TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Fraunhofer ITWM

Université franco-allemande Deutsch-Französische Hochschule

Multi-Class Image Segmentation via Convex and Biconvex Optimization

Vom Fachbereich Mathematik der Technischen Universität Kaiserslautern zur Verleihung des akademischen Grades Doktor der Naturwissenschaften (Doctor rerum naturalium, Dr. rer. nat.) genehmigte Dissertation von

Behrang Shafei

1. Gutachterin:	Prof. Dr. Gabriele Steidl (TU Kaiserslautern)
2. Gutachterin:	Prof. Dr. Sung Ha Kang (Georgia Institute of Technology, Atlanta, USA)
Disputation:	11. November 2013

D 386

Abstract

This thesis is divided into two parts. Both cope with multi-class image segmentation and utilize non-smooth optimization algorithms.

The topic of the first part, namely unsupervised segmentation, is the application of clustering to image pixels. Therefore, we start with an introduction of the biconvex center-based clustering algorithms c-means and fuzzy c-means, where c denotes the number of classes. We show that fuzzy c-means can be seen as an approximation of c-means in terms of power means.

Since noise is omnipresent in our image data, these simple clustering models are not suitable for its segmentation. To this end, we introduce a general and finite dimensional segmentation model that consists of a data term stemming from the aforementioned clustering models plus a continuous regularization term. We tackle this optimization model via an alternating minimization approach called *regularized c-centers* (RcC). Thereby, we fix the centers and optimize the segment membership of the pixels and vice versa. In this general setting, we prove convergence in the sense of set-valued algorithms using Zangwill's Theory [172].

Further, we present a segmentation model with a total variation regularizer. While updating the cluster centers is straightforward for fixed segment memberships of the pixels, updating the segment membership can be solved iteratively via non-smooth, convex optimization. Thereby, we do not iterate a convex optimization algorithm until convergence. Instead, we stop as soon as we have a certain amount of decrease in the objective functional to increase the efficiency. This algorithm is a particular implementation of RcC providing also the corresponding convergence theory. Moreover, we show the good performance of our method in various examples such as simulated 2d images of brain tissue and 3d volumes of two materials, namely a multi-filament composite superconductor and a carbon fiber reinforced silicon carbide ceramics. Thereby, we exploit the property of the latter material that two components have no common boundary in our adapted model.

The second part of the thesis is concerned with supervised segmentation. We leave the area of center based models and investigate convex approaches related to graph p-Laplacians and reproducing kernel Hilbert spaces (RKHSs). We study the effect of different weights used to construct the graph. In practical experiments we show on the one hand image types that are better segmented by the p-Laplacian model and on the other hand images that are better segmented by the RKHS-based approach. This is due to the fact that the p-Laplacian approach provides smoother results, while the RKHS approach provides often more accurate and detailed segmentations. Finally, we propose a novel combination of both approaches to benefit from the advantages of both models and study the performance on challenging medical image data.

Zusammenfassung

Diese Arbeit besteht aus zwei Teilen, welche beide die Multiklassen-Segmentierung von Bildern behandeln und Algorithmen der nichtglatten, konvexen Optimierung verwenden.

Der erste Teil dieser Arbeit befasst sich mit der unüberwachten Segmentierung. Da unüberwachte Segmentierung als Anwendung von Clusteringverfahren auf Bildpixel verstanden werden kann, beginnen wir mit der Einführung der zentrumsbasierten und bikonvexen Clusteringalgorithmen c-Means und Fuzzy c-Means, wobei c die Anzahl der Klassen bezeichnet. Wir zeigen, dass Fuzzy c-Means als Approximation von c-Means bezüglich Hölder-Mittel interpretiert werden kann.

Da Rauschen in unseren Bilddaten allgegenwärtig ist, sind diese simplen Clusteringverfahren nicht zur ihrer Segmentierung geeignet. Deshalb führen wir ein allgemeines, endlichdimensionales Modell ein, dass aus solch einem Clusteringmodell als Datenterm und einem Regularisierer besteht. Als Lösungsansatz für dieses Modell verwenden wir einen alternierenden Algorithmus namens *Regularized c-Centers* (RcC). Dabei halten wir die Zentren fest und optimieren die Klassenzugehörigkeit der Pixel und umgekehrt. In diesem allgemeinen Rahmen beweisen wir Konvergenz im Sinne mengenwertiger Algorithmen basierend auf Zangwills Theorie [172].

Des Weiteren präsentieren wir ein Segmentierungsmodell mit der totalen Variation als Regularisierer. Während das Berechnen der Zentren der Segmente für feste Segmentzugehörigkeit der Pixel unkompliziert ist, benutzen wir zum Aktualisieren der Segmentzugehörigkeit einen iterativen Algorithmus der konvexen Optimierung. Dieser Algorithmus, für den als konkrete Implementierung von RcC die oben erwähnte Konvergenztheorie gilt, wird nicht bis zur Konvergenz iteriert, sondern gestoppt sobald ein bestimmter Abstieg im Zielfunktional festgestellt werden kann. In vielen Beispielen wie 2d Bildern von Gehirngewebe und 3d Volumen von mit Kohlenstofffasern verstärkten SiC-Keramiken (C/SiC) und supraleitenden Verbunddrähten erhalten wir gute Ergebnisse mit diesem Algorithmus. Dabei nutzen wir in einer Modellanpassung die Eigenschaft von C/SiC aus, dass zwei Komponenten des Materials sich nicht berühren.

Im zweiten Teil dieser Arbeit behandeln wir überwachte Segmentierung. Wir untersuchen konvexe Modelle, die auf Graph-p-Laplace-Operatoren und Hilberträumen mit reproduzierendem Kern (RKHSs) basieren. Wir studieren die Auswirkung verschiedener Gewichte in der Konstruktion des Graphen und zeigen anhand praktischer Experimente, welches Verfahren für welchen Bildtypen besser geeignet ist. Der Hauptunterschied der Resultate ist, dass die p-Laplace-Modelle glattere Ergebnisse liefern, während Segmentierungen mit RKHS-Modellen oft genauer sind und mehr Details erkennen lassen. Schließlich präsentieren wir eine neues Modell das beide kombiniert, um von den jeweiligen Vorzügen zu profitieren. Experimente mit schwer segmentierbaren medizinischen Bilddaten werden zur Evaluierung der Verfahren herangezogen.

Acknowledgements

Many thanks ...

... to my supervisor Gabriele Steidl for her guidance and constant support throughout my time as her student.

... to Sung Ha Kang from the Georgia Institute of Technology, Atlanta, USA, for our fruitful collaboration on supervised segmentation.

... to my colleagues from the image processing department at the Fraunhofer ITWM, Kaiserslautern, Germany, and the mathematical image processing group of the University of Kaiserslautern for a pleasant and productive working atmosphere.

... to my colleague René Ciak for helpful discussions about mathematical details and the corresponding devils as well as for providing me with the definition of the function in Figure 2.2.

... to Jürgen Meinhardt from the Fraunhofer ISC, Würzburg, Germany, and Alexander Rack from the ESRF, Grenoble, France, for providing me with the C/SiC image data.

... to Alexander Rack, Christian Scheuerlein from CERN, Geneva, Switzerland, and the Applied Superconductivity Center of the National High Magnetic Field Laboratory, Florida State University, Tallahassee, USA, for providing me with the image data of the multifilament superconductor.

... to the Deutsch-Französische Hochschule for support.

... to Christian Schmidt, Henrike Stephani, and Thomas Weibel for proofreading.

... most of all to my parents and my Anna for their unshakable belief in me and for enduring all my moods.

Contents

Publications and Collaboration

Parts of this thesis have been published in the following 3 articles with authors in alphabetical order:

- [88] S. H. Kang, B. Shafei, and G. Steidl. Supervised and transductive multi-class segmentation using p-Laplacians and RKHS methods, 2012. Submitted,
- [79] Y. He, M. Y. Hussaini, J. Ma, B. Shafei, and G. Steidl. A fuzzy c-means method with total variation regularization for image segmentation. *Pattern Recognition*, 45(9): 3463–3471, 2012,
- [140] B. Shafei and G. Steidl. Segmentation of images with separating layers by fuzzy c-means and convex optimization. *Journal of Visual Communication and Image Representation*, 23(4):611–621, 2012.

We also worked on a 4^{th} publication that is not considered in this thesis, namely

- [45] R. Ciak, B. Shafei, and G. Steidl. Homogeneous penalizers and constraints in convex image restoration. *Journal of Mathematical Imaging and Vision*, 47(3):210–230, 2013.

More precisely, Section 2.4.1 about convex optimization with fixed codebook and a slightly different regularizer, the corresponding model and results in Section 2.6.1, and Section A.1 about d-dimensional computations have been published in [140]. Section 2.4.1 on the optimization with fixed codebook has been restated in [79]. The experimental Section 2.5.1 and the corresponding method STVcM as stated in Algorithm 7 have been published in [79] *without* results on convergence. Chapter 3 on supervised segmentation and the part of the concluding Chapter 4 corresponding to Chapter 3 have been published in [88]. The review of vector-valued reproducing kernel Hilbert spaces in Section 1.1.3 has been published in the corresponding preprint [89].

Regarding the part that has not yet been published, Section 2.2 and the corresponding part in the concluding Chapter 4 is joint work with Gabriele Steidl. The review of Zangwill's Theorem in Section 2.3.1 is joint work with René Ciak.

