UPPSALA
UNIVERSITET

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 1898

Modeling and Visualization for Virtual Interaction with Medical Image Data

FREDRIK NYSJÖ

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2020

ISSN 1651-6214
ISBN 978-91-513-0864-7
urn:nbn:se:uu:diva-403104

Dissertation presented at Uppsala University to be publicly examined in ITC 2446, Lägerhyddsvägen 2, Hus 2, Polacksbacken, Uppsala, Friday, 13 March 2020 at 10:15 for the degree of Doctor of Philosophy. The examination will be conducted in English. Faculty examiner: Professor Alexandru C. Telea (Dept. of Information and Computing Sciences, Utrecht University, The Netherlands).

Abstract
Nysjö, F. 2020. Modeling and Visualization for Virtual Interaction with Medical Image Data. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 1898. 50 pp. Uppsala: Acta Universitatis Upsaliensis. ISBN 978-91-513-0864-7.

Interactive systems for exploring and analysing medical three dimensional (3D) volume image data using techniques such as stereoscopic rendering and haptics can lead to new workflows for virtual surgery planning. This includes the design of patient-specific surgical guides and plates for additive manufacturing (3D printing). Our applications, medical visualization and cranio-maxillofacial surgery planning, involve large volume data such as computed tomo\-graphy (CT) images with millions of data points. This motivates the development of fast and efficient methods for visualization and haptic rendering, as well as the development of efficient modeling techniques for simplifying the design of 3D printable parts. In this thesis, we develop methods for visualization and haptic rendering of isosurfaces in volume image data, and show applications of these methods to medical visualization and virtual surgery planning. We further develop methods for modeling surgical guides and plates for cranio-maxillofacial surgery, and integrate them into our system for haptics-assisted surgery planning called HASP. This system is now installed at the department of surgical sciences, Uppsala University, and is being evaluated for use in clinical research.

Keywords: medical image processing, volume rendering, haptic rendering, medical visualization, virtual surgery planning

Fredrik Nysjö, Department of Information Technology, Division of Visual Information and Interaction, Box 337, Uppsala University, SE-751 05 Uppsala, Sweden.

ISSN 1651-6214
ISBN 978-91-513-0864-7
urn:nbn:se:uu:diva-403104 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-403104)

List of papers

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

I Olsson, P., **Nysjö, F.**, Hirsch, J.M., and Carlbom, I.B., "Snap-to-Fit, a Haptic 6 DOF Alignment Tool for Virtual Assembly," Proc. IEEE World Haptics Conference (WHC), Daejeon, South Korea (2013).

II Olsson, P., **Nysjö, F.**, Hirsch, J.M., and Carlbom, I.B., "A Haptics-Assisted Cranio-Maxillofacial Surgery Planning System for Restoring Skeletal Anatomy in Complex Trauma Cases," Int. Journal of Computer Assisted Radiology and Surgery (IJCARS) (2013).

III **Nysjö, F.**, Olsson, P., Malmberg, F., Carlbom, I.B., and Nyström, I., "Using Anti-Aliased Signed Distance Fields for Generating Surgical Guides and Plates from CT Images", Journal of WSCG, Vol.25, No.1–2 (2017).

IV Nilsson, J., **Nysjö, F.**, Kämpe, J., Nyström, I., Thor, A., "Evaluation of In-House, Haptic Assisted Surgical Planning for Virtual Reduction of Complex Mandibular Fractures", submitted to journal (2020).

V **Nysjö, F.**, Malmberg, F., Nyström, I., "RayCaching: Amortized Isosurface Rendering for Virtual Reality", Computer Graphics Forum, published online in June 2019 (2019).

VI **Nysjö, F.**, "Clustered Grid Cell Data Structure for Isosurface Rendering", submitted to peer-reviewed conference (2020).

VII **Nysjö, F.**, Malmberg, F., Nyström, I., "Vectorised High-Fidelity Haptic Rendering with Dynamic Pointshell", submitted to peer-reviewed conference (2020).

The author has made substantial contributions to all the papers listed above. In Papers I–II, the author developed much of the software and the image processing pipeline, and contributed to the writing. In Papers III, V, and VII, the author was the main contributor to the method development, implementation, experiments, and writing; Paper III was also presented by the author at a conference. In Paper IV, Nysjö and Nilsson contributed equally to the paper, with Nysjö performing method development, data analysis, and contributing to the writing. In Paper VI, the author was the main contributor and single author.

Related work

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

1. Olsson, P., **Nysjö, F.**, Seipel, S., and Carlbom, I.B., "Physically Co-Located Haptic Interaction with 3D Displays," Haptics Symposium, Vancouver, Canada (2012).

2. Olsson, P., **Nysjö, F.**, Aneer, B., Seipel, S., and Carlbom, I.B., "Spline-Grip - An Eight Degrees-of-Freedom Flexible Haptic Sculpting Tool," ACM SIGGRAPH 2013 (abstract), Anaheim, USA (2013).

3. **Nysjö, F.**, Olsson, P., Hirsch, J.M., and Carlbom, I.B., "Custom Mandibular Implant Design with Deformable Models and Haptics", Proc. Computer Assisted Radiology and Surgery (CARS) (abstract), Fukuoka, Japan (2014).

4. Olsson, P., **Nysjö, F.**, Singh, N., Thor, A., and Carlbom, I.B., "Visuohaptic Bone Saw Simulator: Combining Vibrotactile and Kinesthetic Feedback", ACM SIGGRAPH Asia 2015, Kobe, Japan (2015).

5. Olsson, P., **Nysjö, F.**, Rodriguez-Lorenzo, A., Thor, A., Hirsch, J.M., and Carlbom, I.B., "Novel Virtual Planning of Bone, Soft-tissue, and Vessels in Fibula Osteocutaneous Free Flaps with the Haptics-Assisted Surgery Planning (HASP) System", Plastic and Reconstructive Surgery Global Open (PRSGO) (2015).

6. Blache, L., **Nysjö, F.**, Malmberg, F., Thor, A., Rodriguez-Lorenzo, A., Nyström, I., "SoftCut: A Virtual Planning Tool for Soft Tissue Resection on CT Images", Annual Conference on Medical Image Understanding and Analysis, Southampton, UK (2018).

Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
AO	Ambient Occlusion
CBCT	Cone-Beam Computed Tomography
CMF	Cranio-Maxillofacial
CSG	Constructive Solid Geometry
CT	Computed Tomography
DOF	Degrees-of-Freedom
DT	Distance Transform
GPU	Graphics Processing Unit
HASP	Haptics-Assisted Surgery Planning
HMD	Head-Mounted Display
HU	Hounsfield Units
LOD	Level-of-Detail
VR	Virtual Reality
VPS	Voxmap PointShell
VSP	Virtual Surgery Planning

Contents

1. Introduction

Interactive systems for exploring and analysing medical three-dimensional (3D) volume image data using techniques such as stereoscopic rendering and haptics can lead to new workflows for virtual surgery planning (VSP). This includes the design of patient-specific surgical guides and plates for additive manufacturing (3D printing). Applications such as cranio-maxillofacial (CMF) surgery planning often involve large volume data such as computed tomography (CT) images, motivating the development of fast and efficient methods for visualization and haptic rendering, as well as the development of efficient modeling tools for simplifying the design of 3D printable parts.

1.1 Objectives

The objectives for this thesis were to:

- Develop fast methods for visualization and haptic rendering of isosurfaces in volume data, including medical CT images.
- Combine image processing, visualization, and haptics into a VSP system for improving the current workflow of CMF surgery planning.
- Evaluate the accuracy, precision, and usability of our VSP system.

1.2 Thesis Outline

The thesis is structured as follows: Chapter 2 provides the relevant background on digital image processing, volume visualization, stereoscopic visualization and haptics, and medical applications, and also describes related work in those areas; Chapter 3 describes the work and contributions of Papers I–VII included in the thesis; finally, Chapter 4 summarises the main thesis contributions and proposes directions for future work.

2. Background and Related Work

In this chapter, we provide the relevant background for the work in this thesis.

2.1 Digital Image Processing

This first section introduces digital images as well as methods from digital image processing and image analysis [49] that we have used in Papers I–VII.

2.1.1 Digital Images

A digital grayscale image on a continuous domain can be represented by the function

$$f : \mathbb{R}^n \mapsto \mathbb{R} \tag{2.1}$$

defined by a set of sample points on a discrete grid and some function for reconstructing the values between the sample points. Two-dimensional (2D) and three-dimensional (3D) digital images are typically represented by regular Cartesian grids (Figure 2.1), with the sample points (pixels or voxels) shown as squares or cubes. In the rest of this thesis, unless otherwise noted, we will refer to such regular Cartesian grids when referring to images. We will also in some places refer to 3D images as volumes.

Reconstruction with nearest-neighbor interpolation results in the visible square pixels or cubic voxels of Figure 2.1, and is not sufficient to reconstruct a continuous image signal. Bilinear and trilinear interpolation compute a weighted average of the corners of 2^2- and 2^3-sized grid cells in the image, respectively, resulting in a reconstruction with C^0 continuity. Higher order interpolation functions such as cubic interpolation can further provide C^1 or higher continuity in the reconstruction, at a higher computational cost.

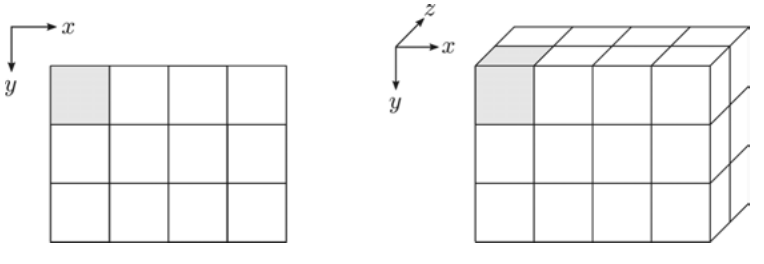
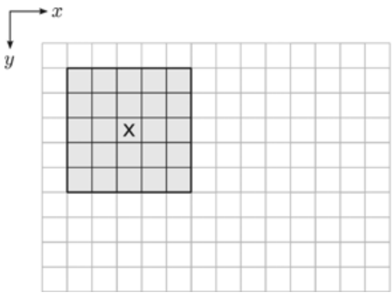


Figure 2.1. Digital images: (left) 2D image with regular grid of 4×3 sample points (pixels) shown as squares; (right) 3D image, or volume, with regular grid of $4 \times 3 \times 2$ sample points (voxels) shown as cubes.



$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Figure 2.2. Image convolution: (left) convolution with a 2D kernel over an image; (right) kernel weights (shown as integers) for a 5×5 Gaussian smoothing filter.

2.1.2 Image Filtering

Many common image filtering operations from image processing [49] can be described as convolution with a kernel over an image (Figure 2.2). An example is the Gaussian smoothing filter used to reduce high frequency image content. Other common uses of filtering include computing image gradients and enhancing features such as edges, and morphological operations.

Real non-synthetic image data such as CT often exhibit artifacts from the imaging device or the imaging process, such as noise, low contrast, or blur. Synthetic image data may similarly exhibit noise and aliasing artifacts from insufficient sampling during the image generation process. Reducing such artifacts by filtering is therefore often necessary in a data pipeline used for image analysis or visualization of the data.

2.1.3 Interactive Segmentation

A classical problem in image analysis is to extract and label objects in an image, a problem known as segmentation. Thresholding with a global threshold value can for example be used to automatically segment images into a background and a foreground label. There has been a transition in recent years from traditional hand-crafted segmentation methods [49] to learning-based methods such as Deep Learning [24] using convolutional neural networks. While such methods extend to 3D [6], lack of training data or ground truth can still be a problem in medical applications. Another potential problem with fully automatic segmentation methods is that the user cannot easily correct the result.

