OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

Context-Aware 3D Model Generation for Biomedical Applications

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von Dipl.-Ing. Tobias Jürgen Mönch

geb. am 28.02.1982 in Magdeburg

Gutachterinnen/Gutachter

Prof. Dr. Bernhard Preim
Prof. Dr. Günther Greiner
Prof. Dr. Jos B.T.M. Roerdink

Magdeburg, den 27.06.2014

Dipl.-Ing. Tobias Jürgen Mönch
Context-Aware 3D Model Generation for Biomedical Applications
Dissertation, Otto-von-Guericke-Universität Magdeburg
Magdeburg, 27.06.2014

Gutachter:

Prof. Dr. Bernhard Preim
Institut für Simulation und Graphik
Otto-von-Guericke-Universität Magdeburg

Prof. Dr. Günther Greiner
Lehrstuhl für Informatik 9 (Graphische Datenverarbeitung)
Friedrich-Alexander-Universität Erlangen-Nürnberg

Prof. Dr. Jos B.T.M. Roerdink
Johann Bernoulli Institute for Mathematics and Computer Science
Universität Groningen

Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, den 27.06.2014

Tobias Jürgen Mönch

Kurzfassung

Anhand medizinischer Bildgebungsverfahren können Kenntnisse über die patientenspezifische Anatomie oder krankhafte Veränderungen gewonnen werden. Aus den akquirierten Schichtbilddaten lassen sich 3D-Modelle extrahieren, welche eine bessere Einordnung der anatomischen Strukturen in ihren räumlichen Kontext erlauben. Die Bilddaten unterliegen verschiedenartigen Artefakten (z.B. schlechtes Signal-Rausch-Verhältnis oder Inhomogenitäten der Intensitätswerte), welche die Generierung von korrekten Oberflächenmodellen erschweren. Durch den Einsatz von Segmentierungsverfahren kann der Einfluss derartiger Bildartefakte reduziert, jedoch nicht vollständig vermieden werden. Oftmals bleiben Treppen- und Terrassenartefakte in den erzeugten Oberflächenmodellen zurück. Diese stören einerseits die visuelle Wahrnehmung der anatomischen Strukturen und deren individueller Merkmale. Darüber hinaus haben sie Einfluss auf die Ergebnisse von Berechnungen (z.B. Distanzmessungen), welche auf Basis der 3D Modelle ausgeführt werden. Aus diesem Grund erfordern die meisten medizinischen Anwendungen (z.B. Planung und Simulation chirurgischer Eingriffe, Blutflusssimulation, Rapid Prototyping) genaue und artefaktfreie Oberflächenmodelle. Insbesondere der Einfluss von Treppenartefakten kann durch Glättungsfilter verringert werden, führt jedoch zu einem Verlust der Genauigkeit und relevanter Oberflächendetails.

Die vorliegende Arbeit untersucht charakteristische medizinische Anwendungen hinsichtlich ihrer individuellen Anforderungen an anatomische Oberflächenmodelle und diskutiert die Komplexität des zugrundeliegenden Modellgenerierungs- und Artefaktreduktionsprozesses. Dabei werden zwei Konzepte eingeführt, welche existierende Verfahren zur Glättung von Oberflächenmodellen robuster und genauer machen. Die kontextabhängige Glättung ermöglicht eine bessere Anpassung an die medizinischen Anforderungen durch das Einbringen von Kontextinformationen. Der Glättungsprozess wird auf bestimmte Oberflächenbereiche beschränkt, wodurch die Genauigkeit der Modelle besser erhalten werden kann und eine höhere Robustheit der Glättungsfilter erreicht wird. Dieser Gedanke wird ebenfalls durch eine effiziente Implementierung von Glättungsfiltern fortgeführt, die den Benutzer während der Glättung und Bewertung der erreichten Modellqualität unterstützt. Die Modellqualität wird für zahlreiche Ausprägungen der Eingabeparameter automatisch bestimmt und in Form verschiedener Qualitätsdiagramme dargestellt. Neben der Interaktion, dem visuellen Feedback und der Darstellung der zu erwartenden Modellqualität können plausible Parameter- und Glättungsvorschläge generiert werden.

Abstract

Medical imaging allows to obtain patient-specific information on anatomy and pathologic variations. 3D surface models may be extracted from the acquired tomographic image data and finally be employed for a better perception of the spatial context. Different artifacts in the image data (e.g., image noise or intensity inhomogeneities) complicate the extraction of accurate surface models. The target structures may be segmented to reduce the impact of the image artifacts, but segmentation may not completely prevent from erroneous surface models. Usually, artifacts, such as staircases and terraces, remain to the surface models. They disturb the perception of the overall shape and affect the results of computational tasks (e.g., distance measurements). Thus, accurate, artifact-free surface models are required for most medical applications, such as surgical planning and simulation, blood flow simulation, or rapid prototyping. In particular, staircase artifacts can be reduced via mesh smoothing filters, which often degrade accuracy and remove relevant details from the surface.

The presented thesis analyzes the characteristic medical applications and their requirements with respect to anatomic surface models and discusses the complexity of the underlying model generation and artifact reduction procedure. In this context, two concepts are introduced, which provide more robustness and a better preservation of accuracy to common mesh smoothing filters. *Context-aware* smoothing allows for a better adaption of smoothing filters to the medical requirements by involving context information. Smoothing is restricted to certain surface areas, which better preserves accuracy and yields more robustness during smoothing. The idea of improving the robustness is continued by *visually-guided* smoothing, which provides an efficient implementation of smoothing filters. The latter assists the user during smoothing and model evaluation, which is achieved by automatically evaluating the model quality for numerous parameter sets and providing this information as *model quality graphs* and *bars* to the user. Besides interaction, visual feedback and model quality preview, plausible parameters and smoothing suggestions can be generated.

Danksagung

Ich möchte mich vor allem bei meinem Betreuer und Doktorvater Prof. Dr. Bernhard Preim bedanken. Während des gesamten Promotionszeitraums stand er mir immer wieder mit Anregungen, Tipps und auch Kritik zur Seite und ermöglichte gleichzeitig das Verfolgen und Umsetzen eigener Ideen. Den (ehemaligen) Kollegen der Arbeitsgruppe Visualisierung möchte ich für das angenehme Arbeitsklima und motivierende Umfeld danken.

Insbesondere Christoph Kubisch hatte immer zahlreiche Ratschläge parat, wodurch meine Ideen in entsprechender Weise umgesetzt und weiterentwickelt werden konnten. In diesem Rahmen möchte ich mich auch bei Kai Lawonn für die vielfältigen kreativen Diskussionen und die Unterstützung bei der Suche nach Lösungsansätzen bedanken. Simon Adler bin ich sehr dankbar für die tolle Zusammenarbeit im Projekt VIERFORES. Seine Unterstützung und Motivation haben vielfach zum Vorankommen des Projektes beigetragen. Mein Dank gilt auch Prof. Dr. Theisel und Prof. Dr. Westermann für ihre Unterstützung bei der Einordnung und Ausrichtung meiner Arbeiten. Ein besonderer Dank geht an Stefanie Quade für das (häufig kurzfristige) Korrekturlesen meiner Arbeiten während der gesamten Promotionszeit.

Ich bedanke mich bei meiner gesamten Familie und insbesondere bei meinen Eltern, die immer für mich da waren und mich jederzeit bedingungslos unterstützt haben. Allen voran möchte ich aber meiner Frau Jeanette und meiner Tochter Pauline danken. Sie haben mich immer wieder motiviert und auch in den richtigen Momenten auf andere Gedanken gebracht.

Contents

1

Introduction

Table of Contents

Medical imaging, such as computed tomography (CT) or magnetic resonance imaging (MRI), allows to obtain patient-specific information on anatomy and pathologic variations. Functional properties may be recorded in a non-invasive way via further imaging modalities, such as positron emission tomography (PET), functional MRI (fMRI), or single-photon emission computed tomography (SPECT). For the planning of surgical interventions or radiation treatment, the morphology of anatomic structures, their spatial relations, as well as their pathologic variations are examined. This comprises different exploration tasks where, e.g., radiologists or surgeons view the acquired tomographic image data slice-wise and build a mental model of the 3D structures. Unfortunately, the precise spatial relation of an anatomic structure towards surrounding tissue and structures is not always obvious from these 2D data. In particular for surgeons who are not trained intensively for analyzing image data, this is a non-trivial task.

In, e.g., ear-neck-throat (ENT) surgery, the location and extent of tumors and lymph nodes, as well as their distance to neighboring risk structures, such as vessels or muscles, are evaluated. Any injury to such risk structures may be a severe risk to patient health. Similarly, the planning of resection planes for live-donor liver transplantations requires detailed knowledge on the liver's vascular system. Thus, information on the topology, location, and diameter of vessel branches is essential. The exploration of the 2D data alone does not provide enough insight to these spatial properties.

3D surface models extracted from the tomographic image data may be employed for a better perception of the spatial context. Medical image data usually suffers from image noise and intensity inhomogeneities, which are often emphasized, since radiation exposure and examination times need to be reduced. These data artifacts can cause frayed or disconnected surfaces or lead to blending of closely located surfaces. Thus, segmentation of the target structures is required to obtain faithfully reconstructed surface models. The delineation of a target structure often yields binary image data. A direct reconstruction of surfaces from binary data causes strong aliasing artifacts and leads to "staircases" and "terraces" instead of smooth anatomic surfaces. Such discontinuities in the surface attract the observers' attention and might severely disturb the perception of the overall shape and structure of the surface. Moreover, surface artifacts have an impact on the results of computational tasks (e.g., distance measurements). Accurate, artifact-free surface models are required, e.g., for the simulation of blood flow in vascular systems. Any surface discontinuity may lead to non-converging simulations or wrong simulation results for the flow behavior. The latter might then yield wrong conclusions for treatment and therapy planning.

Surface artifacts can be reduced at different stages of the model generation process – by processing the image data, during surface extraction, and by postprocessing the surface models. Unfortunately, a lot of manual effort is necessary to fit the artifact reduction procedure to the requirements and properties of patient-specific data and the intended application. The shape and size of anatomic structures plays an important role within this process. As an example, image contrast inhomogeneities may have less impact on large, compact structures (e.g., a liver) than on vascular structures which may suffer from narrowed parts, disconnected, or blended branches. In turn, compact structures may be sensitive to aliasing artifacts and noise. Artifact removal makes two basic assumptions on the correct anatomic surface: continuity and smoothness. For example, anatomic surfaces typically do not contain holes or exhibit sharp edges. Further constraints may be given by the specific shape of the considered structure, e.g., vascular structures may branch, but may not contain loops.

Unfortunately, model generation may also evoke inaccuracies, since each step of the pipeline follows given shape assumptions or since the dataset is processed with uniformly set parameters which affects also non-artifact areas. From the literature, there are many methods available to solve specific problems. The known methods usually do not consider the specific requirements of anatomic structures or do not involve context knowledge for an application-specific artifact reduction. An appropriate tradeoff between high accuracy and perfect artifact removal is often hard to achieve and requires sophisticated knowledge on the underlying algorithms and their parametrization. Model generation for medical applications is thus a complex procedure involving a lot of interaction steps and expert knowledge. Additionally, application-driven constraints might help to make artifact removal more robust and thus to better support users.

In this context, Bade [2007] performed a sophisticated analysis on smoothing filters w.r.t. their capabilities to reduce artifacts and preserve accuracy at the same time. As a result, parameter suggestions could be derived for different categories of anatomic shapes. It became clear that the existing smoothing filters require a better adaption to the specific requirements of medical visualization and its related tasks. A first step towards such an adaption has been made by locally restricting the changes of smoothing filters [Bade et al., 2007a]. The thesis takes the results and related conclusions of Bade [2007] as fundamental input for improving the handling and robustness of smoothing filters and a better adaption to the requirements of medical visualization tasks.

1.1 Goals

The generation of surface models provides numerous options to adjust surface extraction to different needs. Within this thesis, the requirements defined by biomedical applications, such as surgery and treatment planning or the simulation of blood flow, will be examined. Model generation will be discussed for different exemplary medical applications. Since especially tomographic medical image data and subsequent surface extraction may evoke different surface discontinuities, this work focuses on the reduction of artifacts in anatomic surface models.

Within this field, the smoothing of surfaces for a reduction of noise and aliasing artifacts is one of the most commonly used operations. Surface smoothing filters may strongly affect model accuracy. For convenient results, knowledge on the specific algorithmic properties (e.g, the meaning of different parameters) and their impact on differently shaped structures is required. Moreover, if model generation and thus smoothing is performed within a clinical workflow, such deep knowledge on model generation, related filters, and their specific parameters cannot be assumed.

Based on that, the goal of this thesis is to *provide more robustness to surface smoothing filters*. This allows for a better integration into medical workflows and may reduce manual effort. An improvement of the robustness and performance of smoothing filters may be achieved in different ways. The handling in terms of interaction with smoothing filters varies from method to method. Thus, it is relevant to come up with a concept which supports users to find suitable tradeoffs for their desired application. An automated evaluation of smoothing filters with varying input parameters is computationally very expensive due to the potentially large number of parameter combinations and high surface resolution. For model generation workflows embedded into clinical routine, long delays for parameter evaluation and adjustments are not acceptable. Against this background, this thesis shall answer the following questions:

1. *How can users be supported during the use of surface smoothing filters?*

2. *Is there an efficient way to provide individual smoothing suggestions for patient-specific data?*

The design of surface smoothing filters is usually motivated from domains other than medical visualization. These filters try to detect and preserve sharp features, whereas noise is removed from the surface. In contrast, surfaces extracted from tomographic medical image data contain aliasing artifacts, such as staircases and terraces. Since anatomic surfaces typically exhibit smoother shapes, the assumption that sharp features need to be preserved is misleading. Smoothing of such surfaces is often not satisfying with respect to smoothness and accuracy requirements. Specific artifacts might only be removed faithfully, if additional context information is considered. This may comprise information on the image data modality, application-dependent goals, spatial, or anatomic properties. Thus, another question to answer within this thesis is:

3. *How can context information be included into the smoothing procedure for a better consideration of medical requirements and anatomic constraints?*

These research questions focus on surface smoothing primarily. However, a generalization of the solutions to these problems will be discussed.

The investigations within this work consider different medical use cases. These are namely:

- The *planning of surgical interventions and treatments* primarily requires accuracy, but also naturally looking 3D models to prevent focusing of the observer's attention to artifact areas.

- *Computational fluid dynamics (CFD) for blood flow simulation in vascular systems* requires accuracy and artifact-free surfaces for convenient simulation results.

- *Surgical simulation with patient-specific data* requires naturally looking surfaces. The individual shape features may not be omitted during model generation to resemble the considered clinical case. However, smoothness is also relevant for the underlying simulation.

- *Rapid prototyping of anatomic structures* builds physical models from the 3D surfaces. Individual anatomic shapes are required for the training of intervention techniques, procedures and teaching.

The presented thesis evolved from the requirements and work within the ViERforES[1] and ViERforES-II[2] projects. Among other topics, one sub-project treated the requirements of surgery simulation in the context of minimally-invasive surgery (see also Sec. 2.2.1). This particular application considered the generation of surface models from patient-specific data for subsequent usage within a virtual surgery simulation environment. In his PhD thesis, Simon Adler summarized the research and results from the development of this virtual training and testing environment [Adler, 2014]. Patient-specific surface models were generated with the methods presented in the following chapters and served as input for the simulation environment and related preprocessing steps.

1.2 Structure

Chapter 2 describes the above-mentioned medical applications in detail and puts the spot on their specific requirements to surface models of anatomic structures.

For a better understanding of the causes of artifacts in anatomic surfaces, **Chapter 3** gives an overview of the general data processing pipeline. This overview treats the acquisition of tomographic image data, image processing, and surface extraction with respect to surface artifacts. As one of the most commonly applied techniques to reduce staircase artifacts, surface smoothing is discussed.

Against this background, **Chapter 4** describes some difficulties of an exemplary model generation pipeline for CFD and rapid prototyping. The chapter reveals the tremendous effort which is often necessary to prepare the surface model for a specific application.

Chapter 5 introduces *context-aware* smoothing, which represents an extension to common mesh smoothing filters for a better adaption of such filters to anatomic surface models. By adding context information, smoothing can be restricted to certain surface areas which allows for a better preservation of accuracy and yields more robustness during smoothing.

The robustness of smoothing filters is also treated in **Chapter 6**, where an efficient implementation of smoothing filters is used to assist the user during smoothing and model evaluation. This is achieved by evaluating model quality in real-time and providing this information as *model quality graphs* and *bars* to the user. Besides visual feedback and interaction, the evaluation of the model quality allows to provide model-specific, plausible smoothing suggestions.

Chapter 7 concludes this thesis by summarizing the contributions and by providing an outlook to possible further extensions of the presented work as well as related research topics.

[1] ViERforES - “Virtuelle und Erweiterte Realität für höchste Sicherheit und Zuverlässigkeit von Eingebetteten Systemen”, funded by the German Federal Ministry of Education and Research (BMBF, grant no. 01IM08003C)

[2] grant no. 01IM10002B

2

Applications for Medical Surface Models

Table of Contents

Surface models of anatomic structures are basic elements in different medical treatment and training scenarios. Each single application contains specific workflows and raises different requirements. The goal of this chapter is to introduce typical applications for medical surface models. For this, the anatomic background and clinical requirements are discussed with respect to constraints for the use of 3D surfaces. The latter represent one kind of indirect volume visualizations, which means that an intermediate representation of the original dataset is used to depict properties, such as shapes or spatial relations [Preim and Bartz, 2007]. Among others, anatomic surface models are employed in the following applications:

- fast rendering of specific target structures,
- rendering in mobile or web-based environments,
- planning of surgical interventions or radiation treatment,
- simulation, e.g., of tissue behavior or blood flow,
- preparation for building physical models via rapid prototyping

Surface rendering is only suited for structures which can be described by a clear boundary. For structures with fuzzy boundaries (due to their material properties or as a result of image acquisition), direct volume rendering is often more appropriate. Direct volume visualization techniques show the voxels of the volume data without extracting an intermediate representation, such as an isosurface. A detailed insight into direct volume rendering is, e.g., given by Preim and Bartz [2007]; Preim and Botha [2013].

This thesis focuses on anatomic surface models and their applications, which are discussed in the following sections.

2.1 Planning of Surgical Interventions and Treatment

The morphology of anatomic and pathologic structures and their spatial relations are usually examined for the planning of surgical interventions or radiation treatment. Depending on their experience, physicians are able to interpret the gray values in tomographic image data and transform the voxels into a mental 3D representation. Due to a strong inter-patient variability of anatomic structures or even due to anatomic changes of one patient over time, this is a non-trivial task. For a successful intervention or treatment, the extents of malignant structures, such as a tumor or metastasis, or the morphology of branching anatomic structures, such as vascular or bronchial trees, has to be known as precisely as possible. For instance, the surgical removal of a tumor requires an exact knowledge on the spatial relations to surrounding vessels to prevent them from being injured. Additionally, it has to be ensured that the malignant tissue is completely removed and no tumor cells remain. Otherwise, these cells may spread and yield anew tumor growth.

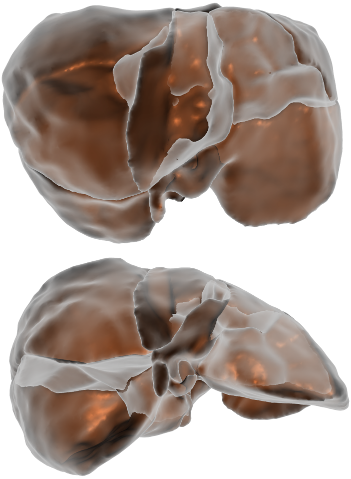


Figure 2.1: *A surface model of a liver rendered from two different view points. The rendering emphasizes the liver segments according to the drainage via the portal vein.*

2.1.1 Liver Surgery

Pathologic deviations, such as tumors, may evolve in many different tissues and organs. Depending on this location in the human body and thus their effects on the affected organs and distance to vasculature, these deviations result in varying mortality. As an example, liver carcinoma were the sixth (men) and 11th (women) most cause of death in Germany in 2010 German Cancer Research Center (DKFZ) [2012]. For their treatment, conventional surgical interventions, minimally-invasive (laparoscopic) interventions, or thermoablations are commonly used. The specific decision for one of these or other treatment options depends on the exact location of a tumor or metastases. The liver contains a complex vascular system, which supplies it with blood via the hepatic artery and drains it via the hepatic veins. According to this vasculature, the liver can be subdivided into different segments (see Fig. 2.1) [Couinaud, 1957]. Any surgical intervention leads to injuries of this vasculature and, hence, the respective liver segment and tissue.

The goal of an intervention is the *complete* removal of the malignant tissue while as much healthy tissue as possible shall be maintained. At least 30-40% of healthy liver tissue must remain in order to guarantee a sufficient functionality. To ensure a complete removal of the tumor, tissue within a safety margin (usually 10 mm) around the known tumor is additionally removed. Such margins may vary, e.g., according to the distance to nearby hepatic vessels. Any intervention represents a tradeoff between a safe tumor resection and a maximum preservation of healthy tissue. Thus, a careful planning of resection strategies and safety margins is required. The decision on the feasibility of a surgical intervention and the resection strategy is often made based on tomographic

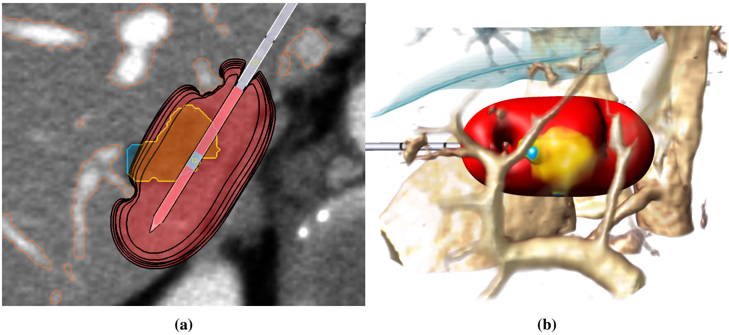
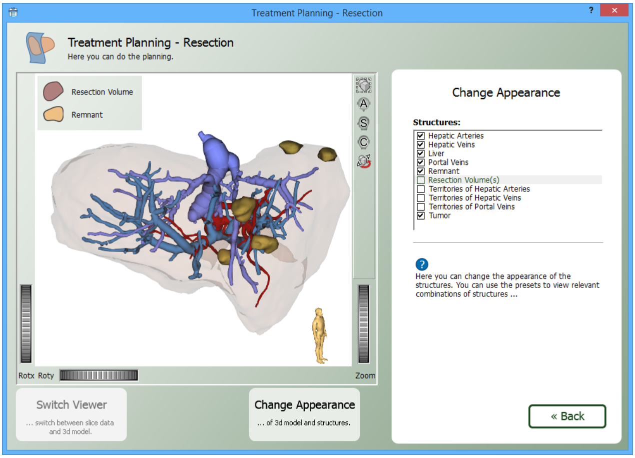


Figure 2.2: *Planning of a radiofrequency ablation. Left: The applicator and the segmented tumor are overlayed to the image data. The ablation zone is shown in red. Right: The same situation is shown in 3D for better perception of the spatial relations during interaction. © 2011 IEEE. Images reprinted, with permission, from [Rieder et al., 2011].*

image data. The reading of the image data slices provides detailed insight to the specific anatomy, but the spatial relation between the complex vasculature and one or several lesions is difficult to judge. This judgment is further complicated by a strong variation of patient anatomy and anatomic changes caused, e.g., by tumor growth [Lamade et al., 2000].

Computer-based planning can be employed to analyze and detect the risk areas which would not be supplied or drained after resection [Lang and Schenk, 2010]. As an example, Meinzer et al. [2002] generate resection suggestions by considering the main stems of the portal and venous trees according to the Couinaud classification. Preim et al. [2002] take the individual anatomy of the portal and hepatic vein into account for a more precise suggestion. Such resection planning is also required for the split-liver technology [Broelsch et al., 1990], where a patient receives a part of a healthy liver. Again, the patient-specific anatomy and its vasculature need to be considered to preserve blood supply and drainage after intervention – this time for both, the transplant recipient and the donor.

An alternative to such a conventional intervention is the thermoablation, which comprehends the destruction of the malignant tissue by a local application of heat, e.g., via radio-frequency pulses [Pereira, 2007], laser beams [Gillams et al., 2005], or microwaves [Tanaka et al., 2006; Hompes et al., 2010]. Although this procedure reduces the invention risk, other organs and vessels may still be injured during insertion of the applicator or during heat application. The access path, the optimal needle placement (with respect to the tumor), as well as the dosage need to be planned to consider the effects on the surrounding tissue and, in particular, nearby vessels. During this procedure, the visual exploration of the data and ablation zone, as wells as the interactive placement of the applicator are major tasks involving 3D surface models (e.g., [Rieder


Treatment Planning - Resection
Resection Volume
Remnant
A
S
C
Rotx Roty
Zoom
Switch Viewer
... switch between slice data and 3d model.
Change Appearance
... of 3d model and structures.
« Back

Treatment Planning - Resection

Here you can do the planning.

Change Appearance

Structures:

- ☑ Hepatic Arteries
- ☑ Hepatic Veins
- ☑ Liver
- ☑ Portal Veins
- ☑ Remnant
- ☐ Resection Volume(s)
- ☐ Territories of Hepatic Arteries
- ☐ Territories of Hepatic Veins
- ☐ Territories of Portal Veins
- ☑ Tumor

Here you can change the appearance of the structures. You can use the presets to view relevant combinations of structures ...

Figure 2.3: *Screenshot of the LiverSurgeryTrainer [Mönch et al., 2013a]. The planning of a resection is shown with the planned resection volume in brown and the remnant as white, transparent surface.*

et al., 2011]). Figure 2.2 depicts the relevance of 3D models for such exploration and interaction tasks during intervention planning.

Intervention planning serves as preparation prior to an intervention, but may also be used as additional information in the operating room. It is often performed as a service as, e.g., provided by MEVIS DISTANT SERVICES[1] or the German Cancer Research Center[2]. For the planning, image data need to be prepared (segmented) and 3D surface models are generated. This is often done by staff members with medical visualization background, radiology technicians, or even surgeons. It is obvious that the planning of such interventions requires high accuracy during image data processing and subsequent model generation for a correct estimation of, e.g., resection planes or ablation zones.

In relation to liver surgery (planning), training and learning environments have evolved to convey the specific anatomy and resection/intervention strategies. These systems (e.g., INTERACTIVE 3D LIVERANATOMY [Crossingham et al., 2009], LIVERSURGERYTRAINER [Logge et al., 2007; Mönch et al., 2013a] (see Fig. 2.3), SURGERYNET[3], WEBOP[4]) convey anatomic knowledge, specific clinical cases, and related treatment options to physicians and medical students. Relevant intervention strategies may then be learned for the presented cases.

[1] http://www.mevis.de/mms

[2] http://www.dkfz.de

[3] http://projekt.surgerynet.de/

[4] http://www.webop.de

2.1.2 Ear-Neck-Throat Surgery

Cancer of the oral cavity and the pharynx had a mortality rate of 5.1% (men) and 1.2% (women) among all cancer deaths (2010, Germany) German Cancer Research Center (DKFZ) [2012]. The ear, neck, and throat (ENT) region contains several relevant, but closely located structures. Namely, these are the trachea, pharynx, larynx, salivary glands, lymph nodes, muscles, nerves, and vessels (see Fig. 2.4) [Köpf-Maier, 2000]. Due to the closeness of these anatomic structures, the growth of malignant tissue may rapidly affect neighboring structures. In addition to the tumor, cancer cells may spread to other parts of the body via the lymphatic system (leading to lymph node metastasis) and via vasculature into other organs and tissues. Thus, treatment aims at a complete removal of the tumor and of lymph node metastases to prevent further spreading. Depending on the location and size of the tumor and the existence of enlarged lymph nodes as an indicator for lymph node metastases, different treatment options are available, ranging from brachytherapy to radical surgical interventions. In the latter case, a loss of function of relevant structures and hence a loss of quality of life has to be expected. The removal of tumors and metastases from the neck is called neck dissection [Bocca et al., 1984]. According to the AMERICAN ACADEMY OF OTOLARYNGOLOGY – HEAD AND NECK SURGERY [AAO-HNS, 2012], neck dissections are classified as follows:

1. Radical neck dissection
2. Modified radical neck dissection
3. Selective neck dissection
4. Extended neck dissection

These are different intervention strategies, which account for the tumor's size, location (with respect to surrounding tissue), as well as the number, size and location of lymph node metastases. Thus, depending on the information gained from preoperative image acquisition, but often also intraoperative information, the resection strategy which minimizes the critical consequences for the patient is chosen. Although neck dissections are just one example for surgical interventions in the ENT region, it is clear that the estimation of spatial extents of specific structures as well as the distance to neighboring structures plays an essential role for the planning of many interventions. For instance, Hansen [2012] presented new techniques for risk assessment in liver surgery planning.

The distances are usually estimated on both the image data and derived surface models. In particular the latter are relevant for a visual exploration, where the viewer can rotate the rendered scene and zoom into specific regions. This assumes that the surface models have been generated with preferably high accuracy in terms of surface details and extents. Such a visual inspection requires a natural look of the surface models, which comprises the absence of sharp features. These are usually not contained in anatomic structures and would attract the observers' attention due to shading discontinuities. This is not critical for measurement tasks, but might severely disturb the perception of the overall shape and structure of the surface model(s) [Preim and Botha, 2013].

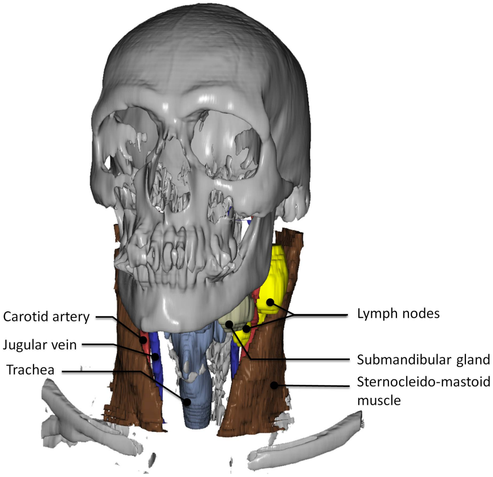


Figure 2.4: *Selected surface models generated from segmented CT data. The shown models represent only a small subset of the structures in the neck.*

2.2 Simulation

The tasks evolving during intervention and treatment planning depend on the specific disease and the affected anatomic structures. In the former sections, planning was basically performed using spatial information gained from assessing the extents of structures and distances to neighboring structures. Such static information is often not sufficient, e.g., if different intervention strategies shall be evaluated or if dynamic behavior, such as blood flow or tissue deformation, needs to be considered. The following sections will describe these settings by some examples.

2.2.1 Surgery Simulation

The simulation of surgical interventions comprehends the computer-based imitation of an intraoperative situation for practicing and training different skills [Kühnapfel et al., 2000; Lehmann et al., 2012]. Typical examples are:

- the training of intervention techniques, e.g., as part of the medical education or as preparation for a specific, potentially complicated intervention,
- the evaluation of intervention strategies and access paths for a specific clinical case, or
- the training and testing of different or new surgical instruments.

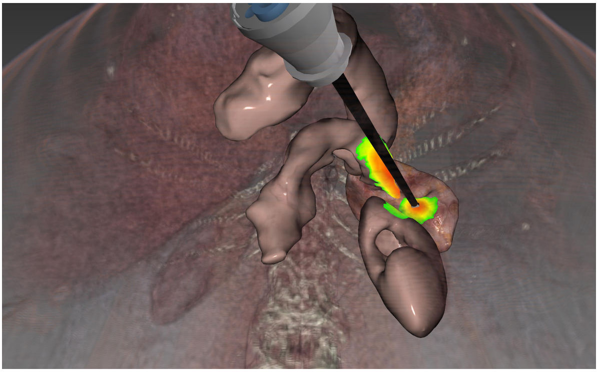


Figure 2.5: *Screenshot of a simulation environment for the training of minimally invasive surgery and testing of related instruments. For training purposes, the view includes a color-coded highlighting of structures close to the instrument (Courtesy of Simon Adler, Fraunhofer IFF Magdeburg).*

Surgery simulation is employed to improve surgical skills, e.g., in minimally-invasive surgery. Laparoscopic interventions are often simulated to practice the complicated handling of the instruments or to evaluate different paths for accessing the target area (e.g, a tumor located on the dorsal side of the liver). Especially the latter serves for assessing the feasibility of an intervention by virtually practicing the complete intervention procedure. Thus, surgery simulation reduces the risk for the patient to be exposed to preventable errors [Ayodeji et al., 2007; Aggarwal et al., 2009]. The virtual intervention should exhibit the same characteristics as the real intervention. This comprises:

- a virtual environment with all relevant structures to visually provide context information and to resemble the real spatial conditions,
- visually plausible behavior of the tissue during interaction with the virtual instruments and organs, and
- a plausible haptic feedback to the user during interaction with the virtual environment via dedicated input devices.

These properties are realized in most surgery simulators. Commercial simulators are offered by, e.g., SIMBIONIX[5], SURGICAL SCIENCE[6], or CEA HEALTHCARE[7]. Sutherland et al. [2006] and Preim and Botha [2013] provide a detailed overview of available surgical simulators from different fields and related studies comparing the training effects of these surgery simulators.

[5] http://simbionix.com

[6] http://www.surgical-science.com

[7] http://www.cae.com

For all of these environments, surface models are a basic prerequisite. They are employed for visualizing spatial context and build the fundamental basis for all simulation-related computations. The latter requires the generation of volume meshes to describe the properties of the tissue (see Sec. 3.3.3). For obtaining a volume mesh, (segmented) volume data has to be transformed into a surface mesh. This surface mesh bounds the volume for which the volume mesh is generated. Of course, the accuracy of the surface mesh affects the overall accuracy of the volume mesh. However, very small surface details yield a locally high volume mesh resolution, which increases computational effort and may even prevent from convergence of the underlying computations. The triangle quality and homogeneity of the triangle size and shape distribution affect the simulation in the same way. A further description of triangle quality is given in Section 3.3.

Surgical simulators have to cope with a number of complex problems. A realistic-looking visualization is a basic requirement. Tissue simulation and haptics between user-controlled devices and the simulated environment pose high demands on the performance of the algorithms and the system hardware. Due to this, tissue simulation and haptics are often performed with moderately sampled, less accurate surface and volume meshes.

Most of the available commercial simulators (refer to [Preim and Botha, 2013] for an overview) employ predefined surface and volume models for visualization and simulation. In particular for the training of surgical skills, the usage of predefined 3D models is often sufficient. If, however, specific pathologic variations have to be considered, patient-specific data may be employed (see Fig. 2.5). The preparation of surface and volume models for such an individual simulation requires sophisticated anatomic knowledge and powerful software tools. An overview of simulators involving patient-specific data is given by Soler and Marescaux [2008]. For instance, Adler [2014] developed several algorithms for preparing patient-specific surface models for surgery simulation within a virtual testing and training environment [Adler et al., 2012]. The modeling of dynamic behavior of patient-specific data and in particular of vascular structures has been of special interest. As a part of the collaboration within the ViERforES and ViERforES-II projects, the methods presented in the following chapters were integrated into the model generation pipeline for this simulation environment. The exchanged surface models were required to depict the patient-specific characteristics. On the other hand, the surfaces should exhibit a low resolution to guarantee high rendering and simulation performance, since the behavior of multiple structures and tissues was simulated at the same time. Thus, the underlying surface model generation workflow was focused on initially removing artifacts (e.g., via selective mesh smoothing) and subsequently reducing the mesh resolution.

2.2.2 Blood Flow Simulation

Vascular diseases, such as the coronary heart disease or abdominal and cerebral aneurysms (see Fig. 2.6a), are severe, often life-threatening diseases. Initiation and progress of these diseases are not fully understood. Many research projects clearly show that such diseases primarily occur at regions of complex and instable blood flow. Blood flow information from patient-specific data is essential for understanding the formation and progress of vascular diseases [Preim and Botha, 2013]. Functional properties of vascular structures need to be evaluated for selecting an appropriate therapy.

Blood flow data may be measured directly, e.g., via MRI scanners and specialized imaging sequences. To examine the development of vascular pathologies, the visualization and exploration of measured flow data becomes increasingly important. This is demonstrated, i.e. by van Pelt et al. [2011], who presented an approach for exploring cardiovascular MRI blood flow data. Illustrative visualization techniques have been used for rendering particles, integral lines and integral surfaces to convey the blood flow characteristics and enhance the perception of the complex flow dynamics.

In addition to measured flow data, CFD simulations of blood flow are often employed to understand certain vascular malformations and their development. The analysis of hemodynamic characteristics is thus often subject to the clinical research of biomedical engineers, cardiologists, or neuroradiologists. For example, cerebral aneurysms result from a local weakness of the vessel wall. If an aneurysm ruptures and bleeds into the space around the brain (subarachnoid hemorrhage), the outcome for the patient may be critical [Brisman et al., 2006]. Due to this risk, an appropriate treatment is necessary, but the intervention risk is usually very high. If a reliable prediction of the rupturing risk is not possible and no severe symptoms are observed, an intervention is usually postponed.

According to the literature [White and Wardlaw, 2003; Wiebers, 2003; Cebral et al., 2011], several factors are known to affect the rupturing risk, where the most relevant ones are:

- aneurysm size
- location
- hemodynamics

Each factor may not be considered distinct from the others. As an example, both, small and large aneurysms may rupture, even if larger ones have a higher rupturing risk [Cebral et al., 2011]. Typical treatment options comprise stenting, coiling, or the clipping of aneurysms [Currie et al., 2010; Bradac et al., 2012]:

- *Stenting* comprises the insertion of a tubular mesh into the vessel, which is usually used to support the vessel wall. However, stents may be designed as flow diverters, which are placed to cover the neck of an aneurysm to alter the hemodynamics and support thrombosis formation in the aneurysm [Janiga et al., 2012].
- *Clipping* has been the standard procedure for many years and comprehends the usage of a small metal clip applied to the aneurysm neck to prevent further blood flow into the aneurysm.

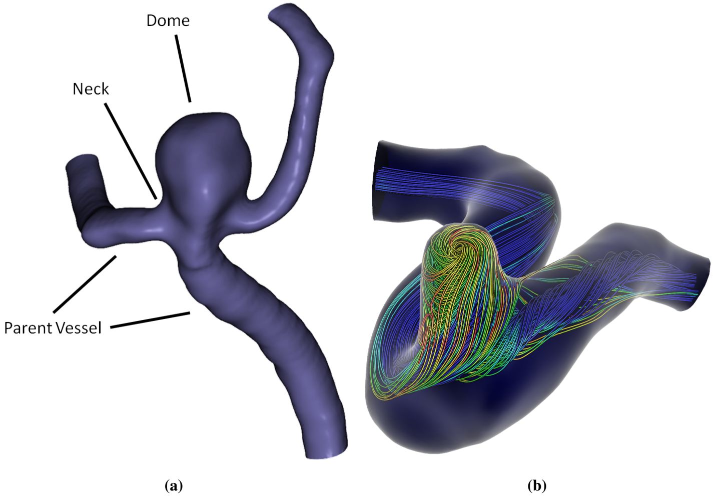


Figure 2.6: *Left: Example of an aneurysm surface model with a clipped parent vessel, the aneurysm dome, and the aneurysm neck. Right: a cerebral aneurysm surface model (semi-opaque) with a visualization of simulated blood flow via streamlines. The vessel has been clipped on both sides of the aneurysm to restrict the simulation to the relevant volume.*

- *Coiling* is the insertion of one or more coils into the aneurysm to support thrombosis formation within the aneurysm. This reduces the rupturing risk and might even prevent any rupture, since pressure is taken from the vessel wall [Seshadhri et al., 2011]. Depending on the aneurysm shape, an additional stent may be inserted to the parent vessel to prevent the coil(s) from dropping out.

Besides information on the current hemodynamics prior to an intervention, the simulation of hemodynamics via CFD may additionally help to select from the available treatment options or to modify them. By doing so, the placement of flow diverters and stents may be planned in advance. The blood flow is simulated based on 3D models of the patient-specific vasculature, extracted from CT or MRI [Schwenke et al., 2011] data. Likewise, 3D models of different stents may be inserted to aneurysm models for estimating each stent's impact on the blood flow inside the aneurysm. An example of simulated blood flow visualized via streamlines is shown in Figure 2.6b.

For a faithful simulation, the employed models have to be accurate with respect to their shape and size [Cebral et al., 2001, 2005]. Again, the surface meshes form the basis for volume mesh generation. Thus, fine details on the surface influence the granularity of the resulting volume mesh and finally the quality of CFD results (e.g., wall shear stress, flow velocity). To obtain meaningful simulation results, the following requirements are essential [Preim and Botha, 2013]:

- The surface models have to be as *accurate* as possible to represent all relevant anatomic and pathologic characteristics.
- *Smoothness* is required, since fine details and small surface artifacts may yield wrong simulated blood flow properties [Mönch et al., 2011a].
- The *mesh resolution* should be high enough to represent all relevant surface details.
- A *high triangle quality* is required to prevent from non-converging simulations. The triangle quality has impact on the quality of cells of the volume mesh, which is the basis for all simulation computations.
- *Homogeneity* with respect to triangle size is essential, since abrupt changes of the size of neighboring triangles would also affect volume mesh generation and thus the simulation.
- To reduce the simulation complexity, the simulated volume should be *restricted to the relevant parts*.

The extraction of the individual patient vasculature and generation of adequate 3D models is an essential step within the complex pipeline for the evaluation of different treatment options. Typical problems arising during the generation of vascular surface models for usage in CFD are further explained in Chapter 4.

2.3 Rapid Prototyping

Rapid prototyping (RP, also: solid freeform fabrication, layered manufacturing) [M. Gurr and R. Mülhaupt, 2012] comprises the building of physical models for different applications. In the medical domain, these are treatment planning, simulation, measurement tasks, or even teaching and training of interventional techniques. The models are used, e.g., for the planning of heart valve replacements [Schmauss et al., 2012], for stent implantation planning, or for educational purposes [Knox et al., 2005]. For instance, inverse transparent silicon models of anatomic structures are used to experimentally simulate blood flow. Optical velocimetry methods are then applied to gain information about the complex flow patterns within the different vessel configurations [Tateshima et al., 2001].

Models from RP are also used for the development of protheses and reconstruction of bones. For example, in oral and maxillofacial surgery, bone replacements are planned using RP technology [Joffe et al., 1999; Winder, 1999; D'Urso et al., 2000]. In a clinical case documented by Gibson et al. [2006], a portion of the jawbone was missing on one side of the patient's face. This missing part has been remodeled virtually and afterwards a physical model has been created. This model was then used to fit a flexible titanium mesh to the estimated patient-specific bone shape. The titanium mesh was implanted to remodel the patient's face and to improve life quality. In another case, RP has been used for planning an intervention to separate conjoined twins [Objet Ltd. and UCLA Medical Center, 2012]. Many cases document that RP models are frequently used for the planning and simulation of complex surgical tasks [Petzold et al., 1999]. Even orthodontic aligners are industrially produced via RP, e.g., by ALIGN TECHNOLOGIES[8].

It is obvious that RP technology has a wide range of applications. For each specific case, the production process may vary. Most methods have in common that surface models serve as input. As in the previous sections, these surface models are usually generated from patient-specific image data. However, a precise physical reproduction of all surface features is often not possible due to procedural- and material-related constraints.

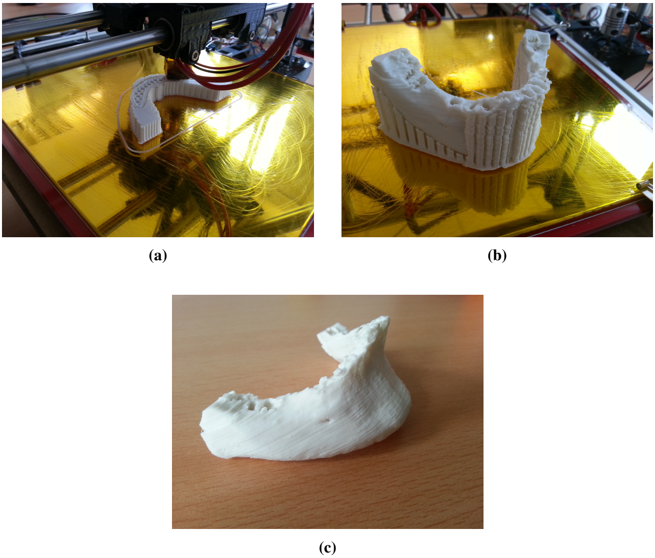
As an example, the initially mentioned inverse transparent silicon models, e.g., of vascular structures, may be generated by constructing a mold via RP and subsequently filling it with a low melting material. The resulting cast is enclosed by silicon and finally the cast is removed from the silicon block by melting. Depending on the specific material, the final inverse silicon model and all intermediates of this process can represent surface details only to a certain degree. Additionally, it must be ensured that the mold can be opened without destroying the cast. In turn, it must be possible to remove the cast, without melted parts remaining in the silicon block. Thus, in order to reconstruct a physical model from the reconstructed surface, it must satisfy certain constraints, e.g., adequate distances between adjacent surface parts, no strong bending angles of vessels and the possibility to define a more or less planar cutting plane through the whole vessel reconstruction. Therefore, in some cases it is necessary to deviate from the patient-specific vessel representation and to perform local adaptations. These adaptations need to be performed carefully to ensure that the results are plausible from a medical point of view. Otherwise, the results of phantom tests and measurements might be useless. This delicate task needs to be performed with the help of an expert in an iterative process balancing between anatomic correctness and producibility of the physical model. This complex manufacturing procedure depends on the specific RP technology.

