Friedrich Schiller University Jena
Faculty of Mathematics and Computer Science

Design and Implementation of High-Performance, Adaptive, and Robust Curve Smoothing on Surface Meshes and its Application to Medical Visualization

MASTER'S THESIS

for obtaining the academic degree

Master of Science (M.Sc.) in Mathematics

submitted by Markus Pawellek

born on May 7th, 1995 in Meiningen
Student Number: 144645

Primary Supervisor:	Kai Lawonn
Secondary Supervisor:	Noeska Smit

Bergen, November 19, 2022

Abstract

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Zusammenfassung

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Acknowledgments

I am grateful to Kai Lawonn and Noeska Smit for their supervising, their helpful suggestions, and for our interesting discussions. Also great thanks to Ann Sommerfeld for assisting in writing the thesis and proofreading.

Contents

List of Figures

List of Tables

List of Definitions and Theorems

List of Code

List of Abbreviations and Acronyms

Abbreviation	Definition
iid	Independently and Identically Distributed
CDF	Cumulative Distribution Function
SLLN	Strong Law of Large Numbers
LTE	Light Transport Equation
API	Application Programming Interface
RAII	Resource Acquisition is Initialization
SFINAE	Specialization Failure is not an Error
STL	Standard Template Library

Symbol Table

Symbol	Definition
Logic	
$\exists \ldots : \ldots$	There exists . . ., such that . . .
$a := b$	a is defined by b.
Set Theory	
$\{\ldots\}$	Set Definition
$\{\ldots \mid \ldots\}$	Set Definition with Condition
$x \in A$	x is an element of the set A.
$A \subset B$	The set A is a subset of the set B.
$A \cap B$	Intersection — $\{x \mid x \in A \text{ and } x \in B\}$ for sets A, B
$A \cup B$	Union — $\{x \mid x \in A \text{ or } x \in B\}$ for sets A, B
$A \setminus B$	Relative Complement — $\{x \in A \mid x \notin B\}$ for sets A, B
$A \times B$	Cartesian Product — $\{(x, y) \mid x \in A, y \in B\}$ for sets A and B
A^n	n-fold Cartesian Product of Set A
$\emptyset$	Empty set — $\{\}$.
$\#A$	Number of Elements in the Set A
$\mathcal{P}(A)$	Power Set of Set A
Special Sets	
$\mathbb{N}$	Set of Natural Numbers
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$
$\mathbb{P}$	Set of Prime Numbers
$\mathbb{Z}$	Set of Integers
$\mathbb{Z}_n$	Set of Integers Modulo n
$\mathbb{F}_m$	Finite Field with $m \in \mathbb{P}$ Elements
$\mathbb{F}_m^{p \times q}$	Set of $p \times q$-Matrices over Finite Field $\mathbb{F}_m$
$\mathbb{F}_2$	Finite Field of Bits
$\mathbb{F}_2^n$	Set of n-bit Words
$\mathbb{R}$	Set of Real Numbers
$\mathbb{R}^n$	Set of n-dimensional Real Vectors
$\mathbb{S}^2$	Set of Directions — $\left\{x \in \mathbb{R}^3 \;\middle|\; \|x\| = 1\right\}$
Functions	
$f\colon X \to Y$	f is a function with domain X and range Y.
id_X	Identity Function over the Set X
$f \circ g$	Composition of Functions f and g
f^{-1}	Inverse Image of Function f
f^n	n-fold Composition of Function f
Bit Arithmetic	
$x_{n-1} \ldots x_1 x_0$	n-bit Word x of Set $\mathbb{F}_2^n$
$x \leftarrow a$	Left Shift of all Bits in x by a
$x \to a$	Right Shift of all Bits in x by a
$x \circlearrowleft a$	Circular Left Shift of all Bits in x by a
$x \oplus y$	Bit-Wise Addition of x and y
$x \odot y$	Bit-Wise Multiplication of x and y
$x \mid y$	Bit-Wise Or of x and y

