

Application of Smart Visibility on Medical 3D Ultrasound Datasets

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Johannes Novotny

Matrikelnummer 0325101

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Dipl.-Ing. Dr.techn. Andrej Varchola

Wien, 23.05.2013

(Unterschrift Verfasser)

(Unterschrift Betreuung)

Application of Smart Visibility on Medical 3D Ultrasound Datasets

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Johannes Novotny

Registration Number 0325101

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Assistance: Dipl.-Ing. Dr.techn. Andrej Varchola

Vienna, 23.05.2013

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Johannes Novotny
Ganglbauergasse 21-23/2/14, 1160 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

23.05.2013

(Ort, Datum)



(Unterschrift Verfasser)

Acknowledgements

This work would not have been possible without the help of many friends and colleagues. I would like to express my gratitude to my first supervisor Meister Eduard Gröller for the useful comments, remarks and engagement through the course of this master thesis. Furthermore I would like to thank my second supervisor Andej Varchola for introducing me to the topic as well for the discussions and support on the way. Also, I would like to thank our collaborators at GE Healthcare for the provided resources and technical insight. In particular, Gerald Schröcker and Daniel Buckton helped to shape this work through their extensive domain knowledge. My appreciation goes to my colleagues at the Vienna University of Technology for their help through suggestions, discussions and proofreading. Finally, I would like to thank my wife Eun Young Lee for supporting me throughout the entire process and for motivating me whenever needed.

Abstract

Ultrasonography is one of the standard medical imaging techniques used in pregnancy examinations. It is widely available due to its low cost, portability and simple diagnostic procedure. In recent years, three-dimensional ultrasound (US) imaging has been gaining popularity in the area of prenatal care. It provides examiners with a coherent visualization of the fetus. However, within scanned 3D datasets, the fetus is often surrounded by occluding structures which reduce its visibility in the resulting visualizations. Current ultrasound machines provide several methods to remove these occluders from visualizations. These methods are often difficult to control in the real-time setting of an US examination session. As a result, the work flow of US examinations has to be interrupted in order to apply occlusion removal to selected scans.

In an attempt to reduce the required user interaction, this thesis evaluates a recently developed occlusion removal technique. The smart visibility method for prenatal ultrasound analyzes the ray profiles during an execution of a ray casting algorithm. This analysis identifies anatomic features within a dataset. From these features, a clipping surface that separates the fetus from its occluders is calculated. The clipping surface is used to remove occlusions from the visualization. The use of cutaway and ghosting visualization techniques allow an unoccluded view of the fetus, while retaining its general context within the volume.

The clipping surface calculated by the smart visibility method is obtained by using a surface reconstruction algorithm. Within this thesis, different surface reconstruction techniques were evaluated for their occlusion removal quality and performance. It was possible to show that one of the evaluated approaches provides good results in a majority of test cases. This approach also reduces the required user interaction to a single parameter. To verify these results, the approach has been tested within a state-of-the-art US firmware. By manipulating the adjustable parameter through the hardware controls of an US machine, the smart visibility method can be used in real-time scenarios. GE Healthcare is evaluating the smart visibility method and considers to include it in upcoming versions of their US machine firmwares.

Kurzfassung

Die Ultraschalluntersuchung (Sonographie) ist ein bildgebendes Verfahren, das sich in der Pränataldiagnostik als Standard durchgesetzt hat. Das Verfahren ist weit verbreitet und besticht durch niedrige Betriebskosten, hohe Portabilität und einen verhältnismäßig einfachen Untersuchungsablauf. In den letzten Jahren haben 3D Ultraschallgeräte an Bedeutung gewonnen, da sie, im Gegensatz zu 2D Ultraschallgeräten, eine vollständige Visualisierung des Fetus ermöglichen. Innerhalb der aufgenommenen 3D Datensätze ist der Fetus allerdings oft von Strukturen umgeben, die auf den endgültigen Visualisierungen zu Verdeckungen führen. Aktuelle Ultraschallgeräte verfügen zwar über Methoden zur Entfernung von Verdeckungen, diese sind jedoch oft schwierig zu steuern und werden daher selten direkt während der Untersuchung eingesetzt. Der Untersuchungsablauf wird stattdessen kurz unterbrochen, um Verdeckungen in einzelnen Scans zu entfernen.

Diese Diplomarbeit analysiert ein kürzlich entwickeltes Verfahren zur Entfernung von Verdeckungen, mit dem Ziel benötigte Benutzerinteraktion zu reduzieren. Die *“smart visibility method”*, für pränatale Sonographie basiert auf einer Analyse der Opazitätsprofile von Sehstrahlen während der Ausführung eines Raycasting-Algorithmus. Die Analyse identifiziert anatomische Merkmale in einem 3D Datensatz, die dazu benutzt werden, eine Clipping-Fläche zwischen dem Fötus und den Verdeckungen zu berechnen. Die Clipping-Fläche wird dann in Kombination mit Cutaway- und Ghosting-Techniken dazu benutzt, Verdeckungen aus der Visualisierung zu entfernen, ohne dabei den Kontext zum umgebenden Volumen zu verlieren.

Die *“smart visibility method”*, berechnet die Clipping-Fläche mit Hilfe eines Oberflächenrekonstruktionsalgorithmus. Im Rahmen dieser Diplomarbeit wurden verschiedene Methoden zur Oberflächenrekonstruktion getestet und die Qualität der Resultate ausgewertet. Es konnte gezeigt werden, dass einer der getesteten Ansätze in einem Großteil der Testfälle gute Resultate liefert. Außerdem reduziert dieser Ansatz die benötigten Benutzereingaben auf einen einzigen Parameter. Um die Ergebnisse zu verifizieren, wurde die *“smart visibility method”*, in der Firmware eines aktuellen Ultraschallgeräts getestet. Durch die Steuerung des Parameters mittels eines Hardwarereglers am Ultraschallgerät kann die Methode direkt während der Ultraschalluntersuchung angewendet werden. Die *“smart visibility method”*, wird zur Zeit von GE Healthcare überprüft. Eine Integration der Technik in zukünftige Versionen der GE US Firmware wird in Erwägung gezogen.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Aim of the Work	2
1.3	Smart Visibility Approach	6
2	Related Work	11
2.1	3D Ultrasound	11
2.2	3D Data Acquisition	13
2.3	Volume Rendering	16
2.4	Smart Visibility Techniques	21
2.5	Surface Reconstruction Methods	23
3	Smart Visibility Methods	25
3.1	Characteristics of Ultrasound Data	25
3.2	Initial Point Detection	32
3.3	Surface Reconstruction	36
3.4	Rendering	43
4	Framework and Method Implementation	47
4.1	Framework Overview	47
4.2	Test Datasets	50
4.3	Modules	50
5	Method Evaluation	63
5.1	Test case preparation	63
5.2	Module Evaluation	64
5.3	Test Case Results	79
5.4	Simu3D Integration	92
6	Conclusion and Future Work	95
	Bibliography	97

Introduction

Medical sonography is a common diagnostic method in modern medicine for visualizing internal body structures by measuring the echoes of ultrasound pulses. It is widely available due to its portability, simple diagnostic procedure and relative inexpensiveness compared to other medical imaging techniques, like MRI and CT. In the area of prenatal care, obstetric sonography is one of the standard diagnostic methods. This thesis will focus on improving the results of fetal imaging during pregnancy examinations.

1.1 Motivation

Classic diagnostic sonography provides the examiner with a real-time 2D visualization of the plane scanned by the ultrasound probe (transducer). Only a single slice of the fetus can be shown at a time. It is up to the examiner to maneuver the transducer into a suitable scanning position to perform medical examinations, e.g., bone length measurements. The actively moving fetus increases the difficulty of capturing a useful ultrasound image even further. More recent transducers are capable of examining a 3D-volume in a single scan. A visualization of 3D-scan data allows the examiner to get a more coherent view of the fetus and helps to spot medically relevant deformations. However, the projection of 3D-data to a 2D-image introduces the problem of occlusion. During an examination the view on the fetus might be obstructed by maternal tissue, the umbilical cord, particles within the amniotic fluid or general ultrasound artifacts. 3D ultrasound scanners usually provide several methods to remove these occluders. The most basic one is to define a box around the region of interest (ROI). During rendering all data points outside of the ROI box are omitted. An extension to this approach is often available. By allowing the deformation of one or more faces of the box, as shown in Figure 1.1, the results can be refined. More complex methods allow the exclusion of arbitrary regions of the dataset. However, all of these methods share the problem of requiring an increasing amount of user interaction. They allow the processing of a single scan to achieve the best possible visibility, but the process takes time and requires the attention of the examiner. Since one of the examiner's hands is holding

the transducer, her or his ability to adjust numerous parameters at the ultrasound (US) machine is limited (as seen in Figure 1.2b). Additionally, the manual adjustments are in general not valid for subsequent 3D-scans. The interruption of the examination workflow in order to get an unoccluded view of the fetus is one of the issues of 3D sonography that require improvement.

Apart from the diagnostic use of 3D sonography there is also a commercial aspect to be considered. It is possible for obstetricians to offer images, video sequences or 3D models of the examination results to their patients as a keepsake. Parents are interested in an unoccluded view of their child, especially of its face. In this case great care has to be taken to avoid any accidental removal of fetal tissue. Incomplete visualization of the fetus might have a detrimental effect on parents.

1.2 Aim of the Work

Based on the problems and requirements mentioned before, this thesis accomplishes two goals: The implementation of a framework for testing and comparing occlusion removal methods and the use of this framework to evaluate a set of methods developed specifically for 3D obstetric ultrasound datasets. In order to minimize the effort of implementing and evaluating different methods, the framework fulfills the following requirements:

- Modularity

By providing a general module interface for data processing and rendering, the framework simplifies the development of occlusion removal methods that can be separated into distinct processing steps. The modular structure allows an effortless exchange of alternative implementations of the individual processing stages. The occlusion removal methods presented in Section 1.3 make extensive use of this feature.

- Easy user interface setup

Each module implementation might depend on user-controllable parameters. The framework provides a set of simple GUI-elements for parameter manipulation. These elements can be used by the modules to allow adjustment of their parameters during the test execution.

- Support functionality for evaluation

In order to verify results, the framework provides specific support for the evaluation of the following quality criteria:

- Occlusion removal quality

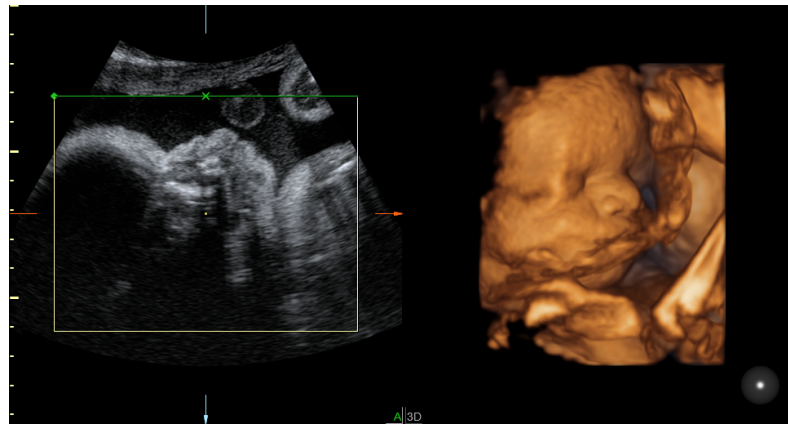
The effectiveness of occlusion removal methods can be measured by comparing their results to ground-truth datasets. To allow a quick visual verification, the framework provides the side by side rendering of test case results. The occlusion removal results are shown alongside with visualizations of the original and ground-truth volumes. Error measurements between result and ground truth can be viewed directly within the framework or exported for in-depth evaluation.

- Real-time performance

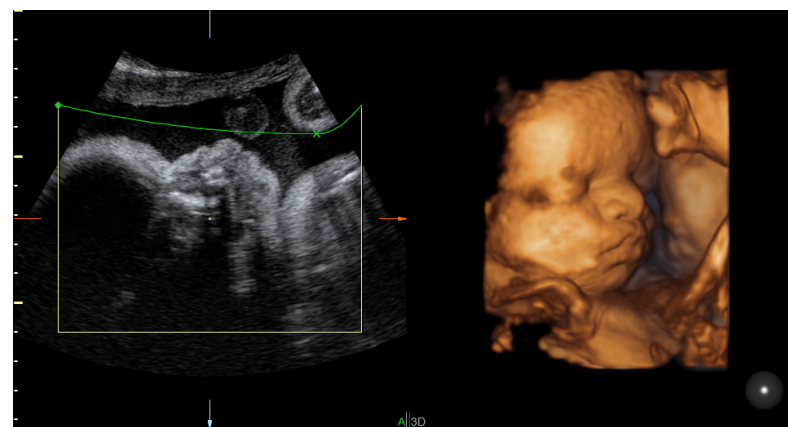
Current 3D ultrasound hardware can acquire and render about 20 scans per second. This frame rate should not be notably reduced by the use of occlusion removal methods. The framework helps with the performance evaluation by providing time measurements at the module level.

- Simplified transition to ultrasound machine firmware

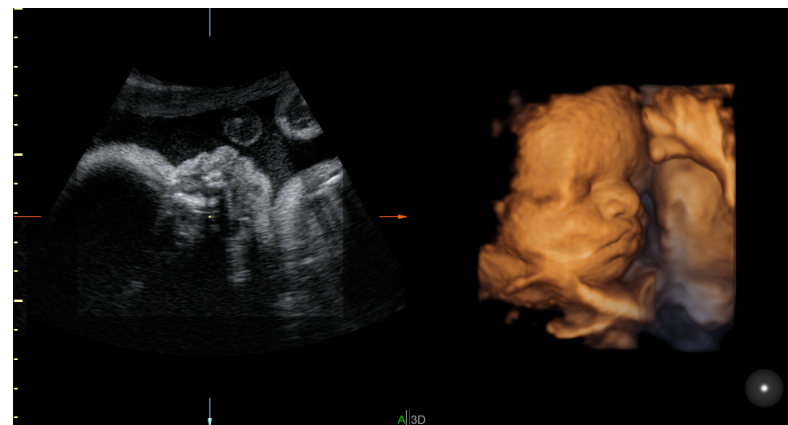
The integration of the tested occlusion removal functionalities to state-of-the-art ultrasound machine firmware is one of the goals of this project. The modular structure of the framework supports the transition by allowing the implementation of an occlusion removal algorithm to be separated from loading and rendering functions.



(a) ROI box



(b) ROI box with deformable face



(c) Removal of arbitrary volume regions

Figure 1.1: Examples of manual occlusion removal methods applied on a 3D-ultrasound scan showing the face of a fetus. A horizontal slice through the scan is shown on the left, the resulting visualization on the right: (a) Scaled region of interest (ROI) box. (b) Scaled ROI box with deformed front face. (c) Manual removal of arbitrary volume regions using a volume cutting tool.



(a) Ultrasound user interface



(b) Ultrasound examination setup

Figure 1.2: Obstetric sonography examination: (a) User interface of a state-of-the-art ultrasound (US) workstation [1]. (b) Example of an examination routine showing the occupation of an examiner's hands [2].

1.3 Smart Visibility Approach

The occlusion removal methods implemented and evaluated in this thesis were developed as part of an ongoing research collaboration between the Vienna University of Technology and GE Healthcare [3]. All presented methods follow a basic concept, similar to the already existing method of using a deformable ROI-box face. Instead of requiring the user to define the curvature of the face manually, a clipping surface is calculated by the occlusion removal algorithm. The shape of the surface is based on anatomical features found within the dataset. The resulting surface separates the region of interest from view-obstructing tissue and can be used to generate cut-away and ghosting effects within the visualization. Based on the definition given by Viola and Gröller [4] the presented methods can be classified as smart visibility techniques. For the remainder of this work these methods will therefore be referred to as *smart visibility methods*.

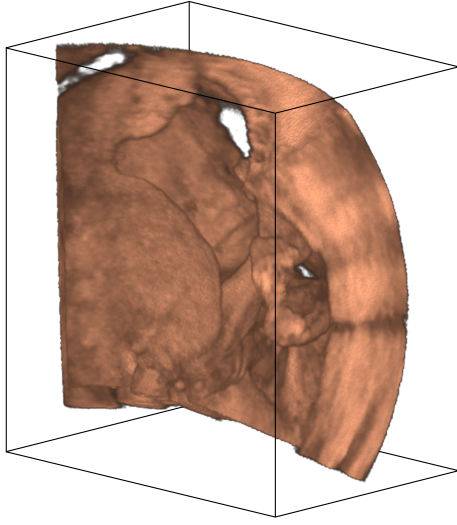
The core concept used by all tested smart visibility methods consists of three consecutive processing steps:

- **Initial point selection**
By analyzing the features of the input dataset, this processing step selects a set of initial points. These points have a high probability of being positioned between the fetus and its closest occluders towards the camera position. By using a ray casting algorithm, for each ray a single initial point is selected if one or more are available. Otherwise the corresponding pixel is marked as undefined. This results in a sparse depthmap storing the distances between initial points and camera position.
- **Surface reconstruction**
The previously produced depthmap is processed by a surface reconstruction technique. Inter- and extrapolation of the sparse set of initial points results in the desired clipping surface. The tested smart visibility methods differ mainly in the surface reconstruction algorithm utilized in this processing step.
- **Rendering**
The rendering step integrates the clipping surface into the visualization using a direct volume rendering (DVR) algorithm. Using the clipping surface as starting point for all rays removes occlusions. However, incorrect placement of the clipping surface is possible. Regions in front of the clipping surface can therefore be rendered in a highly transparent way using ghosting techniques to mitigate incorrect surface placement. Figure 1.3 shows visualizations of different steps of the algorithm.

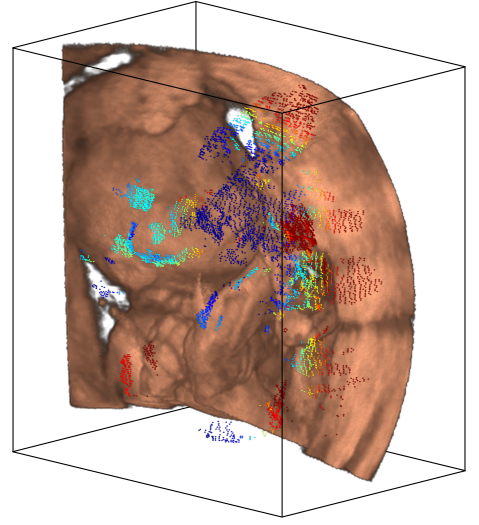
The approaches were tested on nine 3D ultrasound test cases of independent obstetric examinations, provided by GE Healthcare. Each test case contained the original scan dataset and a reference dataset optimized by expert users, using the currently available occlusion removal methods. In order to verify the results, manual segmentation was used to create ground-truth datasets for all provided test cases. Figure 1.4 shows visualizations of an original, a ground-truth and a reference dataset side by side to the result of our smart visibility method. Our method reduces the amount of required user interaction to a single parameter. The results are similar to visualizations of the reference datasets.

The tested smart visibility approaches were evaluated together with experts from GE Healthcare. The most promising approach has been ported into a plugin of the Simu3D US machine firmware for detailed analysis. In order to reduce the computation time, all parts of the selected algorithm were implemented to utilize CUDA GPGPU kernels.

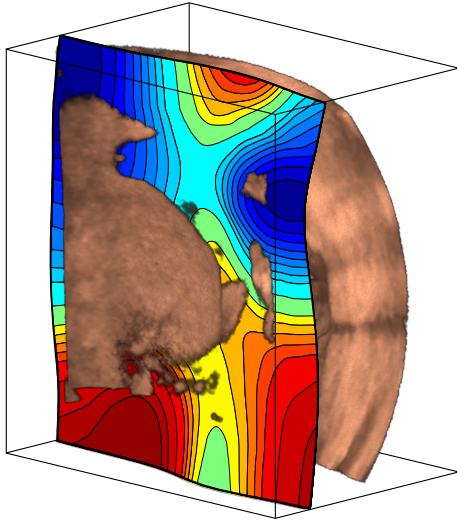
The remainder of this work is structured as follows: Chapter 2 covers the basic principles of 3D ultrasound measurements and gives an overview of state-of-the-art direct volume rendering and smart visibility techniques. Chapter 3 explains the concepts used by the tested smart visibility methods in detail, while Chapter 4 elaborates their implementation specifics. Chapter 5 presents and discusses the results of the tested methods and Chapter 6 finishes with the conclusion and possible future enhancements.



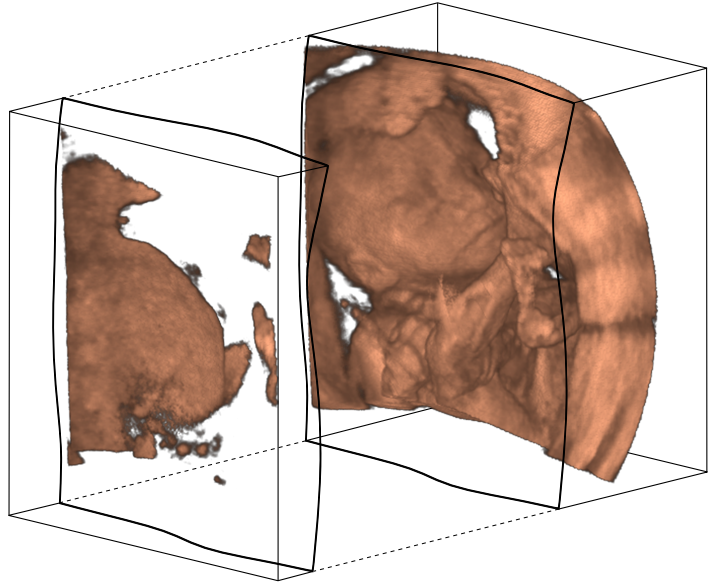
(a) Entire Dataset



(b) Locations of initial points

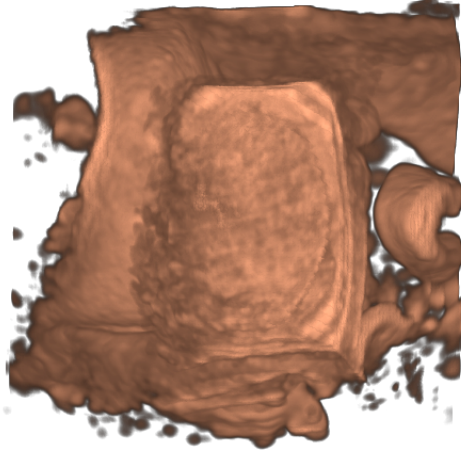


(c) Reconstructed clipping surface

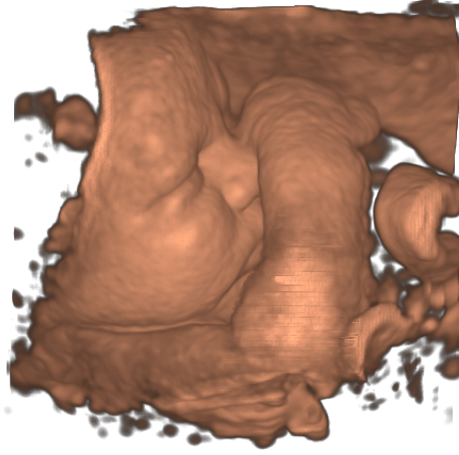


(d) Smart visibility

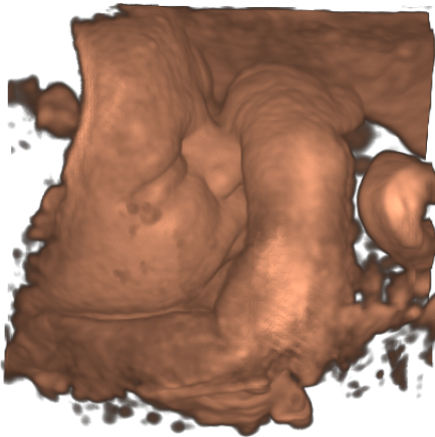
Figure 1.3: Visualizations of various steps of the smart visibility method in test case "M01": (a) Occluded fetus within the entire dataset. (b) Location of the initial points selected by the smart visibility algorithm in the dataset. (c) The reconstructed clipping surface separating fetus and occluder. (d) Unoccluded view on the fetus by utilizing the clipping surface during rendering.



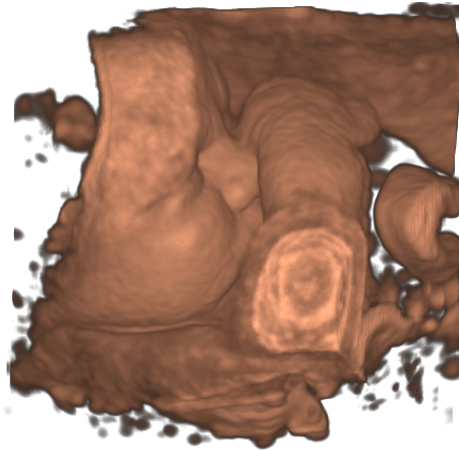
(a) Entire dataset



(b) Ground-truth dataset



(c) Reference dataset



(d) Smart visibility approach

Figure 1.4: Visualizations of the available input datasets and smart visibility results of test case "E01". The datasets contain the head of a fetus with one arm in front of it. (a) Occluded fetus within the entire dataset. (b) Manually created ground-truth dataset. (c) Reference dataset provided by GE Healthcare. (d) Result of our smart visibility method.

Related Work

This chapter gives a brief overview of current 3D ultrasound data acquisition methods, to help understand the characteristics of datasets produced by them. Additionally, it contains a summary of state-of-the-art direct volume rendering, smart visibility and surface reconstruction techniques.

2.1 3D Ultrasound

Basic principle

The basic principle of 3D ultrasound examinations can be shown on the example of 2D sonography. 3D sonography uses the same fundamentals of sending ultrasound pulses into the tissue and analyzing the received echoes. In general, three steps are necessary to obtain a sonographic image [5]:

- Pulse transmission

Ultrasound pulses for medical applications are short sound pulses with frequencies ranging from 1 to 20 MHz. They are usually emitted by piezoelectric elements within an ultrasound probe (transducer) and transmitted into the examined tissue by physical contact [5]. Water-based gels are used to prevent air gaps between transducer and the patient's skin. Within an ultrasound probe, the transducer elements can be arranged in different array layouts. The common sonographic B-scan (brightness) imaging mode requires the transducer elements to be arranged in a linear array, emitting pulses subsequently in order to scan a single plane through the examined body [6].

It is possible to enhance the scan results of specific areas by focusing the ultrasound pulses to a specific depth using physical lenses or electric beamforming. The more recent beamforming uses phased transducer elements to shape the ultrasound pulse by using constructive interference [6]. Additionally, the quality of a scan can be influenced by the pulse

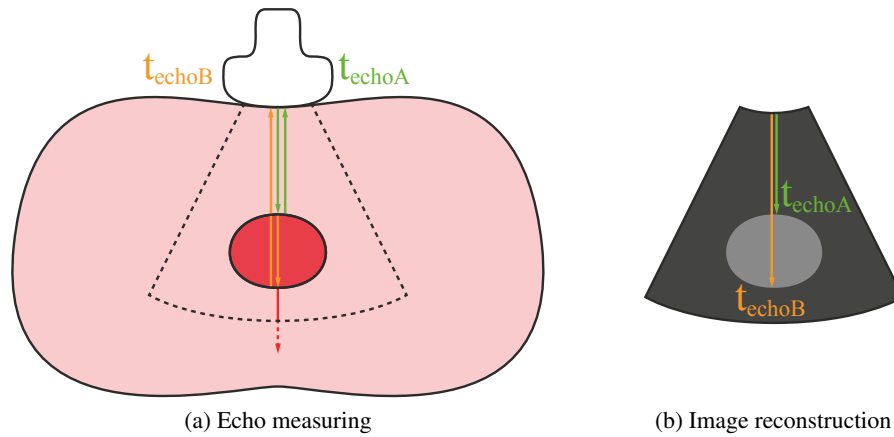


Figure 2.1: Basic principle of ultrasound examinations explained on a single pulse (a) An ultrasound pulse generated by the transducer is reflected at two tissue borders. The traversal-times (t_{echoA} , t_{echoB}) and intensities of these echoes are recorded. (b) With the recorded data it is possible to reconstruct the depth of encountered tissue changes on the corresponding line of a 2D-slice [6].

frequency. A higher frequency allows a more detailed scan, but reduces the penetration depth in the examination [5].

- Echo measuring

As the ultrasound pulses are traversing through tissue, changes in the tissue density causes them to be partially reflected back towards the transducer as echoes. A high density difference causes a stronger echo and attenuates the pulses, which limits the penetration depth. The echoes are received by a sensor array within the transducer, measuring echo direction, strength and traversal time [6].

- Image reconstruction

The recorded data allows a visualization of the inner body structures of a patient, by calculating the depth of the received echoes from their measured traversal times. This calculation uses general assumptions on the speed of sound in human tissue. The intensity of a measured echo determines the grey-scale value at the echo location. A strong echo results in a bright representation [6]. Slice visualizations can be obtained by drawing the calculated depth values on a 2D plane according to their pulse directions. In the mentioned B-mode scan, this slice often has the shape of a rectangle or ring segment fanning out from the transducer. Figure 2.1 illustrates these three steps on the example of a single ultrasound pulse.

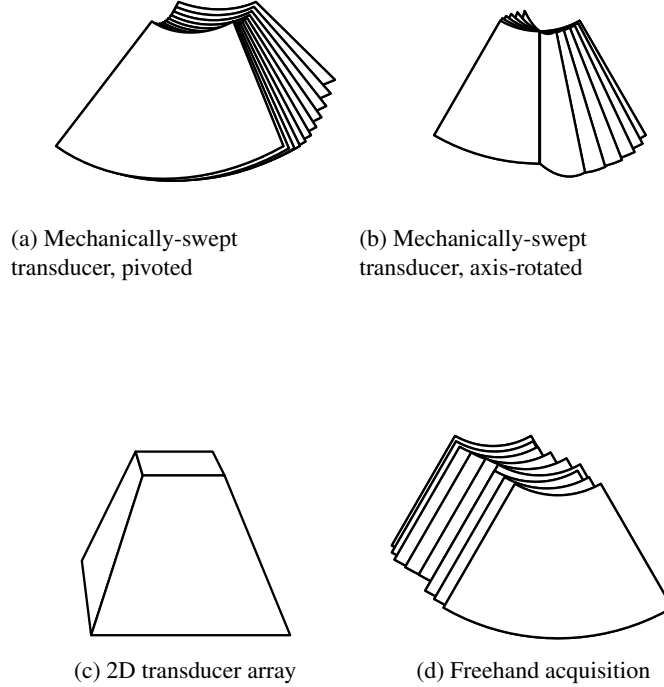


Figure 2.2: Shape of the volume data acquired by different scanning methods [8]

2.2 3D Data Acquisition

2D B-mode sonography is a tomographic imaging technique used to display structures in a thin slice of the patient's body. The high acquisition rate of this mode allows the reconstruction of three-dimensional volumetric data from consecutive slice scans. In practice there are several different acquisition methods used to obtain 3D datasets [7] [8]. They can be classified into three main groups:

- Acquisition by mechanically-swept transducers

The most common method to obtain 3D data in commercial ultrasound scanners is to mechanically sweep the probe over the region to be examined along a known trajectory. Volumetric datasets can then be reconstructed from a set of single scans and their known relative orientations. The sweeping mechanism is usually built into the transducer. Instead of being fixed directly to the case of the probe, the transducer array is mounted to a pivotable or axis-rotatable yoke controlled by a high precision motor. Figures 2.2a and 2.2b illustrate volume data obtained by this kind of data acquisition. In this example, method 2.2b provides a higher volume resolution around the central rotation axis.