An extensive overview of existing RP technologies is given by M. Gurr and R. Mülhaupt [2012]. There exist two basic categories of RP methods [Pham and Gault, 1998]. Models can be built by gradually adding material (e.g., layer-by-layer) [Gibson et al., 2010] or they can be formed by removing material from a larger block. A further subcategorization can be conducted according to the employed materials and specific building procedures. Among others, stereolithography is used in medicine [Petzold et al., 1999], since the material can be cut or drilled with usual surgical instruments. Knox et al. [2005] presented several examples of RP applied to pathologic vascular structures for such purposes. Available literature, however, focuses more on the fabrication process [Lermusiaux et al., 2001; Armillotta et al., 2007] than on the surface model generation. Requirements for generating suitable surface models are rarely specified. Bibb and Winder [2010] made suggestions on how to achieve high-quality RP models w.r.t. the image acquisition procedure. However, the subsequent model generation step does also have strong influence on the quality of RP models.

Since 2006, the development of 3D printing technology yields an increasing number of low-budget printers, e.g., based on the REPRAP project[9], such that RP becomes affordable for smaller companies, research groups, and even for home usage. These

[8] `http://www.aligntech.com`

[9] REPRAP stands for **Rep**licating **Rap**id-prototyper and comprises a certain 3D printer architecture which allows to replicate itself by printing essential components (e.g., mounts, gears); http://www.reprap.org/



(a)

(b)

(c)

Figure 2.7: *A model of the lower jaw is printed via a* RepRap MendelMax *printer. (a) To be able to print overhanging/floating parts, additional support material (striped areas) is printed around the desired structure. The inner parts of the model are filled with a honeycomb pattern to reduce printing times and to gain stability. (b) The printing result with support material. (c) The lower jaw model after manual removal of the supports.*

printers utilize the additive manufacturing technology *fused filament fabrication* (FFF, also: *fused deposition modeling* – FDM). It comprises the extrusion of a plastic filament through a heated, moving nozzle (see Fig. 2.7). The increasing availability of such low budget printers increases the need for convenient 3D modeling functionality and tools, which allow to transform different kinds of data into surface models or for editing and adjusting available models for this printing procedure.

Besides the quality requirements, the tools and workflows used for generating the virtual models vary strongly. Most often, software tools from the computer-aided design (CAD) area are employed, where considerable experiences with such software and its integrated methods are required [Gibson et al., 2010]. This should, however, be feasible without too much knowledge of CAD or computer graphics, e.g., by clinicians. In this context, further issues, such as speed and costs, are also relevant [Gibson et al., 2006]. The reconstruction speed is, of course, mainly influenced by the time consumed for building the RP model. However, the surface model generation part may also be very tedious and time-consuming, since surface models need to be modified, e.g., for artifact removal. This effort is described by some examples in Chapter 4.

2.4 Summary

Surface models are widely used in the medical domain. Typical applications comprise intervention and treatment planning, the simulation of dynamic behavior, or rapid prototyping. The specific requirements to the employed surface models may vary to a certain degree, but can be summarized as shown in Table 2.1.

Requirement	Intervention and	Surgery	CFD	RP
	treatment planning	simulation		
Small features	+	-	+	+
Spatial relations	+	○	-	-
Smoothness	+	+	+	+
High resolution	○	○	+	○
High triangle quality	-	+	+	-
Triangle size homogeneity	-	+	+	-

Table 2.1: *The table indicates the relevance of the specific requirements to surface models. The symbols +,○,- represent high, medium, and low relevance of the specific requirement.*

For *intervention and treatment planning*, a surface model should depict the original structure as precisely as possible. Relevant features must not disappear or be reduced during model generation. The spatial extents have to be preserved to allow for a meaningful interpretation of the spatial situation. For the visual interpretation of the rendered scene, the surface models should resemble the real surface appearance without sharp edges, as they may often be introduced during image acquisition and model generation (see Chap. 3). The mesh resolution should be chosen to be able to depict all relevant surface features. Triangle quality and homogeneity play a minor role in this setting.

Simulation applications may demand very different requirements. *Surgery simulation* is a real-time application, such that visualization, simulation computations, and interaction happen at the same time. To achieve the required performance, the resolution of surface models is often reduced at the expense of surface details and accuracy. However, a natural, smooth look is still required. Since the surface model serves as input for volume mesh generation, the mesh quality is supposed to be high and the triangle size to be changing only gradually over the surface.

In contrast, *CFD* requires high-resolution surface meshes, which should contain all relevant features. At the same time, smoothness, high triangle quality and a homogeneous triangle size distribution are essential. The demand for a preservation of spatial relations to surrounding structures does not fully apply here, since CFD focuses on the dynamic behavior within a single vessel. However, the morphology of the vessel with its furcations and bending parts should be represented accurately. These and further CFD-specific requirements to the model generation pipeline are discussed in more detail in Chapter 4.

For *rapid prototyping*, the surface models are required to look naturally. Thus, they exhibit a smooth, artifact-free surface with all relevant details. The mesh resolution should be adapted to the required level of detail. Since RP usually comprises the production of a physical model for a specific structure, the preservation of distances to surrounding structures does not apply here. Furthermore, triangle quality and homogeneity are not relevant. Depending on the specific rapid prototyping procedure, further requirements which account for the building procedure may have to be considered (see Chap. 4).

The sections in this chapter described example applications for a usage of anatomic surface models. The derived requirements represent a summary of the main aspects of each application. They may vary for other related application fields, but should be generally applicable for generating anatomic surface models.

3

From Anatomy to Smooth Surface Models

Table of Contents

As discussed in Chapter 2, detailed knowledge on the individual patient anatomy but also on its functional properties are required for a variety of medical applications. Anatomic and functional information are acquired by measuring the respective body region using different imaging techniques (e.g. X-ray imaging, ultrasound, computed tomography, magnetic resonance imaging). From the variety of imaging techniques, especially computed tomography (CT) and magnetic resonance imaging (MRI) are capable of measuring a 3D volume. The signals measured within a defined volume are transformed into intensity values using complex algorithms finally composed as 3D image data. This 3D image data may be considered as a stack of 2D images representing "virtual cuts" through the patient, whereas each slice consists of small volume elements (voxels). Each of these voxels represents a small portion of the measured volume and describes a specific, measured property with a computationally obtained intensity value.

Unfortunately, this transformation of anatomic or functional properties into such discretized representations yields several inaccuracies, which affect the interpretation of the image data or 3D models extracted from these data. The generation of 3D models may often not be accomplished automatically and requires user interaction and expert knowledge to achieve a faithful virtual representation of the specific structures. The following sections will discuss these steps and provide background information for understanding the origin of artifacts in 3D surface models. After discussing the most relevant imaging techniques – CT and MRI – with respect to resulting artifacts, the processing of the image data, surface extraction, and the processing of surface meshes are investigated. Special attention is given to the reduction of staircase artifacts.

There exist different representations for approximating 3D surfaces. Triangles represent the simplest polygons which may be used to approximate a surface. Up to now, triangular meshes are most frequently used in the field of medical visualization and its related tasks. As an example, triangles form the basis for tetrahedral mesh generation as required in many simulation applications [Rassineux, 1998]. Due to their relevance in medical visualization, the presented thesis focuses on triangle meshes. During the last years, however, quadrangle (quad) meshes gained more importance. An overview of the benefits of quad meshes is, e.g., given by Bommes et al. [2012] (see also Sec. 3.3.3).

3.1 Image Acquisition

Depending on the specific application and purpose of image acquisition, a variety of techniques is available [Preim and Bartz, 2007, Chap. 3]. From this variety, especially CT and MRI are employed to obtain image data for applications such as surgical planning and simulation, CFD, or rapid prototyping. The following sections outline the basic principles of these imaging techniques.

3.1.1 Computed Tomography

CT employs the principle of X-ray imaging, where an object is radiographed and the ray attenuation is measured behind the object with a radiographic film. Different structures and tissues lead to a different attenuation of the X-rays. As a consequence, this projection of the measurement volume (the measured body region) to the 2D radiographic film yields different brightness values, according to which structures/tissue have been radiographed. This provides a fast overview of the measured region, but lacks any 3D information, since all structures being hit by the same ray attenuate that one and contribute to the same final intensity value.

In contrast to standard X-ray imaging, CT samples the target object with X-rays from different angles by rotating the X-ray source around it and measuring the attenuated ray with a detector on the opposite side. Hence, the radiation dose[1] of CT is clearly higher than standard X-ray imaging, but is justified by the diagnostic benefit of CT imaging.

For each rotation angle, a 1D attenuation profile is created. These profiles can be transformed into volumetric image data via filtered backprojection. Since the specific measurement setup may vary, several reconstruction algorithms can be employed. Herman [2009] provides more insight to this topic.

The slice orientation depends on the rotation axis of the X-ray source and detector. This rotation axis is usually aligned with the patient bed and may thus not be modified. During image acquisition, the patient is required to remain still to prevent movement artifacts, which would distort the ray attenuation, since different tissue portions would move through the same ray(s). The resulting intensity values would thus represent a mixture of the attenuation values of the involved tissues. Such movement artifacts may appear, e.g., from breathing motion. However, short acquisition times, especially with multi-slice CT, reduce the risk of such movement artifacts.

Barrett and Keat [2004] identified four categories of CT data artifacts according to their origins. They are caused by:

- The physics of the acquisition process. Examples are beam hardening, partial volume, and photon starvation artifacts.
  - **Beam hardening** means that the lower energy parts of an X-ray beam are absorbed more rapidly than its higher energy parts. As a consequence, the average energy of the beam increases, which leads to a reduced ray attenuation when passing through an object. For the final image data, this yields inhomogeneous intensity values within the same measured material and streaks.
  - **Partial volume artifacts** occur if a structure is not captured by all beams during the rotation of the ray source/detector or if a structure extends only partially into the measured slice. The resulting intensity values are composed as the average of the materials within the measured voxel. Such artifacts are most likely to appear in body regions where different tissues are located very close along the slice direction. Thus, a higher slice thickness

[1] The radiation dose is given in the unit GRAY (Gy). One GRAY represents the energy absorbed per kilogram of matter.

fosters partial volume artifacts. In the image data, this becomes obvious as shading artifacts and blurring of structure boundaries.

  - **Photon starvation** means that an X-ray is strongly attenuated, such that not enough photons reach the detectors. This results in severe image noise and streaking artifacts.

- Patient-based factors. This comprises all artifacts resulting from improper materials in/on the patient or movements of the patient.

  - **Movement artifacts** are not only caused by conscious movements of the patient, but also by unconscious movements inside the human body. Examples include the motion of anatomic structures due to respiration or pulsation. Some of these artifacts can be reduced by, e.g., breath holding or positioning aids. Also, short acquisition times reduce the effects of motion artifacts, which become visible as streaks or shading artifacts in the image data.

  - Due to the very high material density, **metal objects** attenuate the X-rays much stronger than all human tissue. The result are strong star-shaped, streaking artifacts in the image data (see Fig. 3.1a).

- Scanner-based factors, which summarize all those artifacts resulting from **erroneous calibrations** of X-ray sources and detectors. This is the case, e.g., if the center of rotation of the ray source is not equal to the detector's center of rotation. Such artifacts are often represented as circular stripes in the image data.

- Image reconstruction: Besides measuring a 3D volume slice by slice, there exist further acquisition techniques which proceed, e.g., in a helical manner along the z-axis. Such special acquisition techniques require other image reconstruction and also interpolation methods compared to the standard CT procedure. However, the helical measurement path and required interpolation during reconstruction may lead to severe **distortions** and windmill-like stripes.

The appearance of these artifacts does strongly depend on the specific measurement technique and procedure. Different hard- and/or software solutions for suppressing these artifacts are usually provided by the specific device manufacturers. Artifacts, such as inhomogeneities (shading), stripes (see Fig. 3.1) or noise are, however, still visible in many CT datasets. These effects complicate the 3D model generation, as they may yield frayed parts, staircases, wrong shapes and spatial relations.

Besides only measuring patient anatomy, CT offers more acquisition modi, i.e., CT angiography and CT fluoroscopy. To improve the soft tissue contrast, contrast agents may be applied to the patient. These fluids modify the attenuation of X-rays and may thus enhance certain structures, e.g., vessels (angiography). Furthermore, contrast agent may accumulate in soft tissue, allowing to observe how different substances are transported to specific structures and if they are handled correctly (e.g., hypo-/hyperthyroidism). On the one hand, contrast agent enhances the visual contrast in the acquired CT data and eases the segmentation and surface extraction from these data. Unfortunately, the good contrast does also mean that surface reconstructions may be incorrect, e.g., if the contrast agent did not spread uniformly in the target vasculature. Thus, reconstructed vessels may appear narrowed or even separated (see also Chap. 4).

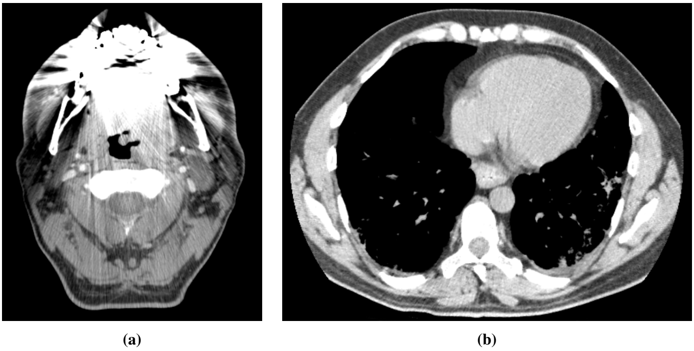


Figure 3.1: *Examples of streaking artifacts in CT data. (a) Neck dataset with dental metal implants. (b) Thorax dataset with slight streaking artifact in the heart region, possibly due to motion.*

3.1.2 Magnetic Resonance Imaging

In contrast to CT, MRI allows for high soft tissue contrast without any ionizing radiation, which makes it especially useful for measuring soft tissue. MRI utilizes the nuclear spin of hydrogen atoms, which comprises the effect that each atom precesses around a rotation axis. These spins yield magnetic moments. Thus, by applying a constant magnetic field, called B_0 field, to the complete measurement volume, the rotation axes of all hydrogen atoms inside this volume are oriented parallel or antiparallel to B_0. Moreover, the field strength affects the precession frequency ($\equiv$ Larmor[2] frequency). By applying a radio frequency pulse, all those spins are disturbed to lose their orientation along B_0, whose Larmor frequency matches the frequency of the excitation pulse. Each time an RF pulse is applied, the spins try to recover to their orientation along B_0, which generates a measurable echo. Additional magnetic fields with spatially varying field strengths (“gradient fields”) modify the Larmor frequency along the gradient directions, which is used for slice selection and position encoding within a slice. However, there exists a large variety of MR sequences that make different usage of RF pulses and gradients.

The opportunity to combine RF pulses and gradients in a different manner makes MRI a powerful tool to measure anatomy and functional properties. As an example, the contrast of vascular structures may be enhanced in a non-invasive way. Time-of-flight imaging (ToF-MRA) or phase-contrast MR angiography (PC-MRA) are used to emphasize flowing blood and to encode the flow velocity. Furthermore, even information about brain activity can be obtained by utilizing the magnetization changes of oxygenated and deoxygenated blood.

Roughly summarized, the image data is finally obtained by measuring slice-wise and transforming the measured signals via a 2-dimensional Fourier Transform into a 2D

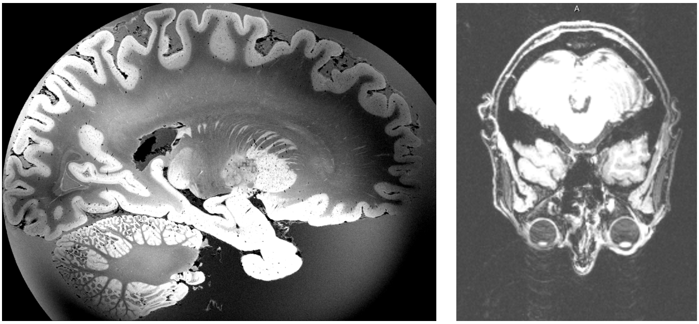
[2] Named after the physicist and mathematician Sir Joseph Larmor.

image. Similar as for the measurement, different strategies exist for faster image reconstruction and artifact suppression. Compared to CT, MRI is more sensitive to artifacts, since very small effects (spin echoes) are measured. This requires an effective shielding against external electromagnetic fields (e.g., caused by electricity) to guarantee homogeneous magnetic fields for the measurement. Moreover, any kind of ferromagnetic material in and near the MRI scanner affects the homogeneity of the employed gradient fields.

Similar to the classification of Barrett and Keat [2004] for CT artifacts, Erasmus et al. [2004] categorized the artifacts in MRI into three classes:

- Patient-related artifacts comprise again all those effects which are caused by improper materials in the patient as well as motion of the body and anatomic structures. As an example, blood flow causes the spins to change their position during the measurement. Thus, magnetized spins leave the measurement volume (e.g., a slice) and non-magnetized spins or spins in another excitation state enter the volume. Such motion and flow artifacts may yield intensity inhomogeneities and ghosting (see Fig. 3.2). Further artifacts originate from metal objects (e.g., surgical clips, orthopaedic devices) in and near the measurement volume. Metal affects the Larmor frequency of the nearby spins, which leads to signal loss, image distortion, or ghosting.
- Signal processing-dependent artifacts comprise all effects resulting from the measurement procedure and sequences. This includes partial volume effects as well as wrap-around artifacts, which means a displaced mapping of anatomic regions to the opposite side of an image slice. Further artifacts in this category are chemical shift and ringing artifacts.
- Machine/hardware-related artifacts originate from the measurement equipment and related technical defects. As an example, inhomogeneities of the B_0 field or of the gradient fields lead to distortions of the mapped anatomy and the resulting intensity values. Likewise, intensity values change depending on the distance to the employed coil.

In summary, MRI can be configured to obtain high soft tissue contrast, high resolution, and image data slices with an arbitrary orientation. Unfortunately, measurements may easily take from a few minutes up to half an hour and more. Movement artifacts, e.g., from breathing or heart beat, have to be accepted or treated. Intensity inhomogeneities and noise are typically contained in MR images and complicate the extraction of surface models. In addition, several other artifacts (e.g., wrap-arounds) may often lead to false structures in the resulting surface models.



(a) (b)

Figure 3.2: *Examples of artifacts in MRI data: (a) strong intensity inhomogeneities and (b) ghosting artifact (bottom part, below the eyes).*

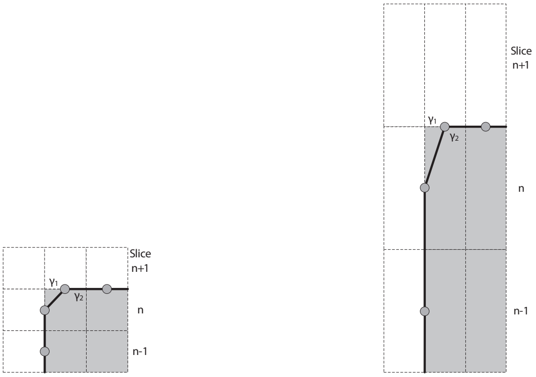
3.2 Surface Extraction

In the ideal case, the boundary of a target structure can be defined by a specific intensity value in the image data. This is theoretically feasible for CT data, where the intensity values are given in the Hounsfield scale[3]. For MRI data, the intensity values depend on several factors, which make an assumption on concrete intensity values impossible. Moreover, due to the different kinds of image data artifacts (see Sec. 3.1.1 and Sec. 3.1.2), a direct transformation of the image data into a naturally looking, accurate surface model is usually not possible. A segmentation of the target structure is required to clearly delineate it from surrounding structures and tissue. Otherwise, artifacts, such as image noise and local and global intensity inhomogeneities as a synonym for the effects of the earlier described artifacts (see Sec. 3.1), lead to additional structures being included by mistake or yield frayed, separated and missing parts. To account for these artifacts, segmentation methods are applied, but often result in staircases in the resulting surface models. Such staircase artifacts can be tackled at different levels of the model generation pipeline. They can be prevented/reduced:

1. at the image level,
2. during surface extraction,
3. and as surface postprocessing step (mesh smoothing)

The following subsections discuss surface extraction under special consideration of such artifacts. Afterwards, staircase reduction at the image level will be outlined. Due to its relevance for the presented thesis, mesh processing – and especially mesh smoothing – will be discussed in a separate section (see Sec. 3.3.4).

[3] The units of intensity values resulting from CT acquisition and filtered backprojection are referred to as Hounsfield Units (HU), named after Godfrey N. Hounsfield. The values range from -1024 to 3071 HU, whereas the value of water is 0 HU, air is -1000 HU, and bone structures have typically values of, e.g., >500 HU.



(**a**) Isotropic Grid

(**b**) Anisotropic Grid

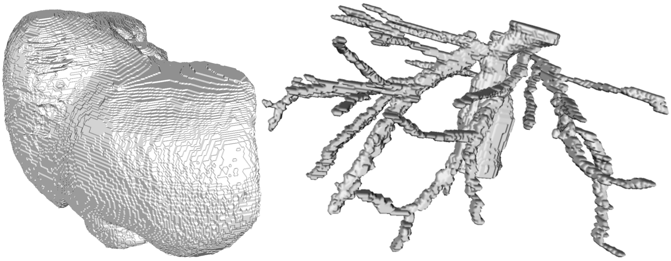
Figure 3.3: *Comparison of surfaces extracted from binary data (white rectangles = background; gray rectangles = object). Extraction from an anisotropic grid (b) yields sharper features than from an isotropic grid (a). For the grid in (b), the voxel height (y-direction in the illustration) has been tripled compared to (a).*

3.2.1 Binary Data

Segmenting tomographic image data typically yields a binary decision if a voxel belongs to the target structure (e.g., described by a certain intensity value) or not. Binary segmentation removes all intensity information from the image data and reduces the structure to a classification into background and object voxels. Such a binary volume can be transformed into a surface mesh via various methods, e.g., Marching Cubes (MC) algorithm [Lorensen and Cline, 1987], Constrained Elastic Surface Nets (CESN) [Gibson, 1998], level-set methods [Whitaker, 2000], or MC variants treating, e.g., multiple segmentation labels in the same volume.

The most commonly used mesh extraction algorithm is the Marching Cubes. It traverses volumetric image data by examining volume cells of eight neighboring voxels (2×2 voxels in each two adjacent image slices, see Fig. 3.3). Each cell is compared to a look-up table with 15 different cases [Lorensen and Cline, 1987]. By doing so, vertices are generated and connected to triangles slice by slice. The position of each new vertex is determined according to the intensity values of the surrounding voxels. Thus, for binary data, a new vertex is always located on the center between each two adjacent voxels. For an isotropic volume grid, where all voxels have the same dimensions in x-, y-, and z-direction, the interpolation of MC yields 135° dihedral angles (see Fig. 3.3a, label γ_2). An increase of the slice distance, as it is common in tomographic medical image data, results in a modification of the dihedral angles towards 90° (see Fig. 3.3b, label γ_2).

The original MC has, however, some limitations, e.g., regarding the handling of non-overlapping contours in adjacent slices, and suffers from ambiguities. Ambiguities re-



(a) Liver **(b)** Vascular Tree

Figure 3.4: *Examples of surface models generated from binary segmented data via MC.*

sult from the cell-based decision of what is considered inside/outside. For two adjacent cells, this may lead to holes in the surface [Dürst, 1988]. To resolve this, Nielson and Hamann [1991] introduced the asymptotic decider, which performs the inside/outside decision based on a point interpolated at the shared face of the adjacent cells. Nowadays, the commonly used MC variants (e.g., as implemented in VTK [Schroeder et al., 2004]) employ a 255-case table to avoid inconsistencies. Further modifications of the MC consider specific cases, where, e.g., two or more structures are very close such that a considered cell contributes to different surfaces. Thus, multiple materials (segmentation labels) occur within the cell resulting in an extension of the MC case table. Besides introducing additional cases, the topology of the contributing surfaces has to be analyzed carefully to achieve plausible surface meshes [Hege et al., 1997; Wu and Sullivan, 2003].

As visible in Figures 3.4 and 3.5a, surface extraction from binary segmented data leads to staircase artifacts all over the surface. The CESN method of Gibson [1998] tries to prevent such artifacts by generating an initial mesh with vertices at the center of each (boundary) volume cell[4] and subsequently translating the vertices within the bounds of their specific cell. The adjustment of the vertex positions represents an additional smoothing step (see Sec. 3.3.4), whereas the movement restriction prevents the surface model from too strong volume shrinkage. CESN guarantees a certain smoothness reducing staircase artifacts with concurrent feature preservation. Larger staircases, however, may still remain to the model, and thin structures, such as smaller blood vessels (diameter $<=$ voxel size) tend to collapse to a line.

The Dual MC algorithm of Nielson [2004] proceeds similarly to the CESN approach. From the binary image data, an initial surface is generated. This is, however, comparable to the dual surface of the initial CESN surface. Artifacts are then reduced by iteratively generating the dual of the current surface. Again, this iterative “smoothing” yields volume shrinkage, which is counteracted by restricting the placement of the vertices to the original cells.

[4] A boundary volume cell means those cells which contribute to the surface, as it consists of background and object voxels.

Whitaker [2000] applied the concept of CESN directly to the binary image data without generating an intermediate surface. Thus, the method operates on the level sets of the volume, which are treated as deformable surfaces. The goal is then to find an optimal level set which minimizes the surface area, includes all voxels belonging to the objects, and excludes all background voxels.

Depending on the model generation algorithm, surface extraction from binary segmented data may introduce additional artifacts (e.g., staircases, terraces, frayed parts or holes) to the final model. Several mesh generation methods do already take care of these artifacts by performing additional correction steps but can only be successful to a certain extent.

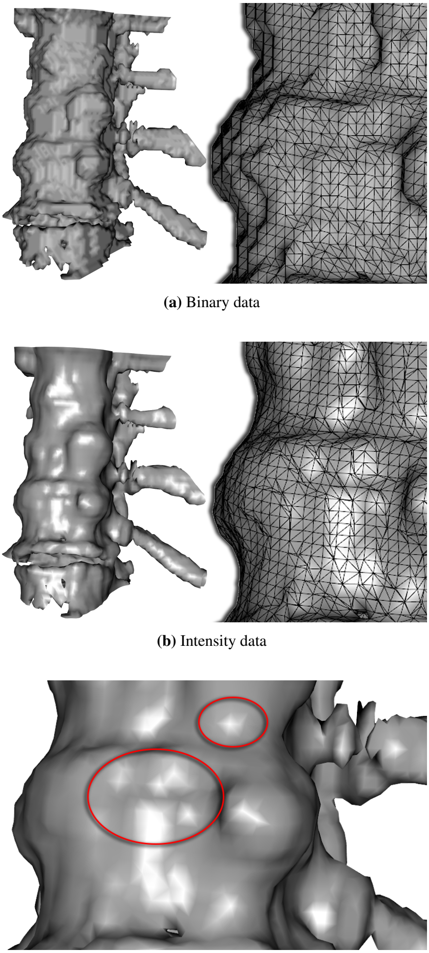
3.2.2 Intensity Data

If possible, the intensity values of tomographic medical data should be used for surface extraction. As described in Section 3.2.1, the MC algorithm proceeds through each volume cell and determines the vertex positions based on the values of the considered voxels. In contrast to the binary case, the intensity values allow for a more precise interpolation of the vertex locations such that the surface exhibits a certain smoothness by default and staircase artifacts are prevented (compare Fig. 3.5). As described in Section 3.1, most image data suffer from intensity inhomogeneities (and artifacts with similar effects) which cause surface extraction algorithms to generate malformed surface models.

Looking at MC-based surfaces in more detail (see Fig. 3.5c) reveals that the surface is often not perfectly smooth and shows some “diamond” artifacts. These artifacts result from the linear interpolation of MC and become most often visible when the camera gets close to the surface. Besides applying better rendering and shading techniques, other MC variants exist, which improve the quality of surface models. As an example, the Precise MC [Allamandri et al., 1998; Cignoni et al., 2000] refines an initial MC mesh adaptively by estimating the local error and generating new vertices by means of trilinear interpolation. The drawback of such methods is a high computational effort for surface generation but also rendering, since the resulting surface contains more elements than the initial MC surface.

Bruin et al. [2000] modified the original CESN approach (see Sec. 3.2.1) to involve intensity information during the relaxation procedure to allow for higher accuracy. They employ the gradients of the intensity data to translate vertices closer to the isosurface of the target structure. Another way to improve accuracy is to detect relevant features in the data and adjust the surface extraction accordingly. As an example, the Extended MC [Kobbelt et al., 2001] accounts for aliasing artifacts degrading feature edges in the surface. As a result, features are better reproduced during model generation, but both relevant geometric details and staircase artifacts would be maintained.

These and other surface extraction methods can only be successful if the data itself does not contain too many or too strong artifacts, such as inhomogeneities which lead to severe reconstruction problems. Figure 3.6 shows such an example, where the carotid artery could not be separated from surrounding tissue. Involving additional information (e.g., on the expected shape) can help to remedy such identification problems. A



(**a**) Binary data

(**b**) Intensity data

(**c**) Close-up view

Figure 3.5: *Isosurfaces of a part of the spine generated from binary data (a) and intensity data (b) via MC (right: close-up view of the wireframe). (c) shows a close-up view of the model exhibiting diamond artifacts (encircled regions).*

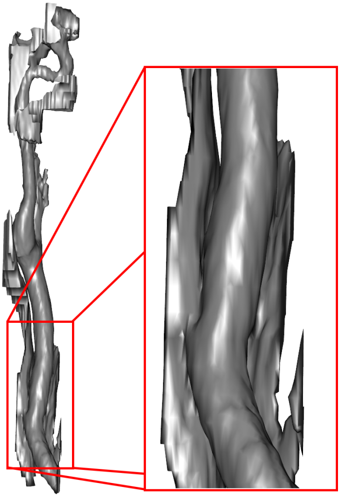
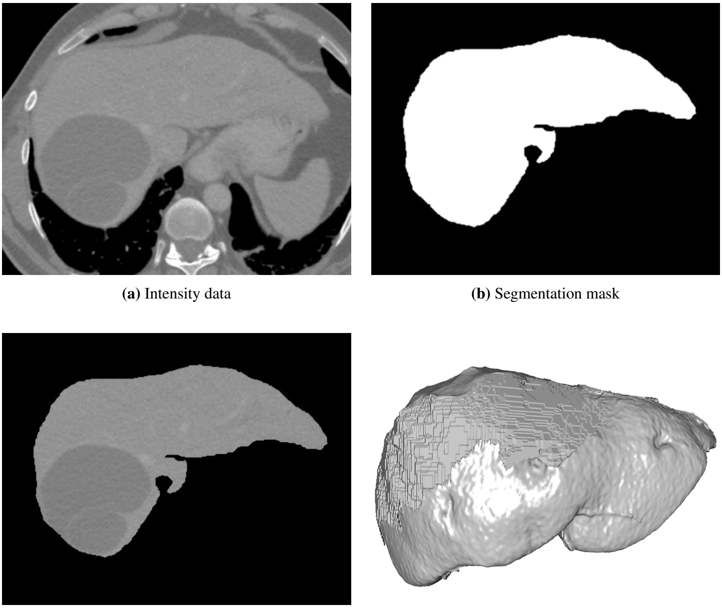


Figure 3.6: *Model of a carotid artery with additional surface parts due to inhomogeneities in the intensity data.*

promising example for direct mesh generation from intensity data are 3D stable mass-spring models [Dornheim et al., 2006, 2007, 2010], since the model initially represents an average shape of the target structure without voxelization-related artifacts and subsequently adapts to the real data. Important (small) features can still be detected and extracted as long as they are included in the model. Unfortunately, due to their aberrant shape, pathologic structures can typically not be reconstructed. Especially for vascular structures, surface extraction often requires dedicated algorithms (see Sec. 4.1 for a detailed discussion). The usage of 3D primitives, such as cylinders and truncated cones, to approximate the shape of a vessel would prevent staircases. However, model-based approaches cannot reproduce arbitrary shapes, e.g., of aberrant anatomic structures. Model-free approaches (e.g., [Grimm and Hughes, 1995; Ohtake et al., 2003; Braude et al., 2007; Jalba and Roerdink, 2006, 2009]) yield smooth surfaces, but often require to set and tune several input parameters. The usage of MPU implicits for surface generation from point cloud data (e.g., acquired via laser scanning) is considered as fast and robust. However, the intensity information of tomographic medical images is ignored.

Another simple solution to the inhomogeneity problem is to include the results of an intermediate segmentation step (eventually performed by medical experts). The (binary) segmentation result describes the location and extent of the target structure in the image data. A careful dilation of the segmentation mask (e.g., with a kernel of 3×3×3 voxels) allows to slightly increase the segmented volume. This contour information can then be used to mask the intensity data leaving only the target structure. Afterwards, surface



(a) Intensity data

(b) Segmentation mask

(c) Masked intensity data

(d) Resulting surface model

Figure 3.7: *Masking procedure for CT data of the liver: (a) the initial CT data, (b) the segmentation mask, (c) the masked intensity data, and (d) the resulting surface model.*

extraction can be performed on the remaining intensity data. Areas formerly suffering from inhomogeneities are now clipped by the dilated segmentation mask, whereas areas matching the desired intensity range (specified by isovalues) are properly extracted. Unfortunately, the inhomogeneous regions may still lead to staircase artifacts – but locally restricted [Mönch et al., 2010b, 2011a] (see Fig. 3.7).

3.2.3 Artifact Reduction at the Image Level

The above-mentioned method of applying segmentation masks to intensity data allows to deal with intensity inhomogeneities to a certain extent but may introduce local staircase artifacts. These are additionally emphasized by anisotropic voxel dimensions. A reduction of the slice distance can be achieved via resampling of the image data to yield almost isotropic voxel sizes, e.g., via shape-based interpolation [Raya and Udupa, 1990]. The quality of the resulting surface model regarding accuracy is especially affected by the employed interpolation method (e.g., linear, hermite spline, or cubic b-spline) [Mönch et al., 2010c]. Even though larger staircase artifacts can be prevented this way, they cannot be prevented generally. Furthermore, by adding intermediate slices, much more data is generated resulting in further computational effort during surface extraction and any subsequent mesh application. Especially in clinical routine, this additional effort is often prohibitive, since data with lower resolution is usually acquired deliberately to save time and storage.

Smoothing operators (e.g., Gaussian smoothing, diffusion filters) can be used to reduce noise within structures and at their contours. Smoothing of binary segmented data may reduce staircase artifacts, since the interpolation during surface extraction benefits from the blurred contours. However, there is no control over the accuracy for uniform smoothing. For intensity data, anisotropic smoothing (e.g., anisotropic diffusion) is usually performed to eliminate noise and to concurrently preserve structure borders by performing smoothing only perpendicular to the intensity gradient. Smoothing may be used to reduce noise in particular, but cannot account for intensity inhomogeneities or larger staircase artifacts. The latter could only be achieved by extensive smoothing at the expense of model accuracy.

During segmentation, intensity inhomogeneities and noise become especially obvious, since they lead to frayed parts and small holes in the binary segmented data and in the resulting surface models. Besides smoothing, morphological operators (erosion/dilation and their combinations opening/closing) with differently sized kernels are an option for further treatment of smaller frayed parts or holes. This has, however, only marginal effects on staircase artifacts but might essentially alter the shape of the considered structures – in particular for smaller/thin structures, such as vessels. As an example, Neubauer et al. [2004] combined a sequence of erosion and dilation operations to reduce the effects of binary segmented data. To preserve features, the algorithm checks if the shape of the target structure has been altered too much. Morphological operations have also been employed by Lakare and Kaufman [2003] to remove aliasing artifacts from binary segmented data. In their workflow, they apply erosion and dilation operations to obtain fuzzy object boundaries, which finally allows for a visually improved direct volume rendering.

3.3 Postprocessing of Surface Models

As summarized in Chapter 2, most applications have very specific requirements to surface models. Besides accuracy in terms of a correct, distortion-free representation of the real anatomic shape, the mathematical models and algorithms applied by the specific application may have further requirements to surface models. The following sections will discuss the most common mesh processing tasks to give a brief insight to this field. A more detailed view on mesh processing is given by Botsch et al. [2010].

3.3.1 Decimation and Subdivision

The resolution of a surface mesh is a major criterion for several applications. A high-resolution is usually required for faithfully depicting fine surface details. Vice versa, surface regions without relevant features (e.g., flat areas) can easily be composed of only a few triangles. Meshes produced, e.g., via the MC algorithm, often consist of too many triangles, since the algorithm proceeds through the volume cells and generates triangles for each single cell without considering the surrounding. Thus, the mesh resolution is almost constant over the surface irrespective of the required level of detail. Especially surfaces obtained from high resolution image data contain a large number of vertices and triangles. For most current rendering applications, this is not a serious problem with respect to performance. Environments with multiple rendered structures (e.g., in surgical simulation) or web-based rendering setups may, however, still be pushed to their limits and may require low resolution meshes.

The process of reducing the mesh resolution is usually referred to as decimation or simplification. The typical procedure behind mesh decimation consists of an evaluation and a triangulation step. At first, vertices are classified according to an error term to identify those vertices which are less relevant to express the surface characteristics. These vertices are then removed and the resulting gap is locally remeshed [Schroeder et al., 1992]. Garland and Heckbert [1997] introduced mesh decimation based on quadric error metrics. Using the quadric error distance measure for vertex evaluation, edges are collapsed to a new point which carries the error metric of the collapsed vertices. Hoppe [1996] described the *Progressive meshes* method, which generates a series of successively decimated meshes. The underlying data structure is organized such that one can easily switch between all meshes in the series by performing edge split or collapse operations.

In contrast to mesh simplification, applications may also demand for higher resolution, e.g., as initial surface mesh for volume mesh generation in CFD or to obtain a smoother look via a higher level of detail. Subdivision algorithms traverse a surface mesh and proceed recursively with subdividing surface elements and generating new vertices according to different criteria. Typical subdivision schemes have already been described in 1978 [Catmull and Clark; Doo and Sabin] and 1987 [Loop]. An overview of simplification algorithms is given by Garland [1999] and Luebke [2001].

3.3.2 Remeshing

Remeshing comprehends a modification or re-organization of the mesh structure. Alliez et al. [2008] identified five categories of remeshing techniques according to their specific goals (structured, compatible, high quality, feature, and error-driven remeshing). Within the scope of this thesis, remeshing is considered as a mesh processing step to prepare surface models for, e.g., CFD or tissue simulation. This includes the optimization of the triangle quality, mesh homogeneity (w.r.t. triangle shape and size), but also locally adaptive sampling according to local shape characteristics (see Fig. 3.8).

A surface mesh has a high quality, if most of its triangles are equilateral. In contrast, deformed/stretched triangles have a low quality. In the context of simulation applications, it is usually required that a certain minimum quality is guaranteed. Amongst others, the

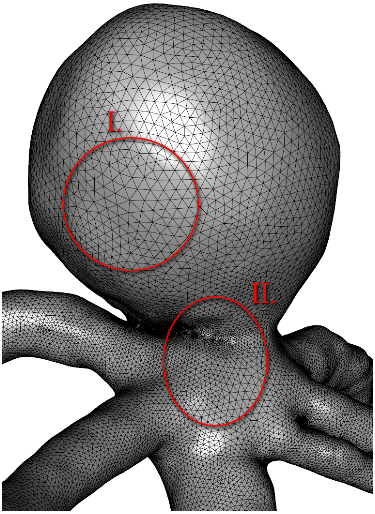


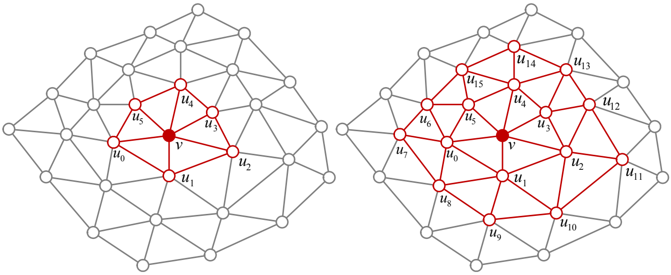
Figure 3.8: *Surface model of a cerebral aneurysm superimposed with the wireframe. The encircled regions I. and II. highlight the adaptive sampling which is required for depicting the specific regions.*

most common triangle quality measures are the equiangle skewness (EAS) and the ratio between the incircle and the outcircle of a triangle [Pébay and Baker, 1991; Knupp, 2003; Preim and Botha, 2013]. Slight variations of the triangle shape and size within a mesh are not critical as long as abrupt changes are avoided.

3.3.3 Volume Mesh Generation

Simulation applications (e.g., simulation of blood flow, tissue behavior, or heat distribution) require volume/tissue characteristics (e.g., elasticity or heat capacity) as input. This input is usually a volume mesh, whose grid structure can be be generated from a surface mesh. The relevant properties, however, have to be assigned based on assumptions or (tissue) measurements.

A volume mesh consists of volume elements, e.g., tetrahedra or prisms, which represent a local description of relevant tissue characteristics. As an example, if one would virtually bear pressure onto one volume element, this element would be locally compressed and might pass some pressure to adjacent cells. In the same manner, heat transfer through a volume might be described. Besides a faithful description of the specific tissue properties, the size and shape play an important role for the accuracy and convergence of such simulation computations. A higher volume mesh resolution and a gradual



(a) **(b)**

Figure 3.9: *Illustration of the 1-ring and 2-ring neighborhood. The 1-ring includes all vertices which are directly connected to the considered vertex. The 2-ring adds the direct neighbors of the 1-ring vertices.*

change of element sizes allows for a more precise simulation. As examples, the algorithms of Molino et al. [2003] and Labelle and Shewchuk [2007] generate tetrahedral meshes being suitable for simulation applications. The shape of surface elements does usually affect the properties of the volume mesh. For instance, a thin, needle-like triangle results in a stretched, almost flat tetrahedron (e.g., needle, sliver, or cap [Adler, 2014]). In this context, surface meshes consisting of quadrangles may be more suitable and are preferred in some simulation tasks, since they reduce the error introduced by the resulting hexahedral volume approximation and the number of volume elements (compared to triangles and tetrahedra) [Shepherd and Johnson, 2008; Bommes et al., 2012]. However, even hybrid volume meshes consisting of both tetrahedral and hexahedral volume elements are used [Berg et al., 2013].

3.3.4 Mesh Smoothing

Noise and staircase artifacts in surface models can be reduced by applying mesh smoothing filters (e.g., [Desbrun et al., 1999; Vollmer et al., 1999; Tasdizen and Whitaker, 2003; Chica et al., 2008]). During mesh smoothing, the vertices of a surface mesh are displaced without modifying the connectivity between the vertices. The displacement of each vertex is usually computed depending on the location of its neighboring vertices. Most algorithms consider the direct (1-ring) neighborhood of each vertex (see Fig 3.9). However, even more vertices from a larger surrounding (e.g., 2-ring neighborhood) may be included. Besides the topological neighborhood, other neighborhood definitions, e.g., Euclidean distances, are feasible. An increase of the neighborhood size reduces the sensitivity for small features. If not explicitly stated, smoothing is always applied to the 1-ring within this thesis.

The simplest smoothing method is the *Laplace filter*. It iterates over the vertices of a mesh and determines the new location of each vertex by averaging the locations of its n-ring neighbors (see Eq. 3.1). The exact displacement of each vertex is scaled by the

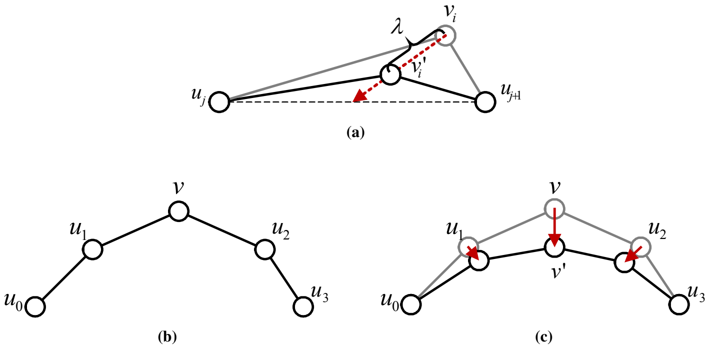


Figure 3.10: *Illustration of Laplacian smoothing. After one smoothing iteration, the vertex v is translated to its new position v'. Since also the vertex neighbors u_j are smoothed (u_0 and u_3 are not moved), the resulting surface tends to shrink.*

weighting factor λ. This is repeated for a given number of iterations or until a defined criterion (e.g., a maximum error is exceeded) becomes true. This basic principle is depicted in Figure 3.10. The Laplace filter requires two input parameters: the number of iterations and the uniform weighting factor λ. The latter weights the displacement of each vertex along the specifically determined displacement vector resulting from the averaged location of the vertex neighbors. A small value for λ yields only small vertex displacements during each iteration.

$$\forall v_i \in V : \quad v_i' = v_i + \frac{\lambda}{m} \sum_{j=1}^{m} (u_j - v_i) \tag{3.1}$$

$v_i, u_j \in V, \forall u_j \in U_{v_i}^1, m = \left|U_{v_i}^n\right|$

$U_{v_i}^n$ - n-ring neighbors of vertex v_i

λ - uniform smoothing factor

Comparing the illustrations in Figures 3.10b and 3.10c, it becomes clear that Laplacian smoothing yields smooth shapes at the expense of strong volume shrinkage for convex surface areas and a loss of features. The vertices illustrated in the figures would finally collapse to one point for a high number of iterations. Especially surface models of elongated, thin structures (e.g., vessels) suffer from a loss of volume. This effect is noticeable during visual inspection of anatomic structures – in particular in endoscopic views of vascular structures (see Fig. 3.11).