Notation and Preliminaries

Symbols

$\operatorname{dom} g$	$\{x \in \Omega : g(x) \neq \infty\}$ where $g : \Omega \to \mathbb{R} \cup \{\infty\}$ and $\Omega \subseteq \mathbb{R}^d$
∇	Gradient
∇_x	Vector of partial derivatives with respect to the components of $x \in \mathbb{R}^n$
$\frac{\partial}{\partial x_i}$	Partial derivative
$\partial g(x_0)$	Subdifferential of the function $g : \Omega \to \mathbb{R} \cup \{\infty\}$ at x_0, i.e., $\partial g(x_0) = \{y \in \mathbb{R}^d : f(x) - f(x_0) \geq \langle y, x - x_0 \rangle \text{ for all } x \in \mathbb{R}^d\}$
∂S	Lipschitz boundary of a set S
$\operatorname{int} S$	Interior of a set S
$\operatorname{ri} S$	Interior of a set S relative to its affine hull
2^S	Power set of S
$\lvert S \rvert$	Number of elements of a finite set S
S^c	c-fold Cartesian product of a set S where $c \in \mathbb{N}$
$\operatorname{conv} S$	Convex hull of a set S
ι_S	Indicator function of a set S, i.e., $\iota_S(x) = 0$ if $x \in S$ and $\iota_S(x) = \infty$ else
χ_S	Characteristic function of a set S, i.e., $\chi_S(x) = 1$ if $x \in S$ and $\chi_S(x) = 0$ else
$B_{\lVert\cdot\rVert}(y, \rho)$	$\{x \in \mathbb{R}^n : \lVert x - y \rVert \leq \rho\}$
$\mathcal{F}(A)$	If $\mathcal{F}$ is a set of functions, its subset of functions that map from A to $\mathbb{R}$
$\mathcal{F}(A, B)$	If $\mathcal{F}$ or $\mathcal{F}(A)$ is a set of functions, its subset of functions that map from A to B
$\mathcal{C}^n$	Set of n-times continuously differentiable functions where $\mathcal{C} := \mathcal{C}^1$
$\mathcal{C}_c^n$	Set of n-times continuously differentiable functions with compact support
$L^p(\Omega)$ for $p \in [1, \infty)$	Let $g : \Omega \to \mathbb{R}$ denote a measurable function and $\lVert g \rVert_{L^p} := (\int \lvert g(x) \rvert^p dx)^{1/p}$. Then, we define L^p by the quotient space $L^p(\Omega) := \{g : \lVert g \rVert_{L^p} < \infty\} / \{g : \lVert g \rVert_{L^p} = 0\}$, i.e., for two functions $g_1, g_2 \in L^p$ we have $g_1 = g_2$ if and only if $g_1(x) = g_2(x)$ almost everywhere.

$L^\infty(\Omega)$	Set of measurable functions $g : \Omega \to \mathbb{R}$ such that $g(x) < \kappa$ almost everywhere for a constant $\kappa \in \mathbb{R}$
$L^p_{\text{loc}}(\Omega)$	$\{g : g\|_K \in L^p(K)$ for all compact sets $K \subseteq \Omega\}$
$W^{n,p}(\Omega)$	$\{g \in L^p(\Omega)$: all weak derivatives of g up to order n are elements of $L^p(\Omega)\}$
$(x_j)_{j=1}^n$	$(x_1, \ldots, x_n)^\top \in \mathbb{R}^n$, analogously for $(x_j)_{j\in J}$ with some finite index set J
$(a_{i,j})_{i,j=1}^{m,n}$	Matrix $A \in \mathbb{R}^{m\times n}$ with elements $a_{i,j}$, analogously for $(a_{i,j})_{i\in K, j\in J} =: A_{KJ}$ with finite index sets K, J
$A \otimes B$	Kronecker product of matrices $A \in \mathbb{R}^{n_1\times n_2}$ and B, i.e., $A \otimes B = (a_{i,j}B)_{i,j=1}^{n_1,n_2}$
$\text{vec}(A)$	The matrix A column-wise reshaped into a vector
$I,\ I_n$	Identity matrix where $I_n \in \mathbb{R}^{n\times n}$;
$\mathcal{N}_i$	Neighborhood of pixel i
N	Number of image pixels
c	Number of segments (phases, classes)
f	Constant given image; usually $f : \{1, \ldots, N\} \to \mathbb{R}^m$
$u_k(j)$	$u_{(k-1)n+j}$ for $u \in \mathbb{R}^{cN}$, $k = 1, \ldots, c$, and $j = 1, \ldots, N$; denotes the degree of membership from pixel j to segment k
u_k	$(u_k(j))_{j=1}^N$
$u(j)$	$(u_k)_{k=1}^c$
$u(J)$	$(u_k(j))_{j\in J})_{k=1}^c$ for a finite index set J
$\|u\|$	modulus for $u \in \mathbb{R}$, $\sqrt{\sum_{k=1}^c ((u_k(j))_{j=1}^n)^2} \in \mathbb{R}^N$ for $u \in \mathbb{R}^{cN}$; not to be confused with the Euclidean norm $\|\cdot\|_2$
$r \in \mathbb{R}^{cm}$	Vector of segment centroids also called codebook where m denotes the feature dimension, i.e., $f(j) \in \mathbb{R}^m$

Properties of Functions, Sets, and Optimization Problems

Proper	A function $g : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is called proper if $g \not\equiv \infty$.
Lower semi-continuous (lsc)	A function $g : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is called lower semi-continuous at x_0 if $g(x_0) = \liminf_{x\to x_0} g(x)$. If such a function g is lower semi-continuous everywhere its epigraph $\{(x, y) \in \mathbb{R}^{d+1} : g(x) \le y\}$ is closed.
Biconvex	Let $X \subseteq \mathbb{R}^{d_1}$ and $Y \subseteq \mathbb{R}^{d_2}$ be convex and nonempty. A set $B \subseteq X \times Y$ is called biconvex if the set $\{y \in Y : (x, y) \in B\}$ is convex for all fixed x and the set $\{x \in X : (x, y) \in B\}$ is convex for all fixed y. An optimization problem of the form $$\operatorname*{argmin}_{x,y} f(x, y) \text{ subject to } (x, y) \in B$$ is called biconvex if B is biconvex and if the functions $f(\cdot, y)$ and $f(x, \cdot)$ are convex for every fixed y and x, respectively.

Calculation Rules

∞	$a\infty = a + \infty = a/0 = \infty$ and $0\infty = a/\infty = 0$ for $a \in \mathbb{R}$
Vectors	For $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ or $y \in \mathbb{R}$, the inequalities $x \bullet y$ and $y \bullet x$ are meant componentwise where $\bullet \in \{\leq, <, >, \geq, \}$. Similarly, $x^t = (x_j^t)_{j=1}^n$ for $t \in \mathbb{R}$ and $xy = (x_i y_i)_{j=1}^n$ for $x, y \in \mathbb{R}^n$.

Chapter 1

Introduction

As a fundamental step towards the analysis of an image, e.g., in terms of object recognition or characterization of materials, image segmentation has attracted a lot of attention. For instance, the simple representation of a segmented image supports the extraction of features describing objects. Moreover, volume fractions of materials' different components are easily computable from segmented 3d image data.

The images we cope with in this work consist of finitely many elements called pixels (picture elements) or voxels (3d volume picture elements). Gray-scale images assign to their elements a scalar value often in $\{0, 1, \ldots, 255\}$, $[0, 1]$, or $\mathbb{R}$. Color images assign to every element a vector whose dimension depends on the color model. For instance, the RGB color model consists of three channels representing the colors red, green, and blue. A straightforward representation of a 2d image with $n_1 \times n_2$ pixels and m feature (e.g., color) channels is a 3d matrix denoted by $F \in \mathbb{R}^{n_1 \times n_2 \times m}$ or equivalently by $F : \{1, \ldots, n_1\} \times \{1, \ldots, n_2\} \to \mathbb{R}^m$. Since our 3d volume images with $n_1 \times n_2 \times n_3$ voxels are always gray-valued, they can also be represented by a 3d matrix $F \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ or equivalently by $F : \{1, \ldots, n_1\} \times \{1, \ldots, n_2\} \times \{1, \ldots, n_3\} \to \mathbb{R}$. Further, we often use a representation in vector form simplifying notation, i.e., an image with $N = n_1 \ldots n_d$ pixels and m feature (e.g., color) channels is denoted by $f \in \mathbb{R}^{Nm}$ or equivalently by $f : \{1, \ldots, N\} \to \mathbb{R}^m$. The features of an image can be anything that is representable as a vector.

Segmentation is the task of partitioning an image into substantively related regions. In a segmented image every element is labeled with one of $c \in \mathbb{N}$ labels and all elements with the same label represent a segment also called phase or class. Figure 1.1 shows two different segmentations of a color image.

Segmentation can be categorized into unsupervised and supervised segmentation. In supervised segmentation we know for a subset of the image pixels beforehand the labels. It remains to determine the labels of the unlabeled pixels. On the contrary, in unsupervised segmentation the labels of all pixels have to be determined. In this work, we cope with unsupervised as well as supervised segmentation approaches.

The simplest technique for unsupervised segmentation of gray-scale images we can think of is thresholding. All pixels with values above a certain threshold are assigned to one segment and all others are assigned to another segment. Instead of setting the threshold manually, a way to find a threshold automatically is the application of the c-means clustering algorithm (see

(a) (b) (c)

Figure 1.1.: Segmentations of a traffic sign. Input image from [149].

Section 2.1) to the pixels of an image where c denotes the number of segments. In contrast to thresholding, c-means clustering is also applicable directly to color images. Figure 1.1b shows the c-means segmentation of Figure 1.1a. It is clearly visible that this simple approach can fail even if the segments are easily distinguishable for humans. Figure 1.1c depicts the result of a more sophisticated method, see Chapter 2 and Section 2.6.1 in particular. This method uses the fact that neighboring pixels often belong to the same segment.