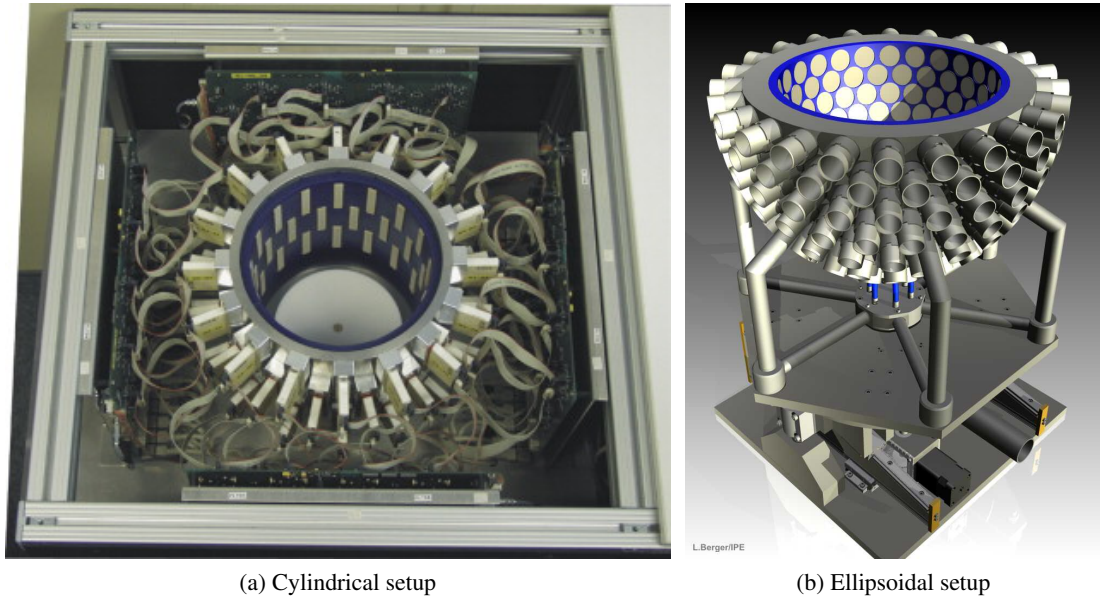


Figure 2.3: Two experimental 3D ultrasound computer tomography setups used for breast-cancer examinations: (a) Cylindrical configuration with 48 transducer arrays (32 receivers and 8 emitters each) [10], (b) Rotation symmetrical ellipsoid configuration with 157 cylindrical transducer arrays (9 receivers and 4 emitter each) [12].

- Acquisition by 2D transducer arrays

A more complex approach to 3D data acquisition is to use one [9] or more [10] two-dimensional transducer arrays. By performing 2D scans using subsequent lines of the transducer array, it is possible to acquire results similar to the ones of a mechanically swept transducer. Since there are no moving components a high sampling rate can be achieved. Currently the technical limitation of this method lies in the resolution of the transducer matrix. An array of dimension 128×128 has already been realized [11]. Scans of these probe-types result in frustum-shaped volumetric datasets as shown in Figure 2.2c.

Strong density differences cause a strong attenuation of ultrasound pulses penetrating deeper tissue regions, resulting in a loss of contrast in these areas of the scan. These shadowing artifacts can be prevented by employing a large number of 2D arrays in a fixed setup around the scanned tissue [10]. This technique is known as 3D ultrasound tomography. The fixed configuration of sensors often prevents direct contact between tissue and transducer. Therefore, water is used as mediator to limit losses of the ultrasound pulse strength. While the penetration depth of ultrasound scans is too low to be used at arbitrary body sections, smaller body parts can be examined entirely. US breast examinations (mammasonography) are a common application of this technique. Figure 2.3 shows two example setups of these scanner types, used for mammasonographic examinations.

- Freehand acquisition

The basic principle of this method is to record the spacial position and rotation of the transducer during each scan of a standard B-mode examination. The recorded information is used to reconstruct a 3D-dataset of the swept volume [13]. The scanned volume can be arbitrarily shaped, as shown in Figure 2.2d, which allows the acquisition of volume data covering the whole body. Since the probe has to be swept manually, this acquisition method can not produce real-time 3D-data. It also performs poorly on patients moving during an examination sweep. There are several ways to track the position and orientation of conventional transducers with the required six degrees-of-freedom.

- Magnetic position sensors

In a time-varying 3D magnetic field it is possible to compute the position and orientation of a transducer. Three small coils are attached to it in orthogonal directions. Position and orientation can be calculated from the current induced in each coil [14] [15] [16]. Magnetic positioning systems do not require a direct line of sight between the field generators and the sensor coils. Their main disadvantage is that metal parts, especially those made of ferromagnetic metals, are deteriorating the accuracy of the reading. Since metal beds are very common in clinical environments Sheehan et.al. [17] proposed a system that places the field generators between patient and bed in order to mitigate accuracy losses.

- Optical position sensors

Optical tracking systems consist of two or more calibrated cameras that are tracking passive or active targets attached to the transducer [18]. A passive target is composed of three or more retro reflective spheres arranged asymmetrically with known relative positions to each other. By using timed infrared light pulses the cameras can determine the centers of the individual spheres within the captured images. Their positions are then triangulated in 3D space and used to determine the transducer orientation.

In active targets, the spheres are replaced by infrared LEDs, each emitting light pulses in known and pairwise different sequences. This removes the necessity of infrared flashlights mounted on the cameras and increases the accuracy of the measured orientation, since the tracked LEDs can be uniquely identified by their pulse sequence.

Optical position sensors produce very accurate position measurements [19]. However, their major drawback is the dependency on an uninterrupted line of sight between cameras and targets. This problem can be diminished by using more than the minimum number of two cameras, but that in turn increases the price and reduces the portability of the tracking system.

- Inertial position sensors

Inertial sensors are a combination of motion-sensing devices attached to the transducer. They usually consist of three orthogonally positioned accelerometers and gyroscopes. These systems do not measure the absolute position of the ultrasound

probe, instead they determine the relative change of position and orientation between two measurements. This has the advantage of not requiring any external tracking equipment apart from the sensors mounted on the transducer. Errors of the measurements are accumulating over time (drift), which effectively limits the maximal examination duration. The translation measurements are more sensitive to drift, therefore Poulsen et al. [20] and Goldsmith et al. [21] proposed the use of additional sensors to refine the measured position changes.

- Sensorless acquisition

Housden et al. [22] have shown that under special circumstances it is possible to determine the relative position and orientation change between two scans using purely image-based techniques. This method requires the transducer to be swept slowly and without any drastic changes in position or orientation to produce a dense stack of scans. By measuring the decorrelation of speckles on subsequent ultrasound images their relative positions to each other can be estimated. Subsequently a 3D-dataset can be reconstructed. Due to the accumulating error this technique cannot be used for large scale measurements.

None of the presented methods directly produce volume data aligned to a Cartesian grid. In order to simplify further processing, the acquired data is often resampled to a Cartesian representation using scan conversion. The datasets used in this thesis have been provided by GE Healthcare in scan converted form. They were produced by a mechanically swept transducer with a pivotable yoke (Figure 2.2a).

2.3 Volume Rendering

The field of volume rendering has been very actively researched in the last two decades. This resulted in several different methods to create two-dimensional visualizations of three-dimensional scalar fields (e.g., isosurface rendering via marching cubes). In the medical field, this kind of 3D data is provided by CT, MRI, 3D-US and other imaging modalities. Direct Volume Rendering (DVR) is a commonly used technique that renders 3D-datasets without calculating an intermediate geometric representation of the volume. The development of programmable shaders and GPGPU programming made it possible to use DVR at interactive frame rates on standard computer hardware. Since this work is utilizing DVR for visualization, this chapter will be focusing solely on direct volume rendering methods.

Optical Model

DVR is based on a simplified model of light passing through a volume filled with particles. As the traversing light is interacting with particles, various physical effects (e.g., absorption, emission, scattering) affect its color and intensity. Eventually a portion of light is reflected towards the view position. Each sample taken within the 3D-dataset is treated as a small volume filled with particles. The particle density and their optical properties are usually depending on the intensity value of the sample. A common formalization of the volume rendering integral is given by Max [23]:

$$I_{\lambda}(x, r) = \int_0^L C_{\lambda}(s) \mu(s) \underbrace{e^{(-\int_0^s \mu(t) dt)}_{t(s)}}_{t(s)} ds \quad (2.1)$$

The result I_{λ} is the light intensity of wavelength λ coming from the direction of ray r that is received at location x on the image plane. L is the length of the ray r . μ is the density of volume particles that receive light from the light sources and reflect it according to their optical properties. C_{λ} is the light of wavelength λ emitted at location s in the direction of r . t is the transparency accumulated along the ray at position s starting from the image plane.

Solving Equation 2.1 analytically is often not possible and infeasible. In practical applications it is therefore discretized into sequential intervals i of size Δs . This leads to the general compositing equation [23]:

$$I_{\lambda}(x, r) = \sum_{i=0}^{L/\Delta s} \underbrace{C_{\lambda}(i\Delta s) \mu(i\Delta s) \Delta s}_{g_i} \cdot \prod_{j=i+1}^{L/\Delta s} \underbrace{e^{-\mu(j\Delta s)\Delta s}}_{t_i} \quad (2.2)$$

Where g_i is the source term defining the color and t_i is the transparency term of an interval Δs at sampling location i . As a further simplification both values can be derived from predefined transfer functions defined over all possible sample intensity values. Equation 2.2 is commonly referred to as discrete volume rendering integral (DVRI) [24] and is used as a basic optical model for many DVR-algorithms. It covers the optical effects of emission and absorption.

While there are many approaches to extend this model in order to gain more physically plausible results there are also visualizations models based on less intricate assumptions. Examples are:

- First hit projection
Only the first sample of a ray with a density above a preset threshold is used in the rendering process.
- Average projection
All samples along a ray are averaged resulting in a single density value. This is also known as x-ray simulation.
- Maximum Intensity Projection (MIP)
Only the highest density value of a ray is used in the rendering process.

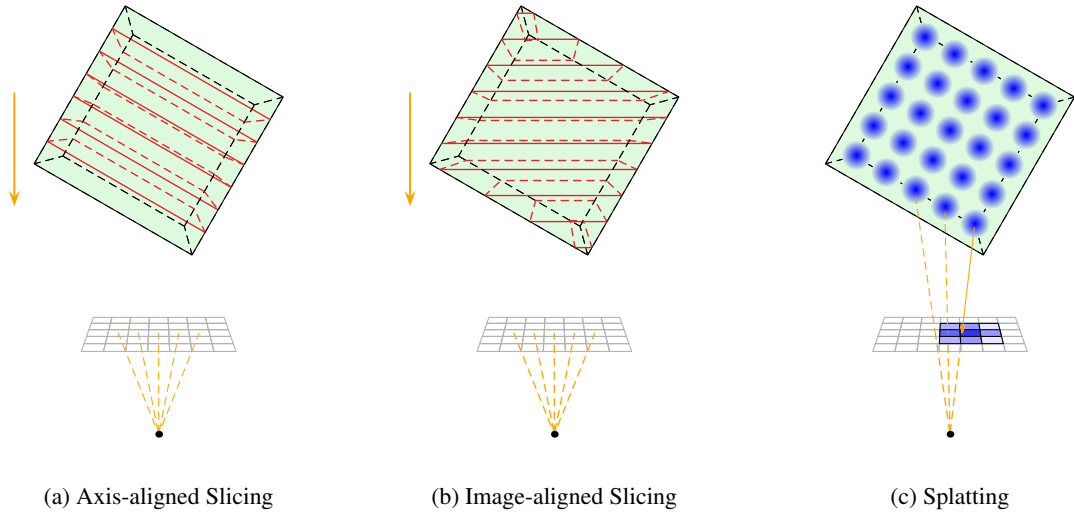


Figure 2.4: Object-order direct volume rendering methods: (a) DVR by back-to-front blending of axis-aligned slices. (b) DVR by back-to-front blending of image-aligned slices. (c) DVR by projecting a reconstruction kernel onto the image plane for each voxel.

Direct Volume Rendering

A variety of different approaches to compute the DVRI have been developed. They can be categorized into two main groups: image-order and object-order algorithms. In an image-order approach the integral is evaluated for every pixel of the output image individually. Object-order methods iterate over geometric primitives in the scene. Example techniques are:

- Slicing

Slicing is an object-order rendering algorithm that uses 2D texture mapping and blending to visualize a volume [25]. As first step, the side of the volume dataset most parallel to the image plane is determined. A stack of 2D primitives (quads) parallel to the chosen side is placed within the extent of the volume dataset. Each primitive is textured according to its slice position. By rendering this stack in back-to-front order using alpha-blending, a 2D volume representation can be obtained (Figure 2.4a). The quality of the result is dependent on the viewing position, since axis-aligned slices may cause artifacts whenever they are not parallel to the image plane.

Utilizing graphics hardware with 3D-texturing support, Rezk-Salama et al. [26] improved the algorithm by always aligning the slice stack to be parallel to the image plane (Figure 2.4b). While this diminishes some of the artifacts, the basic concept of the slicing approach makes it difficult to implement optimizations, e.g., opacity-based early ray termination.

- Splatting

Splatting is an object-order method proposed by Westover [27]. It visualizes a volume by projecting reconstruction kernels onto the image plane for each voxel of the dataset. These kernels, also known as footprints or splats, are commonly Gaussian kernels. This allows the precomputation of their projected 2D representation due to radial symmetry. Size, color and opacity of splats are in general depending on the density of their corresponding voxels. In case of perspective projection they also need to be scaled according to their distance and position relative to the image plane for a correct visualization (Figure 2.4c). Since every voxel is handled separately it is possible to improve the overall performance of the algorithm by skipping voxels that would result in transparent splats. It also allows the visualization of datasets with arbitrary grid geometries, as interpolation is done exclusively by compositing splats in image space.

The initial approach of projecting all splats directly to the image plane often leads to color bleeding artifacts. In order to avoid these artifacts splatting is performed on a volume subdivided into a stack of volume slices (buffers) that are blended together as final step. The alignment of these sheet buffers is handled similarly as done in the previously explained slicing method. Westover's [27] original algorithm used object-aligned sheets causing the already mentioned artifact problems.

Mueller and Crawfis [28] proposed to use image-aligned splatting to diminish the slicing artifacts. Additionally, to improve the overall performance of the algorithm, early splat termination was introduced [29]. By dynamically computing a screen occlusion map it is possible to identify splats that are not visible on the output image and exclude them from the further rendering process.

Vega-Higuera et al. [30] improved the algorithm by employing point sprites to render splats instead of textured quad primitives. Utilizing the same principle, Neophytou and Mueller [31] presented a GPU-based implementation of splatting, utilizing fragment shaders.

- Ray casting

Ray casting is an image-order algorithm proposed by Levoy [32] in 1988 and has been receiving a lot of attention in the research field. Starting from the view point, rays are cast through the center of every pixel of the image plane. Rays passing the volume are sampled at fixed intervals in the dataset (Figure 2.5a). The samples are composited according to the DVRI (Equation 2.2). Since the compositing is processed iteratively in a front-to-back order the over operator of alpha compositing [33] can be employed:

$$C_{dst} = C_{dst} + C_{src}\alpha_{src}(1 - \alpha_{dst}) \quad (2.3)$$

$$\alpha_{dst} = \alpha_{dst} + \alpha_{src}(1 - \alpha_{dst}) \quad (2.4)$$

Where C_{dst} and α_{dst} are the color and transparency values accumulated along the ray up to the current sample point and C_{src} and α_{src} are the color and transparency values of the currently evaluated position.

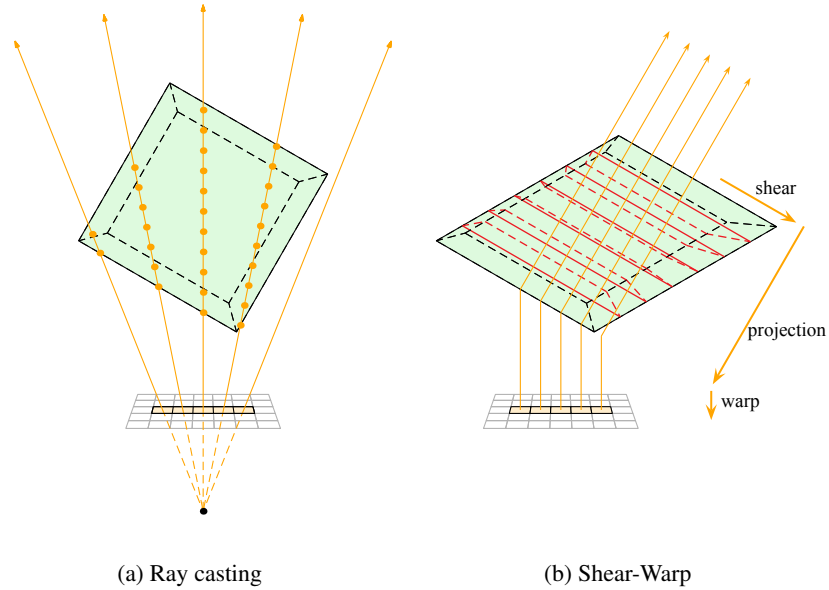


Figure 2.5: Image-order direct volume rendering methods: (a) DVR by casting rays through every pixel of the image plane. Rays are sampled at regular positions and the results are accumulated. (b) DVR by shearing the volume, projecting object aligned slices onto an image and warping the image to the image plane (example uses parallel projection).

In general, every ray is traced separately which allows algorithm improvements and extensions on a per pixel basis. It also divides the rendering computation into a large number of similar compositing tasks, suitable for parallel execution on current graphics hardware. This led to multi-pass GPU implementations by Roettger et al. [34] and Krüger and Westermann [35], as well as numerous single-pass implementations that were developed after the introduction of shader model 3.0.

Depending on the type of 3D-dataset it is possible that a large number of samples do not contribute to the result image because they are classified as transparent by the transfer function. Levoy [36] exploited this property to increase the algorithm performance by implementing hierarchical space skipping using binary pyramids. Here the occurrence of opaque voxels is encoded into a precalculated pyramid representation. By starting the sampling at the top of the pyramid and descending into lower levels it is possible to efficiently skip large portions of voxels that do not contain visible samples.

Taking Levoy’s approach even further, van Walsum et al. [37] proposed the use of adaptive sampling in order to skip over regions of homogeneous density values. This is done by sampling the ray at large intervals and comparing adjacent samples. If their density difference exceeds a preset limit, the step width of samples between them is refined. While saving some computation time, this method might cause a loss of image quality in areas of high frequency density changes.

Another contribution by Levoy was the adaptive refinement of the output visualization [38]. In order to ensure interactivity during changes of the viewpoint, ray casting is only executed for a fraction of output pixels while the remaining pixels are interpolated. The resulting image is of lower quality, but often sufficient as a fast preview.

Apart from performance optimizations, research to extend the basic optical model has been conducted as well. Shadows for instance notably improve the depth perception in visualizations. Ropinski et al. [39] give an overview on the three predominant approaches in this field: shadow rays, shadow mapping and deep shadow maps. Hernell et al. [40] proposed the use of ambient occlusion to obtain shadows independent of light positions, by evaluating the neighborhood of a sample and shading it depending on its local occlusion.

Other improvements, to gain a more physically plausible visualization, include advanced light transport and illumination models, light scattering effects and the use of Monte-Carlo integration. An extensive overview on these techniques can be found in the works of Hadwiger et al. [41] [42].

- **Shear-warp**

In order to benefit from the fast computation of object-order methods and the flexibility of image-order methods, Lacroute and Levoy [43] proposed the use of shear-warp factorization, a combination of ray casting and slicing. By shearing the volume, the viewing rays are transformed to be parallel to the principal viewing axis in the sheared object-space. This allows the efficient simultaneous sampling of rays through the volume. Compositing results are projected onto an intermediate image plane. The intermediate result is then warped according to the viewing direction to produce the final image (Figure 2.5b). The approach supports orthogonal as well as perspective projection. Shear-warp factorization is highly optimized for CPU execution and is recognized as one of the fastest software rendering algorithms [24]. Shearing in combination with 2D texture lookups causes notable artifacts in the resulting image. Due to the availability of more powerful graphics hardware, this method has been mainly replaced by ray casting.

2.4 Smart Visibility Techniques

A recurring problem in 3D visualizations is occlusion. In general, it is not possible to map three-dimensional datasets to a two-dimensional representation which maintains the spatial context without losing information. The view on interesting details of a visualization is often obstructed by one or more opaque regions of low importance. This problem is not a new one, it frequently occurs in technical and medical illustrations. Illustrators have developed many different techniques to emphasize regions of interest, while showing enough of the occluders to keep the overall context of an image intact. Based on these methods several rendering techniques for occlusion removal have been conceived. These techniques are often referred to as smart visibility techniques, a term that has been introduced and defined in the work of Viola and Gröller [4]. The following are examples of smart visibility techniques:

Exploded views are illustration styles typically used in technical drawings, but they can also improve the visibility in medical visualization. This technique exploits the fact that the human

anatomy partially consists of different tissue layers nested within each other. Bruckner and Gröller [44] introduced methods to divide outer tissue layers of medical volume datasets into small sections. These sections are then spatially separated from the remaining volume to allow the viewing of deeper structures. Correa et al. [45] proposed the use of feature-aligned volume manipulation. By cutting the surface of a dataset and retracting the outer tissue layers to the sides, it is possible to see inner structures without losing the context of surrounding tissue. The final images can mimic the results of surgical examinations.

Another approach taking advantage of tissue layers has been proposed by Salama and Kolb [46] for DVR with conventional transfer functions. By analyzing the opacity profile of each ray, it is possible to estimate continuous density layers within a dataset. This information is used to render the outermost layers with low opacity to create a ghosting effect. Malik et al. [47] improved the estimation of layer borders by utilizing more elaborate methods of ray profile analysis. The quality of the results is strongly dependent on the complexity of the layer structure in the visualized volume.

Another approach to the occlusion problem is the use of cutaway views. The visibility of important structures is achieved by completely omitting occluding regions from the rendering process. Feiner and Seligmann [48] proposed the combination of cutaway and ghosting techniques. It is possible to improve the perception of the overall context in many cases, by rendering areas that would usually be omitted in a highly transparent and stylized way. Diepstraten et al. [49] presented implementations of different cutaway techniques. Additionally, they discussed rules of their usage in order to maintain a general overview of the image. Burns et al. [50] extended the cutaway approach, by proposing an adaptive cutaway method for polygonal scenes. Birkeland et al. [51] proposed the use of a simulated elastic membrane as clipping surface for a cutaway visualization. The deformable membrane uses underlying data properties to adjust itself to salient structures, in order to prevent it from passing through features of the dataset.

In contrast to meshed geometry, volumetric datasets often do not contain distinct boundaries between the objects. Transfer functions for mapping of colors and opacities to sample intensities are commonly adjusted to generate a smooth transition between tissue types. This results in a fuzzy classification that cannot be exploited to generate cutaway geometries directly. Viola et al. [52] addressed this problem by introducing an additional transfer function, assigning an importance value to every sample. This value allows users to specify their interest in the visibility of different density ranges. The paper presents importance-driven volume rendering methods. They utilize the additional classification of the importance transfer function to emphasize regions of interest. Notable examples are Maximum Importance Projection, importance-driven color and opacity modulation and volume thinning. Based on this work, Burns et al. [53] proposed cutaway techniques for direct volume rendering, aiming to visualize registered 2D ultrasound scans within volumetric computer tomography datasets. Clearview, developed by Krüger et al. [54], combines several of the presented focus+context approaches into one framework in order to visualize a wide range of different datasets.

The additional importance values increase the complexity of setting useful transfer functions. Wu and Qu [55] addressed this problem by using interactive transfer functions. In the proposed framework, users edit and create transfer functions by interacting directly with the result image using a set of tools, e.g., object selection. It also allows the specification and combination of

transfer function layers to create a wide variety of visualization effects. Mak et al. [56] proposed a similar framework. It improves the classification quality of the generated transfer functions by using a genetic algorithm.

By calculating the contribution of individual samples to the visualization, intensity based visibility histograms can be generated. These histograms, proposed by Correa and Ma [57] [58], provided a base for improved classification methods. Peaks in the visibility histogram show layers of similar tissue and allow an automatic refinement of opacity transfer functions to show deep structures. Correa and Ma [59] later used a similar approach to improve the classification by analyzing the ambient occlusion of individual samples. A 2D-occlusion spectrum of a dataset can be precalculated. By using an occlusion based transfer function it is possible to visualize distinct structures even in complex volumetric datasets.

2.5 Surface Reconstruction Methods

The smart visibility algorithm presented in this thesis requires the reconstruction of a continuous smooth surface from a sparse set of initial points. Methods for inter- and extrapolation of scattered spatial datapoints are needed in a wide variety of fields. Computer animation, scientific simulation, medical visualization, machine learning and geoscientific modeling are examples for fields that rely on surface reconstruction.

Surface reconstruction algorithms aim to generate a surface from a set of sample points that preserves the topology of the original surface and reproduces its sharp features and surface boundaries. Robustness towards sampling density and measurement errors are desired additional features of these algorithms [60]. A wide range of reconstruction methods have been proposed, for general as well as application-specific use cases. An overview of the field is given in the surveys of Mencl and Müller [61], Cazsal and Giesen [62] and Chang [60].

A basic approach to surface reconstruction is, to use sample points as vertices of a triangle mesh. The alpha shapes algorithm was proposed by Edelsbrunner and Mücke [63]. It calculates the 3D Delaunay complex of a given point set and subsequently removes large triangles based on their circumference. Surfaces resulting from this algorithm are not necessarily continuous and can have a different topology than the original surface. To address this problem Amenta et al. [64] proposed the crust algorithm. It uses the Voronoi diagram to approximate the medial axis of the surface and selects triangles in the Delaunay complex based on their adjacency to Voronoi vertices. If specific sampling conditions are met, the crust algorithm guarantees to preserve the original topology. Improvements on performance as well as robustness against undersampling and noise have been proposed by Amenta et al. [65] with the power crust and by Dey and Giesen [66] with the robust cocone algorithm.

Implicit surfaces, defined as the isocontour of their corresponding scalar functions, can be used for surface reconstruction. Several algorithms to define suitable scalar functions have been proposed. A common approach is to fit radial basis functions (RBF) to the given scattered point sets. Muraki [67] and Carr et al. [68] proposed methods to fit RBFs by generating additional off-surface control points. Walder et al. [69] presented an improved fitting algorithm that does not require off-surface points. A different way to define an implicit surface from scattered points is the use of the moving least squares technique, proposed by Shen et al. [70]. By including a

position-dependent weighting term into the scalar function of the implicit surface, only the local neighbourhood is considered during the least square approximation.

Another surface reconstruction technique is to fit multiple parametric surface patches and subsequently combine them to a complete surface. Lee et al. [71] employed multilevel B-splines to compute a continuous surface through a set of irregularly spaced points. This allowed surface interpolation as well as approximation. Haber et al. [72] proposed the use of cubic splines to approximate surfaces in large-scale scattered unorganized point clouds. In their approach, the subdivision of the surface into patches of triangle meshes greatly reduces the complexity of the reconstruction. Working on similar datasets, Bertram et al. [73] proposed the use of B-splines in combination with adaptive quadtree-based clustering. The resulting surface is approximated locally. The size of the neighborhood is defined by the density of points in that region. Nielson [74] presents the use of normalized implicit eigenvector least squares models to approximate smooth surfaces even in noisy point clouds.

The approaches presented are only few examples of the many techniques that have been developed in the field of surface reconstruction. Additional algorithms can be found in the surveys mentioned at the beginning of this section.

Smart Visibility Methods

This chapter describes the smart visibility methods that were implemented and evaluated in the scope of this thesis. The presented methods have been developed as part of a research collaboration between the Vienna University of Technology and GE Healthcare. A. Varchola [3] was the main contributor in the research of occlusion removal methods. The methods aim to provide an unoccluded view of the region of interest in obstetric ultrasound examinations. All presented methods follow the same basic concept, which will be referred to as *smartvis algorithm* for the remainder of this work. The algorithm consists of three steps:

1. Selection of initial points lying between fetus and occluders.
2. Reconstruction of a clipping surface from the obtained sparse set of initial points.
3. Direct volume rendering of the dataset, utilizing the clipping surface as separator to process occluders and region of interest differently.

In detail, the evaluated methods differ mainly in the surface reconstruction algorithm used in the second step of the *smartvis* algorithm.

The following sections cover the basic assumptions used in the *smartvis* algorithm as well as detailed explanations of different steps of the algorithm. Section 3.1 lists the characteristics of ultrasound data in general and discusses the specific case of obstetric sonography datasets. The detection of initial points is explained in Section 3.2. Methods for the reconstruction of a clipping surface from initial points are covered by Section 3.3. Section 3.4 presents ways to incorporate the clipping surface into the visualization process.

3.1 Characteristics of Ultrasound Data

Ultrasound examinations, based on the principles explained in Section 2.1, have several advantages over other medical imaging modalities. Inexpensiveness, portability and the low risk of the procedure are some of them. However, in datasets generated by US tissue boundaries are not as

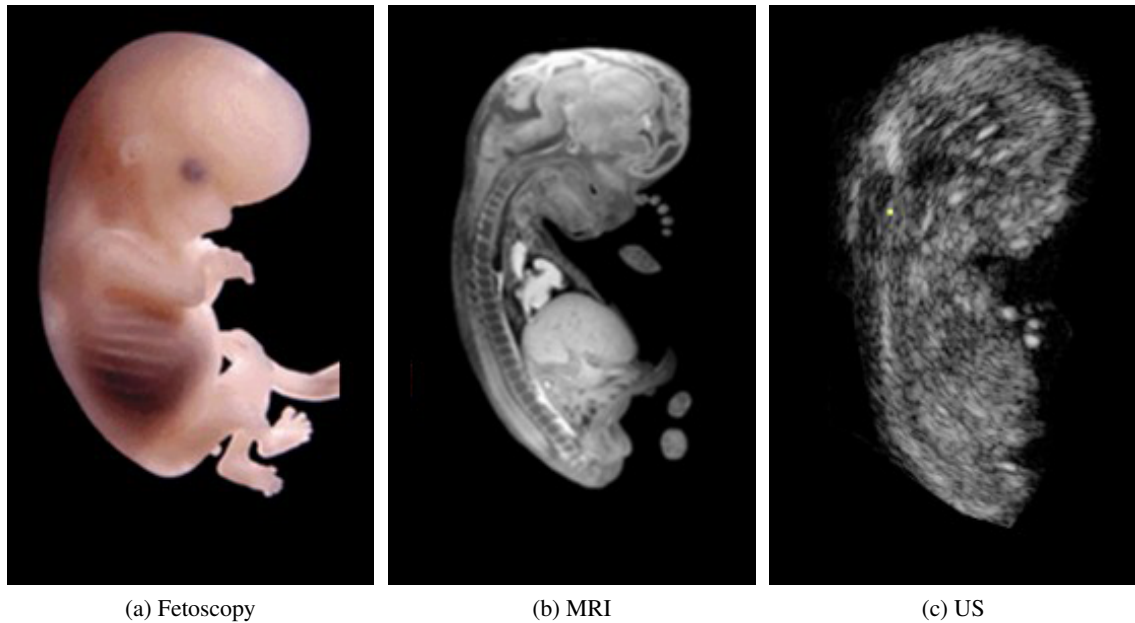


Figure 3.1: A side by side view of the results of three medical imaging techniques [75]: (a) Fetoscopic image of a fetus. (b) Visualization of a single slice through a fetus obtained by Magnetic Resonance Imaging (MRI). (c) Visualization of a single slice through a fetus obtained by B-mode ultrasound.