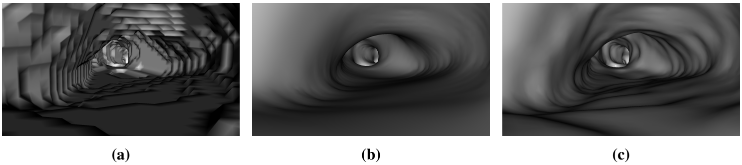


Figure 3.11: *Illustration of Laplacian smoothing for an endoscopic view inside a bronchus. (a) shows the original surface extracted from binary data. (b) The result of uniform Laplacian smoothing and (c) LowPass filtering.*

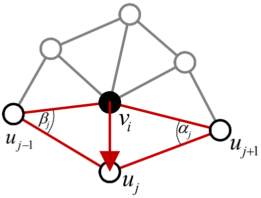


Figure 3.12: *Illustration of a 1-ring with the labels of Equation 3.2. For each of the vertex neighbors u_j in the 1-ring, specific weights are determined from the angles of the corresponding triangles.*

Mean Curvature Flow

Besides volume shrinkage, the uniform weighting of all vertex neighbors leads to tangential drifts of the vertices. Thus, vertices tend to drift into the direction of their furthest neighbors leading to further deformations – especially for irregularly sampled surface meshes [Desbrun et al., 1999]. The *Mean Curvature Flow* filter decouples Laplacian smoothing from this dependency by adaptively weighting the influence of each vertex neighbor. This is achieved by restricting the vertex displacement to the direction of its *Mean Curvature Flow Normal*. Applying this to Equation 3.1 yields:

$$\forall v_i \in V: \quad v_i' = v_i + \frac{\lambda}{\sum_{j=1}^{m}(cot\,\alpha_{ij} + cot\,\beta_{ij})} \sum_{j=1}^{m} (cot\,\alpha_{ij} + cot\,\beta_{ij})(v_i - u_j) \tag{3.2}$$

Suitability for Anatomic Surfaces

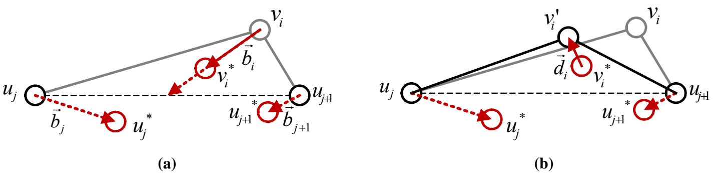
A variety of mesh smoothing filters have been compared by Belyaev and Ohtake [2003]. They evaluated the suitablity of smoothing filters for denoising surface models and preservation of sharp features. The latter, however, interferes with the requirements for anatomic surface models which usually do not contain sharp features. Thus, Bade et al. [2006; 2007] performed a comprehensive analysis on mesh smoothing filters w.r.t. their suitability for anatomic surface models. They compared three different shape categories:

- *Compact objects* have a relatively large volume compared to a small surface area, e.g., liver, milt, kidneys, or lymph nodes.
- *Flat objects* are characterized by a large expansion in two directions, whereas the extension in the third direction is considerably small. Examples are the sternocleidomastoid muscle and some bones (e.g., shoulder blades).
- *Elongated objects* exhibit a strong extension in one direction, e.g., vascular structures, thin bones, or a bronchial tree.

As a result of these investigations, Bade et al. [2006; 2007] identified the LowPass filter [Taubin, 1995] and the Laplace+*HC* filter [Vollmer et al., 1999] as suitable for anatomic structures and presented parameter recommendations for these object shape categories. The above-mentioned *Mean Curvature Flow* filter yielded strong volume shrinkage and is considered as inappropriate for anatomic surface models.

Laplace+*HC* Filter

The idea behind the Laplace+*HC* algorithm of Vollmer et al. [1999] is to apply a Laplacian smoothing step and subsequently push the modified vertices back towards the initial surface (see Fig. 3.13).The first step is performed as standard Laplacian smoothing (see Eq. 3.3). The resulting displacement $\vec{b_i}$ is then modified using the parameter α to weight between considering the original vertex position o_i versus the previous vertex position v_i (see Eq. 3.4). In the second step, the weighting factor β determines to what extent the intermediate displacement of the neighboring vertices is considered (see Eq. 3.5) for the back correction $\vec{d_i}$. Thus, setting α and β both close to 0 yields stronger smoothing, since the influence of the original vertex position as well as the back correction is reduced. This two-step procedure is performed during each smoothing iteration. As a result, volume shrinkage can be reduced, but larger staircase artifacts are usually not completely reduced.



(a) **(b)**

Figure 3.13: *Illustration of Laplace+HC smoothing. During each smoothing cycle, two steps are executed: (a) a first smoothing step and (b) subsequently a correction step.*

$$\forall v_i \in V: \qquad v_i^* = v_i + \tfrac{\lambda}{m} \sum_{j=1}^{m} (u_j - v_i) \tag{3.3}$$

$$\vec{b}_i = v_i^* - (\alpha o_i + (1-\alpha) v_i) \tag{3.4}$$

$$\vec{d}_i = -\beta \vec{b}_i - \tfrac{1-\beta}{m} \sum_{j=1}^{m} \vec{b}_j \tag{3.5}$$

$$v_i' = v_i^* + \vec{d}_i \tag{3.6}$$

LowPass Filter

The principle of the LowPass filter (also: Taubin's $\lambda|\mu$ filter) is similar to the Laplace+*HC* algorithm: perform smoothing and afterwards a back correction step (see Fig. 3.14). For the first step, as previously shown, Laplacian smoothing is performed with the weighting factor λ (see Eq. 3.7) resulting in the intermediate vertex position v_i^*. By considering the intermediate locations of the vertex neighbors u_j^* and the weighting factor μ, v_i^* is translated to its final position v_i' (see Eq. 3.8). The factor μ is selected to be negative, such that $\|\mu\| = \lambda + \varepsilon$ (e.g., $\mu = -1.02\lambda$). These two steps are again subsequently applied during each smoothing iteration. As a result, the volume is generally preserved, whereas a certain smoothness is achieved [Bade et al., 2006; Bade, 2007].

$$\forall v_i \in V: \qquad v_i^* = v_i + \tfrac{\lambda}{m} \sum_{j=1}^{m} (u_j - v_i) \tag{3.7}$$

$$v_i' = v_i^* + \tfrac{\mu}{m} \sum_{j=1}^{m} (u_j^* - v_i^*) \tag{3.8}$$

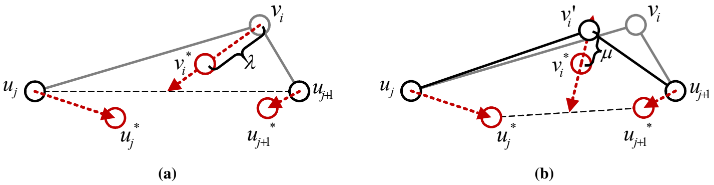


Figure 3.14: *Illustration of smoothing with Taubin's $\lambda|\mu$ filter. During each smoothing cycle, two steps are executed: (a) a first smoothing step and (b) subsequently a correction step.*

Node Position Constrained Smoothing

The Laplace+*HC* and LowPass filter try to preserve the volume and shape of the original structure by performing additional correction steps. This may, however, still alter the shape and lead to deformations. As described in Section 3.2.1, CESN [Gibson, 1998] and Dual MC [Nielson, 2004] restrict the location of the generated vertices to the cells defined by the size of the considered voxels. This concept can also be adopted for mesh smoothing by restricting the vertex movement to predefined cells. In this context, Bade et al. [2007a] presented diamond-constrained smoothing, which restricts vertex movement to a diamond-shaped volume centered at the cell edges. The method is especially dedicated to surface meshes generated via MC from binary segmented data, but can be used to extend any smoothing filter as an additional constraint. It does not require any additional parameters and prevents the surface from being altered too much. Similarly, Hildebrandt and Polthier [2007] restricted vertex movement to a user-defined distance to their initial positions. Finally, the restriction of vertex movement can preserve accuracy to a certain amount. The reduction of staircase artifacts depends on the smoothing algorithm to which the constraint is applied.

Feature-preserving Smoothing

The above described smoothing filters are applied to all vertices of a mesh with uniform weights. Even though the effects of these weights vary locally, smoothing does not account for local features and application-specific properties. The intention of feature-preserving smoothing is to detect salient regions and prevent these regions from being modified during model generation and smoothing. An example has been presented by Kobbelt et al. [2001]. They reduce aliasing artifacts by adjusting the data sampling according to local features. As a result, a standard MC algorithm can be used to extract a surface containing sharp, unaliased features. For surface meshes, anisotropic smoothing comprehends the modification of the smoothing filter according to local features. Thus, smoothing is performed along feature edges but not perpendicular to a feature (e.g., [Yagou et al., 2003; Chen et al., 2003; Meyer et al., 2003; Hildebrandt and Polthier, 2004; Yoshizawa et al., 2006]). Several authors adopted the idea of bilateral filtering in image processing [Tomasi and Manduchi, 1998] to 3D meshes. In their non-iterative smoothing filter, Jones et al. [2003] regard mesh denoising as a statistics problem where surface noise is considered as data outliers. At first, local first-order predictors (tangential planes) are defined for all triangles. The new (smoothed) vertex positions are then found using a weighted sum of the predictions in each vertex' spatial neighborhood. The weight function and predictors ensure a preservation of sharp features. Fleishman et al. [2003] proceed similarly, but have some slight differences in the generation of the surface predictor. The goal of these anisotropic, feature-preserving methods is also their main drawback during the application to anatomic surfaces. Anatomic structures exhibit organic smooth shapes without sharp features. Any method preserving or emphasizing such features would also maintain staircase artifacts while concurrently reducing noise in other surface areas.

3.4 Assessment of Smoothing Quality

In the former sections, model generation and especially mesh smoothing were discussed. During that, the words *accuracy* and *smoothness* have been used to sketch the constraints during medical model generation. Trying to achieve both accuracy and smoothness is usually a contradictory task. In addition, simulation applications may also demand a certain quality of the mesh structure itself.

3.4.1 Accuracy

Accuracy comprehends the preservation of the volume and features. As described earlier, Laplacian smoothing alters the volume such that convex regions tend to collapse. Similarly, concave regions would grow and thus yield an increase of the volume. Both effects should be minimized during mesh smoothing. Moreover, since relevant shape characteristics shall be maintained, local differences between the original and the final surface should also be minimized. This implies that also distances to neighboring structures should not be modified to keep the spatial relations. Hence, achieving high accuracy can usually be understood as performing only minor surface modifications.

Besides comparing the volume before and after smoothing, accuracy is assessed by determining, e.g., the average Euclidean distance and two-sided Hausdorff distance between the initial surface V and the processed surface V'. For a precise measurement, the Euclidean distances should be determined from the vertices of one mesh to the faces of the reference mesh, e.g., from V' to V and vice versa (see also Sec. 6.4.1). In this context, the Hausdorff distance gives an impression on the maximum deviation between the two meshes. The average Euclidean distance serves as approximation of the overall error.

3.4.2 Smoothness

A smooth surface does not contain any sharp features and exhibits only small bumps and bulges. Thus, a surface is considered smooth if its curvature values are low. For every point on a continuous 3D surface, curvature can be measured along two principal curvature directions – the directions with the maximum and the minimum surface bending κ_1 and κ_2 whose values are determined from the reciprocal of the local curvature radius. Both curvature directions are orthogonal to each other. From κ_1 and κ_2 one can construct the mean and the Gaussian curvature. The mean curvature corresponds to the average of the principal curvatures $H = \frac{\kappa_1+\kappa_2}{2}$. The Gaussian curvature is calculated as their product: $K = \kappa_1 \cdot \kappa_2$.

Since these curvature measures are derived from the 2nd order partial derivatives, they can only be defined for at least C^1 continuous surfaces. Thus, such curvature calculation cannot be applied to triangle meshes, which may only be C^0 continuous. For polygonal meshes, as considered in this thesis, curvature values have to be approximated by fitting quadric or cubic surfaces to the mesh (e.g., [Meyer et al., 2003; Razdan and Bae, 2005; Botsch et al., 2010]). Thus, there exist several approximations for calculating curvature values. An example of the mean curvature, as presented by Meyer et al. [2003], is employed in Section 6.4.2.

As mentioned before, smoothing aims at an overall reduction of curvature values. In the context of mesh smoothing, 3D surfaces are usually evaluated by their average, minimum, and maximum curvature values before and after smoothing.

3.5 Summary

The generation of surface models from tomographic medical image data is a highly complex process which underlies several error sources. Beginning with the image acquisition, different artifacts are introduced which need to be handled by subsequent data processing and model generation. Preventing surface artifacts by processing the source image data can be helpful for some issues (e.g., noise or merging structures due to intensity inhomogeneities), but might also be dangerous, since altering the image data may strongly modify the resulting surface. A reason is that image processing operators (smoothing, erosion, dilation, resampling) do not involve information on the individual topology of the structure. Hence, initially separated but close surface parts can easily get merged. Thus, the processing of the resulting surface mesh is often a valid alternative which gives better control over certain aspects.

As typical steps in common model generation pipelines, decimation/subdivision, remeshing, and smoothing are applied to prepare the extracted surface for the desired application (see Chap. 2). From these tasks, especially mesh smoothing serves for reducing staircase artifacts. The literature provides several algorithms which are successful at reducing noise in, e.g., CAD data, but are not suited for organic looking surfaces. The work of Bade et al. [2006] showed the suitabilty of some mesh smoothing filters for anatomic models. They showed that smoothing filters cannot be applied to arbitrary medical surface models without adapting them to the specific shape. The usage of smoothing parameter presets for different shape categories (compact, flat, elongated) seems to be promising. Since anatomic structures might exhibit more complex shapes, this idea might be further improved. In addition, the local restriction of vertex movement can preserve a certain accuracy, but smoothing is still performed globally. Except for surfaces from binary segmented data, staircase artifacts are usually not uniformly spread over the surface, such that a local treatment of artifacts might be more suitable.

4

Optimization of Surface Models for Computational Fluid Dynamics and Rapid Prototyping

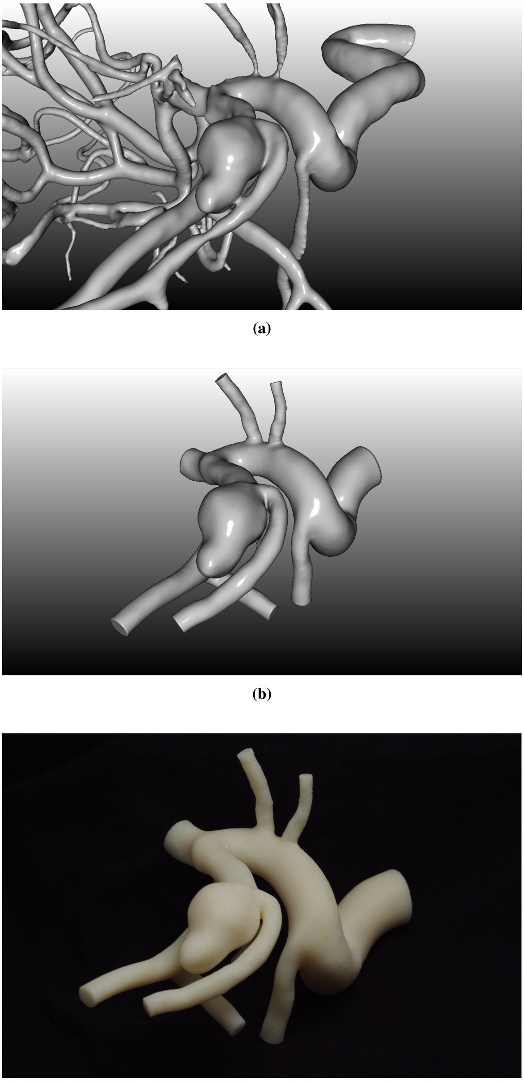
Table of Contents

The previous chapter described the general background for generating surface models from tomographic medical image data. In practice, the model generation workflows are highly dependent on application-specific requirements and artifact types and sources. Depending on the employed segmentation and mesh generation methods, an initial surface model may contain several and different artifacts. Due to their often complex shape, surface extraction of vascular structures is sensitive to the various artifact causes. During image acquisition, low resolution, partial volume, and beam hardening effects as well as inhomogeneous or insufficient contrast agent distribution disturb the representation of the vessel lumen. As a consequence, vessels might occur locally narrowed or interrupted. Beam hardening can cause small vessels to visually blend with closely located, bigger, high-contrast vessels. These artifacts vary to a certain degree, e.g., such that adjacent or outgoing vessels are represented as high frequency noise on the surface of a bigger vessel. Other artifacts, like staircases, may occur during segmentation and subsequent mesh generation. These steps are highly dependent on the contrast agent distribution. The dispersal of a contrast agent takes a certain amount of time, such that veins are enhanced only after a certain period of time. Thus, veins and thin vessels are often underestimated during segmentation and model generation. Additionally, vascular structures are not rigid. Among other effects, they pulsate according to blood pressure and pulse rate. These local deformations may also interfere with the time for image acquisition and yield an incorrect vessel representation.

The resulting noise, staircases, blended vessels, abruptly changing vessel diameters, and deformed vessel profiles need to be removed faithfully. These tasks are, considered separately, not very complicated and can often be handled with specific tools and algorithms. However, the overall process to achieve appropriate models for CFD or RP requires a lot of manual effort in different software tools. This may alter patient-specific properties of the target structure. Thus, medical experts need to be involved to validate the intermediate model adjustments as well as the final surface model. To give an impression of the complexity and manual effort in multiple software tools, the following sections describe such an exemplary model generation pipeline for vascular surface models and their subsequent usage in CFD and rapid prototyping. The contents of this chapter have originally been published in [Mönch et al., 2011b]. They are joint work with Mathias Neugebauer, who generated numerous aneurysm surface models and developed methods for exploring the blood flow in cerebral aneurysms [Neugebauer, 2014].

4.1 Generation of Vascular Surface Models

For vascular structures, special acquisition techniques (e.g., magnetic resonance angiography (MRA), computed tomography angiography (CTA), rotation angiography, time-of-flight MR angiography) are employed to achieve a high contrast to the surrounding tissue. This eases direct mesh extraction from the intensity data, but may still yield several artifacts in the resulting surface meshes. Basically, standard segmentation methods (e.g., thresholding, contouring methods, watershed transform, region growing) can be employed for delineating the vasculature and subsequently generating a surface model.



(a)

(b)

(c)

Figure 4.1: *(a) Example of an initial vascular surface model with an aneurysm and several artifacts at the branches. (b) Result of the complex model generation and artifact correction pipeline. (b) Final model obtained via 3D printing.*

These methods do not account for the characteristics of vascular structures. For faithfully depicting the complex shape, e.g., of the cerebral artery or the liver's portal vein, specialized segmentation and reconstruction methods are more suitable.

Segmentation

For instance, Bogunović et al. [2011] segment cerebral vasculature and contained aneurysms via customized geodesic active contours. The quality of the achieved segmentation result depends on the complexity of the vascular geometry as well as the homogeneity of the intensity values. Frangi et al. [1999] used a B-spline curve for representing the vessel centerline, which is coupled to the vessel wall surface. A shape model has also been employed by de Bruijne et al. [2002] for segmenting abdominal aortic aneurysms. These and other available methods are usually designed for extracting very specific vessels, but are not generally applicable to arbitrary vasculature. Depending on the type of the underlying model, even pathologic shapes can be detected and extracted to a certain degree. An overview of these and related vessel segmentation techniques was given by Lesage et al. [2009].

Mesh Generation

The segmentation information can then be employed for mesh extraction via, e.g., Marching Cubes algorithm [Lorensen and Cline, 1987], level-sets [Whitaker, 2000], or model-free implicit methods [Braude et al., 2007]. Again, these methods are generally applicable, but do not include knowledge on vascular shapes. Their usage may lead to strong inaccuracies and surface artifacts, such as unwanted vessel blending or narrowed vessels, since the surface is generated purely using intensity values. Model assumptions, such as the description of a vessel via its skeleton and local diameter information, can be used to dispose of segmentation errors and artifacts. In this sense, Hahn et al. [2001] described the usage of truncated cones, whereas Oeltze and Preim [2005] used convolution surfaces for generating naturally looking vessels. Convolution surfaces model the surface implicitly, which finally allows to generate smooth transitions at branches. Unfortunately, the model assumptions make such methods useless for representing pathologies in vascular structures, such as aneurysms. Image acquisition artifacts may also be critical for many model-based segmentation and surface extraction methods. Recently, Kretschmer et al. [2012] presented a surface generation method based on implicit sweep surfaces which handles varying vessel profiles and thin branches. Based on centerlines and corresponding outline information, the technique generates intersection-free, smooth surfaces which are suitable for CFD applications [Kretschmer et al., 2013]. Sibbing et al. [2012] employed the skeleton of the the vascular structure, too. Their method analyzes the vessel topology to detect junctions and tube elements. For each of these parts, the quad mesh is generated separately to finally obtain a smooth and simulation-suitable surface.

Schumann et al. [2007, 2008] modified the MPUI method of Braude et al. [2007] for obtaining an appropriate vessel representation. They refined the point cloud generation to account for typical features of vascular trees, such as the elongated shape and branchings. With an appropriate image subsampling scheme, small vessels can be reconstructed and since no shape model is used, even pathologic deviations can be expressed. However, only binary masks serve as input, which makes the MPUIs less accurate and partially sensitive to image inhomogeneities if these are not considered during the preceding segmentation. Bade et al. [2007b] employed the approach of Schumann et al. [2007] and combined it with an additional mesh optimization step to obtain surface meshes being suitable as input for CFD. They improved the triangle quality by flipping and collapsing the edges of very small or thin elongated triangles.

Similar to these MPUI methods, Wu et al. [2010] employed point clouds from binary masks for generating a 3D implicit indicator function, which is subsequently used as input for polygonization. Such specialized model generation methods for vascular structures succeed with reducing some artifacts, as they guarantee a certain smoothness and vascular shape regarding the vessel profile, but also at branching points. Manual effort is often required for artifacts, such as incomplete contrast agent dispersal or touching vessels. As an example, beam hardening artifacts and closely located vessels may yield an unwanted blending of separated structures (see Fig. 4.5 and Sec. 5.5). There are no specialized methods available to remove, e.g., blending artifacts or narrowed vessels faithfully. Available algorithms for feature detection, feature depending smoothing, and interpolation-based methods are only specific solutions, especially to the staircase problem, but cannot remove all potential artifacts. To receive an appropriate surface mesh for CFD or RP, several external tools and expert feedback need to be employed (see Fig. 4.1) to manually remove artifacts. Smoothness is not only an essential requirement for simulation applications – it may also be a basic prerequisite for generating certain visualizations. as an example, depicting flow characteristics and surface properties at the same time requires to reduce sharp surface features such as artifacts to allow for a better visual perception of the flow information inside the vessel [Gasteiger et al., 2010; Lawonn et al., 2013a, 2014], but also of flow characteristics mapped onto the surface [Neugebauer et al., 2009].

The VASCULAR MODELING TOOLKIT (VMTK) [Antiga et al., 2003, 2008; Piccinelli et al., 2009] is a framework for the reconstruction and geometric analysis of vascular structures. It supports the development of software tools which require an analysis of vascular structures. However, methods for a systematic removal of specific artifacts are not included.

Mesh Quality

For the patient-specific simulation of blood flow, further requirements, such as mesh resolution and triangle quality play an important role [Cebral et al., 2001, 2011]. The shape of the mesh's triangles is required to be almost equilateral and homogeneous over the mesh, since the surface models are used as input for volume mesh generation (see Sec. 2.2.2). Thus, triangle quality and size influence convergence and accuracy of the simulation computations. As an example, Neugebauer et al. [2008] described a workflow for generating vascular surface models as input for CFD simulations. They generated the initial surface mesh with the MPUI method of Schumann et al. [2007] and performed a simplification step to remove very small and degenerated triangles. The mesh quality is then achieved by applying isotropic remeshing [Surazhsky et al., 2003].

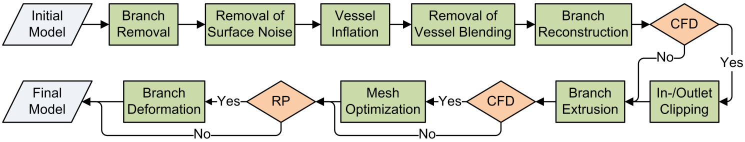


Figure 4.2: *The steps of the employed surface optimization pipeline. (Image reprinted from [Mönch et al., 2011b])*

After generating a volume mesh, they discussed the relation between triangle quality and simulation results (e.g., wall shear stress, flow velocity). Neugebauer et al. [2008] emphasized the importance of a surface and volume grid resolution which is adapted to the desired accuracy, especially for potentially complex flow patterns in aneurysms.

An optimization of the mesh quality can also be achieved by employing an advanced front remeshing algorithm [Schöberl, 1997]. The surface generation method by Wu et al. [2010] does also take care of mesh quality during polygonization of their implicit function, which may reduce the number of steps to be taken within the model generation pipeline. Augsburger et al. [2009] have shown that the mesh extraction procedure may yield strongly varying simulation results depending on the involved segmentation and artifact reduction methods. Similarly, Cebral and Löhner [2001] describe the strong dependence of blood flow characteristics on the vessel geometry. This dependence is also discussed in Section 5.5.

The above-mentioned artifacts and tasks are well-known. There are several methods and software tools available to account for individual artifacts, but in several cases only medical experts can distinguish between an artifact and a pathology. There exists no unified solution that allows users to generate an artifact-free surface model for CFD or RP.

4.2 Optimization Pipeline

In this chapter, especially CFD and RP are considered as sample applications for model generation. As discussed in Sections 2.2.2 and 2.3, these applications have more specific requirements than, e.g., pure visualization tasks. Both CFD and RP have the following tasks in common:

- definition of a region of interest
- removal of artifacts resulting from image acquisition:
  - incomplete contrast agent dispersal
  - beam hardening
  - inhomogeneities
  - image noise
- staircase and surface noise reduction

- branch extrusion

Within this chapter, these tasks will be discussed as parts of a processing pipeline for a cerebral aneurysm (see Fig. 4.2): starting with a mesh of the vascular tree and the aneurysm, unnecessary distant vessel branches or branches that are – in terms of flow direction – located behind the aneurysm, are removed. Subsequently, vessel blending artifacts, underestimated vessel diameters and surface noise are corrected. If necessary, insufficiently represented branches that are necessary for CFD or RP are reconstructed. After faithfully correcting the vessel surface, optimization steps which are specific either for CFD or RP have to be performed. CFD requires a definition of inlet and outlet areas by cutting vessel branches perpendicular to their centerline. This may also include an artificial elongation of the respective branches. Moreover, a final remeshing step is required to globally improve the mesh quality. For usage as RP input, the model may require further modification, e.g., if surfaces are too close, if the vessel bending is too strong, or if it is not possible to apply a more or less planar cut through the complete model. The latter is required if the RP model is obtained from consecutively filling a mold. Other RP variants may have additional special requirements. These mentioned alterations are done by locally changing vessel diameters or bending vessels.

Details of each step are presented in the following sections, where the focus is put on artifact reduction solely on the mesh level. Alternatively, some artifacts could be reduced by modifying segmentation masks and the image data. Depending on the respective artifact, this may be more complicated and might introduce further artifacts [Mönch et al., 2011a]. The following processing pipeline is one possible workflow, which has been consolidated according to the demands of clinical and technical partners.

4.2.1 Employed Software Tools

To achieve the above-mentioned optimization of surface models for CFD and RP, different software tools have to be employed within a complex, manual workflow. These tools had been chosen due to their specific functionality and personal experience. However, they may be replaced by alternative software according to individual preferences and requirements.

- BLENDER[1] and 3DS MAX[2] both belong to the category of 3D modeling software. The functionality comprises tools for rendering, animation, and modeling. The latter refers to creating and modifying 3D models. Due to this functional range, basically all workflow steps could be performed within such software suites. However, the handling requires a lot of experience and the interaction is often too complex for single tasks. GEOMAGIC FREEFORM[3] could also be suitable in this context. It allows for haptic 3D input devices for model editing, but it has not been tested in the presented workflow.
- SCULPTRIS[4] provides several tools, which comprise, e.g., dilation, extrusion, and smoothing of surface meshes. The operators are applied locally by brushing

[1] http://www.blender.org/

[2] http://usa.autodesk.com/3ds-max/

[3] http://geomagic.com/en/products/freeform/overview/

[4] http://www.sculptris.com/

them over the surface. Operator size and strength can be adapted to individual requirements and local shape properties.

- MESHLAB[5] is a software which provides several well-known mesh processing algorithms, e.g., for mesh repair, hole filling, decimation, subdivision, and smoothing. Several of these filters can be applied to local selections of vertices and faces.
- NETGEN[6] [Schöberl, 1997] is a tool for surface remeshing and volume mesh generation. Surface meshes serve as input and can be optimized w.r.t. mesh quality under consideration of several user preferences. Besides surface mesh optimization, tetrahedral volume meshes can be generated.

4.2.2 Branch Removal

In CFD, the simulation is performed for all volume cells which are circumscribed by the surface model. Although a volume cell may have no or only a small relevance for the simulation result, it is considered during all computations. To limit computational effort, the simulation has to be restricted to those parts of the model which are required for the subsequent data interpretation. Thus, branches that do not directly affect the blood flow behavior in the target area (e.g., the aneurysm), need to be removed.

In the current workflow, branch removal is achieved by a combination of different software tools. At first, the target branch is clipped using a 3D modeling software (e.g., BLENDER or 3DS MAX). To remove the branch, a clipping geometry (e.g., a cube, a plane, ...) is specified. After adjusting position and orientation of the clipping geometry, Boolean operations are applied to remove the negligible branches. The clipping operation results in a slight, flat bump on the main vessel and a closed surface. This bump can be reduced via mesh smoothing, which should be performed only locally to prevent from alterations in other parts of the model. Local smoothing is achieved by iteratively using SCULPTRIS (see also Sec. 4.2.3).

Alternatively, branch removal can be performed by cutting the target branch directly at the furcation on the larger main vessel. Cutting out this branching area leaves a hole in the main vessel which needs to be closed afterwards. This hole can be closed, e.g., via MESHLAB, BLENDER, or 3DS MAX. In this specific case, 3D modeling software provides better control for adjusting the new surface patch to the profile of the parent vessel. Depending on the size and profile of the cutting area, the closed hole may also be subject to manual local deformation via SCULPTRIS. If available, dedicated algorithms, such as presented by Zhao et al. [2007], could also be employed for an automated, plausible hole filling.

[5] http://meshlab.sourceforge.net/

[6] http://sourceforge.net/projects/netgen-mesher/

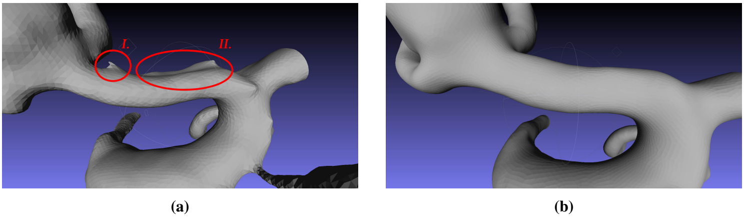


Figure 4.3: Example of rudimentary vessel branches and slight vessel blending (see labels I. and II.) (a) before and (b) after local smoothing with SCULPTRIS. *Screenshots of* MESHLAB. *(Images reprinted from [Mönch et al., 2011b])*

4.2.3 Noise and Bump Removal

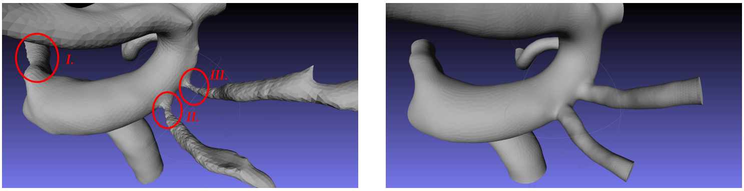
After initial mesh extraction and branch removal, the surface may contain vessel rudiments (see Fig. 4.3a) due to incomplete contrast agent dispersal, beam hardening, or remainders from previous surface editing tasks. A possible solution are smoothing filters. However, typical uniform smoothing will cause strong volume shrinkage of the whole model and removes relevant details. Especially for vascular structures, locally adaptive filters are required to perform smoothing only in the artifact areas.

Since the artifacts being target for smoothing operations may vary in their shape and size, an automated approach detecting the artifacts reliably is very complicated. Thus, the most useful solution is an interactive approach where the user brushes over the artifact area. During brushing, all vertices in a defined neighborhood (topological or Euclidean distance) are smoothed appropriately. SCULPTRIS is an appropriate tool to achieve this. For a removal of bumps, a local smoothing operator can be used, which is parametrized by its size and strength. By doing so, artifacts can be removed in an iterative, manual procedure. This has the benefit that context and expert knowledge about the specific anatomic shape is used subconsciously.

4.2.4 Vessel Inflation

Locally narrowed vessels (see Fig. 4.4a) may arise from image inhomogeneities resulting from the image acquisition technique or incomplete contrast agent dispersal. They have to be adjusted for two reasons:

1. The geometry of the structure is incorrect and may thus yield wrong conclusions during visual inspection or during exploration of resulting RP models.
2. For usage in CFD, non-converging simulations or wrong simulation results have to be expected for, e.g., wall shear stress and flow velocity, but also vortices may occur. Altogether, these affect the (simulated) flow behavior within the whole model.



(a)

(b)

Figure 4.4: *(a) Thin, anatomically incorrect vessel branches (see labels I., II., and III.). (b) The vessels after local inflation and branch clipping. Screenshots of* MESHLAB. *(Images reprinted from [Mönch et al., 2011b])*

To resolve this, the vessels need to be inflated locally (see Fig. 4.4b). The artifact areas can basically be detected automatically by determining the vessel centerlines and comparing the change of the vessel diameters along the centerlines. This implies a circular vessel shape, which must not be true in all cases. Vessels can be slightly flattened due to pressure from surrounding structures or they may contain pathologic variations, such as aneurysms. Thus, an automated procedure may be error-prone, but could be used to support the user to identify these areas faster. After identification of possibly narrowed vessels it is essential to refer to the image data again to validate the narrowing before any further correction.

The narrowing artifact can be corrected by using SCULPTRIS with a combination of the provided "Inflate" and "Smooth" tools. For both tools, operator size and strength need to be adjusted to fit to the size and diameter of the target vessel. By doing so, the vessel profile is iteratively corrected – again supported by the human perception and understanding of the most plausible shape.

4.2.5 Removal of Vessel Blending

Vessel blending is a frequent artifact, which occurs if two structures are located very close such that they (almost) touch each other. In such a case, partial volume effects, beam hardening artifacts, and image inhomogeneities may yield blending of both structures at their closest point in the image data and, hence, in the resulting surface models (see Fig. 4.5a). Such artifacts may arise locally restricted at touching vessels, but also very expanded if a vessel passes another vessel over a long range. In particular the latter is a critical situation, since the involved vessels are visually hard to distinguish and thus hard to faithfully divide.

The separation of blended structures is a complex problem whose specific solution depends on the data and the extent and shape of the artifact. Once more, 3D modeling software is employed to perform the mesh editing tasks. Thus, the mesh can be cut along the desired path to split the blended vessels. This process may be tedious, since the cutting path can be complex and needs to be drawn precisely on all sides of the artifact (see Fig. 4.5b). Especially finding an appropriate alignment of the cuts on the front and back side may be very complicated.

The resulting hole needs to be closed and can be of very complex shape. To simplify this, the hole is "subdivided" into several smaller ones by manually adding support triangles. This eases final hole filling, since each single small hole may be easier to fill. The support triangles are added at critical points, where the "shape" of the hole changes significantly.

Hole filling can again be performed via MESHLAB or 3D modeling software. In the current case, MESHLAB was used to detect the multiple small holes automatically and to fill them afterwards. For complex artifact shapes, this cutting and hole filling procedure may not be efficient anymore.

As an alternative, a *stamping-like* procedure in 3D modeling software is also performed. This requires to generate a stamping geometry (e.g., a cylinder or cuboid), which is then aligned with the blending area. This gives a good preview of the resulting separation and the holes in the target model. After correct placement and slight adjustment of the stamping geometry, the artifact is cut using constructive solid geometry (CSG). CSG employs Boolean operations and yields correctly closed surface meshes. Depending on the artifact shape, this needs to be repeated several times but does still save a lot of effort, since the manual specification/drawing of support triangles and subsequent hole filling is not necessary.

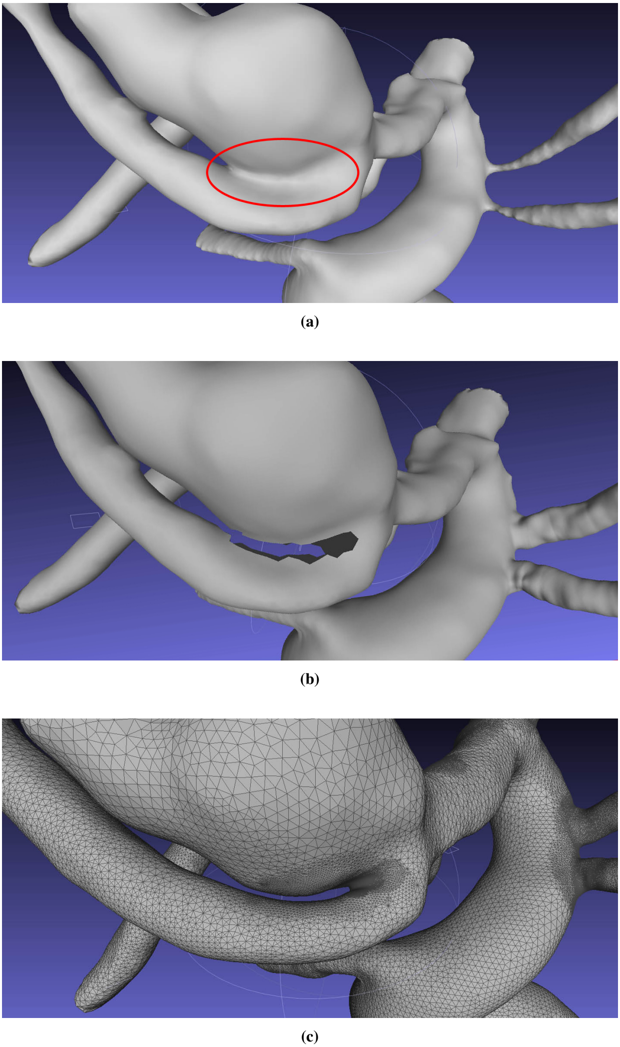
After applying one of the above described artifact removal operations, local adjustments via SCULPTRIS may still be necessary. As a prerequisite for usage in SCULPTRIS, local remeshing and subdivision may be necessary (e.g., via BLENDER, 3DS MAX). Applying local, interactive smoothing or inflation, the artificial vessel surface is modified to achieve a plausible vessel shape and profile. Furthermore, for usage in RP, the distance between the separated vessels needs to be considered to prevent an anew blending during the final building of the physical model (see Sec. 4.2.10).

4.2.6 Branch Reconstruction

Incomplete contrast agent dispersal and segmentation may also yield detached vessels. If such detached parts are essential for visual inspection or further evaluation of, e.g., the flow behavior, they need to be reconstructed to allow for a faithful virtual representation of the specific patient anatomy. The reconstruction of single branches requires a sophisticated understanding of medical image data and patient anatomy. Moreover, experience with 3D modeling software is implied. Branch reconstruction consists of a lot of manual effort, since two vessel rudiments need to be reconnected and thus be manually modified.

For each vessel rudiment, cutting operations (and possibly local deformations) are necessary in order to reconstruct a valid vessel profile. These open vessel profiles may then be connected in different ways:

1. They may be extruded until they match each other, whereas it is unlikely that both endings will perfectly match after a linear extrusion. In addition, this will cause



(a)

(b)

(c)

Figure 4.5: *Example of vessel blending and removal of the artifact. (a) Initial model with blending artifacts, (b) after manual cutting, and (c) after hole filling and smoothing. Screenshot of* MESHLAB. *(Images reprinted from [Mönch et al., 2011b])*

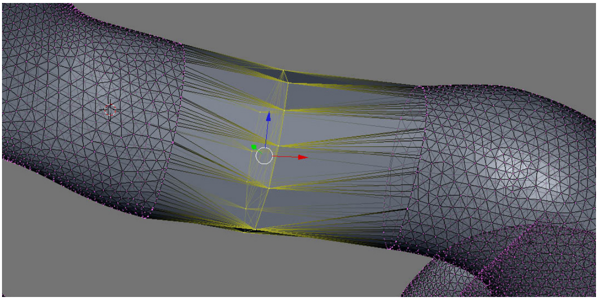


Figure 4.6: *Screenshot of* BLENDER *during reconstruction of a formerly disconnected vessel branch. (Image reprinted from [Mönch et al., 2011b])*

extremely stretched triangles, which comprehends a low triangle quality (see also Sec. 4.2.8 and Fig. 4.6). Thus, a plausible deformation (see Sec. 4.2.10) and local remeshing are also required for a better approximation of the vessel geometry.

2. Both open vessel rudiments may be connected via an artificial, predefined vessel geometry, e.g., a tube. The endings of the tube have to be fitted to the open vessel profiles and their vertices require correct merging. Again, the tube needs to be deformed to fit the centerline of the real vessel geometry.

Due to the complexity of these tasks, 3D modeling software is required to perform these operations (see Fig. 4.6). The procedure involves a lot of interaction to select the correct vertices and to drag them towards their new position. Along the newly reconstructed vessel geometry, the centerline needs to be adjusted to plausibly match the estimated shape of the original vessel. As a last step, the vertices at the "touching" open vessel profiles need to be merged for a correct triangulation.

An automated procedure could support the user by interpolating the vessel centerline between the vessel rudiments. This would allow to extrude the vessel profiles along this centerline without the necessity of subsequent deformation. For the tube geometry approach, the centerline of the tube could automatically be aligned with the interpolated vessel centerline. However, the vertices at the open profiles of the tube and the vessel rudiments still have to be merged correctly.

Besides a deformation along the vessel centerline, slight deformations of the vessel profile will also be necessary to align the perfect circular shape to the vessel profiles. For that, again, SCULPTRIS can easily be applied by using the provided brushing tools for inflation, deflation, and smoothing.

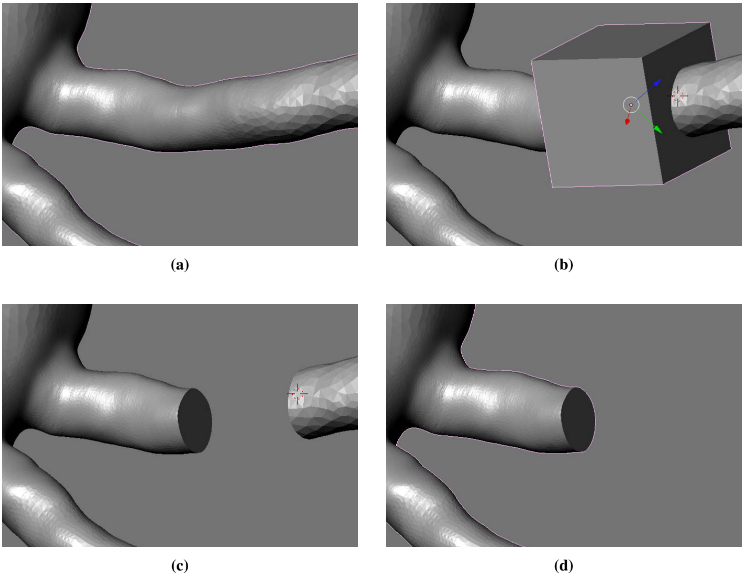


Figure 4.7: *Generation of an in-/outlet: (a) Initial branch; (b) Vessel with clipping box; (c) Result after clipping of an intermediate part; (d) Result after removal of the remainder. Screenshot of* BLENDER*. (Images reprinted from [Mönch et al., 2011b])*

4.2.7 In-/Outlet Clipping

For application in CFD, inlets and outlets need to be specified (see Fig. 4.4b and Fig. 4.7d) to define regions of blood inflow and outflow. Similar to branch removal in Section 4.2.2, 3D modeling software is employed for generating the in- and outlets.

After creating a clipping geometry, its location and orientation are iteratively adjusted according to the vessel centerline (see Fig. 4.7b). For CFD, the clipped areas are required to be preferably perpendicular to the vessel centerline to prevent from additional flow artifacts. This process has to be repeated for all contained vessel branches. Finally, Boolean operations are performed to clip the vessel perpendicular to the centerline (see Fig.4.7d). It has to be ensured that clipping yields a closed surface mesh.

This clipping procedure is not complicated but does still take some time if several vessel branches have to be edited. User interaction could be supported by an automated alignment of clipping volumes perpendicular to the centerline, where the user only needs to drag each clipping geometry along the centerline. Such functionality is, however, typically not contained in the available 3D modeling software.

