



TECHNISCHE UNIVERSITÄT MÜNCHEN

Department of Computer Science

Computer Vision Group

**A Convex Optimization Framework for
Connectivity Constraints in Image Segmentation and
3D Reconstruction**

Jan Stühmer



TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik

Lehrstuhl für Bildverarbeitung und Mustererkennung

**A Convex Optimization Framework for
Connectivity Constraints in Image Segmentation and
3D Reconstruction**

Jan Stühmer

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Nassir Navab

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Daniel Cremers
2. Prof. William T. Freeman, PhD
Massachusetts Institute of Technology, Cambridge, MA

Die Dissertation wurde am 01.01.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 01.01.2016 angenommen.

Summary

This thesis presents a framework for image segmentation and 3D reconstruction under topological constraints, specifically constraints on the connectivity of the reconstructed object, a problem class that is not yet sufficiently solved by existing methods.

Image segmentation, the partition of an image into meaningful parts, is one of the most important and well studied problems in Computer Vision. This stems on the one hand from its high importance for practical applications, often the first step towards a semantic interpretation of the image data, for example in object detection and classification, or image based quantification in medical imaging. On the other hand, image segmentation problems are interesting from a theoretical perspective. Even moderate tasks, for example segmenting an image into more than two regions, result in problems which remain hard to solve and require advanced knowledge in mathematical optimization.

Although the field has been studied for decades, existing methods often share a common drawback. They fail in cases when the object of interest has an elongated and thin shape. This occurs often in practical applications, for example the segmentation of blood vessels in medical imaging or in 3D reconstruction, when the object contains thin or small detailed structures.

In this thesis, we present a framework for image segmentation and 3D reconstruction, that is specifically successful in these cases of thin and elongated structures and allows improved results in comparison to existing methods. The framework allows to incorporate constraints on the topology of the object, specifically on its connectedness. The major contribution of this thesis is, that the constraints can be formulated as linear constraints in a convex optimization framework, which allows to solve the resulting optimization problem to global optimality. In comparison to most prior work, the method is very efficient both in runtime and memory requirements, which allows applications on large scale practical problems in medical imaging and 3D reconstruction. Furthermore, an efficient projection method onto the constraint set is presented, that significantly reduces the runtime complexity in cases when the connections have to span larger distances.

Additionally, the thesis presents an approach to define a probabilistic model of the data term used for image segmentation and, based on the uncertainty of the classifier, to improve the probabilistic model over time by presenting informative queries to the user. Furthermore, a novel approach for model based object tracking using phase based Time-of-Flight depth cameras is presented.

keywords: image segmentation, 3D reconstruction, convex optimization, topological constraints, connectivity constraints, minimal surfaces

Zusammenfassung

Diese Arbeit beschreibt eine einheitliche Formulierung des Problems der Bildsegmentierung und 3D Rekonstruktion, die es erlaubt Nebenbedingungen über die Topologie des rekonstruierten Objektes zu formulieren, eine Problemstellung die von bisherigen Ansätzen nur unzureichend für praktische Anwendungen gelöst wurde.

Verfahren zur Bildsegmentierung, der Aufteilung eines Bildes in spezifische Teilsegmente, ist eines der am längsten und besten untersuchten Teilgebiete des Forschungsfeldes "Maschinelles Sehen". Dies ist zum einen durch die hohe praktische Relevanz der Problemstellung gegeben, meist ist die Bildsegmentierung eine Voraussetzung für eine semantische Analyse von Bild-daten, zum Beispiel für die Objekt-Detektion und Klassifizierung und bei Anwendungen in der medizinischen Bildverarbeitung zur bildbasierten Quantifizierung. Zum anderen sind Bildsegmentierungsprobleme aus algorithmisch-analytischer Perspektive interessant: Bereits relativ einfache Aufgaben, zum Beispiel die Segmentierung in mehr als zwei Regionen, resultieren in mathematisch anspruchsvolle Problemstellungen und erfordern fortgeschrittene Methoden der mathematischen Optimierung.

Obwohl das Aufgabengebiet der Bildsegmentierung bereits seit mehreren Jahrzehnten erforscht wird, sind existierende Verfahren unzureichend, wenn das zu rekonstruierende Objekt dünne Strukturen aufweist. Dies ist in der Praxis ein häufig auftretendes Problem, zum Beispiel bei der Segmentierung von Blutgefäßen in der medizinischen Bildverarbeitung.

In dieser Arbeit wird ein Methode zur Bildsegmentierung und 3D Rekonstruktion vorgestellt, die insbesondere in solchen Fällen von dünnen Strukturen eine deutliche Verbesserung der Ergebnisse im Vergleich zu existierenden Verfahren erzielt. Der Ansatz erlaubt, Nebenbedingungen über die Topologie des zu rekonstruierenden Objektes zu formulieren, insbesondere über die Konnektivität des Objektes. Einer der Hauptbeiträge der Arbeit ist, dass sich die Nebenbedingungen als lineare Nebenbedingungen in einem Ansatz der konvexen Optimierung formulieren lassen. Hierdurch ist es möglich, das resultierende Optimierungsproblem global optimal zu lösen. Im Vergleich zu existierenden Ansätzen ist das vorgestellte Verfahren sehr effizient im Hinblick auf Laufzeit und Speicher, und erlaubt daher die praktische Anwendung bei Fragestellungen größerer Komplexität, zum Beispiel in der medizinischen Bildverarbeitung von dreidimensionalen Datensätzen der Computertomographie oder der 3D Rekonstruktion von dynamischen Szenen. Darüberhinaus wird ein effizientes Projektionsverfahren auf den Lösungsraum vorgestellt, der die Laufzeit besonders in solchen Fällen signifikant verkürzt, in denen Verbindungen größerer Distanz zwischen den einzelnen Teilen des Objektes gefunden werden müssen.

Weitere Teile der Arbeit umfassen die Bestimmung eines Modells für den Datenterm des Segmentierungsmodells mit Hilfe von Gauß-Prozessen, das basierend auf der Unsicherheit der Klassifikation verbessert wird, indem unsicher klassifizierte Bildbereiche identifiziert und dem Benutzer zu Begutachtung präsentiert werden. Ein weiterer Abschnitt der Arbeit stellt einen neuartigen Ansatz zur Objektverfolgung mit Hilfe von phasenmodulierten Tiefenkameras vor.

Stichworte: Bildsegmentierung, 3D-Rekonstruktion, Konvexe Optimierung, Topologische Nebenbedingungen, Konnektivität, Minimalflächen

Acknowledgements

Thanks to all the people who supported me during the years, first and foremost my advisors: My supervisor **Daniel Cremers**, who enabled me to learn and explore the various fields which are touched by computer vision, among them the theory of convex optimisation, differential geometry, computational topology, discrete optimisation and machine learning. Thanks for all the freedom to explore. My co-advisor **Peter Schröder**, who introduced me to the field of discrete differential geometry and enabled me to learn mathematics in such an inspiring and cheerful way, and who managed to find time to listen to my ideas, even with nine hours of time shift. Thanks to **Sebastian Nowozin** and **Jamie Shotton**, for this productive, cooperative and very joyful working experience at Microsoft Research Cambridge, where I learned a lot about probabilistic models and the propagation of light, not to forget the foosball matches.

Then there were my colleagues at the **Computer Vision Group** at TUM, who I thank for inspiring discussions, all the good times during coffee breaks, group retreats, our spare time and the lively atmosphere during the seminars. Thanks to Michael, for his precision in answering mathematical questions, his overall excitement about math, and for proofreading the manuscript. Thanks to Rudolph for the great atmosphere in our office, explanation of uncertainty in machine learning and associated white board sketches, and for proofreading the manuscript. Thank you Thomas and Thomas, for inspiring mathematical discussions that delve into the fascinating world of convex optimisation, topology and measure theory, and for proofreading the manuscript. Thanks to Christian for proofreading the manuscript. Special thanks to Konstantin, for discussing convex optimisation and proximal algorithms with me, and for proofreading the manuscript. At **Caltech**, there were also so many people who supported me: Armeen, Keenan, Michael, Albert, Fernando, Martin, Daniela and Justin.

I am also very grateful to the **TUM Institute for Advanced Studies** for funding and the opportunity to broaden my knowledge at inspiring talks, seminars, and lectures, and to the **European Research Council** for funding.

Contents

Abstract	v
Acknowledgements	ix
Contents	xi

Part I	Introduction	1
1	Introduction	3
1.1	Motivation	3
1.2	Key Contributions	3
1.3	Publications	3
1.4	Collaborations	4
1.5	Notation and Mathematical Symbols	5
2	Introductory Material on Convex Optimization	7
2.1	Convex Analysis	7
2.2	Convex Optimization Problem	7
2.3	Convex Conjugate	8
2.4	Lagrange Duality	9
2.4.1	Weak and Strong and Duality	9
2.4.2	Legendre-Fenchel Transform and the Lagrangian	10
2.5	Primal-Dual Hybrid Gradient Method	11
3	Total Variation and its Minimization	13
3.1	Definition of Total Variation	13
3.2	Minimizing Functionals with Total Variation Regularizer as Saddle Point Problem	14
3.3	Geometric Interpretation of the Total Variation	14
3.4	The Coarea Formula	14
3.5	Total Variation as Regularizer in Image Segmentation and 3D Reconstruction	15
3.6	The Shrinking Bias	15
4	Connectedness as Topological Property	17
4.1	Connectedness of Topological Spaces	17
4.1.1	Path Connectedness	17
4.1.2	Simply Connected Topological Space	18
4.2	Connectedness of a Graph	18
4.2.1	k-Connected Graph	19

Part II	Image Segmentation	21
5	Connectivity Constraints for Image Segmentation	23
5.1	Introduction	23
5.1.1	Related Work	25
5.1.2	Problem Formulation	26
5.2	The Continuous Case: Connectivity Along Geodesics	26
5.2.1	An Image Depending Geodesic Topology	27
5.2.2	Connectivity Constraint as Monotonicity Constraint	27
5.2.3	The Thresholding Theorem	28
5.3	Image Segmentation on the Discrete Domain of a Weighted Graph	30
5.3.1	Gradient and Divergence Operators on Weighted Graphs	30
5.3.2	The Segmentation Model in the Weighted Graph Framework	31
5.3.3	A Primal-Dual Method for Vertex Labelling	32
5.3.4	Comparison of the Primal Dual Algorithm on a Graph and the Graph-Cut Framework	32
5.4	The Connectivity Constraint on a Discrete Domain	33
5.4.1	Discrete Geodesics	35
5.4.2	Legendre-Fenchel Duality	37
5.5	Experimental Results	38
5.6	Conclusion	41
6	A Fast Projection Method for Connectivity Constraints	43
6.1	Introduction	43
6.1.1	Related Work	43
6.1.2	Contribution	44
6.2	Connectivity Constraints in Image Segmentation	44
6.3	Constrained Convex Optimization	45
6.3.1	Optimization via Fenchel Duality	46
6.3.2	Projection onto the Constraint Set	46
6.3.3	Isotonic Regression on a Tree	47
6.4	Experimental Results	50
6.5	Conclusion	53
7	Active Online Learning for Interactive Segmentation Using Sparse Gaussian Processes	55
7.1	Introduction	55
7.1.1	Related Work	56
7.2	Algorithm Overview	57
7.3	Gaussian Process Classification	59
7.3.1	Information-theoretic Sparsification	60
7.4	Segmentation Model	60
7.5	Experimental Results	61
7.5.1	Benefits of the GP classifier	61
7.6	Conclusion	63
Part III	3D Reconstruction and Tracking	65
8	Connectivity in 3D Reconstruction	67
8.1	Introduction	67

8.1.1	Contributions	68
8.1.2	Related Work	68
8.2	3D Reconstruction with Connectivity Constraints	69
8.2.1	Spatio-temporal Multi-view Reconstruction	69
8.2.2	Connectivity Constraints for 3D Reconstruction	70
8.3	Loop Connectivity	71
8.3.1	Loop Connectivity Constraints	72
8.3.2	Handle and Tunnel Loops	73
8.4	Numerical Optimization	76
8.5	Experiments	78
8.6	Conclusion	80
9	The Direct Geometry Approach	81
9.1	Introduction	81
9.1.1	Dense Depth Map Estimation from Multiple Images	81
9.1.2	Extension to Multiple Images	82
9.1.3	Half Quadratic Splitting	83
9.1.4	Dualization of the Data Term	83
9.1.5	Huber loss	84
9.2	Implementation	85
9.3	Experimental Results	85
9.4	Conclusion	85
10	3D Tracking from Raw ToF data	89
10.1	Introduction	89
10.2	Background	91
10.2.1	Phase-modulation time-of-flight	92
10.2.2	Model-based tracking	93
10.3	Method	94
10.3.1	Motion model $P(X_{t+1} X_t)$	94
10.3.2	Observation model $P(Y_t X_t)$ for raw ToF	94
10.4	Implementation and Validation	96
10.5	Experiments	98
10.5.1	Failure of Depth Based Tracking	98
10.5.2	Tracking with Raw ToF Observations	99
10.5.3	Benefit of Equispacing	100
10.6	Discussion	101
10.6.1	Tracking from Raw ToF	101
10.6.2	Equispaced ToF Captures	102
10.6.3	Limitations and Future Work	102
10.7	Conclusion	102
<hr/> Part IV Conclusions and Outlook		103
<hr/> 11 Concluding Remarks		105
<hr/> Part V Appendix		107
<hr/> 12 Projective Geometry and Camera Models		109

13 Sequential Monte Carlo Filtering	111
References	113

Part I.

Introduction

1. Introduction

1.1. Motivation

There seems to be a blind spot, which we would like to shed some light on with this thesis.

1.2. Key Contributions

First practicable approach for connectivity constraints in image segmentation and 3D reconstruction. The method is

- Fast
- Convex
- Independent of initialization (only dependency is the choice of the root node)
- Does not suffer from metrication artefacts (see Section 5.3.4)
- First time that topological constraints have been used for full 3D reconstruction.

Furthermore we present an efficient projection onto the constraint set.

Real-time 3D Reconstruction:

- The first real-time capable approach for dense multiview 3D reconstruction

Raw ToF based Tracking:

- Novel approach for tracking directly from raw ToF data

1.3. Publications

Most of the work in this thesis appears in the following publications:

- [1] J. Stühmer, S. Nowozin, A. Fitzgibbon, R. Szeliski, T. Perry, S. Acharya, D. Cremers, and J. Shotton. Model-based tracking at 300hz using raw time-of-flight observations. In *Proc. International Conference on Computer Vision*. Santiago, Chile, 2015.
- [2] J. Stühmer and D. Cremers. A fast projection method for connectivity constraints in image segmentation. In X.-C. Tai, E. Bae, T. F. Chan, and M. Lysaker (Editors), *Proceedings of the International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, LNCS. 2015.
- [3] M. R. Oswald, J. Stühmer, and D. Cremers. Generalized connectivity constraints for spatio-temporal 3d reconstruction. In *Proc. European Conference on Computer Vision*, pages 32–46. 2014.
- [4] R. Triebel, J. Stühmer, M. Souiai, and D. Cremers. Active online learning for interactive segmentation using sparse gaussian processes. In *German Conference on Pattern Recognition*. 2014.

- [5] J. Stühmer, P. Schröder, and D. Cremers. Tree shape priors with connectivity constraints using convex relaxation on general graphs. In *Proc. International Conference on Computer Vision*. Sydney, Australia, 2013. (Oral Presentation).

The following publications contain parts of my Diploma thesis [125] and are the foundation of chapter 9:

- [6] J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Pattern Recognition (Proc. DAGM)*, pages 11–20. Darmstadt, Germany, 2010. (Oral Presentation). (Part of Diploma Thesis).
- [7] J. Stühmer, S. Gumhold, and D. Cremers. Parallel generalized thresholding scheme for live dense geometry from a handheld camera. In *ECCV Workshop on Computer Vision on GPUs (CVGPU)*. Heraklion, Greece, 2010. (Oral Presentation). (Part of Diploma Thesis).

This thesis extends the work from the Diploma thesis by a dual optimization scheme for the data term and a robust Huber loss for the regularizer.

1.4. Collaborations

The research presented in this thesis was partly conducted as a member of the student researcher visiting program at the California Institute of Technology, Pasadena, USA and the research intern program at Microsoft Research Cambridge, UK.

1.5. Notation and Mathematical Symbols

α	a scalar
x	a vector
\mathbb{R}	the real numbers
I	an image
Ω	the image domain
u^*	with a star we denote an optimal solution to an optimization problem
Σ_μ	the upper levelset of a function thresholded at the value μ
∇	gradient operator
div	divergence operator
$\partial_i f_j$	for a function f , defined over the vertices of a graph, this is the directional derivative along the edge ij
π	projection
Π	orthogonal projection onto a set
G	graph
T_s	tree with root vertex s
\mathcal{N}	neighbourhood of a vertex
V	vertex set of a graph
E	edge set of a graph
E_T	edge set of the tree T
$E_=$	edge set of equality constraints for 2-connectivity constraints
\mathbb{V}	volume domain
\mathbb{T}	time domain
$\mathbb{V} \times \mathbb{T}$	space-time domain
Σ	surface, obtained by thresholding of an indicator function (compare to upper level set)
s	infinitesimal surface element
$int(\Sigma)$	surface interior
$ext(\Sigma)$	surface exterior
S	silhouette
\mathcal{VH}	visual hull
\mathbb{M}	visual hull surface
\mathbb{I}	visual hull surface interior
\mathbb{E}	visual hull surface exterior
h	handle loop (cycle of edges)
t	tunnel loop (cycle of edges)
t^{G_s}	geodesic shortest cycle of the graph that is associated to the tunnel loop t
H	segmented handle

2. Introductory Material on Convex Optimization

This chapter gives an introduction to convex optimization methods, which provide a unifying framework for many of the algorithms derived in this thesis. Throughout the chapter, we follow the definitions and notation of Boyd and Vandenberghe [17] and Parikh and Boyd [103] while adding some additional remarks on the equivalence of the Legendre-Fenchel-transform and Lagrangian multipliers in convex optimization. Furthermore, we add references to relevant algorithms in the field of computer vision when necessary.

2.1. Convex Analysis

Affine Set We consider the set $C \subset \mathbb{R}^n$. Let x and x' be two distinct points in C . The set C is called *affine* if the line through x and x' lies in C , i.e. if for any $x, x' \in C$ and $\theta \in \mathbb{R}$, we have $\theta x + (1 - \theta)x' \in C$ [17].

We extend this concept and consider a set of points x_1, \dots, x_k and a set of coefficients $\theta_1, \dots, \theta_k$ with $\theta_1 + \dots + \theta_k = 1$, i.e. that sum up to one, and call a point $\theta_1 x_1 + \dots + \theta_k x_k$ an *affine combination* of the points x_1, \dots, x_k .

The set of all affine combinations of points in a set $C \subset \mathbb{R}^n$ is called the *affine hull* of C , which we denote with **aff**.

Relative Interior The *relative interior* of a set C is defined as

$$\mathbf{relint} C = \{x \in C \mid B(x, r) \cap \mathbf{aff} C \subset C \text{ for some } r > 0\}, \quad (2.1)$$

where $B(x, r) = \{y \mid \|y - x\| \leq r\}$ is a ball of radius r centred at x . The norm $\|\cdot\|$ can be any norm, and all norms define the same relative interior [17].

Convex Set We consider the set $C \subset \mathbb{R}^n$. Let x and x' be two distinct points in C . The set C is called *convex* if the line segment between x and x' lies in C , i.e. if for any $x, x' \in C$ and $\theta \in \mathbb{R}$ with $0 \leq \theta \leq 1$, it holds that $\theta x + (1 - \theta)x' \in C$ [17].

2.2. Convex Optimization Problem

In mathematical optimization, a *convex optimization problem* is of the form

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{s.t.} && f_i(x) \leq b_i, \quad i = 1, \dots, m. \end{aligned} \quad (2.2)$$

where the functions $f_0, \dots, f_m : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ are *convex*, which means that they satisfy

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

for all $x, y \in \mathbf{dom} f$ and all $\alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$, and $\mathbf{dom} f_i$ are convex sets for $i = 0, \dots, m$.

With $\mathbf{dom} f$ we denote the *effective domain* of a function f which is defined as

$$\mathbf{dom} f = \{x \in \mathbb{R}^n \mid f(x) < +\infty\}, \quad (2.3)$$

thus it is the set of points for which f maps to finite values.

Fundamental for this research is the following property:

Theorem 2.2.1. *Every locally optimal point of (2.2) is globally optimal.*

Proof [17]. To show this, let us assume that x is a locally optimal point of a convex optimization problem. This means x is feasible and for some $R > 0$ it holds that

$$f_0(x) = \inf\{f_0(z) \mid z \text{ feasible, } \|z - x\|_2 \leq R\}. \quad (2.4)$$

Let us assume that x is not globally optimal, then there has to exist a feasible y with $f_0(y) < f_0(x)$. From (2.4) it follows that $\|y - x\|_2 > R$, otherwise the globally optimal point would be within the distance R considered for the locally optimal point.

We consider a point z , a linear combination of x and y

$$z = (1 - \theta)x + \theta y$$

for some

$$0 < \theta < \frac{R}{\|y - x\|_2}$$

such that $\|z - x\|_2 < R$. From convexity of the feasible set it follows that this point z is feasible. Because f_0 is convex it follows that

$$f_0(z) \leq (1 - \theta)f_0(x) + \theta f_0(y)$$

and from $\theta > 0$ and $f_0(y) < f_0(x)$ that

$$f_0(z) < f_0(x)$$

which contradicts (2.4). □

2.3. Convex Conjugate

Let $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$. The *convex conjugate* $f^* : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ of the function f is defined as the supremum

$$f^*(y) = \sup_{x \in \mathbf{dom} f} (y^T x - f(x)), \quad (2.5)$$

It is also known as the *Legendre-Fenchel transform* of f .

The Fenchel *bi-conjugate* f^{**} yields the *convex envelope*, the largest closed convex underapproximation of f [17]. Furthermore, an important property of the bi-conjugate is that $f = f^{**}$ iff f is convex and lower semi-continuous, which in finite dimensions holds for all convex functions with closed *epigraph* and nonempty domain, where the *epigraph* of a convex function f is defined as

$$\mathbf{epi}(f) = \{(x, d) \in \mathbb{R}^{n+1} \mid f(x) \leq d\}. \quad (2.6)$$

2.4. Lagrange Duality

Let's consider the following *constrained optimization problem* [17]

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{s.t.} && f_i(x) \leq 0, \quad i = 1, \dots, m \\ & && h_i(x) = 0, \quad i = 1, \dots, p. \end{aligned} \tag{2.7}$$

The variable $x \in \mathbb{R}^n$ is the *optimization variable* and the function $f_0 : \mathbb{R}^n \mapsto \mathbb{R}$ is called the *objective function*. The inequalities $f_i(x) \leq 0$ are called the *inequality constraints* and the functions $f_i(x) : \mathbb{R}^n \mapsto \mathbb{R}$ are the *inequality constraint functions*. Correspondingly, the equations $h_i(x) = 0$ are called the *equality constraints* and the functions $h_i(x) : \mathbb{R}^n \mapsto \mathbb{R}$ are the *equality constraint functions*. Because the right hand side of the constraints is 0, this optimization problem is considered being in *standard form*. We do not necessarily assume that the optimization problem is convex.

The associated *Lagrangian* $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ is defined as

$$L(x, \lambda, \mu) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \mu_i h_i(x). \tag{2.8}$$

The variables λ_i and μ_i are called *Lagrange multiplier* and their vectors λ and μ are called *dual variables*.

The *Lagrange dual function* $g : \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R} \cup \{-\infty\}$ is defined as the minimum value of the Lagrangian over x :

$$g(\lambda, \mu) = \inf_x L(x, \lambda, \mu) = \inf_x \left(f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \mu_i h_i(x) \right). \tag{2.9}$$

For $\lambda \geq 0$ the Lagrange dual yields a lower bound on the optimal value of (2.7). Maximization of this lower bound leads to the *Lagrange dual problem* associated to (2.7)

$$\begin{aligned} & \sup_{\lambda, \mu} && g(\lambda, \mu) \\ & \text{s.t.} && \lambda \geq 0. \end{aligned} \tag{2.10}$$

2.4.1. Weak and Strong and Duality

So far we only have the guarantee, that the optimal value of the Lagrange dual problem (2.10), which we denote with d^* , gives the maximum lower bound on the optimal value of the original constrained optimization problem (2.7), which we denote with p^* :

$$d^* \leq p^* \tag{2.11}$$

This property of the Lagrange dual holds even when the original problem is not convex and is called *weak duality*.

When this property holds as equality

$$d^* = p^*$$

we say that *strong duality* holds. In this case, the *optimal duality gap* $p^* - d^*$ is zero, and we say that the maximum lower bound retrieved by the Lagrange dual is tight. Strong duality holds when special conditions on the constraints are fulfilled, the so called *constraint qualifications*.

In case the primal problem is convex, with f_0, \dots, f_m convex, and h_1, \dots, h_p affine, one of these constraint qualifications is *Slater's condition*, which holds when there exists an x in the relative interior of the domain, $x \in \mathbf{relint} \mathcal{D}$, such that it is strictly feasible:

$$f_i(x) < 0, \quad i = 1, \dots, m, \quad \text{and} \quad h_j(x) = 0, \quad j = 1, \dots, p.$$

When also some of the inequality constraint functions f_i are affine, Slater's condition can be refined: Let the first k functions f_1, \dots, f_k be affine, then strong duality holds provided that there exists an x in the relative interior of the domain with

$$f_i(x) \leq 0, \quad i = 1, \dots, k, \quad f_i(x) < 0, \quad i = k + 1, \dots, m, \quad \text{and} \quad h_j(x) = 0, \quad j = 1, \dots, p.$$

The interesting property of this refined Slater condition is, that Slater's condition reduces to feasibility when all of the constraint functions are affine. Thus feasibility of the solution of a convex optimization problem with affine constraints is already a sufficient condition for strong duality.

2.4.2. Legendre-Fenchel Transform and the Lagrangian

The Legendre-Fenchel transform and the Lagrangian are closely related. Let us again consider the constrained optimization problem in standard form (2.7)

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{s.t.} && f_i(x) \leq 0, \quad i = 1, \dots, m \\ & && h_i(x) = 0, \quad i = 1, \dots, p. \end{aligned} \tag{2.12}$$

We define the *indicator functions* of the inequality constraints as

$$\delta_{\leq 0}(f_i(x)) = \begin{cases} 0 & \text{if } f_i(x) \leq 0, \\ +\infty & \text{else.} \end{cases} \tag{2.13}$$

Correspondingly, the indicator functions of the equality constraints are defined as

$$\delta_{=0}(h_i(x)) = \begin{cases} 0 & \text{if } h_i(x) = 0, \\ +\infty & \text{else.} \end{cases} \tag{2.14}$$

Recall that the conjugate, also called the Legendre-Fenchel transform, $f^* : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ of a function $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ is defined as

$$f^*(y) = \sup_{x \in \mathbf{dom} f} (y^T x - f(x)). \tag{2.15}$$

We get as *conjugate of the inequality indicator function*

$$\delta_{\leq 0}^*(y) = \sup_x (y^T x - \delta_{\leq 0}(x)) = \begin{cases} 0 & \text{if } y \geq 0, \\ +\infty & \text{else} \end{cases} \tag{2.16}$$

$$= \delta_{\geq 0}(y). \tag{2.17}$$

The *bi-conjugate of the inequality indicator function* is again the original indicator function itself

$$\delta_{\leq 0}^{**}(y) = \sup_x (y^T x - \delta_{\leq 0}^*(x)) \tag{2.18}$$

$$= \sup_{x \geq 0} (y^T x) = \begin{cases} 0 & \text{if } y \leq 0, \\ +\infty & \text{else} \end{cases} \tag{2.19}$$

$$= \delta_{\leq 0}(y). \tag{2.20}$$

Accordingly, we get as *conjugate of the equality indicator function*

$$\delta_{=0}^*(y) = \sup_x (y^T x - \delta_{=0}(x)) \quad (2.21)$$

$$= 0 \quad \forall y. \quad (2.22)$$

The *bi-conjugate of the equality indicator function* again equals the original indicator function

$$\delta_{=0}^{**}(y) = \sup_x (y^T x - \delta_{=0}^*(x)) \quad (2.23)$$

$$= \sup_x (y^T x) = \begin{cases} 0 & \text{if } y = 0, \\ +\infty & \text{else} \end{cases} \quad (2.24)$$

$$= \delta_{=0}(y). \quad (2.25)$$

Now we can rewrite the constrained optimization problem (2.7) using the indicator functions of the constraints as

$$\inf_x f_0(x) + \sum_{i=1}^m \delta_{\leq 0}(f_i(x)) + \sum_{i=1}^p \delta_{=0}(h_i(x)) \quad (2.26)$$

With the bi-conjugate of the indicator functions we get

$$\begin{aligned} &= \inf_x f_0(x) + \sum_{i=1}^m \delta_{\leq 0}^{**}(f_i(x)) + \sum_{i=1}^p \delta_{=0}^{**}(h_i(x)) \\ &= \inf_x f_0(x) + \sum_{i=1}^m \sup_{\lambda \geq 0} (\lambda f_i(x)) + \sum_{i=1}^p \sup_{\mu} (\mu h_i(x)) \end{aligned} \quad (2.27)$$

We introduce individual variables λ_i and μ_i for each constraint function and arrive at the associated Lagrangian

$$\begin{aligned} &= \inf_x \sup_{\lambda_i \geq 0} \sup_{\mu_i} f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \mu_i h_i(x) \\ &= \inf_x \sup_{\lambda_i \geq 0} \sup_{\mu_i} L(x, \lambda, \mu). \end{aligned} \quad (2.28)$$

Later in this thesis we will use this framework to solve constrained convex optimization problems with affine inequality constraints. In case of affine constraints, feasibility of a solution is already sufficient for Slater's condition and we have strong duality. In this case above result is equal to the Lagrange dual

$$= \sup_{\lambda \geq 0} \sup_{\mu} \inf_x L(x, \lambda, \mu) \quad (2.29)$$

and a feasible solution to the Lagrange dual yields an optimal solution to the primal problem.

2.5. Primal-Dual Hybrid Gradient Method

In [25] the authors generalize their algorithm for minimizing total variation regularized functionals [24, 106, 107] to general optimization problems of the structure

$$\min_{x \in X} \max_{y \in Y} G(x) + \langle Kx, y \rangle - F^*(y), \quad (2.30)$$

where X and Y are two finite-dimensional real vector spaces equipped with the inner product $\langle \cdot, \cdot \rangle$, the map $K : X \mapsto Y$ is a continuous linear operator, and $G : X \mapsto [0, +\infty)$ and $F^* : Y \mapsto [0, +\infty)$ are proper convex lower-semicontinuous functions.

The authors provide three different variants of an algorithm to solve optimization problems of this structure. The central idea of the algorithm is to iterate a gradient ascent in the primal variable and a gradient descent in the dual variable. Because F and G do not need to be smooth and differentiable, the gradient steps are evaluated using the prox operator:

The **prox**-operator of a closed proper convex lower-semicontinuous function f is defined as

$$\mathbf{prox}_{\lambda f}(v) = \arg \min_x f(x) + \frac{1}{2\lambda} \|x - v\|_2^2. \quad (2.31)$$

The first variant (Algorithm 1 in [25]) can be applied to the most general problem, the variant 2 allows accelerated convergence when either G or F^* are strongly convex, and algorithm 3 can be applied when both G and F^* are strongly convex functions. For the problems studied in this thesis strong convexity does not hold, thus here we present the variant for this most general case.

Algorithm 1 Primal Dual Hybrid Gradient method from [25]

- 1: Initialize with $x^0, y^0 \in X \times Y$ and $\bar{x}^0 = x^0$.
- 2: Iterate

$$y^{n+1} = \mathbf{prox}_{\sigma F^*}(y^n + \sigma K \bar{x}^n) \quad (2.32)$$

$$x^{n+1} = \mathbf{prox}_{\tau G}(x^n - \tau K^* y^{n+1}) \quad (2.33)$$

$$\bar{x}^{n+1} = x^{n+1} + \theta (x^{n+1} - x^n) \quad (2.34)$$

The step sizes $\tau > 0$ and $\sigma > 0$ can be chosen by a diagonal preconditioning scheme as described in [105]. In this case, one gets instead of a unique τ and σ the diagonal matrices $T = \text{diag}(\tau)$ and $\Sigma = \text{diag}(\sigma)$, with

$$\tau_j = \frac{1}{\sum_{i=1}^m K_{i,j}^{2-\alpha}}, \quad \sigma_i = \frac{1}{\sum_{j=1}^n K_{i,j}^\alpha}, \quad (2.35)$$

with $n = \dim X$ and $m = \dim Y$ for some $\alpha \in [0, 2]$.

The primal dual hybrid gradient method can be interpreted as an approximative Douglas-Rachford splitting [37] on the dual problem which is also equivalent to the alternating direction method of multipliers (ADMM) [50, 52] on the primal problem. An overview of proximal algorithms can be found in [103] and a survey about ADMM and its applications in [16].

3. Total Variation and its Minimization

With the background material on convex optimization at hand, in this chapter we focus on a particular optimization problem, the minimization of the *total variation* of a function of *bounded variation*. It turns out that we can formulate the total variation minimization problem as a convex-concave saddle point problem, that can be optimized with the primal dual splitting algorithm discussed in the end of the previous chapter. Further information on the topic, especially in the context of image analysis, can be found in the publications [24, 25, 106, 107], and the introduction by Chambolle *et al.* [23].

3.1. Definition of Total Variation

Let Ω be an open subset of \mathbb{R}^n . The *total variation* of a function $u : \Omega \mapsto \mathbb{R}$ is defined as

$$TV(u, \Omega) = \sup \left\{ - \int_{\Omega} u(x) \operatorname{div} \phi(x) \, dx : \phi \in C_c^\infty(\Omega, \mathbb{R}^n), |\phi(x)| \leq 1, \forall x \in \Omega \right\}, \quad (3.1)$$

where $|\phi(x)|$ is the ℓ^2 norm of the vector valued function ϕ evaluated at x and C_c^∞ is the set of arbitrarily often continuously differentiable functions with compact support.

We choose this definition of the norm to clarify that we actually evaluate the ℓ^2 norm of $\phi(x)$ point-wise on Ω . In the literature, *e.g.* [2], the constraint on ϕ is also defined as $\|\phi\|_\infty \leq 1$, which denotes the essential supremum norm of $|\phi(x)|$ on Ω . To avoid confusion with the point-wise supremum norm, we instead propose above definition.

Indeed, our definition $|\phi(x)| \leq 1, \forall x \in \Omega$ is equivalent to $\|\phi\|_\infty \leq 1$, which in [20] is defined more clearly as

$$\|\phi\|_\infty = \operatorname{ess\,sup}_{x \in \Omega} \left(\sum_i^n |\phi_i(x)|^2 \right)^{\frac{1}{2}} \quad (3.2)$$

This norm takes the maximum over x of the point-wise ℓ^2 norm of ϕ . It is easy to see that $\|\phi\|_\infty \leq 1$ exactly holds iff $|\phi(x)| \leq 1, \forall x \in \Omega$.

We say that a function $u : \Omega \mapsto \mathbb{R}$ in L^1 has *bounded variation* on Ω when $TV(u, \Omega) < +\infty$ and we write $u \in \mathcal{BV}$.

The function ϕ can be interpreted as distributional derivative of u . For a differentiable function u it follows from the divergence theorem that

$$\int_{\Omega} u(x) \operatorname{div} \phi(x) \, dx = - \int_{\Omega} \nabla u(x) \cdot \phi(x) \, dx. \quad (3.3)$$

For the total variation we take the supremum over ϕ with the supremum norm $|\cdot|_{L^\infty} \leq 1$. Thus we get for differentiable functions u

$$\sup_{|\phi|_{L^\infty} \leq 1} - \int_{\Omega} u(x) \operatorname{div} \phi(x) \, dx = \sup_{|\phi|_{L^\infty} \leq 1} \int_{\Omega} \nabla u(x) \cdot \phi(x) \, dx = \int_{\Omega} |\nabla u(x)| \, dx, \quad (3.4)$$

where the last equality holds in the limit when $\phi(x) \rightarrow \frac{\nabla u}{|\nabla u|_{L^2}}$.

3.2. Minimizing Functionals with Total Variation Regularizer as Saddle Point Problem

In the previous section we saw that the total variation of a differentiable function is defined as the supremum over ϕ . Minimizing the total variation of a differentiable function u thus leads to the saddle point problem

$$\inf_u \int_{\Omega} f(u) + |\nabla u(x)| \, dx = \inf_u \sup_{|\phi|_{L^\infty} \leq 1} \int_{\Omega} f(u) - u(x) \operatorname{div} \phi(x) \, dx. \quad (3.5)$$

On a discrete finite domain Ω and for $\mathbf{dom}(u)$ convex this problem can be solved with the primal dual optimization method introduced in Section 2.5.

3.3. Geometric Interpretation of the Total Variation

In the following we will see that minimizing the total variation has a geometric interpretation and allows to compute sets of minimal surface. First, we define an indicator function of a set.

Let $S \subseteq \Omega \subset \mathbb{R}^n$. The *indicator function* $\mathbb{1}_S : \Omega \mapsto \{0, 1\}$ of S is defined as

$$\mathbb{1}_S(x) = \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{else.} \end{cases} \quad (3.6)$$

The *perimeter* of a measurable set $S \subseteq \Omega \subset \mathbb{R}^n$ is defined as $Per(S, \Omega) = \mathcal{H}^{n-1}(\partial S)$, the $(n - 1)$ -dimensional Hausdorff measure $\mathcal{H}^{n-1}(\cdot)$ of the boundary ∂S of S . It is a geometric measure, *e.g.* for $n = 2$ it measures the length of the curve outlining the set S , for $n = 3$ it measures the surface area of the boundary surface of S .

We will see in the following section that the $(n - 1)$ -dimensional Hausdorff measure of the boundary of a set is equivalent to the total variation $TV(\mathbb{1}_S)$ of the indicator function of the set. This allows to measure boundary lengths of subsets in \mathbb{R}^2 and surface areas of subsets of \mathbb{R}^3 by computing the total variation of the indicator function.

3.4. The Coarea Formula

Recall that a convex function needs to have a convex domain. The indicator function maps to the discrete domain $\{0, 1\}$, which is not a convex set. To be able to minimize the total variation with convex optimization, we therefore often define a relaxed indicator function $u : \Omega \mapsto [0, 1]$ of bounded variation, which maps to the continuous interval between 0 and 1.

A relation between the geometric measure of a perimeter and the total variation of a continuous \mathcal{BV} -function is given by the *coarea formula* [43, 45]

$$\int_{\Omega} |\nabla f(x)| \, dx = \int_{-\infty}^{\infty} TV(\mathbb{1}_{\{f \geq \mu\}}, \Omega) \, d\mu = \int_{-\infty}^{\infty} Per(\{x : f(x) \geq \mu\}, \Omega) \, d\mu. \quad (3.7)$$

Thus the total variation of the continuous function u is the integral over the length of all its level lines. Note that whenever we consider functions u whose range is restricted to the interval $[0, 1]$ we have to evaluate the integral only on this interval and get

$$\int_{\Omega} |\nabla u(x)| \, dx = \int_0^1 Per(\{x : u(x) \geq \mu\}, \Omega) \, d\mu. \quad (3.8)$$

It follows from the definition of the perimeter and the definition of functions with bounded variation that a set has finite perimeter in Ω iff $\mathbb{1}_S \in \mathcal{BV}(\Omega)$, i.e. its indicator function has bounded variation.

3.5. Total Variation as Regularizer in Image Segmentation and 3D Reconstruction

In image segmentation, the goal is to partition an image into meaningful parts. It is one of the best studied problems in computer vision, as image segmentation is often the first step to acquire semantic information from an image. While here we focus on the two region image segmentation problem, we want to mention recent research results that allow a convex relaxation of the multilabel image segmentation problem [54, 90, 106, 108].