In manual segmentation, an expert user draws (usually in 2D) the regions to be segmented. This can be time consuming, especially for volumes when the segmentation is performed per slice in the image. Manual segmentation may still be a practical approach for smaller datasets and when more automated methods are not available. In Papers I–II, we manually segment fractured bones in a small number of CT images to show proof-of-concept of the haptics system in the papers. However, many applications require high throughput and high repeatability, for which manual segmentation is not an option.

Figure 2.3. Coverage representation: (left) original grayscale image (CT slice of fibula); (center) binary segmentation mask; (right) fuzzy coverage representation.

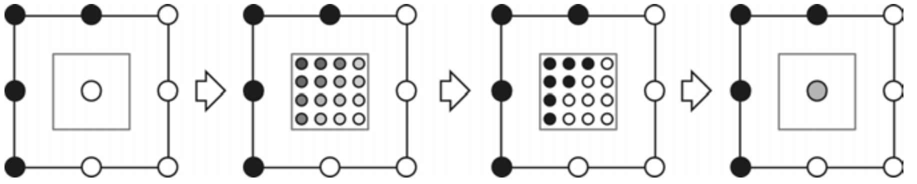


Figure 2.4. Binary-to-coverage conversion of voxel in a binary segmentation mask, using first trilinear supersampling in the original grayscale image data and then thresholding to compute a membership value α.

Semi-automatic segmentation allows the user to guide the segmentation algorithm and interactively correct the result. BoneSplit [36], for example, is a semi-automatic tool that provides a 3D texture-painting interface for bone segmentation in CT images. It enables quick marking and separation of individual bone fragments after an initial threshold value for bone has been specified, using a graph-based segmentation method to compute label masks from user-painted seeds. In Papers III–IV, we use BoneSplit to segment a larger number of CT images for the method evaluation and user study of the papers.

Many segmentation methods and tools only provide binary segmentation masks with limited contour information. To include sub-voxel information, a fuzzy coverage representation (Figures 2.3) with membership values in normalize range [0,1] can also be computed from the segmentation mask, using for example the method illustrated in Figure 2.4. This was used in Paper III.

2.1.4 Distance Transforms

A distance transform (DT) is a way of computing a distance map, or distance field, D_f from a binary image. The signed or unsigned scalar values in D_f represent internal or external distances to the object boundary, according to some distance function, as shown in Figure 2.5. The discrete weighted DT [3] with integer weights $\langle 3,4,5 \rangle$ is fast to compute in 3D and has a low absolute difference of 11.8% compared to the Euclidean distance. Efficient parallelisable algorithms for computing the Euclidean DT in 3D include vector propagation [10] and the more recent method by Felzenszwalb and Huttenlocher [13].

Figure 2.5. Distance maps: (left) binary input image (bunny); (center) distance map computed from internal weighted DT; (right) signed distance map computed from internal and external weighted DTs, and visualized by absolute value. Brighter regions in the distance maps indicate larger absolute distance, with black representing zero.

Anti-aliased DTs can produce more accurate distance values compared to binary DTs, in particular close to object boundaries, by utilising sub-pixel or sub-voxel information in the grayscale image data. Gustavson and Strand [15] extend the classical vector propagation algorithm in [10] to use coverage information when computing the DT. They also suggest a fast linear approximation d_f for mapping any coverage value α to a sub-pixel or sub-voxel distance,

$$d_f = 0.5 - \alpha. \tag{2.2}$$

In Papers I, II, and IV, we use the weighted $\langle 3,4,5 \rangle$ DT to compute distance maps for collision detection and haptic rendering. For the modeling in Paper III, we use Equation 2.2 with an anti-aliased version of the weighted DT to compute signed distance maps from coverage representations.

2.2 Volume Visualization

This section introduces volume visualization, with a focus on isosurfaces in scalar volume data. For an overview of the general visualization pipeline used in volume visualization and other data visualization, see Telea [51].

2.2.1 Isosurface Rendering

Isocontours in sampled data and implicit functions are encountered in applications in many areas, including scientific visualization, constructive solid geometry (CSG), computer animation, and games. The isocontour S of an image f for a particular scalar isovalue k can be defined as

$$S = \{x \in \mathbb{R}^n \mid f(x) = k\}. \tag{2.3}$$

In 2D images, isocontours become isolines, with examples shown in Figure 2.6. In 3D images, isocontours become isosurfaces, with Figure 2.7 showing isosurfaces representing different isovalues in the same volume data.

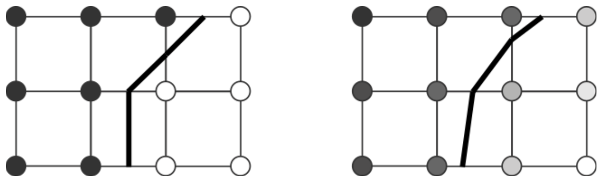


Figure 2.6. Isocontours: (left) isolines in binary 2D image; (right) corresponding isolines in grayscale 2D image, for isovalue $k = 0.5$.

Indirect volume rendering techniques extract an explicit isosurface representation and can provide efficient isosurface rendering for static volume data when the complete isosurface does not have to be recomputed in every frame. Direct volume rendering techniques (further described in Section 2.2.2) can also be used to render implicit isosurfaces directly from the volume data.

Polygonisation algorithms such as Marching Cubes [28] and dual contouring [18] extract the isosurface from the active grid cells (the grid cells with isosurface intersections) of the volume as a mesh that can be rasterised efficiently by the graphics processing unit (GPU). An example of an extracted isosurface mesh is shown in Figure 2.8. A useful property of the Marching Cubes algorithm is that it generates closed (watertight) meshes, which we utilise in Paper III to generate 3D printable models from volume data.

Point-based rendering methods have also been proposed for high-quality isosurface rendering. In point-based rendering, the isosurface is extracted as a point cloud of surface elements (surfels) [42], which is typically more efficient to store than a triangle-based representation. In elliptical weighted average (EWA) splatting [59][4], surface points are rasterised as disks in an initial visibility pass, with normals and other attributes interpolated later in a separate blending pass. An example of disk-based EWA splatting is shown in Figure 2.8. Splatting with billboards can also be used to reconstruct surfaces from median-axis skeletons [17]. In Papers I–II, we perform point-based rendering of point clouds (pointshells) also used for haptics.

Figure 2.7. Isosurfaces for different isovalues k in the same volume data (MANIX CT from the OsiriX DICOM Image Library), rendered in the BoneSplit [36] software.

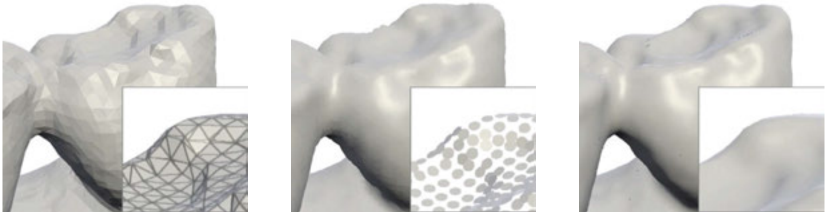


Figure 2.8. Explicit isosurface representations obtained from the same volume data (tooth from mandible in cover figure): (left) Marching Cubes mesh, with triangles rendered with face normals; (center) point cloud, with points (surfels) rendered as disks using EWA-splatting; (right) implicit raycasted isosurface shown for comparison.

2.2.2 Direct Volume Rendering

Direct volume rendering techniques avoid storing an explicit representation of the isosurface and can therefore be more suitable for dynamic volume data and dynamic isovalues. Direct volume rendering also enables other projections of the data, such as maximum intensity projection or compositing with color and opacity transfer functions [11, 51], as shown in Figure 2.9.

Object-order volume rendering include texture mapping [11] that renders the volume by rasterising textured slices with alpha blending. Volume splatting [57] is another object-order method in which a footprint function of each voxel is projected on the pixel grid. In shell rendering [52], which is similar to splatting, voxels are mapped by a fuzzy membership function and rasterised in front-to-back order to obtain a fuzzy representation of the isosurface.

Image-order volume rendering such as GPU-based volume raycasting [22] sample the volume along rays cast through the pixel grid (Figure 2.10). In Paper VII, we implement isosurface raycasting to visualize isosurfaces in vol-

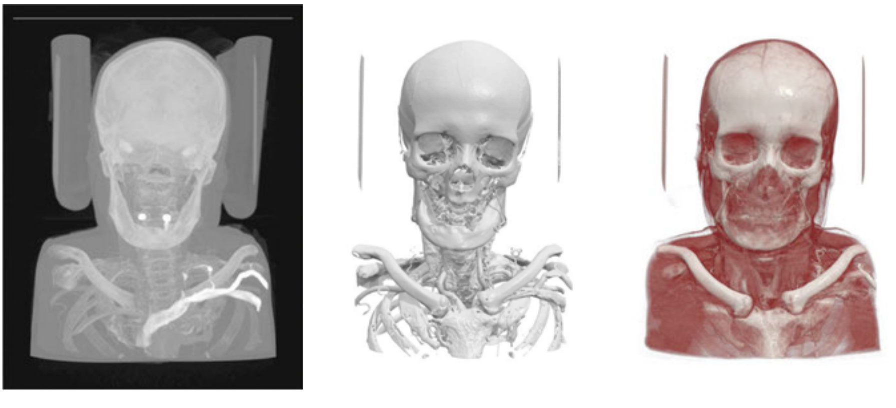


Figure 2.9. Different volume renderings of the same volume data (MANIX CT): (left) maximum-intensity projection; (middle) isosurface raycasting; (right) compositing with a transfer function. Images courtesy of Johan Nysjö.

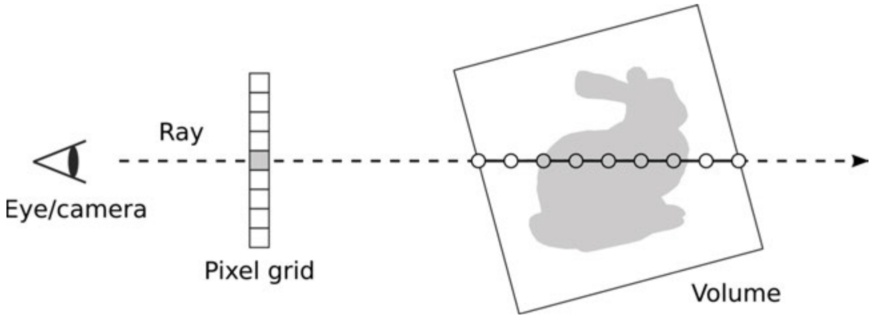


Figure 2.10. Illustration of volume raycasting.

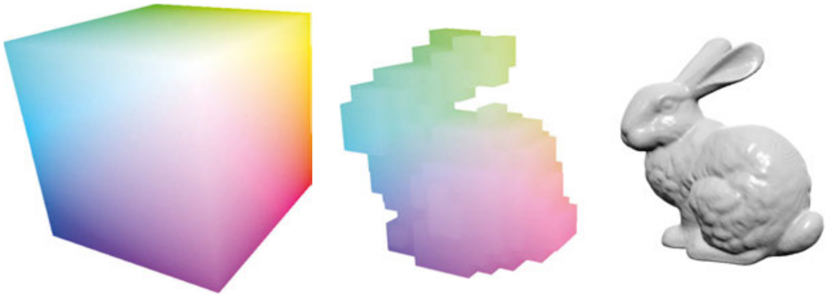


Figure 2.11. Empty-space skipping for isosurface raycasting: (left) proxy geometry used for coarse empty-space skipping outside the volume; (middle) min-max blocks used for finer empty-space skipping inside the volume; (right) raycasted isosurface. Colors in the two left images represent normalized 3D texture coordinates.

ume data also used for haptic rendering. In Paper V, we propose an isosurface rendering method that amortises the cost of isosurface raycasting over time.