Symbol	Definition
Probability Theory	
$\mathcal{B}(\mathbb{R})$	Borel σ-Algebra over $\mathbb{R}$
$(\Sigma, \mathcal{A})$	Measurable Space over Σ with σ-Algebra $\mathcal{A}$
λ	Lebesgue Measure
$\int_U f \,\mathrm{d}\lambda$	Lebesgue Integral of f over U
$\mathrm{L}^2(U, \lambda)$	Set of Square-Integrable Functions over the Set U with Respect to the Lebesgue Measure λ
$(\Omega, \mathcal{F}, P)$	Probability Space over Ω with σ-Algebra $\mathcal{A}$ and Probability Measure P
$\int_\Omega X \,\mathrm{d}P$	Integral of Random Variable X with respect to Probability Space $(\Omega, \mathcal{A}, P)$
$\int_\Omega X(\omega) \,\mathrm{d}P(\omega)$	$\int_\Omega X \,\mathrm{d}P$
P_X	Distribution of Random Variable X
$\mathbb{E}\, X$	Expectation Value of Random Variable X
$\mathrm{var}\, X$	Variance of Random Variable X
$\sigma(X)$	Standard Deviation of Random Variable X
$\mathbb{1}_A$	Characteristic Function of Set A
δ_ω	Dirac Delta Distribution over $\mathcal{S}^2$ with respect to $\omega \in \mathcal{S}^2$
$\bigotimes_{n\in I} P_n$	Product Measure of Measures P_n Indexed by the Set I
Miscellaneous	
$(x_n)_{n\in I}$	Sequence of Values x_n with Index Set I
$\lvert x\rvert$	Absolute Value of x
$\lVert x\rVert$	Norm of Vector x
$x \mod y$	x Modulo y
$\gcd(\rho, k)$	Greatest Common Divisor of ρ and k
$\max(x, y)$	Maximum of x and y
$\lim_{n\to\infty} x_n$	Limit of Sequence $(x_n)_{n\in\mathbb{N}}$
$\sum_{k=1}^n x_k$	Sum over Values x_k for $k \in \mathbb{N}$ with $k \leq n$
$\dim X$	Dimension of X
$\lceil x\rceil$	Ceiling Function
$\langle x \mid y\rangle$	Scalar Product
$[a, b]$	$\{x \in \mathbb{R} \mid a \leq x \leq b\}$
(a, b)	$\{x \in \mathbb{R} \mid a < x < b\}$
$[a, b)$	$\{x \in \mathbb{R} \mid a \leq x < b\}$
Constants	
∞	Infinity
π	$3.1415926535\ldots$ — Pi
Units	
$1\,\mathrm{B}$	$1\,\mathrm{Byte} = 8\,\mathrm{bit}$
$1\,\mathrm{GiB}$	$2^{30}\,\mathrm{B}$
$1\,\mathrm{s}$	$1\,\mathrm{Seconds}$
$1\,\mathrm{min}$	$1\,\mathrm{Minutes} = 60\,\mathrm{s}$
$1\,\mathrm{GHz}$	$1\,\mathrm{Gigahertz} = 10^9\,\mathrm{Hertz}$

1 Introduction

Nowadays, the majority of application domains vital to the life of humanity is supported by computer-aided systems. These are typically programs that provide a set of tools to facilitate the automatic generation, transfer, manipulation, and visualization of domain-specific data by keeping user interaction at a required minimum. Computer systems have enabled humanity to streamline processes and to abstract and encapsulate low-level tasks. As a consequence, this resulted in the ability to solve harder problems even more efficiently.