Weighted partition with minimum perimeter Let $f : \Omega \mapsto \mathbb{R}$. We define the optimal weighted partition with minimum perimeter as the set $S \subseteq \Omega$ that minimizes

$$\min_{S \subseteq \Omega} \int_S f(x) \, dx + \lambda \text{Per}(S, \Omega), \quad (3.9)$$

for a given $\lambda \geq 0$. We call the set S the *foreground region* or *object* and $\Omega \setminus S$ the *background region*. Both sets S and $\Omega \setminus S$ define a partition of Ω which we call a *segmentation*. Because the function f usually depends on the image data, the first term $\int_S f(x) \, dx$ is called *data term*.

For $\lambda = 0$ above problem is easy to solve, and the optimal solution can be achieved by choosing the set $S = \{x : f(x) < 0\}$. However, in image segmentation often the data term is affected by noise and simple thresholding of the data term would result in an irregular set. Therefore we regularize the solution by setting $\lambda > 0$. The second term in (3.9) is also called *boundary length regularizer* and favours a smooth partition.

For $\lambda > 0$ we cannot achieve a minimum partition of (3.9) by simple thresholding of f . Therefore we formulate the minimum weighted partition problem as a convex relaxation using the relaxed indicator function $u : \Omega \mapsto [0, 1]$ of bounded variation

$$\min_{u \in \mathcal{BV}(\Omega; [0,1])} \int_{\Omega} f(x)u(x) \, dx + \lambda \text{TV}(u, \Omega). \quad (3.10)$$

In Section 5.2.3 we will see that by thresholding the solution of (3.10) by choosing the set $\{x : u(x) > \mu\}$ for any $\mu \in [0, 1)$ we get an optimal solution of the minimum weighted partition problem (3.9).

3.6. The Shrinking Bias

The boundary length regularizer achieves good results for segmenting compact objects in practice. However, this approach fails when the object contains thin structures, as these structures have a higher boundary length in comparison to their area than more round, compact objects.

The regularizer therefore tends to favour compact round objects, while smoothing fine detailed features of the boundary. This effect is called the *shrinking bias* which leads to a shrinking of the foreground region to minimize its boundary length.

To overcome the shrinking bias, researchers have proposed to penalize the curvature of the boundary instead of its length, e.g. [39, 55, 115, 116, 117]. However, because curvature is a

second order measure of the boundary, this usually leads to higher order cost functions that are hard to optimize efficiently.

4. Connectedness as Topological Property

In this chapter we will give an introduction to a fundamental topological property, the connectedness of a topological space and the connectedness of graphs. We will study different properties of connectedness, with a special look on path connectedness, which is underlying the connectivity constraints developed in this thesis. The following material follows the introductory literature on topology of [88, 131], and on graph theory of [34].

4.1. Connectedness of Topological Spaces

Given X is a topological space. We say that X is *connected* if there exists no separation of X into a pair of nonempty, disjoint, open subsets $U, V \subset X$ such that $X = U \cup V$.

This is the definition of a connected *space*, in the following we will study the connectedness of *subsets* of a topological space, for which we define the following topology.

Subspace Topology [88] Let $A \subset X$ be a subset of the topological space X . The *subspace topology* \mathcal{S}_a on A is defined as

$$\mathcal{S}_a = \{U \subset A : U = A \cap V \text{ for some open subset } V \subset X\}, \quad (4.1)$$

i.e. the open subsets of \mathcal{S}_a are the intersections of the open subsets of X with A .

We call a subset A of a topological space X *connected on X* if A is connected with respect to the subspace topology \mathcal{S}_a .

4.1.1. Path Connectedness

In this section, we will introduce an easier to use sufficient condition for connectedness, the *path connectedness*. First let's consider the connectedness of mappings on a topological space.

Theorem 4.1.1. Main Theorem on Connectedness [88] Let X, Y be topological spaces and let $f : X \mapsto Y$ be a continuous map. If X is connected, then $f(X)$ is connected.

Proof. [88] Let $f(X)$ be not connected, then there exist two open sets $U, V \subset Y$ which intersections with $f(X)$ are nonempty and disjoint and for which $f(X) \subset U \cup V$. Lets consider the preimages of those subsets $f^{-1}(U)$ and $f^{-1}(V)$. It follows immediately that these provide a separation of X , so X is not connected. \square

Now we are able to give a definition for *path connectedness*, which is much simpler than the definition of connectedness of a space, but yet provides a sufficient condition for its connectedness.

Path Connectedness [88] Let X be a topological space and $x, x' \in X$ two points in X . A path in X from a x to x' is a continuous map $C_x^{x'} : [0, 1] \mapsto X$ with $C_x^{x'}(0) = x$ and

$C_x^{x'}(1) = x'$. We say that X is *path connected* if for every $p, q \in X$, there is a path in X from p to q .

An important property that we will use in the following is, that path connectedness of a topological space X implies connectedness of X .

Furthermore, path connectivity is transitive in the sense that if there is a path connecting two points a and b and there is a path connecting the points b and c , then there also exists a path connecting a and c .

This becomes obvious when we define the path $C_a^c : [0, 1] \mapsto X$ from a to c as [131]

$$C_a^c(s) = \begin{cases} C_a^b(2s), & \text{if } s \in [0, \frac{1}{2}], \\ C_b^c(2s - 1), & \text{if } s \in [\frac{1}{2}, 1], \end{cases} \quad (4.2)$$

where $C_a^b : [0, 1] \mapsto X$ is the path from a to b and $C_b^c : [0, 1] \mapsto X$ is the path from b to c .

4.1.2. Simply Connected Topological Space

There are several definitions for *simply connectedness*, we choose an intuitive definition that is based on the homotopy of paths. First we provide the definition of a homotopy in general and then define the homotopy for paths.

Homotopy [88] Let X and Y be topological spaces, and let $f, g : X \mapsto Y$ be continuous maps. A *homotopy* from f to g is a continuous map $H : X \times [0, 1] \mapsto Y$ such that

$$H(x, 0) = f(x); \quad H(x, 1) = g(x), \quad (4.3)$$

for all $x \in X$.

Now we study the homotopy of two paths:

Path Homotopy Let X be a topological space and $x, x' \in X$ two points in X . We consider two paths f and g in X from x to x' . The *path homotopy* $H : [0, 1] \times [0, 1] \mapsto X$ from f to g is a family of paths with fixed endpoints x and x' , i.e. $H(0, t) = x$ and $H(1, t) = x'$ for all $t \in [0, 1]$, that is a continuous map from f to g , i.e. $H(s, 0) = f(s)$ and $H(s, 1) = g(s)$ for all $s \in [0, 1]$. We say that f and g are *path homotopic*, if there exists a path homotopy between them, and write $f \sim g$.

This allows us to formulate the definition of a *simply connected* topological space:

We call a topological space X *simply connected* when any two paths in X with the same initial and terminal points are path homotopic.

Hence all paths with the same initial and terminal points form an equivalence class. Simply connected in this case means that there is only a single equivalence class of paths.

This allows an intuitive understanding why a simply connected subset $A \subset X$ may not contain a hole, i.e. may not enclose another subset $B \subset X$ which is not in A . In this case no continuous mapping between any two paths "on different sides of B " exists that does not leave A .

4.2. Connectedness of a Graph

While the previous section considered continuous topological spaces, in this section we provide definitions to work on discrete graphs as topological spaces. We follow the notation and definitions in [34].

A graph contains of a set of vertices V and edges $E \subset V \times V$. We denote the vertex set of G with $V(G)$ and the edge set of G with $E(G)$. We call a vertex *incident* with an edge e if $v \in e$. The edges incident to a vertex v define a local neighbourhood of *adjacent vertices* or *neighbours* of v . Two vertices i and j are *adjacent*, when there exists an edge $ij \in E$. The number of adjacent vertices is the *degree* of a vertex.

Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. If $V(G') \subset V(G)$ and $E(G') \subset E(G)$, then we call G' a *subgraph* of G , and write $G' \subset G$.

If the subgraph $G' \subset G$ contains all edges in E with both endpoints in V' , i.e. all $ij \in E(G)$ for $i, j \in V[G']$, then we call G' an *induced subgraph* of G . We say that the set V' *induces* the subgraph G' in G , and denote the induced subgraph with $G' =: G[V']$. Thus any set of vertices $U \subset V$ induces a subgraph $G[U]$, which edges are those edges of G with both endpoints in U .

We define a *path in a graph* G as a non-empty graph $P = (V_p, E_p)$ with $V_p \subset V(G)$ and $E_p \subset E(G)$ and

$$V_p = \{x_0, x_1, \dots, x_k\} \quad E = x_0x_1, x_1x_2, \dots, x_{k-1}x_k \quad (4.4)$$

where all x_i are distinct from other. We call the vertices x_0 and x_k *linked* by P .

Let the graph $P = x_0 \dots x_{k-1}$ be a path with $k \geq 3$, then we call the graph $C := P \cup x_{k-1}x_0$, i.e. a path where both ends are connected, a *cycle*.

A connected *acyclic* graph, which does not contain any cycles, is called a *tree*. Vertices of degree 1 in a tree are called *leaves* or *leave vertices*. Furthermore, any interior vertex, i.e. a vertex which is not a leaf, of a tree has a degree of at least degree 2.

We can now define the property of *connectedness* for a graph [34], which reminds us of the definition of path-connectivity of a topological space.

A non-empty graph G is called *connected* if any two of its vertices are linked by a path in G . If the induced subgraph $G[U]$ of a subset $U \subset V(G)$ is connected, we also call U itself connected in G .

4.2.1. k-Connected Graph

We will use a special type of connectedness of graphs, the *k-connectedness*, sometimes called *k-vertex-connectedness*.

A graph $G = (V, E)$ is called *k-connected* for $k \in \mathbb{N}$ if $|V| > k$ and $G[V \setminus X]$ is connected for every set $X \subseteq V$ with $|X| < k$. Thus, G is connected when less than k vertices are removed from its vertex set. In case G is *2-connected*, we also call G *biconnected*. The greatest integer k for which G is k -connected is the *connectivity* of G .

In this chapter we provided definitions of connectivity as a topological property of a continuous space and on the discrete topology of a graph. This chapter also concludes the introductory material.

In the next part of this thesis we will see how the mathematical tools described so far can be used to solve image partition problems with the constraint, that the foreground partition is connected.

Part II.

Image Segmentation

5. Connectivity Constraints for the Segmentation of 2D and 3D Images

In this chapter, we present one of the first practicable methods to include connectivity constraints into image segmentation and 3D reconstruction. We propose to reformulate the connectivity constraint along geodesics, which in the discrete domain results in a fixed topology of a discrete graph, in this case a geodesic shortest path tree. While solving the original problem is NP-hard, the reformulated problem can be efficiently solved to global optimality. We show how a-priori information about the geometry of the structure of interest can be included when constructing the shortest path tree. To solve the resulting labelling problem, we generalize a recent primal-dual algorithm for continuous convex optimization to an arbitrary graph. The results presented in this chapter have been published in [130].

The chapter is organised as follows: first, we give an overview on related work on topological constraints in image segmentation. Then we describe how the geodesic shortest path tree is constructed with a distance measure that is related to the image data and the bending energy of each path in the tree. We show how to formulate a global connectivity prior as a local constraint on this tree. The connectivity constraint results in a linear constraint on the labelling function and thus preserves convexity of the image segmentation problem. This allows to compute a globally optimal solution. In the end of the chapter we present results on data from medical imaging in angiography, retinal blood vessel segmentation and user interactive image segmentation.

5.1. Introduction

The task of image segmentation, the separation of an image into meaningful parts, is one of the most important and well studied problems in image processing and computer vision. While state-of-the art segmentation methods [18, 59, 135] perform well for segmenting compact objects, their performance on thin and elongated structures is often not satisfying. The commonly used length regularizer suppresses small structures and the correct topology cannot be reconstructed.

To overcome this shrinking bias, recently two different approaches have been suggested in the literature. First, curvature based measures have attracted the interest of researchers in computer vision to include them in image segmentation frameworks [39, 55, 117]. However, introducing these regularizers into segmentation algorithms lead to higher order cost functions, which are hard to optimize.

Another way to preserve thin structures is to use topological constraints. A special subclass of these constraints are connectivity constraints, which ensure the connectedness of a labelled region and therefore allow that thin connections between foreground regions are preserved in the final segmentation result. To overcome the limitation of topology preserving level set methods [63], that only locally optimal solutions can be achieved, recent approaches include topological constraints in random field models [28, 97, 138]. So far, these methods only allow to compute an approximate solution of the global optimization problem.

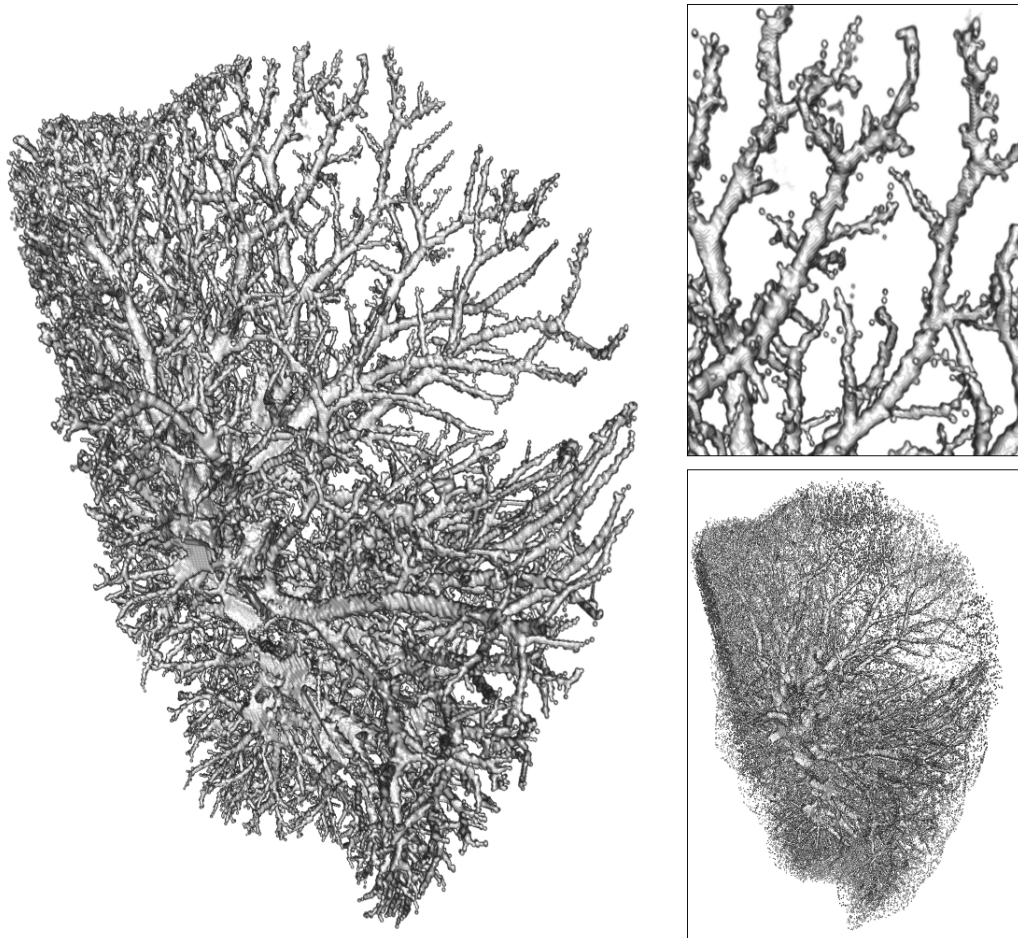


Figure 5.1.: By modelling the object of interest as a connected tree, the method is well suited for medical image segmentation tasks, in this case the segmentation of a blood vessel tree in angiography. Left image and top right: Lung vessel tree segmented with the proposed method. Bottom right: Segmentation result without connectivity constraint.

5.1.1. Related Work

Topology preserving constraints have been recently proposed for different algorithmic frameworks. For the graph cut [19] framework, Zeng *et al.* [143] present an extension, that allows to preserve the topology of the result with respect to an initial segmentation. Beginning on a coarse scale, their method preserves the topology of the initial segmentation during refinement. A similar approach was proposed by Han *et al.* [63] for the level set framework. The drawback of both methods is that they depend on the initialization and therefore only reach a local optimum.

Vicente *et al.* [138] introduce connectivity priors into interactive segmentation in a Markov random field framework and enforce connectivity to user given seed points. The authors show that the original problem is NP-hard and propose a greedy approximation scheme consisting of a Dijkstra algorithm where in every expansion step a graph cut needs to be solved. Their method also only reaches a local optimum.

Chen *et al.* [28] propose to alternately solve a graph cut and modify the unary terms based on a level-set representation until predefined topological constraints are fulfilled. However, they do not find minimal connections, for which the integral along the curve is minimized, instead they minimize the value of the maximum penalty along the curve which can lead to undesired results if this penalty is very high, as the path cost does not depend on the values of the data term at other positions than the maximum. Furthermore, the runtime complexity of this method prevents to use it for large scale problems.

Recently, methods that aim to reach a global optimum were proposed by different authors. First, Nowozin and Lampert [97] propose to formulate the image segmentation problem with topological constraints as a linear program relaxation. However, even for small image sizes the runtime complexity of the method does not scale well and the relaxation is not tight. In contrast to the method presented in this publication, their method is not suitable for large scale problems in 3D segmentation.

Gulshan *et al.* [61] introduce geodesic star shape priors into the graph-cut framework. The solution of the segmentation is restricted to the shape of a geodesic star around an input seed, while the geodesic distance depends on the image gradient. If multiple input seeds are given, the foreground segment takes the form of a geodesic forest, the union of the geodesic stars for every seed. A drawback of their method is that because they solve for the boundary length regularizer using the graph-cut framework their method is affected by a metrication error that depends on the discretization of the pixel neighbourhood.

One application field of methods that preserve thin structures is in angiography, where the object of interest that should be segmented are blood vessels. Some of the most prominent existing methods for this special task are based on geodesic shortest paths. By using a local anisotropic metric and modelling the segmentation task as a path search problem with varying radius, of circles for 2D images [11] and spheres for 3D data [10], such methods are well suited for the special case of tubular structures like blood vessels, but at the same time are restricted to this specific task. Instead of modelling the objects that should be segmented explicitly as connected paths some authors propose to first pre-process the image data with filters that show a strong response in areas where elongated structures are present [47, 82]. In the recent work of Bauer *et al.* [9] a similar approach leads to an explicit model of short tubular segments that are in a second step connected to a whole tree of branching tubular structures. Therefore a connection confidence measure to join adjacent tube segments is defined, that depends on the distance and joining angle of the segments. The resulting minimization problem is solved by using the graph cut algorithm [19]. For a review on recent work in the particular application domain of blood vessel segmentation see [91].

Instead of optimizing over the boundary as proposed by Kass *et al.* [75] or using a level set formulation [27, 63], we introduce our connectivity constraints for the convex image segmentation framework of Chan *et al.* [26]. This has the benefit that, because the constraint can be formulated as linear constraints, the whole image segmentation problem with connectivity constraints remains a convex optimization problem, and thus allows to be solved to global optimality. In comparison to image segmentation methods that are based on the graph cut framework [19, 28], our methods does not suffer from discretization artefacts and instead measures the Euclidean norm of the boundary length.

5.1.2. Problem Formulation

Given an image I with the domain Ω , a bounded connected subset of \mathbb{R}^m , we wish to solve the constrained optimization problem

$$\min_{l:\Omega\rightarrow\{0,1\}} \int_{\Omega} f(x)l(x) + \lambda Per(\Sigma_l) \, dx \quad (5.1)$$

$$\text{s.t.} \quad \forall x, x' \in \Sigma_l : \exists C_x^{x'} \in \Sigma_l \quad (\text{C0})$$

where $\Sigma_l \subseteq \Omega$ is the foreground segment, the part of the image which was labelled by the labelling function $l : \Omega \mapsto \{0, 1\}$

$$\Sigma_l = \{x \in \Omega : l(x) = 1\}. \quad (5.2)$$

We assume that we are given a probabilistic model that depends on the image data and describes the likelihood for foreground and background

$$f(x) = \log \frac{P(I(x)|l(x) = 0)}{P(I(x)|l(x) = 1)} \quad (5.3)$$

for every $x \in \Omega$, i.e. pixel of a 2D image or voxel of a 3D image. The second term $Per(\Sigma_l)$ is the perimeter of the foreground segment Σ_l , the $m - 1$ dimensional Hausdorff measure of its boundary, details can be found in the introduction in Section 3.3. With $C_x^{x'}$ we formalize a connected trajectory from x to x' as a continuous function $C_x^{x'} : [0, 1] \mapsto \Omega$ with $C_x^{x'}(0) = x$ and $C_x^{x'}(1) = x'$.

The solution of the optimization problem should satisfy the connectivity constraint (C0):

For each pair of points $x, x' \in \Omega$ that belong to the foreground Σ_l there must exist a connected path from x to x' such that all $p \in C_x^{x'} \subset \Omega$ in the path between x and x' belong to the foreground.

This constraint is equivalent to the definition of *path connectedness* (Section 4.1.1) of Σ_l and ensures that the foreground segment is connected. Unfortunately even for the special case $\lambda = 0$, minimizing Eq. (5.1) with (C0) is NP-hard because the minimum Steiner tree problem can be reduced to this problem [138]. We will see in the following, how to reformulate the problem such that it becomes feasible to solve.

5.2. The Continuous Case: Connectivity Along Geodesics

To approximate a solution to Eq. (5.1) we propose to reformulate the connectivity constraint along the geodesics from each point $x \in \Sigma_l$ inside the foreground segment to a specific point $s \in \Sigma_l$ inside the foreground segment. In general, we call the connectivity requirement along geodesics *g-connectivity*, and in this case study the special case that these geodesics pass

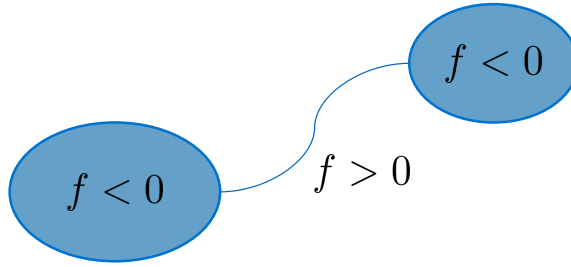


Figure 5.2.: Without the boundary length regularizer ($\lambda = 0$), the foreground segment is defined by the sign of the data term f . Obviously, in this case it holds for the data term f along the path connecting two unconnected parts of the foreground that $f > 0$.

through the point $s \in \Sigma_l$, which we call *rooted g-connectivity*. The point s takes the role of a *root*, from which all the geodesics originate from, and specifying s and the local metric tensor g uniquely defines the topology of the geodesics.

5.2.1. An Image Depending Geodesic Topology

The geodesic shortest path topology is inspired by image segmentation methods based on geodesic distances, that depend on an image depending local metric. Such approaches have been successfully applied to medical image segmentation [10] as well as general image segmentation [6, 30]. In this context, we will show how to define this image depending local metric and how to additionally incorporate a-priori knowledge about the geometry of the object of interest.

First we have to choose an appropriate local metric. We propose to use the non negative cost function $f^+ = \max(0, f(x))$ as metric for the computation of \mathcal{G}_s . This is motivated by the following: Lets consider the special case $\lambda = 0$, then the labelling function $l(x)$ takes on the value 1 for $f(x) < 0$ and 0 for $f(x) > 0$. We leave out the special case $f(x) = 0$ as it does not occur in practice. For all $x_p \in \Omega$ that do not belong to the foreground but need to be added to the foreground to satisfy the connectivity constraint obviously $l(x_p) = 0$ and therefore $f(x_p) \geq 0$. This is also illustrated in Fig. Fig. 5.2. The optimal cost of the connecting path between a fixed s and any x in the region that should be connected on \mathcal{G}_s is then given by

$$\bar{C}_s^x = \min_{C_s^x} \int_0^1 f^+(C(t)) dt, \quad (5.4)$$

which motivates our choice of f^+ as image depending metric.

5.2.2. Connectivity Constraint as Monotonicity Constraint

We formalize the segmentation model with rooted g-connectivity constraint as

$$\min_{l: \Omega \rightarrow \{0,1\}} \int_{\Omega} f(x) l(x) dx + \lambda |\Sigma_l| \quad (\text{P1})$$

$$\text{s.t.} \quad \forall x \in \Sigma_l : \exists \bar{C}_s^x \subset \mathcal{G}_s, \bar{C}_s^x \subset \Sigma_l \quad (\text{C1})$$

where \mathcal{G}_s is the set of all geodesics through s and \bar{C}_s^x is the geodesic curve between s and x .

In contrast to the original image segmentation problem, we will show in the following that this constraint can be introduced as a linear constraint in a convex optimization framework and therefore allows the computation of an optimal solution.

First, we relax the discrete labelling function $l : \Omega \mapsto \{0, 1\}$ by introducing a labelling function with continuous co-domain $u : \Omega \mapsto [0, 1]$ and replace the perimeter of the foreground region with the *total variation* of the continuous labelling function, for details see Section 3.3. Now the connectivity constraint can be expressed as constraint on the directional derivative of u along a geodesic.

$$\min_{u: \Omega \mapsto [0,1]} \int_{\Omega} f(x) u(x) + \lambda |\nabla u| \, dx \quad (\text{P2})$$

$$\text{s.t.} \quad \forall x \in \Omega, \exists \bar{C}_s^x \subset \mathcal{G}_s, \nabla_{\bar{C}_s^x} u(t) \leq 0, t \in [0, 1] \quad (\text{C2})$$

where $\nabla_{\bar{C}_s^x} u(t)$ is the directional derivative of u at t along the geodesic \bar{C}_s^x , in the direction of increasing distance from s , defined as

$$\nabla_{\bar{C}_s^x} u(t) = \lim_{h \rightarrow 0} \frac{u(\bar{C}_s^x(t+h)) - u(\bar{C}_s^x(t))}{h}. \quad (5.5)$$

In fact, we can solve the discrete labelling problem by computing a thresholded solution of the problem with continuous co-domain. A proof for this relation is given in the following section.

5.2.3. The Thresholding Theorem

In the following we show that every thresholded version of the optimal solution of the relaxed optimization problem (P2) provides a minimizer for the original binary labelling problem (P1). We show *optimality* of the thresholded version using the thresholding theorem of [26]. What is left to show is *feasibility* of the thresholded version: that every continuous solution which is feasible regarding the connectivity constraints in (P2) is connected in the discrete case of (P1).

First we reproduce the more concise version of the thresholding theorem as appeared in [79]. Therefore we consider the image segmentation problem without connectivity constraints. First we formulate the functional of the labelling problem with discrete indicator function $l : \Omega \mapsto \{0, 1\}$ as

$$\hat{E}(l) = \int_{\Omega} f(x) l(x) \, dx + \lambda \text{Per}(\Sigma_l). \quad (5.6)$$

We introduce a continuous differentiable function $u : \Omega \mapsto [0, 1]$ of bounded variation and replace the perimeter with the total variation of u . Further details on the relation between the total variation and perimeter of a set are given in the introduction in Section 3.3.

$$E(u) = \int_{\Omega} f(x) u(x) + \lambda |\nabla u| \, dx \quad (5.7)$$

The thresholding theorem now states the following:

Theorem 5.2.1. [79] *Let $u^* : \mathbb{R}^n \mapsto [0, 1]$ be a global minimizer of (5.7). Then all upper level sets, i.e. the thresholded versions*

$$\Sigma_{\{u^* \geq \mu\}} = \{x \in \mathbb{R}^n \mid u^*(x) > \mu\}, \quad \mu \in (0, 1), \quad (5.8)$$

of u^ are minimizers of the original binary labelling problem (5.6).*

Proof. [79] We use the layer cake representation of the function $u^* : \mathbb{R}^n \mapsto [0, 1]$:

$$u^*(x) = \int_0^1 \mathbb{1}_{\{u^* \geq \mu\}}(x) \, d\mu \quad (5.9)$$

and rewrite the first term in the functional (5.7) as

$$\int_{\mathbb{R}^n} f u^* \, dx = \int_{\mathbb{R}^n} f \left(\int_0^1 \mathbb{1}_{\{u^* \geq \mu\}} \, d\mu \right) \, dx = \int_0^1 \int_{\Sigma_{\{u^* \geq \mu\}}} f(x) \, dx \, d\mu \quad (5.10)$$

The functional (5.7) then can be written as

$$E(u^*) = \int_0^1 \left\{ \int_{\Sigma_{\{u^* \geq \mu\}}} f(x) \, dx + |\partial \Sigma_{\{u^* \geq \mu\}}| \right\} \, d\mu \equiv \int_0^1 \hat{E}(\Sigma_{\{u^* \geq \mu\}}) \, d\mu, \quad (5.11)$$

where the total variation norm in (5.7) is written as the integral over the length of all level lines of u by using the coarea formula (see Section 3.4)

$$\int_{\Omega} |\nabla u(x)| \, dx = \int_{-\infty}^{\infty} \mathcal{H}_{n-1}(u^{-1}(t)) \, dt \quad (5.12)$$

We see that the functional (5.11) can be expressed as an integral of the original binary labelling problem \hat{E} evaluated on the upper level sets of u^* .

Assume that theorem 5.2.1 does not hold for some threshold value $\tilde{\mu} \in (0, 1)$, i. e. there exists a minimizer Σ^* of the binary labelling problem with smaller energy

$$\hat{E}(\Sigma^*) < \hat{E}(\Sigma_{\tilde{\mu}, u^*}). \quad (5.13)$$

Then for the indicator function $\mathbb{1}_{\Sigma^*}$ of the set Σ^* we get

$$E(\mathbb{1}_{\Sigma^*}) = \int_0^1 \hat{E}(\Sigma^*) \, d\mu < \int_0^1 \hat{E}(\Sigma_{\{u^* \geq \mu\}}) \, d\mu = E(u^*), \quad (5.14)$$

which contradicts the assumption that u^* was a global minimizer of (5.7). \square

We now show that every thresholded version of a feasible minimizer of u^* (P2) is a feasible minimizer of (P1):

Theorem 5.2.2. *Let $u^* : \mathbb{R}^n \mapsto [0, 1]$ be a global minimizer of (P2) that is feasible with respect to the constraints (C2). Then all upper level sets, i.e. the thresholded versions*

$$\Sigma_{\{u^* \geq \mu\}} = \{x \in \mathbb{R}^n \mid u^*(x) > \mu\}, \quad \mu \in (0, 1), \quad (5.15)$$

of u^ are minimizers of the original binary labelling problem (P1), are connected and thus feasible with respect to the constraint (C1).*

Proof. Optimality of a thresholded version of a minimizer of (P2) for the binary labelling problem (P1) follows from Theorem 5.2.1. What is left to show is feasibility of the thresholded solution with respect to (C1). We show this by contradiction.

Let u^* be feasible with respect to (C2). Let's assume that the constraints (C1) do not hold, i.e. there exists an $x \in \Sigma_{\{u^* \geq \mu\}}$, i.e. $u^*(x) > \mu$ for some μ , and we have a geodesic $C_s^x \subset \mathcal{G}_s$, with $C_s^x(0) = s$ and $C_s^x(1) = x$ for which $C_s^x \not\subset \Sigma_{\{u^* \geq \mu\}}$. In order to let $C_s^x \not\subset \Sigma_{\{u^* \geq \mu\}}$ we have an interval (a, b) with $0 < a < b < 1$, which is not in $\Sigma_{\{u^* \geq \mu\}}$, i.e. $u^*(C_s^x(t)) < \mu$ for all $t \in (a, b)$. Thus we have $u^*(x) = u^*(C_s^x(1)) > u^*(C_s^x(t))$ for $a < t < b < 1$ which violates the monotonicity constraint (C2) along C_s^x . This concludes the proof. \square

We therefore have shown that by thresholding of a feasible minimizer of the continuous optimization problem (C2) we obtain a binary solution of the original binary labelling problem which is optimal and feasible with respect to the original discrete optimization problem (C1).

5.3. Image Segmentation on the Discrete Domain of a Weighted Graph

Before we can solve the optimization problem numerically, we have to find a suitable discretization of the domain. Often, the image segmentation problem with total variation regularizer is discretized on the image grid, and the gradient and the divergence operator are realized using finite differences. Another common approach is to interpret the pixel grid as a graph, with a vertex for every pixel coordinate $x \in \Omega$.

A well known approach for image segmentation on the discrete domain of a graph is the graph-cut framework [19]. In this framework, the optimal subset with minimum boundary length problem is solved by computing a minimum cut through a discrete graph, which is equivalent to computing a maximum flow through the graph, a duality given by the max-flow min-cut theorem [40, 46]. Modern efficient algorithms exist to compute this maximum flow [18], which has led to a widespread application of the min-cut framework. However, a major drawback of this approach is that the boundary length is measured as a discretized quantity: it amounts to the sum of the weights of edges belonging to the cut, which results in a metrication error and leads to discretization artifacts.

When using the total variation regularizer, this metrication error can be avoided: as discussed in the introduction in Section 3.3, the total variation of the labelling function measures the accumulated Euclidean boundary length of the function's level lines. Here, we show instead how to define a regularizer on a discrete graph that is equivalent to the total variation regularizer. This allows, as shown in the following section, to also formulate the connectivity constraint on this discrete graph. First, we define the corresponding operators on a discrete graph and derive a *local variation* regularized segmentation model as theoretically sound equivalent to the continuous *total variation* model. As a consequence, the labelling problem on the weighted graph can be solved efficiently using a recent algorithm for continuous convex optimization [25]. We validate via experiments that, when choosing the ℓ^2 vector norm for the dual variable, the proposed method does not suffer the metrication artifacts of the graph cut framework. Furthermore we show that the metrication errors of the graph-cut framework can be reproduced when taking the ℓ^∞ norm of the dual variable.

5.3.1. Gradient and Divergence Operators on Weighted Graphs

Let $G = (V, E, W)$ be a graph with the set of vertices V with $|V| = n$, a set of edges $E \subset V \times V$ and a positive $n \times n$ weight matrix W that assigns a weight to every edge of the graph. We define the gradient and divergence operators on the graph following [66] and [15, 41]. Let $f : V \mapsto \mathbb{R}$ be a function of $\mathcal{H}(V)$, the Hilbert space of real-valued functions on the vertices of G that is equipped with the inner product $\langle f, g \rangle_{\mathcal{H}(V)} = \sum_{v \in V} f(v)g(v)$. We define the *difference operator* $d : \mathcal{H}(V) \mapsto \mathcal{H}(E)$ of f on an edge $(i, j) \in E$ as

$$(df)(e_{ij}) = \sqrt{w_{ij}}(f(j) - f(i)). \quad (5.16)$$

This difference operator can be interpreted as the *directional derivative* $\partial_i f_j := (df)(e_{ij})$ of a function f at a vertex i along the edge to vertex j .

The *weighted gradient operator* is the vector operator $\nabla_i f = (\partial_i f_j : (i, j) \in E)^T$. The ℓ_2 norm

of this vector is the *local variation* of f at v

$$|\nabla_i f| := \sqrt{\sum_{ij \in E} (\partial_i f_j)^2} \quad (5.17)$$

$$= \sqrt{\sum_{ij \in E} w_{ij} (f_j - f_i)^2} \quad (5.18)$$

Equivalently, let $p : E \mapsto \mathbb{R}$ be a function of $\mathcal{H}(E)$, the Hilbert space of real-valued functions on the edges of G , that is equipped with the inner product $\langle f, g \rangle_{\mathcal{H}(E)} = \sum_{(i,j) \in E} f(i, j) g(i, j)$. The *adjoint* $d^* : \mathcal{H}(E) \mapsto \mathcal{H}(V)$ of the difference operator is given by

$$\langle df, p \rangle_{\mathcal{H}(E)} = \langle f, d^*p \rangle_{\mathcal{H}(V)}. \quad (5.19)$$

Following the definitions of the inner products, the *divergence operator* of p at a node i is

$$\operatorname{div}_i p = -d^*(p)_i \quad (5.20)$$

$$= \sum_{ji \in E} \sqrt{w_{ji}} p_{ji} - \sum_{ij \in E} \sqrt{w_{ij}} p_{ij}. \quad (5.21)$$

With the directional derivative, the gradient, and the divergence operator, we have all required operators to formulate the total variation regularized image segmentation model on a weighted graph.

5.3.2. The Segmentation Model in the Weighted Graph Framework

In this section we derive our image segmentation algorithm in the weighted graph framework. In the following we denote with $u : V \mapsto [0, 1]$ the relaxed labelling function which assigns a value to every vertex, thus u is a function of $\mathcal{H}(V)$. With $f : V \mapsto \mathbb{R}$ we denote the term that depends on the conditional probabilities for foreground and background at every vertex, defined as $f(i) = -\log P_{\mathcal{F}}(x_i) + \log P_{\mathcal{B}}(x_i)$. As a shorthand we write f_i for the value of f at a vertex i , and u_i for the value of u at a vertex i . The dual function $p : E \mapsto \mathbb{R}$ is defined over the edges of the graph and belongs to $\mathcal{H}(E)$. As a shorthand we write p_{ij} for the value of p on the edge ij . Also, we will use the notation x_i to refer to the position $x_i \in \Omega$ associated to the vertex i in the graph.

Given above definitions, we are able to formulate an image segmentation model with *local variation regularization* on a weighted graph

$$\min_{u: V \mapsto [0,1]} \sum_{i \in V} \left\{ f_i u_i + \lambda |\nabla_i u| \right\}. \quad (5.22)$$

By comparing this term with the definition of the local variation Eq. (5.18) we observe that the weight of the regularizer $\lambda \in \mathbb{R}^+$ corresponds to taking the edge weight $w_{ij} = \lambda^2$ for every edge $ij \in E$.

Taking different weights w_{ij} allows to define a local metric that measures the boundary length. How those weights are chosen strongly depends on the application, *e.g.* in image segmentation one can choose weights that depend on the gradient of the image to favour object boundaries at strong image gradients. Furthermore, the presented framework can be used to process 2-manifolds represented as discrete meshes. In [15, 41] the authors propose to apply the framework for surface denoising by smoothing over the vertex coordinates in \mathbb{R}^3 .

5.3.3. A Primal-Dual Method for Vertex Labelling

The definition of the weighted gradient and weighted divergence operators allows to formulate Eq. (5.22) as *saddle-point problem* on a weighted graph

$$\min_u \max_p \quad \langle f, u \rangle_{\mathcal{H}(V)} + \langle u, \operatorname{div} p \rangle_{\mathcal{H}(V)} \quad (5.23)$$

$$\text{s.t.} \quad \forall i \in V, u_i \in [0, 1], |p_i| \leq 1, \quad (5.24)$$

where $|p_i|$ is the ℓ_2 norm of p defined over the edges incident with the vertex i

$$|p_i| = \sqrt{\sum_{ij \in E} p_{ij}^2}. \quad (5.25)$$

The update equations for the segmentation problem on a weighted graph can be derived following [25, 107]. As described in the introduction in section Section 2.5, the update steps in Algorithm 1 in [25] are computed using the **prox**-operator, which for a proper convex lower-semicontinuous function f is defined as

$$\mathbf{prox}_{\lambda F}(v) = \arg \min_x F(x) + \frac{1}{2\lambda} \|x - v\|_2^2. \quad (5.26)$$

For our optimization problem, the **prox**-operator can be decomposed for the values of u_i for each vertex $i \in V$ and for the values of p_{ij} for each edge $ij \in E$. Because of operator duality of ∇_i and div_i , the update equations are given by

$$\begin{aligned} p_{ij}^{k+1} &= \mathbf{prox}_{\sigma F^*} \left(p_{ij}^k + \sigma \partial_i \bar{u}_j^k \right) \\ u_i^{k+1} &= \mathbf{prox}_{\tau G} \left(u_i^k + \tau \operatorname{div}_i p^{k+1} \right) \\ \bar{u}_i^{k+1} &= u_i^{k+1} + \theta \left(u_i^{k+1} - u_i^k \right). \end{aligned} \quad (5.27)$$

With $F^*(p) = \delta_{\leq 1}(|p_i|)$, the indicator function of the unit ball constraints on p , and $G(u_i) = f_i u_i$, the image based data term, we evaluate the **prox**-operators in closed form and get the update equations

$$\begin{aligned} p_{ij}^{k+1} &= \pi_{|p_i| \leq 1} \left(p_{ij}^k + \sigma \partial_i \bar{u}_j^k \right) \\ u_i^{k+1} &= u_i^k + \tau \operatorname{div}_i p^{k+1} - \tau f_i \\ \bar{u}_i^{k+1} &= u_i^{k+1} + \theta \left(u_i^{k+1} - u_i^k \right), \end{aligned} \quad (5.28)$$

with step sizes τ and σ , that are determined using the diagonal precondition method described in [105]. The projection $\pi_{|p_i| \leq 1}(\cdot)$ projects the values of p_{ij}^{k+1} onto the unit ball defined over the edges $ij \in E$ incident to $i \in V$ such that constraint (5.25) is fulfilled.