Cell rasterisation was proposed by Liu et al. [27] as a fast method for rasterising the active grid cells of a volume. Active grid cells are rasterised as point primitives via so-called billboard splatting, and the original volume data is sampled in the cells in the fragment shader to find intersections and compute smooth normals. The authors also use a data structure for fast sorting and traversal of the cells in front-to-back order for transparency rendering. In Paper VI, we extend cell rasterisation with an efficient data structure that stores only the scalar values of the active grid cells for data reduction.

2.2.3 Acceleration Data Structures

Image-order methods such as GPU-based raycasting usually benefit from acceleration data structures for ray traversal and empty-space skipping [22]. Such data structures can further provide efficient storage of the volume data. Object-order isosurface rendering can also benefit from similar data structures during the isosurface extraction step.

A coarse min-max block volume extracted from the full volume data is a simple acceleration data structure that can be used for empty-space skipping, for example by rasterising the blocks for ray start and stop points as in Figure 2.11. Min-max octrees are also used for similar purpose.

For memory efficiency, large volume data is often stored in bricks, such as in the GigaVoxels [9] and VDB [32] data structures. Sparse voxel octrees have also been proposed for storage and ray traversal of large voxel models [23]. The octree typically stores surface voxels in the leaf nodes, and is traversed during ray tracing. However, for high-quality isosurface rendering, a drawback of traditional sparse voxel octrees representing voxels as small cubes is the limited contour information they provide, leading to a blocky appearance unless each voxel is projected to less than a few pixels. In Paper VI, we store the corners of grid cells in a sparse data structure, which provides the same implicit isosurface as the original volume data.

Object-order volume rendering can also benefit from visibility culling to improve the rendering performance. A hierarchical Z-buffer [14] can for example be used to cull non-visible objects or clusters of geometry. In Paper VI, we improve the rasterisation performance of the isosurface rendering method in the paper by implementing a visibility buffer for cluster culling.

2.2.4 Global Illumination

Global information such as shadows and ambient occlusion (AO) provide important depth cues to the visualization, that can improve the perception of a shape. Classical Whitted-style ray tracing allows hard shadows and mirror-like reflections to be rendered, but not fuzzy effects such as soft shadows or reflections from glossy surfaces. In Paper V, we use a rendering technique similar to the Render Cache [54] to cache raycasted hard shadows and AO in the isosurface visualization. Full global illumination with indirect light can

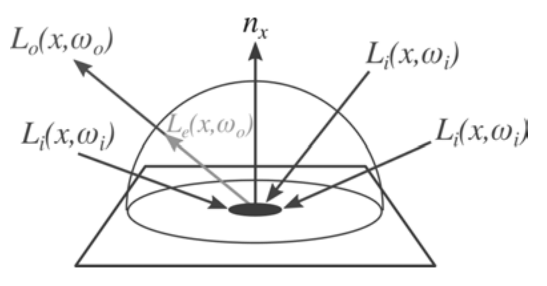


Figure 2.12. Ray sampling over a hemisphere Ω for the surface normal n_x, illustrating incoming radiance (L_i), emissive radiance (L_e, and outgoing radiance (L_o) in the Rendering Equation.

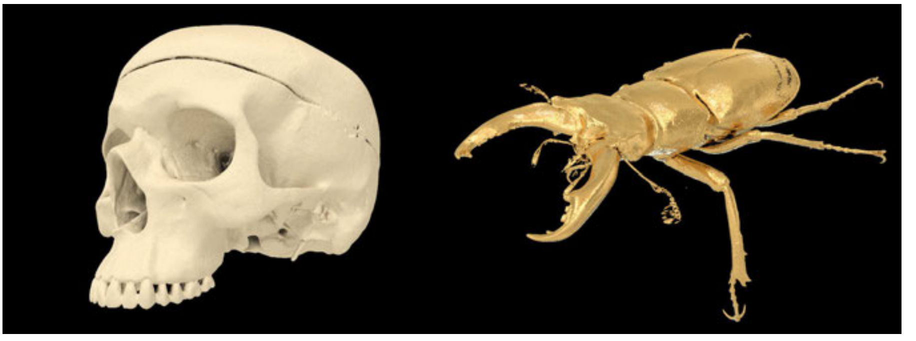


Figure 2.13. Isosurfaces visualized with global illumination, using the method in Paper VI with cell rasterisation for primary rays and path tracing with three bounces and a high-dynamic range environment map for secondary rays.

also be computed by solving the Rendering Equation [19] (Figure 2.12),

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} f_r(l, \omega_i, \omega_o) L_i(x, \omega_i)(\omega_i \cdot n_x) d\omega_i \qquad (2.4)$$

via path tracing and stochastic sampling. Path-traced isosurfaces are shown in Figure 2.13. The convergence rate of the path tracing can be improved by careful choice of sampling sequences [43]. For interactive rendering, denoising and temporal anti-aliasing [20] may also be used.

2.3 Stereoscopic Visualization and Haptics

This section introduces stereoscopic displays [16] and basic stereoscopic rendering. We also introduce haptic devices for kinesthetic feedback (force and torque feedback) and methods for haptic rendering [25].

2.3.1 Stereoscopic Displays

Traditional monoscopic displays can only provide a subset of the visual cues that the human vision system uses to perceive depth in the real world. This includes important monoscopic depth cues such as perspective and shadows, but excludes stereoscopic depth cues such as retinal parallax and vergence. Stereoscopic displays, on the other hand, can provide stereoscopic depth cues to further improve the depth perception of virtual scenes. In addition, monoscopic and stereoscopic displays can be coupled with head-tracking to provide motion parallax, which is another important depth cue.

Most planar stereoscopic displays use image multiplexing to present the viewer with a stereoscopic image. This requires the viewer to wear either active (powered) or passive (non-powered) glasses. Active glasses are used with time multiplexing that alternates between displaying for the left and right eyes at a high display refresh rate, typically 120 Hz instead of 60 Hz. Passive glasses are used with for example polarisation-based multiplexing that displays the left and right eye images simultaneously at different polarisation angles. When coupling the display with head-tracking, so-called "Fish Tank Virtual Reality" [56], additional depth cues from motion parallax are provided.

Figure 2.14. Planar stereoscopic displays: (left) DevinSense Display 300 with active glasses; (right) zSpace 200 display with passive polarised glasses and built-in infrared cameras for head-tracking.

Figure 2.15. Head-mounted displays: (left) Oculus Rift CV1; (right) HTC Vive.

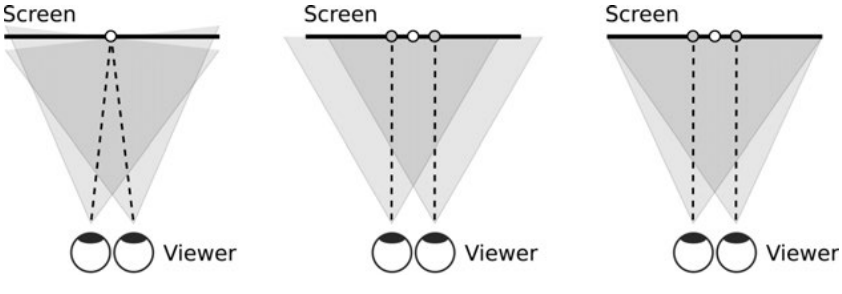


Figure 2.16. Projections for stereoscopic rendering: (left) "toed-in" on-axis frustums introducing parallax errors; (center) parallel on-axis frustums used with some HMDs; (right) off-axis frustums used with planar stereoscopic displays.

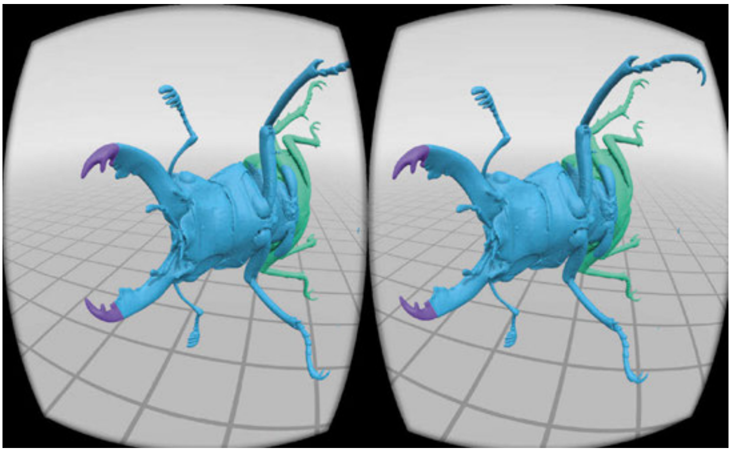


Figure 2.17. Stereoscopic rendering of Stag beetle micro-CT dataset, using the Ray-Caching method of Paper V and with the rendering shown pre-distorted for display on an Oculus Rift HMD.

For the interaction in Papers I and II, we use a DevinSense Display 300, which uses active NVIDIA 3D Vision glasses and a half-transparent mirror for co-location of the visual and haptic workspaces [40]; we further pair the display with an OptiTrack optical tracking system for head-tracking. For the interaction in Papers III and IV, we use a zSpace 200 display with polarisation-based passive glasses and built-in infrared head-tracking cameras.

Head-mounted displays (HMDs) are used for immersive virtual reality (VR) to provide a stereoscopic image and motion parallax from head-tracking. Unlike planar stereoscopic displays that use image multiplexing, HMDs use separate displays attached in front of the viewer's eyes. Recent HMDs such as the Oculus Rift CV1 and the HTC Vive (Figure 2.15) can provide high image resolution and a wide field-of-view, as well as robust 6-DOF tracking via either external sensors or sensors for inside-out-tracking mounted on the HMD and controllers. We use an Oculus Rift CV1 for the experiments in Paper V.

Stereoscopic rendering requires the scene to be rendered for the left and right eyes (Figure 2.17). The image for each eye is typically rendered at either half or full resolution, depending on the type of image multiplexing or display technology used. Planar stereoscopic displays require the use of off-axis projections (Figure 2.16) to provide the viewer with correct stereo parallax [26], and also correct perspective when head-tracking is used. HMDs further require a pre-distortion of the image to compensate for the distortion introduced by the HMD lenses, a task which is usually handled by VR APIs such as OpenVR[1]. In stereoscopic rendering, there is also a high amount of spatial and temporal coherence between the left and right eye, which we exploit in Paper V for faster stereoscopic rendering of isosurfaces.

[1]OpenVR: `https://github.com/ValveSoftware/openvr`, accessed on January 23, 2020.

Figure 2.18. Haptic devices: (left) Novint Falcon with 3-DOF position tracking and 3-DOF force feedback; (right) Geomagic Touch with 6-DOF position and orientation tracking and 3-DOF force feedback.

2.3.2 Haptic Devices

Haptic feedback include tactile feedback, for example vibration or skin stretching, and kinesthetic feedback, which is force and torque feedback. A common type of kinesthetic haptic device is an interface with which a user interacts with an object by grasping and manipulating a handle (or stylus) attached to a mechanical linkage. The device contains sensors that continuously track motion (in the case of impedance-driven devices) or force (in the case of admittance-driven devices) applied to the handle, combined with actuators for generating force and torque feedback to the user. This type of device can provide a bi-directional interface between the user and the virtual environment.