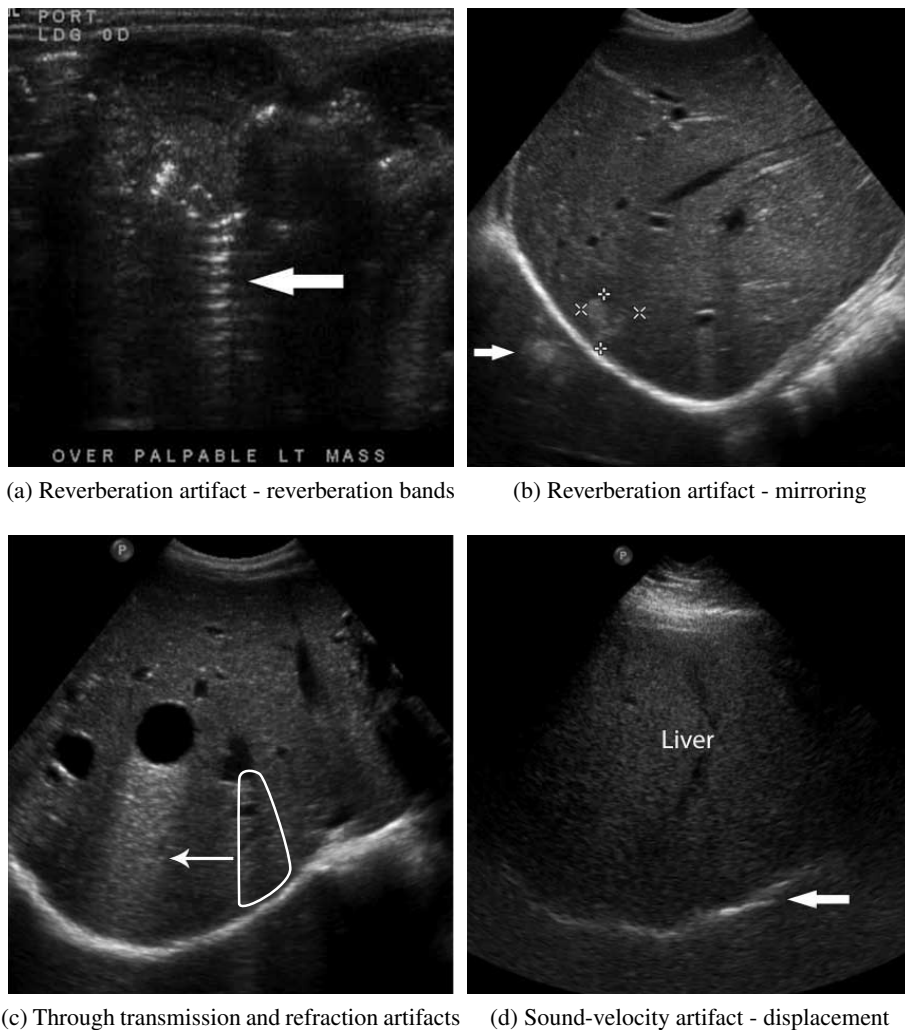
clearly defined as in datasets obtained by other imaging modalities, most notably MRI and CT. Figure 3.1 shows this property on a comparison between MRI and B-mode US visualizations. The comparatively low image quality is caused by artifacts and physical limitations of the basic principle of ultrasound examinations [76] [77]:

- Speckle noise

The most obvious artifact in US-data is speckle noise [78]. The depth resolution of ultrasound machines is limited by the length, frequency and bandwidth of its generated ultrasound pulses. Echoes created by reflectors with a distance below the resolution limit interfere with each other. This often causes artificially large (constructive interference) or small (destructive interference) signals [79]. Even areas of a single tissue type feature minor differences in density and are affected by speckle noise. The US-scan in Figure 3.1c shows this artifact evidently by its grainy structure.

- Reverberation artifacts

An echo received by the transducer of an ultrasound machine is commonly interpreted as a single reflection in linear direction of the associated pulse. Strong reflectors within the scanned area are sometimes causing multiple echoes and more complex sound wave paths which lead to a number of artifacts in the obtained data. Reverberation bands, as seen in



(a) Reverberation artifact - reverberation bands

(b) Reverberation artifact - mirroring

(c) Through transmission and refraction artifacts

(d) Sound-velocity artifact - displacement

Figure 3.2: Examples of common ultrasound artifacts [77].

Figure 3.2a, are occurring if two or more strong parallel reflectors are located in direct line of the pulse. The echo of the more distant object has to pass the closer object on the way to the transducer. This produces inter-reflections between these two objects, visible as parallel bands of uniform distance and decreasing intensity in the scan. If these objects are spaced closely together the reverberation bands form an almost continuous comet-tail artifact [77].

A similar artifact can be observed if the pulse encounters closely-spaced liquid/gas interfaces in a cluster of micro bubbles. The ultrasound pulses cause vibrations in small pockets of liquid between the bubbles. The liquid then emits continuous ultrasound waves towards the transducer. This is visualized as an uninterrupted area of high intensity behind the object known as ring-down artifact. If the scanned area contains a large, smooth and

strong reflector, e.g., a membrane, single inter-reflections may occur. In this situation even objects that generate weak echoes can appear as mirror image artifacts in the scan [77]. Figure 3.2b shows this artifact in the lower left corner.

- Through transmission artifacts

Ultrasound pulses get attenuated depending on the thickness of the penetrated tissue, therefore deep-lying structures cannot respond with strong echoes. To prevent a loss of contrast of these structures a user defined "Time gain compensation" (TGC) can be employed, to amplify echoes with a long response time. Since the TGC assumes a uniform attenuation over the whole scanned area, a structure attenuating the pulse stronger than expected will cause a shadowing effect in the areas behind itself. Likewise a weaker attenuator will cause a brightening of more distant areas [77]. This is shown in Figure 3.2c marked by the white arrow.

- Refraction artifacts

Based on the same principle as optical refraction, the direction of ultrasound pulses gets diverted at oblique interfaces between media with different sound propagation velocities. Refraction has a particularly strong effect at the edges of objects with different sound propagation properties. The flat angle of the pulse incidence causes shadowing artifacts behind these edges. Pulses and their corresponding echoes are strongly diverted which prevents them from properly returning to the transducer. Figure 3.2c shows this artifact in the area marked by the white border.

Large and smooth refraction interfaces can cause a strong apparent displacement of features due to the fact that single refraction diverts the echo of an already refracted pulse back towards the opposite direction of the initial pulse [77].

- Sound-velocity artifacts

Similar to attenuation properties ultrasound machines also assume a constant speed of sound within the scanned object. Pulses passing through areas with strongly deviating sound velocity properties (e.g., fluids or bones) will arrive earlier or later than expected. This causes inaccuracies in the depth recording of all echoes coming from behind such areas. Figure 3.2d shows the displacement of a membrane caused by this artifact [77].

- Beam-related artifacts

Depending on the transducer, the shape of the ultrasound beam may vary. Many transducers can be set to focus the beam on a certain depth to get improved contrast in that area of the scan. The beam itself has a more complex three-dimensional shape than just the scanned area. It is possible for the main beam to diverge out of the scanned area behind the focal plane. Additionally, due to physical effects, off-axis low-energy beams, called side- and grating-lobes, are emitted by the transducer. In both cases echoes from outside the scanned area might be visible as artifacts within the scan [77].

So far only artifacts of two-dimensional B-mode ultrasound scans have been presented. The same artifacts occur in 3D ultrasound datasets as well. However, 3D datasets are in general

visualized using the position of the transducer as viewpoint. In many cases this reduces the influence of the presented artifacts in the resulting visualization, since they are often occluded by the volume areas that caused them.

3D Obstetric Sonography Datasets

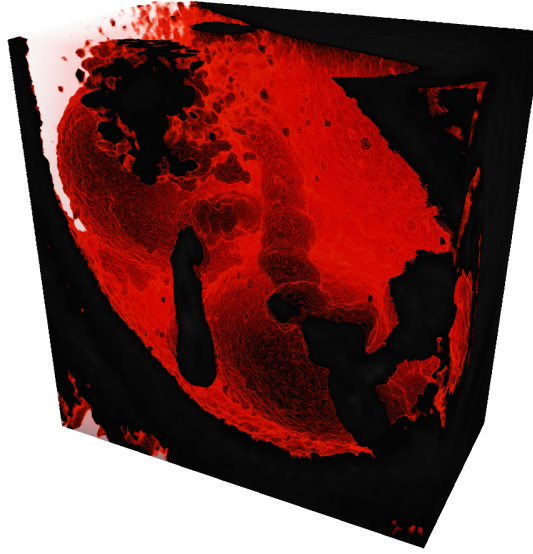
The smart visibility methods evaluated in the scope of this thesis are designed to improve the visibility of fetuses in visualizations of 3D obstetric sonography datasets. These datasets are obtained by using mechanically swept transducers (Section 2.2) during prenatal US examinations. They show the general characteristics of US datasets presented in the previous section. An analysis of nine test datasets provided by GE Healthcare indicated the prevalence of two anatomic features in all datasets, i.e., amniotic fluid and fetal bones. A detailed examination of both properties led to the following results:

- Amniotic fluid

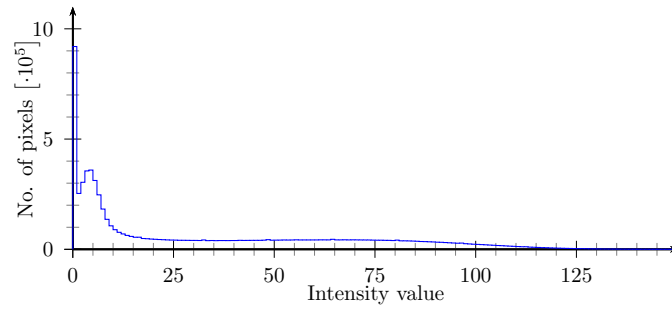
From an early stage of a regular pregnancy, the fetus is surrounded by a protecting and nourishing liquid called amniotic fluid. One of the functions of this liquid is to inflate the amniotic sac, allowing the fetus to move and develop within. It also protects it from physical impacts to the mother's abdomen. With the exception of the last weeks of pregnancy, the amount of amniotic fluid is increasing proportionally to the growth of the fetus. It can therefore be assumed that in 3D obstetric sonography datasets the fetus will be surrounded by amniotic fluid.

Fluid-filled areas do not produce a strong response in ultrasound scans, due to their homogeneous density and the low echogenicity between interfaces of soft tissue and fluid [6]. The analysed histograms exhibited two notable peaks present in all available test datasets. The first peak originates from zero values in the dataset, while the second, lower one is mainly composed of the boundary area between amniotic fluid and soft tissue. Figure 3.3b shows both peaks in an example histogram. Density values in the interval between these peaks can be associated with amniotic fluid, as shown in Figure 3.3a. The distinct densities of the amniotic fluid can be exploited for segmentation by thresholding. Standard 3D ultrasound machines provide the user with an adjustable parameter for this threshold.

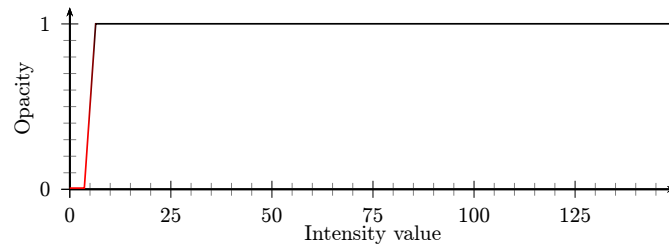
Ultrasound pulses travel faster and are attenuated less within amniotic fluid (or fluid in general) than in other tissues. This can cause sound-velocity and attenuation artifacts as previously mentioned.



(a) DVR - amniotic fluid

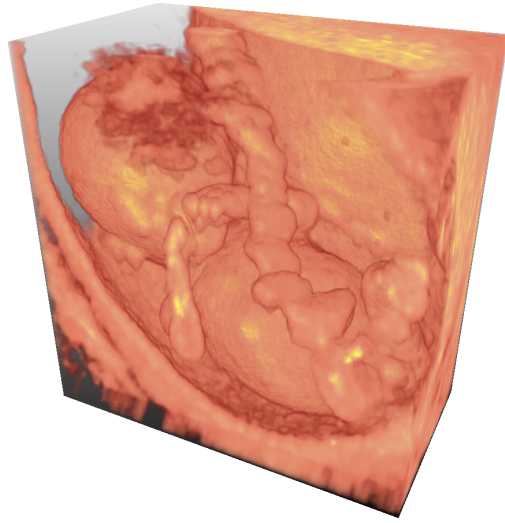


(b) Intensity Histogram

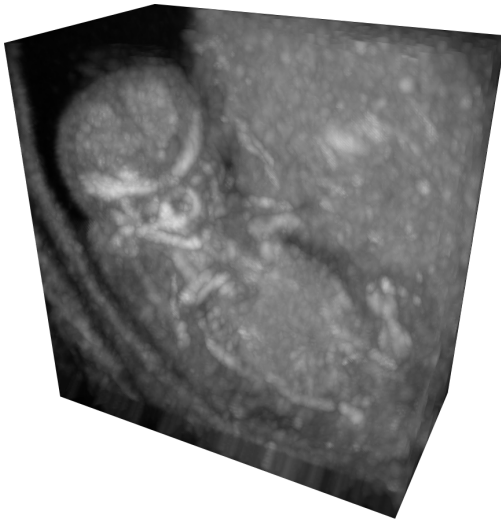


(c) Color and opacity transfer function

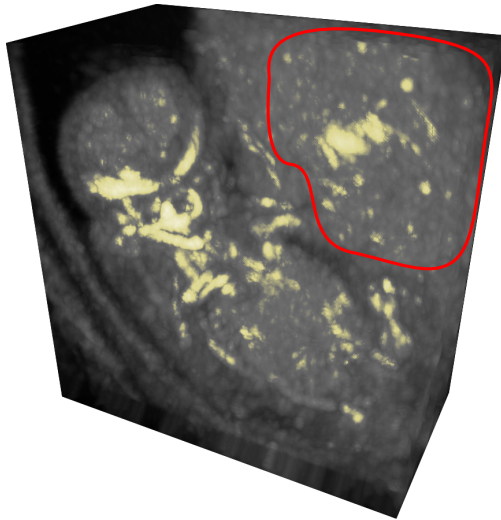
Figure 3.3: Amniotic fluid visualization in dataset "E04": (a) DVR visualization of the amniotic fluid contained in the dataset using a transfer function that depicts it highly transparent red and other tissues in opaque black (b) Histogram of the intensity values of the dataset (c) The color and opacity transfer function used for the DVR visualization.



(a) DVR



(b) MIP



(c) MIP & transfer function

Figure 3.4: Bone visualization in dataset "E04": (a) Standard DVR of the dataset to show the context of the MIP. (b) MIP showing the skeletal structure of the fetus. (c) MIP using a transfer function that emphasizes high intensity values in yellow: facial bones, spine, ribs and leg bones are visible. The regions within the red border are not related to fetal bones and are likely caused by through transmission artifacts.

- Fetal bones

The ossification of bones starts in the third month of embryonic development. The anorganic components of a growing bone matrix are visible as strong reflectors in ultrasound examinations. The development of bones is not a uniform process over the course of a pregnancy. During an examination, bones in many different ossification states are encountered. This prevents them from forming a distinct peak in a datasets histogram. However, bones often cause the strongest echoes within an ultrasound scan. By using Maximum Intensity Projection (MIP, Section 2.3) it is possible to visualize bones in the dataset. Figure 3.4b shows a MIP of a fetus featuring several distinct visible bones of its skeleton. According to GE specialists MIP is the standard visualization mode to perform bone related examinations, e.g., bone counting and length measurement.

In the analyzed datasets, areas with high intensity values were predominantly occupied by bone tissue. However, datasets containing large areas of amniotic fluid also featured through transmission artifacts within the range of bone intensity values. These artifacts were located at the back side of the amniotic sac. The thresholding of MIP images using a high intensity value therefore results in visualizations showing the locations of bones and through transmission artifacts within a dataset. Figure 3.4c emphasizes the resulting locations in yellow and marks areas caused by artifacts rather than bones with a red border.

3.2 Initial Point Detection

The first step of the smartvis algorithm is the selection of a set of points within a given 3D obstetric sonography dataset, that can be used to reconstruct a clipping surface. This surface should separate the fetus from all occluders towards the viewpoint. One of the core assumptions of the smart visibility methods presented in this work is, that it is possible to select a suitable set of points based on the features discussed in the previous Section 3.1. The selected points are referred to as *initial points*.

Due to the gap filled with amniotic fluid between fetus and occluders several solutions for surfaces that correctly separates them are possible, as shown in Figure 3.5c. Each initial point selected for surface reconstruction should be located on one of the possible clipping surface solutions. By this requirement, locations on rays starting at the viewpoint and passing through the volume qualify as potential initial points if they fulfill one of the following conditions:

- If the ray passes through the fetus all points between the exit point from the last occluder in front of the fetus and the entry point to the fetus itself form an interval of potential initial points for the given ray.
- If the ray passes through the amniotic sac but not the fetus all points between the entry point to and exit point from the amniotic fluid form an interval of potential initial points.

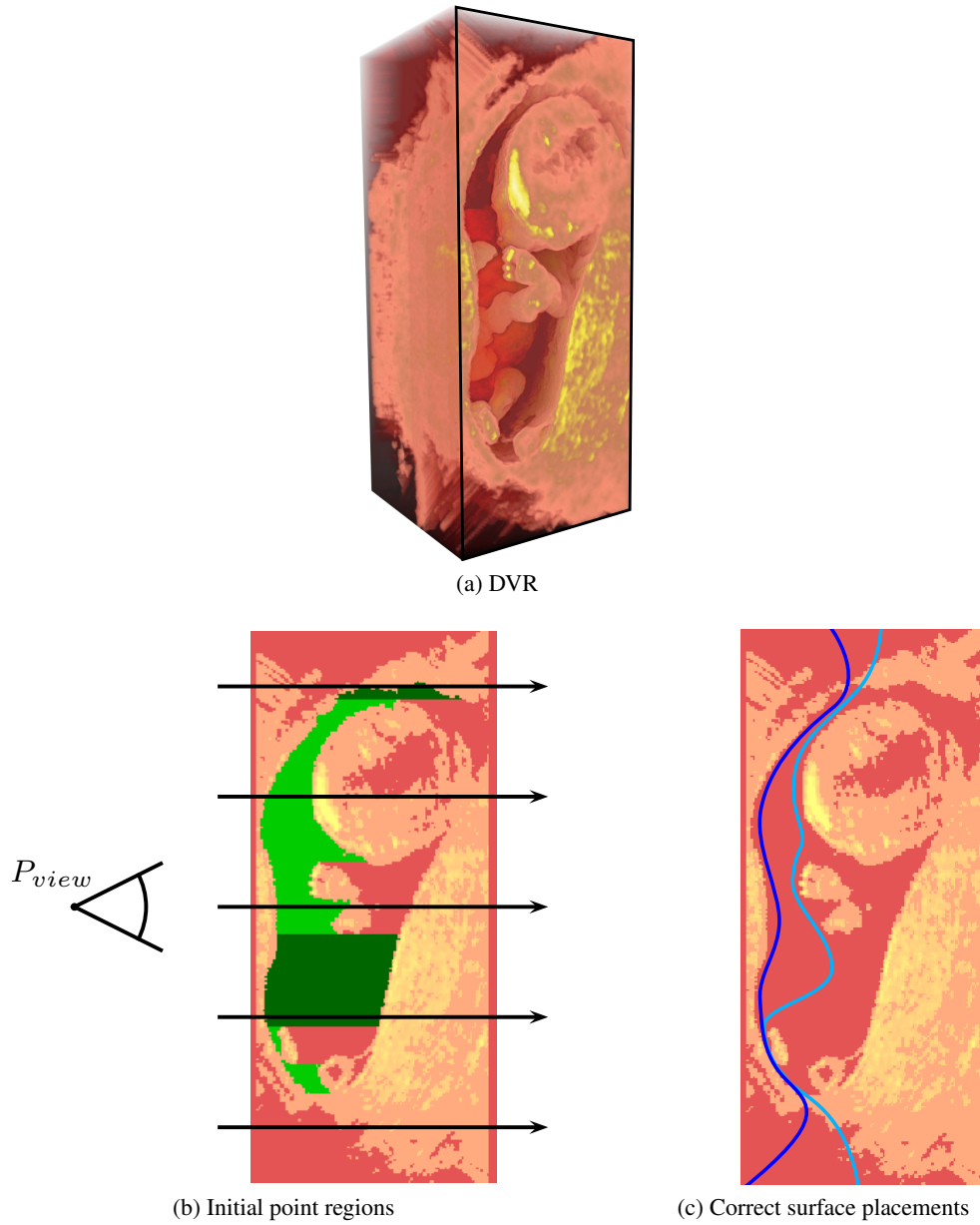


Figure 3.5: Potential initial point regions in "E04": (a) DVR visualization of the dataset clipped to show the context of the used slice (black border). In the visualization thresholding is used to differentiate amniotic fluid (red), soft tissue (pink) and bone/high intensity tissue (yellow). (b) Visualization of the selected slice showing locations of potential initial points for viewpoint P_{view} with orthographic projection. Light green color marks potential initial point regions for rays passing through the fetus, dark green color marks regions for rays passing only through the amniotic sac. (c) Visualization of the selected slice showing two correct clipping surface placements in light and dark blue.

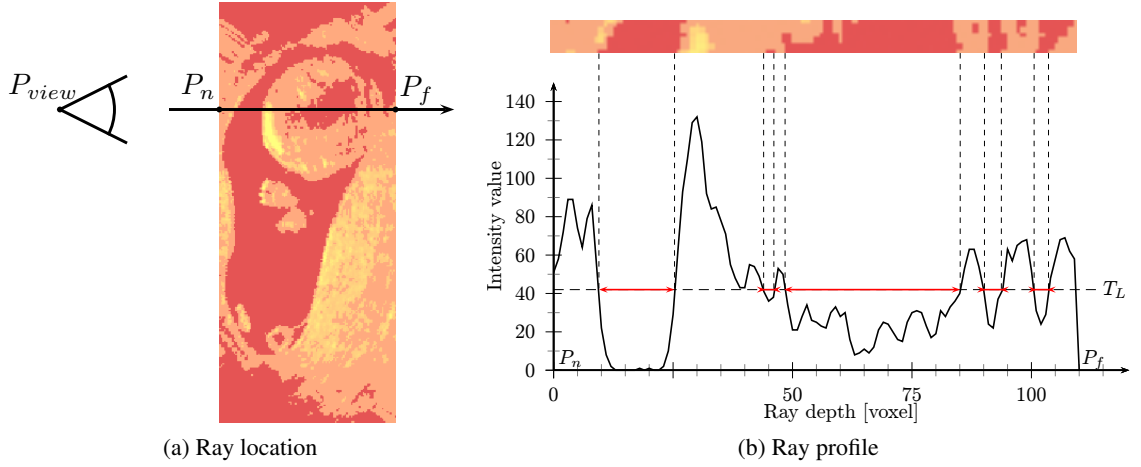


Figure 3.6: Amniotic fluid based initial point intervals: (a) Visualization of the ray position in a slice of dataset "E04". P_n marks the volume entry and P_f the volume exit point of the ray. (b) The intensity profile of the ray. Potential initial point intervals (red arrows) are identified by the ray's entry points into and exit points from the amniotic fluid using the threshold T_L .

Figure 3.5b shows the locations of potential initial points within a single slice of a test dataset. Selecting an initial point along every ray passing through the amniotic sac would solve the occlusion problem. Due to the diffuse boundaries and artifacts in ultrasound datasets finding a point for each individual ray, without prior knowledge about the location of the fetus, is a very difficult task. Therefore, in this step of the smartvis algorithm a heuristic approach is used to select a small number of initial points in the volume. Using a ray casting algorithm, the profile of each ray is analyzed to locate regions passing through amniotic fluid or bone tissue. If a ray profile meets certain conditions an initial point for the corresponding ray is calculated. These experience-based methods are simple to evaluate, but have a chance to select points that do not qualify as initial points.

Approaches based on the thresholding of the amniotic fluid alone have resulted in high numbers of such invalid points. Analyzed rays are often passing through several amniotic fluid regions of various lengths. This can be caused by ultrasound artifacts or an ambiguous placement of the fetus and its umbilical cord. Figure 3.6 illustrates this by showing the potential initial point intervals found on the profile of a single ray. A threshold value T_L is used to determine the ray's entry into and exit from the amniotic fluid. Several intervals are found, but only the first one is located between fetus and occluding tissue and therefore contains potential initial points. Methods of choosing the correct interval using its sequential position (e.g., first-hit) and/or its length have shown unreliable results.

To improve the results of the heuristic, additional properties of the ray profile have to be taken into account. As described in Section 3.1, bones of the fetus are visible in MIP images and can be associated with rays that feature high maximum intensity values. It can be concluded, that the maximum intensity peaks in the profiles of rays passing through fetal bones are located

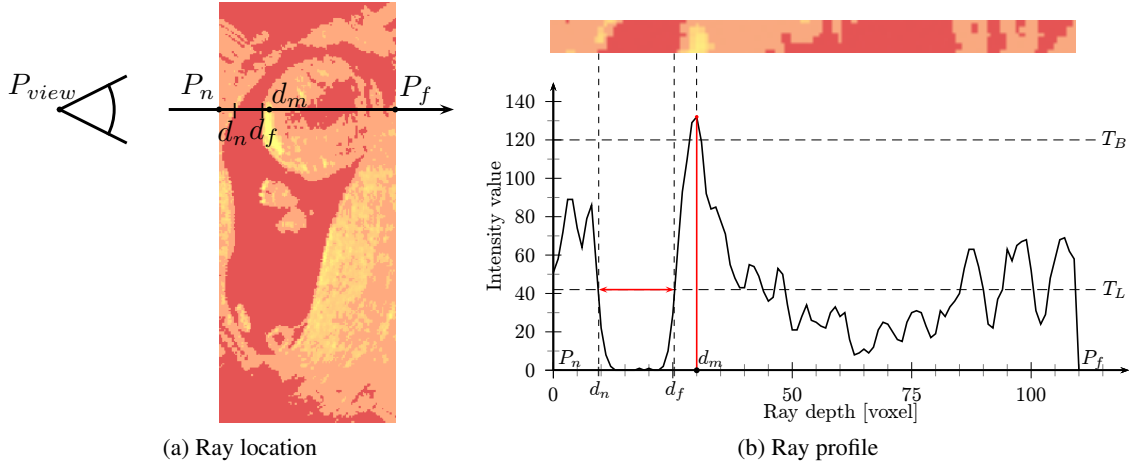


Figure 3.7: Initial point interval selected by ray profile analysis: (a) Visualization of the ray position in a slice of dataset "E04". (b) The ray profile showing the selected interval $[d_n, d_f]$ in front of the sample with depth d_m showing the maximum intensity value.

within the tissue of the fetus. This feature can be exploited to refine the previously described amniotic fluid based thresholding operation. Thresholding rays based on their maximum intensity with an empirically determined threshold T_B ensures that only rays passing through bones are considered for this improvement. Figure 3.7 demonstrates the refinement using the same ray as Figure 3.6. The maximum intensity peak in the ray profile features a higher intensity than the bone threshold T_B . The maximum intensity peak along the ray, located at depth d_m , is subsequently assumed to be located within the fetus. Therefore the last interval in front of d_m , $[d_n, d_f]$, is selected. All points within the interval are considered as initial points. However, the subsequent steps of the smart visibility algorithm require the selection of a single point within a given interval. A single initial point d_p within the interval can be selected using the following equation:

$$d_p = d_n + q(d_f - d_n), 0 \leq q \leq 1, \quad (3.1)$$

where d_n and d_f are the near and far endpoints of the interval and the parameter q defines the position of the selected initial point within the interval. By using the same value of q for every ray it is possible to control the distance of the resulting initial point set with respect to the surface of the fetus.

The depths d_p of the selected initial points are stored in a depthmap for further processing. Due to the bone thresholding, the depthmap is only sparsely filled. Depending on the size of the fetus within the dataset 1-5% of the map contain initial point depth values. Figure 3.8 illustrates a MIP of a dataset side-by-side to its initial point depthmap. While most of the initial points are selected from rays passing through bones a fraction of them are selected due to high intensity through transmission artifacts, as explained in Section 3.1. These artifacts can occur in rays that pass a large region of amniotic fluid. Maximum intensity peaks are often encountered in the tissue behind these fluid regions. In practice, initial points selected due to through transmission

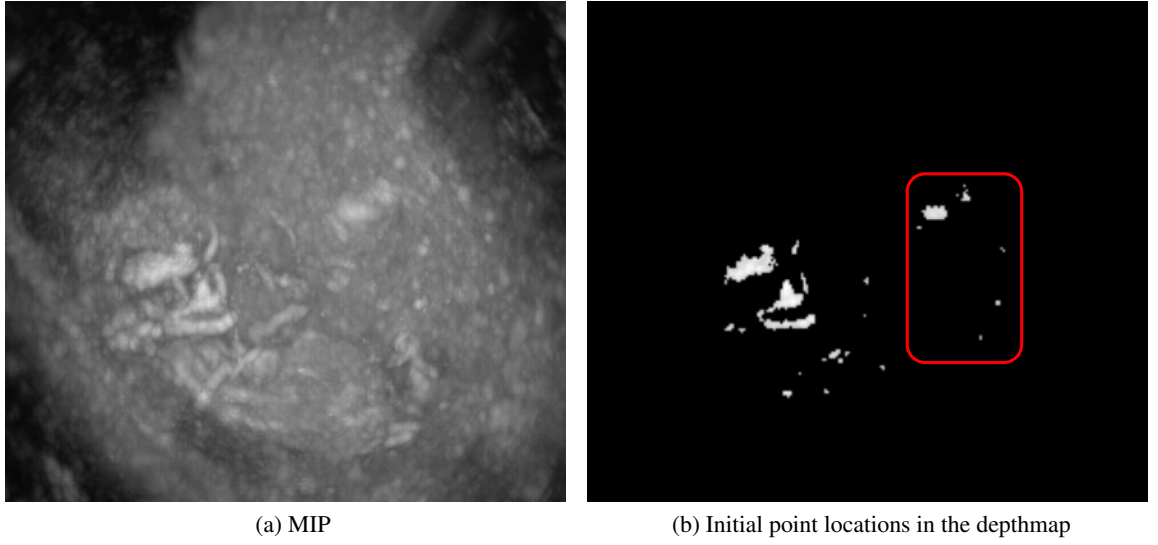


Figure 3.8: Results of the initial point selection in dataset "E04". (a) A MIP visualization of the dataset. (b) The distribution of initial points in the depthmap after applying the bone threshold T_B . Regions marked by the red border are not related to bone tissue.

artifacts are located predominantly in areas not occupied by the fetus. In these cases the heuristic will choose the amniotic fluid area that caused the artifact as potential initial point interval. This is in most cases the correct selection. However, incorrectly selected initial points due to other effects are also possible and have to be considered during the subsequent surface reconstruction and rendering steps.