5.3.4. Comparison of the Primal Dual Algorithm on a Graph and the Graph-Cut Framework

In the previous section, we have shown how to discretize a model for image segmentation with a total variation regularizer on the discrete domain of a graph. As norm over the dual variable of all edges incident to a vertex we define the ℓ^2 norm in Eq. (5.25). The choice of the ℓ^2 norm follows from the definition of the total variation (see Section 3.1). As shown with the co-area formula, this allows to measure the Euclidean length of the level lines of u , which is a major advantage in comparison to the graph-cut framework, that is affected by metrication errors.

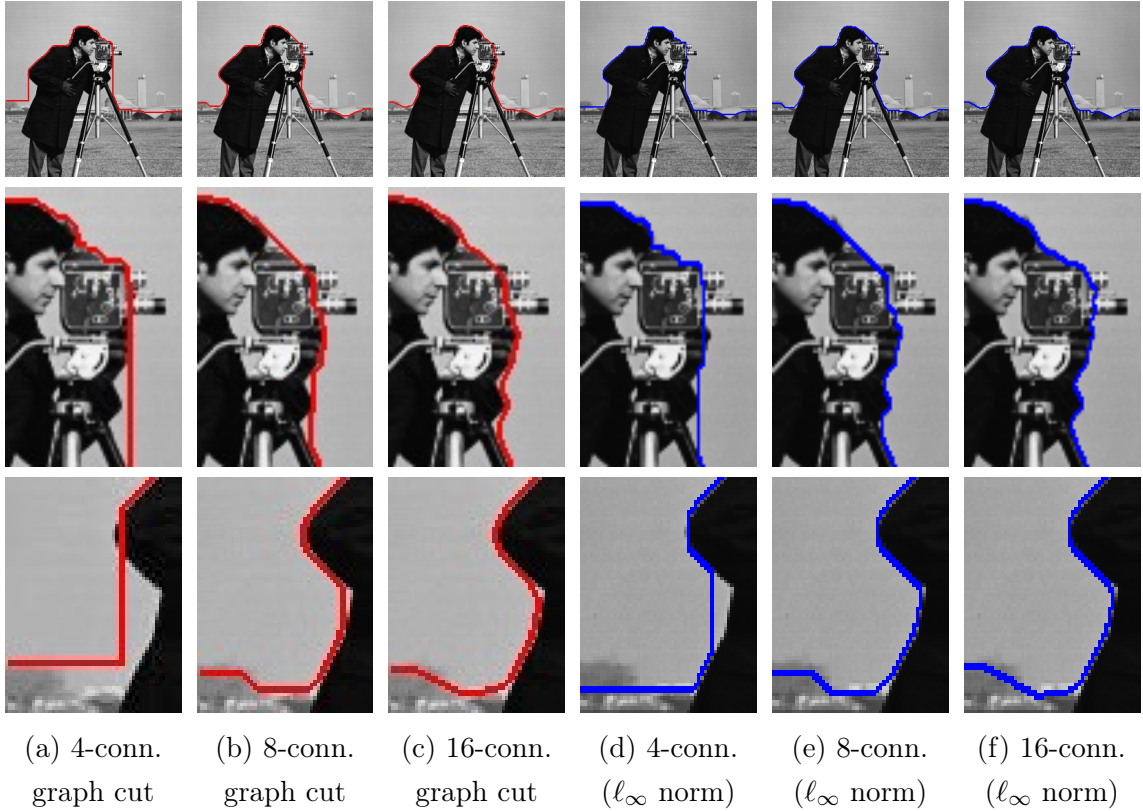


Figure 5.3.: The use of the ℓ_∞ norm for the dual variable p results in metrication artefacts similar to those of the graph-cut algorithm. *Graph cut results reprinted from [78] with kind permission of Maria Klodt.*

Those metrication errors originate from the discrete nature of how the boundary length is defined in the discrete graph-cut framework: the boundary of a set is measured by computing the weight of a cut through the weighted graph, which is the sum of the weights of edges belonging to the cut.

Here we will show that, by choosing a different norm for the dual variable p , we can reproduce the metrication artifacts that occur when using the graph-cut framework. We choose the supremum norm of p_i , defined as

$$|p_i|_\infty = \max_{ij \in E} |p_{ij}|. \quad (5.29)$$

Thus the constraint $|p_i|_\infty \leq 1$ results in a point-wise projection of the values of p_{ij} onto the interval $[-1, +1]$ independently for every edge. As depicted in Fig. 5.3 the constraint on p_i with this ℓ_∞ norm results in metrication artifacts similar to those of the graph-cut framework.

A comparison of segmentation results for the ℓ_2 and the ℓ_∞ norm is provided in Fig. 5.4. Like in the graph-cut framework, the metrication error of the model with the ℓ_∞ norm can be reduced by extending the neighbourhood of the grid. However, for the model with the ℓ_2 norm increasing the degree of the vertices has no effect: the Euclidean length of the boundary is already correctly measured on a 4-grid.

5.4. The Connectivity Constraint on a Discrete Domain

In this section, we describe how the connectivity constraint along geodesics can be formulated on a discrete domain. First we introduce shortest paths as the equivalent to geodesics on the

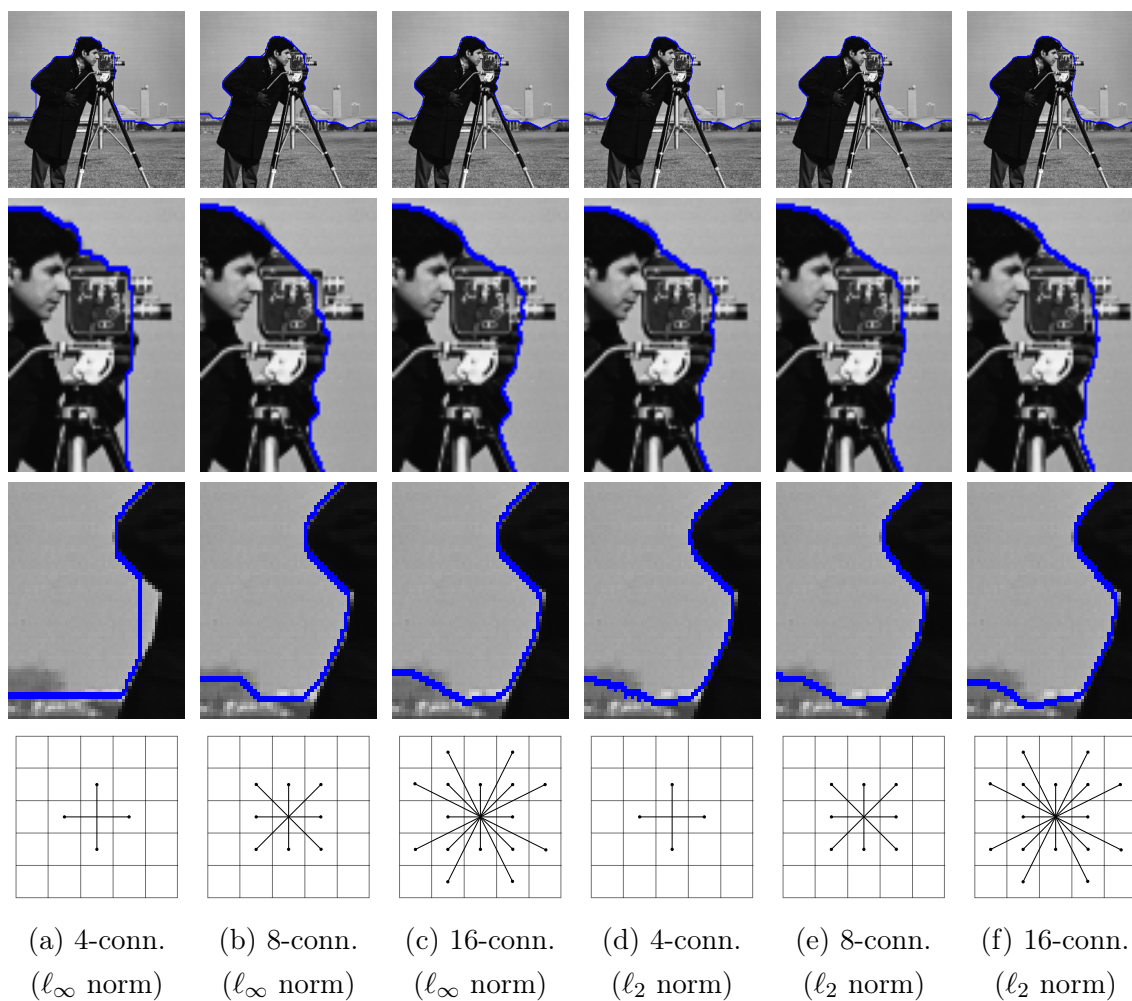


Figure 5.4.: Comparison of different neighbourhood connectivities on the *cameraman* test image using the l_∞ - and l_2 -norm for the dual variable p . The use of the l_∞ -norm results in metrication artefacts similar to those of methods that use the graph cut framework (Compare Fig. 5.3). The l_2 -norm allows to measure the Euclidean norm of the boundary length instead.

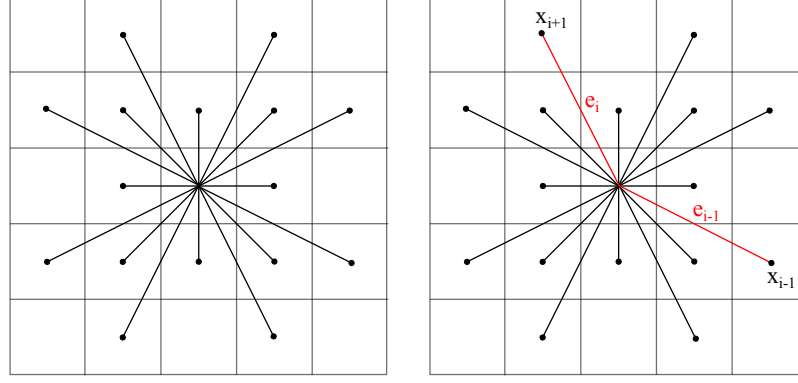


Figure 5.5.: Discretized neighbourhood on the pixel grid that is used for the shortest path search. The bending energy depends on the two edges e_{i-1} and e_i joining at a node x_i .

discrete domain. The shortest paths form a directed graph, the shortest path tree, and we show how the connectivity constraint, a monotonicity constraint along the shortest paths, can be formulated on this tree.

5.4.1. Discrete Geodesics

Shortest Path Tree On the discrete graph, a geodesic from the root s to a point x amounts to the shortest path between s and x . The edges of all those shortest paths originating in s form a tree which is rooted in s . Because we want to compute shortest paths in the graph that are equivalent to the geodesics in the continuous domain described in Section 5.2.1, we compute shortest paths in the metric defined by the positive part of the data term f^+ . Further details are given below.

Bending Energy Prior Additional a-priori information about the geometry of the object that should be segmented can be included in the framework, however this part of the proposed method is optional. For the special case of blood vessel segmentation in medical imaging it is a reasonable assumption, that a blood vessel in a stress free state minimizes its bending energy E_{bend} .

To compute the bending energy in a discretized framework we use the discretized bending energy of Bergou *et al.* [12] that can be applied for curves in 2D as well as spacecurves in 3D. It is expressed using the *curvature binormal*

$$(\kappa \mathbf{b})_i = \frac{2\mathbf{e}^{i-1} \times \mathbf{e}^i}{|\mathbf{e}^{i-1}| |\mathbf{e}^i| + \mathbf{e}^{i-1} \cdot \mathbf{e}^i} \quad (5.30)$$

with $e_i = x_{i+1} - x_i$ and $e_{i-1} = x_i - x_{i-1}$. Because this is an integrated quantity, dividing it by the length of the domain of integration, in this case half the length of the edges joining at x_i , gives the discretized version of the bending energy

$$E_{\text{bend}}(x_i) = \frac{1}{2} \alpha \left(\frac{(\kappa \mathbf{b})_i}{\bar{l}_i/2} \right)^2 \frac{\bar{l}_i}{2} = \frac{\alpha (\kappa \mathbf{b})_i^2}{\bar{l}_i}. \quad (5.31)$$

The curvature binormal $(\kappa \mathbf{b})_i$ depends not only on the position x_i but also on the positions of the neighbouring nodes x_{i+1} and x_{i-1} . The energy $E_{\text{bend}}(x_i)$ is the local part of the bending energy that would add to the total bending energy of a curve, if it was going through x_i given that the positions of its neighbouring nodes stay fixed.

Computing the Shortest Path Tree Finally, the combination of the non negative data term and the bending energy prior leads to the (discretized) geodesic shortest path problem

$$\min_{sPx} \sum_{i=1}^{n(P)} f\epsilon^+(P(i)) + E_{bend}(P(i)), \quad (5.32)$$

where with sPx we denote a path from s to x of length $n(P)$ in the discretized domain and $P(i)$ returns the i -th vertex in P . For numerical stability of the shortest path algorithm, we use the function $f\epsilon^+(i) = \max(\epsilon, f(x_i))$.

Note that this is not a usual geodesic measure, because the bending energy term depends on the angle between the incoming and outgoing edge. Thus, standard first-order techniques [119, 134] can't be used. Instead, such cost functions can be minimized by computing a shortest path on a higher order graph, which contains a node for every edge in the original graph [4]. However, this approach results in a search problem of high complexity. To achieve a feasible runtime also for large datasets, we approximate the minimal path by using a *greedy* optimization scheme, in this case Dijkstra's shortest path algorithm [35] on the pixel grid, using the extended pixel neighbourhood depicted in Fig. 5.5. At every expansion step of Dijkstra's algorithm the value of (5.32) for the candidate nodes x_{i+1} is computed by taking the predecessor x_{i-1} in the current shortest path to x_i . Thus this approximation does not take into account different incoming directions to the node x_i but assumes the incoming directions to be fixed by x_{i-1} . The result of Dijkstra's shortest path algorithm is a geodesic shortest path tree, that spans the whole image, and defines a unique path from the source to every pixel in the image.

In the continuous setting, we formulated the connectivity constraint as monotonicity constraint along geodesics. As discussed in the previous Section 5.4.1, on the discrete domain, a geodesic from the root s to a point x amounts to the shortest path between s and x . Thus, the connectivity constraint corresponds to a monotonicity constraint along shortest paths. All the shortest paths originating from s form a shortest path tree T_s with root vertex s , and the shortest path from s to x in the discretized image domain is a connected path in T_s . Thus, the connectivity constraint along shortest paths can be formulated on the edges of T_s as follows.

On the shortest path tree T_s , the connectivity constraint is equivalent to the constraint that the label u_i of a node i is always greater or equal than the label of a neighbouring node j with a larger distance $d(j)$ to the root node: $d(i) < d(j) \Rightarrow u_i \geq u_j$. Because the graph structure is a shortest path tree, the condition $d(i) < d(j)$ is satisfied for all nodes i and their child nodes j . The constraint $u_i \geq u_j$ then implies $\partial_i u_j \leq 0 \forall ij \in E(T_s)$. This constraint is linear in u , which preserves convexity of Eq. (5.22) and allows for an optimal solution.

The image segmentation problem with connectivity constraints defined over the edges of the tree T_s thus becomes

$$\min_{u:V \rightarrow [0,1]} \sum_{i \in V} \left\{ f_i u_i + \lambda |\nabla_i u| \right\} \quad (5.33)$$

$$\text{s.t.} \quad \forall ij \in E(T_s), \partial_i u_j \leq 0, \quad (5.34)$$

where we define the gradient operator ∇_i on a regular grid and the connectivity constraint over the edges of T_s .

Also for this discretized domain, we can show that every thresholded version of a feasible minimizer of (5.33) is feasible with respect to the constraints (C1) of the original optimization problem (P1).

Theorem 5.4.1. *Let u^* be a feasible minimizer of (5.33). Let $T_s = (E_T, V_T)$ denote the directed graph of the connectivity constraints, i. e. for each inequality constraint $u^*_i \geq u^*_j$*

there exists a directed edge $(i, j) \in E$. Then every upper level set $\Sigma_{\{u^* \geq \mu\}}$ of u^* is a connected subset on G .

Proof. Assume that $\Sigma_{\{u^* \geq \mu\}}$ is not connected on T_s . Then there has to exist a node $j \in V_T$, for which the parent $i \in V_T$ with $(i, j) \in E_T$ is not in $\Sigma_{\{u^* \geq \mu\}}$.

$$\begin{aligned} i \notin \Sigma_{\{u^* \geq \mu\}} &\implies u^*_i < \mu \\ j \in \Sigma_{\{u^* \geq \mu\}} &\implies u^*_j \geq \mu \end{aligned} \tag{5.35}$$

and therefore $u^*_i < u^*_j$.

This contradicts the connectivity constraint $(i, j) \in E_T \iff u^*_i \geq u^*_j$. \square

5.4.2. Legendre-Fenchel Duality

Note that the connectivity constraints (5.34) of the optimization problem (5.33) are linear constraints. Thus Slater's condition holds and we have strong duality (see Section 2.4.1 for further details).

We include the connectivity constraint by adding the indicator function

$$\delta_{\leq 0}(\partial_i u_j) = \begin{cases} 0 & \text{if } \partial_i u_j \leq 0, \\ \infty & \text{else.} \end{cases} \tag{5.36}$$

to the segmentation model and get

$$\min_{u: V \rightarrow [0,1]} \sum_{i \in V} \left\{ f_i u_i + \lambda |\nabla_i u| \right\} + \sum_{ij \in E(T_s)} \delta_{\leq 0}(\partial_i u_j). \tag{5.37}$$

As we have seen in Section 2.4.2, the indicator function of the inequality constraints in Eq. (5.37) can be included in the primal-dual framework by replacing it with the bi-conjugate of the indicator function

$$\delta_{\leq 0}^{**}(\partial_i u_j) = \sup_{\alpha_{ij} \geq 0} \alpha_{ij} \partial_i u_j, \tag{5.38}$$

where α_{ij} are dual variables defined over the edges $ij \in E(T_s)$ of the constraint graph T_s

We optimize also in these dual variables α_{ij} and get as final update equations with connectivity constraints

$$\begin{aligned} p_{ij}^{k+1} &= \pi_{|p_i| \leq 1} \left(p_{ij}^k + \sigma \partial_i \bar{u}_j^k \right) \\ \alpha_{ij}^{k+1} &= \pi_{|\cdot| \geq 0} \left(\alpha_{ij}^k + \nu \partial_i \bar{u}_j^k \right) \\ u_i^{k+1} &= u_i^k + \tau \operatorname{div}_i p^{k+1} + \tau \operatorname{div}_i^{T_s} \alpha^{k+1} - \tau f_i \\ \bar{u}_i^{k+1} &= u_i^{k+1} + \theta \left(u_i^{k+1} - u_i^k \right), \end{aligned} \tag{5.39}$$

where div^{T_s} is evaluated on the edges of the constraint graph.

Also in this case, we determine the step sizes τ , σ , and ν using the diagonal precondition method described in [105].

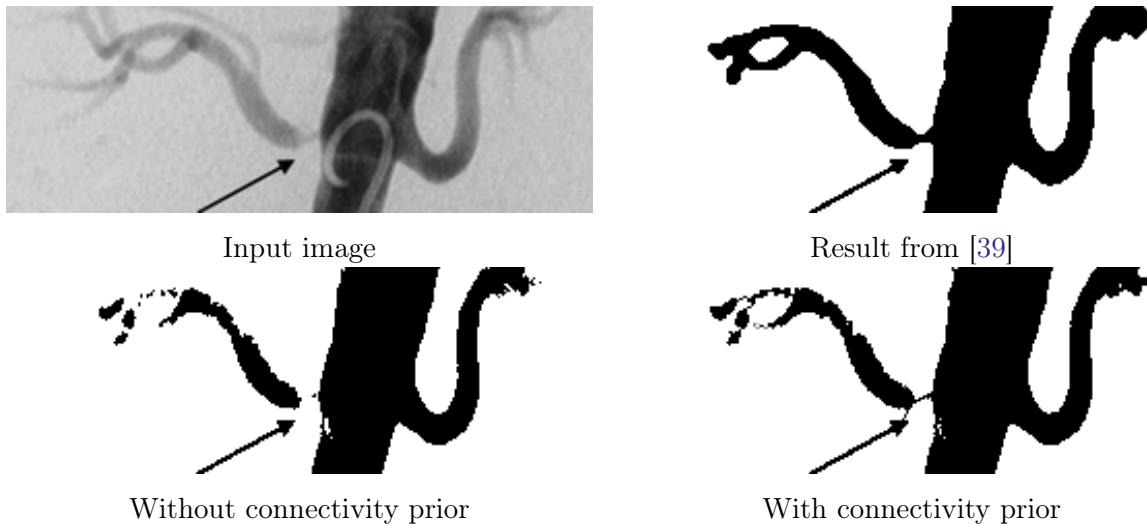


Figure 5.6.: Results on two dimensional medical image data. For comparison we show the results on images from [39]. First column: input image. Second column: Segmentation results from [39]. Third column: Segmentation without connectivity prior. Fourth column: Segmentation result with the proposed method. The connectivity prior enables to connect sparse foreground regions.

5.5. Experimental Results

We applied our method to different types of image data. Figure 6.1 shows the strong capabilities of our segmentation algorithm for the task of blood vessel segmentation in three dimensional CT angiography data ¹. With the tree shape prior, even the small distal tips of the blood vessels are preserved in the final segmentation, while image noise that does not belong to the connected foreground region is successfully suppressed. To segment the whole volume of size $512 \times 512 \times 355$ voxels our algorithm needs 330 seconds on a single threaded 2.27 GHZ Intel Xeon architecture, which is less than 1 second per 512×512 volume slice. Figure 5.6 shows additional results of our tree shape prior on two dimensional medical image data.

Furthermore, the connectivity prior is also a useful extension in an interactive segmentation framework. Figure 5.7 shows an input image with additional user scribbles, that provide hard constraints for foreground and background regions. With these scribbles the user can describe how the shortest path tree is constructed. One foreground region acts as the root node of the shortest path tree. Additional foreground regions can be added via brush strokes that should be connected to the root region.

In all our experiments, we estimate the probability density functions for foreground and background from user scribbles using a Parzen window estimator, with a Gaussian kernel $k_\sigma(I(x) - I_s)$ centred at every image value I_s of the user scribbles.

The method was quantitatively evaluated on the DRIVE database [122] of digital retinal images for vessel extraction. Because the main contribution in this work is the connectivity prior and not the design of a special data term for retinal blood vessel detection, the performance of the tree shape prior was evaluated by using the method of Staal [122] as data term. This is the currently best performing method in the benchmark with an accuracy of 94,42%. By combining this method with the proposed connectivity prior the accuracy can be increased to 94,57%. Therefore this is the highest accuracy reported for this database, almost reaching the accuracy of a human observer (94,73%).

¹ CT dataset taken from the *Vessel Segmentation in the Lung 2012 Grand Challenge* <http://vessel12.grand-challenge.org>.

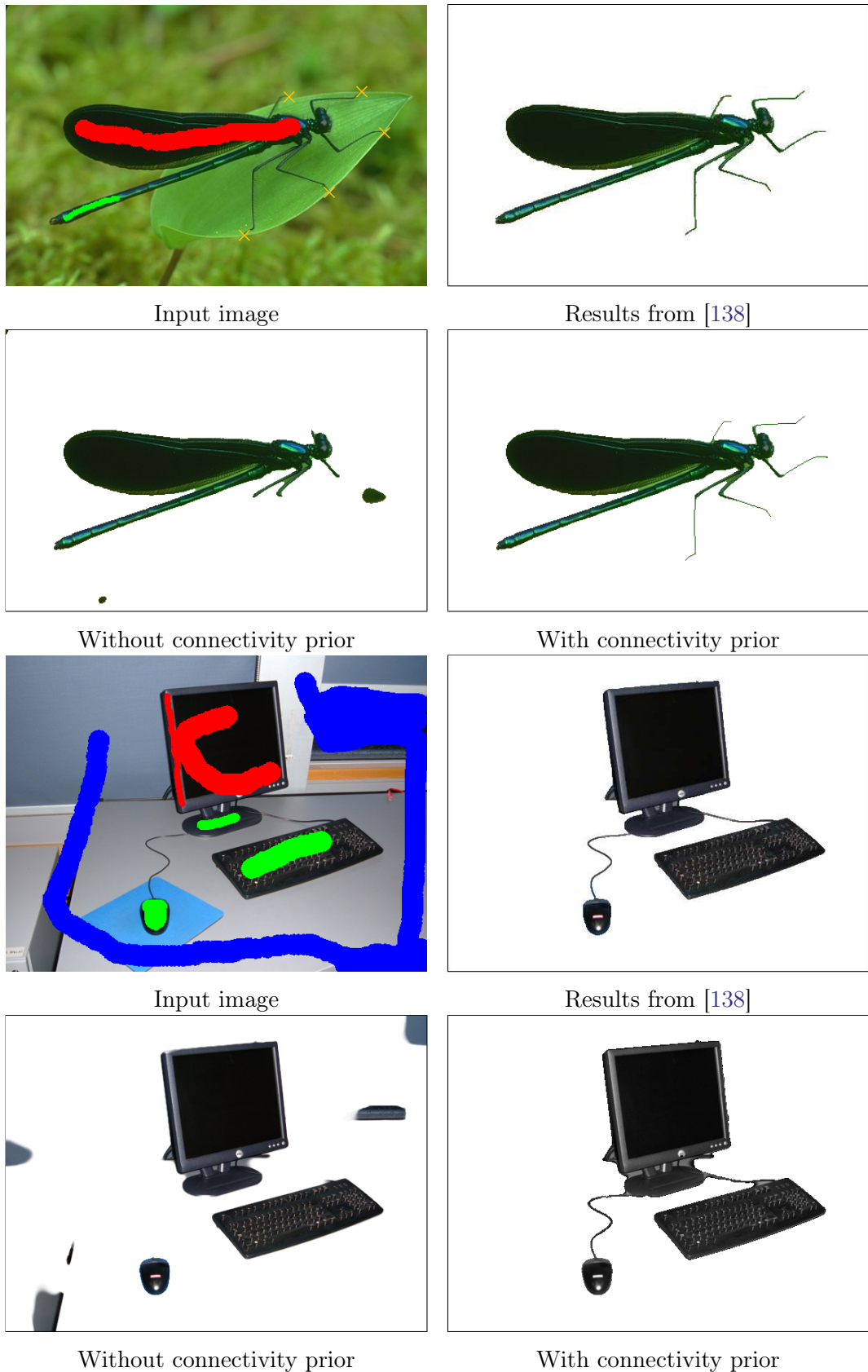
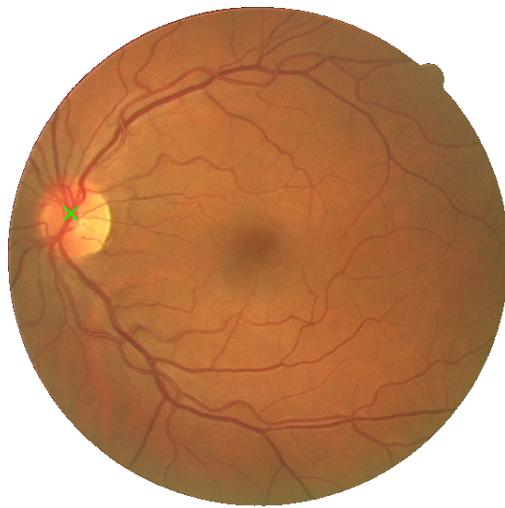


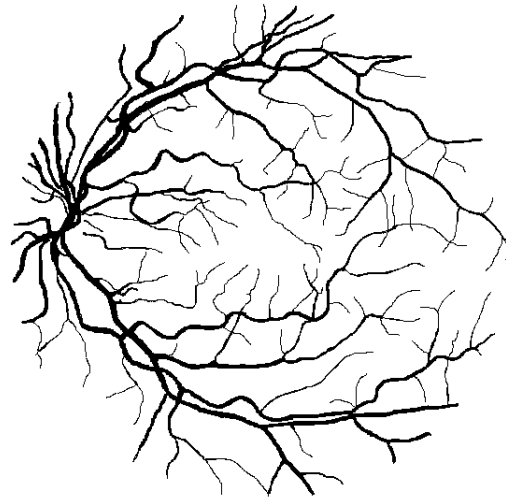
Figure 5.7.: Results for user interactive segmentation. For comparison we show the results on an image from [138]. First column: Input image with user scribbles. The red scribble is the source foreground region of the geodesic shortest path tree, green scribbles are foreground regions that should be connected and red scribbles are background regions. Second column: Segmentation results from [138]. Third column: Segmentation without connectivity prior. Fourth column: Segmentation result with the proposed connectivity prior.

	Accuracy	Sensitivity	Specificity
2 nd Observer	94,73%		
Connectivity Prior			
w Bending Energy	94,57%	84,50%	95,83%
w/o Bending Energy	94,56%	84,65%	95,79%
Staal	94,42%		

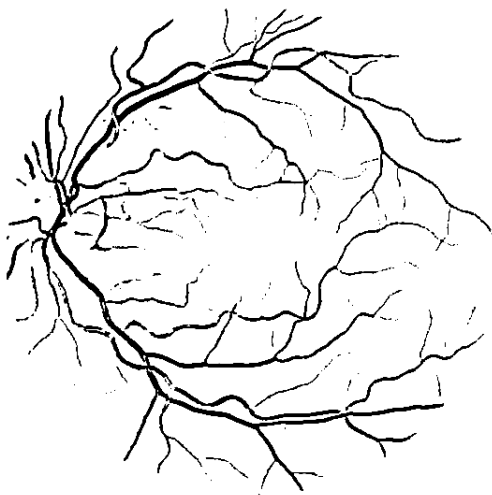
Figure 5.8.: Quantitative evaluation results of the proposed method on the DRIVE database [122]. A combination of the connectivity prior with the method of Staal leads to the most accurate method on this dataset, almost reaching the performance of a second human observer.



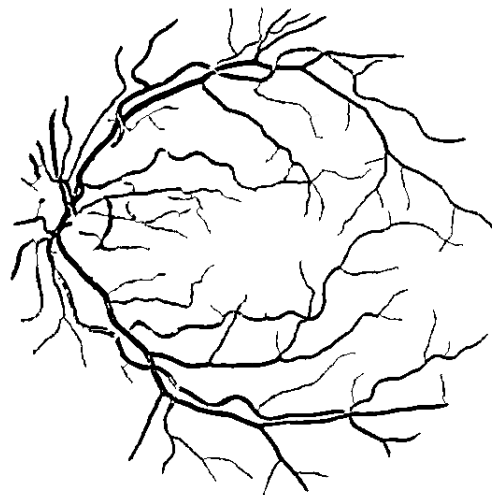
Input image with root s marked in green



Ground truth



Result from [122]



With connectivity prior

Figure 5.9.: Results on an image of the DRIVE benchmark. The connectivity prior increases the segmentation accuracy and allows to connect previously unconnected parts to the vascular network.

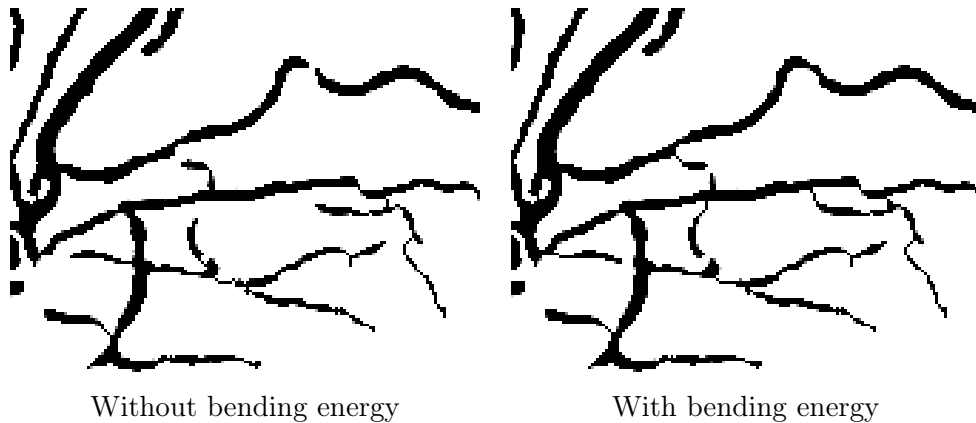


Figure 5.10.: Magnified result from the DRIVE dataset. The bending energy term changes the topology of the connected structure.

In a recent study, Rempfler *et al.* [109]

Including the bending energy term leads to an increased specificity and a slightly increased accuracy, while the sensitivity is decreased. Overall, the number of true positive classified pixels is increased and the number of true negatives is decreased.

5.6. Conclusion

In this work we presented a novel method for image segmentation with connectivity constraints. While solving the image segmentation problem with general connectivity constraints is NP-hard, we propose to formulate the constraint on a geodesic shortest path tree, leading to the novel tree shape prior.

We show that our method can be successfully applied to medical image segmentation problems in angiography and retinal blood vessel extraction, where thin structures otherwise would not be preserved by boundary length regularizers. Experiments on a public dataset show that combining the connectivity prior with existing image segmentation methods clearly improves the performance.

To solve the optimization problem, we generalized an efficient primal dual optimization algorithm for arbitrary graphs. Future work will focus on utilizing the iterative structure of the algorithm for a parallelized implementation on the GPU.

Acknowledgements

The research presented in this chapter was supported by the ERC Starting Grant "ConvexVision" and the Technische Universität München - Institute for Advanced Study, funded by the German Excellence Initiative.

6. A Fast Projection Method for Connectivity Constraints

In this chapter we present how to efficiently project onto the feasible set of the connectivity constraints presented in the previous chapter. The constraints form a convex set and the convex image segmentation problem with a total variation regularizer can be solved to global optimality in a primal-dual framework. Efficiency is achieved by directly computing the update of the primal variable via a projection onto the constraint set, which results in a special quadratic programming problem similar to the problems studied as isotonic regression methods in statistics, which can be solved with $O(n \log n)$ complexity. We show that especially for segmentation problems with long range connections this method is by orders of magnitudes more efficient, both in iteration number and runtime, than solving the dual of the constrained optimization problem. Experiments validate the usefulness of connectivity constraints for segmenting thin structures such as veins and arteries in medical image analysis. The results presented in this chapter have been published in [126].

6.1. Introduction

To allow to preserve thin structures, topological constraints, and especially those that preserve connectivity [130, 138], have been introduced into image segmentation methods.

These constraints have a great advantage in several application areas, including the segmentation of arteries and veins in medical imaging but also in a user interactive setting for general image segmentation. They are very useful when thin structures should be extracted from image data, allowing to extract the whole branching tree of blood vessels in the lung, as shown on the left in Fig. 6.1. For comparison, a total variation regularized segmentation of the dataset without connectivity constraints is shown on the right. In order to preserve the thin structures, only a very small weight of the regularizer can be chosen. Therefore a lot of noise is still present in the final segmentation.

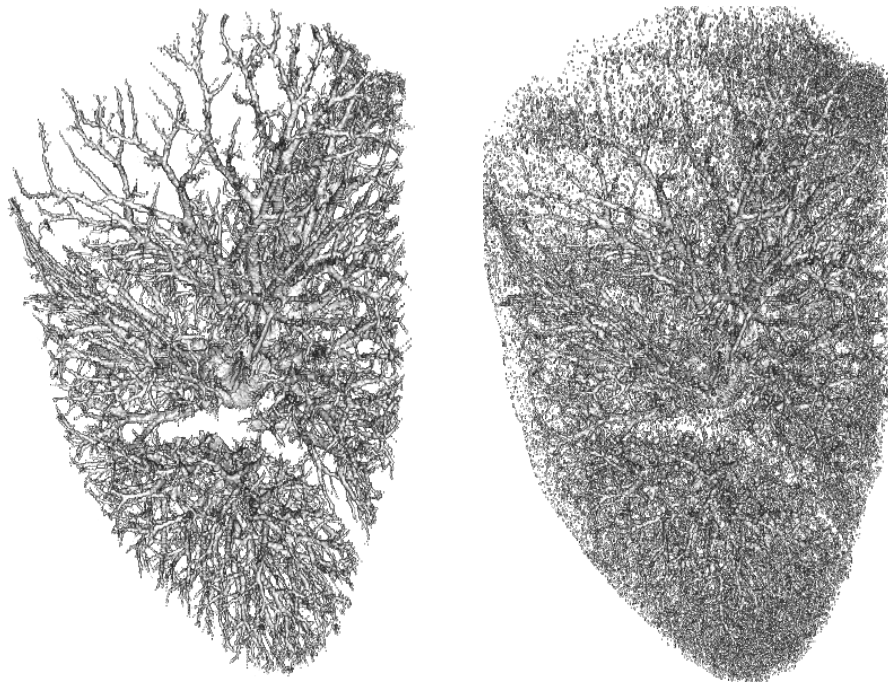
Including these constraints in the segmentation model either leads to a higher algorithmic complexity [28, 138] or slow convergence when solving the dual of the constrained optimization problem [130].

6.1.1. Related Work

For an overview of related work on image segmentation with topological constraints please refer to Section 5.1.1 in the previous chapter.

In the previous chapter and our work [130] we propose a global optimal segmentation method with connectivity constraints in a convex optimization framework. The combination of a total variation regularizer with a connectivity constraint allows to segment thin structures even in very noisy image data. Compared to the work of Gulshan *et al.* [61] our method uses a continuous segmentation framework and therefore the boundary length regularizer is not

¹CT dataset from the *Vessel Segmentation in the Lung 2012 Grand Challenge*



Result with connectivity constraint Without connectivity constraint

Figure 6.1.: Connectivity constraints allow to extract the whole branching tree of blood vessels in the lung, as shown on the left ¹. For comparison, a total variation regularized segmentation without connectivity constraints is shown on the right. In order to preserve the thin structures, only a very small weight of the regularizer can be chosen, therefore a lot of noise is still present in the final segmentation.

biased by discretization artifacts. The constrained optimization problem in [130] is solved by computing a solution of the dual optimization problem. In this work, we propose an efficient projection scheme to directly compute a solution for the update of the primal variable, by a projection of the primal variable onto the feasible set.

6.1.2. Contribution

We propose to solve an image segmentation problem with connectivity constraints via projection onto the constraint set. We show that the constraints form a convex set and derive a projection algorithm from isotonic regression methods in statistics. We show that especially for segmentation problems with long range connections this method is by orders of magnitudes more efficient, both in iteration number and runtime, than solving the dual of the constrained optimization problem.