Both c-means and the method of Section 2.6.1 have in common that they are center based. For instance, the prototype, centroid, or center of a segment can be the average of all gray values of the pixels assigned to that segment and all pixels most similar to one such center are assigned to its segment. We call the vector of all centers of the segments *codebook* denoted by $r \in \mathbb{R}^{cm}$ for images $f : \{1, \ldots, N\} \to \mathbb{R}^m$. Our approaches to supervised segmentation in Chapter 3 are not center based. Instead, we have the given labeled pixels whose similarity to unlabeled pixels are utilized to decide about their segment membership.

Depending on the task, not only labeled pixels but also different kinds of additional prior knowledge such as the shape [48, 134] can be incorporated into the segmentation process. Another type of prior knowledge is described in Section 2.6.1 where we exploit a material's property that two components have no common boundary. Indeed, this is also applied for the unsupervised segmentation of the traffic sign in Figure 1.1c enforcing a white separating segment between the red part of the sign and the background.

If we abstract from image segmentation and talk about partitioning of arbitrary data sets that do not have to be image pixels we enter the field of machine learning. In general, machine learning or clustering methods are applicable to image pixels for segmentation. For instance, the popular machine learning classifier random forests [26] has been applied to image pixels as implemented in the segmentation software Ilastik [146]. However, segmentation involves a lot of data points. For example, a rather small image of 256×256 pixels contains already 65536 elements. In practical applications one usually has to deal with significantly larger images. Some methods developed for machine learning might be applicable to tasks with fewer data points but reach their limits in terms of computational time and memory consumption when used for image segmentation.

1.1 Related Work

A vast number of methods to tackle image segmentation have been developed and are still subject of ongoing research. This section provides a review of segmentation approaches and concepts related to our work. All presented approaches consist of solving an optimization problem, since our work copes exclusively with energy minimization approaches.

1.1.1 Variational Segmentation Models

Many methods that tackle image segmentation as optimization problem are formulated in function spaces. This means images of dimension $d \in \mathbb{N}\setminus\{0\}$ are given as functions $f \in L^\infty(\Omega)$ where $\Omega \subset \mathbb{R}^d$ rather than as finite dimensional vectors or matrices. Thereby, Ω denotes a bounded, connected, and open set with Lipschitz boundary.

The Mumford-Shah Model and Related Concepts

In the first part of this section, we introduce the quite popular Mumford-Shah model for image segmentation and define related concepts such as the Hausdorff measure, the perimeter and the total variation.

The aim of the Mumford-Shah model is to obtain a segmentation of f into c classes where c is unknown in advance. To this end, Mumford and Shah [113] propose to solve

$$\text{(MS)} \qquad \operatorname*{argmin}_{v\in W^{1,2}(\Omega\setminus K),K\subseteq\Omega} E_{\mathrm{MS}}(v,K)$$
$$\text{where } E_{\mathrm{MS}}(v,K) = \int_\Omega (f(x)-v(x))^2 dx + \mu \int_{\Omega\setminus K} \|\nabla v(x)\|_2^2\, dx + \lambda \mathcal{H}^{d-1}(K),$$

where $\mathcal{H}^{d-1}(K)$ denotes the $d-1$-dimensional *Hausdorff measure* of the discontinuity set K, cf. [4].

Let $\hat{v}, \hat{K}$ denote a solution of (MS). The c areas of $\hat{v}$ separated by $\hat{K}$ are caused to be smooth by the second term of E_{MS}. In terms of image segmentation each of the smooth areas corresponds to a segment. Further, the segments are separated by the jump set $\hat{K}$. By minimizing the Hausdorff measure $\mathcal{H}^{d-1}(K)$, regularity on the borders of the smooth areas of $\hat{v}$ is imposed.

More precisely, the q-dimensional Hausdorff measure $\mathcal{H}^q$ determines the q-dimensional volume of subsets of $\mathbb{R}^d$ such as the length of a boundary curve in $\mathbb{R}^2$ or the area of a boundary surface in $\mathbb{R}^3$. To give an exact definition, we follow [4]. Let $\omega_q = \pi^{q/2}/\Gamma(1+q/2)$ where $\Gamma : t \mapsto \int_0^\infty s^{t-1}e^{-s}ds$ is the Euler Γ-function. Note that ω_k coincides with the volume of the unit ball in $\mathbb{R}^q$ if $q \geq 1$ is an integer. We define for every subset A of $\mathbb{R}^d$ the measure $\mathcal{H}^q$ by

$$\mathcal{H}^q(A) := \frac{\omega_q}{2^q} \lim_{\varepsilon\to 0} \left(\inf \left\{ \sum_{i\in\mathcal{I}} (\operatorname{diam} A_i)^q : \operatorname{diam} A_i < \varepsilon, A \subseteq \bigcup_{i\in\mathcal{I}} A_i \right\} \right) \tag{1.1}$$

where $\mathcal{I}$ is a countable index set and $\operatorname{diam} A := \sup\{\|a-b\| : a,b \in A\}$.

In literature the notion of the *perimeter* of a set often replaces the Hausdorff measure of the boundary. Descriptively, the perimeter of a set $A \in \mathbb{R}^d$ measures the length (surface, volume) of its boundary. We follow [4, Section 3.3] and define the perimeter of a set as total variation of its characteristic function.

The total variation semi-norm is defined for functions $g \in L^1(\Omega)$ by

$$\mathcal{TV}(g) := \sup \left\{ \int_\Omega g(x) \operatorname{div} \varphi(x) dx : \ \varphi \in \mathcal{C}_c^1(\Omega, \mathbb{R}^d), \ \|\varphi(x)\|_2 \leq 1 \text{ for all } x \in \Omega \right\}. \quad (1.2)$$

Note that since Ω is bounded we have $L^1(\Omega) = L^1_{\text{loc}}(\Omega)$. The space of functions of *bounded variation*

$$BV(\Omega) := \left\{ g \in L^1(\Omega) : \mathcal{TV}(g) < \infty \right\}$$

becomes with the norm $|g|_{BV} := \|g\|_{L^1} + \mathcal{TV}(g)$ a Banach space. For sufficiently smooth functions $g \in W^{1,1}(\Omega)$ the *total variation* can also be written as

$$\mathcal{TV}(g) = \int_\Omega \|\mathcal{D}g(x)\|_2 \ dx$$

where $\mathcal{D}g$ denotes the vector of weak partial derivatives of g. Descriptively, the total variation is the amount of change or variation measured by the modulus of the derivative summed over the whole domain. However, note that the space $W^{1,1}(\Omega)$ does not include piecewise smooth functions in general. For instance, the restriction of the Heaviside function

$$H : \mathbb{R} \to \mathbb{R}, \ H(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise} \end{cases} \quad (1.3)$$

to the bounded set $(-1, 1)$ is not in $W^{1,1}((-1, 1))$ but in $BV((-1, 1))$. For more details on functions of bounded variation and the total variation see, e.g., [4, 65, 132].

After introduction of the total variation, we define the perimeter in the following. Let

$$\chi_A : \Omega \to \mathbb{R}, \ \chi_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise} \end{cases}$$

denote the characteristic function of A in Ω. For an arbitrary Lebesgue measurable set $A \subseteq \Omega$ the perimeter denoted by $\operatorname{Per}(A, \Omega)$ is defined as the variation of the characteristic function of A in Ω given by

$$\operatorname{Per}(A, \Omega) := \mathcal{TV}(\chi_A).$$

In case $\mathcal{H}^{d-1}(\Omega \cap \partial A) < \infty$ we obtain $\mathcal{H}^{d-1}(\Omega \cap \partial A) = \operatorname{Per}(A, \Omega)$, see [4, Section 3.3].

Eventually, we note that minimizing (MS) is not trivial. This optimization problem is neither convex nor smooth. There exists a lot of research concerning the properties of E_{MS} and its minimization, see, e.g., [3, 37, 36, 157, 1, 32]. Very recently efficient algorithms for different convex approximations of E_{MS} have been proposed, e.g., in [121, 34]. See also [152] for a convexified version suitable for color images. In [131] an adaption of the data term according to the type of noise given in the image is proposed. They tackle the resulting non-convex problem via alternating minimization over v for fixed K and vice versa.

The Piecewise Constant Mumford-Shah Model

Closer related to our work is the restriction of the functions v in (MS) to be piecewise constant with number of segments c fixed beforehand. The piecewise constant version of (MS) can be written as

$$\text{(PMS)} \qquad \begin{aligned} &\operatorname*{argmin}_{r\in\mathbb{R}^c,\Omega_1,\dots,\Omega_c\subseteq\Omega} E_{\mathrm{PMS}}(r,\Omega_1,\dots,\Omega_c) \\ &\text{subject to} \quad \bigcup_{k=1}^{c}\Omega_k=\Omega,\ \Omega_{k_1}\cap\Omega_{k_2}=\emptyset \text{ for all } k_1\neq k_2 \\ &\text{where } E_{\mathrm{PMS}}(r,\Omega_1,\dots,\Omega_c)=\underbrace{\sum_{k=1}^{c}\int_{\Omega_k}(r_k-f(x))^2dx}_{\text{data term}}+\nu\underbrace{\sum_{k=1}^{c}\mathrm{Per}(\Omega_k,\Omega)}_{\text{smoothing term}} \end{aligned}$$

where $\nu\in\mathbb{R}$ is a fixed regularization parameter. We call the vector $r\in\mathbb{R}^c$ the codebook containing gray-value prototypes or centers for each segment.

