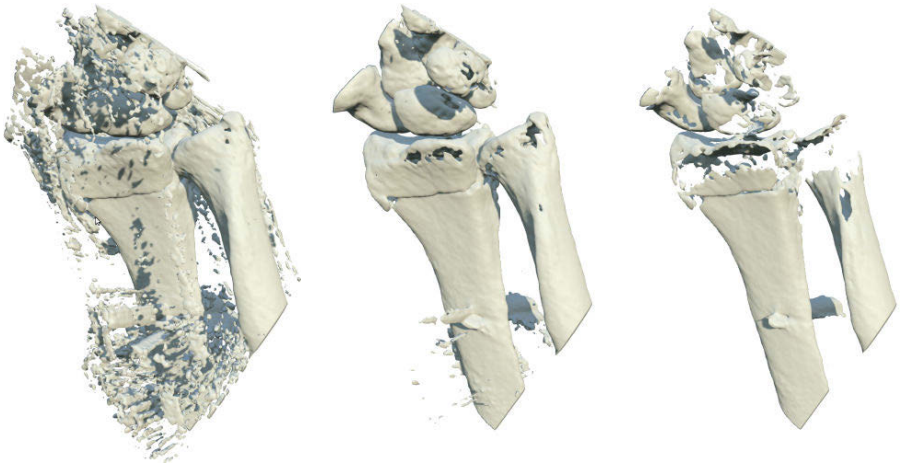


Figure 4.1. 3D multi-planar reformatting (MPR) visualization of a CT image of the facial skeleton. The user can explore the CT image by translating the three orthogonal planes along the image axes



(a) *(b)* *(c)*

Figure 4.2. Surface rendering of a wrist CT image with different HU isovalue settings: (a) 100 HU; (b) 260 HU; (c) 500 HU.

4.2 Surface rendering

Anatomical structures in volume images can be extracted as explicit *isosurfaces* (Figure 4.2), which are surfaces that intersect the volume data at selected intensity values (*isovalues*) and typically are represented and rendered as polygon meshes [9]. One of the most widely used and well-known isosurface extraction algorithm is the marching cubes algorithm [31], which uses a pre-defined lookup table with 33 intersection cases to quickly determine how the isosurface intersects a 2^3 grid cell that is marched through the volume. The *normals* of the isosurface are used for shading [1] and are typically derived from the gradients (Section 3.4) of the volume image. Mesh simplification algorithms [9] can be applied on the mesh to reduce the number of triangles and speed up the rendering. The mesh and underlying volume can also be processed with smoothing algorithms [9] to reduce noise and staircase artifacts in the extracted surface.

One advantage of surface rendering is that the extracted mesh can be rendered very efficiently using standard rasterization techniques [1]. A disadvantage of the technique, however, is that the mesh has to be rebuilt or updated every time the underlying volume data is modified. Another limitation of surface rendering is that it discards all intensity information outside the selected isovalue or isovalues. Similar to global thresholding segmentation (Chapter 5.1.1), an isovalue cannot completely separate bone from the rest of the CT image, so extracted isosurfaces tend to either include noise and soft tissue or exclude soft or thin bone structures.

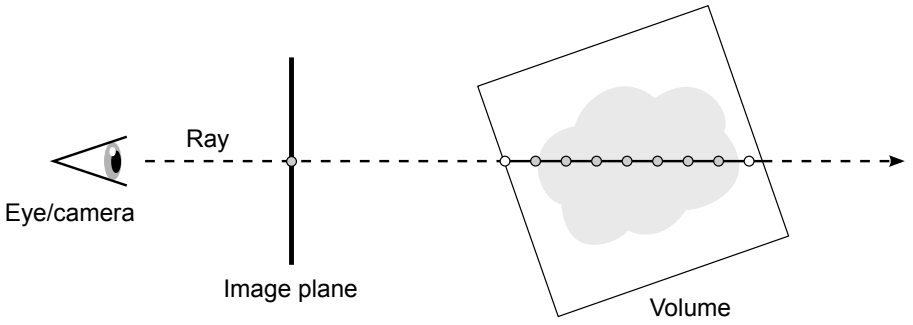


Figure 4.3. Ray-casting principle.

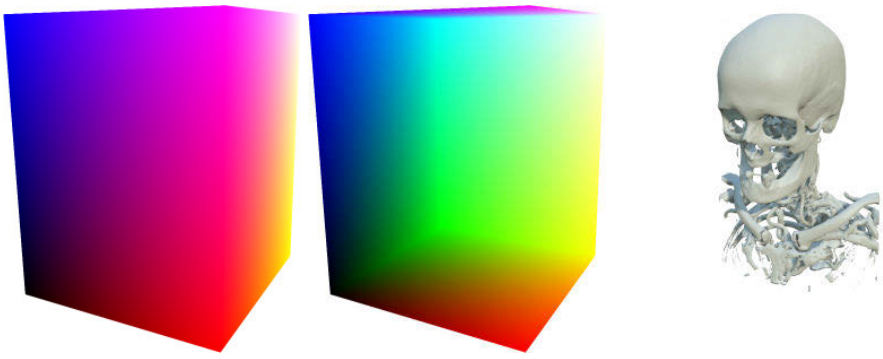
4.3 Direct volume rendering

In contrast to surface rendering techniques, direct volume rendering (DVR) techniques are able to generate projections of volume images directly from the scalar 3D data, without extracting explicit geometric surfaces. DVR techniques can be used for displaying both surfaces and semi-transparent interior structures of anatomical objects in CT images.

GPU-accelerated *ray-casting* [26] is an efficient, flexible, and widely used real-time DVR technique that can produce a wide variety of visualizations. The basic idea, illustrated in Figure 4.3, is to cast a ray from each *fragment* (pixel candidate [1]) in the viewport into the volume image and sample the image along the ray while mapping intensity values to color and opacity. The color and opacity values are accumulated and blended to produce a projection of the volume data. The ray is represented parametrically as $\mathbf{r}(t) = \mathbf{r}_0 + \mathbf{d}t$, where $\mathbf{r}_0 = (x_0, y_0, z_0)$ is the starting point and $\mathbf{d}$ is the normalized ray direction vector. t is initialized to 0 and incremented with some pre-defined step length $\triangle t$ until the ray has passed through the volume.

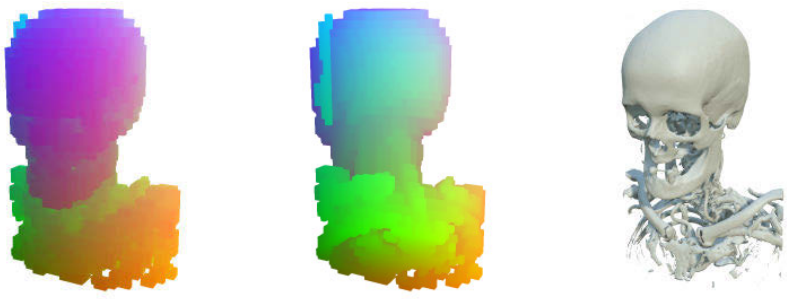
Before the ray-casting starts, the volume image is uploaded as a 3D *texture* [1] to the GPU memory. To obtain the ray starting points for all fragments, the front faces (Figure 4.4a) of the volume image's bounding box are rendered to a 2D RGB texture, encoding the 3D texture coordinates as RGB colors. The ray end points are obtained in a similar manner by rendering the back faces of the bounding box to a second 2D RGB texture (Figure 4.4a). Ray-casting and sampling is then performed by a fragment *shader* (that is, a small program that is executed on the GPU) to create a projection of the volume image.

Ray-casting is computationally expensive due to the large number of 3D texture samples that has to be taken along each ray. One way to speed up the rendering and reduce the number of samples is to use empty-space skipping block structures [26] to identify and skip empty regions of the volume; see Figure 4.5 for an example. For rendering of bone structures, which typically occupy only a small fraction of the total volume, empty-space skipping can provide a substantial speedup.



(a) *(b)* *(c)*

Figure 4.4. GPU-accelerated ray-casting setup: (a) ray-start positions obtained by rendering the front faces of the volumes bounding box and encoding the positions as RGB colors; (b) ray-end positions obtained in a similar manner by rendering the back faces of the bounding box; (c) ray-casting result.



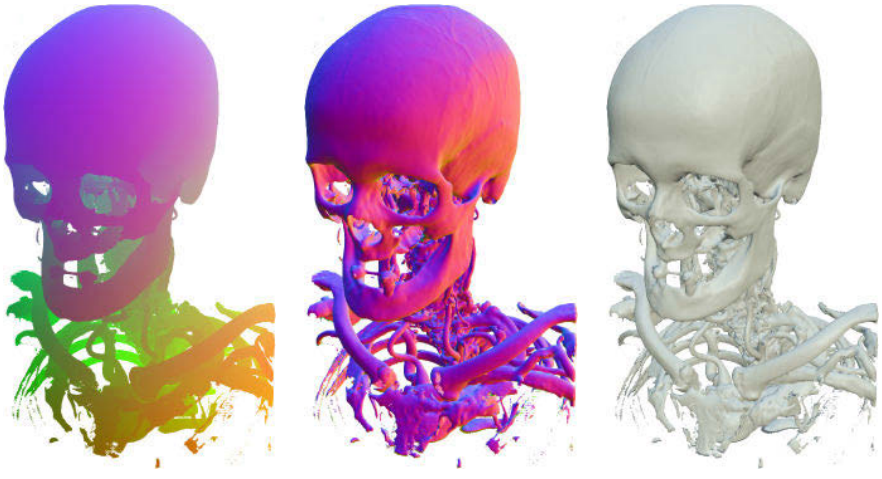
(a) *(b)* *(c)*

Figure 4.5. Ray-casting accelerated with empty-space skipping: (a) ray-start positions; (b) ray-end positions; (c) ray-casting result (identical to Figure 4.4c).

Maximum intensity projection (MIP) (Figure 4.6a) is a simple but often used ray-casting mode that projects the brightest value found along each ray. The resulting image is easy to interpret and is useful for obtaining an overview of the volume data, but lacks depth cues. *Front-to-back compositing* is another commonly used technique that maps, through a so-called *transfer function* [39], voxel intensity to color and opacity and creates a semi-transparent projection (Figure 4.6b) by accumulating opacity values along the ray while compositing (blending) colors in front-to-back order. *Isosurface rendering*, which is the ray-casting technique that has mainly been used in this thesis work, extracts the first sample along the ray that has an intensity value greater than or equal a pre-defined isovalue and uses the gradients of the image for

(a) *(b)*

Figure 4.6. Ray-casting modes: (a) maximum intensity projection (MIP); (b) semi-transparent front-to-back compositing rendering with local illumination and shadows.



(a) *(b)* *(c)*

Figure 4.7. DVR isosurface rendering: (a) first-hit positions encoded as RGB colors; (b) surface normals encoded as RGB colors; (c) local illumination computed from the surface information in (a) and (b).

shading computations (Figure 4.7). An advantage of DVR-based isosurface rendering over explicit surface rendering is that no intermediate polygon mesh representation has to be extracted.