3.3 Surface Reconstruction

The initial point selection results in a depthmap storing the distances of the selected initial points to the viewpoint. Since the map is only sparsely filled (Figure 3.8), it can not be directly used for occlusion removal over the entire image. It is necessary to estimate a depth value for every pixel of the map based on the given initial points. This essentially constitutes a surface reconstruction problem. As presented in Section 2.5, numerous solutions have been proposed for this kind of task. The reconstruction method is required to interpolate a clipping surface between the known initial points as well as to extrapolate it towards the image borders. In order to avoid visual artifacts, e.g., edges caused by sudden changes of the clipping surface, a smooth reconstruction is desirable.

The reconstruction methods tested in the framework proposed in this thesis can be categorized into global and local approaches. The proposed global method attempts to reconstruct the surface using a spline-based approach. This requires the solution of linear equations systems. The tested local methods consider the close neighborhood of each surface point to increase and

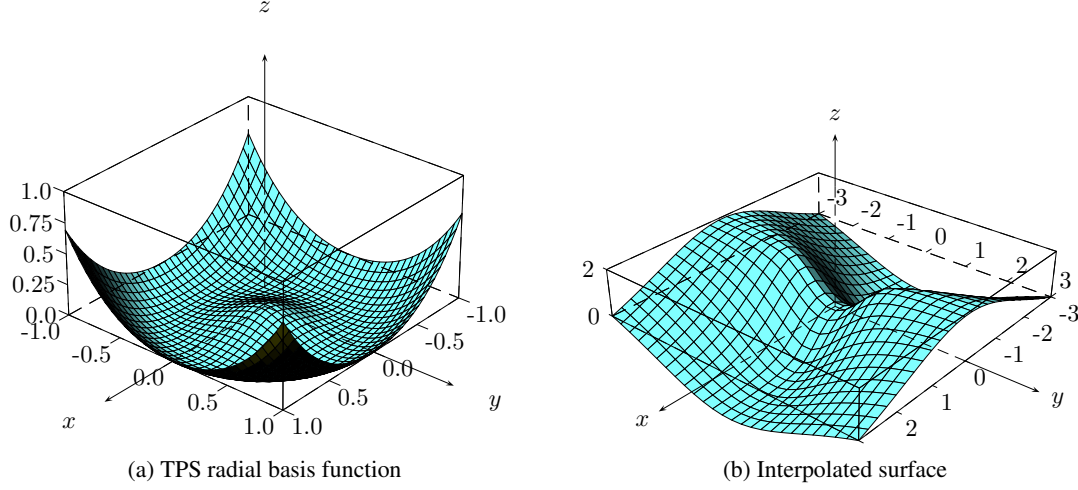


Figure 3.9: The Thin Plate Spline: (a) 3D-plot of the radial basis function for TPS. (b) Example of an interpolated surface using the control points $\{(1, 0, -1), (0, 1, 1), (-1, 0, -1), (0, -1, 1)\}$.

refine the existing clipping surface. They require several iterations to reconstruct the whole surface.

Global Methods

Thin-Plate Splines

Splines are commonly used to solve approximation and interpolation problems. Thin-Plate Splines (TPSs), the two-dimensional extension of the cubic spline, can be employed to solve the surface reconstruction problem stated before. In the medical field TPSs are often used for the co-registration of medical images from different modalities or examination times [80]. The basic physical model of TPS is based on the bending behavior of a thin metal sheet that is constrained only by a sparse set of fixed points. In this situation the metal sheet takes the shape that minimizes the bending energy of the deformation. The radial basis function used in this approach is defined as [81]:

$$\varphi(r) = r^2 \ln r \quad (3.2)$$

where r is the Euclidean distance of an evaluated point to the center of the basis function in the x-y plane. Figure 3.9a shows the shape of function φ , which is a fundamental solution to the biharmonic equation $\Delta^2 \varphi = 0$.

An interpolation problem of a surface passing through a set of distinct control points $\{(x_i, y_i) : i = 1, 2, \dots, p\}$ with known height values $\{z_i : i = 1, 2, \dots, p\}$ is solved by finding a function F that satisfies

$$F(x_i, y_i) = z_i, \quad \forall i = 1, \dots, p \quad (3.3)$$

where the number of control points $p \geq 3$, two control points are not on the same position and the set of control points is not collinear. Such a solution can be found by minimizing the physical bending energy of the surface defined as:

$$I(F) = \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 F}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 F}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 F}{\partial y^2} \right)^2 \right) dx dy \quad (3.4)$$

It has been shown [81] that the function F can be expressed as a polynomial and a linear combination of radial basis functions located at the control points of the form

$$F(x, y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^p \omega_i \varphi(r_i(x, y)) \quad (3.5)$$

$$r_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (3.6)$$

where a_0, a_1 and a_2 are the coefficients of the planar term of the spline and ω_i is the coefficient of the i th spline term. The coefficients $\{\omega_i : i = 1, 2, \dots, p\}$ are further constrained by the conditions:

$$\sum_{i=1}^p \omega_i = 0, \quad \sum_{i=1}^p \omega_i x_i = 0, \quad \sum_{i=1}^p \omega_i y_i = 0 \quad (3.7)$$

The unknown coefficients a_0, a_1, a_2 and $\{\omega_i : i = 1, 2, \dots, p\}$ can be determined by solving the square linear system formed with the constraints given by Equation 3.3 and Equation 3.7. The corresponding matrix equation can be written in the the form

$$\left[\begin{array}{c|c} K & P \\ \hline P^T & O \end{array} \right] \cdot \left[\begin{array}{c} \vec{\omega} \\ \vec{a} \end{array} \right] = \left[\begin{array}{c} \vec{z} \\ \vec{o} \end{array} \right] \equiv M_{(p+3) \times (p+3)} \cdot \vec{x}_{(p+3) \times 1} = \vec{b}_{(p+3) \times 1} \quad (3.8)$$

where the submatrices K , P , O and the column vectors $\vec{\omega}$, \vec{a} , \vec{z} and \vec{o} are defined as [82]

$$K_{p \times p} = \begin{bmatrix} 0 & \varphi(r_{12}) & \cdots & \varphi(r_{1p}) \\ \varphi(r_{21}) & 0 & & \vdots \\ & & \ddots & \\ \vdots & \ddots & \ddots & \varphi(r_{(p-1)p}) \\ \varphi(r_{p1}) & \cdots & \varphi(r_{p(p-1)}) & 0 \end{bmatrix} \quad (3.9)$$

$$r_{ij} = \|[x_i y_i] - [x_j y_j]\| \quad (3.10)$$

$$P_{p \times 3} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_p & y_p \end{bmatrix} \quad (3.11)$$

$$O_{3 \times 3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.12)$$

$$\vec{\omega}_{p \times 1} = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_p \end{bmatrix}, \vec{a}_{3 \times 1} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \vec{z}_{p \times 1} = \begin{bmatrix} z_1 \\ \vdots \\ z_p \end{bmatrix}, \vec{o}_{3 \times 1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.13)$$

The matrix M is symmetric and nonsingular, therefore the control points uniquely define the function $F(x, y)$ in Equation 3.5 [83]. The solution of the equation system can be obtained using arbitrary solvers for linear systems. For a high number of control points optimized iterative numerical solving methods tend to show the best performance. An overview of different solution algorithms has been published by Powell [84].

The resulting coefficients allow the calculation of z values at arbitrary points of the x - y plane, by using Equation 3.5. The resulting surface smoothly interpolates between control points (x_i, y_i) . It asymptotically approaches the least squares plane through all control points outside the region of control points. Figure 3.9b shows the interpolated surface of a set of 4 control points. The bending energy is calculated by:

$$I(F) = \vec{\omega}^T K \vec{\omega} \quad (3.14)$$

It can be used as a measure of smoothness of the surface. By extending the definition of submatrix K (see Equation 3.9) with the regularization parameter λ it is possible to globally adjust the smoothness of the surface:

$$K_{ij} = \varphi(r_{ij}) + I_{ij}\lambda, \quad \lambda \geq 0 \quad (3.15)$$

where I is a unit matrix of size $p \times p$ and λ is the regularization parameter. A λ value of zero results in the standard TPS interpolation. Using values of $\lambda > 0$ results in a smoother surface which only approximates the control points and does not guarantee to pass through them.

Local methods

Since the results of the initial point selection are stored in a depthmap, it is possible to use image filtering methods to locally spread the depth values and complete the clipping surface. Neighborhood based image filtering techniques usually execute the following sequence of operations [85]:

1. Selection of a reference pixel $f(x, y)$ within an input image.
2. Execution of operations using only information of the reference pixel and pixels in its neighborhood to produce an output value.
3. Writing the determined value to an output image at the location of the reference pixel, $g(x, y)$.
4. Execution of steps 1-3 for every pixel of an input image.

Image filtering methods can be categorized into linear and non-linear methods depending on the operations performed. Linear filters compute the value of an output pixel by summing up the products of pixel values and mask coefficients within the selected neighborhood (e.g., mean filter). The mask coefficients are provided by a predefined filter kernel. Non-linear filters allow the use of arbitrary and more complex operations. They often operate on a sorted list of the local pixel values to compute the output pixel value (e.g., median filter). In case of the sparse initial point depthmap, linear filtering methods can not be applied directly, due to the fact that its empty regions initially do not contain depth values. This prevents the multiplication with a filter kernel if the neighborhood of a reference pixel contains an empty region. Replacing empty regions with depth values of zero causes them to directly influence the filtering results. In order to prevent this effect some linear filters can be separated into two filtering operations. The first filter is applied to the depthmap with empty regions marked as zero depths, while the second filter operates on a binary map defining empty and non-empty regions within the depthmap. The filtering result of the binary map is used to avoid the influence of empty regions on the final result. The following section shows this on the example of the mean filter.

Mean filter method

Low-pass filters are used to attenuate high-frequency components in an image resulting in a smoother transition of values between adjacent pixels. By extending a low-pass filter to work with sparse data, this feature can be exploited to interpolate a continuous surface from the initial point depthmap. The mean filter has been selected for testing within the proposed framework. The kernel of a mean filter is defined as

$$G_{n \times n} = \left(\frac{1}{n^2} \right)_{i \in \{1, \dots, n\}, j \in \{1, \dots, n\}} \quad (3.16)$$

$$G_{3 \times 3} = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \quad (3.17)$$

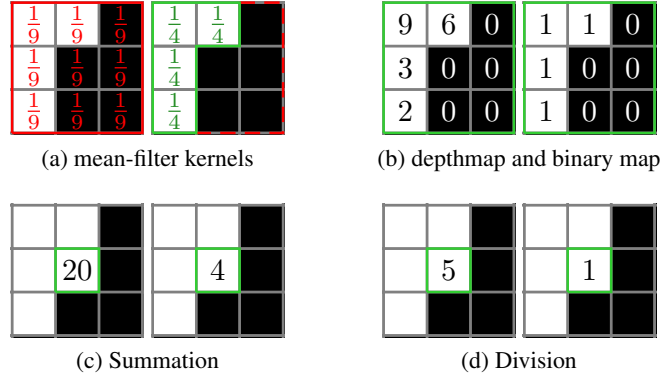


Figure 3.10: Example of the mean filter extension for sparse image data: (a) A standard 3×3 mean filter kernel (left) would include areas without existing depth values (black). The desired filter kernel is shown on the right. (b) The depthmap and binary map used for this example. (c) Summation of values within the 3×3 filter kernel on the depthmap and binary map. (d) Dividing the sum of depths by the number of existing depth values (sum of the binary map kernel) results in a depth value in a previously empty location of the depthmap.

where G is a filter kernel with an extent of n rows and columns. $G_{3 \times 3}$ is an example of a mean filter kernel where $n = 3$.

The linear mean filter presumes that all pixels of a neighborhood contain values and therefore the filter kernel for an average value calculation can be precomputed. To obtain a filter with similar behavior that works on sparse data, it has to be ensured that only non-empty pixels of a neighborhood contribute to the average calculation. This can be achieved by applying summation filters to the depthmap f and the corresponding binary map b and combining their results. In the binary map b non-empty regions are marked by the value one, empty regions have a value of zero in both maps. The applied operations can be formulated as

$$g(x, y) = \sum_{i=-n/2}^{n/2} \sum_{j=-n/2}^{n/2} f(x+i, y+j) \quad (3.18)$$

$$c(x, y) = \sum_{i=-n/2}^{n/2} \sum_{j=-n/2}^{n/2} b(x+i, y+j) \quad (3.19)$$

$$h(x, y) = \begin{cases} \frac{g(x, y)}{c(x, y)}, & \text{if } c(x, y) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.20)$$

$$d(x, y) = \begin{cases} 1, & \text{if } c(x, y) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.21)$$

where n defines the size of the filter kernel. g and c are the output images of the summation filters applied on f and b . h is the resulting depthmap and d is the updated binary map.

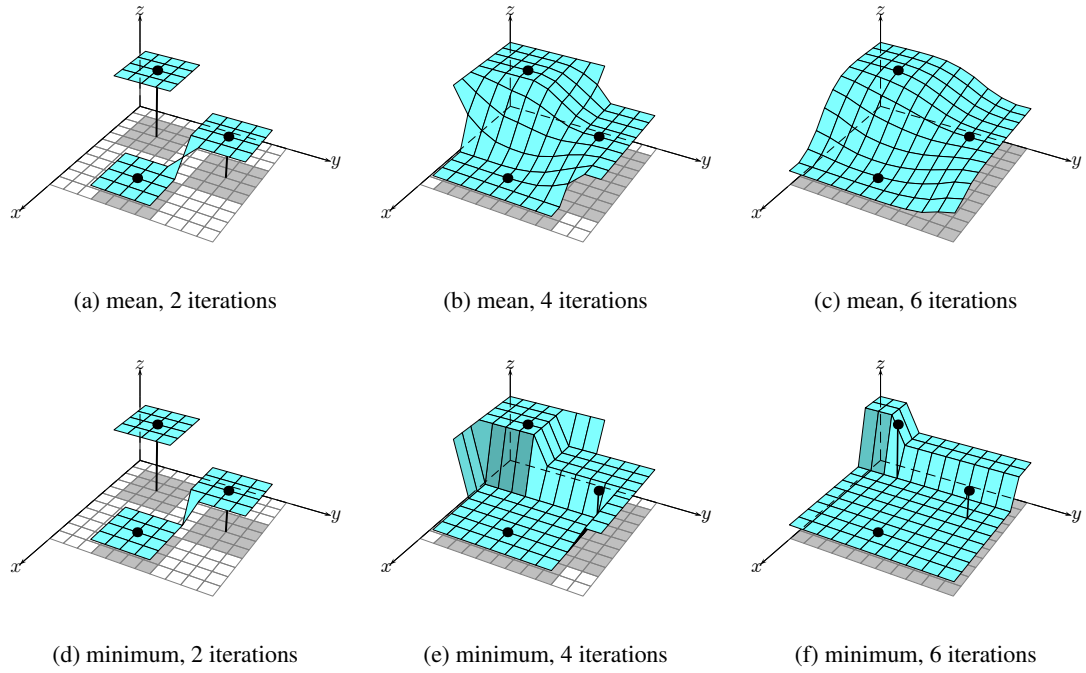


Figure 3.11: Results of local filtering methods after different iteration steps using a 3×3 filter kernel and the initial points $\{(3, 3, 5), (9, 5, 1), (5, 9, 3)\}$. Initial points are marked by black dots and reconstructed depth values (pixels) are shown on the intersections of the grid.

Using the mean filter method produces a low-pass filter effect on all areas of the depthmap with existing depth values as well as on areas bordering to existing values. Figure 3.10 illustrates this on a single border pixel. Repeated filtering of the result images eventually fills all empty pixels of the depthmap. As Figure 3.11a shows, areas with existing depth values (shown in light blue) are extended parallel to the x-y plane until they interact with another area. Successive iterations result in a smooth transition between the previously planar areas as seen in Figure 3.11b and 3.11c. The surface created by this method converges to a plane parallel to the xy-plane located at the average depth of the initial points.

Minimum filter method

In some cases the set of initial points contains outliers, which have negative impact on the reconstructed clipping surface. While the mean filter method generates a smooth clipping surface, it often takes many iterations to diminish the influence of even single outliers. At the same time this causes a loss of detail of the overall surface. By selecting the lowest depth value within the filter kernel the minimum filter is less prone to outliers with a high depth value. Unlike the mean filter, the minimum filter is a non-linear filter that compares all values contained within the filter kernel in order to return the lowest one. As the Figures 3.11d, 3.11e and 3.11f show, the resulting surface features hard edges which might lead to visual artifacts if used as starting region (see

Figure 5.3b). Repeated application of this method propagates the minimum value found in the depthmap and will rarely lead to a useful clipping surface. However a single application of this filter can be used as preprocessing step to eliminate outliers within an initial point depthmap. The resulting clipping surface is stored in the same data structure as its sparse input data and passed on to the rendering module.

3.4 Rendering

The rendering step incorporates the reconstructed clipping surface into the visualization. A basic DVR ray casting algorithm, which is also available in the Simu3D ultrasound firmware, is used in order to allow visual verification of the resulting images. The algorithm presented in Section 2.3 is extended to consider the depth information provided by the reconstructed clipping surface during the rendering process. Rays are started from the clipping surface instead of the first intersection of the volume-bounding box. Each ray is sampled at consecutive equidistant points and the results are classified by a transfer function. Following the capabilities of the Simu3D firmware, the proposed framework offers limited options for transfer function manipulation. The transparency transfer function can be defined by a windowing function with controllable lower and upper boundaries T_L and T_H . To remove the amniotic fluid from the rendering, the amniotic fluid threshold of the initial point selection step is used as lower boundary. The color of a sample point is obtained by multiplying the intensity at the sample location i_{src} with the color value C_{skin} , which represents skin color. Using these color and transparency values results in images similar to the ones of the reference US firmware. The classified values are blended together using front-to-back compositing with the over operator:

$$C_{src} = i_{src}C_{skin} \quad (3.22)$$

$$C_{dst} = C_{dst} + C_{src}\alpha_{src}(1 - \alpha_{dst}) \quad (3.23)$$

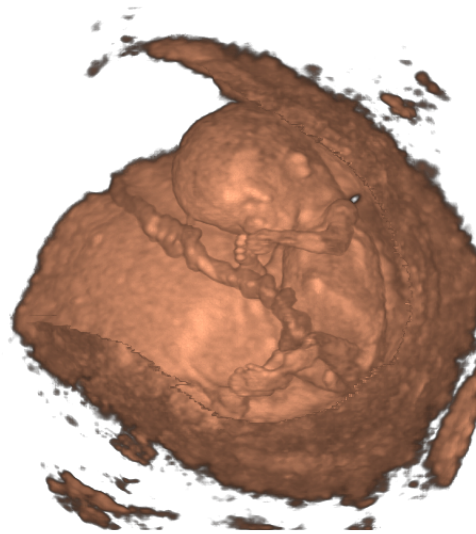
$$\alpha_{dst} = \alpha_{dst} + \alpha_{src}(1 - \alpha_{dst}) \quad (3.24)$$

where C_{dst} and α_{dst} are the color and transparency values accumulated along the ray up to the current sample point. i_{src} and α_{src} are the intensity and transparency values at the current sample position.

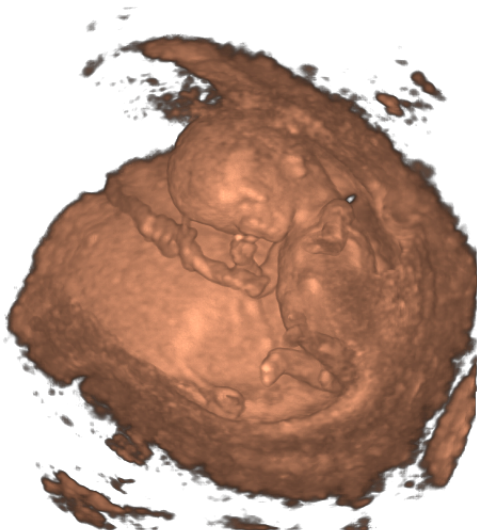
As explained, the clipping surface is a smooth interpolation between the sparse set of initial points. It is possible that the surface passes through the fetus in interpolated regions. Additionally, initial points that were incorrectly selected due to US artifacts can divert the clipping plane and cause similar adverse effects. As Figure 3.12b shows, this might create visual artifacts and incomplete visualizations of the fetus. Ghosting techniques can be utilized to diminish the effects of an incorrectly positioned clipping surface by visualizing areas in front of the surface with low opacity. This is realized by multiplying the transparency value at the sample point α_{src} with a depth-dependent opacity parameter o_{src} during compositing. The used opacity parameter is defined by following windowing function

$$o_{src}(d_p, d_{src}) = \begin{cases} 0 & , d_{src} \leq d_p - s \\ \frac{d_{src} - (d_p - s)}{w} & , d_p - s < d_{src} < d_p - (s + w) \\ 1 & , d_{src} \geq d_p - (s + w) \end{cases} \quad , 0 \leq o \leq 1 \quad (3.25)$$

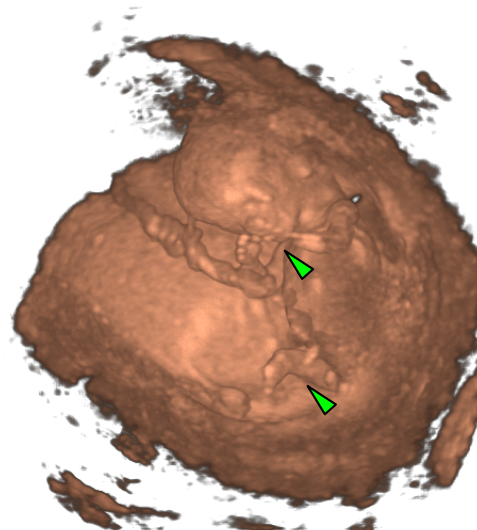
where d_p is the depth of the clipping surface, d_{src} is the sample depth, s is the offset to the starting depth relative to d_p and w is the width of the windowing function. The windowing function of the opacity parameter is illustrated on a ray profile in Figure 3.13. The example shows the depth d_p of the clipping surface to be located within fetal tissue. By using the opacity windowing function rendering starts in front of the tissue boundary with low opacity.



(a) Ground truth volume



(b) DVR - clipping



(c) DVR - ghosting

Figure 3.12: Application of ghosting in dataset "E04": (a) DVR visualization of the ground truth dataset. (b) DVR visualization using the clipping surface as ray starting point for DVR rendering. Due to an incorrect placement of the clipping surface the extremities of the fetus are partially removed. (c) DVR visualization using the presented ghosting method. Extremities that would have been clipped are now shown with low opacity (green arrows).

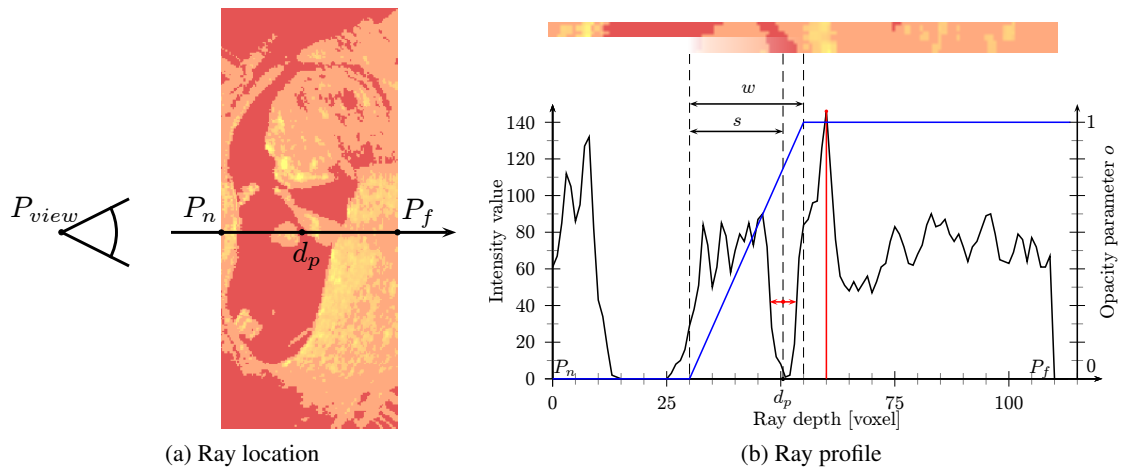


Figure 3.13: The windowing function of the opacity parameter: (a) Visualization of the ray position in a slice of dataset "E04". (b) The windowing function of the opacity parameter (blue) is defined by the starting offset s and the width w in relation to the clipping surface depth d_p . The example above the plot shows that tissue in front of the clipping surface is considered during rendering. In this case one arm of the fetus is partially recovered as seen in Figure 3.12.

Framework and Method Implementation

This chapter presents the design of the testing framework developed as part of this thesis, referred to as the *smartvis framework*. It covers implementation details of the framework itself and the smart visibility approaches discussed in Chapter 3. Section 4.1 provides an overview of the modular architecture of the smartvis framework and the basic features it provides. The datasets used for testing are described in Section 4.2. Section 4.3 introduces the modules used for the testing and evaluation of smart visibility methods.

4.1 Framework Overview

The smartvis framework is designed as a very general rendering framework that executes a defined sequence of processing modules. Figure 4.1 gives an overview of the framework's architecture. In order to mimic the minimum requirements of the Simu3D ultrasound firmware provided by GE Healthcare, the framework is based on Microsoft's DirectX 9 API. GPGPU functions are accessible in the framework through Nvidia's CUDA (Compute Unified Device Architecture) programming model. The framework provides following the functionalities:

- **Module execution**

The core concept of the smartvis framework is that all volume processing and rendering functions are implemented in modules. The framework executes a defined sequence of these modules as part of the rendering loop. This is realized through an abstract interface definition implemented by every module. As part of the module execution a collection based data-structure containing all input parameters is passed to a given module. After its execution the module adds its processing results to the data-structure before passing it to the subsequent module.

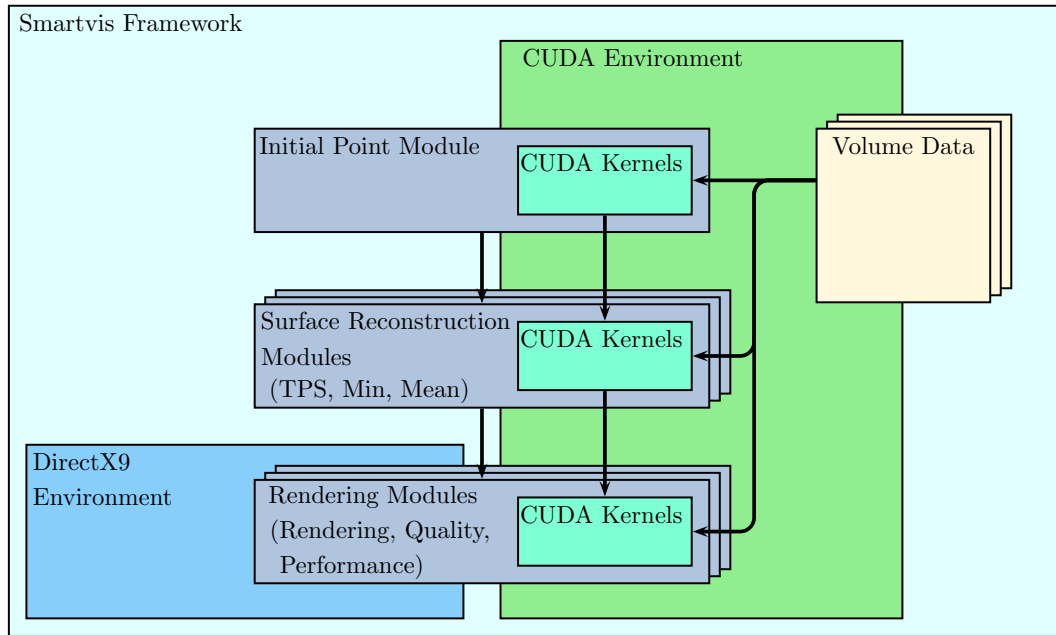


Figure 4.1: General architecture of the smartvis framework.

The smartvis algorithm presented in the previous chapter can be separated into three independent steps: initial point selection, surface reconstruction and rendering. Within the framework they are implemented as modules. Multiple module implementations of a smartvis algorithm step are possible. The framework supports the testing of arbitrary combinations of modules by simply switching the execution order.

- Data import

The test datasets provided by GE Healthcare consist of volume data in raw format, accompanied by MetaImage files containing basic information about the volume (described in detail in Section 4.2). The framework provides methods to load datasets into the device memory of the graphics card and makes them accessible to the modules.

- User input

Each of the implemented modules can define a number of user-controllable parameters. The framework provides direct access to them during the testing procedure through a set of simple GUI components (e.g., sliders, checkboxes and buttons). Mouse input to control camera rotation and distance are also handled by the framework.

Figure 4.2 shows the user interface of the framework during runtime. Three viewports with linked camera positions show the current test result, the reference volume for the tested dataset as well as the full dataset. Below the DVR visualizations each module maintains a column of GUI-controls that allow the manipulation of their respective algorithm parameters.

