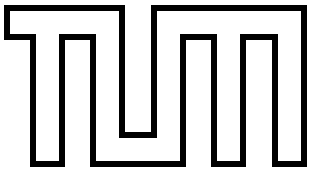


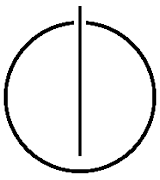
FAKULTÄT FÜR INFORMATIK

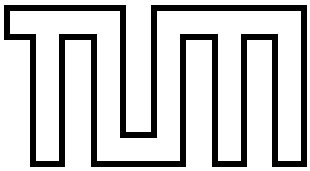
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Interactive Image Segmentation Using Space-Varying Color and Texture Distributions

Wadim Kehl





FAKULTÄT FÜR INFORMATIK

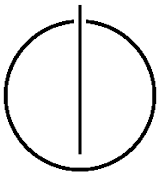
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Interactive Image Segmentation Using Space-Varying Color and Texture Distributions

Interaktive Bildsegmentierung mittels ortsvariierender Farb- und Texturverteilungen

Author:	Wadim Kehl
Supervisor:	Prof. Dr. Daniel Cremers
Advisor:	Dr. Claudia Nieuwenhuis
Date:	January 15th, 2013



I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Ich versichere, dass ich diese Masterarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. Januar 2013 Wadim Kehl

Acknowledgments

First and foremost, I want to give many thanks to Daniel Cremers and Claudia Nieuwenhuis for supervision and many pieces of advice during the creation of this work.

Secondly, I thank my family and friends for mental support.

I also want to give credit to Jan Kremer for proofreading the draft version of the thesis.

And lastly, I thank Mohamed Souiai for keeping the whole coffee machine system up and running through the troublesome times.

Abstract

In the domain of computer vision image segmentation presents itself as a diverse field that has been tackled by research for decades. In this work we look into the area of supervised multi-label segmentation, meaning an input-driven partitioning of an image into multiple disjoint segments. We also strife for an interactive methodology which is able to run and provide visual response to the user in real-time.

This work exploits recent developments in variational segmentation methods together with a statistical approach to estimate probability densities from user input and create results which compete with the state-of-the-art. Building on a model that uses spatially-varying color distributions from Parzen windows we show how to extend the model to incorporate textural information to further increase the accuracy of results. Another aspect concerns the supervised transformation of given information to raise its discriminative power. We also show how anisotropic information can improve the spatial estimation and how the variance of color and texture can be taken as a measure for automatically identifying good kernel parameters.

The proposed texture-enhanced model, implemented on a GPU, is tried on different datasets where it defeats current research and presents itself as state-of-the-art while still offering real-time user interaction. The evaluation also shows that anisotropic information helps to segment elongated objects more reliably and that automatic parameter estimation frees the user from tediously finding hand-picked settings while providing results on the same visual level.

Contents

Outline of the Thesis

Part I: Introduction and Theory

CHAPTER 1: INTRODUCTION

This chapter gives an introduction into the domain of image segmentation and a summary about important developments in this field of research. Secondly, the given interactive setting and the basic idea of the thesis are sketched briefly prior to its full explanation in the second part.

CHAPTER 2: THEORY

At first, the theoretical foundation for variational multi-label segmentation is presented. After that, an explanation of Wavelet theory as well as subspace methods and density estimation lays the groundwork for the chapters to follow.

Part II: The proposed Model

CHAPTER 3: OVERVIEW OF THE PROPOSED APPROACH

This chapter deals with the general setting of interactive image segmentation and how distributions of space, color and texture information, inferred from user input, can drive the segmentation process.

CHAPTER 4: MODEL AND OPTIMIZATION

Building on the theory this chapter gives rise to a model that incorporates the aforementioned pieces of information. The extraction of information from the image and the optimization of the energy receive a detailed focus here.

Part III: Evaluation and Outlook

CHAPTER 5: EVALUATION

An evaluation of the proposed model on different datasets as well as an analysis of specific aspects are given in this chapter together with a discussion of the results.

CHAPTER 6: SUMMARY AND OUTLOOK

The last chapter gives a summary of this work and provides an outlook on possible future work.

Part I.

Introduction and Theory

1. Introduction

This chapter gives an introduction into the interesting area of image segmentation and how this problem has been tackled for years with different approaches in research. Subsequently, the notion of interactive image segmentation gets explained and motivated.

1.1. Image Segmentation

The problem of image segmentation is one of the most studied topics in the field of computer vision. Mathematically, it corresponds to the problem of having an image $\Omega \subset \mathbb{R}^2$ and partitioning it into k disjoint sets $E_1, ..., E_k$:

$$\bigcup_{l=1}^{k} E_l = \Omega \quad , \quad \forall i \neq j : E_i \cap E_j = \emptyset. \tag{1.1}$$

Usually, the literature divides the problem of partitioning into the binary case ($k = 2$) and the multi-label case ($k > 2$). The binary case has proven to be mathematically easier to solve due to its complementary nature and for many models a global optimum can be determined [Boykov et al., 2001, Taskar et al., 2004, Chan and Esedoglu, 2006]. In the multi-label case the (discrete and continuous) models become NP-hard in general and approximate solutions can sometimes be found by different relaxation techniques that render the problems tractable [Boykov et al., 2001, Taskar et al., 2004, Rother and Kolmogorov, 2007, Chambolle et al., 2011]. A special case is shown in the work [Ishikawa, 2003] where a global optimum can be found in polynomial time when having a linearly ordered label set.

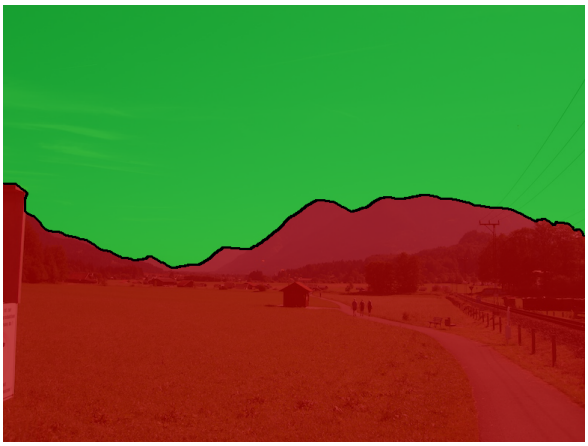
An exemplary segmentation can be seen in Figure 1.1. It is not inherently clear which parts of the image should reside in the same set. In the given example, the binary segmentation separated sky from ground whereas in the multi-label case the image was divided up into sky, mountains, ground and the wooden hut in the front.

Apparently, properties are needed to drive the segmentation towards "meaningful" partitions of the image. Such properties could include (but are certainly not limited to) intensity, color, texture and spatial or temporal relationships. The second aspect of the topic is the actual segmentation methodology that takes the aforementioned properties and yields a partitioning. The first part of the theory deals with the two more sophisticated approaches, namely graphical and variational models.

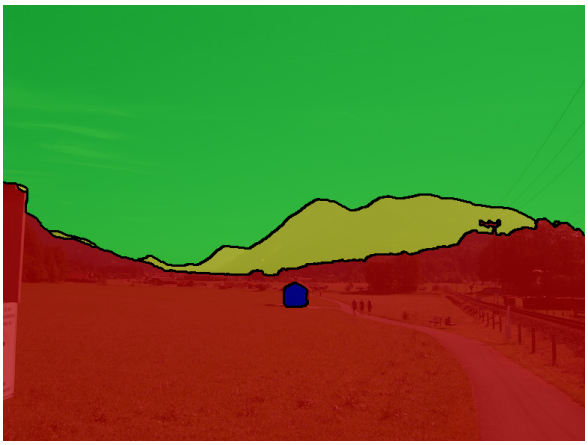
1.2. Interactive Image Segmentation

This work also employs the idea of interactive image segmentation, meaning a supervised, semi-automatic approach to drive the segmentation process. By providing user input, the

(a) The original image



(b) Binary labeling: sky, ground



(c) Multiple labels: sky, mountains, ground, hut

Figure 1.1.: Segmenting the image (a) into $k = 2$ disjoint sets (b) and $k = 4$ disjoint sets (c).

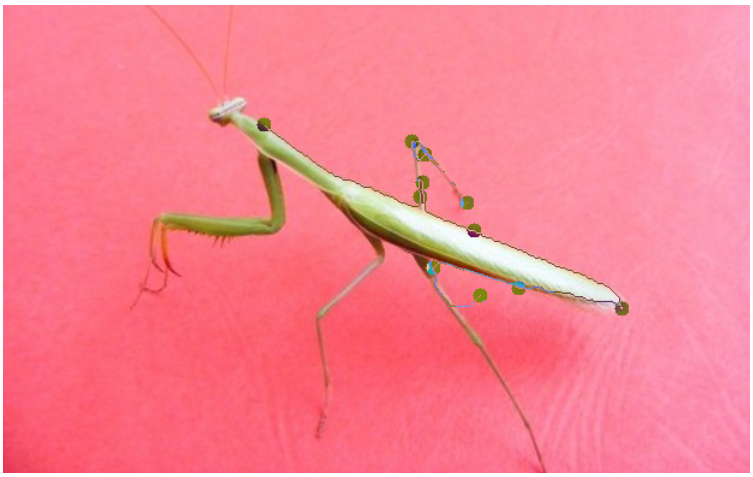
segmentation can be controlled to produce (better) results by exploiting the user's knowledge of the image context. The important aspect is to provide a way for the input to be both intuitive for the user and rich in information for the segmentation model. User input has been introduced in the literature in multiple ways:

One famous approach is the "Intelligent Scissor" [Mortensen, 1998] where the user moves the mouse along the approximate contour of the object. On-the-fly, the method then tries to find a minimal cost path through the gradient magnitudes that is given by the user input "seeds". This approach works well for regions that are easy to discern visually but fails for heavily textured or completely homogeneous region borders since the minimum cost path is not unique in that case. Furthermore, it only allows the separation of foreground and background. Mostly though, user input is given by rectangles [Rother and Kolmogorov, 2004], brush strokes or so-called "scribbles" [Boykov et al., 2001, Grady, 2006, Unger et al., 2008, Nieuwenhuis et al., 2011] or both [Santner et al., 2009]. The advantage of having an intuitive way of user input is the ability to manually improve a given segmentation in an iterative way until a satisfactory result is achieved. See Figure 1.2 for details. It shows that the scissors are convenient to use since they are interactive in real-time, meaning that the contour can be seen while moving the mouse. It becomes problematic though for small object parts, e.g. the thin legs, which are hard for the user to include with the scissors. The second image shows the user rectangle. The problem with this approach is that the area covered by that shape is usually bigger than the object, leading to possible problems in later stages of the segmentation. Nonetheless, it can give a good initial hint on the position and the intensities of the object to segment. The third picture shows user scribble input that gives the user the needed steerable granularity in the spatial domain, e.g. by providing different brush sizes.

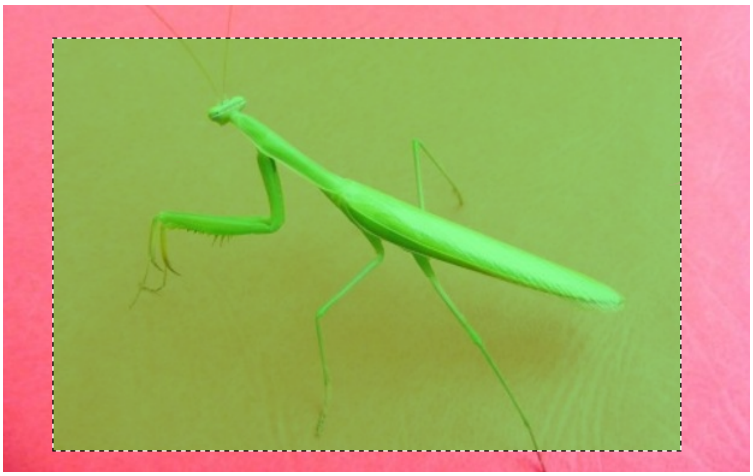
In this thesis user scribbles are the choice of input since they will form an integral part of the computational model. Formally, the set of scribbles S can be written as

$$S := \{S_1, ..., S_k\} \quad , \quad S_i := \{\mathbf{x}_j^i, j = 1, ..., m_i\} \tag{1.2}$$

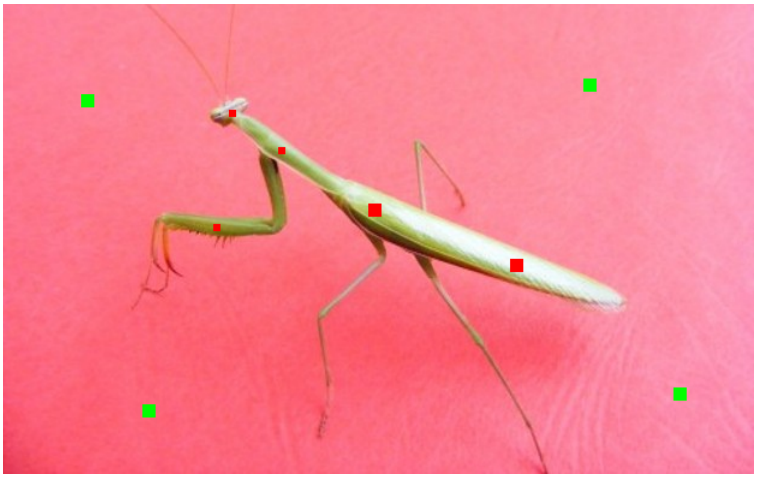
where i denotes the label and j the index of the scribble for the given label. The idea of this work will be to compute likelihoods for a partitioning given the user scribbles, i.e. compute $P(E_1, ..., E_k|S)$ and use the likelihoods to lead the segmentation process. Details and a full explanation of the proposed model will be clarified in the third chapter of the thesis. Using scribbles also has the advantage of easily adding further scribbles to influence the likelihoods. See the following Figure 1.3 for a visual example.



(a) Intelligent Scissor

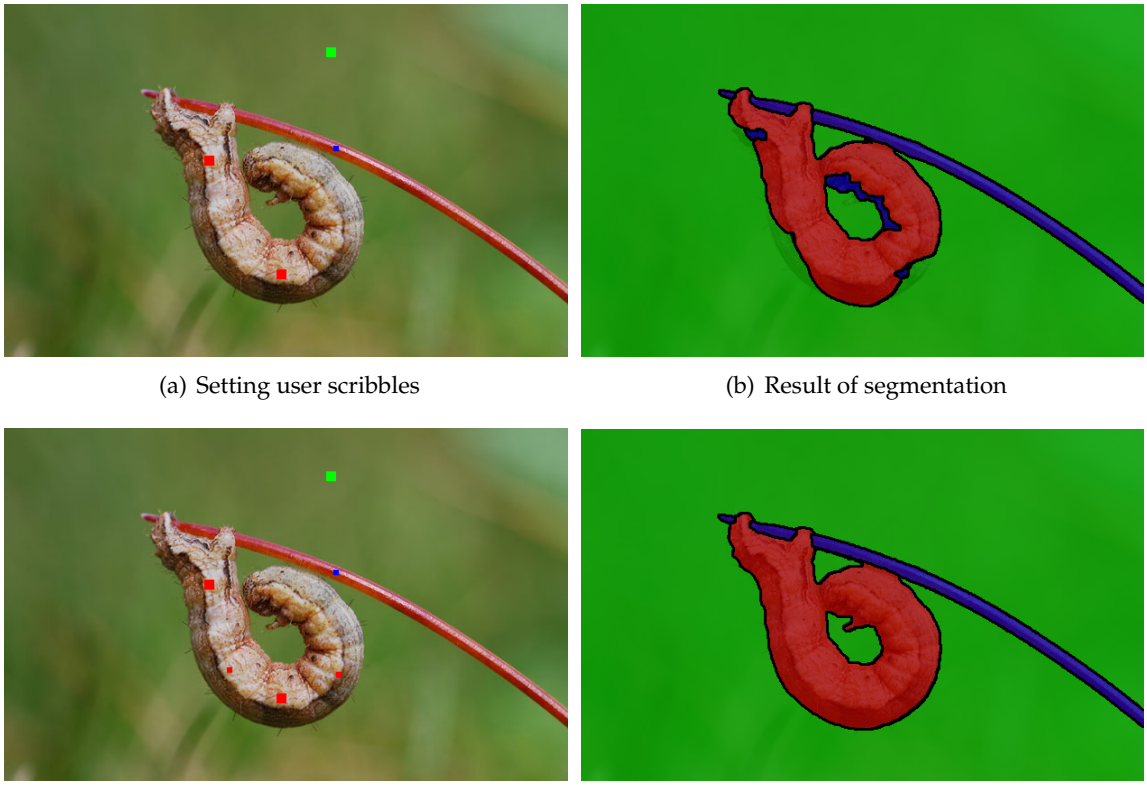


(b) User rectangle



(c) User scribbles

Figure 1.2.: Three different methods for providing user input. For Figure (a) the "Intelligent Scissor" tool from GIMP was used. The image itself is taken from the IcgBench benchmark [Santner et al., 2010].



(a) Setting user scribbles

(b) Result of segmentation

(c) Improving scribble input

(d) Better segmentation

Figure 1.3.: Using the initial user scribble input in (a) leads to moderate segmentation results (b). Intuitively improving the input by adding some further red scribbles to the badly segmented areas (c) leads to a more satisfying result (d).

2. Theory

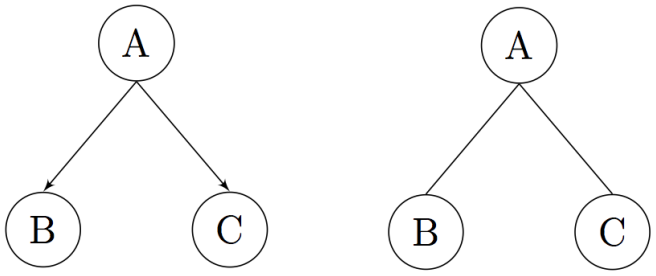
This chapter introduces the reader into the field of variational segmentation. After that an introduction into signal and wavelet theory gives the reader sufficient background for the details to come. A short outline of subspace methods, and especially linear discriminant analysis, is given and a brief introduction into kernel density estimation finally concludes the chapter.

2.1. Graphical models

Graphical models are used to represent joint and conditional distributions of multiple random variables where nodes correspond to the variables and edges model dependencies. In the literature two kinds of (probabilistic) graphs are common: (directed) Bayes networks and (undirected) Markov networks. The depiction in Figure 2.1 shows the same joint distributions $P(A, B, C)$ but with different dependency assumptions. In this work the explanation will be confined to Markov networks since they are more suitable to the problem at hand.

A Markov network (also Markov Random Field, MRF) is defined as a tuple $G = (V, E)$ with random variables $X = (X_v)_{v \in V}$. By employing certain assumptions towards conditional independence and the density the probability of a joint state of X can be factorized over the cliques x_c of the graph:

$$P(X) = \prod_{c \in cl(G)} \phi_c(x_c)$$



(a) Directed Bayes network (b) Undirected Markov network

Figure 2.1.: Two graphical models describing the same joint distribution but different dependency assumptions.

with $cl(G)$ the set of cliques and ϕ_c non-negative potential functions. Usually, the potentials are separated into two classes: unary potentials that are evaluated for singleton cliques and binary potentials for pairwise interactions between neighboring vertices. Thus, we have:

$$P(X) = \prod_v \phi_v(X_v) \cdot \prod_{(v,w)} \phi_{(v,w)}(X_v, X_w)$$

For a detailed introduction one can revert to [Koller and Friedman, 2009].

2.1.1. Gibbs model

In the field of computer vision tasks are often formulated as energy minimization problems. Hence, the MRF is transformed into the Gibbs model of the form

$$P(X) = \frac{1}{Z}\exp(-E(x)) \text{ with } E(x) = \sum_v \phi_v(X_v) + \sum_{(v,w)} \phi_{(v,w)}(X_v, X_w) \tag{2.1}$$

and then optimized by minimizing the negative log-likelihood

$$\max P(X) = \max \frac{1}{Z}\exp(-E(X)) = \max \log \exp(-E(X)) = \min E(X)$$

Here, the analogy to image segmentation is to treat every pixel as a vertex in the graph and to base the unary potentials ϕ_v on said image properties. This is often referred to as the data term in the literature. The pairwise potentials $\phi_{(v,w)}$ are typically used as a smoothing term to penalize incoherent regions, with the Potts model [Wu, 1982] being one of the most famous:

$$\phi_{(v,w)} = \begin{cases} 0 & \text{if } X_v = X_w \\ \lambda & \text{else} \end{cases} .$$

This model basically penalizes configurations where cliques do not agree on a common labeling by adding a constant λ to the energy in that case.

2.1.2. Training and inference

Usually, these models are applied to problems that are tackled in a supervised manner. Therefore a training set is provided on which the model can be conditioned upon by learning suited potential functions. These potential functions can have arbitrary formulations and are taken from different function families although one generally reverts to weighted linear functions to make the computation feasible. Learning is then usually done by solving Convex Programs or employing gradient-based methods [Ratliff et al., 2003, Taskar et al., 2004].

Given a model with potential functions and an unlabeled data set the minimizer configuration needs to be inferred. The literature has shown that these energies can be minimized via various well-known graph-cut methods [Boykov et al., 2001, Boykov and Jolly, 2001, Komodakis and Tziritas, 2007] as long as they are submodular. There also exist other methods, for example message passing algorithms like (loopy) Belief Propagation, that use exact and approximate iterative schemes which are not guaranteed to converge but do so in many cases [Pearl, 1988, Kschischang et al., 2001, Jordan and Weiss, 2002].

2.2. Variational approach

In opposite to the graphical models that exploit the spatially discrete nature of the image variational methods strive to formulate the problem in a continuous manner. Thus, the energy corresponds to a functional and the minimization yields a function that represents, implicitly or explicitly, the partitioning of the image. Essentially, the models that are used in the image domain are based on energies that consist of a data term E_{data} that upholds the solution's fidelity to a given input and a regularizer term E_{reg} that forces the solution to be smooth in some sense:

$$E(u) = E_{data}(u, I) + \lambda \cdot E_{reg}(u) \tag{2.2}$$

together with a weighting parameter λ that balances both terms. Note that the discrete model from the last section also employs this idea by having the unary potentials for the fidelity and the pairwise potentials as some form of regularization. The two following models presented here will give an introduction into the ideas and the methodology to come up with a Weighted-TV segmentation model.

