Faculty of Informatics

Chair for Computer Vision and Pattern Recognition

Convex Variational Methods for Semantic Image Analysis

Julia Diebold

Technische Universität München

Technische Universität München

Fakultät für Informatik
Lehrstuhl für Bildverarbeitung und Mustererkennung

Convex Variational Methods for Semantic Image Analysis

Julia Diebold

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. Nils Thuerey
Prüfer der Dissertation:	1. Univ.-Prof. Dr. Daniel Cremers
	2. Univ.-Prof. Dr. Gabriele Steidl Technische Universität Kaiserslautern

Die Dissertation wurde am 20. August 2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 9. November 2015 angenommen.

To my family.

Abstract

Semantic image analysis – also referred to as class-specific image analysis – is a fundamental component in any computer vision system. In the course of the last years, a continuous effort has been made by the research community to derive algorithms which can analyze and understand visual scenes. The development of formalisms which can extract image information matching the human perception is a challenging task. Mathematically sophisticated methods have been developed to make the algorithms more powerful. However, in particular for complex images, current methods often yield unsatisfactory results.

In this thesis we extend state-of-the-art image analysis methods to allow for a more efficient and more reliable semantic analysis. At first, we consider the analysis of binary line drawings. We algorithmically localize target figures and show that diffusion improves the localization, as it makes the resulting energy easier to optimize. In the following, we focus on the extension of variational segmentation methods and the minimization of convex energy functionals. The advantage of such a formulation is that globally optimal solutions – in the sense of the energy – can be determined by applying established methods of convex optimization.

Semantic segmentation aims at jointly computing a partitioning of the image plane and a semantic labeling of the various regions in terms of previously learned object classes. In this thesis we consider energy functionals which are composed of a *regularization term* and a *data fidelity term* and propose improvements for both parts.

Regularization term The regularization term models prior knowledge. We propose midrange geometric priors which make use of geometric information, *e.g.*, that 'sky' lies above 'ground' or that 'wolf' and 'sheep' usually do not occur together. By penalizing the co-occurrence of specific labels within a certain spatial neighborhood the novel priors are beneficial for various segmentation scenarios. We show how to formulate the problem as a convex optimization problem and employ a novel primal-dual algorithm to find the globally optimal solution.

Data fidelity term Our data fidelity terms model the characteristics of the object classes. A discriminative model can be learned from a set of labeled training images or from user input, *e.g.*, bounding boxes or scribbles. For non-interactive applications we introduce a framework which automatically selects the required features and efficiently estimates the class probabilities. The data fidelity term is then computed as the negative logarithm of the class probabilities. Our proposed fully automatic algorithm achieves state-of-the-art semantic classifications and segmentations at drastically reduced computation time. If no training images are available user input can be used to learn the color model. We therefore introduce a novel approach based on RGB-D data and interactive user input via scribbles. We extend the idea of spatially varying color distributions in a plane to additionally incorporate depth information. By locally adapting the influence of nearby scribbles around each pixel we further improve the result.

Each segmentation method can be applied to video tracking, in particular video inpainting. Instead of processing the video frame by frame we propose a framework for temporally consistent video completion and introduce a flow-based propagation of the user scribbles. We achieve competitive results five times faster and with substantially less user input than required in competing methods.

The developed methods for semantic image analysis are implemented either in Matlab or in C++ with parallelization on GPU and compare well to state-of-the-art methods. All included works were published in highly ranked journals and international conferences.

Zusammenfassung

Semantische Bildanalyse – auch als klassenspezifische Bildanalyse bekannt – ist ein wesentlicher Bestandteil in jedem Bildverarbeitungssystem. Die Wissenschaftsgemeinde hat im Laufe der vergangenen Jahre kontinuierlich an neuen Algorithmen gearbeitet, die bildliche Darstellungen analysieren und verstehen können. Die Erforschung von Formalismen, die Bildinformationen ähnlich der menschlichen Wahrnehmung erkennen können, ist eine anspruchsvolle Aufgabe. Es wurden mathematisch ausgefeilte Verfahren entwickelt, welche die Algorithmen leistungsstärker machen. Insbesondere bei komplexen Eingabebildern liefern dem Stand der Technik entsprechende Methoden allerdings keine zufriedenstellenden Ergebnisse.

In dieser Arbeit erweitern wir aktuelle Bildverarbeitungsverfahren, um eine effizientere und zuverlässigere semantische Bildanalyse zu ermöglichen. Zunächst betrachten wir die Analyse von binären Strichzeichnungen. Wir suchen algorithmisch nach vorgegebenen Formen und zeigen, dass die Lokalisierung durch Diffusion verbessert wird, da die Energie dadurch einfacher zu optimieren ist. Anschließend konzentrieren wir uns auf die Erweiterung von Variationsansätzen für Bildsegmentierung und die Minimierung von konvexen Energiefunktionalen. Der Vorteil einer solchen Formulierung ist, dass bewährte Optimierungsverfahren eingesetzt und so global optimale Lösungen – im Sinne von Energie – bestimmt werden können.

Basierend auf zuvor gelernten Objektklassen berechnet die semantische Segmentierung gleichzeitig die Zerlegung der Bildebene und des semantischen Labels der verschiedenen Regionen. In dieser Doktorarbeit betrachten wir Energiefunktionale, welche aus einem *Regularisierungsterm* und einem *Datenterm* zusammengesetzt sind und stellen Verbesserungen für beide Teile vor.

Regularisierungsterm Der Regularisierungsterm modelliert Vorwissen. Wir bringen geometrische Informationen mit sogenannten *midrange geometric priors* ein. Beispielsweise befindet sich der ‚Himmel‘ über dem ‚Boden‘ und ‚Wolf‘ und ‚Schaf‘ treten gewöhnlich nicht zusammen auf. Durch die Bestrafung des gemeinsamen Auftretens spezifischer Label innerhalb einer bestimmten räumlichen Nachbarschaft verbessern die neuen Priors die Segmentierung verschiedenster Szenarien. Wir zeigen, wie das Problem als konvexes Optimierungsproblem formuliert werden kann und verwenden einen neuen Primal-Dualen Algorithmus um die global optimale Lösung zu bestimmen.

Datenterm Unsere Datenterme bilden die charakteristischen Merkmale von Objektklassen ab. Ein diskriminatives Modell kann aus einer Reihe von gelabelten Trainingsbildern oder aus Benutzereingaben, z. B. Objektrahmen oder Scribbles, gelernt werden. Für Anwendungen ohne Benutzerinteraktion stellen wir ein Konzept vor, das automatisch die benötigten Merkmale (sogenannte Features) auswählt und die Wahrscheinlichkeit der Klassen effizient schätzt. Der Datenterm wird anschließend als negativer Logarithmus der Klassenwahrscheinlichkeit bestimmt. Unser vollautomatischer Algorithmus erzielt dem Stand der Technik entsprechende semantische Klassifikationen und Segmentierungen bei drastisch reduzierter Rechenzeit. Wenn keine Trainingsbilder verfügbar sind, kann das Farbmodell mittels Benutzereingaben gelernt werden. Wir stellen dazu einen neuen Ansatz vor, welcher auf RGB-D-Daten und interaktiven Benutzereingaben via Scribbles basiert. Wir erweitern die Idee der räumlich variierenden Farbverteilungen in einer Ebene, um zusätzlich Tiefeninformation mit einzubeziehen. Da wir in der Umgebung jedes Pixels den Einfluss von nahegelegenen Scribbles lokal anpassen, können wir das Ergebnis weiter verbessern.

Jedes Segmentierungsverfahren kann für Videotracking eingesetzt werden, insbesondere für Video Inpainting. Anstatt die Einzelbilder des Videos zu verarbeiten, stellen wir ein Konzept für zeitlich konsistente Videovervollständigung sowie eine flussbasierte Verschiebung der Scribbles vor. Wir erzielen wettbewerbsfähige Ergebnisse bei einer um den Faktor fünf reduzierten Laufzeit und wesentlich geringeren Benutzereingaben als Konkurrenzverfahren.

Die entwickelten Methoden für die semantische Bildanalyse sind entweder in Matlab oder in C++ mit Parallelisierung auf GPU implementiert und können einem Vergleich mit dem neuesten Stand der Technik standhalten. Alle erwähnten Arbeiten wurden in hochrangigen Fachzeitschriften und internationalen Konferenzen publiziert.

Acknowledgements

To begin with, I would like to thank my doctoral advisor Prof. Daniel Cremers who made this work possible. In his excellent lectures he introduced me to the field of computer vision and excited my curiosity to learn more about image analysis. By doing my dissertation at his chair I got part of a great research community. He gave me the opportunity to work on a topic of my choice and to participate at several workshops and conferences where I could meet great people from all over the world.

Moreover, I want to thank my second referee Prof. Gabriele Steidl and also Prof. Nils Thuerey who both immediately agreed to be committee members of the disputation.

During this work I was supported by a lot of amazing people. I would like to thank for all the nice collaborations and fruitful discussions. Particularly, I would like to thank my co-authors Claudia, Michael M., Caner, Sibel, Michael S., Nikolaus, Mohamed, Daniel, Guy, Gulce, Erin, Ellen, Ruizhen, Kathryn, Matineh and Carola for the pleasant teamwork. This dissertation would not have been possible without the excellent collaboration with all of you.

Moreover, I would particularly like to thank my office mate Lingni for her constant willingness to help and for our numerous ice cream breaks. It always was a pleasure to share the office with her. My thanks also go to our secretary Sabine and our system administrator Quirin for their continuous support and valuable assistance with any questions and problems that came up.

I also would like to thank the Studienstiftung des deutschen Volkes e.V. and the TUM Graduate School for supporting this work and offering many opportunities to establish interdisciplinary contacts and acquire insights and competences beyond the boundaries of my own dissertation project.

Finally, I would like to thank my family most sincerely for supporting me in everything I do. I also like to thank most warmly my friends for their support and friendship and I am deeply grateful for the love and support of my husband Franz that makes this all possible.

Contents

List of Figures xv

List of Tables xix

I Introduction & Motivation 1

1 Introduction 3

1.1 Semantic Image Analysis 3

1.2 Literature Overview 6

1.2.1 Contour Matching Techniques 6

1.2.2 Semantic Segmentation 7

1.3 Outline of the Thesis 11

2 Contributions 13

2.1 Own Publications 15

3 Convex Variational Methods 19

3.1 Semantic Image Segmentation 19

3.1.1 Problem Definition 19

3.1.2 Optimization Problem 19

3.2 Total Variation Regularization 20

3.2.1 Total Variation 20

3.2.2 Rudin-Osher-Fatemi Model 21

3.2.3 Functions of Bounded Variation 23

3.3 Convexity 23

3.3.1 Convex Relaxation 24

3.3.2 Existence of Solutions 25

3.3.3 Binarization 26

3.4 Extremality Condition 26

3.4.1 Gâteaux Derivative 27

3.4.2 Euler-Lagrange Equation 27

3.4.3 Subdifferential 28

II Own Publications 29

4 The Role of Diffusion in Figure Hunt Games 31
4.1 Introduction 32
4.2 Formalization 35
4.2.1 Matching Cost 36
4.2.2 Optimization Via a Genetic Algorithm 38
4.3 The Role of Diffusion 39
4.3.1 Spreading the Edge-Information 40
4.3.2 Improved Localization and Speed-Up 42
4.4 Diffusion Methods 42
4.4.1 Averaging 43
4.4.2 Distance Function 43
4.4.3 v-Transform 45
4.4.4 Erosion Followed by Averaging 45
4.4.5 Overview 47
4.5 Experimental Results 47
4.5.1 Propagation of the Contour Information 47
4.5.2 Proof of Concept 55
4.5.3 Diverse Target Figures 60
4.6 Summary and Conclusion 61

5 Midrange Geometric Interactions for Semantic Segmentation 63
5.1 Introduction 64
5.1.1 Related Work 66
5.1.2 Contributions and Organization 68
5.1.3 Extensions and Improvements over the Previously Published Variant of our Model 68
5.2 Variational Multi-label Segmentation 69
5.2.1 The Multi-label Optimization Problem 69
5.2.2 Choices of Appearance Models 70
5.3 The Novel Midrange Geometric Priors 71
5.3.1 A Continuous Formulation of the Dilation 72
5.3.2 Introducing Midrange Geometric Constraints 73
5.3.3 Structuring Elements 73
5.3.4 Specification of the Penalty Matrix 76
5.3.5 Real-World Examples 77
5.4 Integrating the Geometric Constraints into a Convex Problem 80
5.4.1 A Convex Relaxation of the Midrange Geometric Constraints . 80
5.4.2 Implementation 82
5.5 Experiments and Results 82
5.5.1 Part-based Articulated Objects: Humans, Animals, Clothes . 83

5.5.2 Part-based Rigid Objects . . . 87
5.5.3 Scene Segmentation . . . 87
5.5.4 Analysis of Failure Cases . . . 92
5.5.5 Preventing Ghost Labels . . . 92
5.5.6 Benchmark Evaluation . . . 95
5.5.7 Runtimes . . . 101
5.6 Conclusion . . . 102

6 Optimizing the Relevance-Redundancy Tradeoff for Efficient Semantic Segmentation 103
6.1 Introduction . . . 104
6.1.1 Semantic Segmentation and Feature Selection . . . 104
6.1.2 Related Work . . . 106
6.1.3 Contributions . . . 106
6.2 The Feature Set . . . 107
6.3 Feature Ranking and Selection for Object Recognition . . . 108
6.3.1 Feature Ranking . . . 108
6.3.2 Automatic Feature Selection . . . 109
6.3.3 Implementation . . . 110
6.4 Which and How Many Features? . . . 110
6.4.1 Which Features . . . 111
6.4.2 How Many Features . . . 113
6.5 Experimental Results . . . 113
6.5.1 Significantly Improved Runtime . . . 114
6.5.2 Competitive Classification and Segmentation Results . . . 114
6.6 Conclusion . . . 116

7 Interactive Multi-label Segmentation of RGB-D Images 117
7.1 Introduction . . . 118
7.2 Variational Interactive Segmentation of RGB Images . . . 120
7.2.1 Multi-label Segmentation . . . 120
7.2.2 Convex Relaxation . . . 120
7.2.3 Likelihood Estimation Based on User Scribbles . . . 121
7.3 From RGB to RGB-D Images . . . 121
7.3.1 Pre-Processing the Depth Image . . . 121
7.3.2 Depth as an Additional Color Channel . . . 122
7.3.3 Active Scribbles . . . 122
7.3.4 Revised Pixel Distance by Depth Values . . . 123
7.3.5 The Novel Formulation . . . 124
7.4 Implementation . . . 125
7.5 Experimental Results . . . 125
7.5.1 Experimental Data . . . 125

7.5.2 Depth Information is Crucial 126
7.5.3 Impact of the Proposed Components 128
7.6 Conclusion 129

8 Flow and Color Inpainting for Video Completion 131
8.1 Introduction 132
8.1.1 Related Work 132
8.1.2 Contributions 134
8.2 Interactive Mask-Definition 134
8.2.1 Scribble Relocation via Optical Flow 135
8.2.2 Segmentation According to Nieuwenhuis and Cremers 135
8.3 Flow and Color Inpainting for Video Completion 136
8.3.1 Inpainted Flow for Temporal Coherence 136
8.3.2 Flow Inpainting 137
8.3.3 Exemplar-Based Inpainting 137
8.3.4 Flow Preserving Distance Function 138
8.3.5 Overview of the Algorithm 139
8.4 Experiments and Results 139
8.4.1 Implementation and Runtime 142
8.5 Conclusion 142

III Conclusion 143

9 Summary 145

10 Limitations and Future Research 149

Own Publications 151

Bibliography 153

List of Figures

1.1 Semantic labeling and interpretation of a typical road traffic scene 4
1.2 How many butterflies are depicted in the mandala? 5

2.1 Studied input data 14

3.1 Segmentation of the image plane 20
3.2 TV regularization 22
3.3 Convex versus non-convex function 24
3.4 Illustration of the definition of a convex function 24

4.1 Samples of figure hunt games 33
4.2 Erosion followed by averaging 35
4.3 Matching cost 36
4.4 Optimal match 37
4.5 Expressive energy by diffusion 40
4.6 Comparison of matching costs 41
4.7 How many bees? 42
4.8 Exemplary distance transform 44
4.9 Distance transform 44
4.10 v-transform 45
4.11 Comparison of introduced diffusion approaches 47
4.12 All hypotheses of 1000 independent runs 49
4.13 Matching cost corresponding to 84 distinct locations 50
4.14 Several dissimilar objects found 51
4.15 Three cars of the same type are detected 51
4.16 Coarse-to-fine approach 51
4.17 Distinctive jump in the energy 52
4.18 Hidden cloverleaf 53
4.19 Black spots at intersections of the triangles 53
4.20 Diffusion of target figure and/or illustration 54
4.21 Collection of hangers 55
4.22 Possible target figures to search for 55
4.23 Hypotheses obtained by spreading the edge information via diffusion . . 56
4.24 Several target figures 56
4.25 Robustness to pose variations 57

4.26 Illustrations including figures of different scales 58
4.27 Robustness to scale variations 58
4.28 Robustness to scale variations 58
4.29 Results of 60 runs with different diffusion approaches 60
4.30 Closest match 61
4.31 Hypotheses for pi 61
4.32 A yet hidden symmetry appears 61
4.33 Desired results 62

5.1 Midrange geometric constraints improve semantic segmentation results . 65
5.2 Introducing midrange geometric priors 72
5.3 Horizontal, vertical and sparse structuring elements 74
5.4 Impact of structuring elements 74
5.5 SEs and penalty matrix A learned on the Penn-Fudan training set 75
5.6 Penalty matrix A learned on the MSRC training data 76
5.7 Penalty matrix A and corresponding structuring elements 77
5.8 Effect of different sets $\mathcal{S}_i$ 78
5.9 Penalization of the proximity to improve the color-based segmentation . 79
5.10 Effect of novel geometric constraints 79
5.11 Color legend 83
5.12 Part-based articulated objects such as humans or clothes 84
5.13 Part-based articulated objects such as animals or humans 85
5.14 Improved results on the Penn-Fudan dataset 86
5.15 Part-based rigid objects such as cars or bicycles 88
5.16 Improved results on the MSRC benchmark 89
5.17 Improved labeling of facades on the eTRIMS benchmark 91
5.18 Segmentation of corridors 93
5.19 Analysis of failure cases 94
5.20 Midrange geometric priors prevent ghost labels 94
5.21 Ground truth and trimap segmentations 96
5.22 Pixel-wise classification error on the MSRC benchmark 100
5.23 Minimizing runtime 102

6.1 Impact of features on the classification accuracy 105
6.2 The feature set 108
6.3 Qualitative classification and segmentation results 115

7.1 Depth information significantly improves the segmentation result 119
7.2 Exemplary RGB-D input, scribbles, ground truth and trimap labeling . . 122
7.3 Recovering the scene geometry with depth information 123
7.4 Depth information improves the segmentation 127
7.5 Depth yields less user input 127

7.6 Each of the proposed components improves the segmentation 129

8.1 Sketched approach 133
8.2 Automatic segmentation by scribble propagation via optical flow 135
8.3 Inpainted flow ensures temporal consistency 137
8.4 Transition comparison 140
8.5 Comparison to Motion SSD dataset 140
8.6 Comparison to state-of-the-art methods 141

List of Tables

1.1 Overview of included publications 12

2.1 Journal articles 17
2.2 Conference publications with oral presentation 17
2.3 Additional publications 17

4.1 Parameters and average runtime 48
4.2 Average results 54
4.3 Smallest deviation with erosion and averaging 60

5.1 Penn-Fudan benchmark scores 96
5.2 Confusion matrix on the Penn-Fudan dataset 97
5.3 MSRC benchmark scores 99
5.4 eTRIMS benchmark scores 101
5.5 Improved score for the task of geometric class labeling 101
5.6 Average runtimes 101

6.1 Ranked features 112
6.2 Comparison of runtimes 114
6.3 Quantitative results 115

7.1 Comparison of the dice scores 128

8.1 Comparison of runtimes 142

Part I

Introduction & Motivation

Chapter 1

Introduction

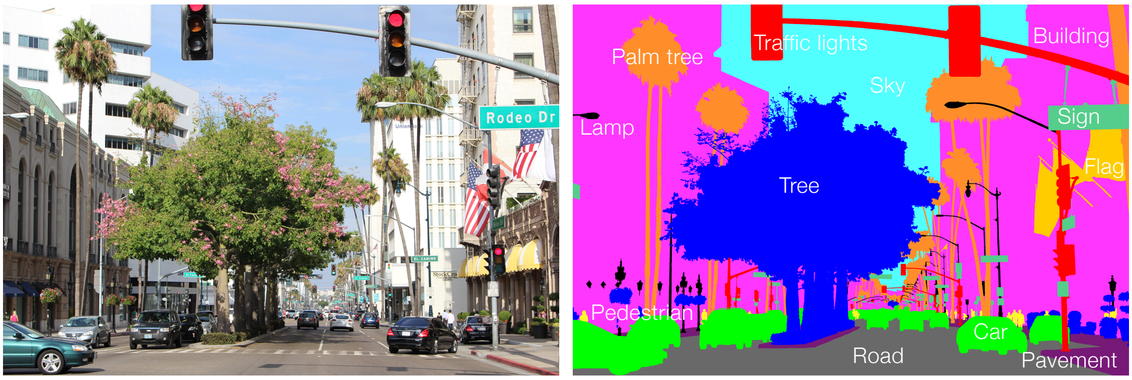
1.1 Semantic Image Analysis

The word *semantics* finds its origins in the ancient Greek word σημαντικός (semantikos), the study of meaning. In the context of semantic image analysis, the goal is to determine the regions and objects of which the image is composed and to label them according to their meaning.

Humans are using semantic analysis in their everyday life. From experience, they learn the meaning of objects. They perceive and memorize which objects are likely to occur together in a scene and learn to categorize their surrounding. Kids have, for example, already acquired the intuition that the ball which is used in a soccer stadium is called a soccer ball, whereas on a basketball court the team is playing with a basketball. Moreover, small kids already know the meaning of things like 'water', 'banana' or 'cat' and are able to classify the objects which are depicted in picture-books. They are able to distinguish an 'apple' from a 'pear' although both have a similar size, color and even their shape is comparable.

Humans need this ability of precise cognition to survive in the everyday life. Figure 1.1 shows a typical road traffic scene where cars and pedestrians are moving along next to each other. When a person looks at this image, he or she immediately starts with a semantic image analysis (compare Figure 1.1 b): "The image shows a road traffic scene. The traffic lights are red. Several green street signs indicate the names of the intersecting streets. Some pedestrians are walking on the pavements. The road is separated by a traffic island. High palm trees are lining the streets." Signs, road markings and traffic lights help to guide the road users. Still, it is quite hard to fully realize all actions in order to avoid dangers.

To support the humans in complicated situations, semantic image analysis gained increasing interest in recent years. Some examples include: driver assistance systems assisting car drivers with the recognition of signs, traffic lights or pedestrians; medical vision systems supporting surgical operations; or industrial applications



a) Observed scene b) Labeled scene

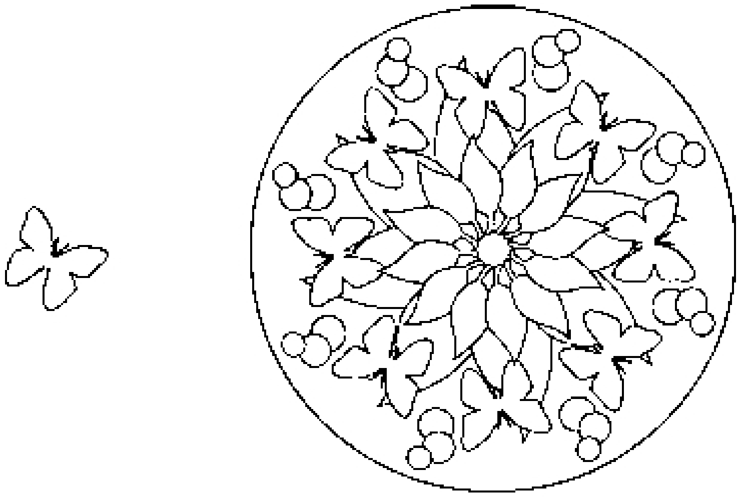
Figure 1.1: **Semantic labeling and interpretation of a typical road traffic scene.** "Multiple cars are driving on the road guided by signs, road markings and traffic lights. Pedestrians are walking on the pavements between street lamps and palm trees." Humans are confronted with such scenes in their daily life. Drivers as well as pedestrians have to analyze the road traffic and interpret the observed scene to ensure road safety.

facilitating quality control.

In general, human-machine communication systems and robots assisting humans in their daily life became a very popular field of research in the recent past. To teach a computer to analyze an image and to semantically interpret the scene is a challenging task. Extensive research is required to develop advanced algorithms which are able to imitate the human perception. Some specific tasks with clear requirements are already solved. An example is the automated speed limit sign recognition which is available in production vehicles. This task is clearly limited in the sense that only a given number of different speed limit signs exist and that they all have a characteristic size, shape and color. Thus, the algorithm is specifically designed for the recognition of these particular objects.

As opposed to this, a reliable and efficient automated recognition of pedestrians is still an open problem. The reason is that pedestrians cannot be described by clear criteria. They can be children or adults, they have different hair and skin colors and they wear clothes with all kinds of patterns and styles. Regarding the *shape*, most of the people on the streets are pictured as elongated regions in images. However, when the person bends down to a child the captured shape corresponds more to a bulb. Hence, in contrast to the speed limit signs which can be specified in detail, it is a very complex task to define general criteria matching the perception of pedestrians.

An object that is easier to describe is a car. Although sizes and colors differ, cars are built from similar components: 'license plate', 'wheel', 'window', 'light', 'door', 'mirror', *etc.* Therefore, an algorithm can be tuned to recognize the single components and to draw a conclusion about the object category. Still, to receive



a) Target figure b) Mandala

Figure 1.2: **How many butterflies are depicted in the mandala?** Humans will most likely answer ‘eight’ whereas an algorithm might only find one resemblant butterfly. Namely, the butterfly at the top on the right.

correct object classifications, various additional factors have to be considered for the analysis of outdoor scenes. A major challenge are the lighting conditions varying with weather and time and affecting the visibility of the scene and the color appearance. During the night, traffic lights might be turned off and trees that were shining light green during the day may appear almost black. Moreover, also the season of the year has to be kept in mind. Typically gray streets can turn white when covered with snow during winter. Thus, if an algorithm is tuned to recognize white color on the road as road markings, it will completely fail during a winter with snow-covered streets. Humans learn to correctly interpret such situations by memorizing experiences starting from their childhood. This knowledge has to be included in the computing system to allow for a reliable semantic image analysis in real-life scenarios.

The development of formalisms which enable the computing system to get close to the human perception is still a challenging task. Computers have an excellent computational power, however, necessary qualities for scene understanding are missing. The human brain constantly builds connections that help to improve the cognitive skills. Computing systems, in contrast, only have the connections which are provided by the respective algorithms. If a person is asked, how often the butterfly in Figure 1.2 a) appears in the mandala in Figure 1.2 b), the immediate answer will most likely be ‘eight’. When the same task is demanded from a computing system, the algorithm might output ‘one’. Namely, the butterfly at the top on the right. Only this butterfly is identical to the figure shown in Figure 1.2 a). All others are rotated versions. Thus, if all eight butterflies should be detected, it has to be clearly defined in the algorithm that rotations of the shape are allowed. In contrast to humans, computing systems have to be explicitly trained to identify coherences and transfer their knowledge to a novel problem.

The challenge of computer vision is therefore to develop algorithms that can perceive relations and principles. The development of formalisms including prior knowledge allows the computing system to recognize repeated patterns and repetitive structures. Over the past years, extensive research has been made to derive and refine such algorithms. However, in particular for complex images current methods often yield unsatisfactory results.

In this thesis we extend state-of-the-art image analysis methods to allow for a more efficient and more reliable semantic analysis. We show that diffusion improves the localization of target figures in binary line drawings, as it makes the resulting energy easier to optimize. Then, we focus on the extension of variational segmentation methods towards semantic scene analysis by incorporating suitable prior information in a convex fashion.

1.2 Literature Overview

In the course of the last years, a continuous effort has been made by the research community to design machines imitating human sensory. There has been growing interest in the research of semantic image analysis. In the following literature overview we distinguish between contour matching techniques and semantic segmentation.

1.2.1 Contour Matching Techniques

A commonly addressed problem in the field of image analysis is finding objects in images. Many different methods were proposed to approach this problem. Among others, contour matching techniques became a popular choice for finding an object's position in a given image. The literature can be split into two main categories, namely the search for *deformable* and the search for *rigid* templates.

Deformable templates are particularly popular for handwritten numeral recognition (see *e.g.*, the works of Burr [36] and Jain and Zongker [74]). In 1997, Beveridge and Riseman [26] studied rigid templates for solving 2D line matching problems. They applied local search to find the best match of the rotated, translated and scaled template with the illustration. Both, the template and the illustration are represented as sets of 2D straight line segments. In 1962, Hough [72] presented a smart technique for the detection of shapes that can be expressed in some parametric form. Later in 1981, Ballard [19] proposed the generalized form of the Hough Transform. When template or illustration, however, contain embedded shapes the Hough Transform is not applicable. Another well known contour matching technique is chamfer matching [21]. The algorithm matches edges from two different images by applying a distance transform to the illustration. This transform propagates local information to neighboring areas such that each pixel value denotes the distance to the nearest contour pixel. *I.e.*, the binary image is converted into a non-binary one.

In our work [8], we proposed to replace the distance transform with a more informative diffusion field which diffuses the image while retaining the contour information. Moreover, we experimentally demonstrated that transforming the template rather than the illustration can be beneficial in certain settings.

1.2.2 Semantic Segmentation

Segmentation of images based on semantics is a significant component for scene understanding, surveillance systems and 3D scene reconstruction (see *e.g.*, [30, 70, 139]). The accuracy of the segmentation algorithms heavily depends on the imposed prior knowledge and the quality of the appearance model. Extensive research has been made to achieve precise and reliable semantic segmentations.

Regularization Term

The integration of label priors into multi-label segmentation approaches has been a common means of improving segmentation algorithms. Numerous works focused on the development of sophisticated regularizers. In general, one can differentiate *global* and *local* label priors.

Global constraints impose prior knowledge on the union of all pixel labels in the image. An example is the minimum description length (MDL) prior of Leclerc [86], which penalizes the number of labels appearing in the segmentation result. Its first continuous formulation was given by Zhu and Yuille [175], and a convex relaxation by Yuan *et al.* [173].

Shape priors. Global object shape priors were proposed in Cremers *et al.* [46] and Lempitsky *et al.* [89]. Both consider binary image segmentation tasks and include knowledge about the object's shape in the segmentation process.

Connectivity priors. State-of-the-art segmentation approaches commonly use a length regularization that suppresses small structures and therefore tend to cut off thin elongated structures in the image. To overcome this shrinking bias, connectivity priors were recently developed by Vicente *et al.* [162], Nowozin and Lampert [113] and Stühmer *et al.* [150].

Proportion priors. To preserve the size ratios of segmented regions across multiple images, Nieuwenhuis *et al.* [110] introduced so-called proportion priors. The constraints relate different region sizes and restrict the relative size of, *e.g.*, body or object parts.

Volume constraints. Klodt and Cremers [76] introduced the concept of moment constraints for interactive image segmentation. They demonstrated that constraints on moments of different order can be imposed as convex constraints within the optimization framework. In particular, the lower-order moments correspond to the overall volume, the centroid, and the variance or covariance of the shape.

Hierarchical constraints. Another type of global constraints for semantic segmentation are hierarchical constraints, which were introduced by Delong *et al.* [51] and Souiai *et al.* [144]. These constraints penalize the co-occurrence of objects from different scene contexts. For example, the co-occurrence of indoor objects, such as 'desk' or 'chair' and outdoor objects such as 'cow' or 'car', is penalized.

Ordering constraints. For the task of geometric scene labeling, ordering constraints were introduced into multi-label optimization. Liu *et al.* [93] focused on a specific five-part model including the regions 'sky', 'ground', 'left', 'right' and 'center'. Felzenszwalb and Veksler [57] generalized the five regions layout and introduced the tiered layout. In 2011, Strekalovskiy and Cremers [147] unified and generalized existing formulations such as the five-regions and the tiered layout by proposing a spatially continuous framework for label order constraints.

Co-occurrence priors. Co-occurrence priors were suggested by Ladicky *et al.* [82] and Souiai *et al.* [145] to impose prior knowledge on label combinations that are likely to co-occur in a given image. For example, the knowledge that a piano and a horse are very unlikely to appear in the same image. The approach by Ladicky *et al.* [82] is formulated in the discrete setting and suffers from metrication errors which often occur in discrete optimization: region boundaries tend to run either horizontally, vertically or diagonally (compare [111]). In contrast, Souiai *et al.* [145] integrated the co-occurrence priors into a single convex continuous optimization approach leading to smooth boundaries.

Local constraints In contrast to global label priors, local priors penalize labels by means of distance functions on adjacent labels. A simple prior is, *e.g.*, the linear label distance $d(i, j) = c|i - j|$ for two labels i and j and a constant c. In the continuous setting, this formulation corresponds to total variation regularization. In the discrete case, convex distance functions on ordered label spaces can be minimized globally optimally by Ishikawa's approach [73].

Metric priors. A popular metric prior is the Potts model [123], which penalizes label changes of neighboring pixels. However, in the case of metric distance functions global optimality is in general not given for more than two labels. To solve the multi-label problem, general metric label distance functions are optimized in [32, 40, 88].

Nonmetric priors. For segmentation, semi-metric distance functions are indispensable to handle label distances without imposing the triangle inequality. A convex optimization approach for such distance functions was given by Strekalovskiy *et al.* [149]. Based on these nonmetric distances the authors formulated a co-occurrence prior on directly neighboring pixels. The drawback of this approach is that the strong locality of the prior allows for regions to appear very close to each other despite high co-occurrence penalties. In particular, since the distance function does not obey the triangle inequality, costs of direct label transitions can be reduced by taking a 'detour' over a third unrelated but less expensive label. This leads to the undesired introduction of thin artificial 'ghost regions'.

What has been less explored so far are *relative spatial priors* imposing constraints on spatial relations between different objects. Gould *et al.* [65] proposed relative location priors. The authors formulated a two stage optimization problem. In the first step, superpixels are computed together with an occurrence based label likelihood. Based on the most likely label each superpixel then votes for labels at other superpixels in the image based on the relative location prior. Considering more complex spatial label relationships, we proposed a single stage optimization problem [7] which models presence and relative location likelihoods at the same time. By generalizing the nonmetric priors [149] for arbitrary relative spatial relations, we were able to avoid the thin artificial ghost regions.

Data Fidelity Term

A precise and accurate appearance model is a central component for every image segmentation task. During the last decade, sophisticated and powerful data fidelity terms have been developed. Depending on the application different requirements have to be met. We therefore distinguish between *non-interactive* and *interactive* approaches.

Non-Interactive Dense object detection approaches, like *e.g.*, the works of Ladicky *et al.* [83] and Shotton *et al.* [139], focus on detecting objects at a pixel level and already provide a preliminary segmentation. In contrast, conventional object detectors deal with the task of finding bounding boxes around each object [49, 95, 164]. The major challenge is to find the most representative features to distinguish dissimilar objects in terms of their shape and textural differences.

In 2001, Viola and Jones [164] presented simple but robust Haar-like features for real-time face detection. These Haar-like features are simple to implement, computationally low cost and very accurate in capturing the shape of objects. Lowe [95] presented a distinctive scale-invariant feature transform (SIFT) and Dalal and Triggs [49] presented so-called histograms of oriented gradients (HOG) which are computationally more expensive. However, these proposed methods are mostly used in the context of sliding window techniques which detect objects in a bounding box.

In 2006, Shotton *et al.* [139] proposed texture-layout filters based on textons which jointly model patterns of texture and their spatial layout for dense object detection. Moreover, Ladický *et al.* [84] combined different features for unary pixel classification by using Joint Boosting [159]. Their approach, however, is sensitive to a large set of parameters. Fröhlich *et al.* [59] proposed an iterative approach for semantic segmentation of a facade dataset. They learned a single random forest and incrementally added context features derived from coarser levels. This approach uses millions of features and refines the semantic segmentation of the scene iteratively. Hermans *et al.* [70] discussed 2D semantic segmentation for RGB-D sensor data in order to reconstruct 3D scenes. They used a very basic set of features and learned random forests for object classification.

None of the above approaches, however, provides a justification for the chosen set of features. In our publication [9], we therefore introduced a framework which analyzes a given feature set, automatically selects a small number of the most significant features and efficiently estimates the class probabilities. The data fidelity term is then computed as the negative logarithm of the class probabilities. Our proposed fully automatic algorithm achieves state-of-the-art semantic classifications and segmentations at drastically reduced computation time.

Interactive Fully *automatic* image segmentation methods are usually tailored to a very specific task. Examples include the approaches for indoor segmentation by Silberman and Fergus [140] as well as the facade segmentation approach by Teboul *et al.* [154]. Interactive methods, in contrast, can be applied to various tasks and have recently attracted a lot of interest. A common way to develop general purpose segmentation tools is to incorporate user input [92, 109, 163]. These approaches demand the user to *interact* and specify the objects and regions which should be segmented.

The user interaction can be designed in various different ways. The most popular input modalities are bounding boxes [92, 128, 163], contours [13, 27] and user scribbles [22, 85, 109]. In terms of user scribbles, the user indicates the pixels which belong to a certain object by drawing lines across the image. The indicated pixel colors and locations can then be used to learn the color distributions and to compute the appearance model.

Interactive methods are extremely beneficial for the segmentation of medical image data, see *e.g.*, the publications of Boykov and Jolly [31] and Lombaert *et al.* [94]. Moreover, they are very popular in the field of video processing, *e.g.*, image sequence segmentation [110] or video completion [10]. For similar composed scenes it can be sufficient that the user only scribbles the first frame. The provided input can then be propagated to subsequent frames. *E.g.*, in their work [85], Lang *et al.* introduced a temporal continuity assumption and showed that it can be used to propagate sparse user input to colorize similar composed videos scenes. For the task of video completion we suggested a semi-automatic procedure to replace the tedious hand-labeling of inpainting regions in all video frames [10]. The user input is required as user scribbles drawn on the first frame of each scene. These scribbles are then automatically relocated throughout the video sequence via optical flow and the inpainting masks are defined by applying the interactive segmentation algorithm proposed by Nieuwenhuis and Cremers [108] in a frame-wise manner.

Various works studied foreground/background [27, 91, 92, 163, 166] as well as multi-label [109, 130, 136] segmentation of RGB and medical images. What has been less explored so far is the extension of interactive segmentation methods to RGB-D images. In 2012, Shao *et al.* [136] proposed an interactive multi-label RGB-D segmentation formulation, particularly designed for the application of furniture segmentation. To further extend interactive methods to RGB-D images, we extended the spatially varying color distributions in a plane [108] to be volumetrically varying in 3D [5].

1.3 Outline of the Thesis

This cumulative thesis is structured into three parts.

Part I provides an introduction and motivation of the thesis. In Chapter 1 the research topic is motivated and a review of relevant literature is provided. Chapter 2 summarizes the contributions of this thesis and provides an overview of all research papers that were published during this thesis. Chapters 3 provides an introduction to the methodology employed in this thesis.

Part II includes the five peer-reviewed publications summarized in Table 1.1 that were published in the context of this thesis. Chapter 4 presents the journal article [8] tracing out target figures hidden in teeming figure pictures. Chapter 5 presents the journal article [7] introducing midrange geometric priors for variational semantic segmentation. The conference publications [5, 9] in Chapters 6 and 7 focus on the data term and present an *automatic (non-interactive)* respectively *interactive* method for the computation of accurate color descriptions. The conference publication [10] presented in Chapter 8 shows an application for video analysis.

Part III concludes the thesis with a summary, a discussion of the results, the limitations and future research opportunities.

Table 1.1: **Overview of included publications.** Part II includes the following five peer-reviewed publications that were published in highly ranked journals and international conferences.

Chap.	Publication		Status
4	[8]	J. Diebold, S. Tari, and D. Cremers The Role of Diffusion in Figure Hunt Games *Journal of Mathematical Imaging and Vision (JMIV)*	Published in 2015
5	[7]	J. Diebold, C. Nieuwenhuis, and D. Cremers Midrange Geometric Interactions for Semantic Segmentation *International Journal of Computer Vision (IJCV)*	Published in 2015
6	[9]	C. Hazirbaş, J. Diebold, and D. Cremers Optimizing the Relevance-Redundancy Tradeoff for Efficient Semantic Segmentation *Scale Space and Variational Methods in Computer Vision (SSVM)*	Published in 2015
7	[5]	J. Diebold, N. Demmel, C. Hazirbaş, M. Möller, and D. Cremers Interactive Multi-label Segmentation of RGB-D Images *Scale Space and Variational Methods in Computer Vision (SSVM)*	Published in 2015
8	[10]	M. Strobel, J. Diebold, and D. Cremers Flow and Color Inpainting for Video Completion *German Conference on Pattern Recognition (GCPR)*	Published in 2014

Chapter 2

Contributions

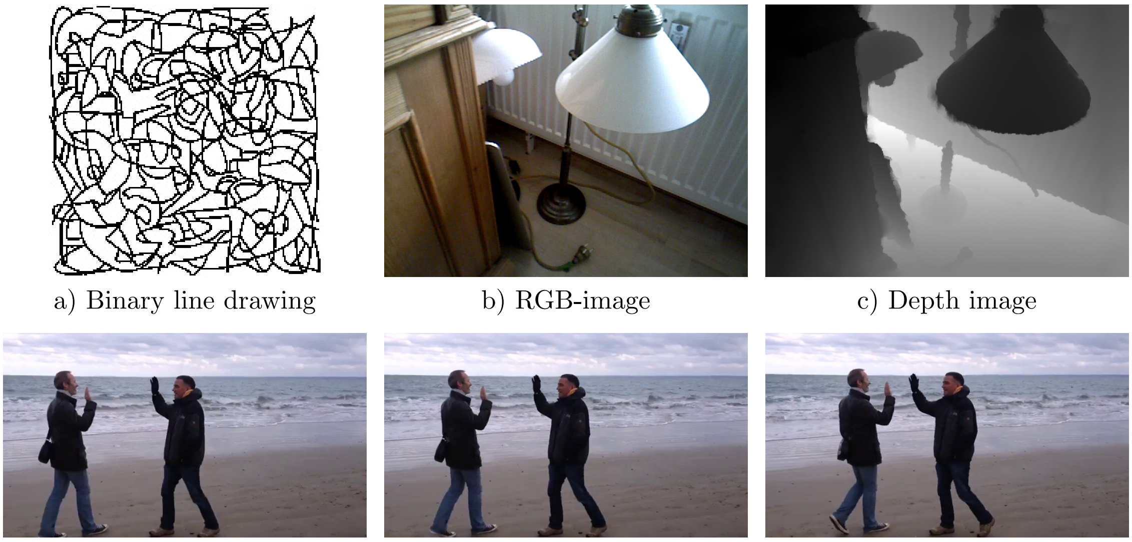
This thesis summarizes the work presented in [5, 7, 8, 9, 10], which is the result of the joint work with Nikolaus Demmel, Michael Strobel, Caner Hazırbaş, Claudia Nieuwenhuis, Michael Möller, Prof. Sibel Tarı, and Prof. Daniel Cremers. All included papers are peer-reviewed publications and were published in highly ranked journals and international conferences.

In the field of computer vision and image analysis multiple kinds of input data might be given. In this thesis we study the analysis of the data illustrated in Figure 2.1: Chapter 4 covers the analysis of binary line drawings [8], Chapters 5 to 7 focus on the semantic analysis of RGB(-D) images [5, 7, 9] and Chapter 8 investigates the analysis of video sequences [10].

A popular way to approach the study of binary line drawings are contour matching techniques. We particularly address the search task of tracing out target figures hidden in teeming figure pictures known as *figure hunt games* [8]. An example is illustrated in Figure 1.2 and in Figure 2.1 a). We experimentally demonstrate that the key idea is to diffuse the information localized on a contour to a plane in which the contour is embedded. Diffuse representations can be obtained in a variety of ways. Particularly suited to the considered task, we propose a diffuse representation which diffuses the image while retaining the contour information. Moreover, we introduce a coarse-to-fine strategy to speed up the search process. Extensive evaluations show that we can handle various illustrations and diverse target figures.

For the analysis of RGB(-D) images and video sequences, a semantic segmentation of the image plane is a central component. Since the accuracy of the segmentation algorithms heavily depends on the imposed prior knowledge and the quality of the appearance model, in this thesis we study the derivation of a novel regularizer [7] and also discuss the development of a precise and accurate data fidelity term for non-interactive [9] as well as interactive [5] applications.

We incorporate prior knowledge into the segmentation algorithm by introducing *midrange geometric constraints* for variational semantic segmentation [7]. The proposed constraints allow to discourage the occurrence of labels in the vicinity of each



a) Binary line drawing

b) RGB-image

c) Depth image

d) Video sequence

Figure 2.1: **Studied input data.** This thesis studies the semantic analysis of multiple kinds of input data: a) Binary line drawings, b,c) RGB(-D) images and d) video sequences.

other, *e.g.*, ‘wolf’ and ‘sheep’. It is up to the user to specifically define the spatial extent of the constraint between each two labels. We call these constraints ‘midrange’ since they generalize both global and local co-occurrence priors to co-occurrence priors with arbitrary spatial relationships. By capturing richer semantic information on spatial relations, the proposed constraints allow to obtain improved segmentation results. Our experimental results show that the novel constraints are beneficial for many segmentation scenarios, such as the segmentation of scenes, part-based articulated or rigid objects. Since our definition of neighborhood regards a larger number of pixels, our approach does not suffer from thin artificial ghost regions, as opposed to purely local priors.

Beside the regularization term, the appearance model is an essential component for every image segmentation task. Depending on the application a non-interactive or an interactive approach is desired. In terms of non-interactive methods a major challenge is the selection of the feature set. State-of-the-art methods usually, however, do not provide any justification for their chosen features. We therefore investigate a systematic information-theoretic feature analysis method [9]. The central idea is to only choose the most significant features by optimizing the relevance and redundancy tradeoff of the respective feature set. Integrated in a variational multi-region segmentation approach, we evaluate our method on five popular benchmarks. Our experimental results demonstrate that the right selection of the feature set allows for state-of-the-art semantic classifications and segmentations at drastically reduced computation time.

Whereas non-interactive approaches are mostly tailored to a very specific problem, interactive methods can be applied to more general segmentation tasks. In this thesis we additionally investigate the application of interactive segmentation, namely interactive RGB-D multi-label segmentation [5]. We extend the concept of spatially varying color distributions proposed for RGB images in three different ways: a) We introduce *active scribbles* to overcome the problem of non-uniformly distributed user scribbles. b) We incorporate depth information to capture the real scene geometry. c) We consider the depth as an additional color channel. We show that the incorporation of active scribbles in the concept of spatially varying color distributions on RGB images already improves the segmentation results. Moreover, we demonstrate that the additional depth information leads to reliable segmentations with significantly less user input. Overall, the experimental evaluations on our benchmark dataset point out that the proposed volumetrically varying color distributions in 3D yield much more distinct color descriptions and thus better segmentation results than spatially varying color distributions in a plane.

In terms of video data, we particularly consider the task of video completion. At first, we introduce spatially varying color distributions to replace the tedious hand-labeling of inpainting regions in all video frames [10]. Rather than hand-labeling the inpainting region in every single frame, we demonstrate a flow-based propagation of the user input followed by an automatic segmentation step. Then, we additionally propose a framework for temporally consistent video completion by a combination of color- and flow-based inpainting. Our proposed semi-automatic technique requires substantially less user input and allows for a temporal consistent video completion at drastically reduced runtime compared to competing approaches.

2.1 Own Publications

Tables 2.1, 2.2 and 2.3 summarize the peer-reviewed research papers that were published during this thesis in highly ranked journals and international conferences. The publications are grouped into journal articles (Table 2.1), conference publications with oral presentation (Table 2.2) and additional publications (Table 2.3).

In the course of this thesis two journal articles were published. The article [7] was published in the International Journal of Computer Vision (IJCV) for the Special Issue on Graphical Models for Scene Understanding. It extends the publication [2] which was presented at the ICCV Workshop on Graphical Models for Scene Understanding. Moreover, the article [8] extends the works [3, 4] and was published in the Journal of Mathematical Imaging and Vision (JMIV) for the Special Issue on Scale Space and Variational Methods in Computer Vision.

The publications [4, 9] were selected for an oral presentation at the Conference on Scale Space and Variational Methods (SSVM) and for submission to the Journal of Mathematical Imaging and Vision (JMIV). Furthermore, the research paper [10] was selected for an oral presentation at the German Conference on Pattern Recognition (GCPR).

Beyond that, the book chapter [1] was written in the context of the Workshop *Women in Shape* at the Institute for Pure and Applied Mathematics (IPAM) at the University of California, Los Angeles and the research paper [5] was presented as a poster at the Conference on Scale Space and Variational Methods (SSVM). Furthermore, the research paper [6] was submitted to the International Conference on Computer Vision (ICCV) and achieved the ratings: poster, poster, oral/poster. The final decision will be announced on September 3, 2015.

Table 2.1: **Journal articles.** In the context of this thesis the following peer-reviewed journal articles were published.

	Authors	Title	Publication medium
[7]	Diebold et al.	Midrange Geometric Interactions for Semantic Segmentation	*International Journal of Computer Vision (IJCV)*
[8]	Diebold et al.	The Role of Diffusion in Figure Hunt Games	*Journal of Mathematical Imaging and Vision (JMIV)*

Table 2.2: **Conference publications with oral presentation.** The following double-blind peer-reviewed conference papers were published within the scope of this thesis and selected for an oral presentation.

	Authors	Title	Publication medium
[4]	Bergbauer et al.	Wimmelbild Analysis with Approximate Curvature Coding Distance Images	*Scale Space and Variational Methods in Computer Vision (SSVM)*
[9]	Hazırbaş et al.	Optimizing the Relevance-Redundancy Tradeoff for Efficient Semantic Segmentation	*Scale Space and Variational Methods in Computer Vision (SSVM)*
[10]	Strobel et al.	Flow and Color Inpainting for Video Completion	*German Conference on Pattern Recognition (GCPR)*

Table 2.3: **Additional publications.** In the course of this thesis the following peer-reviewed research papers were published.

	Authors	Title	Publication medium
[1]	Bal et al.	Skeleton-Based Recognition of Shapes in Images via Longest Path Matching	*Research in Shape Modeling*
[2]	Bergbauer et al.	Proximity Priors for Variational Semantic Segmentation and Recognition	*IEEE International Conference on Computer Vision (ICCV), Workshop on Graphical Models for Scene Understanding (GMSU)*
[3]	Bergbauer et al.	Top-down visual search in Wimmelbild	*Proceedings of SPIE, Human Vision and Electronic Imaging XVIII*
[5]	Diebold et al.	Interactive Multi-label Segmentation of RGB-D Images	*Scale Space and Variational Methods in Computer Vision (SSVM)*

Chapter 3

Convex Variational Methods

This thesis investigates convex variational methods for semantic image analysis. To provide a first introduction to the methodology employed in this thesis, this chapter discusses the basic concepts of convex variational methods.

3.1 Semantic Image Segmentation

3.1.1 Problem Definition

Semantic segmentation aims at jointly computing a partitioning of the image plane and a semantic labeling of the various regions in terms of previously learned object classes. Let the input image be denoted by $I\colon \Omega \to \mathbb{R}^c$, mapping the image plane $\Omega \subset \mathbb{R}^2$ to $\mathbb{R}^c$. The dimension c depends on the image type: $c = 1$ for gray scale images, $c = 3$ for RGB images and $c = 4$ for RGB-D images. Moreover, Ω is mapped to $\{0, 1\}$ in the special case of binary line drawings: $I\colon \Omega \to \{0, 1\}$.

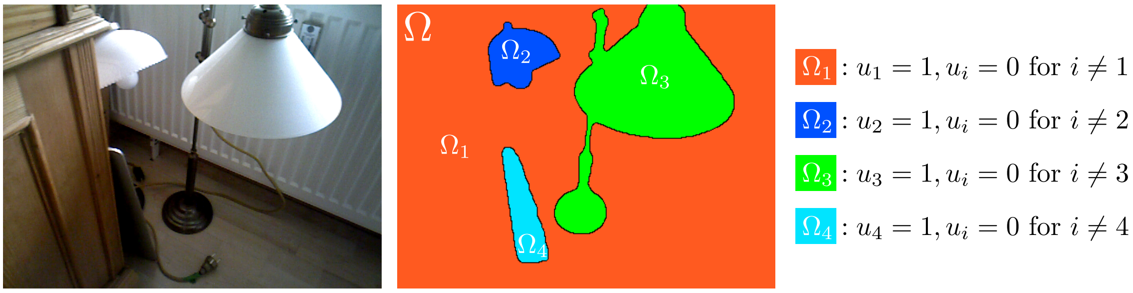
Image segmentation denotes the task of partitioning the image plane Ω into a set of n pairwise disjoint regions Ω_i:

$$\Omega = \bigcup_{i=1}^{n} \Omega_i \quad \text{with} \quad \Omega_i \cap \Omega_j = \emptyset \quad \forall\, i, j = 1, \dots, n \quad \text{with} \quad i \neq j. \tag{3.1}$$

This task is usually solved by computing binary labeling functions $u_i\colon \Omega \to \{0, 1\}$ such that $\Omega_i = \left\{x \;\middle|\; u_i\left(x\right) = 1\right\}$ [41, 42]. Figure 3.1 illustrates a partitioning of the input image a) into four disjoint regions in b). The labeling functions u_i – also known as region indicator functions – take the value one within the related region Ω_i and the value zero outside. Thus, $\sum_{i=1}^{n} u_i\left(x\right) = 1$ for all $x \in \Omega$.

3.1.2 Optimization Problem

For computing the binary labeling functions u_i, $i = 1, \dots, n$ an energy functional can be formulated. In this thesis we focus on the minimization of energy functionals


Ω
Ω_2
Ω_3
Ω_1
Ω_4
Ω_1 : $u_1 = 1, u_i = 0$ for $i \neq 1$
Ω_2 : $u_2 = 1, u_i = 0$ for $i \neq 2$
Ω_3 : $u_3 = 1, u_i = 0$ for $i \neq 3$
Ω_4 : $u_4 = 1, u_i = 0$ for $i \neq 4$

a) Input image

b) Segmentation of the image plane Ω into 4 regions

Figure 3.1: **Segmentation of the image plane** Ω into 4 pairwise disjoint regions Ω_i, $i = 1, \ldots, 4$. The binary labeling functions u_i take the value one within the related region Ω_i and zero outside.

of the following form:

$$E(u_1, \ldots, u_n) = \sum_{i=1}^{n} \big(\mathcal{D}(u_i, I) + \lambda \mathcal{R}(u_i)\big) \tag{3.2}$$

with $u_i \colon \Omega \to \{0, 1\}$ such that $\sum_{i=1}^{n} u_i(x) = 1$ for all $x \in \Omega$ and some positive weighting parameter $\lambda \in \mathbb{R}$. The first term $\mathcal{D}(u_i, I)$ denotes the data fidelity term that models the relationship between the observation I and the solution u_i. The second term $\mathcal{R}(u_i)$ imposes some regularity on the solution. Moreover, certain assumptions on the solution are included by means of this term. The parameter λ regulates the tradeoff between the regularity constraints and the fidelity to the observation.