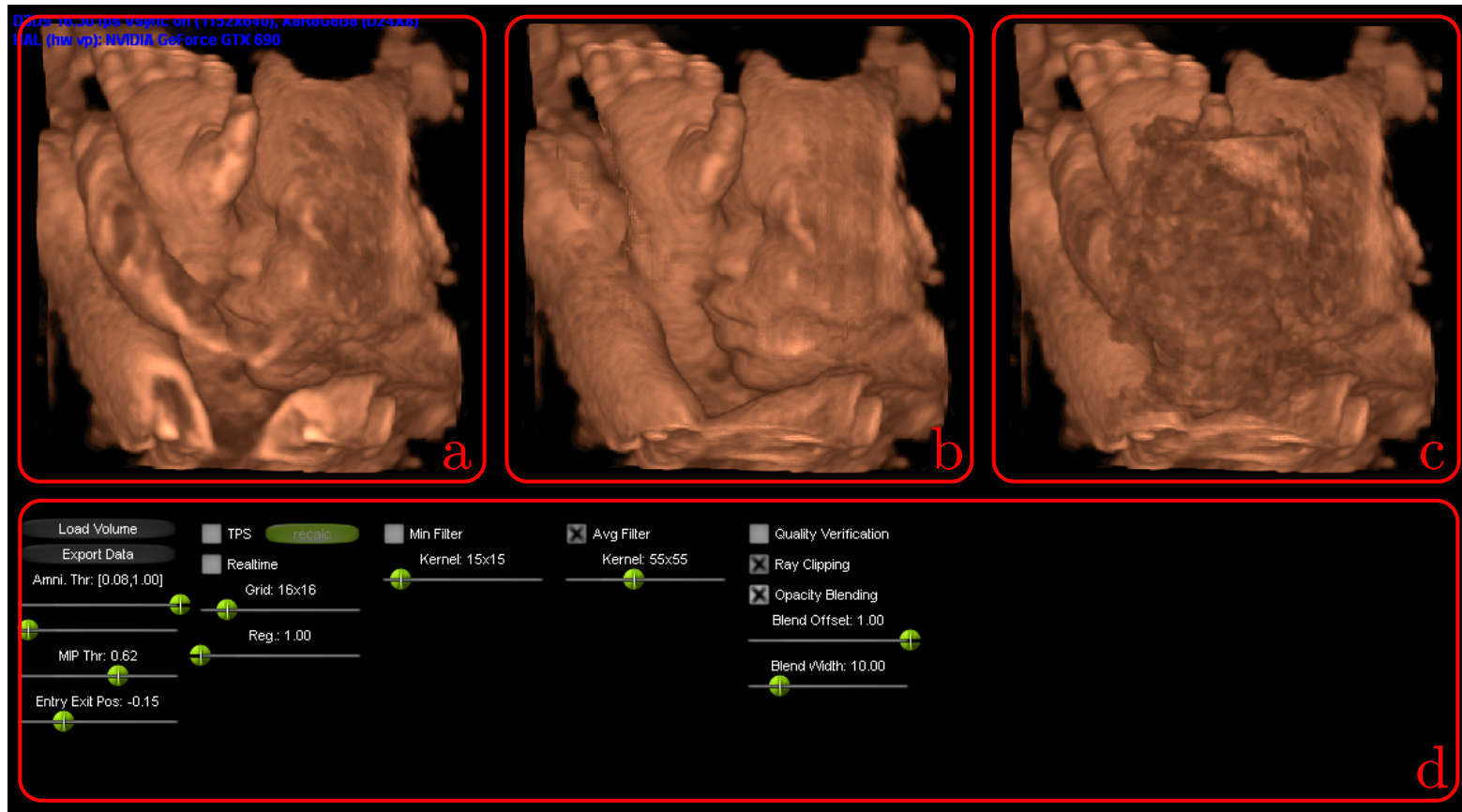


Figure 4.2: Basic user interface of the framework (a) Result of currently tested algorithm settings. (b) Ground-truth dataset. (c) Full dataset. (d) GUI-control area showing controls of 5 modules (initial point, TPS, mean filter, min filter, rendering (visual verification)).

4.2 Test Datasets

GE Healthcare provided anonymized volume datasets of independent 9 obstetric US examinations as test cases. Preprocessing of this data resulted in three volume datasets for each individual test case:

- Full volume dataset:

A full volume dataset contains the unedited result of a 3D ultrasound scan. Datasets of this type are used as input volumes for the evaluation of the smart visibility methods. Figure 4.3a shows an example of this dataset type.

- Reference volume dataset:

Reference datasets are provided by GE Healthcare for each test case. They contain volume data edited by domain experts using the built-in occlusion removal tools of the Simu3D US machine firmware. These removal methods include basic clipping using a resizable ROI box with a single deformable side and arbitrary volume cutting tools. Figure 4.3b shows the reference dataset of the previously shown full dataset.

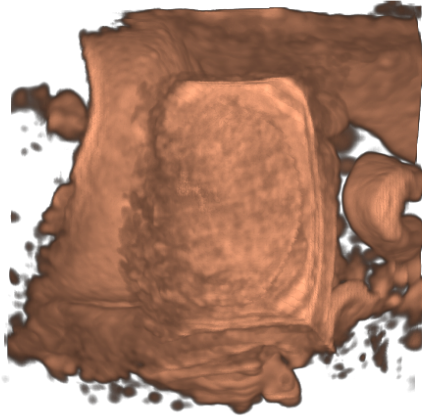
Reference volume datasets represent expert-defined occlusion removal with existing tools rather than ground-truth data of an unoccluded fetus. Occluders are still present within these datasets and in some cases parts of the fetus have been removed. The reference datasets can therefore not be used for an automatic verification of test results.

- Ground-truth dataset:

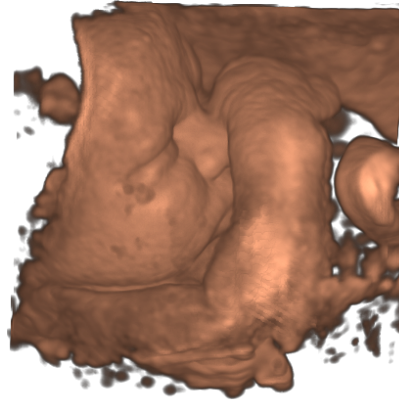
Ground-truth datasets are the result of manual slice-by-slice segmentation of full volume datasets. They were created as part of this thesis. The datasets contain the volume data after clipping with a clipping surface formed by correctly selected initial points for every ray. The definition in Section 3.2 implies that initial points only exist in regions where viewing rays pass through the amniotic sac. Since datasets are not completely occupied by the amniotic sac an additional label map has been created for each dataset. For a defined viewpoint with preset camera parameters the label map assigns one of three states to all pixels of the result image. These states are: passing through the fetus, passing through the amniotic sac but not the fetus and not passing through the amniotic sac. The ground-truth volumes can be used in combination with the label map to verify the quality of the results. Figure 4.3c and Figure 4.3d show a ground-truth dataset and the corresponding label map.

4.3 Modules

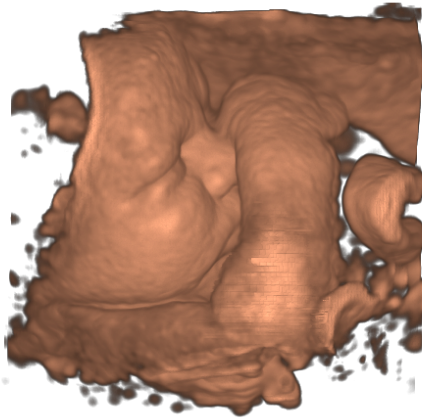
As explained in the framework overview, modules contain the actual implementation of the separate steps of the smartvis algorithm. Each module consists of initialization functions to register the GUI elements of the module and to allocate the required memory. During the rendering loop of the smartvis framework the execute function of activated modules is called. Modules process their input data using CUDA function kernels. Texture results are returned as pointers



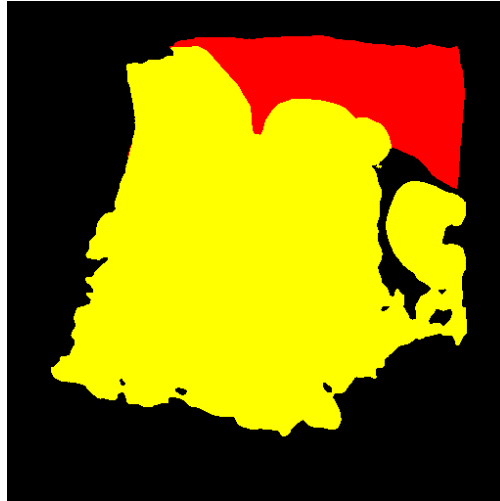
(a) Full volume



(b) GE reference volume



(c) Ground-truth volume



(d) Label map

Figure 4.3: (a), (b) and (c) show DVR visualizations of the available input dataset types for test case "E04". (d) The corresponding label map, yellow marks rays passing the fetus and red marks rays passing just the amniotic sac.

to the device memory to minimize the required data transfer. Figure 4.1 shows the basic module execution sequence, each module is explained in detail in the following sections.

Initial Point Module

The initial point module implements the method of selecting a set of valid initial points presented in Section 3.2. The points are computed using ray-profile analysis as part of a basic ray casting algorithm. The implementation utilizes the parallel computing capabilities of the CUDA environment. Rays are cast from the view point through the center of each pixel of the output depthmap. The traversal of rays through the volume is implemented in parallel CUDA kernels. The pseudo code of these kernels is outlined in Algorithm 4.1.

A kernel is initialized by calculating the near and far intersection points of its ray with the bounding box of the volume P_n and P_f . Afterwards boundaries for two intervals are initialized with the depth value P_n of the rays entry into the volume. Finally, two variables used to store intensity values are initialized with zero. During the traversal the lastIntensity variable stores the intensity value of the previously evaluated sample position. By comparing the lastIntensity and the intensity of the current sample with the threshold for amniotic fluid T_L entry and exit points of tissue regions can be identified. The lastExit and lastEntry variables store depth values and are updated whenever the ray leaves or enters these tissue areas. The maxIntensity variable is updated whenever an intensity higher than the current maximum is encountered. In this case the lastExit and lastEntry values are copied into the maxIntExit and maxIntEntry variables. After the traversal is complete the initial point interval in front of the depth of the maximum intensity peak is defined by $[\text{maxIntExit}, \text{maxIntEntry}]$. The maxIntensity variable is used to check if the ray passed through bone tissue, using the threshold T_B . Afterwards the initial point depth d_p for the ray is calculated using Equation 3.1. The resulting depth is stored in the depthmap. Additionally, the location of the initial point is stored in a binary map of the same resolution for further processing by a surface reconstruction module. This map is referred to as initial point status map in the following modules.

Surface Reconstruction Modules

Surface Reconstruction Modules implement the surface reconstruction methods presented in Section 3.3. The framework contains modules for all three proposed surface reconstruction algorithms.

TPS Module

The TPS module utilizes the Thin-Plate Splines, as presented in Section 3.3, to reconstruct a smooth surface from the sparse initial point set. In order to be processed as linear equations system the initial points of the depthmap have to be rearranged into a list data structure. The coordinates of the initial points are stored alongside their depth values. The module calculates the matrix M of the TPS from this list and solves the linear equation system using LU decomposition. LU decomposition factorizes the equation matrix M into a lower triangle matrix L and

input : volume, \overrightarrow{ray} , T_L , T_B , q
output: Initial point d_p , Initial point status b_p

```

1  $P_n, P_f \leftarrow \text{intersectBox}(\overrightarrow{ray}, \text{volume})$ ;
2  $\text{lastEntry}, \text{lastExit} \leftarrow P_n$ ;
3  $\text{maxIntEntry}, \text{maxIntExit} \leftarrow P_n$ ;
4  $\text{lastIntensity}, \text{maxIntensity} \leftarrow 0$ ;
5 for  $d \leftarrow P_n$  to  $P_f$  do
6    $\text{sample} \leftarrow \text{tex3D}(d, \text{volume})$ ;
7   if  $\text{sample} > T_L$  &  $\text{lastIntensity} < T_L$  then  $\text{lastEntry} \leftarrow d$ ;
8   if  $\text{sample} < T_L$  &  $\text{lastIntensity} > T_L$  then  $\text{lastExit} \leftarrow d$ ;
9   if  $\text{sample} > \text{maxIntensity}$  then
10     $\text{maxIntensity} \leftarrow \text{sample}$ ;
11     $\text{maxIntEntry} \leftarrow \text{lastEntry}$ ;
12     $\text{maxIntExit} \leftarrow \text{lastExit}$ ;
13  end
14 end
15 if  $\text{maxIntensity} > T_B$  then
16    $d_p \leftarrow q(\text{maxIntExit} - \text{maxIntEntry}) + (\text{maxIntEntry} - P_n)$ ;
17    $b_p \leftarrow 1$ ;
18 end
19 else
20    $d_p \leftarrow 0$ ;
21    $b_p \leftarrow 0$ ;
22 end

```

Algorithm 4.1: Initial point kernel: The function returns the initial point d_p and its binary point status b_p for a single ray based on the bone threshold T_B , the amniotic fluid threshold T_L and the offset coefficient q

an upper triangle matrix U :

$$M = LU \tag{4.1}$$

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_{n1} & \cdots & l_{nn-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_{12} & \cdots & \cdots & u_{1n} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix} \tag{4.2}$$

After the decomposition the linear equation system $Mx = b$ can be rewritten as:

$$LUx = b \tag{4.3}$$

The system solution of the system can now be obtained by solving the two linear equation systems shown in Equation 4.4. Since L and U are triangle matrices they can be solved by

variable substitution.

$$\begin{aligned} Ly &= b \\ Ux &= y \end{aligned} \tag{4.4}$$

The TPS module is utilizing the Doolittle LU decomposition algorithm [86] to calculate the triangle matrices. This algorithm solves the decomposition problem for the first column and row and is then recursively applied to the remaining submatrix. Afterwards the equation system is solved by using basic variable substitution on the resulting matrices.

Even with a high bone intensity threshold T_B the number of initial points is often higher than 200. The matrix of a TPS linear equation system is symmetrical but dense which results in a low performance in the current implementation of the module, due to the large number of unknowns. To address this problem a method to reduce the number of initial points is introduced. The depthmap is subdivided into a grid. For each grid cell an average initial point depth and position is calculated from all initial points it contains. Algorithm 4.2 outlines the procedure executed at every grid cell. The method returns a single merged initial point depth d_c and position p_c for each cell that contains at least one initial point. It therefore reduces the size of the equation system to be solved. Additionally, the resulting merged initial points have a higher distance to each other. This reduces the chance of the occurrence of artifacts which are explained in detail in Section 5.2.

After solving the linear equation system, the module evaluates Equation 3.5 at every pixel of the depthmap. The resulting reconstructed depthmap is passed to the rendering module alongside with the unchanged initial point status map.

```

input : depthMap, initialPointStatusMap, grid cell  $c$ 
output: Merged initial point depth  $d_c$ , Merged initial point position  $p_c$ 

1 depthSum, pixelCount  $\leftarrow$  0;
2 positionSum  $\leftarrow$  [0, 0];
3 foreach pixel  $p$  in  $c$  do
4   if tex2D( $p$ , initialPointStatusMap) = 1 then
5     depthSum  $\leftarrow$  depthSum + tex2D( $p$ , depthMap);
6     positionSum  $\leftarrow$  positionSum +  $p.xy$ ;
7     pixelCount  $\leftarrow$  pixelCount + 1;
8   end
9 end
10 if pixelCount > 0 then
11    $d_m \leftarrow$  depthSum / pixelCount;
12    $p_m \leftarrow$  positionSum / pixelCount;
13 end

```

Algorithm 4.2: Initial point reduction by merging initial points within a cell of the depthmap.

Mean Filter Module

The mean filter module implements the mean filter method presented in Section 3.3. The mean filter is a separable 2D filter, therefore the filtering can be achieved by subsequently applying a horizontal and vertical 1D filter to the input image. This reduces the computation time for large filter sizes. The filtering process uses three temporary map data structures to store the intermediate filtering results. The surface point status map is a binary map that stores which pixels have already been reconstructed and it is initialized with the initial point status map. This additional map is used in order to distinguish between initial and reconstructed points. The depth and weighted pixel sum maps are used to store the filtering results of the horizontal filter and are processed by the vertical filter kernel.

An iterative application of the mean filter to the depthmap, as explained in Section 3.3 and shown in Figure 3.11, extends all existing initial point areas in the depthmap by a border half as wide as the filter kernel. Repeated iterative applications will cause the depthmap to converge to the average depth of all initial points. In areas that originally contained initial points that might be an unintended behavior. Initial points are assumed to be located at a correct depth. Changing their depths could move them into regions occupied by tissue. However, the filtering of initial point depth values is still required in order to reduce the impact of outliers and to obtain a smooth surface in general. To address this problem a weight coefficient ω is introduced and applied to all points that were not present in the initial point depthmap. Using a weight $\omega < 1$ effectively slows down the convergence to the global average depth and preserves surface detail around the initial points.

The number of iterations required to reconstruct the entire clipping surface is depending on the distribution of initial points in the input depth map. A global variable *gcomplete* is used to stop the module once every missing value in the depthmap has been filled. This functionality is explained in detail below.

The horizontal and vertical filter functions are outlined in Algorithm 4.3. The horizontal filter sums up the depth values and the weighted pixels within the horizontal extent of the filter kernel. The coefficient ω is applied to pixels based on the initial point status map. The resulting sums are stored in the depth and weighted pixel sum maps. The vertical filter adds up the horizontal filter results and divides the summed up depth by the weighted sum of encountered pixels to obtain an averaged depth value. Additionally, it updates the surface point status map. If no surface points are present in the extent of the filter kernel the global variable *gcomplete* is set to *false* to indicate that the depthmap is not yet completely filled. This variable is initialized with the value *true* before every iterative application of the mean filter and can be accessed by all parallel CUDA kernels.

If the mean filter finishes with *gcomplete* set to *true* the surface reconstruction is complete. The resulting depthmap and the unchanged initial point status map are then passed to the rendering module.

input : depthMap, initialPointStatusMap, surfacePointStatusMap, weight ω
output: depthSumMap, weightedPixelSumMap

```

1 depthSum, weightedPixelSum  $\leftarrow$  0;
2 foreach pixel  $p$  in horizontalKernel do
3   if tex2D( $p$ , initialPointStatusMap) = 1 then
4     depthSum  $\leftarrow$  depthSum + tex2D( $p$ , depthMap);
5     weightedPixelSum  $\leftarrow$  weightedPixelSum + 1;
6   else if tex2D( $p$ , surfacePointStatusMap) = 1 then
7     depthSum  $\leftarrow$  depthSum + ( $\omega \cdot$  tex2D( $p$ , depthMap));
8     weightedPixelSum  $\leftarrow$  weightedPixelSum +  $\omega$ ;
9   end
10 end
11 if weightedPixelSum > 0 then
12   tex2D( $p$ , depthSumMap)  $\leftarrow$  depthSum;
13   tex2D( $p$ , weightedPixelSumMap)  $\leftarrow$  weightedPixelSum;
14 end

```

(a) Mean filter horizontal kernel function

input : depthSumMap, weightedPixelSumMap
output: depthMap, surfacePointStatusMap

```

1 depthSum, weightedPixelSum  $\leftarrow$  0;
2 foreach pixel  $p$  in verticalKernel do
3   if tex2D( $p$ , weightedPixelSumMap) > 0 then
4     depthSum  $\leftarrow$  depthSum + tex2D( $p$ , depthSumMap);
5     weightedPixelSum  $\leftarrow$  weightedPixelSum + tex2D( $p$ ,
6       weightedPixelSumMap);
7   end
8   if weightedPixelSum > 0 then
9     tex2D( $p$ , depthMap)  $\leftarrow$  depthSum / weightedPixelSum;
10    tex2D( $p$ , surfacePointStatusMap)  $\leftarrow$  1;
11  else
12     $g_{complete} \leftarrow false$ ;
13  end

```

(b) Mean filter vertical kernel function

Algorithm 4.3: Mean filter kernel functions: The consecutive execution of both functions updates one pixel of the depthmap and the surface point status map based on kernel size.

Minimum-Filter Module

The minimum filter module implements the minimum filter method presented in Section 3.3. It is implemented similar to the mean filter module by using separate horizontal and vertical filter kernels. Like the mean filter module, the minimum filter module uses a surface point status map to preserve the initial point status map it receives as input. Additionally, a temporary depth map and an temporary point status map are used to store the intermediate results between the horizontal and vertical kernel executions. The temporary map data structures are introduced for the same reasons as in the mean filter module. The minimum filter is iteratively applied until the global $g_{complete}$ variable indicates that the surface has been completely reconstructed.

The horizontal and vertical kernel functions used are outlined in Algorithm 4.4. Apart from the check for the termination condition both kernel functions have the same functionality. They iterate over all pixels within their respective filter kernel and store the existence of depth values as well as the smallest encountered value. The $g_{complete}$ variable is set to *true* before every iterative application of the minimum filter and is used to stop further iterations once the depthmap has been completely reconstructed. After the reconstruction is completed the resulting depthmap is passed to the rendering module alongside with the unchanged initial point status map.

Rendering Modules

Rendering modules provide means to view and compare the results of the smart visibility methods presented in Chapter 3 to visualizations of the corresponding full, reference and ground-truth volumes. Result images for visual verification are generated using ray casting DVR methods that incorporate the clipping surface into the visualization. In addition to the visual verification specialized rendering modules for detailed quality and performance analysis are available.

Visual verification module

The visual verification module is the standard rendering module used in the smartvis framework. It generates a side by side view of three DVR images of a given test case. The result of a tested smart visibility method is shown next to images of the full, reference or ground-truth volumes. Figure 4.2 presents an example of this image arrangement. This rendering module is used to test different parameter settings for an evaluated smart visibility method and allows the free exploration of the parameter space. It includes an export functionality for all rendered images.

The module implements the ray casting DVR algorithm described in Section 3.4. In the rendering algorithm the clipping surface is used in combination with the depth-dependent opacity coefficient o . The rays are started at the beginning of the windowing function defined by the offset to the clipping surface s and the width w . Both parameters can be manipulated during the test execution. Figure 4.4 shows the results of a smart visibility method using different windowing function settings.

Performance testing module

The performance testing module uses the same rendering method as the previously presented visual verification module. It only renders the result of the tested smart visibility method and is

input : depthMap, surfacePointStatusMap
output: temporaryDepthMap, temporaryPointStatusMap

```

1 minDepth  $\leftarrow \infty$ ;
2 pointStatus  $\leftarrow 0$ ;
3 foreach pixel  $p$  in verticalKernel do
4   if tex2D( $p$ , surfacePointStatusMap) > 0 then
5     minDepth  $\leftarrow \min(\text{minDepth}, \text{tex2D}(p, \text{depthMap}))$ ;
6     pointStatus  $\leftarrow 1$ ;
7   end
8 end
9 if pointStatus > 0 then
10   tex2D( $p$ , temporaryDepthMap)  $\leftarrow$  minDepth;
11   tex2D( $p$ , temporaryPointStatusMap)  $\leftarrow 1$ ;
12 end

```

(a) Minimum filter horizontal kernel function

input : temporaryDepthMap, temporaryPointStatusMap
output: depthMap, surfacePointStatusMap

```

1 minDepth  $\leftarrow \infty$ ;
2 pointStatus  $\leftarrow 0$ ;
3 foreach pixel  $p$  in verticalKernel do
4   if tex2D( $p$ , temporaryPointStatusMap) > 0 then
5     minDepth  $\leftarrow \min(\text{minDepth}, \text{tex2D}(p, \text{temporaryDepthMap}))$ ;
6     pointStatus  $\leftarrow 1$ ;
7   end
8 end
9 if pointStatus > 0 then
10   tex2D( $p$ , depthMap)  $\leftarrow$  minDepth;
11   tex2D( $p$ , surfacePointStatusMap)  $\leftarrow 1$ ;
12 else
13    $g_{complete} \leftarrow false$ 
14 end

```

(b) Minimum filter vertical kernel function

Algorithm 4.4: Minimum filter kernel functions: The consecutive execution of both functions updates one pixel of the depthmap and the surface point status map based on kernel size.

used during performance measurements.

Quality verification module

The quality verification module is used to analyze the quality of the occlusion removal effect of a tested smart visibility method. As explained in Section 3.2 multiple correct solutions for the placement of the clipping surface may exist. Therefore a direct comparison of resulting clipping surface shapes and locations with the surfaces used in the corresponding ground-truth datasets is not effective. Instead, the quality is determined by comparing the DVR rendering results. Rays that enter regions of tissue usually reach their full opacity after a few samples and are terminated by the ray casting algorithm. Rays started from different but correctly placed clipping surfaces are therefore terminated at a similar depth. This effect can be used to quantify the differences of smart visibility results to the ground-truth volume.

The quality verification module uses the same rendering method as the visual verification module, but generates depthmaps storing ray termination depths. It creates depthmaps for the evaluated smart visibility method and the ground-truth dataset using preset viewpoint and camera parameters. The resulting depthmaps are combined with the label map presented in Section 4.2 and used to calculate the error distance e in regions of the fetus and the amniotic sac. The error distances are then used to calculate the mean absolute error $\overline{e_{abs}}$:

$$e_i = g_i - d_i \quad (4.5)$$

$$\overline{e_{abs}} = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (4.6)$$

where g_i is the value of the ground-truth depthmap and d_i is the corresponding depth of the smart visibility method. The determined errors distances can be classified into two categories. A positive value of e_i means, the ray of the smart visibility rendering has reached its maximum opacity at a depth closer to the viewpoint than the corresponding ray in the ground-truth rendering. This suggests that the ray passed through occluding tissue areas. Consequently, a negative value of e_i means that the ray of the smart visibility rendering ignored tissue areas that were present in the ground-truth volume. This indicates that the clipping surface was placed inside or behind the fetal tissue at the position of the ray. To quantify the occurrence of these two error

types the mean positive error $\overline{e_{pos}}$ and negative error $\overline{e_{neg}}$ is calculated.

$$e_{pos}(i) = \begin{cases} e_i, & \text{if } e_i > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.7)$$

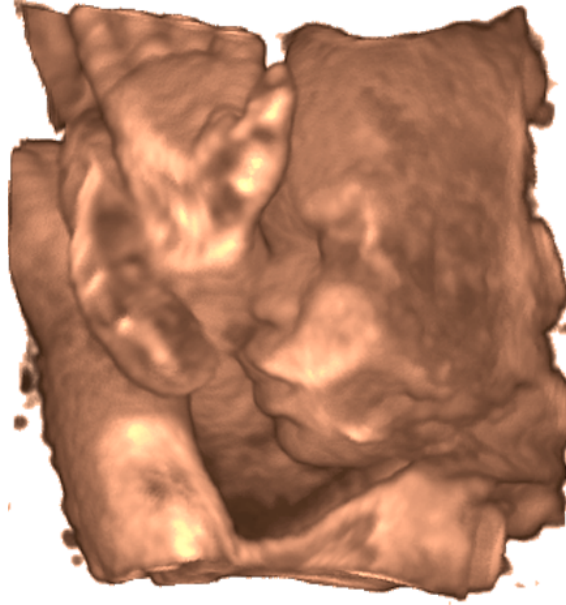
$$\overline{e_{pos}} = \frac{1}{n_{pos}} \sum_{i=1}^n e_{pos}(i) \quad (4.8)$$

$$e_{neg}(i) = \begin{cases} e_i, & \text{if } e_i < 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.9)$$

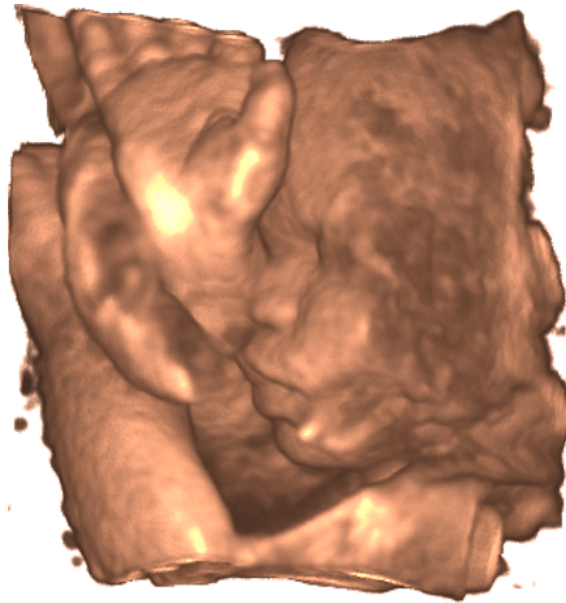
$$\overline{e_{neg}} = \frac{1}{n_{neg}} \sum_{i=1}^n |e_{neg}(i)| \quad (4.10)$$

Where n_{pos} is the number of positive and n_{neg} is the number of negative values of e_i .

The module exports the calculated error metrics as text files for further processing. Additionally it includes methods to change the input parameters of other modules at the end of each execution, which simplifies the analysis of the available parameter space. The quality verification module is used to obtain the quality measurements presented in the next chapter.



(a) Rendering from the clipping surface $s = 0, w = 0$



(b) Rendering with windowing function $s = 10, w = 10$

Figure 4.4: Rendering of dataset "M02" using the windowing function of the visual verification module. The dataset contains the head of a fetus with both arms in front of it. (a) Ray casting with rays starting directly from the clipping surface by using the windowing function $s = 0, w = 0$. (b) Ray casting using the windowing function $s = 10, w = 10$.

Method Evaluation

This chapter presents the evaluation results of smart visibility methods obtained by using the smartvis framework. Nine test cases provided by GE Healthcare were used to test the presented methods with different parameter settings. The evaluation process was divided into three stages:

- Test case preparation

In order to obtain quantifiable quality measurements, the datasets of the test cases had to be preprocessed. This resulted in ground-truth volume datasets and corresponding region label maps for each test case.

- Module evaluation

In this stage the results of different module combinations and module parameter setting were evaluated. The three rendering modules presented in the previous chapter were used to perform the tests. The main focus was the selection of the most promising smart visibility method from the set of proposed approaches. The primary selection criterion was the quality of occlusion removal by using the reconstructed clipping surface. Additionally the general performance characteristics of the methods were taken into consideration. Both criteria were analyzed together with domain experts from GE Healthcare.

- Plugin evaluation

The module combination selected in the previous phase was integrated into a state-of-the-art ultrasound firmware as plugin. The results obtained by using the firmware plugin were evaluated for differences to the smartvis framework in quality and performance caused by the different program environment.

5.1 Test case preparation

As explained in Section 4.2, the reference volume datasets provided by GE Healthcare were obtained using the built-in occlusion removal methods of a current ultrasound firmware. In

some test cases, the reference volume still contained a considerable number of occluders and/or removed parts of the fetal tissue. This greatly reduces the significance of the comparison of smart visibility method results to the reference volumes. To address this problem, ground-truth datasets have been created for each test case. A ground-truth dataset is defined as the result of clipping a full volume dataset with a correctly placed clipping surface. The transducer position was selected as viewpoint for this surface definition and an orthographic camera model was used.

The ground-truth datasets were manually created by defining a correct clipping surface placement for every slice of the corresponding full volume datasets. In the process, occluders in front of the clipping surface were removed from the dataset. The open source software package 3DSlicer was used to process the volume data.

In order to focus the evaluation to image regions occupied by the amniotic sac a label map has been created for each dataset. Using the defined viewpoint of the ground-truth volume and preset camera parameters the label map assigns one of three states to all pixels of the result image. These states are: passing through the fetus, passing through the amniotic sac but not the fetus and not passing through the amniotic sac. The created ground-truth volumes and label maps are used by the quality verification module to calculate error measurements. An overview of all dataset types used during the evaluation is given in Section 4.2.

5.2 Module Evaluation

The smartvis framework was developed to evaluate smart visibility methods by allowing a quick assessment of module combinations and parameter settings. One of the final goals of this work is, to test smart visibility methods within actual ultrasound firmware. GE Healthcare agreed to provide a testing environment under the condition that the tested methods were approved by in-house domain experts. The evaluation focused mainly on the three surface reconstruction modules. These methods were examined for:

- Quality of occlusion removal

The occlusion removal quality of a smart visibility method is defined by the amount of remaining occluding tissue and incorrectly removed fetal tissue in the visualization. A visual comparison between DVR images of the smart visibility results and the corresponding ground-truth dataset provided immediate insight about the occlusion removal quality. Additionally, the quality verification module was used to quantify the deviation of the test results from the ground-truth volumes.

- Performance

While the modules were not specially optimized towards high frame rates, a general estimation of the possible performance impact of regular DVR rendering was included into the evaluation. A frame rate of at least 20 frames per second was targeted.