2.2.1. Mumford-Shah functional

One of the most famous variational models has been the Mumford-Shah functional from the late 80s [Mumford and Shah, 1989] where the minimization of the presented energy is a piecewise smooth approximation of the input image. The model is presented as the following functional:

$$E(u, C) = \int_{\Omega} (I - u)^2 dx + \lambda \int_{\Omega \setminus C} |\nabla u|^2 dx + \upsilon |C| \tag{2.3}$$

with $u : \Omega \to \mathbb{R}$ being the approximation and $C \subset \Omega$ being the one-dimensional discontinuity set. The first term increases when u and I do not match and therefore enforces a good approximation. The second term with a weighting factor λ tries to set u smooth everywhere except for the set C. The last term, again with weighting υ, assures that the length $|C|$ of the discontinuity is minimal.

The problem with this formulation is that the variable of interest C is part of the energy itself and in the original work no numerical scheme is given to compute a minimizer. In the literature multiple approaches have been presented to solve Mumford-Shah related models and this thesis also employs a segmentation model partially based on the Mumford-Shah functional. See two segmentation examples of a color-based model in Figure 2.2.

2.2.2. The ROF model

To introduce the notion of Total Variation in the domain of image processing one can look at the famous denoising ROF model [Rudin et al., 1992]:

$$\min_{u \in BV(\Omega)} \frac{\lambda}{2} ||u - g||_2^2 + \int_{\Omega} |Du| \tag{2.4}$$

(a) Example image (b) Result with $k = 10$ and $\lambda = 6$

(c) Example image (d) Result with $k = 10$ and $\lambda = 4$

Figure 2.2.: Using the Matlab code supplied by [Chambolle and Pock, 2011]. After finding k means in the color space, the segmentation yields a Mumford-Shah piecewise smooth solution which is clearly seen in the results.

with $g \in L^1(\Omega)$ being a noisy image and $u \in L^1(\Omega)$ the sought denoised version of g with bounded variation. The variational energy consists of a data term that strives to minimize the Euclidean error $||u - g||_2$, basically resembling a least-squares approach over the intensities, and a regularizer $\int_\Omega |Du| = TV(u)$ that measures the so-called total variation of the function (here, Du denotes the distributional derivative). Generally, the total variation of a function $u \in L^1(\Omega)$ admits a dual representation, defined as (see, for example [Giusti, 1984])

$$TV(u) := \sup \left\{ -\int_\Omega u(x) \cdot \operatorname{div} \xi dx \;\middle|\; \xi \in C_c^1(\Omega, \mathbb{R}^n), ||\xi||_{L^\infty(\Omega)} \leq 1 \right\} \tag{2.5}$$

with $C_c^1(\Omega, \mathbb{R}^n)$ being the dual space of continuously differentiable vector functions of compact support in Ω and $|| \cdot ||_{L^\infty(\Omega)}$ the essential supremum norm. Further note that $u \in BV(\Omega)$ is constrained to be of bounded variation, defined as

$$BV(\Omega) := \left\{ u \in L^1(\Omega) \;\middle|\; TV(u) < +\infty \right\} \tag{2.6}$$

and meaning that the total variation is finite. If the function u is differentiable and Ω is a bounded open set, then the total variation can also be written (using Gauss' theorem) as

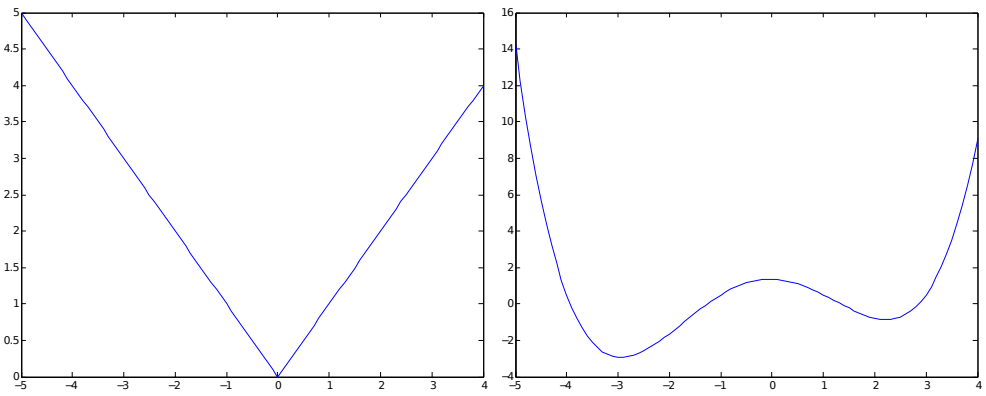
$$\begin{aligned} \int_\Omega |Du| = \int_\Omega -u \cdot \operatorname{div} \xi = \int_\Omega \nabla u \cdot \xi &\leq ||\xi||_\infty \int_\Omega |\nabla u| \\ \text{yielding equality with } \hat{\xi} := \frac{\nabla u}{|\nabla u|} &= \int_\Omega \nabla u \cdot \hat{\xi} = \int_\Omega |\nabla u| \end{aligned}$$

and since $\hat{\xi}$ can be approximated by a sequence $(\xi)_n \subset C_c^1(\Omega)$ it holds that

$$\int_\Omega -u \cdot \operatorname{div} (\xi)_n = \int_\Omega \nabla u \cdot (\xi)_n \longrightarrow \int_\Omega \nabla u \cdot \hat{\xi} = \int_\Omega |\nabla u|. \tag{2.7}$$

The ROF model is also known as the $TV\text{-}L^2$ model and it has been practically shown that the advantage of the TV-regularizer is its discontinuity-preserving (read: edge-preserving) nature and therefore very suited for image processing tasks. One problem of the ROF model is the non-differentiable nature of the TV-regularizer which is overcome by introducing its differentiable dual representation, in e.g. [Chan et al., 1999].

Another general advantage of the total variation is its convex nature. A function f is convex iff its epigraph $epi(f) := \{ (x, y) \mid f(x) \leq y \}$ is a convex set. An epigraph can be regarded as the set of points that lie on or above the graph of f. Assuming that there exists a minimizer for a given convex energy it can be easily shown that this minimizer is indeed globally optimal. See Figure 2.3 for a visualization of the matter. Convex problems also have the advantage that found solutions (if they exist) are independent of the initialization and that there exist well-researched methods to solve them.



(a) Convex energy f_1 (b) Non-convex energy f_2

Figure 2.3.: Examples of a convex (a) and a non-convex (b) function. It is clear that f_2 has local minima and therefore $epi(f_2)$ cannot yield a convex set whereas (a) has only one global minimum. Also note from f_1 that convex functions are not necessarily differentiable.

2.3. TV Segmentation

Building on equations 2.1 and 2.3 a generic Potts model for the continuous case will be used that has the following form:

$$\min_{E_1,...,E_k} \sum_{i=1}^{k} \int_{E_i} f_i(x)dx + \frac{\lambda}{2} \sum_{i=1}^{k} Per(E_i, \Omega) \tag{2.8}$$

where $f_i : \Omega \to \mathbb{R}_+$ are (w.l.o.g.) non-negative potential functions defined for every label. The second term, weighted with λ, is half the sum of the perimeters of the sets $E_1, ..., E_k$ and corresponds to the total length of the partition interface $\bigcup_{i<j} \partial E_i \cap \partial E_j$ (since otherwise every perimeter would be counted twice). Thus, we want to determine a partition of k disjoint sets that yields the lowest energy and ensures a minimal interface at the same time.

To bring the energy into a computationally tractable form the usual way in literature is to represent a partition by characteristic functions $u_1, ..., u_k : \Omega \to \{0, 1\}$ with

$$u_i(x) = \begin{cases} 1 & \text{if } x \in E_i \\ 0 & \text{else} \end{cases} \qquad \text{satisfying } \sum_{i=1}^{k} u_i(x) = 1 \text{ a.e. } x \in \Omega$$

so that the first term becomes

$$\sum_{i=1}^{k} \int_{E_i} f_i(x)dx = \sum_{i=1}^{k} \int_{\Omega} u_i(x) f_i(x)dx$$

and is therefore not depending on such an explicit representation of the sets. See Figure 2.4 for a visualization of the indicators. Furthermore, if the characteristic functions $u_i \in BV(\Omega)$ of measurable sets $E_i \subset \Omega$ are scalar-valued functions of bounded variation (also called Caccioppoli sets) the co-area formula ([Fleming and Rishel, 1960]) states that the perimeter equals the total variation:

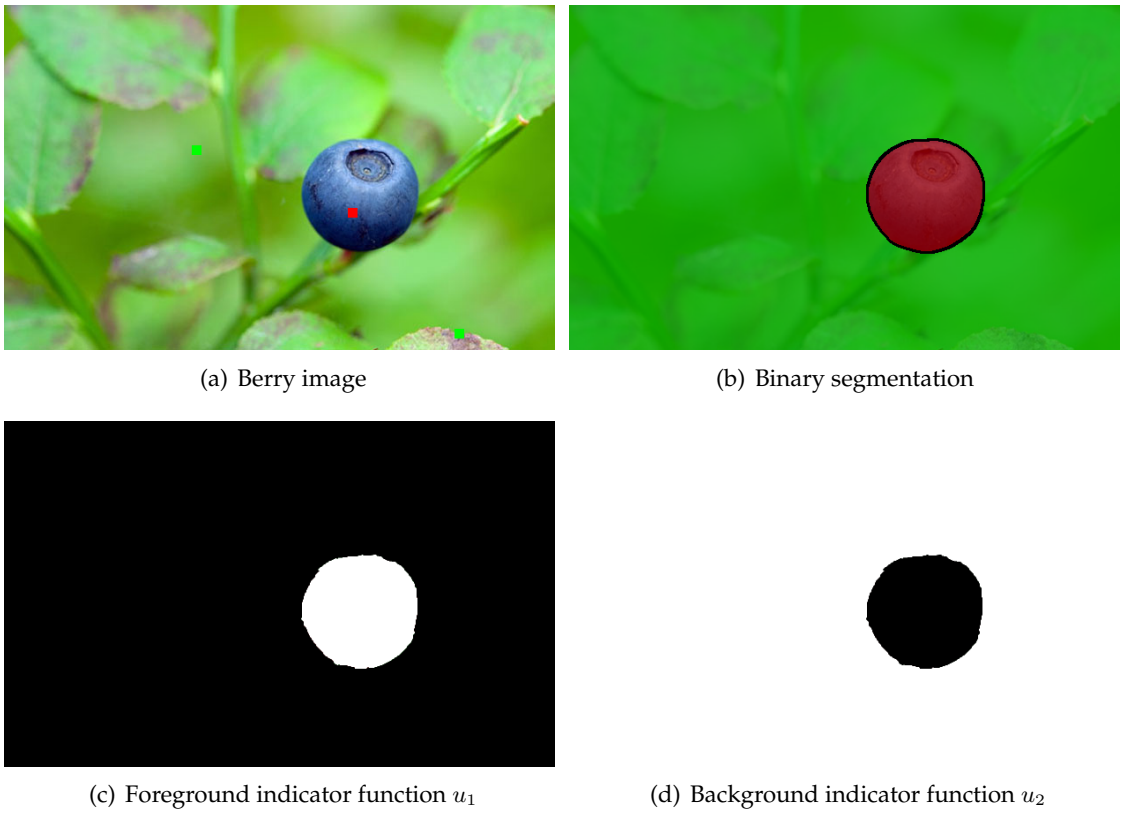
$$Per(E_i, \Omega) = Per(u_i, \Omega) = TV(u_i) = \int_{\Omega} |Du_i|dx.$$

Thus, we arrive at an intermediate minimization problem of the following form:

$$\min_{\mathbf{u} \in \mathcal{B}} \sum_{i=1}^{k} \int_{\Omega} u_i(x) f_i(x)dx + \frac{\lambda}{2} \sum_{i=1}^{k} \int_{\Omega} g(x)|Du_i|dx \tag{2.9}$$

$$\mathcal{B} := \left\{ (u_1, ..., u_k) \in BV(\Omega, \{0, 1\})^k \;\middle|\; \sum_i u_i(x) = 1 \text{ a.e. } x \in \Omega \right\}$$

together with a coherency constraint on the indicator functions u_i. The minimizer $\mathbf{u} = (u_1, ..., u_k)$ now lies in $\mathcal{B}$ which is the k-dimensional space of binary functions with bounded variation and fulfills the point-wise characteristic property, meaning that at every position x only one u_i is allowed to be non-zero. Also note that to receive a so-called Weighted-TV model we introduced a further space-depending function $g(x)$ into the energy that will express how much influence the regularizer should have at a specific position, thus a



(a) Berry image (b) Binary segmentation

(c) Foreground indicator function u_1 (d) Background indicator function u_2

Figure 2.4.: An image (a) after a segmentation with two classes (b). The two binary indicators for foreground (c) and background (d) are either black=0 or white=1.

weighted total variation. If the value of $g(x)$ is low or even zero then the minimization is supposed to merely take the data term into consideration and vice versa.

The optimization of the energy is still hard since it is neither convex nor differentiable. To convexify the problem one needs to get rid of the binary (non-convex) nature of the indicator functions. Therefore, the thesis follows the procedure from [Chambolle et al., 2008], i.e. to relax the binary functions by letting them map into the whole interval $u_i : \Omega \to [0, 1]$. With this, the indicators change from hard to soft assignments where one position x can have multiple non-zero entries although the coherency constraint still holds in that case, i.e. the sum has to yield 1. To remedy the non-differentiable nature of the total variation we replace it with its differentiable dual representation (2.5): $\int_\Omega |Du| = \sup_{\xi \in \mathcal{K}} \int_\Omega -\text{div } \xi \cdot u$.

The final convex and differentiable optimization problem then reads:

$$\min_{\mathbf{u} \in \mathcal{S}} \sup_{\xi \in \mathcal{K}} \sum_{i=1}^{k} \int_\Omega u_i(x) f_i(x)\, dx - \lambda \sum_{i=1}^{k} \int_\Omega \text{div } \xi_i(x) \cdot u_i(x)\, dx \tag{2.10}$$

with minimizing over the set $\mathcal{S}$ of BV functions moving in the k-dimensional simplex

$$\mathcal{S} := \left\{ \mathbf{u} = (u_1, ..., u_k) \in BV(\Omega, [0,1])^k \;\middle|\; \sum_i u_i(x) = 1 \text{ a.e. } x \in \Omega \right\}$$

and the dual variable ξ is in the convex set $\mathcal{K}$. The exact definition of the convex dual set varies with the authors. In [Lellmann et al., 2008, Zach et al., 2008] the authors use variations of a straight-forward formulation that arises from the TV definition of the regularizer. They enforce the dual variable ξ to stay inside a norm boundary, e.g. $(\sum_i ||\xi_i||^2)^{\frac{1}{2}} \leq 1$, that limits the amount of flow to happen at every position. Since we also employ a weighted model that has a space-dependent weighting function g, we can rewrite the dual space:

$$\mathcal{K} := \left\{ \xi = (\xi_1, ..., \xi_k) : \Omega \to \mathbb{R}^{2\times k} \;\middle|\; \sqrt{\sum_i ||\xi_i(x)||^2} \leq \frac{g(x)}{2} \text{ a.e. } x \in \Omega \right\}.$$

Another advanced approach is used in the work [Chambolle et al., 2011]. The authors introduce the notion of a paired calibration of the dual variables' components that represents a local convex envelope of the energy:

$$\mathcal{K}_C := \left\{ \xi = (\xi_1, ..., \xi_k) : \Omega \to \mathbb{R}^{2\times k} \;\middle|\; \left| \sum_{i_1 \leq i \leq i_2} \xi_i(x) \right| \leq \frac{g(x)}{2} \text{ a.e. } x \in \Omega \right\}.$$

They have proven that this model has a tighter bound for $k > 2$ in comparison to the two other publications meaning that the found minimizer is closer to the original global optimum. While this dual space creates tighter solutions the projection of a variable onto $\mathcal{K}_C$ is computationally cumbersome, since it involves the projection onto multiple convex sets. The projection onto $\mathcal{K}$ is very fast because it consists of point-wise truncation operations.

Since the relaxed energy in 2.10 is only an approximation to the original problem it is not guaranteed to find the exact solution. The literature has shown that in the binary case ($k = 2$) the thresholded solution of the relaxed version does indeed yield a global minimizer to the original problem, independent of the chosen threshold. In the multi-label case this does not longer hold for any binarized solution. The optimization of the energy will be introduced in the next chapter together with an explanation of saddle point problems and primal-dual formalism.

2.4. Wavelet theory

In the domain of signal analysis and processing wavelets have become a very important tool. The basic idea is to transform an arbitrary signal into a representation of basis functions and associated coefficients. In contrast to Fourier analysis however, wavelet bases are not confined to trigonometric functions but can be chosen from a wide variety of function families. This section of the thesis will briefly show the relation between Fourier and wavelet transformation together with a further explanation of the so-called wavelet families and of the discrete version of the transform. For further reference, see [Mallat, 2006].

2.4.1. Fourier and Wavelet analysis

To analyze signals one strives to decompose the signal into meaningful components. One method is the famous Fourier transform that decomposes any integrable signal f into an infinite sum of sinusoidal waves $e^{i\omega t}$:

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{f}(\omega)\, e^{i\omega t} d\omega \quad , \quad \hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)\, e^{-i\omega t} dt$$

with the coefficients $\hat{f}(\omega)$ being the Fourier transform for frequency ω. These coefficients can be interpreted as amplitudes since they represent the strength of a particular frequency in that signal over the whole time interval. Since the transform integrates over the whole frequency space, one can reconstruct the original signal by computing the inverse transformation.

One big disadvantage is the infinite support of the transformation in the time domain which makes it not suited for transient signals. If there is a distinct pattern in a non-stationary signal the Fourier transform will not tell at which position it is situated because of the lack of time locality. To compensate for that problem a short-time Fourier transform (also called Gabor transform) of the following form can be used:

$$\hat{f}(\omega, \tau) = \int_{-\infty}^{+\infty} f(t) g(t-\tau)\, e^{-i\omega t} dt$$

with an additional window function g that limits the view of the transformation. While this transformation gives a time locality it is very dependent on the choice of the function g which can lead to spectral leakage or unwanted smoothing.

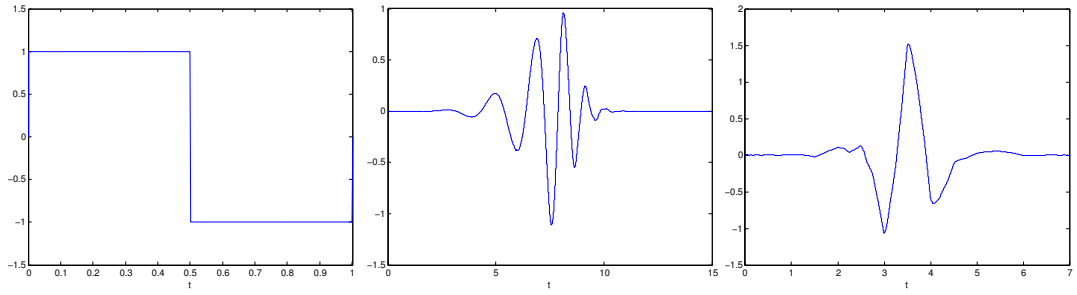
The wavelet transformation is similar to the Gabor transformation. Given a (mother) wavelet $\Psi(t) \in L^2(\mathbb{R})$ that fulfills

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0$$

the transformation at u with scale s is

$$\tilde{f}(u, s) = f \star \frac{1}{\sqrt{s}} \Psi^*(\frac{-u}{s}) = \int_{-\infty}^{+\infty} f(t)\, \psi^*_{u,s}(t) dt.$$

Again, the transformed version uses coefficients and a basis. This basis consists of the scaled and translated wavelet function Ψ or the complex conjugate of its children $\psi^*_{u,s}$. Note that the transformation can be written as a convolution because of its linear nature and that there is a scale s associated with the wavelets that gives rise to a multi-scale analysis. In Figure 2.5 one can see wavelet functions that are used in many applications. The big advantage of wavelets is the ability to create custom bases that suit one's needs and that multidimensional analysis is easy to accomplish for separable wavelets $\Psi(x, y) = \Psi(x)\Psi(y)$.



(a) Haar (b) Daubechies (c) Symlet

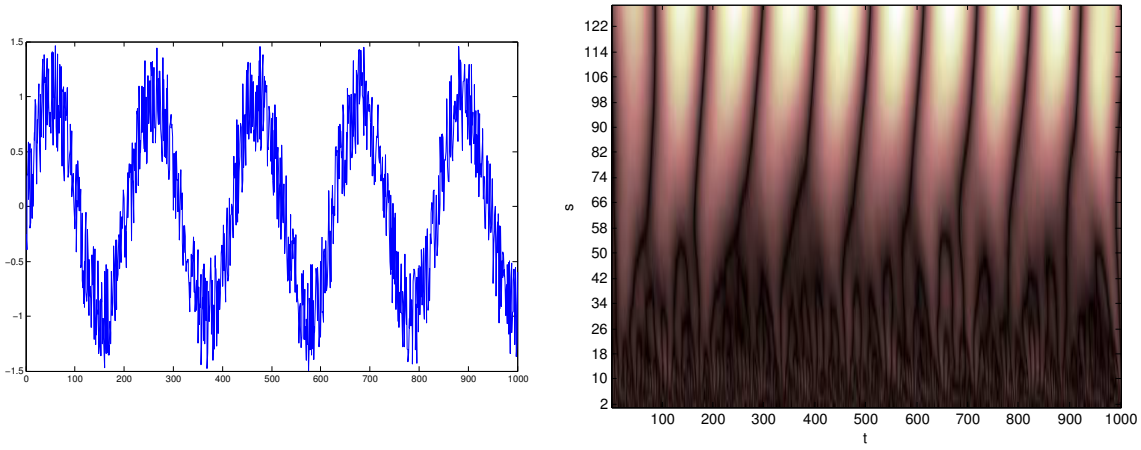
Figure 2.5.: Some typical wavelets commonly used in signal analysis.

2.4.2. Mother wavelets and children

In general, one speaks of a mother wavelet Ψ and its spawn children wavelets $\psi_{(u,s)}$ that are defined as

$$\psi_{(u,s)}(t) = \frac{1}{\sqrt{s}}\Psi(\frac{t-u}{s}).$$

These translated and scaled versions of the mother wavelet allow for a finer analysis of the signal since it gives a (filter) response on different scales and at different positions. See an example in Figure 2.6 that shows the advantage of having an analysis taking different scales into account. The result of the wavelet transform shows clearly that it is essential to



(a) Noisy input signal with a periodicity that is clearly visible.