3.2 Total Variation Regularization

In 1992, Rudin, Osher and Fatemi proposed the popular Rudin-Osher-Fatemi (ROF) model [129] for edge preserving image denoising and introduced the total variation regularization for the solution of problems in the field of computer vision.

3.2.1 Total Variation

The total variation of a function $u \in L^1(\Omega)$ is classically defined by duality as follows [63]:

Definition 1 *(Total variation (TV))* Let $\Omega \subset \mathbb{R}^N$ with $N \geq 2$ be a bounded domain. Then, the *total variation* (TV) of a function $u \in L^1(\Omega, \mathbb{R})$ is defined by:

$$TV(u) := \sup\left\{\int_\Omega u \,\mathrm{div}\xi \, dx \,:\, \xi \in C_0^1\left(\Omega, \mathbb{R}^N\right),\, \|\xi\|_\infty \leq 1\right\}, \tag{3.3}$$

where $\xi = (\xi_1, \ldots, \xi_N)^\top$, $\text{div}\xi = \sum_{i=1}^{N} \frac{\partial \xi_i}{\partial x_i}(x)$, dx is the Lebesgue measure and $C_0^1(\Omega, \mathbb{R}^N)$ the space of continuously differentiable functions with compact support in Ω.

$TV(u)$ is finite if and only if its distributional derivative Du of u is a bounded vector-valued Radon measure in Ω. In this case $TV(u) = |Du|(\Omega)$, also known as $TV(u) = \int_\Omega |Du|$. In particular, for $u \in W^{1,1}(\Omega)$ the total variation of u can be written as [40, 146]:

$$TV(u) = \int_\Omega |\nabla u| dx. \tag{3.4}$$

Moreover, let $u = \chi_A$ be the characteristic function of a subset $A \subset \Omega$ with smooth boundary. Then, the supremum in Equation (3.3) reduces to [18]

$$TV(u) = \sup \left\{ \int_A \text{div}\xi \, dx : \xi \in C_0^1(\Omega, \mathbb{R}^N), \|\xi\|_\infty \leq 1 \right\}, \tag{3.5}$$

and it can be shown that the perimeter of the set A in Ω, is the total variation of its characteristic function [171]. We write:

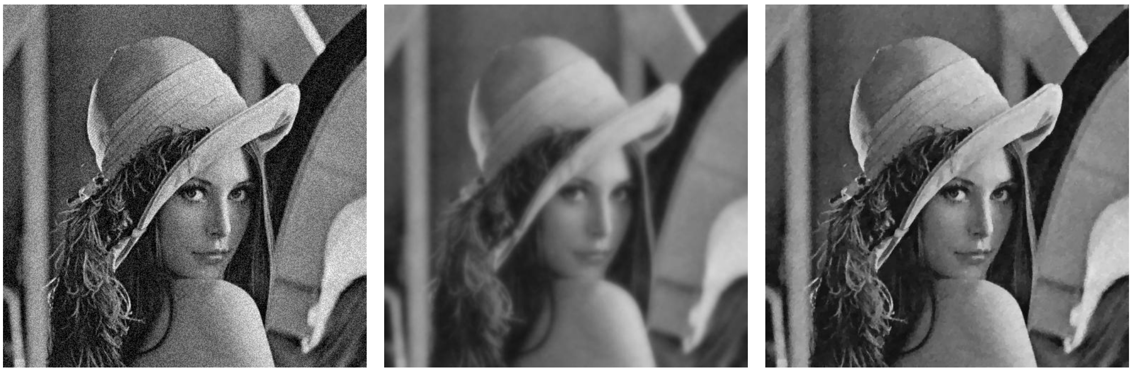
$$TV(u) = Per_\Omega(A). \tag{3.6}$$

In the recent past, various modifications and improvements of the original total variation regularization have been contributed by the community. Several publications have, for example, been devoted to incorporate higher order derivatives in the regularization term [33, 87, 115, 132, 134]. Second order derivatives, for example, were demonstrated to be beneficial in certain cases, *e.g.*, to overcome the staircasing effect, a tendency of TV to produce piecewise constant regions with artificial edges. In particular, the combination of first and second order derivatives has been studied, and we refer the reader to the work of Steidl [146] and the references therein. Another popular subclass of total variation (TV) approaches are nonlocal TV models. In 2009, Gilboa and Osher [61] proposed a framework for nonlocal image and signal processing and demonstrated that nonlocal operators can better handle textures and repetitive structures than local ones.

3.2.2 Rudin-Osher-Fatemi Model

The Rudin-Osher-Fatemi (ROF) model is a famous model for image denoising. In their work [129], Rudin *et al.* suggested the following optimization problem for estimating the denoised version u of a given corrupted image f:

$$\min_u \left\{ \int_\Omega (f-u)^2 \, dx + \lambda TV(u) \right\}, \tag{3.7}$$



a) Noisy input image b) Tikhonov model c) ROF model

Figure 3.2: **TV regularization preserves edges.** The noisy input image a) is denoised by using b) the Tikhonov model and c) the ROF model with $\lambda = 30$. While the ROF model preserves the edge information in the recovered image, the Tikhonov regularization technique not only removes the noise but also blurs prominent structures such as edges.

where $\lambda \in \mathbb{R}$ is a positive weighting parameter and $TV\left(u\right)$ denotes the *total variation* of u given in Definition 1. The first summand in Equation (3.7) can be interpreted as the data fidelity term $\mathcal{D}\left(u, f\right)$ (compare Equation (3.2)) measuring the fidelity to the data. It ensures that the recovered image u is similar to the input image f. The second summand is a smoothing term and corresponds to λ times the regularizer $\mathcal{R}\left(u\right)$. Overall, the optimization problem seeks for a solution u that fits the data and has a small total variation such that noise is removed.

Figure 3.2 compares the results of the ROF model using total variation regularization to the Tikhonov model [156]. The restored image with the Tikhonov technique in Figure 3.2 b) looks oversmoothed. For ease of interpretation, assume $u \in W^{1,1}\left(\Omega\right)$. In contrast to the ROF model using the L_1-norm of the gradient of the image: $\mathcal{R}\left(u\right) = TV\left(u\right) = \int_\Omega |\nabla u| dx$, the Tikhonov model uses an L_2-norm regularization term: $\mathcal{R}\left(u\right) = \int_\Omega |\nabla u|^2 dx$. The L_2-norm regularization removes noise, but does not allow discontinuities of the image. Thus, important structures such as edges are blurred. In contrast, the total variation regularization in c) puts a strong penalization on oscillations and random fluctuations, but at the same time preserves the edge information in the recovered image.

Edges usually indicate the position and shape of individual objects in the image plane. Thus, edge preservation is crucial for most imaging problems. Due to the favorable performance, TV regularization has been extended to a variety of tasks in the field of computer vision and image analysis beyond image denoising.

3.2.3 Functions of Bounded Variation

To optimize the energy functional in Equation (3.2) the feasible set for the labeling functions u_i, $i = 1, \ldots, n$ has to be specified. By definition (compare Section 3.1.1), the functions u_i map the image domain Ω to the binary set $\{0, 1\}$. However, for the task of image analysis not all such functions are useful. As an example, arbitrary oscillations should be prevented whereas informative edges should be preserved. We therefore seek to find a simple functional space for u_i, $i = 1, \ldots, n$ that permits edges but is not too loose to include arbitrary details.

Rudin *et al.* [129] proposed the Banach space of functions with bounded variations (BV space), which allows jumps but also has a sufficient control over arbitrary oscillations [18]:

Definition 2 *(Bounded variation (BV) space)* The space of functions with bounded variation, also known as bounded variation space or BV space is defined as

$$BV\left(\Omega\right) = \left\{u \in L^1\left(\Omega\right) : TV\left(u\right) < \infty\right\}. \tag{3.8}$$

In the optimization problem (3.2), we want to allow for jumps in the indicator functions which correspond to sharp transitions between adjacent regions. Therefore, we specify the feasible set as follows:

$$\min_{u_i \in BV(\Omega;\{0,1\})} \sum_{i=1}^{n} \left(\mathcal{D}\left(u_i, I\right) + \lambda \mathcal{R}\left(u_i\right)\right) \quad \text{s.t.} \; \sum_{i=1}^{n} u_i\left(x\right) = 1 \;\; \forall\, x \in \Omega, \tag{3.9}$$

where $\lambda \in \mathbb{R}$ and the restriction to the binary set $\{0, 1\}$ is given by:

$$BV\left(\Omega; \{0,1\}\right) = \left\{u \in L^1\left(\Omega; \{0,1\}\right) : TV\left(u\right) < \infty\right\}. \tag{3.10}$$

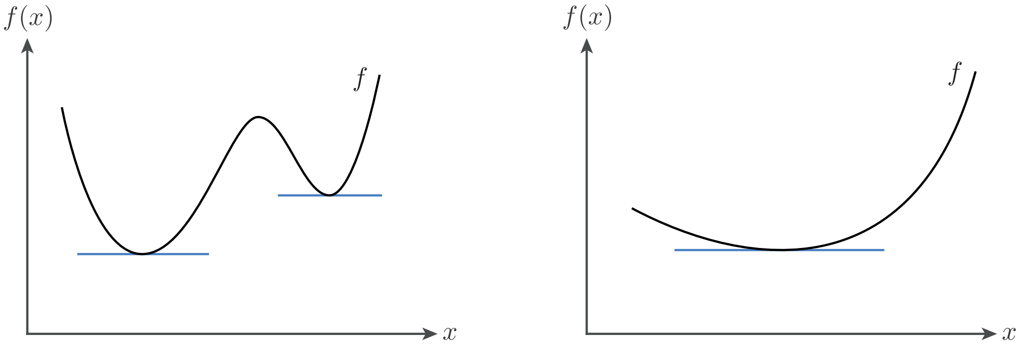
3.3 Convexity

In this thesis we focus on the solution of convex optimization problems. Convex problems are popular in diverse research areas as convexity yields several nice properties of the optimization problem. Figure 3.3 illustrates a) a non-convex and b) a convex function. As indicated in b), each local minimum of a convex function is a global minimum. Thus, the global minimum of a convex optimization problem can be efficiently computed independent of the initialization with a good precision. In contrast, several local minima make the solution of non-convex problems sensitive to the initialization.

Convex minimization aims at minimizing convex functions over convex sets. A convex set, respectively function can be defined as follows [126]:

Definition 3 *(Convex set)* Let $D \subset \mathbb{R}^d$ be a set. Then D is said to be convex if

$$\lambda x_1 + \left(1 - \lambda\right) x_2 \in D \quad \forall\, x_1, x_2 \in D \text{ and } \forall\, \lambda \in \left(0, 1\right).$$



a) Several local minima b) Local minima are global minima

Figure 3.3: **Convex versus non-convex function.** In contrast to a non-convex function, any local minimum of a convex function is also a global minimum.

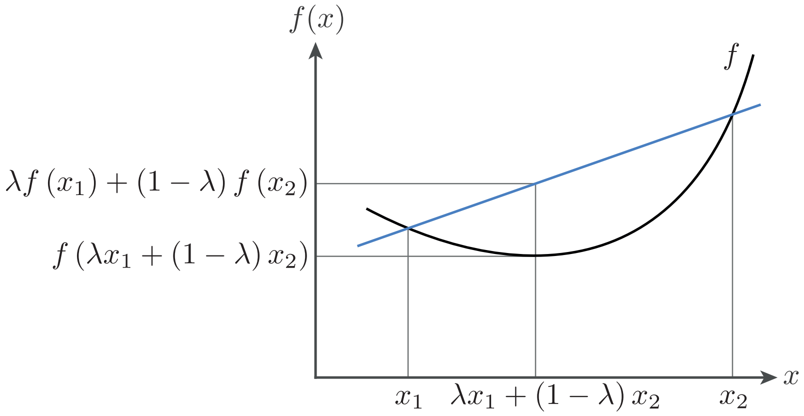

$f(x)$
f
$\lambda f(x_1) + (1-\lambda) f(x_2)$
$f(\lambda x_1 + (1-\lambda) x_2)$
x
x_1
$\lambda x_1 + (1-\lambda) x_2$
x_2

Figure 3.4: **Illustration of the definition of a convex function f.** By definition, the evaluation of a convex function f at any convex combination of two points x_1 and x_2 is less or equal the same convex combination of $f(x_1)$ and $f(x_2)$. In other words, the line segment connecting $(x_1, f(x_1))$ and $(x_2, f(x_2))$ is always located above the graph of f.

Definition 4 *(Convex function)* Let $f\colon D \to \mathbb{R}$ be a function, where D is a convex set. Then f is convex on D if and only if

$$f(\lambda x_1 + (1-\lambda) x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \quad \forall\; x_1, x_2 \in D \;\text{ and }\; \forall\; \lambda \in (0,1)\,.$$

An illustration is given in Figure 3.4.

3.3.1 Convex Relaxation

In practice, only a few problems can be formulated as a convex optimization problem. To make use of the above discussed advantages of convex minimization, so-called *convex relaxation* techniques became popular. By dropping certain constraints from the overall optimization problem, these techniques usually generalize the given problem to a convex problem which is then easier to solve. To ensure that the solution of the relaxed problem is close to the solution of the original problem, tight relaxations were proposed which ensure certain optimality bounds.

In order to achieve a convex optimization problem, we need to 'convexify' the feasible set (3.10) of the energy functional (3.9). Rather than optimizing over the non-convex set $BV(\Omega; \{0,1\})$, we relax the feasible set and optimize over the convex hull $BV(\Omega; [0,1])$:

$$\min_{u_i \in BV(\Omega;[0,1])} \sum_{i=1}^{n} \big(\mathcal{D}(u_i, I) + \lambda \mathcal{R}(u_i)\big) \quad \text{s.t.} \sum_{i=1}^{n} u_i(x) = 1 \ \ \forall\, x \in \Omega, \tag{3.11}$$

with $\lambda \in \mathbb{R}$ and

$$BV(\Omega; [0,1]) = \left\{ u \in L^1(\Omega; [0,1]) : \int_\Omega |Du| < \infty \right\}. \tag{3.12}$$

This minimization problem is convex if the functionals $\mathcal{D}$ and $\mathcal{R}$ are convex.

3.3.2 Existence of Solutions

An essential question in the context of optimization is the existence of optimal solutions. Consider, *e.g.*, a topological space $(\mathcal{X}, \tau)$ and an extended real-valued functional $f\colon \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$. The solution set of the minimization problem $\min_{\mathcal{X}} f$,

$$\arg\min f = \left\{ \bar{u} \in \mathcal{X} : \ f(\bar{u}) = \inf_{u \in \mathcal{X}} f(u) \right\}, \tag{3.13}$$

can be possibly empty. In terms of optimization problems, the existence of solutions can usually be examined via a generalization of the fundamental theorem of optimization. In order to state this theorem, firstly, we introduce the definition of lower level sets and lower semicontinuous functionals [16].

Definition 5 *(Lower level set)* Let $(\mathcal{X}, \tau)$ be a topological space and let $f\colon \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$. For any $\alpha \in \mathbb{R}$, the lower α-level set of f is given by

$$lev_\alpha f = \{ u \in \mathcal{X} : \ f(u) \le \alpha \}. \tag{3.14}$$

Definition 6 *(Lower semicontinuity)* Let $(\mathcal{X}, \tau)$ be a topological space and let $f\colon \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$. For any $u \in \mathcal{X}$, we denote by $\mathcal{V}_\tau(u)$ the family of the neighborhoods of u for the topology τ. The functional f is said to be τ-lower semicontinuous at u if

$$\forall\, \lambda < f(u) \quad \exists\, V_\lambda \in \mathcal{V}_\tau(u) \quad \text{such that} \quad f(v) > \lambda \quad \forall\, v \in V_\lambda. \tag{3.15}$$

If f is τ-lower semicontinuous at all $u \in \mathcal{X}$, then f is said to be τ-lower semicontinuous on $\mathcal{X}$.

Theorem 1 *(Fundamental theorem of optimization)* Let $(\mathcal{X}, \tau)$ be a topological space and let $f \colon \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ be an extended real-valued functional which is τ-lower semicontinuous and such that for some $\alpha \in \mathbb{R}$, $lev_\alpha f$ is τ-compact.

Then, $\inf_{\mathcal{X}} f > -\infty$ and there exists some $\bar{u} \in \mathcal{X}$ which minimizes f on $\mathcal{X}$: $f(\bar{u}) \leq f(u) \; \forall \; u \in \mathcal{X}$, *i.e.*,

$$f(\bar{u}) = \inf_{u \in \mathcal{X}} f(u) \,. \tag{3.16}$$

Because of its importance, this theorem is often referred to as the *Weierstrass theorem* [16]. With the help of this theorem the existence of minimizers can be verified for most common variational formulations as for instance done in the work of Burger and Osher [35] for a class of TV reconstruction problems.

3.3.3 Binarization

By allowing the variables u_i, $i = 1, \ldots, n$ in Equation (3.11) to take on intermediate values between zero and one, the optimum $u^* = (u_1^*, \ldots, u_n^*)$ of the relaxed problem usually is not binary. To obtain a binary solution to the original optimization problem (3.9), we assign each pixel x to the label L with maximum value after optimizing the relaxed problem:

$$L(x) = \underset{i=1,\ldots,n}{\arg\max} \{u_i^*(x)\}\,, \;\; x \in \Omega. \tag{3.17}$$

In our experiments, we observed that the computed relaxed solutions u^* are binary at the vast majority of the pixels. *I.e.*, for most pixels $x \in \Omega$ and $i = 1, \ldots, n$: $u_i^*(x) < 0.01$ or $u_i^*(x) > 0.99$.

3.4 Extremality Condition

A common approach to determine the minima and maxima of a differentiable function is to locate the points where the first derivative equals zero. The French mathematician Pierre de Fermat summarized this in the following theorem [37]:

Theorem 2 *(Fermat's theorem (stationary points))* Let $g \colon (a, b) \to \mathbb{R}$ be a continuous function and suppose that $x_0 \in (a, b)$ is a local extremum of g. If g is differentiable in x_0 then $g'(x_0) = 0$.

In this thesis we aim to minimize the energy functional E (compare Equation (3.2)). E is called a *functional* as it maps from a set of functions u to the real numbers, *i.e.*, E is a 'function of a function'. The field of mathematical analysis that deals with maximizing or minimizing functionals is known as *calculus of variations*. The interest is therefore extremal functions in contrast to extremal points. Similar to Fermat's theorem in calculus, the extremal functions in calculus of variations may be obtained by finding functions where the functional derivative is equal to zero.

3.4.1 Gâteaux Derivative

A formulation of the functional derivative which is commonly used in the calculus of variations has been developed by the French mathematician René Gâteaux († 1914). He generalized the concept of the directional derivative for functions and proposed the so-called *Gâteaux derivative* [23, 60]:

Definition 7 *(Gâteaux derivative)* Let X be a Banach space and $F\colon X \to \mathbb{R}$. F is said to be Gâteaux differentiable at $u \in X$ if there is a bounded linear operator $DF\colon X \to \mathbb{R}$ such that for every $h \in X$,

$$DF\left(u\right)\left(h\right) = \lim_{\epsilon \to 0} \frac{F\left(u + \epsilon h\right) - F\left(u\right)}{\epsilon} = \frac{d}{d\epsilon} F\left(u + \epsilon h\right)\bigg|_{\epsilon = 0}. \tag{3.18}$$

The operator DF is called the Gâteaux derivative of F at u.

If F is Gâteaux differentiable and if the problem $\inf_{u \in X} F\left(u\right)$ has a solution u_0, then, analogous to Fermat's theorem, we have

$$DF\left(u_0\right) = 0. \tag{3.19}$$

Conversely, if F is convex, then a solution u_0 of $DF\left(u_0\right) = 0$ is a solution of the minimization problem. The equation $DF\left(u_0\right) = 0$ is called an *Euler-Lagrange equation* [18].

3.4.2 Euler-Lagrange Equation

The Euler-Lagrange equation is a necessary condition for the extremum of a particular form of (sufficiently smooth) energy functionals and has been developed by the two mathematicians Leonhard Euler and Joseph-Louis Lagrange in the 1750s. It can be derived by reducing the variational problem to a problem in the differential calculus. Let F be an integral of the form:

$$F\left(u\right) = \int_{x_0}^{x_1} \mathcal{L}\left(x, u, u'\right) dx, \tag{3.20}$$

where the values x_0, x_1, $u\left(x_0\right)$, $u\left(x_1\right)$ are given. The function $\mathcal{L}$ is to be twice continuously differentiable with respect to its three arguments x, u, u' and the second derivative u'' of the function u is also assumed continuous [45].

Definition 8 *(Euler-Lagrange equation)* Suppose that u is the desired extremal function yielding the minimum of the integral F in Equation (3.20). As a necessary condition for the existence of an extremum the function u has to satisfy the Euler-Lagrange equation given as follows:

$$\mathcal{L}_u - \frac{d}{dx} \mathcal{L}_{u'} = 0. \tag{3.21}$$

The differential expression $\mathcal{L}_u - \frac{d}{dx}\mathcal{L}_{u'}$ is called the *variational derivative* of $\mathcal{L}$ with respect to u and can be seen analogous to that of the gradient in ordinary minimum problems. For a detailed derivation, we refer to Courant and Hilbert [45].

3.4.3 Subdifferential

In the field of convex optimization, subdifferentials are popular as a generalization of the derivative to functions which are not differentiable. Thus, the subdifferential calculus can be employed for solving convex minimization problems [16].

Definition 9 *(Subdifferential)* Let $(V, \|\cdot\|)$ be a normed space with topological dual space $(V^*, \|\cdot\|_*)$ and $f\colon V \to \mathbb{R} \cup \{+\infty\}$ be a closed convex proper function. We say that an element $u^* \in V^*$ belongs to the subdifferential of f at $u \in V$ if

$$\forall\, v \in V \quad f\,(v) \geq f\,(u) + \langle u^*, v - u\rangle_{(V^*,V)}\,. \tag{3.22}$$

We then write $u^* \in \partial f\,(u)$.

Using this definition we can formulate the following theorem stating the central role of the subdifferential calculus in convex optimization [16].

Theorem 3 Let $(V, \|\cdot\|)$ be a normed space and $f\colon V \to \mathbb{R} \cup \{+\infty\}$ be a closed convex proper function. Then, for an element $u \in V$ the following statements are equivalent:

(i) $f\,(u) \leq f\,(v)\ \ \forall\, v \in V$;

(ii) $\partial f\,(u) \ni 0$.

The above theorem gives a necessary and sufficient condition for an element $u \in V$ to be a solution of the convex minimization problem $\min_{v\in V} f\,(v)$. This necessary and sufficient condition

$$\partial f\,(u) \ni 0 \tag{3.23}$$

is an extension to nonsmooth convex functions of the classical first-order condition of optimality for convex C^1 functions, namely

$$\nabla f\,(u) = 0 \tag{3.24}$$

and can be seen as a generalization to the classical Fermat rule (compare Theorem 2). Thus, for a given convex optimization problem, the problem of finding the optimal solution can be approached by using the subdifferential calculus and solving the generalized equation $\partial f\,(u) \ni 0$ [16].

Part II

Own Publications

Chapter 4

The Role of Diffusion in Figure Hunt Games

Authors

Julia Diebold[1] *julia.diebold@tum.de*
Sibel Tarı[2] *stari@metu.edu.tr*
Daniel Cremers[1] *cremers@tum.de*

[1]Technische Universität München, Munich, Germany
[2]Middle East Technical University, Ankara, Turkey

Status Published

Publication J. Diebold, S. Tari, and D. Cremers. The Role of Diffusion in Figure Hunt Games. In *Journal of Mathematical Imaging and Vision (JMIV)*, 52(1):108–123: *Special Issue on Scale Space and Variational Methods in Computer Vision.* Springer US, 2015. DOI: 10.1007/s10851-014-0548-6

Individual contribution: Leading role in realizing the scientific project

Problem definition	*significantly contributed*
Literature survey	*helped*
Method development & evaluation	*significantly contributed*
Implementation	*significantly contributed*
Experimental evaluation	*significantly contributed*
Preparation of the manuscript	*significantly contributed*

Abstract We consider the task of tracing out target figures hidden in teeming figure pictures known as *figure hunt games.* Figure hunt games are a popular genre of visual puzzles; a timeless classic for children, artists and cognitive scientists. We argue and experimentally demonstrate that diffusion is a key to algorithmically search for a target figure in a binary line drawing. Particularly suited to the considered task, we propose a diffuse representation which diffuses the image while retaining the contour information.

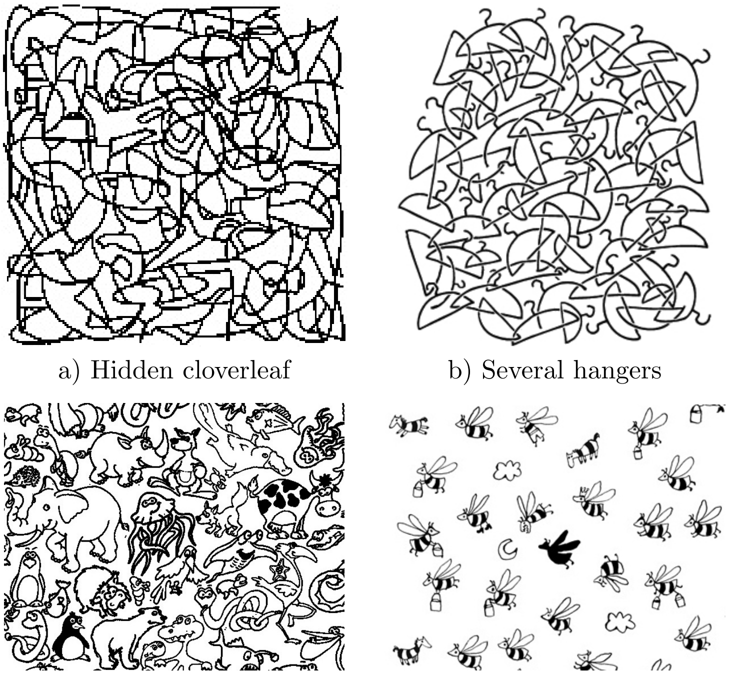
Keywords Screened Poisson PDE and variants · Level sets · Non-linear diffusion · Figure hunt games · Teeming figure pictures · Applications of variational and PDE methods

4.1 Introduction

In 1986, the British illustrator Martin Handford created the distinctive red-and-white dressed character Wally. Since that day, *Where's Wally?* became an extremely popular series of children's books consisting of diverse illustrations, depicting dozens of people. Readers are challenged to find Wally in illustrations where an abundant number of small figures including Wally are brought together.

Where's Wally? is only a sample, though the most famous, in a popular genre of visual puzzles called *figure hunt games.* Figure hunt games have been a timeless classic for children, artists and cognitive scientists. As early as 1926 Kurt Gottschaldt experimented with intentionally designed hidden figures – simple drawings where simple shapes such as polygons are embedded within more complex organizations – to study the influence of experience on perception and the extent to which holes influence the perception of parts [64]. Gottschaldt type puzzles (Figure 4.1 top row) together with the *Where's Wally?* type ones (Figure 4.1 bottom row) form the focus of this paper. This sub-genre of the figure hunt can be generalized based on two factors as exemplified in Figure 4.1. The first factor is the co-dimension: The individual figures in the top-row illustrations are one-dimensional objects drawn on top of each other whereas the ones in the bottom-row illustrations are two dimensional. The second factor is the number of targets. In the first column, each illustration contains a single target, whereas in the second column several targets (hangers and bees, respectively) are placed among distractors.

In the course of this paper, we will discuss how a computer program can trace out the contours of the hidden clover or hangers and locate the desired animals. The question might for example be: “Can you trace out the hidden cloverleaf?” or “How many hangers are there?”, “Where is the sea star?” or “Are there other animals hidden among the bees?”. We shall adopt the most straightforward solution: searching the entire illustration; hypothesize a pose and scale for the target figure and measure how well it fits. Among other things, this requires 1) a randomized search



a) Hidden cloverleaf

b) Several hangers

c) Hidden elephant

d) Multiple bees

Figure 4.1: **Samples of figure hunt games.** Games with a single target (left column) vs. several targets (right column) and Gottschaldt type puzzles (top row) vs. *Where's Wally?* type ones (bottom row).

algorithm that will return multiple answers; 2) a fitness function which distinguishes a bad fit from a good fit. We will argue and experimentally demonstrate that replacing illustrations and/or target figures with diffuse forms significantly helps. Thus, the paper shows the effect of diffusion for the particular considered task of figure hunt games. A secondary issue is how to speed up the search process. To this end, we will experiment with a coarse-to-fine strategy using diffuse forms.

A preliminary conference version examining only the Gottschaldt type puzzles has appeared in SSVM 2013 [4]. In this paper, we concentrated on a thorough evaluation including detailed insights about the experiments in terms of parameters and applied methods.

Related Work Finding an object's position in an image is a commonly addressed problem. There are many methods. For example, one may treat the output of an edge detector as an illustration and try to fit the target shape's boundary to the correct position. Such a fitting can even be done using the generalized form of the Hough Transform [19] provided that the shape can be expressed in some parametric form. Such methods, however, are not applicable when the illustration contains embedded shapes (Figure 4.1 a-b), or when the target figure is a complex form with embedded subfigures. The closest contour matching technique to ours is *chamfer matching* [21]. In chamfer matching, the template is correlated with the distance transform of the illustration. In case of clutter, chamfer matching requires additional improvements, *e.g.*, learning [98]. As one improvement, we propose to replace the distance transform with a more informative diffuse field which implicitly codes curvature. Moreover, our experiments indicate that applying the transformation to the target rather than the illustration may also offer benefits in certain settings. We provide a complete evaluation. Distance transforms have also been instrumental in level set methods. Most typically, shape knowledge is coded via the signed distance transform, embedding the $1-D$ shape boundary as the zero-level set of a function defined on a connected bounded open subset of $\mathbb{R}^2$ [114, 116]. Level set methods, however, are not applicable when there are embedded shapes.

It is possible to replace the sharp interface model in level set based segmentation methods with diffuse ones. For example, smooth distance fields that exhibit exponential decay rather than linear growth are obtained by solving a screened Poisson PDE. These kind of distance fields are more informative in the sense that they implicitly code curvature in addition to distance. The whole topic has a recent revival with a wide range of applications and new theoretical insights [15, 17, 68, 152]. The earliest work by Tarı *et al.* [153] addresses the connection between screened Poisson and image segmentation by the Ambrosio-Tortorelli approximation [11] of the Mumford-Shah model [105]. This particular work has recently been used in [75] to address a search problem where a small fragment of the illustration is searched in order to reveal the underlying global repetition structure in abstract ornaments. The curvature-coding field we propose improves search and does not require solving a PDE.

In reconstructing frescos, Fornasier *et al.* [58] addressed the problem of locating small fragments within a whole. For each small piece of plaster that still showed an element of the design of the fresco, the authors were able to find where it belonged. This is quite an elegant method, but the non-additive and non-linear nature of the binary illustrations that we consider prevents its use.

To the best of our knowledge, discovery of hidden figures as we describe has not been studied within the mathematical imaging community. Nevertheless, Saarbrücken group's recent inpainting based steganography application [99] addresses the opposite problem: to hide a secret image by embedding it into arbitrary cover images. Both the secret and the cover are dense images, and the recovery of the secret is possible only via a password. That is, ordinary observer cannot detect whether an image contains a secret or not. Object camouflage is also a problem of interest in the computer graphics community [43].

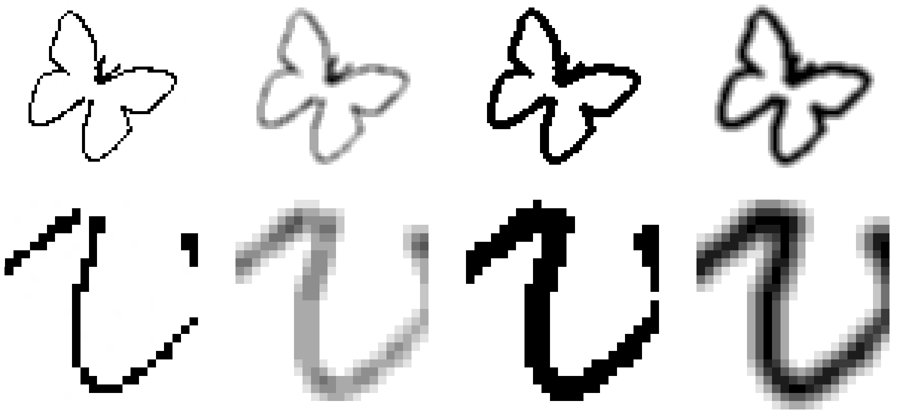

a) Input image b) Averaging (mean filter) c) Erosion d) Erosion and averaging

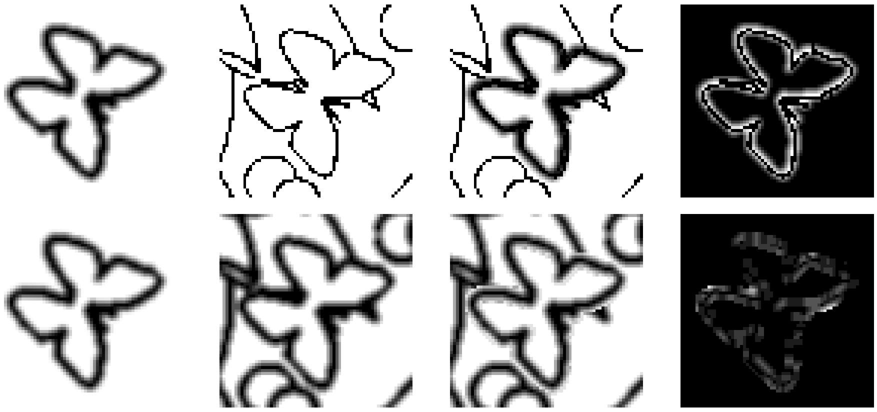
Figure 4.2: **Erosion followed by averaging.** When pure averaging – *e.g.* mean filter – is applied to a binary line drawing, the contour location vanishes. In contrast, when erosion with subsequent averaging is applied, the contour location information is retained.

4.2 Formalization

We consider the task of figure hunt games: tracing out *target figures* hidden in binary *illustrations*. Let $\Omega \subset \mathbb{R}^2$ be the image domain, and let $\mathcal{F}$, $\mathcal{I}\colon \Omega \to \{0,1\}$ be the target figure and the illustration, respectively. The goal is to localize a target figure, such as the butterfly Figure 4.2 a) in an illustration, such as the mandala Figure 4.4 a). Values 1 (white) correspond to the background and 0 (black) to the foreground.

We start by uniformly eroding the white space, or equivalently, dilating the target figure (*e.g.* Figure 4.2 a) and/or the illustration. Hence, the drawing becomes thicker (see Figure 4.2 c). Then, we diffuse by computing a local isotropic average. It is sufficient to compute the local average only for the points falling on the thickened figural loci or in a slightly wider band surrounding it. This transforms the sketch-like binary drawing to a gray-tone picture which may be referred as a diffuse drawing $\mathcal{F}_d$ (Figure 4.2 d). In the following, we use the term *diffusion* to entitle the transformation of the binary drawing to a gray-tone picture, although this transformation does not necessarily describe a diffusion in the technical sense.

The key idea is to propagate information restricted to figural loci to neighboring areas. Thus, it becomes possible to judge whether a background location is close to or far away from a figural loci. If the averaging and the erosion radii are identical, the highest value is attained on the figural loci; from thereof values decrease as a function of distance in the normal direction. Thus, diffusion produces iso-intensity contours, each following the figural loci from a fixed distance. The lower the intensity, the further away the iso-intensity curve from the figural loci. The second column in



a) Deformed target fig. $\mathcal{F}_{\mathcal{D}}$ b) Zoom of illustration $\mathcal{I}$ c) Placed target fig. $\mathcal{F}_{\mathcal{P}}$ d) Matching cost E_{cost}

Figure 4.3: **Matching cost.** Visualized computation of the matching cost. In the top row, the diffuse target is searched in a binary illustration. In the bottom row, the diffuse target is searched in a diffused illustration.

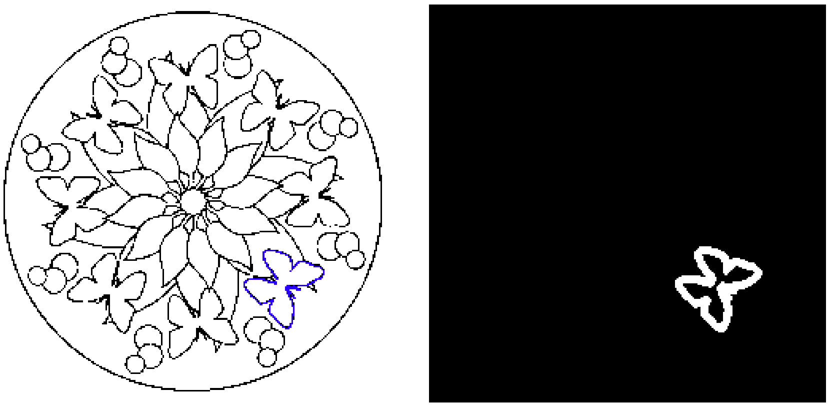
Figure 4.2 depicts the result of local isotropic averaging applied to the original thin drawing. There, one cannot observe the distance-coding behavior, *i.e.* the initial thickening is a crucial step.

4.2.1 Matching Cost

Once the target figure $\mathcal{F}$ (Figure 4.2 a) is converted to a diffused form $\mathcal{F}_d$ (Figure 4.2 d), the best match is determined by the deformation parameters (*i.e.* location, orientation and scale) yielding the best *matching cost.* The matching cost is measured as the sum of the gray-value differences between the illustration $\mathcal{I}$ and the *placed target figure* $\mathcal{F}_{\mathcal{P}}$ (Figure 4.3 c). A visualization is shown in Figure 4.3. We introduce the matching cost by means of the binary illustration (top row) as well as the diffused illustration (bottom row). A discussion about the role of diffusion follows in Section 4.3.

The placed target figure $\mathcal{F}_{\mathcal{P}}$ is obtained by the combination of the deformed target figure $\mathcal{F}_{\mathcal{D}}$ (Figure 4.3 a) and the illustration $\mathcal{I}$ (Figure 4.3 b). Depending on the application, $\mathcal{F}_{\mathcal{P}}$ can be computed by means of the binary illustration or the diffused illustration. In Matlab coding language $\mathcal{F}_{\mathcal{P}}$ can be obtained as follows:

$\mathcal{I}$ binary illustration	$\mathcal{I}$ diffused illustration
$\mathcal{F}_{\mathcal{P}} := \mathcal{F}_{\mathcal{D}}$;	$\mathcal{F}_{\mathcal{P}} := \mathcal{I}$;
$\mathcal{F}_{\mathcal{P}}(\mathcal{I} \neq 1) = \mathcal{I}$;	$\mathcal{F}_{\mathcal{P}}(\mathcal{F}_{\mathcal{D}} \neq 1) = \mathcal{F}_{\mathcal{D}}$;



a) Illustration $\mathcal{I}$ b) Band $\mathcal{B}$ surr. the figural loci

Figure 4.4: **Optimal match.** a) Perfect hint of the target figure in the illustration $\mathcal{I}$; b) Set $\mathcal{B}$ defined by a band surrounding the figural loci.

Hereby, $\mathcal{F}_{\mathcal{D}}$ is obtained by the deformation of the diffused target figure $\mathcal{F}_d$ with the respective deformation parameters:

$$\mathcal{F}_{\mathcal{D}} = \mathcal{F}_{\mathcal{D}}(\mathcal{F}_d, \text{deformation parameters}) \tag{4.1}$$

The visual matching cost in Figure 4.3 d) is defined as the absolute value of the pixel-wise difference between the illustration $\mathcal{I}$ (Figure 4.3 b) and the placed target figure $\mathcal{F}_{\mathcal{P}}$ (Figure 4.3 c). In general, E_{cost} can be obtained as follows:

$$E_{\text{cost}} = \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} |\mathcal{I}(x) - \mathcal{F}_{\mathcal{P}}(x)| \,, \tag{4.2}$$

where $\mathcal{I} \colon \Omega \to [0,1]$ is the (binary/diffused) illustration, $\mathcal{F}_{\mathcal{P}} \colon \Omega \to [0,1]$ the placed target figure and $\mathcal{B} \subset \Omega$ the set indicating the band surrounding the figural loci.

The set $\mathcal{B}$ is illustrated in white in Figure 4.4 b). $\mathcal{B}$ is the collection of pixels which belong to gray values of the deformed target figure (Figure 4.3 a). Hence, the sum is taken over those locations that fall within the band surrounding the figural loci within which the diffuse field has been constructed. Moreover, the cost is normalized by dividing it by the number of locations that contributed to its computation.

Example Let the illustration be the mandala consisting of butterflies shown in Figure 4.4 a). As target figure we take the butterfly in Figure 4.2 a). A perfect hint for the position of the butterfly is indicated in blue right on the bottom of the mandala in Figure 4.4 a). The components of the matching cost (4.2) for this figure hunt are illustrated in Figure 4.4 b) and Figure 4.3. The resulting matching cost is visualized in Figure 4.3 d).

To find a solution with minimal matching cost, we optimize the set of deformation parameters leading to the deformed target figure $\mathcal{F}_{\mathcal{D}}$. We determine these optimizing parameters via a probabilistic algorithm which returns multiple solutions. We use genetic algorithm based optimization which is readily available in the Matlab environment. It minimizes an energy functional by varying its input variables. A detailed discussion follows in the next section.

4.2.2 Optimization Via a Genetic Algorithm

A genetic algorithm is a search heuristic that mimics the process of natural evolution. The evolution starts with a population of random generated initial solutions of the problem. In every step new populations are created, such that a fitness function is minimized. The populations are evolved towards an optimal solution by *selection*, *combination* and *modification* of the intermediate results:

- **Selection** identifies good solutions in a population and discards the rest (*e.g.* by measuring against the fitness function).
- **Combination** – also known as crossover – creates new solutions from existing ones.
- **Modification** (or mutation) introduces new features into the solution to maintain diversity in the population.

This process is repeated as long as either a satisfactory matching cost has been reached, or a maximum number of generations has been produced.

In figure hunt games, the best matching of the target figure with the illustration can be described by the deformation parameters leading to the best matching cost. Therefore, we aim to solve the following optimization problem:

$$\min_{(\theta, t_{\mathrm{r}}, t_{\mathrm{c}}, h_{\mathrm{r}}, h_{\mathrm{c}}) \in \mathcal{D}} E(\theta, t_{\mathrm{r}}, t_{\mathrm{c}}, h_{\mathrm{r}}, h_{\mathrm{c}}) \tag{4.3}$$

where $\mathcal{D} \subset \mathbb{R}^5$ is the domain of the deformation parameters:

- θ being the rotation angle,
- $t_{\mathrm{r}}, t_{\mathrm{c}}$ describing the translation in row/column direction,
- $h_{\mathrm{r}}, h_{\mathrm{c}}$ for scaling in direction of rows/columns.

The energy E to be minimized is defined as follows:

$$\begin{aligned} &E(\theta, t_{\mathrm{r}}, t_{\mathrm{c}}, h_{\mathrm{r}}, h_{\mathrm{c}}) := E_{\mathrm{cost}}, \quad \text{(compare Equation (4.2))} \\ &\text{with } \mathcal{F}_{\mathcal{D}} = \mathcal{F}_{\mathcal{D}}(\mathcal{F}_d, \theta, t_{\mathrm{r}}, t_{\mathrm{c}}, h_{\mathrm{r}}, h_{\mathrm{c}}). \end{aligned} \tag{4.4}$$

In order to find the optimal set of values in this five dimensional search space, we make use of the genetic algorithm built in Matlab:

```
ga(fitnessfcn, nparams, [], [], [], [], lb, ub, [], IntCon)
```

The output of the algorithm includes the parameter set $(\theta, t_\mathrm{r}, t_\mathrm{c}, h_\mathrm{r}, h_\mathrm{c})$ corresponding to the best matching cost E_cost for a given figure hunt problem. The input variables have the following meaning:

- fitnessfcn = {@*energy_functional*, illustration, target figure}, where the *energy_functional* is a function which takes the parameter set as well as the illustration and the target figure as input and returns the corresponding matching cost E_cost
- nparams: number of parameters to optimize (= 5)
- lb/ub: lower/upper bound (*e.g.* $\theta \in [0, 360]$)
- IntCon: integer constraints on parameters (= $[2, 3]$: parameters $t_\mathrm{r}, t_\mathrm{c}$ should be integers)[1]

In every step of the genetic algorithm, new populations are created. Hereby, the lower/upper bound constraint as well as the integer constraints have to be fulfilled. The selection, combination and modification process is guided by the values of the *energy_functional*, aiming to obtain a minimal matching cost E_cost. We are aware that the genetic algorithm also has disadvantages. In particular, the algorithm is non-deterministic and there is no proof of optimality known. While alternative algorithms are conceivable, we chose the genetic algorithm because it provides a good tradeoff between speed and quality of computed solutions.

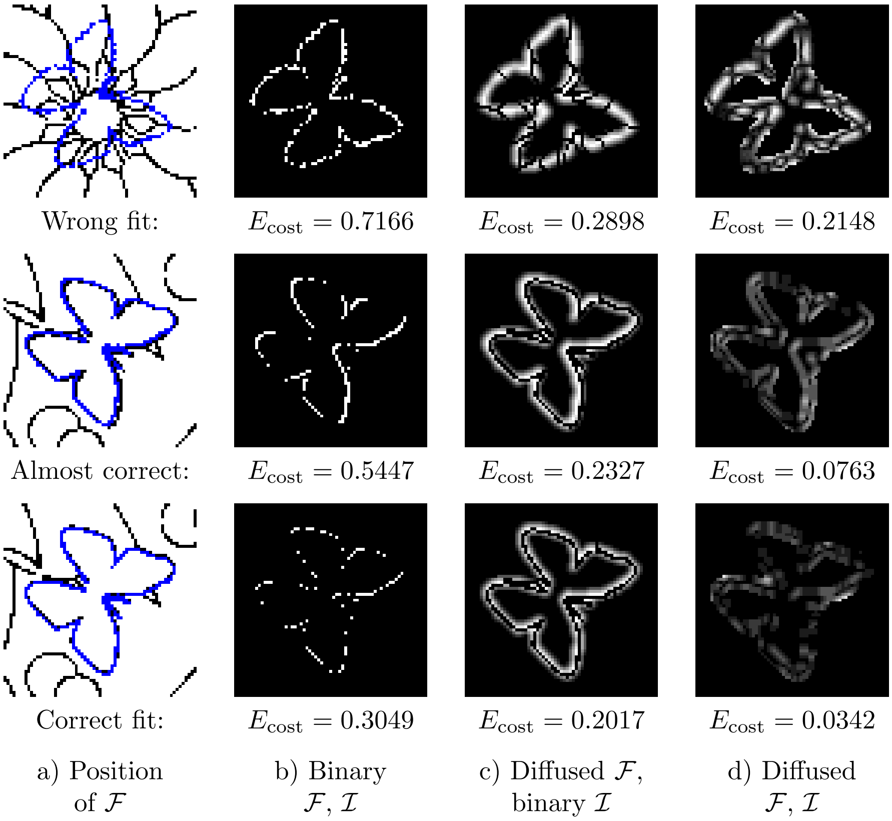
4.3 The Role of Diffusion

In every step of the genetic algorithm, new populations are created such that the matching cost is minimized. To propagate information restricted to figural loci to neighboring areas we use a diffused representation of the target figure (and the illustration). Thus, it becomes possible to know whether a location is close or far away from one of the desired locations.

Diffusion of the target figure (and the illustration) helps in two different ways:

1. **Uninformative pixels become informative.**
   Binary line drawings like the ones shown in Figure 4.1 contain large empty (white) regions without any information. By diffusing the drawing, the information restricted to the figural loci becomes visible within a neighborhood. This allows the search for the cloverleaf in Figure 4.1 a) and the hangers in Figure 4.1 b).

[1] The integer constraints on t_r, t_c can also be omitted with the drawback of higher computational costs. However, our experiments showed a sufficient accuracy when restricting the translation to integers.



Wrong fit: $E_{\text{cost}} = 0.7166$ $E_{\text{cost}} = 0.2898$ $E_{\text{cost}} = 0.2148$

Almost correct: $E_{\text{cost}} = 0.5447$ $E_{\text{cost}} = 0.2327$ $E_{\text{cost}} = 0.0763$

Correct fit: $E_{\text{cost}} = 0.3049$ $E_{\text{cost}} = 0.2017$ $E_{\text{cost}} = 0.0342$

a) Position of $\mathcal{F}$ b) Binary $\mathcal{F}, \mathcal{I}$ c) Diffused $\mathcal{F}$, binary $\mathcal{I}$ d) Diffused $\mathcal{F}, \mathcal{I}$

Figure 4.5: **Expressive energy by diffusion.** Observe the energy drop in the last column.

2. **Improved search process.**
   A strong diffusion may simplify and hence speed up the search process (*e.g.* to get the rough positions of the bees in Figure 4.7). In addition, diffusion convexifies the energy and thereby improves the localization.

4.3.1 Spreading the Edge-Information

In order to minimize the matching cost, a correlation between the quality of the match and the cost is required. This correlation is not given, if the binary representations of the target figure and the illustration are used.

Figure 4.5 compares the matching costs of three different matches: 1) wrong fit; 2) almost correct fit; 3) correct fit. The first column indicates the position of the butterfly within the mandala. The second column depicts the matching cost obtained by means of the binary illustration and target figure. Columns three and four give the matching costs obtained with the diffused target figure together with the binary and the diffused illustration. Observe that the matching costs computed

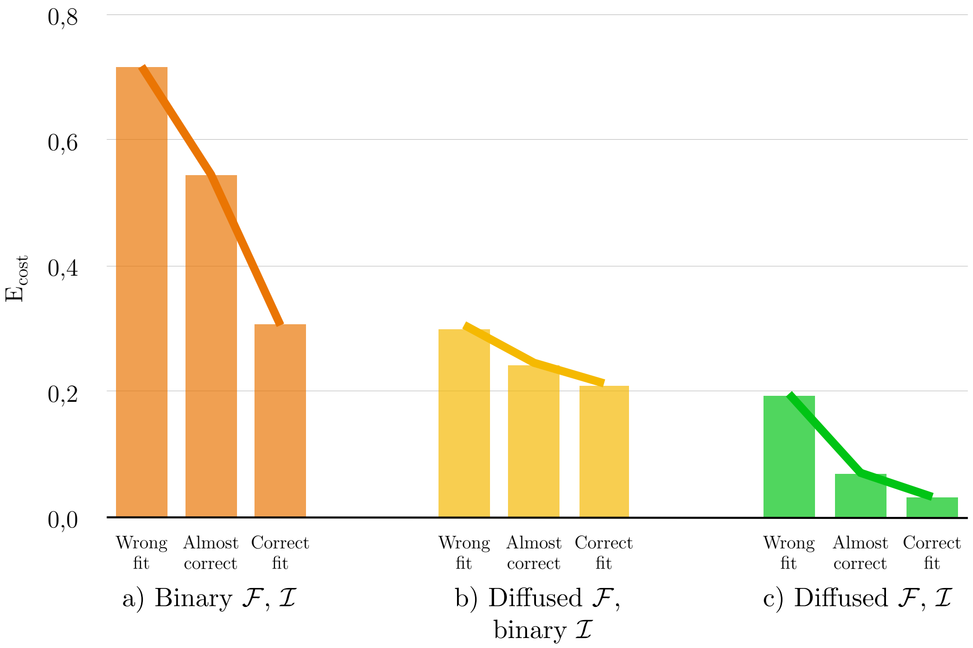


Figure 4.6: **Comparison of matching costs.** The decay of the energy represents the convergence towards a good fit. Compare the first graph to the last.

with the non-diffused drawings (column b), are almost equal. In particular, the visualized energies of a wrong fit and an almost perfect fit are indiscernible. Hence, there is no reliable optimization criterion. It is unclear whether an intermediate match leads to a good fit. In contrast, the energy of the wrong fit in column c,d) is significantly higher (lighter) than the energy of the (almost) correct fit. This means that the cost becomes informative.

Figure 4.6 demonstrates the different matching costs visualized in Figure 4.5. To be able to make decisions about the goodness of a fit, the matching cost corresponding to a bad fit should be substantially higher than the one corresponding to a good fit. In particular, a graph indicating the matching cost of a wrong, an almost correct and a correct fit should first have a strong decay followed by a weak decay. Figure 4.6 shows the graphs corresponding to the three columns b-d) of Figure 4.5. Whereas the two rightmost graphs show the expected decay, the first graph does not imply the position of the target figure. The reason is that the binary figures include lots of uninformative (white) pixels and therefore cannot decide whether a fit is good. In contrast, the diffusion propagates information about the figural loci from purely local to a neighborhood (compare Figure 4.2). Hence, the desired location can be observed from some distance and leads to an informative energy.

With the diffused representations, the genetic algorithm has a clear optimization criterion and thus returns the optimal match of the chosen target figure in the given illustration.

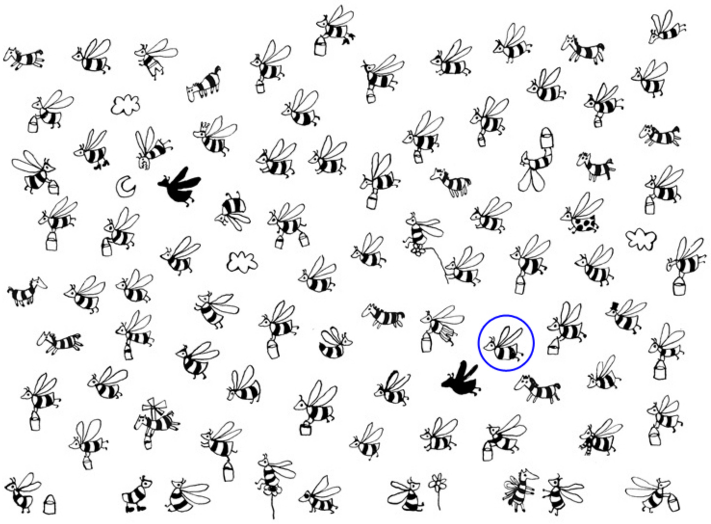


Figure 4.7: **How many bees** are there in the image?