Commercial haptic devices vary greatly in workspace size, input degrees-of-freedom (DOF), feedback (force and torque) DOF, maximum feedback amplitude, and price. Devices commonly used in research include the Geomagic Touch and the Novint Falcon, both shown in Figure 2.18. The Geomagic Touch tracks the handle position and orientation in 6-DOF and provides 3-DOF force feedback (maximum force 3.3 N), whereas the low-cost Novint Falcon is based on a delta-robot design that tracks position in 3-DOF and provides 3-DOF force feedback (maximum force 8.9 N). Other commercially available but more expensive devices include the Geomagic Phantom series, which support torque feedback in the Premium 1.5/3.0 models, and the Force Dimension Delta devices based on the same delta-robot design as the Novint Falcon but intended for professional applications. In Papers I–II, we use the Phantom Premium 1.5 in order to render 6-DOF force and torque feedback for the collision and attraction forces, whereas in Papers III, IV, and VII we use the Touch and Falcon devices for more basic 3-DOF force feedback.

Sending force commands and reading input from the haptic device is typically handled via an application programming interface (API), such as the proprietary OpenHaptics API used with Geomagic devices. Open-source hap-

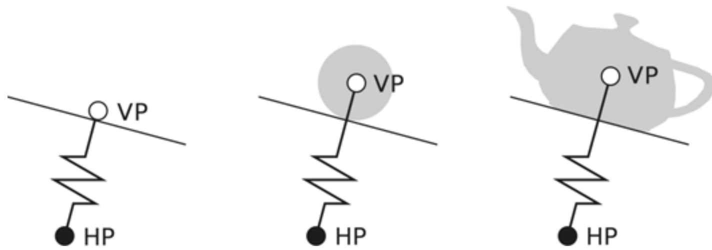


Figure 2.19. Haptic rendering algorithms: (left) God-object (point proxy); (center) Ruspini (sphere proxy); (right) 6-DOF haptic rendering with complex geometrical shape (teapot). HP represents the position of the haptic device handle in the scene, whereas VP represents the proxy position from the virtual coupling spring.

tic frameworks such as the H3DAPI[2] by SenseGraphics AB and CHAI3D [8] by ForceDimension also allow the programmer to write device-independent code. H3DAPI is used in Papers I–IV for the haptic device communication, whereas CHAI3D is used in Paper VII.

2.3.3 Haptic Rendering

Haptic rendering algorithms compute real-time force-feedback when the user interacts with the virtual environment. The force-feedback can for example consist of contact forces when the user is in contact with a virtual object, or guiding forces assisting the user in performing specific tasks. Realistic force feedback requires a high update rate, as much as 1000 Hz, compared to typically 60 Hz required for smooth graphical update. Most haptic rendering software therefore use separate CPU threads for graphics and haptics.

Early proxy-based haptic rendering algorithms were able to simulate the contact between a simple proxy object representing the haptic device and a more complex virtual object such as a polygonal mesh. The God-object algorithm [58] uses an infinitesimal point as proxy, whereas the Ruspini algorithm [48] extends the proxy to a sphere of arbitrary radius. The proxy is usually connected to a virtual coupling spring in order to provide damping and to avoid penetration of the virtual object when the haptic device force limit is exceeded. Implementations of these algorithms for polygonal scenes are found in general haptic rendering frameworks such as H3DAPI and CHAI3D. Lundin Palmerius [29] extends both of these methods to direct volume haptic rendering, which is implemented in for example the WISH toolkit [53] and used for the purpose of interactive segmentation [37, 35].

A limitation of proxy-based algorithms is that each proxy only supports 3-DOF interaction via a single contact point or contact sphere, not allowing for collision between more complex shapes. There has been a large amount of

[2] `https://www.h3dapi.org`, accessed on January 23, 2020.

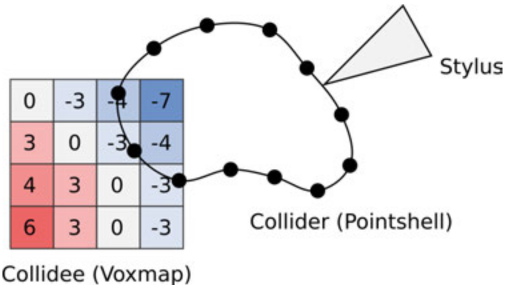


Figure 2.20. The Voxmap PointShell data structure and algorithm.

research into so-called 6-DOF haptic rendering algorithms, both for rigid and deformable shapes. In this thesis, we will restrict the discussion to rigid-rigid body collisions. The Voxmap PointShell (VPS) algorithm [30] is a penalty-based method for 6-DOF haptic rendering of rigid-rigid body collisions that uses a dual virtual object representation, a voxmap and a pointshell, for contact queries. Penalty forces are computed by querying pointshells against the voxmaps of other virtual objects (Figure 2.20) and applying Hooke's law. VPS was later improved with quasi-static virtual coupling [55], distance maps [31], and adaptive sampling via a uniformly distributed pointshell [1]. Other rigid-rigid 6-DOF haptic rendering algorithms are discussed in [41] and [47].

Haptic rendering, like many other types of real-time physics simulation, often relies on fast collision detection in order to achieve the minimum update rates required. For performance reasons, the collision detection is often divided into a coarse phase and a fine phase. In the coarse phase, larger regions determined not to be in contact are culled from further computation. In the fine phase, contact points for collision response such as penalty forces are computed. An overview of different collision detection techniques such as bounding volumes, intersection tests, and spatial partitioning schemes can be found in [12]. For rigid-rigid 6-DOF haptic rendering, hierarchical data structures such as octrees [30] or sphere-trees [47] are typically used in the coarse collision phase. In Papers I–IV, we implement an octree with a fixed leaf-node size for the coarse collision detection between virtual objects such as segmented bone fragments. In Paper VII, we replace the octree with a simpler min-max block volume, which is further used in the volume visualization in the paper. VPS was used for all fine collision detection in the papers of this thesis.

2.4 Medical Applications

This final section introduces medical visualization and virtual surgery planning, which are the medical applications we have investigated in the thesis. For a more detailed introduction to medical visualization, see Preiem et al. [45]. Recent work on virtual surgery planning can be found in Ritacco et al. [46].

2.4.1 Medical Visualization

Medical imaging such as computed tomography (CT) and magnetic resonance imaging generate grayscale volume data that can be visualized by many of the volume rendering techniques described previously in this chapter. The volume data is generally reconstructed as a stack of 2D slices defining the sample points of the 3D image. Because the spacing between slices can differ from the spacing within slices, voxels in medical images can be anisotropic (elongated) which should be taken into account during visualization when sampling the image for raycasting or computing image gradients for normals.

The grayscale values in CT images represent Hounsfield units (HUs) that provide a direct mapping between scalar values and different tissue types found in the human body. The Hounsfield scale typically uses a 12-bit scalar range, $[-1024, 3071]$, with air defined at -1000 HU, water at 0 HU, soft tissues in the range $[10, 60]$ HU, and bone (soft trabecular bone and hard cortical bone) in the range $[400, 3000]$ HU. Imaging artifacts such as the partial volume effect, resulting from limited resolution in the volume reconstruction [45], mean that grayscale values also can represent a mix of different tissue types. To compensate for the low contrast in the soft tissue HU range, the patient may further be injected with a contrast agent during the imaging to highlight for example blood vessels in a region. Cone-beam computed tomography (CBCT) is sometimes used in medical and in particular dental applications instead of traditional CT, because of the lower radiation and cost and easier image acquisition. CBCT images typically have lower contrast than CT images, and do not provide grayscale values in HU, which should be considered when for example selecting an isovalue or transfer function for visualization.

Tasks such as pre-processing for segmentation often require the user to explore different isovalues in order to find a good isosurface with respect to image noise or blurring from the partial volume effect. The visualization method should in such cases support dynamic isovalues to enable interactive isosurface updates. Direct volume rendering methods such as isosurface raycasting can be used, but might be slow without proper acceleration data structures that must be maintained when the volume is modified. In Paper V, we propose an isosurface raycasting method that does not use acceleration data structures and instead amortises the raycasting cost over time to maintain the performance.

Stereoscopic rendering and head-tracking provide additional depth cues that can improve the visualization of complex structures such as bone or vessels in CT images. Haptic rendering presents another way of exploring medical volume data [29], by providing additional information via the haptic feedback. Structures that can be extracted and visualized as isosurfaces can also be 3D printed as physical props. In Papers I–IV, we present and evaluate a system that combines stereoscopic visualization and haptic rendering for visuohaptic interaction with bone fragments segmented from CT images.

2.4.2 Cranio-Maxillofacial Surgery Planning

In CMF surgery, one fundamental task is restoring the skeletal anatomy in patients after injuries such as trauma to the face. This complex task resembles solving a 3D puzzle in which individual fractured bones need to be moved into their correct positions and orientations. Because small errors can accumulate and result in a poor reconstruction, the accuracy and precision requirements are very high. Careful planning before the surgery can help reducing such errors and shortening the operation time. Another fundamental task in CMF surgery is restoring the skeletal anatomy after tumor resection, using for example a bone and soft tissue transplant from the patient's leg [39].

Virtual surgery planning (VSP) is routinely used in CMF surgery to improve the outcome. Compared to traditional planning performed by paper and pen from 2D X-ray images, VSP allows 3D image data such as CT to be used in the planning and for the design of patient-specific surgical guides and plates. Early work on VSP relied on off-the-shelf software for tasks such as 3D modeling and manual segmentation, which would be too time consuming for most surgeons to learn and use efficiently. Specialised VSP tools such as Materialise Mimics[3] have complex user interfaces limited to 2D interaction with 3D data such as CT, preventing the systems from being widely used by the surgeons themselves. Companies such as Materialise and Planmeca[4] provide planning as a commercial service, which is now the preferred VSP method allowing part of the work such as design of patient-specific guides and plates to be outsourced. However, using a VSP company can be expensive and may require several iterations before a plan can be approved by the surgeon, which can prevent its use in time-critical trauma surgery.

If VSP tools would allow surgeons to efficiently perform the planning themselves, in-house VSP has the potential to reduce time and cost of the planning [34]. This would also allow surgeons to create and evaluate several different plans and select the most promising one. 3D printers are becoming common at hospitals and could be used to fabricate designed guides and plates. There are a few in-house planning and design systems proposed in the literature, for example in orthopedic fracture surgery [21]. In Paper III, we develop methods for modeling surgical guides and plates from segmented CT data, and implement those methods in our HASP system.

[3] Materialise: `https://www.materialise.com`, accessed on January 23, 2020.
[4] Planmeca: `https://www.planmeca.com`, accessed on January 23, 2020.

3. Thesis Contributions

In this chapter, we summarise briefly the contributions of Papers I–VII.

3.1 Virtual Surgery Planning

The main application for the methods developed in this thesis has been CMF surgery planning. In Papers I–IV, we develop methods and a system for performing virtual planning from CT volume data.

3.1.1 The Haptics-Assisted Surgery Planning System

In Paper I, we propose Snap-to-fit, a method for virtual assembly of fractured objects using haptic attraction forces and torques to pull the fractured pieces towards locally optimal fits. The force model for Snap-to-fit is shown in Figure 3.1. In order to reach haptic rendering rates of around 1000 Hz, we employ a fixed-depth octree and a static hierarchical pointshell (Figure 3.2) with LODs for the collision detection. Distance maps and gradient direction fields used in the force model are also pre-computed for fast look-up.