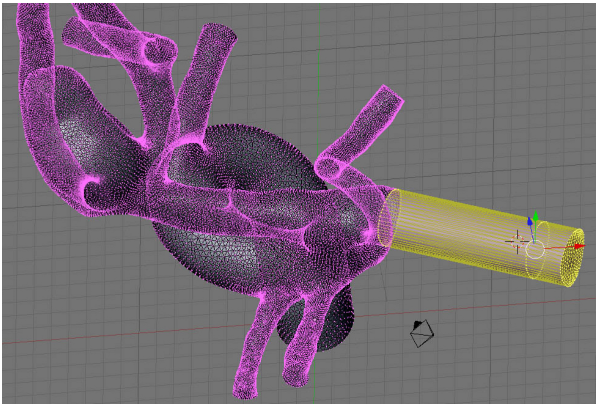


Figure 4.8: The in-/outlet area is selected and extruded to fit the requirements of CFD for a minimum vessel length at the in-/outflow areas. Screenshot of BLENDER. *(Image reprinted from [Mönch et al., 2011b])*

4.2.8 Branch Extrusion

For CFD, the in- and outlets require a minimum length of the adjacent vessel. The inflow and outflow areas must not be directly adjacent to bent vessel parts to achieve more stability during simulation. Since this condition is often not fulfilled, the branch has to be modified. This task is currently solved via BLENDER. The branch is edited by selecting the in-/outlet area and subsequently extruding it (see Fig. 4.8). Again, local remeshing will be necessary to prevent a bad quality at elongated triangles. However, this remeshing is also required to guarantee similar mesh properties (resolution and quality) as the adjacent vessel.

The manual effort could be supported algorithmically by extrapolating the centerline or by using the average triangle normal of the in-/outlet area. The extrusion operation is then performed along this new path – either manually or fully automatic.

Besides CFD, RP may also require branch extrusion. This requirement arises from the phantom building procedure, e.g., if a mold is created and needs to be filled. Thus, the vessel extrusion is used as casting channel. In such a case, the extrusion may not match further anatomic requirements.

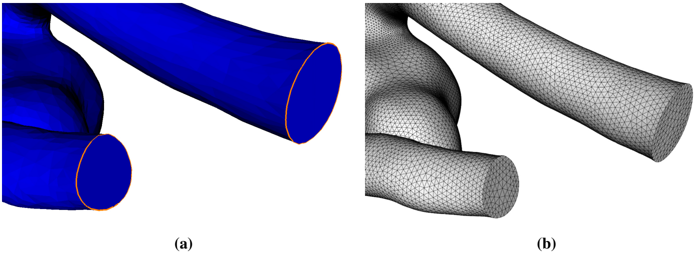


Figure 4.9: (a) The feature edges at the in-/outlets are highlighted after manual parameter adjustment (feature angle and feature size; Screenshot of NETGEN*). (b) Result after remeshing. (Images reprinted from [Mönch et al., 2011b])*

4.2.9 Mesh Optimization

Mesh quality has been mentioned several times in the previous sections. It refers especially to a good triangle edge ratio and a homogeneous triangle size. These are required to ensure stability and convergence of the simulation computations.

A common tool to achieve a this is NETGEN. To preserve the sharp edges, e.g., at the earlier specified in-/outlets, the feature edges have to be detected (see Fig. 4.9). Feature detection is performed semi-automatically by adjusting two parameters until the visual result (edge highlighting) fits the user's requirements. After user interaction and specification of the desired mesh granularity, NETGEN proceeds automatically with an advancing front algorithm. As a result, the complete surface model is remeshed, whereas the feature edges at the in-/outlets are maintained.

After performing all of the above tasks, the surface model is ready for subsequent generation of a volume mesh required for CFD.

4.2.10 Branch Deformation

Usually, the most important requirement during surface model generation is *accuracy*. Especially for usage in surgical planning, radiation treatment or CFD simulation it is prohibited to alter the shape of the target structure. In contrast, RP may require a local deformation of closely located vessel branches. Due to the building procedure (e.g., casting or fused deposition modeling), very close structures and vessel branches might not be represented correctly. Thus, building the physical model may introduce similar artifacts as those which are tried to be removed during virtual surface model generation (e.g., vessel blending).

For deforming vessel branches, hull volumes and harmonic coordinates [Joshi et al., 2007] are employed as provided by BLENDER. Thus, a hull volume is generated which is used to control the deformation of the inner target structure. In the current case, a hull volume is generated for each vessel branch that is located too close to other parts of the

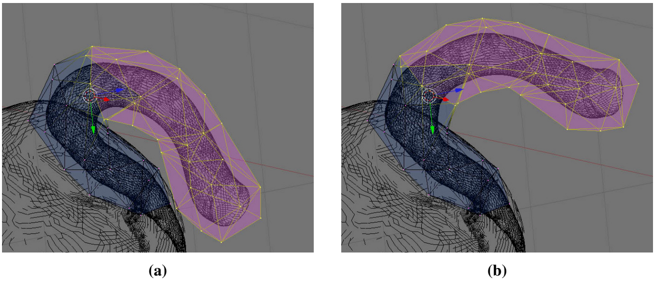


Figure 4.10: Example of vessel branch deformation via hull volumes. (a) Before deformation and (b) after deformation. Screenshots of BLENDER. *(Images reprinted from [Mönch et al., 2011b])*

model. The structure can then be deformed via mouse interaction until it fits the specific local demands. The underlying algorithm guarantees that volume and local properties of the vessel are preserved while its shape is modified (see Fig. 4.10).

4.3 Summary

The reconstruction of vascular surface models for applications, such as CFD or RP, is a complex task which involves a number of different steps. Among several requirements, accuracy and mesh quality play an important role, whereas RP may even require local deformations to account for constraints of the physical model building procedure.

Within this chapter, several specific problems and the related tasks for solving these problems have been discussed. An automated procedure to resolve the described issues is currently not known. A fully automated procedure can usually not be used, since the occurring artifacts require an extensive, often interactive treatment and expert knowledge to distinguish artifacts and pathologies. A further automation of individual tasks resulting in a semi-automatic, expert-guided procedure is, however, conceivable.

The presented solutions represent suggestions for preparing an exemplary surface model for usage in CFD and RP. Some of these steps may require to alter the shape of a target structure locally. For several artifacts, the original shape of the model (e.g., the vessel radius) is obvious, even to non-medical employees, and can thus be adjusted appropriately. The plausibility of these changes as well as the general shape of the intermediate and final models require validation by medical experts.

It is obvious that the described model generation process is heavily based on user-driven mesh editing. The target model needs to be exchanged between different separated software tools. A unified solution offering the described tools for remeshing, local smoothing, local inflation, and cutting is desirable and could help to guide the user faster through this process. In addition, such a platform would serve as a better basis for discussion and collaboration with medical experts.

The user may be supported in different ways. Most important is to organize the individual steps in a plausible manner, such that each step builds on the previous ones. Next, the provided tools should allow for resolving the various artifacts, but concurrently maintain relevant characteristics of the model. In this context, Chapter 5 describes context-aware smoothing to restrict smoothing to dedicated areas. During all of the described steps, the user is required to keep an eye on the original image data and on plausibility of the local modifications and reconstructions. Thus, visual and quantitative feedback – especially concerning model accuracy – may reduce the number of mesh editing, evaluation, and expert feedback cycles. Chapter 6 discusses this in more detail for mesh smoothing operations.

5

Context-Aware Smoothing of Medical Surface Models

Table of Contents

In their simplest form, smoothing methods apply a uniform filter to the surface mesh which may lead to uncontrolled loss of volume, features and accuracy. A large variety of filters is available which focus on the detection and preservation of features (see Sec. 3.3.4). The fact that vertices and edges with high curvature are treated as features represents context information. This context information is defined by the way how surface models are generated and by the application domain. For example, models derived via laser scanning usually suffer from noise. This noise has to be removed, but those features which define certain characteristics shall be maintained. Otherwise, the model might not be appropriate for conveying the domain-specific information, e.g., shapes or distances. Feature-preserving smoothing methods (e.g., Jones et al. [2003]; Fleishman et al. [2003]) try to detect these characteristics and adjust their weighting functions to perform less or no smoothing in these relevant mesh areas. Unfortunately, this context knowledge cannot be adopted for anatomic 3D models. Feature-preserving smoothing would reduce noise and small features, but staircase artifacts would remain and even get emphasized.

For the medical domain, such context information is often employed unconsciously in manual workflows (see Chap. 4). The surface is generated and modified locally according to knowledge on anatomic shapes, malformations, and characteristics of typical artifacts. Alternatively, local adjustments are performed with respect to application-driven requirements. An integration of context information (e.g., on the origin of artifacts) into smoothing methods is not known. This chapter discusses *context-aware* smoothing, which refers to the usage of such context information for adopting common mesh smoothing filters to artifacts in anatomic surface models. Thus, context-aware smoothing summarizes the publications for distance- [Mönch et al., 2010a] and staircase-aware smoothing [Mönch et al., 2010b, 2011a].

5.1 Adjustment of Smoothing Filters

The principle of mesh smoothing was already explained in Section 3.3.4. For each vertex, a displacement vector is determined according to its neighbors and different criteria. Since the length of this displacement vector determines how far each vertex is moved during one smoothing cycle, it can also be modified to promote or restrict smoothing locally. A standard Laplacian mesh smoothing filter is modified with an additional weighting factor w_{v_i}:

$$\forall v_i \in V: \quad v_i' = v_i + \tfrac{\lambda \cdot w_{v_i}}{m} \textstyle\sum_{j=1}^{m} (u_j - v_i) \tag{5.1}$$

$v_i, u_j \in V, \forall u_j \in U_{v_i}^1, m = \left|U_{v_i}^1\right|$

$U_{v_i}^1$ - 1st order neighbors of vertex v_i

λ - uniform smoothing factor

w_{v_i} - vertex-specific weighting factor

This principle can be applied to any smoothing algorithm. More complex algorithms, such as Laplace+*HC* or Taubin's $\lambda|\mu$ filter, can be employed with their default parameters for smoothing and back correction as usual. The vertex-specific weighting w_{v_i} is then applied to the final displacement vector, which is determined by each smoothing method with its individual parameters and weights.

The following sections illuminate this concept and generate locally varying weights to restrict or promote smoothing according to different conditions.

5.2 Distance Awareness

An example for application-driven requirements are the distances between neighboring structures in 3D diagnostic and surgical planning [Mönch et al., 2010a]. There, potential infiltrations of closely located structures need to be assessed. Especially the neck is a good example where several critical structures (e.g. carotid artery, jugular vein, sternocleidomastoid muscle (SCM), lymph nodes, salivary glands) are located very close and the exact local distances are relevant for the planning of surgical interventions or further treatment. For planning of an intervention to remove a tumor or enlarged lymph node, which is directly adjacent to these structures, the distances need to be determined and visualized correctly. If mesh generation and smoothing would alter those distances, the intervention planning could lead to wrong conclusions. To account for these requirements, smoothing should consider these dependencies. They can be expressed by measuring the (minimum) Euclidean distance between the target structures [Preim and Bartz, 2007, Chap. 13]. Involving the knowledge on the local neighborhood during mesh smoothing offers a way to generate surface models with reduced staircase artifacts on the one hand, while keeping accuracy in terms of distances to reference structures. Thus, *distance-aware* (DA) smoothing does not alter minimum inter-structure distances, whereas the distant parts of a structure are smoothed according to the selected method and corresponding parameters. Such an example is shown in Figure 5.1 with surface models of the trachea, jugular vein, and a tumor generated directly from binary masks without further steps for artifact reduction. The tumor model is color-coded according to the Euclidean distance to the neighboring structures. It seems to intersect the trachea and gets very close to the jugular vein. Any further modification of the data, e.g., smoothing of the image data or uniform Laplacian mesh smoothing, might have a strong impact on these spatial relations. In the image, it looks as if the tumor would almost touch the vessel. Any careless artifact correction operation might increase this gap, which could lead to another visual judgment of this relation.

To overcome the problem of altered inter-structure distances during mesh smoothing, smoothing weights have to be aware of the spatial context. For each vertex v_i of a given surface mesh M_1, the Euclidean distance to a given reference mesh M_2 is determined. This distance d_{v_i} is determined from each vertex v_i of M_1 to the faces of M_2. The resulting distance values are scaled to the range of [0, 1], whereas the closest vertex of mesh M_1 to M_2 is assigned a 0 and the furthest vertex is rated with 1. For more than one neighboring structure, distance computation is repeated for each reference model. If more than one distance value is available for a vertex, the minimum value is used. The scaled distance value is then applied to the smoothing filter (see Eq. 5.1) by: $w_{v_i} = d_{v_i}$.

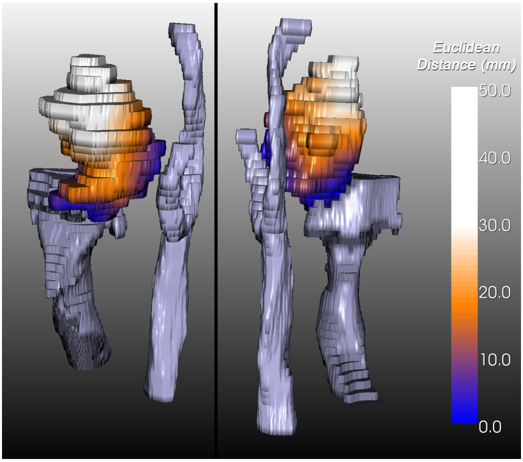


Figure 5.1: *Two views of three sample structures (trachea, jugular vein, tumor; the latter is colored by the Euclidean distance to the other structures) are shown which were generated from binary masks using Marching Cubes algorithm without further pre-/postprocessing.*

Inspired by the major application scenario (risk analysis around a tumor), an artificial dataset was created, which consists of a noisy plane with a sphere located close to it. Figure 5.2 shows this testing scenario to describe the influence of locally adaptive mesh smoothing. By default, the scaled Euclidean distance can directly be used for vertex displacement weighting. This yields a linear increase of the degree of smoothing with increasing distance. However, modified scaling themes will give the opportunity to focus on regions with higher or lower weighting and define "safe" regions which are not altered by mesh smoothing at all (see Fig. 5.3).

In surgery, relevant safety margins are e.g. 2 mm, 5 mm, and 10 mm. A sample safety margin of 10 mm would completely preserve vertex positions (weighting equals zero) within this range, whereas vertices with higher distance values are linearly weighted. Additionally, other, non-linear scaling functions such as in Equation 5.2 may be applied, which gives the opportunity to define differently smoothed regions with smooth transitions. Otherwise, the transition between smoothed and unsmoothed areas may be visually disturbing.

$$\forall v_i \in V: \quad w_{v_i} = \frac{e^{\omega d_{v_i}} - e^{\omega \min(D)}}{e^{\omega \max(D)} - e^{\omega \min(D)}} \tag{5.2}$$

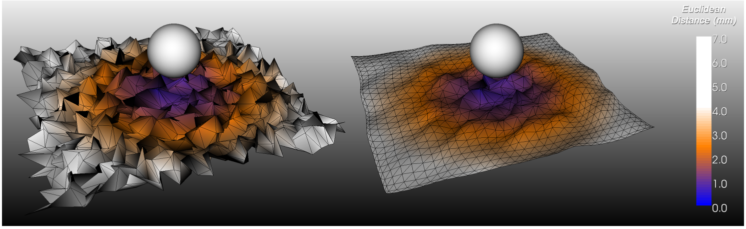


Figure 5.2: *Artifical testing scenario. Left: The initial surface models. Right: Distance-aware Laplacian smoothing applied to the plane with linear scaling. The plane is colored by Euclidean distance to the sphere (in mm).*

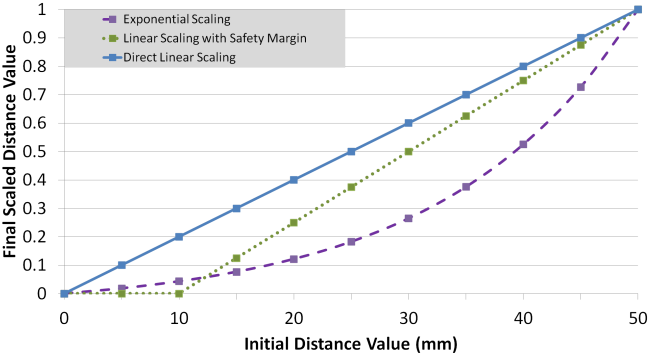


Figure 5.3: *The applied sample scaling functions are shown. By default, direct linear scaling of the Euclidean distances is used. Additionally, a safety margin or an exponentional function might be appropriate.*

ω - scaling parameter, e.g., $\omega = 0.06$ in Fig. 5.3
$v_i \in V; V$ - set of vertices of mesh M_1
$d_{v_i} \in D; D$ - set of Euclidean distances of M_1 to M_2

5.3 Staircase Awareness

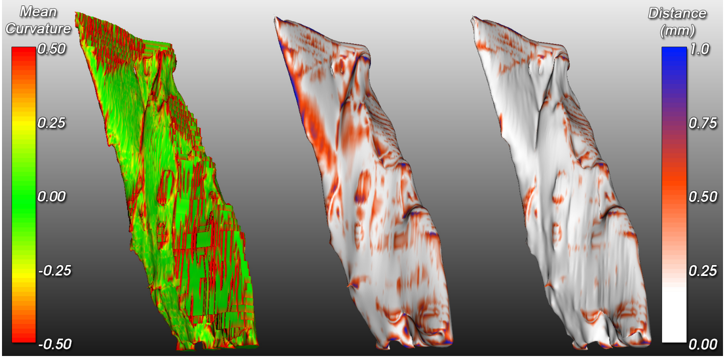
Depending on the individual methods used for segmentation and model generation, staircase artifacts may arise locally. If the artifacts are not uniformly spread over the surface (see Fig. 5.4a), non-artifact areas might get altered too much with typical smoothing filters (see Fig. 5.4b). Feature-sensitive smoothing (e.g., [Jones et al., 2003; Fleishman et al., 2003]) would again detect the staircases as features and indirectly emphasize them.

A removal of such artifacts is only possible, if they could be discriminated from anatomic features. The major differences between staircases and anatomic features are their shape and orientation. With a successful detection and localization of artifacts, they can adaptively be removed, whereas accuracy is preserved in non-artifact areas. This is important for 3D diagnostic or surgical planning applications.

For the simulation of blood flow, flow inlets and outlets are defined by clipping the vessel branches perpendicular to the centerline (see Chap. 4). The resulting flat areas (caps), which may be oriented arbitrarily with respect to the image stack orientation, would get distorted during further mesh processing (e.g., smoothing). Thus, manual clipping is usually the last step of workflow employed for generating volume meshes for CFD simulations. The detection and preservation of arbitrarily oriented caps would thus yield more flexibility in these workflows, since additional smoothing steps could automatically be carried out after defining the in- and outlets.

The segmentation of image data usually focuses on a defined region of interest, which often contains only a part of the structure. Especially elongated structures, such as vessels and muscles, are cut during the segmentation process resulting in flat, clipped areas. Caps, whose flat regions are oriented according to the image stack orientation (e.g., z-axis), might get detected as staircase areas and subsequently be smoothed. Furthermore, caps with a surface normal differing strongly from the image stack orientation might not get smoothed. In contrast to CFD, medical visualization might also require rounded caps to prevent abrupt endings [Hahn et al., 2001]. A correct and consistent handling of these endings requires their identification. This allows to preserve the caps for measurement tasks or to consistently generate smooth endings for visualization.

Staircase-aware smoothing implies knowledge on the image stack orientation (usually the z-axis) as basic information for the differentiation of staircase artifacts. If the coordinate system of the target structure is modified, e.g., the surface model is being aligned with another reference structure, staircase-aware smoothing could not be applied due to the missing correlation between the staircase artifact orientation and the original image stack orientation. In such a case, the orientation information needs to be recovered. The following section describe, how context information, such as slice orientation and artifact shape properties, can be employed for an identification and reduction of staircases. As mentioned, this requires an estimation of the image stack orientation and optionally the detection of caps.



(a) Original Model **(b)** Uniform Laplacian Smoothing **(c)** Staircase-aware Laplacian Smoothing

Figure 5.4: *Staircase-aware smoothing applied to a surface model of the SCM. (a) The initial model generated via the marching cubes algorithm, colored by mean curvature; (b) Uniform Laplacian filtering; (c) Staircase-aware Laplacian filtering. (b) and (c) are colored by the Euclidean distance of each vertex to the surface of the original model shown in (a). (Images reprinted from [Mönch et al., 2011a] , © 2011, with permission from Elsevier)*

5.3.1 Estimation of the Image Stack Orientation

As prerequisites for staircase-aware smoothing, all surface normals are assumed to be consistently pointing either towards the outside or the inside of the model and the normals of neighboring faces do not suddenly point to the opposite side. Thus, faces being perfectly orthogonal to the image stack orientation (typically along the z-axis) have normals parallel to it and vice versa. This orientation information can be derived directly from the image data, since it is usually represented in the header information. If the image stack orientation is not known, e.g., because smoothing is not applied within the surface reconstruction process, or the surface model's orientation was modified for registration/alignment reasons, it needs to be recovered.

For the estimation of the image stack orientation it is assumed that the model contains several staircase artifacts. It is not necessary that the exact location of the artifacts is known. The presence of staircase artifacts, however, allows to detect and extract clusters of similar face normals and approximate the overall model orientation. Within this section, the term "normals" will also be referred to as "samples", since the normal vectors are treated as points on a unit sphere (see Fig. 5.5).

To identify these clusters, all face normals are extracted and regarded as samples in 3D space, where similar normals form clusters. For example, for a spherical surface model, the normals would result in a uniformly sampled Gauss map.

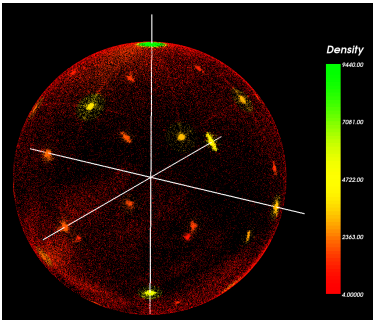


Figure 5.5: *Gauss map of the face normals of a liver model. Regions with higher density are clearly visible (colored yellow/green). Density is shown according to the number of samples within a radius r. (Image reprinted from [Mönch et al., 2011a] , © 2011, with permission from Elsevier)*

As a next step, the density of the samples is determined by counting the number of samples within a given radius r (default value is 0.1). Afterwards, a density threshold is applied to the whole point cloud (see Fig. 5.5). This threshold allows to remove samples which do not belong to one of the dominant clusters. The value is automatically adjusted according to the average of the mean and maximum density value. This value has turned out to be sufficient among all tested datasets. After reducing the point cloud to the high density clusters, labels are assigned to all samples according to their Euclidean distance to other clusters. For that, a random sample is chosen and all samples within the defined range r are iteratively considered as candidates for further processing. After labeling the current sample, the algorithm proceeds with the identified candidate samples. If no new samples for the current cluster are found, the remaining unlabeled samples are processed in the same way. This procedure is related to the DBSCAN algorithm of Ester et al. [1996] or a connected component analysis, where the connectivity is defined from the distance range r. It does not require a predefined number of clusters or further parameters. The center of each labeled point cloud region representing clustered face normals is regarded as potential image stack orientation vector.

To determine the vector with the highest probability of describing the image stack orientation, the relative orientation θ_{f_i} of all faces f_i with respect to each potential orientation vector is computed. The relative face orientation is defined as the dot product of the face normal and a given orientation vector. The resulting values are scaled to the range of [0,1] according to Equation 5.3. Thus, for faces with normals being orthogonal to the orientation vector, the relative orientation θ_{f_i} equals 1, whereas for faces with normals being parallel to the vector it equals 0.

$$\forall f_i \in F : \quad \theta_{f_i} = 1 - \left|\vec{n}_{f_i} \cdot \vec{n}_{stack}\right| \tag{5.3}$$

$\vec{n}_{f_i}$,$\vec{n}_{stack}$ - normal vector of f_i and
image stack orientation
$f_i \in F; F$ - set of faces of mesh M
$\theta \in [0,1]; \theta_{f_i}$ - orientation of face f_i

This computation is performed for each potential orientation vector to find the one maximizing the number of orthogonal faces. Hence, all faces are counted, for which θ_{f_i} is less than 0.05, which equals a tolerated deviation angle of 4.5 degree. This value was also chosen empirically. The results are quite robust with respect to this parameter. Higher values, e.g., 0.1, which equals 9 degree deviation, gave almost similar results. Finally, the vector maximizing the number of orthogonal faces is chosen as image stack orientation.

The described method for the estimation of the image stack orientation to re-establish the correlation between the surface model and the orientation of slices in the image data obviously depends on the existence of staircase artifacts. The correct vector can only be detected if the number of faces belonging to staircases exceeds a certain level, depending on the size and resolution of the model. Thus, if there are just very few and small staircases present in the model, the estimation will return a vector which does not correctly reflect the image stack orientation. This fact will be examined further in the results section.

5.3.2 Identification of Staircase Artifacts

Staircases can be characterized as surface areas that are perpendicular to each other. This information is usually not sufficient to reliably detect staircase artifacts for two reasons:

- Other (relevant) features with similar feature angles might be contained in the model which should not receive a high weighting for the smoothing algorithm.
- Depending on the initially applied mesh generation algorithm, these staircase "borders" might have already been smoothed. Thus, the features within the staircases would exhibit similar curvature values as other "natural" features.

As a result, knowledge on the (estimated) image stack orientation, slice distance and on relative changes between faces in and orthogonal to the image stack orientation is employed. Especially for data with strongly anisotropic voxel dimensions, the dihedral angles (γ_1, γ_2) at the feature edges tend to get closer to 270 and 90 degree (see Fig. 3.3). For nearly isotropic voxels, γ_1 and γ_2 approach 225 and 135 degree. Thus, staircase-aware smoothing allows to interactively adjust its sensitivity for different sizes of staircase artifacts. After computing the initial orientation rating, the vertices, belonging to staircase artifacts, are weighted to allow for subsequent usage during mesh smoothing.

At first, it is necessary to determine the relative orientation θ_{f_i} of each single face f_i (see Figs. 5.6a and 5.7) with respect to the image stack orientation. The orientation may

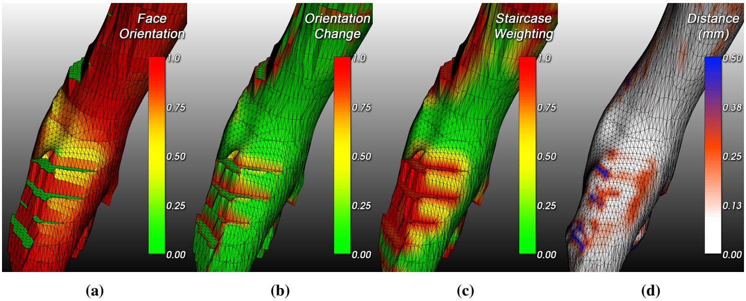


Figure 5.6: *The single steps of the staircase-aware smoothing procedure for a part of the geometric model of a carotid artery: (a) Colored by orientation of the faces in relation to the slice orientation (z-axis). (b) Coloring of the vertices where the orientation of incident faces changes. (c) Vertex weighting according to distance to staircase edges. (d) Final smoothing result colored by local Euclidean distance to the initial model. (Images reprinted from [Mönch et al., 2011a] , © 2011, with permission from Elsevier)*

have been estimated earlier (see Section 5.3.1) or a manually specified vector (usually the vector is set along the z-axis (0,0,1)) is employed for further computation. The relative face orientation is then determined as described in the previous subsection (see Eq. 5.3).

As a next step, the orientation change θ'_{v_j} at each vertex v_j is computed as the difference between the maximum and the minimum face orientation of all incident faces F_{v_j} at that vertex:

$$\forall v_j \in V: \quad \theta'_{v_j} = \max(\theta_{f_k}) - \min(\theta_{f_k}) \tag{5.4}$$

α_i - angle between normal of face f_i
and slice orientation vector
$v_j \in V; V$ - set of vertices of mesh M
$f_k \in F_{v_j}; F_{v_j} \in F; F_{v_j}$ - incident faces at v_j
$\theta, \theta' \in [0,1];$
θ'_{v_j} - orientation change of incident faces at vertex v_j

Thus, for vertices at perfect staircase edges, where the maximum difference of the incident faces would equal 90 degree, θ'_{v_j} would equal 1. For flat areas, θ'_{v_j} equals 0. The orientation change yields visually similar results as typical curvature measures (see Fig. 5.6b), as it highlights feature edges, but is related to the image stack orientation. Thus, features, which are related to the image stack orientation, get highlighted, whereas

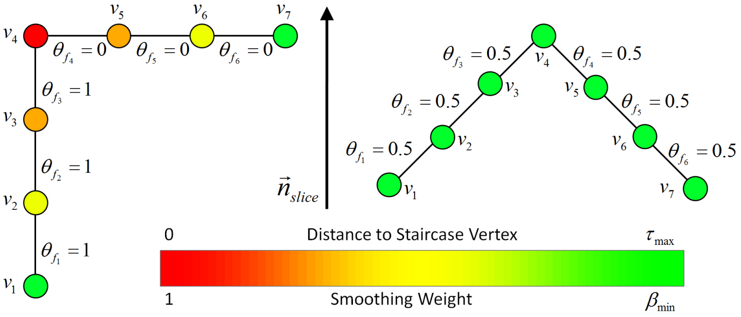


Figure 5.7: *2D illustration of a staircase (left) and a similar feature (right) that is not related to the image stack orientation ($\vec{n}_{stack}$). (Image reprinted from [Mönch et al., 2011a] , © 2011, with permission from Elsevier)*

non-artifact features, which are not caused by segmentation and a large slice distance, receive a lower weighting (compare Fig. 5.7 left and right).

5.3.3 Artifact Weighting

In the previous step, all vertices were weighted according to changes of the relative orientation which depends on the image stack orientation. To enable adaptive smoothing of the staircases, weights are defined for all vertices belonging to the artifact areas. Thus, the threshold $\tau_{\theta'}$ is employed to extract only the vertices with $\theta'_{v_j} > \tau_{\theta'}$. The value of $\tau_{\theta'} = 0.7$ was chosen empirically. A decrease of $\tau_{\theta'}$ will include smoother staircases, whereas a high value of $\tau_{\theta'}$ extracts only staircases with 90 degree feature edges. Since the type of staircases should be almost homogeneous within one surface model, the user can adjust the threshold $\tau_{\theta'}$ easily.

For each vertex v_j in the model, the Euclidean distance d_{v_j} is computed to the closest identified staircase vertex. As a special case it might happen that a non-artifact part of the surface passes a staircase corner vertex very close. This would result in misleading smoothing weights for vertices which do not belong to an artifact. To prevent this, topological connectivity within a given range $\tau_{\max}$ is required. The resulting weights w_{v_j} are again scaled according to Equation 5.5 to [0,1], where values of 1 occur at the staircase vertices and values of 0 occur in distant parts (see Fig. 5.6c). Additional thresholds give a better control of the staircase weighting. Setting $\tau_{\max}$, the user can define how far the influence of staircase smoothing reaches into the non-artifact areas. Furthermore, a minimum weight (e.g., $\beta_{min} = 0.1$) for each vertex can be applied to allow for a user-defined smoothing effect in non-artifact areas without the need to apply an additional smoothing step after staircase-aware smoothing. As another effect of β_{min}, visually disturbing borders between the smoothed staircases and areas without smoothing (e.g., suffering from noise) can be prevented. The previously computed weights are readjusted to the range above the applied minimum value:

$$\forall v_j \in V: \qquad w_{v_j} = \begin{cases} \left(1 - \frac{d_{v_j}}{\max(D)}\right) & \text{if } d_{v_j} \leq \tau_{\max}, \\ 0 & \text{if } d_{v_j} > \tau_{\max}. \end{cases} \tag{5.5}$$

$$w'_{v_j} = w_{v_j} \cdot (1 - \beta_{\min}) + \beta_{\min} \tag{5.6}$$

$\tau_{\max}$ - max. distance threshold
$\beta_{\min}$ - min. weighting offset
$d_{v_j} \in D; D$ - set of Euclidean distances
of the vertices V to V'
$V' \in V; V'$ - the extracted staircase vertices
w_{v_j}, w'_{v_j} - distance-related weights for each vertex v_j

After these steps, each vertex holds information on the (scaled) distance to the closest vertex belonging to a staircase corner. These values can be used as weights w_{v_i} (see Sec. 5.1) during the smoothing process to enable an adaptive artifact correction.

5.3.4 Detection of Caps

The described approach for staircase-aware smoothing of surface meshes identifies staircase artifacts which are related to the image stack orientation. During segmentation and model generation, different operations (manual clipping, volume-of-interest bounding box) may yield caps. For consistently handling these flat regions (include/exclude during smoothing), arbitrarily oriented caps need to be detected.

First, flat areas are identified by extracting all faces where all vertices exhibit low maximum normal angles (≤ 5 degree). The result of this operation is a set of separated flat regions (usually all staircases and other low curvature areas), which are cap candidates.

Caused by inhomogeneities in the image data, surface reconstruction yielded small, cap-like artifact structures (see Fig. 5.9, label (III); Fig. 5.8, label (II)). However, looking at their size, these structures, as well as single triangles scattered over the whole surface, which meet the first low curvature criterion, can be neglected as potential caps using a region size threshold (e.g., 10 faces minimum size). Thus, all small regions will still be subject to smoothing, whereas the larger ones will be examined further. The remaining areas are used as seeds for local region growing. For each triangle, all neighboring cells are considered for further processing if their orientation differs only slightly from the current one. This expands the initially identified areas until feature edges are found.

The basic assumption for the differentiation between caps and staircases is that the direction of the surface at the outer cap/staircase vertices ($\equiv$ vertices with a high curvature) should be constant for caps (thus pointing consistently along or contrary to the local

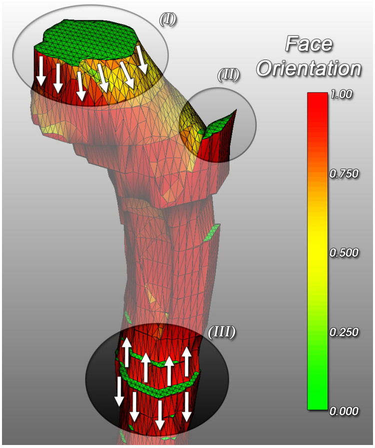


Figure 5.8: *Differentiation between staircases and caps: The latter are characterized by a consistent face direction at the outer cap vertices (I), whereas the direction changes along the vertices of staircases (III). Label (II) highlights a cap-like staircase artifact. (Image reprinted from [Mönch et al., 2011a] , © 2011, with permission from Elsevier)*

cap orientation vector; see Fig 5.8, label (I)). In contrast, for staircases, the direction of the surface at the staircase vertices will change at least twice: from positive to negative and vice versa (see Fig. 5.8, label (III)). Algorithm 1 indicates this differentiation. In each remaining region, all those vertices are examined which were identified as outer staircase vertices. For these vertices, the maximum distance to the first order neighborhood vertices is determined along the specific average cap normal (averaged of all faces belonging to the potential cap). Thus, only those neighboring vertices are considered for comparison, which do not belong to the previously identified cap candidate region. The maximum distance vertex is only needed to determine the direction of the surface: a positive sign means that the surface continues along the specific cap normal, whereas a negative sign indicates that the surface continues contrary to the orientation vector. Counting the direction changes for each candidate region, caps and staircases can be distinguished. Finally, regions where only one direction was detected at the outer staircase vertices, can be regarded as caps, whereas all other regions with higher values are staircases. As a result, the smoothing weights for the vertices belonging to detected caps are set to zero (see Fig. 5.9b, label (I) and (II)). A subsequent smoothing procedure will exclude these vertices to preserve accuracy, whereas staircase areas are reduced (see Fig. 5.9c, label (III)).

Algorithm 1 Pseudocode of the cap detection algorithm. (Reprinted from [Mönch et al., 2011a] , © 2011, with permission from Elsevier)

```
// Flat areas: set of separated low curvature areas
$FlatAreas \leftarrow Faces.getFlatAreas()$
$CapCandidates \leftarrow FlatAreas.labelConnectedComponents()$
for each $Cap \in CapCandidates$ do
    $Vertices \leftarrow Cap.getBorderVertices()$
    $DirectionChange \leftarrow$ false
    for $i \leftarrow 0$ to $Vertices.getSize()$ do
        // getDistance() returns the signed maximum distance
        // to the vertex neighbors along the orientation vector
        if $i = 0$ then
            $Orientation \leftarrow sgn(v.getDistance())$
        else if $Orientation \neq sgn(v.getDistance())$ then
            $DirectionChange \leftarrow$ true
            $break$
        end if
    end for
    if $DirectionChange =$ true then
        $Cap.delete()$
    end if
end for
```

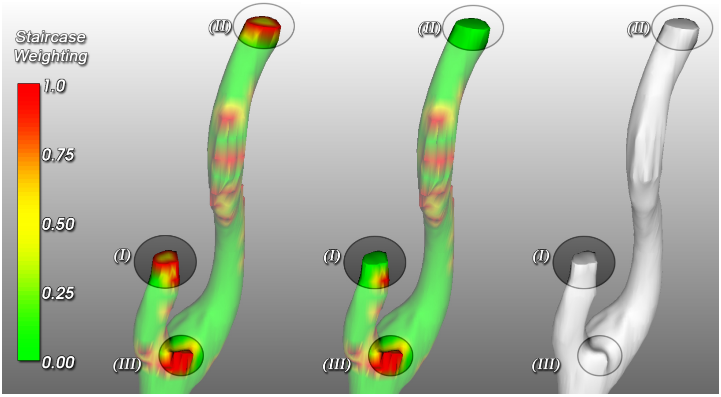
5.4 Results

The above presented methods pursue different goals. Distance-aware smoothing is designed to preserve spatial relations of neighboring structures, especially when reducing very strong artifacts from binary segmented image data. Staircase-aware smoothing aims at a reduction of small, locally restricted artifacts. The results of each specific method are described in the following sections.

5.4.1 Distance-Aware Smoothing

To evaluate the described distance-aware smoothing, a CT dataset (voxel size 0.453×0.453×3 mm) of the neck was employed and three closely located structures were picked for demonstration (see Fig. 5.1). Smoothing is applied to the models of the tumor and the jugular vein. For the tumor model, the jugular vein and the trachea serve as reference structures to which distances should be preserved. For the jugular vein, the tumor is used as reference model. For the latter, the trachea is ignored, since it is to far away from the vessel to be relevant for special consideration.

All structures were segmented manually by medical experts. The models were generated from the binary segmentation masks yielding strong staircase artifacts. The latter could also be reduced by involving, e.g., intensity data to model generation, but the problem of strong staircase artifacts still persists, since the usage of intensity data is not always feasible (e.g., due to image inhomogeneities) [Mönch et al., 2010c]. Thus,



(a) Identified Staircases **(b)** + Cap Detection **(c)** After Smoothing

Figure 5.9: *Application of cap detection for a model of the carotid artery: (a) staircase-like structures are highlighted in red; (b) caps are excluded from staircase weighting; (c) result of staircase-aware Laplacian smoothing with cap detection. Labels (I) and (II) are caps, whereas (III) is a cap-like staircase structure. (Images reprinted from [Mönch et al., 2011a] , © 2011, with permission from Elsevier)*

the surface models were generated from the binary data via MC and different sample mesh smoothing methods were applied: uniform Laplacian smoothing, Laplace+*HC*, Laplace with node position constraint (NPC) [Gibson, 1998; Bade et al., 2007a]. For the latter, cubical voxel cells with the size of the original voxels were used for each vertex. Distance-aware smoothing was combined with uniform Laplacian smoothing to demonstrate its influence to distance preservation – even for filters with serious accuracy problems. Very strong smoothing was applied to emphasize the differences. For the tumor model, 30 iterations with λ=1.0 and for the vessel model, 20 iterations were applied (for all involved methods). The additional parameters of the Laplace+*HC* filter were set to λ=0 and μ=0.5 according to the recommendations of Bade et al. [2006].

The resulting surface models were compared regarding smoothness, distance and volume preservation. For smoothness, the maximum angle between the vertex normal and the normals of all incident faces was employed (normal curvature, similar to [Goldfeather and Interrante, 2003]). This modified curvature measure has shown to be less sensitive for degenerated parts of the model (where the radius of the fitted sphere would be very close to 0 and the resulting default curvature value would thus get extremely high). Volume preservation is used to demonstrate the global error introduced by each mesh smoothing method. Distance preservation is evaluated with two measures:

- the Hausdorff distance, which is determined between the smoothed and the initial surface M_{ref} (to show changes within the model) and

Model/Method	Hausdorff Distance to M_{ref} (mm)	Euclidean Distance to Jugular Vein (mm)	Volume Change (%)	ØNormal Curvature (degree)
Original	0	0.35	0.00	14.69
Laplace	3.06	2.17	-11.09	4.02
Laplace with NPC	1.53	1.65	-5.57	9.02
Laplace+*HC*	1.16	1.07	-2.60	11.12
DA Laplace	2.51	0.39	-6.54	7.30
DA Laplace (10 mm margin)	2.41	0.35	-4.66	9.96
DA Laplace (exp. scaling)	2.47	0.36	-5.16	9.30

Table 5.1: *Results of a comparison of the smoothed tumor models (with the mentioned methods) and the initial MC reference model of the same data. To emphasize the differences between the methods, very strong smoothing with the following parameters (used for all related methods) was applied: 30 iterations, λ=1. ([Mönch et al., 2010a] © 2010 Association for Computing Machinery, Inc., Reprinted by permission)*

- the Euclidean distance, which serves to show the relation between neighboring structures.

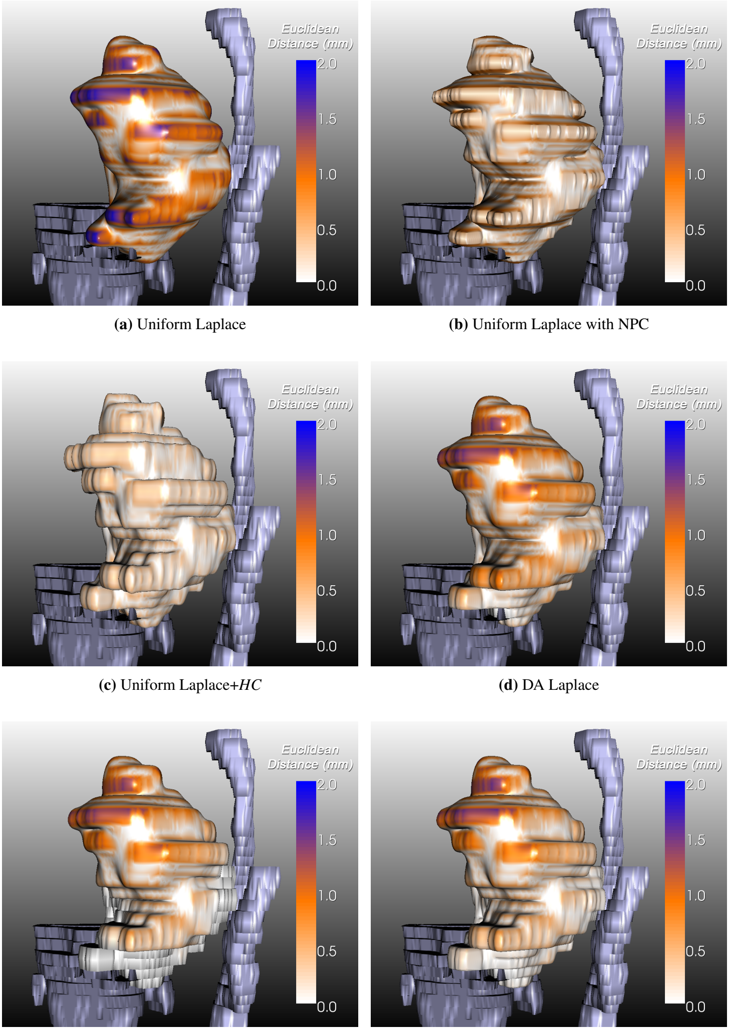
It is obvious that the presented smoothing modification can only focus on the preservation of the inter-structure distances. For smoothing with and without distance awareness, the Hausdorff distance will definitely be similar, since the smoothing weights in less relevant regions of the model will only be modified slightly.

The comparison of distance-aware smoothing to the other standard methods showed that it is possible to receive visually smooth surface models (even with strong staircase artifacts due to binary data) while preserving relevant distances to neighboring structures.

Model of the Tumor

As expected, standard Laplacian smoothing of the tumor model (Fig. 5.1) yields strong volume shrinkage and distance changes compared to the initial (unsmoothed) surface model and neighboring reference structures (see Tab. 5.1). On the other hand, the average normal curvature (and thus the staircase artifacts) could be reduced best (curvature decreased from 14.69 degree to 4.02). Other methods, which are focussed on preservation of accuracy, performed better w.r.t. distance changes (to the initial model) and volume preservation, but could not produce visually satisfying surface models (see Fig. 5.10b and 5.10c).