4.3.2 Improved Localization and Speed-Up

A diffused representation not only propagates edge information to a neighborhood, it also simplifies the search process. This behavior comes from the fact that diffusion improves localization by convexifying the energy. Similar blurring strategies are known by continuation approaches such as graduated non-convexity [28].

A typical example is the swarm of bees in Figure 4.7. To count the number of bees, a strong diffusion can be applied, leading to an accumulation of gray splotches. Taking one of them as target figure, the genetic algorithm quickly detects the splotches throughout the bee swarm. In a second step, a finer search can be applied around the detected positions to obtain a more precise hint of the bees. Extensive experiments will be shown in Section 4.5.1. A crucial point is the *preservation of the original contour* features. Despite diffusing the binary drawings the contour location has to be preserved. This is not given for all diffusion methods. Edges can be washed-out without coding the original contour location or discretization artifacts can be amplified.

To choose the best diffusion method for the particular considered task, we will discuss different diffusion methods in the next section.

4.4 Diffusion Methods

We discussed that diffusion is an essential step for the algorithmic solution of a figure hunt game. During our studies we tested several diffusion approaches. In the course of this section we will give a detailed discussion of the four most interesting ones:

- Averaging
- Distance function
- v-transform [153]
- Erosion followed by averaging

In the following, let $\Omega \subset \mathbb{R}^2$ denote the image domain and $I\colon \Omega \to \{0,1\}$ be the binary image.

4.4.1 Averaging

In order to spread the edge-information, the most intuitive way is to apply a diffusion filter, *e.g.*, the mean filter, where each pixel value is replaced by the mean of the pixel values in its neighborhood. Let $\sigma > 0$ be a parameter. The averaging is a basic convolution of I, where each pixel $(x, y) \in \Omega$ is assigned the average value of its neighborhood of size $\sigma \times \sigma$. Hence, the diffused version can be obtained by:

$$(I * k)(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x-a, y-b)\ k(a,b)\ da\ db, \tag{4.5}$$

where $(x, y) \in \Omega$ and

$$k\,(a,b) = \begin{cases} \frac{1}{\sigma^2} & \text{if } \ |a| \leq \frac{\sigma}{2} \ \text{ and } \ |b| \leq \frac{\sigma}{2}, \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

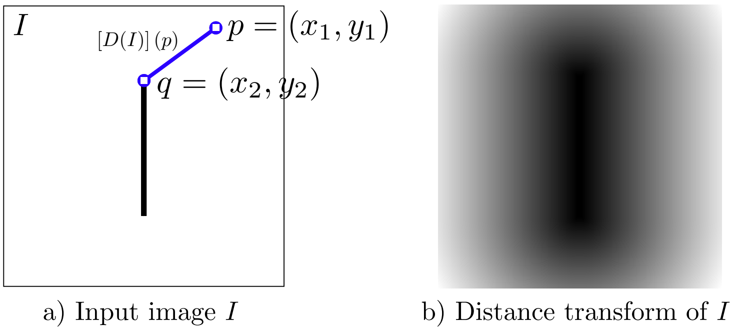
is a standard kernel of the mean filter. For the boundary condition, *zero padding* is used, *i.e.* the boundary of the image is augmented by zeros.

The butterfly diffused by averaging is shown in Figure 4.2 b). The parameter σ was set to $\sigma = 3$. In the zoomed part of the figure (second row), one cannot identify the original location of the contour. The contour is completely blurred, and it merges with the background. One would rather like to spread the edge information but at the same time keep the information about the original edge location. Hence, we propose a second method: *erosion followed by averaging*, which will be described in Section 4.4.4.

4.4.2 Distance Function

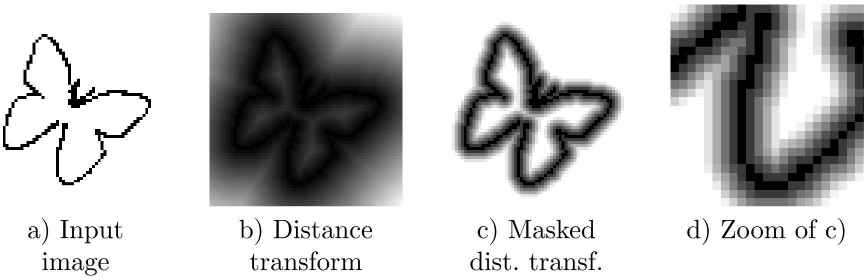
Another option for spreading the edge information is the usage of the distance function (sometimes referred to as the *distance transform*). The *distance function D* of a binary image I associates each pixel p of the domain Ω of I with its distance to the nearest zero-valued pixel:

$$[D(I)]\,(p) = \min\left\{d(p,q) \mid I(q) = 0\right\}. \tag{4.7}$$



a) Input image I b) Distance transform of I

Figure 4.8: **Exemplary distance transform.**



a) Input image b) Distance transform c) Masked dist. transf. d) Zoom of c)

Figure 4.9: **Distance transform** of the butterfly line drawing. By restricting the values to a band surrounding the figural loci, the desired diffused target figure results. However, the zoom shows a strong effect of discretization noise.

The distance function of an image of a black line on a white background is illustrated in Figure 4.8.

Here, the metric d for a space $\mathbb{E}$ is a function associating a nonnegative real number with any two points p and q of $\mathbb{E}$ and satisfying the three conditions of a norm. *E.g.* the *Euclidean distance* d:

$$d\left[(x_1, y_1), (x_2, y_2)\right] = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{4.8}$$

where $p = (x_1, y_1)$ and $q = (x_2, y_2)$.

The distance transform of the butterfly is shown in Figure 4.9 b). Each pixel includes information about its distance to the contour. The desired diffused version, however, requires the information to be restricted to a band around the contour. Thus, we omit the values beyond a band surrounding the figural loci and stretch the remaining gray-values to fill the whole range from 0 to 255. The resulting ‘masked’ version is shown in Figure 4.9 c) with a closeup in d). The closeup reveals the effect of discretization noise which remained (even amplified) despite the diffusion.

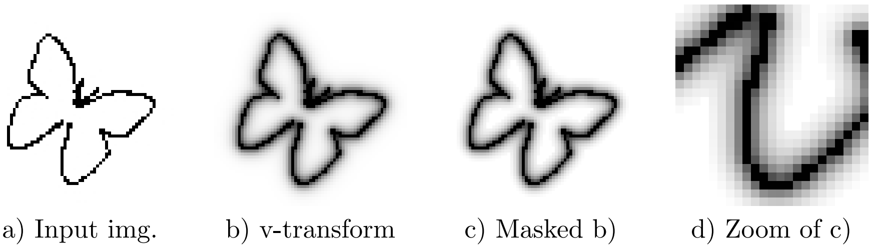

a) Input img. b) v-transform c) Masked b) d) Zoom of c)

Figure 4.10: **v-transform** of the sketch-like binary butterfly.

4.4.3 v-Transform [153]

To obtain a gray-tone picture of a sketch-like binary drawing, one other option is to use the v-transform. It is the minimizer of the following functional

$$\frac{1}{2}\int\int_{\Omega}\left\{\rho\left\|\nabla v\right\|^{2}+\frac{(1-v)^{2}}{\rho}\right\}dxdy \tag{4.9}$$

subject to $v(x,y)=0$ on $\{(x,y):I(x,y)=0\}$.

Numerically, we solve for v iteratively using the following update step:

$$v^{t+1}=\left(1+\frac{\tau}{\rho^{2}}\right)^{-1}\cdot\left(v^{t}+\frac{\tau}{\rho^{2}}+\tau\nabla^{2}v^{t}\right) \tag{4.10}$$

where τ denotes the step size.

For the butterfly drawing, the resulting gray-tone picture computed with the parameters $\tau=0.5$ and $\rho=3$ is shown in Figure 4.10 b). Again, diffused pixels are spread throughout the whole image. Hence, we mask the image and omit values too far away. The result is displayed in Figure 4.10 c) with a closeup in d).

Unlike the usual distance transform (Section 4.4.2), v is an implicit coder of the curvature, a valuable geometric feature, without explicit estimation of higher order derivatives. (One of the original goals in proposing v was to bridge low level and high level vision [151, 153]).

In this paper, in the setting of our specific task, we advocate a much simplistic way of obtaining an analogous behavior in a band around the contour of the drawing. We present this idea in the next section. Our computation does not require the computation of the entire v function on the entire domain Ω by solving a PDE.

4.4.4 Erosion Followed by Averaging

To keep the edge information while blurring the contour, we combine pure averaging presented in Section 4.4.1 with the morphological operation 'erosion'. The intuitive idea is to: 1) broaden the edge of the binary line drawing; 2) smooth the thicker edge.

If the averaging and the erosion radii are identical, the highest value is attained on the figural loci; from thereof values decrease as a function of distance in the normal direction. Figure 4.2 c,d) illustrate the broadened edge and the diffused version for the butterfly drawing.

In the first step, an erosion is applied to broaden the contour line of the binary line drawing. In principle, one is used to the term 'dilation' for enlarging. To stick with the standard terminology referring to the white space, we use the term 'erosion' (instead of 'dilation' of the black space). In the following, we show a detailed explanation of the applied erosion.

Let $\mathcal{S}$ be the structuring element. We denote the erosion of the image I by $\mathcal{S}$ via $\varepsilon_{\mathcal{S}}(I)$:

$$\varepsilon_{\mathcal{S}}(I) = \bigwedge_{s \in \mathcal{S}} I_{-s}, \tag{4.11}$$

the minimum of the translations of I by the vectors $-s$ of $\mathcal{S}$. In other words, the eroded value at a given pixel x is the minimum value of the image in the window defined by the structuring element $\mathcal{S}$ when its origin is at x:

$$\left[\varepsilon_{\mathcal{S}}(I)\right](x) = \min_{s \in \mathcal{S}} I\left(x+s\right). \tag{4.12}$$

The butterfly eroded by the set

$$\mathcal{S} \;=\; \left\{(x,y) \in \mathbb{Z}^2 \;\middle|\; \|(x,y)\| \leq S\right\} \tag{4.13}$$

with $S = 1$ is shown in Figure 4.2 c).

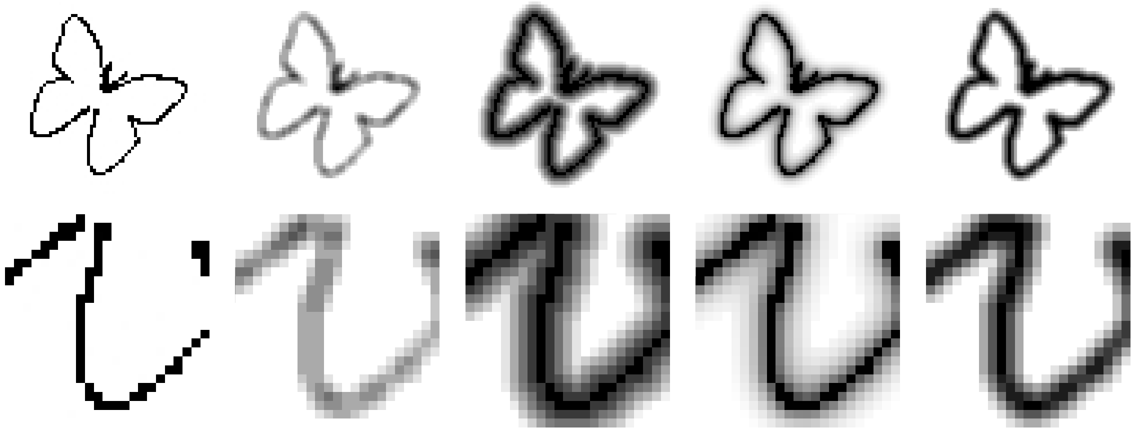
In the next step, the eroded image is smoothed (as in Section 4.4.1):

$$(\varepsilon_{\mathcal{S}}(I) * k)(x) = \int_{\mathbb{R}^2} [\varepsilon_{\mathcal{S}}(I)](x-a)\; k(a)\; da, \tag{4.14}$$

where $x \in \Omega$ and k is a standard kernel of the mean filter as defined in Equation (4.6) with $\sigma = 2S+1$. Again, zero padding is used to augment the boundary. The butterfly obtained by erosion with $S = 1$ (in Equation (4.13)) and subsequent averaging with $\sigma = 3$ is shown in Figure 4.2 d).

Advantages

Within a band surrounding the figural loci, our diffuse drawing (obtained by erosion of the white space followed by averaging) mimics a curvature coding distance field similar to the v-transform, the solution of a screened Poisson PDE [153]. We avoid solving Poisson PDEs or variants for two reasons. Firstly, our approximation is both easier and faster to compute. But more importantly, a Poisson based distance field, being the steady state solution to a biased diffusion equation is too much influenced by long-range interactions among opposing boundaries. This may be detrimental if several figural loci overlap as in Figure 4.1 top row.



a) Input image

b) Pure averaging

c) Distance transform

d) v-transform

e) Erosion & averaging

Figure 4.11: **Comparison** of introduced diffusion approaches. The proposed erosion followed by averaging e) gives the most informative diffused image whilst being much simpler than c) or d).

4.4.5 Overview

In Figure 4.11, we compare our diffuse drawing to the alternatives: pure averaging, usual distance image and the v-transform. All diffuse drawings are restricted to a band surrounding the figural loci. Whereas the pure averaging b) returns a blurred image where the contour location vanished, the other diffusion approaches keep the edge information while blurring. However, the effects of discretization noise remain (even amplified) in the usual distance image c), whereas the iso-intensity contours in our diffuse model e) *smoothly* follow the boundary. Additionally, our diffuse model e) is obtained by a simplistic approach. In contrast, the approach d) requires solving a PDE.

4.5 Experimental Results

In this section, we show extensive validations and demonstrate the performance of the proposed concept for solving figure hunt games.

The parameters used for the experiments are summarized in Table 4.1. Unless specified otherwise, diffusion is computed by *erosion followed by averaging* with Equations (4.12)-(4.14) where $S = 1$ (and $\sigma = 3$).

4.5.1 Propagation of the Contour Information

Depending on the application, one has to specify a) the intensity of the diffusion and b) if the placed target figure $\mathcal{F}_\mathcal{P}$ should be computed by means of the binary or

Table 4.1: **Parameters** of the optimization function and the average runtime per experiment. The diffusion is computed by means of Equation (4.14).

Results		Input					Diff. Prop.		Param. Ranges				
Fig.	Time	Illustration $\mathcal{I}$	Fig.	$\lvert\mathcal{I}\rvert = \mathcal{I}_c \times \mathcal{I}_r$	Targ. fig. $\mathcal{F}$	Fig.	of $\mathcal{F}$	of $\mathcal{I}$	θ	t_r	t_c	h_r	h_c
4.12	24s	Multiple bees	4.7	766×556	Bee	4.7	$S = 5$		$[0, 360]$	$[-\mathcal{I}_r, \mathcal{I}_r]$	$[-\mathcal{I}_c, \mathcal{I}_c]$	$[0.7, 1.5]$	h_r
4.16	21s	Collection of Cars	4.15a)	483×254	Car	4.15b)	$S = 4$			$[-\mathcal{I}_r, \mathcal{I}_r]$	$[-\mathcal{I}_c, \mathcal{I}_c]$	$[0.7, 1.3]$	h_r
4.15a)	64s	Collection of Cars	4.15a)	483×254	Car	4.15b)	$S = 1$			Eq. (4.15)	Eq. (4.16)	$[0.7, 1.3]$	h_r
4.23a)	38s	Hidden cloverleaf	4.18 l.	188×188	Cloverleaf	4.18 r.	$S = 1$	-		$[-\mathcal{I}_r, \mathcal{I}_r]$	$[-\mathcal{I}_c, \mathcal{I}_c]$	1	1
4.31	45s	Hidden pi	4.18 l.	188×188	Pi (π)	4.30a)	$S = 1$	-				$[0.7, 1.3]$	h_r
4.23b)	205s	Several hangers	4.21	304×321	Hanger	4.22a)	$S = 1$	-				$[0.8, 1.2]$	$[0.8, 1.2]$
4.25	176s	Butterfly mandala	4.24	250×250	Butterfly	4.24 r.	$S = 1$					1	1
4.32	210s	Butterfly mandala	4.24	250×250	Segment	4.24 l.	$S = 1$					1	1
4.33c)	97s	Hidden elephant	4.1c)	273×205	Elephant	4.33c)	$S = 1$		$[0, 360]$			1	1
4.27	209s	Mandala circles	4.26a)	550×550	Circle	4.26b)	$S = 1$		0			$[0.13, 2.9]$	h_r
4.28	811s	Triangles	4.26a)	1706×640	Triangle	4.26b)	$S = 4$	-	$[0, 360]$	$[-\mathcal{I}_r, \mathcal{I}_r]$	$[-\mathcal{I}_c, \mathcal{I}_c]$	$[0.3, 1.1]$	$[0.3, 3.2]$

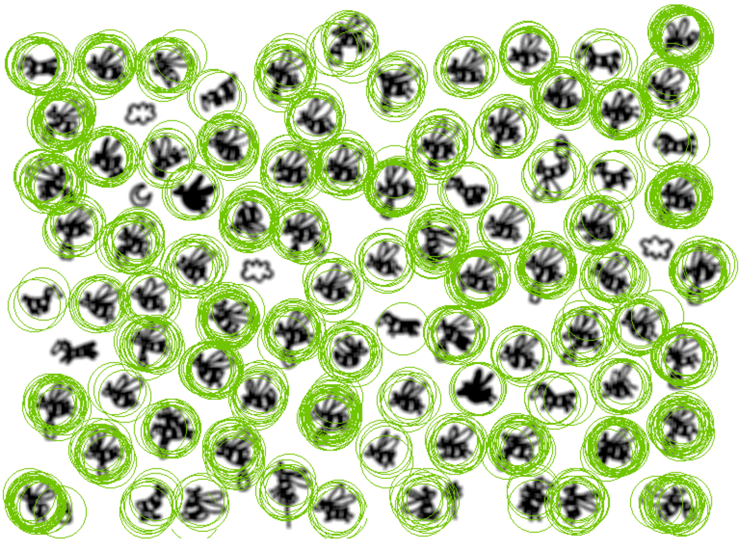


Figure 4.12: **All hypotheses** of 1000 independent runs of the algorithm on a coarse scale.

the diffuse illustration. Therefore, we categorize the figure hunt games as follows:

- *Where's Wally?* type images (Figure 4.1 bottom row)
- Gottschaldt type puzzles (Figure 4.1 top row)

Where's Wally? Type Images

Where's Wally? type illustrations consist of several objects being placed next to each other like Figures 4.7, 4.15. Different questions can arise here, like *e.g.*: "How many bees are in the bee swarm?", "Can you find the objects not belonging to the scene?" or "How many cars of the same type are there?". These questions can be allocated to two general problem settings:

1. Get a rough idea about the drawing (Figure 4.33 d).
2. Find the exact position of a given target figure (Figure 4.33 c).

Both problems can be approached in a first step by localizing the approximate positions of the objects using a strong diffusion (see *e.g.* Figures 4.12, 4.16). If additionally, the exact locations of the objects are desired, the resulting approximate positions can be used to constrain the search space for the search on the fine scale.

Swarm of Bees In order to analyze the swarm of bees, we searched for a strongly diffused version of the target figure marked by the blue circle in Figure 4.7 in the diffused illustration shown in the background of Figure 4.12. For the diffusion we

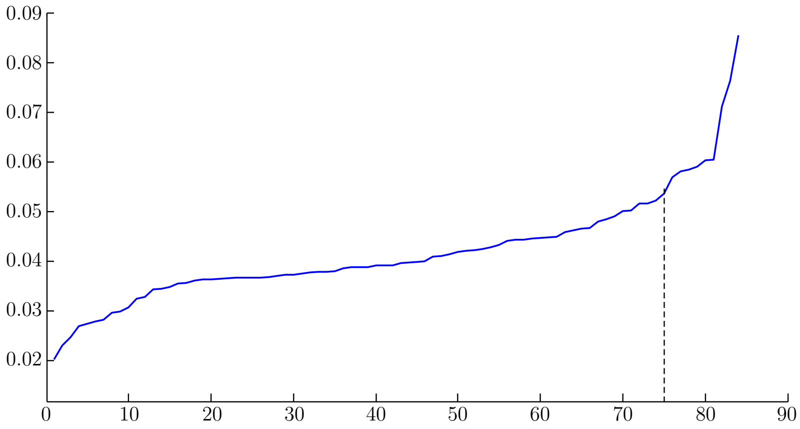


Figure 4.13: **Matching cost** corresponding to the 84 distinct locations. The matching cost is plotted against the hypotheses, ordered by increasing cost.

use Equation (4.14) and set $S = 5$. The ranges of the parameters $(\theta, t_{\mathrm{r}}, t_{\mathrm{c}}, h_{\mathrm{r}}, h_{\mathrm{c}})$ are set as follows: $\theta \in [0, 360]$, $t_{\mathrm{r}} \in [-\mathcal{I}_{\mathrm{r}}, \mathcal{I}_{\mathrm{r}}]$, $t_{\mathrm{c}} \in [-\mathcal{I}_{\mathrm{c}}, \mathcal{I}_{\mathrm{c}}]$, $h_{\mathrm{r}} \in [0.7, 1.5]$ and $h_{\mathrm{c}} := h_{\mathrm{r}}$.

Due to the strong diffusion, we are able to quickly detect the rough locations of the bees – marked by green circles in Figure 4.12 – in this large illustration (766×556 pixels). The number of circles around each bee is an evidence that good fits are found more often than bad ones. In the next step, we omit the duplicates and mark each location by exactly one circle. We obtain 84 distinct circles corresponding to 84 objects in the bee swarm.

By analyzing the matching costs corresponding to the 84 hypotheses we obtain the plot in Figure 4.13. For the first 74 hypotheses the energy increases steadily. In contrast, the energy ascends steeply for the last 10 hypotheses. Hence, we declare the worst 10 fits as objects not belonging to the bee swarm and indicate their position in orange. Figure 4.14 shows the 84 determined locations, whereof the worst 10 fits are indicated in orange. Observe that all orange marked objects are no bees.

In a second step, the approximate locations could be used to obtain a more exact location and orientation of the bees. Such a search on a fine scale will be explained in the next paragraph by means of the *collection of cars*.

Collection of Cars To find all cars of the same type as the target car (Figure 4.15 b) we use a strong diffusion for the target figure and the illustration, which is exemplary shown in Figure 4.15 c). Therefore, we use Equation (4.14) and set $S = 4$. On the coarse scale, the algorithm returns the locations illustrated in Figure 4.16. In a second step we use these locations to initialize the algorithm for the search on a fine scale. For each position obtained by the search on the coarse scale, an additional search with a slightly diffused target figure and illustration is carried out ($S = 1$). Thereby, we constrain the search space to a small area around these positions.

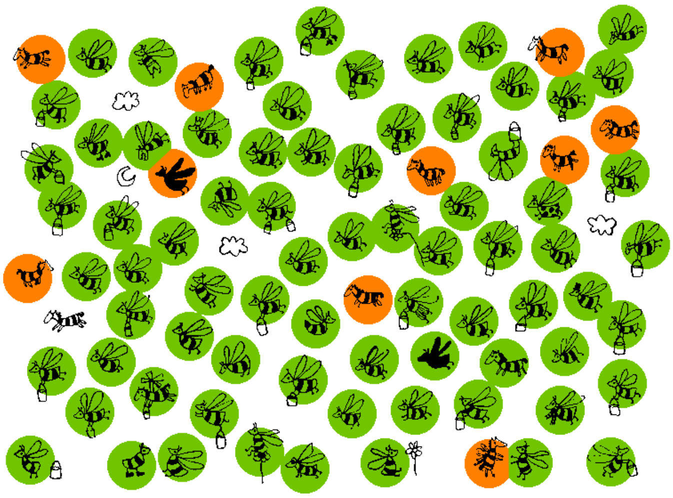
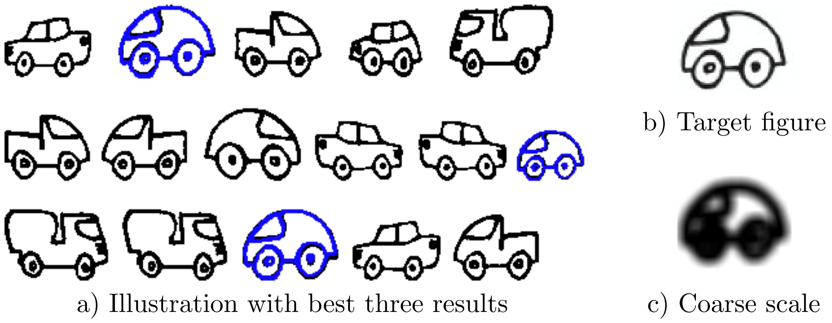


Figure 4.14: **Several dissimilar objects found.** The number of bees in the bee swarm can easily be counted by using our search algorithm together with a strong diffusion. We detected 84 objects, including 10 objects which do not belong to the bee swarm.



a) Illustration with best three results

b) Target figure

c) Coarse scale

Figure 4.15: **Three cars of the same type** are detected with the search on the fine scale. The search was initialized by the positions obtained by the search on the coarse scale (Figure 4.16). Additionally, the search space was constrained to a small area around these positions.

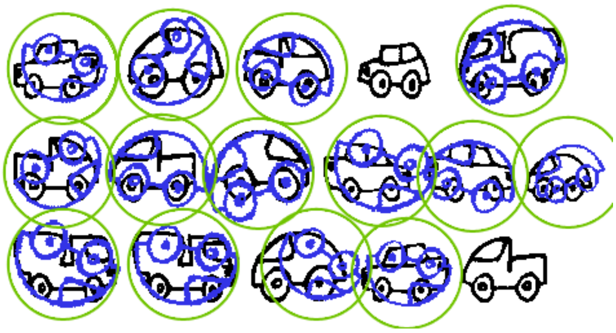


Figure 4.16: **Coarse-to-fine approach.** The search for a single target in a collection of multiple individual objects is performed in two steps: Search on 1) coarse scale; 2) fine scale. The search on the coarse scale returns the approximate positions of objects similar to the target figure.

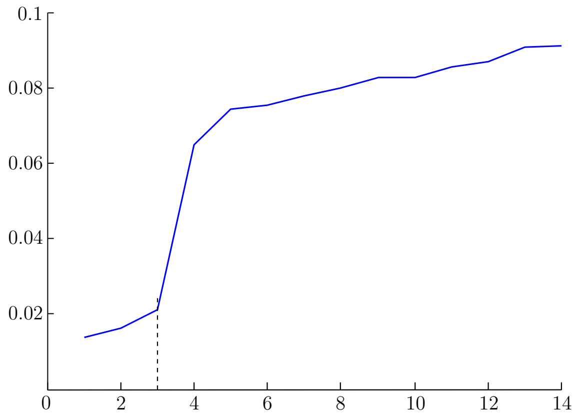


Figure 4.17: **Distinctive jump in the energy.** The graph of the matching costs corresponding to the hypotheses resulting from the search on the fine scale has a distinctive jump at the fourth hypothesis.

Let $(T_\mathrm{r}, T_\mathrm{c})$ be a position obtained by the search on the coarse scale and let $|\mathcal{F}| = \mathcal{F}_\mathrm{c} \times \mathcal{F}_\mathrm{r}$ be the size of the target figure $\mathcal{F}$. For the search on the fine scale, we restrict the parameter ranges of t_r and t_c as follows:

$$t_\mathrm{r} = \left[T_\mathrm{r} - \left\lceil\frac{\mathcal{F}_\mathrm{r}}{4}\right\rceil, T_\mathrm{r} + \left\lceil\frac{\mathcal{F}_\mathrm{r}}{4}\right\rceil\right], \tag{4.15}$$

$$t_\mathrm{c} = \left[T_\mathrm{c} - \left\lceil\frac{\mathcal{F}_\mathrm{c}}{4}\right\rceil, T_\mathrm{c} + \left\lceil\frac{\mathcal{F}_\mathrm{c}}{4}\right\rceil\right]. \tag{4.16}$$

The matching costs corresponding to the resulting hypotheses are plotted in Figure 4.17. A distinctive jump of the cost can be observed at the fourth hypothesis. The three hypotheses with the best energy are thresholded and the binary shapes are depicted in blue in Figure 4.15 a). Three cars of the given type occur in the image.

This two-step approach allows to get a rough analysis of the illustration followed by a precise definition of the deformation parameters in the second step. For the search on the coarse scale, a strong diffusion of both the illustration and the target figure is helpful to have as much information as possible throughout the image. For the subsequent search on the fine scale slightly diffused versions are used. The parameters used for both steps are summarized in Table 4.1.

Gottschaldt Type Puzzles

Gottschaldt type puzzles are line drawings where several lines overlap, like in Figures 4.18, 4.21. A figure hunt game might for example challenge to find the cloverleaf hidden in the line drawing in Figure 4.18. Another task might be to find one of the target figures drawn in Figure 4.22 in the collection of hangers, Figure 4.21. Two different problem settings can appear:

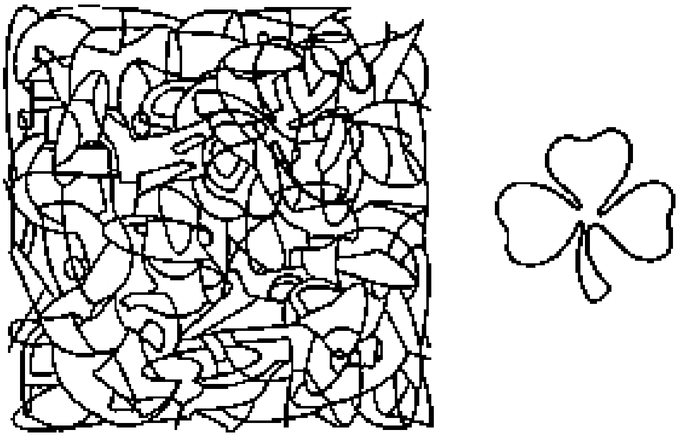
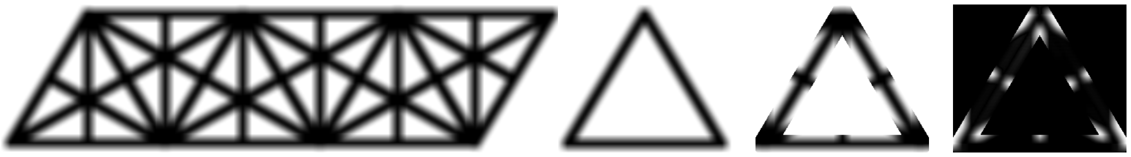


Figure 4.18: **Hidden cloverleaf.** Can you trace out the hidden cloverleaf?



a) Diffused illustration

b) Diffused target figure

c) Cutout of illustration

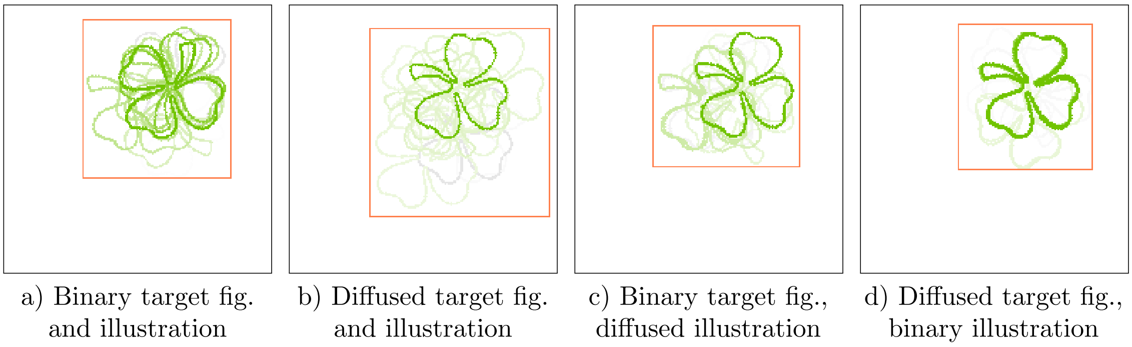
d) Difference b) - c)

Figure 4.19: **Black spots at intersections.** The diffused illustration shows black patches at the intersection of the lines. However, the diffused target figure does not include those dark spots. Hence, the visual matching cost d) of a perfect fit includes white spots.

1. The target figure $\mathcal{F}$ is a cutout of the illustration $\mathcal{I}$ (compare Figure 4.22 b). *I.e.* it reflects a complete segment of the illustration.
2. The target figure $\mathcal{F}$ does not reflect a complete segment of the illustration $\mathcal{I}$ (*e.g.* Figure 4.22 a). *I.e.* at a perfect position of $\mathcal{F}$ in the illustration, $\mathcal{I}$ has additional crossing lines (compare Figure 4.23).

The first type of problems where $\mathcal{F}$ is a cutout of $\mathcal{I}$ is mentioned in Section 4.5.3 and an example is shown in Figure 4.32.

For the second type of problems, a diffusion of both, the target figure and the illustration may lead to unwanted effects: Due to several overlaps of the lines within the drawing, dark black patches appear at the intersections. However, the diffused target figure does not reflect a complete segment of the illustration and hence does not have such black patches along the contour line. See Figure 4.19 for an exemplary illustration. Hence, the matching cost of a perfect fit in a Gottschaldt type puzzle is considerably higher than the matching cost of a perfect fit in a *Where's Wally?* type image. For these cases, we recommend the computation of the matching cost by means of the binary illustration (compare Figure 4.3 top row). By only diffusing the



a) Binary target fig. and illustration

b) Diffused target fig. and illustration

c) Binary target fig., diffused illustration

d) Diffused target fig., binary illustration

Figure 4.20: **Diffusion of target figure and/or illustration** computed by means of Equation (4.14) with $S = 1$. Hypotheses resulting from 10 individual runs of the algorithm. Strong green color indicates low matching cost. Observe that the best hypotheses are obtained by only diffusing the target figure.

Table 4.2: **Average results** by means of the binary/diffused target figure and illustration: Average deviation of the placed target figure from the optimal position, area spread percental to the size of $\mathcal{I}$ and average runtime per run. The best results are given in bold.

	Average deviation	Area spread perc. to $\|\mathcal{I}\|$	Average runtime
a) Binary $\mathcal{F}$, $\mathcal{I}$	12.464	0.327	38.428
b) Diffused $\mathcal{F}$, $\mathcal{I}$	23.588	0.471	38.395
c) Binary $\mathcal{F}$, diffused $\mathcal{I}$	14.809	0.289	39.961
d) Diffused $\mathcal{F}$, binary $\mathcal{I}$	**7.344**	**0.271**	**38.297**

target figure, information restricted to figural loci can be propagated to neighboring areas and at the same time the black-spot-problem can be preserved.

In Figure 4.20 and Table 4.2 we demonstrate the results of the different combinations of a binary and a diffused target figure and illustration by means of the cloverleaf line drawing (Figure 4.18). Strong green colors indicate a position which leads to a lower energy compared to the other hypotheses. The orange box indicates the *overall size of the spread*, *i.e.* the area where hypotheses are placed. The area spread percental to the size of the illustration $\mathcal{I}$ and the average deviation from the optimal position are summarized in Table 4.2. All hypotheses obtained with the binary target figure together with the binary illustration a) are misplaced and have approximately the same matching cost. In contrast, for the remaining cases, the hypotheses belonging to the correct position have a significantly lower energy than the misplaced hypotheses. Due to the diffusion in b-d) the contour information is propagated to the neighborhood making uninformative (white) pixels informative. Due to the black-spot-problem, the output in b) has the largest average deviation and area spread. The best matches are obtained in d) for the usage of the diffused target figure together with the binary illustration (Figure 4.21).

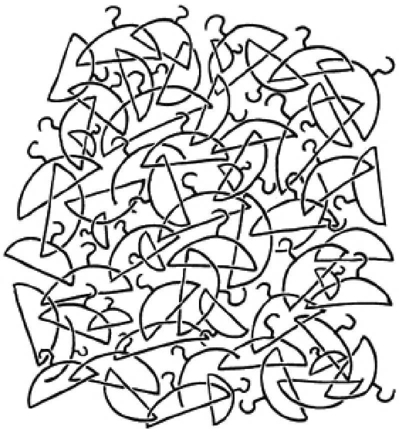
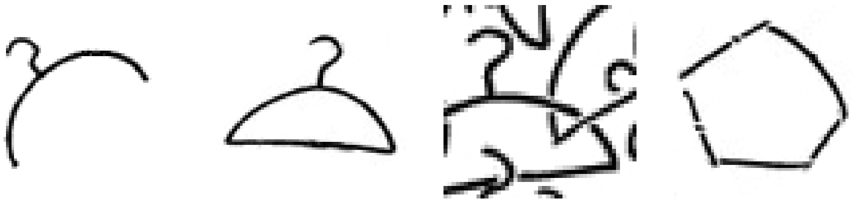


Figure 4.21: **Collection of hangers.**



a) Different hangers b) Segment c) Pentagon

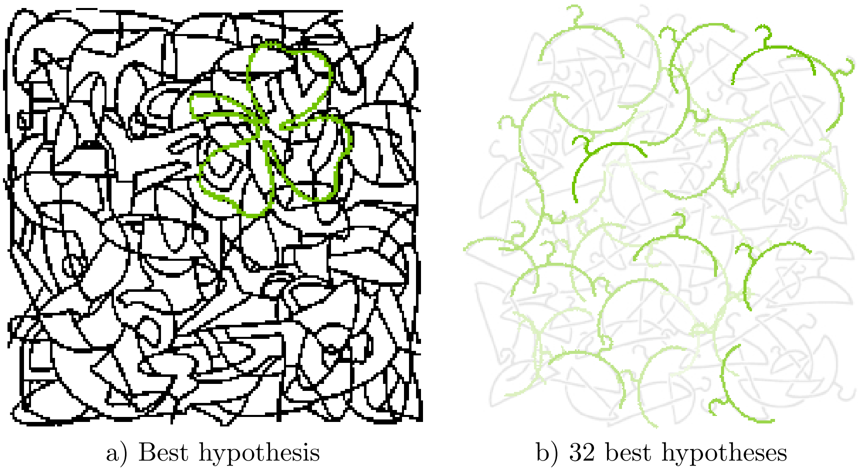
Figure 4.22: **Possible target figures** to search for: two types of hangers, a segment or a pentagon.

Hidden Cloverleaf To find the hidden cloverleaf, we use the approach in Figure 4.20 d) and compute the matching cost by means of the binary illustration (Figure 4.3 top row). The best hypothesis is depicted in Figure 4.23 a). The average runtime is 38.3 seconds.

Collection of Hangers The same combination of the diffused target figure and the binary illustration is used for the computation of the matching cost in the search for the hangers. The search for the hanger leftmost in Figure 4.22 a) leads to the hypotheses shown in Figure 4.23 b). Strong green colors indicate a position which leads to a lower energy compared to the other hypotheses.

4.5.2 Proof of Concept

In this section, we will demonstrate that our approach can handle the similarity transformations translation, rotation and scaling. Furthermore, based on our results we will show that the genetic algorithm together with the defined cost is reliable for the particular problem.


a) Best hypothesis

b) 32 best hypotheses

Figure 4.23: **Hypotheses** obtained by spreading the edge information via diffusion. In b) strong green colors indicate a position which leads to a lower energy compared to the other hypotheses.

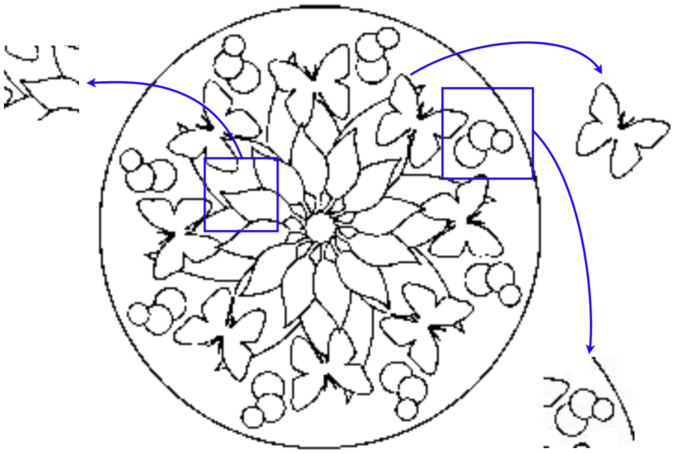


Figure 4.24: **Several target figures** taken from a mandala consisting of butterflies.

Robustness to Pose Variations

To observe the robustness with respect to pose, we consider a simple mandala pattern (Figure 4.24). Possible target figures are indicated by the blue boxes and arrows. For the first experiment the butterfly is chosen as target figure. We diffuse the target figure as well as the illustration by *erosion followed by averaging* (Section 4.4.4) with $S = 1$. Figure 4.25 shows the hypotheses of 100 individual runs of the genetic algorithm described in Section 4.2.2. To enhance the visibility, the hypotheses are thresholded and the binary shapes are drawn in shades of green. The different shades of green show the energy weighted by the number of detections. Strong

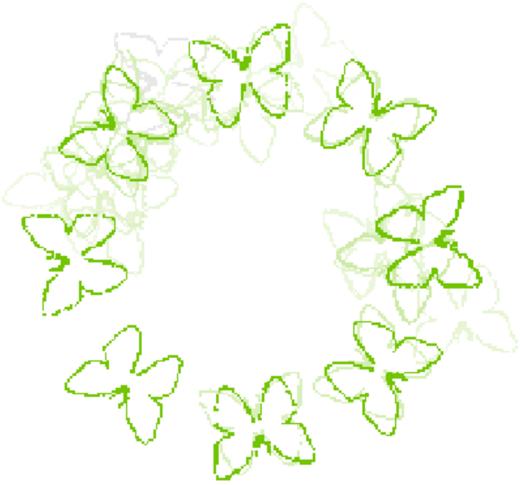


Figure 4.25: **Robustness to pose variations.** Independent of their location and rotation, all butterflies are found. Correct fits appear more often and have a lower energy.

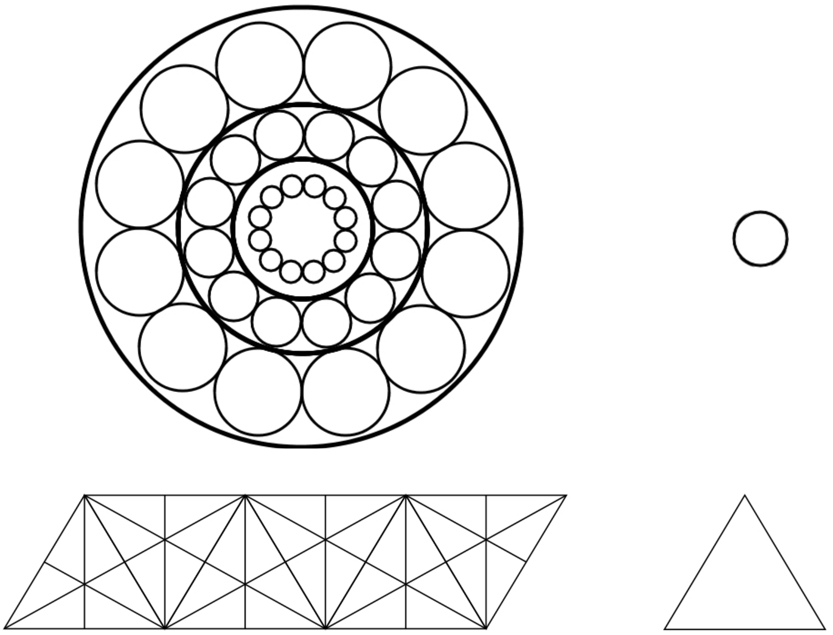
green colors indicate: a) a position which leads to a lower energy compared to the other hypotheses; and b) a match which has been returned more often in comparison to the other ones. Independent of the position and orientation, all butterflies in the mandala are successfully detected. Figure 4.25 provides experimental evidence that our method is robust to pose variations in the translational and rotational sense.

Robustness to Scale Variations

To evaluate the robustness to scaling we consider a composition of circles/triangles of varying size. One of the circles/triangles is selected as the target figure, see Figure 4.26. The goal is to find all occurrences irrespective of their scaling.

In Figure 4.27, we depict the energetically best 99 percent of circles detected after 1200 runs of the genetic algorithm. The same color coding as in Figure 4.25 was used. Observe that the method can handle scale variations. In particular, an unexpected and a very interesting solution is obtained: the innermost circle defined by the twelve smallest circles. This emergent circle may not be immediately perceivable.

In Figure 4.28, we depict selected triangles obtained by several runs of the algorithm. Observe that triangles with diverse edge length have been detected, *e.g.* the red triangle was elongated twice in x-direction and contracted in y-direction. The parameters used for these experiments together with the obtained average runtimes are summarized in Table 4.1.



a) Illustration b) Target figure

Figure 4.26: **Illustrations including figures of different scales.** The goal is to find all circles/triangles of arbitrary scale.

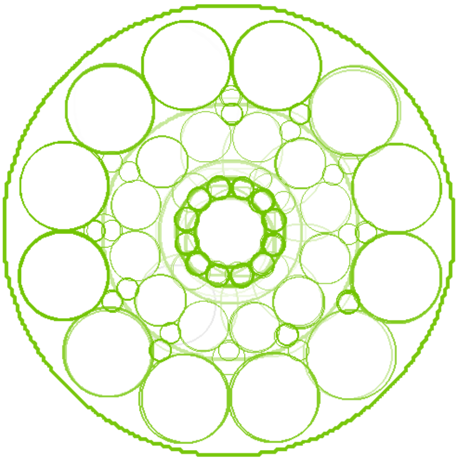


Figure 4.27: **Robustness to scale variations.** Circles of various scales are detected.

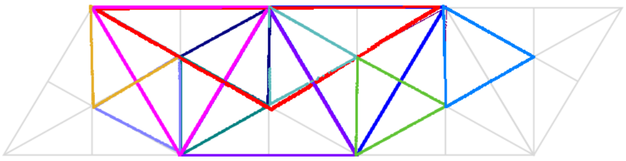


Figure 4.28: **Robustness to scale variations.** Triangles of arbitrary scale are detected. Due to clarity, only part of the best hypotheses are depicted in different colors.

Tendency to Return Good Fits

We evaluated whether the good fits (those of lower matching cost) are obtained more often than the bad fits. This is important as the algorithm is not a deterministic one.

We performed independent runs of the genetic algorithm, each run producing several hypotheses. We then computed the average of the batches of independent runs. As the results shown in Figures 4.25, 4.27 compellingly demonstrate, the algorithm has a tendency to return good fits more often than the bad ones. Furthermore, none of the bad fits has a nice green color. Hence, we can conclude:

1. The genetic algorithm has a tendency to return good fits:
   (a) Good fits appear more often than bad fits.
   (b) The same wrong fit is not detected several times.
2. The matching cost is an indicator for the goodness of the match:
   (a) A low matching cost indicates a good fit.
   (b) A large matching cost indicates a bad fit.

The Most Descriptive Diffusion Approach

In Section 4.3 we discussed the role of diffusion as key-ingredient of algorithmic search for target figures in a drawing. In this section we will compare the experimental results obtained with the introduced diffusion methods by means of the cloverleaf line drawing (Figure 4.18). For the diffusions the parameters are set as given in Section 4.3.

The hypotheses of 60 individual runs of the genetic algorithm obtained by using the different diffusion approaches introduced in Section 4.3 are depicted in Figure 4.29. Strong green colors indicate a match which has been returned more often in comparison to the other ones. In contrast to the figures shown above, the color-coding does not include the energy of the single hypotheses. The orange box again indicates the *overall size of the spread*. The area spread percental to the size of the illustration $\mathcal{I}$ and the average deviation from the optimal position are summarized in Table 4.3.

The hypotheses in Figure 4.29 a-c) have all about the same color, *i.e.*, none of the hypotheses was found more often than the others. In contrast, the hypotheses obtained with our proposed diffusion approach d) accumulate at the correct position. This fact reflects in the average deviation and the area spread indicated by the orange boxes and summarized in Table 4.3. Compared to the other diffusion approaches, the proposed *erosion and averaging* leads to a significantly smaller average deviation from the optimal fit and to the smallest area spread. The average runtimes are about

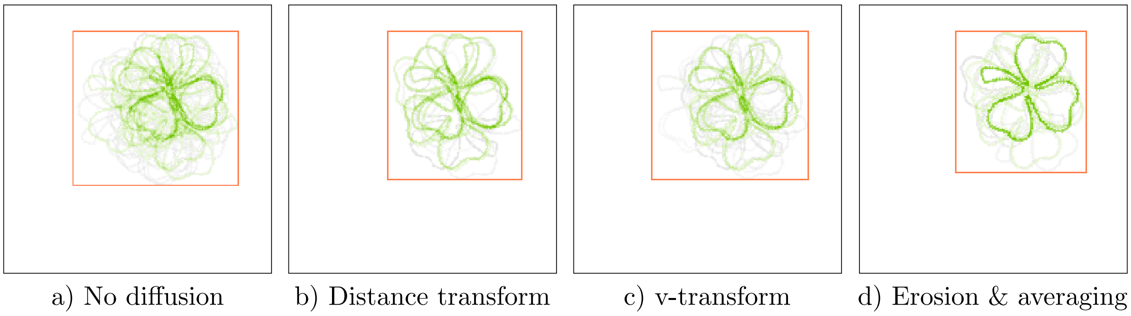

a) No diffusion b) Distance transform c) v-transform d) Erosion & averaging

Figure 4.29: **Results of 60 runs with different diffusion approaches.** Strong green colors indicate a match which has been returned more often in comparison to the other ones. The color-coding does not include the energy of the single hypotheses. Hence, with the proposed diffusion approach *erosion and averaging*, it is more likely that the target figure is placed at the correct position.

Table 4.3: **Smallest deviation with erosion and averaging.** Average results of 60 runs with the different diffusion approaches shown in Figure 4.29. The best results are given in bold.

	Average deviation	Area spread perc. to $\lvert\mathcal{I}\rvert$
a) No diffusion	16.481	0.354
b) Distance transform	12.432	0.277
c) v-transform ($\tau = 0.5$, $\rho = 4$)	12.732	0.324
d) Erosion and averaging	**7.632**	**0.258**

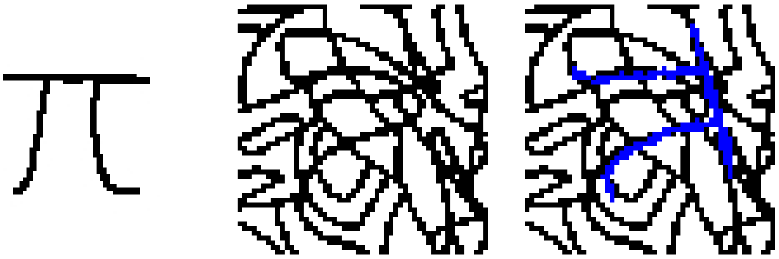
the same for the different diffusion approaches, however, using the v-transform the runtime increases due to the required solution of the PDE by a factor of twenty.

The resulting numbers point to the fact that our proposed diffusion approach *erosion and averaging* gives the best results.

4.5.3 Diverse Target Figures

Up to now, we focused on well-known shapes being included in the illustration. Of course our algorithm can also handle segments cut out of the illustration or target figures which are actually not contained in the drawing.

Figure 4.30 a) shows the target figure, pi, detected in the line drawing Figure 4.18. The algorithm determined the location where pi obtained the best matching cost. Hypotheses of 15 runs, weighted by the matching cost, are shown in Figure 4.31. The hypothesis leading to the best cost is thresholded and the binary shape is depicted in blue on the illustration and shown as a closeup in Figure 4.30 c). Indeed, the letter ‘pi’ is hidden in the line drawing.



a) Target figure

b) Zoom of the drawing

c) Located target figure

Figure 4.30: **Closest match.** A very good match is found. Indeed, the letter 'pi' is hidden in the line drawing.

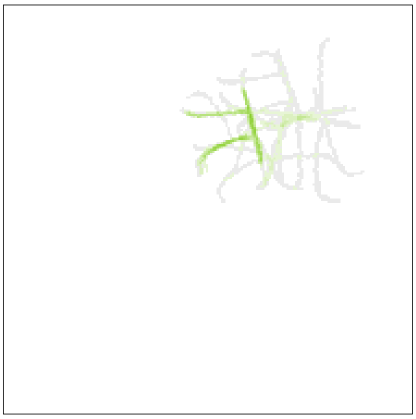


Figure 4.31: **Hypotheses for pi** weighted by their matching costs.

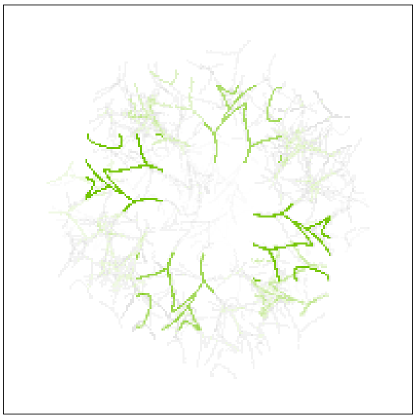
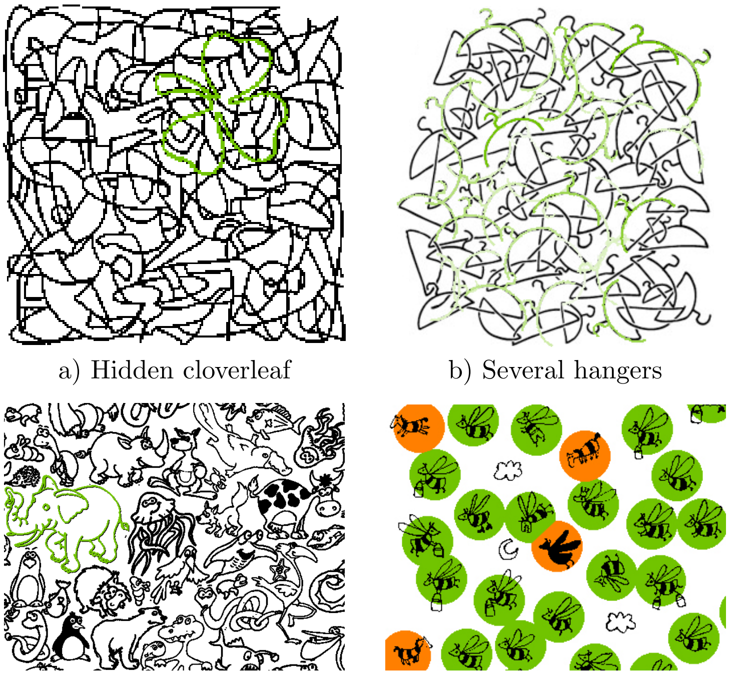


Figure 4.32: **A yet hidden symmetry appears** when using the target figure at the top left in Figure 4.24.

Another option is to search for a cutout of the drawing, like *e.g.*, the segment shown top left in Figure 4.24. The results of 100 independent runs searching for this segment are illustrated in Figure 4.32. The collection of butterfly locations in Figure 4.25 already revealed the outer circular structure of the pattern. The collection of these results leads to an emergence of a diamond scepter (as common in a mandala) together with a weak inner circular arrangement. A yet hidden symmetry appears.