Especially in the area of medicine, examples such as the resection of liver tumors for long-term survival (Alirr and Abd. Rahni 2019) and osteotomy planning (Zachow et al. 2003), that involves reshaping and realigning bones to repair or fix bone-specific issues, show that the use of computer-aided systems for surgery planning reduces the duration of treatment and heavily increases the chance of long-term survival. Both of the named medical applications use curves on the two-dimensional reconstructed surface of scanned medical objects, such as livers and bones, to represent and visualize surgery cuts. The reconstructed surfaces will thereby be provided as triangular meshes and are often referred to as surface meshes.
(Alirr and Abd. Rahni 2019; Zachow et al. 2003)[1]

By construction, initially chosen curves on these surfaces are jagged due to the finite precision of the underlying mesh and emit curvature noise that is not neglectable and perceivable by the human eye. Hence, a smoothing process is applied to initial curves to reduce their overall curvature and attain surface cuts with well-defined properties. In general, the result of curve smoothing might strongly deviate from the initially given curve to fulfill the given constraints. For medical surface cutting applications, though, the shape of an initial curve is defined by domain experts, such as physicians or bioengineers, and most likely indicates relevant anatomical landmarks or surface regions. Thus, under these circumstances, the smoothing additionally requires the resulting curve to be close to its original such that no essential information is lost during the process. (Lawonn et al. 2014)

Futhermore, it is a matter of fact, that curves on surface meshes and algorithms for smoothing them are basic building blocks for mesh processing and segmentation (Ji et al. 2006; Kaplansky and Tal 2009). Consequently, their fundamental role in the areas of computer-aided geometric design, computer graphics, and visualization, that are heavily based on mesh processing, is unconcealable. So, curve smoothing on surface meshes is not only relevant in specific areas of medicine but is a generally applicable and important tool to many other domains of applications building on the above research areas. Further domain areas, such as machine learning (Benhabiles et al. 2011; Park et al. 2019) and engineering, therefore provide many more direct and significant applications.

Besides their mathematical correctness and convergence, curve smoothing algorithms should exhibit a certain level of adaptivity with respect to the given surface mesh and its initially chosen curve. Surface meshes are most typically an irregular grid of triangular faces that may highly vary in diameter and area. In addition, the initial curve might be extreme concerning its length, curvature, and overall shape. An algorithm to smooth curves on surface meshes needs to adapt to all these situations and still figure out the best possible result that abides to the given criteria. In conjunction with its correctness, this also means that such an algorithm needs to be robust for many different kinds of scenarios, such as self-intersecting curves and noisy surface geometries, that may result in wrong calculations based on the finite

[1] In this thesis, citations concerning a whole paragraph will be given after the last sentence of the very paragraph.

precision of floating-point values. Yet another property to take into account is the efficiency of the algorithm. To seamlessly integrate curve smoothing into the user interface of a computer-assisted system for domain-specific applications, it at least needs to provide an interactive up to real-time performance. (Lawonn et al. 2014)

There are a few already existing algorithms for producing smoothed curves on surface meshes (Hofer and Pottmann 2004; Lawonn et al. 2014; Mancinelli et al. 2022; Martínez, Carvalho, and Velho 2007). Still, the implementation and API design of such algorithms is assumed to be an involved task and error-prone when the programmer intends to apply the algorithm on a wide variety of cases. All the given references define their algorithm and explain its properties in great detail. They compare the quality of generated curves to alternative algorithms and describe the algorithm's programming procedures at least with respect to a high-level point of view based on pseudocode. However, the very low-level details about the composition of data structures, advice for an implementation in a specific programming language, or ways to handle difficult corner cases are left out. This makes the comparison of the performance and robustness of algorithms much harder and unreproducible, because custom implementations would need to be used. Furthermore, up to this point there is no widely accepted metric to compare the smoothness of two different generated curves which leads to highly subjective treatment and evaluation of different algorithms.