(b) Result of the 1D multiscale (continuous) Wavelet transformation with brighter values being higher.

Figure 2.6.: Input signal (a) and the multiscale wavelet transform with a Daubechies basis (b). Only the higher scale reveals the periodic nature of the signal while the smaller wavelet children only respond to the noise.

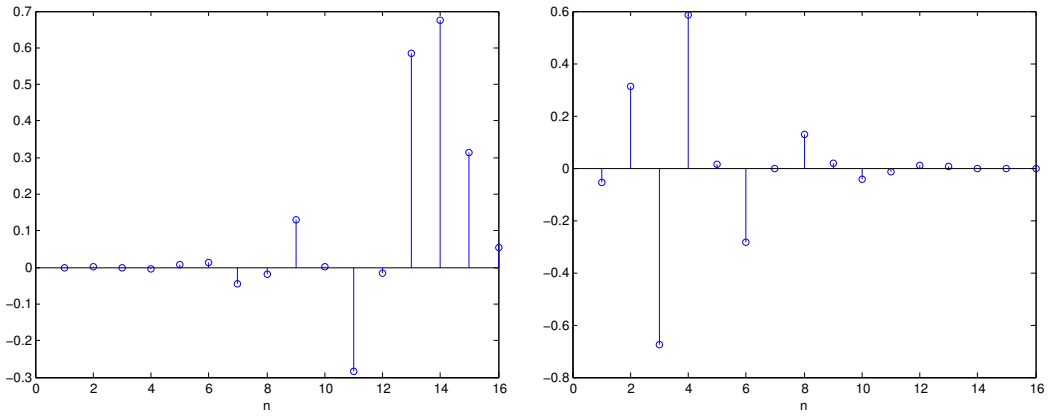
consider a variety of scales to truly capture all the signal's characteristics.

2.4.3. Discrete Wavelet Transform

Since the data at hand is discrete in most cases one needs to also deal with a discrete version of the wavelet transform (DWT). There are multiple approaches to tackle the problem but they all start by a finite sampling of the wavelet basis, often in dyadic scale steps to reduce redundancy: $\psi_s[n] = \frac{1}{2^s}\Psi(\frac{n}{2^s})$. A straightforward approach is to create discrete high-pass filters ψ_s^h and low-pass filters ψ_s^l from the wavelet (see example in Figure 2.7) for multiple scales s which are then applied to the image in the form of a convolution with a filter bank:

$$\tilde{f}_l[n,s] = f \star \psi_s^l[-n] \qquad \tilde{f}_h[n,s] = f \star \psi_s^h[-n]$$

There are further methods that involve lifting schemes or more involved filtering steps. Also note that if the wavelet is separable, a kD-DWT can be computed by computing a 1D-DWT in every of the k dimensions.



(a) Low-pass filter ψ^l (b) High-pass filter ψ^h

Figure 2.7.: Examples of discrete filters taken from a Daubechies wavelet.

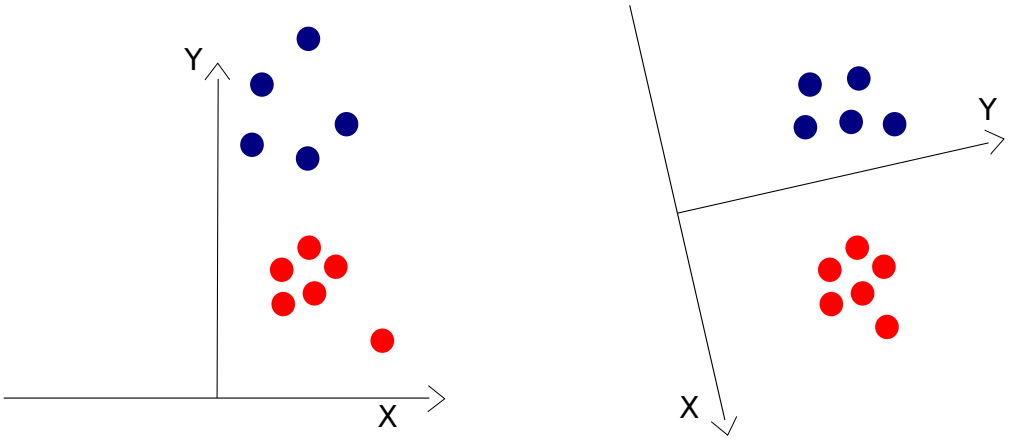
2.5. Subspace methods

Subspace methods form an integral part of many machine learning, data mining and computer vision algorithms. For lots of problems the given data is high-dimensional although often enough the important pieces of information reside in a lower-dimensional subspace. The main goal of these subspace methods, with PCA among the most prominent (see, for example, [Bishop, 2006]), is to determine a new basis so that the most meaningful information can be kept while unimportant parts of the space can be discarded.

2.5.1. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a supervised approach to find a basis transformation maximizing the distance between different label means while simultaneously keeping samples of the same label close to each other. Here, supervised means that we have k labels and for each sample x it is known to which label set $C_1, ..., C_k$ it belongs to. In Figure 2.8

an exemplary application of LDA to a given set of labeled samples is shown. The newly computed system now consists of transformed samples which are easier to separate.



(a) Original system (b) Newly computed system

Figure 2.8.: A system with given labeled samples before (a) and after (b) applying LDA.

Mathematically, the within-scatter matrix S_w is the sum of label-wise matrices S_w^i, defined by

$$S_w = \sum_{i=1}^{k} S_w^i \quad , \quad S_w^i = \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T \quad \text{with} \quad \mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

and then the between-scatter matrix S_b is calculated as follows:

$$S_b = \sum_{i=1}^{k} |C_i|(\mu - \mu_i)(\mu - \mu_i)^T \quad \text{with} \quad \mu = \frac{1}{\sum_i |C_i|} \sum_{x \in C_1, \ldots, C_k} x.$$

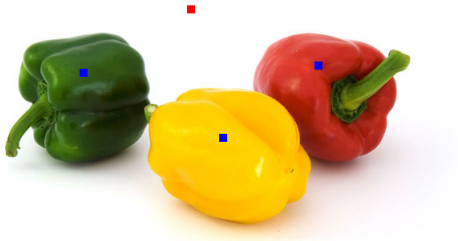
Since the goal is to maximize between-scatter and minimize within-scatter, a matrix G is sought (i.e. a set of vectors projecting into the new basis) which maximizes the multi-label version of the related Fisher's discriminant [Fisher, 1936]:

$$\max J(G) = \max \frac{\det(G^T S_b G))}{\det(G^T S_w G)} \quad \text{leads to} \quad \frac{dJ}{dG} \stackrel{!}{=} 0 \Leftrightarrow S_w^{-1} S_b G = J(G) G.$$

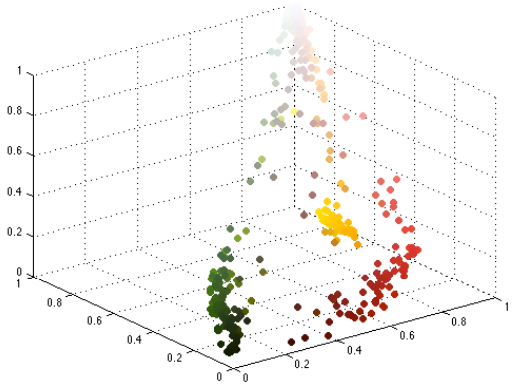
Now it is obvious that G can be computed by an eigenvalue decomposition of $S_w^{-1} S_b$ and using the $k - 1$ eigenvectors corresponding to the largest eigenvalues. An application of this method can be seen in Figure 2.9.

2.5.2. Orthogonal Linear Discriminant Analysis

The standard LDA has serious drawbacks. Firstly, to find a solution the matrix S_w has to be non-singular which is not always the case in practice (e.g. when having few samples).

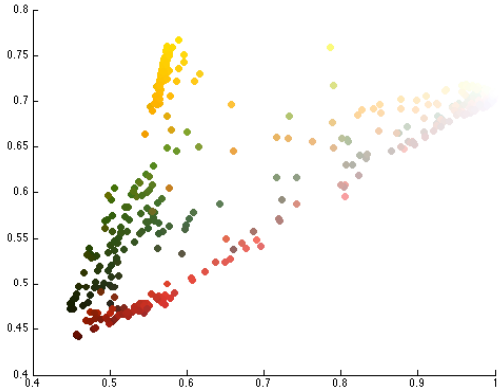


(a) The pepper image



(b) The pixels in the RGB space

(c) The same image after applying LDA



(d) The pixels in the new subspace

Figure 2.9.: Image before (a) and after (c) LDA in the RGB space. The three peppers were marked as belonging to the foreground. After LDA, the RGB colors were projected with the eigenvector corresponding to the largest eigenvalue. Figures (b) and (d) show scatter plots of the pixel colors. It shows that a separation can now be achieved by simply thresholding instead of finding a separating plane in the former case. The pepper image is taken from [Santner et al., 2010].

Secondly and contrary to popular belief, it has been shown that LDA does not produce an orthogonal solution and therefore can lead to unsatisfying results in dimensionality reduction [Beveridge, 2001, Luo et al., 2011]. In recent literature, a lot of different models have been proposed to tackle these problems or improve certain aspects (see [Ye, 2006]). Here, the model of choice is one formulation of the Orthogonal Linear Discriminant Analysis (OLDA) from [Ye, 2005] which yields an orthogonal solution by construction and implicitly avoids the singularity problem.

The new objective function has the form

$$G = \arg\max_{G \in \mathcal{G}} \text{ trace}((G^T S_t G)^+ G^T S_b G)$$

with $\mathcal{G} := \{G \in \mathbb{R}^{m \times l} \mid G^T G = I_l\}$, m the data dimensionality, l the reduced dimensionality and the total scatter matrix $S_t = S_w + S_b$. The optimization is based on simultaneously diagonalizing the three scatter matrices in the following way:

- Compute $H_t := \frac{1}{\sqrt{n}}(x_1 - \mu, ..., x_n - \mu)$ as the mean-centered sample matrix.
- Compute $H_b := \frac{1}{\sqrt{n}}(\frac{\mu_1 - \mu}{|C_1|}, ..., \frac{\mu_k - \mu}{|C_k|})$ as the label-wise mean-centered means matrix.
- Do a SVD of $H_t = U \Sigma V$.
- Extract Σ_t from $\Sigma = \begin{pmatrix} \Sigma_t & 0 \\ 0 & 0 \end{pmatrix}$ with $\text{rank}(S_t) = t$ non-zero entries.
- Partition $U = (U_1, U_2)$ with image $U_1 \in \mathbb{R}^{m \times t}$ and null space $U_2 \in \mathbb{R}^{m \times (m-t)}$.
- Set $B = \Sigma_t^{-1} U_1^T H_b$ and do a SVD $B = P \Xi L^T$.
- Define $X_p = U_1 \Sigma_t^{-1} P_q$ with $q = \text{rank}(S_b)$.
- Compute the QR decomposition of $X_q = QR$ and set $G = Q$.

In [Ye, 2005] it is shown that $X = \begin{pmatrix} X_p & 0 \\ 0 & I_{m-t} \end{pmatrix}$ diagonalizes the three scatter matrices, meaning $X^T S_w X$, $X^T S_b X$ and $X^T S_t X$ are all diagonal.

2.6. Kernel Density Estimation

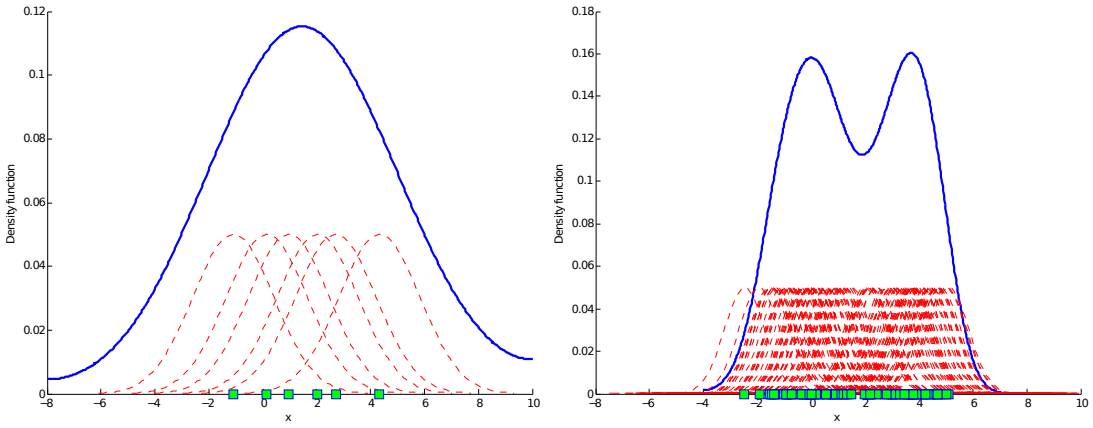
The kernel density estimation (KDE, also called the Rosenblatt-Parzen window method, [Rosenblatt, 1956, Parzen, 1962]) is an approach to estimate the underlying probability density function of a random variable and can be regarded as the continuous counterpart to histograms. Similar to histograms, KDE is non-parametric which means that there is no a-priori assumption about the distribution from which the samples are drawn.

Mathematically, we are given iid samples $x_1, ..., x_n$ taken from an unknown distribution $\mathcal{P}$. Given a new sample x, we can estimate a density at this point by

$$\hat{\mathcal{P}}(x) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{K}_\sigma(x - x_i)$$

with $\mathcal{K}_\sigma$ being a so-called (symmetric and non-negative) kernel function with an associated bandwidth parameter σ. Estimating the density at every point will then yield the approximate density function. There exist many different kernel functions that exhibit slightly different properties, although in general the Gaussian kernel $\mathcal{K}_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \, \mathrm{e}^{\frac{-x}{2\sigma^2}}$ is mostly used due to its smoothness and separability.

Apart from the choice of the kernel itself there are essentially two important factors that influence the density estimation: firstly, the number of provided samples and secondly, the chosen bandwidth for the used kernel. See Figure 2.10 for a visualization. It is obvious that when the number of samples increases the bandwidth should be chosen smaller, since it has a smoothing effect on the shape of the function. It is therefore essential to find an optimal bandwidth that approximates well while at the same time minimizes smoothing. This is analogous to histograms where one would decrease the bin width with an increasing amount of drawn samples.



(a) 5 samples with $\sigma = 1.5$ (b) 100 samples with $\sigma = 0.7$

Figure 2.10.: Two estimates of the same density function using Gaussian kernels. The green boxes show the samples' positions and the red lines the kernel for each sample. The blue contour represents the estimated density function. By increasing the number of drawn samples while decreasing the bandwidth the true distribution, namely two Gaussians, can be observed.

Part II.

The proposed Model

3. Overview of the proposed approach

This chapter gives an explanation of interactive image segmentation and the foundation on which this work is built on. It further motivates the addition of texture information to alleviate problems that arise when only regarding space and color. Lastly, it gives a short summary of texture descriptors that are used in literature and how wavelets are applied in this context.

3.1. Interactive Image Segmentation via spatially-varying distributions

As already stated in the introduction the goal of this work is to provide a robust method to segment images in an interactive setting. The advantage of such a setting is the inherent supervised approach: the user can see the intermediate result of his or her input and further improve it iteratively until the segmentation yields satisfying results. The disadvantage, on the other hand, is that the segmentation is forced to be fast enough, i.e. responsive to frequent changes in the input, to be a viable solution for the problem. Therefore, complex models that employ different learning techniques and involve heavy computations are not favorable to this task. Hence, the focus is a model that is fast to calculate while being discriminant enough.

The thesis follows the work of [Nieuwenhuis et al., 2011, Nieuwenhuis and Cremers, 2012] in which they consider a data model that incorporates spatial and color information into a statistical framework. They showed that their work is state-of-the-art by outperforming other established approaches like GrabCut [Rother and Kolmogorov, 2004] or the work presented in [Santner et al., 2009] by exploiting the natural correlation of space and color in images. They argue that other publications employing color information neglect this correlation and solely use already marginalized distributions that are independent of location.

Next to space and color, images can further be described by texture information. Basically, texture is a combination of space and color, created in patterns which are most often impossible to grasp since they follow no parametric description. Textural information will be described in more detail in the next subsection. Similar to the color case, texture is highly space-dependent. An object can consist of multiple textures which may fade into each other or it may even share texture with other objects from the background. Therefore, describing texture should naturally come in a spatially-aware manner which is going to be done in this work. See Figure 3.1 for some real-life images where texture plays a descriptive role.

Figure 3.1.: Some examples taken from IcgBench that show the important role of texture information in natural images. In many cases only texture can distinguish different objects with similiar colors.

3.2. Texture information

Usually, texture is a very important clue when it comes to image analysis since it can help in discerning objects that have the same colors but exhibit different color patterns. Such information is not only used to segment images but also to synthesize and inpaint missing image parts [Harrison, 2005, Kawai et al., 2009, Arias et al., 2011]. The problem with describing textured regions is that they do not typically relate to piecewise-smooth or piecewise-constant assumptions but are quite irregular in terms of orientation, magnitude, scale or periodicity.

Texture information has been widely used and captured in the literature with different approaches. Many authors try to capture local regularity in the intensities by graph structures

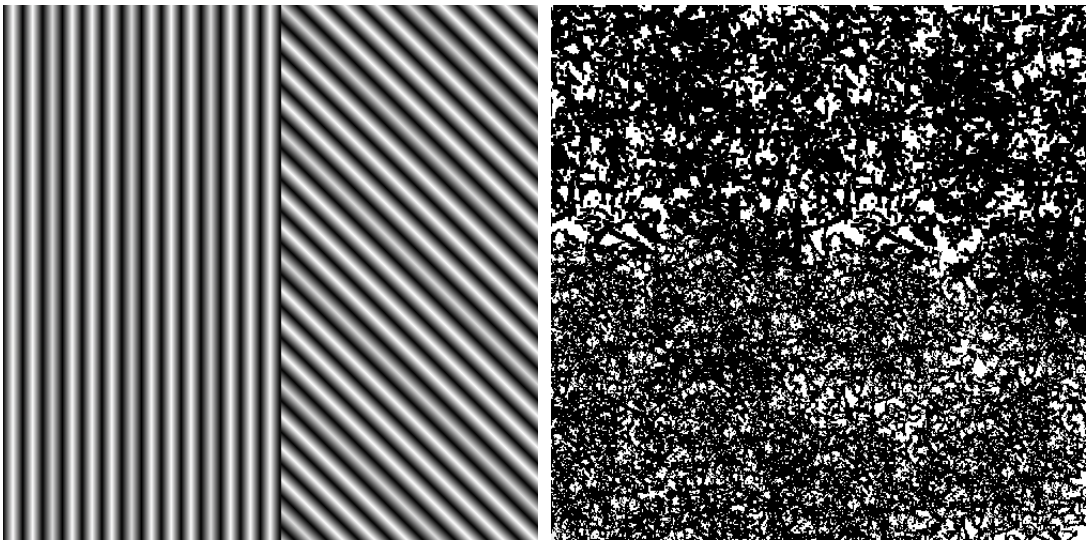
[Efros and Leung, 1999, Gimel'Farb, 1997, Awate et al., 2006, Cremers and Grady, 2006] and while these methods show promising results, they lack robustness when textures are highly irregular in orientation or scale. Although this can be weakened by using differently-sized and rotated graphs, the structural problem that arises from using a discrete lattice to describe continuous phenomena remains.

Therefore, many authors tend to describe texture by local (higher-order) statistics over intensities inside a window. While these are able to capture a whole different set of information and do not suffer from some of the irregularity problems of the discrete textural models, they seldom provide a visual explanation of the information in terms of direction or orientation of the color pattern. This is why sometimes statistical models are supplemented with complimentary information (e.g. directional information from HoG [Dalal and Triggs, 2005]) [Emrich et al., 2010]. A widely established statistical texture descriptor is the Haralick feature set [Haralick, 1979] that computes statistics over gray-scale cooccurences and has seen extensive and successful use in many publications. One of the problems with Haralick features is their exponential growth in gray-scale quantization steps. In order to compute the cooccurences the image must be thresholded into multiple gray-scale levels (e.g. 8,16) which causes an avoidable loss in textural information. Other approaches include the computation of statistics following a given pattern like LBP [Ojala et al., 1996] which basically mixes structural and statistical information into one common descriptor. While LBPs seem to perform well on different textures, they fail in terms of capturing varying scales. See Figure 3.2 for two pathological cases where structural and statistical descriptors defy each other. Conclusively, finding a robust and well-performing descriptor for texture information is a difficult task and therefore still subject to ongoing research.

A completely different idea is the computation of texture by means of an image transformation and decomposition into a different representation. Especially wavelets have proven themselves to be the method of choice in that specific domain because of their descriptive power [Sebe and Lew, 2000, Busch and Boles, 2002] and because they are easy to parallelize in computation [Franco et al., 2009]. In the theory section it was mentioned that discrete wavelet transformations are usually implemented by a pair of orthogonal low-pass and high-pass filters ψ^l, ψ^h. Since we use a 2D-wavelet decomposition, we assume the wavelet base Ψ to be separable: $\Psi(x, y) = \Psi(x)\Psi(y)$ which holds true for most filters that are commonly used.