The Active Contour Model of Chan and Vese In case $c=2$ where $\Omega_1=\Sigma$ is and $\Omega_2=\Omega\setminus\bar{\Sigma}$ are open sets, this problem has been tackled in [42] (see [158] for a generalization of this approach to $c\geq 2$). Based on [90, 116, 173], Chan and Vese [42] formulate the problem using level sets of a Lipschitz continuous function ϕ with

$$\begin{aligned} \partial\Sigma &= \{x\in\Omega:\phi(x)=0\}, \\ \Sigma &= \{x\in\Omega:\phi(x)>0\}, \\ \Omega\setminus\bar{\Sigma} &= \{x\in\Omega:\phi(x)<0\} \end{aligned}$$

where the contour separating the segments is given by $\partial\Sigma$. Utilizing the Heaviside function H defined in (1.3) and its distributional derivative $\delta_0(\phi)=\phi(0)$, the functional E_{PMS} is reformulated in [42] to

$$\begin{aligned} &\int_\Omega (r_k-f(x))^2H(\phi(x))dx+\int_\Omega(r_k-f(x))^2(1-H(\phi(x)))dx \\ &\qquad +\nu\int_\Omega \overbrace{\delta_0(\phi(x))\,\|\nabla\phi(x)\|_2}^{\|\nabla H(\phi(x))\|_2=}dx. \end{aligned} \tag{1.4}$$

In [42], the minimization of (1.4) is done alternatingly for fixed ϕ over r and vice versa. Observe that one cannot expect to find globally optimal solutions due to non-convexity. For fixed ϕ the minimizer $(\hat{r}_1,\hat{r}_2)^\top$ of (1.4) is simply the mean over the segments $\{x\in\Omega:\phi(x)\geq 0\}$ and $\{x\in\Omega:\phi(x)<0\}$, respectively, given by

$$\hat{r}_1=\frac{\int_\Omega f(x)H(\phi(x))dx}{\int_\Omega H(\phi(x))dx} \quad\text{and}\quad \hat{r}_2=\frac{\int_\Omega f(x)(1-H(\phi(x)))dx}{\int_\Omega(1-H(\phi(x)))dx}.$$

For the minimization with fixed r, a function ϕ is computed that fulfills necessary optimality conditions, namely the Euler-Lagrange equation of (1.4), after replacing H and δ_0 by twice continuously differentiable approximations.

The question whether the sequence of iterates $(r^{(n)}, \phi^{(n)})_{n\in\mathbb{N}}$ converges was not handled in [42]. After discretization, however, the solution of (PMS) via alternating minimization is a special case of our general segmentation method introduced in Section 2.3 with the according convergence theory. Therefore, one has to compute a global minimizer of (1.4) for fixed r which is possible for $c = 2$ due to [115, 41].

Global Parametric Optimization for Two Classes The work of Strandmark et al. [151] copes with the global optimization of E_{PMS} in case $c = 2$ with respect to both Σ and r. The central theorem in [151] states that a function $\hat{w}$ solves

$$\operatorname*{argmin}_{w\in BV(\Omega)} \int_\Omega G(x, w(x))dx + \lambda\, \mathcal{TV}(w) \tag{1.5}$$

if and only if $\chi_{\{x:\hat{w}(x)>s\}}$ solves

$$\operatorname*{argmin}_{u\in BV(\Omega,\{0,1\})} \int_\Omega u(x)g(x, s)dx + \lambda\, \mathcal{TV}(u)$$

where $\frac{\partial}{\partial s}G(x, s) = g(x, s)$ besides other rather technical assumptions. This is a generalization of a result from [38].

To apply this theorem to (PMS), Strandmark et al. rewrite (PMS) in terms of characteristic functions as

$$\operatorname*{argmin}_{u\in BV(\Omega),r_1,r_2\in\mathbb{R}} \int_\Omega u(x)(f(x) - r_1)^2 + (1 - u(x))(f(x) - r_2)^2 dx + \nu\, \mathcal{TV}(u) \tag{1.6}$$
$$\text{subject to} \quad u(x) \in \{0, 1\} \text{ for all } x \in \Omega.$$

Substituting $\delta = r_1 - r_2$ and $\rho = r_1^2 - r_2^2$ in (1.6) yields

$$E(u, \rho, \delta) = \int_\Omega u(x)(\rho - 2\delta f(x)) + \left(\frac{\rho - \delta^2}{2\delta} - f(x)\right)^2 dx + \lambda\, \mathcal{TV}(u).$$

To evaluate $m(\delta) = \min_{u,\rho} E(u, \rho, \delta)$ Strandmark et al. propose to proceed as follows:

1. Compute a solution $\hat{w}$ of (1.5) where $G(x, \rho) = \frac{1}{2}(\rho - 2\delta f(x))^2$ via convex optimization, see, e.g., Section 2.4.1 for efficient algorithms.

2. For all possible values ρ, evaluate $E(\chi_{\{x:\hat{w}(x)>\rho\}}, \rho, \delta)$.

3. The pair $(\chi_{\{x:\hat{w}(x)>\hat{\rho}\}}, \hat{\rho})$ found in step 2 with smallest value $E(\chi_{\{x:\hat{w}(x)>\rho\}}, \rho, \delta)$ is the global solution for fixed δ denoted by $m(\delta)$.

In the next step Strandmark et al. derive a lower bound of $m(\delta)$ in a closed interval $[\delta_1, \delta_2]$ with an approach similar to the previous steps. These bounds are utilized in [151] for the application of a branch-and-bound optimization over the real parameter δ to obtain the final solution.

Phase Balancing and c as Optimization Variable The model of Sandberg, Kang, and Chan [130] uses a different regularizer than (PMS). The goal is to segment the image into reasonably sized and balanced parts or phases. More precisely, the inverse scale term $s_k := \mathcal{TV}(u_k)/\int_\Omega u_k(x)dx$ with characteristic function u_k of Ω_k is introduced. Smaller s_k enforces a larger size of the segment Ω_k. To obtain a partitioning of the image with size-balanced segments the unweighted sum $\sum_{k=1}^{c} s_k$ is considered. Then, the proposed model reads

$$\underset{\substack{u\in BV(\Omega,\{0,1\}),\\ r\in\mathbb{R}^c,\ c\in\mathbb{N}\setminus\{0\}}}{\operatorname{argmin}} \sum_{k=1}^{c}\int_\Omega u_k(x)(f(x)-r_k)^2dx + \lambda\sum_{k=1}^{c}s_k\sum_{k=1}^{c}\mathcal{TV}(u_k).$$

In contrast to (PMS), c is an optimization variable in this model. To solve this involved non-convex optimization problem, Sandberg et al. present a fast heuristic algorithm that observes the change of the energy functional when the segment assignment of a pixel is changed. This approach is based on a similar algorithm for the two class model (1.4) by Song and Chan [147].

The scale term is used again by Kang et al. [87] to segment objects in an image with respect to their size. In this paper, a two step approach is proposed. In a pre-processing step an intensity based segmentation has to be computed. To this end, Kang et al. use the method from [130]. With this segmentation at hand, the scale values s_k for all $k \in \{1\dots,c\}$ of all connected components of this segmentation are clustered with a regularized c-means approach that aims to solve

$$\underset{c,r,\mathbf{C}_k}{\operatorname{argmin}}\,\lambda\left(\sum_{k=1}^{c}\frac{1}{n_k}\right)+\sum_{k=1}^{c}\sum_{s_k\in\mathbf{C}_k}|s_k-r_k|^2$$

where n_k denotes the number of elements in cluster $\mathbf{C}_k$, r_k is the average value of all elements in $\mathbf{C}_k$, and c is the number of clusters. As above, a heuristic algorithm based on the changes in the energy functional is proposed to tackle this problem.

A Convex Relaxation of (PMS) utilizing Box Functions A convex relaxation of (PMS) is given by Brown et al. [31] for $c = 2$ and $c = 3$. Their work is based on ideas of [85, 119, 68, 30].

In the first step, Brown et al. review how to obtain a convex relaxation for optimization problems of the form

$$\underset{v\in BV(\Omega,\Gamma)}{\operatorname{argmin}}\int_\Omega \rho(x,v(x))dx+\sum_{k=1}^{n}\mathcal{TV}(v_k) \tag{1.7}$$

where $\Gamma := \{0,1,\dots,N_1\}\times\cdots\times\{0,1,\dots,N_n\}$ and $\rho:\Omega\times\Gamma\to\mathbb{R}$ is a continuous, non-negative, and possibly non-convex function. For the embedding of (1.7) into a higher dimensional space, box functions

$$b_v:\underbrace{\Omega\times\tilde{\Gamma}}_{\subset\mathbb{R}^d\times\mathbb{R}^n}\to\mathbb{R},\ b_v(x,\gamma):=\chi_{\{\tilde{x}:v(\tilde{x})\geq\gamma\}}(x)=\begin{cases}1 & \text{if } v(x)\geq\gamma,\\ 0 & \text{otherwise}\end{cases}$$

where $\tilde{\Gamma} := \{0,\dots,N_1+1\}\times\cdots\times\{0,\dots,N_n+1\}$ are utilized. For every $x\in\Omega$, we can recover $v(x)$ from $b_v(x,\cdot)$ via $v_k(x)=\sum_{l=1}^{N_k}b_v(x,le_k)$ where e_k is the k^{th} standard basis of $\mathbb{R}^n$.