The distance-aware smoothing yielded worse Hausdorff distance values than Laplace+*HC* or smoothing with node position constraint. This is obvious, since these values are reached at parts of the model which were assigned higher smoothing values because of lower relevance for neighboring structures. The parts of the tumor model, which were target for distance preservation and thus received lower weighting during smoothing, could preserve the relevant distances to the reference structures (0.35 mm and 0.39 mm). Furthermore, the average normal curvature was reduced significantly (7.30 degree). Smoothness is very close to the result of standard Laplacian smoothing (compare Fig. 5.10a and 5.10d), whereas accuracy in terms of spatial relationship was



(a) Uniform Laplace

(b) Uniform Laplace with NPC

(c) Uniform Laplace+*HC*

(d) DA Laplace

(e) DA Laplace with 10 mm Safety Margin

(f) DA Laplace with Exp. Scaling

Figure 5.10: *Sample results of different smoothing methods applied to the tumor model. The surface models are colored by Euclidean distance (in mm) to the original tumor model (without smoothing).*

Model/Method	Hausdorff Distance to M_{ref} (mm)	Euclidean Distance to Tumor (mm)	Volume Change (%)	ØNormal Curvature (degree)
Original	0	0.35	0.00	19.44
Laplace	3.35	1.82	-18.62	5.43
Laplace with NPC	1.53	1.39	-8.14	11.03
Laplace+*HC*	1.02	0.97	-3.29	9.10
DA Laplace	2.42	0.35	-9.82	7.37
DA Laplace (10 mm margin)	2.42	0.35	-8.00	9.61
DA Laplace (exp. scaling)	2.40	0.35	-5.88	11.56

Table 5.2: *Results of a comparison of the smoothed models of the jugular vein (with the mentioned methods) and the initial MC reference model of the same data. To emphasize the differences between the methods, very strong smoothing with the following parameters (used for all related methods) was applied: 20 iterations, λ=1. ([Mönch et al., 2010a] © 2010 Association for Computing Machinery, Inc., Reprinted by permission)*

preserved. The usage of an additional safety margin to define parts without any smoothing gives similar results. However, the visual quality of such a model is slightly worse and the visual difference between smoothed and completely unsmoothed regions might influence visual perception. Using an exponential scaling function for the distance-aware weighting might be an appropriate tradeoff between a direct usage of the (scaled) distance values and additional safety margins (see Tab. 5.1).

Model of the Jugular Vein

The results for smoothing applied to the vessel model (see Tab. 5.2) with the tumor as spatial reference could basically confirm the results described for the tumor model. Laplacian filtering yields a very smooth surface (curvature reduction from 19.44 degree to 5.43), but introduces very strong errors to the model (3.35 mm Hausdorff distance compared to the initial model, volume shrinkage to 81.38%). Laplacian smoothing with node position constraint results in lower Hausdorff distance values, but alters the minimum distance to the tumor model and suffers from volume shrinkage. Laplace+*HC* produced lower values for the distance measures and shows only slight loss of volume. Again, the distance-weighted approaches resulted in worse values for Hausdorff distance (both 2.42 mm), but could precisely preserve the minimum distance to the reference tumor model. The values achieved for average normal curvature, showed a strong smoothness gain (7.37 degree), but applying additional safety margins to the weighting yielded slightly higher curvature values (9.61 degree). The additional application of an exponential scaling function yielded similar values for distance preservation, whereas the volume was better preserved and the normal curvature values increased slightly (compared to distance-aware smoothing with linear scaling). However, compared to the initial model, the effect of smoothing is still sufficient.

Model/Method	Hausdorff Dist. to HighRes Model (mm)		ØEuclidean Dist. to HighRes Model (mm)		Volume Change (%)		ØNormal Curvature (degree)	
	Uniform	SA	Uniform	SA	Uniform	SA	Uniform	SA
HighRes MC	0		0		0.00		13.55	
Default MC	3.14		0.20		-1.05		26.03	
Laplace	5.17	4.09	0.29	0.22	-7.05	-4.57	6.46	9.72
Laplace+*HC*	3.08	3.08	0.18	0.18	-1.71	-1.56	12.68	16.15
Laplace with NPC	2.94	2.95	0.23	0.21	-4.44	-3.68	11.04	11.59
Taubin's $\lambda\|\mu$	3.15	3.08	0.18	0.18	-1.69	-1.59	12.14	18.00
Mean Curvature Flow	4.43	3.45	0.23	0.20	-4.91	-3.45	8.07	12.00

Table 5.3: *Results for the comparison of smoothing methods for the data of the SCM: Each smoothing method was combined with staircase-aware smoothing (SA). NPC stands for "node position constraint". The ØNormal Curvature is slightly higher for the SA approaches, since major parts of the original surface remain unchanged. (Table reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

5.4.2 Staircase-Aware Smoothing

Staircase-aware mesh smoothing is usually applied within a pipeline, which may have introduced several inaccuracies (e.g., image noise, inhomogeneities, segmentation errors). Thus, the goal of staircase-aware smoothing is to strictly avoid additional errors in non-artifact areas and simultaneously make obviously erroneous areas (staircase artifacts) look more plausible. An evaluation and the related results for the components are described in Section 5.3. Specific results for the usage of these methods in relation to CFD are shown in the separate Section 5.5.

Data

Two exemplary anatomic structures (jugular vein, SCM) from CT neck data (voxel size is 0.351×0.351×3.956 mm for both structures) were employed for evaluation. These structures contain non-artifact areas as well as parts suffering from staircase artifacts.

The surface meshes were extracted using a typical workflow. They were segmented semi-automatically by medical experts and the resulting binary masks were dilated (kernel: 3×3×3 voxel). Afterwards, this contour information was used to mask the intensity data to exclude neighboring structures with similar intensity values. Due to the dilation operation, the considered volume was slightly increased which allows to account for image inhomogeneities, such that the subsequent MC-based iso-surface extraction could include more voxels along the structures' contour. This did still result in staircase artifacts, which tend to be large because of the strongly anisotropic voxel dimensions. For better evaluation of distance changes during mesh smoothing, the masked intensity data were resampled to isotropic voxel dimensions (0.351×0.351×0.351 mm) using cubic B-spline interpolation. The default MC mesh (obtained from the anisotropic image data) and the high resolution MC mesh (from resampled data) serve both as reference models for evaluation of distance, volume, and curvature changes.

Furthermore, a software phantom was employed to evaluate staircase-aware smoothing in a context where the ground truth is known. First, image data containing a sphere was generated, where parts of the sphere structure were deleted in the image data in order

Model/Method	Hausdorff Dist. to HighRes Model (mm)		ØEuclidean Dist. to HighRes Model (mm)		Volume Change (%)		ØNormal Curvature (degree)	
	Uniform	SA	Uniform	SA	Uniform	SA	Uniform	SA
HighRes MC	0		0		0.00		12.83	
Default MC	1.40		0.22		+3.07		25.74	
Laplace	2.63	1.40	0.24	0.21	-3.53	+1.08	7.31	12.12
Laplace+*HC*	1.60	1.54	0.22	0.22	+3.68	+3.48	13.08	17.64
Laplace with NPC	1.80	1.40	0.23	0.21	-1.40	+1.87	10.38	13.18
Taubin's $\lambda\|\mu$	1.50	1.45	0.22	0.22	+3.78	+3.29	15.01	22.26
Mean Curvature Flow	1.80	1.44	0.21	0.20	-0.22	+1.62	8.92	13.05

Table 5.4: *Results for the comparison of smoothing methods for the data of the jugular vein: Each smoothing method was combined with staircase-aware smoothing (SA). NPC stands for "node position constraint". (Table reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

to induce staircases (see Fig. 5.12b). The image data has a resolution of 128×128×43 voxels. Anisotropic voxel dimensions of 1×1×3 mm were chosen to keep the data processing workflow similar to the pipeline used for medical data. Afterwards, Gaussian smoothing was applied to the image data to allow for the reconstruction of a smooth sphere surface model using MC. As for the previous structures, the data was resampled to isotropic voxel dimensions (0.3×0.3×0.3 mm) to achieve an accurate high resolution surface model as reference. For all structures, staircase-aware and uniform smoothing approaches were compared:

- Laplacian smoothing (with and without node position constraint (NPC)),
- Laplace+*HC*,
- Taubin's $\lambda|\mu$,
- Mean Curvature Flow (Laplace with cotangential weights).

For Laplacian smoothing with NPC, cubical voxel cells were defined with the original voxel dimensions for each vertex, whereas the displacement of the vertices during smoothing was restricted to these cells. Since standard Laplacian smoothing may yield strong tangential shifts, the Mean Curvature Flow method with cotangent weights was included. Staircase-aware smoothing was applied to all of these smoothing methods to allow for a direct comparison. For staircase identification, $\tau_{\theta'}$ was set to 0.7 and $\tau_{\max}$ to 3mm. In non-artifact areas, no smoothing was carried out ($\beta_{min} = 0$) . For the vessel data, 20 iterations and λ=0.5 were used in combination with cap detection. For the muscle data, λ was set to 1 to account for the large staircases. The parameters allow for a sufficient reduction of staircase artifacts for all methods. The additional parameters of the Laplace+*HC* filter were set to $\lambda = 0$ and $\mu = 0.5$ according to Bade et al. [2006]. For Taubin's $\lambda|\mu$ filter, μ equals 0.52 for the vessel data and 1.02 for the muscle data. The resulting surface models were compared regarding smoothness, shape and volume preservation. The preservation of shape is again evaluated via the two-sided Hausdorff distance and the average Euclidean distance between the smoothed and the high resolution reference model.

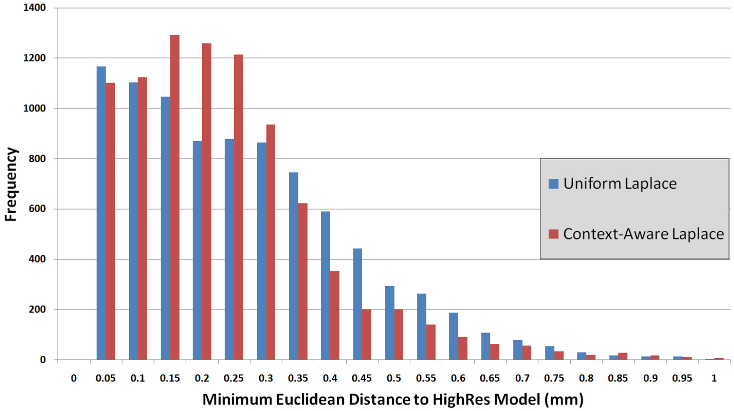


Figure 5.11: *Changes of Euclidean distance of uniform and staircase-aware Laplacian smoothing of the model of the jugular vein. Staircase-aware smoothing reduces the number of larger distance changes. (Image reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

Evaluation

The comparison of the employed methods reveals that staircase awareness could successfully restrict the smoothing process to the artifact areas. In combination with staircase-aware smoothing, the accuracy of all involved uniform smoothing methods could be increased.

As expected, standard Laplacian smoothing yielded strongest volume shrinkage for both anatomic structures (SCM:-7.05%; j. vein: -3.53%; see Tab. 5.3 and 5.4) as well as for the phantom data (-4.02%, see Tab. 5.5). This error could clearly be decreased by staircase-aware smoothing (SCM:-4.57%; j. vein: +1.08%; phantom: -0.48). The volume changes for the more volume- and feature-preserving methods, such as Laplace+*HC*, Laplace with NPC, Taubin's $\lambda|\mu$ filter, and Mean Curvature Flow, are smaller. The individual results could generally be improved. For the vessel model (see Tab. 5.4), most methods yield a volume increase which results from a large concavely shaped area. The sphere phantom data shows the strongest differences between the uniform and staircase-aware methods.

The error in terms of distance changes (Hausdorff distance, average Euclidean distance) could also be reduced by staircase-aware smoothing for the Laplace, Taubin's $\lambda|\mu$ filter, and Mean Curvature Flow for all tested models. The combination with Laplace+*HC* as well as Laplace with NPC did not yield relevant changes for the Hausdorff distance. Slight improvements for the average Euclidean distance were achieved compared to the high-resolution model for Laplace with and without NPC. Figure 5.11 shows for Laplacian smoothing that staircase-aware smoothing (red bars) is able to decrease the number

Model/Method	Hausdorff Dist. to HighRes Model (mm)		ØEuclidean Dist. to HighRes Model (mm)		Volume Change (%)		ØNormal Curvature (degree)	
	Uniform	SA	Uniform	SA	Uniform	SA	Uniform	SA
HighRes MC	0		0		0.00		0.44	
Default MC	2.73		0.04		-0.01		1.72	
Laplace	1.96	1.51	0.57	0.06	-4.02	-0.48	1.37	1.49
Laplace+HC	2.08	2.09	0.03	0.03	-0.02	0.00	1.61	1.67
Laplace with NPC	1.51	1.51	0.42	0.04	-2.91	-0.04	2.45	1.47
Taubin's $\lambda\|\mu$	2.20	2.21	0.03	0.03	-0.01	0.00	1.70	1.71
Mean Curvature Flow	1.64	1.61	0.29	0.04	-2.17	-0.02	2.46	1.475

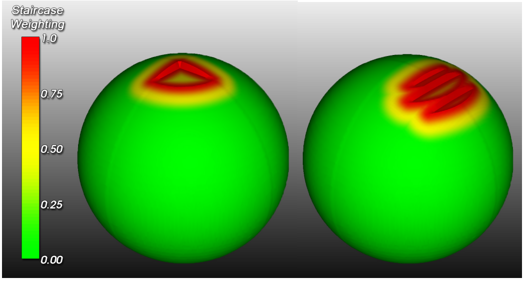
Table 5.5: *Results for the comparison of smoothing methods for the sphere phantom data: Each smoothing method was combined with staircase-aware smoothing (SA). NPC stands for "node position constraint". (Table reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

of large distance changes compared to the uniform method (blue bars). These differences are obviously smaller for more restrictive smoothing methods (e.g., Laplace+HC, Laplace with NPC, Taubin's $\lambda|\mu$ filter).

The smoothness in terms of average normal curvature is always slightly higher for the staircase-aware approaches. This is obvious, since major parts of the surface models were less smoothed. Since the high resolution mesh is smoother by definition (due to the interpolated image data), the default MC mesh was included as reference for smoothness. Staircase-aware smoothing could reduce the curvature in all cases, but represents a tradeoff between accuracy and smoothness. Thus, the uniform smoothing methods yielded smoother surfaces. The average normal curvature for staircase-aware smoothing can be reduced by setting β_{min} to 0.1 or 0.2 for a restrained smoothing effect in non-artifact areas.

According to Tab. 5.3, the sequence of the five methods in terms of accuracy (especially Hausdorff distance) remains the same for the uniform and the staircase-aware approach. For the vessel data (see Tag. 5.4), the sequence is modified. Staircase-aware smoothing improves the accuracy of uniform Laplacian smoothing to be similar and even slightly better than the more restrictive methods.

The results have shown that staircase-aware smoothing could at least keep the accuracy of the initial models and, in almost all cases, preserve the volume better than the employed uniform smoothing methods. Furthermore, it does not introduce essential additional effort to the model generation pipeline. Identification and weighting of staircases took less than one second for all employed surface models. The default parameters did not have to be adjusted. The parameters yield, however, stable results for small changes. Thus, a time-consuming parameter tuning is not necessary.



(a) **(b)**

Figure 5.12: *Samples of the phantom data employed for evaluation of orientation estimation and staircase identification: (a) was derived from phantom image data, where a quarter of the topmost slice of the sphere was removed. (b) is an example for manually introduced staircases to evaluate accuracy. (Images reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

Rotation Axis	Rotation Angle	Carotid Artery	SCM	Liver
	0°	0.830	0.479	0.000
X	30° 60° 90°	0.828 0.830 0.830	0.484 0.483 0.479	0.046 0.026 0.000
Y	30° 60° 90°	0.830 0.831 0.830	0.480 0.482 0.479	0.032 0.034 0.000
Average	-	0.830	0.481	0.021

Table 5.6: *Results for the estimation of the image stack orientation applied to three differently shaped structures: For the vessel (carotid artery), the SCM and the liver, the deviation angles towards the known orientation vector are presented (in degree). (Table reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

Estimation of the Image Stack Orientation

Orientation estimation was applied to three differently shaped MC surface models of the carotid artery, the SCM, and the liver, which were all extracted with the workflow described in Section 5.4.2. The initial models were rotated around the x-axis and the y-axis separately up to 90 degree in 30 degree steps. Finally, the accuracy of the estimated orientation vector was compared to the one known by the applied rotation. The radius r (see Sec. 5.3.1) was set to 0.1 for all employed datasets. Even small changes (e.g., 0.05 or 0.2) did not yield significant changes.

The results of the experiments are shown in Table 5.6. The average estimation error (deviation from the known orientation vector) of all structures is 0.444 degree. For the individual structures containing large staircase areas, but also non-artifact parts, slight

Amount Removed	% Flat Faces	Deviation Angle
1/16	0.17	10.284
1/8	0.40	1.629
1/4	0.78	1.621
1/2	1.63	1.174
Complete Slice	3.09	0.000

Table 5.7: *Results for the comparison of phantom sphere data with differently sized staircase-like artifacts: In the topmost sphere slice, differently sized parts were removed. The percentage of flat faces is the number of faces oriented along the known image stack orientation in relation to the total number of faces in the model. (Table reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

differences were measured: 0.830 (carotid artery), 0.481 (SCM), and 0.021 (liver) degree.

The accuracy of the orientation estimation depends strongly on the presence and size of flat areas related to the image stack orientation, as contained in staircases and caps. Thus, for models with very few staircase artifacts, the estimation might be less accurate. To evaluate this aspect, phantom image data of a sphere was employed, where parts of the topmost image data slice were cut out with increasing size to generate a staircase-like flat area. Five different “artifact sizes” were tested: 1/16th, 1/8th, 1/4th, 1/2, and a complete removal of the topmost sphere slice (see Fig. 5.12a). All of these cuts were applied in the axial images. Thus, the target orientation, which the approach should detect, is a vector along the z-axis, e.g., (0,0,1).

Subsequently, the size of the resulting flat areas was measured in terms of the percentage of each model’s surface (see Tab. 5.7). For each of the models, the orientation was estimated. Less than 0.40% of the faces in these phantom models were sufficient to detect the image stack orientation with an acceptable accuracy. The estimation failed for the model with only 0.17% of the faces being oriented along the image stack orientation. For the latter, not only faces orthogonal to the orientation vector were generated. Especially at the borders between the original sphere surface and the cut parts, a lot of diagonal faces emerged. Thus, an accurate detection of the image stack orientation was not possible.

Detection of Caps

To demonstrate the capabilities of the cap detection method, phantom data in terms of a cylinder merged with a cuboid was generated. Thus, the resulting structure contains perfect caps due to the cylinder and staircases due to the cuboid (see Fig. 5.13a and 5.13b). The surface model was again generated via MC from artificially created image data. The additional step via image data was taken to keep the mesh properties close to the results with the pipeline used for medical structures. Subsequently, uniform Laplacian mesh smoothing (up to 50 iterations with $\lambda = 0.5$) was applied to the phantom surface model to generate variations with increasing smoothness and thus less feature edges. The goal of this procedure was to evaluate if the cap detection method is still able to differentiate between the cap-like parts of the model and the staircases. Sample results are shown in Figure 5.13. Figure 5.13c (50 iterations) demonstrates that the caps

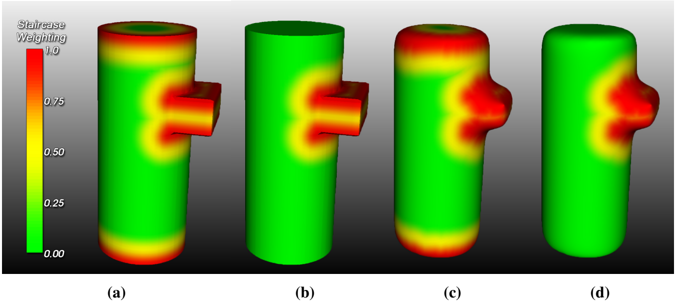


Figure 5.13: *Phantom data used to evaluate cap detection for differently smoothed caps: (a) and (b) depict the initial surface model after default staircase weighting (a) and with cap detection enabled (b). The models in (c) and (d) show the same procedure, but for smoothed models (Laplace, 50 iterations, $\lambda = 0.5$) without sharp feature edges. (Images reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

and staircases were identified correctly and could be excluded from the list of potential staircases (see Fig. 5.13d).

The detection of caps and further exclusion from staircase weighting requires that the cap was initially detected as staircase. Since the feature edges at the caps and staircases exhibit very low curvature, $\tau_{\theta'}$ was decreased to 0.2 to detect these parts as staircases. This low value for $\tau_{\theta'}$, however, was never required for clinical data where potential caps exhibit sharper feature edges. Finally, the caps could be detected for all created phantom models.

5.4.3 Computation Time

For the described methods, the computation time is only important at second glance. Distance-aware smoothing can be applied without noticeable delays. The required computation of minimum distances per vertex has to be performed only once after model initialization (loading from a file or mesh generation). Scaling of the distance values can be performed during modification of the smoothing weights (see Sec. 5.1) and does thus not yield relevant computational effort.

Staircase-aware smoothing is more complex, since it analyzes and filters the mesh vertices according to user-defined parameters. Any change of the parameter $\tau_{\theta'}$ and $\tau_{\max}$ requires an anew filtering of the mesh vertices and recomputation of the staircase weighting. The default MC model of the jugular vein (see Tab. 5.4) consists of 8,783 vertices, the SCM (see Tab. 5.3) of 37,522 vertices. For the vein, staircase identification took up to 130 ms. The SCM required up to 500 ms. For larger surface models, such as the liver (94,343 vertices) employed in [Mönch et al., 2010b], computation took about 1,500 ms. Modifications of $\tau_{\theta'}$ and $\tau_{\max}$ yielded a slight decrease of these computation times. As a result, parameter modifications become visible after very short delays and can be utilized for smoothing.

The measurements were conducted using an older CPU (Intel Core 2 Quad Q9400 @ 2.66 GHz) utilizing only one core. Further parallelization of the code, e.g., by using all CPU cores or even the GPU, might push these computation times into the real-time direction.

5.5 Application of Staircase-Aware Smoothing to CFD

The patient-specific simulation of blood flow, e.g., in cerebral aneurysms, is an application where accuracy and smoothness are especially relevant. The simulation results in terms of flow velocity and wall shear stress may strongly depend on details of the patient-specific geometry and corresponding mesh smoothing operations. Thus, the careful reduction of segmentation artifacts, such as merged vascular branches due to image inhomogeneities, is often a tedious, time-consuming manual procedure.

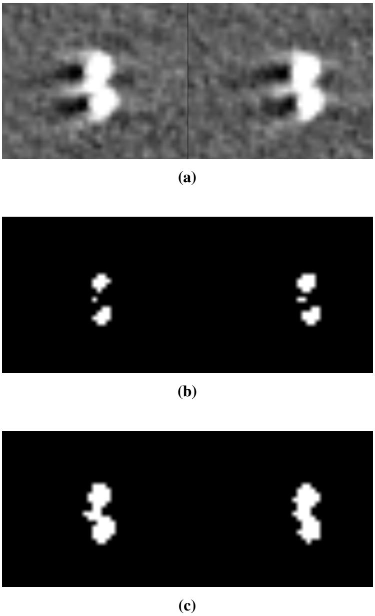
Applying staircase-aware smoothing to models for CFD, two questions shall be investigated:

(I) Can staircase-aware smoothing reduce the effort for manual correction of local artifacts?
(II) What are the differences between the employed smoothing methods with respect to the blood flow behaviour?

This section describes the employed data and the specific model generation pipeline. Finally, the results of the blood flow simulation are discussed for different smoothing methods with and without staircase awareness.

5.5.1 Data and Workflow

The steps of the CFD model generation pipeline are shown in Figure 5.15. Clinical or phantom image data for blood flow simulation can be derived from different modalities (e.g., magnetic resonance angiography (MRA), computed tomography angiography (CTA), rotation angiography, time-of-flight MR angiography). For evaluation of staircase-aware smoothing, CTA data of a phantom aneurysm model (voxel size is 0.4×0.4×0.4 mm) was employed (see Fig. 5.14a). Due to the high contrast in the angiography data, vascular structures can be delineated easily by an intensity threshold. Afterwards, a connected component analysis was applied to extract the aneurysm and its parent vessel (inlets, outlets) from background noise. Manual effort was necessary to define a ROI around the aneurysm to exclude bone structures and distant vessels. The resulting segmentation information was used to mask the initial intensity data. As in many datasets, image inhomogeneities yield wrong contour information. In the employed phantom dataset, beam hardening artifacts at the two aneurysm outlets lead to blending effects during segmentation (see Fig. 5.14) and in the subsequent MC model generation step. Additional effort was necessary to correct the segmentation mask by



(a)

(b)

(c)

Figure 5.14: *Two subsequent slices of the CTA data. (a) shows a close-up of the intensity data. (b) and (c) depict the application of two different thresholds to the same image slices.*

manually drawing contours in the image slices of the affected vessel regions. Unfortunately, this procedure leads to staircase-like artifacts in the surface model. Thus, a final smoothing of the surface mesh is necessary to remove surface noise, but especially staircase artifacts from image segmentation. A typical solution is the manual correction of local artifacts, e.g., employing tools such as *Sculptris* (see Sec. 4.2.1) for dynamic mesh tessellation, surface modeling and smoothing. Such a manual procedure is, however, time-consuming and error-prone (see Chap. 4). In contrast, automatic mesh smoothing might not guarantee the required accuracy and smoothness simultaneously. Thus, adaptive mesh smoothing might allow for accurate and smooth surface models applicable to CFD. As a result of mesh smoothing, the homogeneity of the triangle size and quality is usually slightly improved. After the smoothing step, the inlets and outlets are cut orthogonal to the local vessel centerline to define valid inflow and outflow regions for the CFD simulation. Finally, the reconstructed surface mesh was optimized with respect to the mesh quality by employing an advancing front remeshing algorithm [Schöberl, 1997]. The resulting mesh was subsequently used for volume grid generation, which served for the CFD simulation.

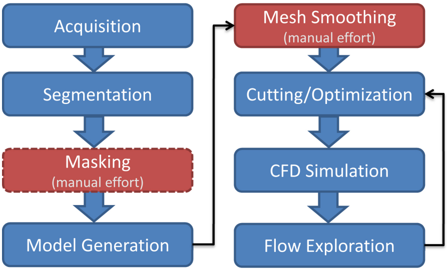


Figure 5.15: *Overview about the CFD data flow pipeline from image acquisition to a meaningful CFD simulation model. In some cases, a masking step with manual effort is necessary to resolve blending artifacts which lead to staircase-like artifacts. Hence, an adaptive smoothing approach is necessary to resolve these artifacts. (Image reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

Numerical Computations

The numerical computations are performed in parallel using up to six computing cores applying the commercial CFD solver ANSYS Fluent 12. Blood rheology is represented using a Newtonian description with constant density and viscosity, where the blood density is chosen as 1000 $\mathrm{kg/m^3}$ and the dynamic viscosity as $4{\cdot}10^{-3}$ Pa·s. In the present case, a steady flow condition is finally retained with an inlet flow rate of 1.2 $\mathrm{cm^3/s}$ at the inlet. All vascular walls are assumed to be rigid, as in most published studies, since real wall material properties are unknown. A standard, no-slip boundary condition is employed at all contact points with surfaces. At the outlets, traction-free boundary conditions are applied.

Influence of Staircase-Aware Smoothing

To evaluate the influence of staircase-aware mesh smoothing, Laplace and Laplace+*HC* filtering was applied to the initial MC mesh. Each method was also combined with staircase-aware smoothing. Taubin's $\lambda|\mu$ smoothing and Laplace with NPC were omitted here, since they usually yield intermediate smoothing results compared to Laplace, which provokes strongest errors, and Laplace+*HC*, which is very restrictive. Furthermore, a manually smoothed and adjusted model was employed as reference model. An overview about the models with their quantitative differences compared to the manually smoothed model is given in Table 5.8. This surface was considered as reference model because of the best trade-off between adaptive smoothness and volume preservation to the original MC model. It is assumed to fit best to the original vessel surface and the resulting CFD results are more valid in contrast to the other models. Note that the CFD results were compared under the certain boundary conditions. Predictions about the validation of simulated blood flow to the real flow behavior are still a research task [Boussel et al., 2009]. The numerical results of the CFD simulation, namely velocity and wall shear stress (WSS), are listed in Table 5.8.

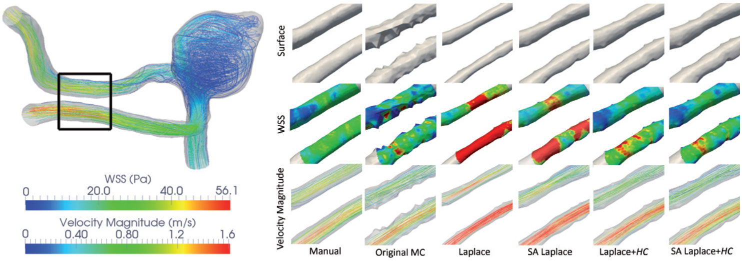


Figure 5.16: *Comparison of the CFD results for the employed smoothing methods. The manually smoothed reference model is shown in the upper left corner with color-coded streamlines. A region of interest is defined (black rectangle) around the artifact area and is enlarged for the six models on the right side. The maximum of the color scales is related to the reference model. (Images reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

5.5.2 Results and Discussion

In the upper left corner of Figure 5.16, the reference model of the reconstructed aneurysm surface with its inlet and two outlets is shown. Blood flow and velocity magnitude is visualized with color-coded streamlines. A ROI around the staircase-like artifact (existent in the original MC model) is marked and enlarged for each surface model next to the aneurysm (first row). Additionally, the corresponding WSS (second row) and velocity magnitude (third row) of that ROI are presented. Laplacian smoothing caused the strongest volume change (-18.69%) resulting in an increased velocity magnitude and WSS because of the small vessel diameters. Staircase-aware smoothing could reduce volume shrinkage to -4.53%, which comes along with a strong reduction of WSS. Comparing uniform and staircase-aware Laplacian smoothing to the manually smoothed reference model shows still strong differences. Uniform Laplace+*HC* filtering could again reduce volume shrinkage and distance changes resulting in less errors for velocity and WSS. Adding staircase awareness to Laplace+*HC* filtering could improve the results and get closer to the reference model.

The results demonstrate clearly that mesh smoothing strongly influences the results of blood flow simulations. Staircase-aware smoothing could clearly improve the results of the employed uniform smoothing methods. Laplace+*HC* smoothing combined with staircase awareness achieved the smallest error compared to manual smoothing. Keeping in mind that the manually smoothed model might also not perfectly describe the real flow behavior, staircase-aware Laplace+*HC* smoothing might be a promising alternative to the time-consuming manual artifact correction procedure.

Model	Hausdorff Dist. (mm)	ØEuclid. Dist. (mm)	Volume Change(%)	ØNormal Curv. (deg.)	Max. Vel. (m/s)	Max. WSS ROI (Pa)	Difference Vel. / WSS
Manual	0	0	0	6.224	1.611	56.135	0 / 0
Original MC	0.497	0.083	-1.05	6.528	1.696	142.824	0.085 / 86.689
Laplace	0.649	0.253	-18.69	6.239	4.281	345.573	2.670 / 289.438
SA Laplace	0.511	0.085	-4.53	6.372	2.591	164.901	0.980 / 108.766
Laplace+*HC*	0.499	0.078	-3.0	6.382	1.941	103.923	0.330 / 47.788
SA Laplace+*HC*	0.401	0.07	-1.49	6.32	1.717	94.037	0.106 / 37.902

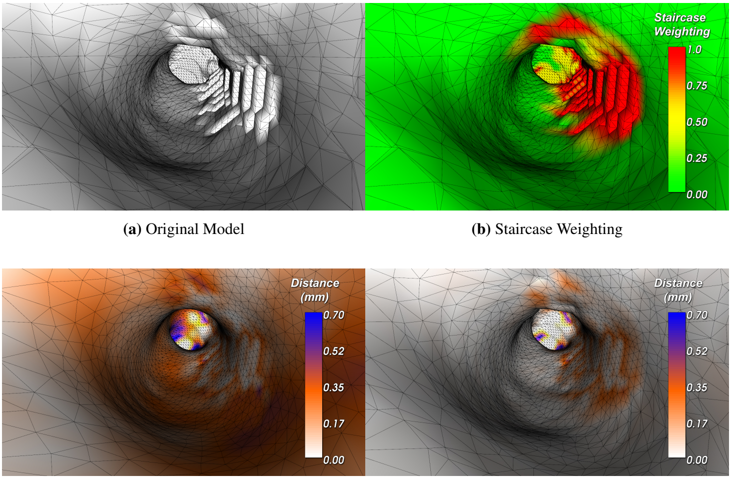
Table 5.8: *Overview of the six smoothing variations of the aneurysm model on which a CFD simulation is performed. In all cases, staircase-aware smoothing results in better volume preservation compared to standard Laplace or Laplace+HC. Distance differences and volume change are related to the manually smoothed model. (Table reprinted from [Mönch et al., 2011a], © 2011, with permission from Elsevier)*

5.6 Summary

Context-aware smoothing can be employed to reduce staircase-artifacts in anatomic surface models. It extends common mesh smoothing filters and improves their accuracy by making use of context information, such as application requirements. Moreover, the methods described in this chapter are not restricted to surfaces from anatomic structures. Context-aware smoothing aims at staircase artifacts in surfaces with natural, organic shapes (in contrast to artificial/mechanical shapes).

Distance awareness is useful for preserving spatial relations but with smoothing of large parts of the model. This is relevant to ensure a correct computation and visualization of safety margins and potential infiltrations, as required for the planning of surgical treatment based on segmented anatomic structures. An inhomogeneous surface, where artifacts may remain in individual regions, is, however, not suited for other applications. Smoothness shall rather be consistent over the surface due to visual, experimental, and computational demands (e.g., rapid prototyping, CFD).

In contrast, staircase awareness serves for reducing staircase artifacts, whereas accuracy and features are preserved in artifact-free areas. Comparing the distance and volume changes to high resolution reference models instead of the default MC models has shown improvements for all employed anatomic structures as well as for phantom data. The changes of Hausdorff distance and average Euclidean distance lie within the submillimeter/subvoxel range. This is related to the size of the artifacts and voxel size. The visual results and the application of staircase-aware smoothing to CFD demonstrate that staircase-aware smoothing is able to preserve non-artifact areas better than uniform smoothing approaches. For surface models where only small parts suffer from staircase artifacts, the strongest visual and quantitative gain is achieved. Hence, a consistently smooth surface is achieved, which is suited for all described applications (see Chap. 2). Figure 5.17 shows an example of an endoscopic view inside of a carotid artery. The usage of staircase-aware smoothing preserves the original surface and reduces the artifacts at the same time – even in combination with a standard Laplacian smoothing filter, which itself would yield very strong inaccuracies (see Fig. 5.17c). The detection of arbitrarily oriented caps extends staircase-aware smoothing by additional context information. It accounts for further requirements, e.g., for a special treatment of these



(a) Original Model

(b) Staircase Weighting

(c) Uniform Laplacian Smoothing

(d) Staircase-aware Laplacian Smoothing

Figure 5.17: *Endoscopic view of a carotid artery. (a) The initial surface with artifacts; (b) Vertex weights according to detected staircases; The results of uniform (c) and staircase-aware (d) Laplacian smoothing colored by the Euclidean distance of each vertex to the surface of the original model shown in (a).*

flat areas for visualization or for preserving in-/outlets in surfaces prepared for CFD simulation.

If surface areas containing staircases represent the closest parts of two neighboring structures, staircase-aware smoothing leads to changes of these spatial relations. As a remedy, distance awareness could additionally be applied by averaging or multiplying the weights from both methods.

Context-aware smoothing allows to apply different smoothing filters with less attention, since:

1. artifacts can systematically be reduced and
2. spatial relations are preserved.

Thus, it brings more robustness into the handling of smoothing filters. However, it also adds further parameters to the existing smoothing parameters. For distance-aware smoothing, no additional parameters other than, e.g., a margin, have to be specified. The only exception is if one wants to apply an arbitrary multi-dimensional function, which requires several parameters itself. Staircase awareness is basically configured by three parameters: $\tau_{\theta'}$ to filter depending on the sharpness of staircases, $\tau_{\max}$ to define the spatial expansion of the staircase weighting, and $\beta_{\min}$ as global minimum weight. The suggested default value for $\tau_{\theta'}$ yielded good results for all tested data. $\tau_{\max}$ accounts for the size of staircases and should thus be set according to the slice distance of the source image data. $\beta_{\min}$ does not influence the staircase detection algorithm and can thus be set according to application- or user-specific requirements. Thus, the configuration of these parameters should not be a too complex and time-consuming task – even if the user wants to explore the parameter influences iteratively. For the latter, an efficient hardware-based implementation would be beneficial.

Besides smoothing, the distance- and staircase-dependent weights might also be applied to other algorithms in the model generation pipeline. As an example, decimation could be modified according to the inter-structure distances and simplify those parts of the model which are less relevant for the interpretation and measurement of spatial relations. Similarly, decimation could primarily be performed in staircase areas, since these areas already have to be considered as inaccurate.

6

Visually Guided Mesh Smoothing for Medical Applications

Table of Contents

Mesh smoothing is usually only one part of a complex model generation pipeline. Depending on the specific use case, such pipelines may be static, which means that all parameters are fixed and mesh postprocessing, such as smoothing, is performed without further changes to earlier defined parameter values. In such static pipelines, the computation time of smoothing filters is usually not considered as relevant.

In contrast, the usage of fixed default parameters interferes with any accuracy requirements. As confirmed by Bade et al. [2006] (see Sec. 3.3.4), minimizing artifacts with concurrent preservation of the individual shape and volume requires careful testing of different smoothing methods and parameters. Considering the described applications (see Chap. 2), the workflows are not static. Users dealing with medical surface model generation may want to compare different smoothing filters and their parameters to find an appropriate smoothness-accuracy tradeoff for patient-specific structures. Within such interactive workflows, even small delays in the range of a few seconds may disturb if they appear repeatedly. Delays occur, if parameters are modified and the visualization has to be updated after performing smoothing anew. Moreover, visual feedback does not only comprise the rendering of the smoothed surface model, but also hints on the achieved curvature reduction or current accuracy in terms of color coding or quantitative measures. Such an additional quantitative analysis for each intermediate smoothing result yields further delays.

The handling of mesh smoothing filters in typical medical model generation workflows can be summarized as follows:

- Smoothing parameters are adjusted several times and smoothing has to be performed repeatedly.
- The quality of the intermediate smoothing results needs to be evaluated for proper visual feedback. This is most often performed using additional tools and not directly embedded into smoothing workflows.
- Any delays during manual parameter adjustments and related computations (smoothing, estimation of smoothness and accuracy) may disturb the interactive model generation.

Because of these reasons, it is desirable that mesh smoothing and its effects on smoothness and accuracy become more obvious and easier to handle to achieve a fast and flexible integration into different model generation procedures.

This chapter is based on the work described in [Mönch et al., 2012, 2013b] and describes a mesh smoothing framework, which reduces the computing delays and provides an interface for decreasing interaction delays. The framework employs the idea of using mouse movements to control relevant smoothing parameters (weights, number of iterations). Visual feedback on accuracy and curvature reduction is given by determining and mapping these measures onto the surface in real-time. Moreover, smoothness and accuracy are precomputed for different parameter sets to provide a fast overview of the expected model quality. *Model quality graphs* and *model quality bars* have been introduced as an effective means to perform parameter adjustments in mesh processing. This allows for examining the sensitivity of the input parameters and to make suggestions for optimized parameter sets. Automatically generated parameter suggestions are then used to provide differently smoothed models, from which the user may select one without having to modify any parameters. Moreover, the model-dependent suggestions serve as preview of local artifact reduction effects during interactive model exploration.

The chapter is structured as follows. First, options for parallelizing Laplacian mesh smoothing filters are discussed. This comprises the technical details, such as the data structure and workflow, which is then used for GPU-based mesh smoothing. Afterwards, the achieved performance, the interaction concept, and visual feedback (Sec. 6.4) are described. The computations performed for error and curvature approximation can then be used to enable a visual and (automated) quantitative sensitivity analysis for the required smoothing parameters (Sec. 6.5).

6.1 Parallelization Techniques

For parallelizing algorithms, different concepts need to be considered. One wide-spread technique to speed up an algorithm running on CPU is OpenMP [Dagum and Menon, 1998; Chandra et al., 2001]. OpenMP can be utilized by adding a few commands to the source code, which set the number of threads and the parallel regions. The OpenMP compiler directives as well as its runtime can be found for all major platforms and compilers. Since most of today's computers provide a multi-core processor (usually with two or four cores), computations, such as smoothing operations, can easily be sped up. However, the overhead created by initializing and merging the threads needs to be considered. Another option for improving the performance of multi-core CPU applications is SSE (Streaming SIMD Extensions) [Thakkur and Huff, 1999], which provides a set of new instructions allowing to process data in parallel. The availability of SSE instructions depends on the employed CPU, but with current CPUs even newer extensions, such as SSE2 to SSE5, should be available.

Besides the abilities of modern multi-core CPUs, GPUs provide high computational power and massive parallelization. The usage of GPUs for non-rendering tasks can be performed using different programming interfaces (e.g., OpenGL [Segal and Akeley, 2012], OpenCL, CUDA). The OpenGL Shading Language (GLSL) [Kessenich et al., 2010], which is officially part of the OpenGL specification since version 2.0, provides shaders which can also be used to do arbitrary computations. OpenGL is, however, foremost designed for graphical operations, which may complicate the handling for general purpose computing. For the latter, especially CUDA (Compute Unified Device Architecture) [NVIDIA, 2010a,b; Che et al., 2008] and OpenCL [Stone et al., 2010; KHRONOS Group, 2010] are available. CUDA is currently available only on NVIDIA GPUs, however, CPU support has been announced. It offers a flexible, higher-level programming interface with a focus on ease of integration. OpenCL allows to use both, GPU and CPU, is vendor-independent and similar to CUDA's driver API. However, it has a large environment setup overhead [Du et al., 2012; Karimi et al., 2010]. In this context, DirectCompute is another API to address the GPU, but can only be used within DirectX 11.

Owens et al. [2007, 2008] discuss the opportunities in general purpose GPU (GPGPU) computing and present a performance comparison of a single-core CPU, OpenGL and CUDA implementation (based on data from [Harris et al., 2007]). They show that the benefit from OpenGL or CUDA versus CPU depends on the number of considered elements. Cohen and Garland [2009] employed CUDA for solving equations in the field of computational fluid dynamics and gain performance up to factor 8 comparing CUDA to their Fortran code utilizing an eight-core CPU. Similarly, Rossinelli et al. [2011] compared GPU performances of OpenCL and Cuda versus multi-core CPU based on SSE and OpenMP for particle-mesh interpolations. For GPU performance, they achieved a speed-up of up to factor 155 (vs. CPU single-core) and factor 9 (vs. CPU multi-core, 16 threads). Furthermore, CUDA yielded slightly higher performance than OpenCL, whereas this relation might slightly change depending on the data size.

In summary, multi-core CPUs can easily be employed for improving algorithm performances. For visualizing the results, data has to be transferred to GPU memory, which brings additional delays. In particular, if intermediate results shall be visualized repeatedly, computation on the graphics card is preferable. Graphics processors benefit especially from large data which can be processed in parallel.

6.2 GPU-based Mesh Smoothing

GPU-based mesh smoothing is often related to subdivision algorithms, since the desired smoothness may require finer models (see Sec. 3.3.1). Examples for GPU versions of Catmull-Clark subdivision have been given by Bolz and Schröder [2002] and Shiue et al. [2005]. Bolz and Schröder [2002] give special attention to caching issues, such that expensive memory references are avoided. They precompute the limit values of the Catmull-Clark generating functions for the most common parameter configurations. During runtime, a linear combination of the precomputed basis functions is then evaluated to obtain the subdivided, smooth surface. Shiue et al. [2005] did also optimize the data access, but are not restricted to any precomputed setups.

Ni et al. [2008] presented a GPU-based algorithm for smoothing quad meshes by subdividing and converting them into bicubic and composite patches. The resulting shapes are similar to Catmull-Clark subdivision surfaces. For an example mesh with 12,000 quads, approximately 50 fps have been achieved. Similarly, Myles et al. [2008] convert a polyhedral mesh into a smooth, piecewise polynomial surface. Both, Ni et al. [2008] and Myles et al. [2008] are real-time capable and employed vertex and geometry shaders within DirectX 10. However, the data they are referring to are small (e.g., 200 to 1300 vertices). Compared to typical anatomic surface models, these meshes are relatively small. In this chapter, larger surface models are considered, for which no further subdivision is required during rendering. In an early GPGPU work, Bolz et al. [2003] presented mesh smoothing as use case for their GPU sparse matrix solver. They used mesh smoothing just as show case and no further discussion of this topic has been given.

Up to now, no other literature is known that contains information on performance improvements for Laplacian mesh smoothing filters on the GPU.

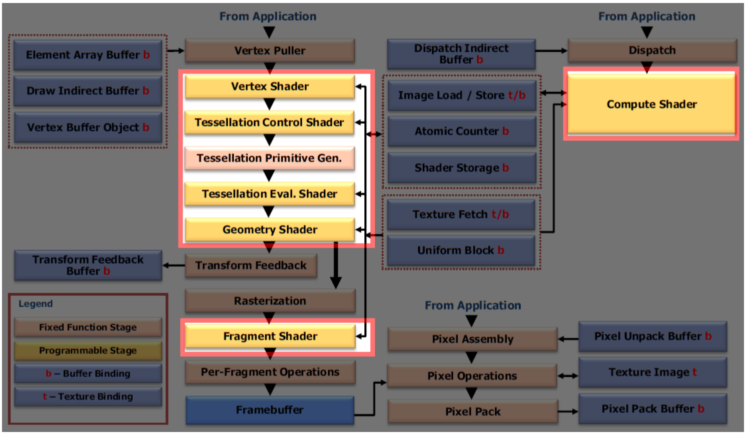


Figure 6.1: *The dataflow in the OpenGL pipeline. The different shader stages are highlighted in red (modified from [Segal and Akeley, 2012]).*

6.3 OpenGL-based Mesh Smoothing

Among the variety of parallelization techniques, OpenGL is one option for massive parallel computations. Other frameworks (CUDA, OpenCL) are often more suitable for GPGPU solutions, but the shader concept of OpenGL is ideally suited for mesh operations being closely related to graphical operations, such as displacement mapping or computation of vertex attributes (normals, colors, illumination).