6.2. Connectivity Constraints in Image Segmentation

First lets review the results from [130] and the previous chapter, where image segmentation with connectivity constraints is formalized as the constrained optimization problem

$$\min_{l: \Omega \rightarrow \{0,1\}} \int_{\Omega} f(x) l(x) dx + \lambda |\Sigma_l| \quad (\text{P1})$$

$$\text{s.t.} \quad \forall x \in \Sigma_l : \exists \bar{C}_s^x \subset \mathcal{G}_s, \bar{C}_s^x \subset \Sigma_l \quad (\text{C1})$$

with the domain Ω , a bounded connected subset of \mathbb{R}^m , $BV(\Omega; [0, 1])$ is the space of functions with bounded variation and $f : \Omega \rightarrow \mathbb{R}$ depends on the image data. The data term f is chosen in such a way that it is negative for image values which are more likely to be foreground and positive in regions which should be regarded as background, e.g. the log ratio $f(x) = \log \frac{P(I(x)|l(x)=0)}{P(I(x)|l(x)=1)}$, with the image I and the discrete label assignment $l : \Omega \mapsto \{0, 1\}$, that describes if an image region belongs to the object of interest $l(x) = 1$ or the image background $l(x) = 0$. With \mathcal{G}_s we denote the set of all geodesics that pass through a given point s , for example defined by user input, defined by a metric that depends on a probabilistic model for foreground and background probabilities, and $\bar{C}_s^x \subset \mathcal{G}_s$ denotes the shortest geodesic from s to a terminal point x .

The solution of the optimization problem should satisfy the connectivity constraint **C1**:

For each $x \in \Omega$ that belongs to the foreground there must exist a connected shortest geodesic path from a given $s \in \Omega$ to x such that all $p \in \Omega$ in the path between x and s belong to the foreground.

This constraint not only ensures the connection of every labeled foreground region to s but also ensures that the whole foreground segment is connected.

6.3. Constrained Convex Optimization

On a discretized domain, the geodesics form a geodesic shortest path tree, a directed acyclic graph $\mathcal{G}_s = \{V, E\}$ with the set of vertices V with $|V| = n$ and the set of directed edges $E \subset V \times V$ with $|E| = m$. We follow [130] and formulate the global connectivity constraint as a monotonicity constraint over each edge of this graph. To satisfy the connectivity constraint we observe that the value of the discretized value function u_i of a node i with distance to the root node d_i should always be greater or equal than the labels of its neighbors with a larger distance $d_j > d_i$ to the root node. This implies that the *directional derivative*

$$\partial_i u_j := (du)(e_{ij}) = (u(j) - u(i))$$

of u at vertex i along the edge to vertex j should always be less or equal to zero.

The image segmentation problem Eq. (P1) thus can be written as the constrained optimization problem

$$\begin{aligned} \min_{u_i \in [0, 1]} \int_{\Omega} f(x) u(x) + \lambda |\nabla u| dx & \quad (6.1) \\ \text{s.t.} & \\ \partial_i u_j \leq 0, \forall (i, j) \in E, & \end{aligned}$$

where the discrete label has been relaxed by introducing the continuous indicator function $u : \Omega \rightarrow [0, 1]$. The total variation regularizer measures the boundary length of the foreground segment.

This image segmentation problem can be optimized using the Primal-Dual framework of [25, 107] which can be applied to convex optimization problems with a saddle-point structure

$$\min_{u \in U} \max_{p \in P} \langle Ku, p \rangle + G(u) - F^*(p), \quad (6.2)$$

where U and P are finite-dimensional vector spaces, $K : U \rightarrow P$ is a continuous linear operator and $G : U \rightarrow [0, +\infty)$ and $F^* : P \rightarrow [0, +\infty)$ are proper, convex, lower semicontinuous

functions. Recall from section Section 2.5 that the update steps in [25] are computed using the **prox**-operator, which is defined as

$$\mathbf{prox}_{\lambda f}(v) = \arg \min_x f(x) + \frac{1}{2\lambda} \|x - v\|_2^2. \quad (6.3)$$

Using this **prox**-operator, the updates in the primal variable u and the dual variable p are computed as

$$\begin{aligned} p^{k+1} &= \mathbf{prox}_{\sigma F^*} \left(p^k + \sigma K \bar{u}^k \right) \\ u^{k+1} &= \mathbf{prox}_{\tau G} \left(u^k - \tau K^* p^{k+1} \right) \\ \bar{u}^{k+1} &= u^{k+1} + \theta \left(u^{k+1} - u^k \right) \end{aligned} \quad (6.4)$$

To formulate the image segmentation problem Eq. (6.1) in the Primal-Dual framework we reformulate the total variation regularizer by introducing a dual variable $p \in R^2$ [107] and after discretization arrive at the saddle point problem

$$\min_{u_i \in [0,1]} \max_{|p| \leq 1} \lambda \langle \nabla u, p \rangle + \langle f, u \rangle + \delta_{\leq 0}(\nabla_i u), \quad (6.5)$$

where $\nabla_i u$ is the stacked vector of the directional derivatives $\partial_i u_j$ and the connectivity constraint is included by adding its indicator function¹. We identify the function $G(u)$ in Eq. (6.2) with $G(u) = \langle f, u \rangle + \delta_{\leq 0}(\nabla_i u)$.

While the constraints over the domains of u and p can be solved by simple projections, the optimization with respect to the connectivity constraint is more involved. In the following, we will investigate two different strategies to incorporate the connectivity constraint.

6.3.1. Optimization via Fenchel Duality

In [130] we propose to optimize the dual of the constrained optimization problem

$$\min_{\substack{u_i \in [0,1] \\ \alpha \geq 0}} \max_{\substack{|p| \leq 1 \\ \alpha \geq 0}} \lambda \langle \nabla u, p \rangle + \langle f, u \rangle + \langle \alpha, \nabla_i u \rangle. \quad (6.6)$$

The connectivity constraint is ensured by introducing an additional dual variable α_{ij} for each edge $(i, j) \in E$. Especially for long range connections the convergence of these multipliers is very slow as we show in our experiments in section 6.4.

6.3.2. Projection onto the Constraint Set

In this section we describe how the connectivity constraint can be included by directly computing the update of the primal variable subject to this constraint. Therefore we propose an efficient projection scheme to solve the constrained quadratic programming problem, which results from the definition of the **prox**-operator.

According to [25] the update in the primal variable u is defined as

$$u^{k+1} = (I + \tau \partial G)^{-1} (u^k + \tau \operatorname{div} p^{k+1}) \quad (6.7)$$

$$= \arg \min_{v \in [0,1]} \left\{ \frac{\|v - (u^k + \tau \operatorname{div} p^{k+1})\|^2}{2\tau} + \langle f, v \rangle + \delta_{\leq 0}(\nabla_i v) \right\}. \quad (6.8)$$

¹Note that while $\nabla_i u$ is defined on the graph \mathcal{G}_s , the gradient ∇u used in the total variation regularizer is computed using standard forward operators on the image grid.

By completing the square and omitting terms independent of v we arrive at

$$u^{k+1} = \arg \min_{v \in [0,1]} \left\{ \|v - (u^k + \tau \operatorname{div} p^{k+1} - \tau f)\|^2 + \delta_{\leq 0}(\nabla_i v) \right\} \quad (6.9)$$

which is of the general form

$$\begin{aligned} & \arg \min_{v_i \in [0,1]} \|v - \tilde{u}\|^2 \\ & \text{s.t.} \\ & v_i \geq v_j, \quad \forall (i, j) \in E, \end{aligned} \quad (6.10)$$

with $\tilde{u} = (u^k + \tau \operatorname{div} p^{k+1} - \tau f)$.

Proposition 6.3.1. *The feasible set C determined by the constraints of the optimization problem Eq. (6.10) is a convex set.*

Proof. Let C_1 be the feasible set determined by the inequality constraints and C_2 the constraint on the range of v . The feasible set of Eq. (6.10) then is $C = C_1 \cap C_2$. First we show that C_1 is convex. If for every $a, b \in C_1$ and $\alpha, \beta > 0$ it holds that $\alpha a + \beta b \in C_1$ then C_1 is a convex cone. Because $a, b \in C_1$ it holds that

$$a_i \geq a_j, \quad b_i \geq b_j, \quad \forall (i, j) \in E, \quad (6.11)$$

and because $\alpha, \beta > 0$ it follows

$$\begin{aligned} \alpha a_i &\geq \alpha a_j, \quad \beta b_i \geq \beta b_j, \quad \forall (i, j) \in E, \\ \alpha a_i + \beta b_i &\geq \alpha a_j + \beta b_j, \quad \forall (i, j) \in E. \end{aligned} \quad (6.12)$$

Hence the set C_1 is a convex cone. In addition to the inequality constraints we also have the constraint on the range of v . We call the feasible set of this constraint $C_2 = [0, 1]$. This set is convex, so $C = C_1 \cap C_2$, the intersection of two convex sets, is convex. \square

Thus the optimization problem Eq. (6.10) is strictly convex subject to convex constraints. Its solution is an Euclidean projection of \tilde{u} onto the set C and can be solved to global optimality. Furthermore the inequality constraints describe a partial order on the values of v . A quadratic programming problem with this structure is known in statistics as isotonic regression [8].

6.3.3. Isotonic Regression on a Tree

In Pardalos *et al.* [102] the authors investigate a class of algorithms for isotonic regression where the constraints define a partial order which can be represented by a directed graph. In particular the authors propose an $O(n \log n)$ algorithm for the case when the directed graph is a directed tree with n vertices. For convenience we present the algorithm IRT-BIN here as Algorithm 2.

We call the isotonic regression problem subject to partial order constraints *IRT*. This problem does not include the range constraints of Eq. (6.10). In the following, we will show that a projection of the optimal solution of *IRT* on the range constraint yields the optimal solution of Eq. (6.10).

First we follow the presentation of Pardalos *et al.* [102] and describe the algorithm for isotonic regression with partial order constraints, using the concept of *upper sets*, *lower sets* and *level sets*:

Definition Let X be a nonempty finite set. Let \preceq be a partial order on X . Let Y be a nonempty subset of X . We define the *average* of Y as $Av(Y) = \frac{1}{|Y|} \sum_{i \in Y} \tilde{u}_i$. We call a subset $L \subset X$ a *lower set* of X with respect to \preceq if $i \in X, j \in L$ and $i \preceq j$ implies $i \in L$. Consequently a subset $U \subset X$ is an *upper set* if $i \in U, j \in X$ and $i \preceq j$ implies $j \in U$. We call a subset $S \subset X$ a *level set* if there are an upper set U and a lower set L such that $S = L \cup U$. A *block* B of X is a nonempty level set such that for each upper set $U \subset X$ for which $U \cap B \neq \emptyset$ it holds that $Av(B) \geq Av(U \cap B)$.

Furthermore the authors of [102] introduce the concept of a *block class*:

Definition A collection Δ of blocks of X is called a *block class* of X if

1. the blocks in Δ are pairwise disjoint and their union is the set X .
2. the collection Δ can be ordered by a partial-order \preceq such that $A \preceq B$ for $A, B \in \Delta$ if there exist $i \in A$ and $j \in B$ such that $i \preceq j$.

Note that the collection of all singleton subsets $\{x\}$ with $x \in X$ is a block class.

The authors prove that the optimal solution of *IRT* on a block B is $v_i = Av(B)$ for every $i \in B$. Furthermore they show that if a block class Δ has no adjacent violators, then the optimal solution of the isotonic regression is given by $v_i^* = Av(B(i))$, where $B(i)$ is the block which contains i , for each element i of X .

Algorithm 2 IRT-BIN from Pardalos *et al.* [102]

- 1: Let Δ be the singleton block class and let \mathcal{T} be a copy of the underlying rooted tree.
 - 2: Mark each leaf node of \mathcal{T} as solved and all other nodes as unsolved.
 - 3: **for** each node x_i of \mathcal{T} **do**
 - 4: Create a block $B(x_i) = \{x_i\}$ and a binomial heap H_i .
 - 5: **end for**
 - 6: **if** all nodes of \mathcal{T} are marked as solved **then**
 - 7: output the blocks corresponding to the nodes in \mathcal{T} as the final block class and **stop**;
 - 8: **end if**
 - 9: Let x_i be an unsolved node of \mathcal{T} such that all the children nodes of x_i are solved.
 - 10: Let $B(x_i)$ (resp. H_i) be the block (resp. binomial heap) corresponding to node x_i .
 - 11: **while** $Av(B(x_i)) < Maximum(H_i)$ **do**
 - 12: $ExtractMax(H_i)$ and let $B(x_k)$ be the corresponding block
 - 13: Shrink the edge connecting x_i to x_k \triangleright the new vertex is still called v_i
 - 14: Create a new block $B(x_i) \leftarrow B(x_i) \cup B(x_k)$ \triangleright the new block is still called $B(x_i)$
 - 15: Calculate the $Av(B(x_i))$ for the new block $B(x_i)$
 - 16: $H_i \leftarrow Union(H_i, H_k)$ \triangleright this is the binomial heap for the new block $B(x_i)$
 - 17: **end while**
 - 18: Mark the node x_i of \mathcal{T} as solved.
 - 19: Let x_p be the parent node of x_i in \mathcal{T} . Let H_p be the binomial heap corresponding to $B(x_p)$ and let a_i be the node in H_p which corresponds to $B(x_i)$. $ChangeKey(a_i, Av(B(x_i)), H_p)$.
 - 20: **go to** 6.
-

We will show with the proof of the following proposition that given a solution v^* of *IRT* the optimal solution to Eq. (6.10) is achieved by projecting v^* on C_2 . Thus, we can directly project onto the constraints of the optimization problem Eq. (6.10) by first projecting onto the isotonicity constraint and then onto the $[0, 1]$ -box constraint.

Obviously, projecting first onto the $[0, 1]$ -box constraint and then onto the isotonicity constraint will not lead to a valid projection. When the averaging step is performed after the

$[0, 1]$ clipping, in case that the isotonicity constraint is violated and some values are smaller 1, only block average values well below 1 can be achieved, even when the average of the block before projection was larger than 1.

Proposition 6.3.2. Direct Projection onto the Constraint Set

Let B be a block of X . Let $v_i^* = Av(B)$ for every $i \in B$ be the solution of IRT. Let $\pi_{[0,1]} : \mathbb{R} \rightarrow [0, 1]$ be a projection that projects negative values to 0 and values larger 1 to 1. Then $\{\pi_{[0,1]}(v_i^*) : i \in B\}$ is the optimal solution to the optimization problem (6.10) on B .

Proof. Let us assume that B has m elements x_1, x_2, \dots, x_m . We look at the three cases $Av(B) > 1$, $Av(B) \in [0, 1]$ and $Av(B) < 0$. Obviously these three cases are exhaustive. If $Av(B) \in [0, 1]$ then the solution v^* of IRT also fulfills the range constraint and the solution of Eq. (6.10) for the set B is identical to the solution of IRT on B .

If $Av(B) > 1$ we follow a similar proof as in [102] and show that the point

$$\begin{aligned} \{\pi_{[0,1]}(v_i^*) : i \in B\} &= (\pi_{[0,1]}(Av(B)), \pi_{[0,1]}(Av(B)), \dots, \pi_{[0,1]}(Av(B))) \in \mathbb{R}^m \\ &= (1, 1, \dots, 1) \in \mathbb{R}^m \end{aligned} \quad (6.13)$$

is the optimal solution to Eq. (6.10) by showing that the inner product of the gradient of Eq. (6.10) with any feasible direction $d \in \mathbb{R}^m$ at that point is a non-negative number.

Let $d = (d_1, d_2, \dots, d_m)$ be a feasible direction of the isotonic regression problem on B . Then, in order to preserve isotonicity, feasibility of the direction d implies $d_i \leq d_j$ when $x_i \preceq x_j$.

Therefore there exists a permutation $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(m))$ such that

$$d_{\sigma(1)} \geq d_{\sigma(2)} \geq \dots \geq d_{\sigma(m)} \quad (6.14)$$

and

$$x_{\sigma(i)} \preceq x_{\sigma(j)} \implies i \leq j. \quad (6.15)$$

To prove that for $Av(B) > 1$ the point in (6.25) is the optimal solution of the optimization problem (6.10) on the set B it is sufficient show that

$$\sum_{i \in B} (1 - \tilde{u}_{\sigma(i)}) \times d_{\sigma(i)} \geq 0. \quad (6.16)$$

From Eq. (6.14) and from the definition of a block it follows that

$$\frac{1}{m - k + 1} \sum_{i=k}^m u_{\sigma(i)} \geq Av(B) > 1 \text{ for all } 1 < k \leq m. \quad (6.17)$$

This implies that

$$\sum_{i=k}^m (1 - u_{\sigma(i)}) \leq 0 \text{ for all } 1 < k \leq m. \quad (6.18)$$

Equations (6.18) and (6.14) imply that for all $1 < k \leq m$ that the following inequality holds

$$\sum_{i=k}^m (1 - u_{\sigma(i)}) \times d_{\sigma(k-1)} \geq \sum_{i=k}^m (1 - u_{\sigma(i)}) \times d_{\sigma(k)}. \quad (6.19)$$

Because $Av(B) > 1$ the feasibility of d implies that $d_{\sigma(i)} \leq 0$ for all $i \in \{1, \dots, m\}$. Combining everything together we get

$$\begin{aligned}
 & \sum_{i=1}^m (1 - u_{\sigma(i)}) \times d_{\sigma(1)} & (6.20) \\
 = & \sum_{i=1}^1 (1 - u_{\sigma(i)}) \times d_{\sigma(i)} + \sum_{i=2}^m (1 - u_{\sigma(1)}) \times d_{\sigma(1)} \\
 \leq & \sum_{i=1}^1 (1 - u_{\sigma(i)}) \times d_{\sigma(i)} + \sum_{i=2}^m (1 - u_{\sigma(2)}) \times d_{\sigma(2)} \\
 = & \sum_{i=1}^2 (1 - u_{\sigma(i)}) \times d_{\sigma(i)} + \sum_{i=3}^m (1 - u_{\sigma(2)}) \times d_{\sigma(2)} \\
 \leq & \sum_{i=1}^2 (1 - u_{\sigma(i)}) \times d_{\sigma(i)} + \sum_{i=3}^m (1 - u_{\sigma(3)}) \times d_{\sigma(3)} \\
 & \dots \\
 \leq & \sum_{i=1}^m (1 - u_{\sigma(i)}) \times d_{\sigma(i)} & (6.21)
 \end{aligned}$$

From $Av(B) > 1$ it follows that

$$\sum_{i=1}^m (1 - u_{\sigma(i)}) < 0. \quad (6.22)$$

Together with $d_{\sigma(i)} \leq 0$ for all $i \in \{1, \dots, m\}$ it follows for Eq. (6.20)

$$\sum_{i=1}^m (1 - u_{\sigma(i)}) \times d_{\sigma(1)} \geq 0. \quad (6.23)$$

Therefore from Eq. (6.20) to Eq. (6.21) we have proved that if $Av(B) > 1$

$$\sum_{i=1}^m (1 - u_{\sigma(i)}) \times d_{\sigma(i)} \geq 0. \quad (6.24)$$

If $Av(B) < 0$ we have to show that the inner product of the gradient of Eq. (6.10) with any feasible direction $d = (d_1, d_2, \dots, d_m) \in \mathbb{R}^m$ at the point

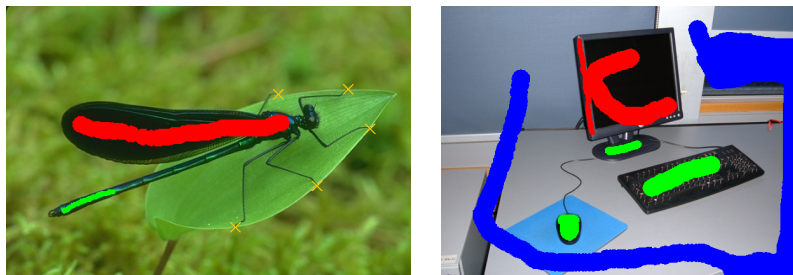
$$\{\pi_{[0,1]}(v_i^*) : i \in B\} = (0, 0, \dots, 0) \in \mathbb{R}^m$$

is a positive number. This proof is equivalent to the proof for $Av(B) > 1$. \square

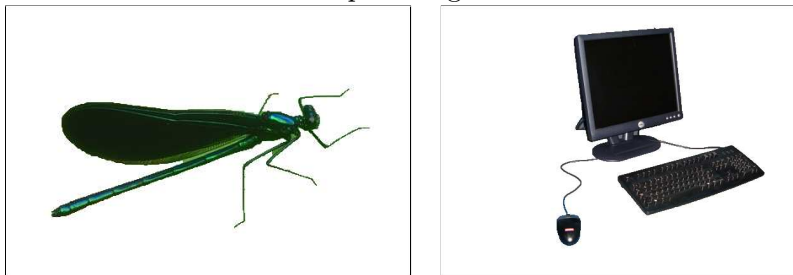
6.4. Experimental Results

For comparison we performed experiments for interactive segmentation on images from [138] that also have been used in other publications, e.g. [39, 130]. As depicted in Fig. 6.2, the segmentations acquired with the projection method are not different from the results of the algorithm based on Fenchel duality [130].

We provide convergence results of the two different methods on a set of synthetic test images. The set contains images of two circles that are connected by a 2 pixel wide faint path of a



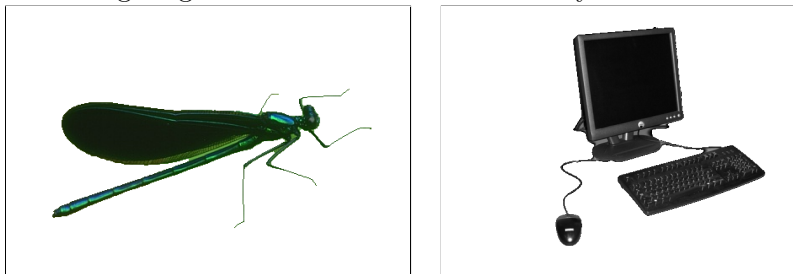
Input images



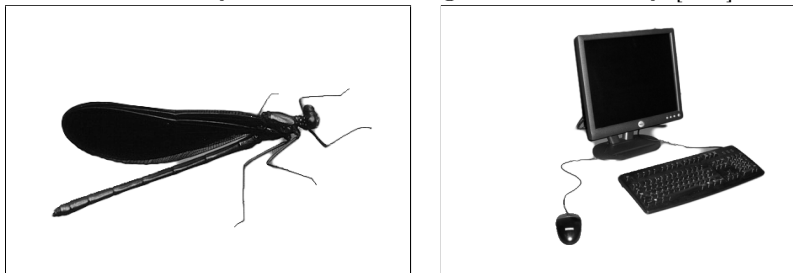
Results from [138]



Image segmentation without connectivity constraints



Connectivity constraints using Fenchel Duality [130]



Connectivity constraints using the projection scheme [126]

Figure 6.2.: Comparison of the projection method and Fenchel duality Both methods produce the same results for interactive segmentation. First row: Input image with user scribbles. The red scribbles are the source of the geodesic shortest path tree, green scribbles are foreground regions that should be connected and blue scribbles are background regions. Second column: Results from [138]. Third column: Segmentation without connectivity constraints. Fourth column: Segmentation with connectivity constraints by solving the dual problem [130]. Fourth column: Segmentation with connectivity constraints using the proposed projection scheme.

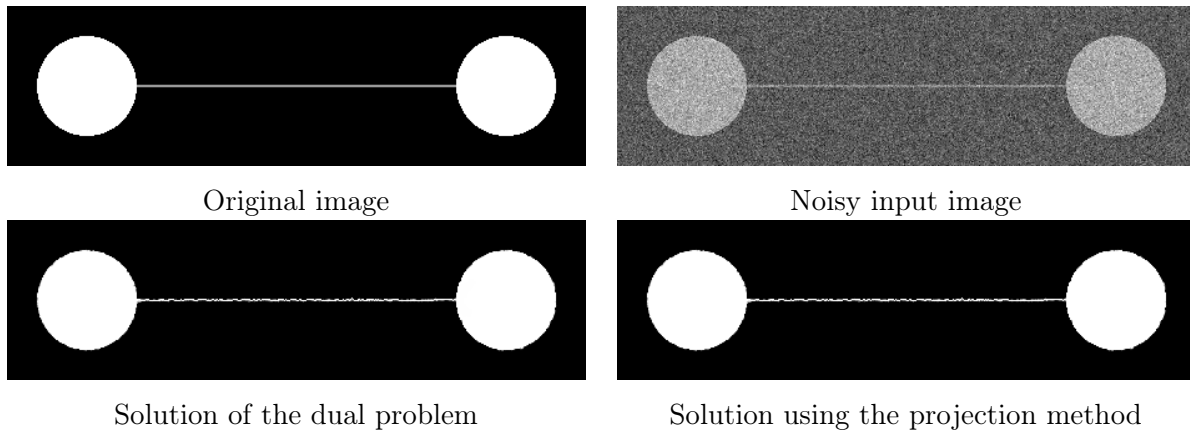


Figure 6.3.: Synthetic test image. Upper row: The input image with added Gaussian noise. Lower row: Identical results of the two different methods to include the connectivity constraint.

Table 6.1.: Comparison of runtime and number of iterations until convergence. Especially when the images contain long range connections, the projection method is by magnitudes more efficient than solving the dual problem.

Image	Fenchel Duality		Projection Method	
	Iterations	Runtime	Iterations	Runtime
Test Circle 64	5396	10.12 s	19	0.29 s
Test Circle 128	18318	41.11 s	20	0.52 s
Test Circle 256	81987	251.17 s	20	1.06 s
Test Circle 512	344030	1639.15 s	20	2.89 s
Fly	1226	9.13 s	54	3.66 s
Desk	3440	42.00 s	109	13.40 s

length of 64, 128, 256 and 512 pixels. As an example, the image for the path length of 256 pixels is shown in Fig. 6.3.

Plots of the convergence of the two methods with respect to runtime are shown in Fig. 6.4. The projection method clearly outperforms the method based on Fenchel duality. The longer the connection, the higher the runtime difference of both methods. Convergence of the dual method takes from 10.12 seconds for the 64 pixel connection, over 41.11 seconds for 128, 251.17 seconds for 256 to 1639.15 seconds for the 512 pixel connection, whereas the projection method converges within less than 3 seconds for all different images. Although solving the isotonic regression problem results in a higher complexity of each iteration, by magnitudes fewer iterations are required for the projection method to converge. The needed runtime and number of iterations until convergence for both methods are also shown in Table 6.1. To measure the speed of convergence we first compute a segmentation result that is reached after a large number of iterations (10000). Then we restart the algorithm and stop when the absolute difference between the current result and the converged result is below 0.1 % of the number of pixels of the image. All Experiments were performed on a a single threaded 2.27 GHZ Intel Xeon architecture.

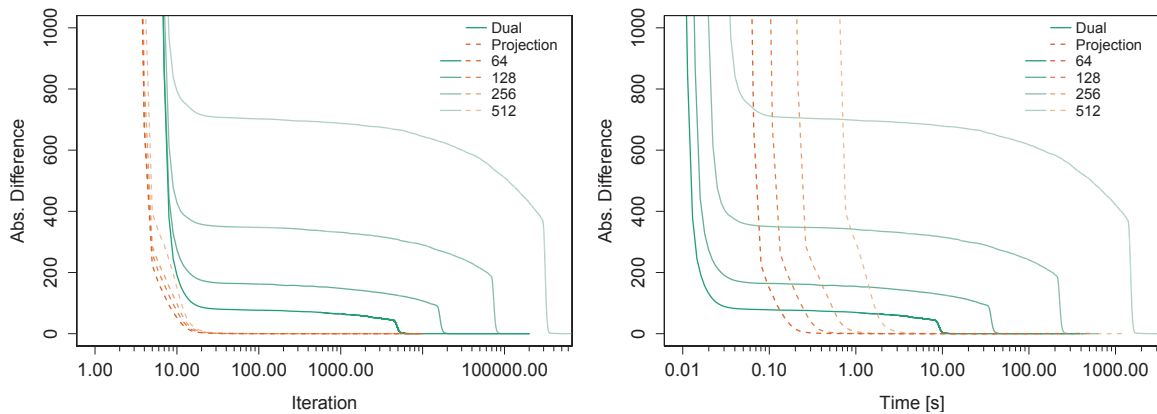


Figure 6.4.: Convergence of the two different methods to include the connectivity constraint on a set of test images as shown in Fig. 6.3. The set contains images with two circles that are connected by a 2 pixel width path of a length of 64, 128, 256 and 512 pixels. Note that the plots have a logarithmic scale at the x axes. When using the projection method (dashed line), by order of magnitudes fewer iterations are needed than for solving the dual problem (solid). This results in a by order of magnitudes better runtime performance.

6.5. Conclusion

We presented a very efficient projection scheme to include connectivity constraints in a convex image segmentation framework. The method outperforms commonly used approaches that are based on Fenchel duality by orders of magnitudes. Instead of using the common approach to solve the dual problem of the constrained optimization problem we directly project onto the constraint set thus significantly fewer iterations are needed until a sufficient convergence is reached. This enables to use connectivity constraints for large segmentation problems as they arise for example in medical image segmentation of three dimensional CT angiography.

Acknowledgements We thank Michael McCoy, Michael Möller and Konstantin Pieper for fruitful discussions. This research was supported by the ERC Starting Grant "ConvexVision" and the Technische Universität München - Institute for Advanced Study, funded by the German Excellence Initiative.

7. Active Online Learning for Interactive Segmentation Using Sparse Gaussian Processes

The image segmentation methods presented so far, focused on the regularizer and appropriate constraints to extract a topologically connected object from the image. Another important part of an image segmentation model is the *data term*, which can be defined by a probabilistic model, that for an image region defines the probability for this region to be part of an object, in two class image segmentation also called foreground, or instead to be part of the background.

But how exactly is this probabilistic model specified? Possible choices include non-parametric models, for example a histogram or a Parzen window estimator, and parametric models, for example a mixture of Gaussian. These models are fitted to labelled data, the so called training set. In its most simple form, this training set is used once specified and then used to fit the probabilistic model of the data term.

As an alternative and more flexible approach, we present an active learning framework to define this probabilistic model. This active learning framework allows to improve the data term in an user interactive approach. Our system uses a sparse Gaussian Process classifier (GPC) trained on manually labelled image pixels (user scribbles) that is refined in every active learning round. In every learning round, our method presents a set of image regions to be labelled by the user. These regions are selected based on the *classification uncertainty* of the classifier, such that regions with high uncertainty are presented to the user. For small images, this seems unnecessary, when visual inspection of the segmentation result allows to quickly identify incorrectly classified regions. However, our method is well suited for large datasets. These large datasets occur for example in satellite and aerial imagery, high resolution microscopy in biology and also large image datasets [113].

The results presented in this chapter are joint work with Rudolph Triebel and Mohammed Souiai and have been published in [133].

7.1. Introduction

Image segmentation is one of the most important problems in computer vision with a large range of applications, including medical imaging and robotics. However, image segmentation is an ill-posed problem in general, because the definition of a correct segmentation strongly depends on the application. In this chapter we therefore focus on *interactive* image segmentation, where the user provides information about the image to be segmented, by manually selecting regions and assigning them a specific class label. These selected regions can for example be selected with strokes drawn in the image, these strokes are also called *user scribbles*, and are used as ground truth information to infer a good segmentation of the image. The scribbles can be used in two ways, first they can define constraints on the final segmentation, because the image labelling result should be consistent with the labels assigned manually by the user. Second, the labelled regions in the image can be used to train a probabilistic classifier to assign the correct class label to an image region. There exist many approaches for

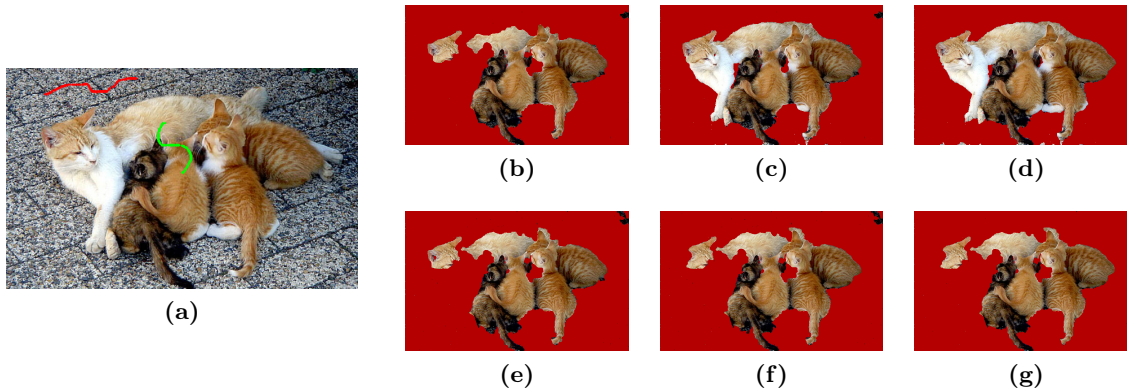


Figure 7.1.: Comparison between a Parzen window estimator [95] and our sparse GP classifier for foreground classification. (a) Both approaches use the initial user input to fit a probabilistic model for foreground and background. As no white region has been marked as foreground, both approaches fail to classify the white neck of the cat correctly (b, e). However, in the next active learning round, the Gaussian process classifier queries exactly this part from the user based on its accurate estimation of the predictive uncertainty (c). In contrast, the Parzen window estimator does not query this part, because its uncertainty is low despite its incorrect classification (f). After 6 rounds the GP achieves a very good segmentation (d), while the Parzen window estimator still gives a lower-quality segmentation (g).

interactive image segmentation, one of the most important and central problems in computer vision, with impressive results. However, current methods reach a high classification rate only by requiring comparably many user scribbles, and the amount of user input needed usually grows rapidly when the segmentation quality should approach 100%.

Therefore, we present a method that asks for user input more intelligently, by actively querying image regions to be labelled where the classification was made with high uncertainty. This way, the amount of user input needed to obtain a high quality segmentation is significantly reduced. While our approach seems unnecessary for a single image, where the user immediately can recognize image regions that were not correctly classified, our approach is well suited for large datasets, where a manual inspection of the classification result is not feasible. To obtain an accurate classification uncertainty estimate, we use a Gaussian Process classifier (GPC) to learn a background and foreground model. For increased runtime performance, we use an efficient sparse version of the GPC.

7.1.1. Related Work

Since the work of Boykov *et al.* [19], many approaches, e.g. [92, 111], have been proposed to compute a segmentation based on the graph cut framework. Another line of research [95, 135] models the image segmentation problem in a continuous domain and is based on the convex relaxation technique of Chan *et al.* [26]. Both discrete and continuous approaches impose spatial smoothness as a prior on the image labelling. Our work is related to [95] where the data term describing the pixel class probabilities includes spatial information while estimating the colour distribution using a Parzen window estimator. However, we use an Informative Vector Machine (IVM) [86], a sparse version of the Gaussian Process Classifier, and employ active learning, which improves the segmentation result quickly after only a few training rounds (see Fig. 7.1). In contrast to the sparse GP algorithm of Csató and Opper [31], the IVM has advantages in the context of active learning, mainly due to the information-theoretic criterion used to select the subset of the training points.

In the field of active learning, Kapoor *et al.* [74] address object categorization using a Gaussian

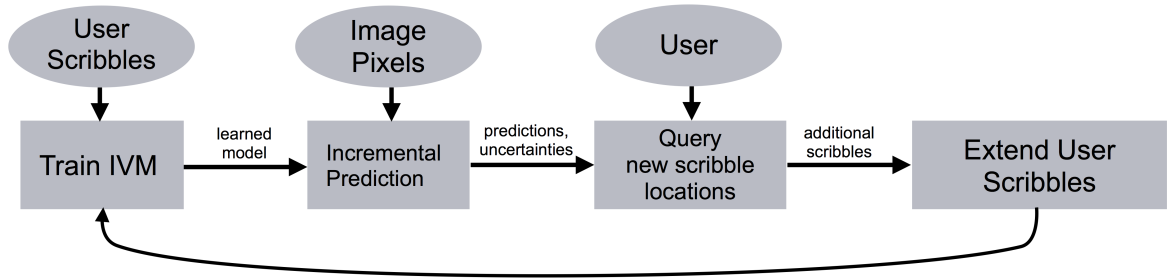


Figure 7.2.: Flowchart of our active learning framework. Starting from an initial set of user scribbles, a sparse GP classifier is trained and the remaining image pixels are classified. The obtained class predictions are analysed with respect to their uncertainty estimates. Then, new user scribbles are queried at locations that are randomly sampled, with probabilities proportional to the classification uncertainties. The newly added user scribbles are added to the training data, and the next training round begins. *Figure used with kind permission of Rudolph Triebel.*

process classifier (GPC) and improve the classification result by querying labels for data points which possess large uncertainty, which is estimated by using the posterior mean and variance. Triebel *et al.* [132] use an IVM in an active learning framework to detect traffic lights in urban traffic images. Vezhnevets *et al.* [137], as well as Wang *et al.* [139] also use active learning for interactive image segmentation, but either with a conditional random field (CRF) with a naive Bayes classifier [137] or a Gaussian Mixture Model (GMM) [139] as the underlying classifier. We favour to use a Gaussian process classifier, because it is non-parametric, i.e. it does not assume a functional model for the data, and it was shown to provide very accurate uncertainty estimates [104], which is crucial in active learning.

We use a sparse version of a Gaussian process classifier, the Informative Vector Machine (IVM) [86]. A sparsification of the GP classifier is achieved by selecting only a sub-set of the training data that is most informative with respect to the expected information gain. This smaller sub-set is used to approximate the posterior. Csató and Opper [31] also proposed a sparse GP algorithm. They select the elements of the sub-set by minimizing the KL-divergence between the approximate posterior and the sparse representation. For our active learning classification problem we favour the IVM, mainly for the information-theoretic criterion used to select the subset of the training points.

7.2. Algorithm Overview

A typical sequence of our active learning framework for interactive segmentation is depicted in Fig. 7.3. From the initial user input, a set of labelled pixels (Fig. 7.3a) of both foreground and background, the Gaussian process classifier is trained. Then this classifier is used to estimate foreground and background probabilities for the remaining pixels of the image. The resulting segmentation of the image using this initial training set for classification is shown in Fig. 7.3b.