To rewrite the regularizing term by means of box functions the co-area formula [59] for functions $g \in BV(\Omega)$ given by

$$\mathcal{TV}(g) = \int_{-\infty}^{\infty} \mathrm{Per}(\{x : g(x) > t\})dt \tag{1.8}$$

is used. Applying (1.8) yields

$$\sum_{l=1}^{N_k} \mathcal{TV}(b_v(\cdot, le_k)) = \sum_{l=1}^{N_k} \mathcal{TV}(\chi_{\{x:v(x)\geq le_k\}}) = \sum_{l=0}^{N_k-1} \mathrm{Per}(\{x : v_k(x) > l\}) \underset{(1.8)}{=} \mathcal{TV}(v_k).$$

Let us now turn to rewrite the data term by means of box functions. To this end, the forward difference operator D is defined by

$$(\mathrm{D}_k\phi)(x,\gamma) := \begin{cases} 0 & \text{if } \gamma_k = N_k + 1, \\ \phi(x, \gamma + e_k) - \phi(x,\gamma) & \text{otherwise} \end{cases}$$

where $\phi : \Omega \times \tilde{\Gamma} \to \mathbb{R}$. The composition is denoted by $\mathrm{D}_{1,\dots,n} := \mathrm{D}_n \circ \cdots \circ \mathrm{D}_1$ and can be computed for box functions to

$$\mathrm{D}_{1,\dots,n} b_v(x,\gamma) = \begin{cases} (-1)^n & \text{if } \gamma = u(x), \\ 0 & \text{otherwise.} \end{cases}$$

Thus, (1.7) can be written as

$$\underset{b_v \in \mathcal{B}}{\operatorname{argmin}} \; E(b_v) \text{ where} \tag{1.9}$$

$$E(b_v) = (-1)^n \sum_{\gamma \in \Gamma} \int_\Omega \rho(x,\gamma) \mathrm{D}_{1,\dots,n} b_v dx + \sum_{k=1}^{n} \sum_{l=1}^{N_k} \mathcal{TV}(b_v(\cdot, le_k))$$

and $\mathcal{B}$ is the set of all box functions.

Problem (1.9) is convex in b_v but the set $\mathcal{B}$ is not. Therefore, the admissible set $\mathcal{B}$ is relaxed to the convex set

$$\begin{aligned} \mathcal{Y} := \{\phi : \Omega \times \tilde{\Gamma} \to [0,1] : \\ \phi(x,0) = 1, \\ \phi(x,\gamma) = 0 \text{ if there exists a } k \text{ with } \gamma_k = N_k, \\ (-1)^n \mathrm{D}_{1,\dots,n}\phi \geq 0\} \supset \mathcal{B}. \end{aligned} \tag{1.10}$$

Additionally, let $\hat{\phi} \in \operatorname{argmin}_{\phi \in \mathcal{Y}} E$. Then, Brown et al. show that v with

$$v_k(x) = \sum_{l=1}^{N_k} \chi_{\{\tilde{x}:\hat{\phi}(\tilde{x}, le_k)\geq 1\}}(x)$$

is a solution of (1.7) if

$$E(\hat{\phi}) = E(\chi_{\{x:\hat{\phi}(x)\geq 1\}}). \tag{1.11}$$

With criterion (1.11) it is easy to evaluate whether a relaxed solution is also a solution of the original problem.

To optimize (PMS) for $c = 2$, Brown et al. consider

$$\begin{aligned} &\operatorname*{argmin}_{u,r_1,r_2} \int_\Omega u(x)(hr_1(x) - f(x))^2 + (1-u(x))(hr_2(x) - f(x))^2 dx + \nu\, \mathcal{TV}(u) \\ &\text{subject to} \quad \mathcal{D}r_1 \equiv \mathcal{D}r_2 \equiv 0,\ u \in BV(\Omega, \{0,1\}),\ r_1, r_2 \in BV(\Omega, \{0,\dots,N_r\}) \end{aligned} \tag{1.12}$$

where $h = 1/N_r$ is the step size of the discretization and $\mathcal{D}$ denotes the vector of distributional partial derivatives. Contrary to (PMS), the center or codebook vector r is allowed to attain just finitely many values. Similarly to [29], Brown et al. solve the sequence of problems

$$\begin{aligned} \operatorname*{argmin}_{\substack{u\in BV(\Omega,\{0,1\}),\\ r_1,r_2\in BV(\Omega,\{0,\dots,N_r\})}} &\int_\Omega u(x)(hr_1(x) - f(x))^2 + (1-u(x))(hr_2(x) - f(x))^2 dx \\ &+\nu\, \mathcal{TV}(u) + a^{(i)}\, \mathcal{TV}(r_1) + a^{(i)}\, \mathcal{TV}(r_2) \end{aligned} \tag{1.13}$$

where $a^{(i)} \to \infty$ is strictly increasing. Let $(\hat{u}^{(i)}, \hat{r}^{(i)})$ denote a solution of (1.13) for fixed i. Then, Brown et al. state that $(\hat{u}, \hat{r}) = \lim_{i\to\infty}(\hat{u}^{(i)}, \hat{r}^{(i)})$ is a global solution of (1.12).

For fixed i they apply their convex relaxation approach to (1.13) where $v = (u, r_1, r_2) : \Omega \to \Gamma = \{0,1\} \times \{0,\dots,N_r\}^2$ and $\rho : \Omega \times \Gamma$, $\rho(x, u(x), r_1(x), r_2(x)) = u(x)(hr_1(x) - f(x))^2 + (1 - u(x))(r_2(x) - f(x))^2$. The resulting convex functional is optimized over $\mathcal{Y}$ defined in (1.10). Finally, let us mention that the case $c = 3$ is handled analogously in [31].

A Convex Relaxation of (PMS) also Suitable for c Unknown in Advance Similarly to [31], Bae et al. [10] consider an approach where the (PMS) is embedded into a higher dimensional space to obtain a convex relaxation of (PMS). In the variant of [10] the number of segments c can also be handled as an optimization variable. This yields in terms of characteristic functions

$$\begin{aligned} &\operatorname*{argmin}_{\substack{u\in BV(\Omega)^c,\\ r\in\{l_1,\dots,l_n\}^c,\\ c\in\mathbb{N}\setminus\{0\}}} \sum_{k=1}^{c} \int_\Omega u_k(x)(f(x) - r_k)^2 dx + \lambda \sum_{k=1}^{c} \mathcal{TV}(u_k) \\ &\text{subject to} \quad \sum_{k=1}^{c} u_k(x) = 1,\ \text{for all } x \in \Omega, \\ &\qquad\qquad u_k(x) \in \{0,1\} \text{ for all } x \in \Omega, k = 1,\dots,c. \end{aligned} \tag{1.14}$$

Observe that each element of the codebook r_k for $k = 1,\dots,c$ is optimized over a finite set of n elements $\{l_1,\dots,l_n\} \subseteq \mathbb{R}$ and not over $\mathbb{R}$ where normally $c \ll n$. As shown in [10], problem

(1.14) is closely related to

$$\begin{aligned} &\operatorname*{argmin}_{u\in BV(\Omega)^n} \sum_{k=1}^{n} \int_\Omega u_k(x)(f(x)-l_k)^2 dx + \lambda \sum_{k=1}^{n} \mathcal{TV}(u_k) \\ &\text{subject to} \quad \sum_{k=1}^{n} u_k(x) = 1, \text{ for all } x \in \Omega, \\ &\qquad\qquad u_k(x) \in \{0,1\} \text{ for all } x \in \Omega, k = 1,\dots,n. \end{aligned} \tag{1.15}$$

More precisely, let $\hat{u}$ denote a solution of (1.15) and let $\mathcal{I} \subseteq \{1,\dots,n\}$ be the set of indices with $\hat{u}_k \not\equiv 0$ for all $k \in \mathcal{I}$. Then, Bae et al. state that $((\hat{u}_k)_{k\in\mathcal{I}}, (l_k)_{k\in\mathcal{I}}, |\mathcal{I}|)$ is a global solution of (1.14). Usually, $|\mathcal{I}| \ll n$. For a fixed number of classes c in advance, Bae et al. [10] provide a similar model.

As for the work of Brown et al. [31], a drawback of this approach is a finite set $\{l_1,\dots,l_n\}$ of admissible codebook vectors r. The more elements are admissible for r, the slower the according optimization algorithms become.

Problem (1.15) is a particular case of a spatially continuous Potts model [123]. This can be convexified as discussed in the next part of this section.

A Spatially Continuous Potts Model

For fixed r, (PMS) is also referred to as a special case of a spatially continuous version of the Potts model [123]. In terms of characteristic functions this can be written as

$$\text{(PO)} \qquad \operatorname*{argmin}_{u\in BV(\Omega)^c} E_{\mathrm{PO}}(u) \quad \text{subject to} \quad u(x) \in \{0,1\}^c, \quad \sum_{k=1}^{c} u_k(x) = 1 \quad \text{for all } x \in \Omega$$

$$\text{where } E_{\mathrm{PO}}(u) = \sum_{k=1}^{c} \int_\Omega u_k(x) g_k(x) dx + \lambda \sum_{k=1}^{c} \mathcal{TV}(u_k)$$

with $g_k \in L^\infty(\Omega)$. Since problem (PO) is not convex, several convex relaxations of (PO) and minimization algorithms have been proposed to be able to compute at least approximate solutions efficiently such as in [171, 11, 98, 120, 99, 170].