Starting at the first scale $s = 0$ and setting $A_0 = I$ as the gray-scale input image we compute four subbands per scale with the following recursive scheme:

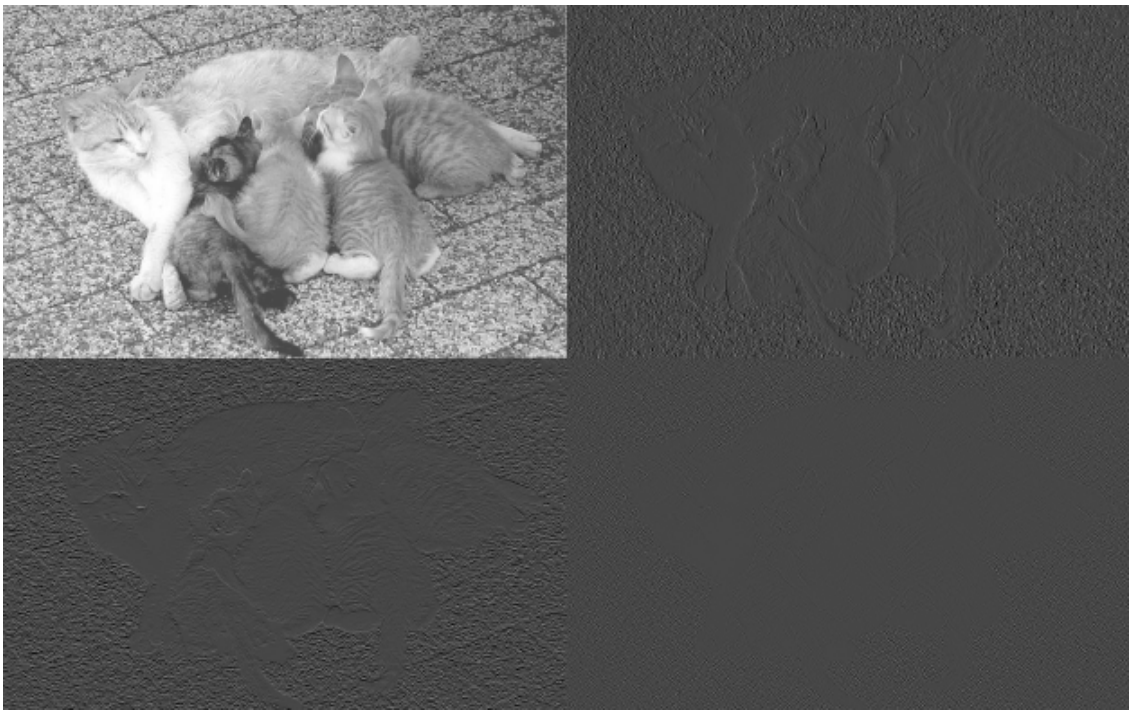
$$
\begin{aligned}
A_s &= (\psi_x^l \star (\psi_y^l \star A_{s-1})_{\downarrow x})_{\downarrow y} \\
H_s &= (\psi_x^h \star (\psi_y^l \star A_{s-1})_{\downarrow x})_{\downarrow y} \\
V_s &= (\psi_x^l \star (\psi_y^h \star A_{s-1})_{\downarrow x})_{\downarrow y} \\
D_s &= (\psi_x^h \star (\psi_y^h \star A_{s-1})_{\downarrow x})_{\downarrow y}
\end{aligned}
$$



(a) Textures with structure (b) Textures with noise

Figure 3.2.: Two different texture segmentation problems. The textures in (a) would be easy to separate with structural features but harder with statistical ones. The textures in (b), on the other hand, could be easily distinguished by their difference in statistics but hardly by their structure.

which are also referred to as the three detail subbands H_s, V_s, D_s and one approximation subband A_s. The underscored ψ^l, ψ^h denote in which dimension the convolution has taken place and the $\downarrow$ denotes a down-sampling by the factor 2 in the given dimension. The approximation A_s is computed by a low-pass filtering in both dimensions while the detail coefficients H_s, V_s, D_s are obtained by high-pass filtering in one or both dimensions, revealing information about spatial changes in the horizontal, vertical or diagonal direction. Also note that the number of coefficients equals the number of pixels due to the quadratic subsampling involved in the computation. To finally end up with three values per pixel for one scale we resize all subbands to the original image size and use bilinear interpolation. See Figure 3.3 for two examples of a discrete wavelet decomposition.



(a) DWT using a Haar basis on one scale

(b) DWT using a Daubechies basis on three scales

Figure 3.3.: Visualization of 2D-DWTs of the grayscale cat image. In (a) the upper-left corner shows the approximation for the next scale whereas the other three are the detail subbands. In (b) one can see the recursive pattern, capturing information with different granularities.

4. Model and Optimization

This chapter introduces the computational model that will be used in this work. Firstly, Bayesian inference for segmentation will be presented followed by an explanation of how to estimate densities for space, color and texture from the user scribbles. Secondly, the variational energy will be formulated as a saddle point problem together with an optimization technique. Lastly, we briefly visit the weighting term $g(x)$ and show alternative definitions that arise from the given context.

4.1. Modeling the distribution

Given the image and a set of user scribbles the task is to use the spatial as well as the color and texture information by incorporating them into the probabilistic model. The idea is to find a suitable representation of the information so that it may serve as the data term f in the energy functional from equation 2.10.

This work follows [Nieuwenhuis et al., 2011, Nieuwenhuis and Cremers, 2012] to formulate the probabilistic model using Bayes inference. The goal is to maximize the conditional probability $\mathcal{P}(\mathcal{L}|I)$ with respect to $\mathcal{L}$, where I is the image and $\mathcal{L}$ is the labeling, i.e. the segmentation, to be determined. Unfortunately, finding the maximizer is non-trivial since there is no obvious structure for $\mathcal{P}$. Using Bayesian inference it can be rewritten as

$$\arg\max_{\mathcal{L}} \mathcal{P}(\mathcal{L}|I) = \arg\max_{\mathcal{L}} \frac{\mathcal{P}(I|\mathcal{L})\mathcal{P}(\mathcal{L})}{\mathcal{P}(I)}$$

with a prior $\mathcal{P}(\mathcal{L})$ and a normalization $\mathcal{P}(I)$. Since we will employ a TV regularizer that penalizes the length of the boundaries, we define a prior over segmentations that favor short interfaces between regions:

$$\mathcal{P}(\mathcal{L}) \propto \exp(-\frac{1}{2}\sum_{i=1}^{k} Per(E_i, \Omega)).$$

The normalization term can be neglected because the interest lies only in the maximizer of the energy and not the energy itself. To proceed, it is assumed that the color of one pixel is independent from all the other pixels, leading to

$$\mathcal{P}(I|\mathcal{L}) = \prod_{x\in\Omega} \mathcal{P}(I(x)|x, \mathcal{L})^{dx}$$

with the exponent dx denoting an infinitesimal volume in $\mathbb{R}^2$ ensuring correct continuum limits. Furthermore, it is assumed that the color of one pixel is independent of the labeling

of all other pixels, giving

$$\mathcal{P}(I|\mathcal{L}) = \prod_{i=1}^{k} \prod_{x \in \Omega} \mathcal{P}(I(x)|x, \mathcal{L}(x) = i)^{dx}.$$

Of course, these independency assumptions are harsh and virtually violated for every real-life image. Nevertheless, it allows to model the problem in a straight-forward and tractable way. Since the goal is to maximize the expression, the negative log-likelihood

$$\arg\max_{\mathcal{L}} \mathcal{P}(\mathcal{L}|I)\mathcal{P}(\mathcal{L}) = \arg\min_{\mathcal{L}} -\log \mathcal{P}(\mathcal{L}|I) - \log \mathcal{P}(\mathcal{L})$$

then yields the MAP solution

$$\arg\min_{\mathcal{L}} \sum_{i=1}^{k} \sum_{x \in \Omega} -\log \mathcal{P}(I(x)|x, \mathcal{L}(x) = i)^{dx} + \frac{1}{2} \sum_{i=1}^{k} Per(E_i, \Omega) \tag{4.1}$$

which clearly resembles the energy from Equation (2.9). Now it shows that one can define the data term f of the energy functional for every pixel x and every label i as

$$f_i(x) := -\log \mathcal{P}(I(x)|x, \mathcal{L}(x) = i)^{dx}. \tag{4.2}$$

and we arrive at the aforementioned TV-segmentation energy from the introduction.

4.1.1. Estimating the density

Now it remains to clarify how the density $\mathcal{P}(I(x)|x, \mathcal{L}(x) = i)^{dx}$ can be computed for a given set of scribbles. Recall that a set of scribbles was defined as

$$S := \{S_1, ..., S_k\} \quad , \quad S_i := \{\mathbf{x}_j^i, j = 1, ..., m_i\}$$

and that a density can be estimated using KDE from the last chapter as follows

$$\hat{\mathcal{P}}(x) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{K}_\sigma(x - x_i).$$

Since the work focuses on spatially-varying color and texture information, one can estimate the density by

$$\hat{\mathcal{P}}(I(x)|x, \mathcal{L}(x) = i)^{dx} = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathcal{K} \begin{pmatrix} x - x_{ij} \\ I(x) - I(x_{ij}) \\ T(x) - T(x_{ij}) \end{pmatrix}$$

which means that we have a multidimensional distributional space. By exploiting the separability property of the Gaussian kernel we can overcome this more complicated form to get

$$\hat{\mathcal{P}}(I(x)|x, \mathcal{L}(x) = i)^{dx} = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathcal{K}_\alpha(x - x_{ij}) \cdot \mathcal{K}_\sigma(I(x) - I(x_{ij})) \cdot \mathcal{K}_\beta(T(x) - T(x_{ij})) \tag{4.3}$$

with a spatial kernel $\mathcal{K}_\alpha(x)$, a color kernel $\mathcal{K}_\sigma(I(x))$ and a texture kernel $\mathcal{K}_\beta(T(x))$, each supplied with an associated bandwidth parameter.

In [Nieuwenhuis and Cremers, 2012] the authors show that their spatially-varying color model reveals interesting properties when the bandwidths are each taken to infinity. By increasing the spatial bandwidth α the influence of the scribbles' placement on the estimate is weakened. With $\alpha \to \infty$ the data term purely becomes an estimate over color distances:

$$\hat{\mathcal{P}}(I(x)|x, \mathcal{L}(x) = i)^{dx} = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathcal{K}_\sigma(I(x) - I(x_{ij})).$$

If the color bandwidth σ increases the spatial influence on the estimate will rise. In the limit $\sigma \to \infty$ the density will be solely based on spatial distances:

$$\hat{\mathcal{P}}(I(x)|x, \mathcal{L}(x) = i)^{dx} = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathcal{K}_\alpha(x - x_{ij})$$

with the final segmented image being a regularized Voronoi partition based on distances to the input. Accordingly, when those notions ($\sigma \to \infty, \alpha \to \infty$) are both applied to the enhanced model presented here the estimation will be purely based on distances in the texture space:

$$\hat{\mathcal{P}}(I(x)|x, \mathcal{L}(x) = i)^{dx} = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathcal{K}_\beta(T(x) - T(x_{ij})).$$

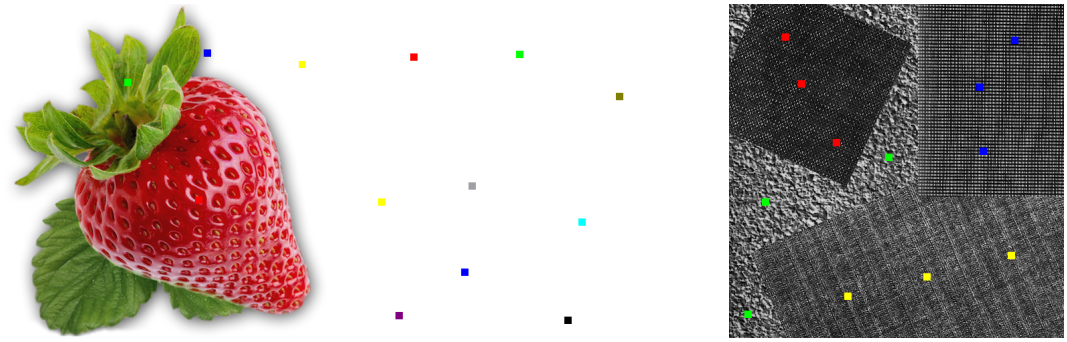
A visualization of this argumentation can be seen in Figure 4.1. Thus, the model can be steered not only be the scribbles' placement but also by these three bandwidths. The user can determine the best weighting in order to produce the best outcome. It should also be mentioned that this amount of freedom in the input can also be disadvantageous, since it forces the user to not only think about the position of the scribbles but also of side effects introduced by these further parameters.

The next sections are going to present how each kernel has been defined in this work. Note that the Parzen method assumes iid samples which is clearly not the case for user-provided scribbles, since they are placed by the user in a visually-driven way. Furthermore, having high-dimensional kernels can lead to the well-known curse of dimensionality, meaning that sparsity of information increases with a growing number of dimensions and can lead to numerical problems during computation.

4.1.2. Spatial kernel

The spatial kernel is supposed to influence the density based on the distance of one pixel to the scribbles. Intuitively, the closer the pixel x is to scribbles of a specific class i the higher the likelihood of that pixel x of belonging to the class i should be. One approach is to employ isotropic kernels that take the Euclidean distance as an argument. To alleviate the problem of having scribbles placed non-uniformly on the image one can endow the bandwidth parameter with an argument so that the isotropic kernel becomes

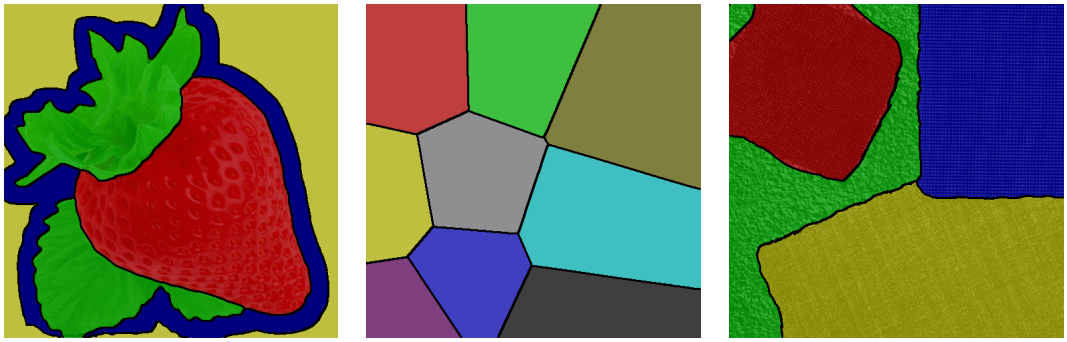
$$\mathcal{K}_{\alpha(x)}(||x - x_{ij}||^2) \quad , \quad \alpha(x) = \delta \cdot \max(1, \min_{x_{ij}} ||x - x_{ij}||^2) \tag{4.4}$$



(a) Image taken from the internet together with 4 labels

(b) Randomly placed scribbles on white background

(c) Custom collage of Brodatz textures [Brodatz, 1966]



(d) Only color information

(e) Only space information

(f) Only texture information

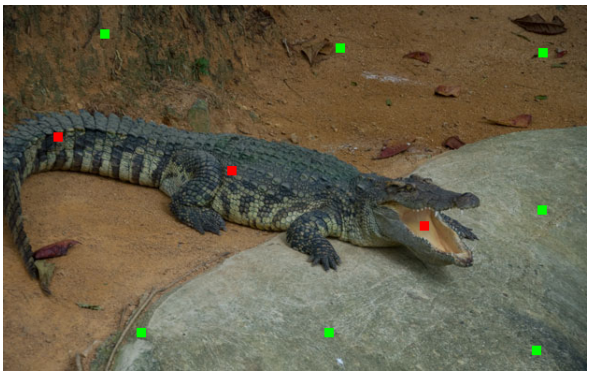
Figure 4.1.: Input and segmentation results. In (a) only the color kernel was used for the estimation, in (b) only the spatial kernel and in (c) only the texture kernel.

This version has a bandwidth δ multiplied with the distance from point x to its nearest scribble. This asserts that pixels that are far off from any placed scribble still get affected by the spatial component of the density estimation. In Figure 4.2 one can see the results of a density estimation using only the spatial isotropic information. It shows that when using a reasonable amount of scribbles the spatial information is often insufficient for a good estimate. Another problem could arise from the isotropic behavior of the kernels which can lead to an undesirably strong circular influence on the likelihoods around scribbles .

A further approach is to introduce a directional dependence by using anisotropic kernels that take into account how the scribbles are positioned in relation to each other. Since humans place the scribbles intuitively along the shape of the object they want to segment, anisotropy arises naturally in that context and can help in shaping the density in a more favorable way. The idea here is that every label has its own spatial distance measure which is computed by the covariance of the input. Mathematically, this leads to

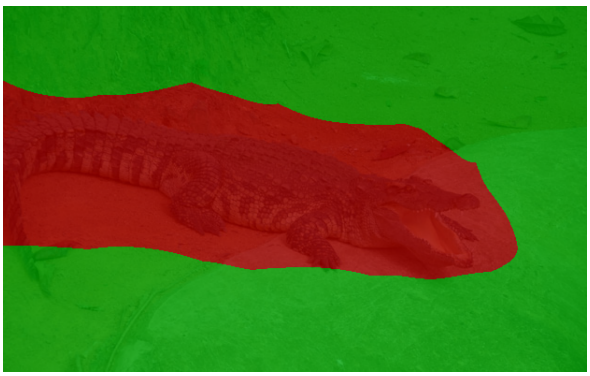
$$\mathcal{K}_{\alpha(x)}(\sqrt{(x-x_{ij})^T\Sigma_i^{-1}(x-x_{ij})}) \quad , \quad \Sigma_i = \begin{pmatrix} \sigma_i^1 & \sigma_i^3 \\ \sigma_i^3 & \sigma_i^2 \end{pmatrix} \tag{4.5}$$

and is also known as the Mahalanobis distance. Here, Σ_i is the normalized covariance



(a) Crocodile image from IcgBench

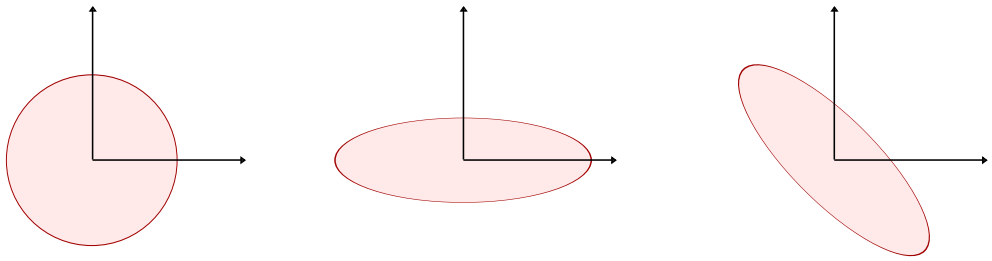
(b) KDE using only isotropic spatial information



(c) KDE using anisotropic spatial information

Figure 4.2.: In (a) an image with scribbles is provided, (b) shows the density estimation using only isotropic spatial information with $\alpha = 2$ and (c) when employing anisotropic information. The pixel color is determined by the most-likely class for each pixel. One can clearly see the circles that arise from the isotropic Euclidean measure and how this problem gets weakened in the anisotropic case.

matrix of the scribble positions for label i. This measure skews the Euclidean space in such a way that points on an ellipsis have the same distance from the center point. See Figure



(a) Standard Euclidean distance measure with $\Sigma = I$

(b) Skewed with $\sigma^1 = 1.5, \sigma^2 = 0.5, \sigma^3 = 0$

(c) Skewed with $\sigma^1 = \sigma^2 = 1$ and $\sigma^3 = -0.5$

Figure 4.3.: Visualization of the Mahalanobis distance. In (a) the standard Euclidean measure is used whereas in (b) and (c) the covariance matrix is not the identity matrix. Every point on the red line has the same distance from the center point.

4.3 for a visual explanation and again Figure 4.2 for a practical result. The advantage clearly shows when the object has an elongated shape or even when the object of interest is fairly distant from a circular shape. In the latter case (and assuming that the scribbles are positioned in an appropriate way) the anisotropic distance measure will revert to an isotropic one since the covariance will be close to the identity matrix.

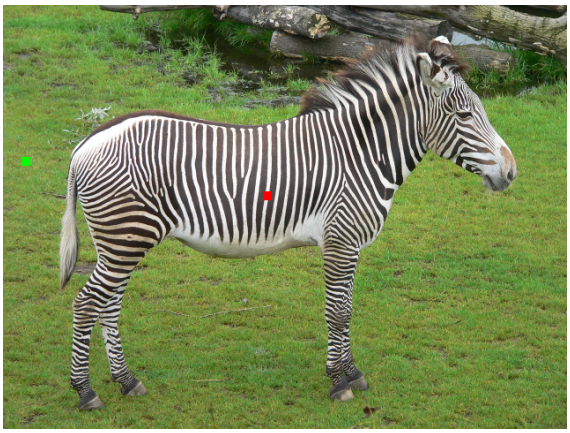
4.1.3. Color kernel

The color kernel should shape the density in such a way that pixel x belongs to a certain class i if its color is close to the pixel colors of the scribbles' positions of class i. Mathematically, the likelihood rises when the color in the color space is near to the color the scribble is placed upon and corresponds in the RGB space to

$$\mathcal{K}_\sigma(I(x) - I(x_{ij})) := \mathcal{K}_\sigma(R(x) - R(x_{ij})) \cdot \mathcal{K}_\sigma(G(x) - G(x_{ij})) \cdot \mathcal{K}_\sigma(B(x) - B(x_{ij})). \quad (4.6)$$

That is, we multiply a kernel for each color channel and the distance therein and end up with a joint estimate. Also note that we can write it in this product form since we use Gaussian kernels.

Although this approach gives generally good results, it can become problematic if the object consists of multiple colors which are too far apart in the color space. In Figure 4.4 a pathological case is presented when estimating the likelihoods only based on color information. The object of interest is a zebra consisting of a black-and-white pattern with both colors having the largest possible distance from each other in the color space. The image background is mostly green with the color being located between black and white in the color space, although a bit closer to black. The subfigure (b) shows that especially those parts of the zebra which are not completely black or white are more likely to belong to the background due to their vicinity to green colors in the RGB space.



(a) The zebra image

(b) RGB color density

(c) OLDA transformed color space

(d) Its OLDA color density

Figure 4.4.: A zebra image (a) taken from the internet and the KDE (b) using only the color kernel in RGB space. In (c) the OLDA transformed color space is shown and (d) the final density estimation. For both estimations the parameter was $\sigma = 0.1$

A possibility to weaken this problem is the idea of transforming the color space to a more suitable representation. Since we already have user scribbles that tell which colors belong to which label, a supervised method can be used that yields such a new and improved representation. In this work LDA (respectively OLDA) was employed to create a new color space in which colors of the same label should reside more closely to each other while enlarging the distance to colors of other labels. See figures (c) and (d) for the image in the transformed color space and the KDE result. To keep maximum discriminancy all three dimensions of the new space are taken for the estimation. It can be seen that the foreground has been encoded in the former green and blue color channels whereas the background mostly resides in the former red channel. By this, the distances in the color space changed and the estimation gave more coherent and globally improved likelihoods.