To improve the segmentation, first an uncertainty measure is computed from the predictive variance returned by the Gaussian process classifier. The benefit of the Gaussian process classifier is, that its uncertainty estimates are more reliable than those produced by other classifiers such as Support Vector Machines, where reliability of uncertainty estimates corresponds to a strong correlation between uncertain and incorrectly classified samples (see, e.g., [104]). For being able to present meaningful parts of the image to the user, we compute a partition of the image into larger regions, called super pixels, using the method of [44]. For each segment, we compute the average classification uncertainty (see Fig. 7.3c) and select the segment with the highest uncertainty to query a ground truth label from the user. After the

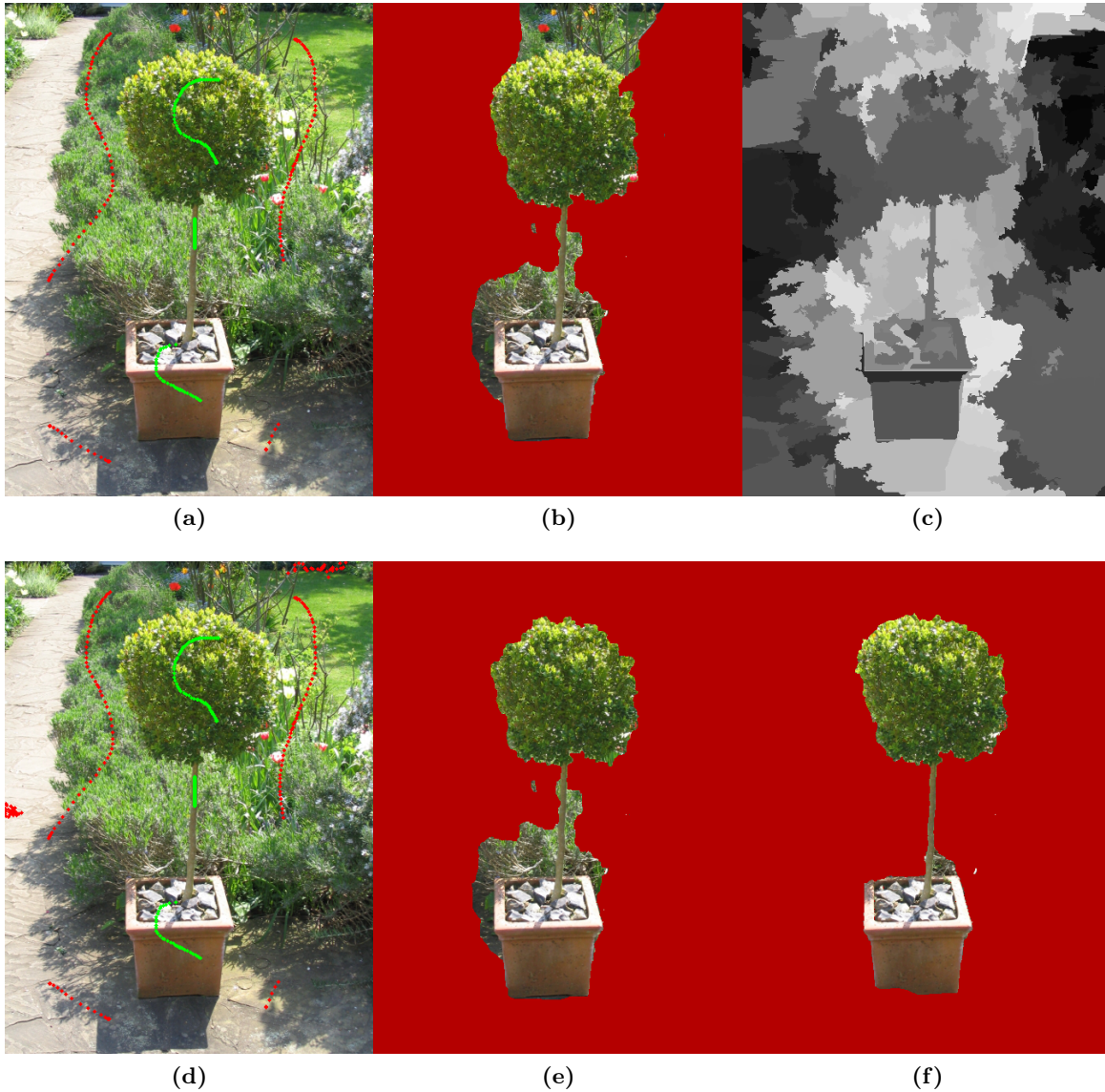


Figure 7.3.: Example sequence of our proposed active learning framework. The algorithm starts with initial user input as shown in (a). A sparse GP classifier is trained and the image is segmented using the GP prediction and a total variation spatial smoothness prior (b). Then, candidate regions for new, informative labels are computed (c). These are based on the normalized entropy of the GP prediction, where bright regions represent a higher classification uncertainty than darker regions. In this case, the segment with highest uncertainty at the upper right border is chosen. A label is queried for this region, here it is background, and a sub-set of uniformly sampled pixels in this region is added to the training data (d). In the next round, the classification is improved and the result is refined (e). After a few rounds, here 4 in total, the final segmentation is obtained (f).

user has assigned a label to this image region, a set of pixels is uniformly sampled from the region and, together with the obtained label, added to the training data set (see Fig. 7.3d). In some cases, a segment can contain both foreground and background of the scene. In that case, the user can select a “don’t know” option. Then the next segment with the next highest classification uncertainty is selected. However, this occurs only rarely in practice and can be avoided by computing a super pixel segmentation with sufficiently small regions. As last step of the active learning cycle the classifier is updated with the extended training set.

Above active learning round is iterated, either for a fixed number of iterations or until the classification uncertainty has become sufficiently small (Fig. 7.3e and 7.3f).

7.3. Gaussian Process Classification

In every round of the active learning algorithm, the Gaussian Process Classifier (GPC) is trained on the current training set. This set consists of user scribbles, these are user defined pixel locations in the image and their respective labels. We represent the training set as pairs $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$, where \mathbf{x}_i are feature vectors and $y_i \in \{-1, 1\}$ are binary labels denoting background or foreground. In our implementation, we use a combination of image coordinates and RGB colour values of the pixels as feature vector \mathbf{x}_i , but our active learning method can easily extend to higher level features, for example the output of special feature detectors. The use of image coordinated is motivated by the work of Nieuwenhuis and Cremers [95], to which we compare our method in the experimental section.

After training of the classifier we can compute the *predictive distribution* $p(y_* = 1 \mid \mathcal{X}, \mathbf{y}, \mathbf{x}_*)$, where (\mathbf{x}_*, y_*) is an unseen pixel/label pair, \mathcal{X} are the feature vectors in the training set, and \mathbf{y} are the labels in the training set. In a Gaussian Process Classifier, the predictive distribution is computed by first estimating a distribution $p(\mathbf{f} \mid \mathcal{X}, \mathbf{y})$ over the *latent variables* $\mathbf{f} \in \mathbb{R}^N$. This distribution is approximated by a multivariate normal distribution with mean $\vec{\mu}$ and covariance matrix Σ , i.e.: $p(\mathbf{f} \mid \mathcal{X}, \mathbf{y}) \approx \mathcal{N}(\mathbf{f} \mid \vec{\mu}, \Sigma)$ using Bayes’ rule:

$$p(\mathbf{f} \mid \mathcal{X}, \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{f})p(\mathbf{f} \mid \mathcal{X})}{\int p(\mathbf{y} \mid \mathbf{f})p(\mathbf{f} \mid \mathcal{X})d\mathbf{f}}, \quad (7.1)$$

where $p(\mathbf{f} \mid \mathcal{X}) = \mathcal{N}(\mathbf{f} \mid \vec{0}, K)$ is the prior of the latent variables, and

$$p(\mathbf{y} \mid \mathbf{f}) = \prod_i p(y_i \mid f_i) \quad (7.2)$$

are the likelihoods, which are conditionally independent. These likelihoods are given by *sigmoid function* Φ , i.e. $p(y_i \mid f_i) = \Phi(y_i f_i)$, such that Eq. (7.1) cannot be computed in closed form. Instead, these likelihoods are commonly approximated with Expectation Propagation (EP) and Assumed Density Filtering (ADF). This approximation yields a Gaussian distribution $q(y_i \mid f_i)$ that minimises the Kullback-Leibler (KL) divergence between $q(\mathbf{y} \mid \mathbf{f})p(\mathbf{f} \mid \mathcal{X})$ and the numerator of Eq. (7.1).

At test time, the GP classifier computes for a given new data point \mathbf{x}_* the mean μ_* and the variance σ_*^2 of the latent variable distribution

$$p(f_* \mid \mathcal{X}, \mathbf{y}, \mathbf{x}_*) = \int p(f_* \mid \mathcal{X}, \mathbf{x}_*, \mathbf{f})p(\mathbf{f} \mid \mathcal{X}, \mathbf{y})d\mathbf{f}. \quad (7.3)$$

Given the distribution of the latent variables, the predictive distribution can be modelled as

$$p(y_* = 1 \mid \mathcal{X}, \mathbf{y}, \mathbf{x}_*) = \int \Phi(f_*)p(f_* \mid \mathcal{X}, \mathbf{y}, \mathbf{x}_*)df_*. \quad (7.4)$$

When choosing Φ as the cumulative Gaussian function, the prediction can be computed in closed form using

$$p(y_* = 1 \mid \mathcal{X}, \mathbf{y}, \mathbf{x}_*) = \Phi \left(\frac{\mu_*}{\sqrt{1 + \sigma_*^2}} \right). \quad (7.5)$$

7.3.1. Information-theoretic Sparsification

A drawback that limits the practical applicability of Gaussian Process Classifiers is their huge demand of memory and run time. The high computational complexity of the method is due to the $N \times N$ covariance matrix that has to be maintained, where N is the number of samples in the training set that can be very large in practice. Therefore, we use a sparse version of the Gaussian Process Classifier, the Informative Vector Machine (IVM) [87]. This sparsification is achieved by only using a sub-set of the training data, the so called *active set* \mathcal{I}_D , which is used to compute an approximation q of the posterior. As in the original Gaussian Process Classifier, q is Gaussian, i.e. $q(\mathbf{f} \mid \mathcal{X}, \mathbf{y}) = \mathcal{N}(\mathbf{f} \mid \vec{\mu}, \Sigma)$. The IVM computes the vector $\vec{\mu}$ and the covariance matrix Σ incrementally, i.e. in every step j a new $\vec{\mu}_j$ and Σ_j are computed:

$$\vec{\mu}_j = \vec{\mu}_{j-1} + \Sigma_{j-1} \mathbf{g}_j \quad (7.6)$$

$$\Sigma_j = \Sigma_{j-1} - \Sigma_{j-1} (\mathbf{g}_j \mathbf{g}_j^T - 2\Gamma_j) \Sigma_{j-1} \quad (7.7)$$

where

$$\mathbf{g}_j = \frac{\partial \log Z_j}{\partial \vec{\mu}_{j-1}}, \quad \Gamma_j = \frac{\partial \log Z_j}{\partial \Sigma_{j-1}}, \quad (7.8)$$

and Z_j is an approximation to the denominator in Eq. (7.1) using the estimate q_j . Initially, we set $\vec{\mu}_0 = \vec{0}$, and $\Sigma_0 = K$, where K is the prior GP covariance matrix.

At every iteration, a new training point (\mathbf{x}_k, y_k) that maximizes the entropy difference between q_{j-1} and q_j is added to the active set, until the active set has reached a desired size D . In our experiments, we defined this size as a fixed fraction of the size of the training set N .

Because both \mathcal{I}_D and the kernel hyper parameters θ depend on each other, the training algorithm of the IVM iterates several times over two steps: estimation of the active set \mathcal{I}_D from θ and, for this given active set \mathcal{I}_D , minimizing the *marginal likelihood* Z_D using $\partial Z_D / \partial \theta$. Although there is no guaranteed convergence, in practice only a few iterations are needed to find good kernel hyper-parameters.

7.4. Segmentation Model

The IVM allows to infer predictions for the foreground and background class probabilities for a given image location. However, these estimates are only based on the local feature vector. To allow a consistent segmentation of the image with smooth object boundaries, we use the boundary length regularized image segmentation model from the previous chapters. We formulate the image segmentation problem according to (3.9) as

$$\min_{S \subseteq \Omega} \int_S f(x) \, dx + \lambda \text{Per}_\alpha(S, \Omega), \quad (7.9)$$

where $\text{Per}_\alpha(S, \Omega)$ measures the perimeter of S in Ω weighted by a local metric $\alpha(x) = e^{-\gamma|\nabla I|}$ that depends on the image gradient.

The probabilistic model of the IVM is included with

$$f(x) = \log \frac{p(y_* = -1 \mid \mathcal{X}, \mathbf{y}, \mathbf{x}_*)}{p(y_* = 1 \mid \mathcal{X}, \mathbf{y}, \mathbf{x}_*)}. \quad (7.10)$$

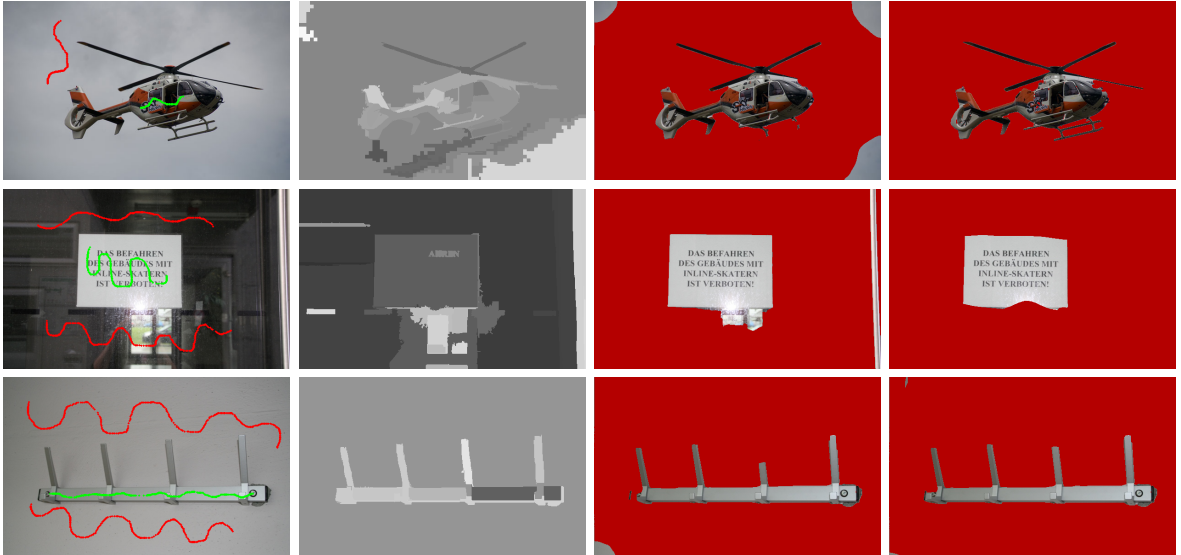


Figure 7.4.: Evaluation of our algorithm on the Graz benchmark. **Left column:** input image with initial user scribbles. **Second column:** classification uncertainties after the first learning round. **Third column:** resulting segmentation after the first round. Note that some small areas are misclassified, but the classification in those same areas is often very uncertain (see, e.g., the third peg on the wardrobe). Thus, the errors can be corrected by querying more useful, i.e. informative user scribbles. **Right column:** final segmentation results, obtained after a few further active learning rounds (between 1 and 5). Here, a segmentation of high-quality is obtained.

As in the previous chapters, we use the convex representation of Chan *et al.* [26] and define a continuous indicator function $u : \Omega \mapsto [0, 1]$ and get the convex optimization problem

$$\min_{u: \Omega \mapsto [0,1]} \int_{\Omega} f(x) u(x) dx + \lambda \int_{\Omega} \alpha(x) |\nabla u(x)| dx, \quad (7.11)$$

which can be minimized with the primal-dual hybrid gradient method, see [24, 25, 106, 107] and Section 2.5.

7.5. Experimental Results

To allow a comparison to the most related work of Nieuwenhuis and Cremers [95], we also use the benchmark data set from the University of Graz [114] for evaluation. The data set consists of images with predefined user scribbles and a groundtruth segmentation for every image. Because we implemented our method for two class image segmentation we chose a subset of 44 images from the dataset which contain only two object classes.

7.5.1. Benefits of the GP classifier

In the work of Nieuwenhuis and Cremers [95] the data term is computed using a Parzen window (PW) estimator, and the feature vector consists of the RGB colour channel and the position of a scribbles. We use the same idea and also use the RGB-colour value and the image coordinates as feature vector. In contrast to [95], we employ a Gaussian Process Classifier instead of the Parzen window estimator, with the benefit that misclassifications can be detected using the predictive uncertainty, which is more strongly correlated to incorrect classifications than for the Parzen windows estimator. We validate this assumption by performing the active learning

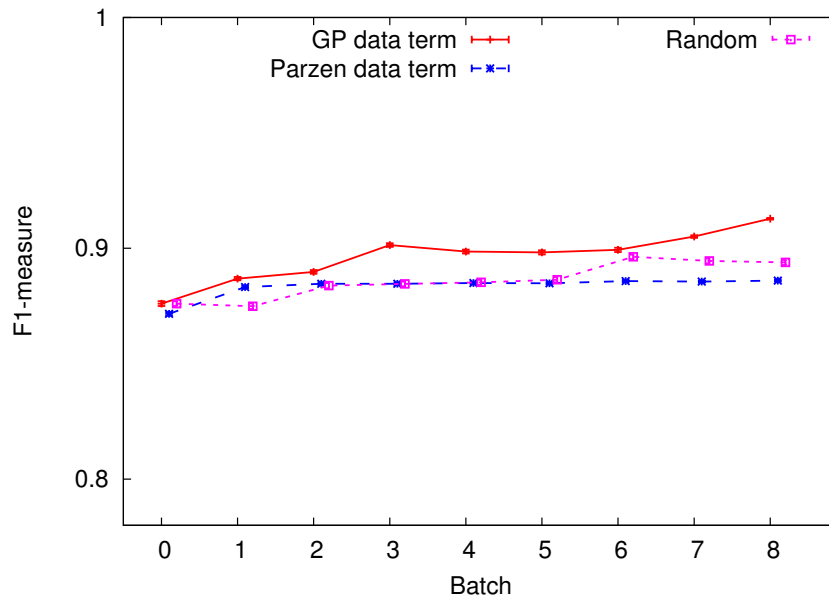


Figure 7.5.: Average f-measure over 8 active learning rounds. The GPC steadily improves the segmentation, because its label queries are more informative for classification. In contrast, the Parzen window estimator only improves slightly and then remains at a lower performance level. As a baseline comparison, we also show GPC results where new user scribbles are chosen randomly and not based on the classification uncertainty. This also improves the segmentation, as it increases the amount of training data, but it does not improve the result as quickly as the GPC with uncertainty based sampling.

approach on the Graz data set (Fig. 7.5). As a result, in active learning the GPC generates more informed questions.

Both approaches, the Gaussian Process Classifier (GPC) and the Parzen window estimator (PW), perform equally well in the first active learning rounds, but then the GPC (red curve) outperforms the PW (blue curve), because it asks more informed label queries. As a baseline comparison, we also show the result for randomly selected scribbles (magenta curve) instead of those with the highest uncertainty. We see that random sampling also improves the classification, as it provides more training data in every round, but the improvement is smaller compared to selecting the most uncertain image region.

Representative results from the Graz data set are shown in Fig. 7.4. The left column shows the images with the initial user scribbles. Columns two and three show the uncertainties of the GPC, where brighter is more uncertain, and the segmentation after the first learning round. The general segmentation quality is good, already after the first learning round, but some small misclassifications occur. However, these often correspond to locations of high uncertainty, e.g. the lower right corner of the helicopter image or the third peg on the wardrobe: here the classification is incorrect, but the uncertainty is also high. This allows to correct for these misclassified regions in subsequent training rounds.

We quantitatively compare our method to the method of Santner *et al.* [114] and Nieuwenhuis and Cremers [95], and report the dice-score in Table Fig. 7.6. The dice score is the relation between the overlap of each segmented region with the ground truth and the sum of their areas. In addition we present the f-measure of our results, which is defined as the harmonic mean of precision and recall.

Batch	Dice	F-Measure			
1	0.890	0.845	5	0.923	0.898
2	0.910	0.875	6	0.928	0.904
3	0.917	0.887	7	0.932	0.908
4	0.920	0.894	8	0.936	0.913
[95]	0.920	0.882			

Figure 7.6.: Dice and F-Measures on the Graz benchmark. Already after 4 active learning rounds our method produces as accurate results as the segmentation method of [95].

7.6. Conclusion

The presented active learning approach for user interactive image segmentation is able to significantly improve the classification performance over several active learning rounds. The method adaptively improves the classifier by informed questions based on the classification uncertainty.

We believe that this approach is particularly useful for the classification of large image data sets, where a manual inspection of the segmentation result is infeasible. Instead, our active learning approach identifies exactly those regions in the image, which have a high classification uncertainty and which are likely to be misclassified. This enables a user interactive validation and improvement of image segmentation of large image data sets.

Part III.

3D Reconstruction and Tracking

8. Connectivity Constraints for Image Based 3D Reconstruction

This chapter describes how to introduce connectivity constraints into spatio-temporal multiview reconstruction. In the previous part of the thesis, the connectivity constraint was introduced for the task of image segmentation: the labeling of an image into two different parts, the object and the background. Here we will use a very similar mathematical framework for spatio-temporal multiview reconstruction, which is based on a labelling of a volume into interior and exterior of the object to reconstruct. The use of this framework allows to adapt the connectivity constraints for image segmentation easily to the task of multiview reconstruction.

We also present an extension of the connectivity constraints: Previously, only connectivity of the object was required. We extend this framework to preserve loops as distinct topological features of the object. This has practical applications in volumetric multiview reconstruction. Starting from the so called visual hull of the object, the intersection of the interior of the silhouettes of the object from all perspectives, in a first step we detect loops in this visual hull and in the subsequent reconstruction step guarantee that these loops stay connected in the final segmentation. The combination of the connectivity constraint with the spatio-temporal multiview reconstruction method of Oswald and Cremers [100] allows a significant improvement in comparison to the state-of-the-art especially for scenes with fine structured details.

The chapter is organized as follows: First, we give an introduction to the spatio-temporal multiview reconstruction method, then we describe the modifications necessary to extend the connectivity prior to allow to preserve loops.

The work of this chapter has been published in [101] and [99]. Martin Oswald contributed the implementation of the spatio-temporal reconstruction algorithm. Jan Stühmer developed the theory and methods to include the connectivity constraint and extend it to preserve loops. The final integration of both methods and necessary implementations for analyzing the topology of the visual hull were performed by both authors.

Here, the work presented in [101] is extended by introducing the mathematically more precise notion of *k-connectivity* of the graph spanned by the edges of the constraints. We will see that loop connectivity constraints correspond to *2-connectivity* in comparison to the tree shape priors presented in the previous part of this thesis, which correspond to *1-connectivity* of the constraint graph.

8.1. Introduction

Multi-view 3D reconstruction is one of the classical topics in computer vision research. Given a set of images from different viewpoints the goal is to reconstruct the three dimensional geometry of the scene. Often, the scene consists of a single object one wishes to reconstruct, another application domain is 3D reconstruction from aerial photographs.

State-of the art reconstruction algorithms make use of a priori information about the scene,

such as smoothness of the surface or shape priors for the object to reconstruct. We will see in the following, how higher level features about the topology of the object can be incorporated into the reconstruction process.

To demonstrate the effectivity of topological constraints for 3D reconstruction we combine our framework for connectivity constraints with a state-of-the art algorithm for dynamic scene reconstruction: the spatio-temporal reconstruction method of Oswald and Cremers [100].

8.1.1. Contributions

We embed the concept of connectivity constraints for image segmentation into a spatio-temporal multi-view reconstruction framework. The connectivity constraints developed in the previous part of this thesis allowed to preserve *1-connectivity* on the directed graph defining the constraints. We extend this framework to allow to preserve loops as topological features, resulting in a *2-connectivity* constraint on the directed graph¹. Because the underlying spatio-temporal multi-view reconstruction approach of [100] is formulated in a convex optimization framework, and the connectivity constraint preserves the convexity, our method allows image based globally optimal 3D reconstruction while preserving connectivity.

8.1.2. Related Work

In their pioneering work for spatio-temporal multi-view reconstruction on dense occupancy grids Goldlücke et al. [56, 57] represent the dynamic scene as a space-time surface using a level set representation. The main drawback of their method is that level set representations only allow for local optimality of the solution, thus their method depends on a good initialization. Another local optimal approach was proposed by Aganj et al. [1]. The scene is also represented as a space-time surface, which is computed as spatio-temporal Delaunay mesh.

Starck and Hilton [123] propose to first estimate shapes from silhouettes and in a second step to refine the reconstruction with photometrically matched features together with temporal coherence. Guillemaut and Hilton [60] propose to use a multi-layer segmentation of the scene and assign a depth value to each layer based on confidence-weighted optical flow.

¹Refer to Section 4.2.1 for details on 1- and 2-connectivity.

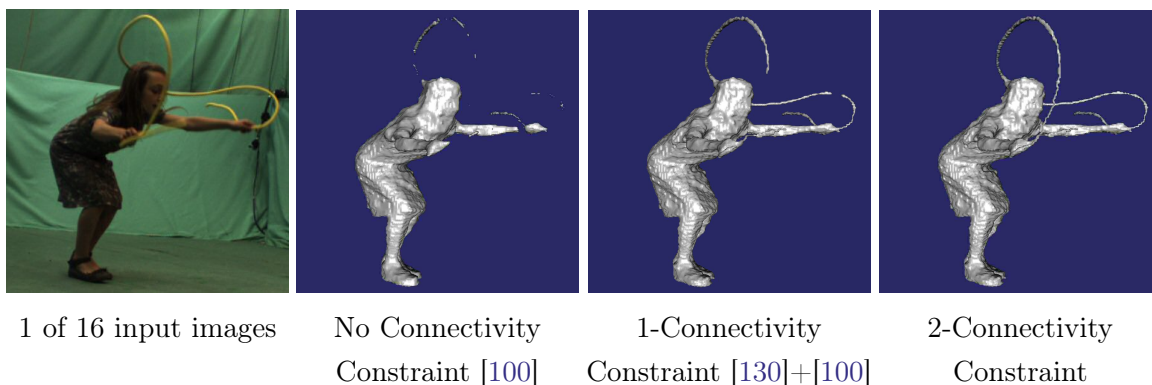


Figure 8.1.: Comparison of reconstruction results with and without connectivity constraints. Connectivity constraints clearly improve state-of-the art multi-view reconstruction methods and allow to recover fine structures like the rope in this example. The tree-shaped connectivity prior introduced in the previous chapter only enforces simple connectivity of the object, thus it is allowed that the rope is disconnected into multiple parts when it touches the head. In the following we will modify the connectivity constraints to allow to preserve loops of the object. Dataset: 'jumping rope' sequence from the INRIA 4D repository [68].

The spatio-temporal reconstruction method from Oswald & Cremers [100], on which the method presented in this chapter is based on, is a generalization of the 3D reconstruction framework by Kolev *et al.* [80] to the temporal domain. Similar to the image segmentation methods studied in the first part of this thesis, Kolev *et al.* and Oswald & Cremers represent the scene as the surface of a labeling function and optimize it using a convex optimization framework. This readily enables us to combine their framework with the connectivity constraints for image segmentation and build on top the results from the previous chapters.

To the best of our knowledge there is only one previous work for connectivity in 3D reconstruction, the work of Bleyer *et al.* [13]. They propose to solve for the stereo matching problem by concurrently computing a segmentation of the scene and impose connectivity on each segment of the scene. While we aim for a full 3D reconstruction in the spatial domain, Bleyer's method only computes a stereo matching: the scene is represented by assigning a depth value for every pixel of one of the images. In contrast to a full 3D reconstruction, such approaches are also called 2.5D stereo reconstruction methods.

8.2. 3D Reconstruction with Connectivity Constraints

First, we give an introduction to the spatio-temporal 3D reconstruction approach of [100] which allows 3D reconstructions of moving scenes by using video data from several viewpoints. Then we show how the connectivity constraint can be incorporated into the reconstruction. We verify with on real world data experiments that the constraints help to reconstruct fine scale details of the scene.

8.2.1. Spatio-temporal Multi-view Reconstruction

This section gives an introduction to the spatio-temporal 3D reconstruction approach of [100]. The task of spatio-temporal 3D reconstruction is formulated as convex optimization problem, which allows to include the connectivity constraints without difficulty. The dynamically changing scene is represented by a hypersurface $\Sigma \subset \mathbb{V} \times \mathbb{T}$ that is embedded in the spatio-temporal product space of the three-dimensional Euclidian space $\mathbb{V} \subset \mathbb{R}^3$ with the time dimension $\mathbb{T} \subset \mathbb{R}_{\geq 0}$. The scene is observed by N static calibrated cameras with projection matrices $\{\pi_i\}_{i=1}^N$. Furthermore, for every image at every timepoint t we have the approximate silhouettes $\{S_i(t)\}_{i=1}^N$. The method does not need exact silhouettes, which allows to extract the silhouettes automatically, for example by 2D image segmentation or, in case of a controlled background, even simpler methods. Recordings in front of a green or blue screen allow to extract the silhouettes with a method called chroma keying, which is broadly used in television broadcasting.

These silhouettes from different perspectives allow to impose geometric and topological constraints on the reconstruction process. The geometric constraint is given by the visual hull $\mathcal{VH}(t) = \bigcap_{i=1}^N \pi_i^{-1}(S_i(t))$ and states that the projections of the reconstructed object into the viewpoint of every camera have to stay within the silhouette $S_i(t)$ of this viewpoint. In the following section we will also see, how constraints on the topology of the object can be derived from the visual hull.

The hypersurface Σ is represented as the 1-level-set of the binary label function $u : \mathbb{V} \times \mathbb{T} \mapsto \{0, 1\}$, indicating either interior or exterior for every point in space-time. This implicit representation ensures that the surface is a closed, not necessarily connected, manifold without a boundary while allowing arbitrary topologies.

Furthermore, we are given a photoconsistency measure $\rho : \mathbb{V} \times \mathbb{T} \mapsto \mathbb{R}_{\geq 0}$, that describes a measure for a point in space-time for being on the surface of the object, as well as a data term

$f : \mathbb{V} \times \mathbb{T} \mapsto \mathbb{R}$ that expresses an affinity to an interior ($f < 0$) or an exterior ($f > 0$) labeling.

We are now able to formulate the space-time reconstruction problem as energy minimization problem which amounts to find a surface of minimum area, where this area is weighted by the local photoconsistency, such that a smaller weight corresponds to higher agreement with the image data. We combine this problem of finding a minimum surface with the region based data term and get the energy minimization problem

$$E(u) = \int_{\mathbb{V} \times \mathbb{T}} \left(\rho |\nabla_x u| + g_t |\nabla_t u| \right) dx dt + \lambda \int_{\mathbb{V} \times \mathbb{T}} f u dx dt \quad (8.1)$$

where $\lambda > 0$ is the weight for a reularizer that measures the smoothness of the reconstructed hypersurface and the function $g_t(x, t) = \exp(-|\nabla f(x, t)|)$ weights the temporal smoothness based on f to allow fast motions.

Following [100], we restrict the solution space of the energy minimization in (8.1) to the visual hull such that the object's outline from every camera viewpoint has to remain inside the visual hull. As a consequence, the visual hull completely contains the interior of the scene.

The photoconsistency measure $\rho(x)$ is a voting scheme that is based on truncated normalized cross-correlation matching scores C_i between neighboring camera pairs:

$$\rho(x) = \exp \left[- \mu \sum_{i \in \mathcal{C}} \underbrace{\delta(d_i^{\max} = \text{depth}_i(x)) \cdot C_i(x, d_i^{\max})}_{\text{VOTE}_i(x)} \right], \quad (8.2)$$

with scaling parameter μ . Together with $d_i^{\max} = \arg \max_d C_i(x, d)$ the delta function δ performs a ray-based denoising of the photoconsistency measures and represents the voting scheme proposed by Hernández and Schmitt [42].

The data term f avoids the empty set as trivial solution of energy Eq. (8.1) by propagating the photometric information from Eq. (8.2) into the volume.

$$f(x, t) = - \ln \left(\frac{1 - P(x \in \text{int}(\Sigma))}{P(x \in \text{int}(\Sigma))} \right). \quad (8.3)$$

The probability $P(x \in \text{int}(\Sigma))$ that point x belongs to the interior $\text{int}(\Sigma)$ of surface Σ is defined based on the locations and qualities of votes along the camera rays $r_i(x, \cdot)$ through point x

$$P(x \in \text{int}(\Sigma)) = \prod_{i=1}^N \prod_{j=1}^N \prod_{\text{depth}_i(x) < d \leq d_i^{\max}} \frac{1}{Z_j} \exp \left[-\eta \cdot \text{VOTE}_j(r_i(x, d)) \right]. \quad (8.4)$$

To limit the memory consumption we follow the approach in [100] and solve for a timepoint t by computing a solution of (8.1) with respect to $t - 1$, t and $t + 1$, thus limiting the number of timepoints used to estimate a surface for timepoint t to $|\mathbb{T}| = 3$. For each timepoint a mesh is extracted with the Marching Cubes algorithm [93] at an iso-level of 0.5.

8.2.2. Connectivity Constraints for 3D Reconstruction

Without loss of generality we assume that the visual hull is connected. For the case that is not connected, the same approach can be applied component-wise after identifying independent connected components of the visual hull. We define connectivity constraints independently for each time step to allow for topology changes between time steps. For better readability we drop the temporal dependency in the following notation.

Graph Structure. For every time step we define a geodesic shortest path tree G_s on the visual hull \mathcal{VH} with respect to a given source node s that contains for each point $x \in \mathcal{VH}$ inside the visual hull the shortest geodesic path $C_s^x : [0, 1] \mapsto \mathbb{R}^3$ with $C_s^x(0) = s$ and $C_s^x(1) = x$ that minimizes the cost function

$$\mathcal{D}_s(x) = \int_0^1 e^{f(C_s^x(t))} dt, \quad (8.5)$$

which is a positive geodesic measure that depends on the data term and results in lower costs for paths that stay inside the interior of the object. $\mathcal{D}_s(x)$ is the geodesic distance from the source node s to every point $x \in \mathcal{VH}$. In a discretized domain, the edges along the shortest paths form the edge set E of the resulting shortest path tree G_s .

Source Node Computation. Because the tree shape prior enforces a certain star shaped topology, it is desirable to choose the source node for the geodesic shortest path tree to be 'central' to the reconstructed object. To this end, we compute the source node $s(t)$ as the point which minimizes a spatio-temporal convolution of the data term f with a sufficiently large Gaussian kernel \mathcal{G} .

$$s(t) = \operatorname{argmin}_x \int_{t-1}^{t+1} (f * \mathcal{G})(x, \tau) d\tau \quad (8.6)$$

We minimize above term because negative data term values $f < 0$ indicate a favor for an interior label and thus ensure a position that has high probability of being interior. This choice favors a smooth temporal change of its position within the data term while maximizing the distance to the surface. In practice however, the exact position of the source node has not much influence on the result. An example rendering of a shortest path from a leaf node to the source is shown in Fig. 8.4a.

Constrained Optimization. The connectivity constraint from [130], which was also presented in the previous chapters, is included into the reconstruction process as a monotonicity constraint of the labeling function u with respect to the edges E in G_s . This monotonicity can be ensured by including inequality constraints on the directional derivative $\partial_i u(x, t)_j$ of u along every edge $ij \in E$. Thus, computing a spatio-temporal 3D reconstruction with connectivity constraints can be achieved by computing a minimizer of the constrained optimization problem

$$\begin{aligned} \min_{u \in \mathcal{BV}(\mathbb{W} \times \mathbb{T}; \{0,1\})} & E(u) \\ \text{s.t.} & \partial_i u_j \leq 0, \quad \forall (i, j) \in E \end{aligned} \quad (8.7)$$

with one constraint for each edge in the edge set E of the shortest path tree G_s . $\mathcal{BV}(\cdot)$ denotes the function space of bounded variations [3].

8.3. Loop Connectivity

In the following we will see how a topological analysis of the visual hull allows to derive suitable topological constraints in a fully automatic process. Therefore, we first describe how to extend the connectivity constraints introduced in the previous part of this thesis to allow to preserve loops. While the tree shape prior only guaranteed that the object is path connected, which means that it is 1-connected on the underlying graph, in the following we will show how the framework can be extended to allow to preserve 2-connectivity on the underlying graph for specific parts of the object. This section is organized as follows: First we describe

how the framework is extended to formulate 2-connectivity constraints which allow to require connected cycles in the underlying graph. Then we describe how the topology of the visual hull can be automatically analyzed to identify topological features of the object which should be preserved.

8.3.1. Loop Connectivity Constraints

So far, the connectivity constraint requires that the vertices in the foreground segment induce a 1-connected subgraph on the tree defined by the connectivity constraints : Let $T = (V, E_T)$ denote the tree of the connectivity constraints that are defined along the edge set E_T . Let $u : V \mapsto \{0, 1\}$ be a feasible labeling with respect to the connectivity constraints defined by E_T . The vertices v with label $u(v \in V) = 1$ form the set of vertices inside the foreground segment $\Sigma_u \subseteq V$, thus $\Sigma_u = \{v : u(v \in V) = 1\}$. As shown in Section 5.2.3, Σ_u is connected on T , thus the subgraph $T_\Sigma = T[\Sigma_u]$ induced by Σ_u is connected. However, because T_Σ is a tree, there is only one possible path from the source vertex s to each vertex $v \in T_\Sigma$. This leads us to the following statement about the connectivity of a connected tree.

Theorem 8.3.1. *Let $T = (V_T, E_T)$ be a connected tree with $|V_T| > 2$, then the connectivity of T is exactly 1.*

For the proof we need the following proposition:

Proposition 8.3.2. *Let $T = (V_T, E_T)$ be a connected tree with $|V_T| > 2$, then T has at least one internal vertex.*

Proof. An internal vertex has a degree of at least 2. Lets assume there is no such vertex, then there exist only single vertices and pairs of vertices. The largest connected graph with vertices of degree less than 2 is a single pair of 2 vertices. Thus, every connected graph with more than two vertices has at least one internal vertex. \square

Now we can prove theorem 8.3.1, that every connected tree with more than two vertices is 1-connected:

Proof. Let $v \in V_T$ be an internal vertex of T . We denote all paths through v with the set $\mathcal{P} : \{P \ni v\}$. Because there is only one single path that connects the end vertices of each $P \in \mathcal{P}$, these end vertices are not connected when v is removed. Thus the graph $T \setminus v$ is not connected which concludes the proof. \square

If the object to reconstruct has a more complex topology of a higher genus, and we would like to preserve this topology, this connectivity constraint is not sufficient, as it does not allow to formulate the requirement of connected loops of the object. However, we will see in the following how to extend the constraint to formulate a connectivity constraint for loops, while still using the shortest geodesic path topology.

Theorem 8.3.3. *Let $T = (V_T, E_T)$ be a connected undirected tree. Let $e = ab$ be an edge with $a \in V_T, b \in V_T, a \neq b$ that is not in E_T , i.e. $e \notin E_T$. The graph $G = (V_T, E_T \cup \{e\})$ that we get by adding the edge e to T , has a cycle which contains e .*

Proof. The tree T contains the unique connected path P from the root vertex s to a , which we denote with sPa , and the unique path Q from s to b , which we denote with sQb . Because both P and Q originate in s it holds that $V(P) \cap V(Q) \ni s$ and therefore $V(P) \cap V(Q) \neq \emptyset$. There exists a vertex x_i , which is both in $V(P)$ and in $V(Q)$, and for which the adjacent

vertex in P , $x_{i+1}^P \in P$, is not in Q , likewise it holds for the adjacent vertex in Q , $x_{i+1}^Q \in Q$, and $x_{i+1}^Q \notin P$. Then it holds that $x_i P a \cap x_i Q b = \{x_i\}$. We form the cycle $x_i P a \cup ab \cup b Q x_i$. \square

Thus by adding an additional edge to tree that connects two vertices which were previously not connected we get a graph which contains a cycle. We will describe in the following, how to define the connectivity constraint on a cycle. For the 1-connectivity constraint on the directed tree, the connectivity constraint was defined as monotonicity constraint of the label function along each path of the tree. The label should not increase when traversing along a directed edge, with each edge pointing into the direction of increasing distance from the root vertex. To get a graph with a cycle from a directed tree, we first discard the directions of the edges and add a cycle to the resulting undirected tree as shown above. Then we assign an arbitrary, but consistent direction to the cycle, that allows to traverse the cycle along the directed edges, we call this an *oriented cycle*.

Theorem 8.3.4. *Introducing a monotonicity constraint of the label function along an oriented cycle results in an equality constraint along the cycle.*

Proof. Let $x_i P a \cup ab \cup b Q x_i$ be a directed cycle. We use the definitions of the paths P and Q and the vertices $x_i, x_{i+1}^P \in P$ and $x_{i+1}^Q \in Q$ of the previous proof. Then there is a directed edge from x_i to x_{i+1}^P and a directed edge from x_{i+1}^Q to x_i . The monotonicity along the edge from x_i to x_{i+1}^P implies that $u(x_i) \geq u(x_{i+1}^P)$. The monotonicity along $x_{i+1}^P P a \cup ab \cup b Q x_{i+1}^Q$ implies that $u(x_{i+1}^P) \geq u(x_{i+1}^Q)$, together with the remaining constraint $u(x_{i+1}^Q) \geq u(x_i)$ we get

$$u(x_i) \geq u(x_{i+1}^P) \geq u(x_{i+1}^Q) \geq u(x_i) \quad (8.8)$$

which only holds when all the values of the label function along the cycle are equal. \square

8.3.2. Handle and Tunnel Loops

In [33], Dey et al. describe a method to automatically analyze the topology of a two-dimensional surface embedded in three dimensional space that allows to extract topological features, specifically the *handles* and *tunnels* of the surface. The surface is embedded in a simplicial complex, a hierarchy of p -simplicies. The surface \mathbb{M} separates the simplicial complex into an interior part \mathbb{I} and an exterior part \mathbb{E} . Both the interior and the exterior part are closed and bounded by the surface, therefore $\mathbb{I} \cap \mathbb{E} = \mathbb{M}$. Since we want to analyze the topology of the visual hull, in the following we will denote the surface of the visual hull with $\mathbb{M} = \partial \mathcal{VH}$, the interior of the visual hull with $\mathbb{I} = \mathcal{VH}$ and the exterior with $\mathbb{E} = (\mathbb{V} \setminus \mathcal{VH}) \cup \partial \mathcal{VH}$.