4.6 Summary and Conclusion

We addressed the task of tracing out target figures in sketch-like binary teeming figure pictures. Some results of our algorithm are summarized in Figure 4.33. We can search for the unique occurrence of a target figure (left column) as well as for various similar objects (right column). Particularly suited to the task, we propose



a) Hidden cloverleaf

b) Several hangers

c) Hidden elephant

d) Multiple bees

Figure 4.33: **Desired results.** The proposed algorithm is able to localize the target figures.

a simple heuristic for generating diffuse drawings that imitate curvature coding distance images which are typically computed as solutions to elliptic PDEs. Our work extends the applications of diffusion based ideas to an interesting problem.

Acknowledgements We thank three anonymous reviewers for their constructive feedback. We also acknowledge the financial support from the Alexander von Humboldt Foundation and Tubitak through grant 112E208.

Chapter 5

Midrange Geometric Interactions for Semantic Segmentation

Authors
Julia Diebold[1] *julia.diebold@tum.de*
Claudia Nieuwenhuis[2] *cnieuwe@berkeley.edu*
Daniel Cremers[1] *cremers@tum.de*
[1]Technische Universität München, Munich, Germany
[2]ICSI, UC Berkeley, Berkeley, USA

Status Published

Publication J. Diebold, C. Nieuwenhuis, and D. Cremers. Midrange Geometric Interactions for Semantic Segmentation. Constraints for Continuous Multi-label Optimization. In *International Journal of Computer Vision (IJCV)*: *Special Issue on Graphical Models for Scene Understanding*. Springer US, 2015. doi: `10.1007/s11263-015-0828-7`

Individual contribution Leading role in realizing the scientific project

Problem definition	*significantly contributed*
Literature survey	*helped*
Method development & evaluation	*significantly contributed*
Implementation	*significantly contributed*
Experimental evaluation	*significantly contributed*
Preparation of the manuscript	*contributed*

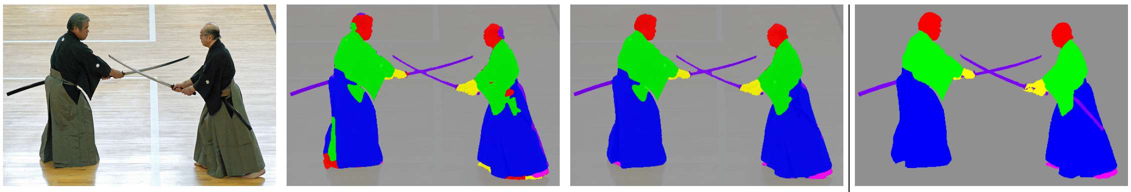
Abstract In this article we introduce the concept of midrange geometric constraints into semantic segmentation. We call these constraints 'midrange' since they are neither global constraints, which take into account all pixels without any spatial limitation, nor are they local constraints, which only regard single pixels or pairwise relations. Instead, the proposed constraints allow to discourage the occurrence of labels in the vicinity of each other, *e.g.* 'wolf' and 'sheep'. 'Vicinity' encompasses spatial distance as well as specific spatial directions simultaneously, *e.g.* 'plates' are found directly above 'tables', but do not fly over them. It is up to the user to specifically define the spatial extent of the constraint between each two labels. Such constraints are not only interesting for scene segmentation, but also for part-based articulated or rigid objects. The reason is that object parts such as for example arms, torso and legs usually obey specific spatial rules, which are among the few things that remain valid for articulated objects over many images and which can be expressed in terms of the proposed midrange constraints, *i.e.* closeness and/or direction. We show, how midrange geometric constraints are formulated within a continuous multi-label optimization framework, and we give a convex relaxation, which allows us to find globally optimal solutions of the relaxed problem independent of the initialization.

Keywords Variational · Image segmentation · Convex optimization · Directional relations · Geometric relations · Midlevel range interactions

5.1 Introduction

Semantic segmentation denotes the task of segmenting and recognizing objects based on class-specific information and/or knowledge of typical object relations. Ultimately, we aim at assigning an object label from a given pool of labels to each pixel in the image. In contrast to common segmentation problems, where little or no prior information is available, semantic segmentation makes use of knowledge such as color models, geometric relationships or the likelihood of object constellations, which can be learned from training data. Based on such information we can increase the accuracy of segmentation results and at the same time recognize specific objects instead of only detecting their boundaries.

Especially, the task of segmenting articulated objects is difficult. Animals usually share some common color or texture model, but humans usually wear variable clothes, which makes them hard to segment. Shape priors are often suited well to describe such objects, but they are usually too rigid and do not allow for large pose variations or occlusions. Besides, they are challenging for optimization due to their long-range relations between pixels leading to high-order potentials. We believe that constraints such as geometric relations between objects are generic enough to de-



a) Original image

b) Color-based segmentation

c) Segmentation with novel priors

d) Ground truth segmentation

Figure 5.1: **Midrange geometric constraints improve semantic segmentation results.** Midrange geometric constraints between labels allow the user to define specific spatial regions (by means of orientation and distance) within which constraints are enforced, *i.e.* specific label combinations are penalized. These constraints improve segmentation results, *e.g.* by imposing penalties for the head being below the jacket or for head and hands being close to the trousers.

scribe a wide range of objects and poses and still limit the ambiguity of color and texture models, features or object detectors, which usually operate on a single pixel or very limited pixel context.

Previous optimization approaches for semantic segmentation mainly make use of two types of constraint ranges: local or global ones. Local constraints are usually formulated on a pixel or pairwise pixel level, *e.g.* color likelihood constraints only consider the deviation of the local pixel color and the precomputed model. In contrast, global constraints are formulated based on the whole image, *e.g.* size [104, 110] and volume constraints [157, 158] or co-occurrence priors [82, 145]. What has been less explored so far in the context of optimization approaches are midlevel range interactions, *i.e.* interactions between pixels which are locally confined to a specific user-defined region around each pixel of a specific size, shape and direction.

We see mainly three fields of application of our novel constraints. First, there is the task of scene understanding, where geometric information is very useful to assign correct labels, *e.g.* knowing that ‘sky’ lies above ‘ground’, that ‘wolf’ and ‘sheep’ usually do not occur together or that ‘boats’ are usually surrounded by ‘water’. Second, there is the task of segmenting objects which consist of several parts, *e.g.* humans consist of ‘head’, ‘arms’, ‘legs’ and ‘torso’, or cars consist of ‘windshield’, ‘doors’, ‘headlights’, ‘bumpers’ and ‘tires’. For such objects there usually exist specific relations between their parts concerning their location, size and distance. Third, there are scenarios, where we have very specific knowledge of where different objects are located with respect to each other, *e.g.* when segmenting human clothes. There are no specific object parts, but specific rules about relative positions, and many labels can be missing in contrast to parts of objects.

In all three scenarios, the integration of geometric information into semantic segmentation will improve the labeling results, see Figure 5.1 for an example. The main challenge in this article is the formulation and efficient solution of a convex energy optimization problem, which allows for the integration of such additional geometric constraints.

5.1.1 Related Work

There has been growing interest in the topic of semantic segmentation in recent years, which combines different disciplines such as object detection, various features, shape priors, scene context information and learning. Especially the joint handling of several tasks such as segmentation, recognition and scene classification is beneficial for achieving results of higher quality, but has only recently been made possible by increased computing power.

The typical pipeline of such systems is the following: in the first step, some object detection, region segmentation or superpixel algorithm is used to obtain basic region proposals. In a second step different features are computed from these proposals, which are finally fed into a object classifier such as a random forest, a support vector machine or a neural network (*e.g.* [39]).

For example, in [12], Arbelaez *et al.* combined object detectors, poselets and different features such as color, shape and texture to a powerful semantic segmentation system, which can handle articulated objects in particular. The power of employing millions of features within a random forest approach was demonstrated by Fröhlich *et al.* in [59]. To learn such complex feature hierarchies from large amounts of training data, deep learning was used by Girshick *et al.* [62]. Instead of non-linear classifiers, Carreira *et al.* [38] demonstrated that second order statistics in conjunction with linear classifiers improve semantic segmentation results. A holistic approach to semantic segmentation and the full scene understanding problem which also includes geometric relations such as location or the spatial extent of objects or the type of scene was given by Yao *et al.* in [172].

In contrast to this typical pipeline processing, we aim at formulating a single optimization problem, which contains all information we have within a single energy. In this way we will be able to guarantee optimality bounds of the solution. To avoid ambiguous solutions which depend on the initialization we will give a convex relaxation of the energy.

The particular novelty of this article in contrast to previous discrete or continuous optimization approaches to semantic segmentation [2, 50, 82, 110, 112, 144, 145, 149] is the introduction of midrange geometric constraints between regions concerning relative location, distances and directions.

Ladicky *et al.* [82] and Souiai *et al.* [145] introduced co-occurrence priors into semantic segmentation which penalize the simultaneous occurrence of specific label combinations within the same image. In contrast to our approach, these constraints do not consider any spatial information such as location, direction or distance of objects. [82, 145] model co-occurrence by an additional cost function which can be seen as potentials of the highest order. MRF algorithms for high-order vision problem include [77, 79]. While higher order potentials are generally hard to optimize, the proposed approach is of order two and can be relaxed to a convex optimization problem which can be optimized with standard methods.

Strekalovskiy *et al.* [149] took in a way the opposite path and only penalize directly adjacent label combinations. It can be understood as a highly local co-occurrence prior. As the geometric relations in this approach are limited to directly adjacent pixels, they do not include distances or directions. In contrast to previous methods, the method does not require the distance penalty to be a metric but allows combinations which do not adhere to the triangle inequality. While labels ‘wolf’ and ‘grass’, for example, are common within an image and labels ‘sheep’ and ‘grass’ as well, sheep are rarely found next to wolves, which violates the triangle inequality. This often leads to one pixel wide ghost regions of hallucinated objects, which make transitions between two regions cheaper. Our approach does not suffer from ghost regions since our definition of neighborhood regards a larger number of pixels, which makes ghost regions very expensive.

Another type of global constraints for semantic segmentation are hierarchical constraints, which were introduced by Delong *et al.* [51] and Souiai *et al.* [144] and penalize the occurrence of objects from semantically different groups or scenes. Constraints relating different region sizes, *e.g.* of object parts, were introduced by Nieuwenhuis *et al.* [110]. These constraints are also global and integrate a notion of proportion and size into the segmentation, but they do not take into account distance or directional relations such as that the head of a person usually is above the body.

Topological constraints, which require that some label lies within another label, were proposed within a discrete optimization framework by Delong and Boykov [50] and within a continuous optimization framework by Nosrati *et al.* [112]. Geometric scene labeling has been studied by Felzenszwalb and Veksler [57] considering labelings that have a tiered structure. So-called ordering constraints, which require labels to only occur within a certain direction of other labels, were applied to geometric scene labeling by Liu *et al.* [93] for a specific five-part model. Strekalovskiy and Cremers [147] unified the existing approaches such as the five-regions and the tiered layout and proposed generalized ordering constraints. None of these constraints include any notion of label distance and thus cannot be considered as midrange constraints due to their global nature.

Finally, relative location based geometric relations have been introduced before into segmentation. In Gould *et al.* [65] the authors propose a two-stage process, which first uses an appearance model to assign labels and then employs a relative location prior based on the most likely label for each pixel in the first step to improve the segmentation. In contrast to our approach, this is a two-stage process and thus does not allow for any optimality guarantees. In the context of learning, relative spatial label distances have also been successfully applied, *e.g.* in [80, 131].

5.1.2 Contributions and Organization

In this article, we show how midrange geometric constraints characterized by label direction and distance can be integrated into variational semantic segmentation approaches. We give a convex relaxation of the energy minimization problem, which can be solved with fast primal-dual algorithms [122] in parallel on graphics hardware (GPUs). Results on various images and benchmarks show that the novel constraints improve semantic segmentation results.

The article is organized as follows: In Section 5.2 we give a formal definition of the multi-label segmentation problem together with different appearance models. In Section 5.3 we introduce the novel midrange geometric priors followed by a convex relaxation of the optimization problem in Section 5.4. In Section 5.5 we present results on various datasets and compare our segmentation results to state-of-the-art approaches.

5.1.3 Extensions and Improvements over the Previously Published Variant of our Model [2]

This journal paper extends our previously published ICCV workshop paper [2] by a more general formulation of the proximity priors to midrange geometric constraints and by more detailed and thorough evaluations on various image datasets.

The novel midrange geometric constraints are beneficial for the segmentation of part-based articulated and for part-based rigid objects as well as for the segmentation of scenes. The novel formulation allows to define different structuring elements for each label in contrast to only a single one in [2] (Sections 5.3.2, 5.3.3). We give an overview of different appearance models (Section 5.2.2), an analysis of different choices of structuring elements and the penalty matrix (Sections 5.3.3, 5.3.4) as well as a detailed explanation of the impact of different structuring elements (Section 5.3.5).

Additionally, we provide extensive evaluations including failure cases in Section 5.5. We present additional experiments on part-based articulated (Section 5.5.1) and part-based rigid objects (Section 5.5.2) on the CMU-Cornell iCoseg dataset [22], the People dataset [124] and the Penn-Fudan pedestrian database [167]. Moreover, we show results for the recognition of facades on the eTRIMS image database [81] and for the task of geometric class labeling of indoor images [93] (Section 5.5.3). Furthermore, we provide detailed insights about our experiments including the chosen parameters such as the structuring element $\mathcal{S}_i$ for label i, its size d and the choice of the penalty matrix A.

5.2 Variational Multi-label Segmentation

We begin with the formal definition of the multi-label segmentation problem and show several choices for the appearance term.

Although any numerical algorithm used for the implementation of the method presented below requires a discretization of the image domain, our general multi-label segmentation framework can be formulated continuously. We present the continuous setup below and give more details regarding the discretization and implementation in Section 5.4.2.

5.2.1 The Multi-label Optimization Problem

Let $I\colon \Omega \to \mathbb{R}^d$ denote the input image defined on the image domain $\Omega \subset \mathbb{R}^2$. The general multi-label image segmentation problem with $n \geq 1$ labels consists of the partitioning of the image domain Ω into n regions $\{\Omega_1, \ldots, \Omega_n\}$. This task can be solved by computing binary labeling functions $u_i\colon \Omega \to \{0,1\}$ in the space of functions of bounded variation (BV) such that $\Omega_i = \big\{x \mid u_i(x) = 1\big\}$. The BV space is important, since it allows jumps in the indicator functions which correspond to sharp transitions between adjacent regions. We compute a segmentation of the image by minimizing the following energy [174] (see [111] for a detailed survey and code)

$$E(\Omega_1, .., \Omega_n) = \frac{\lambda}{2} \sum_{i=1}^{n} \mathrm{Per}_g(\Omega_i) + \sum_{i=1}^{n} \int_{\Omega_i} f_i(x)\, dx. \tag{5.1}$$

f_i denotes the appearance model for the respective region Ω_i. Different ways to define f_i are discussed in Section 5.2.2. $\mathrm{Per}_g(\Omega_i)$ denotes the perimeter of each set Ω_i, which is minimized in order to favor segments of shorter boundary. These boundaries are measured with either an edge-dependent or a Euclidean metric defined by the non-negative function $g\colon \Omega \to \mathbb{R}^+$. For example,

$$g(x) = \exp\left(-\frac{|\nabla I(x)|^2}{2\sigma^2}\right), \; \sigma^2 = \frac{1}{|\Omega|}\int_{\Omega} |\nabla I(x)|^2 dx$$

favors the coincidence of object and image edges.

To rewrite the perimeter of the regions in terms of the indicator functions we make use of the total variation and its dual formulation [111, 120]:

$$\mathrm{Per}_g(\Omega_i) = \int_{\Omega} g(x)|Du_i| = \sup_{\xi_i\colon |\xi_i(x)| \leq g(x)} -\int_{\Omega} u_i \operatorname{div} \xi_i \; dx.$$

Since the binary functions u_i are not differentiable Du_i denotes their distributional derivative. Furthermore, $\xi_i \in \mathcal{C}_c^1(\Omega; \mathbb{R}^2)$ are the dual variables and $\mathcal{C}_c^1$ denotes the space of smooth functions with compact support. We can rewrite the energy in (5.1) in terms of the indicator functions $u_i \colon \Omega \to \{0,1\}$ [111, 174]:

$$\begin{aligned} E(u_1, .., u_n) &= \sup_{\xi \in \mathcal{K}} \sum_{i=1}^{n} \int_{\Omega} (f_i - \operatorname{div} \xi_i)\, u_i \; dx, \qquad (5.2)\\ \text{where } \mathcal{K} &= \left\{ \xi \in \mathcal{C}_c^1 \left(\Omega; \mathbb{R}^{2\times n}\right) \;\middle|\; |\xi_i(x)| \le \frac{\lambda g(x)}{2} \right\}. \end{aligned}$$

5.2.2 Choices of Appearance Models

In this article, we use different appearance models for the appearance term f_i in (5.2) depending on the task to solve.

Color Likelihoods

The simplest model is based on an estimated color probability distribution, *e.g.* by means of Parzen density estimators. Given a set of scribbles or training data we can extract RGB or HSV sample data for each label in the image or database. A Parzen density for a specific object class i with m_i color samples, each denoted by $I_{ij} \in \mathbb{R}^3$, is then given by

$$f_i(x) := -\log P_i\left(I\left(x\right)\right) := \frac{1}{m_i} \sum_{j=1}^{m_i} \frac{1}{\sqrt{(2\pi)^3 |\Sigma|}} \exp^{-\left((I-I_{ij})^T \Sigma^{-1} (I-I_{ij})\right)}. \qquad (5.3)$$

The density depends on the covariance matrix Σ of the multivariate Gaussian, which is usually a diagonal matrix and can be adapted by the user. Large values on the diagonal will assign a higher probability to less common colors. Low values on the diagonal will, in contrast, make the distribution more peaked. $|\Sigma|$ denotes the determinant of Σ. In order to avoid infinity values in the appearance term caused by color probabilities of 0 we modify the expression as follows

$$f_i(x) := -\log\Big(P_i\big(I\left(x\right)\big) \cdot (1-\epsilon) + \epsilon\Big), \qquad (5.4)$$

where ϵ is a very small constant close to 0.

Spatially Varying Color Likelihoods

In the case of scribble based segmentation we can make use of additional spatial information to estimate color likelihoods. The idea is that close to the scribble we are quite certain about the color in this location, which will be similar to the closest scribble points. On the contrary, far from the scribbles we have to deal with uncertainty in the color density estimation. This level of confidence depends on the distance to the closest scribble point of the current label. It can be integrated into the Parzen density estimator in (5.3) by computing a different covariance matrix $\Sigma_i(x)$ at each pixel proportional to the distance to the closest scribble of label i:

$$\Sigma_i\,(x) = \alpha \min_{j=1,..,m_i} |x - x_{ij}|_2, \tag{5.5}$$

where $\{x_{ij},\ j = 1, .., m_i\}$ is the set of user scribbles for region i and $\alpha \in \mathbb{R}$. This yields a space-dependent color density estimator. Details of this approach are given in [108].

Texton Likelihoods

In order to integrate not only color, but also shape and context information, Shotton *et al.* [139] proposed to learn a discriminative model to distinguish between object classes. This model is based on texton features, which incorporate shape and texture information jointly. Training is done by means of a shared boosting algorithm. Using the softmax function, the predicted confidence values $H_i\,(x)$ can be interpreted as a probability distribution. By taking the negative logarithm, we obtain the appearance model

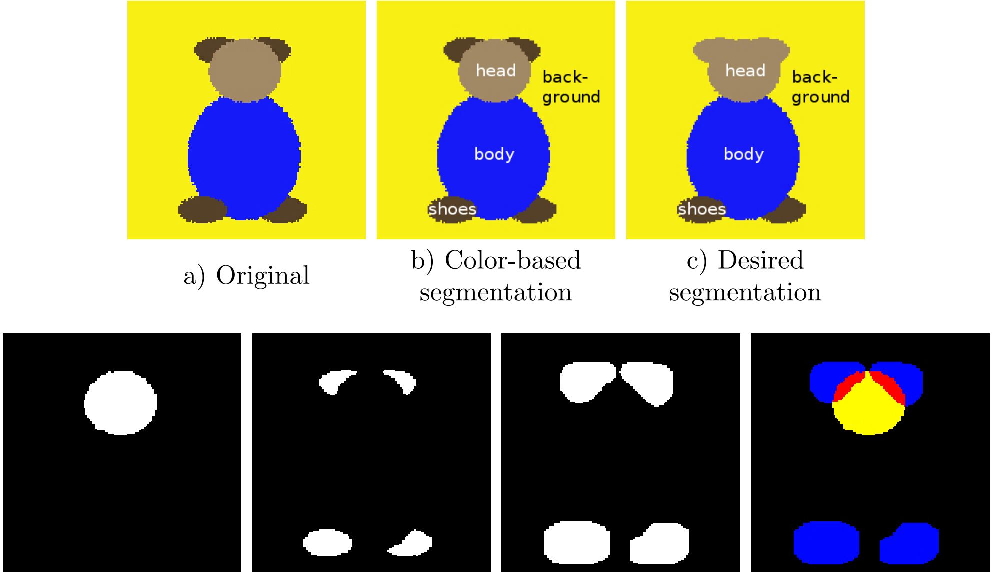
$$f_i(x) = -\log\Big(\frac{\exp\left(H_i\,(x)\right)}{\sum_{j=1}^{n}\exp\left(H_j\,(x)\right)}\Big), \tag{5.6}$$

which is also known as unary pixel potential. This model is computed with the ALE library [82, 83] and used for the experiments on the Penn-Fudan, eTRIMS and MSRC dataset in order to guarantee comparability to other approaches.

5.3 The Novel Midrange Geometric Priors

We motivate the midrange geometric priors by means of the simple artificial teddy bear example in Figure 5.2 a). Common segmentation approaches group pixels mainly according to their color, hence the ears of the bear are associated with the region ‘shoes’ (Figure 5.2 b). The desired result, however, would rather connect the ears to the head instead of the shoes as shown in Figure 5.2 c).

To obtain the desired solution, we make use of a *dilation*, an operation from mathematical morphology. To examine if two regions are close to each other in a certain direction we dilate one of the regions in this direction and compute the overlap between the dilation and the second region. For the teddy example, we want to penalize that head and shoes are close without considering any specific direction. Therefore, we enlarge the region ‘shoes’ in all directions simultaneously and compute the overlap with the region ‘head’ as shown in Figure 5.2. In this way, we do not only consider directly neighboring pixels as close but we consider



a) Original

b) Color-based segmentation

c) Desired segmentation

d) ‘Head’

e) ‘Shoes’

f) Dilation

g) Overlap

Figure 5.2: **Introducing midrange geometric priors.** First row: Color-based segmentation often fails. The ears of the bear are b) assigned the label ‘shoes’ instead of c) being combined with the label ‘head’. Second row: The novel priors can be used to penalize the ‘closeness’ of two labels, in this example d) ‘head’ and e) ‘shoes’. f) Dilation of the indicator function ‘shoes’; g) Overlap of the dilated region ‘shoes’ (blue) and the region ‘head’ (yellow). Appropriate penalties for such overlap (red) introduce semantic information into the segmentation.

proximity with respect to arbitrary neighborhoods of any size, shape or direction, which allows us to introduce midrange geometric constraints. The size and shape of these neighborhoods is determined by the *structuring element* of the dilation and can thus be easily adapted.

5.3.1 A Continuous Formulation of the Dilation

Dilation is one of the basic operations in mathematical morphology. Since we ultimately aim at introducing the dilation operation into a continuous energy optimization problem instead of using a suboptimal two-step procedure, we require a continuous formulation of the dilation, which can be defined as follows:

Definition 10 *(Dilation of an image [143])* Let $I\colon \Omega \to \mathbb{R}^d$ be an image and $\mathcal{S}$ a structuring element. The dilation of I by $\mathcal{S}$ is denoted by $\delta_{\mathcal{S}}\left(I\right)$. The dilated value at a given pixel $x \in \Omega$ is given as follows:

$$\left[\delta_{\mathcal{S}}\left(I\right)\right](x) = \sup_{z \in \mathcal{S}} I(x+z). \tag{5.7}$$

Thus, the dilation result at a given location x in the image is the maximum value of the image within the window defined by the structuring element $\mathcal{S}$, when its origin is at x.

5.3.2 Introducing Midrange Geometric Constraints

To compute the proximity of two labels, we first introduce the notion of an extended region indicator function u_i denoted by $d_i \colon \Omega \to \{0, 1\}$, which dilates the indicator function in a specific direction and distance (see Figure 5.2 and Definition 10):

$$d_i(x) := \left[\delta_{\mathcal{S}_i}(u_i)\right](x) = \sup_{z \in \mathcal{S}_i} u_i(x+z). \tag{5.8}$$

The set $\mathcal{S}_i$ determines the type of geometric spatial relationship we want to penalize for label i, *i.e.* distance and direction, for example 'less than 20 pixels above'. $\mathcal{S}_i$ is often denoted by *structuring element*. We will give a more detailed explanation of $\mathcal{S}_i$ in Section 5.3.3.

To detect if two regions i and j are close to each other, we compute the overlap of the extended indicator function d_i and the indicator function u_j, as shown in bright red in Figure 5.2 g). For each two regions i and j we can now penalize their proximity by means of the following energy term:

$$E_{geom}(u) = \sum_{1 \le i < j \le n} A(i,j) \int_{\Omega} d_i\,(x)\, u_j\,(x)\, dx. \tag{5.9}$$

The penalty matrix $A \in \mathbb{R}^{n \times n}_{\geq 0}$ indicates the penalty for the occurrence of label j in the proximity of label i. Information on how to define or learn this matrix are given in Section 5.3.4.

5.3.3 Structuring Elements

The dilation operation requires a *structuring element* (SE) for probing and expanding label indicator functions. The option to use structuring elements of different sizes and shapes is one of the major benefits of the proposed algorithm.

There are many different ways to define SEs. We can specify one set $\mathcal{S}_i$ for each label i. If $\mathcal{S}_i$ is for example a line we can penalize the proximity of specific labels in specific directions, *e.g.* the occurrence of a book below a sign (compare Figure 5.4 c). Symmetric sets of specific sizes consider the proximity of two labels without preference of a specific direction. Sparse sets $\mathcal{S}_i$ as shown in Figure 5.3 c,d) lead to similar results but can speed up the runtime. Examples for structuring elements are shown in Figure 5.3 and their application in Figure 5.4.

The larger $\mathcal{S}_i$ the more pixels are considered adjacent to x. Let the occurrence of label j in the proximity of label i be denoted by $i \sim_{\mathcal{S}_i} j$. If training data is available

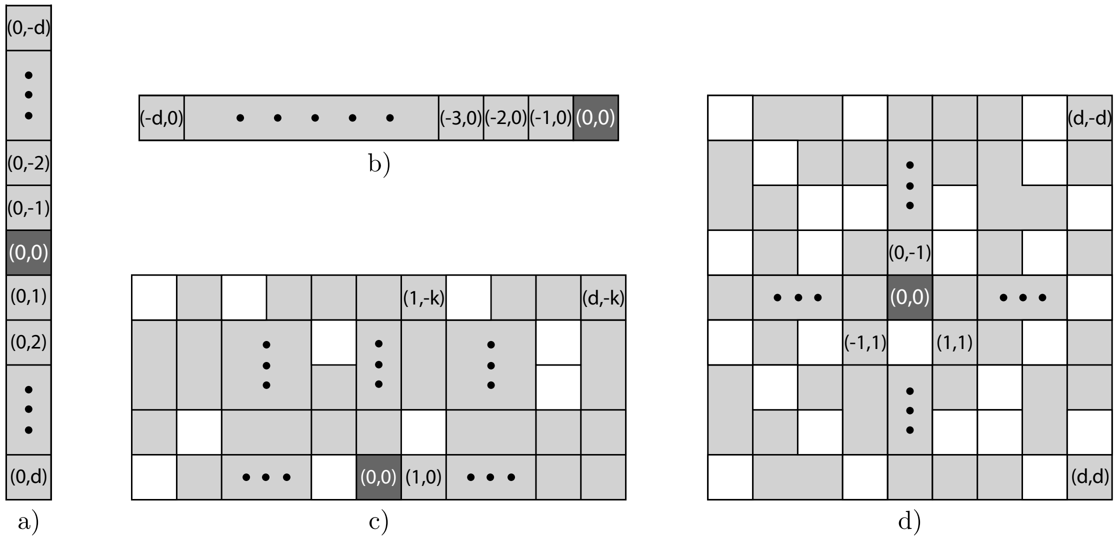
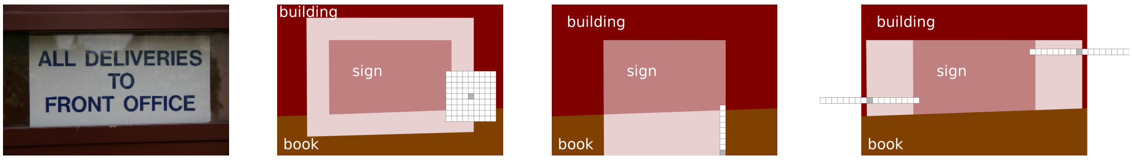


Figure 5.3: **Horizontal, vertical and sparse structuring elements.** Knowledge of the occurrence of regions *above/below* or *left/right* within a distance d can be included by using different structuring elements. Each structuring element has an origin which is indicated in dark gray. a) The vertical line dilates a region d pixels upward and downward, b) the horizontal line centered on the rightmost pixel enlarges a region d pixels to the right. c,d) To save computation time sparse structuring elements can be used. White pixels are chosen randomly and left out, *i.e.* they are not included in the set $\mathcal{S}$. c) A sparse element, which dilates to the bottom, right and left. d) A sparse element, which dilates equally in all directions and thus only regards pixel distance.



a) Original image

b) - d) Indicator function extended by different sets $\mathcal{S}_i$.

Figure 5.4: **Impact of structuring elements.** Different sets $\mathcal{S}_i$ convey different geometric priors. The light pink color illustrates the extended 'sign' region. b) Symmetric sets $\mathcal{S}_i$ only consider object distances, but are indifferent to directional relations. c) If $\mathcal{S}_i$ is chosen as a vertical line centered at the bottom, the indicator function of the region 'sign' is extended to the bottom of the object, *e.g.* penalizing 'book' appearing closely (within d pixels) below 'sign'. d) Horizontal lines penalize labels to the left and right.

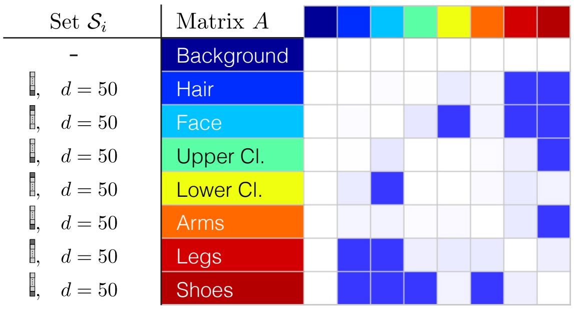


Figure 5.5: **SEs and penalty matrix A learned on the Penn-Fudan training set** (label colors according to benchmark conventions). We penalize the labels 'lower clothes', 'legs' and 'shoes' above 'face', as well as 'hair', 'face', 'upper clothes' and 'arms' below 'shoes'.

we can learn the probabilities $P\left(i \sim_{\mathcal{S}_i} j\right)$ for different types and sizes of SEs and then define $\mathcal{S}_i$ as the SE which provides the highest information gain for label i.

The information gain for a label i and structuring element $\mathcal{S}$ can be computed by means of the Shannon entropy [135]:

$$H(i, \mathcal{S}) = - \sum_{\substack{j \in \{1,..,n\} \\ j \neq i}} P\left(i \sim_{\mathcal{S}} j\right) \cdot \log\left(P\left(i \sim_{\mathcal{S}} j\right)\right). \tag{5.10}$$

The probabilities $P\left(i \sim_{\mathcal{S}} j\right)$ can *e.g.* be obtained by estimating the relative frequencies of the labels within the range of the selected structuring element $\mathcal{S}$ in the training data. We can either treat the relative frequencies as a joint probability distribution, which requires normalization by the sum of all elements, or we can treat it as a conditional distribution, which requires normalization per label separately. In the first case, the occurrence probability of each label is inherently part of the estimated probability distribution, *i.e.* labels occurring rarely in the training data also occur rarely close to other labels. The second case removes the influence of the frequency of label occurrences and only judges if a second label is common within the vicinity of a first label, which is already given. A slightly different way, which does not involve probability distributions, is to count all pairwise label co-occurrences in the training data weighted by their inverse distance in a matrix $B_{\mathcal{S}}$, to normalize $B_{\mathcal{S}}$ and then to estimate $P\left(i \sim_{\mathcal{S}} j\right)$ by $B_{\mathcal{S}}(i, j)$. For the Penn-Fudan dataset and different types and sizes of SEs for each label (except the 'background'), for example, we use the latter approach and obtain the SEs in Figure 5.5.

Note that the optimal structuring element $\mathcal{S}_i$ for label i will be dependent on the viewpoint. According to whether a scene is captured from a front or a top view, the size, shape and position of the objects in the scene varies in the captured image. Hence, to define one structuring element $\mathcal{S}_i$ for all labels i in a benchmark, some uniformity of the training and test images has to be assumed.

If a learning approach is not desired or not possible due to lack of training images

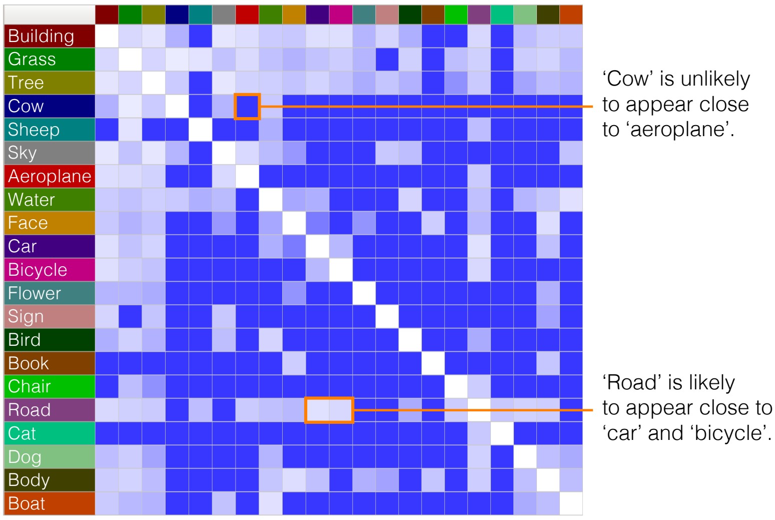


Figure 5.6: **Penalty matrix A learned on the MSRC training data** (objects are color coded corresponding to benchmark convention in first row and column). The lighter the color the more likely is the co-occurrence of the corresponding labels within the relative spatial context, and the lower is the corresponding penalty.

or non-uniformity of the dataset, appropriate sets $\mathcal{S}_i$ can easily be chosen manually as done for the experiments in Figures 5.9 and 5.10 in Figure 5.7.

5.3.4 Specification of the Penalty Matrix

To introduce the novel geometric priors into the original optimization problem in (5.2), we have to define the penalty matrix $A \in \mathbb{R}^{n\times n}_{\geq 0}$ in (5.9). Each entry $A(i,j)$, $i \neq j$ indicates the penalty for the occurrence of label j in the proximity of label i, where the proximity is defined by the respective structuring element $\mathcal{S}_i$. For $i = j$ we set $A(i,i) := 0$.

If training data is available we can learn the probabilities $P\left(i \sim_{\mathcal{S}_i} j\right)$ as described in Section 5.3.3 and define the entries $A(i,j)$ for label j being close to label i, *e.g.* by $A(i,j) := \min(-\log(P\left(i \sim_{\mathcal{S}_i} j\right)), m)$ with a fixed number $m \in \mathbb{N}$. This assigns a penalty close to zero to frequent and a penalty of m to less frequent co-occurrences. For the MSRC benchmark and a symmetric set $\mathcal{S}$ of size 9 × 9 for all labels, for example, we estimate $P\left(i \sim_{\mathcal{S}} j\right)$ by $B_{\mathcal{S}}(i,j)$ (compare Section 5.3.3) and define $A(i,j) := \min(-\log(B_{\mathcal{S}}\left(i,j\right)), 20)$ and obtain the penalty matrix in Figure 5.6. The first column in Figure 5.6, *e.g.*, indicates that the occurrence of 'building' close to 'tree' or 'sky' is very likely (light colored cells), whereas the occurrence of 'building' close to 'sheep' is very unlikely (dark colored cell).

If there is no appropriate training data available or if a learning approach is not desired, the penalty matrix A can easily be defined by hand as done for the experiments in Figures 5.9 and 5.10 in Figure 5.7.

Set $\mathcal{S}_i$	Matrix A		🟥	🟨	🟩	🟪	⬜
▦, $d = 15$	Head	🟥	0	12	0	12	0
▦, $d = 20$	Arms	🟨	12	0	0	12	0
▯, $d = 50$	Shirt	🟩	12	12	0	0	0
▤, $d = 20$	Legs	🟪	24	24	0	0	0
-	Background	⬜	0	0	0	0	0

Set $\mathcal{S}_i$	Matrix A		🟥	🟦	🟦	🟨	🟪	⬜	🟪
-	Head	🟥	0	0	0	0	0	0	0
▯, $d = 20$	Jacket	🟦	10	0	0	0	0	0	0
▦, $d = 20$	Trousers	🟦	10	0	0	50	0	0	0
-	Hands	🟨	0	0	0	0	0	0	0
-	Feet	🟪	0	0	0	0	0	0	0
-	Background	⬜	0	0	0	0	0	0	0
▦, $d = 25$	Weapon	🟪	10	0	0	0	0	0	0

Figure 5.7: **Penalty matrix A and corresponding structuring elements (SE)** defined to improve the segmentation results in Figures 5.9 and 5.10. For each label a specific SE with specific size d has been chosen by the user. For each label pair the corresponding matrix entry indicates the penalty in case these labels appear close to each other in the specified direction.

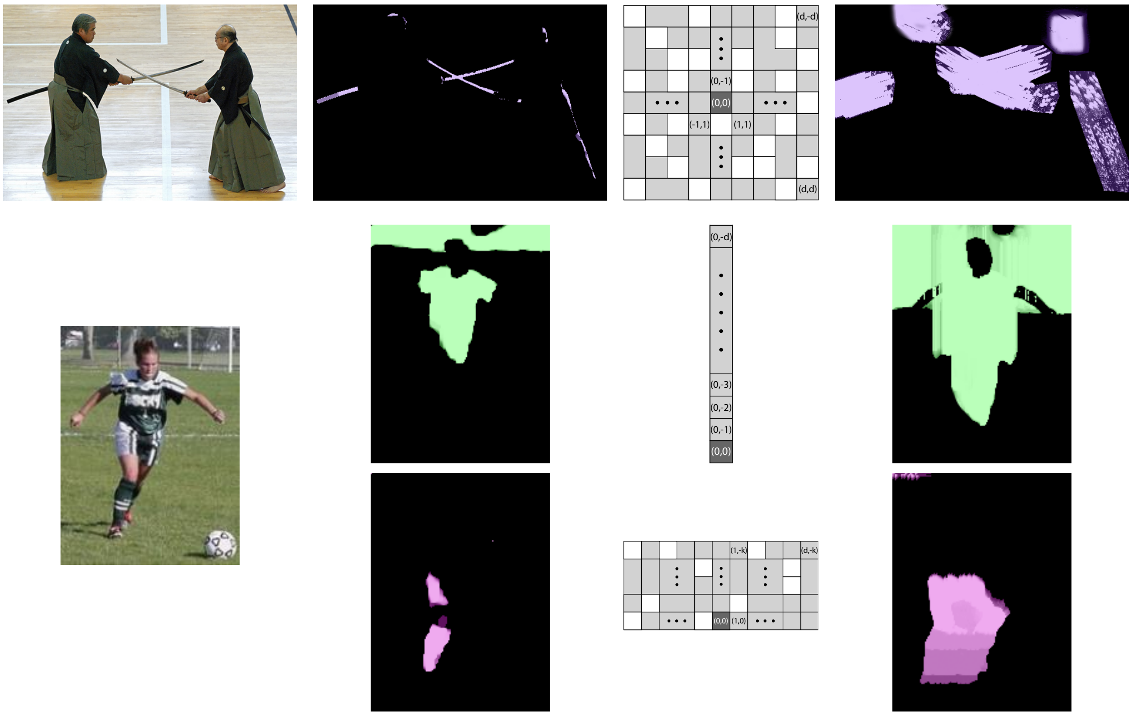
5.3.5 Real-World Examples

We demonstrate the impact of the novel midrange geometric priors by means of two examples shown in Figures 5.8, 5.9 and 5.10. The corresponding color-legend can be found in Figure 5.11.

Figure 5.7 gives an overview of the structuring elements and the penalty matrices defined for the segmentation of the soccer player and the fighters. For each label i an individual structuring element $\mathcal{S}_i$ with specific size d has been defined by the user. In the example of the soccer player we penalize the label 'head' being close to 'arms', below 'shirt' or below 'legs', as well as 'arms' below 'shirt' or 'legs'. For the fighters, we penalize the occurrence of 'head' below 'jacket', close to 'trousers' or close to 'weapon'. Furthermore, we penalize 'hands' next to 'trousers'.

Figure 5.8 shows the generation of the extended indicator functions d_i by means of different structuring elements from the original indicator functions u_i.

Figures 5.9 and 5.10 show how segmentation results can be improved by imposing midrange geometric constraints by penalizing the overlap of the specified indicator functions.



a) Original image b) u_i c) Set $\mathcal{S}_i$ d) $d_i = \delta_{\mathcal{S}_i}(u_i)$

Figure 5.8: **Effect of different sets $\mathcal{S}_i$** shown by means of the extended region indicator functions d_i. a) Original images and b) indicator functions u_i for the 'weapon' region of the fighters and the 'shirt' and 'legs' region of the soccer player. c) Sets $\mathcal{S}_i$ chosen for the dilation. Top: Symmetric sets $\mathcal{S}_i$ consider proximity in all directions. Center: If $\mathcal{S}$ is chosen as a vertical line centered at the bottom, the indicator function of the region 'shirt' is extended to the bottom of the object, *e.g.* penalizing 'head' appearing below 'shirt'. Bottom: Horizontal lines penalize labels to the left and right and can be extended to probe *downwards* to the left and right. Sparse sets save runtime. d) Extended indicator functions obtained with $\mathcal{S}_i$.

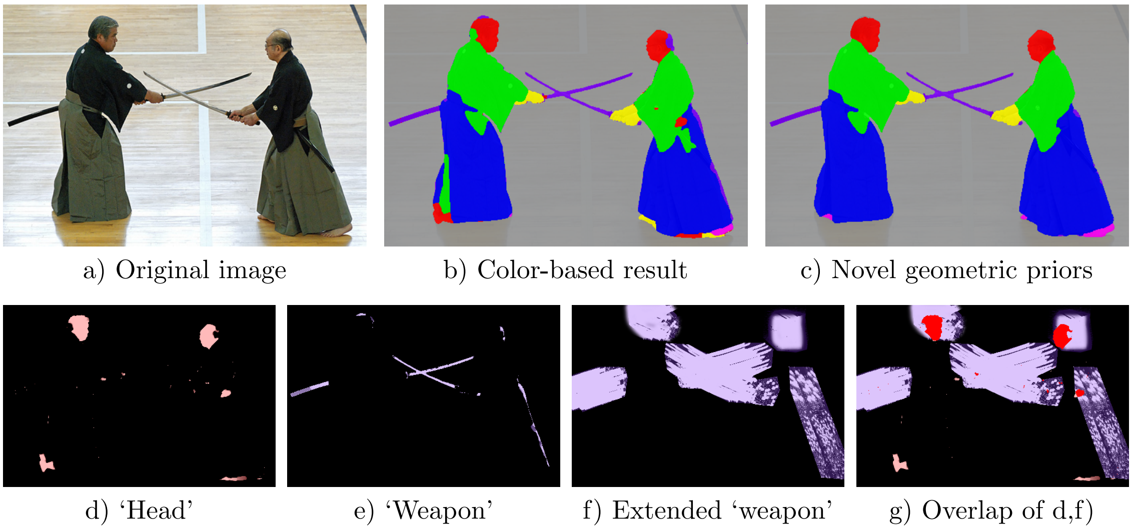


Figure 5.9: **Penalization of the proximity** of the labels 'head' and 'weapon' for the fighter image to improve the color-based segmentation in b), see Figure 5.11 for a color legend. d,e) Region indicator functions u_i for 'head' and 'weapon'. f) Extended region indicator function d_i for 'weapon'. g) Bright red indicates the penalized overlap. The overlap of the head with the weapon forces the weapon to retract in the area of the head. Using several geometric constraints together yields the result in c).

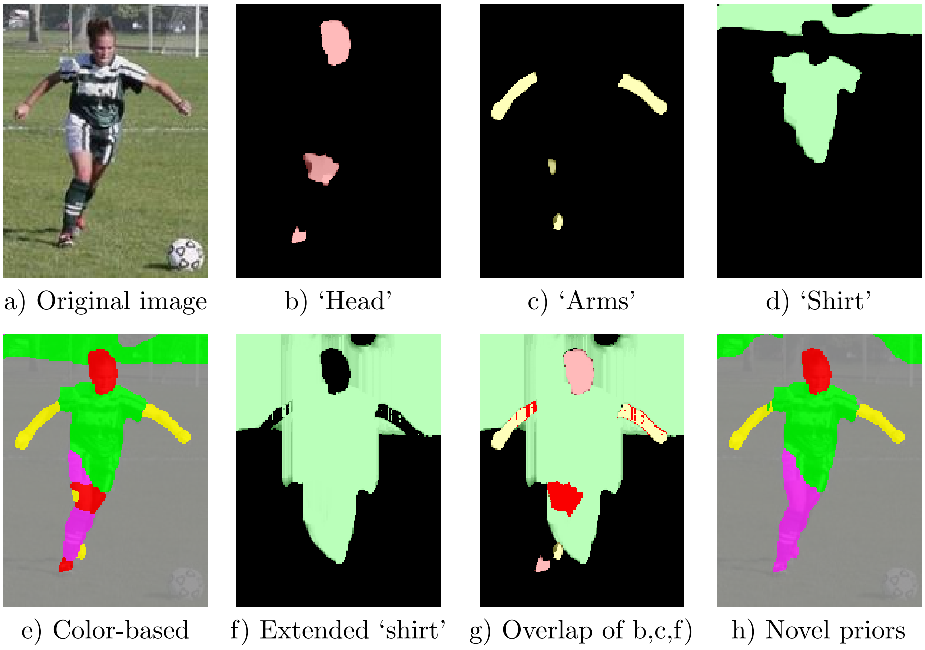


Figure 5.10: **Effect of novel geometric constraints**, which improve the color-based segmentation in e) for the soccer player image, see Figure 5.11 for a color legend. b-d) Region indicator functions u_i for 'head', 'arms' and 'shirt'. f) Extended indicator function d_i for the shirt region. g) The overlap of the head and the arms with the dilated shirt force the shirt in the top of the image to retract and the head to disappear from the trousers.

5.4 Integrating the Geometric Constraints into a Convex Optimization Problem

After introducing and defining the novel midrange geometric constraints (5.9) with $A \in \mathbb{R}^{n\times n}_{\geq 0}$ it remains to integrate these constraints into the original convex optimization problem for segmentation (5.2)

$$\min_{u\in\mathcal{G}} \quad E(u) + E_{geom}(u) = \tag{5.11}$$

$$\min_{u\in\mathcal{G}} \quad \sup_{\xi\in\mathcal{K}} \sum_{i=1}^{n} \int_{\Omega} (f_i - \operatorname{div}\xi_i)\, u_i \; dx + \sum_{1\leq i<j\leq n} A(i,j) \int_{\Omega} d_i u_j \; dx \tag{5.12}$$

$$\text{s.t.} \quad d_i(x) = \left[\delta_{\mathcal{S}_i}(u_i)\right](x) = \sup_{z\in\mathcal{S}_i} u_i(x+z), \tag{5.13}$$

$$\mathcal{G} = \left\{ u \in BV\left(\Omega; \{0,1\}^n\right) \,\Big|\, \sum_{j=1}^{n} u_j\left(x\right) = 1 \;\; \forall\, x \in \Omega \right\}, \tag{5.14}$$

$$\mathcal{K} = \left\{ \xi \in \mathcal{C}^1_c\left(\Omega; \mathbb{R}^{2\times n}\right) \;\Big|\; |\xi_i(x)| \leq \frac{\lambda g(x)}{2} \right\}. \tag{5.15}$$

5.4.1 A Convex Relaxation of the Midrange Geometric Constraints

In the following we will propose a convex relaxation of the segmentation problem (5.2) combined with the proposed priors in (5.9) as stated in (5.11)-(5.15). To obtain a convex optimization problem, we require convex functions over convex domains.

Relaxation of the Binary Functions u_i

The general multi-labeling problem is not convex due to the binary region indicator functions $u_i \colon \Omega \to \{0,1\}$ in (5.14). To obtain a convex problem where each pixel is assigned to exactly one label, instead of optimizing over the set $\mathcal{G}$ in (5.14) optimization is carried out over the convex set

$$\mathcal{U} = \left\{ u \in BV\left(\Omega; [0,1]^n\right) \,\Big|\, \sum_{j=1}^{n} u_j\left(x\right) = 1 \;\; \forall\, x \in \Omega \right\}.$$

Relaxation of the Dilation Constraints

The dilation constraints in (5.13) are relaxed to

$$d_i\left(x\right) \geq u_i\left(x+z\right) \quad \forall\, x \in \Omega, \; z \in \mathcal{S}_i. \tag{5.16}$$

By simultaneously minimizing over the functions d_i we can assure that at the optimum d_i fulfills the constraints in (5.13) exactly. The inequality (5.16) can easily be included in the segmentation energy by introducing a set of Lagrange multipliers β_{i_z} and adding the following energy term:

$$\begin{aligned}
&\min_{d\in\mathcal{D}} \max_{\beta\in\mathcal{B}} \sum_{i=1}^{n} \sum_{z\in\mathcal{S}_i} \int_{\Omega} \beta_{i_z}(x) \big(d_i(x) - u_i(x+z)\big)\, dx, && (5.17)\\
&\mathcal{B} = \left\{\beta_{i_z} \;\middle|\; \beta_{i_z}\colon \Omega \to [-\infty, 0] \quad \forall\, z \in \mathcal{S}_i,\ i = 1, .., n\right\},\\
&\mathcal{D} = BV\left(\Omega; [0,1]^n\right).
\end{aligned}$$

Relaxation of the Product of the Indicator Functions

The product of the dilation d_i and the indicator function u_j in (5.12) is not convex. A convex, tight relaxation of such energy terms was given by Strekalovskiy *et al.* [148]. To this end, we introduce additional dual variables q_{ij} and Lagrange multipliers α_{ij}:

$$\begin{aligned}
\mathcal{Q} &= \Big\{q_{ij} \;\Big|\; q_{ij}\colon \Omega \to \mathbb{R}^4, 1 \le i < j \le n\Big\}, && (5.18)\\
\mathcal{A} &= \Big\{\alpha_{ij} \;\Big|\; \alpha_{ij}\colon \Omega \to [-\infty, 0]^4, 1 \le i < j \le n\Big\}.
\end{aligned}$$

Resulting Optimization Problem

After carrying out these relaxations we finally obtain the following convex energy minimization problem

$$\begin{aligned}
\min_{\substack{u\in\mathcal{U}\\ d\in\mathcal{D}\\ \alpha\in\mathcal{A}}} \max_{\substack{\xi\in\mathcal{K}\\ \beta\in\mathcal{B}\\ q\in\mathcal{Q}}} \sum_{i=1}^{n} \Big\{ &\int_{\Omega} \big(f_i(x) - \operatorname{div}\xi_i(x)\big)\, u_i(x)\, dx + \sum_{z\in\mathcal{S}_i} \int_{\Omega} \beta_{i_z}(x) \big(d_i(x) - u_i(x+z)\big) dx\\
+ \sum_{j=i+1}^{n} &\int_{\Omega} q_{ij}^1(x) \big(1 - d_i(x)\big) + q_{ij}^2(x)\, d_i(x) + q_{ij}^3(x) \big(1 - u_j(x)\big) + q_{ij}^4(x)\, u_j(x)\\
&+ \alpha_{ij}^1(x) \big(q_{ij}^1(x) + q_{ij}^3(x)\big) + \alpha_{ij}^2(x) \big(q_{ij}^1(x) + q_{ij}^4(x)\big) \qquad (5.19)\\
&+ \alpha_{ij}^3(x) \big(q_{ij}^2(x) + q_{ij}^3(x)\big) + \alpha_{ij}^4(x) \big(q_{ij}^2(x) + q_{ij}^4(x) - A(i,j)\big) dx \Big\}.
\end{aligned}$$

The projections onto the respective convex sets of ξ, d, β and α are done by simple clipping while that of the primal variable u is a projection onto the simplex in $\mathbb{R}^n$ [103].

5.4.2 Implementation

In the previous sections, we proposed our method in a continuous framework with the image domain $\Omega \subset \mathbb{R}^2$. For this reason we discretize the problem using a regular Cartesian grid [41] as is commonly done, *e.g.* [121]. In order to find the globally optimal solution to this relaxed convex optimization problem, we employ the primal-dual algorithm published in [122]. Optimization is done by alternating a gradient descent with respect to the functions u, d and α and a gradient ascent for the dual variables ξ, β and q interlaced with an over-relaxation step in the primal variables. The step sizes are chosen optimally according to [120].

Due to the inherent parallel structure of the optimization algorithm [122], each pixel can be updated independently. *E.g.*, the update of the indicator function $u(x)$: $u^n \to u^{n+1}$ can be computed in parallel for each pixel $x \in \Omega$. Hence, the approach can be easily parallelized and implemented on graphics hardware. We used a parallel CUDA implementation on an NVIDIA GTX 680 GPU.

We stopped the iterations when the average update of the indicator function $u(x)$ per pixel was less than 10^{-5}, *i.e.* if

$$\frac{1}{|\Omega|} \left| u^k - u^{k-1} \right| < 10^{-5}. \tag{5.20}$$

By relaxing the indicator variables, *i.e.* allowing the primal variables u_i to take on intermediate values between 0 and 1, we may end up with non-binary solutions. In order to obtain a binary solution to the original optimization problem, we assign each pixel x to the label L with maximum value after optimizing the relaxed problem:

$$L(x) = \arg\max_i \left\{ u_i(x) \right\}, \; x \in \Omega. \tag{5.21}$$

We observed that the computed relaxed solutions u are binary almost everywhere. For the benchmark experiments, the computed solutions $u_i(x) < 0.01$ or $u_i(x) > 0.99$ for 97-98% of all pixels $x \in \Omega$ and $i = 1, \ldots, n$ and for 2-3% $u_i(x) \in [0.01, 0.99]$.