To obtain a convex relaxation Zach et al. [171] relax the functions u_k to attain values in the whole interval $[0,1]$ yielding

$$\begin{aligned} &\operatorname*{argmin}_{u\in BV(\Omega)^c} \sum_{k=1}^{c} \int_\Omega u_k(x) g_k(x) dx + \lambda \sum_{k=1}^{c} \mathcal{TV}(u_k) \\ &\text{subject to} \quad u(x) \in [0,1]^c, \quad \sum_{k=1}^{c} u_k(x) = 1 \quad \text{for all } x \in \Omega. \end{aligned} \tag{1.16}$$

In the two-class case there exists the following result [115, 41]. If $c = 2$ the solution $\hat{u}^{\mathrm{bin}}$ of (PO) can be obtained by thresholding $\hat{u}_1^{\mathrm{rel}}$ at $t \in (0,1)$ where $\hat{u}^{\mathrm{rel}}$ denotes the solution of the relaxed problem (1.16), see also Section 2.4.3. To our knowledge, a similar result for the multi-class case $c > 2$ is still open.

The model of Lellmann et al. [98] is very similar to (1.16). The only difference is the coupled regularizer. More precisely, for $u_k \in W^{1,1}$ the regularizer can be written as

$$\int_\Omega \sqrt{\|\mathcal{D}u_1(x)\|_2^2 + \cdots + \|\mathcal{D}u_c(x)\|_2^2}\, dx \tag{1.17}$$

instead of $\sum_{k=1}^c \int_\Omega \|\mathcal{D}u_k(x)\|_2\, dx$.

The work of Bae et al. [11] is build on [171, 98]. In [11] the starting point is

$$\begin{aligned}
&\min_{u\in BV(\Omega)^c} \sum_{k=1}^c \int_\Omega u_k(x) g_k(x) dx + \underbrace{\max_{\varphi\in\mathcal{V}_\lambda} \int_\Omega u_k(x) \operatorname{div} \varphi_k(x) dx}_{=\lambda\, \mathcal{TV}(u_k)} \\
&= \min_{u\in BV(\Omega)} \max_{\varphi\in\mathcal{V}_\lambda} \sum_{k=1}^c \int_\Omega u_k(x)(g_k(x) + \operatorname{div}\varphi_k(x)) dx \\
&\text{subject to} \quad u(x) \in S_c \quad \text{for all } x \in \Omega
\end{aligned}$$

where $\mathcal{V}_\lambda := \{\varphi \in \mathcal{C}_c^1(\Omega, \mathbb{R}^d) : \|\varphi(x)\|_2 \le \lambda \text{ for all } x \in \Omega\}$ and $S_c := \operatorname{conv}\{e_1, \ldots, e_c\}$ with k^{th} unit basis vector $e_k \in \mathbb{R}^c$. Further, Bae et al. employ that for a vector $y \in \mathbb{R}^n$ the relation

$$\min_{x\in S_n} \sum_{k=1}^n x_i y_i = \min\{y_1, \ldots, y_n\} \tag{1.18}$$

holds true. Exchanging min and max and using (1.18) yields

$$\max_{\varphi\in\mathcal{V}_\lambda} \sum_{k=1}^c \int_\Omega \min\{g_k(x) + \operatorname{div}\varphi_k(x)\} dx.$$

From this representation Bae et al. derive the result that under very specific conditions a solution of the relaxed problem (1.16) is also a solution of the Potts model (PO).

A different approach to obtain a convex representation of (PO) is presented by Pock et al. [120]. As equivalent formulation for the labeling function $l : \Omega \to \{0, \ldots, c-1\}$, i.e., $l(x) = k$ if and only if $x \in \Omega_{k+1}$, upper level sets are utilized. The upper level sets $\theta = (\theta_1, \ldots, \theta_{c-1})^\top$ are defined by

$$\theta_k(x) = \begin{cases} 1 & \text{if } l(x) \ge k, \\ 0 & \text{otherwise.} \end{cases}$$

Vice versa, the labeling function can be recovered via $l(x) = \sum_{k=1}^{c-1} \theta_k(x)$. An appropriate definition of the set of upper level sets is given by

$$\mathcal{U} = \{\theta : \Omega \to \{0,1\}^{c-1} : 0 \le \theta_{c-1} \le \cdots \le \theta_1 \le 1 \text{ for all } x \in \Omega\}.$$

With $\theta_0 \equiv 0$ and $\theta_c \equiv 1$ the characteristic functions u_k of Ω_k can be written in terms of upper level sets as $u_k(x) = \theta_k(x) - \theta_{k+1}(x)$. Similarly to [171, 98, 11], the data term of (PO) is rewritten in terms of the characteristic functions as

$$\sum_{k=0}^{c-1} \int_{\Omega_k} g_k(x) dx = \sum_{k=0}^{c-1} \int_\Omega (\theta_k(x) - \theta_{k+1}(x)) g_k(x) dx.$$

Although it does not resemble the smoothing term of the Potts model (PO), Pock et al. discuss the usage of $\sum_{k=0}^{c-1} \mathcal{TV}(\theta_k)$ as regularizer. However, they detect that interfaces where m functions θ_k jump from 0 to 1 are counted m times. To overcome this drawback, Pock et al. propose to use a constrained version of $\mathcal{TV}$ given by

$$\sup \left\{ \int_\Omega g(x) \operatorname{div} \varphi(x) dx : \ \varphi \in \mathcal{C}_c^1(\Omega, \mathbb{R}^d), \ \|\varphi(x)\|_2 \leq 1, \ \left| \sum_{k_1 \leq k \leq k_2} \varphi_k(x) \right| \leq 1 \right.$$
$$\left. \text{for all } x \in \Omega, 1 \leq k_1 \leq k \leq k_2 \leq c-1 \right\}$$

as smoothing term and state that this resembles (PO). See [40] for a proof. Of course, the optimization of the resulting problem over $\mathcal{U}$ is non-convex. Hence, the set $\mathcal{U}$ is relaxed to the convex admissible set

$$\{\theta : \Omega \to [0,1]^{c-1} : 0 \leq \theta_{c-1} \leq \cdots \leq \theta_1 \leq 1 \text{ for all } x \in \Omega\}$$

yielding a convex approximation of (PO). Further, Pock et al. give error bounds on the objective of their approximation and state that their approximation provides solutions closer to binary than the one of Zach et al. [171].

In case the number of segments c is very large and not prime, Goldlücke and Cremers [66] propose a method to obtain a significant gain in performance regarding computational time and memory consumption. As mentioned in [31], this or similar techniques could also be used for convex relaxations of (PMS) discussed in the previous part of this section [10, 31]. The idea in [66] is to reformulate the vector of characteristic functions

$$u \in U := \{u \in BV(\Omega, \{0,1\}^c) : \textstyle\sum_{k=1}^c u_k(x) = 1 \text{ for all } x \in \Omega\}$$

as follows. In problem (PO) the vector $u(x)$ consists of zeros everywhere except for the component of the segment (label) of pixel x. Let $\Lambda = \{1, 2, \ldots, c\}$ denote the set of labels. Further, let $\Lambda = \Lambda_1 \cup \cdots \cup \Lambda_n$ with $c_i = |\Lambda_i|$ and $c = \prod_{i=1}^n c_i$. Then, Goldlücke and Cremers [66] propose to replace the admissible set in (PO) by

$$\Upsilon := \left\{ (u_k^i)_{1 \leq i \leq n, k \in \Lambda_i} : u_k^i \in BV(\Omega, \{0,1\}) \text{ and } \textstyle\sum_{k \in \Lambda_i} u_k^i(x) = 1 \text{ for all } x \in \Omega, 1 \leq i \leq n \right\}.$$

The dimension of $u(x)$ for $u \in U$ is $c = \prod_{i=1}^n c_i$ while the dimension of $\tilde{u}(x)$ for $\tilde{u} \in \Upsilon$ is only $\sum_{i=1}^n c_i$. Figure 1.2 shows an example. To obtain $u(x) \in \{e_1, \ldots, e_c\}$ one can apply $u_k(x) = \prod_{i=1}^n u_k^i(x)$. Further, Goldlücke and Cremers [66] propose an according convex relaxation to obtain an approximate solution.

Finally, let us mention that there exists work on variants of the Potts model with regularizers different from the total variation. Just to mention a few examples, see [167] for a non-local version or [78] for shearlet regularizers. Lellmann et al. [97] use a more general, weighted version of (2.51) that can be written for sufficiently smooth functions $u \in W^{1,1}$ as

$$\int_\Omega \sqrt{\|\mathcal{D}u_1(x)\|_A^2 + \cdots + \|\mathcal{D}u_c(x)\|_A^2} \, dx$$

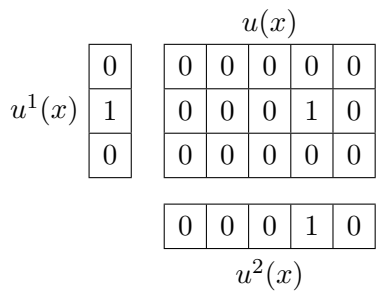


Figure 1.2.: Example for the reduction of dimension proposed in [66]. The vector of characteristic functions $u(x)$ at some image pixel x is shown in matrix shape for $c = 15$. It can be reformulated in two vectors $u^1(x)$ and $u^2(x)$ containing together only 8 elements instead of 15. Analogously, e.g., for $c = 64$ a reduction of dimension to 16 is possible.

where A is a weight matrix and $\|v\|_A = \sqrt{v^\top A^\top A v}$. The matrix A can be chosen in such a way that the jump from segment i to j is penalized differently than the jump from j to k where $1 \leq i, j, k \leq c$ and $i \neq j \neq k$.