4.1.4. Texture kernel

To bring texture information into this work the choice here is to use a multiscale discrete wavelet transformation of the grayscale input image. As already mentioned in the theory section one normally resorts to using analysis banks with orthogonal high-pass and low-pass filters (also referred to as quadrature mirror filters) which decompose the signal into multiple bands. The low-frequency subbands are then recursively decomposed to achieve the analysis on multiple scales. Mathematically, every pixel x then has three values per scale s from the DWT bands:

$$T(x) := (\, x_D \, , \, x_H \, , \, x_V \,)_s$$

with underscored D, H, V denoting the detail coefficients for the diagonal, horizontal and vertical subbands. These coefficients describe the very local behavior of the signal and thus, are sometimes not sufficient to provide a good description. Therefore, it is common to compute statistics of these coefficients inside a window which significantly raises the descriptive power of the wavelet decomposition. Here, we compute the absolute mean and variance of each subband at every scale to endow every pixel x with a signature

$$T(x) := (\, \mu_D(x) \, , \, \sigma_D(x) \, , \, \mu_H(x) \, , \, \sigma_H(x) \, , \, \mu_V(x) \, , \, \sigma_V(x) \,)_s \tag{4.7}$$

while the signature's components are computed inside a given window $W(x)$ as follows:

$$\mu_{band}(x) := \int_{W(x)} |x_{band}| \, dx \quad , \quad \sigma_{band}(x) := \sqrt{(\int_{W(x)} |x_{band}| - \mu_{band}(x))^2 \, dx}.$$

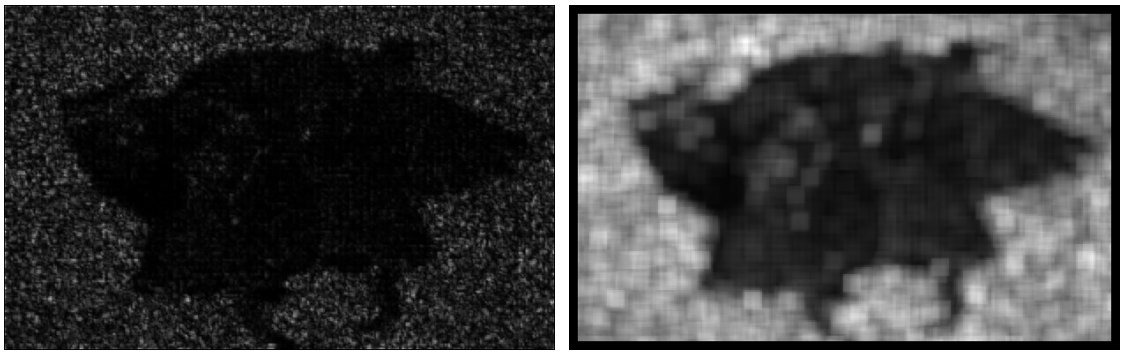
The following Figure 4.5 shows the output of using statistics for the texture information with different window sizes. Although we improved the texture information with this step we also raised the dimensionality of our kernel estimate. Hence, we can once again employ OLDA to achieve a supervised reduction in the number of used dimensions while retaining a high amount of information. Another downside is that we now introduced two further degrees of freedom: the choice of a suiting wavelet base as well as determining a good choice for the window size of the statistics computation.

4.1.5. Automatic bandwidth estimation

One disadvantage of having multiple kernels with different bandwidths is the amount of freedom that is exposed to the user. Not only is the scribble placement important for the final segmentation but also the choice of suitable kernel bandwidths which can have a tremendous influence. One can therefore think of ways on how to estimate the bandwidths automatically from the given user information. The idea pursued here is to analyze how much variation one label has in the corresponding color and texture space and set parameters accordingly. This means that we break up the bandwidths into a label-wise parameterization:

$$\beta = (\beta_1, ..., \beta_k) \quad , \quad \sigma = (\sigma_1, ..., \sigma_k).$$

By this, we can give tighter bounds for labels with small color and texture changes and looser bounds for labels with highly varying colors and textures. See Figure 4.6 for a



(a) Window size of 2 (b) Window size of 15

Figure 4.5.: Visualization of one component from the computed signature using different window sizes. The image itself is the cat image from Figure 3.3.

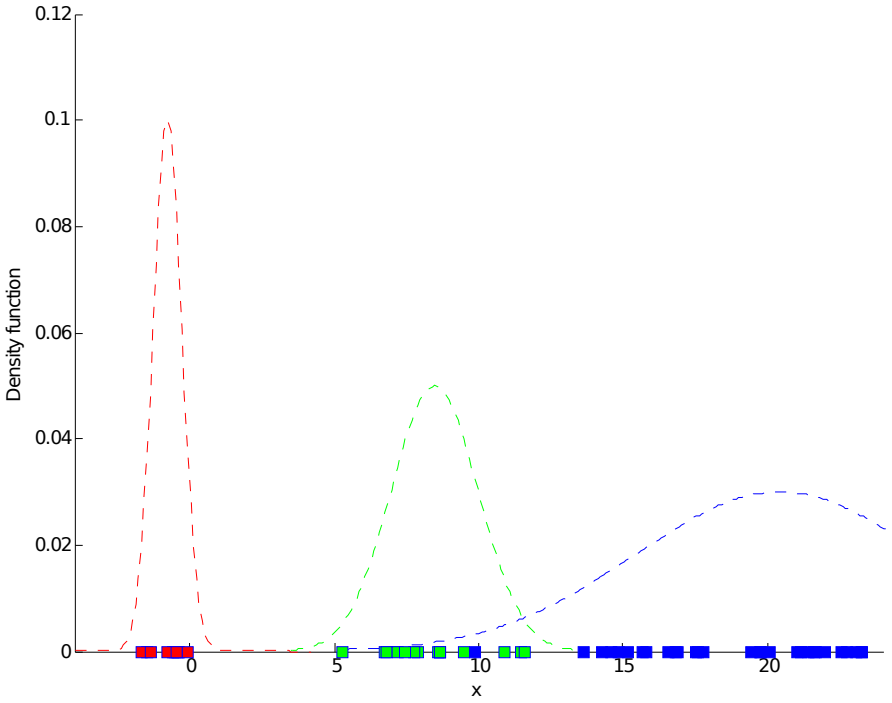


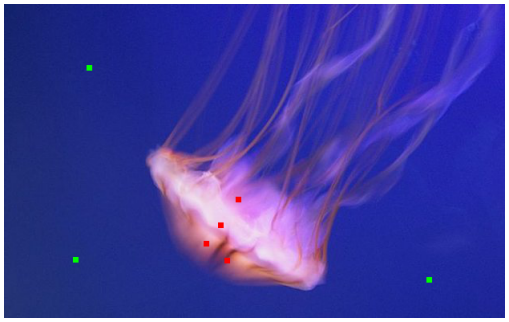
Figure 4.6.: Visualization of samples where each label has a distinctive variance in the information. This can be exploited for automatic bandwidth estimation.

visual explanation. To define the bandwidths we just compute the variance for each label by treating color and texture information for one label i as observations from two random variables I_i, T_i and take the biggest scalar value to ensure maximum information:

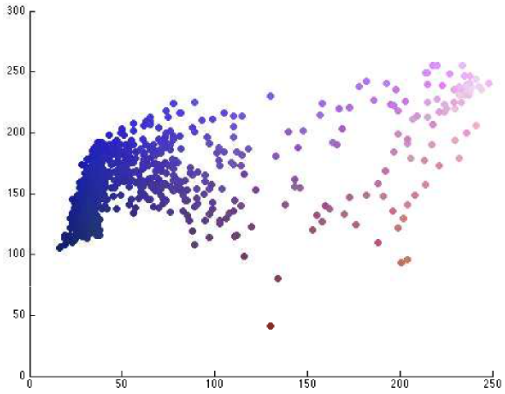
$$\beta_i = \max(Var(T_i)) \quad , \quad T_i = [T(x_{i1}), \cdots, T(x_{im_i})],$$

$$\alpha_i = \max(Var(I_i)) \quad , \quad I_i = [I(x_{i1}), \cdots, I(x_{im_i})].$$

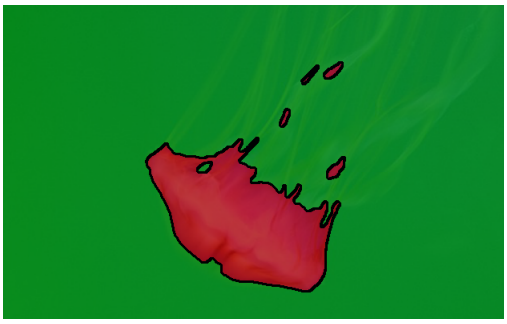
Also note that especially here it is important that the user sets the scribbles in a representative manner, i.e. making sure that the marked colors and textures convey enough information for the automatic estimation. See Figure 4.7 for a practical result. There are also



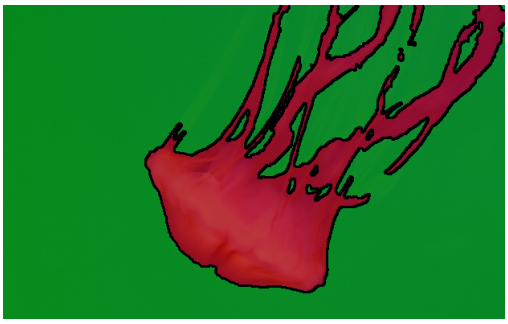
(a) Input image with with two classes



(b) Distribution in red and blue subspace



(c) Manually set bandwidth



(d) Automatically estimated bandwidth

Figure 4.7.: Input image with two classes (a) and segmented results (c) and (d) using only color information and $\lambda = 1$. In (c) the bandwidth $\sigma = 0.15$ was manually determined for the best visual result. In (d) the parameters were set automatically to $\sigma_1 = 0.21$ and $\sigma_2 = 0.04$, reflecting the fact that the background has far less variation than the foreground. This can be seen in the color plot (b).

more involved techniques to estimate bandwidths for kernels, e.g. using plug-in methods [Jones et al., 1996] or diffusion [Botev et al., 2010] but they are usually computationally more intense and are therefore not taken into consideration here.

4.2. Optimizing the energy

As already stated in the introduction many problems in computer vision are formulated as discrete or continuous energy minimization problems that can be tackled with mathematical optimization techniques. In this work an energy in the form of a Mumford-Shah-functional is employed and solved in the form as a saddle point problem. To optimize it an alternating, iterative scheme is used that reprojects the variables to its convexity con-

straints after each step.

4.2.1. Saddle point problems and primal-dual formulation

One sort of problems in mathematical optimization are the so-called saddle point problems which possess the following general form:

$$\min_{x\in X}\max_{y\in Y} E(x,y)$$

with X, Y being two arbitrary vector spaces. The point $(\hat{x},\hat{y})$ is a saddle point if satisfying

$$\hat{x}\in\arg\min_{x\in X} E(x,\hat{y}) \quad \text{and} \quad \hat{y}\in\arg\max_{y\in Y} E(\hat{x},y).$$

Also note that the saddle value $\max_y E(\hat{x},y) = E(\hat{x},\hat{y}) = \max_x E(x,\hat{y})$ is always unique. If furthermore, the functional $E(x,y)$ is a convex-concave function (meaning that $E(x,y)$ is convex in x and that $-E(x,y)$ is convex in y), then a saddle point can be characterized by subgradient conditions

$$0\in\partial_x E(\hat{x},\hat{y}) \quad \text{and} \quad 0\in\partial_y(-E)(\hat{x},\hat{y}).$$

Such a saddle point exists if the functional is coercive and closed in respect to each argument. The goal now is to bring our energy into the form of a saddle point problem and exploiting the mentioned properties to reach a global optimum, i.e. the saddle point.

Given two finite-dimensional real vector spaces X, Y, a continuous linear operator $K : X \to Y$ and two proper, convex, l.s.c. functions $F : Y \to [0,+\infty), G : X \to [0,+\infty)$, we define the primal problem (P), its primal-dual version (PD) and the dual (D) as

$$\begin{array}{lll}(P) & \min_x & F(Kx)+G(x)\\ (PD) & \min_x\max_y & \langle Kx,y\rangle + G(x) - F^*(y)\\ (D) & \max_y & -(F^*(y)+G^*(-K^*y))\end{array}$$

where F^*, G^* being itself the convex conjugates of functions F and G. Assuming that there exists a solution $(\hat{x},\hat{y})$ one can specify that it must satisfy $K\hat{x}\in\partial F^*(\hat{y})$ and $-(K^*\hat{y})\in\partial G(\hat{x})$. For further details see [Chambolle and Pock, 2011]. Recall now that our relaxed energy was

$$\min_{\mathbf{u}\in\mathcal{S}}\sum_{i=1}^{k}\int_\Omega u_i\cdot f_i + \frac{\lambda}{2}\sum_{i=1}^{k}\int_\Omega g\cdot|\nabla u_i|$$

with differentiable indicators u_i. It is obvious that we can bring this energy into the primal form by setting $G(u)=\sum_{i=1}^k\int_\Omega u_i\cdot f_i$ and $F(\nabla u)=\frac{\lambda}{2}\sum_{i=1}^k\int_\Omega g\cdot|\nabla u_i|$. Furthermore, we also introduced the dual formulation of the TV-norm and ended up with

$$\min_{\mathbf{u}\in\mathcal{S}}\sup_{\xi\in\mathcal{K}}\sum_{i=1}^{k}\int_\Omega u_i\cdot f_i - \frac{\lambda}{2}\sum_{i=1}^{k}\int_\Omega \operatorname{div}\xi_i\cdot u_i$$

which resembles the (PD) form when setting $\langle Kx,y\rangle = \langle x,K^*y\rangle := -\sum_{i=1}^k\int_\Omega \operatorname{div}\xi_i\cdot u_i$ with K^* being the adjoint operator. Note that here the missing $F^*(y)$ is the characteristic

function for the convex dual space which can be omitted, since we already restrain $\xi \in \mathcal{K}$.

A huge advantage of the saddle formulation is the duality gap $\mathcal{G}(x, y)$ that can be evaluated for a given point (x, y). The gap tells how good a solution is by giving an estimate on how far the primal and dual energies differ from each other. Since the saddle point $(\hat{x}, \hat{y})$ has to lie between the primal and dual energy: $F(Kx) + G(x) \geq E(\hat{x}, \hat{y}) \geq -F^*(y) - G^*(-K^*y)$, the gap can be defined as

$$\mathcal{G}(x, y) := F(Kx) + G(x) + F^*(y) + G^*(-K^*y)$$

and has to be exactly zero at the saddle point. This can be used as a measure for solutions and also as an termination criterion during the optimization process.

4.2.2. Optimization scheme

Let us now look at optimizing our saddle point problem. Our energy for the Weighted-TV model has the following form (Equation 2.10):

$$\min_{\mathbf{u}\in\mathcal{S}} \sup_{\xi\in\mathcal{K}} \sum_{i=1}^{k} \int_\Omega u_i \cdot f_i - \lambda \sum_{i=1}^{k} \int_\Omega \text{div } \xi_i \cdot u_i$$

$$\mathcal{S} := \left\{ \mathbf{u} = (u_1, ..., u_k) \in BV(\Omega, [0,1])^k \;\middle|\; \sum_i u_i(x) = 1 \text{ a.e. } x \in \Omega \right\}$$

with the dual space $\mathcal{K}$ being one of the following:

$$\mathcal{K}_Z := \left\{ \xi = (\xi_1, ..., \xi_k) : \Omega \to \mathbb{R}^{2\times k} \;\middle|\; \sum_i |\xi_i(x)| \leq \frac{g(x)}{2} \text{ a.e. } x \in \Omega \right\}$$

$$\mathcal{K}_L := \left\{ \xi = (\xi_1, ..., \xi_k) : \Omega \to \mathbb{R}^{2\times k} \;\middle|\; \sqrt{\sum_i ||\xi_i(x)||^2} \leq \frac{g(x)}{2} \text{ a.e. } x \in \Omega \right\}$$

$$\mathcal{K}_C := \left\{ \xi = (\xi_1, ..., \xi_k) : \Omega \to \mathbb{R}^{2\times k} \;\middle|\; \left| \sum_{i_1 \leq i \leq i_2} \xi_i(x) \right| \leq \frac{g(x)}{2} \text{ a.e. } x \in \Omega \right\}$$

with Z, L, C denoting each dual space defined in [Zach et al., 2008, Lellmann et al., 2008, Chambolle et al., 2011]. The aim here is to find an optimum of the energy and since it is convex, there exists an optimal, global solution. In [Chambolle et al., 2011] the idea is to first discretize the data term on a $M \times N$ grid and to define the discrete divergence as the negative adjoint of the discrete gradient $\nabla^* = -\text{div}$, i.e. defining the gradient using forward differences while enforcing von-Neumann boundary conditions $\frac{\partial u}{\partial n} = 0$ and the divergence via backward differences and zero Dirichlet boundary conditions:

$$(\nabla u)^1_{i,j} := \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < M \\ 0 & \text{if } i = M \end{cases} \qquad (\nabla u)^2_{i,j} := \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}$$

$$(\text{div } \xi)_{i,j} := \begin{cases} \xi^1_{i,j} - \xi^1_{i-1,j} & \text{if } 1 < i < M \\ \xi^1_{i,j} & \text{if } i = 1 \\ -\xi^1_{i-1,j} & \text{if } i = M \end{cases} + \begin{cases} \xi^2_{i,j} - \xi^2_{i,j-1} & \text{if } 1 < j < N \\ \xi^2_{i,j} & \text{if } j = 1 \\ -\xi^2_{i,j-1} & \text{if } j = N \end{cases}$$

Eventually, the authors use an Arrow-Hurwicz -style algorithm that alternatively descents in the primal variable and ascents in the dual variable until convergence. Fixing each variable and deriving the energy yields two equations

$$\frac{\partial}{\partial u}E = f - \text{div } \xi \quad , \quad \frac{\partial}{\partial \xi}E = \nabla u$$

and gives the update scheme for the optimization. In the first step the primal variable $u^0 = 0$, the dual $\xi^0 = 0$ and an auxiliary variable $v^0 = 0$ (that will be used for an acceleration step) are initialized with zero. Then the algorithm iteratively runs the following scheme:

$$\begin{aligned} \xi_i^{n+1} &= \Pi_{\mathcal{K}}(\xi_i^n + \tau \cdot \nabla v_i^n) \\ u_i^{n+1} &= \Pi_{\mathcal{S}}(u_i^n + \rho \cdot \text{div } \xi_i^{n+1} - f_i) \\ v^{n+1} &= 2u^{n+1} - u^n \end{aligned}$$

with τ, ρ being two fixed time steps and $\Pi_{\mathcal{K}}, \Pi_{\mathcal{S}}$ being the projections onto the corresponding sets. It has been shown that the algorithm converges as long as $\tau\rho \leq \frac{1}{8}$. The advantage of that algorithm is that it can be easily parallelized and is efficient in terms of memory, since it requires every variable to be stored only once.

4.2.3. Projection of variables

In the algorithm a projection for the primal and dual variables has to occur. The projection $\Pi_{\mathcal{S}}$ of the primal variable u onto the canonical simplex $\mathcal{S}$ is quite easy and can be done in linear time [Chen and Ye, 2011]. The following Algorithm 1 from [Michelot, 1986] is computable without sorting and therefore better suited for a GPU implementation. The

$x \leftarrow u$
while $x \notin \mathcal{S}$ **do**
  $I \leftarrow \{i \mid x_i < 0\}$
  $\sigma \leftarrow (\sum_j x_j - 1)/|I|$
  $\forall j \notin I : x_j \leftarrow x_j - \sigma$
  $\forall i \in I : x_i \leftarrow 0$
end

Algorithm 1: Simplex projection $\Pi_{\mathcal{S}}(u) := \inf_{x \in \mathcal{S}} \frac{1}{2}||x - u||^2$

algorithm converges within k iterations, since the dimension $|I|$ is reduced by at least one unit in every step. To check if x is in the convex set after each iteration it suffices to assert that all entries are positive and their sum is "close enough" to 1.

If the dual space $\mathcal{K}$ is either defined as $\mathcal{K}_Z$ or $\mathcal{K}_L$, then the projection is fairly simple: it requires the computation of the variable's point-wise norm and then divide by it if it is greater than admitted. Mathematically, this means

$$\Pi_{\mathcal{K}}(\xi(x)) = \frac{\xi(x)}{\max(1, \frac{||\xi(x)||}{g(x)})}$$

with the norm $||\xi(x)||$ defined according to the respective authors.

When employing Chambolle's dual space $\mathcal{K}_C$ the projection $\Pi_{\mathcal{K}}$ of the dual variable is more involved and can be done using Dykstra's algorithm [Dykstra and Boyle, 1986]. The general idea is to write the convex set $\mathcal{K}_C$ as an intersection of simpler convex sets on which the projections are easier to compute. These projections are then performed in an alternating matter until the constraints of $\mathcal{K}_C$ are satisfied. Regard $\mathcal{K}_C$ as

$$\mathcal{K}_C := \bigcap_{1 \le i_1 < i_2 \le k} K_{i_1,i_2} \quad , \quad K_{i_1,i_2} = \left\{ (q_i)_{i=1}^k \in \mathbb{R}^{2\times k} \;\middle|\; |q_{i_1} - q_{i_2}| \le \lambda_{i_1,i_2} \right\}$$

then Algorithm 2, taken from [Chambolle et al., 2011], will compute the projection. The

$q \leftarrow \xi, a \leftarrow 0, \delta \leftarrow \infty$
while $\delta > \epsilon$ **do**
  $\bar{q} \leftarrow q$
  for $1 \le i_1 < i_2 \le k$ **do**
    $b \leftarrow q_{i_2} - q_{i_1} + a$
    $c \leftarrow (|b| - \lambda_{i_1,i_2})^+ (\frac{b}{|b|})$
    $q_{i_1} \leftarrow q_{i_1} + (b-a)/2$
    $q_{i_2} \leftarrow q_{i_2} - (b-a)/2$
    $a \leftarrow c$
  end
  $\delta \leftarrow ||q - \bar{q}||$
end

Algorithm 2: Dykstra projection $\Pi_{\mathcal{K}}(\xi) := \inf_{q\in\mathcal{K}_C} \frac{1}{2}||q-\xi||^2$

number δ is used to measure the numerical change in the variable and the algorithm aborts if it is smaller than a given tolerance ϵ. The method is computationally cumbersome, since it involves infinitely many iterations to converge and therefore dominates the segmentation process, especially when having more than a few labels (e.g. $k > 5$). In the referenced work the authors also give a modified version which is significantly faster when the surface tensions $\lambda_{i,j}$ satisfy certain triangular constraints.

4.3. Weighting term

The used energy from equation 2.10 is known as the Weighted-TV functional because it imposes a weighting g on the TV regularizer. This space-dependent weighting can be regarded as the amount of flow that is permitted to pass at that position, i.e. the neighborhood's strength of influence on the segmentation. A low value of $g(x)$ means that at position x the data term $f(x)$ will dominate during the optimization and vice versa. Apparently, a suitable choice of g can also drive the segmentation process towards more favorable solutions.