For the design and implementation of a basic framework for curve smoothing on surfaces that allows for high-performance, reproducibility, and robustness, adequate candidates are the modern standards of the C++ programming language in conjunction with the OpenGL graphics API. C++ is a multi-paradigm language that integrates many different programming styles, such as object-oriented, functional, and data-oriented programming. It is still the de-facto standard for graphics applications and well-known to be one of the fastest languages in the world which incorporates low-level programming based on assembler routines and efficient high-level abstraction mechanisms, like template meta programming. The design of the whole language keeps on advancing to make programs faster and easier to develop. In the most common cases, C++ can be seen as a superset of the older C programming language which is typically used by other programming languages to provide the possibility of code being called from different languages. Therefore the users of the framework are not even restricted to use C++ but instead are able to use other languages, like Python, to communicate with a C interface to achieve similar results. OpenGL is the open-source graphics API that allows programs to efficiently communicate and interact with the driver of the graphics card to visualize provided data independently of the manufacturer or the operating system. By using them, no strong constraints are imposed on the software environment that the software framework is running on. Both tools allow for a sophisticated modularization of the whole framework. So, no user needs to pay for features that are not needed.
(*cppreference.com* n.d.; Meyers 2014; *OpenGL: The Industry's Foundation for High Performance Graphics* 2023; Reddy 2011; *Standard C++ Foundation* 2023; Stroustrup 2014; Vandevoorde, Josuttis, and Gregor 2018)

In this thesis, precisely in sections 4 and 5, we develop a new library and program, called *reflex*[2], using the C++ programming language in conjunction with OpenGL graphics API. *reflex* implements parallelized and tweaked variants of the curve smoothing algorithm given by Lawonn et al. (2014) on the CPU and GPU which should be applicable in a wide variety

[2]Markus Pawellek (2023). *reflex. Reactive and Flexible Curve Smoothing on Surface Meshes*. URL: https://github.com/lyrahgames/reflex (visited on 01/15/2023).

of cases. Hereby, a special emphasis lies on the robust and fast implementation for medical purposes. The program and library are open-source and can be found on GitHub. The necessary theoretical background to understand the design- and the implementation-specific aspects is given in the section 2. Here, we will give a brief introduction to differential geometry, polyhedral manifolds, and computer architecture. A mathematical rigorous discussion about the algorithm will be part of section 4 to properly encapsulate all the information specific to the implementations. Section 3 refers to the previous work concerning general curves, geodesics and the smoothing of curves on surfaces. At the end in section 6, we apply the constructed algorithm to the problem of segmentation of lung lobes (Park et al. 2019). In the sections 7 and 8, the evaluation is shown followed by a discussion dealing with further improvements.

2 Preliminaries

Differential Geometry on Polyhedral Surfaces (Polthier and Schmies 2006)

Curvature Estimation on Surfaces (Rusinkiewicz 2004)

Generation of Surface Normals (Jin, Lewis, and West 2005; Max 1999; Meyer et al. 2001)

3 Previous Work

As marked in the introduction in section 1, the smoothing of curves on surface meshes is an essential operation for mesh processing and, as a consequence, for many other domain areas, like computer graphics, image-based medicine, and engineering, that rely on such tools (Ji et al. 2006; Kaplansky and Tal 2009). In most of its applications, initial curves are either provided by means of direct user interaction or by automatic or semiautomatic feature detection algorithms (Lawonn et al. 2014; Zachow et al. 2003). The finite precision of the underlying surface mesh together with all the steps included to define an initial curve usually makes resulting lines contain non-smooth artifacts which may violate given constraints or expected properties and therefore degrade its quality (Kaplansky and Tal 2009; Lawonn et al. 2014). Introducing a smoothing stage into the curve processing pipeline, the mesh segmentation is expected to be of much higher quality which greatly increases its usage for areas like machine learning (Benhabiles et al. 2011) or medicine (Alirr and Abd. Rahni 2019; Zachow et al. 2003). During the last two decades, there have been multiple successful attempts for constructing algorithms to smooth curves on surfaces (Bischoff, Weyand, and Kobbelt 2005; Hofer and Pottmann 2004; Lawonn et al. 2014; Mancinelli et al. 2022). In this section, a brief overview of their major contributions is given.