In Paper II, we present the haptics-assisted surgery planning (HASP) system that combines stereoscopic visualization with 6-DOF haptic rendering for

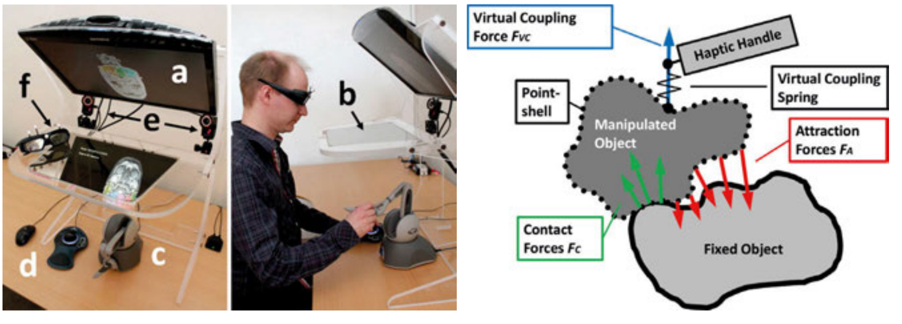


Figure 3.1. Hardware and force model used for the HASP system in Papers I–II: (left) stereoscopic mirror display (a–b), with OptiTrack cameras (e) used for head-tracking of glasses (f), and haptic device (c); (right) proposed Snap-to-fit force model extending the classical 6-DOF Voxmap PointShell algorithm with attraction forces. A Phantom Premium 1.5 haptic device was used for the experiments in Papers I–II instead of the Touch device shown in the left photos.

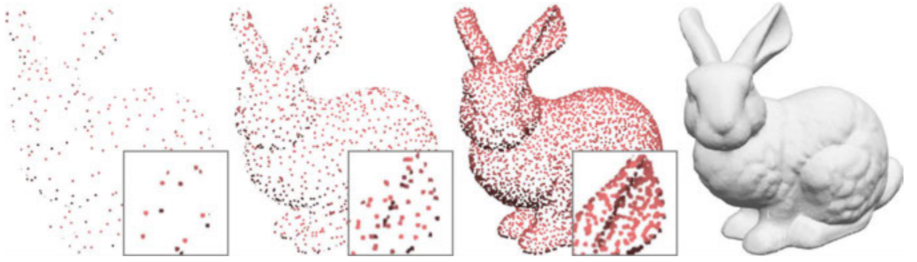


Figure 3.2. Static hierarchical pointshell used in Papers I–IV and also in Paper VII: (left) first three pointshell LODs of voxelized model (bunny); (right) full pointshell.

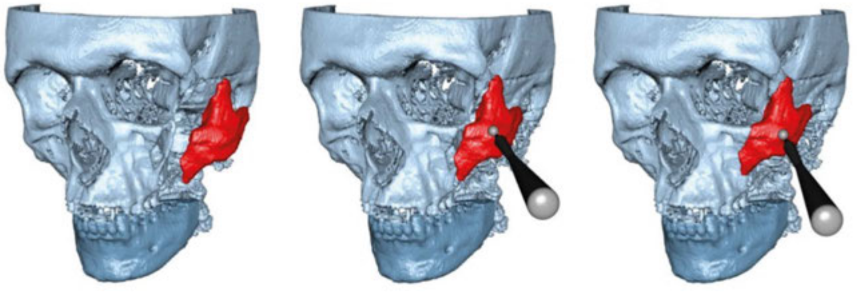


Figure 3.3. Snap-to-fit used in Paper I to assemble a fractured zygoma bone in segmented CT data: (left) original location of the fractured piece (red object); (center) coarse fit using contact forces; (right) final fit using attraction forces.

in-house surgery planning. The system loads segmented CT data to generate Voxmap Pointshell (VPS) representations for the force model in Paper I, and is implemented in C++ and Python, using OpenGL and H3DAPI for graphical rendering and haptic device communication. The system features head-tracking cameras to enable user "look-around" in the graphical scene, and we also implement the ability to group bone fragments for easier assembly of complex cases with many pieces.

Results

We demonstrate the use of Snap-to-fit and the HASP system on applications from CMF surgery and archaeology, and show that the force model with combined contact and attraction forces can achieve a stable fit at haptic rates for surfaces with over 5000 points.

3.1.2 Modeling of Surgical Guides and Plates

The HASP system of Papers I–II was extended by Olsson et al. [39, 38] to further support planning of so-called fibula free flaps, a reconstruction technique in which a bone graft from the patient's leg is used to reconstruct a defect after for example a tumor has been removed. However, this version of HASP

Figure 3.4. Virtual plate design from Paper III, using landmarks that the user places with the haptic device on the mandible to generate the plate model from CSG operations on signed distance fields.

Figure 3.5. Fibula cutting guides generated with the methods in Paper III.

did not support in-house design of patient-specific plates (Figure 3.4) for bone fixation, or cutting guides (Figure 3.5) with slots or flanges to help the surgeon perform precise osteotomies or resections (cuts) according to the plan.

In Paper III, we propose a method for generating 3D printable models of surgical guides and plates from segmented CT data. Models are generated from user-placed landmarks or resections by constructive solid geometry (CSG) operations on signed distance fields computed from implicit functions and anti-aliased distance transforms of coverage representations. We then export 3D printable models from signed distance fields using the Marching Cubes implementation in [5]. Further details are provided in our paper.

The method and modeling tools were implemented in HASP, which had further been extended with a workflow-based graphical user-interface (Figure 3.6) and a new rendering framework replacing H3DAPI and supporting physically based rendering, shadows, and a stencil routed A-buffer [33] for showing transparent parts. A zSpace 200 display and a GeoMagic Touch haptic device were used for stereoscopic and haptic display. Segmented CT data was converted into Voxmap PointShell representations similar to in previous papers, to enable haptic interaction between bone surfaces and the stylus.

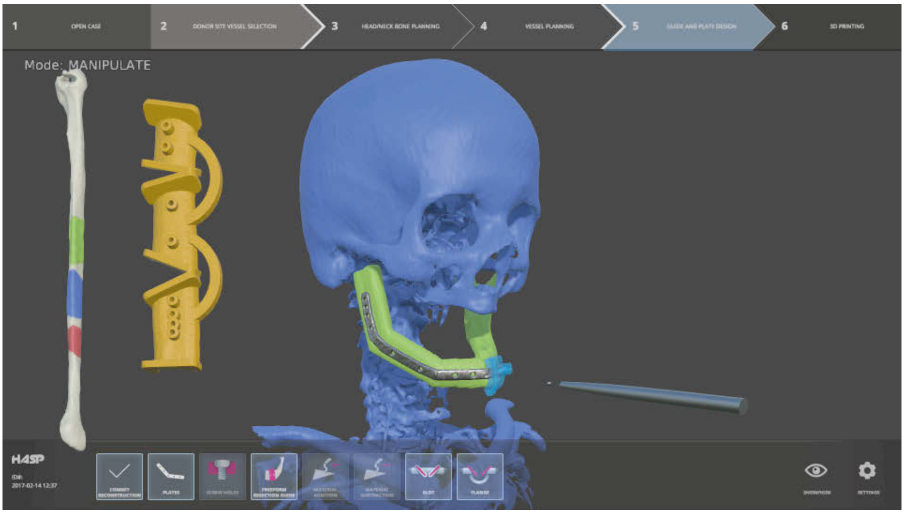


Figure 3.6. Fibula reconstruction in HASP, showing a planning of a tumor resection of the mandible with the fibula on the left, and surgical guides and plates generated by the modeling methods in Paper III.

Results

Using an anti-aliased DT resulted in minimum distances closer to the expected shell offsets, and visibly smoother iso-surfaces, compared to a binary DT. Figures 3.4–3.5 show 3D-printable models generated in HASP. The method and design tools we have developed allows a user of the system to quickly design such models within a few minutes.

3.1.3 Virtual Reconstruction of Mandibular Fractures

The aim of the study in Paper IV is to evaluate the new workflow for virtual reduction of complex mandible fractures proposed and implemented in Paper II, and to investigate its usability, accuracy, and precision.

An overview of the evaluation study is shown in Figure 3.7. The reconstruction planning was performed by three users in the extended version of HASP

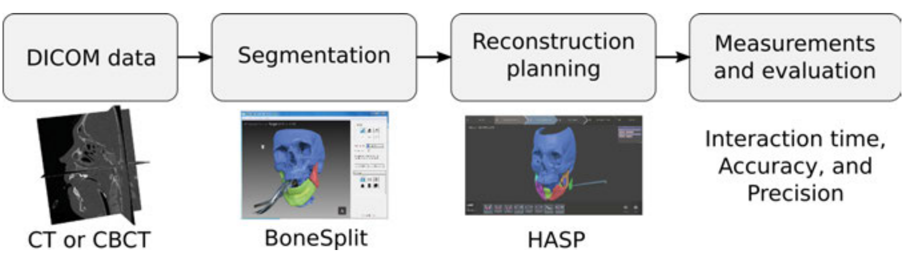


Figure 3.7. Overview of the evaluation study in Paper IV.

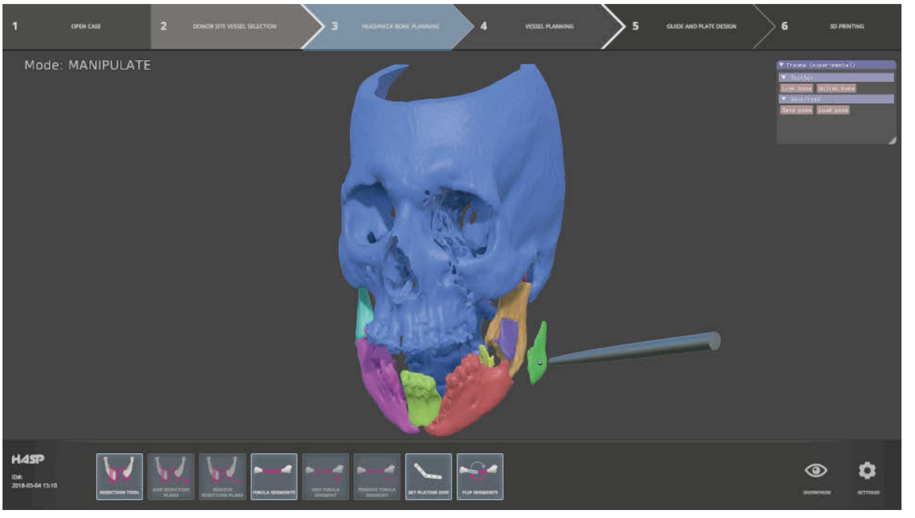


Figure 3.8. Trauma reconstruction in HASP, from the evaluation study in Paper IV.

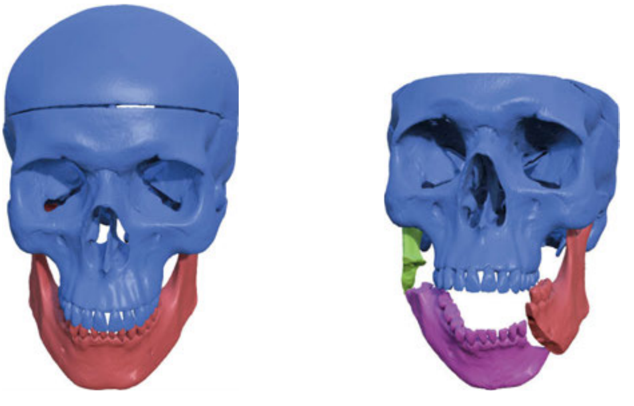


Figure 3.9. Plastic skull CT data used for the evaluation in Paper IV: (left) segmentation of intact plastic skull; (right) segmentation of the same plastic skull after the mandible has been artificially fractured before CT imaging.

from Paper III (Figure 3.8), with two trials per user and case, on a scanned plastic skull used for reference (Figure 3.9), and twelve retrospective cases (Figure 3.10). After each trial, the poses of the bone fragments were recorded. Segmentation of the cranium and individual bone fragments of the mandible in the datasets was also performed by one of the users, with input from a clinical expert, and the segmentation times recorded. Segmentation was performed with BoneSplit [36] from CT and CBCT data provided in DICOM format. The segmented cases were then loaded into HASP for the reconstruction planning.