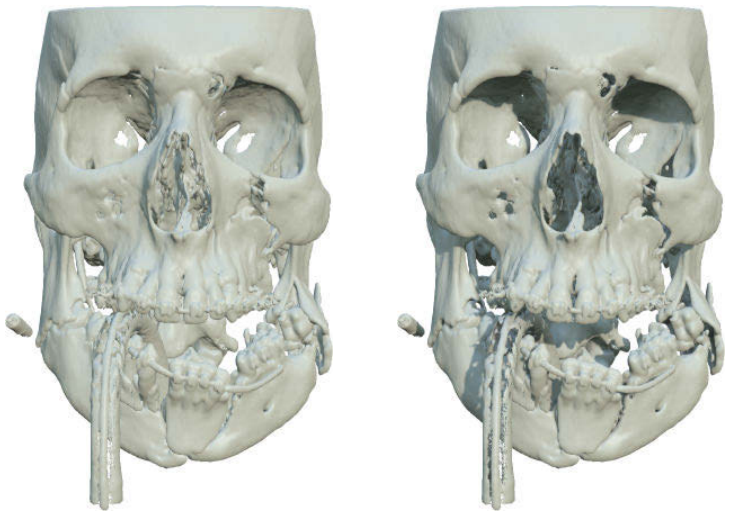


Figure 4.8. The importance of shadows for perceiving depth and spatial relationships between bone structures in a complex trauma case: (a) local illumination only; (b) local illumination with cast shadows (shadow mapping) and ambient occlusion.

Local illumination [1] can be computed with, for example, the Blinn-Phong shading model [6] to simulate how light is reflected on surfaces. For isosurface rendering, the local illumination can be combined with shadow mapping [40] and ambient occlusion [28] to produce shadow depth cues that make it easier for the user to perceive spatial relations between, for example, bones and bone fragments, as demonstrated in Figure 4.8. Similar, but more computationally expensive volumetric shadow rendering techniques [22] can be used for semi-transparent renderings, as shown in Figure 4.6b.

4.4 Haptic 3D interaction

When interacting with real-life 3D objects, the sense of touch (*haptics*) is an important addition to the visual perception. The development of haptic 3D input devices and haptic rendering techniques [32], combined with stereoscopic displays that convey strong depth cues by means of stereopsis [10], has made it possible to create 3D user interfaces where the user can touch, feel, and manipulate virtual objects in an intuitive way (Figure 4.9). These types of interfaces are highly useful for virtual surgery planning [36] and can also facilitate interactive segmentation of volume images [51]. In this thesis, haptics and stereoscopic displays have been utilized in the segmentation and measurement methods in Papers I and II.

A haptic device (Figure 4.10) is usually constructed as a small robot arm with a stylus that the user holds on to. Haptic feedback is given at the haptic probe, an interaction point located at the tip of the stylus or at the center of the

Figure 4.9. A stereoscopic mirror display with head-tracking and haptic 3D input. The system makes it easier for the user to perceive depth in the volume image and facilitates complex 3D interaction tasks such as navigation and selection.

Figure 4.10. A Sensable Phantom Omni haptic device with six degrees of freedom (6-DoF) input and three degrees of freedom (3-DoF) output.

ball. The number of independent translations or rotations a haptic device can monitor and control (as input or output) are referred to as its degrees of freedom (DoF). The feedback generated by a haptic device can be kinesthetic or tactile. Kinesthetic feedback refers to directional forces, whereas tactile feedback refers to sensations felt by the skin, for example, texture and vibrations.

Haptic rendering is the process of computing and generating kinesthetic or tactile feedback in response to a user's interaction with virtual 3D objects through the haptic device. To avoid vibrations, the haptic rendering must be updated at 1 KHz or higher. In the haptic rendering implementation used in this thesis, kinesthetic feedback is rendered by connecting the haptic probe to a virtual proxy that is constrained by volume image gradients. See [51] for further details.

5. Interactive segmentation

Image segmentation is the process of separating objects of interest from the rest of the image. It can be divided into two fundamental tasks [48]: *recognition*, which is the task of determining the approximate location of an object of interest; and *delineation*, which is the task of marking the exact extent of the object. Humans tend to outperform computers on the recognition task, whereas computers often are more efficient and precise at performing delineation. Semi-automatic or interactive segmentation methods [16, 14, 33] take advantage of these complementing abilities by combining imprecise user input with efficient and precise delineation algorithms to produce accurate segmentation results with a limited amount of user interaction. This chapter covers several interactive segmentation techniques that can be applied for bone or bone cavity segmentation in CT images.

5.1 Thresholding

The basic concept of thresholding is that an object of interest can be separated from the rest of the grayscale image by selecting all voxels above or below a certain *intensity threshold* [19]. For CT images, the HU scale can be used as guideline to find thresholds that will separate different tissue types. This section describes two common thresholding techniques: global thresholding; and hysteresis thresholding.

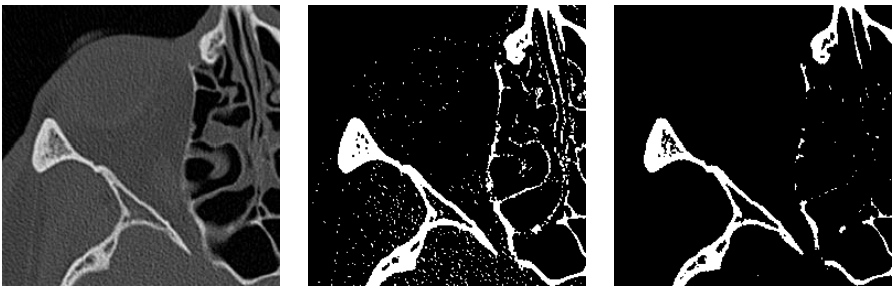
5.1.1 Global thresholding

Global thresholding [19] is a simple, fast, and widely used segmentation technique that can be use to extract bright objects (bone) from a darker background (air and soft tissue), or vice versa. Given a grayscale volume image $f(\mathbf{x})$, a binary segmentation $g(\mathbf{x})$ is generated by selecting a threshold t corresponding to the lowest intensity value of the object of interest and thresholding f at t as

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } f(\mathbf{x}) \geq t, \\ 0 & \text{otherwise,} \end{cases} \tag{5.1}$$

where 1 represents object and 0 background. See Figure 5.1 for an example.

A major limitation of the global thresholding technique is that it cannot separate objects with overlapping intensities. This becomes a problem when segmenting thin or soft bone structures, which, due to noise and PVE, often have similar intensity as soft tissue.



(a) *(b)* *(c)*

Figure 5.1. Global thresholding of thin orbital bone structures in a grayscale CT image: (a) original image; (b) thresholding result for $t = 200$ HU; (c) thresholding result for $t = 400$ HU.

5.1.2 Hysteresis thresholding

To separate low-intensity bone voxels from noise and soft tissue, it is necessary to employ segmentation methods that not only consider intensity values but also spatial relationships between voxels [50]. Hysteresis thresholding [11] is one such technique that uses dual thresholds, t_{low} and t_{high}, to first extract high-intensity bone voxels and then performs a connectivity analysis to find all low-intensity bone voxels that are connected to high-intensity bone voxels. See Figure 5.2 for an example. An efficient queue-based implementation of hysteresis thresholding is described in Algorithm 1.

Hysteresis thresholding is implemented in Papers I, IV, VII, and VIII to segment bone structures in the orbit and wrist.

(a) *(b)* *(c)*

Figure 5.2. Hysteresis thresholding applied on the CT image in Figure 5.1: (a) low-intensity bone voxels extracted with threshold $t_{low} = 200$ HU; (b) high-intensity bone voxels extracted with threshold $t_{high} = 400$ HU; (c) hysteresis thresholding segmentation obtained by removing low-intensity bone voxels that are not connected to high-intensity bone voxels.

Algorithm 1: Hysteresis thresholding

Input: Grayscale CT image f, low threshold t_{low}, high threshold t_{high}
Output: Binary segmentation g
Auxiliary: Queue Q

$Q \leftarrow \emptyset$;
/* Find high-intensity bone voxels and push them on Q */
for each voxel $\mathbf{x} \in f$ **do**
    if $f(\mathbf{x}) < t_{high}$ **then**
        $g(\mathbf{x}) \leftarrow 0$;
    else
        $g(\mathbf{x}) \leftarrow 1$;
        push $\mathbf{x}$ on Q;
/* Find low-intensity bone voxels that are connected to high-intensity bone voxels */
while Q is not empty **do**
    pop $\mathbf{x}$ from Q;
    for each neighbor voxel $\mathbf{y} \in N_{26}(\mathbf{x})$ **do**
        if $f(\mathbf{y}) \geq t_{low}$ and $g(\mathbf{y}) = 0$ **then**
            $g(\mathbf{y}) \leftarrow 1$;
            push $\mathbf{y}$ on Q;

5.2 Random walks segmentation

Random walks [21] is semi-automatic image segmentation method that takes user-defined markers (*seeds*) as input to segment objects of interest. For every voxel, the algorithm calculates the probability that a "random walker" starting at the voxel will reach a particular seed label. The input image is represented as a weighted graph $G = (V, E)$, where the vertices $v \in V$ represent the voxels and the edges $e \in E$ represent the connections between adjacent voxels in a 6-connected neighborhood. Each edge e_{ij} between two neighbor vertices v_i and v_j is assigned a gradient magnitude-based weight w_{ij}, defined as

$$w_{ij} = \exp(-\beta(g_i - g_j)^2) + \varepsilon, \tag{5.2}$$

where g_i and g_j are the intensities of v_i and v_j in the underlying grayscale image, β is a parameter that determines the influence of the gradient magnitude, and ε is a small positive constant added to ensure that v_i and v_j are connected, that is, $w_{ij} > 0$. Increasing the value of β makes the random walkers less prone to traverse edges with high gradient magnitude.

The graph and seed nodes are used to construct a sparse linear system, which can be solved efficiently using multi-threaded or GPU-accelerated [3] iterative solvers [43]. A labelling is obtained by, for each voxel, selecting the label with the highest probability value. The segmentation result can be edited

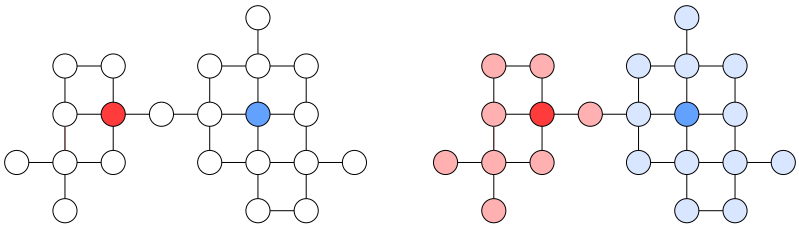


Figure 5.3. Concept of using random walks for separating two connected objects in a graph. The algorithm takes a weighted graph and user-defined seeds (red and blue) as input and produces a labelling by, for each vertex in the graph, computing the probability that a random walker starting at the vertex will reach a particular seed.

interactively by adding more seeds and running the solver again, using the previous solution as starting guess to speed up computations.

Figure 5.3 illustrates the general concept of using random walks for segmenting objects in graphs. The main advantages of the random walks method are that it enables fast editing and update of the segmentation result, is robust to noise and weak boundaries, and extends easily to multi-label (K-way) segmentation. A drawback of the method, however, is its high computational and memory cost.