The settings of the rendering modules during the evaluation were chosen based on the settings found in the Simu3D ultrasound machine firmware. The result renderings were obtained by

using an orthographic camera viewing the volume from the position of the US transducer. The results were rendered to images with a resolution of 512×512 pixels. The camera viewport was adjusted to show the volume data to the greatest possible extent in the resulting visualization. The evaluation process yielded the following results:

TPS Module

The Thin-Plate Spline approach is a widely used surface reconstruction technique. Initial tests of the module revealed that it is not the optimal method to complete the sparse depthmap produced by the initial point module. The examination identified the following problems:

- Distribution of initial points

The TPS method provides best results if used on a set of initial points well distributed on the depthmap. However, densely placed initial points with significant depth differences can cause an "overshooting" effect of the reconstructed surface. As explained in Section 3.2, the sparse depthmap provided as input is obtained by thresholding the MIP. In general, this results in several areas of adjacent initial points. In some cases, the depth differences of adjacent initial points lead to undesired interpolation behavior of the reconstructed surface. Figure 5.1a shows this problem at the region of the surface marked by the green arrow. In the example, the clipping surface is reconstructed in a wide arch between two sets of densely packed initial points. The depth values in this reconstructed surface region are considerably higher than those of the original initial points. Figure 5.2a shows the resulting visualization. Large parts of the fetus' leg are removed due to an incorrect placement of the clipping surface.

- Number of initial points

While the initial point depthmap is in general very sparsely filled, the number of initial points often exceeds 200 points, even for high bone intensity thresholds T_B . This has a significant impact on the dimensionality of the underlying linear equation system, since the size of the coefficients' matrix is directly dependent on the number of initial points.

The implementation using all initial points available on the depthmap required several seconds of computation time in all available test cases using the hardware setup listed in Table 5.5. Merging adjacent initial points using the method described in Section 3.3 with a 8×8 grid reduces the number of initial points to 64 or less. Additionally the resolution of the depthmap, storing the reconstruction results, has been reduced to 32×32 . This reduced the number of times Equation 3.2 has to be evaluated. Through these improvements real-time performance could be achieved. Due to the lower number of initial points and their higher distance toward each other, the interpolation problem mentioned before is mitigated. Figure 5.2c shows that the surface resulting from fewer initial points does not exhibit the unwanted interpolation behavior. Due to the merging process, the number of initial points used during surface reconstruction is limited. The threshold T_B can therefore be set to lower values without reducing the overall performance. In general, a lower

T_B value results in more initial points with a wider distribution on the depthmap. However, the merging of initial points reduces the surface detail and can in some situations emphasize outliers.

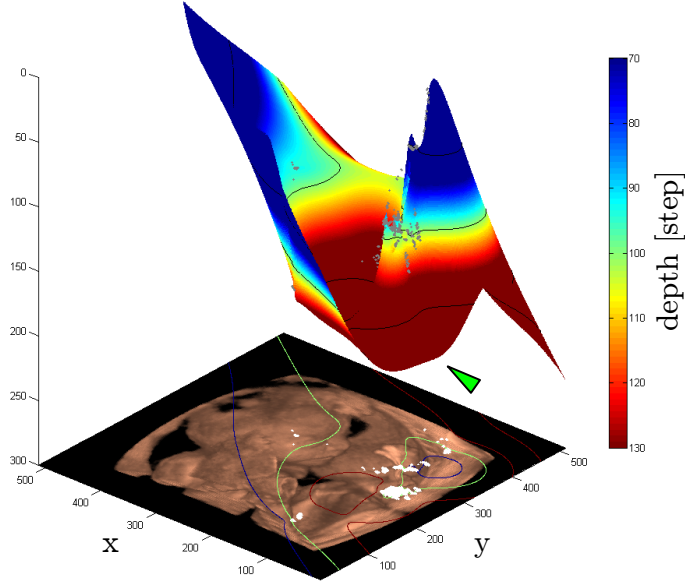
- Extrapolation behavior

One of the features of the TPS method is that the resulting surface converges to the least-squares plane fitted through all control-points in regions outside the convex hull of the control-point set. In worst case situations this least-squares plane can be significantly tilted away from the image plane. This can cause incorrect placement in extrapolated regions of the reconstructed surface. Both surfaces in Figure 5.1 exhibit this effect.

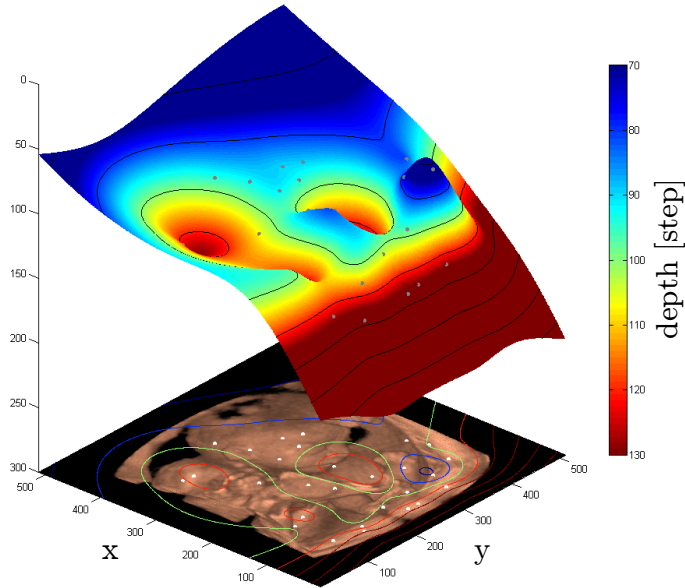
- Performance

During the evaluation the performance of the TPS module was in the range of 4 frames per second. Performance improvements of the module, by a better utilization of the graphics hardware are possible. However, the overall quality of smart visibility results using the TPS method were unconvincing to the domain experts of GE Healthcare. Therefore, the development of the TPS module was not further pursued in favor of the local surface reconstruction modules.

Result images of the smart visibility algorithm using TPS for surface reconstruction are shown in Figure 5.2. Figure 5.2a uses the bone threshold $T_B = 0.88$ resulting in 1401 initial points. The previously explained problems lead to an unsatisfying surface reconstruction result. Large portions of the fetus' legs are omitted while occluders in the shoulder region remain. Figure 5.2c shows the results with a lower bone threshold of $T_B = 0.74$ and merged initial points using an 8×8 grid. As shown in Figure 5.1b the resulting 28 initial points are well distributed over the image plane and the smart visibility result shows an improved occlusion removal quality.

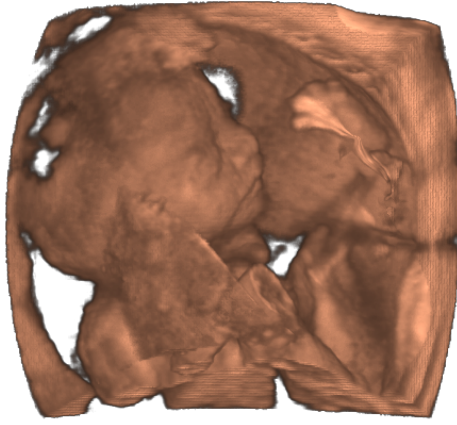


(a) TPS, 1401 initial points, $T_B = 0.88$

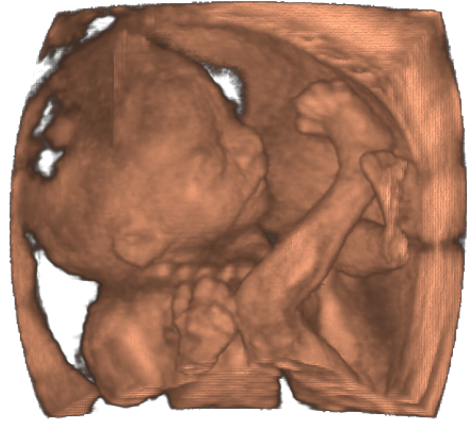


(b) TPS, 28 initial points, $T_B = 0.74$

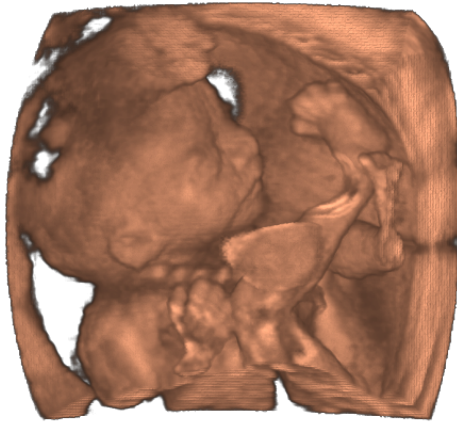
Figure 5.1: Visualizations of clipping surfaces reconstructed by the TPS method in test case "M01". (a) Clipping surface reconstructed from 1401 initial points with $T_B = 0.88$. The green arrow marks a region of unwanted interpolation behavior. Extrapolated surface regions are strongly tilted towards the image plane. (b) Clipping surface reconstructed from 28 merged initial points with $T_B = 0.74$. The lower threshold T_B increases the number of initial points. The merged initial points are distributed widely over the image plane.



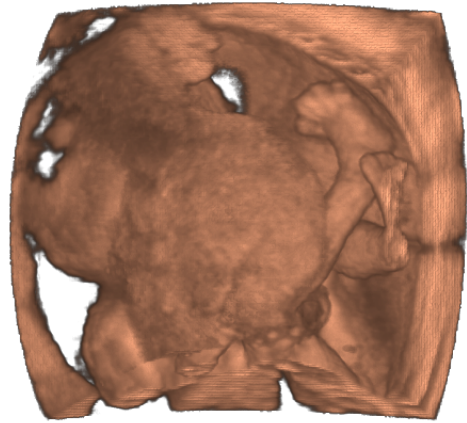
(a) TPS, 1401 initial points, $T_B = 0.88$



(b) Ground-truth dataset



(c) TPS, 28 initial points, $T_B = 0.74$



(d) Full dataset

Figure 5.2: Result images of the smart visibility method using TPS surface reconstruction applied on test case "M01". (a) Smart visibility result of the TPS surface reconstruction using all 1401 initial points of the depthmap with $T_B = 0.88$. (b) DVR visualization of the ground-truth dataset. (c) Result using 28 merged initial points with $T_B = 0.74$. (d) DVR Visualization of the full dataset.

Minimum Filter Module

As mentioned in the detailed explanation in Section 3.3, the minimum filter method has more value as a preprocessing operation than as a method for complete surface reconstruction. The reconstruction results were evaluated to back up this claim. The following results were obtained:

- Loss of initial point information

A core problem of the minimum filter method is the minimum selection itself. While this selection scheme can effectively eliminate outlying initial points with high depth values, it often discards the depth information of surface regions that were already correctly placed. The clipping surface resulting from the minimum filter method is in most cases dominated by the minimum value found in the initial point depthmap. This effect is shown in Figure 5.3b. Useful results can only be obtained in cases that require very few filter iterations to complete the surface reconstruction.

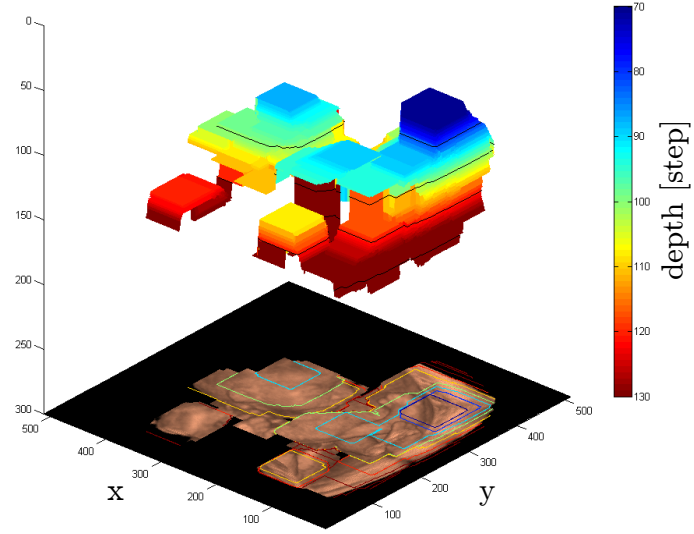
- Result stability

The shape of the reconstructed clipping surface is highly dependent on the spacial distribution of initial points on the initial point depthmap. The surface reconstruction from initial point sets distributed evenly over the depthmap requires fewer iterations to complete than of sets with a heterogeneous distribution. Each additional iteration has the potential to overwrite correct surface placements with high depth values. The removal or addition of a single initial point can therefore greatly influence the shape of the reconstructed surface, if it changes the number of iterations required to complete the reconstruction.

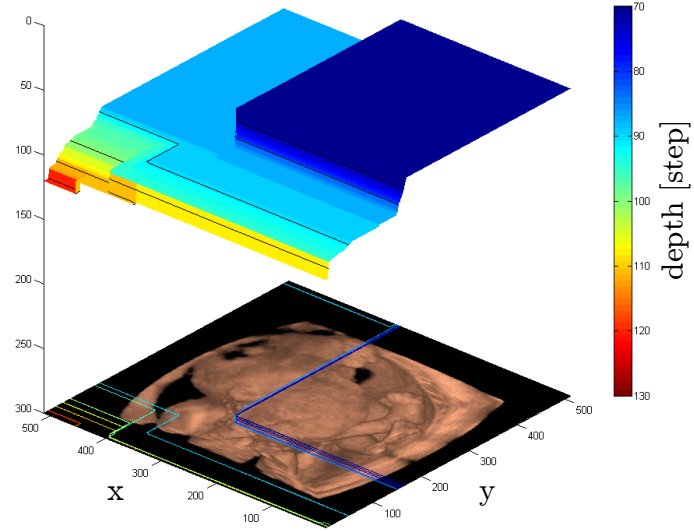
- Performance

Even for filter kernel sizes k of 50×50 and larger the minimum filter module retains real-time performance. Utilizing the separable filter kernel approach reduces the number of texture lookups required per filter kernel application to $4 \cdot k$. Using a simple 2D filter kernel implementation would require $2 \cdot k^2$ lookups.

In general, the minimum filter module generates unsatisfying clipping surfaces results. As Figure 5.4a shows, the results are acceptable after the first three iterations. After the 19 iterations required to reconstruct the complete depthmap, most occluders are reintroduced (Figure 5.4c). The use of the module as preprocessing step for a subsequent surface reconstruction with the mean filter module has been tested. However, no significant improvements could be observed. The few existing incorrect outliers that occurred within the test datasets were in general located with some distance to the closest correct initial points. Increasing the minimum filter kernel size to reach these points in general introduced more additional errors rather than improving the result.



(a) Minimum filter method, 3 iterations



(b) Minimum filter method, 19 iterations

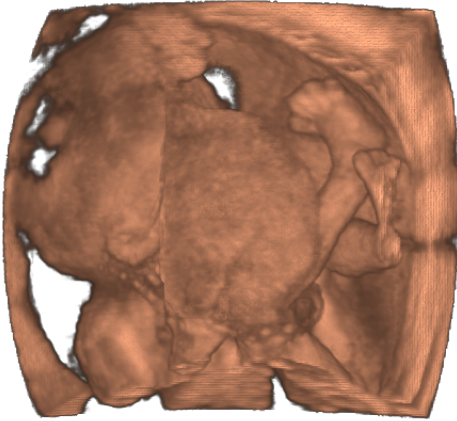
Figure 5.3: 3D visualization of the clipping surface reconstructed by the minimum filter method in test case "M01". The parameter setting $T_B = 0.74$, $k = 21$, $q = 0.25$ is used. Where T_B is the bone threshold, k is the filter kernel size and q is the surface offset parameter (Equation 3.1). (a) 3D surface plot of the reconstructed incomplete surface after 3 iterations. (b) 3D surface plot of the completely reconstructed surface after 19 iterations



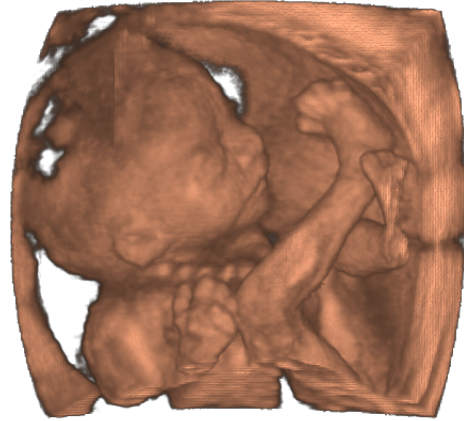
(a) Minimum filter method, 3 iterations



(b) Ground-truth dataset



(c) Minimum filter method, 19 iterations



(d) Ground-truth dataset

Figure 5.4: Result images of the smart visibility method using minimum filter method applied to test case "M01". The parameter setting $T_B = 0.74$, $k = 21$, $q = 0.25$ is used. (a) Smart visibility results of an incomplete surface reconstruction after 3 iterations. (b) Visualization of the ground-truth dataset using the partially reconstructed surface as mask. (c) Smart visibility results of the complete minimum filter surface reconstruction (19 iterations). (d) Visualization of the complete ground-truth dataset.

Mean Filter Module

By using an average calculation of all depth values inside the filter kernel, it is possible to reconstruct a surface with a smooth interpolation between the initial points. In empty areas the mean filter exhibits a similar surface extrapolation behavior as the minimum filter method, as shown in Figure 3.11. The evaluation of the mean filter module revealed the following surface reconstruction properties:

- Initial point approximation

The clipping surface reconstructed by the mean filter module is an approximation surface and does not necessarily pass through the original initial points. The surface converges to a plane parallel to the image plane at the average depth of all initial points. However, to reach this state a large number of iterations is required. Additionally, the weight coefficient ω introduced in Section 3.3 is used to slow down the convergence rate. During the evaluation a value of $\omega = 0.5$ proved to be practical.

- Non-uniform smoothing

Smoothing occurs mainly at the border areas of regions with different depth values. Due to the iterative extrapolation of the surface the filter operation is applied more often in regions close to the initial points, while regions at the border of the depthmap receive fewer smoothing iterations. This behavior can be seen to a small extent in Figure 3.11a where the two regions in the front start to interact after two iterations. The third one is unaffected for one more iteration. In general this results in a smoother surface in the central areas of the depthmap. As mentioned the weight coefficient ω slows down the convergence rates in these areas. The filtering of initial point depth values is required in order to reduce the impact of outliers and to obtain a smooth surface in general.

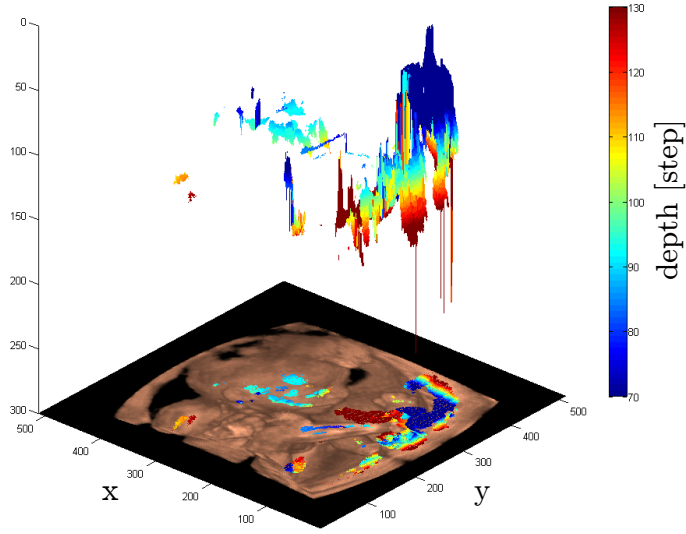
- Result stability

For the same reasons as in the minimum filter case the number of iterations required to reconstruct a clipping surface can vary greatly depending on the on the spacial distribution of initial points in the input depthmap. However, due to the slower rate of convergence, this has little impact on the reconstructed surface. The addition or removal of initial points in general only affects the local surrounding but not the reconstructed surface as a whole.

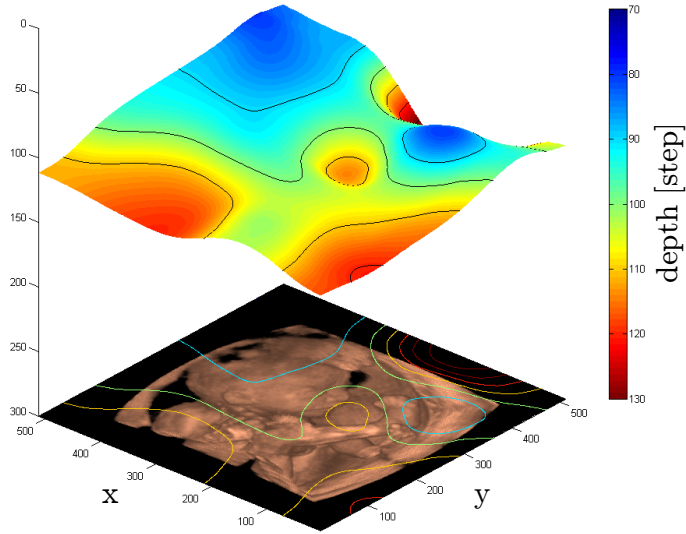
- Performance

Due to the similarity of the kernel functions to those of the minimum filter module, the mean filter module exhibits the same performance. Filter kernel sizes k of 50×50 and larger are possible without losing real-time performance.

Figure 5.6 shows an example of the smart visibility method using the mean filter for surface reconstruction. The clipping removes most of the occluders but also ommits minor parts of the fetus's legs and hand (marked by black arrows).

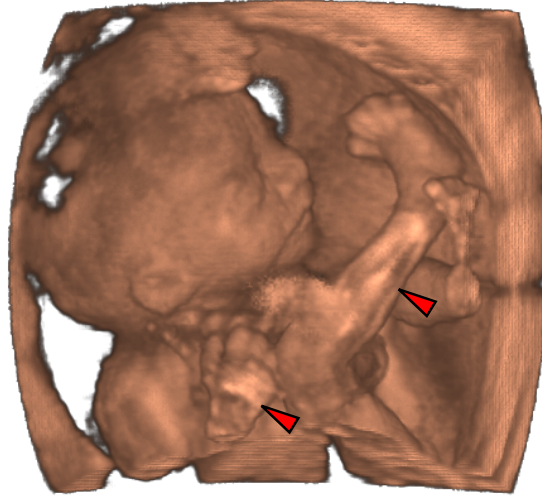


(a) Initial point location

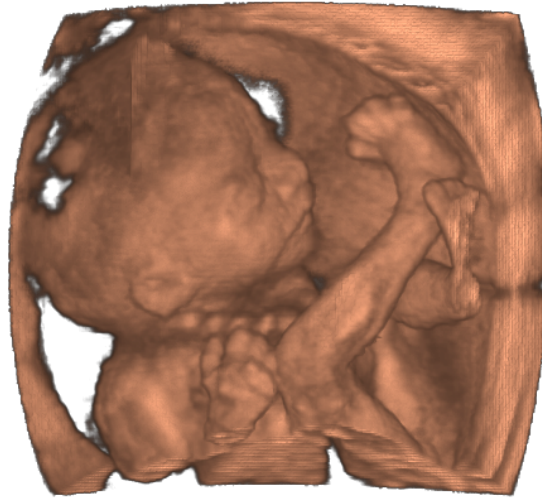


(b) Reconstructed clipping surface

Figure 5.5: 3D visualization of the clipping surface reconstructed by the mean filter method in test case "M01". The parameter setting $T_B = 0.74$, $k = 63$, $q = 0.10$ is used. (a) 3D surface plot of the initial point depthmap. The initial points on the result image are colored with the colormap of the surface to show their corresponding depth. (b) The reconstructed smooth clipping surface.



(a) Mean filter method



(b) Reference dataset

Figure 5.6: Result images of the smart visibility method using mean filter surface reconstruction applied on test case "M01". The parameter setting $T_B = 0.74$, $k = 63$, $q = 0.10$ is used. (a) Smart visibility results of the mean filter surface reconstruction. Only minor occluders remain. The clipping surface is slightly passing through the fetus at its right arm and leg (marked by black arrows). (b) Visualization of the ground-truth dataset.

Parameter Space Analysis

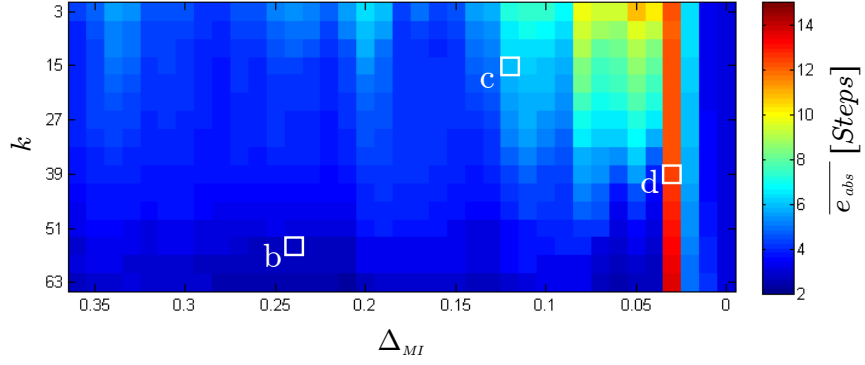
Of all evaluated approaches the mean filter module showed the most promising results in terms of quality as well as performance. To get a better understanding of this method's behavior under varying testing conditions, a parameter space analysis was performed. The smart visibility algorithm with the mean filter method for surface reconstruction provided three parameters to manipulate the results:

- Bone density threshold T_B
controls the thresholding on maximum intensity values, described in Section 3.2. It influences the number of initial points.
- Kernel size $k \times k$
defines the size of the neighborhood considered during the filtering process. It influences the smoothness of the resulting surface.
- Surface offset parameter q
controls the distance of initial points to the skin of the fetus as explained in Equation 3.1. It influences the distance of the surface from the fetus in view direction.

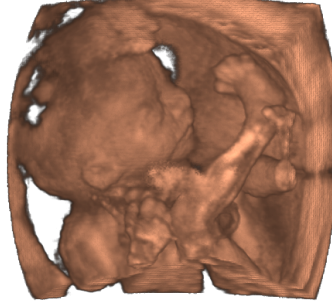
One of the motivations of this work is the reduction of user interaction required during the process of occlusion removal. The parameter space analysis therefore aimed to find settings that minimize the mean error distances calculated by the quality verification module in all available test cases. Table 5.1 lists the intervals and step sizes used to sample all three parameters. Due to its direct influence towards the general shape and location of the surface, the bone density threshold T_B has been analyzed with the finest detail.

Parameter	Interval	Step size
T_B	[0.5, 1.0]	0.01
k	[3, 63]	4
q	[0, 1.5]	0.05

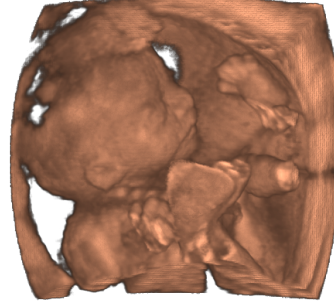
Table 5.1: Sample intervals and step sizes of bone density threshold T_B , kernel size k and surface offset q used for parameter space analysis.



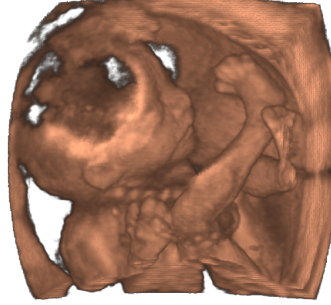
(a) Mean absolute error plot



(b) Mean filter method, $q = 0.1$



(c) Mean filter method, $q = 0.35$



(d) Mean filter method, $q = 0.0$

Figure 5.7: Results of the parameter space analysis of the mean filter method for test case "M01". (a) The mean absolute error plot, showing the minimum error along the q -axis for each sampled Δ_{MI} - k parameter combination. The threshold T_B is represented by its difference Δ_{MI} to the maximum intensity value in the dataset. The white squares mark the parameter settings for three example images: (b) Optimal average error setting (See Table 5.3), $T_B = 0.71$, $k = 55$, $q = 0.10$. (c) Setting with medium error, $T_B = 0.86$, $k = 15$, $q = 0.35$. (d) Setting with high error, $T_B = 0.95$, $k = 39$, $q = 0.0$.

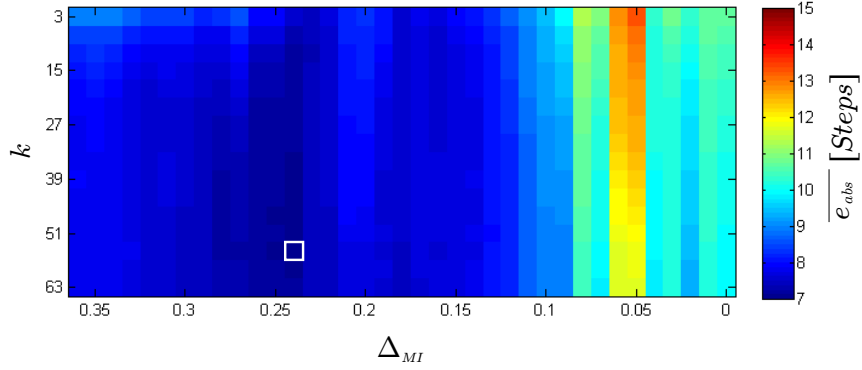


Figure 5.8: Results of the parameter space analysis using the average error values over all test cases. The plot shows the minimum error along the q -axis for each sampled parameter Δ_{MI} - k combination. The white square marks the parameter setting with the lowest average mean absolute error, $\Delta_{MI} = 24$, $k = 55$

The quality verification module was used to calculate the error distances at every sample point of the three-dimensional parameter space. The results were three 3D datasets for every test case, i.e., mean absolute, mean positive and mean negative error ($\overline{e_{abs}}$, $\overline{e_{pos}}$, $\overline{e_{neg}}$). In order to allow a certain degree of fine tuning by the user, at least one parameter of the method has to be manipulable. Of the three listed parameters, the surface offset q is the most intuitive parameter for this task. During the evaluation of the results q was therefore considered to be a variable parameter. For every sample position on the T_B - k plane, the lowest error value in q direction was considered to be the error value of the sample.

Figure 5.7 shows the resulting $\overline{e_{abs}}$ 2D plot for one test case. Additionally, the smart visibility method results for three selected parameter settings are shown. The maximum intensity value of the datasets differed in each test case. A T_B setting higher than the maximum intensity in a dataset does not result in any initial points and no clipping surface can be reconstructed. In the 2D plot, the T_B parameter is therefore represented by Δ_{MI} . Δ_{MI} is the difference between T_B and the maximum intensity value of the given dataset. This removes empty regions from the plot and allows plots of different test cases to be aligned according to their maximum intensity value.

In order to combine the individual results, the average of the mean absolute error distances in all test cases has been calculated for each sample point. The resulting 2D plot is shown in Figure 5.8. The figure shows a correlation between Δ_{MI} and the average $\overline{e_{abs}}$. Selecting the bone threshold value T_B in a distance of 0.1 of the maximum intensity value of the dataset increases the mean absolute error significantly. Thresholding at these high levels result in a low number of initial points. The quality of the resulting clipping surface decreases significantly, due to the lack of input data. However, in some cases the clipping surface at high threshold values degenerates to a flat clipping surface parallel to the image plane. In test case "M01", shown in Figure 5.7, clipping with this flat surface leads to results with good occlusion removal quality.

Table 5.2 shows parameter settings with the minimal mean absolute error $\overline{e_{abs}}$ for each individual test case. The corresponding positive and negative error distances $\overline{e_{pos}}$, $\overline{e_{neg}}$ are shown as

well. The optimal parameter settings vary between all test cases and no global correlation could be determined from the individual results.