5.5 Experiments and Results

We have shown how to integrate midrange geometric priors into a variational multi-label approach and gave a convex relaxation of the resulting optimization problem. One of the major advantages of the proposed algorithm is that we can utilize sets $\mathcal{S}_i$ of different sizes and shapes which allow us to define specific neighborhoods of different spatial extent and direction for each label. In the following we will show qualitative and quantitative results for a number of articulated part-based objects such as humans, animals or clothes from the CMU-Cornell iCoseg [22], People [124] and Penn-Fudan dataset [167], for rigid part-based objects such as cars or bicycles as

- 🟧 Hair
- 🟥 Head / Face
- 🟩 Shirt / Pullover / Dress
- 🟦 Jacket
- 🟨 Arms / Hands / Skin
- 🟦 Trousers
- 🟪 Feet / Legs
- 🟪 Socks
- 🟫 Shoes
- ⬛ Background
- 🟪 Weapon
- 🟨 Beak
- 🟩 Body
- 🟨 Saddle
- 🟧 Handlebar
- 🟦 Tires
- 🟪 Bicycle Frame / Car
- 🟥 Car Window
- 🟨 Light
- 🟩 Mirror
- 🟦 License Plate

Figure 5.11: **Color legend** used in all experiments except for the benchmarks which have their own color coding.

well as for a variety of scenes in the MSRC benchmark, for the recognition of facades on the eTRIMS image database [81] and for the task of geometric class labeling of indoor images [93].

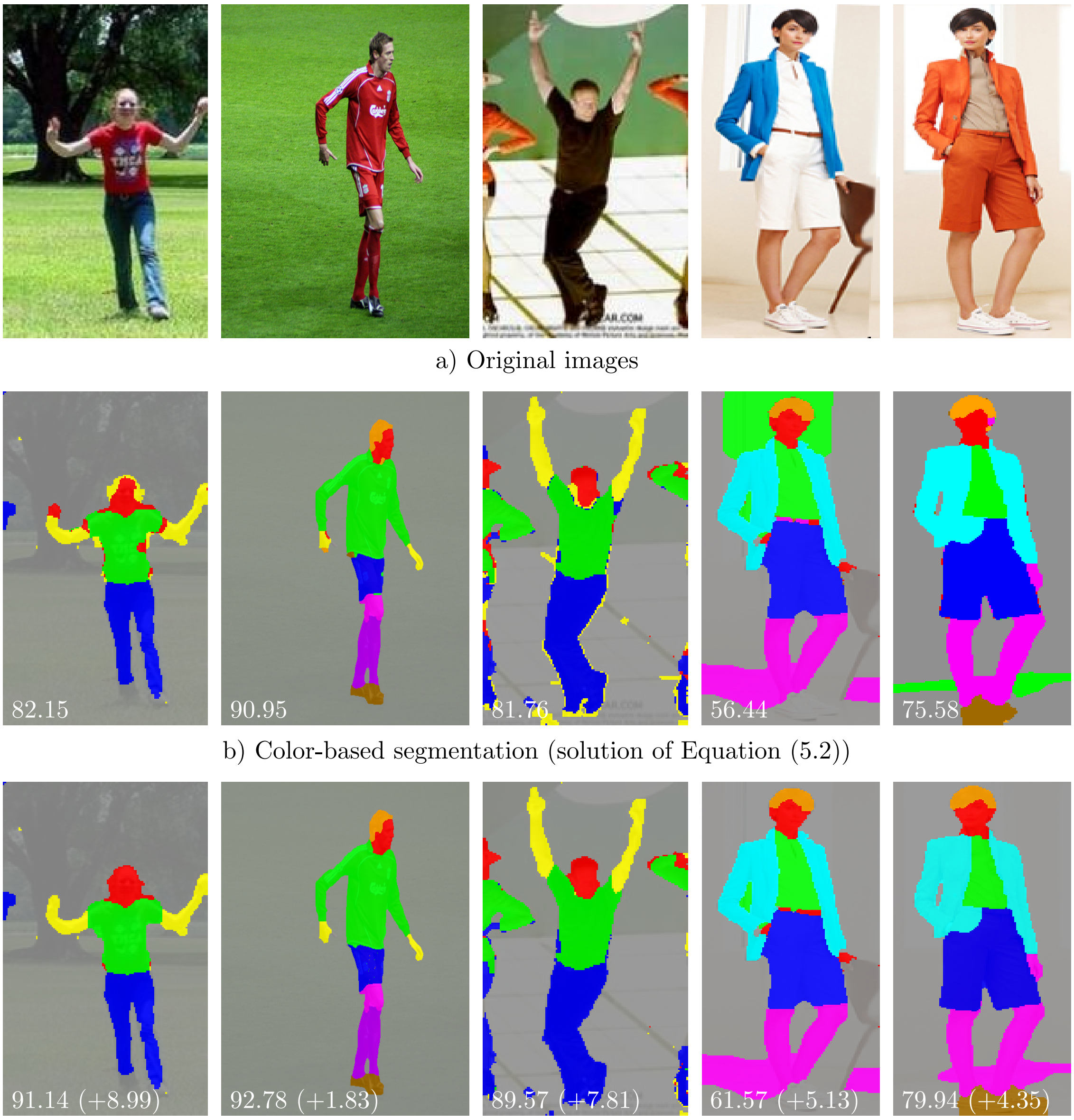
For the iCoseg and People dataset, we defined the labels: 'hair', 'face', 'shirt', 'jacket', 'hands', 'trousers', 'feet', 'socks', 'shoes', 'weapon' and 'background'. The corresponding colors are indicated in Figure 5.11 and consistently used for all experiments except for the benchmarks which have their own standard color legends.[1]

5.5.1 Part-based Articulated Objects: Humans, Animals, Clothes

Articulated objects such as humans, animals and clothes are hard to segment correctly since there are few things that remain constant over a set of images and thus suitable for formulating useful constraints, for example color, shape or absolute location priors are not suitable. Yet, what is typical for many of these objects is that they obey relative geometric constraints, which relate to specific directions and distances and which can be formulated within the proposed framework of the midrange geometric constraints. Especially humans, animals and clothes are good examples for objects, which are difficult to segment, but still follow strict rules imposed on their parts, *e.g.* the head is usually above the feet and trousers can be found below the shirt and hands are usually close to arms.

Figures 5.12, 5.13 and 5.14 show segmentation results for humans, clothes and

[1] The Pascal VOC dataset is not appropriate for the evaluation of the proposed midrange geometric priors since the images of the Pascal VOC segmentation task consist of only very few (often only one) objects and large 'background' areas. 64%/90% of the images contain less or equal one/two objects. The proposed constraints, however, allow to discourage the occurrence of labels in the vicinity of each other, *e.g.* that 'sky' lies above 'ground' or that the 'shoes' of a person appear below the 'head'. We therefore chose datasets with more than three labels for the benchmark evaluations.

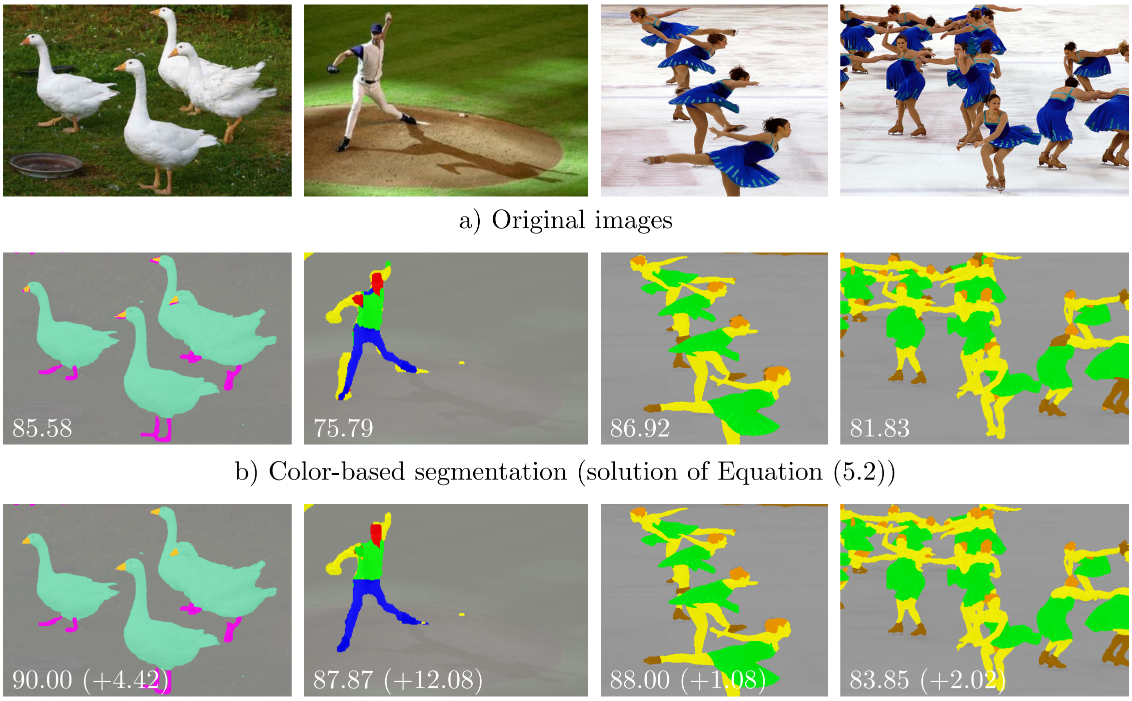


a) Original images

b) Color-based segmentation (solution of Equation (5.2))

c) Segmentation with novel constraints

Figure 5.12: **Part-based articulated objects such as humans or clothes.** Improved segmentation results can be obtained by introducing the proposed novel midrange geometric constraints in order to introduce prior knowledge of relative location, direction and distance of body parts, *e.g.* we penalize 'trousers' above 'body', 'head' and 'arms' below 'legs' and 'shirt' next to 'shoes'. The dice-score (and the improvement over the color-based segmentation) is given in white in the bottom left image corner.



a) Original images

b) Color-based segmentation (solution of Equation (5.2))

c) Segmentation with novel constraints

Figure 5.13: **Part-based articulated objects such as animals or humans.** We obtain improved segmentation results for further articulated objects based on the proposed midrange geometric constraints, *e.g.* we penalize 'feet' close to 'beak' or 'shoes' above 'hair'. The bottom left corner of each image shows the dice-score (and the improvement over the color-based segmentation).

animals. Since no training data is available for the iCoseg and People dataset, we manually defined the structuring elements $\mathcal{S}_i$ and the penalty matrix A. For example, we penalize 'arms' and 'trousers' next to one another using a 31 × 31 sparse symmetric structuring element as well as 'hair' and 'face' next to 'hands' by a 51 × 51 sparse symmetric element $\mathcal{S}_i$ (compare Figure 5.3 d) for $d = 15, 25$). Furthermore, we penalize 'head' below 'body' by a 25 pixel high vertical element centered at the bottom. Each structuring element is selected such that it reflects the common label proximities of the specific dataset. 'Arms', *e.g.*, mostly appear closer to 'trousers' than 'hands' next to 'hair'. Thus, the structuring elements are chosen such that 'hands' and 'hair' are penalized within a larger distance ($d = 25$) than the labels 'arms' and 'trousers' ($d = 15$).

For the experiments on the Penn-Fudan dataset (Figure 5.14) we used the learning approach introduced in Section 5.3.3 and obtained the penalty matrix A and structuring elements $\mathcal{S}_i$ shown in Figure 5.5. For example, we penalize the label 'shoes' appearing closely (within 50 pixels) below 'hair' and the label 'face' appearing closely above 'lower clothes'. Figure 5.14 shows that the proposed constraints

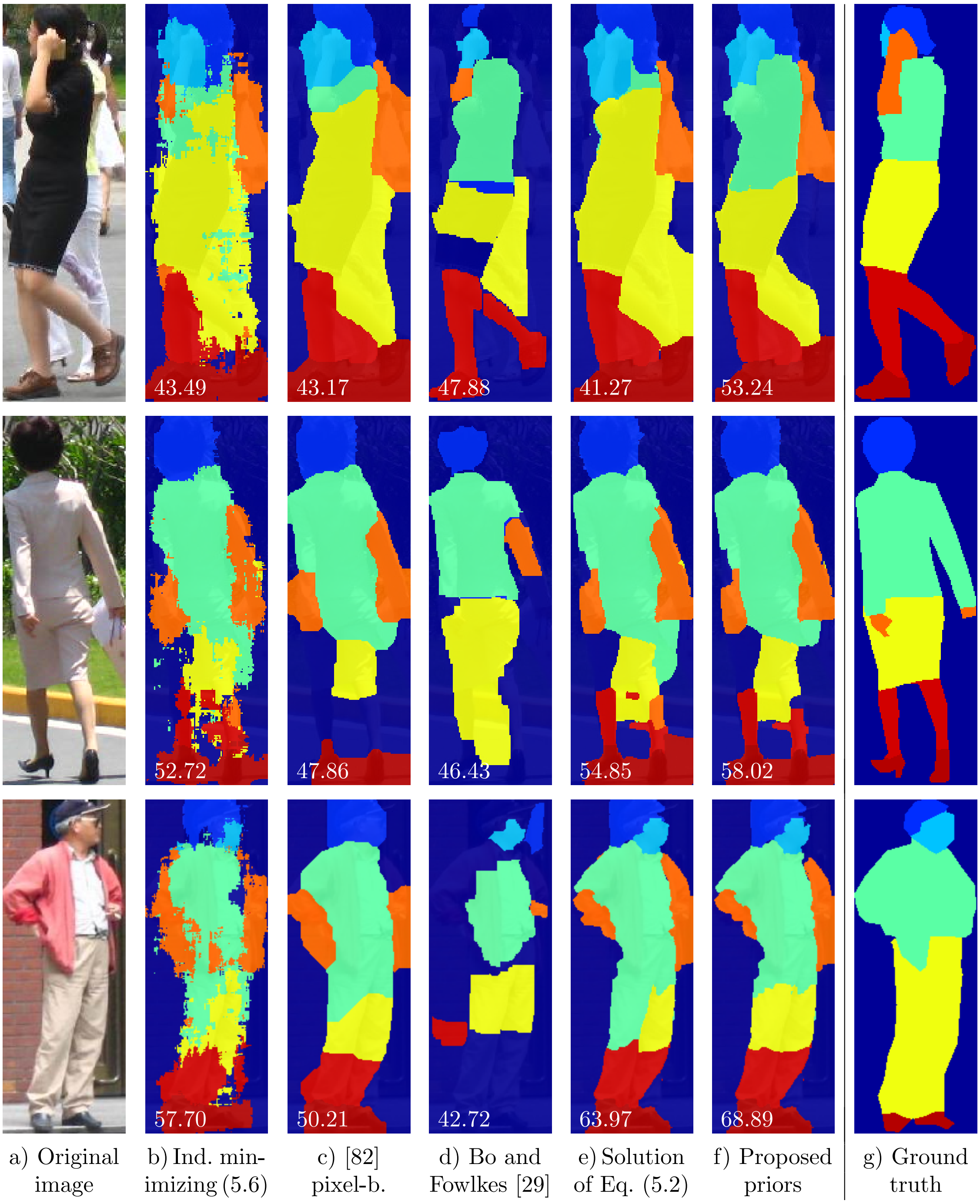


Figure 5.14: **Improved results on the Penn-Fudan dataset** using the learned penalty matrix A and structuring elements $\mathcal{S}_i$ shown in Figure 5.5. The proposed novel midrange geometric constraints allow to obtain improved segmentation results by capturing richer semantic information on spatial object inter-relations of part-based articulated objects such as humans.

improve the semantic labeling of the images compared to c) the pixel-based approach by Ladicky *et al.* [82], d) the approach by Bo and Fowlkes [29] who provided the ground truth annotations and e) the color-based segmentation (solution of Equation (5.2)). In the top row, *e.g.*, the incorrect label transition from 'face' to 'lower clothes' is penalized with the novel constraints and the correct label 'upper clothes' is selected.

To allow for a quantitative analysis, we provide the dice-scores (and the improvement over the color-based segmentation) in the bottom left corner of each image. The dice-score [52] is given as

$$\frac{\mathit{2} \cdot \mathit{True\ Positives} \cdot 100}{\mathit{2} \cdot \mathit{True\ Positives} + \mathit{False\ Negatives} + \mathit{False\ Positives}}. \tag{5.22}$$

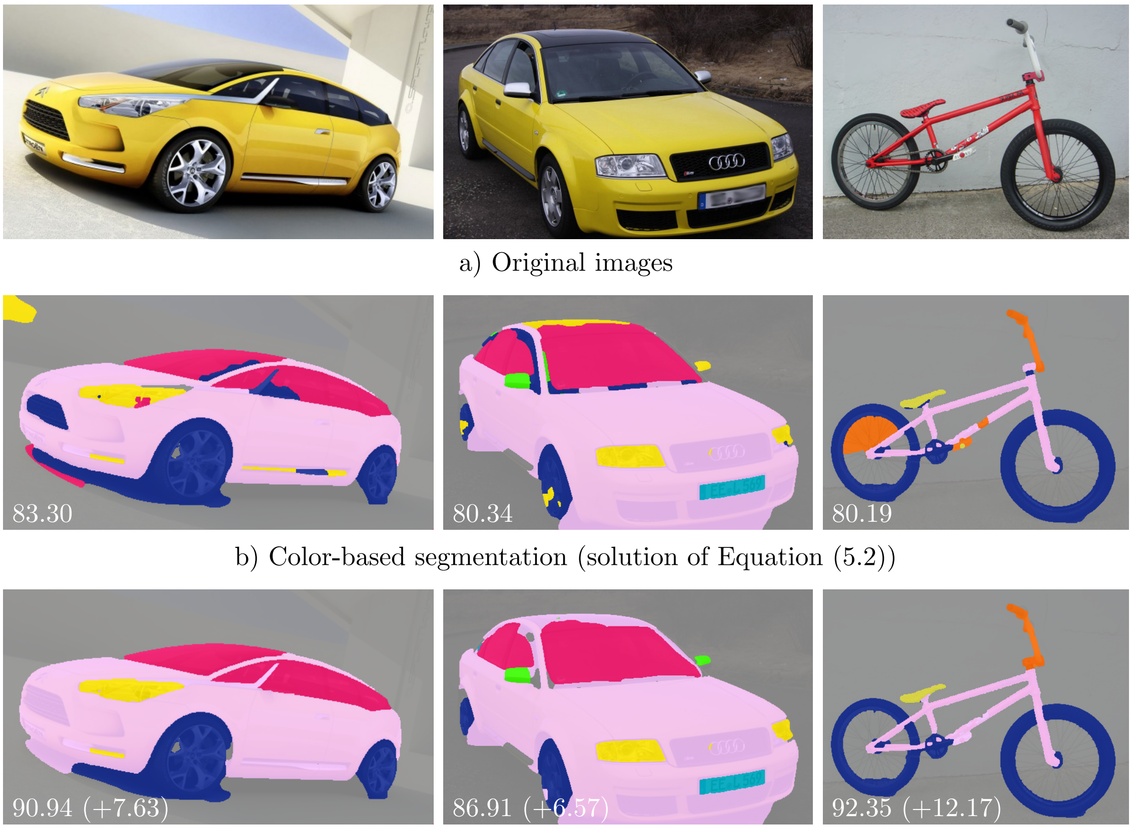
Since no multi-label ground truth segmentations are available for the iCoseg and People datasets, we therefore created accurate ground truth labelings (compare Figure 5.1 d). The qualitative results show improvements up to 12% of the novel constraints over the color-based segmentation. The novel midrange geometric priors capture richer semantic information and thus allow for a correct semantic interpretation. A discussion of the quantitative results on the Penn-Fudan dataset will be given in Section 5.5.6.

5.5.2 Part-based Rigid Objects

An obvious application of the proposed priors are rigid objects consisting of several parts, which is often the case for man-made objects such as cars or bicycles. Using the proposed framework we can improve segmentation results of these objects with all their parts by integrating the proposed priors. Figure 5.15 shows results for a set of part-based rigid objects. For example we penalize 'headlight' and 'window' next to each other and 'tires' next to 'headlight' by using 41 × 41 sparse symmetric elements $\mathcal{S}_i$ (compare Figure 5.3 d) for $d = 20$). The dice-scores (cf. Equation (5.22)) show a significant improvement of more than 6% compared to the color-based segmentation.

5.5.3 Scene Segmentation

The proposed constraints are not only useful for part-based objects but can as well be applied to scene segmentation. The same geometric rules that apply to object parts also apply to whole objects within scenes, for example we know that the sky is above the ground and that sheep do not appear close to wolves. In the following, we show results for a variety of scenes in the MSRC benchmark, for the task of facade recognition on the eTRIMS dataset [81] and for the task of geometric scene labeling of indoor images [93].



a) Original images

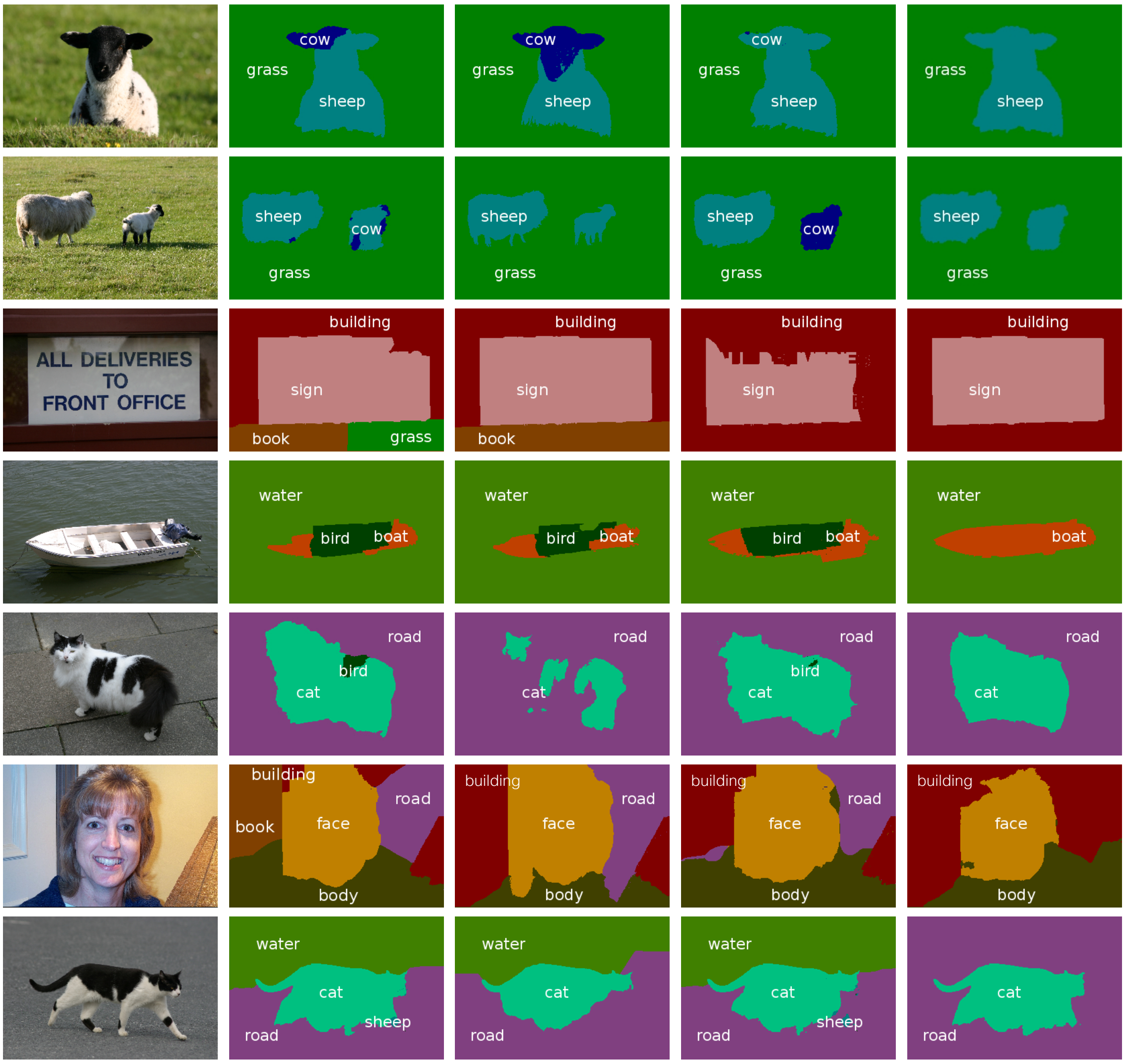
b) Color-based segmentation (solution of Equation (5.2))

c) Segmentation with novel constraints

Figure 5.15: **Part-based rigid objects such as cars or bicycles.** We obtain improved segmentation results by imposing the novel geometric priors. For example we penalize ‘tires’ above ‘window’ or ‘handlebar’ close to ‘tires’.

MSRC Scene Segmentation

In Figure 5.16 we show several results from the MSRC benchmark. We compare our results to previous approaches, which incorporate semantic constraints. The global co-occurrence priors by Ladicky *et al.* [82] penalize the simultaneous occurrence of specific label sets, but they exhibit two drawbacks: a) The quality of the results depends on the quality of the superpixel partition, which is done prior to any segmentation. This can lead to segmentations such as the cat in Figure 5.16 fifth row, where only the black image parts are considered as ‘cat’. b) They altogether disregard spatial information. Since the penalty is independent of the size of the regions and their location in the image, the prior is sometimes not strong enough to prevent incorrect label combinations. As a consequence, if more pixels vote for a certain label then they may easily overrule penalties imposed by the co-occurrence term. This can lead to segmentations such as the sheep with cow head (see Figure 5.16 first row) despite a large co-occurrence cost for ‘sheep’ and ‘cow’. Other examples



a) Original

b) Solution of Eq. (5.2)

c) Global co-oc. prior [82]

d) Local non-metric prior [149]

e) Proposed geometric priors

Figure 5.16: **Improved results on the MSRC benchmark.** Midrange geometric priors capture richer semantic information on spatial object inter-relations such as distances, direction and relative location than previous approaches such as global co-occurrence [82] or local co-occurrence [149].

are the sign above the book (third row) or the cat below the water (seventh row) despite large costs for 'sign' and 'book' or 'water' and 'cat'.

The local nonmetric prior by Strekalovskiy *et al.* [149] can be understood as a purely local co-occurrence prior since it only considers directly adjacent pixels as close. If two sheep are standing further apart as in the second row then this case is not penalized by the prior, which can lead to a sheep and a cow close to each other. Besides, this method can easily produce ghost regions, see Section 5.5.5.

There is no notion of distance, direction or proximity in each of the approaches [82, 149]. In contrast, the proposed label cost penalty is proportional to the size of the labeled regions and also effects object labels at larger spatial distances. Hence, the proposed priors are more flexible and allow for the integration of more specific information, which improves segmentation results as shown in the last column e) of Figure 5.16. The result of the cat (see Figure 5.16 fifth row), *e.g.*, shows that we can avoid problems which appear due to prior superpixel segmentations.

Facade Parsing on the eTRIMS Dataset

We applied our method for the recognition of facades on the 8-class eTRIMS facade dataset [81]. The following eight object classes are considered: 'sky', 'building', 'window', 'door', 'vegetation', 'car', 'road' and 'pavement'.

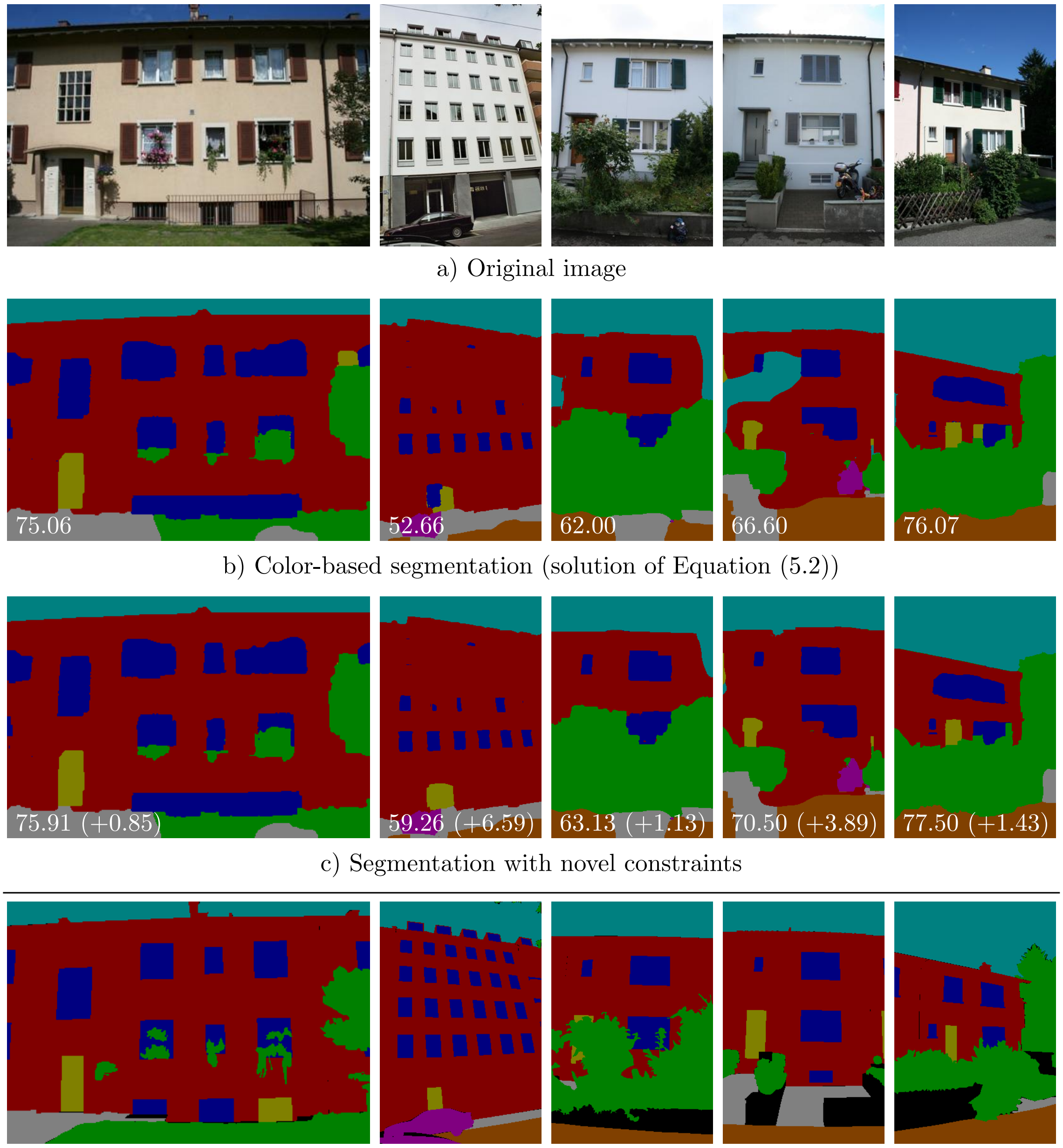
In Figure 5.17 we present five examples of facade segmentations. In columns one and two, the incorrect label transition from 'window' (blue) to 'door' (yellow) is corrected with the novel constraints by penalizing the appearance of 'window' close to 'door'. In columns three and four, the wrong labeled 'sky' pixels (cyan) in the middle of the building disappear by claiming that no other region appears above 'sky'. The combination of both constraints improves the segmentation in the rightmost column, where both the incorrect 'sky' and the incorrect 'door' pixels are removed with the novel constraints.

A first quantitative comparison is provided by the dice-scores. A concrete benchmark analysis will be given in Section 5.5.6.

Geometric Class Labeling of Indoor Images

In tasks like 3D reconstruction or vision-guided robot navigation a rough labeling of the environment is essential. In particular, the geometric classes such as 'floor' or 'right wall' are of importance. We therefore applied our novel constraints on the dataset of indoor images from Liu *et al.* [93] with the five-regions layout: 'left wall' (yellow), 'floor' (green), 'right wall' (pink), 'ceiling' (blue) and 'center' (cyan).

Knowing, for example, that except the ceiling no other region appears above the left wall, the incorrect labels within the region 'left wall' can be removed. The midrange geometric constraints can, *e.g.*, be defined such that they penalize everything above 'ceiling' and everything above 'left'/'right'/'center' except 'ceiling'.



a) Original image

b) Color-based segmentation (solution of Equation (5.2))

c) Segmentation with novel constraints

d) Ground truth

Figure 5.17: **Improved labeling of facades** on the eTRIMS benchmark. By penalizing 'window' (blue) close to 'door' (yellow) and by claiming that no other region appears above 'sky' (cyan) the incorrectly labeled 'door', 'window' and 'sky' pixels disappear.

Results for six different images with the corresponding dice-scores are shown in Figure 5.18. A quantitative benchmark analysis will be given in Section 5.5.6.

5.5.4 Analysis of Failure Cases

In order to evaluate the strengths and weaknesses of our approach we looked into a number of failure cases on the MSRC benchmark and compared our results to the index minimizing the appearance model (5.6) and the results by Ladicky *et al.* [82] and Strekalovskiy *et al.* [149], see Figure 5.19 for some examples. After close investigation of many cases we can formulate one main reason for incorrect labelings:

The appearance term (see Section 5.2.2, Equation (5.6)) used by all three approaches favors incorrect labels over the correct one (Figure 5.19 c). Take for example the 'building' which occurs in the first row in all three results instead of the correct label 'boat'. Since the appearance term clearly favors the white color to belong to a 'building' and 'building' and 'water' is not an uncommon combination in the penalty matrix we obtain incorrect labels. The same happens for the examples in the central and bottom row, where the appearance term yields lots of incorrect labels. Since the appearance term favors 'building' over 'sign' in the bottom row (see column c) and the proposed priors do not favor 'sign' close to 'sky' over 'building', the incorrect segmentation results. This happens in a similar way in the central row, where the appearance term suggests 'car' next to 'road' and 'water'. Since 'car' is more likely to occur above 'road' than 'water' the water is assigned the label 'road'.

Even though none of the methods yields good results for these images, the proposed novel constraints at least yield a reasonable combination of labels in contrast to the other methods. These failure cases suggest that improvements of the method can be gained by using better appearance models.

5.5.5 Preventing Ghost Labels

'Ghost labels' denote thin artificial regions which are easily introduced if label distances are learned from training data, see for example [149]. If the distance function, *i.e.* the penalty matrix A, does not obey the triangle inequality 'ghost labels' can appear. They reduce costs of direct label transitions by taking a 'detour' over a third, unrelated but less expensive label. For example, the labels sheep and grass are common next to each other, and the same holds for cow and grass, but cows usually do not occur directly next to sheep, so the triangle inequality is violated.

Examples are given in Figure 5.20 b) with a close-up in Figure 5.20 c). The segmentation result obtained by [149], *e.g.*, contains very thin 'boat' regions at the edge of the 'grass' label, because the transition between the labels 'water' and 'boat' and 'boat' and 'grass' is in sum less costly than the direct transition between 'water' and

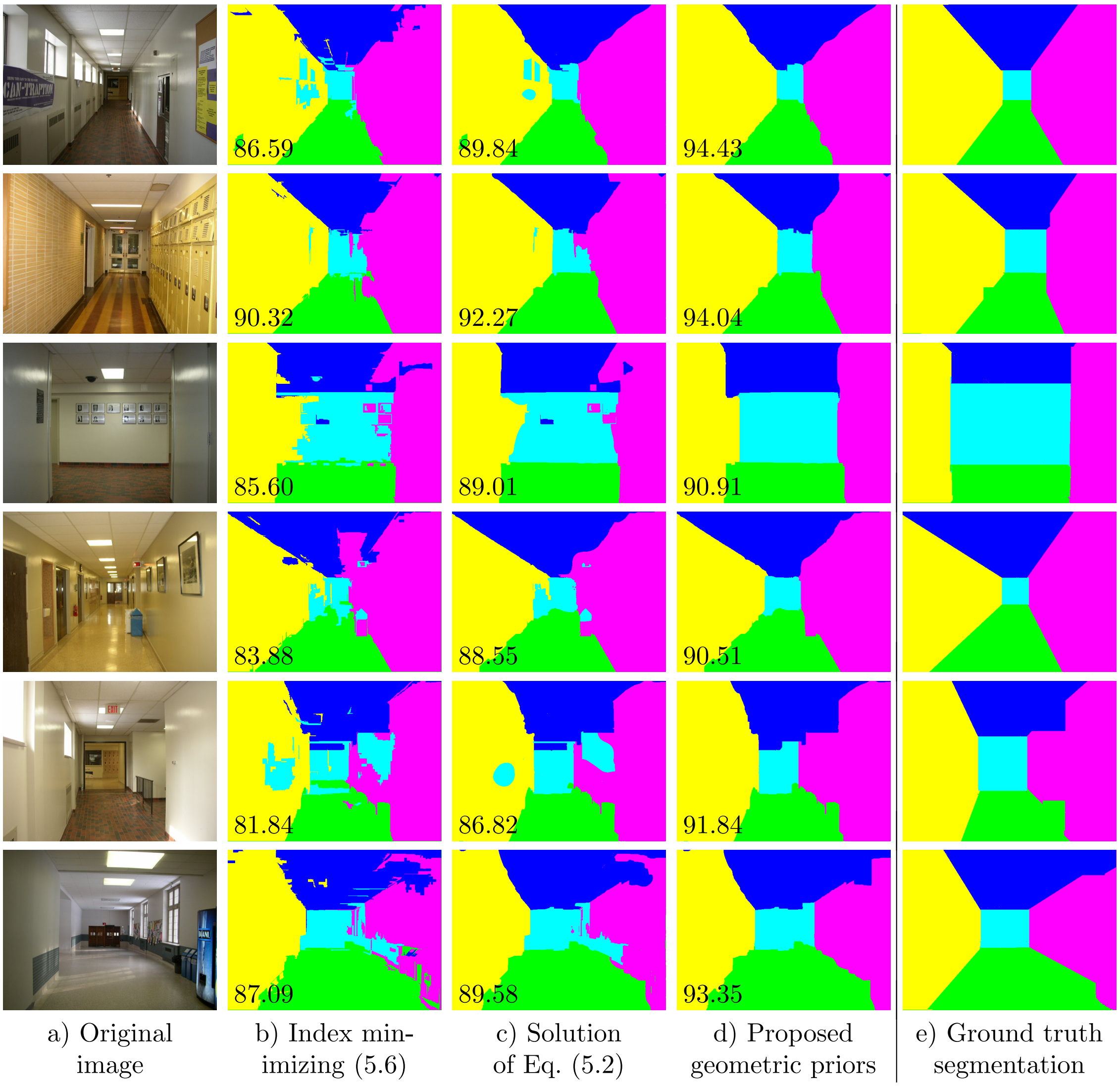
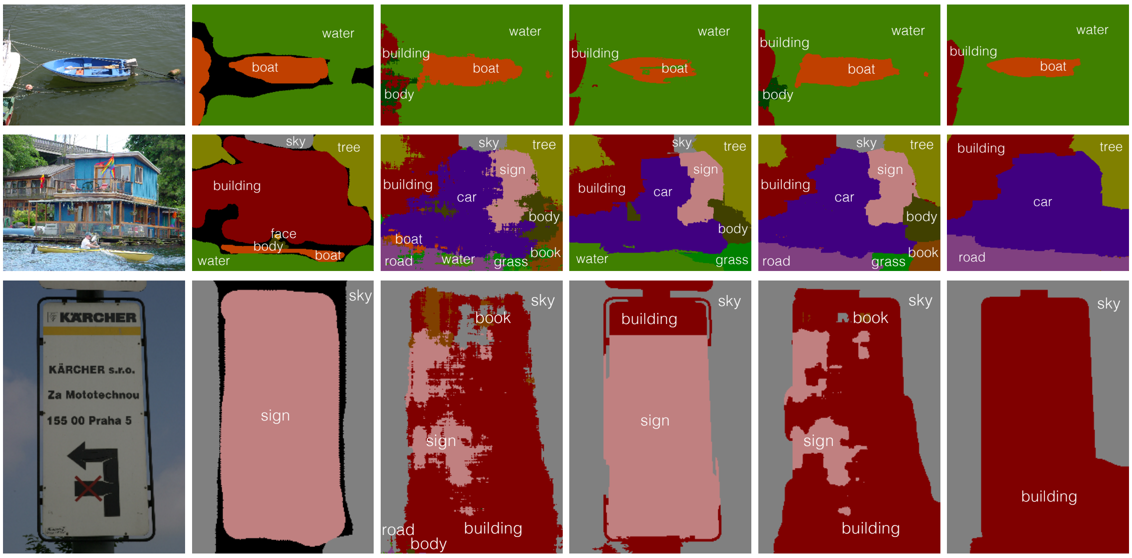
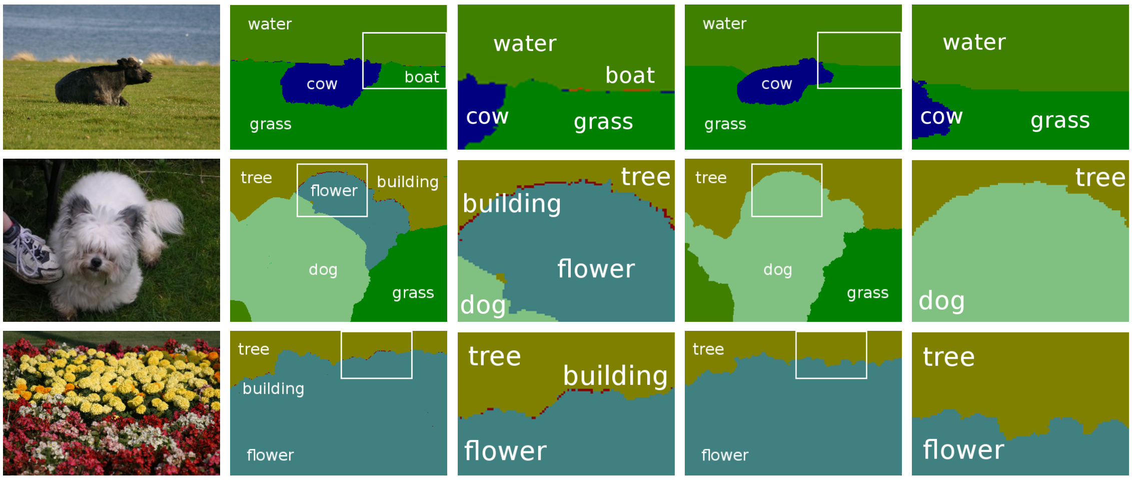


Figure 5.18: **Corrected layout of labels.** The novel priors allow a correct segmentation of the corridors by including directional relations such as that the floor usually is below the ceiling.



a) Original b) Ground truth c) Index minimizing (5.6) d) Global co-occ. prior [82] e) Local non-metric pr. [149] f) Proposed geom. priors

Figure 5.19: **Analysis of failure cases.** For a thorough evaluation we looked into the failure cases of our approach and compared to the index minimizing the appearance term (5.6) and the results of [82] and [149]. We identified one main reason: the appearance term favors incorrect labels.



a) Original b) Local [149] c) Zoom of b) d) Geom. prior e) Zoom of d)

Figure 5.20: **Midrange geometric priors prevent ghost labels.** If the transition of two labels is cheaper via a third label artificial labels will be introduced as shown in b) and as close-up in c). The proposed geometric priors consider regions with more than one-pixel distance still as adjacent and thus avoid ghost labels.

'grass'. The computed label distance matrix denotes the following distances [149]:

$$d(\text{'grass'}, \text{'water'}) = 7.0 > 4.7 = d(\text{'grass'}, \text{'boat'}) + d(\text{'boat'}, \text{'water'}),$$

thus, the more costly label transition from 'grass' to 'water' is avoided by introducing infinitesimal 'boat' regions.

The proposed geometric priors prevent ghost regions since the size of the structuring element is usually larger than two pixels and thus considers more than a single pixel wide margin as close to the object. This leads to overlaps in indicator functions which are larger than a single pixel and thus much more expensive than in the approach by Strekalovskiy *et al.* [149], see for example our results in Figure 5.20 d) with a close-up in Figure 5.20 e). Our learned penalization matrix A, *e.g.*, indicates the following penalties:

$$A(\text{'grass'}, \text{'water'}) = 6.2 < 9.6 = A(\text{'grass'}, \text{'boat'}) + A(\text{'boat'}, \text{'water'}).$$

Thus, the direct transition from 'grass' to 'water' is favored in the optimization process.

5.5.6 Benchmark Evaluation

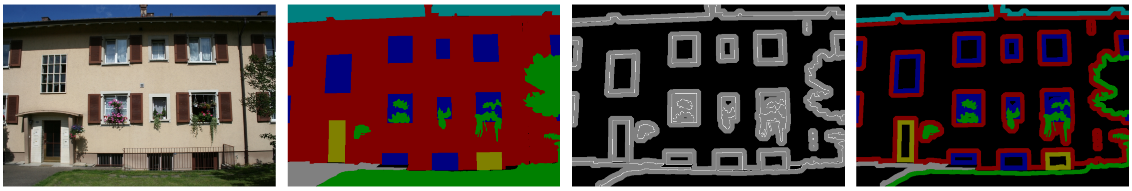
In the following we will show quantitative results on the aforementioned benchmarks and compare our segmentations to state-of-the-art approaches for semantic labeling. For the benchmark analysis, we computed three different evaluation scores. The scores denote the average accuracy on the benchmark given as $\frac{\textit{True Positives} \cdot 100}{\textit{True Positives} + \textit{False Negatives}}$ per pixel and per class and the dice-score averaged over all images. The dice-score [52] additionally takes the false positives into account and is given in Equation (5.22).

We measure the labeling accuracies using the different evaluation scores and using different evaluation regions. The evaluation region can be the whole image domain or restricted to a band surrounding the region boundaries. The restricted evaluation regions are called *trimap* [78]. An exemplary trimap with an evaluation band width of 13 is illustrated in gray in Figure 5.21 c).

Penn-Fudan Benchmark Scores

The Penn-Fudan pedestrian benchmark [167] includes 169 images with an average resolution of 290 × 116 pixels and 12 different labels such as 'hair', 'face', 'left leg' or 'right leg'. We follow Bo and Fowlkes [29] and combine the left and right hand/leg/shoe to one region each, resulting in the 8 different labels: 'hair', 'face', 'upper clothes', 'lower clothes', 'arm', 'leg', 'shoes', 'background'[2]. For the benchmark experiments, we divided the image set randomly into 60% training and 40%

[2] Note that Bo and Fowlkes [29] additionally neglected the region 'shoes'.



a) Original image b) Ground truth segm. c) Trimap of b) d) Trimap segm. of b)

Figure 5.21: **Ground truth and trimap segmentations.** We evaluate the performance using different evaluation domains: b) The whole image domain; c,d) trimap of b) generated by taking a 13 pixel band surrounding the object boundaries.

Table 5.1: **Penn-Fudan benchmark scores.** The proposed constraints outperform the related state-of-the-art segmentation algorithms on the Penn-Fudan benchmark. The best results are given in bold.

	Evaluation on the whole image domain			Evaluation on the trimap (width 13)		
	Accur. per pixel	Accur. per class	Dice-score	Accur. per pixel	Accur. per class	Dice-score
Index minimizing (5.6)	66.97	67.81	55.63	58.09	64.28	53.28
Solution of Equation (5.2)	72.83	70.61	59.98	65.93	68.03	58.58
Ladicky *et al.* [82] pixel-based	71.84	67.36	57.21	64.62	64.70	55.52
Bo and Fowlkes [29]	-	57.29	-	-	-	-
Luo *et al.* [97]	-	54.7	-	-	-	-
Proposed midrange geometric priors	**73.84**	**70.78**	**60.65**	**67.00**	**68.26**	**59.15**

test images and learned the penalty matrix A and structuring elements $\mathcal{S}_i$ (see Figure 5.5) as described above in Section 5.3.3. The parameter lambda is set to $\lambda = 0.8$.

In Tables 5.1 and 5.2 we compare the performance of our method with the approaches by Ladicky *et al.* [82] for the pixel-based prior, Bo and Fowlkes [29] with the shape-based model and the recently proposed work of Luo *et al.* [97] for pedestrian parsing. Furthermore, we present the accuracy of the index minimizing the appearance model (5.6) and the solution of the approach without geometric priors, *i.e.* the solution of Equation (5.2). We evaluate the performance of the approaches on the whole image domain and on the trimap with band width 13. Table 5.1 shows that the proposed midrange geometric constraints outperform the related state-of-the-art segmentation algorithms. In Table 5.2 we additionally compare the confusion matrices on both evaluation domains. Green colored values indicate that the proposed method outperforms the comparative approach for this region. The proposed priors achieve the best performance for the vast majority of regions.

Table 5.2: **Confusion matrix on the Penn-Fudan dataset** obtained for the evaluation on the whole image domain and on the trimap. The elements (i, j) represent the percentage of pixels labeled i by the method and j in the ground truth. We compare the difference between our method and the comparison ones along the diagonal (shown in bold). The values are given in green when the proposed method outperforms the comparative approach, in red otherwise.

	Evaluation on the whole image domain								Evaluation on the trimap (width 13)							
	Backgr.	Hair	Face	Up. Cl.	Low. Cl.	Arms	Legs	Shoes	Backgr.	Hair	Face	Up. Cl.	Low. Cl.	Arms	Legs	Shoes
Background	**72**	3	1	6	9	3	1	5	**59**	5	2	8	13	5	1	8
Hair	4	**81**	6	8	0	1	0	0	4	**81**	6	8	0	1	0	0
Face	4	19	**71**	34	0	2	0	0	4	19	**71**	4	0	2	0	0
Upper Clothes	9	1	1	**78**	5	6	0	0	10	1	2	**73**	6	7	0	0
Lower Clothes	7	0	0	7	**78**	2	1	5	8	0	0	7	**76**	2	1	6
Arms	16	0	4	23	5	**53**	0	0	16	0	4	23	5	**53**	0	0
Legs	19	0	0	0	22	3	**51**	6	19	0	0	0	22	3	**51**	6
Shoes	9	0	0	0	5	0	3	**83**	9	0	0	0	5	0	3	**83**
Index minimizing (5.6)	8	3	1	9	5	-4	1	0	13	4	1	12	6	-4	1	0
Solution of Eq. (5.2)	2	0	1	2	0	-2	0	-1	2	0	1	2	0	-2	0	-1
[82] pixel-based	3	1	5	1	0	3	12	2	4	2	5	0	0	3	12	2
Bo and Fowlkes [29]	-9	36	10	3	7	27	9	-	-	-	-	-	-	-	-	-
Luo *et al.* [97]	-13	36	17	0	3	28	1	-	-	-	-	-	-	-	-	-

MSRC Segmentation Benchmark Results

In the following we will show quantitative results on the MSRC database and compare our segmentations to state-of-the-art approaches for semantic labeling.

The MSRC benchmark comprises 591 images with a resolution of 320×213 pixels which contain 21 different labels such as ‘cow’, ‘book’, ‘building’ or ‘grass’. To conduct experiments on this benchmark, we follow Ladicky *et al.* [82] and divide the image set randomly into 60% training and 40% test images. For the benchmark experiments we chose a symmetric set $\mathcal{S}$ of size 9×9 for all labels (compare Figure 5.3 d) and selected $\lambda = 0.3$. The proximity matrix A is learned on the training set as described above in Section 5.3.4 and illustrated in Figure 5.6.

To evaluate the segmentation accuracy of the proposed method, in Table 5.3 we compare the benchmark scores of our method to state-of-the-art segmentation algorithms with co-occurrence priors: the approaches by Gould *et al.* [65] with relative location priors, Ladicky *et al.* [82] for the pixel-based and the co-occurrence and hierarchical prior, Lucchi *et al.* [96] for the data pairwise global and local models, Vezhnevets *et al.* [161] for the weakly and fully supervised approach and Strekalovskiy *et al.* [149] with the nonmetric distance functions for multi-label problems. Moreover, we present the accuracy of the index minimizing the appearance model (5.6) and the solution of Equation (5.2). The results indicate that we outperform the other co-occurrence based methods in average class and pixel accuracy.

Note that the high score of the approach by Strekalovskiy *et al.* [149] does not reflect the ghost label problem since a) these regions contain only very few pixels, and b) these pixels occur in mostly unlabeled regions of the ground truth near object boundaries, see the second column in Figure 5.19. However, the introduction of entirely unrelated objects, albeit small ones, is often problematic for applications.

The benchmark results in general suggest rather small improvements for the integration of geometric spatial priors. This is somewhat surprising since the images show strong improvements and the prior corresponds to typical human thinking. As already mentioned by Lucchi *et al.* [96] who stated similar findings this is probably due to the rather crude ground truth of the benchmark with large unlabeled regions especially at object boundaries, compare Figure 5.19 b). These regions are not counted in the score, but nevertheless leave a lot of room for misclassification or improvements. Therefore, we think that the benchmark score should not be overstressed here.

To provide a second evaluation measure, we additionally computed the classification error on the precise ground truth provided by [100]. In Figure 5.22 we compare the pixel-wise classification error for different widths of the evaluation region. We consider trimaps with 3 to 21 pixels wide bands surrounding the object boundaries (cf. Figure 5.21). The classification error decreases with increasing width of the trimap. The smallest error is achieved with the proposed midrange geometric priors.

Qualitative comparisons with the two best scoring of the above mentioned methods by Ladicky *et al.* [82] with co-occurrence and hierarchical prior and by Strekalovskiy *et al.* [149] on the MSRC database are given in Figure 5.16. The results show that the proposed method reduces the number of mislabeled objects.

eTRIMS Facade Parsing Benchmark Results

In Section 5.5.3 we already demonstrated some qualitative results for the task of segmenting facades. The 8-class eTRIMS facade dataset [81] consists of 60 images with a resolution of 512×768. Again, we split the dataset into 60/40 for training and testing and set $\lambda = 0.6$.

In Table 5.4 we compare the accuracy per pixel on the whole image domain as well as for different band widths of the trimap. For all evaluation domains the best score is achieved with the proposed priors. The relatively minor improvement in the percentages reflects our observation that significant improvements of the semantic

Table 5.3: **MSRC benchmark scores.** We compare the segmentation accuracy to state-of-the-art segmentation algorithms with co-occurrence priors on the MSRC benchmark. The approach by Ladicky *et al.* [82] in the last row is added for the sake of completeness. They use a more sophisticated appearance model and model co-occurrence by an additional cost function which can be seen as potentials of the highest order $|\Omega|$, instead of order two as in our approach[a]. The best results are given in bold.