Although the cited papers about the Potts model contain minimization algorithms, we just mention that all convex relaxations of the Potts model can be optimized efficiently with algorithms reviewed in Section 2.4.1. These algorithms are the *alternating direction method of multipliers* [62, 54, 67, 56, 138, 24] and the *primal dual hybrid gradient algorithm* [39, 56, 176]. For further speed up of the latter algorithm one can consider the preconditioned variant [118]. Note that there also exist other suitable methods such as the algorithm of Nesterov [114] and the fast iterative shrinkage-thresholding algorithm (FISTA) of Beck and Teboulle [12]. A comparison of all suitable algorithms is far beyond the scope of this work.

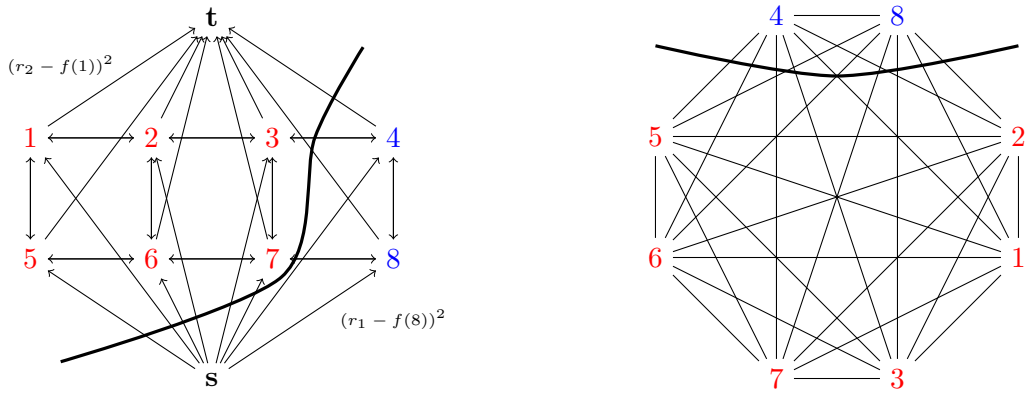
1.1.2 Graph Cuts

A *weighted (directed) graph* consists of a finite set of vertices $\boldsymbol{V}$, a set of edges $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$, and a symmetric weight matrix $W = (w_{i,j})_{i \in V, j \in V}$ with $w_{i,j} \geq 0$ and $w_{i,j} = 0$ if and only if $(i, j) \notin \boldsymbol{E}$. A graph that fulfills $(i, j) \in \boldsymbol{E}$ if $(j, i) \in \boldsymbol{E}$ for all $i, j \in \boldsymbol{V}$ is called *undirected*. Therefore, we can denote the elements of $\boldsymbol{E}$ equivalently by $\{i, j\}$ in the undirected case. We assume that $\boldsymbol{E}$ does not have any loops, i.e., $(j, j) \notin \boldsymbol{E}$. To represent an image as graph, every pixel j corresponds to vertex $j \in \boldsymbol{V}$ and the edges of the vertex j define its neighborhood $\mathcal{N}_j = \{i : (j, i), (i, j) \in \boldsymbol{E}\}$ of pixels. Usually, similar (e.g., in terms of position or feature value) pixels are grouped in neighborhoods.

We distinguish between two different approaches. One is related to the Minimal Cut Theorem for directed graphs due to Elias et al. [55] and Ford and Fulkerson [60]. Closer related to our work is another approach based on undirected graphs also called spectral clustering [105, 51, 58] in the unsupervised case.

Directed Graph Cuts

For this quite popular approach used for various tasks in image processing and computer vision, we just give a brief illustration for a simple two class segmentation model with fixed codebook



(a) Directed graph. The weights are given by $w_{\mathbf{s}j} = (r_1 - f(j))^2$, $w_{j\mathbf{t}} = (r_2 - f(j))^2$, and $w_{ij} = \lambda$ for all pixels $i, j \in \{1, \dots, 8\}$.

(b) Undirected graph. The weights between the pixels i and j are chosen according to their similarity, e.g., $w_{ij} = e^{-\|f(i)-f(j)\|_2^2}$.

Figure 1.3.: The vertices $\boldsymbol{V}$ correspond to image pixels numbered from 1 to 8. The thick black curves sketch the cut and indicate the two-class segmentation of a 2×4 pixel image.

r. For a given image $f \in \mathbb{R}^N$ we consider

$$\operatorname*{argmin}_{l \in \{r_1, r_2\}^N} \underbrace{\sum_{j=1}^{N} (l(j) - f(j))^2}_{\text{data term}} + \lambda \underbrace{\sum_{j=1}^{N} \sum_{i \in \mathcal{N}_j} \psi(l(j), l(i))}_{\text{regularizer}} \tag{1.19}$$

where $\psi(l(j), l(i)) = 0$ if $l(i) = l(j)$ and $\psi(l(j), l(i)) = 1$ otherwise. Model (1.19) with a general data term $g(l)$ is due to Ising [86] and generalized by Potts [123] to $c > 2$. Note that spatially continuous versions of this model are discussed in Section 1.1.1. As shown in Figure 1.3a, the set of vertices of the graph $\mathcal{V}$ is obtained by adding a source $\mathbf{s}$ and a sink $\mathbf{t}$ to the image vertices $\boldsymbol{V}$ and by adding edges from $\mathbf{s}$ to every image pixel and from every image pixel to $\mathbf{t}$ to obtain the extended edge set $\mathcal{E} := \boldsymbol{E} \cup \{(\mathbf{s}, j), (j, \mathbf{t}),\ j = 1, \dots, N\}$. The weights of the edges between $\mathbf{s}$ as well as $\mathbf{t}$ and image pixel j are given by $(r_1 - f(j))^2$ as well as $(r_2 - f(j))^2$, respectively, according to the data term. The edge weight between image pixels i and j is simply λ according to the regularization term. For some $\mathcal{S} \subset \mathcal{V}$ with $\mathbf{s} \in \mathcal{S}$ and $\mathbf{t} \notin \mathcal{S}$ let $C \subset \mathcal{E}$ be defined as $C = \{(i, j) \in \mathcal{E} : i \in \mathcal{S}, j \in \mathcal{V} \setminus \mathcal{S}\}$. Now finding the *minimal cut* $\operatorname{argmin}_C \sum_{(i,j) \in C} w_{i,j}$ is precisely our optimization problem (1.19). This is a standard problem in combinatorial optimization. Based on the Minimal Cut Theorem [55, 60] there exist efficient algorithms providing globally optimal solutions.

Problem (1.19) can be rewritten in terms of characteristic vectors $u \in \mathbb{R}^{2N}$ with $u(j) = (u_1(j), u_2(j))^\top \in \mathbb{R}^2$ as

$$\sum_{j=1}^{N} u_1(j)(l(j) - f(j))^2 + u_2(j)(l(j) - f(j))^2 + \frac{\lambda}{2} \sum_{j=1}^{N} \sum_{i \in \mathcal{N}_j} \|u(j) - u(i)\|_1$$
$$\text{subject to} \quad u_k(j) \in \{0, 1\}, \quad u_1(j) + u_2(j) = 1 \text{ for all } j \in \{1, \dots, N\}$$

where $u_k(j) = 1$ means pixel j is in segment k. The regularizer in this form is a discrete and anisotropic measure for the length of the boundary, i.e., a variant of the discretized total variation. We use for our models in the following chapters isotropic variants of the discrete total variation such as defined later in (2.29) and (2.46). The anisotropic regularizer introduces stronger discretization errors. See [92] and the references therein for an exhaustive comparison, more details, and possible workarounds.

Boykov et al. [25] cope with multi-label approaches and show that the Potts model is NP-hard. In the work of Chaux et al. [44] multi-class segmentation with directed graph cuts and optimization of the codebook is done alternatingly. These graphs can also be motivated stochastically in terms of Markov random fields. For details, see, e.g., [22].

Undirected Graph Cuts and Graph p-Laplacians

Contrary to the method with directed graphs described Section 1.1.2, the ansatz with undirected graphs is not motivated by the solution of some given optimization problem which does not have to be a segmentation model. The motivation of this approach stems from the structure of the graph itself and is suitable for clustering of data points, e.g., segmentation of image pixels. As sketched in Figure 1.3b, every vertex represents a pixel and the edge weight $w_{i,j}$ is the value of a similarity measure between pixels i and j. Usually, in applications with reasonably sized images a lot of the weights have to be zero due to tractability issues. In the following, we give a brief introduction to spectral clustering, i.e., the unsupervised case, and refer to [105] for more details. Supervised segmentation using undirected graphs is addressed in Section 3.1.

Spectral Clustering For some nonempty sets $A_1, \dots, A_c$ with $A_1 \cup \dots \cup A_c = \boldsymbol{V}$ and $\bigcap_{k=1}^{c} A_k = \emptyset$ the *minimal cut*

$$\operatorname*{argmin}_{A_1,\dots,A_c} \sum_{k=1}^{c} \sum_{\substack{i \in A_k, \\ j \in \boldsymbol{V} \setminus A_k}} w_{i,j} \tag{1.20}$$

can be interpreted as the minimal degree of similarity between the minimizing sets $\hat{A}_k$, $k = 1, \dots, c$.