6.3.1 OpenGL Shaders

OpenGL stands for Open Graphics Library and represents a progeramming interface for 2d and 3d computer graphics. Basically, it allows for specifying geometry via vertices and indices. The latter may be used to connect the vertices to further primitives, such as lines or triangles which finally assemble a complete 3d model. These data are transferred to the graphics card and then processed by the GPU. For standard rendering tasks, the so-called fixed-function-pipeline may be used. For custom illumination effects or more complex computations for each graphics primitive, OpenGL's shader pipeline may be employed. As the word already says, shaders are programs running on the GPU, which are primarily intended for shading effects.

There exist different types of shaders (see Fig. 6.1):

1. **Vertex shaders** process attributes for each vertex separately. A typical example is illumation calculation from the vertex normals and the position of a light source. Similarly, vertex coordinates are projected into the camera coordinate system. Vertex shaders can only access vertex-specific attributes (e.g., position,

normal, color) or global (uniform) variables. They do not provide any neighborhood information, such as the number of adjacent triangles or the location of neighboring vertices. Moreover, vertex shaders cannot create new geometry.

2. **Tessellation shaders** can be executed for each triangle and are supposed for on-the-fly subdivision of the input primitive. This subdivision may involve further properties, such as error or curvature metrics. However, no further neighborhood information is available – only the vertices of the respective triangle can be accessed.

3. **Geometry shaders** may also be executed for each graphics primitive (vertices, lines, triangles) and can generate new geometry. The emitted vertices have to be transformed into the camera space and properties, such as normals or colors, may be passed to the next shader stage. There exist different geometry shader execution modes, which allow for accessing all edge neighbors of a triangle. However, this is not sufficient for obtaining the full 1-ring neighborhood of a vertex as it would be required for smoothing applications.

4. **Fragment shaders** (also: pixel shaders) represent the last shader step in this pipeline. They determine the color of each pixel in the render window. This color may depend on several (semi-opaque) primitives processed in earlier shader stages. For example, the same illumination computations as in vertex shaders may be performed – but at a much finer scale: all computations are performed for each visible pixel. Thus, fragment shaders are not designed for per-primitive-computations.

5. **Compute shaders** are a special shader type introduced with the newest OpenGL version (4.3). Basically, they have the same capabilities as vertex shaders, but are less related to graphics primitives. They are generally intended for non-rendering computations and thus not part of the standard shader workflow (see Fig. 6.1).

6.3.2 Data Structures

For being able to perform mesh smoothing via OpenGL shaders, especially the issue of missing neighborhood information had to be resolved. Since mesh smoothing is performed per vertex, vertex shaders are appropriate. For accessing topological information, a custom data structure has been built, which employs vertex attributes (vertex buffer objects, VBOs) and large 1d-arrays (texture buffer objects, TBOs). The latter allow for arbitrary GPU memory access via sampler buffers. In newer OpenGL versions, other extensions for arbitrary memory acccesses are available, such as EXT_shader_image_load_store (OpenGL core since version 4.2) or ARB_shader_storage_buffer_object (OpenGL core since version 4.3). TBOs are supported even by older OpenGL versions and graphics cards and have thus been chosen at the time of implementation.

The basic structure of a Laplacian mesh smoothing filter has already been shown in Section 3.3.4. These filters are executed iteratively – the vertex positions determined within one cycle have to be available for the following iteration. Thus, for parallelizing such filters, only the iteration over the vertices can be considered. Each vertex computes

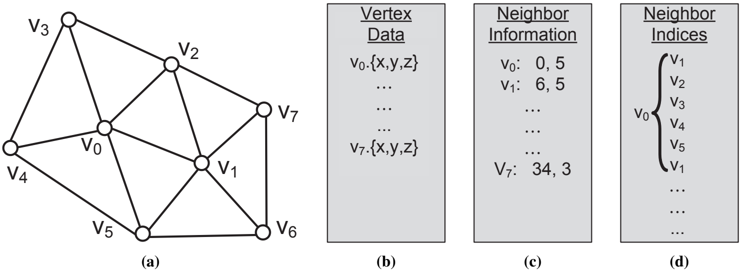


Figure 6.2: *The data is organized in three arrays: vertex coordinates (b), a lookup table (c) holding information on the number of neighboring vertices and their location in the list of neighboring vertices (d).*

its new position based on the coordinates of its neighbors from the former cycle. But within the current cycle, there is no dependence between the vertices. Smoothing filters with additional correction steps, such as Laplace+*HC* or LowPass, proceed similarly. Usually, they iterate over the vertices a second time, which can be processed in the same manner.

As a basic prerequisite, a custom mesh data structure has been realized, where each vertex holds information on its 1-ring neighbors (see Fig. 6.2). This requires an array for the vertex coordinates (see Fig. 6.2b) and a second one for the indices of neighboring vertices (see Fig. 6.2d). For each vertex, the list of neighboring vertices is finalized by adding the first neighbor to the list again. This duplicate neighbor index simplifies the code for average vertex normal generation from the adjacent triangles, which is required in later steps. Through counter-clockwise sorting of the list, the triangle fan can easily be created. Furthermore, a lookup table (see Fig. 6.2c) is necessary, which contains neighborhood information, such as the number of neighboring vertices and the location (offset) of the indices of neighboring vertices in the aforementioned neighbor index list (see Fig. 6.2d). The provided data structure provides fast access to the coordinates of vertex neighbors. Topological changes, e.g., by adding/removing vertices or changing the triangulation, are not intended, since mesh smoothing filters will only move existing vertices, but not remove them or add new vertices. Such an extension is, however, feasible by providing more dynamic access to the data structure from within the shaders via new OpenGL extensions, such as shader storage buffer objects.

In the current GPU architectures, threads are processed in-order as groups. Vertex valences influence the efficiency, as all threads of a group will have to wait for the vertex with the highest valence. On NVIDIA hardware with a group size of 32 threads, the test models which have been employed in this work, yielded a thread utilization of 75%. Compared to perfect efficiency, simulated by forcing equal valences during smoothing and slightly above each model's average, around 12% performance loss have been measured for the anatomic surface models. In the context of the achieved computation times (see Sec. 6.3.4) and considering, that no further measures have been taken for

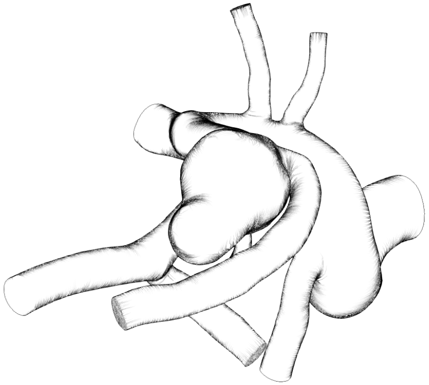


Figure 6.3: *The* ConFIS *method applied to a cerebral aneurysm surface model (see also [Lawonn et al., 2013b]).*

load balancing w.r.t. the valences, this is considered acceptable for the presented use case, but could be optimized. Moreover, the data layout ensures that the cache line mechanism is utilized well, as required data is contiguously stored in memory. The neighborhood indices are stored as densely packed arrays, and thus, can benefit from the hardware's cache line mechanism. Typical cache line sizes are between 32 and 128 bytes, which provides room for 8 to 32 indices with a single fetch. The considered models are typically created via the marching cubes algorithm, whose well-defined vertex creation order greatly favors data locality of neighboring vertices. The linear mesh creation ensures that neighboring vertices are often stored closely in memory. The lookup of neighborhood information is expensive with respect to the memory accesses, since the data might be widely distributed in memory. The memory accesses to the vertex attributes via vertex buffer objects and data output via transform feedback are, however, coalesce.

A modified version of the described data structure (see Sec. 6.3.2) has also been employed for a novel illustrative real-time rendering method presented by Lawonn et al. [2013b]. The *ConFIS* method calculates precise streamlines on the surface of a 3D model with a given curvature vector field. By determining minima and maxima of a related scalar field, salient regions are detected and emphasized (see Fig. 6.3). The algorithm for analyzing the scalar and vector fields as well as the streamline generation and final real-time rendering make heavy use of the above discussed data structure.

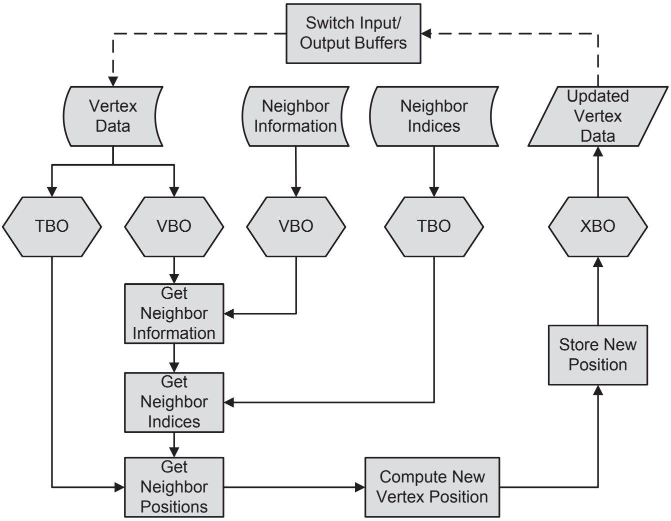


Figure 6.4: *During the vertex shader calls, each vertex needs to catch neighborhood information from the lookup table (number of neighbors and position in index list), passed as vertex attributes (VBO), and subsequently the indices of neighboring vertices being bound as texture buffer (TBO). With the coordinates, also being bound as texture, the smoothed position can be determined and finally be stored within a transform feedback buffer (XBO).*

6.3.3 Workflow

OpenGL provides a buffer-centric data model, in which the user can manage memory on the device and bind it to different functionalities of the rendering pipeline. VBOs allow to upload vertices and vertex attributes to the graphics card. For mesh smoothing, this means the vertex's own position and the neighborhood lookup data. The TBOs allow for arbitrary memory access to the indices of neighboring vertices and their locations.

Thus, the buffer holding all the vertex coordinates and the buffer with the indices of neighboring vertices are bound as sampler buffers. For storing the vertices of the smoothed mesh, a transform feedback buffer (XBO) with the same size as the vertex buffer is used. This setup is shown in Figure 6.4 and allows to read all current vertex positions and store the updated locations from within vertex shaders. Moreover, this setup is not restricted to constant smoothing weights. By simply adding another buffer object (bound as vertex attribute, VBO), local weights may be included (see Sec. 6.3.6). For example, vertex movement could be locally restricted depending on local mesh properties to smooth artifact areas only, to maintain minimum distances to a neighboring, crucial anatomic structure (see Chap. 5), or vertex position constraints (e.g., [Gibson, 1998]).

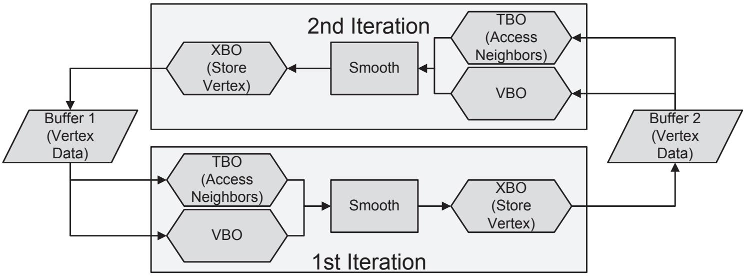


Figure 6.5: *The input and output buffers are switched after each smoothing cycle by binding them as VBO/TBO and XBO.*

After each iteration of the mesh vertices, the input and output buffers need to be switched, such that the modified vertex positions become the input data for the following cycle (see Fig. 6.5). For that, the buffer containing the updated vertex data is bound as VBO and TBO, whereas the former vertex data buffer is bound as XBO to be ready to store the new vertex positions. After this procedure, the positions of the vertices have been modified according to the specific algorithm.

For correct visualization and shading, the related vertex normals have to be updated. The VBOs for vertex data and lookup as well as the neighborhood TBO can again be used to obtain all neighborhood information and finally compute updated vertex normals by iterating over the smoothed vertices once more. Due to the sorted indices in the neighborhood array, the predecessor and successor information can be used to compute the face normals of the adjacent triangles and subsequently the vertex normal. Since the data is already located in graphics card memory, it can directly be used for visualization without further delays. As a result, the OpenGL approach is advantageous over multi-core CPU approaches for tasks requiring an immediate visual feedback. Please note that the engaged buffers could also be shared, e.g., with CUDA for further processing, but the direct usage of OpenGL shaders avoids expensive interoperability with other computing APIs and yields independence of graphics card vendors.

For the described setup, at least OpenGL 3.0 or the appropriate extensions are required (EXT_gpu_shader4, NV/EXT_transform_feedback). The required functionality was introduced in OpenGL by NVIDIA in 2006 and is also part of the DirectX 10 runtime released in 2007.

Model	Vertices	#Neighbors		
		Min	Max	Avg
Artery 01	4,818	3	16	5
Cerebral Aneurysm	6,773	3	12	5
Carotid Artery	6,988	4	11	5
Sternocleidomastoid Muscle	38,069	4	15	5
Liver	247,772	4	13	5

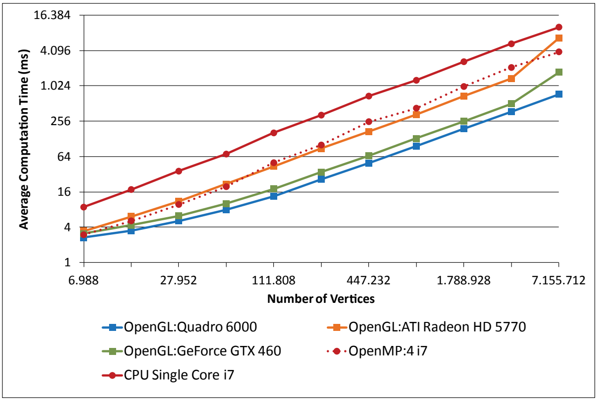
Table 6.1: *The medical surface models employed for evaluation of real-time smoothing capabilities. (Table reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

6.3.4 Performance Evaluation

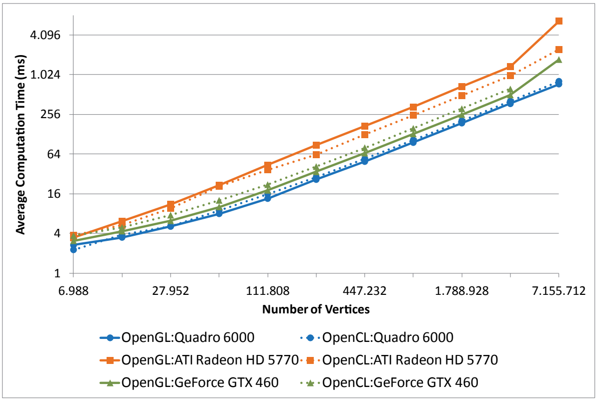
For benchmarking, an anatomic surface model of the carotid artery (see Tab. 6.1) has been used together with the Laplace+*HC* filter, since it is computationally more expensive than the default Laplace filter and yields a good tradeoff between artifact reduction and accuracy for medical surface models [Bade et al., 2006]. Following the recommendations of Bade et al. [2006], approximately 20 iterations should typically be used. The other weighting parameters influence the accuracy and smoothness of the resulting mesh, but not the computational effort.

The mesh data size was increased by duplicating the number of vertices until factor 1024 (7,155,712 vertices) to evaluate the behavior with increasing data size and a constant number of smoothing iterations. For each data increase step, the computations were run five times to subsequently average the results. Figure 6.6 shows the performances for GPU and CPU. It is not surprising that the smoothing performance scales with more computational power. For comparison, also the CPU has been employed with the data structure setup described in Section 6.3.2. Both, single- and multi-core (OpenMP) modes have been tested and compared against the OpenGL implementation.

Comparing the results for the ATI HD5770 and the Intel Core i7 using OpenMP, it is noticeable that strong CPUs are able to provide significant speed-ups similar to older GPUs. However, the simpler CPU implementation and roughly acceptable computational performance does not legitimate the performance loss from transferring the results to GPU memory for the required fast visualization. The comparison of OpenGL and OpenCL, which both run on the GPU, showed only marginal differences. For the ATI HD5770, OpenCL was slightly better, whereas this changed for the NVIDIA devices. For one test case (NVIDIA GTX 460, OpenCL), the model size of more than 7 million vertices could not be handled, since the operating system processes together with the benchmark application required too much GPU memory. The surface models addressed in this chapter have been generated from (segmented) tomographic image data, roughly consisting of up to 250,000 vertices (e.g., the liver model in Tab. 6.1). Using the NVidia GTX460 as common reference graphics card, frame rates of approximately 25 fps (40 ms computation and rendering time) are achieved for this comparably large model with 20 iterations of the Laplace+*HC* algorithm. The subsequent update of the vertex normals can be neglected at this point. Thus, completely smooth rendering is obtained, even if mesh smoothing would be executed for each render frame.



(a) Comparison of OpenGL (GPU) and OpenMP (CPU)



(b) Comparison of OpenGL (GPU) and OpenCL (GPU)

Figure 6.6: *Evaluation of GPU (OpenGL, OpenCL) and CPU (single core, OpenMP) computation times (Laplace+HC, 20 iterations) for increasing data size. The computation time increases approximately linearly with the number of vertices.*

6.3.5 Interactive Adjustment of Parameters

The achieved performance enables to combine smoothing filters and rendering in an interactive manner. In the reading of radiological image data with a radiology workstation, or in software for the exploration of medical image data, brightness and contrast adjustments are performed by mouse movements (i.e., horizontal – contrast; vertical – brightness). This feature is used extensively to investigate suspicious regions and variations carefully while the radiologist remains focused on the image data and its features. An interface with separated controls for the viewing parameters would require to interrupt the visual examination of the radiological images.

Borrowing this idea, mouse movements can be mapped directly onto relevant smoothing parameters. Similar to the radiology scenario, this interaction is activated by holding the right or middle mouse button. Any interaction causes an immediate execution of the selected smoothing filter. Depending on the movement direction and distance, the smoothing parameters related to horizontal and vertical movements are modified. Thus, the user can remain focused on its visual evaluation of artifact reduction and accuracy preservation. Since users familiar with radiology workstations and exploration of medical image data are among the target users, a high acceptance of this feature can be excepted. For occasional users, this should become intuitive very fast. Precise parameter adjustments should not be accomplished with this interaction method, since users would have to keep their eyes on all related smoothing parameters whilst moving the mouse horizontally and vertically. In turn, it is intended for an exploration of the parameter space. The general parameter effects and their interdependencies can be observed as well.

6.3.6 Anisotropic GPU-based Smoothing

The described workflow and underlying data structure have been designed to be able to realize those smoothing algorithms, which have been confirmed to be suitable for anatomic surface models [Bade et al., 2006]. In particular, this comprises the Laplace+*HC* and LowPass filters. These filters have in common that they apply uniform weights to the complete surface mesh. The presented approach is, however, not restricted to such uniform smoothing filters. A local adjustment of smoothing according to specific anatomic properties or features can also be realized.

Usually, anisotropic smoothing (see Sec. 3.3.4) serves to remove noise but to preserve feature regions at the same time. This is not directly relevant for anatomic surfaces, but might be helpful for surface models of implants, such as hearing aids, stents or artificial joints, which may exhibit sharp features. To demonstrate and test the flexibility of the above described framework, anisotropic smoothing as presented by Meyer et al. [2003] has also been implemented using OpenGL shaders. Similarly, anisotropic smoothing can be used to selectively reduce features (e.g., staircases) in anatomic surfaces. Following the idea of staircase-aware smoothing (see Chap. 5), anisotropic smoothing has been modified to smooth only those artifact regions. Feature selection is performed based on the mean curvature of each vertex. If the absolute value of the mean curvature exceeds the threshold λ, the corresponding vertex is permitted to move. Otherwise, the vertex remains at the same position. All vertices which are permitted to move are

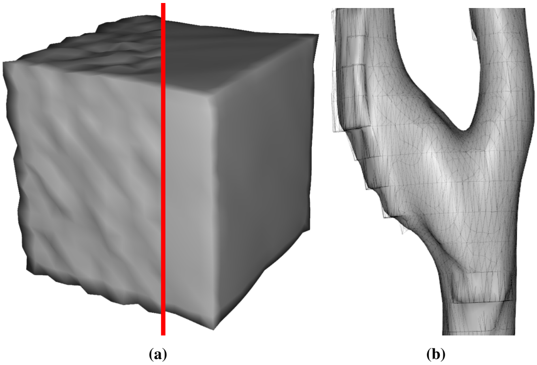


Figure 6.7: *(a) Cube model with added noise and the result of anisotropic smoothing. (b) Close-up view of a smoothed carotid artery model with staircase artifacts in the initial surface (superimposed wireframe). (Images reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

subsequently smoothed via Laplacian smoothing. All subsequent iterations follow the rule that, if one neighbor of a vertex was moved, the vertex itself can be smoothed in the next iteration step. This guarantees that staircase artifact regions are first smoothed at the sharp edges. With each additional iteration, the size of these regions is increased, which leads to gradual smoothing in the surrounding.

Sample results of the OpenGL-based anisotropic smoothing implementations are shown in Figure 6.7. The effects of classical anisotropic smoothing are demonstrated by the artificially generated, noisy cube model. By applying anisotropic smoothing, the noisy non-feature regions are smoothed, whereas the sharp edges are preserved (see Fig. 6.7a). In contrast, Figure 6.7b shows a model of a smoothed carotid artery with the initial wireframe superimposed to indicate the original artifacts. Artifact reduction has been achieved with the modified anisotropic smoothing approach.

These examples demonstrate that the OpenGL-based mesh smoothing framework described in this chapter is suitable for different smoothing filters. The framework can easily be adapted to consider specific local properties.

6.4 Visual Feedback

Real-time mesh smoothing, combined with an interactive approach, gives an expressive visual feedback and allows for a direct perception of modifications caused by different filters and parameters. The user does, however, not obtain detailed information on the resulting accuracy. Local distance changes (Euclidean distances) and changes of the structure's volume are typical accuracy measures. To determine the influence of mesh smoothing on accuracy, a comparison of the modified mesh M' versus a reference mesh M, regarded as accurate, needs to be performed. For a correct comparison, M should have a high resolution and depict the artifact-free, real shape of the considered structure. Usually, such a perfect reference mesh is not available for anatomic surface models. Hence, the changes between the initial mesh M and M' serve as accuracy indicator.

6.4.1 Approximation of the Local Error

Within the presented shader-based framework, a comparison of each vertex $\mathbf{v}' \in M'$ to each vertex of $\mathbf{v} \in M$ would be highly inefficient. Employing the Euclidean distance of $\mathbf{v}'_i$ to its initial position $\mathbf{v}_i$ would roughly reflect the local changes, but does not account for possible drifts along the surface. Thus, error approximation is divided in two steps:

1. the determination of the reference neighborhoods $\mathcal{N}_{ref}$ during mesh smoothing (see Fig. 6.8a) and
2. the final computation of the Euclidean distance (see Figs. 6.8b and 6.8c).

The initial reference neighborhood $\mathcal{N}_{ref}(i)$ of $\mathbf{v}'_i$ is the 1-ring of $\mathbf{v}_i$. As the position of $\mathbf{v}'_i$ is modified during smoothing, it may tend to drift along the surface (see Fig. 6.8a). As a consequence, one of the 1-ring neighbors of $\mathbf{v}_i$ might now be closer to $\mathbf{v}'_i$ than $\mathbf{v}_i$. Thus, this neighbor vertex becomes the new center $\mathbf{v}_{ref}(i)$ of $\mathcal{N}_{ref}(i)$. This procedure is repeated after each smoothing step and moves $\mathcal{N}_{ref}(i)$ according to the drift of $\mathbf{v}'_i$. After completing the smoothing procedure, the final error computation step is executed. All triangles $\mathcal{T}_{ref}(i)$ in the 1-ring of $\mathbf{v}_{ref}(i)$ are evaluated by testing whether $\mathbf{v}'_i$ is located inside the prism spanned by each of the triangles $\Delta_j \in \mathcal{T}_{ref}(i)$ (see Fig. 6.8b) and the corresponding normal. This can be solved by:

$$(a,b,c) = P_{\Delta_j}^{-1}\,(\mathbf{v}'_i - \mathbf{v}_{ref}(i)). \tag{6.1}$$

The columns of the matrix P_{Δ_j} are the vectors e_1, e_2 along the edges of the j-th triangle in $\mathcal{T}_{ref}(i)$, and the triangle normal n_i. For degenerated triangles, P^{-1} does not exist, such that they are are omitted from any further processing and the following is only performed for remaining non-degenerated triangles. If $a,b \geq 0$ and $a+b \leq 1$, $\mathbf{v}'_i$ is inside the prism of the considered triangle. Moreover, if n_i is orthonormal to e_1 and e_2, c returns the distance from $\mathbf{v}'_i$ to the triangle plane ($d = c$). In case $\mathbf{v}'_i$ is not inside any of the prisms of $\mathcal{N}_{ref}(i)$, $\mathbf{v}'_i$ might be close to one of the vertices or edges of $\mathcal{N}_{ref}(i)$ (see Fig. 6.8c). As a special case, $\mathbf{v}'_i$ might have left $\mathcal{N}_{ref}(i)$, but the closest vertex is still $\mathbf{v}_{ref}(i)$. In those cases, d is the Euclidean distance towards the edges of $\mathcal{N}_{ref}(i)$.

The movement of $\mathcal{N}_{ref}(i)$ might be stopped by a local distance minimum, and thus, not find the global (minimum) Euclidean distance to M. This occurs, if, e.g., a smoothed

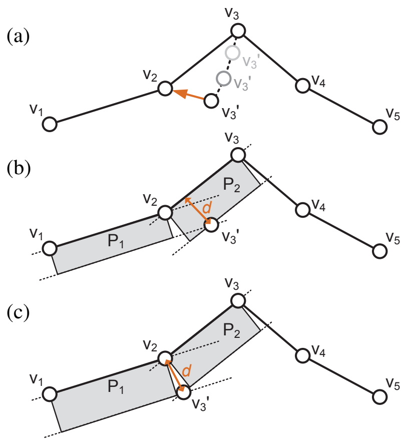


Figure 6.8: *2D illustration of the error approximation. (a) $\mathscr{N}_{ref}$ consists of $\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$. After several smoothing iterations, $\mathbf{v}'_3$ is closer to $\mathbf{v}_2$ than to its initial position ($\mathbf{v}_3$). Thus, $\mathscr{N}_{ref}$ comprises $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$. For determining the minimum distance to the original surface, $\mathbf{v}'_3$ is compared to the vertices and triangles of $\mathscr{N}_{ref}$ by testing whether it is located inside the prisms P_1, P_2 spanned by $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ (b) or outside the prisms (c). (Images reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

vertex is getting close to a topologically very distant part of the model (e.g., another branch of a vascular structure). This real minimum is, however, not important to the error feedback. Finally, the error is calculated by averaging all local errors:

$$\mathscr{E}_{M'} = \frac{1}{|V(M')|} \sum_{i \in V(M')} \max_{j \in \mathscr{T}_{ref}(i)} |(P^{-1}_{\Delta_j}(\mathbf{v}'_i - \mathbf{v}_{ref}(i)))_z|, \qquad (6.2)$$

where $()_z$ denotes the third component of the vector.

6.4.2 Mean Curvature

A decrease of curvature is an indicator for the degree of smoothing. Thus, curvature needs to be visualized as well. In the described framework, curvature can be determined within the final shader execution step, where the error is computed (see Sec. 6.4.1). Within the vertex shader, the discrete mean curvature normal operator $\mathbf{H}_M$ at a vertex $\mathbf{v}_i$ is determined by iterating over all adjacent vertices $N(i)$ and weighting the difference of the two vertices with the cotangent weights and the mixed area $A_{Mixed}(\mathbf{v}_i)$ according to Meyer et al. [2003]:

$$\mathbf{H}_M(\mathbf{v}_i) = \frac{1}{2\,A_{Mixed}(\mathbf{v}_i)} \sum_{j \in N(i)} (\cot(\alpha_{ij}) \;+\; \cot(\beta_{ij}))\;(\mathbf{v}_j \;-\; \mathbf{v}_i).$$

From this expression, the mean curvature $H(\mathbf{v}_i)$ can be determined by multiplying half of the normal $\mathbf{n}_i$ at $\mathbf{v}_i$:

$$H(\mathbf{v}_i) \;=\; \frac{1}{2}\,\mathbf{H}_M(\mathbf{v}_i) \cdot \mathbf{n}_i. \tag{6.3}$$

Finally, the integral of the mean curvature over the geometry is calculated. This can be expressed as the sum of $\mathbf{H}_M(\mathbf{v}_i)$ multiplied by $\frac{1}{2}A_{Mixed}(\mathbf{v}_i)\;\mathbf{n}_i$

$$\mathscr{H}_M \;=\; \int_M H\;dA \;=\; \frac{1}{2} \sum_{i \in V(M)} A_{Mixed}(\mathbf{v}_i)\;\mathbf{H}_M(\mathbf{v}_i) \cdot \mathbf{n}_i. \tag{6.4}$$

Depending on the specific purpose, different curvature approximations (e.g., [Rusinkiewicz, 2004]) can easily be integrated into the presented shader framework, but may also require much more computational effort. However, the framework is not limited to the smoothing context in this chapter, since it is suitable for other applications requiring curvature computation, e.g., for shape estimation or feature detection, such as in illustrative rendering [Praun et al., 2001; Rusinkiewicz et al., 2008; Lawonn et al., 2013b].

6.5 Sensitivity Analysis

The immediate rendering update in combination with curvature or error coloring provides a fast overview on the local effects. By modifying relevant parameters continuously via mouse movement (see Sec. 6.3.5) and observing the changes, the user performs a mental sensitivity analysis to explore the correlation between input parameters and smoothing results. Since parameters of different filters may have different meanings, an intuitive usage of these filters is not guaranteed. For example, the weights of the Laplace+*HC* method describe the influence of the original vertex positions. Thus, higher values represent lesser smoothing, whereas they usually stand for stronger smoothing in other filters. The following section describes how to support this analysis visually and quantitatively with *model quality graphs, model quality bars*, and *smoothing previews.*

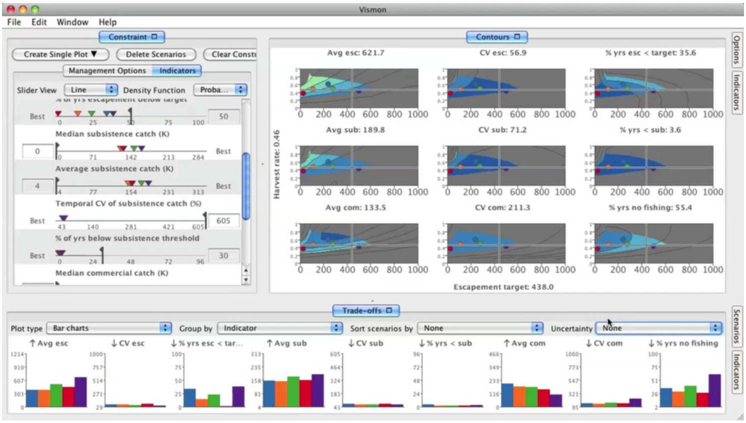
6.5.1 Related Work

The considered smoothing algorithms contain two (Laplace) to four parameters (Laplace+*HC*) as input. The influence of each parameter on curvature reduction and local accuracy may vary. Bade et al. [2006] performed an exhaustive analysis on mesh smoothing filters for anatomic surface models to derive parameter suggestions for different shape categories (e.g., compact vs. elongated). Within this study, numerous time-consuming measurements have been performed to find acceptable parameter sets for a good tradeoff between accuracy and smoothness. These parameter sets are generalized suggestions for different anatomic shape categories. It is not guaranteed that these suggestions are always valid or useful for arbitrary anatomic models.

The correlation between the input (smoothing) parameters and the resulting quality measures (curvature, local error, volume) for patient-specific anatomic shapes is not known. This correlation is usually considered as *sensitivity*. The presented real-time mesh smoothing framework enables users to interactively explore the parameter sensitivity by providing immediate quality feedback. Having the option to compute all metrics very fast, an automated sensitivity analysis would allow to make parameter suggestions for any surface model. In this context, Marks et al. [1997] described the idea of "design galleries", which provide precomputed visual results to the user. The gallery concept is used to provide an overview of the relation between input parameters and the corresponding results. Berger et al. [2011] presented an interactive approach for the investigation of system dynamics by analyzing parameter sensitivity. Chan et al. [2010] determine sensitivity as partial derivative of one variable with respect to another, approximated using linear regression. The sensitivity information is then used to augment scatterplots.

An example for automated parameter optimization has been presented by Selle et al. [2002]. They estimated optimal thresholds for vessel segmentation based on precomputed data and conveyed the effect of the threshold on the volume of the segmented structure. Thus, it becomes obvious where a further decrease of the threshold would lead to oversegmentation. Similarly, Torsney-Weir et al. [2011] presented *Tuner*, a complex system to replace tedious manual parameter testing by an approach that samples the parameter space and estimates the results of the segmentation algorithm. A statistical model guides the user towards areas which require a further segmentation refinement.

VISMON (see Fig. 6.9) [Booshehrian et al., 2012] is designed to support the data analysis of simulation results in the field of fisheries management. It allows people without background in simulation software to explore the simulation results in all of its parameter dimensions. The included sensitivity analysis helps to understand the dependencies between the considered parameters. Bruckner and Möller [2010] presented a software for supporting graphics artists during the generation of visual effects, such as smoke or explosions. The software allows for exploring the simulation parameter space visually. This visual exploration is performed to select different characteristics of the desired animation sequence.



Slider View: Line

Density Function: Proba...

Plot type: Bar charts

Group by: Indicator

Sort scenarios by: None

Uncertainty: None

Figure 6.9: *Screenshot of the software* VISMON *[Booshehrian et al., 2012].*

6.5.2 Volume Computation and Prefix Sums

Besides local distances, volume preservation is a major criterion for the accuracy of a smoothed model. Volume changes shall be minimized to avoid misleading measurement, such as during distance measurements in surgical planning or during simulation of blood flow in vascular structures. For a closed surface, the computation of the volume may be determined from the discrete form of the divergence theorem [Alyassin et al., 1994; Desbrun et al., 1999] (see Eq. 6.5). Following this equation, the volume is calculated by iterating over all m triangles. For each triangle, the face normal $\mathbf{n}_i$, the area A_i, and an arbitrary point $\mathbf{v}_i$ inside the triangle are required. Within the smoothing framework, this can be efficiently achieved by employing geometry shaders, which allow for computations per primitive (see Sec. 6.3.1).

$$\mathscr{V}_M = \frac{1}{3}\sum_{i=0}^{m} A_i\, \mathbf{v}_i \cdot \mathbf{n}_i \tag{6.5}$$

Finally, the volume fractions of the individual triangles need to be accumulated via parallel prefix sums [Blelloch, 1993; Harris et al., 2007]. After calculating the volume fractions of each triangle, all values are stored in one single buffer (see Fig. 6.10). Each four successive values are now interpreted as attribute of an imagined vertex and passed to the vertex shader as a four-element vector. These four values are summed up and stored, e.g., via the transform feedback mechanism. The procedure is repeated until only one result value is left. This one value can then be transferred to CPU memory at negligible costs. To further improve performance and reduce the number of vertex shader calls, up to 16 vertex attributes (each with 4 elements) can be handled for each

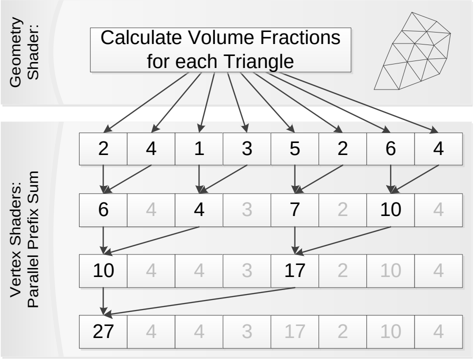


Figure 6.10: *The volume is calculated via geometry shaders and afterwards accumulated via a parallel prefix sum algorithm.*

imagined vertex on current GPUs. Thus, for n values in the buffer, the first vertex shader call processes $n/64$ imagined vertices.

Alternatively, the buffer holding the volume fractions could also be copied into CPU memory, where the sum-up procedure is serialized and would invoke less memory accesses. Depending on the size of the data, the data transfer between GPU and CPU remains a bottleneck.

6.5.3 Model Quality Graphs

Model quality graphs are supposed to improve the handling of mesh smoothing filters. These graphs display the correlation between a smoothing parameter and the model quality. The current setting is visually indicated by a vertical line. After initially loading the model and building the data structures, an initial computation of the quality metrics is performed.

Each smoothing parameter is increased within its defined bounds (e.g., weights $\in [0,1]$ with step size 0.05). During modification of one parameter, all others remain constant. After each smoothing operation, the mesh is evaluated again w.r.t. local distance changes, volume shrinkage, and curvature reduction. This procedure is repeated for all desired input parameters. The resulting data values for volume and error are scaled with the quality values of a very strong smoothing operation, e.g., standard Laplacian smoothing with #Its.=50, $\lambda = 1.0$. Figure 6.11 shows model quality graphs for a surface model of the carotid artery. In this example, three input parameters of the Laplace+*HC* filter have been considered. In contrast to standard Laplacian smoothing, two parameters (λ, μ) control the influence of original and intermediate vertex positions. Thus, stronger smoothing is achieved for small values of λ and μ. Without the model quality

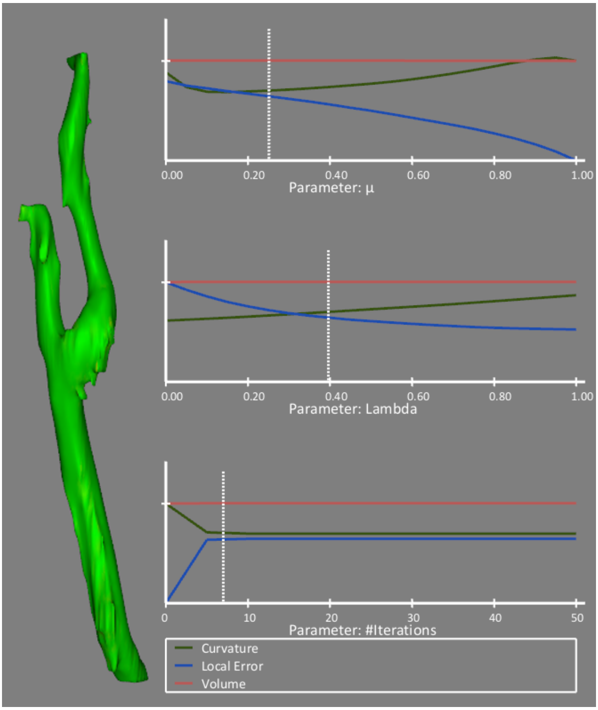
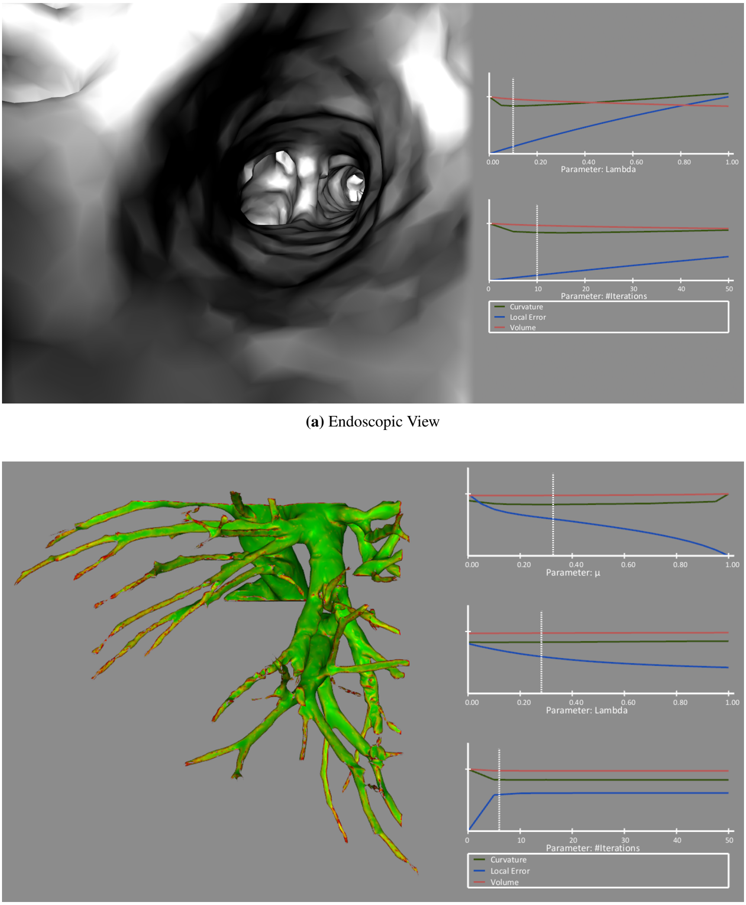


Figure 6.11: *Prediction of model quality for the carotid artery, color-coded by the local error (Laplace+HC with 3 parameters – #Its. & smoothing weights λ,μ). (Image reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

graphs, such filters can only be handled correctly by experienced users with background in model generation and smoothing algorithms. A reasonable usage would not be possible to occasional users. It would take several trial-and-error cycles to get used to the meaning of each parameter and to apply them correctly.

Another example of model quality graphs is shown in Figure 6.12. The endoscopic view (Fig. 6.12a) reveals that even small surface bumps can disturb the visual impression. The graphs indicate that smoothing with a standard Laplace filter without any correction steps would initially yield slight curvature reduction but also uncontrolled volume shrinkage for further parameter changes. The elongated vascular shape leads to an increase of the average curvature since thin branches tend to collapse. In contrast, by applying another filter with correction steps (Laplace+*HC*) the curvature can be reduced more efficiently without too strong accuracy loss (see Fig. 6.12b).



(a) Endoscopic View

(b) Complete Model

Figure 6.12: *Model quality graphs for a surface model of the pulmonary artery: (a) endoscopic view, Laplace filter with 2 parameters; (b) the same model colored from mean curvature, Laplace+HC filter with 3 parameters.*

Model quality graphs can initially display how accuracy and curvature would evolve if the corresponding input parameter is modified by the user. The quality graphs are updated after each interaction step, e.g., after the user stops the parameter modification via mouse movement. Scaling of the vertical axis is performed depending on the quality measures of the initial surface model without smoothing and of a model after very strong smoothing. For each input parameter (and diagram), a value hint is displayed (dashed vertical line in Fig. 6.11 and 6.12a), which helps the user to visually correlate the current parameter set with the precomputed quality measures. This value hint is also updated, if smoothing parameters are modified via mouse movements and serves as a visual reference. However, the user can also modify each smoothing parameter via the graphs. For integer parameters, such as the number of iterations, the value hint is aligned accordingly. The value indicator of floating point parameters can be moved continuously.

The described concept of quality graph previewing supports the user during the interactive parameter optimization and shall decrease the amount of trial and error cycles during parameter testing and model inspection. The graph-based quality preview provides a direct feedback on the influence of each input parameter. Thus, it assists the mental sensitivity analysis performed by the user. This feedback supports primarily experienced users and enables them to adjust the parameters efficiently. Occasional users will still need a few trial-and-error cycles to get used to the meaning of the displayed measures. The graphs enhance the visual feedback provided by the (curvature-/error-)colored, smooth mesh with quantitative measures.

6.5.4 Model Quality Bars

An initial informal evaluation of the model quality graphs showed that the graph representation with several lines and different slopes is confusing to some users [Mönch et al., 2012]. They noted that they paid special attention to the intersections of the curvature, volume, and error curves. In fact, these measures are correlated among themselves. For example, volume shrinkage and curvature reduction come along with an increasing error. The intersection points, however, have no meaning, since each graph is scaled individually. *Model quality bars* are an alternative representation of the same information, but avoid the mentioned issues.

Each quality measure is represented by one single horizontal bar. As a result, three bars (volume, error, curvature) exist for each smoothing parameter. Figure 6.13 shows a model quality bar for a surface model of the liver and the LowPass filter (three parameters). The colors of the bars are obtained from a perceptually linearized heated object lookup table [Levkowitz, 1997]. Each data value is again normalized as described in Section 6.5.3. The quality bars are configured to show bright colors for better values (less volume changes, less local errors, lower curvature). Thus, users do not have to interpret the colors of each quality bar separately. For finding a good quality-smoothness tradeoff, one would choose a bar region exhibiting preferably bright values for all three bars. Similar to the quality graphs, a value hint is displayed on each bar to indicate the current setting (dashed rectangles in Fig. 6.13).