Usually, the weighting term is computed by using the image gradient of the colored image or its grayscale version. Using the image gradient makes sense when the objective is

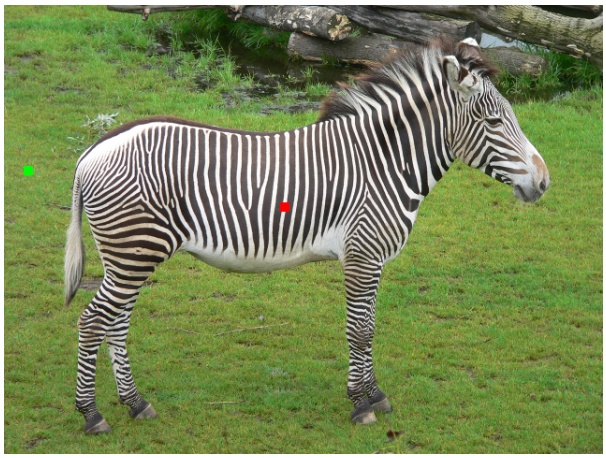
to find segmentations that are based on unsupervised color differences (see, for example, [Chambolle and Pock, 2011]). When using a data term that is based on arbitrary information, thus yielding segmentations where a region is possibly supposed to cover differently colored areas, the image gradient can degrade the result. Since the image is supplied with scribbles, the weighting term should reflect that additional information in its computation.

4.3.1. OLDA color space

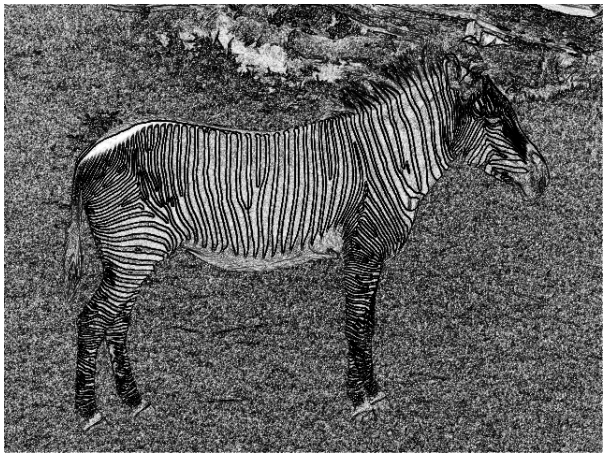
One improvement when having colored in-class structure together with supervised information is to transform the color space and compute the image gradient in the new (not necessarily lower-dimensional) space. In Figure 4.8 one can see a zebra, its image gradient and the image gradient computed in the new one-dimensional OLDA color space. The original image gradient is problematic because it creates high values inside the zebra, although these areas do belong to the same object. In the transformed space the distance between black and white pixels is reduced leading to a smoother appearance inside the body.

4.3.2. Texture space

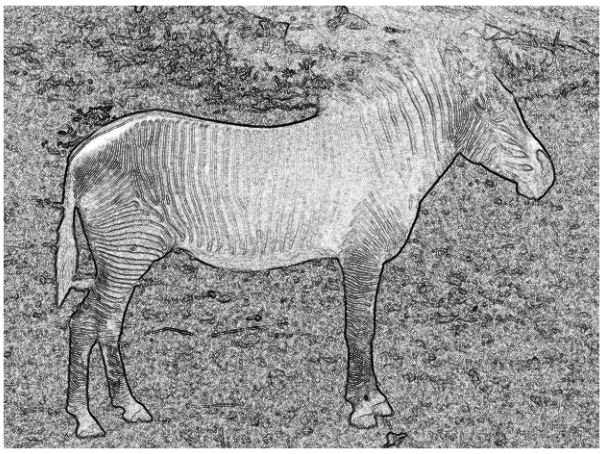
A second idea is to employ the texture information by computing the gradient on it. This is especially suited for noisy images where there is no apparent color structure and where even a color transformation cannot add much to the discriminance. Of course, the texture space can also be transformed by using the scribbles and computing the gradient in the transformed and lower-dimensional OLDA subspace to further improve the weighting. In Figure 4.9 it shows that using the texture gradient strongly reduces the in-class noise and improves the separation of foreground and background.



(a) The zebra image



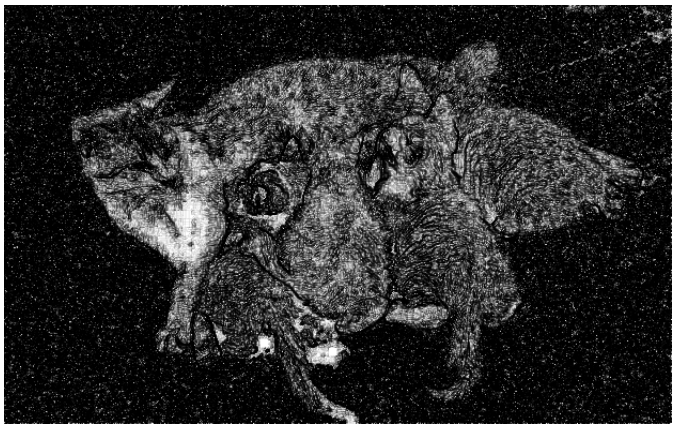
(b) Its color gradient g_1



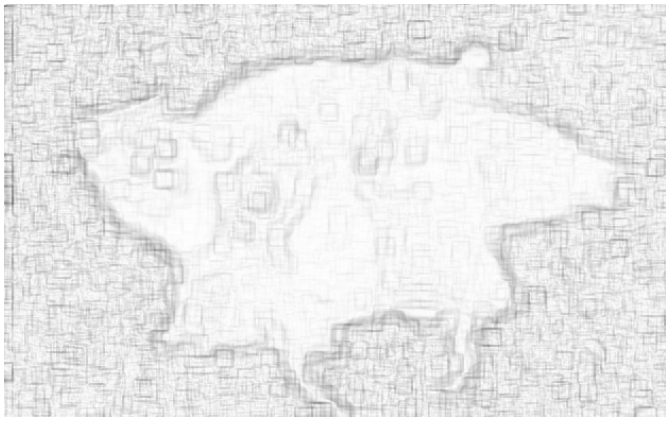
(c) Its OLDA color gradient g_2

Figure 4.8.: A zebra image (a) taken from the internet and its color gradient (b) computed as $g_1 = e^{-\eta|\nabla I|}$ with an additional parameter η. In (c) the gradient of the one-dimensional OLDA color subspace was used. The parameter was $\eta = 0.15$.

(a) The cat image from IcgBench



(b) Its color gradient g_1



(c) Its texture gradient g_2

Figure 4.9.: The cat image (a) taken from IcgBench and its color gradient (b) computed as $g_1 = e^{-\eta|\nabla I|}$ with an additional parameter η. In (c) the gradient of the texture space $g_2 = e^{-\eta|\nabla T|}$ was used. The parameter was $\eta = 0.2$.

Part III.

Evaluation and Outlook

5. Evaluation

The computational model was implemented using C++ and the Nvidia CUDA framework. The evaluation was done on a mid-class notebook with a Core i5 processor and a GT330M graphics card. Calculations involving the LDA transformation, the anisotropy information and the wavelet coefficients have been done on the CPU whereas the computation of the kernel density estimate, the segmentation and the texture statistics have been taken onto the GPU, since they are inherently parallel.

5.1. Estimation and segmentation

The data term is computed by a kernel density estimate which yields a likelihood for each label i. This means that we receive multiple functions $f_i(x)$ and feed this into the segmentation model. See Figure 5.3 for a visualization of the data term. It shows that especially for locations where the scribbles for a specific label are positioned the likelihoods tend to be very weak for every other label. For example, note that the black snake-like shape in f_1 which tells that the red label is very unlikely to be there, also happens to coincide with the green scribbles' locations.

The second part of the energy, the regularizer term, measures the length of the boundary and is steered by the weighting parameter λ. With a small value the importance of the boundary is neglected which often leads to jagged interfaces between labels. The higher the value the smoother those interfaces become. See Figure 5.2 for visuals. Note that with an increasing λ the boundary shortens. From a certain value on the green label would vanish for the given segmentation instance, since its length would be the dominating term in the energy and therefore completely done away with to reach the optimum.

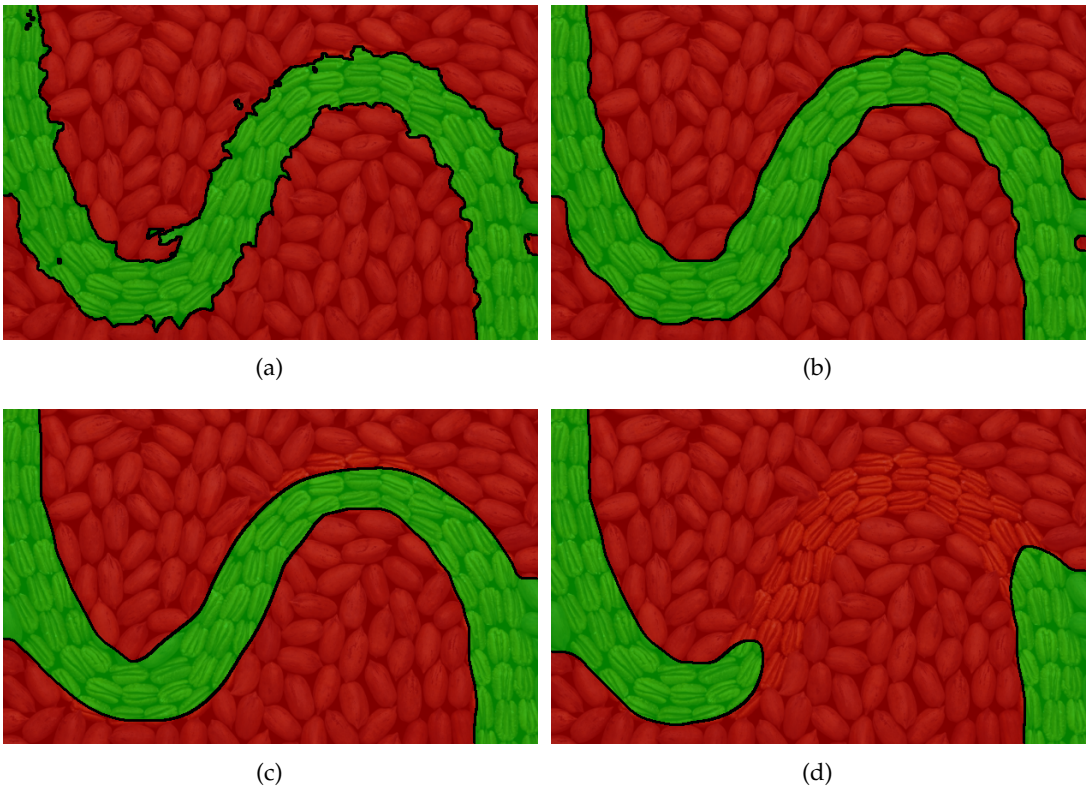
An example for the estimation and segmentation for the multi-label case is given in Figure 5.4. Note again here that every estimate is black for locations where scribbles of other labels have been positioned. Another interesting fact is that the estimate reflects the image colors to a certain extent due to the color kernels' working. Both mentioned aspects show how the fusion of spatial and color information shapes the estimated distribution.

Since we are in an interactive setting, we want the method to be responsive enough for the user. The only important numbers here are the times for the density estimation and the segmentation because the image features need to be computed only once in the beginning. In Figure 5.1 one can see some runtimes for the KDE. The complexity of the estimation depends linearly on the number of scribbles and the computation time also increases with the dimensionality of the kernel space. Here, the dimensions were 2 for the space, 3 for the color and 6 for the texture. The numbers given in the figure reflect the mentioned growth

in computation time and also show that for a moderate amount of scribbles the method is suited for real-time applications, even on average GPU hardware.

# Scribbles	Space (2)	Space+Color (5)	Space+Color+Texture (11)
100	150	198	234
350	227	328	597
700	323	687	1265
1050	548	956	1846

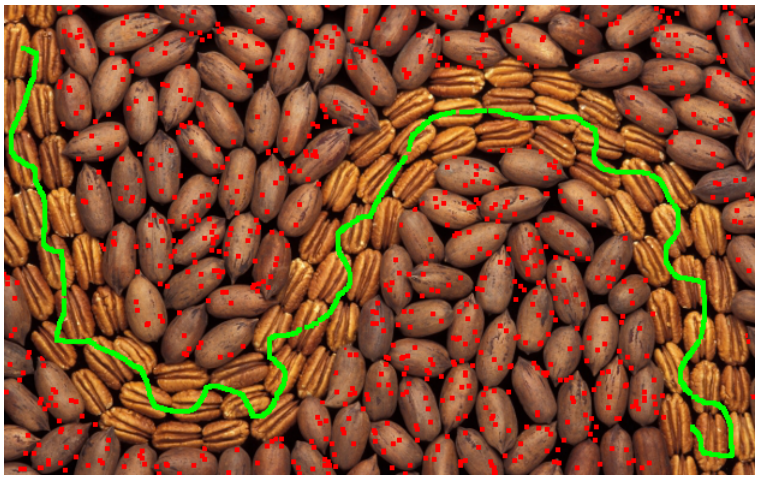
Figure 5.1.: Runtimes in milliseconds for KDEs on an image of size 625x391. The numbers in parentheses denote the dimensionality of the kernel space. With more dimensions and scribbles the computation time rises. Still, for a reasonable amount of scribbles the method remains suitable for real-time applications.



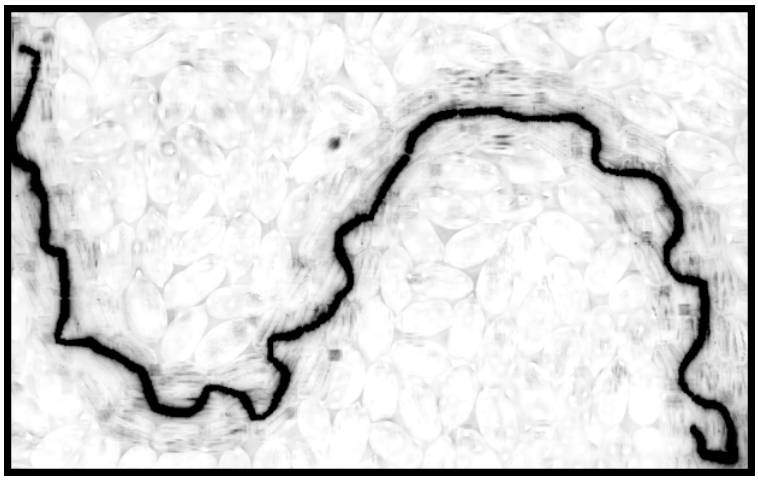
(a) (b)

(c) (d)

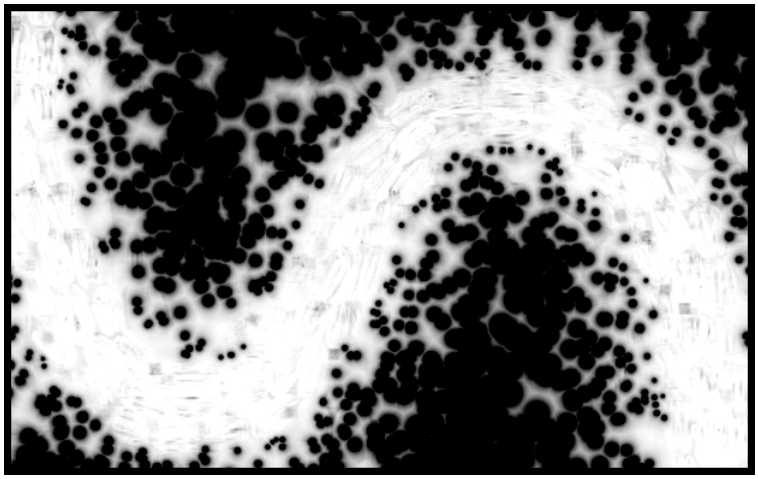
Figure 5.2.: Comparison of segmentation results using different values for the regularization where (a) has $\lambda = 1$, (b) has $\lambda = 10$, (c) has $\lambda = 75$ and (d) has $\lambda = 100$. With a weak regularization the interface exhibits sharp edges since the data term dominates the optimization. With a higher λ the boundary becomes smoother until it starts to disappear.



(a) Input



(b) Estimate for red



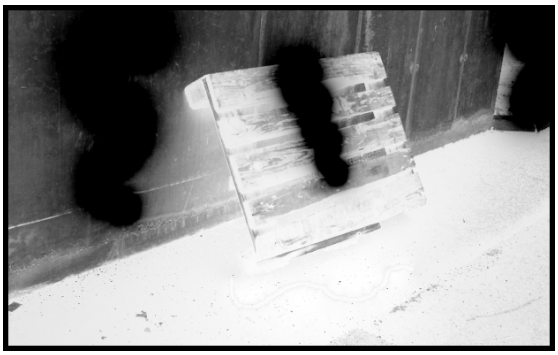
(c) Estimate for green

Figure 5.3.: Input image with two classes and corresponding data terms. Figure (b) shows the term f_1 and in (c) one can see the term f_2 with brighter values being higher. The used parameters for the space-color KDE were $\sigma = 0.2$ and $\alpha = 2$.

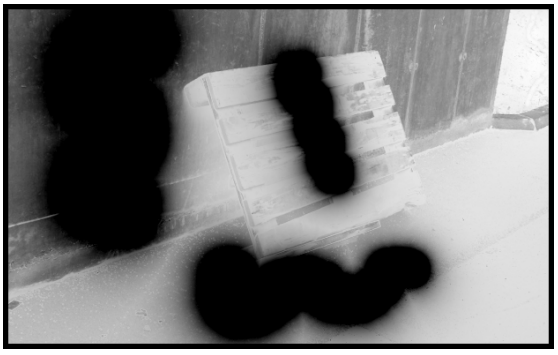
(a) Input

(b) Output

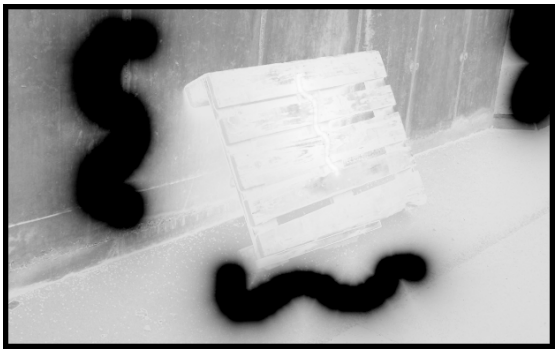
(c) Estimate for red



(d) Estimate for green



(e) Estimate for blue



(f) Estimate for yellow

Figure 5.4.: Input image with four classes and corresponding data terms. The used parameters for the space-color KDE were $\sigma = 0.15$ and $\alpha = 2$. One can clearly identify the influence of the spatial component by the dark shapes in the estimate. The color kernel also influences the density, since the data term resembles the colors of the original image to a certain extent.

5.2. Different dual spaces

We consider three different dual spaces for the energy formulation, namely $\mathcal{K}_Z$ from Zach et al., $\mathcal{K}_L$ from Lellmann et al. and $\mathcal{K}_C$ from Chambolle et al.:

$$\mathcal{K}_Z := \left\{ \xi = (\xi_1, ..., \xi_k) : \Omega \to \mathbb{R}^{2\times k} \;\middle|\; \sum_i |\xi_i(x)| \;\leq\; \frac{g(x)}{2} \;\text{ a.e. }\; x \in \Omega \right\}$$

$$\mathcal{K}_L := \left\{ \xi = (\xi_1, ..., \xi_k) : \Omega \to \mathbb{R}^{2\times k} \;\middle|\; \sqrt{\sum_i ||\xi_i(x)||^2} \;\leq\; \frac{g(x)}{2} \;\text{ a.e. }\; x \in \Omega \right\}$$

$$\mathcal{K}_C := \left\{ \xi = (\xi_1, ..., \xi_k) : \Omega \to \mathbb{R}^{2\times k} \;\middle|\; |\sum_{i_1 \leq i \leq i_2} \xi_i(x)| \leq\; \frac{g(x)}{2} \;\text{ a.e. }\; x \in \Omega \right\}$$

The former two are rather simple definitions, since they only constrain a point-wise vector norm. The latter is more involved because it enforces a label-wise paired calibration for every point which is computationally demanding. One problematic aspect of the paired calibration is the exponential growth of complexity when the number of labels increases. See Figure 5.5 for a runtime comparison of the three energies for 1500 iterations. It can be seen that both Lellmann and Zach are nearly equal in their runtimes which in turn increase slowly with a growing label count. The Chambolle energy on the other hand always takes more time due to the slow convex projection constraints. Another aspect is the visual quality of the results that are shown in Figure 5.6. While it may give mathematically tighter solutions for multi-label problems, the Chambolle energy does not yield visually superior segmentations. Although Chambolle's results tend to look a bit smoother, this can be easily remedied for the other energies by adding further scribbles and increasing the regularizer weighting. Thus, one should resort to one of the easier energies because they are both fast to compute and produce comparative visual output.

# Labels	Lellmann	Zach	Chambolle
2	1.50	1.44	1.77
3	2.33	2.33	2.87
4	3.12	3.16	4.33
5	4.17	4.21	6.41
6	5.70	5.46	8.44
7	6.62	6.70	11.05
8	7.61	7.25	14.27
9	8.36	8.30	18.18

Figure 5.5.: Duration of 1500 iterations in seconds for different dual spaces. While the runtimes for the Lellmann and Zach spaces grow linearly in the number of labels with negligible differences, runtimes for the Chambolle space increase rapidly.