In [33] the authors define and study cycles of edges ('loops') on the surface which form equivalence classes with respect to contraction of the cycle - like a rubber band which can be moved on the surface but not through holes in the surface. We call this equivalence relation $\sim_{\mathbb{M}}$ 'contractible' on the set \mathbb{M} , for example, we denote the relation that a loop $l_1 \subset \mathbb{M}$ is contractible to a loop $l_2 \subset \mathbb{M}$ on the set \mathbb{M} as $l_1 \sim_{\mathbb{M}} l_2$.

Handle and tunnel loops A **handle loop** $h \subset \mathbb{M}$ is a cycle of edges on the surface that is contractible in the interior ($h \sim_{\mathbb{I}} 0$) and not contractible on the surface ($h \not\sim_{\mathbb{M}} 0$). A **tunnel loop** $t \subset \mathbb{M}$ is a cycle of edges on the surface that is contractible in the exterior ($h \sim_{\mathbb{E}} 0$) and not contractible on the surface ($h \not\sim_{\mathbb{M}} 0$).

Handle and tunnel loops have the following two important properties: First, both definitions form equivalence classes of loops. Two loops are in the same equivalence class if there exists a continuous transformation between them. Second, the classes of *handle* and *tunnel* loops are

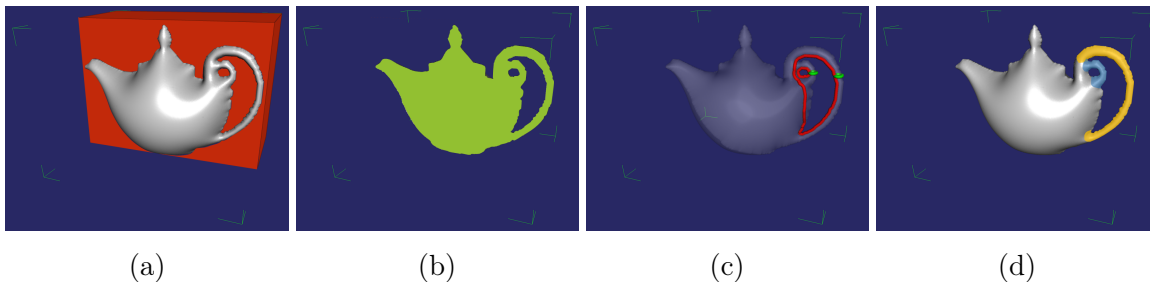


Figure 8.2.: The sets defined in this section are visualized for the surface of a teapot of genus 2. (a) Exterior \mathbb{E} (red), (b) Interior \mathbb{I} (green), (c) Handle and tunnel loops $\{h_1, h_2\}, \{t_1, t_2\}$ (green+red), (d) Handle segments H_1, H_2 (yellow+blue).

dual to each other, for each handle loop there exists a corresponding tunnel loop. Furthermore, a closed surface of genus g has exactly g classes of handle loops and g classes of tunnel loops. We consider one representative loop with approximate minimal geometric length per class and denote them as the set of handle loops $\{h_i\}_{i=1}^g$ and the set of tunnel loops $\{t_i\}_{i=1}^g$. For each hole i of the surface we have a corresponding pair (h_i, t_i) of representative handle and tunnel loops which intersect in at least one point, i.e. $h_i \cap t_i \neq \emptyset$. Figs. 8.2, 8.3, 8.4c show examples of handle and tunnel loops which clearly shows their duality.

Dey et al. propose different algorithms for computing these handle and tunnel loops: In [33] they present a method that is able to process geometries defined by implicit functions, perfectly suited to process the geometry defined by the volumetric labeling function we use here for 3D reconstruction. Also their algorithm allows to compute handle and tunnel loops with approximate minimal length, which is desirable for our purpose of segmenting the handles from the rest of the object. However this method is considerably slower than a recently published algorithm by Dey et al. [32] for meshes. The faster runtime is achieved by using the concept of Reeb graphs to estimate an initial set of handle and tunnel loops and their geometric length is shortened in a subsequent refinement step. Also the method does not require to compute a full 3D tessellation of the scene. Therefore, we extract an iso-surface mesh of the visual hull and use the more efficient method [32] to extract handle and tunnel loops.

Handle Segmentation. We aim to segment the 'thin' geometric parts around the holes of the surface, called handles. These handle segments will help to make the connectivity constraints adaptive to the data term. For this purpose we introduce the following definitions.

Handle Segment Surface We define the handle segment surface as the connected subset of all points $x \in \mathbb{M}$ for which a handle loop $h_x \ni x$ exists which is contractible to h_i subject to the additional constraint that the ratio of the lengths of h_x and h_i does not exceed a given threshold σ :

$$\mathbb{M}_{H_i} = \left\{ x \in \mathbb{M} \mid \exists h_x \subset \mathbb{M} : h_x \sim_{\mathbb{I}}^{\sigma} h_i \right\} \quad (8.9)$$

where $h_x \subseteq \mathbb{M}$ with $h_x \ni x$ denotes a handle loop through the surface point x and $h_x \sim_{\mathbb{I}}^{\sigma} h_i$ means that handle loop h_x is contractible to h_i subject to the length ratio constraint $\ell(h_x) < \sigma \ell(h_i)$.

Handle Segment Given the handle segment surface \mathbb{M}_{H_i} from the previous definition, we define the corresponding volumetric handle segment $H_i \subseteq \mathbb{I}$ as the set of all points in the visual hull for which the closest point on the visual hull boundary is on the handle segment

surface \mathbb{M}_{H_i} .

$$H_i = \left\{ x \in \mathbb{I} \mid \underset{y \in \mathbb{M}}{\operatorname{argmin}} \operatorname{dist}(x, y) \in \mathbb{M}_{H_i} \right\} \quad (8.10)$$

where $\operatorname{dist}(x, y)$ denotes the Euclidean distance between point $x \in \mathbb{I}$ in the interior and point $y \in \mathbb{M}$ on the surface.

In practice, we compute H_i by a breadth first search algorithm on the visual hull. Starting from the handle loop h_i a wavefront is propagated in both directions. Independently for each wavefront, we stop the search if the ratio between the current length of the wavefront and the initial position exceeds the threshold σ .

With the definition of the handle and tunnel segments of the visual hull we are now able to formulate connectivity constraint to preserve these topological features. Therefore, we define a cycle through the interior of each handle segment as follows: In order to add a minimum amount of cost to the energy (8.7) when enforcing loop connectivity, we need to form a cycle in the graph with minimum cost with respect to the data term. We solve for these geodesic shortest cycles by computing cycles $t_i^{G_s} \subset \mathbb{I}$ using the precomputed geodesic shortest path tree G_s . These cycles need to be *path homotopic* to the original tunnel loop on the surface, i.e. $t_i^{G_s} \sim_{\mathbb{I}} t_i$. For an introduction to path homotopy please refer to the introductory material in section Section 4.1.2. The computation of the minimum cost cycle $t_i^{G_s}$ is discussed later in this section. First we discuss possible definitions of constraints to preserve the topology of the visual hull. For each tunnel loop t_i of the visual hull we define a path homotopic cycle $t_i^{G_s} \sim_{\mathbb{I}} t_i$ and introduce a *loop preserving* constraint along this cycle as

$$\forall i \in [1, \dots, g] : \quad \left\{ \forall x \in t_i^{G_s} : u(x) = 1 \right\}. \quad (\text{LC0})$$

Proposition 8.3.5. *The constraint (LC0) preserves the handle and tunnel loops and thus all holes of the visual hull in the reconstructed object. The topological genus of the reconstructed object is larger or equal to the one of the visual hull.*

Proof. [101] Let us assume that the proposition does not hold. To let the genus of the reconstructed object decrease, either (i) at least one hole of the visual hull needs to be filled or (ii) at least one tunnel loop has to be disconnected in the reconstructed object. Because the domain of the reconstructed object is restricted to the visual hull, (i) cannot be fulfilled. By construction, (ii) is fulfilled if (LC0) is fulfilled. Therefore the genus of the reconstructed object has to be larger or equal to the genus of the visual hull. \square

Note that, depending on the data term f the reconstructed object is allowed to have a higher genus than the visual hull. In some cases, it is not desirable to preserve all handles of the visual hull. A possible scenario is depicted in Fig. 8.3 where artifacts of the visual hull lead to spurious handle loops which should not be preserved in the final reconstruction. Therefore we propose to relax the loop preserving constraint (LC0) to an equality constraint (LC1) of the labels along a cycle such that either the connectivity of a handle is preserved in the final reconstruction or, in case the photometric support via f is not strong enough, the handle segment H_i is suppressed completely. We define this relaxed constraint as

$$\forall i \in [1, \dots, g] : \quad \left\{ \forall x \in t_i^{G_s} \cap H_i : \frac{d}{ds} u(x) = 0 \right\} \quad (\text{LC1})$$

where $\frac{d}{ds}$ is the directional derivative along the loop $t_i^{G_s}$.

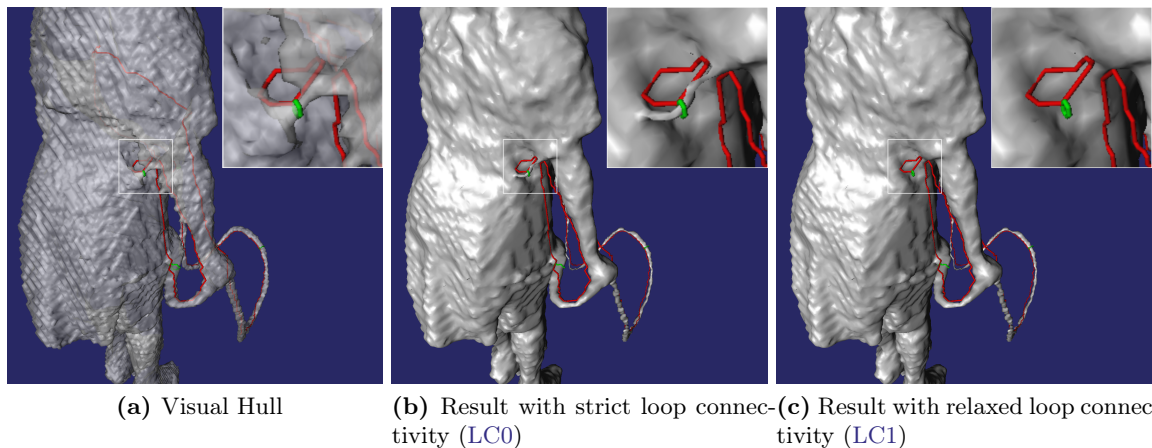


Figure 8.3.: (a) In some cases artifacts of the visual hull can lead to spurious handle loops which should not be preserved in the final reconstruction. (b) The constraint (LC0) strictly preserves all loops in the solution. (c) Relaxing the topology preserving constraint to an equality constraint allows to suppress handles where the photoconsistency is not strong enough. The rope, for which the support of the photoconsistency is sufficient, is still completely preserved. Handle loops are depicted in green and tunnel loops in red.

Finding the optimal connected loop $t_i^{G_s}$. For the 1-connectivity constraint the use of the shortest path tree is motivated by finding a connecting path, that adds a low cost to the final segmentation result. In case of objects with higher genus, we wish to preserve the connectivity with respect to loops in the final segmentation. Therefore a loop through each handle needs to be found, which adds minimum cost to the final segmentation. To compute this best connecting path we don't have to compute another shortest distance on the graph but can utilize the already computed shortest path tree G_s . We can find the shortest loop $t_i^{G_s}$ with respect to G_s for each handle i by the following steps: With a depth first search on G_s , starting from the boundary of a handle segment H_i , we compute the partitions $H_i^1 \cup H_i^2 = H_i, H_i^1 \cap H_i^2 = \emptyset$ which are disconnected on the shortest path tree G_s . These partitions are shown in Fig. 8.4d. If one of these partitions is empty, i.e. all points in the handle segment H_i are connected on G_s , then no further constraints need to be added in order to preserve handle segment H_i . Otherwise, we compute an optimal pair of points

$$(p, q) = \underset{(x \in H_i^1, y \in H_i^2, y \in \mathcal{N}(x))}{\operatorname{argmin}} \mathcal{D}_s(x) + \mathcal{D}_s(y) \quad (8.11)$$

which are leaf-nodes in G_s . The set $\mathcal{N}(x)$ denotes the local spatial neighborhood of a point $x \in \mathbb{V}$. By adding an edge between p and q we generate a cycle $t_i^{G_s}$ in G_s , that is in the interior of H_i , is path homotopic to t_i , and is of minimum cost of all such cycles. We define the set $E_{=} = \bigcup_i E(t_i^{G_s})$ as the set of edges of all $t_i^{G_s}$ and define our loop connectivity constraints as

$$\partial_i u_j = 0, \quad \forall (i, j) \in E_{=}. \quad (8.12)$$

While the tree connectivity constraint resulted in an inequality constraint on the derivative of the label function, the loop connectivity is preserved by adding an equality constraint (Theorem 8.3.4).

8.4. Numerical Optimization

To minimize energy (8.7) using convex optimization we first relax the discrete image function to the continuous interval $[0, 1]$. The constraints defined on the derivative of the image function

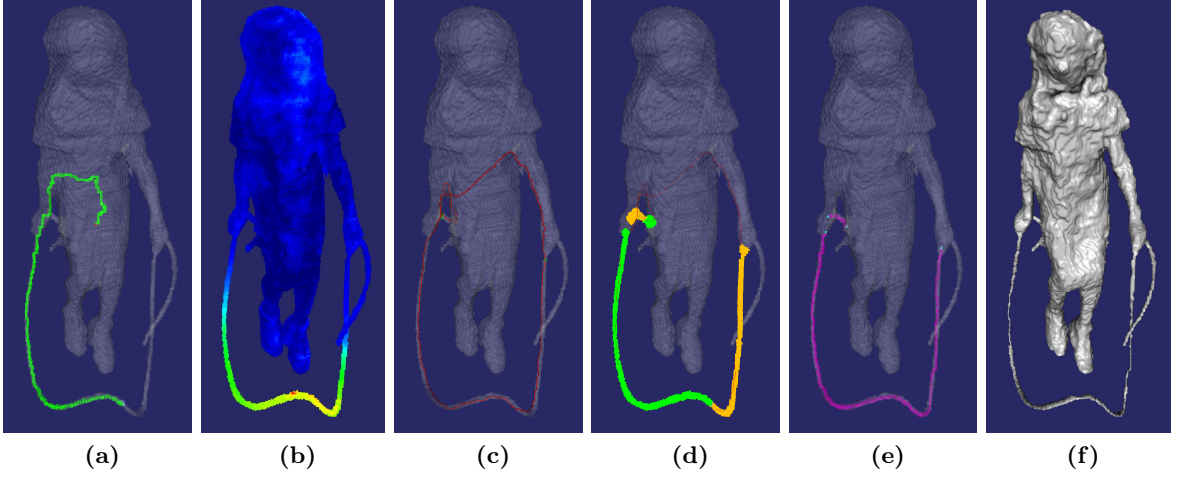


Figure 8.4. Visualization of quantities computed from the data term and visual hull. (a) Example shortest path from the source node to a leaf node in the rope (green); (b) color encoded geodesic distance \mathcal{D}_s with respect to the source node s ; (c) tunnel loop (red), handle loop (green); (d) handle segmentations $H_i = H_i^1 \cup H_i^2$ (green and orange), the color shows the two parts of each handle which are disconnected on the geodesic path tree G_s . (e) shortest path through the handle for which the equality constraints (LC1) are imposed; (f) final reconstruction result.

remain the same as in the discrete setting.

Because the total variation norm is non-differentiable, we introduce a dual variable

$p : \mathbb{V} \times \mathbb{T} \mapsto \mathbb{R}^4$ and reformulate the optimization problem Eq. (8.7) as the equivalent saddle-point problem

$$\begin{aligned} \min_{u: \mathbb{V} \times \mathbb{T} \rightarrow [0,1]} \quad & \max_{\substack{\|p\| \leq 1 \\ \mathbb{V} \times \mathbb{T}}} \int \langle u, -\operatorname{div} p \rangle dxdt + \lambda \int_{\mathbb{V} \times \mathbb{T}} f u dxdt \\ \text{s.t.} \quad & \partial_i u_j \leq 0, \quad \forall (i, j) \in E_T \\ & \partial_i u_j = 0, \quad \forall (i, j) \in E_- \end{aligned} \quad (8.13)$$

The constraints on u over the edge sets E_T and E_- are linear, the optimization problem is convex in u and concave in p . Furthermore, the feasible set defined by the constraints is non-empty, a trivial feasible solution is $u(x, t) = 0$ for all $x \in V$ and $t \in T$, thus Slater's condition holds and the constraints can be included in the optimization using Lagrangian multipliers β and γ . The Lagrangian dual associated to problem (8.13) becomes

$$\begin{aligned} \min_{u: \mathbb{V} \times \mathbb{T} \rightarrow [0,1]} \quad & \max_{\substack{\|p\| \leq 1, \\ \beta \geq 0, \\ \gamma}} \int_{\mathbb{V} \times \mathbb{T}} \langle u, -\operatorname{div} p \rangle dxdt + \lambda \int_{\mathbb{V} \times \mathbb{T}} f u dxdt \\ & + \int_T \left\{ \sum_{ij \in E_T} \beta_{ij} \partial_i u_j + \sum_{ij \in E_-} \gamma_{ij} \partial_i u_j \right\} dt. \end{aligned} \quad (8.14)$$

We optimize this saddle point problem using the preconditioned primal-dual algorithm by Pock and Chambolle [105]. The algorithm results in an iterative update scheme with a projected

gradient ascent in the dual and a projected gradient descent in the primal variable

$$\begin{aligned}
 p^{n+1} &= \Pi_C [p^n + \sigma \nabla \bar{u}^n] \\
 \beta_{ij}^{n+1} &= \Pi_{\geq 0} (\beta_{ij}^n + \mu \partial_i \bar{u}_j^n) \\
 \gamma_{ij}^{n+1} &= \gamma_{ij}^n + \nu \partial_i \bar{u}_j^n \\
 u^{n+1} &= \Pi_{[0,1]} \left[u^n + \tau \left(\operatorname{div} p^{n+1} + \operatorname{div} \beta^{n+1} + \operatorname{div} \gamma^{n+1} - \lambda f \right) \right] \\
 \bar{u}^{n+1} &= 2u^{n+1} - u^n
 \end{aligned} \tag{8.15}$$

where $\Pi_{[0,1]}$ is the projection of u onto the unit interval $[0, 1]$ and $\Pi_{\geq 0}$ projects onto positive values. The projection onto the set $C = \{q = (q_x, q_t)^T : \mathbb{V} \times \mathbb{T} \mapsto \mathbb{R}^4 \mid \|q_x\| \leq 1, |q_t| \leq 1\}$ is a projection on a four dimensional hyperball and defined as

$$\Pi_C(q) = \left(\frac{q_x}{\max(1, \frac{\|q_x\|}{\rho})}, \max(-g_t, \min(g_t, q_t)) \right)^T \tag{8.16}$$

The step sizes τ , σ , μ and ν are chosen as described in [105]. Because the optimization problem is convex in u , concave in p , and the constraints are linear, the update scheme (8.15) converges to a global minimum of the relaxed energy (8.7). An optimal binary labeling can be found by thresholding the relaxed solution.

Implementation. The resulting iterative scheme for minimal surface reconstruction with connectivity constraints (8.15) allows a high degree of parallelization and is implemented using the CUDA programming framework. The connectivity graph precomputation is more difficult to parallelize and therefore is implemented on the CPU using the boost graph library.

8.5. Experiments

We evaluated our method on several spatio-temporal multi-view data sets provided by the INRIA 4D repository [68]. Each dataset contains synchronized video recordings of 16 cameras in a green room environment.

In the experiments we mainly focus on comparing reconstruction results with and without the proposed connectivity constraints. Since no other 4D reconstruction method is publicly available, we compare our results with results of state-of-the-art 3D reconstruction methods and evaluate them on the same dataset. This is the reconstruction method by Jancosek and Pajdla [71], and a combination of Furukawa et al. (PMVS) [49] with Poisson surface reconstruction [76].

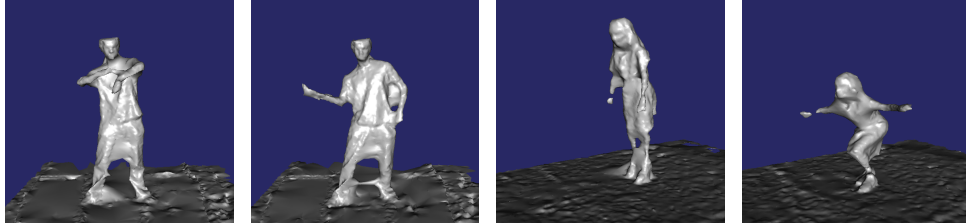
Approximate silhouette information was used by all methods except of the method by Jancosek and Pajdla [71]. The geodesic shortest path tree G_s was computed using a regular 6-neighborhood.

Runtime and Memory Resource Evaluation. To encode the connectivity constraint, the memory requirement of the suggested implementation increases only by $|\mathbb{V} \times \mathbb{T}|$ bytes in comparison to the original approach. The numerical optimization runtime per iteration remains almost unchanged, but depending on the scene structure more iterations are needed for sufficient convergence of long connections. However, this can be solved by using a projection scheme similar to one presented in Chapter 6. All experiments were run on a Linux-based Intel Xeon E5520 PC with 24GB RAM and NVidia GTX Titan graphics card. For the

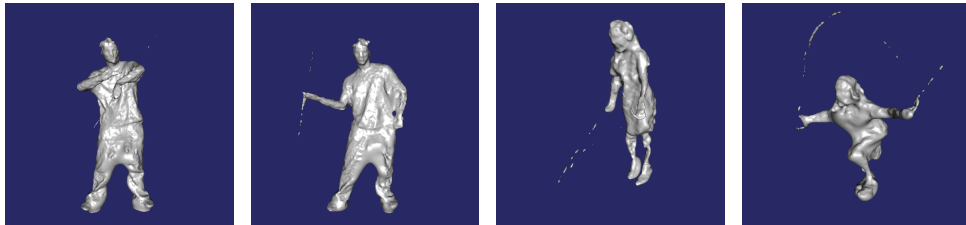
1 of 16 Input Images



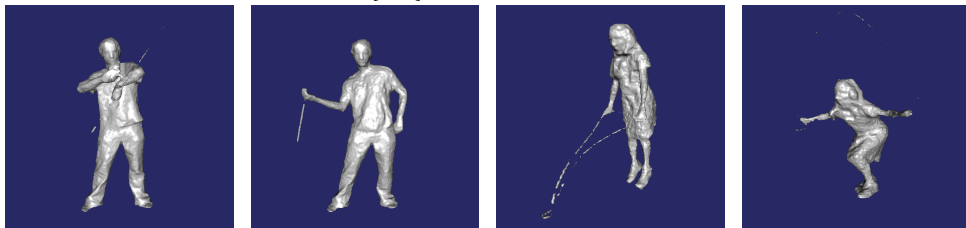
Jancosek and Pajdla [71]



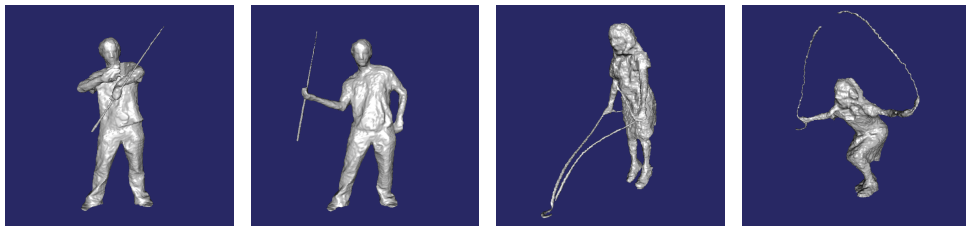
Furukawa et al. (PMVS) [49] + Poisson surface reconstruction [76]



Without Connectivity Constraint [100]



With 1-Connectivity Constraint [130]+[100]



Proposed 2-Connectivity Constraint

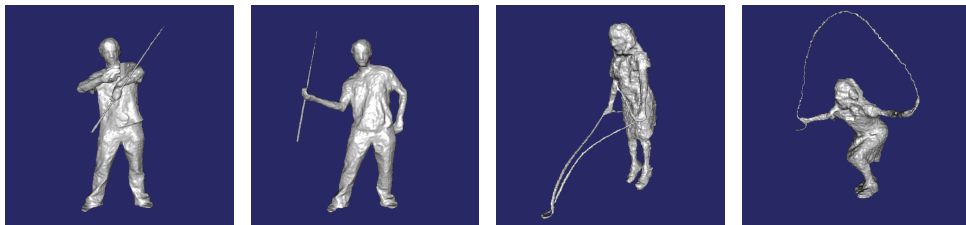


Figure 8.5.: Comparison of the proposed reconstruction method with the state-of-the-art. Existing approaches [49, 71, 76] fail to recover thin structures, in this example the stick and the rope. The 1-connectivity constraint allows to preserve the stick, but for the rope-jump scene, it does not completely preserve the connection of the rope. Our proposed 2-connectivity constraint allows to correctly the connected rope (volume resolution $|V| = 384^3$).

1-connectivity constraint [130] the precomputation time per frame was about 20 sec for computing the tree. For the 2-connectivity constraints the precomputation time was about 1 min for handle and tunnel loop detection, handle segmentation and computation of the tree. The optimization needs about 3 min per frame resulting in a total runtime of about 4 minutes per frame when using the 2-connectivity constraints.

8.6. Conclusion

In this chapter we showed how to include connectivity constraints in spatio-temporal multi-view 3D reconstruction. Because the method used for 3D reconstruction is based on a volumetric labeling by convex optimization, the connectivity constraints for image segmentation described in the first part of this thesis can be directly applied in this new context of 3D reconstruction. Furthermore, we showed how to reformulate the constraints to allow to preserve loops of the object. Therefore, we first showed that the connectivity constraints defined on a tree induce a 1-connectedness of the tree, then we extended these constraints to allow to preserve 2-connectedness of cycles in the constraint graph. By analyzing the visual hull of the scene, we can automatically deduce suitable constraints for a scene in a fully automatic way. We demonstrate in several experiments on real world data that the connectivity constraints significantly improve the reconstruction in the presence of fine elongated structures. To the best of our knowledge, apart from the work of Bleyer et al. [13], a simplification of connectivity for depthmap estimation, this is the first time that connectivity constraints are described in the context of multi-view 3D reconstruction.

9. The Direct Geometry Approach for 3D Reconstruction

In an earlier work [127, 128], we presented a variational approach for online estimation of dense depth maps from a handheld camera. We show that combining the information from multiple images using the robust ℓ^1 norm in the data term geometry reconstructions of high quality and detail can be achieved. Here we review this approach and propose several improvements: First, we show that the resulting optimization problem can be optimized by introducing and dual variable for the data term. In contrast to our previous approach, based on half-quadratic splitting [51], this approach is numerically more stable and the fine tuning of the weight of the quadratic penalty term, the parameter θ in [128], can be avoided.

Second we show that when exactly minimizing the total variation of the depth map, staircasing artifacts occur. However, the previous method that was based on half quadratic splitting never produced these artifacts. We describe the connection between half quadratic splitting and infimal convolution with a quadratic function and propose to avoid the staircasing by using the robust Huber loss instead of the absolute value function in the regularizer. Experimental results confirm that this allows to achieve 3D reconstructions of a quality that is en par with the earlier results, but with improved numerical stability and with less parameters.

Previous results on the research topic presented in this chapter have been published in [128] and [127], and parts of these results were already part of the author's Diploma thesis [125]. As extension to the Diploma thesis here we present the dualization of the data term and the Huber loss as robust penalty function in the regularizer.

9.1. Introduction

9.1.1. Dense Depth Map Estimation from Multiple Images

$$E(h) = \lambda \int_{\Omega_0} \sum_{i \in \mathcal{I}(x)} \left| I_i(\pi(\exp(\hat{\xi}_i) X(x, h))) - I_0(\pi(x)) \right| dx + \int_{\Omega_0} \|\nabla h\| dx, \quad (9.1)$$

data term $I_i(\pi(\exp(\hat{\xi}_i) X(x, h))) - I_0(\pi(x))$

In the following we will use the simplified notation $I_i(x, h)$ for $I_i(\pi(\exp(\hat{\xi}_i) X(x, h)))$.

We begin with a linearization of $I_i(x, h)$ by using the first order Taylor expansion, i.e.

$$I_i(x, h) = I_i(x, h_0) + (h - h_0) \frac{d}{dh} I_i(x, h) \Big|_{h_0} \quad (9.2)$$

where h_0 is a given depth map. The derivative $\frac{d}{dh} I_i(x, h)$ can be considered as a directional derivative in direction of a differential vector on the image plane that results from a variation of h . It can be expressed as the scalar product of the gradient of $I_i(x, h)$ with this differential vector, i.e.

$$\frac{d}{dh} I_i(x, h) = \nabla I_i(x, h) \cdot \frac{d}{dh} \pi(\exp(\hat{\xi}) X(x, h)). \quad (9.3)$$

The differential vector mentioned above needs to be calculated with respect to the chosen camera model.

Using the linear approximation for $I_i(x, h)$ and by reordering the integrals the energy functional now reads

$$E(h) = \int_{\Omega_0} \left\{ \lambda \underbrace{|I_i(x, h_0) + (h - h_0) \frac{d}{dh} I_i(x, h)|_{h_0} - I_0(x)|}_{\rho_i(x, h_0, h)} + \|\nabla h\| \right\} dx. \quad (9.4)$$

Though this energy functional is much simpler than the original functional (Eq. (9.1)), the task of minimizing it is still difficult, because both the regularization term and the data term are not continuously differentiable.

We introduce an auxiliary function u that decouples the data term and the regularizer, leading to the following convex approximation of Eq. (9.11):

$$E_\theta = \int_{\Omega} \left\{ \|\nabla u\| + \frac{1}{2\theta} (u - h)^2 + \lambda |\rho_i(h)| \right\} dx, \quad (9.5)$$

where θ is a small constant and $\rho_i(h)$ denotes the current residual of the data term (by omitting the dependency on h_0 and x). It is immediate to see that for $\theta \rightarrow 0$ the minimization of the above functional results in both h and u being a close approximation of each other.

This minimization problem can be solved efficiently in real-time by minimizing the data term with a simple thresholding scheme and using a primal dual algorithm for the minimization of the ROF energy [22].

9.1.2. Extension to Multiple Images

Let us now consider the case when multiple input images are given. In the previous section we formulate our energy model for the classical stereo task in case of two images. Compared to previous approaches that employ the epipolar constraint by using the fundamental matrix the main difference is that here we formulate the data term relative to the coordinate system of one specific view and use the perspective projection to map this coordinate system to the second camera frame. This makes it easy to incorporate the information from other views by simply adding up their data terms. We propose the following energy functional to robustly estimate a depth map from multiple images

$$E(h) = \lambda \int_{\Omega} \sum_{i \in \mathcal{I}(x)} |\rho_i(x, h)| dx + \int_{\Omega} \|\nabla h\| dx \quad (9.6)$$

where $\mathcal{I}(x)$ contains the indices of all images for which the perspective projection $\pi(\exp(\hat{\xi}_i) \cdot X(x, h))$ is inside the image boundaries. With $\rho_i(x, h)$ we denote the residual of the linearized data term for image I_i

$$\rho_i(x, h) = I_i(x, h_0) + (h - h_0) I_i^h(x) - I_0(x), \quad (9.7)$$

where $I_i^h(x)$ is a simplified notation for the derivative $\frac{d}{dh} I_i(x, h)|_{h_0}$.

By using the above functional we should expect two benefits. First of all algorithms using only two images are not able to estimate disparity information in regions that are occluded in the other view or simply outside of its image borders. The use of images from several different views should help in these cases because information from images where the object is not occluded can be used. The use of the L_1 -norm in the data terms allows an increased robustness towards outliers in cases where objects are occluded. The second benefit of using multiple images is the increased signal to noise ratio that provides much better results when the input images are affected by noise.

9.1.3. Half Quadratic Splitting

We decouple the smoothness and data term by introducing an auxiliary function u and get the following convex approximation of Eq. (9.6):

$$E_\theta = \int_{\Omega} \left\{ \|\nabla u\| + \frac{1}{2\theta}(u-h)^2 + \lambda \sum_{i \in \mathcal{I}(x)} |\rho_i(x, h)| \right\} dx, \quad (9.8)$$

The above functional is convex so an alternating descent scheme can be applied to find the minimizer of E_θ :

1. For h being fixed, solve

$$\min_u \int_{\Omega} \left\{ \|\nabla u\| + \frac{1}{2\theta}(u-h)^2 \right\} dx \quad (9.9)$$

This is the ROF energy for image denoising [22, 112].

2. For u being fixed, solve

$$\min_h \int_{\Omega} \left\{ \frac{1}{2\theta}(u-h)^2 + \lambda \sum_{i \in \mathcal{I}(x)} |\rho_i(x, h)| \right\} dx \quad (9.10)$$

This minimization problem can be solved point-wise.

9.1.4. Dualization of the Data Term

Instead of using the half quadratic splitting method discussed above, here we propose to introduce a dual vector $q \in \mathbb{R}^m$, where m is the number of additional views, with a component for each summand of (9.6).

$$E(h) = \int_{\Omega_0} \left\{ \lambda \underbrace{|I_i(x, h_0) + (h - h_0) \frac{d}{dh} I_i(x, h)|_{h_0} - I_0(x)}_{\rho_i(x, h)} + \|\nabla h\| \right\} dx. \quad (9.11)$$

$$\rho_i(x, h) = I_i(x, h_0) + (h - h_0) \frac{d}{dh} I_i(x, h)|_{h_0} - I_0(x) \quad (9.12)$$

$\rho_i(x, h) = a_i h - b_i$, with $a_i := I_i^h(x)$ and $b_i := a_i h_0 - I_i(x, h_0) + I_0(x)$.

$$E(h) = \lambda \int_{\Omega} \sum_{i \in \mathcal{I}(x)} \{|a_i h - b_i|\} dx + \int_{\Omega} \|\nabla h\| dx \quad (9.13)$$

We define the column vector $a := a_1, \dots, a_m$, with $m := |\mathcal{I}(x)|$, and the column vector $b := b_1, \dots, b_m$ and write above energy functional as

$$E(h) = \lambda \int_{\Omega} \|a h - b\|_1 dx + \int_{\Omega} \|\nabla h\| dx, \quad (9.14)$$

with $\|\cdot\|_1$ we denote the ℓ^1 norm, the sum of the absolute value of each component of a vector.

We introduce dual a variable $p \in \mathbb{R}^m$, one p_i for every image data term with $|p_i| \leq 1$, thus $\|p\|_\infty \leq 1$, and a dual variable $q \in \mathbb{R}^2$ for the total variation regularizer and get the saddle point problem

$$\min_{h:\Omega \rightarrow \mathbb{R}} \max_{\substack{\|p\|_\infty \leq 1, \\ \|q\| \leq 1}} \langle p, \lambda a h - \lambda b \rangle + \langle q, \nabla h \rangle \quad (9.15)$$

with the ℓ^∞ norm for the dual variable of the data term and the ℓ^2 norm for the dual variable of the total variation regularizer.

$$\min_{h:\Omega \rightarrow \mathbb{R}} \max_{\substack{\|p\|_\infty \leq 1, \\ \|q\| \leq 1}} \langle p, \lambda a h \rangle + \langle q, \nabla h \rangle - \lambda b p \quad (9.16)$$

Now we have brought the optimization problem into a standard form that allows to apply Algorithm 1 of [25]. The update equations are defined by the **prox**-operator

$$p^{k+1} = \mathbf{prox}_{\sigma F^*} \left(p^k + \sigma \lambda a \bar{h}^k \right) - \lambda b p \quad (9.17)$$

$$q^{k+1} = \mathbf{prox}_{\nu H^*} \left(q^k + \nu \nabla \bar{h}^k \right) \quad (9.18)$$

$$h^{k+1} = \mathbf{prox}_{\tau G} \left(h^k + \tau \operatorname{div} q^{k+1} - \tau \lambda a^T p^{k+1} \right) \quad (9.19)$$

$$\bar{h}^{k+1} = h^{k+1} + \theta \left(h^{k+1} - h^k \right), \quad (9.20)$$

where we choose the step sizes τ , σ , and ν following the diagonal precondition method described in [105].

We evaluate the **prox**-operators with $F^*(p) = \lambda b p$, $H^*(q) = \delta_{\|q\| \leq 1}$, and $G(h) = 0$ and get as update equations

$$p^{k+1} = p^k + \sigma \lambda \left(a \bar{h}^k - b \right) \quad (9.21)$$

$$q^{k+1} = \pi_{\|q\| \leq 1} \left(q^k + \nu \nabla \bar{h}^k \right) \quad (9.22)$$

$$h^{k+1} = h^k + \tau \operatorname{div} q^{k+1} - \tau \lambda a^T p^{k+1} \quad (9.23)$$

$$\bar{h}^{k+1} = h^{k+1} + \theta \left(h^{k+1} - h^k \right). \quad (9.24)$$

9.1.5. Huber loss

We discovered in experiments, that exactly minimizing the total variation of the depth map by using the dualized data term results in stair-casing artifacts, a well know effect when minimizing the total variation of a function. In our former approach, where we where using a half quadratic splitting approach these effects didn't appear, a possible explanation for this is that the half quadratic splitting results in a smoothing of the functional, comparable to an infimal convolution with a quadratic function.

An exact infimal convolution of the absolute value function with a quadratic function is realized by the Huber loss [67]. It allows to avoid the stair casing artifacts, by using the Huber loss function instead of the absolute value function on the depth map gradient. The Huber loss is a robust estimator defined as

$$\|x\|_\varepsilon = \begin{cases} \frac{x^2}{2\varepsilon} & \text{for } \|x\| \leq \varepsilon \\ \|x\| - \frac{\varepsilon}{2} & \text{else .} \end{cases} \quad (9.25)$$

The functional for depth map estimation from multiple images using the Huber loss function for the regularizer becomes

$$E(h) = \lambda \int_{\Omega} \|a h - b\|_1 \, dx + \int_{\Omega} \|\nabla h\|_{\varepsilon} \, dx, \quad (9.26)$$

with the ℓ^1 norm $\|\cdot\|_1$ that sums up over the absolute values of the data terms of multiple images and the Huber norm $\|\cdot\|_{\varepsilon}$.

To minimize the Huber norm in the primal dual optimization framework, we use the conjugate of the Huber norm. With $H(x) = \|x\|_{\varepsilon}$ the conjugate $H^*(y)$ is

$$H^*(y) = \frac{\varepsilon}{2} \|y\|^2 + \delta_{\|\cdot\| \leq 1}(y). \quad (9.27)$$

The **prox**-operator for the Huber norm is

$$\mathbf{prox}_{\nu H^*}(\tilde{y}) = \arg \min_y \frac{1}{2\nu} \|y - \tilde{y}\|_2^2 + \frac{\varepsilon}{2} \|y\|_2^2 + \delta_{\|\cdot\| \leq 1}(y) \quad (9.28)$$

$$= \pi_{\|\cdot\| \leq 1} \left(\frac{1}{1 + \nu \varepsilon} \tilde{y} \right). \quad (9.29)$$

Thus the update in the dual variable q is given by the projection

$$q^{k+1} = \mathbf{prox}_{\nu H^*} \left(q^k + \nu \nabla \bar{h}^k \right) \quad (9.30)$$

$$= \pi_{\|\cdot\| \leq 1} \left(\frac{1}{1 + \nu \varepsilon} \left(q^k + \nu \nabla \bar{h}^k \right) \right). \quad (9.31)$$

An experimental comparison of the Huber loss applied to the regularizer with the exact total variation is given in the experimental section below.