Test case	T_B	Δ_{MI}	k	q	$\overline{e_{abs}}$	$\overline{e_{pos}}$	$\overline{e_{neg}}$
E01	0.64	0.23	19	0.80	4.45	0.71	0.81
E02	0.90	0.02	63	0.30	2.35	2.29	2.52
E03	0.92	0.07	11	0.95	6.34	6.66	8.03
E04	0.63	0.25	3	0.75	4.06	4.26	5.74
M01	0.74	0.21	63	0.10	2.39	1.56	1.84
M02	0.58	0.28	19	0.20	5.84	2.83	3.39
M03	0.84	0.04	3	0.95	9.22	5.24	6.41
D01	0.73	0.20	47	0.15	1.08	0.63	0.74
D02	0.50	0.48	19	0.85	2.07	1.07	1.08
Average	-	0.24	55	0.25	7.05	-	-

Table 5.2: Parameter settings minimizing $\overline{e_{abs}}$ for all test cases. The $\overline{e_{pos}}$ and $\overline{e_{neg}}$ values show the positive and negative errors at the selected parameter settings. The unit of error distances is the stepsize used during direct volume rendering. The average parameter setting minimizes the average $\overline{e_{abs}}$ over all test cases. It is derived from the plot in Figure 5.8

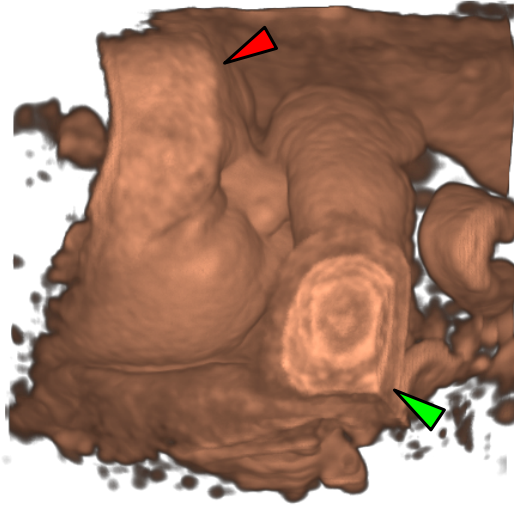
The settings minimizing the average $\overline{e_{abs}}$ suggest a Δ_{MI} value of 0.24 and a filter kernel size k of 55. With these two parameters fixed the minimum absolute error $\overline{e_{abs}}$ was determined again for each test case with q as variable parameter. Table 5.3 shows the resulting values of q for each test case together with the corresponding error values. The results show a moderate increase of error values in most test cases. The test cases "E02" and "E03" exhibit significant increases of $\overline{e_{abs}}$ and are analyzed in detail in Section 5.3.

Test case	q	$\overline{e_{abs}}$	$\overline{e_{pos}}$	$\overline{e_{neg}}$
E01	0.80	4.93	7.97	1.58
E02	0.00	6.06	2.64	9.00
E03	0.25	8.24	7.60	8.02
E04	0.40	4.58	2.33	6.19
M01	0.10	2.74	2.93	2.43
M02	0.15	6.24	8.40	1.77
M03	1.20	10.81	2.21	19.07
D01	0.15	1.23	1.71	0.75
D02	0.80	2.82	3.77	1.63

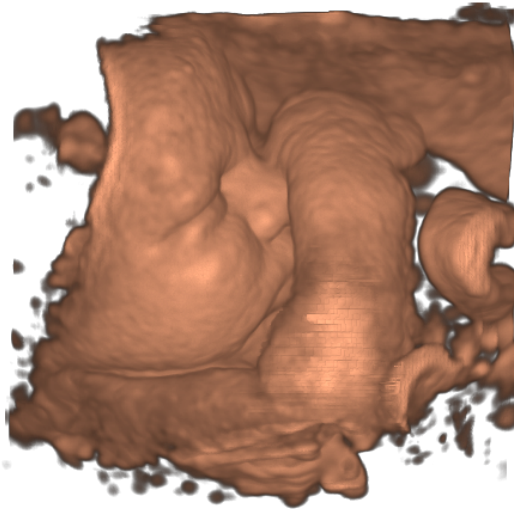
Table 5.3: Surface offset q minimizing $\overline{e_{abs}}$ for all test cases using the settings $\Delta_{MI} = 0.24$ and $k = 55$. The setting is derived from the plot in Figure 5.8 and minimizes the average $\overline{e_{abs}}$ over all test cases. The unit of error distances is the stepsize used during direct volume rendering.

5.3 Test Case Results

This section shows the smart visibility results of all test cases using the parameter settings listed in Table 5.3. The mean filter method is used for surface reconstruction. Artifacts caused by an incorrect placement of the clipping surface are marked by arrows in the example images. Green arrows mark remaining occluders (positive error distance), while red arrows mark regions in which the clipping surface passes through the fetus (negative error distance). The method results are shown on the top and visualizations of the ground-truth volumes are shown on the bottom side of the figures:

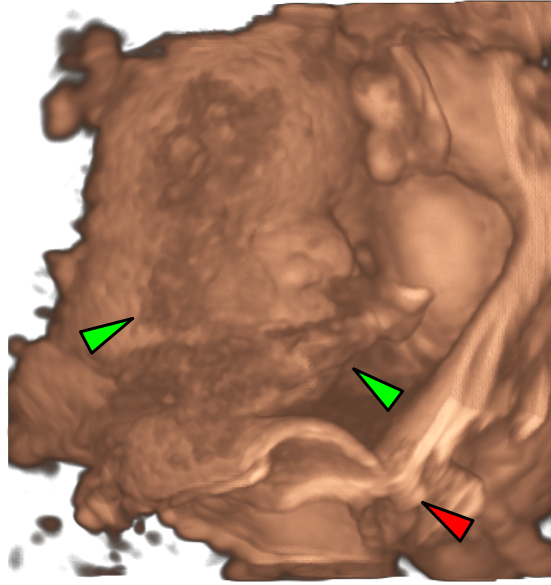


(a) Mean filter method

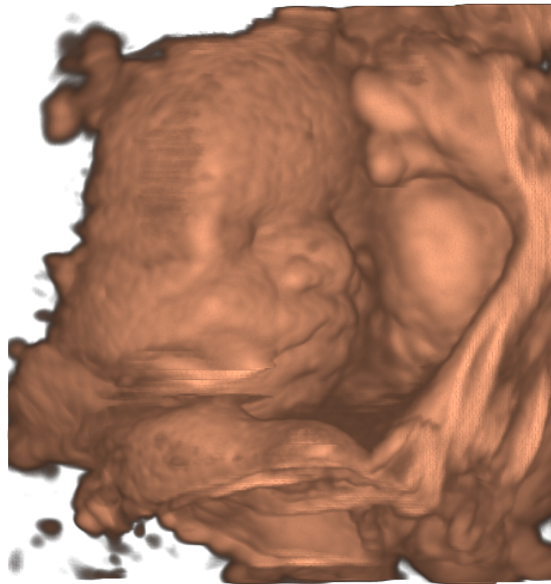


(b) Ground-truth volume

Figure 5.9: Smart visibility results of dataset "E01" (mean filter, $\Delta_{MI} = 0.24$, $k = 55$, $q = 0.80$). (a) Result of the smart visibility method. Occluders are remaining in the region of the right arm. The surface passes through the fetus at the forehead region. (b) Visualization of the corresponding ground-truth volume.

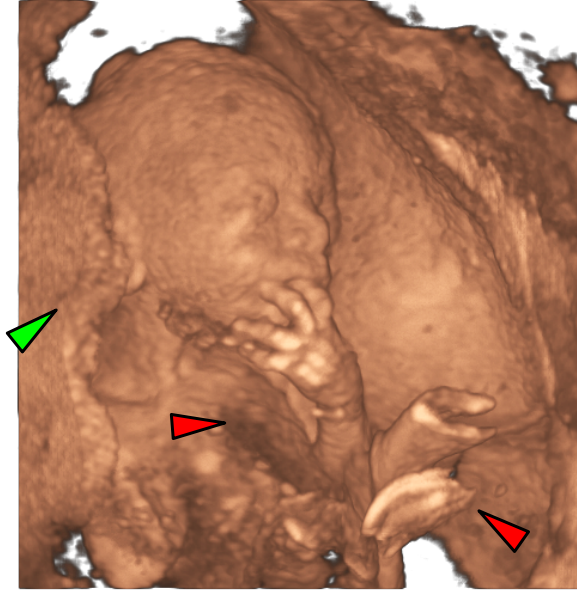


(a) Mean filter method

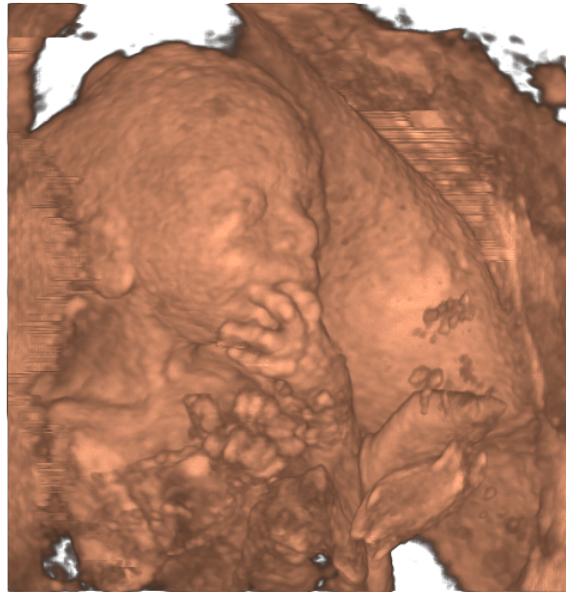


(b) Ground-truth volume

Figure 5.10: Smart visibility results of dataset "E02" (mean filter, $\Delta_{MI} = 0.24$, $k = 55$, $q = 0.00$). (a) Result of the smart visibility method. Several cloudy occluders and parts of the umbilical cord are remaining in the facial region. The surface passes through the right leg of the fetus. (b) Visualization of the corresponding ground-truth volume.

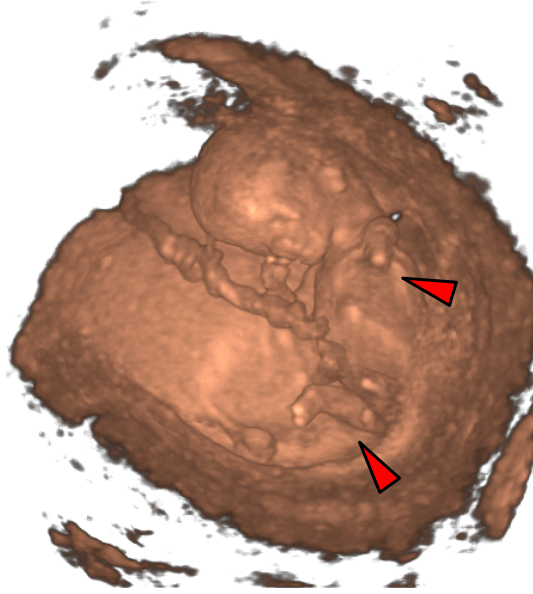


(a) Mean filter method

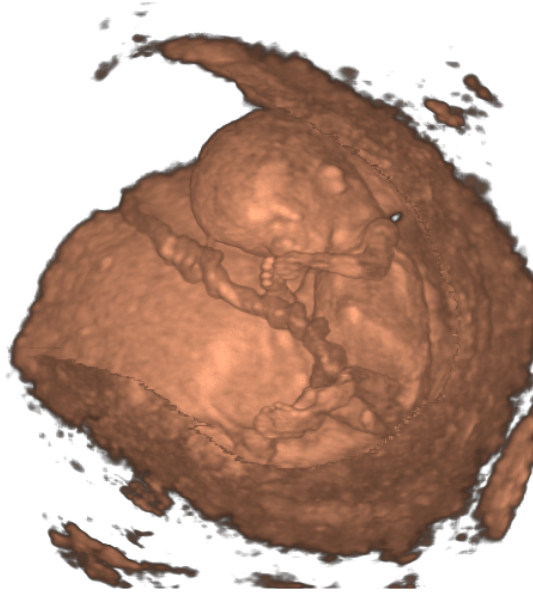


(b) Ground-truth volume

Figure 5.11: Smart visibility results of dataset "E03" (mean filter, $\Delta_{MI} = 0.24$, $k = 55$, $q = 0.25$). (a) Result of the smart visibility method. Occluders in the neck and shoulder region of the fetus remain. The surface passes through the right arm and the legs of the fetus. (b) Visualization of the corresponding ground-truth volume.

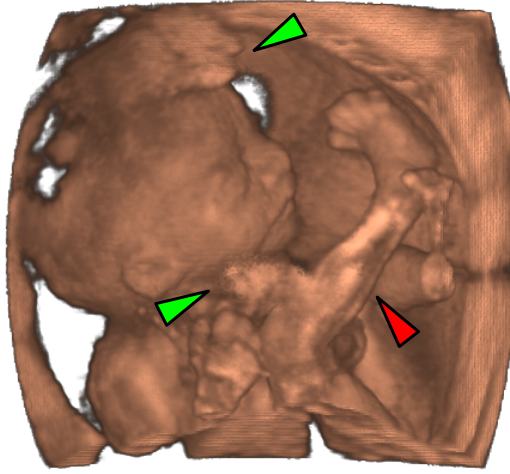


(a) Mean filter method

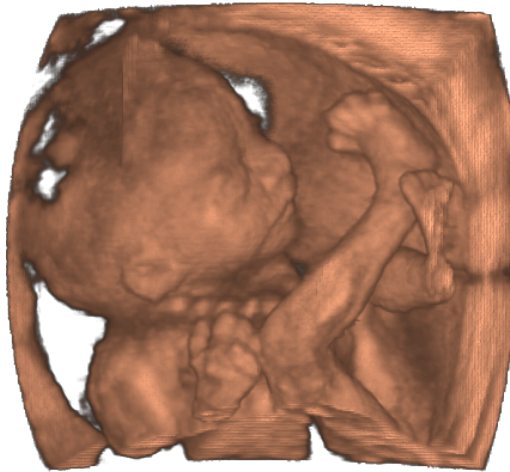


(b) Ground-truth volume

Figure 5.12: Smart visibility results of dataset "E04" (mean filter, $\Delta_{MI} = 0.24$, $k = 55$, $q = 0.40$). (a) Result of the smart visibility method. The surface passes through the left arm and both legs of the fetus. (b) Visualization of the corresponding ground-truth volume.

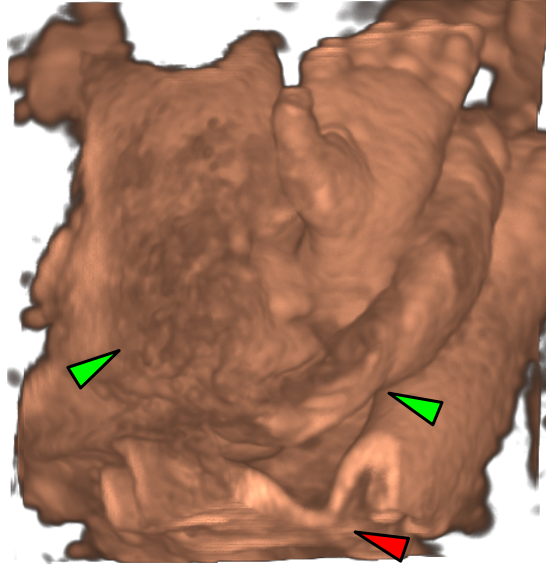


(a) Mean filter method

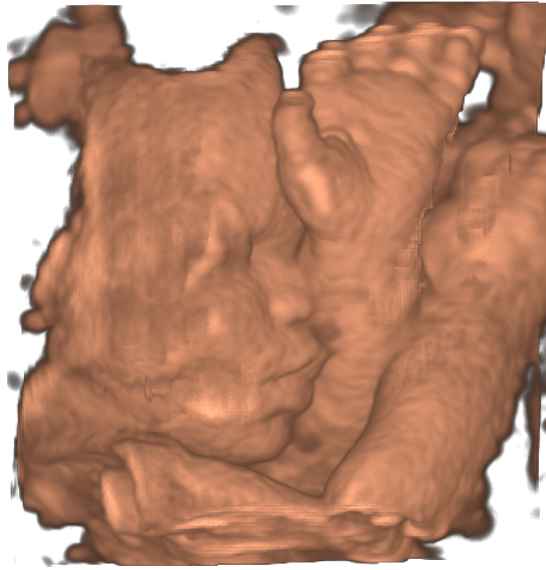


(b) Ground-truth volume

Figure 5.13: Smart visibility results of dataset "M01" (mean filter, $\Delta_{MI} = 0.24$, $k = 55$, $q = 0.10$). (a) Result of the smart visibility method. Occluders in the forehead and knee region of the fetus remain. The surface passes slightly through the right leg of the fetus. (b) Visualization of the corresponding ground-truth volume.

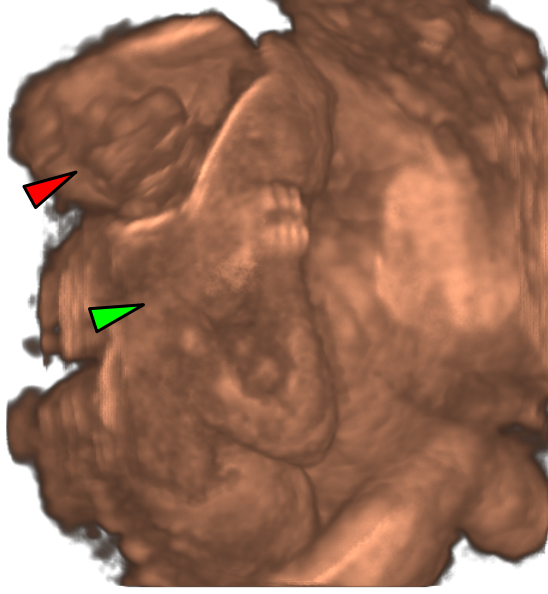


(a) Mean filter method

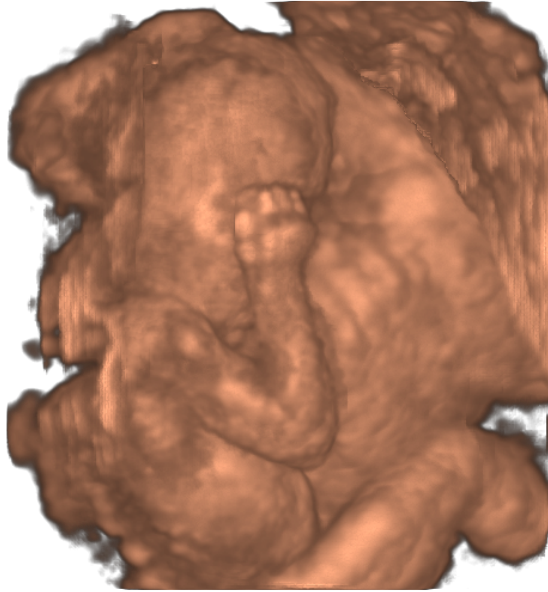


(b) Ground-truth volume

Figure 5.14: Smart visibility results of dataset "M02" (mean filter, $\Delta_{MI} = 0.24$, $k = 55$, $q = 0.15$). (a) Result of the smart visibility method. Several cloudy occluders and parts of the umbilical cord are remaining in the facial region. The surface passes slightly through the right elbow of the fetus. (b) Visualization of the corresponding ground-truth volume.

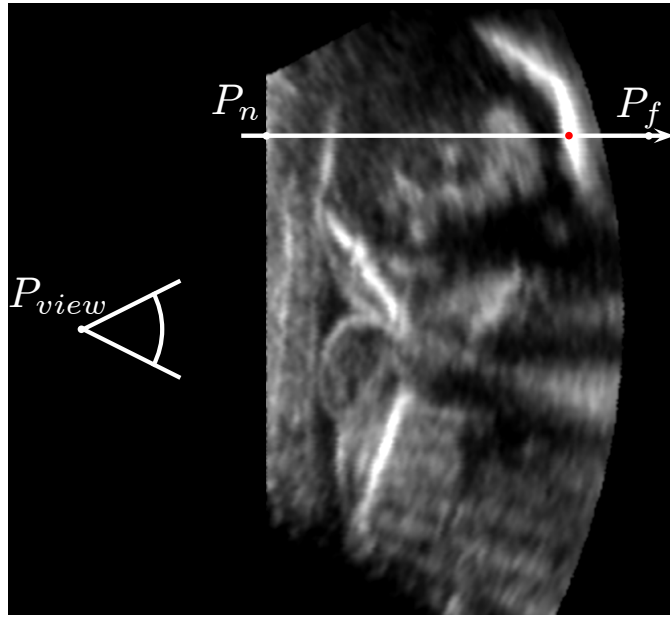


(a) Mean filter method

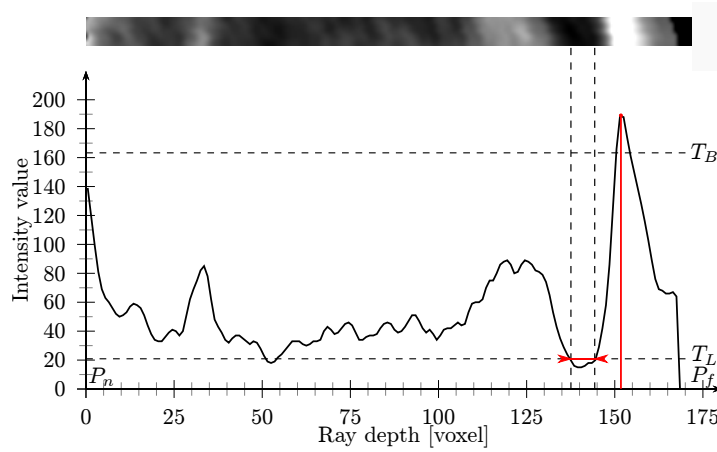


(b) Ground-truth volume

Figure 5.15: Smart visibility results of dataset "M03" (mean filter, $\Delta_{MI} = 0.24$, $k = 55$, $q = 1.20$). (a) Result of the smart visibility method. Several cloudy occluders are remaining in the neck region. A major misplacement of the clipping surface removes the back of the fetus's head and leads to a high negative error of $\overline{e_{neg}} = 19.071$. The cause of this misplacement is elaborated in Figure 5.16. (b) Visualization of the corresponding ground-truth volume.

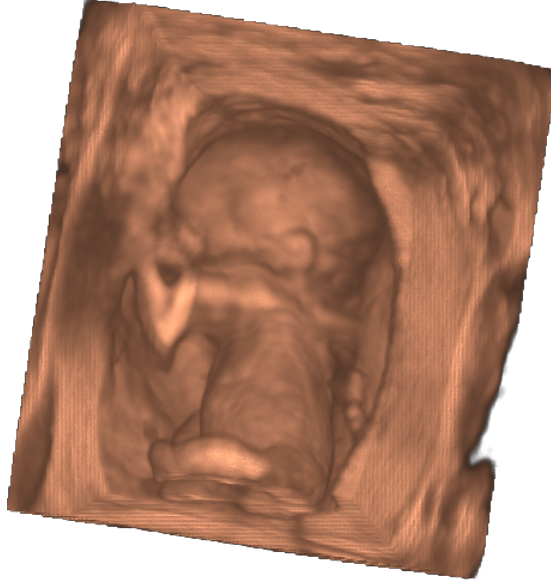


(a) Ray location

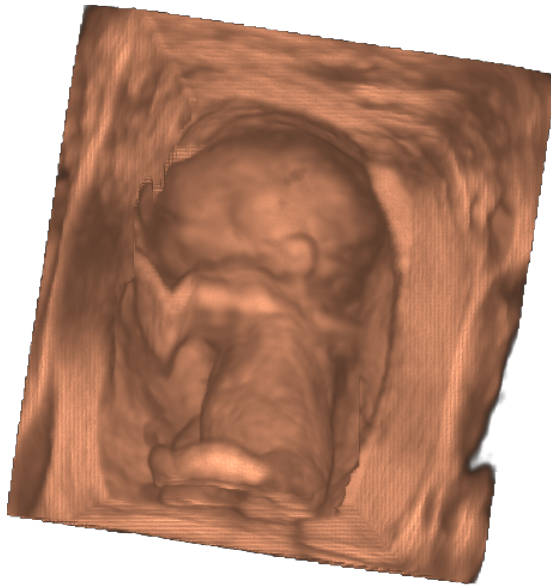


(b) Ray profile

Figure 5.16: The artifact marked by the red arrow in Figure 5.15a is caused by a high intensity region at the back of the fetus' head. (a) A 2D slice of the dataset passing through the head of the fetus. The marked ray passes through the high intensity region which appears to be the head bone. (b) The ray profile corresponding to the marked ray. A small interval with low intensity directly in front of the back part of the head bone causes the selection of initial points directly inside the head of the fetus. It is likely, that the interval as well as the high intensity head bone are results of US artifacts.

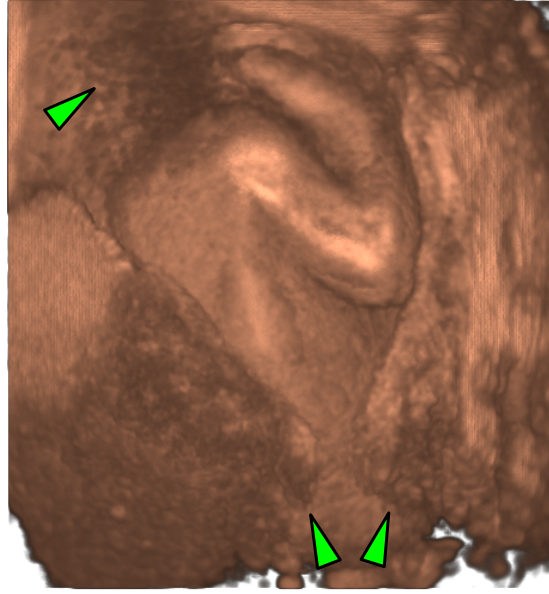


(a) Mean filter method

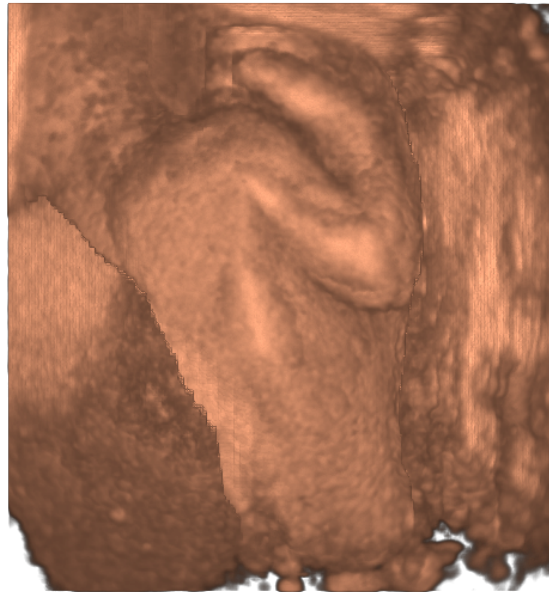


(b) Ground-truth volume

Figure 5.17: Smart visibility results of dataset "D01" (mean filter, $\Delta_{MI} = 0.24$, $k = 55$, $q = 0.15$). (a) Result of the smart visibility method. Nearly perfect occlusion removal, the occurring error distances are in the range of one sampling step. (b) Visualization of the corresponding ground-truth volume.



(a) Mean filter method



(b) Ground-truth volume

Figure 5.18: Smart visibility results of dataset "D02" (mean filter, $\Delta_{MI} = 0.24$, $k = 55$, $q = 0.80$). (a) Result of the smart visibility method. Cloudy occluders are remaining in the neck region. Two large occluders in the back region of the fetus are not completely removed. (b) Visualization of the corresponding ground-truth volume.

Performance

To get an overview of the general performance of the mean filter method, the execution times of its individual modules has been examined. The parameter settings listed in Table 5.3 were used for this evaluation. Figure 5.19 shows the framerates for each test case. The execution time of each module is encoded into each bar as ratio of the framerate. The mean filter module accounts for the largest share of the total execution time. On average over 85% of the execution time is used for surface reconstruction, followed by the general setup process for each frame with 8% (Table 5.4). The initial point and rendering module have a very low influence on the total framerate.

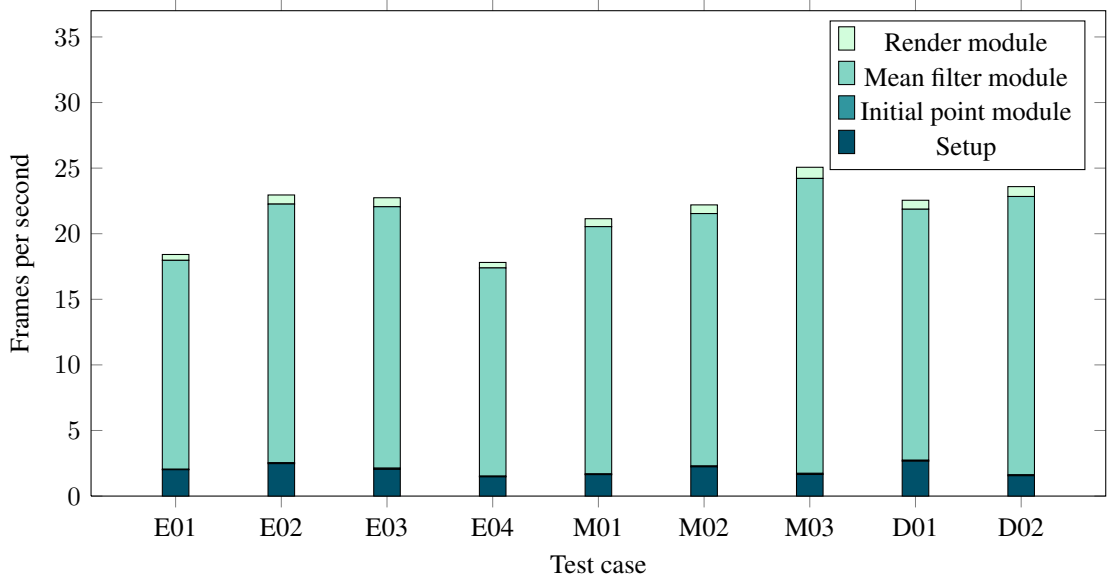


Figure 5.19: Performance of the mean filter method in all test cases. The execution times of the modules as ratio of the frame rate is encoded in each bar. The initial point module execution requires a very low execution time and is not visible in this plot. The parameter settings listed in Table 5.3 were used for the measurements.

Setup [ms]	4.223
Initial point module [ms]	0.140
Mean filter module [ms]	40.631
Render module [ms]	1.335
Total [ms]	46.329

Table 5.4: Average execution times of the modules used by the mean filter smart visibility method. The parameter settings listed in Table 5.3 were used for the measurements.