As stated in the previous section 2, a crucial tool for working with curves on two-dimensional manifolds is the ability to generate geodesics in the sense of the initial and boundary value problem. The rigorous mathematical concepts and definitions for the discrete geodesics problems have been elaborated by Mitchell, Mount, and Papadimitriou (1987) and Polthier and Schmies (2006) first published in 1997. Additionally, Mitchell, Mount, and Papadimitriou (1987) built an algorithm to solve the discrete boundary value problem, that used a continuous version of the algorithm of Dijkstra (1959) to find the shortest path connecting two given points. Furthermore, Polthier and Schmies (2006) provided an iterative algorithm to solve the discrete initial value problem of finding the geodesic given a starting point and a direction. They also introduced the parallel translation of vectors along the surface for particle transportation. This algorithm has been improved by Mancinelli et al. (2022) through the use of optimized data structures and a superior choice of initial curves. Based upon the theory of Polthier and Schmies (2006), Martínez, Velho, and Carvalho (2005) provided an algorithm to the discrete boundary value problem. Hereby, a starting curve on the surface had to be given as initial value to iteratively improve it up to an approximated geodesic. Surazhsky et al. (2005) developed exact and approximate algorithms based on Mitchell, Mount, and Papadimitriou (1987) for the discrete initial and boundary value problem, which could be evaluated efficiently by the use of distance fields. Extending the idea of distance fields as an intermediate step to the generation of geodesics, Bommes and Kobbelt (2007) generalized the algorithm of Surazhsky et al. (2005) to not only handle isolated points for their distance fields but also general polygons on the surface. Also based on the results of Mitchell, Mount, and Papadimitriou (1987), Kimmel and Sethian (1996) introduced the so-called fast marching approach, which used the eikonal equation to build propagating fronts to more efficiently generate the distance fields. Hereupon, Crane, Weischedel, and Wardetzky (2013) also used the gradient of the heat kernel to reconstruct a distance field by solving the Poisson equation.

Application of Cutting Curves in Mesh Processing (Zachow et al. 2003) (Benhabiles et al. 2011) (Ji et al. 2006)

Previous Intuitive Approach called Corner Cutting in Planar (Chaikin 1974) (Dyn, Levin, and Liu 1992) in space (Morera, Velho, and Carvalho 2008) But this may not provide a real surface curve.

Lines as Snakes (Kass, Witkin, and Terzopoulos 1988) Generalization 2D-Manifold (Bischoff, Weyand, and Kobbelt 2005) (Jung and Kim 2004)

Automatic Surface Segmentation and Cutting (Lee and Lee 2002) (Lee et al. 2004)

Feature-Sensitive Curve Smoothing (Lai et al. 2007)

Geodesics on Triangular Meshes (Martínez, Velho, and Carvalho 2005)

Bezier Curves on Meshes (Martínez, Carvalho, and Velho 2007) and splines in manifolds (Hofer and Pottmann 2004) and geodesics in polyhedral surfaces (Polthier and Schmies 2006) (Mitchell, Mount, and Papadimitriou 1987) (Surazhsky et al. 2005)

geodesic distance fields (Bommes and Kobbelt 2007) (Kimmel and Sethian 1996)

heat maps (Crane, Weischedel, and Wardetzky 2013)

(Dijkstra 1959)

(Ma and Chen 2007) (Pottmann and Hofer 2005) (Lévy et al. 2002)

(Mancinelli et al. 2022)

(Yu, Schumacher, and Crane 2021)

(Engelke et al. 2018)

4 Design

5 Implementation

6 Application

7 Evaluation and Results

8 Conclusions and Future Work

References

Books

McCool, Michael, Arch D. Robison, and James Reinders (2012). *Structured Parallel Programming: Patterns of Efficient Computation*. Morgan Kaufmann – Elsevier. ISBN: 978-0-12-415993-8.

Meyers, Scott (2014). *Effective Modern C++*. O'Reilly Media. ISBN: 978-1-491-90399-5.