To evaluate the intra- and inter-operator precision, we computed the measurements surface-to-surface distance [7], maximum absolute distance, and rotational pose difference. To avoid the need for registering reconstructions

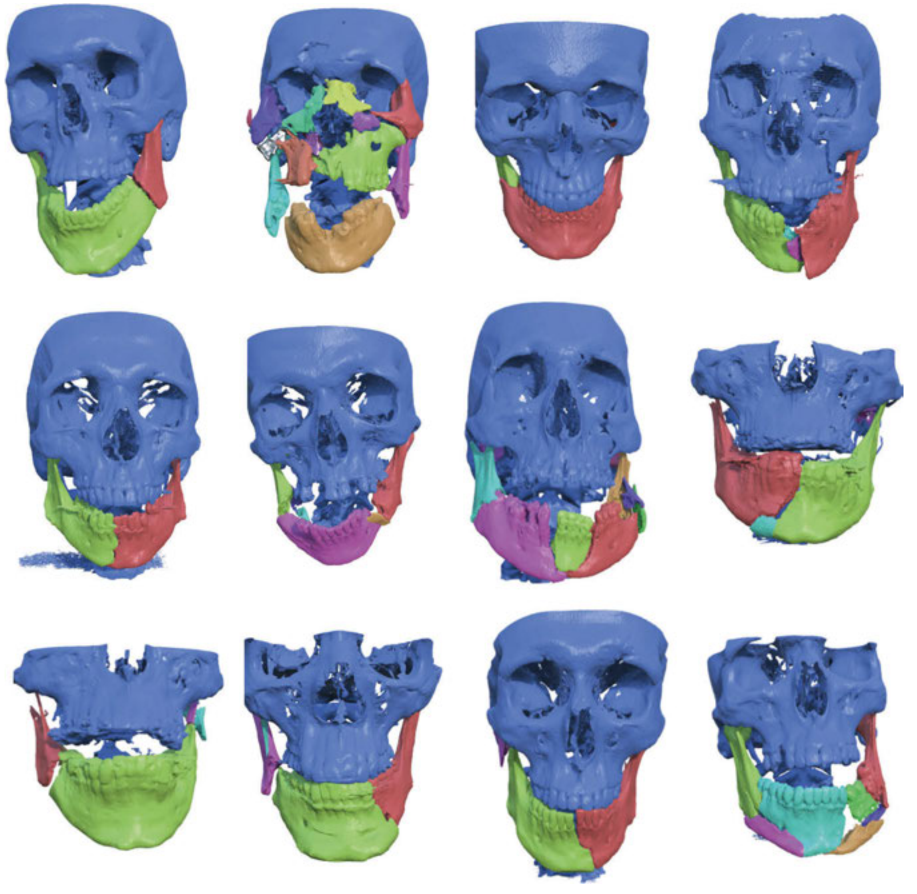


Figure 3.10. Segmented CT data of retrospective cases used for the evaluation study in Paper IV.

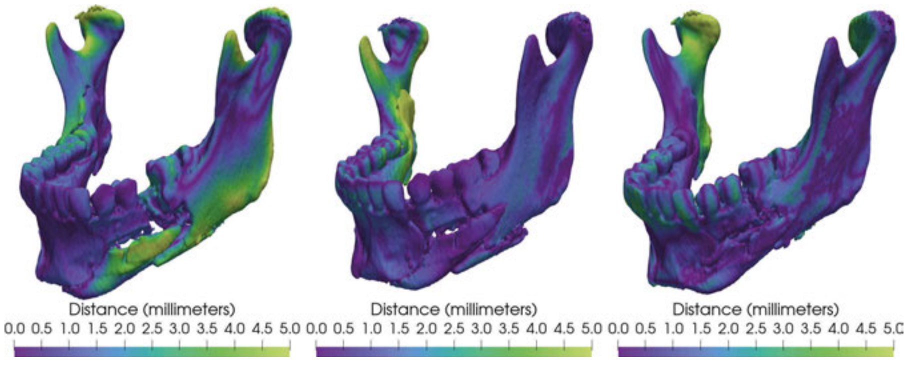


Figure 3.11. Fractured mandible virtually reconstructed by three users, with colormap showing absolute error (surface-to-surface distance) for each user between two trials.

from the same case against each other, which would be challenging for the retrospective cases with no reference, the cranium was used as a fixed reference point. Surface-to-surface distances for a case are shown in Figure 3.11.

Results

The study results showed high precision but fairly low accuracy. Some of the cases were particularly challenging because of lack of easily identifiable landmarks and lack of context from tissue surrounding the bone. Other possible error sources include the segmentation and lack of sufficient fidelity in the haptic rendering. Average planning time (including time for segmentation) was around 15 minutes, which demonstrates the potential for using HASP as a tool in trauma applications.

3.2 Isosurface Visualization

Isosurface visualization is a common task in medical visualization and applications such as VSP that involve large CT volume data. In Papers V–VI, we present fast and efficient rendering methods for isosurface visualization.

3.2.1 RayCaching

In Paper V, we propose a direct volume rendering method, RayCaching, that can be used without pre-computed data structures to render isosurfaces in volume data with dynamic isovalues. An overview of our method is shown in Figure 3.12. The basic idea is to generate a dynamic point cloud that follows the viewer, using a clipmap [50] to cache primary rays and amortising the cost of raycasting over time. We further exploit coherence between the left and right eyes for more efficient stereoscopic rendering, and compute raycasted hard shadows and ambient occlusion that also are cached in the point cloud.

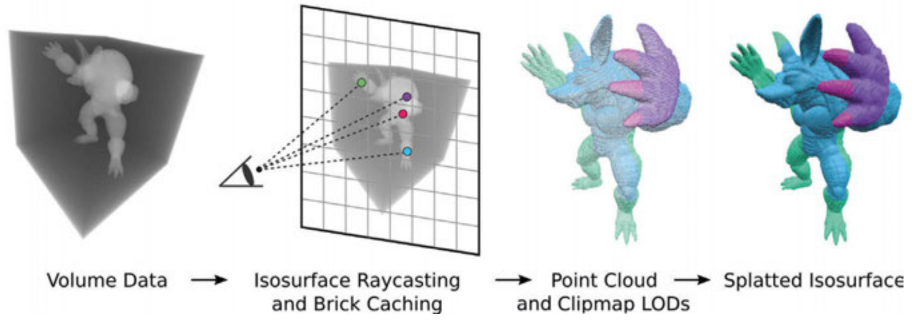


Figure 3.12. Overview of the method in Paper V.

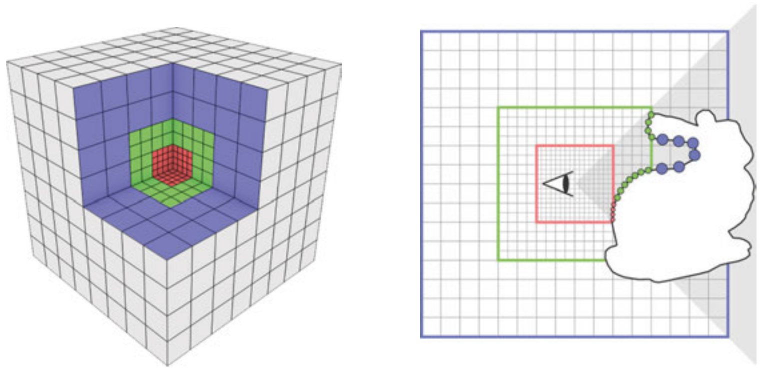


Figure 3.13. Clipmap and dynamic point cloud from method in Paper V.

Results

Performance suggests that the RayCaching method is suitable for immersive virtual reality (VR) and rendering for high-resolution head-mounted displays (HMDs), which was the intended main application. An example of stereoscopic rendering with RayCaching is shown in Figure 2.17. While our method is mainly compared against direct isosurface raycasting without acceleration structures for empty-space skipping, our method would also benefit if such acceleration structures were used. Compared to isosurface extraction and mesh rendering, our method requires the original volume data during rendering, but can be faster for primary rays and enables amortising of secondary rays.

3.2.2 Clustered Grid Cell Data Structure

In Paper VI, we propose an indirect volume rendering method using a memory efficient data structure (Figure 3.14) for storing positions and corner scalar values of active grid cells extracted from a volume. The grid cells are stored in clusters that can be rendered by cell rasterisation via billboard splatting, and also ray traced to compute global illumination (Figure 3.15). Improvements to the basic cell rasterisation include visibility culling of clusters and EWA-like attribute splatting for smooth interpolation of normals and other attributes.

Results

We demonstrate that the proposed grid cell-based data structure can be used for fast cell rasterisation of isosurfaces, as well as for hybrid ray tracing (Figure 2.13) of isosurfaces extracted from large volumes. The data structure uses about 37% memory compared to a typical mesh-based representation, while providing faster rasterisation for larger isosurfaces with high depth complexity. Good use cases for our method include digital sculpting and the CSG modeling in Paper III, as well as visualization of data that need storing additional attributes per cell or voxel.

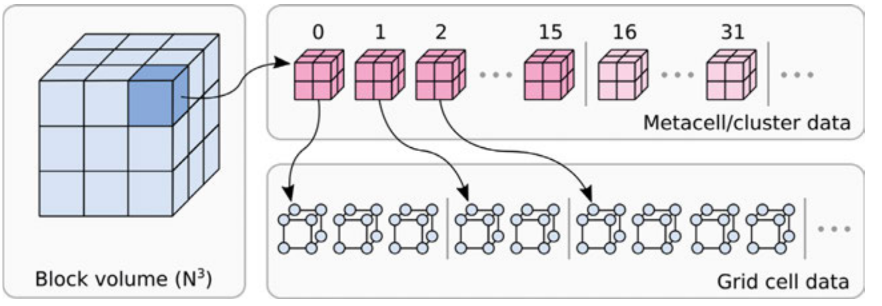


Figure 3.14. Data structure in Paper VI.

Figure 3.15. Overview of the method in Paper VI.

3.3 Haptic Rendering

In the final Paper VII, we look at improving the performance and fidelity of the 6-DOF haptic rendering method used in Papers I–IV.

3.3.1 Vectorised Haptic Rendering

Because 6-DOF haptic rendering of stiff surfaces demands a high update rate (1000 Hz) and imposes strict real-time constraints on the computations, parallelism can be hard to exploit via traditional means such as multi-threading or the GPU. In Paper VII, we show how the classical VPS haptic rendering method can be efficiently vectorised via the ISPC [44] programming language. In addition, we propose a dynamic pointshell (Figure 3.16) that does not require any pre-processing and allows a fixed point budget to be set per frame. The dynamic pointshell follows the contact regions and is updated per haptic frame, and is stored in a circular buffer.

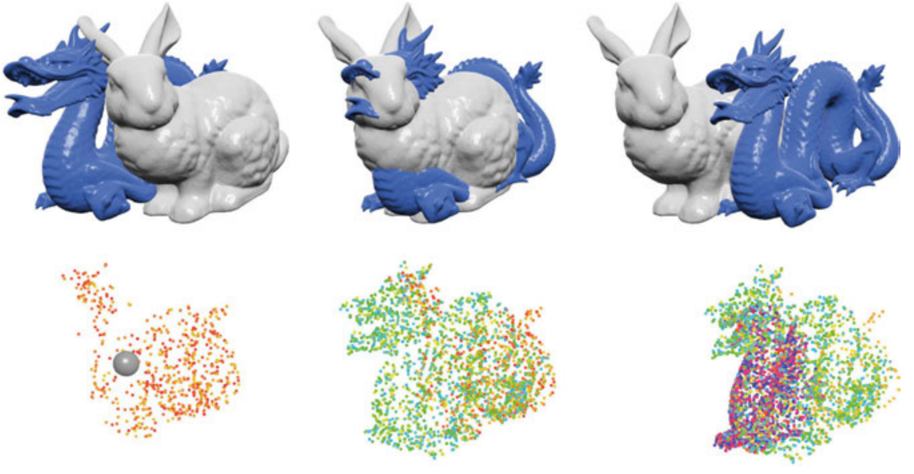


Figure 3.16. Dynamic pointshell proposed for haptic rendering in Paper VII.