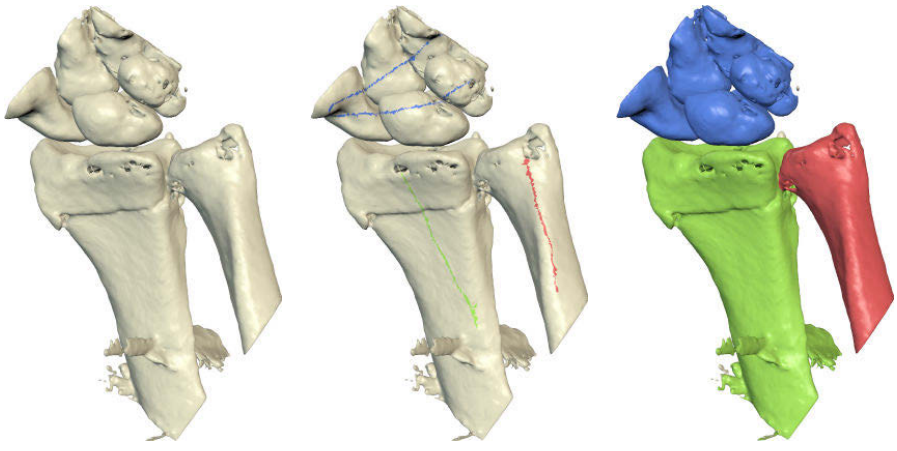
A GPU-accelerated random walks algorithm is adapted for interactive bone segmentation in Paper VII.

5.3 Mesh segmentation

Mesh segmentation techniques are used for labelling and decomposing polygon meshes into sub-components [27]. They are frequently used in 3D modelling and computer-aided design (CAD) applications, but can also be applied on polygon meshes that have been extracted from CT data, as demonstrated in Figure 5.4. Interactive mesh segmentation techniques usually provide intuitive sketch- or landmark-based interfaces that make it easy for the user to mark objects of interest and edit the segmentation result.

Random walks mesh segmentation [27] is based on the same idea as the random walks method for image segmentation, but operates on a graph constructed from a surface mesh and uses surface curvature information instead of gradients to detect object boundaries. The mesh faces $f_1, \ldots, f_m$ represent the nodes in the graph, and the weights between the nodes are set so that the random walker becomes less prone to traverse concave edges, which are assumed to correspond to object boundaries. The probability that a random walk starting from face f_k will reach a seed s_l is given by the equation

$$P^l(f_k) = \sum_{i=1}^{\kappa} p_{k,i} P^l(f_{k,i}), \tag{5.3}$$



(a) (b) (c)

Figure 5.4. Sketch-based random walks mesh segmentation applied on a polygon mesh extracted with marching cubes from CT data: (a) original mesh; (b) user-defined seeds painted on bones that should be separated; (c) segmented bones.

where $f_{k,i}$ denotes a neighbor face sharing an edge $e_{k,i}$ with f_k, and κ is the number of neighbors. A random walker at face f_k will move across an edge $e_{k,i}$ to a neighbor $f_{k,i}$ with the probability

$$p_{k,i} = |e_{k,i}| \exp\left\{-\frac{d_1(f_k, f_{k,i})}{\sigma}\right\}, \tag{5.4}$$

where

$$d_1(f_k, f_{k,i}) = \frac{\eta}{2}\|N_k - N_{k,i}\|^2 \tag{5.5}$$

is a difference function measuring the dihedral angle between f_i and $f_{k,i}$, and σ is a weight (typically set to $\sigma = 1$) used to control the influence of d_1. N_k and $N_{k,i}$ denote the face normals, and η is a weighting term set to $\eta = 1.0$ for concave edges and $\eta = 0.2$ for convex edges to make the random walker less prone to traverse concavities. $P^l(f_k)$ is computed by representing and solving Equation 5.3 as sparse linear system, using the same type of iterative solvers as in Section 5.2. A labelling is then obtained by, for each mesh face, selecting the seed label with the highest probability. See [27] for more details.

Paper IV implements the random walks mesh segmentation technique to separate individual bones in the wrist. One limitation to have in mind when using mesh segmentation for this purpose is that the segmented objects generally cannot easily be converted back to solid volumetric representations, since the decomposed meshes are no longer watertight 2-manifold surfaces.

5.4 Deformable surface models

Deformable surface models [46] may be thought of as elastic surfaces that are defined in the image domain and driven by the minimization of a cost function, of which local or global minima correspond to object boundaries. They are capable of segmenting objects of widely different shapes and are also robust to noise and boundary gaps, which have made them popular tools for segmenting anatomical structures in CT images.

The deformable surface models used in this thesis work are based on the implementation by Vidholm [51] and applied in Paper I for segmenting the orbit in CT images. The following sections provide a brief overview of how the models are represented and deformed.

5.4.1 Geometric representation

Several different geometric representations of deformable surface models have been proposed [34]. The representation used in this thesis is the deformable *2-simplex mesh* introduced by Delingette [15], which is illustrated in Figure 5.5a. Two characteristics of 2-simplex meshes are that every simplex mesh vertex $\mathbf{p}_i$ has exactly three neighbor vertices $(\mathbf{p}_{N_1(i)}, \mathbf{p}_{N_2(i)}, \mathbf{p}_{N_3(i)})$ and that every simplex mesh has a dual triangle mesh. The three neighbor vertices define the surface normal vector $\mathbf{n}_i$ at $\mathbf{p}_i$ and also the simplex angle Φ, which describes the local curvature of the mesh at $\mathbf{p}_i$ [15].

5.4.2 Deformation

The simplex mesh is deformed according to the force formulation

$$m\frac{d^2\mathbf{p}_i}{dt^2} = -\gamma\frac{d\mathbf{p}_i}{dt} + F_{int}(\mathbf{p}_i) + F_{ext}(\mathbf{p}_i), \tag{5.6}$$

where m is the vertex mass unit, t is the time, γ is a damping factor, and F_{int} and F_{ext} are *internal* and *external* forces, respectively [15]. The external force can either be a single force or a weighted sum of several different forces, which also holds for the internal force.

Before the deformation starts, the deformable model needs to be initialized so that its size, shape, and position roughly matches the target object. Otherwise, the model might get stuck in local minima or fail to converge to the target boundary. The initialization can be performed interactively by, for example, asking the user to position the model as a coarse sphere (Figure 5.5b) or other simple primitive on the target object.

Internal forces

Internal forces control the smoothness and elasticity of the model [15]. They help keeping the mesh together and prevent it from leaking through boundary gaps or forming self intersections.

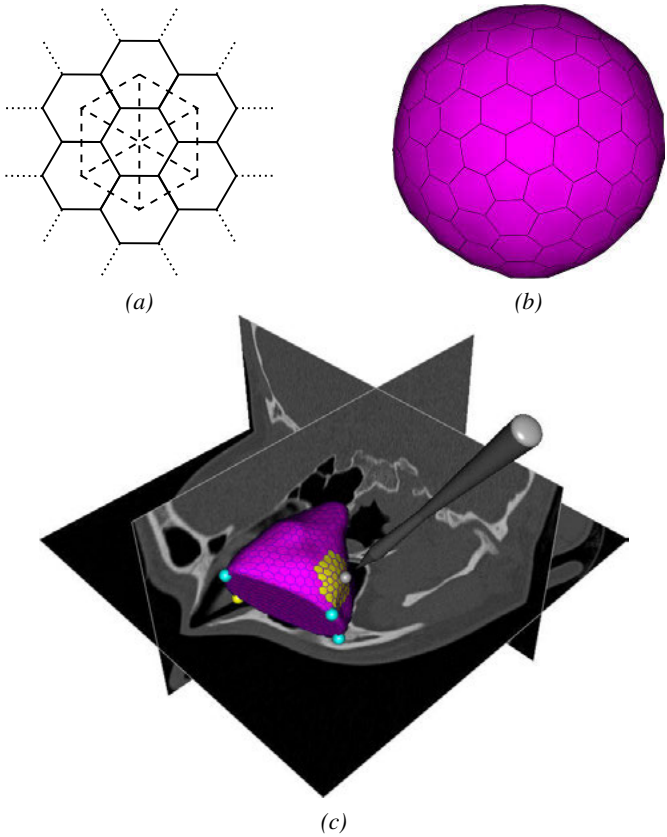


Figure 5.5. (a) A simplex mesh and its dual triangle mesh. (b) Simplex mesh sphere. (c) Interactive segmentation performed with a deformable simplex mesh. The mesh is deformed by external forces computed from the CT image and by interactive forces that the user applies on selected mesh faces (yellow) with a haptic 3D input device.

External forces

External forces attract the deformable model to the boundaries of the object of interest. The external force computation is usually performed in two steps: first by extracting a boundary map from the underlying CT image, and then by computing the external force from the boundary map according to some force definition. Examples of commonly used external forces are gradient vector flow (GVF) [53] and vector field convolution (VFC) [30]. GVF diffuses gradient vectors derived from the input image to produce a vector field that will attract the deformable model towards boundaries, whereas VFC produces a similar but less noise-sensitive force by convolving the gradient vectors with a vector field kernel.

Another frequently used external force is the distance potential force [13], which uses the gradients of a distance field to drive the model towards object

boundaries. Given a binary boundary map f and a distance map D_f computed from f, the distance potential force $\mathbf{f}_{dp}$ is computed as

$$\mathbf{f}_{dp}(\mathbf{x}) = -\nabla D_f(\mathbf{x}) \tag{5.7}$$

The main advantages of the distance potential force is that it is cheap to compute and offers a long capture range. Compared with GVF and VFC, the distance potential force is more sensitive to noise and lacks the ability to deform the model into narrow boundary concavities. However, the latter property can actually be an advantage when segmenting objects that have large boundary gaps. Distance potential forces are used in Paper I.

Interactive forces
Interactive forces can be incorporated in the external force term to enable the user to guide and correct the model interactively if the image-based forces fail. Two examples of such forces are user-positioned landmarks that the model has to pass through (spring forces) or avoid (volcano forces) [53].

In this thesis, haptic interactive forces [51] have been used to guide and correct the deformable simplex mesh. The user interacts with the 3D scene via a haptic 3D input device and can pull or push selected mesh faces toward the object boundary with the haptic stylus to correct the segmentation result (Figure 5.5c). Haptic force feedback enables the user to feel how much force he or she is applying on the mesh during the interaction.

5.5 Segmentation evaluation

Segmentation evaluation is the process of assessing and comparing the performance of different segmentation methods in a specific application domain [49]. It is an essential part of method development and needs to be performed in a standardized and systematic manner to yield comparable results.

Unified frameworks for segmentation evaluation have been proposed by, for example, Chalana and Kim [12] and Udupa et al. [49]. The latter framework, which has been applied in Papers I and VIII, is capable of handling both crisp (binary) and fuzzy (graded) segmentations and states that segmentation methods should be compared based on three factors: *precision*, *accuracy*, and *efficiency*. The following sections describe these factors in detail.

5.5.1 Precision

Segmentation precision describes the degree of repeatability, that is, how sensitive a segmentation method is to user input. The precision of a segmentation method M can be measured as

$$PR^M(O) = \frac{|C^M_{O_1} \cap C^M_{O_2}|}{|C^M_{O_1} \cup C^M_{O_2}|}, \tag{5.8}$$

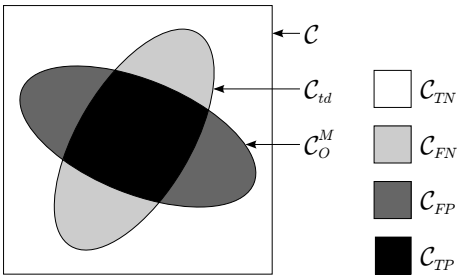


Figure 5.6. Illustration of segmentation accuracy factors. C_{td} represents the true segmentation, whereas C_O^M represents a segmentation obtained with some method M. C_{TN}, C_{FN}, C_{FP}, and C_{TP} represent the true negative, false negative, false positive, and true positive volume fractions, respectively.

where $C_{O_1}^M$ and $C_{O_2}^M$ are two segmentations of an object O in some image C. $PR^M(O)$ is also known as the Jaccard index and measures the spatial overlap between the two segmentations. It ranges from 0 to 1, where 0 indicates no overlap and 1 perfect overlap.