Munzner, Tamara (2014). *Visualization Analysis and Design*. A K Peters Visualization Series. CRC Press. URL: https://www.cs.ubc.ca/~tmm/vadbook/ (visited on 11/16/2022).

Patterson, David A. and John L. Hennessy (2014). *Computer Organization and Design. The Hardware/Software Interface*. Fifth Edition. Morgan Kaufmann – Elsevier. ISBN: 978-0-12-407726-3.

Reddy, Martin (2011). *API Design for C++*. Morgan Kaufmann – Elsevier. ISBN: 978-0-12-385003-4.

Stroustrup, Bjarne (2014). *The C++ Programming Language*. Fourth Edition. Addison-Wesley – Pearson Education. ISBN: 978-0-321-95832-7.

Vandevoorde, David, Nicolai M. Josuttis, and Douglas Gregor (2018). *C++ Templates: The Complete Guide*. Second Edition. Addison-Wesley – Pearson Education. ISBN: 978-0-321-71412-1.

Williams, Anthony (2019). *C++ Concurrency in Action*. Second Edition. Manning Publications. ISBN: 978-1-61-729469-3.

Online

cppreference.com (n.d.). URL: https://en.cppreference.com/w/ (visited on 01/15/2023).

OpenGL: The Industry's Foundation for High Performance Graphics (2023). URL: https://www.opengl.org/ (visited on 01/15/2023).

Pawellek, Markus (2023). *reflex. Reactive and Flexible Curve Smoothing on Surface Meshes*. URL: https://github.com/lyrahgames/reflex (visited on 01/15/2023).

Standard C++ Foundation (2023). URL: https://isocpp.org/ (visited on 01/15/2023).

Articles

Alirr, Omar and Ashrani Aizzuddin Abd. Rahni (August 2019). "Survey on Liver Tumour Resection Planning System: Steps, Techniques, and Parameters". In: *Journal of Digital Imaging* 33. DOI: 10.1007/s10278-019-00262-8.

Benhabiles, Halim et al. (December 2011). "Learning Boundary Edges for 3D-Mesh Segmentation". In: *Computer Graphics Forum* 30, pp. 2170–2182. DOI: 10.1111/j.1467-8659.2011.01967.x.

Bischoff, Stephan, Tobias Weyand, and Leif Kobbelt (January 2005). "Snakes on Triangle Meshes". In: *Proceedings of Bildverarbeitung für die Medizin*, pp. 208–212. DOI: 10.1007/3-540-26431-0_43.

Bommes, David and Leif Kobbelt (January 2007). "Accurate Computation of Geodesic Distance Fields for Polygonal Curves on Triangle Meshes". In: *Proceedings of the Vision, Modeling, and Visualization Conference*, pp. 151–160. URL: https://www-sop.inria.fr/members/David.Bommes/publications/geodesic.pdf (visited on 11/16/2022).

Chaikin, George (December 1974). “An Algorithm for High-Speed Curve Generation”. In: *Computer Graphics and Image Processing* 3, pp. 346–349. DOI: 10.1016/0146-664X(74)90028-8.

Crane, Keenan, Clarisse Weischedel, and Max Wardetzky (September 2013). “Geodesics in Heat: A New Approach to Computing Distance Based on Heat Flow”. In: *ACM Transactions on Graphics* 32. DOI: 10.1145/2516971.2516977.

Dijkstra, Edsger W. (1959). “A Note on Two Problems in Connexion with Graphs”. In: *Numerische Mathematik* 1, pp. 269–271. URL: https://www.semanticscholar.org/paper/A-note-on-two-problems-in-connexion-with-graphs-Dijkstra/45786063578e814444b8247028970758bbbd0488 (visited on 11/17/2022).

Dyn, Nira, D. Levin, and D. Liu (April 1992). “Interpolatory Convexity-Preserving Subdivision Schemes for Curves and Surfaces”. In: *Computer-Aided Design* 24, pp. 211–216. DOI: 10.1016/0010-4485(92)90057-H.