	Accur. per pixel	Accur. per class	Dice-score	Building	Grass	Tree	Cow	Sheep	Sky	Plane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Gould *et al.* [65] CRF + rel. loc.	76.5	64.38	-	**72**	95	81	66	71	93	74	70	70	69	72	68	55	23	82	40	77	60	50	50	14
Ladicky *et al.* [82] pixel-based	84	77.72	81	69	**97**	91	86	88	96	88	82	91	82	93	82	63	44	92	66	86	80	56	73	30
Lucchi *et al.* [96], DPG local	75	68.62	-	54	88	83	79	82	95	87	70	85	81	**97**	69	**72**	27	88	46	60	74	27	49	28
Lucchi *et al.* [96], DPG loc.+glob.	80	74.62	-	65	87	87	84	75	93	**94**	78	83	72	93	86	70	50	**93**	**80**	**86**	78	28	58	27
Vezhnevets *et al.* [161], weak sup.	67	66.52	-	12	83	70	81	**93**	84	91	55	**97**	87	92	82	69	**51**	61	59	66	53	44	9	**58**
Vezhnevets *et al.* [161], full sup.	72	71.71	-	21	93	77	86	**93**	96	92	61	79	**89**	89	**89**	68	50	74	54	76	68	47	49	55
Strekalovskiy *et al.* [149]	84.85	77.49	80.79	70	**97**	**92**	**89**	85	96	81	**83**	90	82	92	83	66	45	92	63	**86**	80	51	73	32
Index minimizing (5.6)	82.58	75.91	79.51	67	**97**	91	85	86	95	88	81	90	82	93	81	62	43	65	66	**86**	79	54	73	31
Solution of Equation (5.2)	82.98	76.28	79.66	68	96	91	86	87	95	88	81	90	82	93	81	62	43	65	66	**86**	80	55	72	30
Proposed midrange geom. priors	**84.97**	**78.19**	**81.04**	69	**97**	**92**	87	87	**97**	87	82	91	83	94	84	62	44	**93**	67	**86**	**83**	**57**	**74**	26
Ladicky *et al.* [82] hier. + co-oc.	86.76	76.76	80.78	82	95	88	73	88	100	83	92	88	87	88	96	96	27	85	37	93	49	80	65	20

[a] One could argue that with the introduction of auxiliary variables the model [82] can be reduced to a model of order two. However, the nature of the problem stays a problem of higher order with respect to the original variables. In contrast, our model is of order two without auxiliary variables.

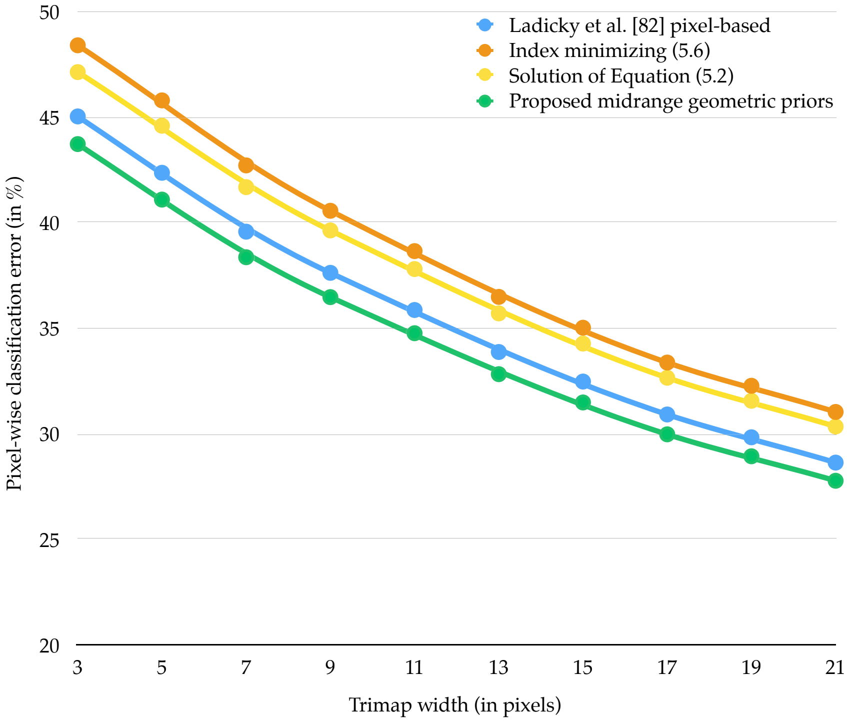


Figure 5.22: **Pixel-wise classification error on the MSRC benchmark.** With increasing width of the evaluation region, the pixel-wise classification error decreases. The best classification is achieved with the proposed midrange geometric priors. For the computation of the trimaps, the more precise ground truth labeling of [100] has been used.

segmentation do not necessarily lead to a substantial improvement of the score. In Figure 5.17 4th column, *e.g.*, a major part of the image – namely the mislabeled 'sky' pixels – is corrected by the proposed constraints. The dice-score for this image, however, only improved by 3.9%.

Score for the Task of Geometric Class Labeling of Indoor Images

The definition of the geometric classes in a scene is another interesting application area. For our experiments we use the indoor dataset from Liu *et al.* [93] which consists of 300 indoor images with a resolution of 640 × 480 pixels. To guarantee comparability we use their appearance model and set $\lambda = 1$.

In Table 5.5 we compare our results to the approaches by Liu *et al.* [93] and Strekalovskiy and Cremers [147] who use the same appearance model. We achieved an overall accuracy of 87.24%, compared Liu *et al.* with 85% and Strekalovskiy and Cremers with 85.3%.

Table 5.4: **The highest scores on the eTRIMS benchmark** are achieved with the proposed priors. The scores are the accuracies per pixel computed on different trimap segmentations and the whole image domain. The best results are given in bold.

	Trimap width 9	Trimap width 13	Trimap width 17	Trimap width 21	Accur. per pixel
Index minimizing (5.6)	63.09	67.90	71.28	73.58	80.56
Solution of Equation (5.2)	69.33	73.35	76.17	78.08	84.36
Ladicky *et al.* [82] pixel-based	68.79	72.84	75.75	77.76	84.22
Proposed midrange geometric priors	**69.37**	**73.46**	**76.34**	**78.31**	**84.82**

Table 5.5: **Improved score for the task of geometric class labeling.** The proposed midrange geometric constraints outperform the approaches by Liu *et al.* [93] and Strekalovskiy and Cremers [147] which use the same appearance model. The best results are given in bold.

	Accur. per pixel	Accur. per class	Dice-score
Index minimizing (5.6)	84.99	79.97	77.67
Solution of Equation (5.2)	86.64	81.59	79.51
Liu *et al.* [93]	85	-	-
Strekalovskiy and Cremers [147]	85.3	-	-
Proposed midrange geometric priors	**87.24**	**81.90**	**80.17**

5.5.7 Runtimes

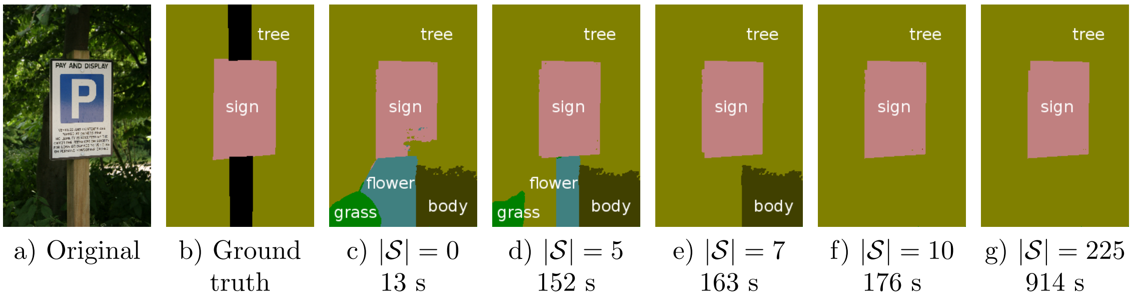
We finally investigate the runtime of the proposed method.

Apart from the size of the $\mathcal{S}_i$, the runtime mainly depends on the number of labels used for the segmentation. For the MSRC benchmark 21 labels have been used. Usually, images consist of less than ten different labels, *e.g.* images of persons can include hair, head, body, arms, hands, trousers, legs, shoes or background.

We obtain average runtimes of 7.7 seconds on the iCoseg [22] and the People [124] dataset (see Table 5.6) compared to 2.3 seconds if we do not use the novel priors. The images have a resolution of around 500×333 pixels and the sets $\mathcal{S}_i$ have a size of around $d = 25$.

Table 5.6: **Average runtimes** for multi-label segmentation of an image of the iCoseg [22] and the People [124] dataset containing 4 to 9 labels.

	Average Runtime
Without Geometric Prior	2.29 s
With Geometric Prior	7.74 s



a) Original b) Ground truth c) $|\mathcal{S}| = 0$ 13 s d) $|\mathcal{S}| = 5$ 152 s e) $|\mathcal{S}| = 7$ 163 s f) $|\mathcal{S}| = 10$ 176 s g) $|\mathcal{S}| = 225$ 914 s

Figure 5.23: **Minimizing runtime.** To minimize runtime in case of large label numbers we use sparse structuring elements (SE). The evolution of the solution for an increasing number of entries in a structuring element $\mathcal{S}$ of size 15×15 shows that very few entries (here 10 entries in a 15×15 SE) are already sufficient to obtain accurate results. The runtimes denote the average runtime on the MSRC benchmark for 21 labels with the respective number of entries $|\mathcal{S}|$.

The MSRC benchmark, in contrast, contains 21 labels, which in theory can appear all at the same time in a single image. This leads to lots of label pairs, most of which are highly unlikely. To reduce the runtime of the approach we used sparse structuring elements $\mathcal{S}_i$ yielding equivalent results to full elements in around 180 seconds on average (note that we do not work on superpixels). We can conclude that already very sparse sets $\mathcal{S}_i$ containing around ten entries yield results very similar to the full set $\mathcal{S}_i$ (compare Figure 5.23).

5.6 Conclusion

In this article we introduced a framework for the integration of midrange geometric priors into semantic segmentation and recognition within a variational multi-label approach. Midrange geometric priors impose constraints on directions and/or distances in which label pairs usually occur. We call them midrange, since the constraints are neither global by taking all pixels into consideration such as co-occurrence priors nor are they purely local by only regarding single pixels or pairwise pixel interactions. Instead, the user is able to define the range and specific shape of the interactions between pixels that are penalized. We have shown how morphological operations such as the continuous formulation of the dilation operation can be employed to formulate these constraints within a continuous optimization approach. We gave a convex relaxation, which guarantees independence of the initialization.

The proposed approach does not require the computation of superpixels and prevents the emergence of one pixel wide ‘ghost labels’. Experiments show that the proposed novel constraints are beneficial for many segmentation scenarios, *e.g.* for part-based articulated objects such as humans, animals or clothes, for part-based rigid objects, especially man-made items, and for semantic scene segmentation.

Acknowledgements We thank three anonymous reviewers for their constructive feedback.

Chapter 6

Optimizing the Relevance-Redundancy Tradeoff for Efficient Semantic Segmentation

Authors

Caner Hazırbaş[1] *c.hazirbas@tum.de*
Julia Diebold[1] *julia.diebold@tum.de*
Daniel Cremers[1] *cremers@tum.de*

[1]Technische Universität München, Munich, Germany

Status

Published

Publication

C. Hazırbaş, J. Diebold, and D. Cremers. Optimizing the Relevance-Redundancy Tradeoff for Efficient Semantic Segmentation. In *Scale Space and Variational Methods in Computer Vision (SSVM). Lecture Notes in Computer Science.* Vol. 9087. Springer International Publishing, 2015, pp. 243–255. isbn: 978-3-319-18460-9. doi: `10.1007/978-3-319-18461-6_20`. Oral Presentation

Individual contribution

Significant contribution in realizing the scientific project

Problem definition	*contributed*
Literature survey	*contributed*
Method development & evaluation	*significantly contributed*
Implementation	*contributed*
Experimental evaluation	*contributed*
Preparation of the manuscript	*significantly contributed*

Abstract Semantic segmentation aims at jointly computing a segmentation and a semantic labeling of the image plane. The main ingredient is an efficient feature selection strategy. In this work we perform a systematic information-theoretic evaluation of existing features in order to address the question which and how many features are appropriate for an efficient semantic segmentation. To this end, we discuss the tradeoff between relevance and redundancy and present an information-theoretic feature evaluation strategy. Subsequently, we perform a systematic experimental validation which shows that the proposed feature selection strategy provides state-of-the-art semantic segmentations on five semantic segmentation datasets at significantly reduced runtimes. Moreover, it provides a systematic overview of which features are the most relevant for various benchmarks.

Keywords Feature analysis · Feature selection · Image segmentation · Semantic scene understanding

6.1 Introduction

6.1.1 Semantic Segmentation and Feature Selection

Semantic segmentation – sometimes also referred to as class-specific segmentation – aims at jointly computing a partitioning of the image plane and a semantic labeling of the various regions in terms of previously learned object classes. Numerous works are focused on the development of sophisticated regularizers for this problem: co-occurrence priors [82, 145] have been suggested to learn and penalize the joint occurrence of semantic labels within the same image. Proximity priors [2] have been proposed to penalize the co-occurrence of labels within a certain spatial neighborhood. Hierarchical priors [144, 172] have been introduced to impose certain label hierarchies – for example that an office is composed of chairs and tables, whereas an outdoor scene is composed of water, grass, cows, *etc.* Proportion priors [110] have been proposed to learn and impose priors on the relative size of object parts. The quantitative performance in terms of segmentation accuracy of respective methods, however, is generally dominated by respective data terms. In this paper we therefore focus on the data term.

A multitude of data terms have been proposed over the last years to take texture, color, spatial location and even depth into account in the construction of appropriate observation likelihoods associated with each pixel. Not surprisingly, depending on the object class and image benchmark, some features are more relevant than others. While in principle taking more and more features into account should improve the segmentation accuracy, in the interest of computational efficiency, the redundancy among features should be minimized. How can we quantify relevance and redun-

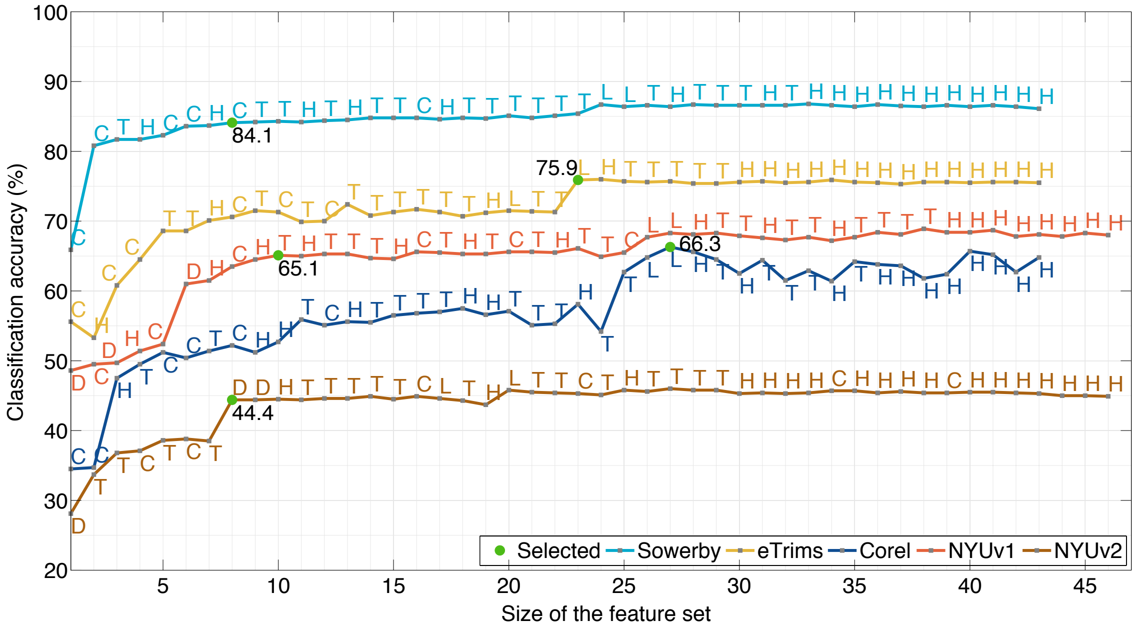


Figure 6.1: **Impact of features on the classification accuracy.** The labels indicate the type of feature added to the feature set: Haar-like (H), color (C), texton (T), location (L) and depth (D). For each benchmark a green dot indicates the feature set which is selected by the proposed approach.

dancy of features? How can we devise a systematic feature selection strategy to identify a small set of optimal features for semantic image segmentation? And how can we automatically determine the number of features to use?

In this work we make use of information-theoretic quantities in order to characterize and optimize the relevance and redundancy tradeoff of respective features for semantic segmentation. An overview of the studied features is given in Section 6.2. For two continuous random variables X and Y, the *mutual information*

$$MI(X;Y) = \int_Y \int_X p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy, \qquad (6.1)$$

is a measure of the mutual dependency of X and Y, where p denotes their probability density function. A feature f_i is relevant for the class labeling c if the mutual information $MI(f_i;c)$ of the feature and the class label is large. Moreover, it is redundant with respect to another feature f_j if the mutual information $MI(f_i;f_j)$ is high. In the following we will show that an appropriate information-theoretic feature selection strategy will lead to semantic segmentation methods which provide state-of-the-art performance at substantially reduced computation time. Figure 6.1 shows the improvement of classification accuracy on different benchmarks with increasing size of the feature set. The features are ordered based on their relevance and redundancy.

This paper is organized as follows: we introduce the studied features in Section 6.2. In Section 6.3 we propose an information-theoretic feature analysis method and in Section 6.4 we show the list of ranked and selected features for five different benchmarks. Finally, we compare our runtime as well as qualitative and quantitative results to state-of-the-art methods (Section 6.5).

6.1.2 Related Work

The literature on object detection can be roughly grouped into two complementary approaches. Conventional object detectors deal with the task of finding bounding boxes around each object [49, 95, 164]. In contrast, dense object detection approaches [70, 139] focus on detecting objects at pixel level. We focus on the choice of the best visual object recognition features for dense object detection.

Shotton *et al.* [139] proposed texture-layout filters based on textons which jointly model patterns of texture and their spatial layout for dense object detection. As they use a large set of features in their computations, their method is not applicable in real-time. On the contrary, our method only chooses the most *significant* features. Thus, we are able to improve the detection performance at a highly reduced runtime.

In 2012, Fröhlich *et al.* [59] proposed an iterative approach for semantic segmentation of a facade dataset. This approach uses millions of features and refines the semantic segmentation by iteratively adding context features derived from coarser levels to a Random Forests classifier. As a result, this approach is fairly slow. In contrast, we determine the optimal set of features and are thus able to receive similar detection accuracies with a significantly smaller set of features at a reduced runtime.

Couprie *et al.* [44] introduced a multiscale convolutional network to segment indoor RGB-D images. They implicitly compute and select features by constructing complex and deep architectures. In contrast, our method is based on a transparent selection criterion.

Recently, Hermans *et al.* [70] discussed 2D semantic segmentation for RGB-D sensor data in order to reconstruct 3D scenes. They use a very basic set of features. However, this basic set of features is determined by experiments and no clear selection criterion is given. In general, none of the above approaches gives justification for their chosen set of features. We specifically address this problem and give detailed explanations on how to choose the best feature set for dense object detection.

6.1.3 Contributions

We present an information-theoretic feature analysis method which resolves the following challenges:

- We answer the questions *which features are the most significant for object recognition* and *how many features are needed for a good tradeoff between accuracy and runtime.*

- The proposed feature analysis method is easy to use and immediately applicable to different datasets. It runs fast in real-time even on large datasets with high-resolution images. All parameters are determined automatically from the information-theoretic formulation.[1]

- We evaluate our method on five different datasets and compare our classification and segmentation results with the state-of-the-art methods by Shotton *et al.* [138], Fröhlich *et al.* [59], Couprie *et al.* [44] and Hermans *et al.* [70]. The proposed feature selection strategy provides state-of-the-art semantic classifications and segmentations at significantly reduced runtimes.

6.2 The Feature Set

We consider 17 shape and texture features composed of 6 Haar-like, 2 color, 4 texton, 2 location and 3 depth features. The features are computed in a patch surrounding the image pixels. Thereby, different patch sizes of the features are used. We convert the images from the RGB(-D) to the CIELab color space and compute the features on the channels: L, a, b (and D). Depth maps are normalized and converted to gray scale.

Haar-like features We use six types of Haar-like features [164]: horizontal and vertical edge (HE/VE) and line (HL/VL), center surround (CS) and four square (FS) illustrated in Figure 6.2 a).

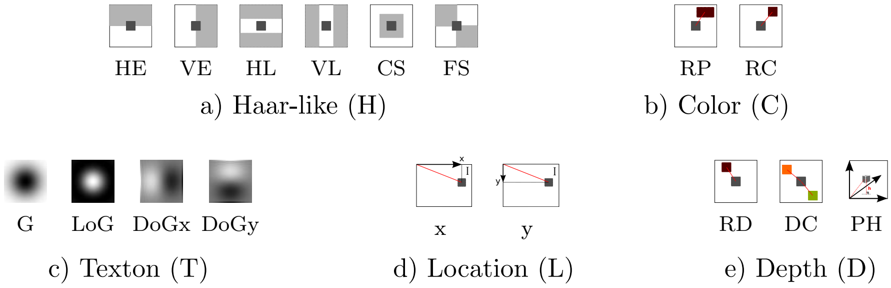
Color features We use the average of the relative patch (RP) and the relative color (RC) feature shown in Figure 6.2 b).

Texton features As texton features we use Gaussian filter (G), Laplacian of Gaussian (LoG) and the first order derivatives of Gaussian filter (DoG) in x and y direction with different bandwidths (see Figure 6.2 c).

Location features We use normalized canonical location features (see Figure 6.2 d) computed for each pixel p in the image I.

Depth features We use the relative depth (RD), the relative depth comparison (DC) and the height of a pixel (PH) [70], illustrated in Figure 6.2 e).

[1] Our code is publicly available at `vision.in.tum.de/data/software`



a) Haar-like (H) b) Color (C)

c) Texton (T) d) Location (L) e) Depth (D)

Figure 6.2: **The feature set.** Illustration of the 17 shape and texture features which are studied in various patch sizes on different image channels. We analyze the significance of the features and explain which and how many of them to use.

6.3 Feature Ranking and Selection for Object Recognition

Among the discussed features, *which* are the most significant for object recognition? And *how many* are needed for a good tradeoff between accuracy and runtime? To this end, we first rank the features according to their significance, then we analyze them and propose an automatic selection criterion.

6.3.1 Feature Ranking

In the first step, a set of training images is used to compute a ranked set of features $\mathcal{F}_R$ of the full feature set $\mathcal{F}$ where the ranking is based on significance. As described in the introduction, features are significant if they are relevant for the classification performance but as little redundant as possible. Ideally, the optimal set of features $\{f_1, .., f_N\}$ is obtained by maximizing the expression

$$\max_{\{f_1,..,f_N\}\in\mathcal{F}} \sum_{f_i\in\mathcal{F}} MI(f_i;c) - \frac{1}{N} \sum_{f_i,f_j\in\mathcal{F}} MI(f_i;f_j), \tag{6.2}$$

where the first term aims at maximizing the relevance of each feature in terms of the mutual information with respect to the target class c and the second term aims at minimizing the redundancy between pairs of features. We call a feature *significant* if it maximizes the relevance for the classification task while minimizing the redundancy with respect to the other features. First of all, the joint optimization over all features is computationally demanding. Secondly, it does not provide us with a ranking of features by significance.

To address these drawbacks, we revert to a greedy strategy for feature selection introduced by Peng *et al.* [119] in the context of biological feature analysis and handwritten digit recognition.

For a fixed target class c, let $\mathcal{F}_{m-1} = \{f_1, \ldots, f_{m-1}\}$ be the *best* feature set with $m-1$ features. To identify the *best additional* feature $f_m \in \mathcal{F} \setminus \mathcal{F}_{m-1}$, we simply optimize its relevance-redundancy tradeoff with respect to the existing features:

$$f_m = \underset{f_i \in \mathcal{F} \setminus \mathcal{F}_{m-1}}{\arg\max} \left[MI(f_i; c) - \frac{1}{m-1} \sum_{f_j \in \mathcal{F}_{m-1}} MI(f_i; f_j) \right]. \tag{6.3}$$

This leads to a set of features $\mathcal{F}_R = \{f_1, \ldots, f_N\}$ which are ranked with respect to their significance for the target class c.

6.3.2 Automatic Feature Selection

Let the first n features in $\mathcal{F}_R$ be denoted by $\mathcal{F}_R\,(n) := \{f_1, \ldots, f_n\}$. In the following step, we determine $n^* \in \{1, \ldots, N\}$ such that $\mathcal{F}_R(n^*)$ consists of only the most significant features. Therefore, we initially apply an *incremental feature analysis* returning the classification accuracy $Acc(n)$ for each feature set $\mathcal{F}_R\,(n)$. Algorithm 1 sketches the steps we carried out to obtain $\big(Acc(1), \ldots, Acc(N)\big)$.

To figure out *how many* features $n^* \in \{1, \ldots, N\}$ to choose, the following conditions have to be met: a) For optimizing the runtime a small n^* is preferred, while b) for the optimization of the accuracy a large n^* is desired. Hence, n^* should be small but still lead to a satisfying accuracy. We therefore propose the following optimization criterion:

$$n^* = \underset{n \in \{1, \ldots, N\}}{\arg\max} \big(Acc\,(n)\,\big)^{\alpha} \,(N + 1 - n)^{\frac{1}{\beta}}\,, \tag{6.4}$$

where $\alpha, \beta \geq 1$ (we set $\alpha = 5,\ \beta = 2$). This function jointly maximizes the accuracy $Acc(n)$ and minimizes the number of features n. Taking $Acc(n)$ to the power of α emphasizes the jumps in the accuracy in which we are interested. Taking the βth root of $(N+1-n)$ prevents too strong influence of the size of $\mathcal{F}_R(n)$. By varying the values of α and β, the method can be adapted to the user's interest focusing on optimal runtime and/or accuracy.

This two-step approach leads to the feature set $\mathcal{F}_R(n^*)$ which consists of only the most significant features for the respective dataset. Compared to other approaches which mostly use arbitrary large feature sets, we are able to obtain competitive classification accuracies at a remarkably reduced runtime.

Related works such as [59, 70] mostly tune the parameters used for training the Random Forests. In contrast, we use default settings for all benchmarks. Our experimental results (Section 6.5) show that the choice of the right features is more important than the best parameter settings for Random Forests. Reduced redundancy in the feature set keeps the accuracy high while it decreases the runtime significantly.

Algorithm 1 Incremental Feature Analysis

```
procedure AnalyzeFeatures($\mathcal{D}, \mathcal{F}_R$)    ▷ $\mathcal{D}$: Dataset, $\mathcal{F}_R$: Ranked Features
    $n = 0$, $Acc = \emptyset$    ▷ $Acc$: Classification Accuracy
    while $n < N$ do
        $n \leftarrow n + 1$
        Extract the features $\mathcal{F}_R(n)$ on the training set.
        Train $K$ Random Trees $\{T_1(\cdot), \ldots, T_K(\cdot)\}$ on the training samples.
        For each class $c \in \{1, \ldots, C\}$ estimate the class probabilities $\widetilde{P}$
        at each pixel $p$ on the validation set:    ▷ $C$: #Classes
            $$\widetilde{P}(c \mid p, \mathcal{F}_R(n)) = \frac{\sum_{k=1}^{K} \left[T_k\big(p, \mathcal{F}_R(n)\big) == c\right]}{K}. \quad (6.5)$$
        Predict the class label $c^*(p)$ for each pixel $p$ with:
            $$c^*(p) = \arg\max_{c \in \{1,\ldots,C\}} \widetilde{P}(c \mid p, \mathcal{F}_R(n)).$$
        Compute $Acc(n)$ with the predicted class labels $c^*$ on the validation set:
            $$Acc(n) = \frac{\text{Number of correctly classified pixels}}{\text{Total number of labeled pixels}}.$$
    end while
    return $Acc$    ▷ List of accuracies $\big(Acc(1), \ldots, Acc(N)\big)$
end procedure
```

6.3.3 Implementation

The algorithm runs fast in real-time even on large datasets with high-resolution images. The whole algorithm runs on a single CPU. We restricted the system to the minimal number of parameters. This makes the application independent from parameter tuning for different benchmarks. Except for the patch size of the features and the grid size Δ_{ss} all other parameters are fixed. Therefore, the proposed method is easy to use and immediately applicable for different datasets.

6.4 Which and How Many Features?

We apply the proposed feature ranking and selection method using the 17 shape and texture features introduced in Section 6.2 on five different benchmarks. In the following we discuss the resulting significance of the different features. We made similar observations on all benchmarks.

The following benchmarks are studied: (i) the 8-class facade dataset eTrims [81], (ii) the 7-class Corel and (iii) Sowerby datasets [69] and (iv) the 12-class NYUv1 [70, 140] as well as (v) the 13-class NYUv2 [44, 141] RGB-D benchmark. For the eTrims dataset we follow Fröhlich *et al.* [59] and split the dataset by a ratio of 60/40 for training and testing. We split the Corel and Sowerby benchmark by a ratio of 60/40, the NYUv1 dataset by a ratio of 50/50 and the NYUv2 by 55/45 for training and testing, similar to [44]. For each benchmark 20% of the training set is used as validation set. On the Corel benchmark we follow Shotton *et al.* [138] and normalize the color and intensity of the images.

For the eTrims, Corel and Sowerby benchmarks we use 50 trees to train the Random Forests, each having at most a depth of 15. For the NYUv1 and NYUv2 benchmark we follow Hermans *et al.* [70] and use 8 trees, each having at most a depth of 10.

To reduce the computational cost during the training process, filter responses are computed on a $\Delta_{ss} \times \Delta_{ss}$ grid on the image [138]. We set $\Delta_{ss} = 3$ for the Corel and Sowerby benchmark and $\Delta_{ss} = 5$ for the eTrims, NYUv1 and NYUv2 benchmark.

6.4.1 Which Features

The ranked set of features $\mathcal{F}_R$, listed in Table 6.1, is computed for each dataset with the method proposed in Section 6.3.1. The following observations can be made on the relevance of the studied features:

Haar-like features (orange) In the literature Haar-like features are commonly evaluated on a gray-scale image or on the luminance channel. Table 6.1, however, shows that for all five benchmarks the top ranked Haar-like features are particularly those ones evaluated on the 'a' and 'b' color channel.

Color features (turquoise) Independently of the benchmark, almost all color features appear among the top ranked features. Hence, color features should definitely be used for training object classifiers.

Texton features (gray) Several texton features are ranked on a top position. Most of the higher ranked texton features (≤ 20) are computed on the 'L' channel. We conclude that texton features are more distinctive on the luminance channel.

Location features (blue) All location features are ranked in the lower half ($\geq$ 17). However, for the eTrims and Corel benchmark, they significantly enhance the classification accuracy (cf. Figure 6.1).

Depth features (purple) are only available for the NYUv1 and NYUv2 benchmark (columns 4,5). All depth features are ranked among the top nine features and strongly boost the accuracy (cf. Figure 6.1).

Table 6.1: **Ranked features** $\mathcal{F}_R$ for the eTrims, Corel, Sowerby, NYUv1 and NYUv2 benchmark. Different colors are set for Haar-like (H), color (C), texton (T), location (L) and depth (D) features. The features are labeled as follows: {feature type}_{feature name}_{patch size}_{color channel}. For an interpretation see Section 6.4.1.

Rank	eTrims	Corel	Sowerby	NYUv1	NYUv2
1	C_RP_25_b	C_RP_11_a	C_RC_7_a	D_PH_25_D	D_PH_25_D
2	H_VL_25_a	C_RC_11_b	C_RP_7_L	C_RP_25_a	T_G_3_L
3	C_RC_25_L	H_CS_11_L	T_DoGy_13x5_L	D_DC_25_D	T_LoG_17_L
4	C_RP_25_a	T_DoGy_13x5_L	H_CS_7_a	H_FS_25_b	C_RP_25_a
5	T_LoG_3_L	C_RP_11_L	C_RC_7_b	C_RC_25_b	T_LoG_5_L
6	T_G_3_L	C_RC_11_a	C_RP_7_a	D_RD_25_D	C_RC_25_L
7	H_CS_25_L	T_DoGx_9x25_L	H_VL_7_b	H_FS_25_L	T_G_5_L
8	C_RC_25_b	C_RP_11_b	C_RC_7_L	C_RP_25_L	D_RD_25_D
9	T_DoGy_25x9_L	H_HE_11_a	T_DoGx_9x25_L	H_CS_25_a	D_DC_25_D
10	C_RP_25_L	H_CS_11_a	T_DoGy_25x9_L	T_LoG_17_L	H_FS_25_L
11	T_DoGy_13x5_L	T_G_9_b	H_HL_7_a	H_VE_25_b	T_DoGy_25x9_L
12	C_RC_25_a	C_RC_11_L	T_G_9_b	T_LoG_3_L	T_G_9_L
13	T_G_9_a	H_CS_11_b	H_CS_7_b	T_DoGx_9x25_L	T_LoG_9_L
14	T_G_5_L	T_DoGy_25x9_L	T_LoG_3_L	T_LoG_5_L	T_DoGx_5x13_L
15	T_LoG_5_L	T_LoG_3_L	T_LoG_5_L	H_CS_25_b	T_LoG_3_L
16	T_LoG_9_L	T_LoG_5_L	C_RP_7_b	C_RC_25_a	C_RP_25_b
17	T_DoGx_9x25_L	T_LoG_9_L	H_CS_7_L	T_LoG_9_L	L_y
18	T_G_9_L	H_FS_11_a	T_LoG_9_L	H_HL_25_L	T_DoGx_9x25_L
19	H_VL_25_b	H_VL_11_a	T_DoGx_5x13_L	T_DoGy_25x9_L	H_HE_25_a
20	L_y	T_DoGx_5x13_L	T_G_3_L	C_RP_25_b	L_x
21	T_DoGx_5x13_L	T_G_5_L	T_G_9_a	T_G_9_a	T_G_5_b
22	T_G_9_b	T_G_9_a	T_G_3_a	H_FS_25_a	T_G_3_b
23	L_x	H_VL_11_b	T_G_3_b	T_G_3_b	C_RP_25_L
24	H_HE_25_L	T_G_3_L	L_x	T_G_5_a	T_DoGy_13x5_L
25	T_G_5_b	T_G_3_a	L_y	C_RC_25_L	H_VL_25_L
26	T_G_5_a	L_x	T_G_9_L	L_x	T_G_5_a
27	T_G_3_b	L_y	H_VL_7_L	L_y	T_G_3_a
28	T_G_3_a	H_VE_11_a	T_G_5_b	H_VE_25_a	T_G_9_b
29	T_LoG_17_L	T_G_9_L	T_G_5_L	T_DoGy_13x5_L	T_G_9_a
30	H_FS_25_a	H_HL_11_a	T_G_5_a	T_DoGx_5x13_L	H_FS_25_a
31	H_VL_25_L	T_G_5_b	H_HE_7_a	H_HE_25_L	H_HL_25_L
32	H_FS_25_b	T_G_3_b	T_LoG_17_L	T_G_5_b	H_CS_25_a
33	H_HL_25_a	T_G_5_a	H_VL_7_a	T_G_9_b	H_FS_25_b
34	H_VE_25_a	H_HL_11_b	H_FS_7_b	H_VL_25_b	C_RC_25_a
35	H_HE_25_a	T_LoG_17_L	H_VE_7_a	T_G_3_a	H_VE_25_a
36	H_CS_25_b	H_FS_11_b	H_FS_7_a	T_G_9_L	H_CS_25_b
37	H_CS_25_a	H_HE_11_b	H_HE_7_b	T_G_5_L	H_HE_25_b
38	H_HE_25_b	H_VE_11_b	H_VE_7_b	T_G_3_L	H_VE_25_b
39	H_VE_25_b	H_FS_11_L	H_FS_7_L	H_VL_25_L	C_RC_25_b
40	H_HL_25_b	H_VL_11_L	H_HL_7_b	H_HE_25_a	H_VL_25_a
41	H_FS_25_L	H_VE_11_L	H_VE_7_L	H_HE_25_b	H_HL_25_a
42	H_VE_25_L	H_HE_11_L	H_HE_7_L	H_VL_25_a	H_VL_25_b
43	H_HL_25_L	H_HL_11_L	H_HL_7_L	H_HL_25_a	H_HL_25_b
44				H_HL_25_b	H_HE_25_L
45				H_VE_25_L	H_VE_25_L
46				H_CS_25_L	H_CS_25_L

We gained a valuable insight into the significance of various features for the task of pixel-wise object recognition. In summary, Haar-like features should particularly be evaluated on the color channels. Color features are important in general. Texton features should be considered especially on the 'L' channel. Location features can be essential and depth features are the most distinctive ones (when available).

6.4.2 How Many Features

In the following we answer the question on the best size of the feature set. For each benchmark, Figure 6.1 illustrates the classification accuracies $Acc(n)$ with increasing n. n indicates the size of the feature set $\mathcal{F}_R(n)$ which leads to $Acc(n)$ (cf. Algorithm 1). The green dots indicate the numbers n^* which are chosen by the proposed optimization criterion in Equation (6.4). The intention is to chose n^* small, but large enough to obtain an optimal tradeoff between accuracy and number of features.

For the eTrims benchmark, *e.g.*, the accuracy has a significant jump from $n = 22$ to $n = 23$. For values of n larger than 23, only very minor improvements can be achieved. Hence, one would prefer n^* to be equal to 23. As marked by the green dot in Figure 6.1, the proposed optimization criterion (6.4) selects $n^* = 23$. The accuracy plot computed for the Corel benchmark has a peak at $n = 27$. Thus, the selected $n^* = 27$ gives the best tradeoff between the accuracy and the size of the feature set. The accuracy plot for the NYUv2 benchmark shows a jump from $n = 7$ to $n = 8$. All values $n \in [9, 46]$ only provide an insignificant increase of $Acc(8)$. Thus, $n^* = 8$ is the perfect value for n and selected by Equation (6.4).

The accuracy plots obtained for the Sowerby and the NYUv1 benchmark show a less significant jump than the plots of the other benchmarks. For the Sowerby benchmark, the proposed method selects $n^* = 8$. Still, this value can be seen as optimal. For smaller values of n, the accuracy is not good enough. For larger values of n, up to $n = 23$, the accuracy improves only very little, whereas the feature set grows much more. The small gain in accuracy would have to be paid for by a much larger runtime. The same holds for the NYUv1 benchmark.

6.5 Experimental Results

Our framework chooses the feature set small but still large enough to obtain a satisfying accuracy. The above observations already show an experimental proof of the proposed feature ranking and selection method. In the following, we compare our runtime, classification and segmentation accuracies as well as qualitative results with state-of-the-art methods.

Table 6.2: **Comparison of runtimes** for object classification in seconds. The training time is given for the whole training set whereas the testing time is averaged over all test images. The proposed method significantly outperforms the other methods in terms of training and testing runtime.

	eTrims		Corel		Sowerby		NYUv1		NYUv2	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Shotton *et al.* [138]	-	-	1800	1.10	1200	2.50	-	-	-	-
Fröhlich *et al.* [59]	-	17.0	-	-	-	-	-	-	-	-
Hermans *et al.* [70]	-	-	-	-	-	-	-	0.38	-	0.38
Couprie *et al.* [44]	-	-	-	-	-	-	-	-	172800	0.70
Proposed	**143**	**6.6**	**20**	**0.27**	**2**	**0.07**	**133**	**0.32**	**183**	**0.26**

6.5.1 Significantly Improved Runtime

We ran our experiments on an Intel® Core™ i7-3770 3.40GHz CPU equipped with 32 GB RAM which is similar to the hardware used by competing approaches. Table 6.2 compares the training and testing runtimes for the classification task. Our framework runs much faster than state-of-the-art methods. In particular for the Sowerby and NYUv2 benchmark, we reduce the training time by a factor of 600 and 900, respectively. Furthermore, our method accelerates the testing runtime on all benchmarks.

6.5.2 Competitive Classification and Segmentation Results

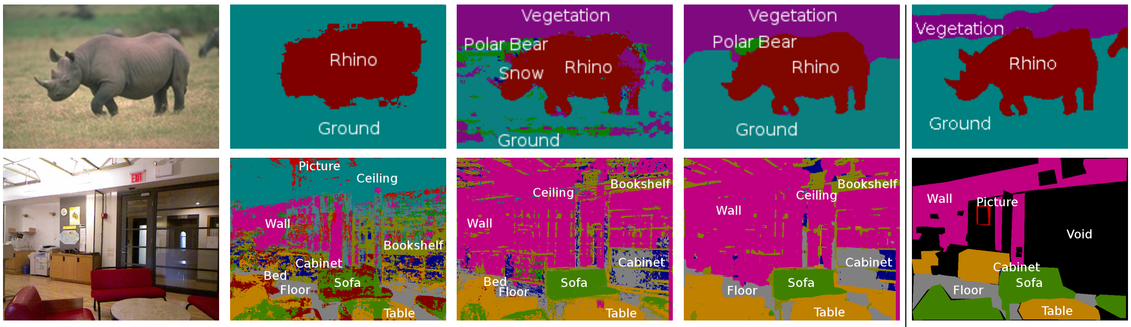
In Table 6.3 we compare the classification and segmentation accuracies to Shotton *et al.* [138], Fröhlich *et al.* [59], Hermans *et al.* [70] and Couprie *et al.* [44]. To obtain a smooth segmentation result we minimize the following energy [40]:

$$E(\Omega_1, \dots, \Omega_C) = \sum_{c=1}^{C} \Big(\mathrm{Per}\,(\Omega_c) + \lambda \int_{\Omega_c} f_c\,(p)\,dp \Big), \tag{6.6}$$

where $\Omega_1, \dots, \Omega_C$ denote the partitions of the image plane, $\mathrm{Per}\,(\Omega_c)$ the perimeter of each set Ω_c which is minimized to favor segments of shorter boundary and $f_c\,(p) = -\log \widetilde{P}(c \mid p, \mathcal{F}_R(n^*))$ the data term, where $\widetilde{P}$ are the class probabilities estimated with the proposed method. λ is a weighting parameter and optimized during the computation.

Table 6.3: **Quantitative results** compared in terms of accuracies. The accuracies are computed as the percentage of correctly labeled pixels on the test set. At significantly reduced runtime our method achieves competitive classification and segmentation accuracies with state-of-the-art methods.

	eTrims		Corel		Sowerby		NYUv1		NYUv2	
	Class.	Segm.	Class.	Segm.	Class.	Segm.	Class.	Segm.	Class.	Segm.
Shotton *et al.* [138]	-	-	68.4	74.6	85.6	88.6	-	-	-	-
Fröhlich *et al.* [59]	-	77.22	-	-	-	-	-	-	-	-
Hermans *et al.* [70]	-	-	-	-	-	-	**65.0**	**71.5**	-	**54.2**
Couprie *et al.* [44]	-	-	-	-	-	-	-	-	-	52.4
Proposed	**77.1**	**77.9**	**74.4**	**78.2**	**87.1**	**88.8**	**65.0**	66.5	**44.0**	45.0



a) Original image b) Classification of [138]/[70] c) Proposed classification d) Proposed segmentation e) Ground truth

Figure 6.3: **Accurate qualitative classification and segmentation results** are achieved with the proposed framework. We compare our classification result to Shotton *et al.* [138] on the Corel benchmark (first row) and to Hermans *et al.* [70] on the NYUv1 benchmark (second row).

Table 6.3 indicates that our classification and segmentation accuracies are competitive with the state-of-the-art approaches. For each benchmark, our method achieves the best accuracies at a remarkably speeded up runtime (cf. Table 6.2).

Most importantly our scores are a) obtained at a significantly improved runtime and b) by using an automatically chosen feature set. We neither tuned the parameters nor the feature set manually to obtain better scores on the specific benchmarks. The proposed method is designed to autonomously compute accurate classifications/segmentations at a significantly reduced runtime for all benchmarks.

Figure 6.3 shows exemplary qualitative classification and segmentation results obtained with the proposed method. In column b), we additionally provide the classification results of the related methods.

6.6 Conclusion

We introduced a framework for automatic feature selection for semantic image segmentation. Starting from a large set of popular features, we sequentially construct a ranked set of features by maximizing the relevance of each feature for the classification task while minimizing its redundancy with respect to the previously selected features. Subsequently, we define an automatic criterion to choose a small number of the most significant features. Integrated in a variational approach to multi-region segmentation, we obtain a fully automatic algorithm which provides state-of-the-art semantic classifications and segmentations on five popular benchmarks at drastically reduced computation time.

Chapter 7

Interactive Multi-label Segmentation of RGB-D Images

Authors

Julia Diebold[1]	*julia.diebold@tum.de*
Nikolaus Demmel[1]	*nikolaus@nikolaus-demmel.de*
Caner Hazırbaş[1]	*c.hazirbas@tum.de*
Michael Möller[1]	*michael.moeller@tum.de*
Daniel Cremers[1]	*cremers@tum.de*

[1]Technische Universität München, Munich, Germany

Status Published

Publication J. Diebold, N. Demmel, C. Hazırbaş, M. Möller, and D. Cremers. Interactive Multi-label Segmentation of RGB-D Images. In *Scale Space and Variational Methods in Computer Vision (SSVM). Lecture Notes in Computer Science.* Vol. 9087. Springer International Publishing, 2015, pp. 294–306. isbn: 978-3-319-18460-9. doi: `10.1007/978-3-319-18461-6_24`

Individual contribution

Significant contribution in realizing the scientific project

Problem definition	*contributed*
Literature survey	*helped*
Method development & evaluation	*helped*
Implementation	*contributed*
Experimental evaluation	*significantly contributed*
Preparation of the manuscript	*significantly contributed*

Abstract We propose a novel interactive multi-label RGB-D image segmentation method by extending spatially varying color distributions [108] to additionally utilize depth information in two different ways. On the one hand, we consider the depth image as an additional data channel. On the other hand, we extend the idea of spatially varying color distributions in a plane to volumetrically varying color distributions in 3D. Furthermore, we improve the data fidelity term by locally adapting the influence of nearby scribbles around each pixel. Our approach is implemented for parallel hardware and evaluated on a novel interactive RGB-D image segmentation benchmark with pixel-accurate ground truth. We show that depth information leads to considerably more precise segmentation results. At the same time significantly less user scribbles are required for obtaining the same segmentation accuracy as without using depth clues.

Keywords Multi-label segmentation · RGB-D images · Interactive segmentation · Spatially varying color distributions · Total variation

7.1 Introduction

A major challenge in computer vision is to compute accurate *image segmentations*, that is, the accurate partitioning of images into meaningful regions. Possible fields of application cover medical imaging, image editing software, object tracking and scene reconstructions. The definition of meaningful regions, however, highly depends on what application the segmentation is needed for. Thus, fully *automatic* image segmentation methods are usually tailored to very specific tasks and try to extract particular objects the methods have learned some prior knowledge about, *e.g.* indoor [44, 70, 140] or facade [71, 154] segmentation.

One way to develop general purpose segmentation tools are *interactive* segmentation methods, where the user indicates the object to be segmented. In this work, we consider user inputs by so called *scribbles*, *i.e.* separate points the user indicated to belong to a certain object. Alternative interactive user input modalities not considered in this work include bounding boxes [92, 128, 163] or contours [13, 27]. Due to their adaptability, interactive segmentation methods have recently attracted a lot of interest. Recent works focus on foreground/background [27, 91, 92, 163, 166] as well as on multi-region segmentation [109, 130, 136], and mostly consider RGB images as input data.

Despite the segmentation constraints given by the user, accurate segmentation remains a challenging task. Extensive studies have led to significant improvements of segmentation quality in recent years [91, 166]. Nevertheless, modern approaches

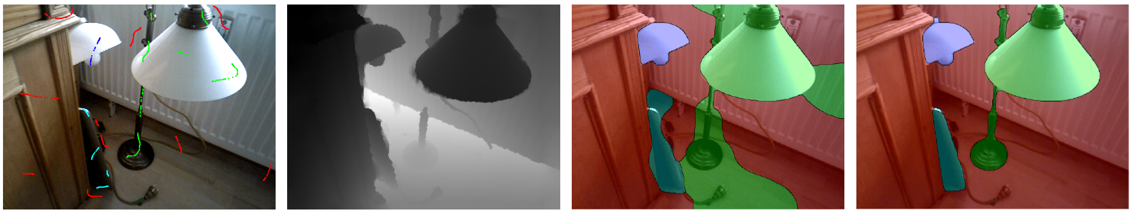

a) Color image with scribbles b) Depth image c) RGB segmentation [108] d) Proposed RGB-D segmentation

Figure 7.1: **Depth information** significantly improves the segmentation result.

often still fail for complex scenes, where objects with similar colors and difficult lightning conditions appear. Moreover, a good segmentation often requires a rather large number of scribbles.

Considering the recent increase and availability of depth-sensing cameras such as the Kinect, we investigate the segmentation of RGB-D images to overcome some of the aforementioned problems. We will mainly focus on the distinction of objects based on color and depth information. While some research has been done on extending interactive segmentation methods to medical imaging data (*e.g.* [31, 94]), very little work has been done on the interactive segmentation of RGB-D images. The only other approach we found which explicitly addresses interactive multi-label RGB-D segmentation is the method by Shao *et al.* [136] on the semantic modeling of indoor scenes. Although this method is related to our approach in the sense that it also formulates the segmentation of RGB-D images as a variational approach, it is tailored towards the application of furniture segmentation. Therefore, the algorithm can use learned a-priori information about the objects to be segmented and the user interaction merely serves as a possible correction step for the first automatic segmentation step.

We investigate the application of interactive RGB-D multi-label segmentation and enhance the recently published work by Nieuwenhuis and Cremers [108] by including depth information. We propose to extend the spatially varying color distributions [108] to RGB-D images in two different ways: a) We consider the depth as an additional color channel. b) We enhance the spatially varying color distributions from varying in a plane to be volumetrically varying. Figure 7.1 d) shows an example of the improvements that can be obtained by taking the depth into account. In the above example, it is almost impossible to distinguish the radiator from the lamp (Figure 7.1 c), because both objects have a similar color and are close in the image plane. The proposed volumetrically varying color distributions (Figure 7.1 d) incorporate the depth information, which yields much more distinct color descriptions and thus better segmentation results.

7.2 Variational Interactive Segmentation of RGB Images

7.2.1 Multi-label Segmentation

Let $I\colon \Omega \to \mathbb{R}^d$ denote the input image, mapping the image domain $\Omega \subset \mathbb{R}^2$ to $\mathbb{R}^d$, with $d = 3$ for an RGB and $d = 4$ for an RGB-D image. Image segmentation denotes the task of partitioning the image plane into a set of n pairwise disjoint regions Ω_i: $\Omega = \bigcup_{i=1}^n \Omega_i$. The regions Ω_i can be computed by minimizing the following energy:

$$E\left(\Omega_1, \ldots, \Omega_n\right) = \frac{1}{2} \sum_{i=1}^{n} \mathrm{Per}_g\left(\Omega_i\right) + \lambda \sum_{i=1}^{n} \int_{\Omega_i} f_i\left(x\right) dx, \tag{7.1}$$

where $\mathrm{Per}_g\left(\Omega_i\right)$ denotes the perimeter of each set Ω_i, which is minimized in order to favor segments of shorter boundaries. These boundaries are measured with either a Euclidean or an edge-dependent metric defined by the non-negative function $g\colon \Omega \to \mathbb{R}^+$. For example, $g(x) = \exp\big(-\gamma|\nabla I(x)|\big)$, favors the coincidence of object border and image edges. f_i denotes the appearance model and λ is a weighting parameter which regulates the influence of the second term.

7.2.2 Convex Relaxation

The usual strategy to address the nonconvex energy minimization problem arising from (7.1) is to use convex relaxation: One represents the disjoint regions Ω_i by indicator functions v_i, with $v_i(x) = 1$ if $x \in \Omega_i$ and $v_i(x) = 0$, else. Since the v_i are indicator functions, we can make use of the fact that the total variation (TV) of an indicator function is nothing but the perimeter of the set described by the functions. Hence, we can reformulate Equation (7.1) as

$$E\left(v_1, \ldots, v_n\right) = \frac{1}{2} \sum_{i=1}^{n} \int_{\Omega} g\left(x\right) \left|D v_i\left(x\right)\right| dx + \lambda \sum_{i=1}^{n} \int_{\Omega} v_i\left(x\right) f_i\left(x\right) dx, \tag{7.2}$$

where Dv_i is the distributional derivative of v_i. Determining the optimal segmentation can be stated as solving the minimization problem

$$\left(\tilde{v}_1, \ldots, \tilde{v}_n\right) = \arg\min_{v_i} E\left(v_1, \ldots, v_n\right) \quad \text{s.t. } v_i(x) \in \{0,1\}, \ \sum_i v_i(x) = 1, \ \forall x. \tag{7.3}$$

Since the nonconvexity of the above problem comes from the integer constraint $v_i(x) \in \{0, 1\}$, a standard convex relaxation is to replace this constraint by $v_i(x) \in [0, 1]$.

The key to obtain a good segmentation method based on (7.3) is to determine f_i that lead to a good data fidelity term guiding the segmentation. In the following, we recall the computation of the f_i motivated by maximum a-posteriori probability (MAP) estimates as suggested in [108].

7.2.3 Likelihood Estimation Based on User Scribbles

Let $I\colon \Omega \to \mathbb{R}^3$ and $u\colon \Omega \to \{1, \ldots, n\}$ be a labeling, such that $\Omega_i = \{x \in \Omega \mid u(x) = i\}$. Motivated by a MAP estimate Nieuwenhuis and Cremers [108] proposed to compute the $f_i(x)$ as the negative log-likelihood of the estimated probability distribution:

$$f_i(x) = -\log \hat{\mathcal{P}}\left(I(x), x \mid u(x) = i\right). \tag{7.4}$$

The expression $\mathcal{P}\left(I(x), x \mid u(x) = i\right)$ denotes the joint probability density of observing a color value $I(x)$ at location x given that x is part of region Ω_i. Based on the ideas of kernel based probability estimates (cf. [142] for an overview), it can be estimated from the user scribbles by

$$\hat{\mathcal{P}}\left(I(x), x \mid u(x) = i\right) = \frac{1}{m_i} \sum_{j=1}^{m_i} k \begin{pmatrix} x - x_{ij} \\ I(x) - I(x_{ij}) \end{pmatrix}, \tag{7.5}$$

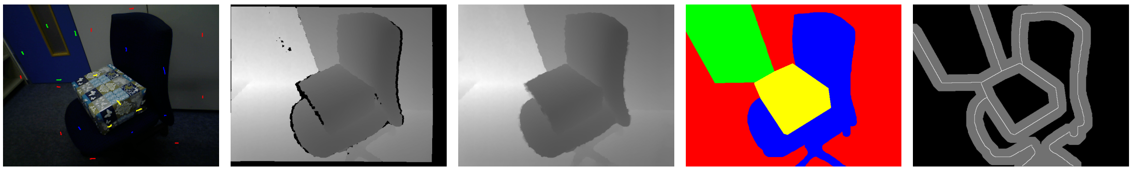
where $\{x_{ij},\ j = 1, \ldots, m_i\}$ is the set of user scribbles for region i, and k a suitable kernel function. The probability estimate in (7.5) only has to be computed for pixels $x \notin \{x_{ij},\ j = 1, \ldots, m_i\}$. For $x \in \{x_{ij}\}$ we keep the label given by the user scribble. We discuss the particular choice of k in more detail below.