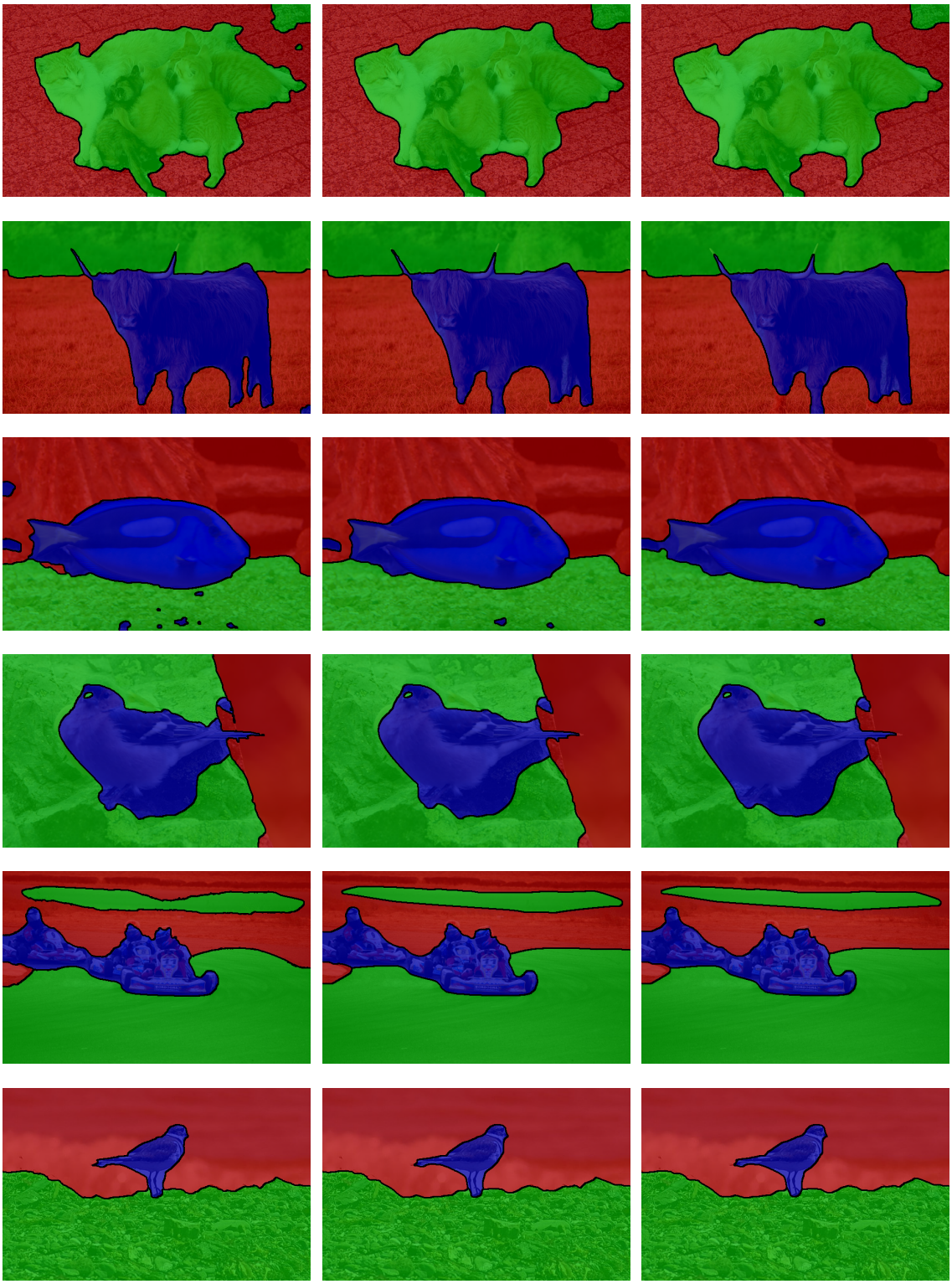


Figure 5.6.: Comparison of segmentation results using different dual spaces with same data term and $\lambda = 10$. From left to right: $\mathcal{K}_Z$, $\mathcal{K}_L$, $\mathcal{K}_C$.

5.3. Texture kernel

The idea of a texture kernel is to supplement the spatially-varying distribution with textural image information. The problem with many real-life images is the already mentioned phenomenon of changing color patterns on the same object which may lead to wrong estimations when resorting only to color distributions. To convey the benefit of using texture some instances from the Graz IcgBench dataset [Santner et al., 2009] were used. This dataset consists of 262 images together with user scribbles which closely resembles our given scenario. The instances from the dataset were taken and segmented without and with the additional usage of textural clues as seen in Figures 5.7 and 5.8. For the spatially-varying color distributions the bandwidths were chosen manually for the best visual result. The bandwidth for the texture kernel and the window size was chosen accordingly. It shows that adding texture information can tremendously improve the visual results.

Basically, we can distinguish two cases where this information helps. Firstly, when dealing with highly-textured objects of similar color it is very important to include texture into the density estimation. This can especially be seen for the images in Figure 5.7. When only relying on space and color the estimation is not able to properly cope with the change in intensity patterns. The spider and the turtle, for example, are closely matching the background color, rendering a proper segmentation impossible without texture information. The second case, which is highlighted in Figure 5.8, deals with instances where input is either sparse or very dense. When having only a few and/or badly positioned samples the spatial kernel cannot bring enough information into the estimation and the KDE is then mostly dependent on the color kernel. When plugging the texture kernel into the estimation this problem is weakened because the textural information compensates for that to some extent. This is especially visible in the two bird images. On the contrary, dealing with a huge amount of scribbles may lead to a dominant spatial component in the KDE where the role of color is undermined. Both the leopard image as well as the people image show exactly that. Here, texture improves the segmentation by relativating the overall influence of the space kernel in the estimate.

To further show the advantage of our model we look at some images from the Berkeley Segmentation Database [Martin et al., 2001] in Figure 5.9 and apply the same methods there. Since this benchmark only comprises images, we supply some input and observe again the segmentation output when using the KDE without and with texture information. Again, we can see that there are cases where texture drastically improves the segmentation. There are some images, for example the woman, the crocodile or the leopard, where a proper segmentation is nearly impossible without additional texture information. The crocodile and the leopard share a lot of their coloring with the background and are only betrayed by their surface patterns. Although the background in the woman image differs in terms of color, texture is the most important clue for the separation. In other cases texture merely helps in finding sharper object bounds. Note that most problems could be tackled with using only space and color information while omitting texture. The downside is that the amount of scribbles that have to be correctly placed increases. Thus, it is more convenient to regard all three pieces of information because it means less hassle for the user during the input process.

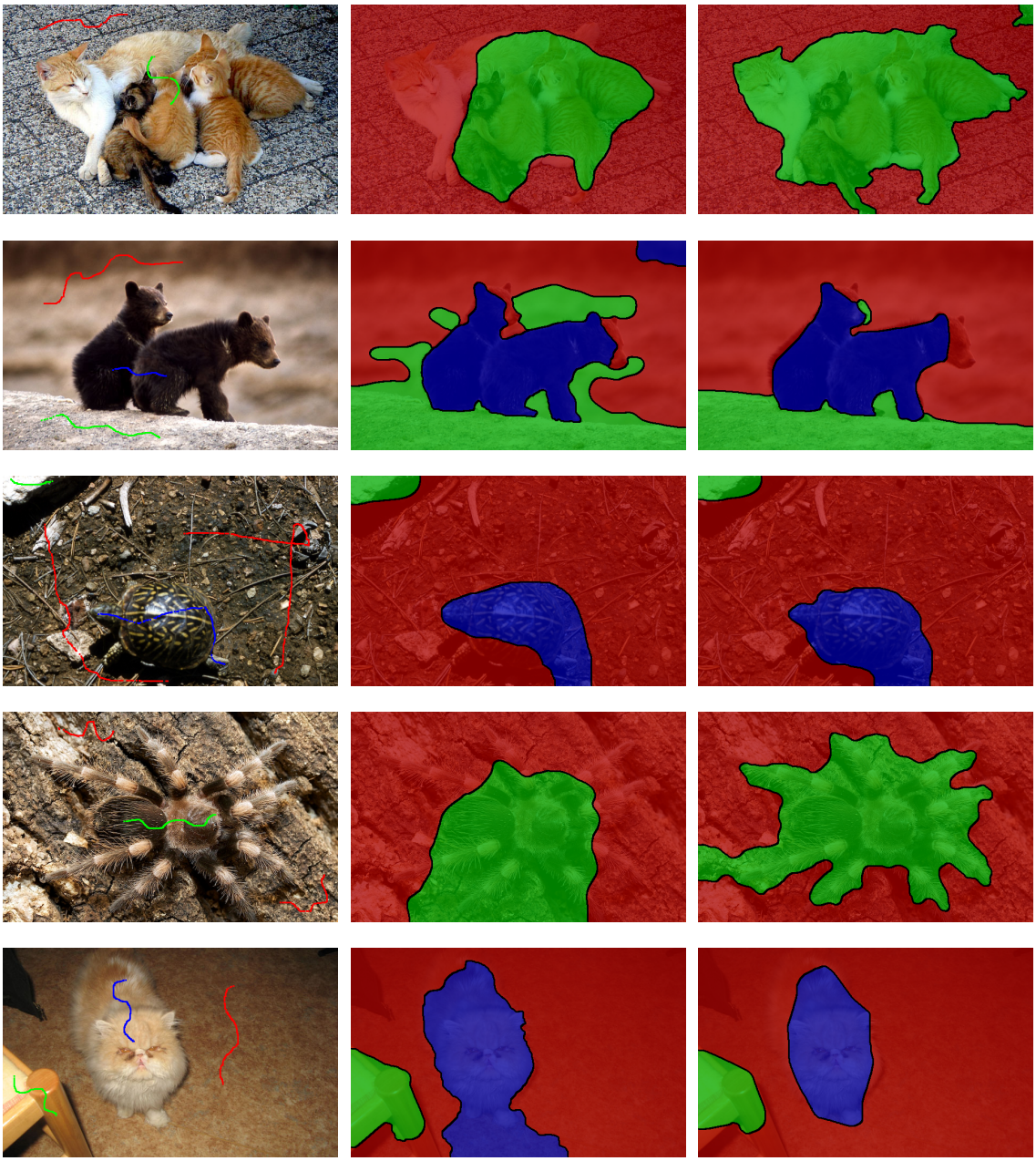


Figure 5.7.: Examples from the Graz benchmark. The left column shows the input, the middle column the segmentations when using only space and color information and the right column shows the result when adding the texture kernel.

To assess the quality of additional texture information for the segmentation in a more comparable framework we apply our work to the whole Graz IcgBench dataset. It is important to note that although this dataset provides us with instances of images and user input, thus allowing comparisons, it defies the idea of an interactive setting. The main advantage

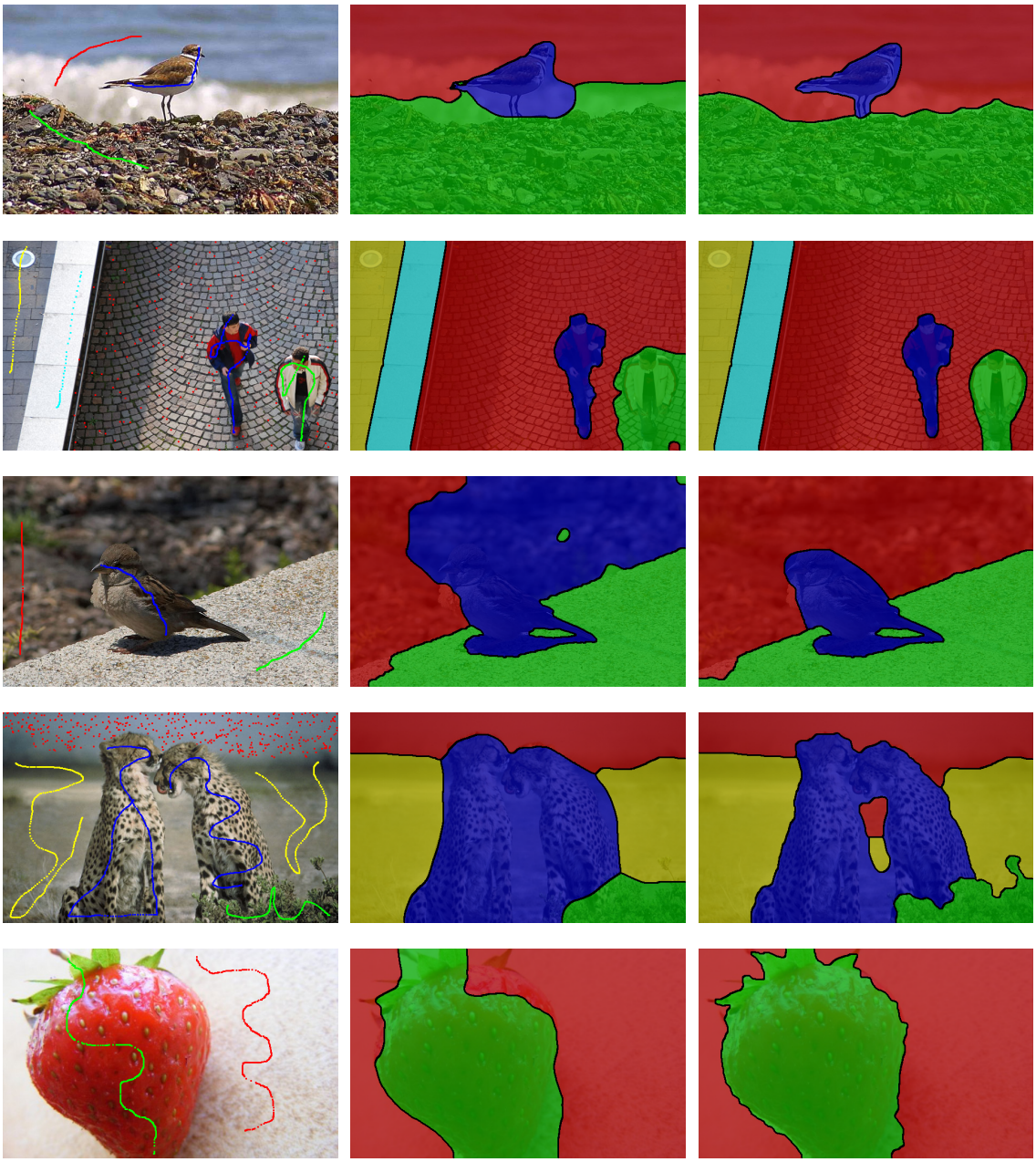


Figure 5.8.: Examples from the Graz benchmark. The left column shows the input, the middle column the segmentations when using only space and color information and the right column shows the result when adding the texture kernel.

of real-time capable and interactive methods is the incremental update procedure. Intermediate results are presented to the user and the input improves in turn by adding further scribbles. Therefore, results on this benchmark can only provide a limited amount of information in the presented context .

To measure segmentations, the authors use a so-called dice score, defined as

$$dice(\Omega_i, \overline{\Omega_i}) := \frac{2|\Omega_i \cap \overline{\Omega_i}|}{|\Omega_i| + |\overline{\Omega_i}|}$$

which computes the ratio between the overlap of the computed region Ω_i with the ground-truth region $\overline{\Omega_i}$ and the sum of their areas. Since that score always ranges between 0 and 1, we compute the arithmetic mean over the dice scores for each label i and end up with an accuracy measure for arbitrarily many labels. Figure 5.11 shows some results on that benchmark when employing different methods. Grady and Santner vary their input by increasing their brush size and by this, constantly improve their accuracy. This will not be done in this work because it distorts comparability. The notion of enlarging the brush size to simulate more user input is completely misleading, since in practice the user would place new scribbles at hitherto unused locations instead of thickening already given scribbles. Also, with bigger brush sizes the chosen method becomes less important since the input covers larger areas in the image to a point where segmentation becomes virtually superfluous (see Figure 5.10 for two actual Graz benchmark examples).

The results show that simply using the spatial information of the scribbles for the density is not enough for good results. Together with color information the proposed method beats Grady et al. as well as Santner et al. with different color spaces. It is also interesting to see that by supervised transformation of the color space, therefore drastically boosting the descriptive power of that component, we can achieve the same score as Santner with mixed features. Note that they use a brush size of 5 together with color and LBP texture information with a high-dimensional space whereas we keep the brush size constant and only use a 5-dimensional space for our computation. Lastly, taking also texture into our KDE and transforming it with OLDA, totaling 10 dimensions, we are able to outperform the state-of-the-art while still keeping the brush size constant. The additional 0.3% indicate that the method performs well when only considering space and color for the KDE and that there is only a subset of images where texture was essential for a good segmentation. Another point is that although the benchmark consists of relatively many instances, a lot of them are supplied with a huge amount of scribbles, rendering color and texture less important in comparison to information from the spatial domain.

5.4. Automatic parameter estimation

Employing KDE for the data term always exhibits the problem of finding suitable bandwidth parameters for the space, color and texture kernel. Removing the number of free parameters is always desirable and the notion of automatically estimating the bandwidths for the color and texture kernels was proposed. Again, we will have a look at instances from the Graz benchmark as well as images from the Berkeley dataset together with supplied input and compare the output of manually chosen bandwidths for color and texture against automatically determined kernel bandwidths.

In Figure 5.12 the results for some of the Graz benchmark instances are presented. Remarkably, the segmentations provided by the automatic estimation are similar to the ones

with manual bandwidths. In some cases, the results were even better by having sharper boundaries, e.g. the vegetation in the bird image or the helicopter in the desert image. The automatic estimation pays tribute to the fact that each label exhibits different variances in its color and texture information. In the bird image both the red label and the blue label have little color variety and therefore, were separated more tightly due to small bandwidths.

We can see the results on the Berkeley images in Figure 5.13. Here again it shows that the method was able to determine appropriate bandwidths and to produce results that are similar to those with manually set bandwidths. In the first and last image the automatic estimation was even able to outperform the standard approach by reflecting the objects' individual variance in color and texture and thus, producing more fitting boundaries. The boy image conveys the one problematic aspect of this approach. Although the result seems good enough (especially around the body where it is even better than with the standard method), it fails at the top rim of the hat. There, no green scribble was placed at the bright rim, leading to a very small color variance for the dark hat, i.e. for that label, and therefore to a suboptimal segmentation.

During the evaluation it became evident that for the automatic estimation it was beneficial to use smaller window sizes for the texture statistics and generally, to decrease the spatial bandwidth. This is due to the fact that the automatically determined parameters for color and texture were very tightly chosen. Conclusively, the automatic estimation works quite well and really helps the user by hiding free parameters while producing comparable visual output. It was also seen that by placing the scribbles in an appropriate manner and increasing their number the bandwidths became more fitting because the objects' appearances in terms of color and texture were captured more distinctively.

5.5. Anisotropic spatial information

One interesting aspect is the usage of anisotropic space information for the kernel density estimation. The argumentation is that the user usually positions the input along the objects' shapes and often enough these shapes have a dominant direction. By computing this direction we may influence the estimation in a positive way. See Figure 5.14 for visual results on some Graz benchmark instances.

The images show that anisotropic information can indeed have a positive influence on the segmentation, especially for elongated objects where it provides more fitting contours. Although the dice score for all of the presented images increased, it shows that sometimes the anisotropic information can also lead to mistakes, especially for fine details, e.g. the insect's leg. It again supports the notion that the presented method may help the user with the input by reducing the amount of scribbles which are necessary to place but it is still of importance that scribbles are positioned in a meaningful manner. Still, directional dependence in the space kernel presents itself as a useful addition to the model.

Figure 5.9.: Examples from the Berkeley benchmark. The left column shows the input, the middle column the segmentations when using only space and color information and the right column shows the result when adding the texture kernel.

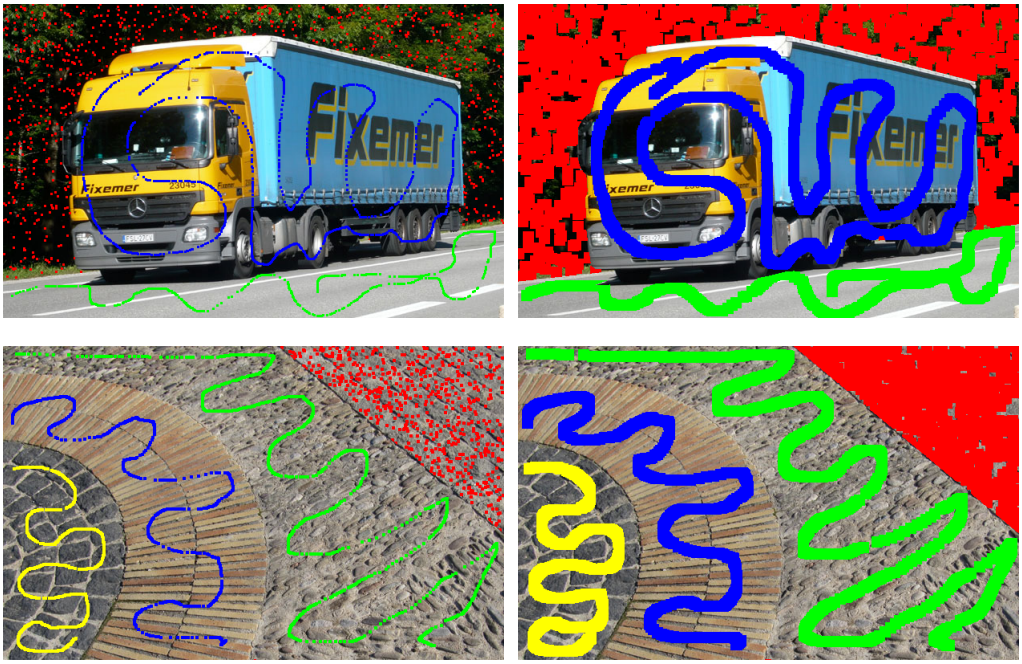


Figure 5.10.: Two instances from the Graz benchmark using a brush size of 2 (left) and a brush size of 13 (right). It is obvious that using large brush sizes improves accuracy but runs contrary to meaningful comparisons. In the given examples the red regions are practically already segmented by the supplied input.

Method	Dim	Brush	Score
[Grady, 2006]	3	13	0.855
[Santner et al., 2010] ,RGB	3	-	0.877
[Santner et al., 2010] ,HSV	3	-	0.897
Space	2	1	0.739
Space + Color	5	1	0.900
[Santner et al., 2010] ,CIELab+LBP	21	5	0.917
Space + OLDA Color	5	1	0.917
Space + OLDA Color + OLDA Texture	10	1	0.92

Figure 5.11.: Results on the Graz dataset using different methods and options.