Engelke, Wito et al. (August 2018). “Autonomous Particles for Interactive Flow Visualization”. In: *Computer Graphics Forum* 38. DOI: 10.1111/cgf.13528.

Hertzmann, Aaron and Denis Zorin (2000). “Illustrating Smooth Surfaces”. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’00. ACM Press/Addison-Wesley Publishing Co., 517–526. ISBN: 1581132085. DOI: 10.1145/344779.345074.

Hofer, Michael and Helmut Pottmann (August 2004). “Energy-Minimizing Splines in Manifolds”. In: *ACM Transactions on Graphics* 23, pp. 284–293. DOI: 10.1145/1015706.1015716.

Ji, Zhongping et al. (September 2006). “Easy Mesh Cutting”. In: *Computer Graphics Forum* 25, pp. 283–291. DOI: 10.1111/j.1467-8659.2006.00947.x.

Jin, Shuangshuang, Robert Lewis, and David West (February 2005). “A Comparison of Algorithms for Vertex Normal Computation”. In: *The Visual Computer* 21, pp. 71–82. DOI: 10.1007/s00371-004-0271-1.

Jung, Moonryul and Haengkang Kim (November 2004). “Snaking Across 3D Meshes”. In: *Proceedings of Pacific Graphics*, pp. 87–93. DOI: 10.1109/PCCGA.2004.1348338.

Kaplansky, Lotan and Ayellet Tal (October 2009). “Mesh Segmentation Refinement”. In: *Computer Graphics Forum* 28, pp. 1995–2003. DOI: 10.1111/j.1467-8659.2009.01578.x.

Kass, Michael, Andrew Witkin, and Demetri Terzopoulos (January 1988). “Snakes: Active Contour Models”. In: *IEEE Proceedings on Computer Vision and Pattern Recognition* 1, pp. 321–331. URL: https://sites.pitt.edu/~sjh95/related_papers/Kass1988_Article_SnakesActiveContourModels.pdf (visited on 11/16/2022).

Kimmel, Ron and J. A. Sethian (1996). *Fast Marching Methods for Computing Distance Maps and Shortest Paths*. Tech. rep. Lawrence Berkeley National Laboratory. URL: https://escholarship.org/uc/item/7kx079v5 (visited on 11/16/2022).

Kindlmann, Gordon et al. (November 2003). “Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications”. In: vol. 2003, pp. 513–520. ISBN: 0-7803-8120-3. DOI: 10.1109/VISUAL.2003.1250414.

Lai, Yu-Kun et al. (January 2007). “Robust Feature Classification and Editing”. In: *IEEE Transactions on Visualization and Computer Graphics* 13, pp. 34–45. DOI: 10.1109/TVCG.2007.19.

Lawonn, Kai et al. (2014). “Adaptive and Robust Curve Smoothing on Surface Meshes”. In: *Computers & Graphics* 40, pp. 22–35. DOI: 10.1016/j.cag.2014.01.004.

Lee, Yunjin and S. Lee (September 2002). “Geometric Snakes for Triangular Meshes”. In: *Computer Graphics Forum* 21, pp. 229 –238. DOI: 10.1111/1467-8659.t01-1-00582.

Lee, Yunjin et al. (January 2004). “Intelligent Mesh Scissoring Using 3D Snakes”. In: pp. 279–287. DOI: 10.1109/PCCGA.2004.1348358.

Lévy, Bruno et al. (July 2002). “Least Squares Conformal Maps for Automatic Texture Atlas Generation”. In: *ACM Transactions on Graphics* 21, pp. 362–371. DOI: 10.1145/566654.566590.

Ma, Li and Dezhong Chen (June 2007). “Curve Shortening in a Riemannian Manifold”. In: *Annali Di Matematica Pura Ed Applicata* 186, pp. 663–684. DOI: 10.1007/s10231-006-0025-y.