In the mean filter module, only the bone density threshold T_B and the filter kernel size k have

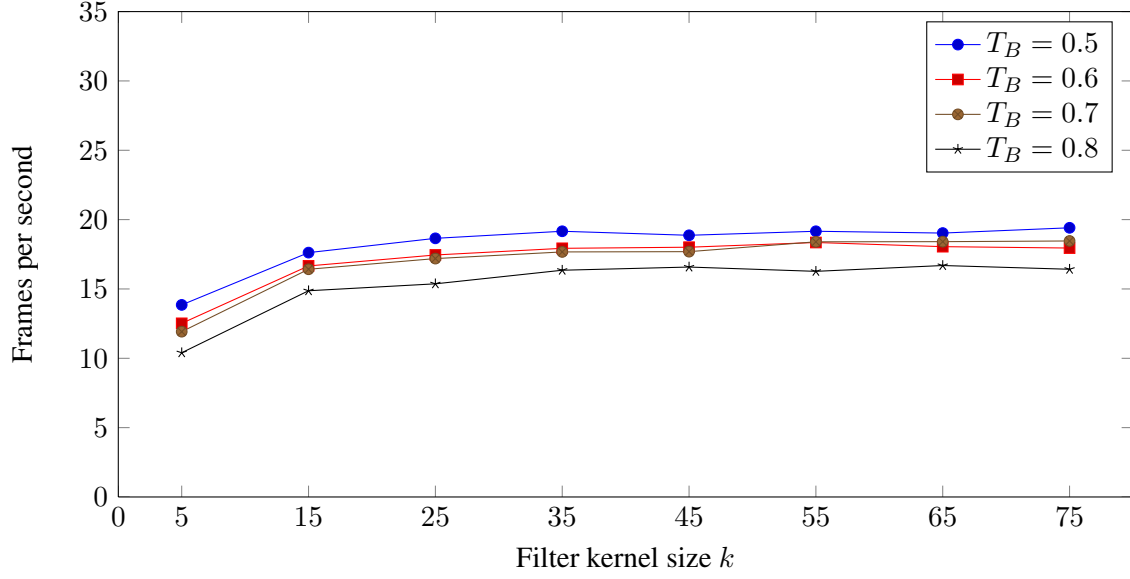


Figure 5.20: Impact of the bone threshold T_B and the kernel size k on the performance in test case "E04". The performance increases for lower values of T_B and larger filter kernel sizes k .

influence on the execution time. Therefore the effect of different settings for both parameters were evaluated in detail. Figure 5.20 shows the performance curves of the mean filter method in test case "E04". While the performance decreased with higher T_B values the performance converges to a fixed value with increasing kernel sizes. As previously mentioned a higher T_B value decreases the number of initial points. In most cases this increases the number of mean filter iterations required to reconstruct the complete clipping surface and has direct influence on the execution time. Conversely, increasing the kernel size reduces the execution time. Iterative filter applications cause a considerable computation overhead. Larger filter kernels require fewer iterations to reconstruct the clipping surface and show a better performance. For kernel sizes of 35×35 only two iterations are required to fill the complete depthmap, therefore the framerate stays stable for larger kernels at an image resolution of 512×512 pixel. All performance measurements were obtained on a PC with specifications as given in Table 5.5.

CPU	Intel Core i7-3930K (12 × 3.20 GHz)
RAM	16 GB
GPU	GForce GTX 690 (4 GB DDR5 Memory)

Table 5.5: PC setup used for all performance tests of the smartvis framework

5.4 Simu3D Integration

After an evaluation performed by domain experts from GE Healthcare, the mean filter approach was selected for integration and testing within the Simu3D US firmware. Simu3D is a proprietary development version of one of the company's ultrasound machine firmwares. A plugin interface offering access to data and parameters required by the smartvis algorithm was provided to enable the transition into the framework. The modular structure of the smartvis framework supported the transition process, since module implementations were not strongly tied to other modules and the smartvis framework. However, several basic changes in the Simu3D environment made modifications necessary:

- Volume data

Simu3D does not provide access to full resolution datasets at once, instead the data is transferred as subsequent slabs. For plugins that require the whole volume for processing a scan-converted lower resolution version of the volume is available as tiled DirectX 9 2D texture. The plugin interface only gives access to the low resolution data set, therefore custom 3D lookup functions for the 2D texture needed to be implemented.

- Camera-model

Like the smartvis framework Simu3D uses an orthographic camera model. This allows measurement and comparison of geometric features without distortions caused by perspective projection. In order to avoid the necessity of a direct access to the camera parameters, the volume data provided by the plugin interface is already transformed with respect to the camera position. This simplified the transition effort. Source code concerning camera parameters and projection model did not have to be integrated.

- Rendering

Simu3D provides a variety of rendering modes for clinical use and provides methods to manipulate the starting positions for every ray as part of the existing occlusion removal methods. The integration of the rendering module functions was therefore not necessary. Instead the plugin directly returns the depthmap of the reconstructed clipping surface to the framework, where it is accessed by the DVR component. The missing access to the rendering functions prevented a transition of the ghosting technique presented in Section 3.4.

Plugin Evaluation

After the integration of the mean filter method into Simu3D, the occlusion removal results were compared with results obtained from the smartvis framework. The visualization results of both smart visibility implementations were compared. The low resolution volume datasets available in the plugin are scan-converted to about one third of the original resolution in each dimension. In order to generate comparable results the resolution of the depthmap used in the smartvis framework was reduced to the resolution employed in the plugin. Additionally the sampling distance during ray profile analysis of the initial point module was increased to match the lower

resolution. Both implementations used a reduced filter kernel size to account for the lower depthmap resolution. Apart from minor errors caused by the different downsampling methods the smart visibility implementations delivered equivalent results.

The influence of the plugin execution on the frame rate of the Simu3D framework was evaluated using the built-in performance measuring tools. The performance was measured using three different settings for each test case:

- DVR: Frame rate with deactivated scan conversion and smartvis plugin.
- DVR & scan conversion: Frame rate with activated scan conversion but deactivated smartvis plugin.
- Smart visibility: Frame rate with activated scan conversion and smartvis plugin.

Figure 5.21 shows the resulting performance of all test cases for the parameter settings $\Delta_{MI} = 0.26$, $k = 23$ and a value of q varying for each test case. The frame rate of DVR rendering without scan conversion and smartvis plugin varies greatly depending on the test case. Due to the unavailable access to the core functions of the Simu3D framework the cause of these fluctuations could not be determined. Activating the scan conversion of datasets lowers the frame rate considerably, however this is likely caused by the emulation of this feature within the Simu3D framework. On average the use of the smartvis plugin reduced the frame rate by 6 frames per second. As test case "M01" shows, this remains true even for higher frame rates after the scan conversion.

Summary

The evaluation of the smart visibility methods presented in Chapter 3 showed that the method using the mean filter for surface reconstruction provides the best occlusion removal results. An analysis of the parameter space constituted by the three parameters of the mean filter method (T_B , k and q) resulted in optimal settings for each test case. Additionally a fixed setting for the two surface reconstruction parameters T_B and k was found. These setting resulted in a useful clipping surface in most test cases, with the exception of test case "M03" (as seen in Figure 5.15a).

The mean filter method was successfully transferred into the Simu3D US firmware. The final performance analysis showed that the mean filter method provides frame rates fit for real-time applications in the smartvis as well as the Simu3D framework. These results convinced our collaborators at GE Healthcare of the effectiveness of the smartvis algorithm. After further evaluation in clinical setups our method is considered for inclusion into the GE US firmware in a future update.

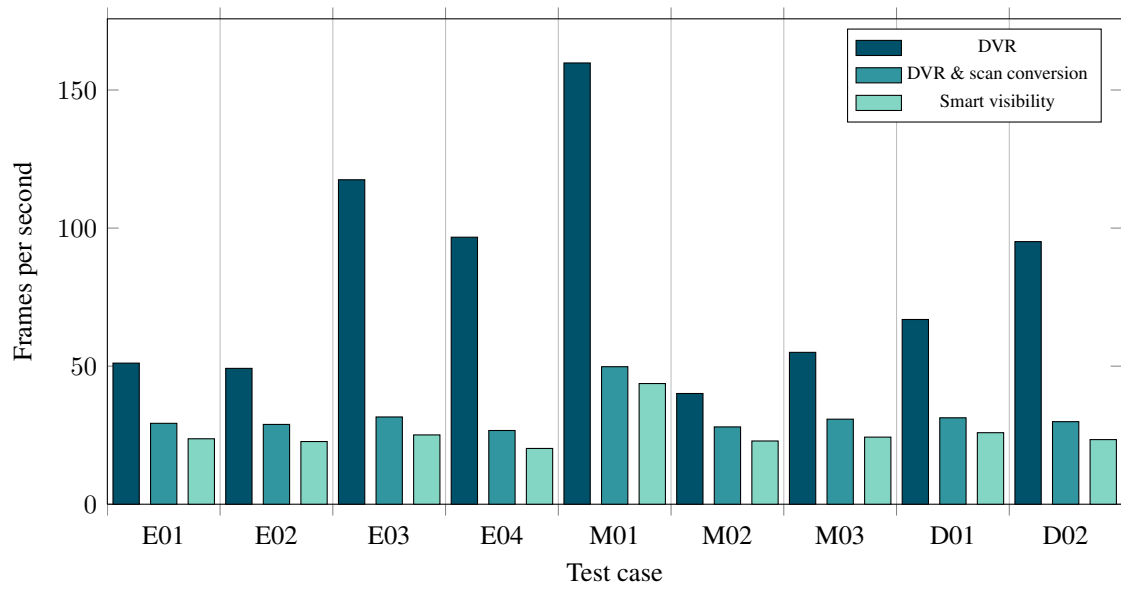


Figure 5.21: Performance comparison of standard DVR, standard DVR including low-resolution scan conversion and DVR with low-resolution scan conversion and enabled smart visibility of all test cases. The parameter settings of the smartvis framework performance test (Table 5.3) were adapted to the lower resolution of the volumes in each test case.

Conclusion and Future Work

This thesis presents and analyzes three methods for automated, user-optimizable occlusion removal in obstetric 3D ultrasound datasets. All tested methods follow the basic concept of initial point selection and surface reconstruction to generate a clipping surface between occluders and the fetus. The reconstructed surface is incorporated into the rendering process by using smart visibility concepts. The resulting images show a significant reduction of occluders.

The initial point selection is based on anatomic features of the fetus and the amniotic sac. These features were found in all available test datasets. The selection result is a set of points which have a high probability to be positioned between the fetus and the first occluder in the direction to the viewpoint. The tested approaches vary mainly by the method of surface reconstruction applied. One spline-based and two local-filtering approaches have been examined. Different smart visibility rendering modes allow the fine tuning of the occlusion-removal effect.

The smartvis framework developed as part of the thesis was used to evaluate the occlusion-removal quality and performance of the three presented approaches. In cooperation with domain experts of GE Healthcare the mean-filter method has been selected as most promising approach and was subsequently integrated into the Simu3D US firmware for further testing. A comparison between results of the mean-filter approach and manually segmented ground truth datasets show good occlusion-removal quality in a majority of the test cases. By using preset parameter settings, the amount of user-input is reduced to a single slider-controllable parameter. This allows the easy use of the smartvis approach during US examinations. The performance analysis of the mean-filter method shows that the frame rate stays within an acceptable range for real-time applications in all test cases. GE Healthcare will continue to evaluate the mean filter method in clinical environments. It can therefore be concluded that the smartvis framework successfully fulfilled the task it was developed for.

Future Work

While the mean-filter method shows promising results, several aspects of the approach have potential for improvement. Defining the bone density threshold by a fixed offset to the maximum intensity value of a dataset is a practical way to select initial points. However, the number of initial points obtained by a fixed offset varies significantly between different datasets. The number could be stabilized by considering the histogram of each dataset in the threshold selection.

Due to the missing access to the core rendering functions in the Simu3D framework the use of ghosting techniques has not yet been evaluated. An inclusion of the depth-dependent opacity coefficient presented in Section 3.4 could reduce the visual impact of incorrectly placed clipping surfaces.

Currently the performance of the method allows real-time usage, but low-level improvements to the mean-filter functions might yield even lower computation times. Lastly the method should be verified by testing it with a larger number of datasets in order to detect and analyze test cases showing suboptimal results. The results of these tests could provide additional insight that might be used to refine the algorithm.

Bibliography

- [1] GE Healthcare, “Voluson S8–Voluson–Ultrasound–Products–gehealthcare.com.” http://www3.gehealthcare.com/en/Products/Categories/Ultrasound/Voluson/Voluson_S8. Accessed: 2013-03-12.
- [2] GE Healthcare, “Voluson E6 Brochure–gehealthcare.com.” http://www3.gehealthcare.de/~media/Downloads/uk/Product/Ultrasound/Voluson/Voluson_E6/Voluson_E6_Brochure.pdf. Accessed: 2013-03-12.
- [3] A. Varchola, *Live Fetoscopic Visualization of 4D Ultrasound Data*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, Oct. 2012.
- [4] I. Viola and M. E. Gröller, “Smart visibility in visualization,” in *Computational Aesthetics in Graphics, Visualization and Imaging, 2005. Proceedings of the First Eurographics conference on, Computational Aesthetics’05*, (Aire-la-Ville, Switzerland, Switzerland), pp. 209–216, Eurographics Association, 2005.
- [5] V. Chan and A. Perlas, “Basics of ultrasound imaging,” in *Atlas of Ultrasound-Guided Procedures in Interventional Pain Management* (S. N. Narouze, ed.), pp. 13–19, Springer New York, 2011.
- [6] J. E. Aldrich, “Basic physics of ultrasound imaging,” *Critical Care Medicine*, vol. 35, pp. 131–137, may 2007.
- [7] R. N. Rankin, A. Fenster, D. B. Downey, P. L. Munk, M. F. Levin, and A. D. Vellet, “Three-dimensional sonographic reconstruction: Techniques and diagnostic applications,” *American Journal of Roentgenology*, vol. 161, pp. 695–702, oct 1993.
- [8] R. W. Prager, U. Z. Ijaz, A. H. Gee, and G. M. Treece, “Three-dimensional ultrasound imaging,” in *Journal of Engineering in Medicine. Proceedings of the Institution of Mechanical Engineers, Part H:*, vol. 224, pp. 193–223, Sage Publications, Inc., feb 2010.
- [9] J.-M. Bureau, W. Steichen, and G. Lebaill, “A two-dimensional transducer array for real-time 3d medical ultrasound imaging,” in *Ultrasonics Symposium, 1998. Proceedings of the IEEE*, vol. 2, pp. 1065–1068, 1998.

- [10] H. Gemmeke, L. Berger, M. Birk, G. Gobel, A. Menshikov, D. Tcherniakhovski, M. Zapf, and N. Ruiter, "Hardware setup for the next generation of 3d ultrasound computer tomography," in *Nuclear Science Symposium Conference Record, 2010. IEEE*, pp. 2449–2454, 2010.
- [11] J. T. Yen and S. W. Smith, "Real-time rectilinear volumetric imaging using a periodic array," *Ultrasound in Medicine & Biology*, vol. 28, no. 7, pp. 923–931, 2002.
- [12] N. V. Ruiter, G. F. Schwarzenberg, M. Zapf, and H. Gemmeke, "Conclusions from an experimental 3d ultrasound computer tomograph," in *Nuclear Science Symposium Conference Record, 2008. IEEE*, pp. 4502–4509, oct 2008.
- [13] A. Gee, R. Prager, G. Treece, and L. Berman, "Engineering a freehand 3d ultrasound system," *Pattern Recognition Letters*, vol. 24, pp. 757–777, Feb. 2003.
- [14] G. Treece, R. Prager, A. Gee, and L. Berman, "3d ultrasound measurement of large organ volume," *Medical Image Analysis*, vol. 5, no. 1, pp. 41–54, 2001.
- [15] E. M. Boctor, M. A. Choti, E. C. Burdette, and R. J. Webster III, "Three-dimensional ultrasound-guided robotic needle placement: an experimental evaluation," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 4, no. 2, pp. 180–191, 2008.
- [16] J. Hummel, M. Figl, M. Bax, H. Bergmann, and W. Birkfellner, "2d/3d registration of endoscopic ultrasound to ct volume data," *Physics in Medicine and Biology*, vol. 53, no. 16, p. 4303, 2008.
- [17] F. Sheehan, M. Schneider, E. Bolson, and B. Webster, "Tracking three dimensional ultrasound with immunity from ferro-magnetic interference," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003* (R. Ellis and T. Peters, eds.), vol. 2879 of *Lecture Notes in Computer Science*, pp. 192–198, Springer Berlin / Heidelberg, 2003.
- [18] G. Penney, J. Blackall, M. Hamady, T. Sabharwal, A. Adam, and D. Hawkes, "Registration of freehand 3d ultrasound and magnetic resonance liver images," *Medical Image Analysis*, vol. 8, no. 1, pp. 81–91, 2004.
- [19] W. Birkfellner, F. Watzinger, F. Wanschitz, R. Ewers, and H. Bergmann, "Calibration of tracking systems in a surgical environment," *Medical Imaging, 1998. IEEE Transactions on*, vol. 17, pp. 737–742, oct 1998.
- [20] C. Poulsen, P. Pedersen, and T. Szabo, "An optical registration method for 3d ultrasound freehand scanning," in *Ultrasonics Symposium, 2005. Proceedings of the IEEE*, vol. 2, pp. 1236–1240, sept 2005.
- [21] A. Goldsmith, P. Pedersen, and T. Szabo, "An inertial-optical tracking system for portable, quantitative, 3d ultrasound," in *Ultrasonics Symposium, 2008. Proceedings of the IEEE*, pp. 45–49, nov 2008.

- [22] R. J. Housden, A. H. Gee, G. M. Treece, and R. W. Prager, "Sensorless reconstruction of unconstrained freehand 3d ultrasound data," *Ultrasound in medicine & biology*, vol. 33, no. 3, pp. 408–419, 2007.
- [23] N. Max, "Optical models for direct volume rendering," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 1, pp. 99–108, jun 1995.
- [24] J. Huang, K. Mueller, R. Crawfis, D. Bartz, and M. Meissner, "A practical evaluation of popular volume rendering algorithms," in *Symposium on Volume Visualization, 2000. Proceedings of the IEEE*, pp. 81–90, 2000.
- [25] B. Cabral, N. Cam, and J. Foran, "Accelerated volume rendering and tomographic reconstruction using texture mapping hardware," in *Symposium on Volume visualization, 1994. Proceedings of the, VVS '94*, (New York, NY, USA), pp. 91–98, ACM, 1994.
- [26] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl, "Interactive volume rendering on standard pc graphics hardware using multi-textures and multi-stage rasterization," in *Graphics hardware, 2000. Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on*, HWWS '00, (New York, NY, USA), pp. 109–118, ACM, 2000.
- [27] L. Westover, "Footprint evaluation for volume rendering," in *Computer graphics and interactive techniques, 1990. Proceedings of the 17th annual conference on*, SIGGRAPH '90, (New York, NY, USA), pp. 367–376, ACM, 1990.
- [28] K. Mueller and R. Crawfis, "Eliminating popping artifacts in sheet buffer-based splatting," in *Visualization, 1998. Proceedings of the conference on*, VIS '98, (Los Alamitos, CA, USA), pp. 239–245, IEEE Computer Society Press, 1998.
- [29] K. Mueller, N. Shareef, J. Huang, and R. Crawfis, "High-quality splatting on rectilinear grids with efficient culling of occluded voxels," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 5, pp. 116–134, apr-jun 1999.
- [30] F. Vega-Higuera, P. Hastreiter, R. Fahlbusch, and G. Greiner, "High performance volume splatting for visualization of neurovascular data," in *Visualization, 2005. Proceedings of the conference on IEEE*, pp. 271–278, oct 2005.
- [31] N. Neophytou and K. Mueller, "Gpu accelerated image aligned splatting," in *Volume Graphics, 2005. Fourth International Workshop on*, pp. 197–242, jun 2005.
- [32] M. Levoy, "Display of surfaces from volume data," *Computer Graphics and Applications, IEEE*, vol. 8, pp. 29–37, may 1988.
- [33] T. Porter and T. Duff, "Compositing digital images," in *Computer graphics and interactive techniques, 1984. Proceedings of the 11th annual conference on*, SIGGRAPH '84, (New York, NY, USA), pp. 253–259, ACM, 1984.

- [34] S. Roettger, S. Guthe, D. Weiskopf, T. Ertl, and W. Strasser, "Smart hardware-accelerated volume rendering," in *Symposium on Data visualisation, 2003. Proceedings of the, VIS-SYM '03*, (Aire-la-Ville, Switzerland, Switzerland), pp. 231–238, Eurographics Association, 2003.
- [35] J. Krüger and R. Westermann, "Acceleration techniques for gpu-based volume rendering," in *Visualization, 2003. Proceedings of the 14th IEEE, VIS '03*, (Washington, DC, USA), pp. 38–, IEEE Computer Society, 2003.
- [36] M. Levoy, "Efficient ray tracing of volume data," *ACM Trans. Graph.*, vol. 9, pp. 245–261, July 1990.
- [37] T. van Walsum, A. J. S. Hin, J. Versloot, and F. Post, "Efficient hybrid rendering of volume data and polygons," *Advances in Scientific Visualization*, pp. 83–96, 1992.
- [38] M. Levoy, "Volume rendering by adaptive refinement," *The Visual Computer*, vol. 6, pp. 2–7, 1990. 10.1007/BF01902624.
- [39] T. Ropinski, J. Kasten, and K. H. Hinrichs, "Efficient shadows for gpu-based volume ray-casting," in *Computer Graphics, Visualization and Computer Vision, 2008. Proceedings of the 16th International Conference in Central Europe on*, pp. 17–24, 2008.
- [40] F. Hernell, P. Ljung, and A. Ynnerman, "Efficient ambient and emissive tissue illumination using local occlusion in multiresolution volume rendering," in *Volume Graphics, 2007. Proceedings of the Sixth Eurographics / IEEE VGTC conference on, VG'07*, (Aire-la-Ville, Switzerland, Switzerland), pp. 1–8, Eurographics Association, 2007.
- [41] M. Hadwiger, J. M. Kniss, C. Rezk-salama, D. Weiskopf, and K. Engel, *Real-time Volume Graphics*. Natick, MA, USA: A. K. Peters, Ltd., 2006.
- [42] M. Hadwiger, P. Ljung, C. R. Salama, and T. Ropinski, "Advanced illumination techniques for gpu volume raycasting," in *ACM SIGGRAPH ASIA 2008 courses*, SIGGRAPH Asia '08, (New York, NY, USA), pp. 1:1–1:166, ACM, 2008.
- [43] P. Lacroute and M. Levoy, "Fast volume rendering using a shear-warp factorization of the viewing transformation," in *Computer graphics and interactive techniques, 1994. Proceedings of the 21st annual conference on*, SIGGRAPH '94, (New York, NY, USA), pp. 451–458, ACM, 1994.
- [44] S. Bruckner and M. E. Gröller, "Exploded views for volume data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, pp. 1077–1084, 9 2006.
- [45] C. Correa, D. Silver, and M. Chen, "Feature aligned volume manipulation for illustration and visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, pp. 1069–1076, sept–oct 2006.
- [46] C. Rezk-Salama and A. Kolb, "Opacity peeling for direct volume rendering," *Computer Graphics Forum*, vol. 25, no. 3, pp. 597–606, 2006.

- [47] M. M. Malik, T. Möller, and M. E. Gröller, "Feature peeling," in *Graphics Interface, 2007. Proceedings of, GI '07*, (New York, NY, USA), pp. 273–280, ACM, 2007.
- [48] S. K. Feiner and D. D. Seligmann, "Cutaways and ghosting: satisfying visibility constraints in dynamic 3d illustrations," *The Visual Computer*, vol. 8, pp. 292–302, 1992.
- [49] J. Diepstraten, D. Weiskopf, and T. Ertl, "Interactive cutaway illustrations," *Computer Graphics Forum*, vol. 22, no. 3, pp. 523–532, 2003.
- [50] M. Burns and A. Finkelstein, "Adaptive cutaways for comprehensible rendering of polygonal scenes," in *ACM SIGGRAPH Asia 2008 papers*, SIGGRAPH Asia '08, (New York, NY, USA), pp. 154:1–154:7, ACM, 2008.
- [51] Å. Birkeland, S. Bruckner, A. Brambilla, and I. Viola, "Illustrative membrane clipping," *Computer Graphics Forum*, vol. 31, pp. 905–914, June 2012.
- [52] I. Viola, A. Kanitsar, and M. E. Gröller, "Importance-driven volume rendering," in *Visualization, 2004. Proceedings of the conference on, VIS '04*, (Washington, DC, USA), pp. 139–146, IEEE Computer Society, 2004.
- [53] M. Burns, M. Haidacher, W. Wein, I. Viola, and E. Gröller, "Feature emphasis and contextual cutaways for multimodal medical visualization," in *Symposium on Visualization, 2007. Proceedings of the Eurographics/IEEE VGTC*, pp. 275–282, May 2007.
- [54] J. Krüger, J. Schneider, and R. Westermann, "Clearview: An interactive context preserving hotspot visualization technique," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, pp. 941–948, sept–oct 2006.
- [55] Y. Wu and H. Qu, "Interactive transfer function design based on editing direct volume rendered images," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, pp. 1027–1040, sept–oct 2007.
- [56] W.-H. Mak, Y. Wu, M.-Y. Chan, and H. Qu, "Visibility-aware direct volume rendering," *J. Comput. Sci. Technol.*, vol. 26, pp. 217–228, Mar. 2011.
- [57] C. Correa and K.-L. Ma, "Visibility-driven transfer functions," in *Visualization Symposium, 2009. Proceedings of the IEEE Pacific*, pp. 177–184, apr 2009.
- [58] C. Correa and K.-L. Ma, "Visibility histograms and visibility-driven transfer functions," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, pp. 192–204, feb 2011.
- [59] C. Correa and K.-L. Ma, "The occlusion spectrum for volume classification and visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, pp. 1465–1472, nov–dec 2009.
- [60] W. Chang, *Surface reconstruction from points*. Department of Computer Science and Engineering, University of California, San Diego, 2008.

- [61] R. Mencl and H. Müller, "Interpolation and approximation of surfaces from three-dimensional scattered data points," in *Scientific Visualization, 1997. Proceedings of the Conference on*, pp. 223–223, 1997.
- [62] F. Cazals and J. Giesen, "Delaunay triangulation based surface reconstruction," in *Effective Computational Geometry for Curves and Surfaces*, pp. 231–276, Springer, 2006.
- [63] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Trans. Graph.*, vol. 13, pp. 43–72, Jan. 1994.
- [64] N. Amenta, M. Bern, and M. Kamvysselis, "A new Voronoi-based surface reconstruction algorithm," in *Computer graphics and interactive techniques. Proceedings of the 25th annual conference on, SIGGRAPH '98*, (New York, NY, USA), pp. 415–421, ACM, 1998.
- [65] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust, unions of balls, and the medial axis transform," *Computational Geometry: Theory and Applications*, vol. 19, pp. 127–153, July 2001.
- [66] T. K. Dey and J. Giesen, "Detecting undersampling in surface reconstruction," in *Computational geometry. Proceedings of the seventeenth annual symposium on, SCG '01*, (New York, NY, USA), pp. 257–263, ACM, 2001.
- [67] S. Muraki, "Volumetric shape description of range data using "blobby model"," in *Computer graphics and interactive techniques. Proceedings of the 18th annual conference on, SIGGRAPH '91*, (New York, NY, USA), pp. 227–235, ACM, 1991.
- [68] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," in *Computer graphics and interactive techniques, Proceedings of the 28th annual conference on, SIGGRAPH '01*, (New York, NY, USA), pp. 67–76, ACM, 2001.
- [69] C. Walder, O. Chapelle, and B. Schölkopf, "Implicit surface modelling as an eigenvalue problem," in *Machine learning. Proceedings of the 22nd international conference on, ICML '05*, (New York, NY, USA), pp. 936–939, ACM, 2005.
- [70] C. Shen, J. F. O'Brien, and J. R. Shewchuk, "Interpolating and approximating implicit surfaces from polygon soup," in *ACM SIGGRAPH 2005 Courses, SIGGRAPH '05*, (New York, NY, USA), ACM, 2005.
- [71] S. Lee, G. Wolberg, and S. Shin, "Scattered data interpolation with multilevel b-splines," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 3, pp. 228–244, jul-sep 1997.
- [72] J. Haber, F. Zeilfelder, O. Davydov, and H. P. Seidel, "Smooth approximation and rendering of large scattered data sets," in *Visualization, 2001. Proceedings of the conference on, VIS '01*, (Washington, DC, USA), pp. 341–348, IEEE Computer Society, 2001.

- [73] M. Bertram, X. Tricoche, and H. Hagen, "Adaptive smooth scattered-data approximation for large-scale terrain visualization," in *Symposium on Data visualisation, 2003. Proceedings of the, VISSYM '03*, (Aire-la-Ville, Switzerland, Switzerland), pp. 177–184, Eurographics Association, 2003.
- [74] G. M. Nielson, "Normalized implicit eigenvector least squares operators for noisy scattered data: radial basis functions," *Computing*, vol. 86, pp. 199–212, Oct. 2009.
- [75] G. R. DeVore, "Genetic sonography." http://www.focus77.com/obgyn/genetic_sonography.htm, 2005. Accessed: 2012-11-28.
- [76] K. A. Scanlan, "Sonographic artifacts and their origins," *American Journal of Roentgenology*, vol. 156, no. 6, pp. 1267–72, 1991.
- [77] M. K. Feldman, S. Katyal, and M. S. Blackwood, "US artifacts 1," *Radiographics*, vol. 29, no. 4, pp. 1179–1189, jul–aug 2009.
- [78] C. Burckhardt, "Speckle in ultrasound B-mode scans," *Sonics and Ultrasonics, IEEE Transactions on*, vol. 25, pp. 1–6, jan 1978.
- [79] D. Becker, M. Washburn, and S. O. Ishrak, "Speckle reduction imaging in high definition (sri-hd)," tech. rep., GE Healthcare, 2005.
- [80] K. Rohr, H. Stiehl, R. Sprengel, T. Buzug, J. Weese, and M. Kuhn, "Landmark-based elastic registration using approximating thin-plate splines," *Medical Imaging, IEEE Transactions on*, vol. 20, pp. 526–534, jun 2001.
- [81] J. Duchon, "Splines minimizing rotation-invariant semi-norms in Sobolev spaces," in *Constructive Theory of Functions of Several Variables* (W. Schempp and K. Zeller, eds.), vol. 571 of *Lecture Notes in Mathematics*, pp. 85–100, Springer Berlin / Heidelberg, 1977.
- [82] J. Elonen, "Thin plate spline editor - an example program in c++." <http://elonen.iki.fi/code/tpsdemo/>, 2005. Accessed: 2013-03-18.
- [83] M. Powell, "Some algorithms for thin plate spline interpolation to functions of two variables," in *Advances in Computational Mathematics*, (New Delhi, India), pp. 303–319, World Scientific, 1994.
- [84] J. Powell, *A Review of Algorithms for Thin Plate Spline Interpolation in Two Dimensions*. Cambridge DAMTP, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 1996.
- [85] O. Marques, *Practical Image and Video Processing Using MATLAB*, ch. 10. Wiley-IEEE Press, 1 ed., 2011.
- [86] L. Cupertino, A. Singulani, C. da Silva, M. Pacheco, and R. Farias, "LU decomposition on gpus: The impact of memory access," in *Computer Architecture and High Performance Computing Workshops, 2010. Proceedings of the 22nd International Symposium on*, pp. 19–24, oct 2010.