Results

Our main objective in this paper was to increase the number of contact points that can be processed per haptic frame. Compared to a corresponding scalar C++ implementation, our ISPC vectorised version of VPS provided a 3× speedup on a single CPU core for a static hierarchical pointshell. This speedup typically allowed an additional LOD to be processed within the haptic time-frame. For the dynamic pointshell, the vectorised version was between 2.5× and 5× faster, which allowed the pointshell capacity to be increased.

In the empirical study, we demonstrated that the increased fidelity in collision simulation afforded by the vectorisation directly leads to a significantly improved precision in virtual assembly tasks. The task was to assemble fractured blocks (Figure 3.17) using a Novint Falcon haptic device with maximum force set to 9 N. The five users were unable to distinguish a difference in haptic response between the two implementations, but performed the alignment tasks

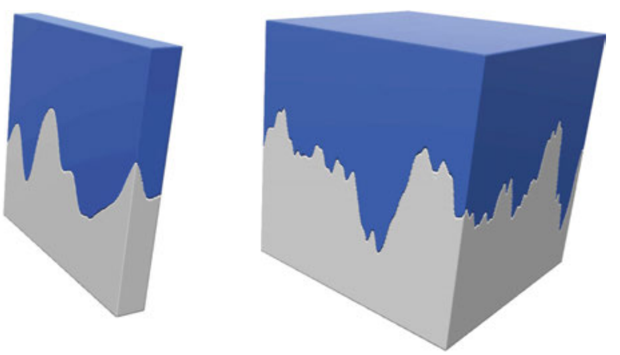


Figure 3.17. Fractured blocks from test scenes of the empirical study in Paper VII.

with improved precision when the haptic fidelity was increased. In the empirical study, the vectorised haptic rendering showed significantly ($p < 0.01$) reduced alignment error compared to the scalar version.

4. Conclusions

This chapter summarises the contributions of this thesis and proposes some directions for future work.

4.1 Summary of Contributions

The objectives for this thesis, as stated in Chapter 1, were to:

- Develop fast methods for visualization and haptic rendering of isosurfaces in volume data, including medical CT images.
- Combine image processing, visualization, and haptics into a VSP system for improving the current workflow of CMF surgery planning.
- Evaluate the accuracy, precision, and usability of our VSP system.

Our contributions in the thesis can be summarised as follows:

- In Papers I, II, and IV, we describe the development and evaluation of an in-house VSP system that uses stereoscopic visualization and haptics to allow 3D interaction with 3D volume data such as segmented CT images. Our proposed system, the haptics-assisted surgery planning (HASP) system, is intended for virtual fracture reconstruction and CMF surgery planning. A novel force model, called Snap-to-fit, that extends an existing 6-DOF haptic rendering method with attraction forces between fractured surfaces is also proposed in Paper I.
- In Paper III, we propose a method for generating 3D-printable surgical guides and plates from segmented CT images using anti-aliased distance transforms, and implement modeling tools in a version of HASP extended to support planning of fibula free flap reconstructions.
- In Papers V and VI, we propose fast and efficient rendering methods for visualizing isosurfaces in volume data. The method in Paper V is a direct volume rendering method that amortises the raycasting cost, whereas the method in Paper VI is an indirekt volume rendering method that implements a memory efficient data structure allowing rasterisation and ray tracing of isosurfaces extracted from large volumes.
- In Paper VII, we propose improvements in performance and fidelity to the existing 6-DOF haptic rendering method used in Papers I–IV.

In addition, the HASP system has been installed at two hospitals for further evaluation in clinical research. We also provide full source code for the implementations of the rendering methods proposed in Papers V–VII.

4.2 Future Work

An important need in VSP that was identified during the project was that the surgeons wanted to use CBCT more in the planning, for the easier imaging and lower radiation to the patient. This would likely require investigating new methods for segmentation and for improving the local contrast of CBCT such that it can be used in-place of CT in the current image processing pipeline. Extending the HASP system to other applications, for example, in archaeology and digital humanities, would also likely require new data pipelines. It would be interesting to explore other uses of haptics for the type of modeling that we developed in Paper III, for example, to test the fit of parts during the modeling before exporting them for manufacturing. There is also an ongoing study at the Mount Sinai Hospital, New York, to evaluate the guide and plate design in HASP on a larger number of post-operative cases.

The use of ray tracing and path tracing for realistic rendering of global illumination is becoming more common in visualization and interactive applications, while previously mainly being used in film rendering and other offline rendering. Combining the caching of global illumination in Paper V with the hybrid ray tracing of Paper VI for improving the interactive performance is something that would also be interesting to investigate.

5. Summary in Swedish

Interaktiva system för att utforska och interagera med tredimensionell (3D) medicinsk bilddata via tekniker som stereoskopisk visualisering och haptik kan leda till nya arbetssätt för virtuell kirurgiplanering inom till exempel craniomaxillofacial (CMF) kirurgi. Sådan planering inkluderar även design och modellering av patientspecifika sågguider och plattor som kan tillverkas med additiv teknik (3D-utskrift) och användas under själva operationen.

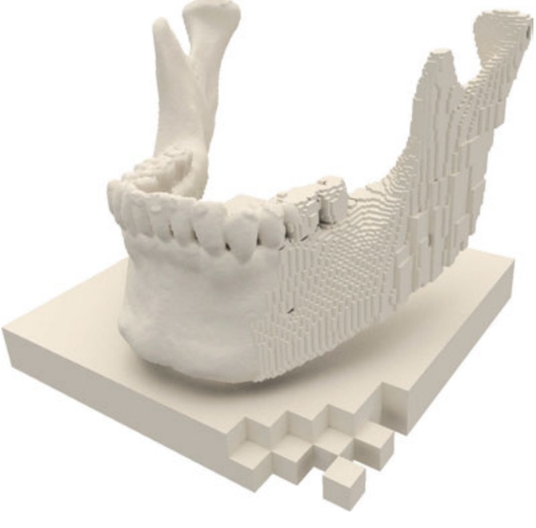
Eftersom medicinsk bilddata ofta består av högupplösta volymbilder såsom datortomografier (DT) med miljontals datapunkter krävs effektiva metoder för att behandla och visualisera datat. Detta gäller framförallt för interaktiva tillämpningar där visualiseringen måste ske i realtid. En vanlig visualiseringsuppgift är exempelvis att rendera isoytor som visar benet eller annan vävnad i datat. För haptik, som innefattar känsel och kraftåterkoppling, krävs i regel dessutom en hög uppdateringsfrekvens på 1000 Hz för realistisk haptisk rendering och återkoppling utan oönskade skakningar eller vibrationer.

I den här avhandlingen har vi undersökt och utvecklat metoder för visualisering och haptisk rendering av isoytor i volumetrisk bilddata, med tillämpningar inom medicinsk visualisering och virtuell kirurgiplanering. Vidare har vi utvecklat metoder för modellering av kirurgiska sågguider och plattor utifrån segmenterad DT-data, samt implementerat dessa i modelleringsverktyg som vi integrerat i vårt system HASP för haptikstödd kirurgiplanering. Detta projekt har utförts in nära samarbete med klinisk forskning vid Institutionen för kirurgiska vetenskaper vid Uppsala Universitet. En annan viktig del i avhandlingen har även varit att utvärdera användbarheten av planeringen i HASP för det tänkta användningsområdet CMF-kirurgi.

För virtuell kirurgiplanering utvecklar vi i artiklarna I–II en metod och ett system för visuohaptisk inpassning av frakturerade benytor. Systemet, HASP, utnyttjar stereoskopisk visualisering och haptik för att låta kirurgen interagera med datat i 3D. I artikel IV utvärderar vi den virtuella planeringen i en användarstudie med DT-data från ett större antal postoperativa frakturfall. Resultaten från användarstudien visar på att den virtuella planeringen kan utföras snabbt (inom 15 minuter, inklusive tid för segmentering) och med hög precision för samma användare, men att noggrannheten i planeringen behöver förbättras. I artikel III utvecklar vi metoder som använder avståndstransformer för att modellera 3D-utskrivbara patientspecifika kirurgiska sågguider och plattor, och integrerar dessa i HASP. Systemet är nu installerat på Akademiska sjukhuset i Uppsala för utvärdering av användning inom klinisk forskning.

För visualisering av isoytor har vi utvecklat två metoder i avhandlingen. Metoden i artikel V kan användas för direkt volymrendering av isoytor i datat, utan några förberäknade datastrukturer och med dynamiskt isovärde, och bygger på ett dynamiskt punktmoln som följer användarens huvudrörelser. Vi utnyttjar redundans mellan vänster och höger öga för snabbare stereoskopisk rendering, och demonstrerar tillämpningar inom virtuell verklighet. Metoden i artikel VI kan användas för indirekt volymrendering av isoytor i datat, och implementerar en minneseffektiv datastruktur som tillåter både rasterisering och strålspårning ("ray tracing") av isoytor extraherade från stora volymer.

Slutligen, för haptisk rendering av isoytor utökar vi i artikel VII den befintliga haptikrenderingsmetod som används i HASP med syfte att snabba upp metoden och möjliggöra fler kontaktpunkter vid inpassning av frakturerade ytor. Utökningen består av ett dynamiskt punktmoln som inte kräver någon förberäkning och som anpassar sig till kontaktytorna, samt vektorisering via programmeringsspråket ISPC. En användarstudie visar att det ökade antalet kontaktpunkter vi uppnår tack vare den nya metoden signifikant minskar felet som användaren gör vid inpassningen av två bitar mot varandra utan att för den skull öka tiden det tar att göra inpassningen.



6. Acknowledgements

I would like to thank past and present colleagues at the Centre for Image Analysis and the Division of Visual Information and Interaction, Uppsala University, as well as others who have in some way contributed to this thesis. In particular:

- Ingela Nyström, my main supervisor, for giving me a lot of freedom and support during these years, and for trying to make me see the larger picture of the work instead of just the technical details.
- Filip Malmberg, my assistant supervisor, for support and good scientific advice, and for the nice discussions about rendering.
- Ingrid Carlbom, my assistant supervisor, for having confidence in me and bringing me onto the haptics project. And for improving my writing and appreciation of the English language.
- Pontus Olsson, for good collaboration on many of the papers, fun times traveling together, and for teaching me a lot about hardware and haptics.
- My brother Johan Nysjö, for comments on the thesis and many of the papers, and for always being an inspiration when it comes to writing, code quality, and knowledge of computer graphics. And thanks for our fly fishing trips together during summers!
- Johanna Nilsson, for good and fruitful collaboration on Paper IV.
- Andreas Thor, for good collaboration, and for always making me feel welcome at Käkkirurgen at the Uppsala University hospital.
- Ludovic Blache, Johan Kämpe, Andres Rodriguez-Lorenzo, Jan-Michael Hirsch, Stefan Johansson, Ewert Bengtsson, and others who have been involved in the haptics and surgery planning projects over the years.
- Daniel Buchbinder and his resident surgeons at the Mount Sinai hospital in New York, for the hospitality showed during the visits.
- Anders Hast, for your enthusiasm, and for good collaboration on the computer graphics and visualization courses.
- Elisabeth Wetzer, Raphaela Heil, Amit Suver, Nadezhda Koriakina, Leslie Solorzano, Kristina Lidarova, and Tomas Wilkinson, for making teaching the computer labs a lot more fun and a lot less stressful...
- Teo Asplund, Mikael Laaksoharju, Kalyan Ram, and Fei Liu, for interesting discussions on emulation, 3D printing, haptics, deep learning, gaming, and other fun topics.
- Hanqing Zhang, Amin Allalou, Eva Breznik, and Ashis Kumar Dhara, for being such nice and friendly office mates!