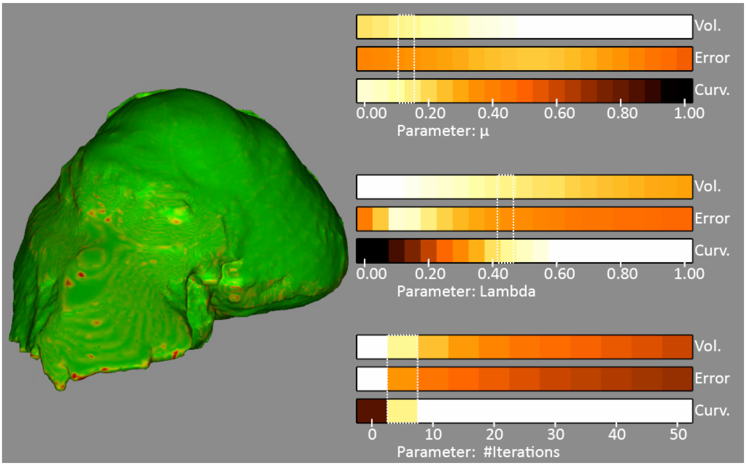


Figure 6.13: *Prediction of model quality for a liver model, color-coded by the local distance to the original model (LowPass with 3 parameters – #Its. & smoothing weights λ, μ). (Image reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

The quality bars can also be sorted according to the different quality measures (see Fig 6.14). Rearranging the bars of different input parameters together allows for a detailed comparison of their influence on the specific quality measure. As an example, it becomes more obvious which parameter affects smoothness most or which parameter causes the highest volume loss. This view is primarily intended for experienced users who want to compare and explore these effects. By default, the bars are arranged according to the smoothing parameters.

6.5.5 Parameter Suggestions

Model quality graphs guide the user during parameter tuning. A further simplification of the manual testing procedure can be achieved via parameter suggestions based on an automated analysis of smoothing parameters and quality measures. This is used for an initial suggestion of model-specific default values. Thus, a large number of parameter combinations needs to be evaluated w.r.t. accuracy and smoothness. Each time, volume preservation, local errors, and curvature are evaluated and stored. For an initial guess, a rough parameter sampling can be chosen, e.g. a step size of 5 or 10 for #Its. $\in [0, 50]$, or a step size of 0.1 for weighting factors $\in [0, 1]$. By executing smoothing and mesh evaluation for all parameter combinations, the quality score S is computed separately for each parameter set as a weighted sum of $\mathscr{H}$ (6.4), the average local error $\mathscr{E}$ (6.2), the

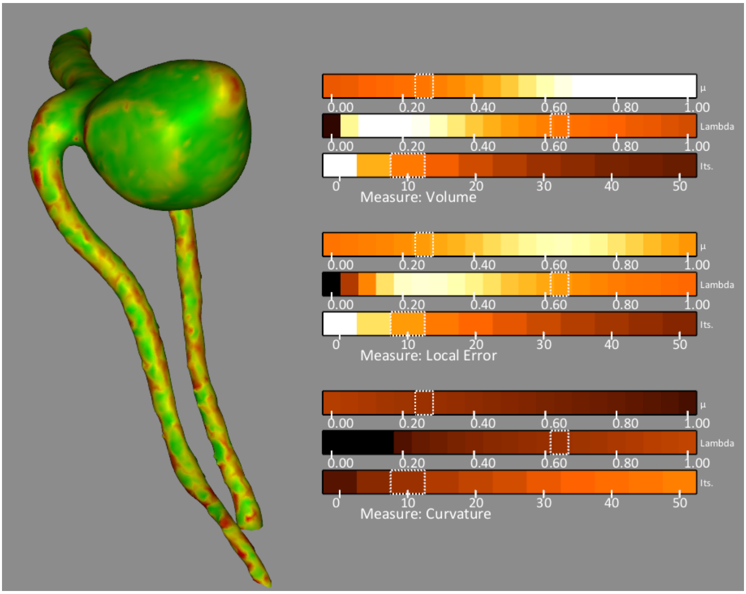
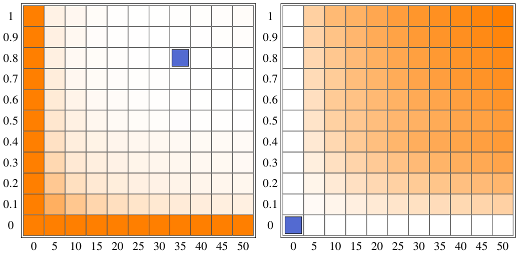


Figure 6.14: *Sorted model quality bars allow to compare the influence of the smoothing parameters to each measure separately (LowPass with 3 parameters – #Its. & smoothing weights λ, μ).*

volume $\mathcal{V}$ (6.5), the initial curvature and volume $\mathcal{H}_{M'}, \mathcal{V}_{M'}$, and the target smoothness coefficient τ:

$$S = \tau \, \frac{\mathcal{H}_{M'}}{\mathcal{H}_{M}} + (1-\tau)\, \frac{1}{2} \cdot \left(\frac{\mathcal{E}_{M'}}{\mathcal{E}_{M'}+1} + \frac{|\mathcal{V}_{M'} - \mathcal{V}_{M}|}{\mathcal{V}_{M}} \right). \tag{6.6}$$

In this equation, the quality measures $\mathcal{H}_{\mathcal{M}'}, \mathcal{E}_{\mathcal{M}'}, \mathcal{V}_{\mathcal{M}'}$ of the smoothed surface are compared to those of the original mesh. Each term is configured to yield small values, e.g., if the curvature of M' is reduced or if there are only small volume or local distance changes. The weighted sum is controlled by the target smoothness $\tau \in [0,1]$ describing the tradeoff between accuracy and smoothness. τ maps all n input parameters onto a single one. Hence, it may be used to bypass interaction with several separated parameters. The suitable parameter set is then found by minimizing S. For $\tau \rightarrow 1$, the suggestion considers more smoothness, whereas it yields higher accuracy for $\tau \rightarrow 0$. By default, τ is set to 0.5. It is not required to be modified by the user and does thus not cause more effort. Nevertheless, τ maps all n input parameters onto a single parameter to control smoothness and accuracy. Hence, it may be used to bypass interaction with several separated parameters.



(a) Score matrix of a liver **(b)** Score matrix of a petrous bone

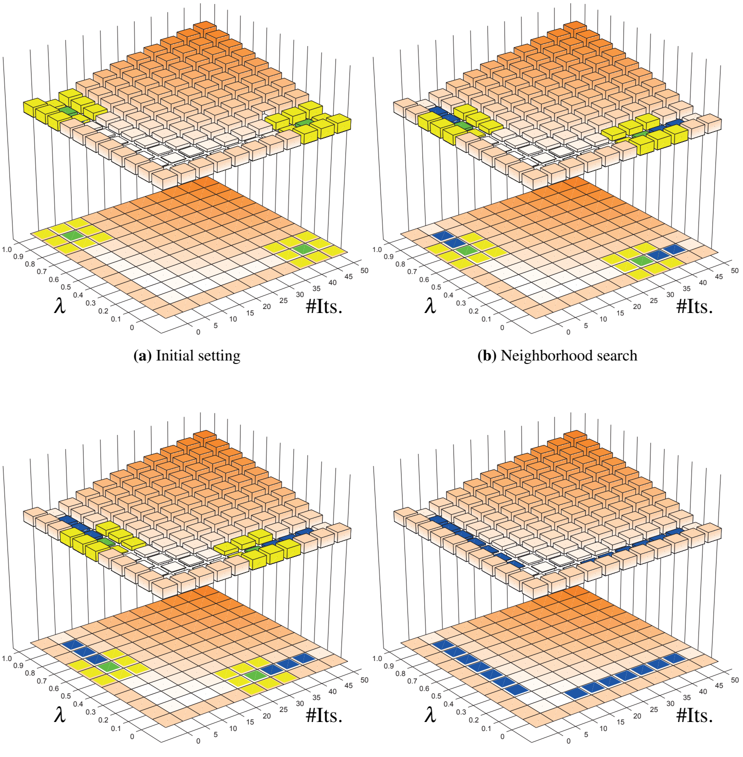
Figure 6.15: *Examples for resulting score matrices for two input parameters of Laplacian smoothing (#Iterations, λ). (a) Results for the liver model with $\tau = 0.45$. $\lambda = 0.8$ and #Iterations$=35$ are suggested (blue square). (b) For the petrous bone model (compare Fig. 6.17), $\lambda = 0.0$ and #Iterations$=0$ are suggested, since the model is initially smooth and exhibits numerous concave and convex parts which would cause an increase of the total curvature for further smoothing.*

	Carotid artery		SCM		Liver	
τ	λ	#Its.	λ	#Its.	λ	#Its.
0.0	0.0	0	0.0	0	0.0	0
0.2	0.4	5	0.9	5	0.6	5
0.4	0.9	5	1.0	15	0.2	40
0.6	1.0	15	0.8	45	0.3	50
0.8	1.0	50	1.0	50	0.6	50
1.0	1.0	50	1.0	50	1.0	50

Table 6.2: *Automatically generated parameter suggestions for differently shaped structures (uniform Laplace, 2 parameters). (Table reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

The effect of τ can be seen in Tab. 6.2. For the carotid artery (see Fig. 6.11), the average curvature increases with stronger smoothing, since thin parts tend to collapse. The presented method is sensitive to such changes. The employed liver in Tab. 6.2 is more compact and has a large volume. Smoothing can thus be stronger, compared to elongated structures.

Figure 6.15 shows examples for score matrices for a model of the liver and the petrous bone. The matrices have been obtained from performing smoothing for all possible parameter combinations of a standard Laplace filter (#Its.,λ) and subsequent quality evaluation. For the petrous bone model, minimum scores are detected for all parameter combinations with $\lambda = 0.0$ or #Iterations$=0$ (no smoothing). This results from large, initially smooth areas and several convex, concave, and frayed parts (see Fig. 6.17). Any Laplacian smoothing operation would increase the average curvature, since curvature decreases only marginally in the large, flat areas, but increases for the frayed parts and creases. This may, in turn, be used as an indicator that the chosen isovalue during surface extraction was not appropriate.



(a) Initial setting

(b) Neighborhood search

(c) Neighborhood search

(d) Minimization finished

Figure 6.16: *Examples for iterative score optimization for two parameters (λ, #Its.). (a) Beginning from two positions in parameter space, similar parameter sets are evaluated. (b, c) The parameter set, which yielded the minimum scores in each neighborhood is used to proceed until a local minimum is found. (d) The quality scores are homogeneous in regions with similar parameter configurations. (Images (a) and (c) reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

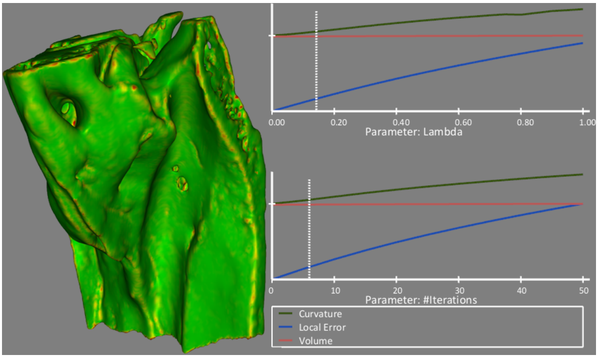


Figure 6.17: *Model quality graphs for the surface model of a petrous bone, color-coded by the local error. The model exhibits several convex and concave areas and an initially smooth surface. Smoothing leads to an increase of the average curvature for Laplacian smoothing (two input parameters).*

This optimization yielded stable and plausible results for all employed models and took up to 400 ms for small and medium sized models (e.g., SCM and carotid artery in Tab. 6.1). For large models, such as the liver, computation takes about 6 s, which is still acceptable, since the optimization is performed only once during model initialization. For more input parameters, as for the Laplace+*HC* and LowPass filter, but also for context-aware smoothing [Mönch et al., 2011a], computation times increase accordingly.

An iterative, local search in the spanned parameter space of S (see Fig. 6.16a) instead of evaluating all possible parameter sets decreases the computation time. The precomputation of $\mathscr{H}, \mathscr{E}, \mathscr{V}$, and S is not required. Starting at a given position in the parameter space, a local gradient search is performed. All parameter sets defined by the direct $(3^n - 1)$-neighborhood are considered, and smoothing, evaluation, and score computation (see Fig. 6.16b for the 2d case) are performed. The parameter set which yielded the minimum score, serves as next search center. This is continued until a local minimum has been found (see Figs. 6.16c and 6.16d). Depending on the number of input parameters n, n search patterns are created. For example, for two input parameters (#Its., λ), two search patterns are initialized. Each pattern is placed near the maximum value of one parameter. In the 2D case, the pattern centers are located at $\lambda = 0.95$, #Its.$= 5$ and $\lambda = 0.05$, #Its.$= 45$. This yields at most n local minima, from which the one with the lowest score is selected. Since the quality scores for the complete parameter space are not known, it cannot be guaranteed to find the global minimum, but the use of n search patterns, starting from opposite sides of the parameter space, yielded the global minimum for most test cases. Otherwise, the suggested parameters were always valid alternatives, since they were located in an area with similar quality scores as the global minimum. The iterative optimization was up to 6× faster than an evaluation of the complete parameter space. The use of only one search pattern is faster, but increases the likelihood that the optimum is not found.

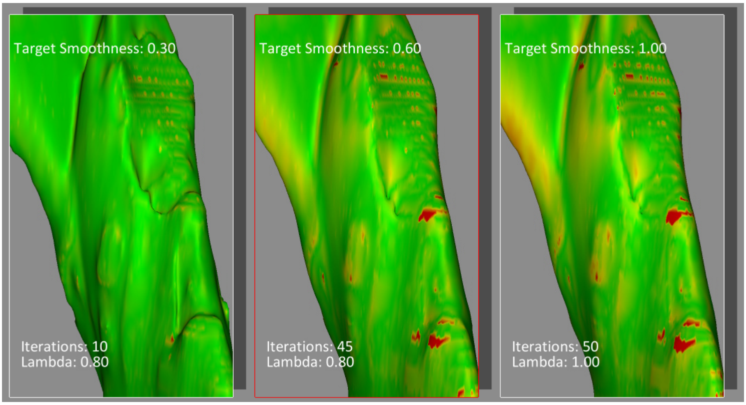


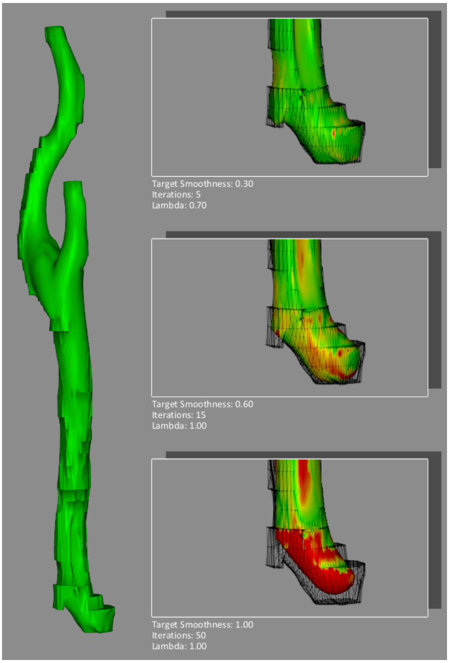
Figure 6.18: *Preview gallery example with three parameter suggestions for a model of a sternocleidomastoid muscle. (Image reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

The performance of the described approaches is limited by the large number of sequential operations for numerous parameter combinations which permanently load the GPU: mesh smoothing, followed by curvature, error, and volume computation, and finally data transfer into CPU memory. Higher performance could be achieved using a multi-GPU setup. This would allow to perform the evaluation procedure in parallel.

During user interaction (e.g., with quality graphs and bars), the suggestions for the parameters are updated and may also be added to the graph/bar representation. For this, each parameter is again modified separately, while all others remain constant. For each modified parameter, curves for the quality measures and scores are obtained according to Eq. 6.6. The location of the minimum score yields the suggestion for the specific parameter. This suggestion may slightly differ from the one derived with the above described optimization, but allows for an immediate suggestion update after each parameter modification.

6.5.6 Preview Galleries and Visual Inspection

Radiologists are highly familiar with comparing images arranged in galleries, e.g., rendered with different transfer functions. They compare these images to find similarities or conspicuous areas. Jankun-Kelly and Ma [2001] presented a spreadsheet-like interface for comparing the effects of different visualization parameters. In a similar manner, Liu et al. [2010] presented an interface for users with little experience in direct volume rendering to find suitable transfer functions. Motivated by these concepts and as an interface to automatically generated parameter suggestions, a *preview gallery* is provided to the user (see Fig. 6.18).



Target Smoothness: 0.30
Iterations: 5
Lambda: 0.70

Target Smoothness: 0.60
Iterations: 15
Lambda: 1.00

Target Smoothness: 1.00
Iterations: 50
Lambda: 1.00

Figure 6.19: *Example of the artifact inspection view, which allows to select specific artifacts and compare their reduction for three automatically generated suggestions. (Image reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

This gallery shows the same model for three different smoothing suggestions. These suggestions are generated for three predefined values of τ (e.g., 0.3, 0.6, 1.0) and circumscribe characteristic smoothing suggestions (little, medium, strong). For each of these target smoothness values, corresponding parameter sets are generated for the specific surface model. The visual results after applying the determined smoothing parameters are then presented to the user side by side. The views onto the smoothed models are synchronized, such that user can easily explore the different smoothing suggestions via rotation and zoom. If one of the suggestions fits the user's needs, it can be directly selected for data export or further processing in the user's specific model post-processing pipeline. Otherwise, a double-click on one of the suggestions switches to the default view with model quality graphs or bars and allows for further guided parameter editing.

The gallery view serves as an initial interface for observing and comparing smoothing suggestions. Since all views and underlying parameters are generated automatically, the necessity of any manual parameter adjustments (λ, μ, #Its., or τ) is reduced to a minimum.

A modification of this gallery concept may be used for model and artifact inspection. For that, the standard view to the 3D model, which can be smoothed, e.g., using mouse movements, is modified to show three preview areas next to the surface model (see Fig. 6.19). The user can now select a specific part of the model which is then shown magnified in the preview areas. Each of the previews provides again different smoothing suggestions (see above), which are superimposed with the wireframe of the initial model. The intention of this tool is that users can keep their focus on selected relevant regions which shall be smoothed or preserved. In parallel, the main view to the model can be used for further interaction and user-specific smoothing settings.

6.6 Framework Evaluation

An informal evaluation with eight participants has been performed to assess the presented functionality: smoothing via mouse movement, interaction with the quality graphs and bars, and plausibility of smoothing suggestions. The following sections describe the evaluation procedure and results.

6.6.1 Procedure

All volunteers had a strong background in medical imaging and visualization, such that they were able to differentiate between staircase artifacts and anatomic shapes. Six participants were using surface models for exploring anatomic scenes on a regular basis. The other two were familiar with 3D models and had more background in visual analytics. Three had in-depth knowledge of model generation and were familiar with mesh smoothing algorithms. The evaluation was conducted with two artificial and five anatomic surface models (see Tab. 6.1). The artificial models were a sphere and a torus. For both, the vertex positions have been modified with additional noise in the direction of the vertex normals. Standard Laplacian smoothing with uniform weights has been used for all models. The evaluation was conducted in four steps:

1. Each participant got a brief introduction to the interface and the interaction options. They were informed that for each subsequent smoothing task the goal was to find a suitable tradeoff between accuracy and artifact reduction. The participants were asked to think aloud about their actions and questions.
2. The first task was to smooth an anatomic model of the carotid artery using mouse movements w.r.t. the above criteria. The model was color-coded by the local deviation from its initial surface and the latter one could be superimposed as wireframe. Afterwards, the participants were introduced to the gallery and artifact inspection views and could try each viewing mode.
3. The second task was to smooth the noisy sphere and torus. This served as introduction to the interaction with the model quality graphs and bars. For each model, smoothing had to be performed via quality graphs and bars (without parameter

suggestions). One half of the participants started with the quality graphs for both sphere and torus, and switched to the bars afterwards. Accordingly, the other half began with the quality bars for both models. Although the focus in the paper is set on anatomic models, the usage of artificial models allows for a comparison to a ground truth (the initial sphere and torus without any noise). Moreover, the sphere and torus are shapes which are known by all participants, whereas the imagination of an ideal shape of an anatomic model may vary among the participants. As quantitative mesure, the two-sided Hausdorff distance between the smoothing results and the noise-free models was determined.

4. The last step comprised smoothing of four differently shaped anatomic models: an aneurysm, a branching vessel, a sternocleidomastoid muscle, and a liver. The participants had the free choice between quality graphs and quality bars. Additionally, parameter suggestions were shown in the graphs and bars. After finishing each model, the testers were asked about the suitability of the suggestions.

6.6.2 Results

In summary, all participants gave positive feedback on the framework. Three people commented that they liked in particular the real-time capability, since interaction and waiting times are reduced and an immediate feedback is given. This has been used throughout all steps of the evaluation. The participants employed this functionality to explore the parameter influences – they configured the parameters to see the original model and applied very strong smoothing subsequently. Afterwards, they began to solve their specific tasks.

During step two, all participants took a while to explore the changes that happen to the model during smoothing. In this context, mouse movement smoothing was rated as helpful, since they did not have to put their attention to parameter editing fields. The presented alternative viewing modes (gallery and artifact inspection view) were considered useful. Most people described the gallery as a good initial overview, where they could select one smoothing suggestion and modify it slightly, if required. It was mentioned that the visual difference between the gallery items has to be strong enough, such that the main differences become obvious at the first glance. Two participants were interested in more gallery items, e.g., to compare multiple suggestions for different smoothing methods. In contrast, the other test persons commented that the visual comparison of more than three suggestions might also take more time, which would compensate the benefit of reducing manual configuration effort. The artifact inspection view was rated more useful than the gallery view, since suggestions for regions could be observed and visually compared against a custom result. One person suggested to add the possibility to select multiple artifacts and observe these regions during manual smoothing instead of showing smoothing suggestions for one selected artifact.

Comparing the results for the artificial models, which have been smoothed using model quality graphs and bars, no relevant differences could be found. Similar accuracy has been achieved with both quality hints. For both modes, the participants described them as helpful, since they get an initial overview of the how model quality would develop. Most volunteers used them to find conspicuous points in the curves or in the bars (e.g., the strongest curvature reduction). However, two participants also mentioned that they were looking for crossings of the curves – although they knew that these points had no meaning and were caused by the scaling of the data values. The majority described the bars as easier to understand, since brighter values could be considered as *better* for all quality measures. Three people focused primarily on the surface models during smoothing and referred to the graphs/bars for subsequent validation and fine tuning. The interaction with the graphs and bars to modify the parameters was also considered helpful. Only two persons preferred to use mouse movement smoothing to remain focussed on the model.

For the anatomic surface models (step 4), four participants used only the quality bars as hint, two preferred the graphs, and two started with the graphs and then switched to the bars. The two latter commented afterwards that they initially considered the graphs as more intuitive, but always had to think about the meaning of the curves (volume: should not deviate too far from original value; average error: should be minimized; curvature: should be minimized). With the bars, however, they only had to look for bright colors without interpreting the measures.

The displayed parameter suggestions (directly after model initialization; updated during parameter modification) were rated helpful and plausible by all participants. In some cases, they used the suggestions as initial recommendation which they slightly modified to fit their subjective comprehension of an accuracy-smoothness tradeoff.

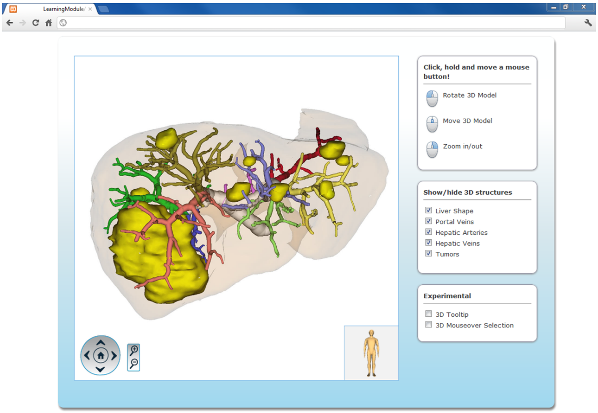
6.7 Application

The described methods are the core of a real-time smoothing framework for medical surface meshes. It is employed within *fixed* model generation pipelines to select smoothing parameters automatically based on a predefined target smoothness τ. During interaction, model quality graphs and τ guide the user.

6.7.1 Interactive Model Generation for Rapid Prototyping

The framework is utilized within a prototype software for the segmentation and surface extraction of structures for building physical models using rapid prototyping technology (see Sec. 2.3). Chapter 4 described the enormous manual effort being necessary to prepare a vascular surface model with complex pathologies for rapid prototyping. In all these cases, the surface and physical models are required to depict the individual anatomic shape and features to provide a realistic look and resemble the intraoperative situation. Thus, the geometry should be smooth and accurate. Within this context, medical experts operate the software which comprises segmentation and model generation. Interaction with the software has to be intuitive, fast, and effective.

Mesh smoothing is applied subsequent to a real-time Marching Cubes implementation (similar to [Dyken et al., 2008]), which yields an immediate surface update after isovalue adjustment. Models range between 50k and 300k vertices. With the immediate



Click, hold and move a mouse button!

Rotate 3D Model

Move 3D Model

Zoom in/out

Show/hide 3D structures

☑ Liver Shape
☑ Portal Veins
☑ Hepatic Arteries
☑ Hepatic Veins
☑ Tumors

Experimental

□ 3D Tooltip
□ 3D Mouseover Selection

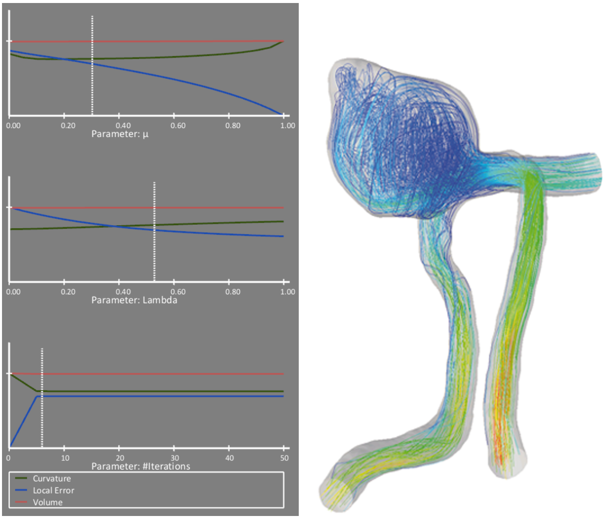
Figure 6.20: *Screenshot of the web-based rendering tool, which is integrated into a medical e-learning platform. Surface models of the liver, metastases, and hepatic vasculature were generated. The colors represent vascular branches supplying different territories (Courtesy of Steven Birr, University of Magdeburg).*

(graph/bar-based) quality feedback, smoothing does not interfere with this highly interactive procedure. The mapping of smoothing parameters to mouse movements integrates well to the interaction schemes known by medical experts. Initial parameter suggestions have been successfully utilized to minimize iterative parameter testing.

6.7.2 Web-based Medical e-Learning

Web-based rendering (e.g., via X3D, WebGL) in combination with higher bandwiths of current internet connections enables web-based e-learning platforms to share and discuss clinical cases. They comprise tomographic data, but also explorable surface renderings. Such models may be provided by platform users or dedicated authors. In both cases, it is most likely that several models need to be generated, e.g., to describe spatial relations of separated structures.

The presented methods have been integrated into the model generation pipeline of a web-based e-learning platform that incorporates Web 2.0 functionality to enhance the cooperation between surgeons (see Fig. 6.20). The users providing content for this e-learning platform prepare 3D models according to the requirements of a medical expert, e.g., a liver surgeon. These users, e.g., technical assistants or co-workers in medical informatics, are familiar with model generation. Since usually several models need to be prepared, manual effort should be reduced, but still allow for full control. Especially the preview gallery view integrates well to this procedure, since initial suggestions are presented and may directly be used. In case that no suggestion fits the specific needs,



(a) **(b)**

Figure 6.21: *(a) Model quality graphs for a cerebral aneurysm (Laplace+HC filter). (b) The aneurysm model with embedded flow information obtained from CFD simulation. (Images reprinted from [Mönch et al., 2013b], © 2013 The Authors Computer Graphics Forum © 2013 The Eurographics Association and John Wiley & Sons Ltd)*

the model quality graphs and bars may be used for a more precise but fast parameter adjustment.

6.7.3 Simulation of Blood Flow

CFD of vascular systems is performed to estimate the rupturing risk of aneurysms and the ability to reduce this risk, e.g., by inserting stents (see Sec. 2.2.2). Accurate and smooth surface models are used as input for the patient-specific simulation of blood flow [Cebral et al., 2005, 2001] (see Fig. 6.21). The extraction of a surface from the image data involves a lot of manual effort to remove artifacts caused by, e.g., low resolution, partial volume, beam hardening effects, or insufficient contrast agent distribution [Mönch et al., 2011b]. In the context of CFD, this procedure is usually executed by people with many experiences in segmentation and model generation. Permanent feedback of medical experts is required to discriminate between surface artifacts and pathologic situations. CFD simulation raises several constraints, which also need to be considered during model generation (see Chap. 4). Thus, the pipeline is often executed several times for the same data. To reduce this effort, the methods within each step should be easy to configure and should not introduce disturbing waiting times. The framework integrates these characteristics, since resulting errors can be evaluated immediately and parameters can easily be adjusted according to different requirements. By employing the quality score, appropriate smoothing suggestions can be obtained automatically without the need for manually adjusting the smoothing parameters after the model input has been changed in earlier pipeline steps. Moreover, the artifact inspection view has been commented to be a helpful tool for observing selected artifacts and comparing their reduction for different smoothing suggestions.

6.8 Summary

Different applications working with anatomic surface models benefit from interactive real-time mesh smoothing. Using OpenGL, uniform and anisotropic mesh smoothing filters have been combined with a mapping of mouse movements to their input parameters. This is strongly motivated by the typical interactions of the intended users (e.g., radiologists, radiology technicians) with medical image data. The interaction scheme is not intended to set parameters precisely, but provides insight to the influence of each parameter. This chapter described how quality measures (local error, volume, curvature) are efficiently determined to provide visual feedback. The achieved performance enabled *model quality graphs* and *model quality bars* for visually guiding the user during parameter adjustments.

By evaluating accuracy and smoothness for a large variety of parameter sets, model-specific parameter suggestions can be made. The quality hints (graphs, bars) and parameter suggestions are especially relevant, since the weighting factors of different smoothing methods may have different meanings. User guidance in terms of quality hints and parameter suggestions is crucial for comparing the effects of different parameters. Especially occasional users without a strong background in mesh smoothing methods benefit from the previewing functionality.

To further reduce any manual effort, the parameter suggestions were employed for generating a preview gallery of different smoothing suggestions. This is used to compare complete models or for the inspection of artifact reduction in selected areas. These tools allow for a better integration into existing model generation workflows, since the necessity of manual parameter adjustments and algorithm knowledge is reduced. Especially the inspection view has shown to be promising and might be extended to allow for a selection of multiple areas.

7

Discussion and Outlook

Table of Contents

Various medical applications employ 3D visualizations and information obtained from these visualizations as additional knowledge source, e.g., for supporting the interpretation of complicated clinical cases or for the planning of surgical interventions. Such visualizations play an increasingly important role from medical education and training up to patient information. In this context, patient-specific surface models generated from tomographic medical image data are commonly used for visually exploring the 3D data, for determining static spatial characteristics and even for simulating dynamic processes, such as blood flow and tissue behavior.

7.1 Conclusion

The presented thesis outlined these applications as prominent examples and explained typical scenarios for the usage of 3D surface models (see Chap. 2). As a result of this discussion, specific requirements to surface models have been summarized. Accuracy is commonly considered as the most important requirement. Generally speaking, 3D models should capture the relevant characteristics of an anatomic structure and the related model generation process should not alter the individual shape – neither locally, nor in a global manner. The required degree of accuracy depends on the specific application and affects, e.g., the resolution of the surface mesh. Besides accuracy, smoothness is considered as another major requirement. The demand for smooth surface models recurs in all discussed applications. 3D surfaces of anatomic structures should exhibit a natural, organic look to resemble, e.g., the intraoperative experience of surgeons. Sharp features, such as staircases, affect the visual perception in two ways: they alter the perceived shape and may even distract the viewer's visual attention from other surface areas. Moreover, smoothness is required to allow for a faithful simulation, e.g., of blood flow and as input for volume mesh generation in relation with tissue simulation. For the latter, staircase artifacts should be removed prior to mesh decimation to prevent from considering the artifacts as preservable features.

As shown in Chapter 3, the whole pipeline from data acquisition to the final model is error-prone and may yield several kinds of artifact. Artifacts in 3D surface models originate from artifacts during data acquisition via, e.g., MRI or CT. Subsequent data processing may reduce some of these image data artifacts, but as a result of segmentation, staircase artifacts may still arise in 3D surface models. Since the available smoothing methods are not generally suited for surfaces from anatomic structures, a better adaptation of existing methods to medical data is essential. In the introduction of this thesis, three major questions have been asked. To answer these questions, the presented thesis demonstrated that enormous manual effort is necessary to faithfully remove typical artifacts of medical data (see Chap. 4). As further steps towards a simplification of the artifact reduction – and thus the model generation – procedure, Chapters 5 and 6 introduced new methods which make existing mesh smoothing filters more robust and simplify their handling.

How can context information be included into the smoothing procedure for a better consideration of medical requirements and anatomic constraints?

Chapter 5 introduced the concept of context-aware smoothing by means of two examples: spatial relations in terms of Euclidean distances and the localization of artifacts are employed to locally promote or damp smoothing. The idea of this chapter is to directly include expert knowledge, e.g., derived from application-specific requirements or from the source and type of artifacts into the smoothing process.

Distance-aware smoothing allows for generating smooth surface models while the spatial relation to surrounding structures is preserved. This is especially relevant for applications which require an evaluation of inter-structure distances (e.g., surgical planning). By including such an easily determinable measure, smoothing is damped in those surface regions which get close to surrounding structures. This allows to apply comparably strong smoothing without worrying about altering spatial relations. For very strong smoothing, the non-uniform application of the smoothing filter may lead to severe surface deformations. Thus, a careful setting of smoothing parameters is still required.

While the idea of distance awareness results from application requirements, staircase awareness employs information on the origin of staircase artifacts. The human visual perception can clearly differentiate staircase artifacts from other “features” on a surface due to their orientation and shape. Chapter 5 explained how to involve this knowledge into the smoothing procedure. By explicitly searching for staircases, promoting smoothing in these artifact areas and damping smoothing in non-artifact areas, even small details can usually be maintained while staircase artifacts are reduced. Hence, staircase-aware smoothing is primarily suited for surface models suffering from local staircases. If staircases are uniformly spread over the surface, such as in models from binary segmented data, the result of staircase-aware smoothing will be similar to the one of a uniform smoothing filter.

The key benefit of context-aware smoothing (as summary of distance and staircase awareness) is the general applicability to any mesh smoothing filter. Irrespective of the specific underlying algorithm, smoothing generates a translation vector for each vertex of the mesh. Thus, the vertex weights resulting from the context awareness may easily be employed to adapt the smoothing procedure to context-dependent constraints.

How can users be supported during the use of surface smoothing filters?

Mesh smoothing is usually applied within complex model generation pipelines. Depending on the relevance of accuracy for the specific application, smoothing may be considered subordinately by using default parameters. For the applications described in Chapter 2, accuracy is usually a major requirement, such that any involvement of smoothing filters requires a careful setting of the required parameters.

By applying context-aware smoothing, standard mesh smoothing filters can be restricted to certain areas for a selective artifact reduction. Thus, the required smoothing parameters can be set less carefully which reduces the necessity of iterative parameter testing and result evaluation. The detection of staircases requires additional parameters, which can usually be left unchanged. Two scenarios can be imagined, where staircase detection fails:

- No staircases are detected: Since the context awareness would perform smoothing in artifact areas and – vice versa – leave non-artifact areas unchanged, no smoothing would be performed. Thus, no additional errors are generated.
- The whole surface is considered as staircases: In this case, smoothing would be performed allover the surface in a uniform manner according to the user-selected smoothing algorithm. Thus, the maximum amount of smoothing depends on the user-defined parameters. Hence, the result would not be aggravated by the (failed) staircase detection.

Besides reducing the need for precise parameter testing by locally restricting smoothing, Chapter 6 presented a concept and GPU-based framework for efficiently performing smoothing operations. A method for quantifying the quality of a surface model by including the volume change, local distance changes, and curvature reduction has been introduced and successfully applied. An efficient implementation allowed for applying smoothing, model evaluation, and visual feedback in real-time. In addition, the user is provided with a visualization of the expected change of model quality, if one of the available smoothing parameters is modified. Thus, the user does not have to explore the effects of parameter changes in an iterative manner (as usual). Instead, it is directly visible which change of smoothing parameters yields either more smoothness or better preserves accuracy. With the option to directly interact with the quality visualizations (graphs, bars), the visually-guided, interactive smoothing approach supports the user during understanding of the parameter effects and promotes the finding of an optimal parameter set for the considered structure.

Is there an efficient way to provide individual smoothing suggestions for patient-specific data?

Bade et al. [2006] provided smoothing suggestions for different shape-categories of anatomic structures. Visually-guided smoothing (see Chap. 6) took this idea and employed the GPU-based framework and model quality evaluation to generate model-specific smoothing suggestions. By performing an iterative optimization, a large number of parameter sets is automatically evaluated. Users can select whether they require higher accuracy or more artifact reduction. The underlying optimization process finds the parameter set which fits best to the needs of the user. Thus, parameter tuning is minimized to the target smoothness parameter which defines the tradeoff between accuracy and smoothness. Assuming that most applications require both artifact reduction and accuracy preservation, the target smoothness can usually be left unchanged at an intermediate value. As a result, the framework then determines appropriate smoothing parameters fully automatically under consideration of the individual shape of the processed anatomic structure.

Furthermore, this functionality has been used to automatically generate smoothing suggestions for different settings of the target smoothness. For each of the resulting parameter sets, smoothing has been applied and the final models are visualized to the user.

Thus, without setting any smoothing parameters, the user can select the model (and corresponding parameter set) which fits best to the individual needs.

7.2 Outlook

Context-aware and visually-guided smoothing have been successfully integrated into several workflows for different applications. However, there is potential for further research and improvement of the presented methods.

The implementation of the staircase awareness is currently not performance-critical, since computation usually took less than a second for medium-sized surface meshes and between one and two seconds for larger ones. In the context of visually-guided smoothing, however, the detection of staircases might be integrated into the GPU-based framework. Currently, staircase detection is performed on the CPU side and the location information is passed to the GPU for restricting smoothing to the artifact areas. By unifying these steps, relevant input parameters could also be optimized automatically. Moreover, context information in terms of scaled distances and staircase locations has solely been applied to smoothing algorithms. As a next step, this information might be fed into, e.g., decimation algorithms, such that a surface mesh is particularly decimated in artifact areas. For instance, the priority queue of the quadric error decimation method of Garland and Heckbert [1997] might be modified to include the staircase-dependent weighting.

The visually-guided smoothing framework consisting of the evaluation of the model quality as well as the interaction concept (quality graphs & bars, previewing functionality) might also be extended to mesh processing tasks other than smoothing (e.g., decimation, subdivision, remeshing). However, this requires a modification of the error estimation methods, since the presented method assumes that the mesh topology is equal between the compared meshes (see Sec. 6.4.1). A starting point for this might be the work of Dick et al. [2011], who determined and visualized distances for interactive preoperative implant planning.

A first step towards taking model generation to the application level was made by introducing the target smoothness parameter to allow the user to specify the tradeoff between the desired smoothness and accuracy without having to understand the underlying smoothing algorithm and parameters. This idea should be continued in the context of the whole model generation process. Thus, the user might specify the intended application for the surface model (e.g., pure visualization, CFD, rapid prototyping), and the framework adjusts the model generation pipeline (and related parameters) on its own to yield more automation and user guidance during the process. This is, in turn, related to the concept of context-aware smoothing (and model generation), since additional contextual information and constraints need to be involved.

The methods presented in the former chapters are dedicated to triangle meshes. As summarized by Bommes et al. [2012], quadrilateral meshes become increasingly important (e.g., for simulation applications), which affects the typical mesh processing tasks and requires modified or even dedicated algorithms. In the context of such generalized model generation framework, the underlying methods should also be able to cope with quadrilateral meshes. This affects not only the implementation, but also the way how model quality has to be determined.

Recapturing the various artifacts occurring in the data acquisition and model generation pipeline (see Chap. 3 and 4), it becomes clear that a widely automated model generation procedure could only be achieved if the possible artifacts could be detected automatically. This requires, however, a previous definition of possible artifacts and an appropriate generalization for handling their various occurrences. A framework considering model generation from the application side could then detect and classify the artifacts to finally perform the required application-specific mesh processing tasks. Though, this may not be possible for all kinds of artifacts, since several issues (e.g., blending artifacts in vascular surface models) may only be detected and resolved with the help of an expert or by tailoring algorithms for the considered target structure.

Bibliography

AAO-HNS. American Academy of Otolaryngology - Head and Neck Surgery (AAO-HNS) - Definition of Lymph Node Groups, August 2012. URL `http://www.entnet.org`.

S. Adler. *Entwicklung von Verfahren zur interaktiven Simulation minimal-invasiver Operationsmethoden*. PhD thesis, Otto-von-Guericke University Magdeburg, April 2014.

S. Adler, L. Dornheim, I. Rössling, J. Dornheim, M. Buser, and R. Mecke. Planungs- und Testumgebung für neue Methoden in der minimal-invasiven Chirurgie. In *9. Fachtagung - Digitales Engineering zum Planen, Testen und betreiben Technischer Systeme, Tagungsband 15. IFF-Wissenschaftstage*, volume 15, pages 329–339. Fraunhofer-Institut für Fabrikbetrieb und -automatisierung, 2012.

R. Aggarwal, P. Crochet, A. Dias, A. Misra, P. Ziprin, and A. Darzi. Development of a virtual reality training curriculum for laparoscopic cholecystectomy. *British Journal of Surgery*, 96(9):1086–1093, 2009. URL `http://dx.doi.org/10.1002/bjs.6679`.

F. Allamandri, P. Cignoni, C. Montani, and R. Scopigno. Adaptively adjusting Marching Cubes output to fit a trilinear reconstruction filter. In *Proc. of EG Workshop on Scientific Visualization*, pages 25–34, 1998. URL `http://dx.doi.org/10.1007/978-3-7091-7517-0_3`.

P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. Recent Advances in Remeshing of Surfaces. In L. De Floriani and M. Spagnuolo, editors, *Shape Analysis and Structuring*, Mathematics and Visualization, pages 53–82. Springer Berlin Heidelberg, 2008. URL `http://dx.doi.org/10.1007/978-3-540-33265-7_2`.

A. M. Alyassin, J. L. Lancaster, J. H. D. III, and P. T. Fox. Evaluation of new algorithms for the interactive measurement of surface area and volume. *Medical Physics*, 21(6): 741–752, 1994. URL `http://link.aip.org/link/?MPH/21/741/1`.

L. Antiga, B. Ene-Iordache, and A. Remuzzi. Computational geometry for patient-specific reconstruction and meshing of blood vessels from MR and CT angiography. *IEEE Trans Med Imaging*, 22(5): 674–84, 2003. URL http://www.biomedsearch.com/nih/Computational-geometry-patient-specific-reconstruction/12846436.html.

L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, and D. Steinman. An image-based modeling framework for patient-specific computational hemodynamics. *Medical and Biological Engineering and Computing*, 46:1097–1112, 2008. URL http://dx.doi.org/10.1007/s11517-008-0420-1.

A. Armillotta, P. Bonhoeffer, G. Dubini, S. Ferragina, F. Migliavacca, G. Sala, and S. Schievano. Use of rapid prototyping models in the planning of percutaneous pulmonary valved stent implantation. *Proc. of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine*, 221(4):407–16, 2007. URL http://www.biomedsearch.com/nih/Use-rapid-prototyping-models-in/17605398.html.

L. Augsburger, P. Reymond, R. Ouared, O. Brina, D. A. Rufenacht, N. Stergiopulos, and V. M. Pereira. Influence of Segmentation on Hemodynamics in Cerebral Aneurysms. In *Proc. of Conference on Modelling Fluid Flow*, pages 81–87, 2009. URL http://dx.doi.org/10.1007/s12573-012-0046-7.

I. Ayodeji, M. Schijven, J. Jakimowicz, and J. Greve. Face validation of the Simbionix LAP Mentor virtual reality training module and its applicability in the surgical curriculum. *Surgical Endoscopy*, 21(9):1641–1649, 2007. URL http://dx.doi.org/10.1007/s00464-007-9219-7.

R. Bade. *Interaktive und dynamische Visualisierung für die chirurgische Ausbildung und Interventionsplanung*. PhD thesis, Otto-von-Guericke University Magdeburg, 2007.

R. Bade, J. Haase, and B. Preim. Comparison of fundamental mesh smoothing algorithms for medical surface models. In *Proc. of Simulation und Visualisierung*, pages 289–304, 2006.

R. Bade, O. Konrad, and B. Preim. Reducing Artifacts in Surface Meshes Extracted from Binary Volumes. *Journal of WSCG*, 15(1-3):67–74, 2007a. URL http://wscg.zcu.cz/WSCG2007/Papers_2007/journal/B29-full.pdf.

R. Bade, C. Schumann, S. Seshadhri, G. Janiga, T. Bölke, Özlem Krischek, M. Skalej, G. Rose, D. Thévenin, and B. Preim. High-quality surface generation for flow simulation in cerebral aneurysms. In *Proc. of Computer- und Roboterassistierte Chirurgie (CURAC)*, pages 125–128, 2007b.

J. F. Barrett and N. Keat. Artifacts in CT: Recognition and Avoidance. *RadioGraphics*, 24(6):1679–1691, 2004. URL http://dx.doi.org/10.1148/rg.246045065.

A. Belyaev and Y. Ohtake. A Comparison of Mesh Smoothing Methods. In *In Proceedings of the Israel-Korea BiNational Conference on Geometric Modeling and Computer Graphics*, pages 83–87, 2003.