7.3 From RGB to RGB-D Images

7.3.1 Pre-Processing the Depth Image

Prior to using the depth image, two pre-processing steps have to be conducted. One has to decide how to handle missing depth information and which range of the depth values to use.

Depth inpainting. Depth cameras such as the Kinect provide metric depth values in addition to color. However, depth information is usually not available for all pixels. We fill in the missing depths in a preprocessing step with an inpainting technique provided in the toolbox of Silberman *et al.* [141]. The implementation is a slight adaptation of the colorization proposed by Levin *et al.* [90]. For an example see Figure 7.2 b,c).



a) Color b) Depth c) Filled depth d) Ground truth e) Trimap

Figure 7.2: **Exemplary RGB-D input, scribbles, ground truth and trimap labeling.** a) Color image with scribbles, b,c) (filled) normalized depth image, d) ground truth segmentation, e) trimap used for measuring the pixel labeling accuracy in a band surrounding the object boundaries [78]. The evaluation region is colored gray and was generated by taking a 25 pixel band surrounding the boundaries of the objects.

Normalization. For Kinect-like cameras the value range of the depth values $z(x)$ in meters is roughly $[0.5, 6]$. To be independent of physical units, for each image we normalize the actual depth range to $[0, 1]$. Similarly, to be independent of the image resolution, we normalize Ω to $[0, 1]^2$.

7.3.2 Depth as an Additional Color Channel

Following Nieuwenhuis and Cremers [108], we use Gaussian kernels with different bandwidths to model the joint probability distribution (7.5). Incorporating the depth image as an additional data channel leads to the following distribution for $\hat{\mathcal{P}}\left(I\left(x\right), D\left(x\right), x \mid u\left(x\right) = i\right)$:

$$\frac{1}{m_i}\sum_{j=1}^{m_i} \underbrace{k_{\rho_i(x)}\left(x - x_{ij}\right)}_{\text{distance kernel}} \underbrace{k_\sigma\left(I\left(x\right) - I\left(x_{ij}\right)\right)}_{\text{color kernel}} \underbrace{k_\tau\left(D\left(x\right) - D\left(x_{ij}\right)\right)}_{\text{depth kernel}}, \tag{7.6}$$

with the bandwidths ρ_i, σ and τ. Due to the comparability of their values, the color channels R, G and B are modeled by the same bandwidth σ. A separate fixed bandwidth τ is used for the depth channel. The bandwidth of the spatial kernel ρ_i on the other hand is chosen proportional to the distance to the closest scribble of label i [108]: $\rho_i\left(x\right) = \alpha \min_{j=1,\dots,m_i} |x - x_{ij}|$. Analogous ideas arise in generalized k-nearest neighbor probability density estimates (cf. [142]), where a similar dependence of the kernel variance on the distance to the nearest samples is considered. Note that although a single multivariate Gaussian could be used for modeling the probability density, this would require an estimation of the covariance matrix, *e.g.* on a training data set.

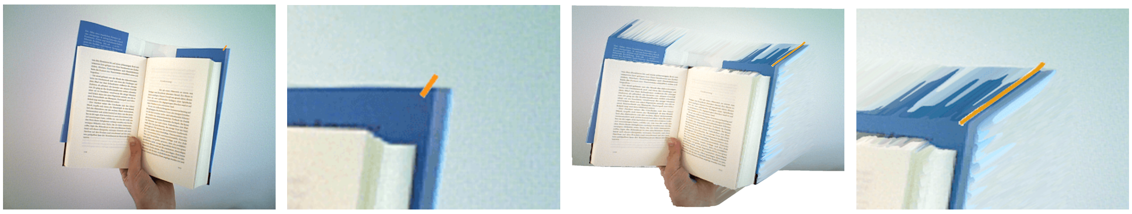
7.3.3 Active Scribbles

To overcome the fact that scribble positions are generally not distributed uniformly throughout the image, we furthermore introduce the idea of *active scribbles*. A general problem of (7.5) and (7.6) is, that the estimated distribution is heavily influenced by the total number m_i of scribbles in class i. This leads to the undesirable behavior that adding many scribbles in one particular region of the image actually reduces the likelihood of far-away-points belonging to the same class. To avoid this, we determine for each pixel x and each class i all scribbles x_{ij}, $j = 1, \ldots, m_i$ that are within a radius of three times the distance to the closest scribble. We call these scribbles active. The distance is computed in 2D or 3D depending on the availability of depth. If less than 80% of the scribbles are active, we compute the probability density (7.6) of the active and inactive scribbles separately and combine the two by $0.8 \cdot \hat{\mathcal{P}}_a \left(I\left(x\right), D\left(x\right),\ x \mid u\left(x\right) = i\right) + 0.2 \cdot \hat{\mathcal{P}}_p \left(I\left(x\right), D\left(x\right),\ x \mid u\left(x\right) = i\right)$, where the subscripts a and p denote the estimates based on the active and passive (inactive) scribbles respectively. Otherwise we use all scribbles to compute (7.6).

7.3.4 Revised Pixel Distance by Depth Values

The main contribution of [108] was to introduce spatially varying color distributions, *i.e.* using a distance kernel in (7.6). The motivation for this kernel was that while an object often looks locally similar, its typical color distribution may change with the position that is considered. With the help of the distance kernel, scribbles that are close to the current position gain more influence than those that are far away. A limitation of this approach for RGB images is that the true 3D geometry cannot be represented: Due to the lack of depth information in RGB data, the method considered in [108] is a projection of a volumetrically varying color distribution onto the image plane.

The depth image allows us to compute color distributions that truly depend on the objects' position in space and thus lead to more distinct color descriptions. For illustration purposes Figure 7.3 a,b) considers a 2D color image. Pixels close in the image are not necessarily close in the 3-dimensional space as we can see in



a) Color image (2D dist. in orange)

b) Zoom of a)

c) Back-projection (3D dist. in orange)

d) Zoom of c)

Figure 7.3: **Recovering the scene geometry with depth information.** Illustration of the distance in the 2-dimensional color image compared to the real distance in the 3-dimensional space. The incorporation of depth information in the computation of the distance kernel allows to capture the real object geometry.

Figure 7.3 c,d). To better reflect the real object geometry, we therefore improve the computation of the distance kernel $k_{\rho_i(x)}\left(x - x_{ij}\right)$ by using the depth information.

Back-Projection. To perform the distance computation in the 3-dimensional space, the 3-dimensional pixel position X has to be computed from the pixel coordinates x and the normalized depth value $D\left(x\right)$. While a physically correct back-projection would be perspective and therefore dependent on the intrinsic parameters of the camera, we found a planar back-projection that simply uses $D\left(x\right)$ as the third coordinate to be the better choice for two reasons: It not only compared favorable in our numerical experiments but also is easier to compute as it does not require the knowledge of camera parameters.

Thus, in Equation (7.6), instead of evaluating the distance kernel $k_{\rho_i(x)}\left(x - x_{ij}\right)$ at $x \in [0,1]^2$ we incorporate the depth as a third dimension and evaluate the distance kernel at $X = \big(x, D\left(x\right)\big)^\top$:

$$k_{\rho_i(X)}\left(X - X_{ij}\right) \quad \text{with} \quad \rho_i\left(X\right) = \alpha \min_{j=1,\ldots,m_i} |X - X_{ij}|. \tag{7.7}$$

7.3.5 The Novel Formulation

Combining the ideas of Sections 7.3.2 and 7.3.4 we propose the following appearance model for RGB-D images

$$f_i\left(x\right) = -\log \hat{\mathcal{P}}\left(I\left(x\right), D\left(x\right), x \mid u\left(x\right) = i\right), \tag{7.8}$$

with

$$\begin{aligned}
&\hat{\mathcal{P}}\left(I\left(x\right), D\left(x\right), x \mid u\left(x\right) = i\right) \\
&= \frac{1}{m_i} \sum_{j=1}^{m_i} \underbrace{k_{\rho_i(X)}\left(X - X_{ij}\right)}_{\text{distance kernel}} \underbrace{k_\sigma\left(I\left(x\right) - I\left(x_{ij}\right)\right)}_{\text{color kernel}} \underbrace{k_\tau\left(D\left(x\right) - D\left(x_{ij}\right)\right)}_{\text{depth kernel}}.
\end{aligned} \tag{7.9}$$

Here $\{x_{ij},\ j = 1,\ldots,m_i\}$ denotes the set of user scribbles for region i, X the three-dimensional position $X = \big(x, D\left(x\right)\big)^\top \in [0,1]^3$ and $\rho_i\left(X\right) = \alpha \min_j |X - X_{ij}|$, σ and τ denote the kernel bandwidths. The effect of both ways of incorporating depth information into the segmentation framework will be studied in detail in the experimental results (Section 7.5).

Finally, let us mention that the two ways the depth information is utilized in the above model is actually equivalent to using a single Gaussian kernel for the depth information. The single kernel would have a bandwidth that contains a spatially varying part as well as a constant part. Since the latter is rather difficult to interpret, we decided to motivate the proposed approach from two different perspectives. Thus, the depth information appears in our proposed model twice.

7.4 Implementation

To find the globally optimal solution to this relaxed convex optimization problem, we employ the primal-dual algorithm published in [55, 120, 122]. It consists of updating a primal and a dual variable in an alternating fashion. The update of each variable decouples for each pixel such that the approach can easily be parallelized and implemented on graphics hardware.

Since we are solving the relaxed problem, there may be pixels x at which $v_i(x)$ take on intermediate values between 0 and 1, *i.e.* we may end up with non-binary solutions. In our numerical experiments, we observed that the computed relaxed solutions $v_i(x) < 0.001$ or $v_i(x) > 0.999$ for 98% of all pixels $x \in \Omega$ and $i = 1, \dots, n$. In order to obtain a binary solution, we assign each pixel x to the label L with maximum value after optimizing the relaxed problem.

7.5 Experimental Results

In this section we demonstrate the effectiveness of all proposed RGB-D image adaptions in several numerical experiments. The numerical study is divided into three parts: First, we discuss the data used for the numerical experiments. Second, we compare RGB to RGB-D segmentation and demonstrate that the segmentation accuracy is improved by the additional depth information. Alternatively, less user scribbles are required by the RGB-D segmentation method to obtain the same accuracy as an RGB method. In a third part we demonstrate that not just one but all of our proposed extensions improve the segmentation results in the sense that the addition of each component individually yields an improvement in segmentation quality.

7.5.1 Experimental Data

As extensively discussed in [130], not every benchmark is suited for testing interactive segmentation. Typical interactive segmentation benchmarks (such as the iCoseg benchmark [22] for foreground/background segmentation or the IcgBench dataset [130] for multi-label segmentation) do not provide RGB-D data, and hence could not be used for our experiments. Popular RGB-D benchmarks such as the NYUv2 dataset [141] are not suitable for interactive segmentation since the scenes are typically composed of very many small objects.

Therefore, we chose the Object Segmentation Database (OSD) [125] as the starting point for numerical experiments. We, however, found that the images contained in the OSD were not challenging enough. They all have the same background and same colors. Furthermore, the objects are relatively small compared to the image size and the given depth. Hence, we decided to use 12 images from the OSD along with 16 images we captured ourselves using an RGB-D sensor. The new images were intentionally taken with challenging color and texture similarities between different objects. For all 28 images, we fixed the scribbles and manually created an accurate ground truth labeling.[1] An example is given in Figure 7.2.

7.5.2 Depth Information is Crucial

We use the aforementioned image data set to compare our algorithm (using $\lambda = 10$, $\gamma = 5$, $\alpha = 1000$, $\sigma = 0.05$, $\tau = 0.2$ for all experiments) to the results obtained by Santner *et al.* [130] and Nieuwenhuis and Cremers [108]. Due to the similarity of our approach with the one in [108], we used the same parameters (without the additional depth information) for the implementation of [108]. For the framework in [130], we took the parameters that were mentioned to be the best general purpose choice.[2] Using exactly the same scribbles (see Figure 7.4 a) for all three interactive segmentation methods, we obtain the results shown in Figure 7.4 c-e).

We have to mention that our comparison is unfair in the sense that the other methods do not make use of the depth information. However, as we could not find other suitable interactive RGB-D segmentation methods, we chose this comparison to illustrate the importance of depth information for image segmentation tasks.

For images with challenging color and lighting conditions, like *e.g.* in Figure 7.4 first row, an RGB based method can hardly find the correct segmentation of the scene. The depth channel, however, provides essential information regarding the spatial relation between the pixels in the image. Thus, the incorporation of the depth image results in significant improvements of the segmentation quality over the RGB based methods. For images in which the depth channel does not provide additional information, such as the image in the bottom row of Figure 7.4, the proposed method yields the same result as [108], as expected.

Another benefit which comes from the additional depth information is that less user scribbles are required compared to an RGB based segmentation method. Figure 7.5 exemplary illustrates this behavior: running our method with the scribbles shown in Figure 7.5 d) we obtain the segmentation result in e). We incrementally add scribbles in order to obtain a similar result with [108], see Figure 7.5 c). Due to the strong color similarity between foreground and background, the RGB based method requires significantly more user scribbles to obtain a similar result.

Finally, let us mention that the runtime of our method is – same as [108, 130] – around one second on 640 × 480 images. The major computational time is needed for the optimization which is independent of our proposed components.

[1] Our framework as well as the RGB-D images, the scribbles and the ground truth labelings are publicly available on our website: `vision.in.tum.de/data/software`

[2] CIELab color space, LBP features with a patch size of 16 and a radius of 3, Random Forests with 200 trees, 750 iterations, $\lambda = 0.2$ and $\alpha = 15$.

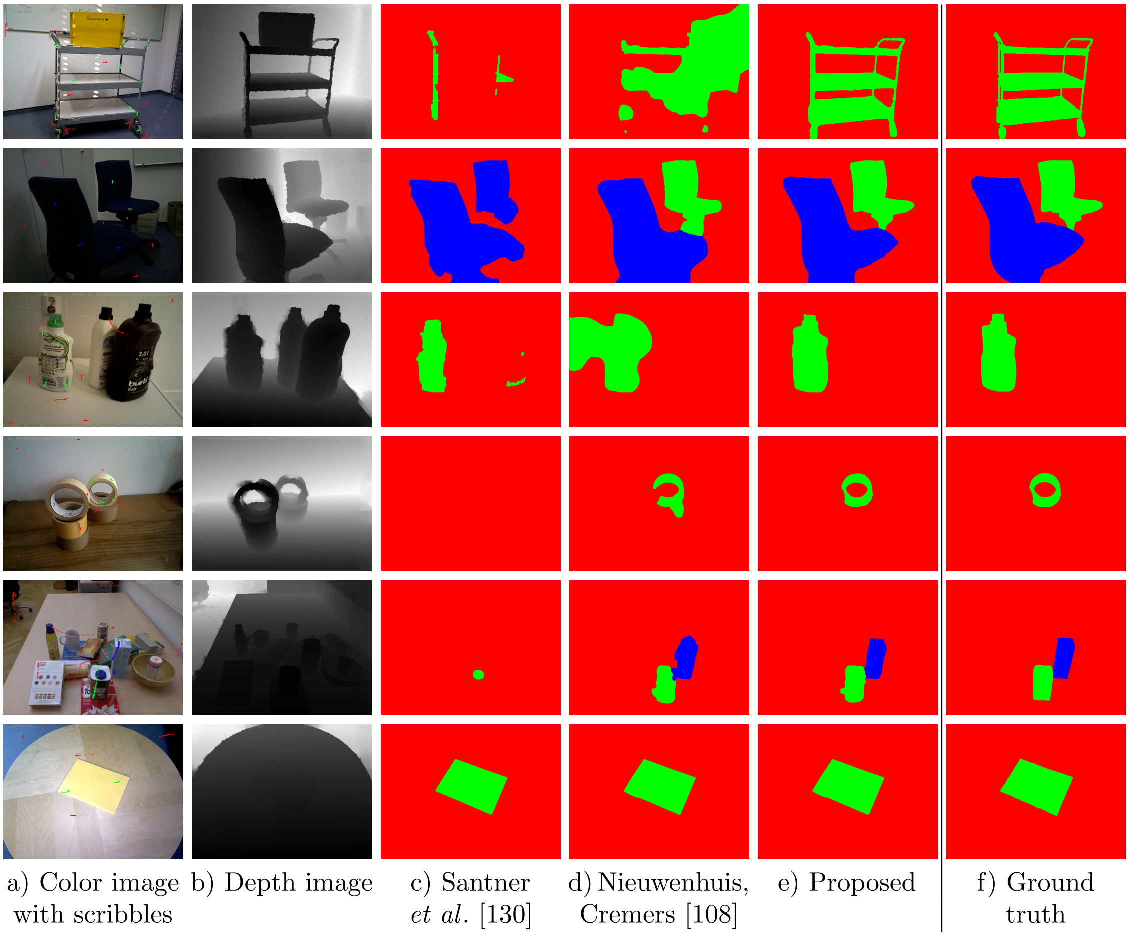

a) Color image with scribbles
b) Depth image
c) Santner *et al.* [130]
d) Nieuwenhuis, Cremers [108]
e) Proposed
f) Ground truth

Figure 7.4: **Depth information improves the segmentation.** The scribbled RGB-D input data is shown in the columns a,b). Columns c-e) compare the proposed RGB-D segmentation to the RGB segmentations of [108, 130].

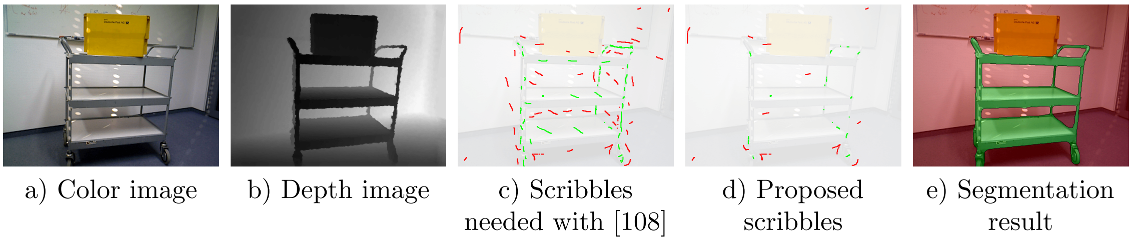


Figure 7.5: **Depth yields less user input.** The depth information provides valuable information which reduces the required user input. To retrieve a similar result as in e), the user needs to place more scribbles with [108] c) than with the proposed volumetrically varying color distributions d).

Table 7.1: **The proposed method outperforms** the previous ones. The dice scores are compared by means of the regular ground truth segmentations as well as the trimap width of 25 (compare Figure 7.2). The usage of active scribbles is abbreviated by 'AS'.

Input	Segmentation method	Reg. GT	Trimap
RGB	Santner *et al.* [130]	72.56	67.69
RGB	Nieuwenhuis and Cremers [108] (Figure 7.6 b)	87.09	86.17
RGB	[108] with proposed AS (2D) (Figure 7.6 c)	87.79	88.40
RGB-D	[108] + AS (3D) + depth for 3D distance (Figure 7.6 d)	91.51	93.63
RGB-D	[108] + AS (3D) + depth as color channel (Figure 7.6 e)	92.93	93.07
RGB-D	Combination of all proposed components (Figure 7.6 f)	**93.70**	**94.84**

7.5.3 Impact of the Proposed Components

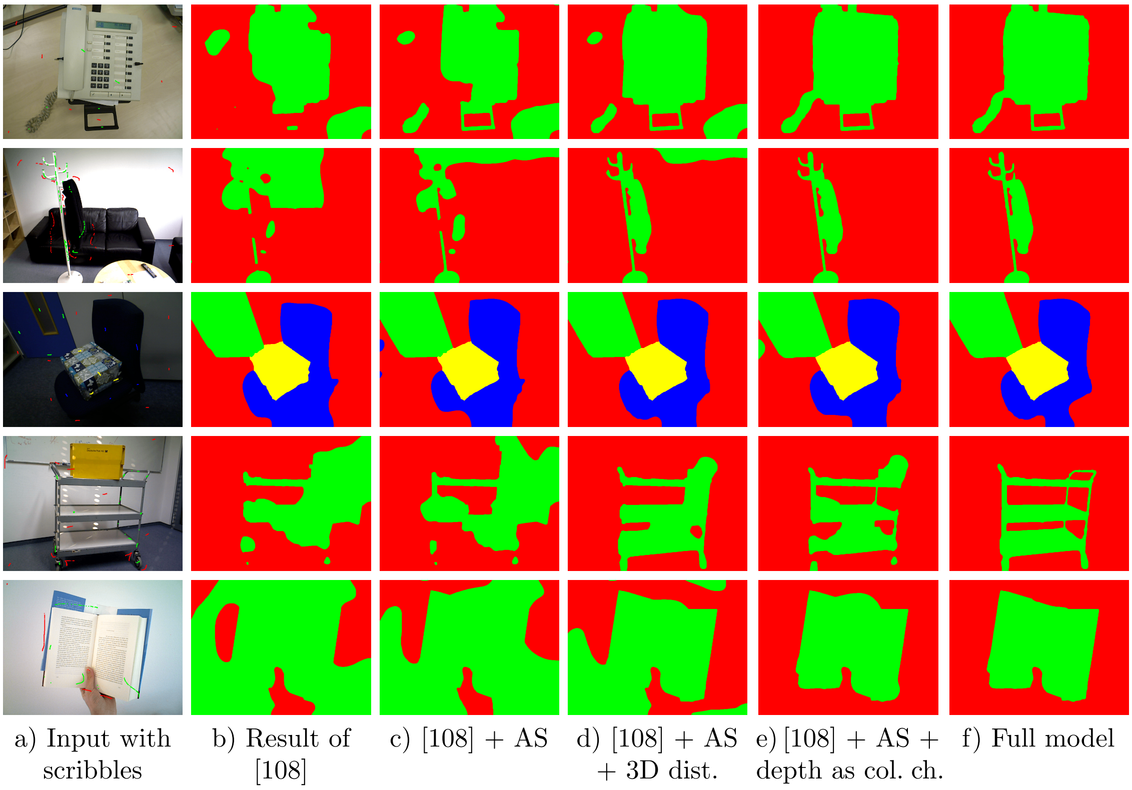
To quantify the results on our benchmark dataset, we compute the dice-scores suggested in [108, 130] on the regular ground truth as well as on a trimap surrounding the object boundaries: Let S denote the labeling obtained for an image, GT the respective ground truth labeling. Then the dice-score is computed as

$$dice\,(S) = \frac{1}{n}\sum_{i=1}^{n}\frac{2|GT_i \cap S_i|}{|GT_i| + |S_i|}, \tag{7.10}$$

where the index i denotes the label i and $|\cdot|$ the area of a segment.

Table 7.1 shows the dice scores averaged over all images obtained by [130], [108], and a step by step addition of the proposed algorithm components. The scores not only give us the possibility of quantitatively evaluating the results obtained by the different methods, but also allow to study the effect of each of the proposed extensions of [108], namely using active scribbles, using depth as an additional data channel and using depth for the 3D distance.

It is interesting to see that the usage of active scribbles – which does not require any depth information – already improves the score on the regular ground truth by 0.7% and on the trimap by 2.2%. Additionally including the depth for either the 3D distance or as an additional color channel again improves the score. The best results are obtained when combining all three components as we can see in the last row of Table 7.1. To visualize the results from Table 7.1, Figure 7.6 shows a qualitative comparison of the different components. As we can see, in each column, from left to right the result improves.



a) Input with scribbles

b) Result of [108]

c) [108] + AS

d) [108] + AS + 3D dist.

e) [108] + AS + depth as col. ch.

f) Full model

Figure 7.6: **Each of the proposed components improves the segmentation.** We compare the segmentations obtained with different components of the proposed method. The usage of active scribbles is abbreviated by 'AS'. f) The combination of all components: active scribbles, depth for 3D distance and depth as an additional color channel leads to the best result.

7.6 Conclusion

We proposed a powerful extension of the spatially varying color distributions [108]. Our contributions include the idea of active scribbles to overcome the problem of non-uniformly distributed user scribbles. Furthermore, we improve the estimation of the data fidelity term by incorporating the depth as an additional color channel as well as using it to construct volumetrically varying color distributions in 3D. We have demonstrated that each of the proposed components contributes separately and improves the segmentation results. Due to the additional depth information, reliable segmentations are obtained with significantly less user input. For future work, one could also use a regularization that takes into account the geometry of the 3D surface as suggested in [127].

Chapter 8

Flow and Color Inpainting for Video Completion

Authors
Michael Strobel[1] *m.strobel@tum.de*
Julia Diebold[1] *julia.diebold@tum.de*
Daniel Cremers[1] *cremers@tum.de*
[1]Technische Universität München, Munich, Germany

Status Published

Publication M. Strobel, J. Diebold, and D. Cremers. Flow and Color Inpainting for Video Completion. In *German Conference on Pattern Recognition (GCPR). Lecture Notes in Computer Science.* Vol. 8753. Springer International Publishing, 2014, pp. 293–304. ISBN: 978-3-319-11751-5. DOI: 10.1007/978-3-319-11752-2_23. Oral Presentation

Individual contribution Significant contribution in realizing the scientific project

Problem definition	*contributed*
Literature survey	*significantly contributed*
Method development & evaluation	*helped*
Implementation	*contributed*
Experimental evaluation	*significantly contributed*
Preparation of the manuscript	*significantly contributed*

Abstract We propose a framework for temporally consistent video completion. To this end we generalize the exemplar-based inpainting method of Criminisi *et al*. [48] to video inpainting. Specifically we address two important issues: Firstly, we propose a color and optical flow inpainting to ensure temporal consistency of inpainting even for complex motion of foreground and background. Secondly, rather than requiring the user to hand-label the inpainting region in every single image, we propose a flow-based propagation of user scribbles from the first to subsequent video frames which drastically reduces the user input. Experimental comparisons to state-of-the-art video completion methods demonstrate the benefits of the proposed approach.

Keywords Video completion · Video inpainting · Disocclusion · Temporal consistency · Segmentation · Optical flow

8.1 Introduction

Videos of natural scenes often include disturbing artifacts like undesired walking people or occluding objects. In the past ten years, the technique of replacing disruptive parts with visually pleasing content grew to an active research area in the field of image processing. The technique is known as video inpainting and has its origin in image inpainting. While image inpainting has been researched very active in the past years the problem of video inpainting has received much less attention. Due to the additional temporal dimension in videos, new technical challenges arise and make calculations much more complex and time consuming. At the same time, video completion has a much larger range of applications, including professional post-productions or restoration of damaged film.

In this work, we focus on two central challenges in video completion, namely temporal consistency and efficient mask-definition.

8.1.1 Related Work

The literature on image inpainting can be roughly grouped into two complementary approaches, namely inpainting via partial differential equations (PDEs) and exemplar-based inpainting. PDE-based inpainting was first proposed by Masnou and Morel [101, 102] and popularized under the name of inpainting by Bertalmio *et al*. [24, 25]. The key idea is to fill the inpainting region by propagating isolines of constant color from the surrounding region. These techniques provide pleasing results for filling small regions, for example to remove undesired text or scratches from images. For larger regions, however, the propagation of similar colors creates undesired smoothing effects. To account for this shortcoming, texture synthesis techniques were promoted, most importantly exemplar-based techniques [14, 53, 54]

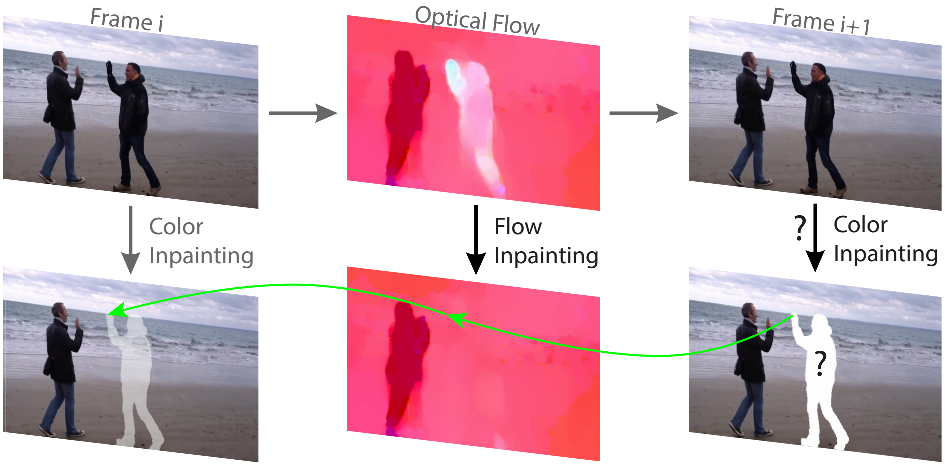


Figure 8.1: **Sketched approach.** We propose an efficient algorithm for semi-automatic video inpainting. In particular, we impose temporal consistency of the inpainting not by a tedious sampling of space-time patches but rather by a strategy of flow- and color inpainting. We inpaint the optical flow and subsequently modify the distance function in an exemplar-based image inpainting such that consistency with corresponding patches in previous frames is imposed.

which can fill substantially larger inpainting regions by copy-pasting colors from the surrounding areas based on patch-based similarity. Criminisi *et al.* [47, 48] presented an approach which combines the two methods to one efficient image inpainting algorithm. The algorithm works at the image patch level and fills unknown regions effectively by extending texture synthesis with an isophote guided ordering. This automatic priority-based ordering significantly improves the quality of the completion algorithm by preserving crucial image structures.

Patwardhan *et al.* [117, 118] and Werlberger [168] extended and adapted Criminisi *et al.*'s [48] method for video inpainting. The approach of Patwardhan *et al.* is using a 5D patch search and takes motion into account. Their approach leads to satisfying results as long as the camera movement matches some special cases. We are not restricted to specific camera motion.

The idea of using graph cuts for video inpainting was recently introduced by Granados *et al.* [67]. They propose a semi-automatic algorithm which optimizes the spatio-temporal shift map. This algorithm presents impressive results however, the approach only has very limited practicability as the runtime takes between 11 and 90 hours for 200 frames.

Newson *et al.* [106, 107] provided an important speed-up by extending the PatchMatch algorithm [20] to the spatio-temporal domain thereby drastically accelerating the search for approximate nearest neighbors. Nevertheless, the runtime for high-resolution videos is about 6 hours for 82 frames.

8.1.2 Contributions

We propose a method for video completion which resolves several important challenges:

- We propose a method to interactively determine the inpainting region over multiple frames. Rather than hand-labeling the inpainting region in every single frame, we perform a flow-based propagation of user scribbles (from the first frame to subsequent frames), followed by an automatic foreground-background segmentation.

- We introduce temporal consistency not by sampling spatio-temporal patches, but rather by a combination of color- and flow-based inpainting. The key idea is to perform an inpainting of the optical flow for the inpainting region and subsequently perform an exemplar-based image inpainting with a constraint on temporal consistency along the inpainted optical flow trajectories – see Figure 8.1. As a consequence, the proposed video completion method can handle arbitrary foreground and background motion in a single approach and with substantially reduced computation time.

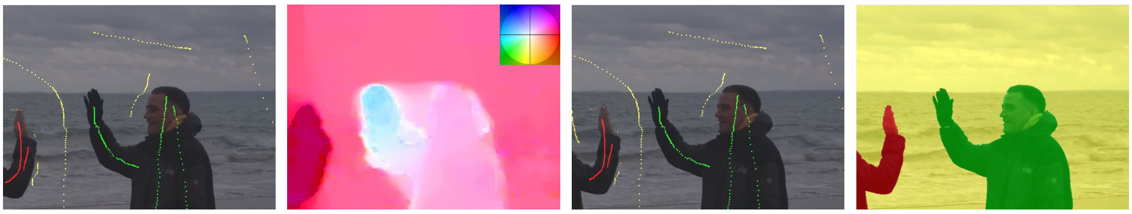
- The inpainting is computed without any pre- or post-processing steps. An efficient GPU-based implementation provides pleasing video completion results with minimal user input at drastically improved runtimes compared to state-of-the-art methods.

8.2 Interactive Mask-Definition

In [24, 67, 169, 170] manual labeling of the inpainting region in all frames of the videos is needed. This makes video editing an extremely tedious and somewhat unpleasant process. We present a simple tool for *interactive mask-definition* with minimal user input. The requirements for such a tool include: (i) an intuitive user interface (ii) a robust mask definition and (iii) a real-time capable algorithm.

The method of Nieuwenhuis and Cremers [108] provides a user-guided image segmentation algorithm that generates accurate results even on images with difficult color and lighting conditions. The user input is given by user scribbles drawn on the input image. The algorithm analyzes the spatial variation of the color distributions given by the scribbles. Thanks to their parallel implementation, computation times of around one second per frame can be obtained.

Based on this user input, we (i) automatically relocate the scribbles throughout the video sequence via optical flow and (ii) frame-wise apply the image segmentation method according to Nieuwenhuis and Cremers [108].



a) Frame I_i b) Flow to I_{i+1} c) Propag. scribbles d) Segmentation

Figure 8.2: **Automatic segmentation by scribble propagation via optical flow.** Scribbles are placed on the first frame and propagated to the next frames by optical flow. Segmentation is computed based on the transported scribbles.

8.2.1 Scribble Relocation via Optical Flow

To transport scribbles over time we use the optical flow method of Brox *et al.* [34] which computes the displacement vector field (u, v) by minimizing an energy functional of the form:

$$E\,(u, v) = E_{Data} + \alpha\ E_{Smooth} \tag{8.1}$$

with some regularization parameter $\alpha > 0$. The data term, E_{Data}, measures the global deviations from the gray value and gradient constancy assumption. The smoothness term, E_{Smooth}, is given by the discontinuity-preserving total variation.

Figure 8.2 b) shows the optical flow between two frames of the image sequence by Newson *et al.* [107]. We use this flow to transport the scribbles from frame to frame (Figure 8.2 a,c). Green scribbles are placed on the region to be inpainted and yellow ones on the search space for the inpainting algorithm. Optionally, red scribbles can be used to mark unrelated image parts in order to shrink the search space. Depending on the user scribbles, a two- or three-region segmentation according to Nieuwenhuis and Cremers [108] is computed.

8.2.2 Segmentation According to Nieuwenhuis and Cremers

Let $I\colon \mathcal{I} \to \mathbb{R}^d$ denote the input frame defined on the domain $\mathcal{I} \subset \mathbb{R}^2$. The task of segmenting the image plane into a set of n pairwise disjoint regions $\mathcal{I}_i$: $\mathcal{I} = \dot{\bigcup}_{i=1}^{n} \mathcal{I}_i$, $\mathcal{I}_i \cap \mathcal{I}_j = \emptyset \quad \forall\ i \neq j$ can be solved by computing a labeling $u\colon \mathcal{I} \to \{1, \dots, n\}$, indicating which of the n regions each pixel belongs to: $\mathcal{I}_i = \Big\{x \;\Big|\; u\,(x) = i\Big\}$. The segmentation time for a video sequence can be speed-up by initializing the indicator function u with the resulting segmentation of the previous frame.

We compute a segmentation of each video frame by minimizing the following energy [108]:

$$E(\mathcal{I}_1, \dots, \mathcal{I}_n) = \frac{\lambda}{2} \sum_{i=1}^{n} \mathrm{Per}_g\,(\mathcal{I}_i) + \lambda \sum_{i=1}^{n} \int_{\mathcal{I}_i} f_i\,(x)\,dx,$$

where $f_i(x) = -\log \hat{\mathcal{P}}\left(I(x), x \,\middle|\, u(x) = i\right)$. $\mathrm{Per}_g(\mathcal{I}_i)$ denotes the perimeter of each set $\mathcal{I}_i$, λ is a weighting parameter. The expression $\hat{\mathcal{P}}\left(I(x), x \,\middle|\, u(x) = i\right)$ denotes the joint probability for observing a color value I at location x given that x is part of region $\mathcal{I}_i$ and can be estimated from the user scribbles. For further details of the segmentation algorithm we refer to [108].

To summarize, our inpainting method brings along a tool which allows the user to quickly define the respective regions on the first video frame, and all the remaining calculations are working automatically. In contrast, state-of-the-art methods require the user to manually draw an exact mask on each single video frame [24, 67, 169, 170] or work with an inflexible bounding box [137].

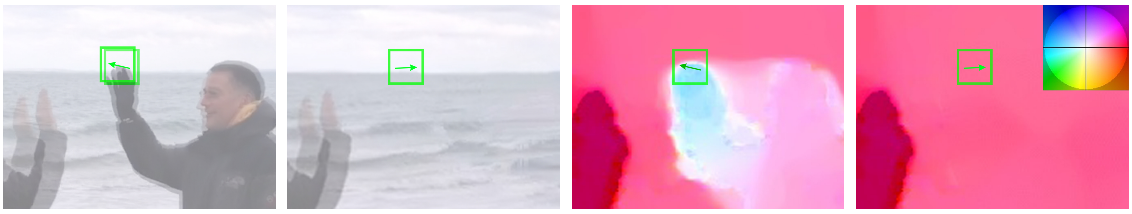
8.3 Flow and Color Inpainting for Video Completion

The major challenge in video inpainting is the temporal dimension: the inpainted regions have to be consistent with the color and structure around the hole, and additionally temporal continuity has to be preserved. When applying image inpainting methods frame by frame, the inpainted videos show artifacts, like ghost shadows or flickering [137]. Several investigations have been done in the past years towards a temporally coherent video completion. State-of-the-art methods, however, have some drawbacks: several pre- and post-processing steps are required [106, 137], only specific camera motions can be handled [67, 106, 118, 169] and the calculations are extremely time consuming [66, 67, 106, 169].

We propose a novel approach inspired by the exemplar-based image inpainting by Criminisi *et al.* [48] overcoming these problems. We apply inpainting to the optical flow and define a *refined distance function* ensuring temporal consistency in video inpainting. No additional pre- or post-processing steps are required.

8.3.1 Inpainted Flow for Temporal Coherence

In a temporally consistent video sequence, the inpainted region follows the flow of its surrounding region. Figure 8.3 a) shows a person who should be removed from the video sequence. The desired patches clearly should not follow the hand of the person, but the flow of the sea. To find the best matching patches, Criminisi *et al.* [48] consider the colors around the hole. We additionally claim a similarity to the patch which naturally flows into this position. This flow is obtained by *inpainting the original flow* – see Figure 8.3 d).



a) Overlayed frames b) Inpainted frames c) Optical flow in a) d) Inpainted flow c)

Figure 8.3: **Inpainted flow ensures temporal consistency.** In order to ensure temporal consistency, we propose to inpaint the optical flow and additionally request the found patch to be similar to its origin. The inpainted flow d) should be approximately the flow of the inpainted video sequence.

8.3.2 Flow Inpainting

For the inpainting of the optical flow we extended the Telea-Inpainting [155] to optical flow. Telea-Inpainting is a fast PDE based approach and hence particularly suited to fill missing parts in optical flow images. Let Ω denote the hole in the optical flow $\mathcal{F}$ which has to be replaced, $\delta\Omega$ the contour of the hole and Ω^c the search region (complement of Ω). Telea-Inpainting approximates the value of a pixel p on the boarder of the fill-front $\delta\Omega$ by a first order Taylor expansion combined with a normalized weighting function $w(p,q)$ for $q \in B_\epsilon(p)$ and $\epsilon > 0$:

$$\hat{\mathcal{F}}(p) = \frac{\sum_{q\in B_\epsilon(p)\cap\Omega^c} w(p,q)[\mathcal{F}(p) - \nabla\mathcal{F}(q)(p-q)]}{\sum_{q\in B_\epsilon(p)\cap\Omega^c} w(p,q)}.$$

The pixel values are propagated into the fill region along the isophotes by solving the eikonal equation: $|\nabla T| = 1$ on Ω, $T = 0$ on $\delta\Omega$ using the Tsitsiklis algorithm [133, 160]. The solution T of the eikonal equation describes the distance map of the pixels inside Ω to its boundary $\delta\Omega$.

8.3.3 Exemplar-Based Inpainting

For the general inpainting, we focused on the exemplar-based inpainting method for region filling and object removal by Criminisi *et al.* [48]. This well known *best-first algorithm* uses texture synthesis and successfully propagates continuities of structures along isophotes to the inpainting region.

Computation of the filling priorities. Let Ω denote the hole to be replaced and $\delta\Omega$ the contour of the hole. For each pixel p along the contour $\delta\Omega$, a filling priority $P(p)$ is computed. $P(p)$ is defined as the product [48]:

$$P(p) = ((1-\omega)\,C(p) + \omega)\,D(p). \tag{8.2}$$

$\omega \in \mathbb{R}$ is a weighting factor. $C(p) := \frac{\sum_{q \in \Psi_p \cap (\mathcal{I} - \Omega)} C(q)}{|\Psi_p|}$ is called the *confidence term* and $D(p) := \frac{\left|\nabla I_p^{\perp} \cdot n_p\right|}{\alpha}$ the *data term*. $|\Psi_p|$ denotes the area of the patch Ψ_p, α is a normalization factor and n_p is a unit vector orthogonal to $\delta\Omega$ in the point p.

The confidence term $C(p)$ measures the amount of reliable information surrounding the pixel p. The intention is to fill first those patches which have more of their pixels already filled. Wang *et al.* [165] introduced the weighting factor ω to control the strong descent of $C(p)$ which accumulates along with the filling. The data term $D(p)$ is a function of the strength of isophotes hitting the contour of the hole. This factor is of fundamental importance because it encourages linear structures to be synthesized first. The pixel $\hat{p}$ with the highest priority: $\hat{p} = \arg\max_{p \in \delta\Omega} P(p)$ defines the center of the target patch $\Psi_{\hat{p}}$ which will be inpainted.

Search for the best matching patch. In the next step, the patch $\Psi_{\hat{q}}$ which best matches the target patch $\Psi_{\hat{p}}$ is searched within the source region Φ. Formally [48]:

$$\Psi_{\hat{q}} = \underset{\Psi_q \in \Phi}{\arg\min}\, d\left(\Psi_{\hat{p}}, \Psi_q\right), \tag{8.3}$$

where the distance $d\left(\cdot, \cdot\right)$ is defined as the sum of squared differences (SSD) of the already filled pixels in the two patches.

This distance, however, is only designed for image inpainting. For the problem of video inpainting the additional temporal dimension is not considered. We present a refined distance function, modeled explicitly to maintain temporal consistency along the video frames. The detailed definition follows in the next Section 8.3.4.

Copy and refresh. When the search for the best matching patch $\Psi_{\hat{q}}$ is completed, the target region $\Psi_{\hat{p}} \cap \Omega$ is inpainted by copying the pixels from $\Psi_{\hat{q}}$ to the target patch $\Psi_{\hat{p}}$. Besides, the boundary of the target region is updated.

The above steps are done iteratively until the target region is fully inpainted.

8.3.4 Flow Preserving Distance Function

The main difficulty of generalizing classical exemplar-based inpainting to videos is maintaining temporal consistency. Therefore, we modify the distance function (8.3) by Criminisi *et al.* [48]. The key idea of our approach is that scenes do not change vastly and changesets can be determined by optical flow. So we assume to already have a well inpainted frame and for further frames to inpaint we demand similarity to this reference frame. The connection between the reference frame and the current inpainting point is obtained via the inpainted optical flow $\hat{F}$ of the original scene (compare Section 8.3.2).

The corresponding distance function reads as follows:

$$\hat{d}(\Psi_{\hat{p}}, \Psi_q) := d(\Psi_{\hat{p}}, \Psi_q) + \frac{\beta}{|\Psi_{\hat{p}} \cap \Phi|}\, d(\Psi_{\hat{F}^{-1}(\hat{p})}, \Psi_q). \tag{8.4}$$

The first term ensures local consistency, as proposed by Criminisi *et al.* The second one enforces similarity to a previous inpainted frame and hence temporal consistency. $\Psi_{\hat{F}^{-1}(\hat{p})}$, using inverse optical flow, points back to the already inpainted image and ensures temporal consistency.

This distance function enables us to reduce complexity of the patch match since we do not have to choose a set of 3D patches. Our algorithm can greedily choose the best patch for the current hole to fill yet can select from all frames to exploit time redundancy. An illustration is shown in Figure 8.1.

8.3.5 Overview of the Algorithm

Interactive Mask Definition. Let $\mathcal{I}[k]$ denote the k'th video frame. The user is asked to roughly scribble (see Section 8.2) the desired regions in the first frame $\mathcal{I}[0]$. These scribbles are propagated via optical flow (Figure 8.2 b) throughout the video. Depending on the user scribbles a two-region segmentation in object Ω (green) and search space Φ (yellow) or a three-region segmentation with additional region Φ_r (red) for neglecting parts is computed: $\mathcal{I} = \Omega \,\dot{\cup}\, \Phi \;(\,\dot{\cup}\, \Phi_r)$.

This processing gives an accurate mask in an easy and quick manner. State-of-the-art methods do not tackle how to obtain an accurate mask definition.

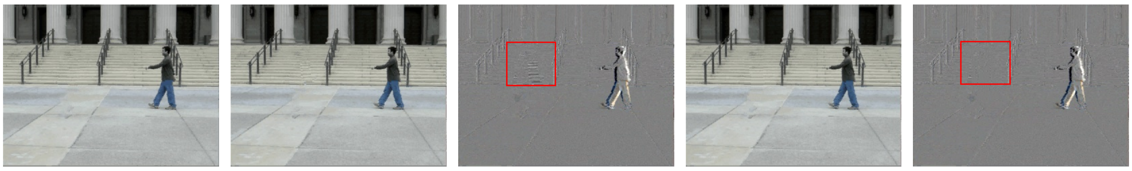
Video Completion by Flow and Color Inpainting. In the proposed image inpainting algorithm one can choose the number of frames to be inpainted at the same time. This allows to exploit redundancy in the video sequence.

Using the inpainted optical flow $\hat{F}$ of the original video sequence we fill the target region Ω step by step according to Criminisi *et al.* using our new distance function (8.4). Our distance function ensures, that the chosen patch is both locally consistent and similar to its origin in a previous inpainted frame. This leads to a temporal consistent inpainted video sequence without any flickering.

8.4 Experiments and Results

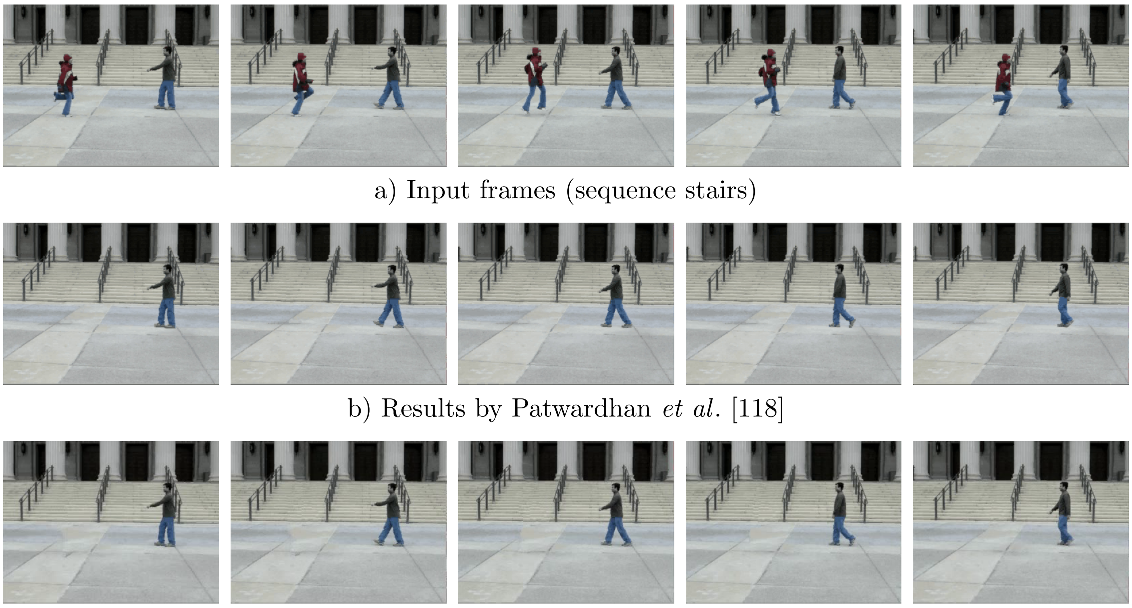
In the following we will show results on various datasets and compare our results to state-of-the-art approaches for video inpainting. The evaluations show that we can handle different object and camera motions.

Depending on the video size we choose a patch-size between 8 × 8 and 12 × 12 and inpaint 3 to 8 frames at the same time to exploit time redundancy. We choose β around 1.1 to weight local and temporal consistency.



a) Frame 1 b) Frame 2a c) Δ_1 d) Frame 2b e) Δ_2

Figure 8.4: **Transition comparison.** Δ_1 shows the transition between a) and b). The transition is computed without regularization and shows strong video flickering. In contrast, the transition Δ_2 with our approach between a) and d) is smooth and does not show disruptive flickering.



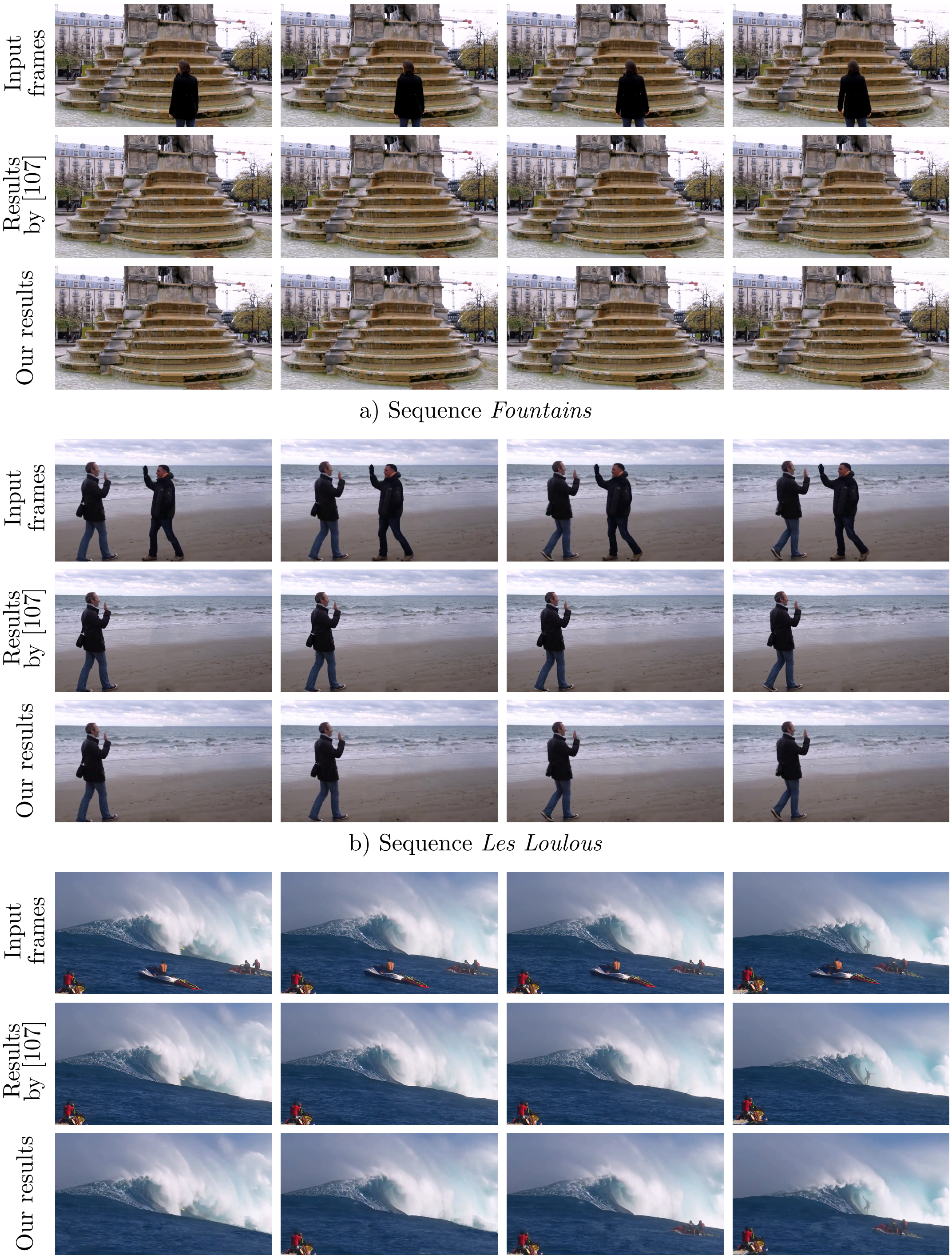
a) Input frames (sequence stairs)

b) Results by Patwardhan *et al.* [118]

c) Our results

Figure 8.5: **Comparison to Motion SSD dataset** with slight camera movement.

In Figure 8.4 we compare two adjacent frames with and without our proposed consistency term. Without the flow consistency term the results have large deviations from one frame to the next one. In the final video such deviations are observed as disruptive flickering. In contrast, the video sequence inpainted with our proposed term shows smooth transitions between the frames. We obtain great results for complex scenes with detailed structures and different types of camera motions at substantially reduced runtime. Figures 8.5 and 8.6 compare our results to the results of Patwardhan *et al.* [118] and Newson *et al.* [107]. Table 8.1 compares the runtime of our method with the state-of-the-art methods [67, 106, 107, 118, 169].



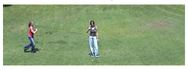
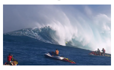
a) Sequence *Fountains*

b) Sequence *Les Loulous*

c) Sequence *Young Jaws*

Figure 8.6: **Our results compared to state-of-the-art methods.** Evaluations on the sequences *Fountains*, *Les Loulous* and *Young Yaws* by Newson *et al.* [107] show that we obtain the same precision of results, whereas our runtime is much faster. Furthermore, we are not restricted to a static mask and can easily remove different objects – see our results of the *Young Jaws* sequence.

Table 8.1: **Runtimes.** Although our approach includes an interactive mask-definition we outperform state-of-the-art methods up to a factor of five.

	Beach Umbrella 264×68×98	Jumping Girl  300×100×239	Stairs 320×240×40	Young Jaws  1280×720×82
Wexler *et al.* [169]	1h	-	-	-
Patwardhan *et al.* [118]	≈ 30 min	≈ 1h 15min	≈ 15 min	-
Granados *et al.* [67]	11 hours	-	-	-
Newson *et al.* [106]	21 min	62 min	-	-
Newson *et al.* [107]	24 min	40 min	-	5h 48 min
Proposed approach	**4.6 min**	**8 min**	**5 min 20 sec**	**3h 20min**

8.4.1 Implementation and Runtime

Runtime is a big challenge to all video inpainting algorithms. Especially on high resolution videos a large amount of data has to be processed. Our parallel implementation takes around 2 to 150 seconds per frame, depending on the resolution of the input video on a NVIDIA GeForce GTX 560 Ti. This outruns state-of-the-art algorithms, requiring much more computing power (like Granados *et al.* [67] on a mainframe with 64 CPUs) and runtime (compare Table 8.1).