Intra-operator precision is assessed by having one user segment the same object in the same image twice. Similarly, inter-operator precision is assessed by having two users segment the same object in the same image once.

5.5.2 Accuracy

Segmentation accuracy describes how close a segmentation C_O^M matches the ground truth C_{td}. Since absolute true segmentations are generally not available for real CT datasets, manual segmentations performed by experts are often used as a replacement for ground truth. To make the assessment less subjective, multiple manual segmentations of the same object performed by different experts can be combined into a crisp or fuzzy ground truth segmentation [49].

The framework in [49] evaluates accuracy by measuring two entities: *sensitivity* and *specificity*. Sensitivity is defined as the true positive volume fraction

$$TPVF_d^M(O) = \frac{|C_{TP}|}{|C_{td}|}, \tag{5.9}$$

whereas specificity is defined as one minus the false positive volume fraction

$$1 - FPVF_d^M(O) = 1 - \frac{|C_{FP}|}{|C_{td}|}. \tag{5.10}$$

The closer to one these entities are, the more accurate is the segmentation. Figure 5.6 illustrates the different accuracy factors.

Other commonly used comparison measures are the previously mentioned Jaccard index, the Dice similarity coefficient [54], which is similar to the Jaccard index but assigns higher weight to true positives, and surface distance

measures such as the symmetric mean absolute (MAD) and max absolute (Hausdorff) distances [12]. Volume agreement is often assessed with Bland-Altman plots [5]. These measures and techniques can be used for precision assessment as well.

5.5.3 Efficiency

Segmentation efficiency describes the practical usability of a method in terms of time and cost. To perform a full efficiency evaluation, one first needs to record the training times, user interaction times, and computational times, and then define a function that converts these time factors into a cost. In this thesis work, only the time factors have been evaluated.

6. Registration and analysis

This chapter describes techniques for comparing and analysing segmented bones or bone cavities.

6.1 Iterative closest point registration

A standard technique for aligning (*registering*) bone surface models extracted from CT data is the iterative closest point (ICP) [4] algorithm. Given a *target model* and a *source model* (Figure 6.1) that should be registered, the ICP algorithm takes an initial coarse alignment of the models as starting guess and then refines the registration iteratively in the following steps:

1. For each point or vertex on the source model, find the closest corresponding point on the target model.
2. Compute a rigid-body transformation (translation and rotation) that minimizes the mean distance between the corresponding points.
3. Apply the transformation on the source points.
4. Repeat steps 1-3 until the mean distance falls below a given threshold.

The starting guess can be generated automatically by, for example, matching the centroids of the models. Alternatively, the initialization can be performed interactively by asking a human user to identify corresponding landmarks on the models, which are then aligned with Procrustes analysis [20].

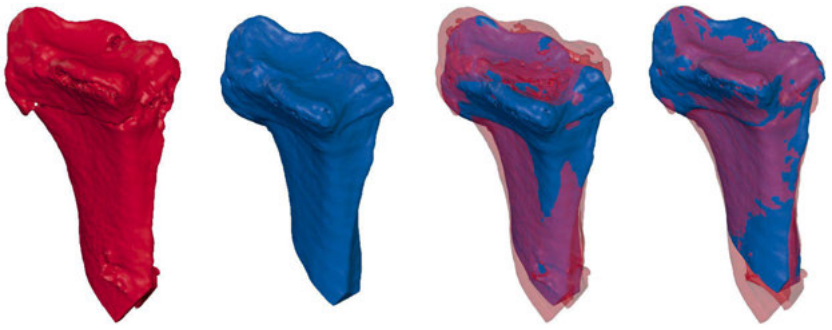


Figure 6.1. Registration of bone surface models extracted from CT data: (a) target model; (b) source model; (c) initial coarse registration obtained by matching the centroids of the models; (d) refined registration obtained with the ICP algorithm.

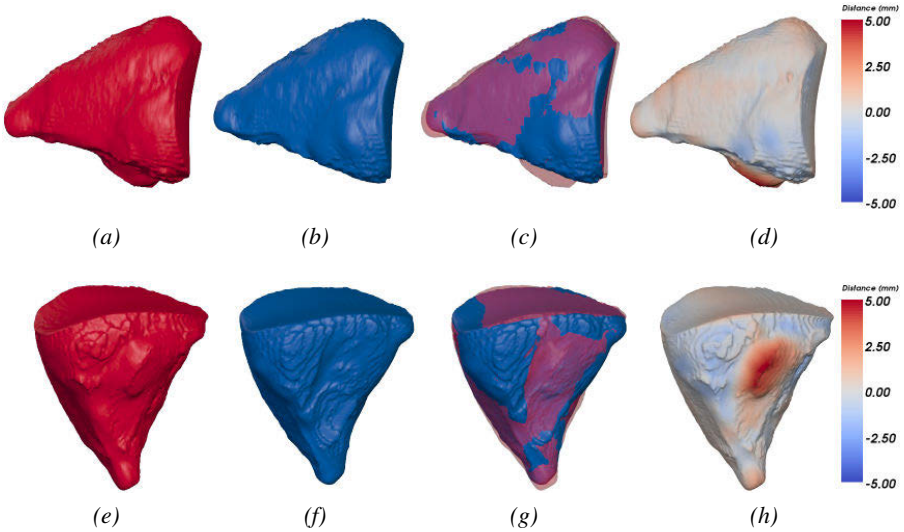


Figure 6.2. Shape comparison of an injured orbit (red) and a contralateral intact orbit (blue) that has been mirrored and registered onto the injured side. The semi-transparent overlays in the third column shows the spatial overlap between the surface models, whereas the color-coded distance maps in the right column shows the signed surface-to-surface distance (in mm).

ICP requires a reasonably good starting guess in order to converge, which is one of the main weaknesses of the algorithm. Another issue with the original ICP algorithm is that it performs poorly if the input models are noisy or only partially overlapping. Moreover, the original ICP algorithm does not take surface normals into account and might therefore align points with opposite normal directions. The latter two issues have been addressed in later implementations [41] by rejecting a certain amount of point pairs as *outliers* (false matches) based on distance and/or surface normal direction, which improves the accuracy and robustness.

An ICP algorithm with outlier rejection is implemented in Papers IV and VII to register fractured wrist bones in CT images.

6.2 Shape analysis

The shape of two segmented and registered objects can be compared visually by generating semi-transparent overlays and computing and visualizing the signed and unsigned surface-to-surface distance [2] between the registered meshes, as illustrated in Figure 6.2. These types of visualizations are useful for comparing pre- and post-operative results or mirroring an injured side onto an intact side. A diverging colormap [35] is typically used to encode negative distances as blue, intersection as white, and positive distances as red.

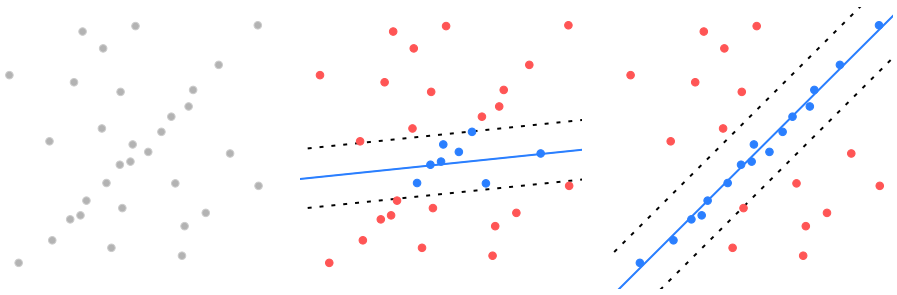


Figure 6.3. Basic concept of using RANSAC for fitting a model (in this case, a line) to observed data that contains outliers. Inliers are displayed in blue and outliers in red.

Additionally, quantitative shape analysis can be performed in a similar manner as segmentation evaluation (Chapter 5.5) by measuring spatial overlap and, for example, the symmetric mean and max absolute surface distances.

6.3 RANSAC

Random sample consensus (RANSAC) [18] is a robust iterative method for fitting a mathematical model, for example, a line or a plane, to observed data that contains outliers. The basic algorithm consists of the following steps:

1. Select, at random, n points from the input data, where n is the minimum number of points required to represent the model.
2. Fit the model to the n points.
3. Determine the *inlier* or *consensus* set, that is, the set of points in the input data that fits the model well (within a given error tolerance).
4. If the inlier ratio is high enough, fit the model to all inliers and terminate.
5. Otherwise, repeat steps 1–4 or terminate if the maximum number of iterations N has been exceeded.

Figure 6.3 illustrates the concept. A RANSAC-based method is used in Papers IV and VII to determine the long-axis of wrist bones.

Two drawbacks of the original RANSAC algorithm, due to its random nature, are (1) that it can get stuck in local minima even when the inlier ratio is high and (2) that it performs poorly when the inlier ratio is less than 50%. In Paper V, a modified RANSAC algorithm is presented that will find the optimal set in every run for certain applications and can be applied on datasets with inlier ratios as low as 5%.

7. Thesis contributions

This chapter summarizes the methods and results presented in the appended Papers I–VIII.

7.1 Orbit segmentation

Measuring the shape and volume of the bony *orbit* (eye-socket) in 3D CT images is a highly useful tool for diagnosis, planning, and evaluation of orbit surgery, as well as for studying the orbital morphology. To measure these properties, however, the orbit first needs to be segmented. The conventional method for this is to manually trace the orbit boundary in individual CT slices and combine the 2D contours into a 3D volume [42]. This approach generally produces accurate results, but is very tedious and time-consuming and thus not suitable for routine clinical practice or studies on large patient groups.

Automatic or semi-automatic orbit segmentation techniques can greatly reduce the segmentation time and improve repeatability [45, 52, 23], but are, for several reasons, difficult to make accurate and robust enough for clinical usage. Image imprecisions such as noise and PVEs make it difficult to separate the thin orbital bones from air and soft tissue, and the orbit also has natural holes and fissures that make segmentations prone to leak. Another key challenge is to define and mark the anterior (frontal) boundary of the orbit in a consistent and anatomically meaningful way [37]. Lastly, it is difficult to identify and mark the bone and soft tissue boundaries in fractured regions. The two interactive segmentation methods presented in this section attempt to overcome these challenges.