Mancinelli, Claudio et al. (May 2022). “b/Surf: Interactive Bzier Splines on Surface Meshes”. In: *IEEE Transactions on Visualization and Computer Graphics* PP. DOI: 10.1109/TVCG.2022.3171179.

Martínez, Dimas, Paulo de Carvalho, and Luiz Velho (November 2007). “Geodesic Bezier Curves: A Tool for Modeling on Triangulations”. In: *Brazilian Symposium on Computer Graphics and Image Processing*, pp. 71–78. ISBN: 978-0-7695-2996-7. DOI: 10.1109/SIBGRAPI.2007.38.

Martínez, Dimas, Luiz Velho, and Paulo de Carvalho (October 2005). “Computing Geodesics on Triangular Meshes”. In: *Computers & Graphics* 29, pp. 667–675. DOI: 10.1016/j.cag.2005.08.003.

Max, Nelson (January 1999). “Weights for Computing Vertex Normals from Facet Normals”. In: *Journal of Graphics Tools* 4. DOI: 10.1080/10867651.1999.10487501.

Meyer, Mark et al. (November 2001). “Discrete Differential-Geometry Operators for Triangulated 2-Manifolds”. In: *Proceedings of Visualization and Mathematics* 3. DOI: 10.1007/978-3-662-05105-4_2.

Mitchell, Joseph, David Mount, and Christos Papadimitriou (August 1987). “The Discrete Geodesic Problem”. In: *SIAM Journal on Computing* 16, pp. 647–668. DOI: 10.1137/0216045.

Morera, Dimas Martínez, Luiz Velho, and Paulo Cezar Pinto Carvalho (2008). “Subdivision Curves on Triangular Meshes”. In: URL: https://www.semanticscholar.org/paper/Subdivision-Curves-on-Triangular-Meshes-Morera-Velho/595d28aacea33ba038d36e7dc403c156a9248905 (visited on 11/16/2022).

Park, Jongha et al. (May 2019). “Fully Automated Lung Lobe Segmentation in Volumetric Chest CT with 3D U-Net: Validation with Intra- and Extra-Datasets”. In: *Journal of Digital Imaging* 33. DOI: 10.1007/s10278-019-00223-1.

Polthier, Konrad and Markus Schmies (2006). “Straightest Geodesics on Polyhedral Surfaces”. In: *ACM SIGGRAPH 2006 Courses*. SIGGRAPH ’06. Association for Computing Machinery, 30–38. DOI: 10.1145/1185657.1185664.

Pottmann, Helmut and Michael Hofer (October 2005). “A Variational Approach to Spline Curves on Surface”. In: *Computer Aided Geometric Design* 22, pp. 693–709. DOI: 10.1016/j.cagd.2005.06.006.

Rusinkiewicz, Szymon (October 2004). “Estimating Curvatures and Their Derivatives on Triangle Meshes”. In: pp. 486–493. ISBN: 0-7695-2223-8. DOI: 10.1109/TDPVT.2004.1335277.

Surazhsky, Vitaly et al. (July 2005). “Fast Exact and Approximate Geodesics on Meshes”. In: *ACM Transactions on Graphics* 24, pp. 553–560. DOI: 10.1145/1073204.1073228.

Yu, Chris, Henrik Schumacher, and Keenan Crane (April 2021). “Repulsive Curves”. In: *ACM Transactions on Graphics* 40, pp. 1–21. DOI: 10.1145/3439429.

Zachow, Stefan et al. (2003). “Draw and Cut: Intuitive 3D Osteotomy Planning on Polygonal Bone Models”. In: *International Congress Series* 1256, pp. 362–369. DOI: 10.1016/S0531-5131(03)00272-3.

A Mathematical Proofs

B Further Code

Statutory Declaration

I declare that I have developed and written the enclosed Master's thesis completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked. The Master's thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.
On the part of the author, there are no objections to the provision of this Master's thesis for public use.

Bergen, November 19, 2022

Markus Pawellek