8.5 Conclusion

We propose an interactive video completion method which integrates two innovations: Firstly, we replace the tedious hand-labeling of inpainting regions in all video frames by a semi-automatic procedure which consists of a flow-based propagation of user scribbles from the first to subsequent frames followed by an automatic foreground-background segmentation. Secondly, we propose a novel solution for assuring temporal consistency of the inpainting. Rather than performing a computationally intense sampling of space-time patches, we perform an optical flow inpainting followed by a flow-constrained image inpainting. An efficient GPU implementation provides a semi-automatic video inpainting method which requires substantially less user input and provides competitive video inpainting results which is around five times faster than competing methods.

Part III

Conclusion

Chapter 9

Summary

In this thesis we have focused on convex variational methods for semantic image analysis. In the course of the last years, researchers have been investigating how the human visual system interprets images and how algorithms can approximate the human perception. Humans use their experience to classify and understand new observations. Therefore, extensive research has been made to develop algorithms incorporating prior knowledge. Semantic image segmentation uses, *e.g.*, shape priors, hierarchical constraints and geometric relationships to increase the accuracy of segmentation results.

In Chapter 1 we have provided an introduction and have motivated the topic of semantic image analysis. Moreover, we have presented a review of relevant literature in the field of convex variational methods and semantic image analysis.

In Chapter 2 we have discussed the contributions of this thesis as well as the research papers that were published within the scope of this thesis.

In Chapter 3 we have given an introduction to the methodology employed in this thesis. We have provided a discussion of the basic concepts of convex variational methods, such as total variation regularization, convexity, existence of solutions and extremality conditions.

In Chapters 4 to 8 we have presented five selected research papers. All included papers are peer-reviewed publications and have been published in highly ranked journals and international conferences. In the following we summarize the chapters one by one.

The Role of Diffusion in Figure Hunt Games In Chapter 4 we have addressed the task of tracing out target figures in sketch-like binary teeming figure pictures, a popular genre of visual puzzles in which simpler shapes are hidden within more complex organizations. We have discussed how these figures that are hidden in a distractive context can be discovered algorithmically. We have experimentally demonstrated that the key idea is to diffuse the figures (and illustrations). The diffusion propagates information about the figural loci from purely local to a neighborhood. Hence, the desired location can be observed from some distance.

Particularly suited to this task, we have proposed a simple approach for generating diffuse drawings. Our diffuse model keeps the edge information while blurring the contour and imitates curvature coding distance images which are typically computed as solutions to elliptic PDEs. Our proposed approach can be used to search for the unique occurrence of a target figure as well as for various similar objects. By introducing a coarse-to-fine strategy we have been able to speed up the search process.

Midrange Geometric Interactions for Semantic Segmentation In Chapter 5 we have introduced midrange geometric priors for semantic segmentation and recognition within a variational multi-label framework. Instead of introducing co-occurrence probabilities of label combinations, the proposed priors incorporate specific geometric spatial relationships of label pairs, *i.e.*, their direction and distance. It is up to the user to specifically define the spatial extent of the constraint between each two labels. We have called them midrange, since the constraints generalize both global co-occurrence priors, which take into account all labels irrespective of their spatial location, and local co-occurrence priors which are only imposed on directly adjacent pixels.

We have shown how the continuous formulation of the morphological dilation operation can be employed to formulate these constraints within a continuous optimization approach. Moreover, we have given a convex relaxation, which guarantees independence of the initialization. In addition, the proposed approach does not require the computation of superpixels and prevents the emergence of thin artificial ghost regions. Extensive experiments have demonstrated that the proposed novel constraints are beneficial for many segmentation scenarios. In particular for part-based articulated objects such as humans, animals or clothes, for part-based rigid objects, especially man-made items, and for semantic scene segmentation.

Optimizing the Relevance-Redundancy Tradeoff for Efficient Semantic Segmentation In Chapter 6 we have presented a comprehensive study on feature ranking and selection for semantic image segmentation and have introduced a framework for systematic feature analysis.

Therefore, we have discussed various types of features to build up the optimal feature set for an efficient and accurate semantic segmentation. We have shown that by exploiting redundancies in feature sets the computational cost for learning and testing can be significantly decreased. Moreover, we have demonstrated that the key idea is to optimize the relevance-redundancy tradeoff in the feature set. Starting from a large set of popular features, a ranked set of features is sequentially constructed by maximizing the relevance of each feature for the classification task.

At the same time its redundancy is minimized with respect to the previously selected features.

Experiments on different benchmarks have provided a deep understanding on how many and what kind of features to use for semantic segmentation. If available, depth features, for example, provide essential information and strongly boost the accuracy. By integrating the proposed feature analysis into a variational formulation of the multi-labeling problem we have obtained a fully automatic framework for semantic classifications and segmentations. Experiments on five popular benchmarks have demonstrated that our algorithm achieves state-of-the-art semantic classifications and segmentations at drastically reduced computation time.

Interactive Multi-label Segmentation of RGB-D Images In Chapter 7 we have discussed the extension of interactive multi-label segmentation to RGB-D images. Among the various types of user interaction, we have particularly focused on the user input via scribbles and have built upon the spatially varying color distributions proposed by Nieuwenhuis and Cremers [108].

To overcome the problem of non-uniformly distributed user scribbles we have introduced the idea of active scribbles. The experimental evaluations on our benchmark dataset have shown that the benchmark score is improved already when using active scribbles in the framework of Nieuwenhuis and Cremers [108] on RGB images.

Moreover, we have proposed a powerful extension of the spatially varying color distributions [108] to RGB-D images. Our extensions improve the estimation of the data fidelity term and can be divided into two parts: a) We have considered the depth image as an additional color channel. b) We have used the depth information to incorporate the true 3D geometry.

Our experimental results have demonstrated that each of the proposed components contributes separately and improves the segmentation results. Due to the additional depth information, more distinct color descriptions are achieved and reliable segmentations are obtained with significantly less user input.

Flow and Color Inpainting for Video Completion In Chapter 8 we have introduced the concept of interactive segmentation for the task of video completion. We have replaced the tedious hand-labeling of inpainting regions in all video frames by a semi-automatic procedure. The user indicates the region of interest by placing scribbles on the first frame. The scribbles are then transported from the first to subsequent frames by using optical flow and an automatic foreground-background segmentation is computed.

Next to the efficient mask-definition, we have additionally focused on temporal consistency which is one of the major challenges in video inpainting. State-of-the-art methods mostly perform computationally intense sampling of space-time patches. In contrast, we have introduced an efficient solution for assuring temporal consistency of the inpainted frames. Our approach is based on an optical flow inpainting followed by a flow-constrained image inpainting.

Our semi-automatic video inpainting method has two major advantages: a) It requires substantially less user input than competitive approaches. b) It achieves competitive video inpainting results at around five times faster runtime than competing methods.

Chapter 10

Limitations and Future Research

In this thesis we have proposed novel methods for several scenarios in the field of semantic image analysis. Our proposed approaches achieve promising results and compare favorably to competitive state-of-the-art methods. Nevertheless, we also see some limitations and possible extensions which might be interesting to address in future research.

In the following we discuss the limitations of the presented approaches and present some ideas for possible extensions and future work.

Algorithmic search in binary line drawings In our publication [8] we have formulated and solved an optimization problem in order to detect the best matching of the target figure with the illustration. We have determined the parameters leading to the optimal fit via a probabilistic algorithm. More specifically, we have used a genetic algorithm based optimization which is readily available in the Matlab environment. A genetic algorithm is a search heuristic. The algorithm is non-deterministic and there is no proof of optimality known. In [8] we have chosen the genetic algorithm because it provides a good tradeoff between speed and quality of computed solutions.

For future work we suggest to work on a new formulation which can be solved globally optimal by applying established methods of convex optimization.

Midrange geometric constraints The midrange geometric constraints introduced in [7] have been designed for the segmentation of RGB images. Considering the recent increase and availability of depth-sensing cameras such as the Kinect, one could investigate the extension to RGB-D input data. In the current approach, the penalty matrix is specified by considering the occurrence of labels in the vicinity of each other. ‘Vicinity’ is defined by the pixel distance in the image plane. To examine if two regions are close to each other, a dilation operation is used.

With RGB-D input data, some information about the 3-dimensional structure of the scene is provided by the depth image. To take advantage of the additional depth information, one could consider the reformulation of the proposed constraints. In particular, one could transfer the 'vicinity' of two labels in the plane to the volume and incorporate the depth information for the specification of the penalty matrix.

Data fidelity term for efficient semantic segmentation In our work [9] we have addressed the question which and how many features are appropriate for an efficient semantic segmentation. To this end, we have first ranked the features according to their significance, then we have analyzed them and have proposed an automatic selection criterion. For the feature ranking, we have applied a min-redundancy max-relevance (mRMR) criterion [119]. The mRMR algorithm, however, is an incremental search scheme that maximizes a certain criterion with respect to a single feature at a time. For the task of image classification, however, several features usually are interacting. This may cause a suboptimal feature set as irrelevant features might be selected earlier than relevant and/or redundant features.

For future applications, we therefore suggest to experiment with a different ranking strategy. Estévez *et al.* [56], for example, suggested a feature selection method called GAMIFS, a genetic algorithm guided by mutual information for feature selection, particularly designed for problems where groups of features are relevant. It is a hybrid method combining a genetic algorithm with a normalized mutual information feature selection.

Interactive RGB-D segmentation For the task of interactive segmentation we have extended the concept of spatially varying color distributions proposed for RGB images to RGB-D input data [5]. We have incorporated the depth information in two ways to obtain more distinct color distributions: a) We have considered the depth image as an additional data channel. b) We have computed the color distributions based on the object's position in the 3D space.

For the computation of the regularizer the proposed approach, however, only makes use of the RGB image. For future work, one could think about using the depth information for the regularization too. Rosman *et al.* [127], for example, suggested a regularization that takes into account the geometry of the 3D surface.

Image sequence segmentation and video completion Most video completion approaches require immense user input to specify the inpainting region. To overcome this problem, we have introduced a semi-automatic procedure with minimal user input [10]. The user input is given by user scribbles drawn on the input image. The scribbles are automatically relocated throughout the video sequence via optical flow and a frame-wise image segmentation method is applied.

While the current approach computes the segmentation frame-by-frame separately, for future work, one could extend the segmentation algorithm to treat the video as a 3D data cube. For complex video sequences, the 3D volume might provide additional information leading to more accurate segmentation results.

Own Publications

[1] G. Bal, J. Diebold, E. W. Chambers, E. Gasparovic, R. Hu, K. Leonard, M. Shaker, and C. Wenk. Skeleton-Based Recognition of Shapes in Images via Longest Path Matching. Chapter in *Research in Shape Modeling. Association for Women in Mathematics Series.* Vol. 1. Springer International Publishing, 2015, pp. 81–99. isbn: 978-3-319-16347-5. doi: 10.1007/978-3-319-16348-2_6 (cited on pp. 16, 17).

[2] J. Bergbauer, C. Nieuwenhuis, M. Souiai, and D. Cremers. Proximity Priors for Variational Semantic Segmentation and Recognition. In *IEEE International Conference on Computer Vision (ICCV), Workshop on Graphical Models for Scene Understanding (GMSU)*, Dec. 2013[1], pp. 15–21. doi: 10.1109/ICCVW.2013.132 (cited on pp. 15, 17, 66, 68, 104).

[3] J. Bergbauer and S. Tari. Top-down visual search in Wimmelbild. In *Proceedings of SPIE, Human Vision and Electronic Imaging XVIII.* Vol. 8651, 2013[2]. doi: 10.1117/12.2006160 (cited on pp. 15, 17).

[4] J. Bergbauer and S. Tari. Wimmelbild Analysis with Approximate Curvature Coding Distance Images. In *Scale Space and Variational Methods in Computer Vision (SSVM). Lecture Notes in Computer Science.* Vol. 7893. Springer Berlin Heidelberg, 2013[2], pp. 489–500. isbn: 978-3-642-38266-6. doi: 10.1007/978-3-642-38267-3_41. Oral Presentation (cited on pp. 15, 17, 33).

[5] J. Diebold, N. Demmel, C. Hazirbaş, M. Möller, and D. Cremers. Interactive Multi-label Segmentation of RGB-D Images. In *Scale Space and Variational Methods in Computer Vision (SSVM). Lecture Notes in Computer Science.* Vol. 9087. Springer International Publishing, 2015, pp. 294–306. isbn: 978-3-319-18460-9. doi: 10.1007/978-3-319-18461-6_24 (cited on pp. 11–13, 15–17, 117, 147, 150).

[1] Includes results of the Master's Thesis: J. Bergbauer. Variational Image Segmentation with Region Adjacency Constraints. MA thesis. Department of Informatics, Technische Universität München, Sept. 2012.

[2] Includes results of the project work: J. Bergbauer. Locating a given partial shape in a complete one using diffused distance-like representations. Lecture Mathematical Morphology, Department of Mathematics, Technische Universität München, Mar. 2012.

[6] J. Diebold, M. Möller, G. Gilboa, and D. Cremers. Learning Nonlinear Spectral Filters for Color Image Reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. Submitted (cited on p. 16).

[7] J. Diebold, C. Nieuwenhuis, and D. Cremers. Midrange Geometric Interactions for Semantic Segmentation. Constraints for Continuous Multi-label Optimization. In *International Journal of Computer Vision (IJCV)*: *Special Issue on Graphical Models for Scene Understanding*. Springer US, 2015[1]. doi: `10.1007/s11263-015-0828-7` (cited on pp. 9, 11–13, 15, 17, 63, 146, 149).

[8] J. Diebold, S. Tari, and D. Cremers. The Role of Diffusion in Figure Hunt Games. In *Journal of Mathematical Imaging and Vision (JMIV)*, 52(1):108–123: *Special Issue on Scale Space and Variational Methods in Computer Vision*. Springer US, 2015[2]. doi: `10.1007/s10851-014-0548-6` (cited on pp. 7, 11–13, 15, 17, 31, 145, 149).

[9] C. Hazırbaş, J. Diebold, and D. Cremers. Optimizing the Relevance-Redundancy Tradeoff for Efficient Semantic Segmentation. In *Scale Space and Variational Methods in Computer Vision (SSVM). Lecture Notes in Computer Science*. Vol. 9087. Springer International Publishing, 2015, pp. 243–255. isbn: 978-3-319-18460-9. doi: `10.1007/978-3-319-18461-6_20`. Oral Presentation (cited on pp. 10–15, 17, 103, 146, 150).

[10] M. Strobel, J. Diebold, and D. Cremers. Flow and Color Inpainting for Video Completion. In *German Conference on Pattern Recognition (GCPR). Lecture Notes in Computer Science*. Vol. 8753. Springer International Publishing, 2014, pp. 293–304. isbn: 978-3-319-11751-5. doi: `10.1007/978-3-319-11752-2_23`. Oral Presentation (cited on pp. 10–13, 15, 17, 131, 147, 150).

Bibliography

[11] L. Ambrosio and V. M. Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. In *Communications on Pure and Applied Mathematics*, 43(8):999–1036. Wiley Subscription Services, Inc., A Wiley Company, 1990. doi: 10.1002/cpa.3160430805 (cited on p. 34).

[12] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 3378–3385. doi: 10.1109/CVPR.2012.6248077 (cited on p. 66).

[13] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009, pp. 2294–2301. doi: 10.1109/CVPR.2009.5206707 (cited on pp. 10, 118).

[14] M. Ashikhmin. Synthesizing Natural Textures. In *Symposium on Interactive 3D Graphics (I3D)*. ACM, 2001, pp. 217–226. isbn: 1-58113-292-1. doi: 10.1145/364338.364405 (cited on p. 132).

[15] C. Asian and S. Tari. An axis-based representation for recognition. In *IEEE International Conference on Computer Vision (ICCV)*. Vol. 2, Oct. 2005, pp. 1339–1346. doi: 10.1109/ICCV.2005.32 (cited on p. 34).

[16] H. Attouch, G. Buttazzo, and G. Michaille. *Variational Analysis in Sobolev and BV Spaces. Applications to PDEs and Optimization.* Of *MOS-SIAM Series on Optimization.* Society for Industrial and Applied Mathematics, 2nd ed., 2014. isbn: 978-1-61197-347-1. doi: 10.1137/1.9781611973488 (cited on pp. 25, 26, 28).

[17] G. Aubert and J.-F. Aujol. Poisson Skeleton Revisited: a New Mathematical Perspective. In *Journal of Mathematical Imaging and Vision (JMIV)*, 48(1):149–159. Springer US, 2014. doi: 10.1007/s10851-012-0404-5 (cited on p. 34).

[18] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing. Partial Differential Equations and the Calculus of Variations.* Vol. 147 of *Applied Mathematical Sciences.* Springer New York, 2006. isbn: 978-0-387-32200-1. doi: 10.1007/978-0-387-44588-5 (cited on pp. 21, 23, 27).

[19] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. In *Pattern Recognition*, 13(2):111–122. Elsevier B.V., 1981. doi: 10.1016/0031-3203(81)90009-1 (cited on pp. 6, 33).

[20] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. In *ACM Transactions on Graphics (TOG)*, 28(3):1–11, July 2009. doi: 10.1145/1531326.1531330 (cited on p. 133).

[21] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching. In *International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 2. William Kaufmann, Aug. 1977, pp. 659–663 (cited on pp. 7, 34).

[22] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 3169–3176. doi: 10.1109/CVPR.2010.5540080 (cited on pp. 10, 68, 82, 101, 125).

[23] Y. Benyamini and J. Lindenstrauss. *Geometric Nonlinear Functional Analysis*. Vol. 1 of *Colloquium Publications*. American Mathematical Society, 2000. isbn: 978-0-8218-0835-1 (cited on p. 27).

[24] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1, 2001, pp. 355–362. doi: 10.1109/CVPR.2001.990497 (cited on pp. 132, 134, 136).

[25] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image Inpainting. In *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2000, pp. 417–424. isbn: 1-58113-208-5. doi: 10.1145/344779.344972 (cited on p. 132).

[26] J. R. Beveridge and E. M. Riseman. How easy is matching 2D line models using local search? In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(6):564–579, June 1997. doi: 10.1109/34.601245 (cited on p. 6).

[27] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive Image Segmentation Using an Adaptive GMMRF Model. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science*. Vol. 3021. Springer Berlin Heidelberg, 2004, pp. 428–441. isbn: 978-3-540-21984-2. doi: 10.1007/978-3-540-24670-1_33 (cited on pp. 10, 11, 118).

[28] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987. isbn: 0-262-02271-0 (cited on p. 42).

[29] Y. Bo and C. C. Fowlkes. Shape-based pedestrian parsing. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 2265–2272. doi: `10.1109/CVPR.2011.5995609` (cited on pp. 86, 87, 95–97).

[30] W. Bouachir, A. Torabi, G.-A. Bilodeau, and P. Blais. A Bag of Words Approach for Semantic Segmentation of Monitored Scenes. In *Computing Research Repository (CoRR)*, abs/1305.3189, 2013 (cited on p. 7).

[31] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *IEEE International Conference on Computer Vision (ICCV)*. Vol. 1, 2001, pp. 105–112. doi: `10.1109/ICCV.2001.937505` (cited on pp. 10, 119).

[32] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239, Nov. 2001. doi: `10.1109/34.969114` (cited on p. 8).

[33] K. Bredies, K. Kunisch, and T. Pock. Total Generalized Variation. In *SIAM Journal on Imaging Sciences (SIIMS)*, 3(3):492–526, 2010. doi: `10.1137/090769521` (cited on p. 21).

[34] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science.* Vol. 3024. Springer Berlin Heidelberg, 2004, pp. 25–36. isbn: 978-3-540-21981-1. doi: `10.1007/978-3-540-24673-2_3` (cited on p. 135).

[35] M. Burger and S. Osher. A Guide to the TV Zoo. Chapter in *Level Set and PDE Based Reconstruction Methods in Imaging.* Vol. 2090. In Lecture Notes in Mathematics. Springer International Publishing, 2013, pp. 1–70. isbn: 978-3-319-01711-2. doi: `10.1007/978-3-319-01712-9_1` (cited on p. 26).

[36] D. J. Burr. Elastic Matching of Line Drawings. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, (6):708–713, Nov. 1981. doi: `10.1109/TPAMI.1981.4767176` (cited on p. 6).

[37] C. Canuto and A. Tabacco. *Mathematical Analysis II.* Springer International Publishing, 2nd ed., 2015. isbn: 978-3-319-12756-9. doi: `10.1007/978-3-319-12757-6` (cited on p. 26).

[38] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic Segmentation with Second-Order Pooling. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science.* Vol. 7578. Springer Berlin Heidelberg, 2012, pp. 430–443. isbn: 978-3-642-33785-7. doi: `10.1007/978-3-642-33786-4_32` (cited on p. 66).

[39] J. Carreira and C. Sminchisescu. CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(7):1312–1328, July 2012. doi: 10.1109/TPAMI.2011.231 (cited on p. 66).

[40] A. Chambolle, D. Cremers, and T. Pock. A Convex Approach to Minimal Partitions. In *SIAM Journal on Imaging Sciences (SIIMS)*, 5(4):1113–1158, 2012. doi: 10.1137/110856733 (cited on pp. 8, 21, 114).

[41] A. Chambolle and T. Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. In *Journal of Mathematical Imaging and Vision (JMIV)*, 40(1):120–145. Springer US, 2011. doi: 10.1007/s10851-010-0251-1 (cited on pp. 19, 82).

[42] T. F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models. In *SIAM Journal on Applied Mathematics (SIAP)*, 66(5):1632–1648, 2006. doi: 10.1137/040615286 (cited on p. 19).

[43] H.-K. Chu, W.-H. Hsu, N. J. Mitra, D. Cohen-Or, T.-T. Wong, and T.-Y. Lee. Camouflage Images. In *ACM Transactions on Graphics (TOG)*, 29(4):1–8, July 2010. doi: 10.1145/1778765.1778788 (cited on p. 34).

[44] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor Semantic Segmentation using depth information. In *International Conference on Learning Representations (ICLR)*, 2013 (cited on pp. 106, 107, 111, 114, 115, 118).

[45] R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Vol. 1. John Wiley & Sons, Inc., 1989. isbn: 978-0-471-50447-4 (cited on pp. 27, 28).

[46] D. Cremers, F. R. Schmidt, and F. Barthel. Shape priors in variational image segmentation: Convexity, Lipschitz continuity and globally optimal solutions. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008, pp. 1–6. doi: 10.1109/CVPR.2008.4587446 (cited on p. 7).

[47] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2, June 2003, pp. 721–728. doi: 10.1109/CVPR.2003.1211538 (cited on p. 133).

[48] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. In *IEEE Transactions on Image Processing*, 13(9):1200–1212, Sept. 2004. doi: 10.1109/TIP.2004.833105 (cited on pp. 132, 133, 136–138).

[49] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).* Vol. 1, June 2005, pp. 886–893. doi: 10.1109/CVPR.2005.177 (cited on pp. 9, 106).

[50] A. Delong and Y. Boykov. Globally optimal segmentation of multi-region objects. In *IEEE International Conference on Computer Vision (ICCV)*, Sept. 2009, pp. 285–292. doi: 10.1109/ICCV.2009.5459263 (cited on pp. 66, 67).

[51] A. Delong, L. Gorelick, O. Veksler, and Y. Boykov. Minimizing Energies with Hierarchical Costs. In *International Journal of Computer Vision (IJCV)*, 100(1):38–58. Springer US, 2012. doi: 10.1007/s11263-012-0531-x (cited on pp. 8, 67).

[52] L. R. Dice. Measures of the Amount of Ecologic Association Between Species. In *Ecology*, 26(3):297–302. Ecological Society of America, July 1945. doi: 10.2307/1932409 (cited on pp. 87, 95).

[53] A. A. Efros and W. T. Freeman. Image Quilting for Texture Synthesis and Transfer. In *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2001, pp. 341–346. isbn: 1-58113-374-X. doi: 10.1145/383259.383296 (cited on p. 132).

[54] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision (ICCV).* Vol. 2, 1999, pp. 1033–1038. doi: 10.1109/ICCV.1999.790383 (cited on p. 132).

[55] E. Esser, X. Zhang, and T. F. Chan. A General Framework for a Class of First Order Primal-Dual Algorithms for Convex Optimization in Imaging Science. In *SIAM Journal on Imaging Sciences (SIIMS)*, 3(4):1015–1046, 2010. doi: 10.1137/09076934X (cited on p. 125).

[56] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada. Normalized Mutual Information Feature Selection. In *IEEE Transactions on Neural Networks*, 20(2):189–201, Feb. 2009. doi: 10.1109/TNN.2008.2005601 (cited on p. 150).

[57] P. F. Felzenszwalb and O. Veksler. Tiered scene labeling with dynamic programming. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 3097–3104. doi: 10.1109/CVPR.2010.5540067 (cited on pp. 8, 67).

[58] M. Fornasier and D. Toniolo. Fast, Robust and Efficient 2D Pattern Recognition for Re-assembling Fragmented Images. In *Pattern Recognition*, 38(11):2074–2087. Elsevier Science Inc., Nov. 2005. doi: 10.1016/j.patcog.2005.03.014 (cited on p. 34).

[59] B. Fröhlich, E. Rodner, and J. Denzler. Semantic Segmentation with Millions of Features: Integrating Multiple Cues in a Combined Random Forest Approach. In *Asian Conference on Computer Vision (ACCV). Lecture Notes in Computer Science.* Vol. 7724. Springer Berlin Heidelberg, 2012, pp. 218–231. isbn: 978-3-642-37330-5. doi: 10.1007/978-3-642-37331-2_17 (cited on pp. 10, 66, 106, 107, 109, 111, 114, 115).

[60] R. Gâteaux. Fonctions d'une infinité de variables indépendantes. In *Bulletin de la Société Mathématique de France*, 47:70–96, 1919 (cited on p. 27).

[61] G. Gilboa and S. Osher. Nonlocal Operators with Applications to Image Processing. In *SIAM Multiscale Modeling & Simulation*, 7(3):1005–1028, 2009. doi: 10.1137/070698592 (cited on p. 21).

[62] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 580–587. doi: 10.1109/CVPR.2014.81 (cited on p. 66).

[63] E. Giusti. *Minimal Surfaces and Functions of Bounded Variation.* Vol. 80 of *Monographs in Mathematics.* Birkhäuser Boston, 1984. isbn: 978-0-8176-3153-6. doi: 10.1007/978-1-4684-9486-0 (cited on p. 20).

[64] K. Gottschaldt. Über den Einfluß der Erfahrung auf die Wahrnehmung von Figuren. In *Psychologische Forschung*, 8(1):261–317. Springer-Verlag, 1926. doi: 10.1007/BF02411523 (cited on p. 32).

[65] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-Class Segmentation with Relative Location Prior. In *International Journal of Computer Vision (IJCV)*, 80(3):300–316. Springer US, 2008. doi: 10.1007/s11263-008-0140-x (cited on pp. 9, 67, 97, 99).

[66] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt. Background Inpainting for Videos with Dynamic Objects and a Free-Moving Camera. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science.* Vol. 7572. Springer Berlin Heidelberg, 2012, pp. 682–695. isbn: 978-3-642-33717-8. doi: 10.1007/978-3-642-33718-5_49 (cited on p. 136).

[67] M. Granados, J. Tompkin, K. I. Kim, O. Grau, J. Kautz, and C. Theobalt. How Not to Be Seen – Object Removal from Videos of Crowded Scenes. In *Computer graphics forum*, 31(2):219–228. John Wiley & Sons, Inc., May 2012. doi: 10.1111/j.1467-8659.2012.03000.x (cited on pp. 133, 134, 136, 140, 142).

[68] K. S. Gurumoorthy and A. Rangarajan. A Schrödinger Equation for the Fast Computation of Approximate Euclidean Distance Functions. In *Scale Space and Variational Methods in Computer Vision (SSVM). Lecture Notes in Computer Science*. Vol. 5567. Springer Berlin Heidelberg, 2009, pp. 100–111. isbn: 978-3-642-02255-5. doi: 10.1007/978-3-642-02256-2_9 (cited on p. 34).

[69] X. He, R. S. Zemel, and M. A. Carreira-Perpinñán. Multiscale conditional random fields for image labeling. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2, June 2004, pp. 695–702. doi: 10.1109/CVPR.2004.1315232 (cited on p. 111).

[70] A. Hermans, G. Floros, and B. Leibe. Dense 3D Semantic Mapping of Indoor Scenes from RGB-D Images. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2631–2638. doi: 10.1109/ICRA.2014.6907236 (cited on pp. 7, 10, 106, 107, 109, 111, 114, 115, 118).

[71] J. Hernandez and B. Marcotegui. Morphological segmentation of building façade images. In *IEEE International Conference on Image Processing (ICIP)*, Nov. 2009, pp. 4029–4032. doi: 10.1109/ICIP.2009.5413756 (cited on p. 118).

[72] P. V. C. Hough. Method and means for recognizing complex patterns. US Patent 3,069,654. Dec. 1962 (cited on p. 6).

[73] H. Ishikawa. Exact optimization for Markov random fields with convex priors. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(10):1333–1336, Oct. 2003. doi: 10.1109/TPAMI.2003.1233908 (cited on p. 8).

[74] A. K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(12):1386–1390, Dec. 1997. doi: 10.1109/34.643899 (cited on p. 6).

[75] H. Keles, M. Özkar, and S. Tari. Weighted shapes for embedding perceived wholes. In *Environment and Planning B: Planning and Design*, 39(2):360–375. Pion Ltd, 2012. doi: 10.1068/b37067 (cited on p. 34).

[76] M. Klodt and D. Cremers. A convex framework for image segmentation with moment constraints. In *IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 2236–2243. doi: 10.1109/ICCV.2011.6126502 (cited on p. 8).

[77] P. Kohli, M. P. Kumar, and P. H. Torr. P3 & Beyond: Solving Energies with Higher Order Cliques. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007, pp. 1–8. doi: 10.1109/CVPR.2007.383204 (cited on p. 67).

[78] P. Kohli, L. Ladický, and P. H. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. In *International Journal of Computer Vision (IJCV)*, 82(3):302–324. Springer US, 2009. doi: `10.1007/s11263-008-0202-0` (cited on pp. 95, 122).

[79] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009, pp. 2985–2992. doi: `10.1109/CVPR.2009.5206846` (cited on p. 67).

[80] P. Kontschieder, P. Kohli, J. Shotton, and A. Criminisi. GeoF: Geodesic Forests for Learning Coupled Predictors. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 65–72. doi: `10.1109/CVPR.2013.16` (cited on p. 67).

[81] F. Korč and W. Förstner. eTRIMS Image Database for Interpreting Images of Man-Made Scenes. Tech. rep. (TR-IGG-P-2009-01). Department of Photogrammetry, University of Bonn, 2009 (cited on pp. 68, 83, 87, 90, 98, 111).

[82] L. Ladický, C. Russell, P. Kohli, and P. H. Torr. Graph Cut Based Inference with Co-occurrence Statistics. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science.* Vol. 6315. Springer Berlin Heidelberg, 2010, pp. 239–253. isbn: 978-3-642-15554-3. doi: `10.1007/978-3-642-15555-0_18` (cited on pp. 8, 65, 66, 71, 86–90, 92, 94, 96–99, 101, 104).

[83] L. Ladický, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical CRFs for object class image segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, Sept. 2009, pp. 739–746. doi: `10.1109/ICCV.2009.5459248` (cited on pp. 9, 71).

[84] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, Where and How Many? Combining Object Detectors and CRFs. In *European Conference on Computer Vision (ECCV).* K. Daniilidis, P. Maragos, and N. Paragios, editors. Vol. 6314. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 424–437. isbn: 978-3-642-15560-4. doi: `10.1007/978-3-642-15561-1_31` (cited on p. 10).

[85] M. Lang, O. Wang, T. Aydin, A. Smolic, and M. Gross. Practical Temporal Consistency for Image-based Graphics Applications. In *ACM Transactions on Graphics (TOG)*, 31(4):1–8, July 2012. doi: `10.1145/2185520.2185530` (cited on p. 10).

[86] Y. G. Leclerc. Region growing using the MDL principle. In *DARPA Image Understanding Workshop*, 1990 (cited on p. 7).

[87] S. Lefkimmiatis, A. Bourquard, and M. Unser. Hessian-Based Norm Regularization for Image Restoration With Biomedical Applications. In *IEEE Transactions on Image Processing*, 21(3):983–995, Mar. 2012. doi: 10.1109/TIP.2011.2168232 (cited on p. 21).

[88] J. Lellmann, F. Becker, and C. Schnörr. Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In *IEEE International Conference on Computer Vision (ICCV)*, Sept. 2009, pp. 646–653. doi: 10.1109/ICCV.2009.5459176 (cited on p. 8).

[89] V. Lempitsky, A. Blake, and C. Rother. Branch-and-Mincut: Global Optimization for Image Segmentation with High-Level Priors. In *Journal of Mathematical Imaging and Vision (JMIV)*, 44(3):315–329. Springer US, 2012. doi: 10.1007/s10851-012-0328-0 (cited on p. 7).

[90] A. Levin, D. Lischinski, and Y. Weiss. Colorization Using Optimization. In *ACM Transactions on Graphics (TOG)*, 23(3):689–694, Aug. 2004. doi: 10.1145/1015706.1015780 (cited on p. 121).

[91] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy Snapping. In *ACM Transactions on Graphics (TOG)*, 23(3):303–308, Aug. 2004. doi: 10.1145/1015706.1015719 (cited on pp. 11, 118).

[92] D. Liu, K. Pulli, L. G. Shapiro, and Y. Xiong. Fast Interactive Image Segmentation by Discriminative Clustering. In *ACM Multimedia Workshop on Mobile Cloud Media Computing*, 2010, pp. 47–52. isbn: 978-1-4503-0168-8. doi: 10.1145/1877953.1877967 (cited on pp. 10, 11, 118).

[93] X. Liu, O. Veksler, and J. Samarabandu. Order-Preserving Moves for Graph-Cut-Based Optimization. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(7):1182–1196, July 2010. doi: 10.1109/TPAMI.2009.120 (cited on pp. 8, 67, 68, 83, 87, 90, 100, 101).

[94] H. Lombaert, Y. Sun, L. Grady, and C. Xu. A multilevel banded graph cuts method for fast image segmentation. In *IEEE International Conference on Computer Vision (ICCV)*. Vol. 1, Oct. 2005, pp. 259–265. doi: 10.1109/ICCV.2005.13 (cited on pp. 10, 119).

[95] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*. Vol. 2, Sept. 1999, pp. 1150–1157. doi: 10.1109/ICCV.1999.790410 (cited on pp. 9, 106).

[96] A. Lucchi, Y. Li, X. Boix, K. Smith, and P. Fua. Are spatial and global constraints really necessary for segmentation? In *IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 9–16. doi: 10.1109/ICCV.2011.6126219 (cited on pp. 98, 99).

[97] P. Luo, X. Wang, and X. Tang. Pedestrian Parsing via Deep Decompositional Network. In *IEEE International Conference on Computer Vision (ICCV)*, Dec. 2013, pp. 2648–2655. doi: 10.1109/ICCV.2013.329 (cited on pp. 96, 97).

[98] T. Ma, X. Yang, and L. J. Latecki. Boosting Chamfer Matching by Learning Chamfer Distance Normalization. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science*. Vol. 6315. Springer Berlin Heidelberg, 2010, pp. 450–463. isbn: 978-3-642-15554-3. doi: 10.1007/978-3-642-15555-0_33 (cited on p. 34).

[99] M. Mainberger, C. Schmaltz, M. Berg, J. Weickert, and M. Backes. Diffusion-Based Image Compression in Steganography. In *International Symposium on Advances in Visual Computing (ISVC). Lecture Notes in Computer Science*. Vol. 7432. Springer Berlin Heidelberg, 2012, pp. 219–228. isbn: 978-3-642-33190-9. doi: 10.1007/978-3-642-33191-6_22 (cited on p. 34).

[100] T. Malisiewicz and A. A. Efros. Improving Spatial Support for Objects via Multiple Segmentations. In *British Machine Vision Conference (BMVC)*. BMVA Press, Sept. 2007, pp. 1–10. isbn: 1-901725-34-0. doi: 10.5244/C.21.55 (cited on pp. 98, 100).

[101] S. Masnou. Disocclusion: a variational approach using level lines. In *IEEE Transactions on Image Processing*, 11(2):68–76, Feb. 2002. doi: 10.1109/83.982815 (cited on p. 132).

[102] S. Masnou and J.-M. Morel. Level lines based disocclusion. In *IEEE International Conference on Image Processing (ICIP)*. Vol. 3, Oct. 1998, pp. 259–263. doi: 10.1109/ICIP.1998.999016 (cited on p. 132).

[103] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. In *Journal of optimization theory and applications*, 50(1):195–200. Plenum Publishing Corporation, July 1986. doi: 10.1007/BF00938486 (cited on p. 81).

[104] T. Möllenhoff, C. Nieuwenhuis, E. Töppe, and D. Cremers. Efficient Convex Optimization for Minimal Partition Problems with Volume Constraints. In *International Conference on Energy Minimization Methods for Computer Vision and Pattern Recognition (EMMCVPR). Lecture Notes in Computer Science*. Vol. 8081. Springer Berlin Heidelberg, 2013, pp. 94–107. isbn: 978-3-642-40394-1. doi: 10.1007/978-3-642-40395-8_8 (cited on p. 65).

[105] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. In *Communications on Pure and Applied Mathematics*, 42(5):577–685. Wiley Subscription Services, Inc., A Wiley Company, 1989. doi: 10.1002/cpa.3160420503 (cited on p. 34).

[106] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Towards Fast, Generic Video Inpainting. In *ACM European Conference on Visual Media Production (CVMP)*, 2013, pp. 1–8. isbn: 978-1-4503-2589-9. doi: 10.1145/2534008.2534019 (cited on pp. 133, 136, 140, 142).

[107] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Video Inpainting of Complex Scenes. In *SIAM Journal on Imaging Sciences (SIIMS)*, 7(4):1993–2019, 2014. doi: 10.1137/140954933 (cited on pp. 133, 135, 140–142).

[108] C. Nieuwenhuis and D. Cremers. Spatially Varying Color Distributions for Interactive Multilabel Segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(5):1234–1247, May 2013. doi: 10.1109/TPAMI.2012.183 (cited on pp. 11, 71, 118, 119, 121–123, 126–129, 134–136, 147).

[109] C. Nieuwenhuis, S. Hawe, M. Kleinsteuber, and D. Cremers. Co-Sparse Textural Similarity for Interactive Segmentation. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science.* Vol. 8694. Springer International Publishing, 2014, pp. 285–301. isbn: 978-3-319-10598-7. doi: 10.1007/978-3-319-10599-4_19 (cited on pp. 10, 11, 118).

[110] C. Nieuwenhuis, E. Strekalovskiy, and D. Cremers. Proportion Priors for Image Sequence Segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, Dec. 2013, pp. 2328–2335. doi: 10.1109/ICCV.2013.289 (cited on pp. 7, 10, 65–67, 104).

[111] C. Nieuwenhuis, E. Töppe, and D. Cremers. A Survey and Comparison of Discrete and Continuous Multi-label Optimization Approaches for the Potts Model. In *International Journal of Computer Vision (IJCV)*, 104(3):223–240. Springer US, 2013. doi: 10.1007/s11263-013-0619-y (cited on pp. 8, 69, 70).

[112] M. S. Nosrati, S. Andrews, and G. Hamarneh. Bounded Labeling Function for Global Segmentation of Multi-part Objects with Geometric Constraints. In *IEEE International Conference on Computer Vision (ICCV)*, Dec. 2013, pp. 2032–2039. doi: 10.1109/ICCV.2013.254 (cited on pp. 66, 67).

[113] S. Nowozin and C. H. Lampert. Global connectivity potentials for random field models. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009, pp. 818–825. doi: 10.1109/CVPR.2009.5206567 (cited on p. 7).

[114] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer New York, 2003. isbn: 978-0-387-95488-2. doi: 10.1007/b97541 (cited on p. 34).

[115] K. Papafitsoros and C. B. Schönlieb. A Combined First and Second Order Variational Approach for Image Reconstruction. In *Journal of Mathematical Imaging and Vision (JMIV)*, 48(2):308–338. Springer US, 2014. doi: 10.1007/s10851-013-0445-4 (cited on p. 21).

[116] N. Paragios. A Variational Approach for the Segmentation of the Left Ventricle in Cardiac Image Analysis. In *International Journal of Computer Vision (IJCV)*, 50(3):345–362. Kluwer Academic Publishers, 2002. doi: 10.1023/A:1020882509893 (cited on p. 34).

[117] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. In *IEEE International Conference on Image Processing (ICIP)*. Vol. 2, Sept. 2005, pp. 69–72. doi: 10.1109/ICIP.2005.1529993 (cited on p. 133).

[118] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video Inpainting Under Constrained Camera Motion. In *IEEE Transactions on Image Processing*, 16(2):545–553, Feb. 2007. doi: 10.1109/TIP.2006.888343 (cited on pp. 133, 136, 140, 142).

[119] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(8):1226–1238, Aug. 2005. doi: 10.1109/TPAMI.2005.159 (cited on pp. 108, 150).

[120] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 1762–1769. doi: 10.1109/ICCV.2011.6126441 (cited on pp. 69, 82, 125).

[121] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009, pp. 810–817. doi: 10.1109/CVPR.2009.5206604 (cited on p. 82).

[122] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *IEEE International Conference on Computer Vision (ICCV)*, Sept. 2009, pp. 1133–1140. doi: 10.1109/ICCV.2009.5459348 (cited on pp. 68, 82, 125).

[123] R. B. Potts. Some generalized order-disorder transformations. In *Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952. doi: 10.1017/S0305004100027419 (cited on p. 8).

[124] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2006, pp. 1129–1136 (cited on pp. 68, 82, 101).

[125] A. Richtsfeld, T. Morwald, J. Prankl, M. Zillich, and M. Vincze. Segmentation of unknown objects in indoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 4791–4796. doi: 10.1109/IROS.2012.6385661 (cited on p. 125).

[126] R. T. Rockafellar. *Convex Analysis*. Of *Princeton Landmarks in Mathematics*. Princeton University Press, 1997. isbn: 9780691015866 (cited on p. 23).

[127] G. Rosman, A. M. Bronstein, M. M. Bronstein, X.-C. Tai, and R. Kimmel. Group-Valued Regularization for Analysis of Articulated Motion. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science*. Vol. 7583. Springer Berlin Heidelberg, 2012, pp. 52–62. isbn: 978-3-642-33862-5. doi: 10.1007/978-3-642-33863-2_6 (cited on pp. 129, 150).

[128] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In. Vol. 23. (3), Aug. 2004, pp. 309–314. doi: 10.1145/1015706.1015720 (cited on pp. 10, 118).

[129] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. In *Physica D: Nonlinear Phenomena*, 60(1):259–268. Elsevier, 1992. doi: 10.1016/0167-2789(92)90242-F (cited on pp. 20, 21, 23).

[130] J. Santner, T. Pock, and H. Bischof. Interactive Multi-label Segmentation. In *Asian Conference on Computer Vision (ACCV). Lecture Notes in Computer Science*. Vol. 6492. Springer Berlin Heidelberg, 2011, pp. 397–410. isbn: 978-3-642-19314-9. doi: 10.1007/978-3-642-19315-6_31 (cited on pp. 11, 118, 125–128).

[131] S. Savarese, J. Winn, and A. Criminisi. Discriminative Object Class Models of Appearance and Shape by Correlatons. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2, 2006, pp. 2033–2040. doi: 10.1109/CVPR.2006.102 (cited on p. 67).

[132] O. Scherzer. Denoising with higher order derivatives of bounded variation and an application to parameter estimation. In *Computing*, 60(1):1–27. Springer-Verlag, 1998. doi: 10.1007/BF02684327 (cited on p. 21).

[133] J. A. Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. In *Proceedings of the national academy of sciences of the united states of america*, 93(4):1591–1595. National Academy of Sciences, 1996 (cited on p. 137).

[134] S. Setzer, G. Steidl, and T. Teuber. Infimal convolution regularizations with discrete l1-type functionals. In *Communications in Mathematical Sciences*, 9(3):797–827, 2011. doi: 10.4310/CMS.2011.v9.n3.a7 (cited on p. 21).

[135] C. E. Shannon. A Mathematical Theory of Communication. In *Sigmobile mobile computing and communications review*, 5(1):3–55. ACM, Jan. 2001. doi: 10.1145/584091.584093 (cited on p. 75).

[136] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo. An Interactive Approach to Semantic Modeling of Indoor Scenes with an RGBD Camera. In *ACM Transactions on Graphics (TOG)*, 31(6):1–11, Nov. 2012. doi: 10.1145/2366145.2366155 (cited on pp. 11, 118, 119).

[137] T. K. Shih, N. C. Tang, and J.-N. Hwang. Exemplar-Based Video Inpainting Without Ghost Shadow Artifacts by Maintaining Temporal Continuity. In *IEEE Transactions on Circuits and Systems for Video Technology*, 19(3):347–360, Mar. 2009. doi: 10.1109/TCSVT.2009.2013519 (cited on p. 136).

[138] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. In *International Journal of Computer Vision (IJCV)*, 81(1):2–23. Springer US, 2009. doi: 10.1007/s11263-007-0109-1 (cited on pp. 107, 111, 114, 115).

[139] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science*. Vol. 3951. Springer Berlin Heidelberg, 2006, pp. 1–15. isbn: 978-3-540-33832-1. doi: 10.1007/11744023_1 (cited on pp. 7, 9, 10, 71, 106).

[140] N. Silberman and R. Fergus. Indoor Scene Segmentation using a Structured Light Sensor. In *IEEE International Conference on Computer Vision (ICCV), Workshop on 3D Representation and Recognition (3dRR)*, Nov. 2011, pp. 601–608. doi: 10.1109/ICCVW.2011.6130298 (cited on pp. 10, 111, 118).

[141] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science.* Vol. 7576. Springer Berlin Heidelberg, 2012, pp. 746–760. isbn: 978-3-642-33714-7. doi: 10.1007/978-3-642-33715-4_54 (cited on pp. 111, 121, 125).

[142] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Of *Monographs on Statistics and Applied Probability.* Chapman and Hall/CRC, 1986. 176 pp. isbn: 9780412246203 (cited on pp. 121, 122).

[143] P. Soille. *Morphological Image Analysis. Principles and Applications.* Springer Berlin Heidelberg, 2nd ed., 2004. isbn: 978-3-642-07696-1. doi: 10.1007/978-3-662-05088-0 (cited on p. 72).

[144] M. Souiai, C. Nieuwenhuis, E. Strekalovskiy, and D. Cremers. Convex Optimization for Scene Understanding. In *IEEE International Conference on Computer Vision (ICCV), Workshop on Graphical Models for Scene Understanding (GMSU)*, Dec. 2013, pp. 9–14. doi: 10.1109/ICCVW.2013.131 (cited on pp. 8, 66, 67, 104).

[145] M. Souiai, E. Strekalovskiy, C. Nieuwenhuis, and D. Cremers. A Co-occurrence Prior for Continuous Multi-label Optimization. In *International Conference on Energy Minimization Methods for Computer Vision and Pattern Recognition (EMMCVPR). Lecture Notes in Computer Science.* Vol. 8081. Springer Berlin Heidelberg, 2013, pp. 209–222. isbn: 978-3-642-40394-1. doi: 10.1007/978-3-642-40395-8_16 (cited on pp. 8, 65, 66, 104).

[146] G. Steidl. Combined First and Second Order Variational Approaches for Image Processing. In *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 117(2):133–160. Springer Berlin Heidelberg, 2015. doi: 10.1365/s13291-015-0113-2 (cited on p. 21).

[147] E. Strekalovskiy and D. Cremers. Generalized ordering constraints for multilabel optimization. In *IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 2619–2626. doi: 10.1109/ICCV.2011.6126551 (cited on pp. 8, 67, 100, 101).

[148] E. Strekalovskiy, B. Goldlücke, and D. Cremers. Tight convex relaxations for vector-valued labeling problems. In *IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 2328–2335. doi: 10.1109/ICCV.2011.6126514 (cited on p. 81).

[149] E. Strekalovskiy, C. Nieuwenhuis, and D. Cremers. Nonmetric Priors for Continuous Multilabel Optimization. In *European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science.* Vol. 7578. Springer Berlin Heidelberg, 2012, pp. 208–221. isbn: 978-3-642-33785-7. doi:

10.1007/978-3-642-33786-4_16 (cited on pp. 8, 9, 66, 67, 89, 90, 92, 94, 95, 98, 99).

[150] J. Stühmer, P. Schröder, and D. Cremers. Tree Shape Priors with Connectivity Constraints Using Convex Relaxation on General Graphs. In *IEEE International Conference on Computer Vision (ICCV)*, Dec. 2013, pp. 2336–2343. doi: 10.1109/ICCV.2013.290 (cited on p. 7).

[151] S. Tari and M. Genctav. From a Modified Ambrosio-Tortorelli to a Randomized Part Hierarchy Tree. In *Scale Space and Variational Methods in Computer Vision (SSVM). Lecture Notes in Computer Science*. Vol. 6667. Springer Berlin Heidelberg, 2012, pp. 267–278. isbn: 978-3-642-24784-2. doi: 10.1007/978-3-642-24785-9_23 (cited on p. 45).

[152] S. Tari and M. Genctav. From a Non-Local Ambrosio-Tortorelli Phase Field to a Randomized Part Hierarchy Tree. In *Journal of Mathematical Imaging and Vision (JMIV)*, 49(1):69–86. Springer US, 2014. doi: 10.1007/s10851-013-0441-8 (cited on p. 34).

[153] S. Tari, J. Shah, and H. Pien. Extraction of Shape Skeletons from Grayscale Images. In *Computer Vision and Image Understanding*, 66(2):133–146. Elsevier Science Inc., May 1997. doi: 10.1006/cviu.1997.0612 (cited on pp. 34, 43, 45, 46).

[154] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape priors. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 3105–3112. doi: 10.1109/CVPR.2010.5540068 (cited on pp. 10, 118).

[155] A. Telea. An Image Inpainting Technique Based on the Fast Marching Method. In *Journal of Graphics Tools*, 9(1):23–34. Taylor & Francis, 2004. doi: 10.1080/10867651.2004.10487596 (cited on p. 137).

[156] A. N. Tikhonov. On the stability of inverse problems. In *USSR Academy of Sciences*. Vol. 39. (5), 1943, pp. 195–198 (cited on p. 22).

[157] E. Töppe, C. Nieuwenhuis, and D. Cremers. Relative Volume Constraints for Single View 3D Reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 177–184. doi: 10.1109/CVPR.2013.30 (cited on p. 65).

[158] E. Töppe, M. R. Oswald, D. Cremers, and C. Rother. Image-based 3D Modeling via Cheeger Sets. In *Asian Conference on Computer Vision (ACCV). Lecture Notes in Computer Science*. Vol. 6492. Springer Berlin Heidelberg, 2010, pp. 53–64. isbn: 978-3-642-19314-9. doi: 10.1007/978-3-642-19315-6_5 (cited on p. 65).

[159] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).* Vol. 2, June 2004, pp. 762–769. doi: `10.1109/CVPR.2004.1315241` (cited on p. 10).

[160] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. In *IEEE Transactions on Automatic Control*, 40(9):1528–1538, Sept. 1995. doi: `10.1109/9.412624` (cited on p. 137).

[161] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised semantic segmentation with a multi-image model. In *IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 643–650. doi: `10.1109/ICCV.2011.6126299` (cited on pp. 98, 99).

[162] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008, pp. 1–8. doi: `10.1109/CVPR.2008.4587440` (cited on p. 7).

[163] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In *IEEE International Conference on Computer Vision (ICCV)*, Sept. 2009, pp. 755–762. doi: `10.1109/ICCV.2009.5459287` (cited on pp. 10, 11, 118).

[164] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).* Vol. 1, 2001, pp. 511–518. doi: `10.1109/CVPR.2001.990517` (cited on pp. 9, 106, 107).

[165] J. Wang, K. Lu, D. Pan, N. He, and B.-k. Bao. Robust object removal with an exemplar-based image inpainting approach. In *Neurocomputing*, 123:150–155, 2014. doi: `http://dx.doi.org/10.1016/j.neucom.2013.06.022` (cited on p. 138).

[166] J. Wang. Discriminative Gaussian Mixtures for Interactive Image Segmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Vol. 1, Apr. 2007, pp. 601–604. doi: `10.1109/ICASSP.2007.365979` (cited on pp. 11, 118).

[167] L. Wang, J. Shi, G. Song, and I.-f. Shang. Object Detection Combining Recognition and Segmentation. In *Asian Conference on Computer Vision (ACCV). Lecture Notes in Computer Science.* Vol. 4843. Springer Berlin Heidelberg, 2007, pp. 189–199. isbn: 978-3-540-76385-7. doi: `10.1007/978-3-540-76386-4_17` (cited on pp. 68, 82, 95).

[168] M. Werlberger. Convex Approaches for High Performance Video Processing. PhD thesis. Graz, Austria: Institute for Computer Graphics and Vision, Graz University of Technology, June 2012 (cited on p. 133).

[169] Y. Wexler, E. Shechtman, and M. Irani. Space-Time Completion of Video. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(3):463–476, Mar. 2007. doi: 10.1109/TPAMI.2007.60 (cited on pp. 134, 136, 140, 142).

[170] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1, June 2004, pp. 120–127. doi: 10.1109/CVPR.2004.1315022 (cited on pp. 134, 136).

[171] M. Willem. *Functional Analysis. Fundamentals and Applications*. Of *Cornerstones*. Birkhäuser Basel, 2013. isbn: 978-1-4614-7003-8. doi: 10.1007/978-1-4614-7004-5 (cited on p. 21).

[172] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 702–709. doi: 10.1109/CVPR.2012.6247739 (cited on pp. 66, 104).

[173] J. Yuan and Y. Boykov. TV-Based Multi-Label Image Segmentation with Label Cost Prior. In *British Machine Vision Conference (BMVC)*. BMVA Press, 2010, pp. 1–12. isbn: 1-901725-40-5. doi: 10.5244/C.24.101 (cited on p. 7).

[174] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast Global Labeling for Real-Time Stereo Using Multiple Plane Sweeps. In *Vision, Modeling and Visualization Workshop (VMV)*, Oct. 2008, pp. 243–252 (cited on pp. 69, 70).

[175] S. C. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(9):884–900, Sept. 1996. doi: 10.1109/34.537343 (cited on p. 7).