P. Berg, G. Janiga, O. Beuing, M. Neugebauer, and D. Thévenin. Hemodynamics in Multiple Intracranial Aneurysms: The Role of Shear Related to Rupture. *International Journal of Bioscience, Biochemistry and Bioinformatics*, 3(3):177–181, 2013. URL `http://dx.doi.org/10.7763/IJBBB.2013.V3.191`.

W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-Aware Exploration of Continuous Parameter Spaces Using Multivariate Prediction. *Computer Graphics Forum*, 30(3):911–920, 2011. URL `http://dx.doi.org/10.1111/j.1467-8659.2011.01940.x`.

R. Bibb and J. Winder. A review of the issues surrounding three-dimensional computed tomography for medical modelling using rapid prototyping techniques. *Radiography*, 16(1):78 – 83, 2010. URL `http://dx.doi.org/10.1016/j.radi.2009.10.005`.

G. E. Blelloch. Prefix Sums and Their Applications. In *Synthesis of Parallel Algorithms*. Morgan Kaufmann, 1993. URL `https://www.cs.cmu.edu/~guyb/papers/Ble93.pdf`.

E. Bocca, O. Pignataro, C. Oldini, and C. Cappa. Functional neck dissection: an evaluation and review of 843 cases. *Laryngoscope*, 94 (7):942–5, 1984. URL `http://www.biomedsearch.com/nih/Functional-neck-dissection-evaluation-review/6738274.html`.

H. Bogunović, J. M. Pozo, M. C. Villa-Uriol, C. B. L. M. Majoie, R. van den Berg, H. A. F. Gratama van Andel, J. M. Macho, J. Blasco, L. S. Román, and A. F. Frangi. Automated segmentation of cerebral vasculature with aneurysms in 3DRA and TOF-MRA using geodesic active regions: an evaluation study. *Med Phys*, 38(1):210–22, 2011. URL `http://www.biomedsearch.com/nih/Automated-segmentation-cerebral-vasculature-with/21361189.html`.

J. Bolz and P. Schröder. Rapid evaluation of Catmull-Clark subdivision surfaces. In *Proceedings of the seventh international conference on 3D Web technology*, Web3D '02, pages 11–17, 2002. URL `http://doi.acm.org/10.1145/504502.504505`.

J. Bolz, I. Farmer, E. Grinspun, and P. Schröder. Sparse Matrix Solvers on the GPU: Conjugate Gradients and Multigrid. *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, 22(3):917–924, Jul 2003. URL `http://dx.doi.org/10.1145/882262.882364`.

D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin. State of the Art in Quad Meshing. In *Eurographics STARS*, 2012.

M. Booshehrian, T. Möler, R. M. Peterman, and T. Munzner. Vismon: Facilitating Analysis of Trade-Offs, Uncertainty, and Sensitivity In Fisheries Management Decision Making. *Computer Graphics Forum*, 31(3pt3):1235–1244, 2012. URL `http://dx.doi.org/10.1111/j.1467-8659.2012.03116.x`.

M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. *Polygon Mesh Processing*. AK Peters, 2010.

L. Boussel, V. Rayz, A. Martin, G. Acevedo-Bolton, M. T. Lawton, R. Higashida, W. S. Smith, W. L. Young, and D. Saloner. Phase-contrast magnetic resonance imaging measurements in intracranial aneurysms in vivo of flow patterns, velocity fields, and wall shear stress: Comparison with computational fluid dynamics. *Magnetic Resonance in Medicine*, 61(2):409–417, 2009. URL `http://dx.doi.org/10.1002/mrm.21861`.

O. Bradac, S. Hide, D. Mendelow, and V. Benes. Aneurysm treatment in Europe 2010: an internet survey. *Acta Neurochirurgica*, 154:971–978, 2012. URL `http://dx.doi.org/10.1007/s00701-012-1340-2`.

I. Braude, J. Marker, K. Museth, J. Nissanov, and D. Breen. Contour-based surface reconstruction using MPU implicit models. *Graphical Models*, 69(2):139–157, March 2007. URL `http://dx.doi.org/10.1016/j.gmod.2006.09.007`.

J. L. Brisman, J. K. Song, and D. W. Newell. Cerebral Aneurysms. *New England Journal of Medicine*, 355(9):928–939, 2006. URL `http://dx.doi.org/10.1056/NEJMra052760`.

C. E. Broelsch, J. C. Emond, P. F. Whitington, J. R. Thistlethwaite, A. L. Baker, and J. L. Lichtor. Application of reduced-size liver transplants as split grafts, auxiliary orthotopic grafts, and living related segmental transplants. *Ann Surg*, 212(3):368–75; discussion 375–7, 1990. URL `http://www.ncbi.nlm.nih.gov/pubmed/2396888`.

S. Bruckner and T. Möller. Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1467–1475, Oct. 2010. URL `http://dx.doi.org/10.1109/TVCG.2010.190`.

P. W. D. Bruin, F. M. Vos, F. H. Post, S. F. Frisken-Gibson, and A. M. Vossepoel. Improving Triangle Mesh Quality with SurfaceNets. In *Proc. of MICCAI*, pages 804–813, 2000. URL `http://dx.doi.org/10.1007/978-3-540-40899-4_83`.

E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350 – 355, 1978. URL `http://dx.doi.org/10.1016/0010-4485(78)90110-0`.

J. R. Cebral and R. Löhner. From medical images to anatomically accurate finite element grids. *International Journal for Numerical Methods in Engineering*, 51:985–1008, 2001. URL `http://dx.doi.org/10.1002/nme.205`.

J. R. Cebral, R. Löhner, O. Soto, P. L. Choyke, and P. J. Yim. Patient-Specific Simulation of Carotid Artery Stenting Using Computational Fluid Dynamics. In *Proc. of MICCAI*, pages 153–160, 2001. URL `http://dx.doi.org/10.1007/3-540-45468-3_19`.

J. R. Cebral, M. A. Castro, S. Appanaboyina, C. M. Putman, D. Millan, and A. F. Frangi. Efficient pipeline for image-based patient-specific analysis of cerebral aneurysm hemodynamics: technique and sensitivity. *IEEE Trans. Med. Imaging*, 24(4):457–467, 2005. URL `http://www.ncbi.nlm.nih.gov/pubmed/15822804`.

J. R. Cebral, F. Mut, J. Weir, and C. M. Putman. Association of hemodynamic characteristics and cerebral aneurysm rupture. *AJNR Am J Neuroradiol*, 32(2):264–70, 2011. URL `http://dx.doi.org/10.3174/ajnr.A2274`.

Y.-H. Chan, C. Correa, and K.-L. Ma. Flow-based scatterplots for sensitivity analysis. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 43 –50, 2010. URL `http://dx.doi.org/10.1109/VAST.2010.5652460`.

R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon. *Parallel programming in OpenMP*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron. A performance study of general-purpose applications on graphics processors using CUDA. *J. Parallel Distrib. Comput.*, 68:1370–1380, 2008. URL `http://dx.doi.org/10.1016/j.jpdc.2008.05.014`.

C.-Y. Chen, M.-S. Su, and K.-Y. Cheng. An Adaptive Diffusion Filtering Method for Polygon Meshes Smoothing. In *Proc. of Conference on Computer Vision, Graphics, and Image Processing (CVGIP 2003)*, 2003.

A. Chica, J. Williams, C. Andujar, P. Brunet, I. Navazo, J. Rossignac, and A. Vinacua. Pressing: Smooth Isosurfaces with Flats from Binary Grids. *Computer Graphics Forum*, 27(1):36–46, 2008. URL `http://dx.doi.org/10.1111/j.1467-8659.2007.01039.x`.

P. Cignoni, F. Ganovelli, C. Montani, and R. Scopigno. Reconstruction of topologically correct and adaptive trilinear isosurfaces. *Computers & Graphics*, 24(3):399–418, 2000. URL `http://dblp.uni-trier.de/db/journals/cg/cg24.html#CignoniGMS00`.

J. Cohen and M. Garland. Solving Computational Problems with GPU Computing. *Computing in Science and Engineering*, 11:58–63, 2009. URL `http://dx.doi.org/10.1109/MCSE.2009.144`.

C. Couinaud. *Le Foie: Études anatomiques et chirurgicales [The Liver: Anatomical and Surgical Studies]*. Masson, 1957.

J. Crossingham, J. Jenkinson, N. Woolridge, S. Gallinger, G. Tait, and C.-A. Moulton. Interpreting three-dimensional structures from two-dimensional images: A web-based interactive 3D teaching model of surgical liver anatomy. *Official Journal of The International Hepato-Pancreato-Biliary Association*, 11(6):523–528, 2009. URL `http://dx.doi.org/10.1111/j.1477-2574.2009.00097.x`.

S. Currie, K. Mankad, and A. Goddard. Endovascular treatment of intracranial aneurysms: review of current practice. *Postgraduate Medical Journal*, 2010. URL `http://dx.doi.org/10.1136/pgmj.2010.105387`.

L. Dagum and R. Menon. OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Comput. Sci. Eng.*, 5:46–55, 1998. URL `http://dx.doi.org/10.1109/99.660313`.

M. de Bruijne, B. van Ginneken, W. J. Niessen, J. B. A. Maintz, and M. A. Viergever. Active-shape-model-based segmentation of abdominal aortic aneurysms in CTA images. *Proc. of SPIE Medical Imaging*, pages 463–474, 2002. URL `http://dx.doi.org/10.1117/12.467188`.

M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 317–324, 1999. URL `http://dx.doi.org/10.1145/311535.311576`.

C. Dick, R. Burgkart, and R. Westermann. Distance Visualization for Interactive 3D Implant Planning. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE Visualization 2011)*, 17(12):2173–2182, 2011. URL `http://dx.doi.org/10.1109/TVCG.2011.189`.

D. Doo and M. Sabin. Behavior of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, 1978. URL `http://dx.doi.org/10.1016/0010-4485(78)90111-2`.

J. Dornheim, H. Seim, B. Preim, I. Hertel, and G. Strauss. Segmentation of Neck Lymph Nodes in CT Datasets with Stable 3D Mass-Spring Models. *Academic Radiology*, pages 1389–1399, 2007. URL `http://dx.doi.org/10.1016/j.acra.2007.09.001`.

L. Dornheim, J. Dornheim, and K. D. Tönnies. Automatic generation of dynamic 3d models for medical segmentation tasks. In *Proc. of SPIE Medical Imaging*, 2006. URL `http://dx.doi.org/10.1117/12.654153`.

L. Dornheim, J. Dornheim, and I. Rössling. Complete fully automatic model-based segmentation of normal and pathological lymph nodes in CT data. *Journal of Computer-Assisted Radiology and Surgery*, 5(6):565–581, 2010. URL `http://dx.doi.org/10.1007/s11548-010-0530-8`.

M. J. Dürst. Letters: Additional Reference to Marching Cubes. *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, 22(2):72–73, 1988.

P. Du, R. Weber, P. Luszczek, S. Tomov, G. Peterson, and J. Dongarra. From {CUDA} to opencl: Towards a performance-portable solution for multi-platform {GPU} programming. *Parallel Computing*, 38(8):391 – 407, 2012. URL `http://dx.doi.org/10.1016/j.parco.2011.10.002`.

P. S. D'Urso, D. J. Effeney, W. J. Earwaker, T. M. Barker, M. J. Redmond, R. G. Thompson, and F. H. Tomlinson. Custom cranioplasty using stereolithography and acrylic. *British Journal of Plastic Surgery*, 53(3):200 – 204, 2000. URL `http://dx.doi.org/10.1054/bjps.1999.3268`.

C. Dyken, G. Ziegler, C. Theobalt, and H.-P. Seidel. High-speed Marching Cubes using HistoPyramids. *Computer Graphics Forum*, 27(8):2028–2039, 2008. URL `http://dx.doi.org/10.1111/j.1467-8659.2008.01182.x`.

L. Erasmus, D. Hurter, M. Naude, H. Kritzinger, and S. Acho. A short overview of MRI artefacts. *South African Journal of Radiology*, 8(2), 2004. URL `http://hdl.handle.net/10520/EJC66512`.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, 1996.

S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, SIGGRAPH '03, pages 950–953. ACM, 2003. URL `http://doi.acm.org/10.1145/1201775.882368`.

A. Frangi, W. Niessen, R. Hoogeveen, T. Van Walsum, and M. Viergever. Model-based quantitation of 3-D magnetic resonance angiographic images. *Medical Imaging, IEEE Transactions on*, 18(10):946–956, 1999. URL `http://dx.doi.org/10.1109/42.811279`.

M. Garland. Multiresolution modeling: Survey & future opportunities. In *Eurographics '99 – State of the Art Reports*, pages 111–131, 1999. URL `http://mgarland.org/files/papers/STAR99.pdf`.

M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 209–216, 1997. URL `http://dx.doi.org/10.1145/258734.258849`.

R. Gasteiger, M. Neugebauer, C. Kubisch, and B. Preim. Adapted Surface Visualization of Cerebral Aneurysms with Embedded Blood Flow Information. In *Eurographics Workshop on Visual Computing for Biology and Medicine (EG VCBM)*, pages 25–32, 2010.

German Cancer Research Center (DKFZ). Overview of Cancer Mortality, 2012. URL `http://www.dkfz.de/en/krebsatlas/index.html`.

I. Gibson, L. Cheung, S. Chow, W. Cheung, S. Beh, M. Savalani, and S. Lee. The use of rapid prototyping to assist medical applications. *Rapid Prototyping Journal*, 12(1): 53–58, 2006. URL `http://dx.doi.org/10.1108/13552540610637273`.

I. I. Gibson, D. W. Rosen, and B. Stucker. *Additive manufacturing technologies : rapid prototyping to direct digital manufacturing*. Springer, 2010. URL http://ci.nii.ac.jp/ncid/BB04858677.

S. F. F. Gibson. Constrained Elastic Surface Nets: Generating Smooth Surfaces from Binary Segmented Data. In *Proc. of MICCAI*, pages 888–898, 1998. URL http://dl.acm.org/citation.cfm?id=646921.709482.

A. Gillams, A. Cassoni, G. Conway, and W. Lees. Radiofrequency ablation of neuroendocrine liver metastases—the Middlesex experience. *Abdominal Imaging*, 30:435–441, 2005. URL http://dx.doi.org/10.1007/s00261-004-0258-4.

J. Goldfeather and V. Interrante. Understanding errors in approximating principal direction vectors. *ACM Transactions on Graphics*, 2003.

C. M. Grimm and J. F. Hughes. Smooth iso-surface approximation. In *In Proceedings of Implicit Surfaces '95*, pages 57–67, 1995.

H. Hahn, B. Preim, D. Selle, and H.-O. Peitgen. Visualization and Interaction Techniques for the Exploration of Vascular Structures. In *IEEE Visualization*, pages 395–402, Oktober 2001. URL http://dx.doi.org/10.1109/VISUAL.2001.964538.

C. Hansen. *Software Assistance for Preoperative Risk Assessment and Intraoperative Support in Liver Resection Surgery*. PhD thesis, School of Engineering and Science, Jacobs University Bremen, 2012. URL http://www.jacobs-university.de/phd/files/phd20120510_hansen.pdf.

M. Harris, S. Sengupta, and J. D. Owens. Parallel Prefix Sum (Scan) with CUDA. In H. Nguyen, editor, *GPU Gems 3*, chapter 39, pages 851–876. Addison Wesley, August 2007. URL http://http.developer.nvidia.com/GPUGems3/gpugems3_ch39.html.

H. C. Hege, M. Seebaß, D. Stalling, and M. Zöckler. A generalized marching cubes algorithm based on non-binary classifications. Technical report, 1997. URL http://nbn-resolving.de/urn:nbn:de:0297-zib-2741.

G. T. Herman. *Fundamentals of Computerized Tomography Image Reconstruction from Projections*. Springer Publishing Company, Incorporated, 2nd edition, 2009.

K. Hildebrandt and K. Polthier. Anisotropic Filtering of Non-Linear Surface Features. *Computer Graphics Forum*, 23(3):391–400, 2004. URL http://dx.doi.org/10.1111/j.1467-8659.2004.00770.x.

K. Hildebrandt and K. Polthier. Constraint-based fairing of surface meshes. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 203–212. Eurographics Association, 2007. URL http://portal.acm.org/citation.cfm?id=1281991.1282019.

R. Hompes, S. Fieuws, R. Aerts, M. Thijs, F. Penninckx, and B. Topal. Results of single-probe microwave ablation of metastatic liver cancer. *European Journal of Surgical Oncology (EJSO)*, 36(8):725 – 730, 2010. URL `http://dx.doi.org/10.1016/j.ejso.2010.05.013`.

H. Hoppe. Progressive Meshes. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 99–108, 1996. URL `http://dx.doi.org/10.1145/237170.237216`.

A. Jalba and J. B. T. M. Roerdink. Efficient Surface Reconstruction From Noisy Data Using Regularized Membrane Potentials. *Image Processing, IEEE Transactions on*, 18(5):1119–1134, 2009. URL `http://dx.doi.org/10.1109/TIP.2009.2016141`.

A. C. Jalba and J. B. T. M. Roerdink. Efficient Surface Reconstruction from Noisy Data Using Regularized Membrane Potentials. In *Proceedings of the Eighth Joint Eurographics / IEEE VGTC Conference on Visualization*, EUROVIS'06, pages 83–90. Eurographics Association, 2006. URL `http://dx.doi.org/10.2312/VisSym/EuroVis06/083-090`.

G. Janiga, C. Rössl, M. Skalej, and D. Thévenin. Realistic virtual intracranial stenting and computational fluid dynamics for treatment analysis. *Journal of Biomechanics*, 46(1):7–12, 2012. URL `http://dx.doi.org/10.1016/j.jbiomech.2012.08.047`.

T. J. Jankun-Kelly and K.-L. Ma. Visualization Exploration and Encapsulation via a Spreadsheet-Like Interface. *IEEE TVCG*, 7:275–287, 2001. URL `http://dx.doi.org/10.1109/2945.942695`.

M. Joffe, F. Harris, S. Kahugu, A. Nicoll, and J. Linney, R.and Richards. A prospective study of computer-aided design and manufacture of titanium plate for cranioplasty and its clinical outcome. *British Journal of Neurosurgery*, 13(6):576–580, 1999. URL `http://informahealthcare.com/doi/abs/10.1080/02688699943088`.

T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, 22(3):943–949, 2003. URL `http://dx.doi.org/10.1145/882262.882367`.

P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, SIGGRAPH '07. ACM, 2007. URL `http://doi.acm.org/10.1145/1275808.1276466`.

K. Karimi, N. G. Dickson, and F. Hamze. A Performance Comparison of CUDA and OpenCL. *CoRR*, abs/1005.2581, 2010. URL `http://arxiv.org/abs/1005.2581`.

J. Kessenich, D. Baldwin, and R. Rost. The OpenGL Shading Language - Version 4.10, 2010.

U. G. Kühnapfel, H. K. Çakmak, and H. Maaß. Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computers and Graphics*, 24(5): 671–682, 2000. URL http://dx.doi.org/10.1016/S0097-8493(00)00070-4.

KHRONOS Group. The OpenCL Specification - Version: 1.1, 2010.

K. Knox, C. W. Kerber, S. Singel, M. Bailey, and S. Imbesi. Rapid prototyping to create vascular replicas from CT scan data: Making tools to teach, rehearse, and choose treatment strategies. *Catheterization and Cardiovascular Interventions*, 65(1):47–53, 2005. URL http://dx.doi.org/10.1002/ccd.20333.

P. M. Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, 39(3):217–241, 2003. URL http://dx.doi.org/10.1016/S0168-874X(02)00070-7.

L. P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel. Feature sensitive surface extraction from volume data. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 57–66, 2001. URL http://dx.doi.org/10.1145/383259.383265.

P. Köpf-Maier. *Wolf-Hedeggers Atlas of Human Anatomie- Head and -neck, Thorax, Abdomen, Pelvis, CNS, Eye, Ear.* 5th completely revised and supplemented Edition. Karger, Freiburg i.B., 2000.

J. Kretschmer, T. Beck, C. Tietjen, B. Preim, and M. Stamminger. Reliable Adaptive Modelling of Vascular Structures with Non-Circular Cross-Sections. *Computer Graphics Forum (EuroVis)*, 31 (3):1055–1064, 2012. URL http://dx.doi.org/10.1111/j.1467-8659.2012.03098.x.

J. Kretschmer, C. Godenschwager, B. Preim, and M. Stamminger. Interactive Patient-Specific Vascular Modeling with Sweep Surfaces. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* , 19(12):2828–2837, 2013. URL http://dx.doi.org/10.1109/TVCG.2013.169.

F. Labelle and J. R. Shewchuk. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, page 57. ACM, 2007. URL http://dx.doi.org/10.1145/1276377.1276448.

S. Lakare and A. E. Kaufman. Anti-Aliased Volume Extraction. In *Proc. of Eurographics/IEEE Symposium on Visualization*, 2003.

W. Lamade, G. Glombitza, L. Fischer, P. Chiu, C. E. Cárdenas, M. Thorn, H. P. Meinzer, L. Grenacher, H. Bauer, T. Lehnert, and C. Herfarth. The impact of 3-dimensional reconstructions on operation planning in liver surgery. *Archives of surgery (Chicago, Ill. : 1960)*, 135(11):1256 – 61, 2000. URL http://www.ncbi.nlm.nih.gov/pubmed/11074877.

H. Lang and A. Schenk. *Computerassistierte Chirurgie*, chapter Planung von Leberresektionen, pages 515–524. Urban & Fischer Verlag/Elsevier GmbH, 2010.

K. Lawonn, R. Gasteiger, and B. Preim. Adaptive Surface Visualization of Vessels with Embedded Blood Flow Based on the Suggestive Contour Measure. In M. Bronstein, J. Favre, and K. Hormann, editors, *VMV 2013 - Vision, Modeling, Visualization*, pages 113–120, September 2013a. URL `http://dx.doi.org/10.2312/PE.VMV.VMV13.113-120`.

K. Lawonn, T. Mönch, and B. Preim. Streamlines for Illustrative Real-time Rendering. *Computer Graphics Forum*, 33(3):321–330, 2013b. URL `http://dx.doi.org/10.1111/cgf.12119`.

K. Lawonn, R. Gasteiger, and B. Preim. Adaptive surface visualization of vessels with animated blood flow. *Computer Graphics Forum*, pages n/a–n/a, 2014. URL `http://dx.doi.org/10.1111/cgf.12355`.

K. S. Lehmann, J. Gröne, J. C. Lauscher, J.-P. P. Ritz, C. Holmer, U. Pohlen, and H.-J. J. Buhr. Simulation training in surgical education - application of virtual reality laparoscopic simulators in a surgical skills course. *Zentralblatt für Chirurgie*, 137(2):130–137, April 2012. URL `http://dx.doi.org/10.1055/s-0031-1283984`.

P. Lermusiaux, C. Leroux, J. C. Tasse, L. Castellani, and R. Martinez. Aortic Aneurysm: Construction of a Life-size Model by Rapid Prototyping. *Annals of Vascular Surgery*, 15(2):131 – 135, 2001. URL `http://dx.doi.org/10.1007/s100160010054`.

D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea. A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis*, 13(6):819 – 845, 2009. URL `http://dx.doi.org/10.1016/j.media.2009.07.011`.

H. Levkowitz. *Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications*. Kluwer Academic Publishers, 1997.

B. Liu, B. Wuensche, and T. Ropinski. Visualization by Example - A Constructive Visual Component-Based Interface for Direct Volume Rendering. In *Proc of. Computer Graphics Theory and Applications*, 2010. URL `http://viscg.uni-muenster.de/publications/2010/LWR10`.

C. Logge, J. Cordes, K. Mühler, B. Preim, and C. Hillert. Training in Living Donor Liver Transplantation - The LiverSurgeryTrainer. In *ILTS 13th Annual International Congress*, 2007.

C. Loop. Smooth Subdivision Surfaces Based on Triangles. Master's thesis, University of Utah, 1987.

W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 163–169, 1987. URL `http://dx.doi.org/10.1145/37401.37422`.

D. P. Luebke. A Developer's Survey of Polygonal Simplification Algorithms. *IEEE Comput. Graph. Appl.*, 21(3):24–35, 2001. URL `http://dx.doi.org/10.1109/38.920624`.

M. Gurr and R. Mülhaupt. 8.04 - Rapid Prototyping. In *Polymer Science: A Comprehensive Reference*, pages 77 – 99. Elsevier, 2012. URL `http://dx.doi.org/10.1016/B978-0-444-53349-4.00202-8`.

J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 389–400, 1997. URL `http://dx.doi.org/10.1145/258734.258887`.

H.-P. Meinzer, M. Thorn, and C. Cárdenas. Computerized planning of liver surgery—an overview. *Computers & Graphics*, 26:569–576, 2002. URL `http://dx.doi.org/10.1016/S0097-8493(02)00102-4`.

M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, 2003. URL `http://dx.doi.org/10.1007/978-3-662-05105-4_2`.

J. Mönch, K. Mühler, C. Hansen, K.-J. Oldhafer, G. Stavrou, C. Hillert, C. Logge, and B. Preim. The LiverSurgeryTrainer: training of computer-based planning in liver resection surgery. *International Journal of Computer Assisted Radiology and Surgery*, January, 2013a. URL `http://dx.doi.org/10.1007/s11548-013-0812-z`.

T. Mönch, S. Adler, P. Hahn, I. Rössling, and B. Preim. Distance-Aware Smoothing of Surface Meshes for Surgical Planning. In *Proc. of the 1st International Workshop on Digital Engineering*, pages 45–51. ACM, June 2010a. URL `http://dx.doi.org/10.1145/1837154.1837161`.

T. Mönch, S. Adler, and B. Preim. Staircase-Aware Smoothing of Medical Surface Meshes. In *Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)*, pages 83–90, 2010b. URL `http://dx.doi.org/10.2312/VCBM/VCBM10/083-090`.

T. Mönch, M. Neugebauer, P. Hahn, and B. Preim. Generation of Smooth and Accurate Surface Models for Surgical Planning and Simulation. In *Proc. of SPIE Medical Imaging*, February 2010c. URL `http://dx.doi.org/10.1117/12.844235`.

T. Mönch, R. Gasteiger, G. Janiga, H. Theisel, and B. Preim. Context-Aware Mesh Smoothing for Biomedical Applications. *Computers & Graphics*, 35(4):755 – 767, 2011a. URL `http://dx.doi.org/10.1016/j.cag.2011.04.011`.

T. Mönch, M. Neugebauer, and B. Preim. Optimization of Vascular Surface Models for Computational Fluid Dynamics and Rapid Prototyping. In *Proc. of the 2nd International Workshop on Digital Engineering*, pages 16–23. Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, 2011b. Technical Report FIN-010-2011, ISSN 1869-5078.

T. Mönch, C. Kubisch, K. Lawonn, R. Westermann, and B. Preim. Visually Guided Mesh Smoothing for Medical Applications. In *VCBM 2012 - Eurographics Workshop on Visual Computing for Biology and Medicine*, pages 91–98, September 2012. URL `http://dx.doi.org/10.2312/VCBM/VCBM12/091-098`.

T. Mönch, K. Lawonn, C. Kubisch, R. Westermann, and B. Preim. Interactive Mesh Smoothing for Medical Applications. *Computer Graphics Forum*, 32(8):110–121, 2013b. URL `http://dx.doi.org/10.1111/cgf.12165`.

N. Molino, R. Bridson, and R. Fedkiw. Tetrahedral mesh generation for deformable bodies. In *Proc. Symposium on Computer Animation*, 2003.

A. Myles, T. Ni, and J. Peters. Fast parallel construction of smooth surfaces from meshes with tri/quad/pent facets. In *Proc. of the Symposium on Geometry Processing*, SGP '08, pages 1365–1372, 2008. URL `http://dl.acm.org/citation.cfm?id=1731309.1731318`.

A. Neubauer, M.-T. Forster, R. Wegenkittl, L. Mroz, and K. Bühler. Efficient Display of Background Objects for Virtual Endoscopy using Flexible First-Hit Ray Casting. In *Proc. of Eurographics/IEEE TVCG Symposium on Visualization*, pages 301–310, 352, 2004. URL `http://dx.doi.org/10.2312/VisSym/VisSym04/301-310`.

M. Neugebauer. *Computer-Aided Exploration of Blood Flow in Cerebral Aneurysms - Geometrical Processing and Interactive Visualization*. PhD thesis, Otto-von-Guericke University Magdeburg, 2014.

M. Neugebauer, G. Janiga, S. Zachow, M. Skalej, and B. Preim. Generierung qualitativ hochwertiger Modelle für die Simulation von Blutfluss in zerebralen Aneurysmen. In *Proc. Simulation and Visualization 2008*, pages 221–236, 2008.

M. Neugebauer, R. Gasteiger, O. Beuing, V. Diehl, M. Skalej, and B. Preim. Map Displays for the Analysis of Scalar Data on Cerebral Aneurysm Surfaces. In *Computer Graphics Forum (EuroVis)*, volume 28 (3), pages 895–902, Berlin, 10.-12. Juni 2009. URL `http://dx.doi.org/10.1111/j.1467-8659.2009.01459.x`.

T. Ni, Y. I. Yeo, A. Myles, V. Goel, and J. Peters. GPU smoothing of quad meshes. In *Shape Modeling International*, pages 3–9, 2008. URL `http://dx.doi.org/10.1109/SMI.2008.4547938`.

G. M. Nielson. Dual Marching Cubes. In *Proc. of IEEE Visualization*, pages 489–496, 2004. URL `http://dx.doi.org/10.1109/VISUAL.2004.28`.

G. M. Nielson and B. Hamann. The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes. In *Proceedings of the 2Nd Conference on Visualization '91*, VIS '91, pages 83–91. IEEE Computer Society Press, 1991. URL `http://dl.acm.org/citation.cfm?id=949607.949621`.

NVIDIA. NVIDIA CUDA$^{\text{TM}}$Programming Guide Version 3.0, 2010a.

NVIDIA. NVIDIA CUDA$^{\text{TM}}$C Best Practices Guide Version 3.2, 2010b.

Objet Ltd. and UCLA Medical Center. Case Study - Conjoined twins' operation planned using the Objet QuadraTempo$^{\text{TM}}$ RP System, August 2012. URL `http://www.mimos.it/chirurgiavirtuale10/11_Abstract_Langfeld.pdf`.

S. Oeltze and B. Preim. Visualization of vasculature with convolution surfaces: method, validation and evaluation. *IEEE Transactions on Medical Imaging*, 24(4):540–548, April 2005. URL `http://dx.doi.org/10.1109/TMI.2004.843196`.

Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 463–470. ACM, 2003. URL `http://dx.doi.org/10.1145/1201775.882293`.

J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. Lefohn, and T. J. Purcell. A Survey of General-Purpose Computation on Graphics Hardware. *Computer Graphics Forum*, 26(1):80–113, 2007. URL `http://dx.doi.org/10.1111/j.1467-8659.2007.01012.x`.

J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips. GPU Computing. *Proceedings of the IEEE*, 96(5):879–899, May 2008. URL `http://dx.doi.org/10.1109/JPROC.2008.917757`.

P. P. Pébay and T. J. Baker. A Comparison Of Triangle Quality Measures. In *Proceedings of the National Academy of Sciences*, pages 9653–9657. Plenum Press, 1991.

P. Pereira. Actual role of radiofrequency ablation of liver metastases. *European Radiology*, 17:2062–2070, 2007. URL `http://dx.doi.org/10.1007/s00330-007-0587-0`.

R. Petzold, H.-F. Zeilhofer, and W. Kalender. Rapid prototyping technology in medicine—basics and applications. *Computerized Medical Imaging and Graphics*, 23(5):277 – 284, 1999. URL `http://dx.doi.org/10.1016/S0895-6111(99)00025-7`.

D. Pham and R. Gault. A comparison of rapid prototyping technologies. *International Journal of Machine Tools and Manufacture*, 38(10–11):1257 – 1287, 1998. URL `http://dx.doi.org/10.1016/S0890-6955(97)00137-5`.

M. Piccinelli, A. Veneziani, D. A. Steinman, A. Remuzzi, and L. Antiga. A Framework for Geometric Analysis of Vascular Structures: Application to Cerebral Aneurysms. *IEEE Transactions on Medical Imaging*, 28(8):1141–1155, August 2009. URL `http://dx.doi.org/10.1109/TMI.2009.2021652`.

E. Praun, H. Hoppe, M. Webb, and A. Finkelstein. Real-Time Hatching. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 579–584, 2001. URL `http://dx.doi.org/10.1145/383259.383328`.

B. Preim and D. Bartz. *Visualization in Medicine - Theory, Algorithms, and Applications*. Morgan-Kaufmann-Verlag, 2007.

B. Preim and C. Botha. *Visual Computing for Medicine: Theory, Algorithms, and Applications*. Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Elsevier Science & Technology Books, 2013. URL `http://books.google.de/books?id=sonOnAEACAAJ`.

B. Preim, H. Bourquain, D. Selle, Heinz-Otto-Peitgen, and K. J. Oldhafer. Resection Proposals for Oncologic Liver Surgery based on Vascular Territories. In *Computer-Assisted Radiology and Surgery (CARS)*. Springer, Juni 2002. URL `http://dx.doi.org/10.1007/978-3-642-56168-9_58`.

A. Rassineux. Generation and optimization of tetrahedral meshes by advancing front technique. *International Journal for Numerical Methods in Engineering*, 41(12): 651–674, 1998. URL `http://dx.doi.org/10.1002/nme.840`.

S. Raya and J. Udupa. Shape-based interpolation of multidimensional objects. *IEEE Transactions on Medical Imaging*, 9(3):32–42, 1990. URL `http://dx.doi.org/10.1109/42.52980`.

A. Razdan and M. Bae. Curvature Estimation Scheme for Triangle Meshes Using Biquadratic Bézier Patches. *Comput. Aided Des.*, 37(14):1481–1491, December 2005. URL `http://dx.doi.org/10.1016/j.cad.2005.03.003`.

C. Rieder, T. Kröger, C. Schumann, and H. K. Hahn. GPU-Based Real-Time Approximation of the Ablation Zone for Radiofrequency Ablation. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization 2011)*, 17(12):1812–1821, December 2011. URL `http://dx.doi.org/10.1109/TVCG.2011.207`.

D. Rossinelli, C. Conti, and P. Koumoutsakos. Mesh-particle interpolations on graphics processing units and multicore central processing units. *Philos Transact A Math Phys Eng Sci*, 369(1944):2164–75, 2011. URL `http://dx.doi.org/10.1098/rsta.2011.0074`.

S. Rusinkiewicz. Estimating Curvatures and Their Derivatives on Triangle Meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, September 2004. URL `http://dx.doi.org/10.1109/3DPVT.2004.54`.

S. Rusinkiewicz, F. Cole, D. DeCarlo, and A. Finkelstein. Line drawings from 3D models. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 39:1–39:356, July 2008. URL `http://dx.doi.org/10.1145/1401132.1401188`.

D. Schmauss, C. Schmitz, A. K. Bigdeli, S. Weber, N. Gerber, A. Beiras-Fernandez, F. Schwarz, C. Becker, C. Kupatt, and R. Sodian. Three-Dimensional Printing of Models for Preoperative Planning and Simulation of Transcatheter Valve Replacement. *The Annals of Thoracic Surgery*, 93(2):e31–e33, 2012. URL `http://dx.doi.org/10.1016/j.athoracsur.2011.09.031`.

J. Schöberl. NETGEN - An advancing front 2D/3D-mesh generator based on abstract rules. In *Computing and Visualization in Science*, volume 1, pages 41–52, 1997. URL `http://dx.doi.org/10.1007/s007910050004`.

W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, 26:65–70, July 1992. URL `http://dx.doi.org/10.1145/133994.134010`.

W. J. Schroeder, K. Martin, and B. Lorensen. *The Visualization toolkit : an object-oriented approach to 3D graphics*. Kitware, Inc., 2004.

C. Schumann, S. Oeltze, R. Bade, and B. Preim. Model-free Surface Visualization of Vascular Trees. In *IEEE/Eurographics Symposium on Visualization, Eurographics*, pages 283–290, 2007. URL `http://dx.doi.org/10.2312/VisSym/EuroVis07/283-290`.

C. Schumann, M. Neugebauer, R. Bade, B. Preim, and H.-O. Peitgen. Implicit Vessel Surface Reconstruction for Visualization and Simulation. In *International Journal of Computer Assisted Radiology and Surgery (IJCARS)*, volume 2(5), pages 275–286, 2008. URL `http://dx.doi.org/10.1007/s11548-007-0137-x`.

M. Schwenke, A. Hennemuth, B. Fischer, and O. Friman. Blood flow computation in phase-contrast MRI by minimal paths in anisotropic media. In *Proceedings of the 14th international conference on Medical image computing and computer-assisted intervention - Volume Part I*, MICCAI'11, pages 436–443. Springer-Verlag, 2011. URL `http://dl.acm.org/citation.cfm?id=2044656.2044717`.

M. Segal and K. Akeley. OpenGL 4.3 Core Specification, August 2012. URL `http://www.opengl.org/registry/doc/glspec43.core.20120806.pdf`.

D. Selle, B. Preim, A. Schenk, and H.-O. Peitgen. Analysis of vasculature for liver surgical planning. *IEEE Transactions on Medical Imaging*, 21(11):1344 –1357, 2002. URL `http://dx.doi.org/10.1109/TMI.2002.801166`.

S. Seshadhri, G. Janiga, O. Beuing, M. Skalej, and D. Thévenin. Impact of stents and flow diverters on hemodynamics in idealized aneurysm models. *J Biomech Eng*, 133 (7):071005, 2011. URL `http://dx.doi.org/10.1115/1.4004410`.

J. F. Shepherd and C. R. Johnson. Hexahedral Mesh Generation Constraints. *Eng. with Comput.*, 24(3):195–213, June 2008. URL `http://dx.doi.org/10.1007/s00366-008-0091-4`.

L.-J. Shiue, I. Jones, and J. Peters. A realtime GPU subdivision kernel. *ACM Trans. Graph.*, 24:1010–1015, 2005. URL `http://dx.doi.org/10.1145/1073204.1073304`.

D. Sibbing, H.-C. Ebke, K. I. Esser, and L. Kobbelt. Topology Aware Quad Dominant Meshing for Vascular Structures. In *Proceedings of the 2012 International Conference on Mesh Processing in Medical Image Analysis*, MeshMed'12, pages 147–158. Springer-Verlag, 2012. URL `http://dx.doi.org/10.1007/978-3-642-33463-4_15`.

L. Soler and J. Marescaux. Patient-specific surgical simulation. *World journal of surgery*, 32(2):208–212, February 2008. URL `http://dx.doi.org/10.1007/s00268-007-9329-3`.

J. E. Stone, D. Gohara, and G. Shi. OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems. *Computing in Science and Engineering*, 12: 66–73, 2010. URL `http://dx.doi.org/10.1109%2FMCSE.2010.69`.

V. Surazhsky, P. Alliez, and C. Gotsman. Isotropic Remeshing of Surfaces: a Local Parameterization Approach. Technical Report RR-4967, INRIA, 2003. URL `http://hal.inria.fr/inria-00071612`.

L. Sutherland, P. Middleton, A. Anthony, J. Hamdorf, P. Cregan, D. Scott, and G. Maddern. Surgical simulation: a systematic review. *Ann Surg*, 243(3):291–300, 2006. URL `http://www.ncbi.nlm.nih.gov/pubmed/16495690`.

K. Tanaka, H. Shimada, Y. Nagano, I. Endo, H. Sekido, and S. Togo. Outcome after hepatic resection versus combined resection and microwave ablation for multiple bilobar colorectal metastases to the liver. *Surgery*, 139(2):263 – 273, 2006. URL `http://dx.doi.org/10.1016/j.surg.2005.07.036`.

T. Tasdizen and R. Whitaker. Anisotropic diffusion of surface normals for feature preserving surface reconstruction. In *Proc. of 3-D Imaging and Modeling*, pages 353–360, 2003. URL `http://dx.doi.org/10.1109/IM.2003.1240269`.

S. Tateshima, Y. Murayama, J. P. Villablanca, T. Morino, H. Takahashi, T. Yamauchi, K. Tanishita, and F. Viñuela. Intraaneurysmal flow dynamics study featuring an acrylic aneurysm model manufactured using a computerized tomography angiogram as a mold. *Journal of Neurosurgery*, 95(6):1020–1027, 2001. URL `http://thejns.org/doi/abs/10.3171/jns.2001.95.6.1020`.

G. Taubin. A signal processing approach to fair surface design. In *Proc. of the ACM SIGGRAPH Conference on Computer Graphics*, pages 351–358. ACM, 1995. URL `http://dx.doi.org/10.1145/218380.218473`.

S. Thakkur and T. Huff. Internet Streaming SIMD Extensions. *Computer*, 32(12):26 –34, December 1999. URL `http://dx.doi.org/10.1109/2.809248`.

C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 839–. IEEE Computer Society, 1998. URL `http://dl.acm.org/citation.cfm?id=938978.939190`.

T. Torsney-Weir, A. Saad, T. Möller, H. Hege, B. Weber, J. Verbavatz, and S. Bergner. Tuner: Principled Parameter Finding for Image Segmentation Algorithms Using Visual Response Surface Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1892 –1901, 2011. URL http://dx.doi.org/10.1109/TVCG.2011.248.

R. van Pelt, J. Olivan Bescos, M. Breeuwer, R. Clough, M. Groller, B. ter Haar Romeny, and A. Vilanova. Interactive Virtual Probing of 4D MRI Blood-Flow. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2153 –2162, December 2011. URL http://dx.doi.org/10.1109/TVCG.2011.215.

J. Vollmer, R. Mencl, and H. Müller. Improved Laplacian Smoothing of Noisy Surface Meshes. *Computer Graphics Forum*, 18(3):131–138, 1999. URL http://dx.doi.org/10.1111/1467-8659.00334.

R. T. Whitaker. Reducing aliasing artifacts in iso-surfaces of binary volumes. In *Proc. of the IEEE symposium on volume visualization*, pages 23–32, 2000. URL http://dx.doi.org/10.1109/VV.2000.10004.

P. M. White and J. M. Wardlaw. Unruptured intracranial aneurysms. *J Neuroradiology*, 30(5):336–50, 2003. URL http://www.ncbi.nlm.nih.gov/pubmed/14752379.

D. O. Wiebers. Unruptured intracranial aneurysms: natural history, clinical outcome, and risks of surgical and endovascular treatment. *The Lancet*, 362(9378):103 – 110, 2003. URL http://dx.doi.org/10.1016/S0140-6736(03)13860-3.

J. Winder. Medical rapid prototyping and 3D CT in the manufacture of custom made cranial titanium plates. *Journal of Medical Engineering & Technology*, 23(1):26–28, 1999. URL http://dx.doi.org/10.1080/030919099294401.

J. Wu, M. Wei, Y. Li, X. Ma, F. Jia, and Q. Hu. Scale-adaptive surface modeling of vascular structures. *Biomed Eng Online*, 9:75, 2010. URL http://dx.doi.org/10.1186/1475-925X-9-75.

Z. Wu and J. M. Sullivan. Multiple material marching cubes algorithm. *International Journal for Numerical Methods in Engineering*, 58(2):189–207, 2003. URL http://dx.doi.org/10.1002/nme.775.

H. Yagou, Y. Ohtake, and A. Belyaev. Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding. In *Proc. of Computer Graphics International*, pages 28–33, 2003. URL http://dx.doi.org/10.1109/CGI.2003.1214444.

S. Yoshizawa, A. Belyaev, and H. P. Seidel. Smoothing by Example: Mesh Denoising by Averaging with Similarity-Based Weights. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 9–9, 2006. URL http://dx.doi.org/10.1109/SMI.2006.38.

W. Zhao, S. Gao, and H. Lin. A robust hole-filling algorithm for triangular mesh. *Vis. Comput.*, 23(12):987–997, November 2007. URL http://dx.doi.org/10.1007/s00371-007-0167-y.

List of Abbreviations

API	Application Programming Interface
CA	Context-Aware
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CT	Computed Tomography
CUDA	Compute Unified Device Architecture
DA	Distance-Aware
ENT	Ear-Neck-Throat
fMRI	functional MRI
GLSL	OpenGL Shading Language
GPGPU	General-Purpose computing on Graphics Processing Units
GPU	Graphics Processing Unit
HC	Refers to the Laplace+*HC* filter. As stated by Vollmer et al. [1999], "HC stands for Humphrey's Classes and has no deeper meaning".
MPUI	Multi-level Partition of Unity Implicits [Ohtake et al., 2003]
MRI	Magnetic Resonance Imaging
NPC	Node-Position-Constraint
OpenCL	Open Computing Language
OpenGL	Open Graphics Library
OpenMP	Open Multi-Processing
PET	Positron Emission Tomography

ROI	Region Of Interest
SA	Staircase-Aware
SCM	Sternocleidomastoid Muscle
SIMD	Single Instruction, Multiple Data streams
SPECT	Single-Photon Emission Computed Tomography
SSE	Streaming SIMD Extensions
TBO	Texture Buffer Object
VBO	Vertex Buffer Object
VTK	Visualization Toolkit – see `http://www.vtk.org`
ViERforES	Virtuelle und Erweiterte Realität für höchste Sicherheit und Zuverlässigkeit von Eingebetteten Systemen – project funded by the German Federal Ministry of Education and Research (BMBF)
WebGL	Web Graphics Library
WSS	Wall Shear Stress
X3D	Extensible 3D – an XML-based file format
XBO	Transform Feedback Buffer Object