- Robin Strand and Gunilla Borgefors, for comments on the thesis drafts.
- Bo Nordin, for the nice lunch conversations.
- My parents Berit and Erik, for being supportive and understanding.

References

[1] BARBIČ, J., AND JAMES, D. L. Six-DoF Haptic Rendering of Contact between Geometrically Complex Reduced Deformable Models. *IEEE Transactions on Haptics 1*, 1 (2008), 39–52.

[2] BIMBER, O., AND RASKAR, R. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A K Peters, 2005.

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

[4] BOTSCH, M., HORNUNG, A., ZWICKER, M., AND KOBBELT, L. High-Quality Surface Splatting on Today's GPUs. In *Proceedings of the Second Eurographics / IEEE VGTC Conference on Point-Based Graphics* (2005), SPBG'05, pp. 17–24.

[5] BOURKE, P. Polygonising a scalar field. `http://paulbourke.net/geometry/polygonise`. Accessed on January 23, 2020.

[6] ÇIÇEK, Ö., ABDULKADIR, A., LIENKAMP, S. S., BROX, T., AND RONNEBERGER, O. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In *Int. conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2016), pp. 424–432.

[7] CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. Metro: Measuring Error on Simplified Surfaces. In *Computer Graphics Forum* (1998), vol. 17, pp. 167–174.

[8] CONTI, F., BARBAGLI, F., BALANIUK, R., HALG, M., LU, C., MORRIS, D., SENTIS, L., WARREN, J., KHATIB, O., AND SALISBURY, K. The CHAI Libraries. In *Proceedings of Eurohaptics 2003* (2003), pp. 496–500.

[9] CRASSIN, C., NEYRET, F., LEFEBVRE, S., AND EISEMANN, E. GigaVoxels: Ray-guided Streaming for Efficient and Detailed Voxel Rendering. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games* (2009), ACM I3D '09, pp. 15–22.

[10] DANIELSSON, P.-E. Euclidean Distance Mapping. *Computer Graphics and Image Processing 14*, 3 (1980), 227–248.

[11] ENGEL, K., HADWIGER, M., KNISS, J. M., REZK-SALAMA, C., AND WEISKOPF, D. *Real-Time Volume Graphics*. A K Peters, 2006.

[12] ERICSON, C. *Real-Time Collision Detection*. Morgan Kaufmann, 2005.

[13] FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. Distance Transforms of Sampled Functions. *Theory of Computing 8* (2012), 415–428.

[14] GREENE, N., KASS, M., AND MILLER, G. Hierarchical Z-Buffer Visibility. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (1993), ACM, pp. 231–238.

[15] GUSTAVSON, S., AND STRAND, R. Anti-aliased Euclidean Distance Transform. *Pattern Recognition Letters 32*, 2 (2011), 252–257.

[16] HAINICH, R. R., AND BIMBER, O. *Displays: Fundamentals and Applications*. CRC Press, 2011.

[17] JALBA, A. C., KUSTRA, J., AND TELEA, A. C. Surface and Curve Skeletonization of Large 3D Models on the GPU. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 6 (2012), 1495–1508.

[18] JU, T., LOSASSO, F., SCHAEFER, S., AND WARREN, J. Dual Contouring of Hermite Data. In *ACM transactions on graphics (TOG)* (2002), vol. 21, pp. 339–346.

[19] KAJIYA, J. T. The Rendering Equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques* (1986), ACM SIGGRAPH '86, pp. 143–150.

[20] KARIS, B. High Quality Temporal Anti-Aliasing. *Advances in Real-Time Rendering for Games, SIGGRAPH Courses* (2014).

[21] KOVLER, I., JOSKOWICZ, L., WEIL, Y., KHOURY, A., KRONMAN, A., MOSHEIFF, R., LIEBERGALL, M., AND SALAVARRIETA, J. Haptic computer-assisted patient-specific preoperative planning for orthopedic fractures surgery. *Int. Journal of Computer Assisted Radiology and Surgery 10*, 10 (2015), 1535–1546.

[22] KRUGER, J., AND WESTERMANN, R. Acceleration Techniques for GPU-based Volume Rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (2003), p. 38.

[23] LAINE, S., AND KARRAS, T. Efficient Sparse Voxel Octrees - Analysis, Extensions, and Implementation. Tech. rep., NVIDIA Research, 2010.

[24] LECUN, Y., BENGIO, Y., AND HINTON, G. Deep Learning. *Nature 521*, 7553 (2015), 436–444.

[25] LIN, M. C., AND OTADUY, M. *Haptic Rendering: Foundations, Algorithms, and Applications*. A K Peters, 2008.

[26] LIPTON, L. *StereoGraphics Developers' Handbook*. StereoGraphics Corporation, 1997.

[27] LIU, B., CLAPWORTHY, G. J., AND DONG, F. Fast Isosurface Rendering on a GPU by Cell Rasterization. *Computer Graphics Forum 28*, 8 (2009), 2151–2164.

[28] LORENSEN, W. E., AND CLINE, H. E. Marching Cubes: a High Resolution 3D Surface Construction Algorithm. *Computer Graphics 21*, 4 (1987), 163–169.

[29] LUNDIN PALMERIUS, K. *Direct Volume Haptics for Visualization (PhD Thesis)*. Linköping University, Norrköping, 2007.

[30] MCNEELY, W. A., PUTERBAUGH, K. D., AND TROY, J. J. Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (1999), ACM SIGGRAPH '99, pp. 401–408.

[31] MCNEELY, W. A., PUTERBAUGH, K. D., AND TROY, J. J. Voxel-Based 6-DoF Haptic Rendering Improvements. *Haptics-e, The electronic journal of haptics research* (2006).

[32] MUSETH, K. VDB: High-Resolution Sparse Volumes with Dynamic Topology. *ACM Transactions on Graphics (TOG) 32*, 3 (2013), 27.

[33] MYERS, K., AND BAVOIL, L. Stencil Routed A-buffer. In *SIGGRAPH Sketches* (2007), p. 21.

[34] NILSSON, J. *On Virtual Surgical Planning in Cranio-Maxillofacial Surgery (PhD Thesis)*. Uppsala University, 2019.

[35] Nysjö, J. *Interactive 3D Image Analysis for Cranio-Maxillofacial Surgery Planning and Orthopedic Applications (PhD Thesis)*. Uppsala University, 2016.
[36] Nysjö, J., Malmberg, F., Sintorn, I.-M., and Nyström, I. BoneSplit - A 3D Texture Painting Tool for Interactive Bone Separation in CT Images. *Journal of WSCG 23*, 2 (2015), 157–166.
[37] Nyström, I., Nysjö, J., and Malmberg, F. Visualization and Haptics for Interactive Medical Image Analysis: Image Segmentation in Cranio-Maxillofacial Surgery Planning. In *International Visual Informatics Conference* (2011), Springer, pp. 1–12.
[38] Olsson, P. *Haptics with Applications to Cranio-Maxillofacial Surgery Planning (PhD Thesis)*. Uppsala University, 2015.
[39] Olsson, P., Nysjö, F., Rodriguez-Lorenzo, A., Thor, A., Hirsch, J.-M., and Carlbom, I. B. Haptics-assisted Virtual Planning of Bone, Soft Tissue, and Vessels in Fibula Osteocutaneous Free Flaps. *Plastic and Reconstructive Surgery - Global Open 3*, 8 (2015).
[40] Olsson, P., Nysjö, F., Seipel, S., and Carlbom, I. B. Physically Co-Located Haptic Interaction with 3D Displays. In *Proceedings of Haptics Symposium 2012, Vancouver, Canada* (2012).
[41] Otaduy, M. A., and Lin, M. C. *High Fidelity Haptic Rendering*. Morgan & Claypool, 2006.
[42] Pfister, H., Zwicker, M., van Baar, J., and Gross, M. Surfels: Surface Elements as Rendering Primitives. In *Proceedings of the 27th Annual Conference on Computer graphics and Interactive techniques* (2000), ACM SIGGRAPH '00, pp. 335–342.
[43] Pharr, M., Jakob, W., and Humphreys, G. *Physically Based Rendering – From Theory to Implementation, 3rd ed.* Morgan Kaufmann, 2016. Online edition available at `https://pbrt.org` (Accessed on January 23, 2020).
[44] Pharr, M., and Mark, W. R. ispc: A SPMD Compiler for High-Performance CPU Programming. In *2012 Innovative Parallel Computing (InPar)* (2012), IEEE, pp. 1–13.
[45] Preim, B., and Bartz, D. *Visualization in Medicine: Theory, Algorithms, and Applications, 1st ed.* Morgan Kaufmann, 2007.
[46] Ritacco, L. E., Milano, F. E., and Chao, E. *Computer-Assisted Musculoskeletal Surgery: Thinking and Executing in 3D*. Springer International Publishing, 2016.
[47] Ruffaldi, E., Morris, D., Barbagli, F., Salisbury, K., and Bergamasco, M. Voxel-Based Haptic Rendering using Implicit Sphere Trees. In *2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* (2008), IEEE, pp. 319–325.
[48] Ruspini, D. C., Kolarov, K., and Khatib, O. The Haptic Display of Complex Graphical Environments. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (1997), ACM SIGGRAPH '97, pp. 345–352.
[49] Sonka, M., Hlavac, V., and Boyle, R. *Image Processing, Analysis, and Machine Vision, 3rd ed.* Thomson, 2008.
[50] Tanner, C. C., Migdal, C. J., and Jones, M. T. The Clipmap: A Virtual Mipmap. In *Proceedings of the 25th Annual Conference on Computer Graphics*

and Interactive Techniques (1998), ACM SIGGRAPH '98, pp. 151–158.
[51] TELEA, A. C. *Data Visualization: Principles and Practices, 2nd ed.* A K Peters, 2014.
[52] UDUPA, J. K., AND ODHNER, D. Shell Rendering. *IEEE Computer Graphics and Applications 13*, 6 (1993), 58–67.
[53] VIDHOLM, E. *Visualization and Haptics for Interactive Medical Image Analysis (PhD Thesis).* Uppsala University, 2008.
[54] WALTER, B., DRETTAKIS, G., AND PARKER, S. Interactive Rendering Using the Render Cache. In *Proceedings of the 10th Eurographics Conference on Rendering* (1999), EGWR'99, pp. 19–30.
[55] WAN, M., AND MCNEELY, W. A. Quasi-Static Approximation for 6 Degrees-of-Freedom Haptic Rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (2003), pp. 257–262.
[56] WARE, C., ARTHUR, K., AND BOOTH, K. S. Fish Tank Virtual Reality. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (1993), ACM, pp. 37–42.
[57] WESTOVER, L. A. *Splatting: A Parallel, Feed-Forward Volume Rendering Algorithm (PhD Thesis).* University of North Carolina, 1991.
[58] ZILLES, C. B., AND SALISBURY, J. K. A Constraint-Based God-Object Method for Haptic Display. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems* (1995), vol. 3, pp. 146–151.
[59] ZWICKER, M., PFISTER, H., VAN BAAR, J., AND GROSS, M. Surface Splatting. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), ACM SIGGRAPH '01, pp. 371–378.

Acta Universitatis Upsaliensis

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 1898

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-403104

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2020