7.1.1 A semi-automatic orbit segmentation method based on deformable models with haptic 3D interaction

Paper I presents a semi-automatic orbit segmentation method that combines deformable models with haptic interaction. In brief, the orbit is segmented by

1. enclosing the orbital opening with a bounding plane fitted to four user-defined landmarks,
2. segmenting the bone structures in the skull and the orbit with hysteresis thresholding,

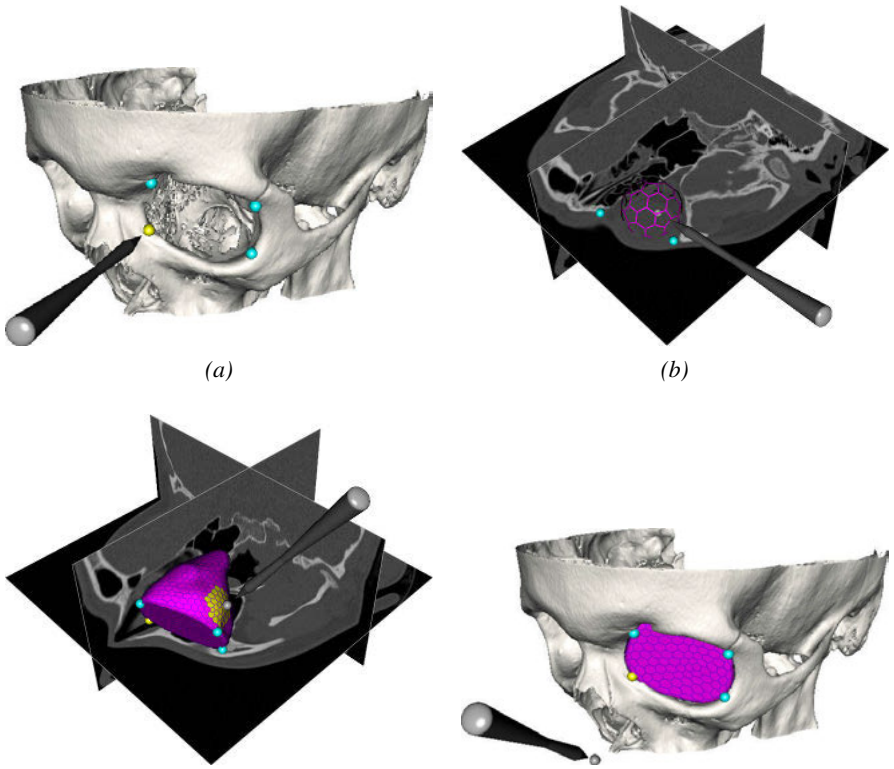
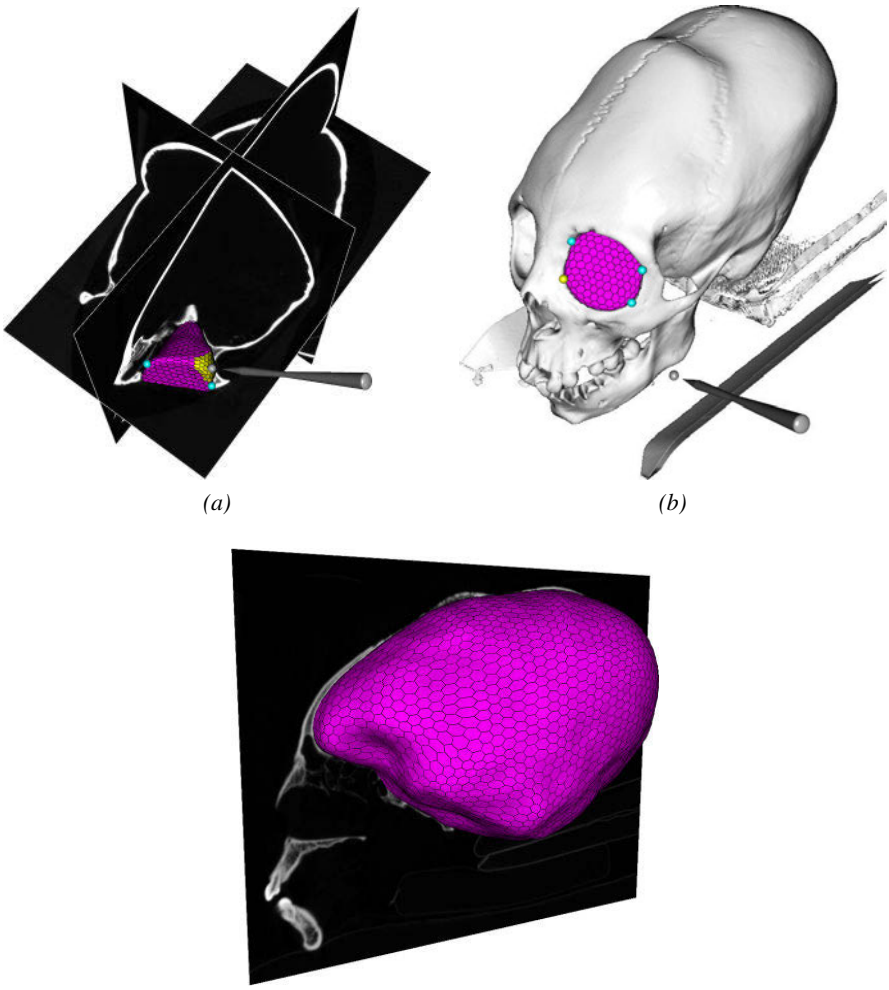


Figure 7.1. Overview of the deformable model-based orbit segmentation process: (a) landmark positioning performed with the haptic input device to mark the extent of the anterior (frontal) opening; (b) simplex mesh initialization; (c) mesh deformation and haptic-aided segmentation correction; (d) segmented orbit.

3. computing a signed distance field from the segmented bones and the bounding plane, and
4. fitting a deformable simplex mesh to the enclosed bone cavity using the gradients of the distance field as external force.

Figure 7.1 illustrates the interactive steps in the segmentation process. A haptic 3D input device is used to initialize, guide, and correct the deformable simplex mesh during the segmentation. Hysteresis thresholding enables extraction of thin (in some areas only a few tenths of a millimeter thick) orbital bone structures that have been blurred due to the PVE, and the distance-based external force provides a long capture range while preventing the deformable model from leaking through boundary gaps.



(a) *(b)*

(c)

Figure 7.2. Deformable model-based segmentation of the orbits and cranial vault in a dry skull affected by intentional cranial deformations.

Results

The method was evaluated on 14 orbits (both normal and fractured) by three users and found to be accurate, precise, and substantially faster than manual segmentation. The semi-automatic segmentations took on average 3 minutes to perform, compared with 17 minutes for the manual segmentations. Minor leaks into the nasal cavity and sinuses occurred, but could be corrected interactively with the haptic device.

Additionally, the method was applied in Paper III to segment the orbits, maxillary sinuses, and cranial vault in 39 intentionally deformed skulls (Figure 7.2) and 19 non-deformed control skulls. The purpose of this study was to

assess how the intentional cranial deformations affected the shape and volume of the cranial cavities and the thickness of the cranial vault. The method was able to segment the cavities in all cases, even though some of the cavities were broken and/or had large boundary gaps.

One limitation of the method is that the planar barrier does not follow the curvature of the anterior boundary, which can lead to over- or underestimation of the orbit volume. Simultaneous leaks at different regions of the orbit can also be difficult to correct effectively since the mesh continues to deform globally during the editing and might reintroduce leaks in regions that were previously corrected.

7.1.2 An interactive orbit segmentation method based on 3D painting

To overcome the limitations of the deformable model-based orbit segmentation method presented in the previous section, an alternative interactive segmentation tool was developed in Paper VIII that offers tighter control over the segmentation result and produces a more accurate and consistent delineation of the anterior opening.

The tool implements a 3D painting interface that allows the user to quickly segment or "paint" the fat and soft-tissue content of the orbit by sweeping a volumetric brush over the CT image, similar to [33]. The brush modifies and updates the segmentation result in real-time and takes distance and gradient information into account to fill out and find the exact boundaries of the orbit. A smooth and consistent delineation of the anterior boundary is obtained by fitting a thin-plate spline [7] to user-selected landmarks. Figure 7.3 shows an overview of the segmentation process.

Results

The tool was evaluated on ten CT images of intact and fractured orbits (20 orbits in total). Two test users segmented the orbits twice and the resulting segmentations were compared with manually corrected reference segmentations. The tool was found to have high intra- and inter-operator precision (95% spatial overlap and $\leq$1 ml volume difference) and produced segmentation results that were similar to the manually corrected reference segmentations, but required only a few minutes of interaction time. The interactive segmentations included a larger part of the anterior soft-tissue content than the manually corrected reference segmentations, but defined the anterior boundary in a way that more closely resembles the boundary of orbit models generated by casts [37].

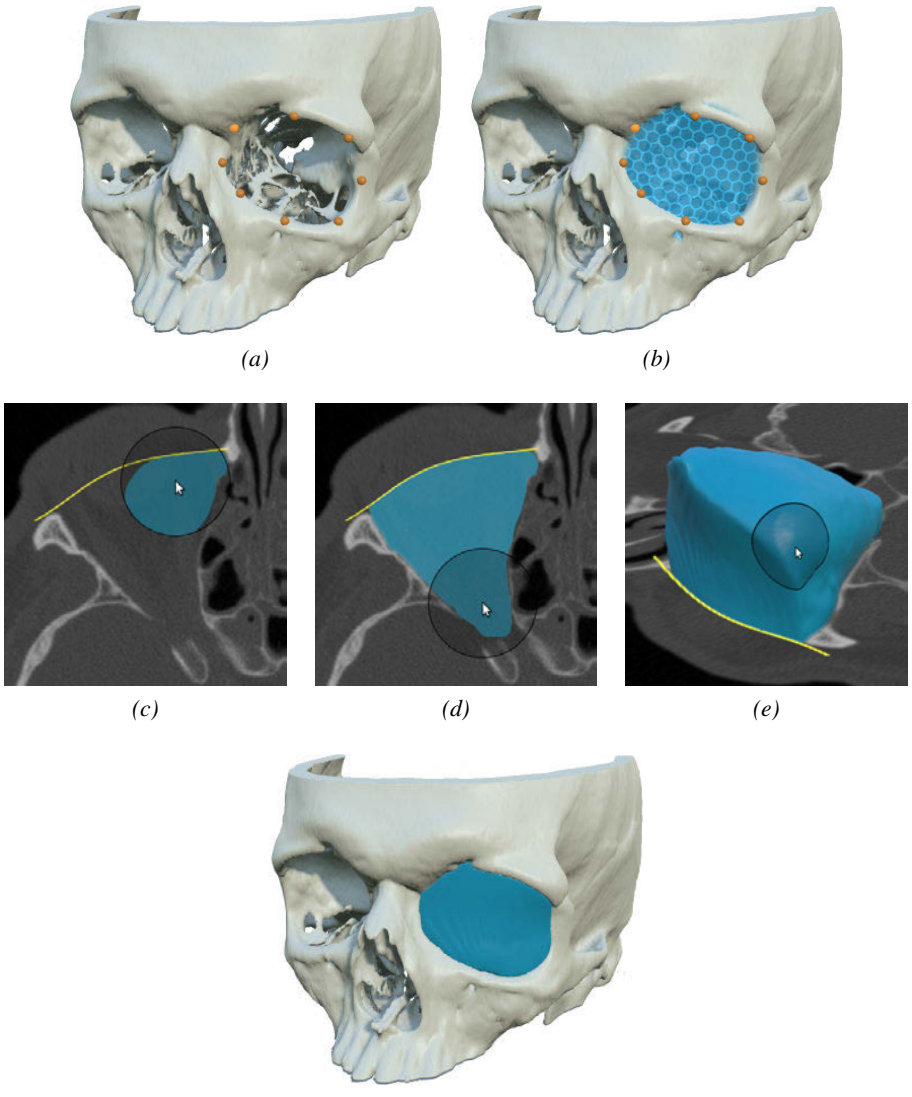


Figure 7.3. Interactive 3D painting tool for orbit segmentation: (a) landmark-based marking of the anterior opening; (b) smooth boundary obtained by fitting a thin-plate spline to the landmarks; (c)–(e) interactive 3D painting; (f) segmented orbit.