9.2. Implementation

9.3. Experimental Results

We now compare our new dual variable approach for the data term with our previous approach that used half quadratic splitting. In Fig. 9.1 we present reconstruction results of both approaches to allow a direct qualitative comparison.

Fig. 9.2 depicts the parameter space of the Huber loss regularized approach for different values of the weight of the data term λ and the Huber loss parameter ε .

Runtime Comparison:

9.4. Conclusion

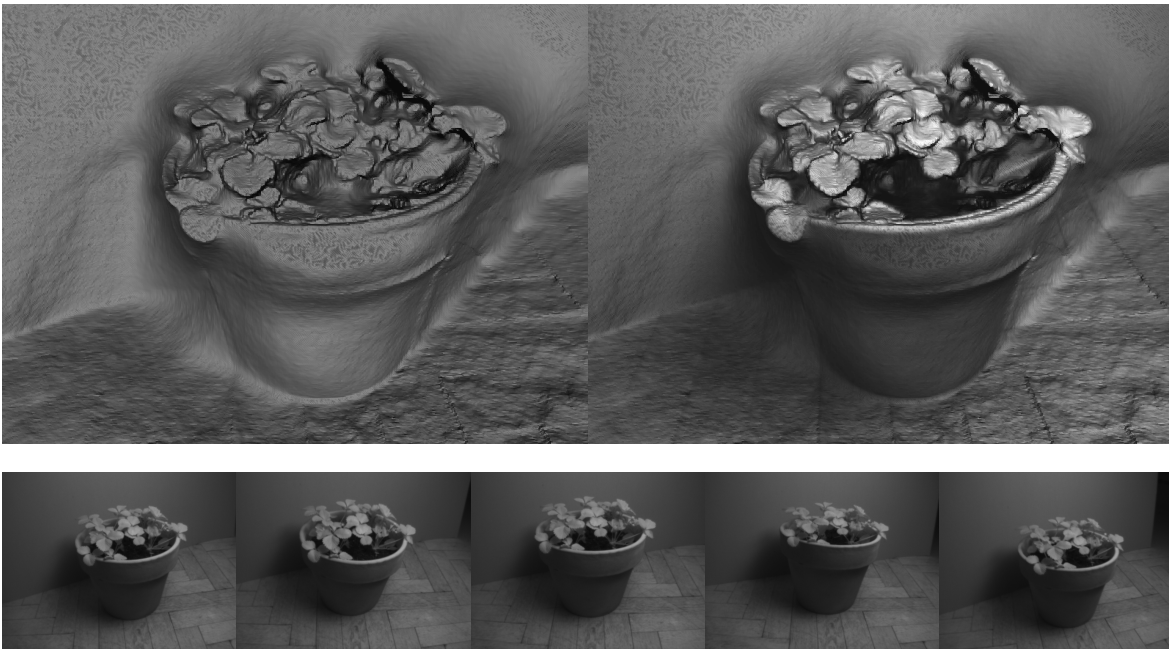


Figure 9.1.

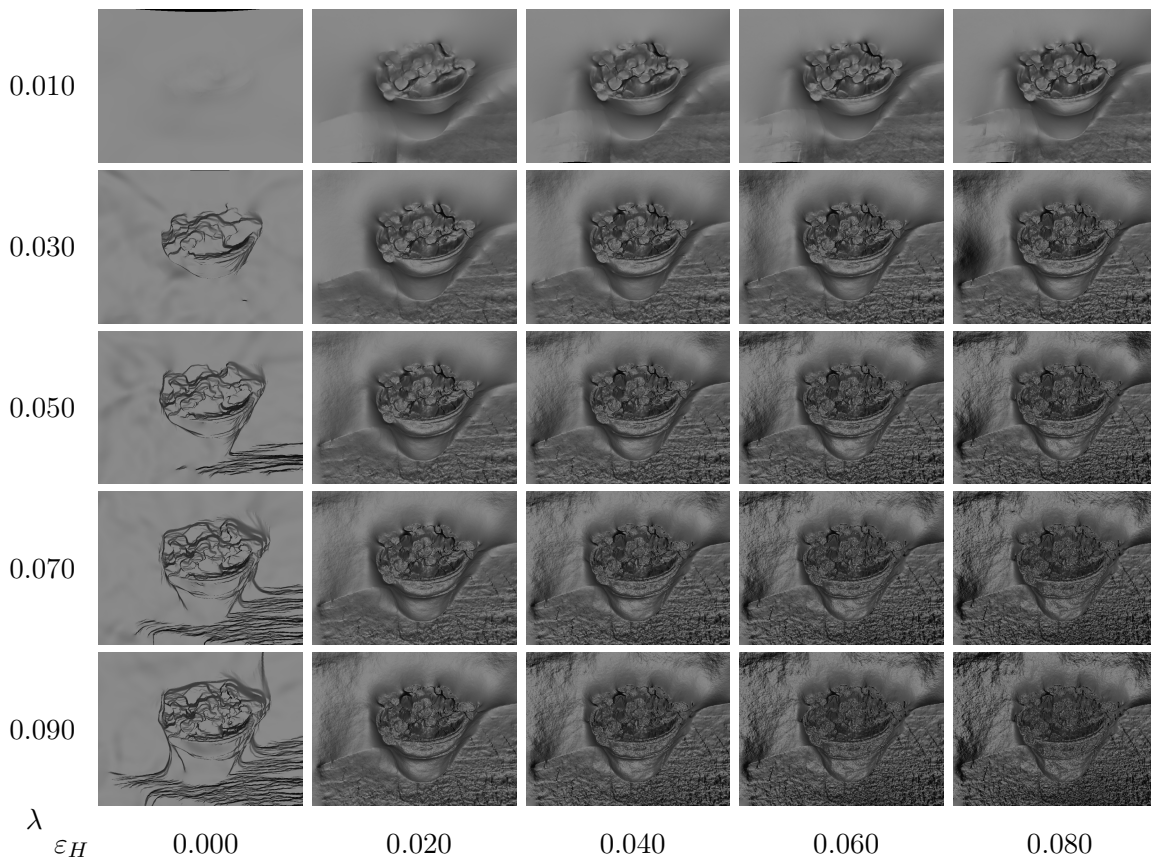


Figure 9.2.: Reconstruction results with Huber loss regularization using the dual variable approach for the data term. Shown are reconstruction results for different parameters for the weight of the data term λ and the Huber loss parameter ε_H .

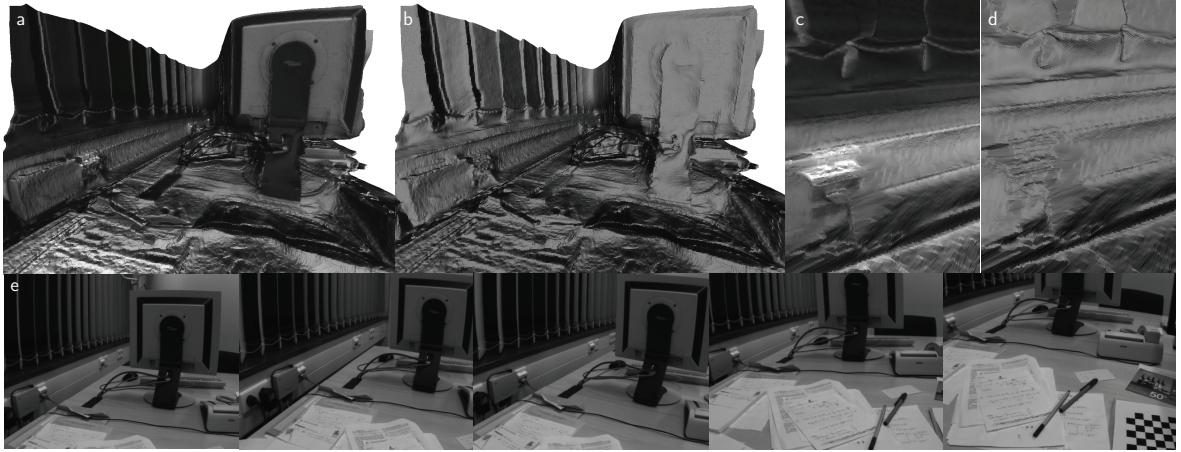
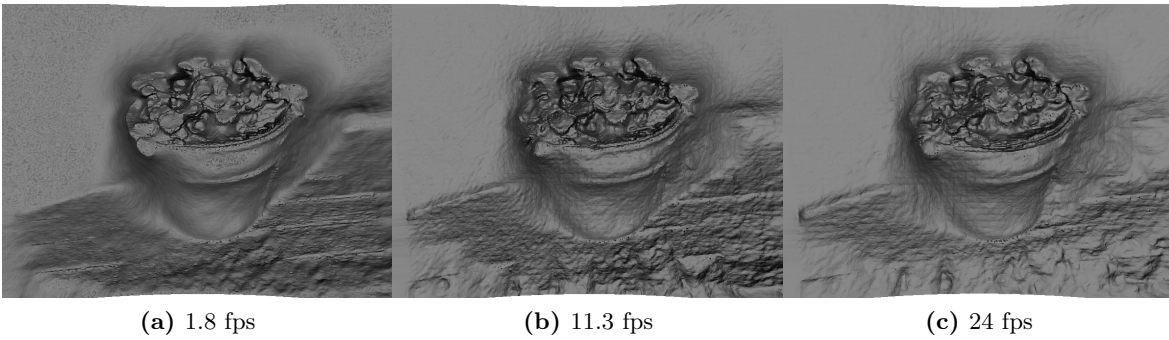


Figure 9.3.: Textured (a,c) and untextured geometry (b,d). Note the accurate reconstruction of small-scale details like the network socket and cords. (e) Images

Table 9.1.: Parameter settings for different frame rates

Quality Setting	High	Medium	Low
Pyramid Levels	24	10	7
Pyramid Scale-Factor	0.94	0.8	0.7
Iterations per Level	120	70	70
Internal Iterations	1	4	4
Frames per Second	1.8	11.3	24



(a) 1.8 fps

(b) 11.3 fps

(c) 24 fps

Figure 9.4.

10. 3D Tracking from Raw ToF data

The 3D reconstruction methods described so far are *image based* reconstruction methods, which reconstruct the geometry from images of traditional video and photo cameras. Because digital colour images are encoded using a red, a green and a blue colour channel, these image modalities are also referred to as *RGB* data. In this chapter, we will present a novel approach for object tracking for a special type of sensor, that in addition to the red, green, and blue colour channel also measures the depth of the scene for every pixel. These sensors are called depth cameras or *RGBD* cameras, to emphasise that in addition to a red, green and blue channel these sensors also measure the depth of the scene for every pixel. This additional information greatly improves segmentation and tracking of rigid, articulated, and even deformable 3D objects in real-time. However, these depth cameras typically have a limited temporal resolution (frame-rate) that restricts the accuracy and robustness of tracking, especially for fast or unpredictable motion. In the following, we show how to perform model-based object tracking at an order of magnitude higher frame-rate. This is achieved through simple modifications to an off-the-shelf depth camera. We focus on phase-based time-of-flight (ToF) sensing, which reconstructs each low frame-rate depth image from a set of short exposure ‘raw’ infrared captures. These raw captures are taken in quick succession near the beginning of each depth frame, and differ in the modulation of their active illumination. Instead of computing a depth frame for this set of raw captures, we propose a model-based tracking approach that allows to infer the depth of the object for each measurement. We make two main contributions. First, we detail how to perform model-based tracking against these raw captures. Second, we show that by reprogramming the camera to space the raw captures uniformly in time, we obtain a 10x higher frame-rate, and thereby improve the ability to track fast-moving objects. Our approach has the side-benefit of avoiding the depth reconstruction step that may be costly for mobile applications.

The results presented in this chapter previously appeared in [129]. The research was conducted in cooperation with Microsoft Research Cambridge and funded by the Microsoft Research internship program.

10.1. Introduction

Tracking objects that move is one of the fundamental research topics in computer vision, that enables higher-level reasoning about the world and allows to build systems that interact with the environment. Realtime tracking and especially tracking of the articulated human body enable applications in human computer interaction which allow a natural interface between a computer system and the user. One way to enable interaction is by tracking the full body pose [121], more recent approaches allow realtime tracking of the articulated hand [120]. These methods allow the user to interact with the computer in an intuitive way without the need of an additional input device, e.g. by pointing with the finger at objects on a screen or even in the scene itself [136].

However, the visual object tracking problem itself bears several key challenges that limit the accuracy of current systems. One of the main difficulties in tracking is the change of the object’s appearance due to object translation, rotation, deformation, and lighting variation.

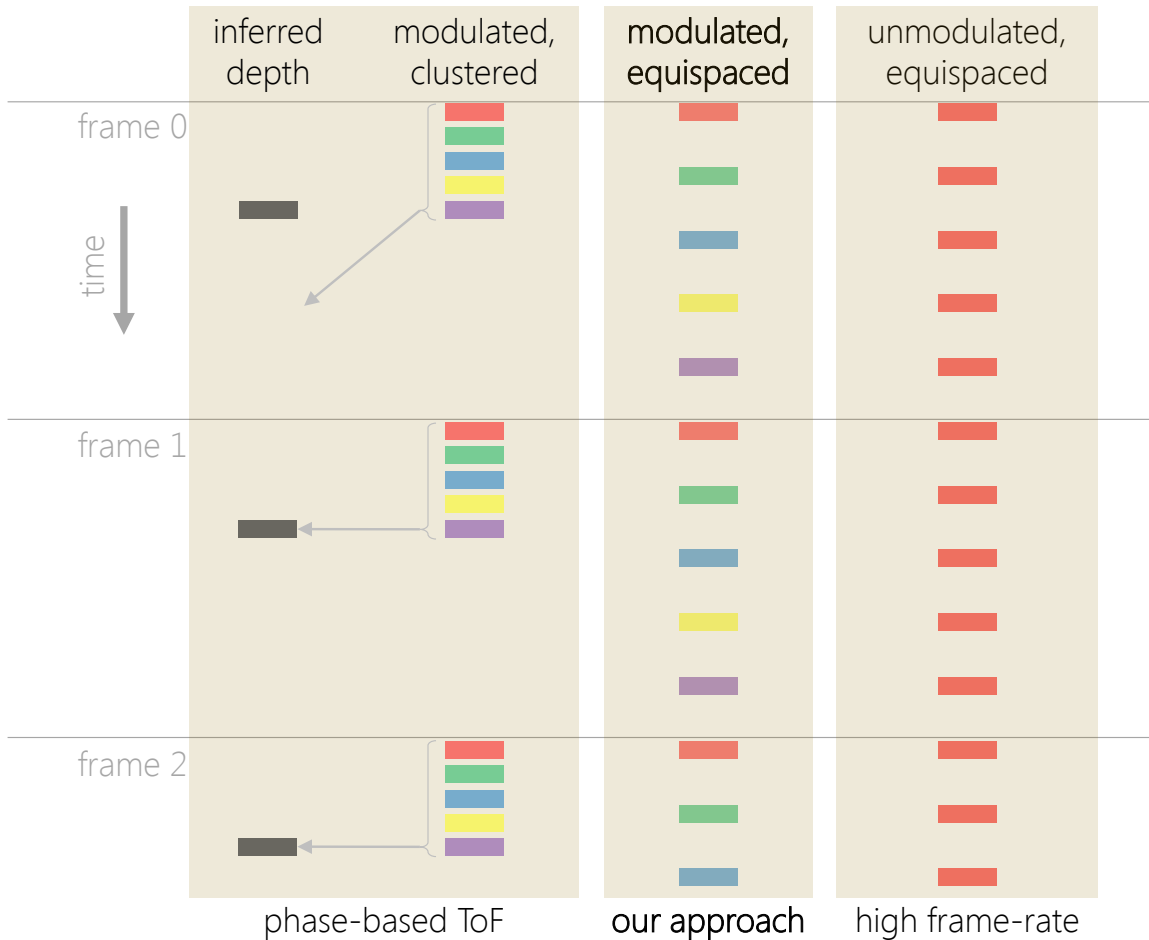


Figure 10.1.: Overview. Phase-based time-of-flight (ToF) sensors infer a low frame-rate stream of depth images from a set of short-exposure ‘raw’ captures that are clustered closely in time to reduce motion artefacts. For illustration purposes, we use five colours here to indicate different frequencies and phase modulations of the illuminant and sensor; see text. For the application of model-based tracking, we propose to forego the depth reconstruction step, and instead track directly from *equispaced* raw captures, resulting in a signal at much higher frame-rate.

Further difficulty stems from object occlusion, and objects leaving the viewing volume. Another key challenge is the simultaneous tracking of multiple objects whose number may vary over time, and tracking fast moving objects. These challenges make it hard to build robust tracking algorithms that can achieve human level accuracy and robustness.

General purpose tracking approaches try to address these challenges through adaptive non-parametric methods, as in mean-shift tracking [29], and by online learning of a flexible object representation [110]. For multiple objects the integration of observations over a longer time frame is required to disambiguate between different objects with similar appearance [5, 98]. Despite significant progress in the last decade, general purpose tracking remains a challenging problem, as illustrated by the recent VOT 2014 challenge [83]. For other general surveys on object tracking, see [141, 142].

In this work we do not address the general purpose tracking problem but instead focus on accurately tracking *fast-moving* rigid objects in three dimensional space from a single viewpoint. One successful strategy to tracking fast-moving objects is to use custom hardware consisting of high speed sensors and processing units [94]. Alternatively to increasing the resolution in the time domain, one could increase the resolution of the captured frames in order to improve

angular accuracy, a point made in an extensive synthetic SLAM study [64]. A third option is to use a setup of multiple cameras [84].

However, increasing the frame rate via a high frame rate camera or even a custom built imaging sensor is expensive and requires control over the imaging setup. Here, we instead show how a single off-the-shelf time-of-flight (ToF) sensor, the Kinect V2 [7], can be re-purposed for high speed object tracking. Furthermore, we validate in experiments that our approach allows to accurately infer the depth of the object. The method should readily extend to other commercially available ToF sensors.

Our approach is based on the working principles of phase-based ToF sensors, as illustrated in Fig. 10.1. The Kinect camera captures a set of actively illuminated infrared frames (the coloured bars in the left column) and infers from these a single depth image frame (each grey bar). In order to provide a 30Hz depth signal, the Kinect sensor internally captures infrared frames at an average frequency of 300Hz. The frames are captured in a short burst at the beginning of the exposure cycle, to minimise movement in the scene during the capture period to allow a consistent depth reconstruction of the scene. Each frame is captured under one of three frequencies of infrared illumination, modulated by one of three phases resulting in nine different combinations of frequency and phase shift.

Contributions. Our main contribution is to show that this phase-based ToF sensor can be re-purposed for tracking. First, we show that model-based tracking can be performed against the raw infrared ToF captures. This allows to directly track the object in the infrared captures, and thus does not require to first reconstruct a depth image, avoiding the potentially computational expensive depth reconstruction process. Second, instead of capturing the nine frames in a burst at the begin of each depth frame cycle, we space the frames out equally in time at approximately 300Hz. While the burst mode of the camera was specifically designed to motion artefacts during depth reconstruction, spacing out the frames evenly allows for much more stable tracking of fast moving objects (see Fig. 10.2).

Here, we present the results of an initial study of the above ideas based on a model-based tracking framework, that employs a probabilistic state space model with a standard temporal prior. While this probabilistic tracking approach is a standard approach for object tracking, our contribution is a generative model of the observation likelihood, that allows us to compare the observation to a rendered simulation of the raw ToF captures, given a 3D model of the object. We validate in experiments that we can accurately track a fast moving rigid object in the regime where the depth reconstruction fails. In contrast to a depth-based tracking method, that would fail completely in this case, we do not need to first infer a depth image but instead directly track using the raw ToF frames. Compared to constant illumination high frame-rate tracking, the benefit of our solution is that we not only achieve a high frame rate but that each frame contains a distinct active illumination response that contains additional depth information.

10.2. Background

In this section we present some background material on phase-based time-of-flight sensors and model-based tracking that will be necessary to explain our main contributions in the subsequent sections.

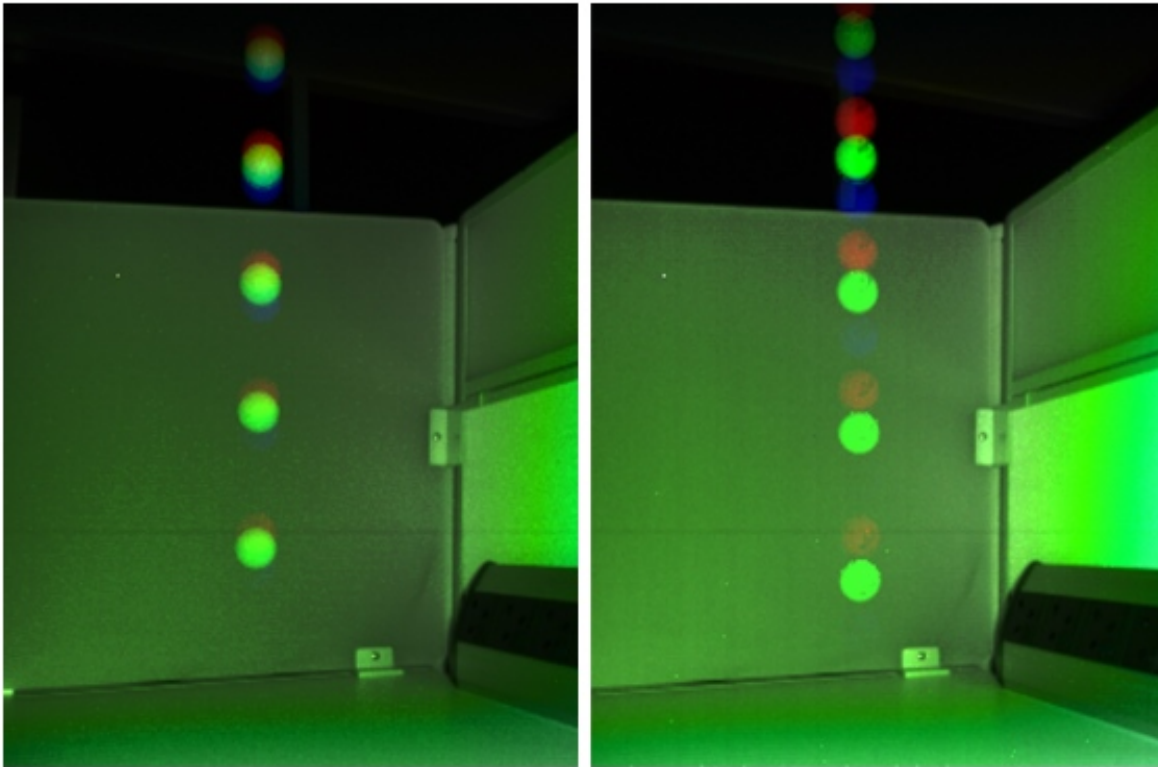


Figure 10.2.: Time-of-flight captures and fast motion. A table tennis ball is dropped, and three raw captures are superimposed for visualisation using the red, green, and blue colour channels. **Left:** phase-based ToF clusters its captures temporally to minimise motion artefacts during depth reconstruction. **Right:** we propose reprogramming the ToF capture profiles to more equally space the captures. We demonstrate how to exploit this extra temporal information by tracking objects without previous reconstruction of a depth image.

10.2.1. Phase-modulation time-of-flight

Modern ToF cameras measure the depth of the scene for every pixel based on phase modulation: a modulated light source emits a sinusoidal light signal modulated at a specific frequency, and a special sensor images the light’s reflection, gain-modulated at the same frequency [85, 118]. During the frame exposure, the sensor integrates over a large number of oscillation periods and the recorded image intensities contain information about the phase shift between emitted light and incoming light. This phase shift depends on the time it took for the emitted light to travel from the light source to the object, where it is reflected, back to the sensor. Usually, the phase shift wraps around several times for depth ranges present in typical scenes. Instead of recording only a single frame at a single frequency, modern cameras therefore record a sequence of frames at multiple modulation frequencies and phase shifts, to solve for this ambiguity of possible depth values that lead to the same single phase shift. The whole set of recorded frames allows a unique disambiguation of surface distances based on phase unwrapping algorithms [62, 65, 89]. To allow for a stable phase unwrapping, it is important that the depth of the object does not change between the measurements in each set of frames. Thus the frames are recorded in a short time span at the beginning of each depth reconstruction cycle, to reduce the motion of the object between different captures. This standard operation mode is illustrated in the leftmost column in Fig. 10.1.

Formally, for each pixel we obtain a sequence of nine measurements R_1, \dots, R_9 (3 frequencies \times 3 phases) via

$$R_i = \frac{\rho}{d^2} S_i(d) + \epsilon_i, \quad (10.1)$$

where $d > 0$ is the depth of the imaged surface at that pixel and $\rho > 0$ is the surface *albedo*. The ideal responses are dependent on modulation frequency and phase delay and are given by an idealised calibrated response curve [7],

$$S_i : [d_{\min}, d_{\max}] \rightarrow \{-I_{\max}, \dots, -1, 0, 1, \dots, I_{\max}\},$$

where d_{\min} and d_{\max} is the range of valid depths and the range of S_i are signed image intensities. For the noise model ϵ_i we simply assume zero mean Gaussian noise of a fixed standard deviation.¹

The standard approach to reconstruct a depth image is to use a depth reconstruction engine which infers the depth from the nine measurements as

$$\hat{d} = f(R_1, \dots, R_9). \quad (10.2)$$

Our system instead uses the raw measurements as detailed below, without first needing to infer a depth image.

10.2.2. Model-based tracking

We focus on the task of model-based object tracking [81, 140], using a generative observation model to relate the tracked position to the observations over time. To provide stable tracking we use a temporal model and follow the influential work by Isard and Blake [69] based on *particle filtering* [58] in state space models. In a state space model we need to specify a state space and both a probabilistic transition model and a probabilistic observation model [38]. We use a state vector

$$X_t = (x_t, v_t), \quad (10.3)$$

encoding a 3D world location $x_t \in \mathbb{R}^3$ and a 3D velocity vector $v_t \in \mathbb{R}^3$. For general rigid objects we could include rotation parameters, i.e. $X_t = (x_t, r_t, v_t)$, but, to demonstrate our key contributions in a setup as simple as possible, we will only use a spherical object (a table tennis ball) in the experiments, which does not require rotational parameters. While not currently demonstrated, our approach is general and our results should extend to more complex rigid and non-rigid objects that have higher-dimensional state spaces. However, one problem that might arise with a higher dimensional state space is that the number of required samples increases exponentially with the number of dimensions.

The stochastic transition model is specified via a distribution $P(X_{t+1}|X_t)$ that encodes the assumed laws of motion (see below for further details). The observation model is specified via an analysis-by-synthesis approach: observation Y_t corresponds to an entire raw ToF frame, and we compute an observation likelihood by comparing the observed image to a synthetic rendering of the scene (see below for further details).

Together, the transition and observation model give a joint distribution over the entire sequence of states $X_{1:T}$ and observations $Y_{1:T}$ as

$$P(X_{1:T}, Y_{1:T}) = \prod_{t=1}^T P(X_t|X_{t-1}) P(Y_t|X_t), \quad (10.4)$$

where $P(X_1|X_0) = P(X_1)$ is assumed to be given.

Once the transition and observation models are defined, inference about the object's state for given observations can be done either by *filtering* or by *smoothing* [38]. In filtering the past

¹Due to the way the Kinect sensor operates [7] the right noise model would be an intensity-dependent Skellam noise, but for simplicity we adopt the Gaussian approach.

observations are used to infer the current believed distribution over positions and velocities. Because filtering allows to infer a distribution about the current object state it is suitable for interactive tracking. Filtering provides as output at each time step t the marginal distribution $P(X_t|Y_{1:t})$ over the state X_t . In smoothing, we instead use observations both from the past and the future, i.e. we perform inference offline after the entire sequence $Y_{1:T}$ of T frames has been observed. This is known to significantly improve tracking accuracy [70] as the inference result $P(X_{1:T}|Y_{1:T})$ now integrates all observations coherently. A strategy between filtering and smoothing is to delay inference by a small number of K frames and perform smoothing on a truncated sequence, i.e. to infer the partial sequence $P(X_{(t-K+1):t}|Y_{1:t})$. This is known as *fixed-lag smoothing* and offers an adjustable tradeoff between both strategies [36]: for $K = 1$ we recover filtering, and for $K = T$ we recover smoothing. This can allow for improved accuracy of interactive tracking at the expense of introducing a fixed latency.

We use standard inference methods: for filtering we use the bootstrap particle filter [58] and for smoothing we use the forward-filtering-backward-realisation method [21, 53].

10.3. Method

We now describe our model for tracking a moving object directly in the raw ToF captures. While the motion model is standard, the observation model for raw ToF captures is a novel contribution.

10.3.1. Motion model $P(X_{t+1}|X_t)$

Using the state representation (10.3) we model the motion linearly via a multivariate Gaussian distribution,

$$P(X_{t+1}|X_t) \sim \mathcal{N} \left(\begin{bmatrix} x_t + \Delta v_t \\ v_t \end{bmatrix}, \begin{bmatrix} \sigma_x^2 I_3 & 0 \\ 0 & \sigma_v^2 I_3 \end{bmatrix} \right), \quad (10.5)$$

where Δ is the difference in time stamps between the frame captured at step $t + 1$ and step t , and $\sigma_x > 0$ and $\sigma_v > 0$ are the noise terms for the position and velocity vectors. Intuitively we can understand the model (10.5) as simply predicting the position x_{t+1} to be the linear extrapolation of the current position x_t using the current estimate of the velocity v_t . The velocity is assumed to remain constant, i.e. $v_{t+1} = v_t$, which is a common simplifying assumption. Other motion models are of course possible. Here we chose (10.5) as probably the simplest possible model that could help demonstrate our main contribution: an observation model that allows for raw ToF-based tracking.

10.3.2. Observation model $P(Y_t|X_t)$ for raw ToF

This section describes one of our contributions: how to create an observation model which removes the dependence on a ToF depth reconstruction and instead compute observation likelihoods directly against the raw ToF captures.

The observation model is specified as $P(Y_t|X_t)$, where Y_t is an observed raw ToF frame of size 512×424 together with its timestamp, and X_t is an object hypothesis. The raw ToF frame takes the form of raw phase-encoded responses (10.1), one for each sensor element (sensel). Let us denote by $\bar{R}_i(u)$ the observed response at sensel location u and shutter type i , a specific configuration of frequency and phase shift of the active illumination, where for a single frame only one such shutter type i is possible. The observed information is then $Y_t = (i, \bar{\mathbf{R}}_i)$, where i is the shutter type, and $\bar{\mathbf{R}}_i$ is the vector of all response at all sensels. The shutter type i

changes in a fixed cyclic order on the camera device, and thus does not need to be modelled probabilistically. Therefore we only need to specify a model for the frame $\bar{\mathbf{R}}_i$.

Our probabilistic model for $\bar{\mathbf{R}}_i$ is based on a 3D rendering approach: given the object hypothesis X_t we first render the distance $d(u)$ and reflectivity $\rho(u)$ for every sensel ray at location u . The reflectivity is computed via a Blinn-Phong model [14] whose coefficients we fit empirically to the object appearance prior to tracking in an offline calibration step. From $d(u)$ and $\rho(u)$ and from the known shutter type i we use equation (10.1) to compute the expected ideal object response $R_i^{\text{obj}}(u)$ for each location u . The whole pipeline of rendering the depth, reflectivity and computing the ideal response is implemented efficiently on the GPU using the HLSL shader language.

The ideal synthesised response $R_i(u)$ is compared to the observed response $\bar{R}_i(u)$ to compute a likelihood. Here, a complication arises: the rendering model expects a non-zero response only at object locations, and does not contain a model for the background. One possibility is to compare only sensels at the assumed object location provided by X_t , however, this does not provide a valid distribution $P(Y_t|X_t)$ for the entire observed frame.

To overcome this difficulty, we explicitly model the background. This is commonly done for RGB images via mixture models, as in the seminal work [48, 124, 144]. Here, for simplicity we assume a static camera and static scene, we use a simpler Gaussian model as described below. For every shutter type i and every location u we capture a few seconds of static background video and compute the empirical mean $\hat{\mu}_i(u)$ to the observed responses $\bar{R}_i(u)$. We then assume the background to be distributed as

$$R_i^{\text{bg}}(u) \sim \mathcal{N}(\hat{\mu}_i(u), \sigma_{\text{bg}}^2), \quad (10.6)$$

where σ_{bg} is a global parameter in raw ToF units, typically in the range of a few hundred units.

The full model $P(Y_t|X_t)$ is now the composition between the background responses R_i^{bg} and the object (foreground) responses R_i^{obj} . For a given object hypothesis X_t the renderer can perform this composition easily as it computes a mask of object locations during rendering. Let us denote the mask by $M(u) \in \{0, 1\}$ where $M(u) = 1$ denotes a location where the object hypothesis causes the location u to be occupied. We obtain the full model as

$$R_i(u) \sim \begin{cases} \mathcal{N}(R_i^{\text{obj}}(u), \sigma_{\text{obj}}^2), & \text{if } M(u) = 1, \\ \mathcal{N}(\hat{\mu}_i(u), \sigma_{\text{bg}}^2), & \text{otherwise.} \end{cases} \quad (10.7)$$

Here the additional parameter σ_{obj} is a constant specifying the assumed noise in the object responses. From (10.7) and assuming independent pixels we see that the full raw ToF frame is modelled by a product of Gaussian distributions, hence the full model is a multivariate Gaussian. Therefore we compute the log-likelihood function $\log P(Y_t|X_t)$ as

$$\begin{aligned} \log P(Y_t|X_t) = & - \sum_{u:M(u)=1} \left[\frac{(\bar{R}_i(u) - R_{\text{obj}}(u))^2}{2\sigma_{\text{obj}}^2} + \log \sigma_{\text{obj}} \right] \\ & - \sum_{u:M(u)=0} \left[\frac{(\bar{R}_i(u) - \hat{\mu}_i(u))^2}{2\sigma_{\text{bg}}^2} + \log \sigma_{\text{bg}} \right] + C, \end{aligned} \quad (10.8)$$

where $C = -\frac{n}{2} \log(2\pi)$ is a constant independent of the observation with $n = 512 \cdot 424$ denoting the number of sensels per frame. Because C is independent of the observation it can be omitted.

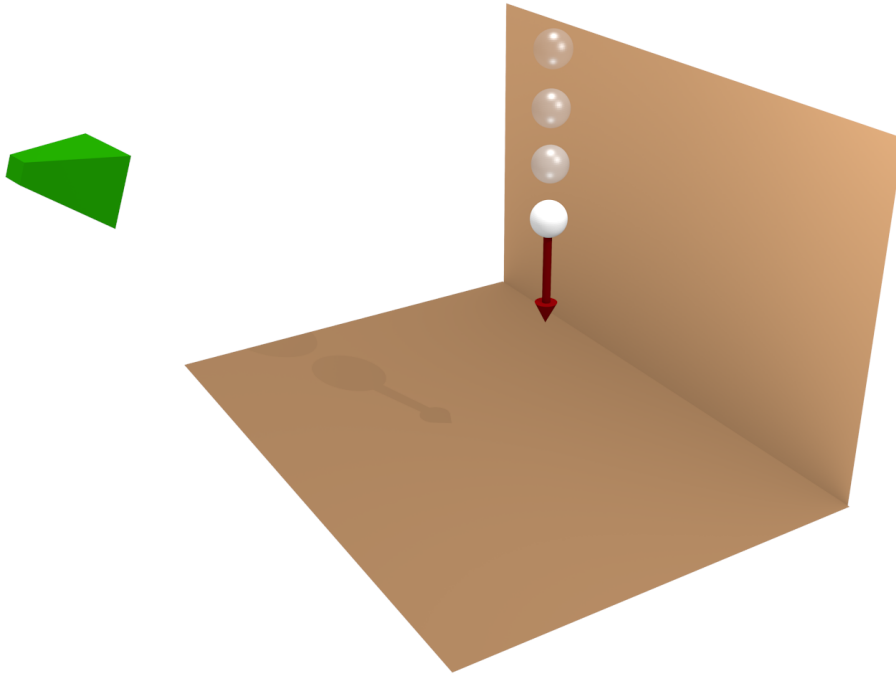


Figure 10.3: Experimental setup A. A table tennis ball is released from a stationary position and accelerates towards the ground. The camera observes this fall at a slightly downwards angle.

10.4. Implementation and Validation

Implementation details We implement the above modifications to the Kinect sensor [7] by using a custom firmware and modified driver software. However, in principle, a similar mode of operation could be supported by other phase-based ToF cameras available commercially, *e.g.* the ones available from PMD, Intel, and Mesa Imaging.

The tracking algorithm is implemented in C++ on a CPU and the rendering and likelihood computations are performed entirely on the GPU, implemented using the HLSL shading language. A tiled layout for the rendering pipeline enables to evaluate over 8000 particles in parallel and when using 4096 particles allows real time tracking at 300 Hz on modern GPU hardware.

Experimental setup A. For our experiments we use two different setups as shown in Fig. 10.3 and Fig. 10.5. In the first setup, we use a static camera mounted on a tripod and a table tennis ball as object model. The ball is released from a fixed position with no inertia and falls downwards driven purely by acceleration due to gravity.

To assess tracking performance quantitatively we use the following procedure. Because the ball starts from rest, the trajectory lies on a line in 3D space. Our tracker predicts object coordinates at each observation as the average of the weighted particle positions. For each sequence we use the predicted coordinates and fit a line through the coordinates in three dimensional space using least squares. We measure the deviation between the predicted coordinate from the line and the magnitude of this deviation is a reasonable assessment of the quality of the tracking result.

The camera is tilted downwards which results in movement of the ball not only along the y axis but also results in a change of the measured depth along in the z axis. Thus, because motion is present only in the y/z plane, we fit a least squares regressor $z_t \approx ay_t + b$ from the

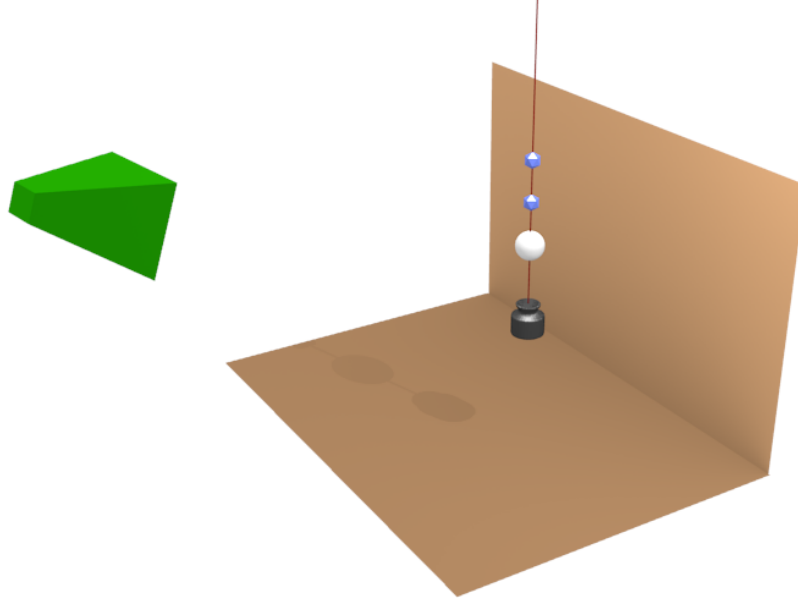


Figure 10.4.: Experimental setup B. A table tennis ball attached to a rope is obtaining a pendulum movement. Attached to the rope are two reflective markers used for motion capture. The rope is ensured to be a straight line by an attached weight. The scene is captured by the Kinect camera and by a commercial Motion Capture system consisting of 11 cameras (not shown).

the object position at step t as predicted by the tracker. The error metric is then the root mean squared error (RMSE),

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1:T} (z_t - (ay_t + b))^2}. \quad (10.9)$$

To avoid potential biases due to initialisation effects we perform the above for only the second half timespan of each sequence, where the result of the tracking algorithm has suitably converged. All our methods use the same transition model and the same parameters in (10.5), so no systematic bias due to assuming a strict linear motion benefits one model over the other.