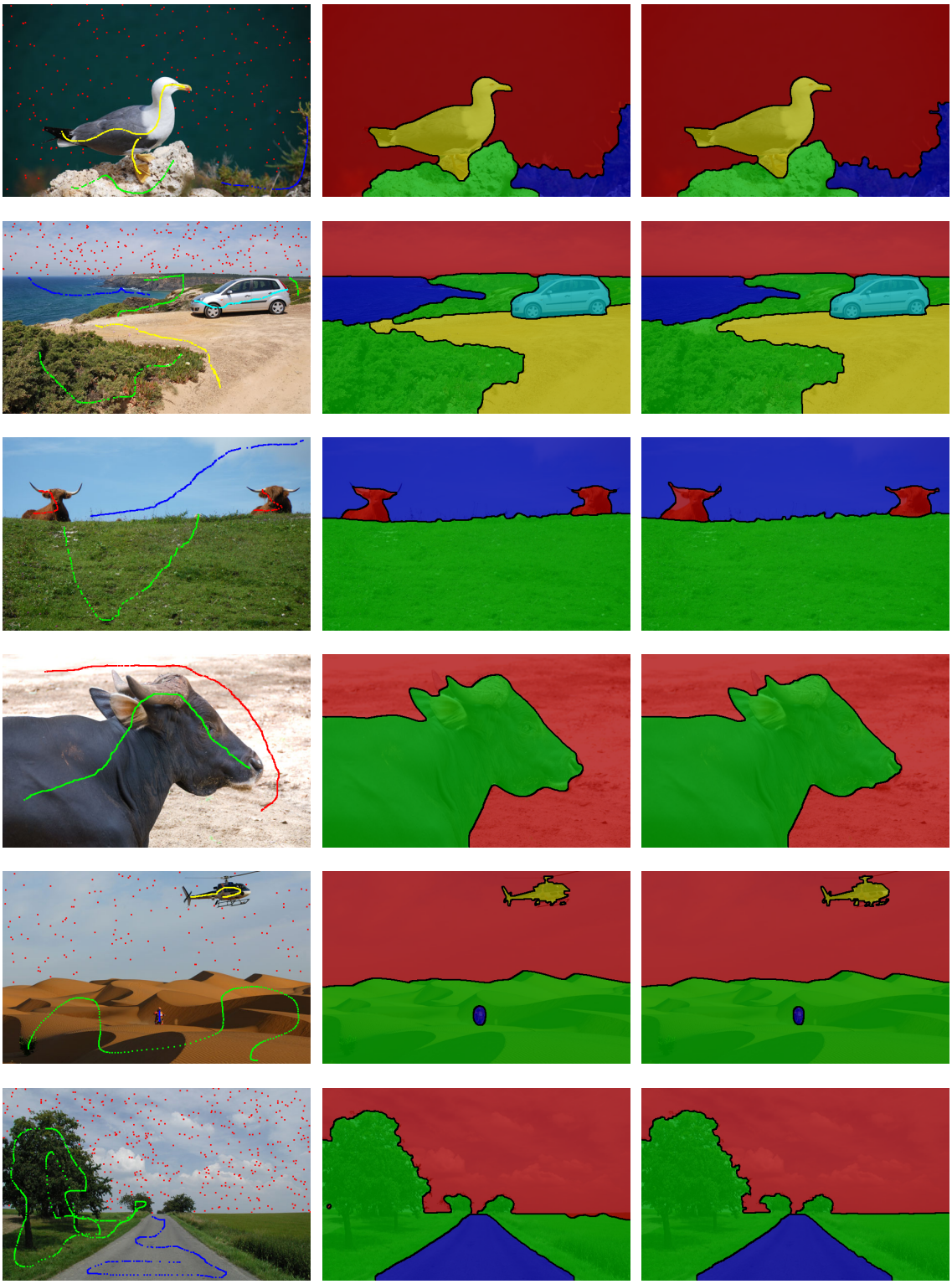


Figure 5.12.: Examples from the Graz benchmark. The left column shows the input, the middle column the segmentations with manually set bandwidths and the right column shows the result when estimating them automatically.

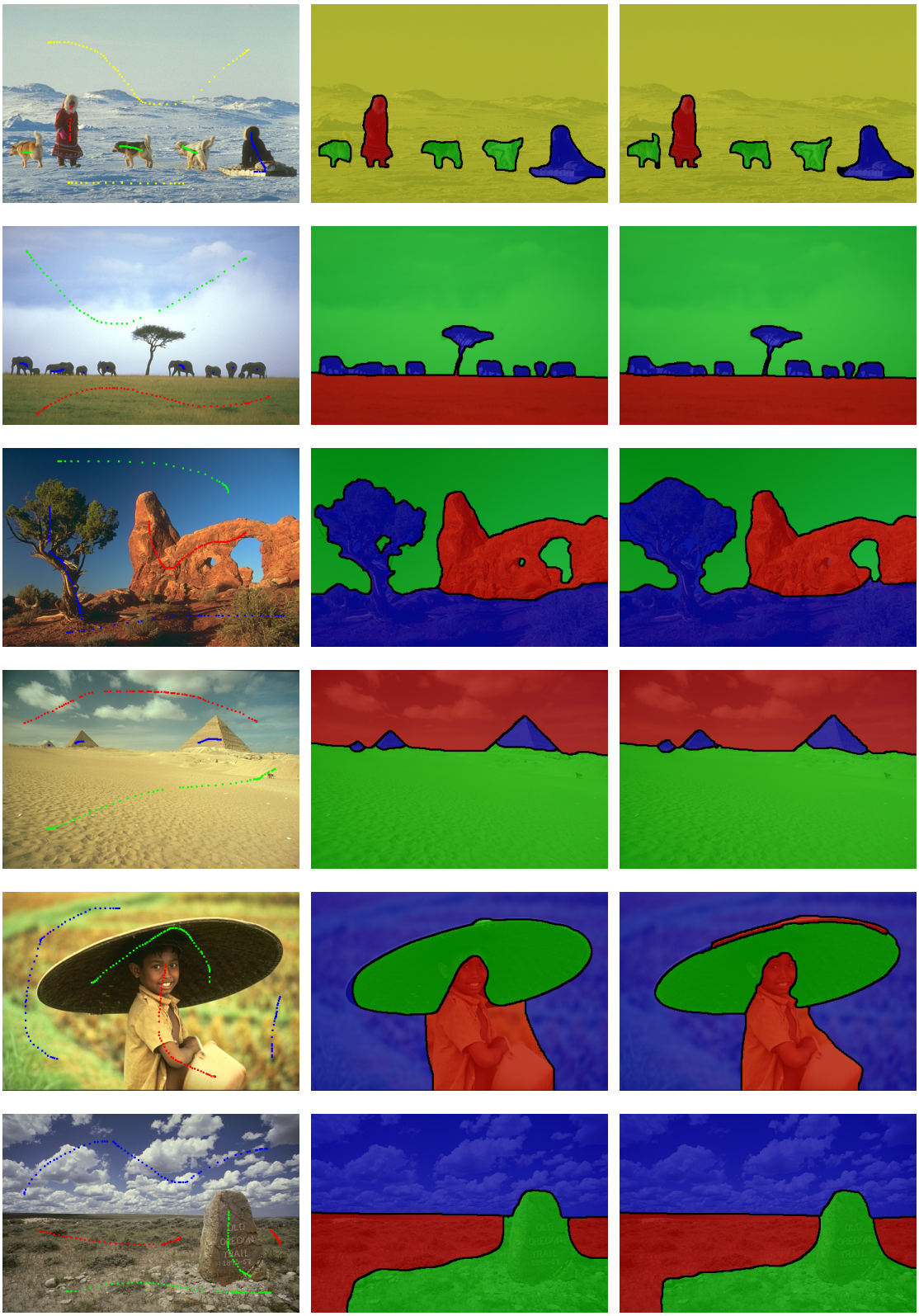


Figure 5.13.: Examples from the Berkeley benchmark. The left column shows the input, the middle column the segmentations with manually set bandwidths and the right column shows the result when estimating them automatically.

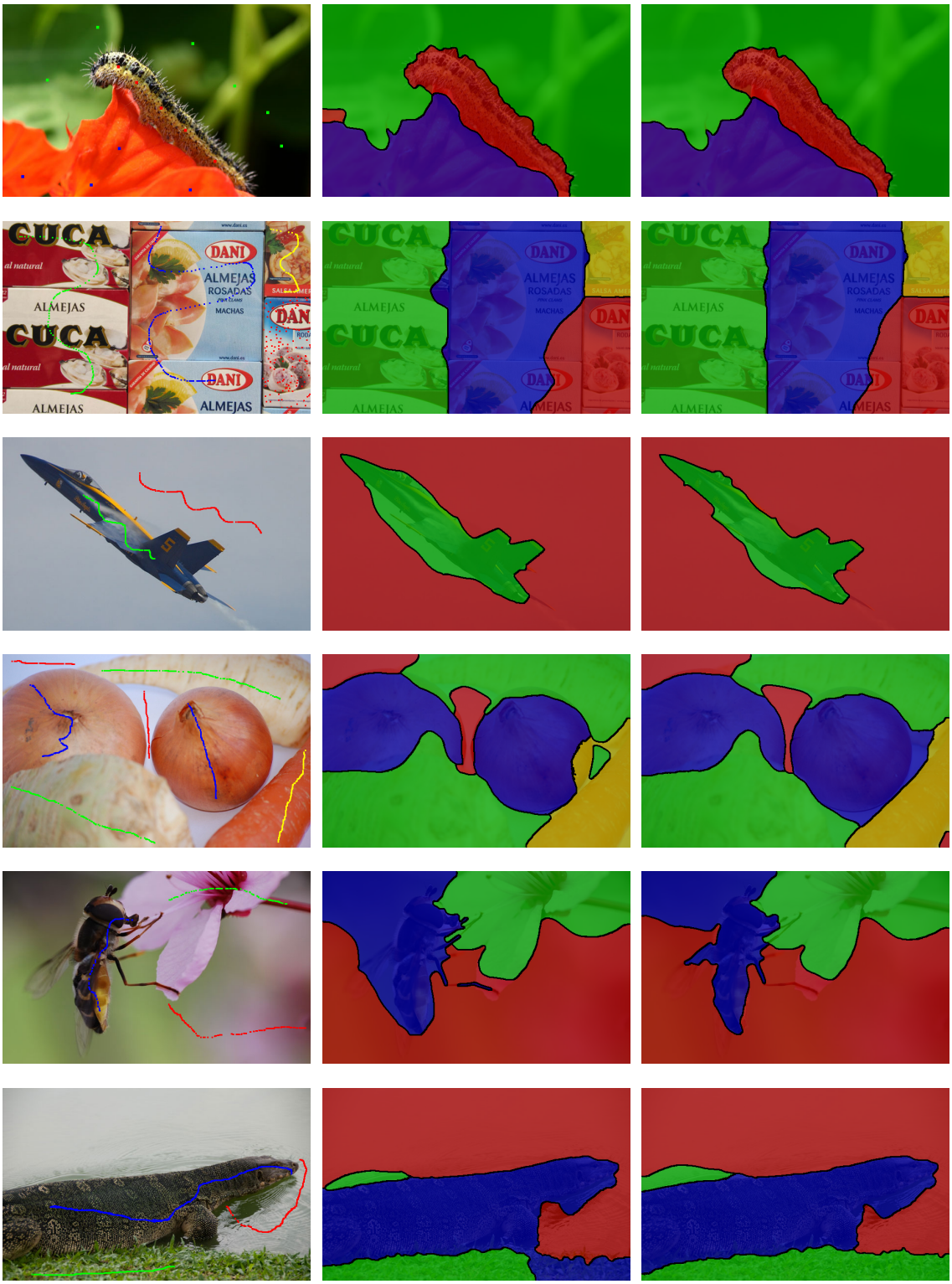


Figure 5.14.: Comparison of isotropic vs. anisotropic color distributions. First row shows the input, second row the segmentation using isotropic space kernels and the last row shows segmentations using anisotropic space kernels.

6. Summary and Outlook

This work employed a variational approach towards multi-label image segmentation where the data term was computed with kernel density estimates over space, color and texture information. Another aspect was the interactive setting in which these various pieces of information have been taken from user input in the form of scribbles. For the space kernel an alternative, anisotropic version was proposed that used a Mahalanobis distance measure. A texture kernel was introduced into the model that employed a discrete wavelet transform of the image together with a measure which was based on the wavelet coefficients' statistics. The transformation of the color and texture space with a subspace method was proposed together with a new variant for setting the bandwidth parameters by estimating them automatically from input. Lastly, two ideas of a possible definition for the TV weighting function were briefly presented.

The evaluation showed that the proposed method, implemented on a GPU, is suitable for real-time applications and that texture information helps in discerning objects more reliably. It also showed that employing supervised transformations for the color and texture space, here (O)LDA, improved the accuracy by a fair amount. This was all demonstrated on two datasets, namely the Berkeley Segmentation benchmark and the Graz benchmark. For the latter, we presented the dice scores for different approaches and proved the superiority of our proposed model in comparison to related work, establishing the new state-of-the-art. We also evaluated the automatic estimation of bandwidth parameters for the color and texture kernels and the notion of anisotropic spatial information. The results conveyed the beneficial influence of directional dependence in the space kernel especially for elongated objects. Furthermore, the automatic estimation presented itself as a viable alternative to manually determining the bandwidth parameters. The difference in the results were mostly indiscernible while at the same time the user was ridden from free parameters.

There is, of course, a variety of ideas for future work. Firstly, one could look into alternative texture descriptors and associated measures for the kernel estimate. While wavelets showed promising results, the search for a good wavelet base alongside appropriate scale dimensions and window sizes introduces an unnecessarily high degree of freedom. Secondly, it would be convenient to find an automatic estimation for the spatial bandwidth by sampling around user scribbles or inferring it through stochastic means over scribble positions. By this, the user would be completely freed from the underlying model's workings and would only need to tweak the regularization parameter for the smoothness of results.

Bibliography

[Arias et al., 2011] Arias, P., Facciolo, G., Caselles, V., and Sapiro, G. (2011). A Variational Framework for Exemplar-Based Image Inpainting. *International Journal of Computer Vision*, 93(3):319–347.

[Awate et al., 2006] Awate, S., Tasdizen, T., and Whitaker, R. (2006). Unsupervised texture segmentation with nonparametric neighborhood statistics. In *Proc. of ECCV, LNCS 3952*, pages 494–507.

[Beveridge, 2001] Beveridge, J. (2001). The Geometry of LDA and PCA Classifiers Illustrated with 3D Examples. Technical report, Colorado State University, Computer Science Department.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[Botev et al., 2010] Botev, Z. I., Grotowski, J. F., and Kroese, D. P. (2010). Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957.

[Boykov et al., 2001] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

[Boykov and Jolly, 2001] Boykov, Y. Y. and Jolly, M.-P. (2001). Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In *Proc. of International Conference on Computer Vision*, pages 105–112.

[Brodatz, 1966] Brodatz, P. (1966). *Textures: A Photographic Album for Artists and Designers*. Dover Publications Inc., New York.

[Busch and Boles, 2002] Busch, A. and Boles, W. W. (2002). Texture classification using multiple wavelet analysis. *Proc. of Digital Image Computing Techniques and Applications*, 6:341–345.

[Chambolle et al., 2008] Chambolle, A., Cremers, D., and Pock, T. (2008). A convex approach for computing minimal partitions. Technical report, University of Bonn, Computer Science Department.

[Chambolle et al., 2011] Chambolle, A., Cremers, D., and Pock, T. (2011). A convex approach to minimal partitions. Technical report, École Polytechnique, Technical University Munich, Graz University of Technology.

[Chambolle and Pock, 2011] Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145.

[Chan and Esedoglu, 2006] Chan, T. and Esedoglu, S. (2006). Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66:1632–1648.

[Chan et al., 1999] Chan, T. F., Golub, G. H., and Mulet, P. (1999). A Nonlinear Primal-Dual Method for Total Variation-Based Image Restoration. *SIAM Journal on Scientific Computing*, 20(6):1964–1977.

[Chen and Ye, 2011] Chen, Y. and Ye, X. (2011). Projection Onto A Simplex.

[Cremers and Grady, 2006] Cremers, D. and Grady, L. (2006). Statistical priors for efficient combinatorial optimization via graph cuts. In *Proc. of ECCV, LNCS 3953*, pages 263–274.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. of Computer Vision and Pattern Recognition*, pages 886–893.

[Dykstra and Boyle, 1986] Dykstra, R. L. and Boyle, J. P. (1986). A method for finding projections onto the intersection of convex sets in Hilbert spaces. *Lecture Notes in Statistics*, 37:28–47.

[Efros and Leung, 1999] Efros, A. A. and Leung, T. K. (1999). Texture Synthesis by Non-parametric Sampling. In *Proc. of International Conference on Computer Vision*, pages 1033–1038.

[Emrich et al., 2010] Emrich, T., Graf, F., Kriegel, H.-P., Schubert, M., Thoma, M., and Cavallaro, A. (2010). CT Slice Localization via Instance-Based Regression. In *Proc. of SPIE Medical Imaging*, page 762320.

[Fisher, 1936] Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179–188.

[Fleming and Rishel, 1960] Fleming, W. and Rishel, R. (1960). An integral formula for total gradient variation. *Archiv der Mathematik*, pages 218–222.

[Franco et al., 2009] Franco, J., Bernabé, G., Fernández, J., and Acacio, M. E. (2009). A parallel implementation of the 2D wavelet transform using CUDA. In *Proc. of International Conference on Parallel, Distributed and Network-based Processing*, pages 111–118.

[Gimel'Farb, 1997] Gimel'Farb, G. (1997). Gibbs fields with multiple pairwise pixel interactions for texture simulation and segmentation. Technical report, INRIA.

[Giusti, 1984] Giusti, E. (1984). *Minimal Surfaces and Functions of Bounded Variation*. Springer.

[Grady, 2006] Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–83.

[Haralick, 1979] Haralick, R. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804.

[Harrison, 2005] Harrison, P. (2005). *Image Texture Tools*. PhD thesis.

[Ishikawa, 2003] Ishikawa, H. (2003). Exact optimization for Markov random fields with convex priors. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1333–1336.

[Jones et al., 1996] Jones, C., Marron, J. S., and Sheather, S. J. (1996). Progress in data-based bandwidth selection for kernel density estimation. *Computational Statistics*, 11:337–381.

[Jordan and Weiss, 2002] Jordan, M. I. and Weiss, Y. (2002). Probabilistic Inference in Graphical Models. Technical report, EECS Computer Science Division, University of California.

[Kawai et al., 2009] Kawai, N., Sato, T., and Yokoya, N. (2009). Image inpainting considering brightness change and spatial locality of textures. In *Proc. of PSIVT*, pages 271–282.

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

[Komodakis and Tziritas, 2007] Komodakis, N. and Tziritas, G. (2007). Approximate labeling via graph cuts based on linear programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1436–1453.

[Kschischang et al., 2001] Kschischang, F. R., Frey, B. J., and Loeliger, H. A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.

[Lellmann et al., 2008] Lellmann, J., Yuan, J., Becker, F., and Schnör, C. (2008). Convex Multi-Class Image Labeling by Simplex-Constrained Total Variation. Technical report, University of Heidelberg, Department of Mathematics and Computer Science.

[Luo et al., 2011] Luo, D., Ding, C., and Huang, H. (2011). Linear Discriminant Analysis : New Formulations and Overfit Analysis. In *Proc. of the AAAI Conference on Artificial Intelligence*, pages 417–422.

[Mallat, 2006] Mallat, S. (2006). *A wavelet tour of signal processing*. Elsevier, 2nd edition.

[Martin et al., 2001] Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *International Conference on Computer Vision*, pages 416–423.

[Michelot, 1986] Michelot, C. (1986). A Finite Algorithm for Finding the Projection of a Point onto the Canonical Simplex of R. *Journal of Optimization Theory and Applications*, 50(1):195–200.

[Mortensen, 1998] Mortensen, E. (1998). Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384.

[Mumford and Shah, 1989] Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685.

[Nieuwenhuis and Cremers, 2012] Nieuwenhuis, C. and Cremers, D. (2012). Spatially Varying Color Distributions for Interactive Multilabel Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14.

[Nieuwenhuis et al., 2011] Nieuwenhuis, C., Töppe, E., and Cremers, D. (2011). Space-varying color distributions for interactive multiregion segmentation: discrete versus continuous approaches. In *Energy Minimization Methods in Computer Vision and Pattern Recognition. LNCS in Computer Science*, pages 177–190.

[Ojala et al., 1996] Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A Comparative Study of Texture Measures with Classification Based on Feature Distributions. *Pattern Recognition*, 29(1):51–59.

[Parzen, 1962] Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33:1065–1076.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufman Publishing Inc.

[Ratliff et al., 2003] Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. (2003). Subgradient Methods for Maximum Margin Structured Learning. Technical report, Carnegie Mellon University, University of Alberta.

[Rosenblatt, 1956] Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a densitiy function. *The Annals of Mathematical Statistics*, 27:832–837.

[Rother and Kolmogorov, 2004] Rother, C. and Kolmogorov, V. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. In *Proc. of ACM SIGGRAPH*, pages 309–314.

[Rother and Kolmogorov, 2007] Rother, C. and Kolmogorov, V. (2007). Optimizing binary MRFs via extended roof duality. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7.

[Rudin et al., 1992] Rudin, L., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268.

[Santner et al., 2010] Santner, J., Pock, T., and Bischof, H. (2010). Interactive multi-label segmentation. In *Proc. of Asian Conference on Computer Vision*, pages 397–410.

[Santner et al., 2009] Santner, J., Unger, M., Pock, T., Leistner, C., Saffari, A., and Bischof, H. (2009). Interactive Texture Segmentation using Random Forests and Total Variation. In *Proc. of the British Machine Vision Conference*.

[Sebe and Lew, 2000] Sebe, N. and Lew, M. S. (2000). Wavelet based texture classification. In *Proc. of International Conference on Pattern Recognition*, volume 3, pages 947–950.

[Taskar et al., 2004] Taskar, B., Chatalbashev, V., and Koller, D. (2004). Learning Associative Markov Networks. In *Proc. of the International Conference on Machine Learning*, pages 102–110.

[Unger et al., 2008] Unger, M., Pock, T., and Trobin, W. (2008). TVSeg-Interactive total variation based image segmentation. In *British Machine Vision Conference*.

[Wu, 1982] Wu, F. Y. (1982). The Potts model. *Reviews of Modern Physics*, 54(1):235–268.

[Ye, 2005] Ye, J. (2005). Characterization of a Family of Algorithms for Generalized Discriminant Analysis on Undersampled Problems. *Journal of Machine Learning Research*, 6:483–502.

[Ye, 2006] Ye, J. (2006). Computational and Theoretical Analysis of Null Space and Orthogonal Linear Discriminant Analysis. *Journal of Machine Learning Research*, 7:1183–1204.

[Zach et al., 2008] Zach, C., Gallup, D., Frahm, J.-m., and Niethammer, M. (2008). Fast Global Labeling for Real-Time Stereo Using Multiple Plane Sweeps. In *Proc. of Vision, Modeling and Visualization Workshop*, pages 243–252.