7.2 Quantitative analysis of wrist fractures in CT images

The conventional method for evaluating fractures on the *radius* bone in the wrist is to measure the *fracture angulation* in plain 2D X-ray images (Figure 7.4). By performing the measurements on consecutive repeat scans of the same patient, the orthopedic surgeon can determine how the angulation has changed over time. The precision and accuracy of this measurement method

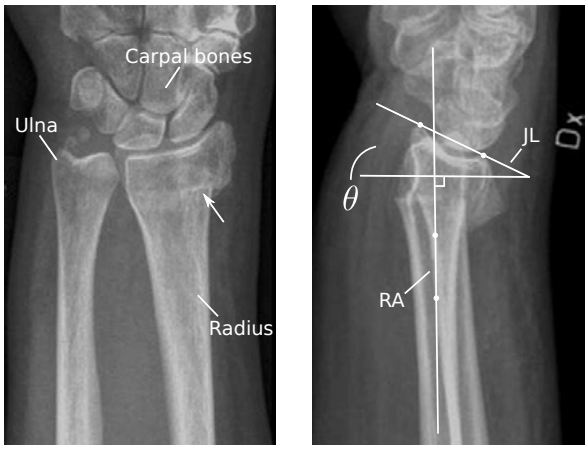


Figure 7.4. Left: frontal X-ray image of a fractured radius bone. The arrow marks the fracture location. Right: the fracture angulation θ, measured in 2D on a lateral X-ray image of the same wrist. θ is defined as the angle between the joint line *JL* and a line that is orthogonal to the long-axis *RA* of the radius.

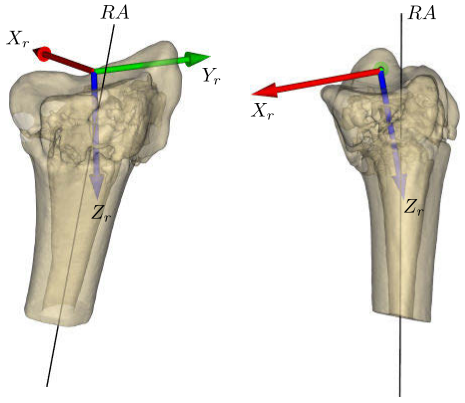


Figure 7.5. Wrist fracture angulation measured in a 3D CT image. *RA* is the long-axis of the radius shaft and (X_r, Y_r, Z_r) is a local coordinate system representing the orientation of the joint surface. The fracture angulation θ is obtained from *RA* and X_r.

is, however, limited since X-ray only provides a static 2D projection of the bones and the angulation is three-dimensional. The lack of depth in the image makes it difficult to identify reliable landmarks for the measurements, and the X-ray acquisition angle can often vary substantially between images. An intra- and inter-operator variability of 3–4° has been reported [24] for the method, which is too high for, for example, orthopedic research studies where more precise angle measurements are required to compare different methods of treatment. Papers II and IV present a more precise method for measuring the angulation in 3D CT images.

Method

The proposed 3D measurement technique combines imprecise user input with precise semi-automatic image analysis algorithms to identify two components: the long axis *RA* of the radius shaft; and a local coordinate system (X_r, Y_r, Z_r) representing the orientation of the joint surface. The components are shown in Figure 7.5. Figure 7.6 illustrates the measurement pipeline.

Given a sequence of N consecutive CT images of the same patient acquired at different occasions, the radius bone in each image is first extracted with an interactive mesh segmentation technique. To determine the orientation of the joint surface in each image, the joint surface in the first image is selected as template and assigned a local coordinate system based on three user-defined landmarks. By registering the resulting template to the segmented radius bones in the remaining follow-up scans, the joint-surface orientation in each scan is obtained. The registration is performed semi-automatically with an ICP-based algorithm that is initialized with user-defined landmarks.

To determine the long axis *RA*, the user selects an approximately 2 cm long section of the radius shaft that is located beneath the fracture and within 5 cm from the joint surface. The surface normals of the selected shaft section are computed and mapped to points on a unit sphere. The normals of the inner and outer surface of the shaft, which are orthogonal or near-orthogonal to the long axis, will form a dense ring on the sphere, and the the long axis can be obtained by fitting a plane to the ring and selecting the normal of the resulting plane as long axis. The plane fitting is performed with a robust RANSAC-based method.

Since the selected part of the bone is slightly flattened and conical, standard RANSAC tends to get stuck in and wobble between local minima in each run and produce axes that are not equally well-aligned to all sides of the bone. The method described in IV address this issue by first running RANSAC multiple (100) times to generate a set of candidate axes and then selecting the average of these axes as the true axis. Paper V presents an improved general RANSAC algorithm, called Optimal RANSAC, that has been tested on the wrist images and shown to find the optimal axis in a single run.

Results

Paper VII presents a study where the 3D measurement method was tested on 33 CT scan sequences of fractured wrists (198 CT images in total) and compared with conventional 2D X-ray measurements via Bland-Altman plots [5]. Two users, U1 and U2, performed repeat angle measurements to enable assessment of intra- and inter-operator precision. The 3D measurement method was found to have substantially higher intra- and inter-operator precision (limits of agreement ± 1.4 to 2.5°) than the conventional 2D measurement method (limits of agreement ± 4.0 to 4.6°).

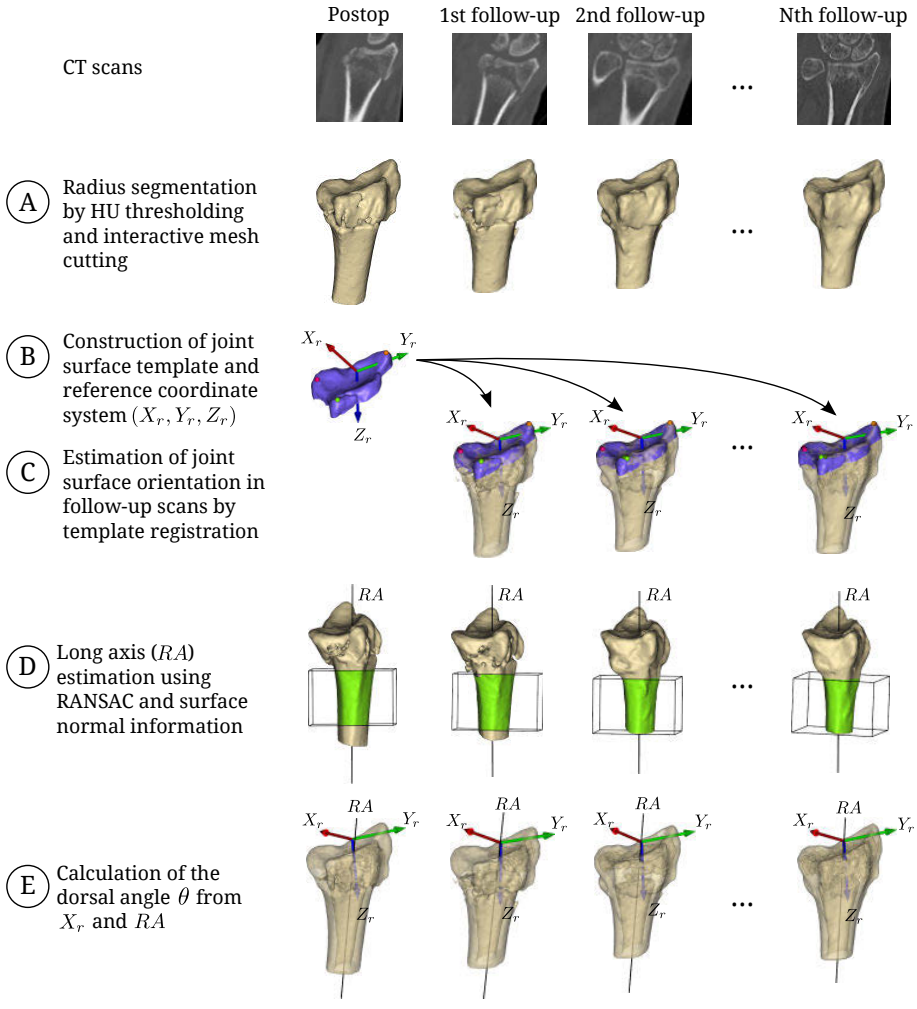
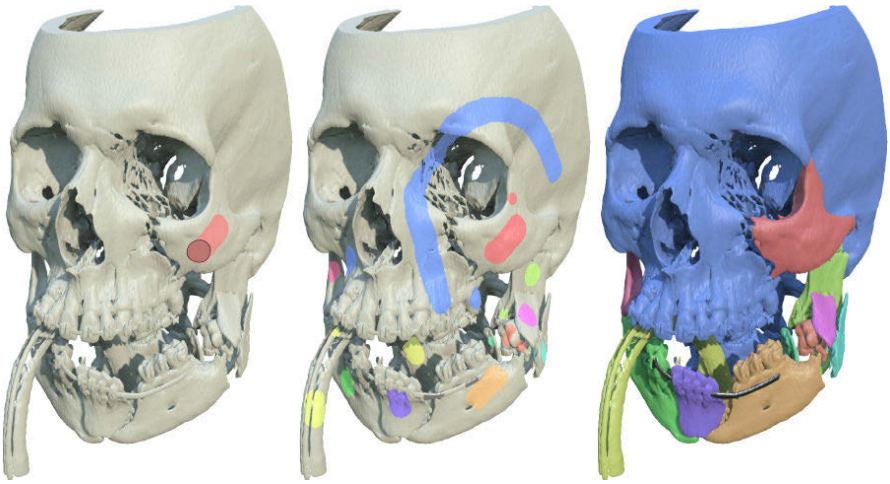


Figure 7.6. Overview of the semi-automatic 3D angle measurement pipeline.

7.3 Interactive segmentation of bones and bone fragments for virtual surgery planning

Restoring the skeletal anatomy after trauma or disease is a complex task that can be facilitated by careful pre-operative planning based on 3D CT images. It is possible to assembly fractured bones virtually [36] or print them as plastic models on a 3D printer to guide the reconstruction, but this requires that the individual bone structures first have been segmented from the CT image. Currently, this type of segmentation is often performed by manually marking the bones in 2D slice views, which is both tedious and time-consuming.

Paper VI presents BoneSplit, an efficient and intuitive interactive 3D painting tool for segmenting individual bones and bone fragments in CT images.



(a) *(b)* *(c)*

Figure 7.7. Interactive separation of bones and bone fragments in a complex fracture case: (a) 3D brush used for painting markers (seeds); (b) marked bones; (c) segmentation result obtained with random walks.

Method

An initial collective bone segmentation is generated by thresholding the CT image at a user-specified HU value (preset to 300 HU). Separation of individual bone structures is then achieved with a GPU-accelerated random walks algorithm that takes a graph constructed from the thresholded bones and user-defined markers (seeds) as input. Since individual bone structures can be difficult to identify and mark in conventional 2D slice views, a 3D painting interface is implemented that allows the user to paint seeds directly on 3D surface renderings of the bones as well as on individual CT slices. The segmentation process is iterative and typically requires that the user paints additional seeds on the bones to guide and correct the random walks algorithm. Figure 7.7 illustrates the segmentation process. The bones are rendered with a deferred isosurface ray-casting technique, which combines local illumination with shadow mapping and ambient occlusion to provide shadow depth cues that makes it easier for the user to perceive depth and spatial relationships between bones and bone fragments. Volume clipping allows the user to hide or expose selected bone structures to visually inspect the segmentation result and, in particular, the contact surfaces between the segmented bones.