Experimental setup B. In our second setup, the table tennis ball is attached to a rope together with two reflective markers. The markers are tracked in 3D at 150 Hz using an eleven-camera motion capture system (Qualisys QTM, Qualisys Inc., Gothenburg, Sweden). Attached to the end of the rope is also a weight, which straightens the rope. For quantitative comparison we transform the 3D trajectory of the motion capture system to the Kinect camera coordinate frame and compute the root mean squared error (RMSE) between the three dimensional coordinates of the raw ToF tracker result and the motion capture system output.

Notes on the accuracy: The motion capture system is calibrated with a residual of less than 2 mm. The coordinate frames of the Kinect camera and the motion capture system are registered by using six reflective markers which are visible in the Kinect camera frame. In the Kinect camera frame, depth values are assigned to the markers using the standard depth reconstruction of the Kinect. Registration of both coordinate frames is achieved by using Kabsch’s algorithm [73] with a residual of 8 – 9 mm.

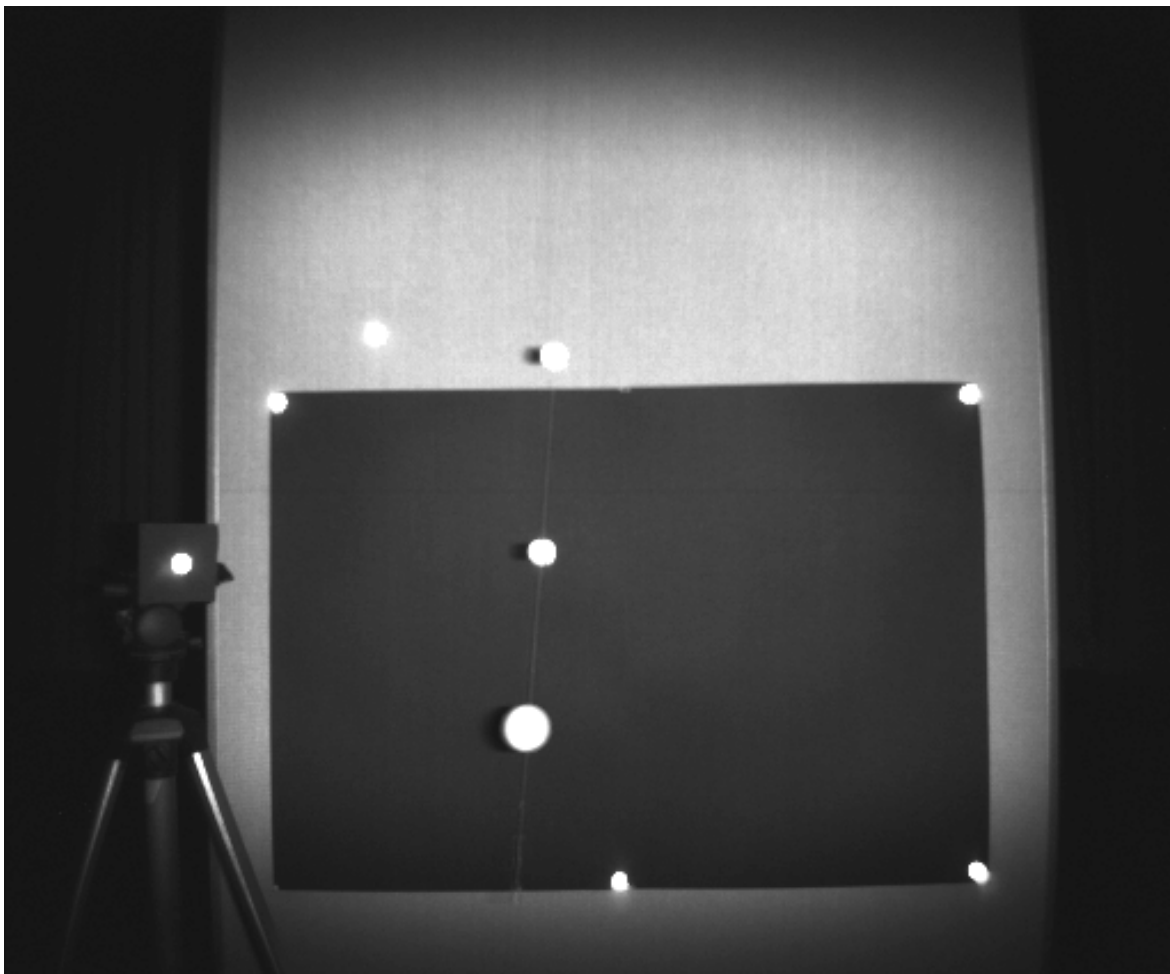


Figure 10.5.: Experimental setup B. A table tennis ball is attached to a rope with two reflective markers. Additional markers in the scene allow to register the coordinate frames of the camera and of the motion capture system. Shown in an averaged infrared image of the ToF camera.

10.5. Experiments

Our approach combines the use of raw ToF observations with a high frame rate. These benefits are complementary, and are best understood as part of a landscape of possible camera modes, as shown in Table 10.1. As a consequence we design the experiments to verify that both the ToF modulation and the equispaced frames are beneficial for tracking and verify that these benefits are complementary.

We first demonstrate that tracking based on phase unwrapping, the standard depth reconstruction method for ToF cameras, fails for fast moving objects because of motion artefacts.

10.5.1. Failure of Depth Based Tracking

The underlying assumption of a reconstruction algorithm based on phase unwrapping is a static scene. As a consequence, when an object moves between two raw frames and those frames are used for reconstructing the depth image, artefacts become visible in the depth reconstruction. Fig. 10.6 shows an overlay of the raw frames of the falling table tennis ball together with the depth reconstructions obtained from these frames. A depth value can only be reconstructed in those regions where the moving object overlaps in all the raw frames used for the depth reconstruction. Therefore it is obvious that the strategy of first reconstructing a

		Camera frame rate	
		Low fps	High fps
Mode	unmodulated	video camera	highspeed camera
	modulated	ToF camera	our raw ToF

Table 10.1.: Relevant dimensions of camera operation and frame rate for object tracking. Within the four quadrants going to the right or going downwards potentially improves tracking performance. Our approach combines the benefits of a high frame rate with phase modulation to provide superior tracking performance.

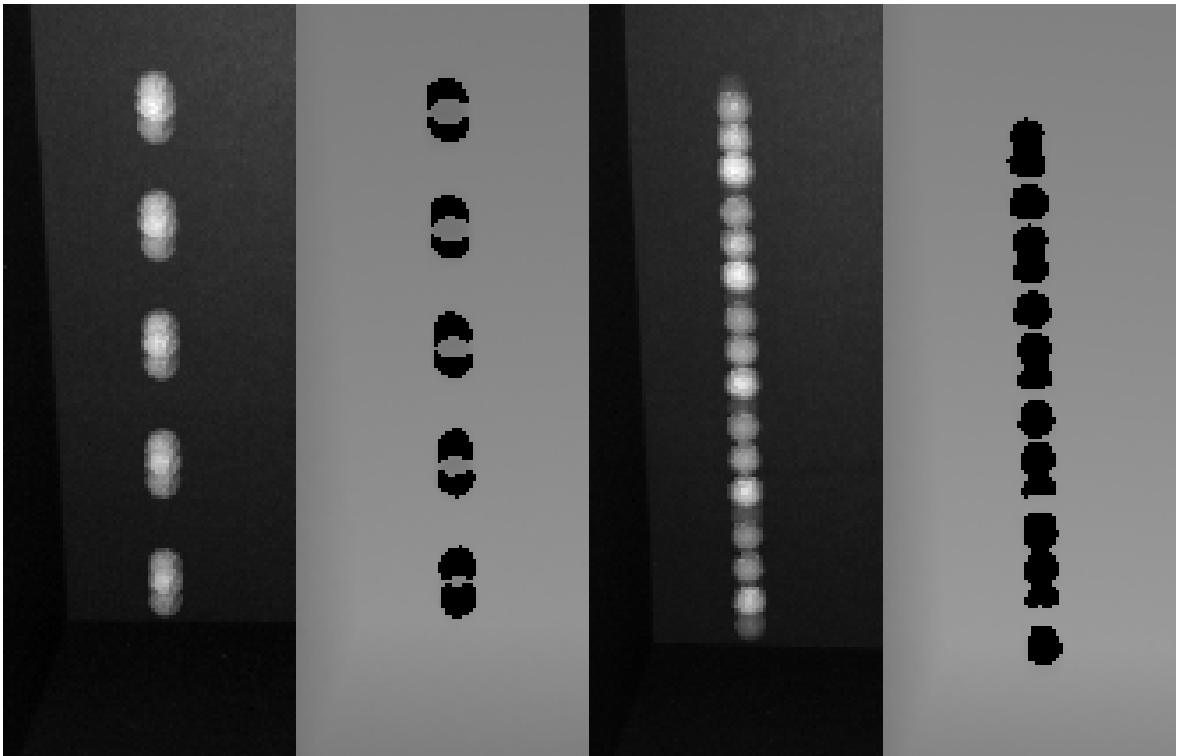


Figure 10.6.: Depth reconstruction failure (Experiment A). The ball quickly reaches a velocity that prevents a successful depth reconstruction by phase unwrapping. This is due to insufficient overlap of the object in the nine frames used for reconstruction. Left two images: an overlay of the raw captures for five frames and the corresponding depth reconstructions, using the standard ‘clustered’ exposure timing of the Kinect. Note the depth reconstruction gets worse with increasing velocity (the black ‘holes’). Right two images: Equidistant exposure timing. Depth reconstruction now completely fails. However, we show that this timing is beneficial for our proposed tracking method because we can directly leverage the high frame-rate raw ToF information.

depth image and then tracking fast moving objects has to fail. We therefore propose to track the object in three dimensions by directly using the raw data of the sensor.

10.5.2. Tracking with Raw ToF Observations

We now show that the unknown depth of an object can be obtained by our model-based tracking method. Fig. 10.7 depicts the estimated depth with respect to the vertical y coordinate of the falling table tennis ball (Experiment A). To validate our method, we show for comparison the depth value of the table tennis ball, which was reconstructed with the standard phase

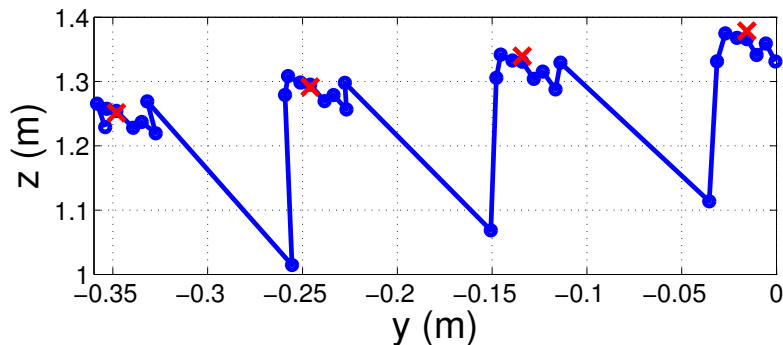


Figure 10.7.: Validation of the estimated depth values when tracking off raw ToF images captured using the standard ‘clustered’ temporal spacing (blue), in comparison to the depth values of the standard time of flight reconstruction method (red). (Experiment A)

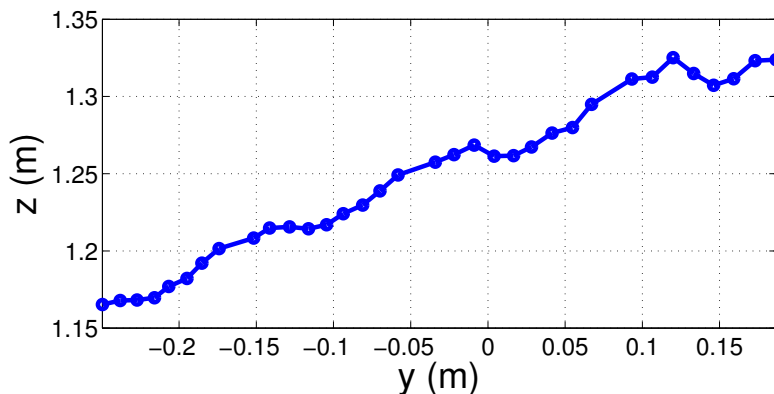


Figure 10.8.: Equispaced exposure timing leads to a more stable depth estimate when tracking from raw ToF captures. (Experiment A)

unwrapping approach. To acquire this value, we average over depth values in the interior of the overlapping region of the table tennis ball in the depth image (this region is visible in Fig. 10.6).

In the standard exposure timing, all nine raw ToF captures are taken at the beginning of each 30Hz depth frame cycle in order to minimise motion artefacts. For tracking purposes these unevenly spaced exposures are suboptimal: the larger time-gap between the captures leads to low quality of the depth estimate for the first frame of each capture, as is visible by the sharp drop of the estimated depth in every 9th frame in Fig. 10.7.

10.5.3. Benefit of Equispacing

To overcome the large time gap between the 30Hz captures we propose to use an equidistant timing of the exposures. Fig. 10.8 clearly demonstrates that this increases the stability of the trajectory estimates. Note that Fig. 10.8 tracks a different sequence than the sequence depicted in Fig. 10.7, since due to interference we are unable to capture the same scene simultaneously with two cameras, and thus it is necessary to capture separate sequences for different exposure timings.

We also quantitatively compared the effect of the equidistant timings in comparison with the standard clustered exposure timings by computing the root mean squared error of the residual towards a linear regressor as explained in the beginning of this section. We compared both exposure timings by individually tracking 10 sequences with 60 frames each of a table tennis ball falling straight to the ground. The camera is slightly tilted downwards which

leads to a linear relation between the y and the z coordinate. Our results in Table 10.2 and Table 10.3 show first that our model-based tracking method is highly accurate and second that the equidistant based shutter profile further improves this accuracy.

		RMSE
Exposure timing	clustered	1.62 cm
	equidistant	1.59 cm

Table 10.2.: Experiment A. Quantitative comparison of the standard clustered exposure timing and our proposed equidistant timing for a table tennis ball accelerated by gravity.

Exposure timing	Object Speed	RMSE	RMSE-z
clustered	1.40 kmh	4.15 cm	0.94 cm
	2.16 kmh	6.59 cm	2.59 cm
equidistant	1.12 kmh	2.40 cm	0.66 cm
	2.54 kmh	4.47 cm	1.39 cm

Table 10.3.: Experiment B. Quantitative comparison of the different exposure timings for a slow and fast moving table tennis ball. Shown is the root mean squared error between the raw ToF tracker and a commercial motion capture system, for all three coordinates (RMSE) and for the z coordinate in the camera coordinate frame (RMSE-z).

10.6. Discussion

The proposed method has two major benefits: Tracking of an object directly in the raw ToF observations avoids to first reconstruct a depth image, and the equispacing of the raw ToF captures, that increases the stability of the tracking result.

10.6.1. Tracking from Raw ToF

The proposed observation model allows to directly compare a model of the object with the raw ToF observation. In contrast to a comparison in the reconstructed depth images, a potential benefit of the proposed approach is a reduction in computational cost: whereas a depth-based tracking approach first has to reconstruct a depth image and then compute the likelihood between the model and the estimated depth, the presented approach allows to skip the potentially costly depth reconstruction process. This can be a significant advantage for power-limited environments, e.g. mobile devices.

An even greater benefit of the proposed raw ToF tracking approach is the avoidance of motion artefacts that would occur in phase unwrapping based depth reconstruction from observations of fast moving objects (examples of fast motion that would cause problems are shown in Fig. 10.2). The individual exposure time of a single raw ToF frame is by and order of magnitude shorter compared to the time interval in which a full sequence of nine frames, used for phase unwrapping based depth reconstruction, is recorded, even when these exposures are clustered in the beginning of the depth frame cycle.

Still very fast object motion can result in motion blurring in the raw ToF capture. But in contrast to the motion artefacts due to phase unwrapping of a sequence of raw captures, motion blurring in a single raw ToF frame is negligible, because the exposure time of a single raw frame is an order of magnitude shorter than the exposure time across all nine ToF raw frames required for phase unwrapping based depth reconstruction.

10.6.2. Equispaced ToF Captures

We proposed to space out the raw captures uniformly in time (see Fig. 10.1) and verified in our experiments that this increases the stability of the tracking result. While for the standard exposure timing the goal was to reconstruct depth, it was useful to cluster the frames in time to minimise motion between the captures and achieve the required consistence of the captures for phase unwrapping. In our approach, we are instead interested in an exposure timing that is beneficial for tracking. By capturing the frames uniformly in time, the measurements better cover the movement of the object along the trajectory and allow an increases stability of the tracking result.

10.6.3. Limitations and Future Work

As we have shown, the ToF tracking framework obtains several benefits, but there are also some limitations in practice. Tracking in ToF captures of a cluttered scene is much harder than tracking in the depth frame. As future work, we hope to explore other model-based tracking algorithms which allow for tracking more general objects including articulated objects, as well as to extend our method to handle arbitrary backgrounds. Also, especially for articulated objects, we believe that the benefits of both methods, depth based tracking and the raw ToF based tracking proposed here, can be combined. Combining both methods allows to combine the strengths of both approaches by using the more robust depth based tracking for slow moving object parts and utilising the higher time resolution of the ToF tracking framework for the faster moving parts.

10.7. Conclusion

We proposed a novel mode of operation of a commonly available ToF sensor for the purpose of high-speed object tracking. Through experiments we demonstrated improved tracking accuracy due to two distinct contributions: an observation model for raw ToF frame observations, and by spacing out the captures uniformly over time.

Our approach is potentially useful for other fast moving objects, for example the human hand We believe our work is just the first step in adapting ToF sensor operation to better fit computer vision tasks. We considered tracking, but other vision applications such as surface reconstruction and camera localisation may benefit similarly from such a co-design of algorithm and sensor. In addition, whereas we still operate the camera with an exposure timing that is fixed apriori, possible future work could explore how to adapt the camera operation online depending on observed data and the estimated object trajectory.

Part IV.

Conclusions and Outlook

11. Concluding Remarks

Part V.

Appendix

12. Projective Geometry and Camera Models

13. Sequential Monte Carlo Filtering

-
- [1] E. Aganj, J.-P. Pons, F. Ségonne, and R. Keriven. Spatio-temporal shape from silhouette using four-dimensional delaunay meshing. In *Proc. International Conference on Computer Vision*, pages 1–8. 2007. (cited on page 68)
 - [2] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variation and free discontinuity problems*. Oxford Mathematical Monographs. The Clarendon Press Oxford University Press, New York, 2000. (cited on page 13)
 - [3] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variation and free discontinuity problems*. Oxford Mathematical Monographs. The Clarendon Press Oxford University Press, New York, 2000. (cited on page 71)
 - [4] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(9):855–867, 1990. (cited on page 36)
 - [5] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011. (cited on page 90)
 - [6] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. (cited on page 27)
 - [7] C. S. Bamji, P. O’Connor, T. A. Elkhatib, S. Mehta, B. Thompson, L. A. Prather, D. Snow, O. C. Akkaya, A. Daniel, A. D. Payne, T. Perry, M. Fenton, and V.-H. Chan. A 0.13 μm CMOS system-on-chip for a 512×424 time-of-flight image sensor with multi-frequency photo-demodulation up to 130 MHz and 2 GS/s ADC. *J. Solid-State Circuits*, 50(1):303–319, 2015. (cited on pages 91, 93, and 96)
 - [8] R. Barlow and H. Brunk. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147, 1972. (cited on page 47)
 - [9] C. Bauer, T. Pock, E. Sorantin, H. Bischof, and R. Beichel. Segmentation of interwoven 3d tubular tree structures utilizing shape priors and graph cuts. *Medical image analysis*, 14(2):172–184, 2010. (cited on page 25)
 - [10] F. Benmansour and L. D. Cohen. Tubular structure segmentation based on minimal path method and anisotropic enhancement. *International Journal of Computer Vision*, 92:192–210, 2011. (cited on pages 25 and 27)
 - [11] F. Benmansour, L. D. Cohen, M. Law, and A. Chung. Tubular anisotropy for 2d vessel segmentation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2286–2293. IEEE, 2009. (cited on page 25)
 - [12] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun. Discrete Elastic Rods. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 27(3):63:1–63:12, 2008. (cited on page 35)
 - [13] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha. Object stereo—joint stereo matching and object segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 3081–3088. IEEE, 2011. (cited on pages 69 and 80)
 - [14] J. F. Blinn. Models of light reflection for computer synthesized pictures. *ACM SIGGRAPH Computer Graphics*, 11(2):192–198, 1977. (cited on page 95)

-
- [15] S. Bougleux, A. Elmoataz, and M. Melkemi. Discrete regularization on weighted graphs for image and mesh filtering. In F. Sgallari, A. Murli, and N. Paragios (Editors), *SSVM*, volume 4485 of *Lecture Notes in Computer Science*, pages 128–139. Springer, 2007. (cited on pages 30 and 31)
- [16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. (cited on page 12)
- [17] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. (cited on pages 7, 8, and 9)
- [18] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004. (cited on pages 23 and 30)
- [19] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001. (cited on pages 25, 26, 30, and 56)
- [20] K. Bredies and D. Lorenz. Mathematische bildverarbeitung. *Vieweg+ Teubner*, 4(6):12, 2011. (cited on page 13)
- [21] O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007. (cited on page 94)
- [22] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, 2004. (cited on pages 82 and 83)
- [23] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340):227, 2010. (cited on page 13)
- [24] A. Chambolle, D. Cremers, and T. Pock. A convex approach for computing minimal partitions. Technical report TR-2008-05, Dept. of Computer Science, University of Bonn, Bonn, Germany, 2008. (cited on pages 11, 13, and 61)
- [25] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, 2011. (cited on pages 11, 12, 13, 30, 32, 45, 46, 61, and 84)
- [26] T. Chan, S. Esedoğlu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006. (cited on pages 26, 28, 56, and 61)
- [27] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, 2001. (cited on page 26)
- [28] C. Chen, D. Freedman, and C. H. Lampert. Enforcing topological constraints in random field image segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 2089–2096. IEEE, 2011. (cited on pages 23, 25, 26, and 43)
- [29] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, 2003. (cited on page 90)

-
- [30] A. Criminisi, T. Sharp, and A. Blake. Geos: Geodesic image segmentation. In *Computer Vision–ECCV 2008*, pages 99–112. Springer, 2008. (cited on page 27)
- [31] L. Csató and M. Opper. Sparse on-line gaussian processes. *Neural Computation*, 14(3):641 – 668, 2002. (cited on pages 56 and 57)
- [32] T. K. Dey, F. Fan, and Y. Wang. An efficient computation of handle and tunnel loops via reeb graphs. *ACM Trans. Graph.*, 32(4):32, 2013. (cited on page 74)
- [33] T. K. Dey, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3d models. *ACM Trans. Graph.*, 27(3), 2008. (cited on pages 73 and 74)
- [34] R. Diestel. *Graph Theory*. Springer-Verlag, Berlin, Heidelberg, 2006. (cited on pages 17, 18, and 19)
- [35] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. (cited on page 36)
- [36] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009. (cited on page 94)
- [37] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82(2):421–439, 1956. (cited on page 12)
- [38] J. Durbin and S. Koopman. *Time Series Analysis by State Space Methods: Second Edition*. Oxford Statistical Science Series. OUP Oxford, 2012. (cited on page 93)
- [39] N. Y. El-Zehiry and L. Grady. Fast global optimization of curvature. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 3257–3264. IEEE, 2010. (cited on pages 15, 23, 38, and 50)
- [40] P. Elias, A. Feinstein, and C. Shannon. A note on the maximum flow through a network. *IRE Transactions on Information Theory*, 2(4):117–119, 1956. (cited on page 30)
- [41] A. Elmoataz, O. Lezoray, and S. Boughleux. Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *Image Processing, IEEE Transactions on*, 17(7):1047 –1060, 2008. (cited on pages 30 and 31)
- [42] C. H. Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004. (cited on page 70)
- [43] H. Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93(3):418–491, 1959. (cited on page 14)
- [44] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. (cited on page 57)
- [45] W. H. Fleming and R. Rishel. An integral formula for total gradient variation. *Archiv der Mathematik*, 11(1):218–222, 1960. (cited on page 14)
- [46] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404, 1956. (cited on page 30)
- [47] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 98*, pages 130–137. Springer, 1998. (cited on page 25)

-
- [48] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, UAI'97*. 1997. (cited on page 95)
- [49] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010. (cited on pages 78 and 79)
- [50] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976. (cited on page 12)
- [51] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. 14(3):367–383, 1992. (cited on page 81)
- [52] R. Glowinski and A. Marroco. Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires. *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(2):41–76, 1975. (cited on page 12)
- [53] S. J. Godsill, A. Doucet, and M. West. Monte carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, 2004. (cited on page 94)
- [54] B. Goldluecke and D. Cremers. Convex relaxation for multilabel problems with product label spaces. In *Proc. European Conference on Computer Vision*. 2010. (cited on page 15)
- [55] B. Goldluecke and D. Cremers. Introducing total curvature for image processing. In *Proc. International Conference on Computer Vision*. 2011. (cited on pages 15 and 23)
- [56] B. Goldluecke, I. Ihrke, C. Linz, and M. Magnor. Weighted minimal hypersurface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1194–1208, 2007. (cited on page 68)
- [57] B. Goldluecke and M. Magnor. Space-time isosurface evolution for temporally coherent 3D reconstruction. In *Proc. International Conference on Computer Vision and Pattern Recognition*, volume I, pages 350–355. 2004. (cited on page 68)
- [58] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993. (cited on pages 93 and 94)
- [59] L. Grady. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768–1783, 2006. (cited on page 23)
- [60] J.-Y. Guillemaut and A. Hilton. Space-time joint multi-layer segmentation and depth estimation. In *Proc. International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 440–447. 2012. (cited on page 68)
- [61] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 3129–3136. IEEE, 2010. (cited on pages 25 and 43)
- [62] M. Gupta, S. K. Nayar, M. B. Hullin, and J. Martin. Phasor imaging: A generalization of correlation-based time-of-flight imaging. Technical Report, Department of Computer Science, Columbia University, 2014. (cited on page 92)

-
- [63] X. Han, C. Xu, and J. L. Prince. A topology preserving level set method for geometric deformable models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(6):755–768, 2003. (cited on pages 23, 25, and 26)
- [64] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison. Real-time camera tracking: When is high frame-rate best? In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII*. Springer, 2012. (cited on page 91)
- [65] M. E. Hansard, S. Lee, O. Choi, and R. Horaud. *Time-of-Flight Cameras - Principles, Methods and Applications*. Springer Briefs in Computer Science. Springer, 2013. (cited on page 92)
- [66] M. Hein, J.-Y. Audibert, and U. von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8:1325–1368, 2007. (cited on page 30)
- [67] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. (cited on page 84)
- [68] Institut national de recherche en informatique et en automatique (INRIA) Rhône Alpes. 4d repository. <http://4drepository.inrialpes.fr/>. (cited on pages 68 and 78)
- [69] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. (cited on page 93)
- [70] M. Isard and A. Blake. A smoothing filter for CONDENSATION. In H. Burkhardt and B. Neumann (Editors), *Computer Vision - ECCV'98, 5th European Conference on Computer Vision, Freiburg, Germany, June 2-6, 1998, Proceedings, Volume I*, volume 1406 of *Lecture Notes in Computer Science*, pages 767–781. Springer, 1998. (cited on page 94)
- [71] M. Jancosek and T. Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 3121–3128. 2011. (cited on pages 78 and 79)
- [72] H. Jin, P. Favaro, and S. Soatto. Real-time 3-d motion and structure of point-features: A front-end for vision-based control and interaction. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 2778–2779. 2000.
- [73] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976. (cited on page 97)
- [74] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Gaussian processes for object categorization. *International Journal of Computer Vision*, 88(2):169–188, 2010. (cited on page 56)
- [75] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. (cited on page 26)
- [76] M. M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, pages 61–70. 2006. (cited on pages 78 and 79)
- [77] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*. Nara, Japan, 2007.

-
- [78] M. Klodt. *Convex Relaxation Methods for Image Segmentation and Stereo Reconstruction*. Dissertation, Technische Universität München, München, 2014. (cited on page 33)
- [79] M. Klodt, T. Schoenemann, K. Kolev, M. Schikora, and D. Cremers. An experimental comparison of discrete and continuous shape optimization methods. In *European Conference on Computer Vision (ECCV)*. Marseille, France, 2008. (cited on pages 28 and 29)
- [80] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009. (cited on page 69)
- [81] D. Koller, K. Daniilidis, and H. H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993. (cited on page 93)
- [82] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Troussel. Model-based detection of tubular structures in 3d images. *Computer Vision and Image Understanding*, 80(2):130–171, 2000. (cited on page 25)
- [83] M. Kristan, R. P. Pflugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay, T. Vojtř, G. Fernández, A. Lukežič, A. Dimitriev, A. Petrosino, A. Saffari, B. Li, B. Han, C. Heng, C. Garcia, D. Pangercic, G. Häger, F. S. Khan, F. Oven, H. Possegger, H. Bischof, H. Nam, J. Zhu, J. Li, J. Y. Choi, J.-W. Choi, J. F. Henriques, J. van de Weijer, J. Batista, K. Lebeda, K. Öfjäll, K. M. Yi, L. Qin, L. Wen, M. E. Maresca, M. Danelljan, M. Felsberg, M.-M. Cheng, P. H. S. Torr, Q. Huang, R. Bowden, S. Hare, S. Y. Lim, S. Hong, S. Liao, S. Hadfield, S. Z. Li, S. Duffner, S. Golodetz, T. Mauthner, V. Vineet, W. Lin, Y. Li, Y. Qi, Z. Lei, and Z. H. Niu. The visual object tracking VOT2014 challenge results. In *ECCV Workshops*. Springer, 2014. (cited on page 90)
- [84] C. H. Lampert and J. Peters. Real-time detection of colored objects in multiple camera streams with off-the-shelf hardware components. *Journal of Real-Time Image Processing*, 7(1):31–41, 2012. (cited on page 91)
- [85] R. Lange and P. Seitz. Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics*, 37(3):390–397, 2001. (cited on page 92)
- [86] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse gaussian process methods: The informative vector machine. *Adv. in neural inf. proc. systems*, 15:609–616, 2002. (cited on pages 56 and 57)
- [87] N. D. Lawrence, J. C. Platt, and M. I. Jordan. Extensions of the informative vector machine. In *Proc. of the First Intern. Conf. on Deterministic and Statistical Methods in Machine Learning*, pages 56–87. Springer-Verlag, 2004. (cited on page 60)
- [88] J. M. Lee. *Introduction to Topological Manifolds*. Springer-Verlag, New York, 2000. (cited on pages 17 and 18)
- [89] D. Lefloch, R. Nair, F. Lenzen, H. Schäfer, L. Streeter, M. J. Cree, R. Koch, and A. Kolb. Technical foundation and calibration methods for time-of-flight cameras. In *Time-of-Flight and Depth Imaging: Sensors, Algorithms, and Applications*, pages 3–24. Springer, 2013. (cited on page 92)
- [90] J. Lellmann, F. Becker, and C. Schnörr. Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In *2009 IEEE 12th International Conference on Computer Vision*, pages 646–653. 2009. (cited on page 15)

-
- [91] 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. (cited on page 25)
- [92] H. Lombaert, Y. Sun, L. Grady, and C. Xu. A multilevel banded graph cuts method for fast image segmentation. In *Proc. International Conference on Computer Vision*, pages 259–265. 2005. (cited on page 56)
- [93] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21:163–169, 1987. (cited on page 70)
- [94] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno. 1ms column parallel vision system and it’s application of high speed target tracking. In *ICRA. IEEE*, 2000. (cited on page 90)
- [95] C. Nieuwenhuis and D. Cremers. Spatially varying color distributions for interactive multi-label segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1234–1247, 2013. (cited on pages 56, 59, 61, 62, and 63)
- [96] D. Nister. Preemptive ransac for live structure and motion estimation. In *Proc. International Conference on Computer Vision*, pages 199–206. 2003.
- [97] S. Nowozin and C. H. Lampert. Global connectivity potentials for random field models. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 818–825. IEEE, 2009. (cited on pages 23 and 25)
- [98] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*. Springer, 2004. (cited on page 90)
- [99] M. R. Oswald. *Convex Variational Methods for Single-View and Space-Time Multi-View Reconstruction*. Dissertation, Technische Universität München, München, 2015. (cited on page 67)
- [100] M. R. Oswald and D. Cremers. A convex relaxation approach to space time multi-view 3d reconstruction. In *ICCV - Workshop on Dynamic Shape Capture and Analysis (4DMOD)*. 2013. (cited on pages 67, 68, 69, 70, and 79)
- [101] M. R. Oswald, J. Stühmer, and D. Cremers. Generalized connectivity constraints for spatio-temporal 3d reconstruction. In *Proc. European Conference on Computer Vision*, pages 32–46. 2014. (cited on pages 67 and 75)
- [102] P. M. Pardalos and G. Xue. Algorithms for a class of isotonic regression problems. *Algorithmica*, 23(3):211–222, 1999. (cited on pages 47, 48, and 49)
- [103] N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014. (cited on pages 7 and 12)
- [104] R. Paul, R. Triebel, D. Rus, and P. Newman. Semantic categorization of outdoor scenes with uncertainty estimates using multi-class Gaussian process classification. In *Proc. of the Intern. Conf. on Intelligent Robots and Systems (IROS)*. 2012. (cited on page 57)
- [105] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1762–1769. IEEE, 2011. (cited on pages 12, 32, 37, 77, 78, and 84)

-
- [106] T. Pock, A. Chambolle, H. Bischof, and D. Cremers. A convex relaxation approach for computing minimal partitions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Miami, Florida, 2009. (cited on pages 11, 13, 15, and 61)
- [107] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the piecewise smooth Mumford-Shah functional. In *Proc. International Conference on Computer Vision*. Kyoto, Japan, 2009. (cited on pages 11, 13, 32, 45, 46, and 61)
- [108] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *European Conference on Computer Vision (ECCV)*. Marseille, France, 2008. (cited on page 15)
- [109] M. Rempfler, B. Andres, and B. Menze. The minimum cost connected subgraph problem in medical image analysis. In *MICCAI*. 2016. (Accepted). (cited on page 41)
- [110] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. (cited on page 90)
- [111] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics (TOG)*, 23(3):309–314, 2004. (cited on page 56)
- [112] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992. (cited on page 83)
- [113] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. (cited on page 55)
- [114] J. Santner, T. Pock, and H. Bischof. Interactive multi-label segmentation. In *Proc. Asian Conference on Computer Vision*, pages 397–410. Springer, 2011. (cited on pages 61 and 62)
- [115] T. Schoenemann and D. Cremers. Introducing curvature into globally optimal image segmentation: Minimum ratio cycles on product graphs. In *Proc. International Conference on Computer Vision*. Rio de Janeiro, 2007. (cited on page 15)
- [116] T. Schoenemann, F. Kahl, and D. Cremers. Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In *Proc. International Conference on Computer Vision*. Kyoto, Japan, 2009. (cited on page 15)
- [117] T. Schoenemann, F. Kahl, S. Masnou, and D. Cremers. A linear framework for region-based image segmentation and inpainting involving curvature penalization. *International Journal of Computer Vision*, 2012. (cited on pages 15 and 23)
- [118] R. Schwarte, Z. Xu, H.-G. Heinol, J. Olk, R. Klein, B. Buxbaum, H. Fischer, and J. Schulte. New electro-optical mixing and correlating sensor: facilities and applications of the photonic mixer device (PMD). In *Proc. SPIE*, volume 3100, pages 245–253. 1997. (cited on page 92)
- [119] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. (cited on page 36)
- [120] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. Izadi. Accurate, robust, and flexible real-time hand tracking. CHI, 2015. (cited on page 89)

-
- [121] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013. (cited on page 89)
- [122] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken. Ridge-based vessel segmentation in color images of the retina. *Medical Imaging, IEEE Transactions on*, 23(4):501–509, 2004. (cited on pages 38 and 40)
- [123] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007. (cited on page 68)
- [124] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*. 1999. (cited on page 95)
- [125] J. Stühmer. *Ein Variationsansatz zur Schätzung von dichten Tiefenkarten im Kontext des Structure-from-Motion*. Diplomarbeit, TU Dresden, Germany, 2010. (cited on pages 4 and 81)
- [126] J. Stühmer and D. Cremers. A fast projection method for connectivity constraints in image segmentation. In X.-C. Tai, E. Bae, T. F. Chan, and M. Lysaker (Editors), *Proceedings of the International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, LNCS. 2015. (cited on pages 43 and 51)
- [127] J. Stühmer, S. Gumhold, and D. Cremers. Parallel generalized thresholding scheme for live dense geometry from a handheld camera. In *ECCV Workshop on Computer Vision on GPUs (CVGPU)*. Heraklion, Greece, 2010. (Part of Diploma Thesis). (cited on page 81)
- [128] J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Pattern Recognition (Proc. DAGM)*, pages 11–20. Darmstadt, Germany, 2010. (Part of Diploma Thesis). (cited on page 81)
- [129] J. Stühmer, S. Nowozin, A. Fitzgibbon, R. Szeliski, T. Perry, S. Acharya, D. Cremers, and J. Shotton. Model-based tracking at 300hz using raw time-of-flight observations. In *Proc. International Conference on Computer Vision*. Santiago, Chile, 2015. (cited on page 89)
- [130] J. Stühmer, P. Schröder, and D. Cremers. Tree shape priors with connectivity constraints using convex relaxation on general graphs. In *Proc. International Conference on Computer Vision*. Sydney, Australia, 2013. (cited on pages 23, 43, 44, 45, 46, 50, 51, 68, 71, 79, and 80)
- [131] W. A. Sutherland. *Introduction to metric and topological spaces*. Clarendon Press, Oxford, 1975. (cited on pages 17 and 18)
- [132] R. Triebel, H. Grimmitt, R. Paul, and I. Posner. Driven learning for driving: How introspection improves semantic mapping. In *Proc of Intern. Symposium on Robotics Research (ISRR)*. 2013. (cited on page 57)
- [133] R. Triebel, J. Stühmer, M. Souiai, and D. Cremers. Active online learning for interactive segmentation using sparse gaussian processes. In *German Conference on Pattern Recognition*. 2014. (cited on page 55)
- [134] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *Automatic Control, IEEE Transactions on*, 40(9):1528–1538, 1995. (cited on page 36)

-
- [135] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof. Tvseg-interactive total variation based image segmentation. In *British Machine Vision Conference*, volume 2. Citeseer, 2008. (cited on pages 23 and 56)
- [136] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Niessner, A. Criminisi, S. Izadi, and P. Torr. Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Trans. on Graphics (TOG)*, 2015. (cited on page 89)
- [137] A. Vezhnevets, E. J. Buhmann, and V. Ferrari. Active learning for semantic segmentation with expected change. In *Proc. International Conference on Computer Vision and Pattern Recognition*. 2012. (cited on page 57)
- [138] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *Proc. International Conference on Computer Vision and Pattern Recognition*. 2008. (cited on pages 23, 25, 26, 39, 43, 50, and 51)
- [139] D. Wang, C. Yan, S. Shan, and X. Chen. Active learning for interactive segmentation with expected confidence change. In *Proc. Asian Conference on Computer Vision*. 2012. (cited on page 57)
- [140] A. D. Worrall, R. F. Marslin, G. D. Sullivan, and K. D. Baker. Model-based tracking. In P. Mowforth (Editor), *BMVC*, pages 1–9. BMVA Press, 1991. (cited on page 93)
- [141] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011. (cited on page 90)
- [142] A. Yilmaz, O. Javed, and M. Shah. Object tracking: a survey. *ACM Computing Surveys*, 38(4):13:1–13:45, 2006. (cited on page 90)
- [143] Y. Zeng, D. Samaras, W. Chen, and Q. Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. *Computer Vision and Image Understanding*, 112(1):81–90, 2008. (cited on page 25)
- [144] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *ICPR*. 2004. (cited on page 95)