Small errors can be corrected with a local 3D editing tool, and a multi-label connected component analysis filter can optionally be applied to remove small structures that should not be included in the final segmentation. The segmented bones are then ready to be exported to VTK image or STL mesh files for further analysis or planning.

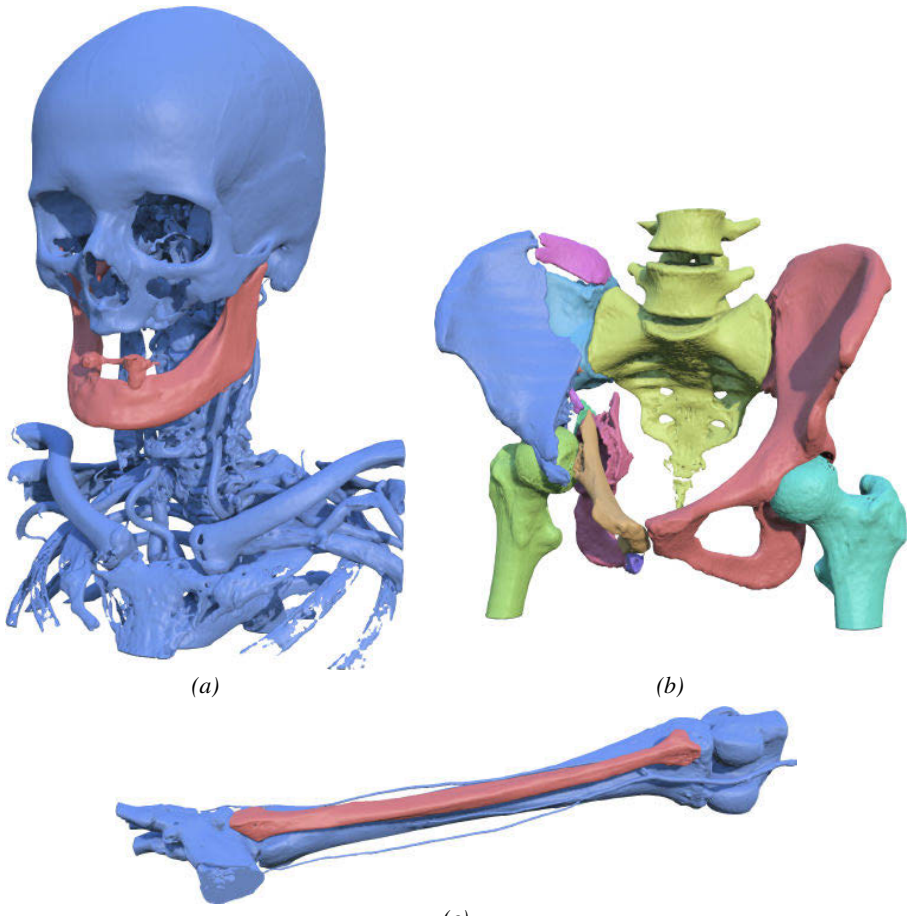


Figure 7.8. Examples of segmentation results produced with BoneSplit: (a) mandible segmentation; (b) pelvis fracture segmentation; (c) fibula segmentation. The test datasets were obtained from the OsiriX DICOM repository.

Results

Two test users performed interactive segmentations of the facial skeleton in CT scans of two complex trauma cases and one tumor case. The interactive segmentations took on average 14 minutes to perform. For comparison, a corresponding manual segmentation of one of the trauma cases with 15 bone fragments took more than five hours to complete. Visual inspection and assessment of the Dice similarity coefficient showed that the interactive segmentations were both accurate and repeatable. The segmentation tool has also been tested on CT images of the pelvis, wrist, and lower limbs, with satisfying results, and been used in a joint project on virtual surgery planning to segment bones in several head trauma and tumor cases. See Figure 7.8 for examples.

8. Summary and future work

This final chapter concludes the work performed and offers insights and suggestions for future directions.

8.1 Summary of contributions

The work performed in this thesis has resulted in a number of interactive methods and software tools for segmenting, visualizing, and analysing 3D CT images of the skeleton. To summarize, the main contributions are:

- A semi-automatic deformable model-based method for segmenting the orbit (eye-socket) in CT images that lets the user guide and correct the result via haptic 3D interaction. The method can produce accurate and precise segmentations of both intact and fractured orbits and requires only a few minutes of interaction time. Moreover, it is demonstrated that the method is not limited to orbit segmentation but can also be applied for segmenting other bone cavities such as the sinuses and cranial vault.
- An interactive 3D painting-based method for orbit segmentation that offers tight control over the segmentation result and produces a smooth and consistent delineation of the anterior (frontal) boundary of the orbit. The segmentation is performed with a volumetric brush that takes distance and gradient information into account to fill out and find the exact boundaries of the orbit. The method has been tested on intact as well as fractured orbits and found to be fast and precise.
- A semi-automatic method for measuring the 3D angulation of wrist fractures in CT images. The method combines graph-based mesh segmentation with surface registration and RANSAC to determine the angulation between the shaft and joint surface of the radius bone with high precision. It is demonstrated that the method produces more precise results than the conventional 2D X-ray measurement method that is currently used in clinical practice.
- An intuitive and interactive tool for segmenting individual bones and bone fragments in CT images. The tool, called BoneSplit, combines GPU-accelerated random walks segmentation with direct volume rendering and 3D texture painting to make it easy for the user to mark and segment bone structures of interest. It enables the user to produce an accurate segmentation result within a few minutes, thereby removing a major bottleneck in virtual surgery planning.

8.2 Future work

One future plan is to investigate if the orbit or bone segmentation methods can be extended to work on CBCT images. CBCT scanners offer higher image resolution at lower radiation dose than conventional CT scanners and are increasingly used at clinics, but the resulting images are more challenging to visualize and segment since the intensity values do not represent actual HU values and the images often are corrupted by artifacts. The presented methods rely on intensity thresholding for extracting bone structures from the CT image, which is not a reliable technique for CBCT. Edge-detection or, for example, fuzzy connectedness [50] segmentation techniques could potentially be used as a replacement for thresholding to overcome this problem.

Further research is also required to develop reliable methods for comparing and analysing the shape of segmented orbits. Previous research have mainly focused on segmentation and volume measurements.

Another interesting direction would be to combine the deformable model-based orbit segmentation method in Paper I with the 3D painting tools in Paper VIII. The deformable model, which is more robust to large boundary gaps, could be used for quickly producing an initial segmentation of the orbit, while the 3D painting tools are used for editing and refinement.

Several of the presented segmentation and measurement methods have already been applied in joint research projects and medical research studies, with positive results and feedback from surgeons: however, the methods need to be extensively tested and evaluated on larger CT datasets before they, eventually, can be applied in clinical settings.

Future plans for the interactive bone segmentation tool are to make a standalone version of the software available for academic use, and also to integrate it with a virtual surgery planning system [36] developed in a joint research project. The bone segmentation tool should also be extended with interactive hole-filling methods [25] to enable generation of solid bone models that can be used for, for example, finite element modelling.

Finally, one interesting future direction would be to extend the interactive 3D painting segmentation tools to a true 3D user interface that provides stereographics, head-tracking, and tracked 3D input devices. This would facilitate some of the complex 3D navigation and selection tasks involved in the segmentation. Key challenges would be to maintain a high enough frame rate (60 FPS for stereoscopic head-tracked planar displays and 90 FPS for stereoscopic head-mounted displays) for the rendering, and to compute and update the segmentation result fast enough to not break the immersion for the user.

9. Summary in Swedish

Dagens medicinska bildtagningsutrustningar kan generera mycket detaljerade tredimensionella (3D) bilder av skelettet som tydligt visar frakturer eller andra skador. Datoriserade bildbehandlings- och bildanalysmetoder, i kombination med volymvisualiseringstekniker, kan i hög grad underlätta tolkningen av sådana bilder och används allt mer inom kirurgin för att virtuellt planera hur skelettet ska återställas efter trauma eller sjukdom. Två svårigheter i den virtuella planeringen är dock att separera (segmentera) benstrukturer från resten av bilden samt att interagera med 3D-datat på ett effektivt sätt.

Segmentering är nödvändigt för att kunna skapa 3D-modeller av benen som kan användas för vidare analys eller planering. Eftersom benstrukturer kan ha väldigt varierande form och storlek, och kontrasten mellan ben och mjukvävnad ibland kan vara låg, är det svårt för en dator att utföra segmenteringen automatiskt. I dagsläget utförs segmentering därför ofta manuellt av medicinska expertanvändare, vilket kan ge noggranna resultat men är väldigt tidskrävande och tröttsamt. En kompromiss som visat sig fungera väl i praktiken är halvautomatiska eller interaktiva segmenteringsmetoder, där användaren vägleder och korrigerar algoritmer som utför merparten av segmenteringsarbetet.

Den här avhandlingen presenterar effektiva och precisa interaktiva metoder för att segmentera, visualisera och analysera tredimensionella datortomografibilder (eng. computed tomography, CT) av skelettet. Metoderna har testats på kliniska CT-bilder och är framförallt avsedda att användas som stöd och hjälpmedel för kraniomaxillofacial (eng. cranio-maxillofacial, CMF) kirurgiplanering och ortopediska tillämpningar.

Två interaktiva metoder för att segmentera och mäta volymen på ögonhålor i CT-bilder presenteras. Den första metoden implementerar en deformerbar model som passas in mot ögonhålan via haptisk 3D-interaktion. Användaren får haptisk återkoppling (kraftåterkoppling) via en robotarm med sex frihetsgrader och kan därigenom känna på bilddatat och manipulera den deformerbara modellen i 3D på ett mycket naturligt sätt. Den andra metoden implementerar en användarstyrd volumetrisk pensel som utnyttjar avstånds- och gradientinformation från bilddatat för att fylla ut och finna exakta gränser hos ögonhålan. Metoderna har testats på CT-bilder av såväl intakta som skadade ögonhålor och visat sig kunna producera segmenteringar som är jämförbara, i avseende på noggrannhet, med manuella referenssegmenteringar, men endast kräver några få minuters interaktionstid.

Avhandlingen presenterar även en halvautomatisk metod för att mäta vinkelförändringar hos handledsfrakturer i CT-bilder. Vinkelförändringar mäts normalt sett i tvådimensionella (2D) slätröntgenbilder, vilket dock ger upphov till

stora mätfel och variationer eftersom sådana bilder bara visar en 2D-projektion av den tredimensionella frakturvinkeln och det är svårt för en (mänsklig) användare att mäta vinkeln lika från gång till gång. I den CT-baserade mätmetoden extraheras det frakturerade benet först från bilden med hjälp av en halvautomatisk polygonytebaserad segmenteringsteknik, där användaren indikerar vilka ben som ska separeras och extraheras genom att rita ut olikfärgade markörer (så kallade frön) på benytorna. Vinkeln mellan ledytan och skaftet på benet bestäms därefter med en teknik baserad på ytregistrering och en metod som heter random sample consensus (RANSAC). Avhandlingen demonstrerar att den CT-baserade metoden kan mäta vinkelförändringar med betydligt högre precision än den konventionella 2D-metoden.

Slutligen presenterar avhandlingen BoneSplit, ett interaktivt verktyg för att segmentera individuella ben och benfragment i CT-bilder. Den här typen av segmentering är väsentlig för virtuell kirurgiplanering av komplicerade trauma- eller tumörfall, men tar flera timmar att utföra med konventionella manuella segmenteringsmetoder. Det presenterade verktyget kombinerar en slumpvandrarbaserad (eng. random walks) segmenteringsmetod med direktvolymrendrering och en interaktiv 3D-pensel för att skapa ett intuitivt och effektivt segmenteringsgränssnitt där användaren snabbt och enkelt kan markera och separera benstrukturer. Exempel på segmenteringar som har utförts med verktyget visas på omslaget. Segmenteringen utnyttjar beräkningskraften hos grafikprocessorer (eng. graphics processing units, GPUs) för att snabbt beräkna och även uppdatera resultatet när användaren lägger till markeringar med penseln. Ett noggrant segmenteringsresultat kan erhållas efter bara några få minuters interaktionstid, vilket tar bort en stor flaskhals i planeringen.

Acknowledgements

The work presented in this thesis was carried out at the Centre for Image Analysis between 2011 and 2016. I wish to thank all my past and current colleagues at CBA, and my collaborators from the medical (and other) fields, for making my PhD studies both enjoyable and challenging. Special thanks to:

- My main supervisor, Ingela Nyström, for always being so supportive, and for giving me the opportunity to do my master thesis project and, later on, my PhD, at CBA. Thanks for giving me freedom to pursue different research ideas and projects (and work somewhat irregular hours).
- My assistant supervisor, Filip Malmberg, for invaluable scientific advice, guidance, and support during the years.
- My assistant supervisor, Ida-Maria Sintorn, for always being so encouraging and helpful and offering feedback on short notice.
- Roman Khonsari, for great collaboration through the years and for supplying me with such interesting datasets to work on.
- My collaborators and co-authors from the hospital: Albert Christersson, Johanna Nilsson, Andreas Thor, and Jan-Michaél Hirsch - it has been a pleasure to work with you!
- Anders Hast, for being an enthusiastic collaborator and partner in crime on teaching the computer graphics and visualization courses.
- My other research collaborators: Jonathan Britto, Benjamin Way, Tomasz Smektala, Alice Clement, and more.
- My brother and co-worker Fredrik Nysjö, for collaboration on courses and projects over the years, fishing trips, and interesting explorations in programming, computer graphics, and VR. Good luck with your PhD!
- Lennart Svensson, for good collaboration on courses and the ProViz project.
- My previous office mates, Pontus Olsson, Martin Simonsson, and Azadeh Fakhrzadeh, for being great colleagues. Special thanks to Pontus for collaboration on teaching the graphics course, to Martin for keeping me awake with coffee, and to Azadeh for the dinners.
- Ingrid Carlbom, for letting me take part of the HASP project.
- Stefan Seipel, for good collaboration on teaching the computer graphics and visualization courses.
- Ewert Bengtsson, for helpful discussions during the macro meetings.
- Lena Nordström, for patiently taking care of the administrative work.
- Astrid Raidl, for keeping my Linux workstation updated and running.

- Fei Liu, for excellent collaboration on courses, interesting conversations about computer graphics, and the (unfortunately too few) Friday night gaming sessions.
- My parents, Berit and Erik, for all your support through the years, and for keeping up with my absence and absent-mindness during the production of this thesis.

References

[1] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., 2008.

[2] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. Mesh: Measuring errors between surfaces using the Hausdorff distance. In *Proceedings of the IEEE International Conference in Multimedia and Expo (ICME) 2002*, volume 1, pages 705–708. IEEE, 2002.

[3] N. Bell and M. Garland. Implementing Sparse Matrix-Vector Multiplication on Throughput-Oriented Processors. In *Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pages 1–11. ACM, 2009.

[4] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[5] J. M. Bland and D. Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *The Lancet*, 327(8476):307–310, 1986.

[6] J. F. Blinn. Models of Light Reflection for Computer Synthesized Pictures. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 192–198. ACM, 1977.

[7] F. L. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):567–585, 1989.

[8] G. Borgefors. Distance Transformations in Arbitrary Dimensions. *Computer Vision, Graphics, and Image Processing*, 27(3):321–345, 1984.

[9] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon Mesh Processing*. CRC press, 2010.

[10] Bowman, Doug and Kruijff, Ernst and LaViola Jr, Joseph J and Poupyrev, Ivan. *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2004.

[11] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[12] V. Chalana and Y. Kim. A Methodology for Evaluation of Boundary Detection Algorithms on Medical Images. *IEEE Transactions on Medical Imaging*, 16(5):642–652, 1997.

[13] L. D. Cohen and I. Cohen. Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images. *IEEE Transactions on Pattern Analysis and machine intelligence*, 15(11):1131–1147, 1993.

[14] A. Criminisi, T. Sharp, and A. Blake. GeoS: Geodesic Image Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 5302 of *LNCS*, pages 99–112. Springer, 2008.

[15] H. Delingette. General Object Reconstruction Based on Simplex Meshes. *International Journal of Computer Vision*, 32(2):111–146, 1999.

[16] A. X. Falcão and J. K. Udupa. A 3D generalization of user-steered live-wire segmentation. *Medical Image Analysis*, 4(4):389–402, 2000.

[17] P. F. Felzenszwalb and D. P. Huttenlocher. Distance Transforms of Sampled Functions. *Theory of computing*, 8(1):415–428, 2012.
[18] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.
[19] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 2008.
[20] C. Goodall. Procrustes Methods in the Statistical Analysis of Shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 285–339, 1991.
[21] L. Grady. Random Walks for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.
[22] M. Hadwiger, P. Ljung, C. R. Salama, and T. Ropinski. Advanced Illumination Techniques for GPU Volume Raycasting. In *ACM SIGGRAPH ASIA 2008 Courses*, pages 1–166. ACM, 2008.
[23] J. Jansen, R. Schreurs, L. Dubois, T. J. Maal, P. J. Gooris, and A. G. Becking. Orbital volume analysis: validation of a semi-automatic software segmentation method. *International Journal of Computer Assisted Radiology and Surgery*, 11(1):11–18, 2016.
[24] P. Johnson and R. M. Szabo. Angle measurements of the distal radius: a cadaver study. *Skeletal Radiology*, 22(4):243–246, 1993.
[25] Y. Kang, K. Engelke, and W. A. Kalender. Interactive 3D editing tools for image segmentation. *Medical Image Analysis*, 8(1):35–46, 2004.
[26] J. Krüger and R. Westermann. Acceleration Techniques for GPU-based Volume Rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, pages 287–292, 2003.
[27] Y.-K. Lai, S.-M. Hu, R. R. Martin, and P. L. Rosin. Rapid and Effective Segmentation of 3D Models Using Random Walks. *Computer Aided Geometric Design*, 26(6):665–679, 2009.
[28] H. Landis. Production-ready global illumination. *ACM SIGGRAPH Course Notes*, 16(2002):11, 2002.
[29] J.-S. Lee. Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24(2):255–269, 1983.
[30] B. Li and S. T. Acton. Active Contour External Force Using Vector Field Convolution for Image Segmentation. *IEEE Transactions on Image Processing*, 16(8):2096–2106, 2007.
[31] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.
[32] K. Lundin, B. Gudmundsson, and A. Ynnerman. General proxy-based haptics for volume visualization. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, pages 557–560. IEEE, 2005.
[33] F. Malmberg, R. Nordenskjöld, R. Strand, and J. Kullberg. SmartPaint: a Tool for Interactive Segmentation of Medical Volume Images. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, (doi:10.1080/21681163.2014.960535):1–9, 2014.
[34] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*,

19(14):1023–1040, 2001.

[35] K. Moreland. Diverging Color Maps for Scientific Visualization. In *Advances in Visual Computing*, volume 5876 of *LNCS*, pages 92–103. Springer Berlin Heidelberg, 2009.

[36] P. Olsson, F. Nysjö, J.-M. Hirsch, and I. B. Carlbom. A haptics-assisted cranio-maxillofacial surgery planning system for restoring skeletal anatomy in complex trauma cases. *International Journal of Computer Assisted Radiology and Surgery*, 8(6):887–894, 2013.

[37] T. H. Osaki, D. K. de Castro, C. Yabumoto, V. Mingkwansook, E. Ting, N. Nallasamy, H. Curtin, and A. Fay. Comparison of Methodologies in Volumetric Orbitometry. *Ophthalmic Plastic and Reconstructive Surgery*, 29(6):431–436, 2013.

[38] P. Perona and J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[39] B. Preim and C. P. Botha. *Visual Computing for Medicine: Theory, Algorithms, and Applications*. Newnes, 2013.

[40] W. T. Reeves, D. H. Salesin, and R. L. Cook. Rendering Antialiased Shadows with Depth Maps. *ACM SIGGRAPH Computer Graphics*, 21(4):283–291, 1987.

[41] S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. IEEE, 2001.

[42] P. Scolozzi and B. Jaques. Computer-Aided Volume Measurement of Posttraumatic Orbits Reconstructed with AO Titanium Mesh Plates: Accuracy and Reliability. *Ophthalmic Plastic and Reconstructive Surgery*, 24(5):383–389, 2008.

[43] J. R. Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Carnegie-Mellon University. Department of Computer Science, 1994.

[44] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson, third edition, 2008.

[45] E. B. Strong, S. C. Fuller, and H. S. Chahal. Computer-Aided Analysis of Orbital Volume: A Novel Technique. *Ophthalmic Plastic and Reconstructive Surgery*, 29(1):1–5, 2013.

[46] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically Deformable Models. In *ACM Siggraph Computer Graphics*, volume 21, pages 205–214. ACM, 1987.

[47] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *Sixth International Conference on Computer Vision*, pages 839–846. IEEE, 1998.

[48] J. K. Udupa and G. T. Herman. *3D Imaging in Medicine*. CRC press, 1999.

[49] J. K. Udupa, V. R. LeBlanc, Y. Zhuge, C. Imielinska, H. Schmidt, L. M. Currie, B. E. Hirsch, and J. Woodburn. A framework for evaluating image segmentation algorithms. *Computerized Medical Imaging and Graphics*, 30(2):75 – 87, 2006.

[50] J. K. Udupa and S. Samarasekera. Fuzzy Connectedness and Object Definition: Theory, Algorithms, and Applications in Image Segmentation. *Graphical Models and Image Processing*, 58(3):246–261, 1996.

[51] E. Vidholm. *Visualization and Haptics for Interactive Medical Image Analysis*. PhD thesis, Uppsala University, 2008.

[52] M. E. Wagner, N.-C. Gellrich, K.-I. Friese, M. Becker, F.-E. Wolter, J. T. Lichtenstein, M. Stoetzer, M. Rana, and H. Essig. Model-based segmentation in orbital volume measurement with cone beam computed tomography and evaluation against current concepts. *International Journal of Computer Assisted Radiology and Surgery*, 11(1):1–9, 2016.

[53] C. Xu, D. L. Pham, and J. L. Prince. Image Segmentation Using Deformable Models. *Handbook of Medical Imaging*, 2:129–174, 2000.

[54] K. H. Zou, S. K. Warfield, A. Bharatha, C. M. Tempany, M. R. Kaus, S. J. Haker, W. M. Wells, F. A. Jolesz, and R. Kikinis. Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index 1: Scientific Reports. *Academic Radiology*, 11(2):178–189, 2004.

Acta Universitatis Upsaliensis

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 1411

Editor: The Dean of the Faculty of Science and Technology

A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology. (Prior to January, 2005, the series was published under the title "Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology".)

Distribution: publications.uu.se
urn:nbn:se:uu:diva-301180

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2016