One problem of the unsupervised model (1.20) is that clusters with isolated vertices are favored, since the objective value is increasing in the number of cut edges. To overcome this drawback, two very common approaches are the minimal *ratio cut*

$$\operatorname*{argmin}_{A_1,\dots,A_c} \sum_{k=1}^{c} \sum_{\substack{i \in A_k, \\ j \in \boldsymbol{V} \setminus A_k}} \frac{w_{i,j}}{|A_k|} \tag{1.21}$$

of Hagen and Kahng [72] and the minimal *normalized cut*

$$\operatorname*{argmin}_{A_1,\dots,A_c} \sum_{k=1}^{c} \sum_{\substack{i \in A_k, \\ j \in \boldsymbol{V} \setminus A_k}} \frac{w_{i,j}}{\sum_{i \in A_k} \sum_{j=1}^{N} w_{i,j}}$$

of Shi and Malik [141].

While (1.20) can be solved efficiently for $c = 2$ [168], computing a ratio or normalized cut increases the difficulty of the optimization problems. More precisely, finding the minimal ratio and normalized cut are both NP-hard problems, see, e.g., [161] for a discussion.

To this end, one can relax the problems as we will see in the following for the ratio cut (the approach for the normalized cut is similar). We start with a reformulation of the ratio cut as stated in [105]. Let $U = (u_1, \ldots, u_c)$ where $u_k = (u_k(j))_{j=1}^N \in \mathbb{R}^N$ denotes a normalized vector of characteristic functions, i.e., $u_k(j) = 1/\left\|u_k\right\|_2$ if $j \in A_k$ and $u_k(j) = 0$ otherwise. Note that this implies $U^\top U = I$. Further, let

$$\Delta_2 := D^{\mathcal{G}} - W \tag{1.22}$$

define the *graph 2-Laplacian* where $D^{\mathcal{G}} = \operatorname{diag}\left(\sum_{j=1}^N w_{1,j}, \ldots, \sum_{j=1}^N w_{N,j}\right)$ denotes the diagonal matrix of row sums of W. Then, problem (1.21) can be written in terms of characteristic functions as

$$\hat{u} := \operatorname*{argmin}_{u \in \mathbb{R}^{cN}} E_{\Delta_2}(u) \tag{1.23}$$

$$\text{subject to} \quad U^\top U = I,\ u_k(j) \in \{0, \frac{1}{\left\|u_k\right\|_2}\},\ j = 1, \ldots, N,\ k = 1, \ldots, c \tag{1.24}$$

where

$$E_{\Delta_2}(u) := \sum_{k=1}^c \langle u_k, \Delta_2 u_k \rangle = \sum_{k=1}^c (U^\top \Delta_2 U)_{k,k} = \operatorname{Tr}(U^\top \Delta_2 U)$$

and $\operatorname{Tr}(M) = \sum_{i=1}^n M_{i,i}$ denotes the trace of a diagonal matrix $M \in \mathbb{R}^{n \times n}$. The relaxation

$$\begin{aligned} \hat{V} := \operatorname*{argmin}_{U \in \mathbb{R}^{N \times c}} & \operatorname{Tr}(U^\top \Delta_2 U) \\ \text{subject to} \quad & U^\top U = I \end{aligned} \tag{1.25}$$

of (1.23) can be solved with a variant of the Rayleigh-Ritz Theorem, see, e.g., [104, Section 5.2.2 (6)]). This variant states that the matrix $\hat{V}$ consisting of the first c eigenvectors of Δ_2 as column vectors, i.e., the c eigenvectors corresponding to the c smallest eigenvalues, minimizes (1.25). In other words, instead of solving an NP-hard problem one computes the eigenvectors of the graph 2-Laplacian.

To assign the pixels to segments one can apply a simple clustering method such as c-means [76, 106, 102] to the rows of $\hat{V}$ where row j corresponds to the j^{th} pixel. The preceding approach is also called *spectral clustering*.

Remark 1.1. *The first eigenvector of Δ_2 is the vector $\mathbf{1} := (1)_{i=1}^N$, i.e., $\Delta_2 \mathbf{1} = 0$. Since Δ_2 is symmetric, eigenvectors of different eigenvalues are orthogonal. Hence, a column of the relaxed solution $\hat{V}$ cannot fulfill the constraints* (1.24)*. Moreover, let $\hat{A}_1, \ldots, \hat{A}_c$ denote the solution corresponding to $\hat{u}$ and $\hat{B}_1, \ldots, \hat{B}_c$ the solution corresponding to $\hat{V}$ after assignment of pixels to segments. Then, the difference of the resulting ratio cuts between the solutions of the unrelaxed and relaxed problem given by*

$$\sum_{k=1}^c \sum_{\substack{i \in A_k, \\ j \in \boldsymbol{V} \setminus A_k}} \frac{w_{i,j}}{|A_k|} - \sum_{k=1}^c \sum_{\substack{i \in B_k, \\ j \in \boldsymbol{V} \setminus B_k}} \frac{w_{i,j}}{|B_k|}$$

can be arbitrarily large, see [105, Section 5.4] and the references therein.

Very recently, approaches to clustering including the *graph p-Laplacian* were investigated [33, 155, 80, 81, 125]. For $p \geq 1$ the non-linear graph p-Laplacian $\Delta_p : \mathbb{R}^N \to \mathbb{R}^N$ is defined by

$$(\Delta_p x)_i := \sum_{j=1}^{N} w_{i,j} \phi(x(i) - x(j)) \text{ where } \phi(t) := |t|^{p-1} \operatorname{sign}(t), \tag{1.26}$$

see [5, 33]. For instance, Hein and Bühler [80] prove that the second smallest eigenvalue of Δ_1 equals the minimal *ratio Cheeger cut*

$$\begin{aligned} &\min_{A,B} \frac{\sum_{i \in A, j \in B} w_{i,j}}{\min\{|A|, |B|\}} \\ &\text{subject to} \quad A \cup B = V,\ A \cap B = \emptyset \end{aligned} \tag{1.27}$$

and present an algorithm for the computation of eigenvectors of Δ_1.

1.1.3 Reproducing Kernel Hilbert Spaces

As in our preprint [89], we briefly review vector-valued reproducing kernel Hilbert spaces (RKHSs). For the theory of vector-valued RKHSs we refer to [35, 109, 117, 136].

A vector-valued RKHS is a Hilbert space $\mathcal{H}$ of vector-valued functions $g : \mathbb{R}^2 \to \mathbb{R}^c$ where the point evaluation operator $\delta_x : \mathcal{H} \to \mathbb{R}^c$ with $\delta_x g := g(x)$ is linear and bounded for all $x \in \mathbb{R}^2$, that is, for all $x \in \mathbb{R}^2$ there exists a constant C_x such that

$$\|g(x)\|_2 \leq C_x \|g\|_{\mathcal{H}} \qquad \text{for all } g \in \mathcal{H}.$$

A function K from $\mathbb{R}^2 \times \mathbb{R}^2$ to the set $\operatorname{Sym}_c(\mathbb{R})$ of real-valued, symmetric $c \times c$-matrices is called a *kernel of positive type* or *operator-valued, positive definite kernel* if for all $n \in \mathbb{N}$, $x_1, \ldots, x_n \in \mathbb{R}^2$ and $a_1, \ldots, a_n \in \mathbb{R}$ the inequality

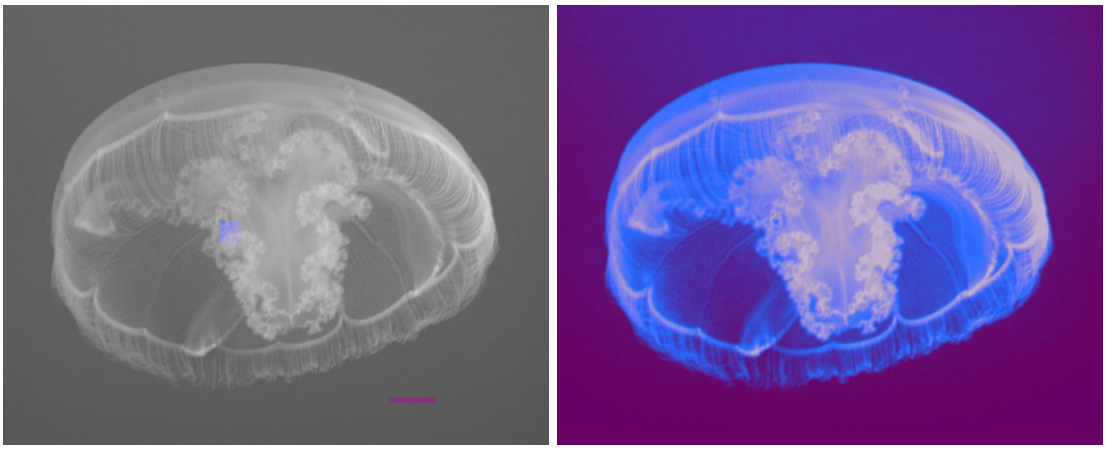
$$\sum_{i,j=1}^{n} a_i a_j \langle v, K(x_i, x_j) v \rangle \geq 0 \quad \text{for all } v \in \mathbb{R}^c$$

holds. As in the scalar case, any RKHS canonically defines a kernel K of positive type by $K(x, y) = \delta_x \delta_y^*$. For all $v \in \mathbb{R}^c$ this kernel fulfills

- the *reproducing property* $\langle g(x), v \rangle = \langle g, K_x v \rangle_{\mathcal{H}}$ for all $x \in \mathbb{R}^2$ and
- $K_x v := K(\cdot, x) v \in \mathcal{H}$.

On the other hand, every kernel of positive type defines a unique vector-valued RKHS. Another approach to vector-valued RKHSs considers the space

$$\mathcal{H}^{\text{pre}} := \operatorname{span}\{K_x v : x \in \mathbb{R}^2, v \in \mathbb{R}^c\}$$



(a) (b)

Figure 1.4.: Image colorization due to [71]: the small given dabs of color in (a) are used to obtain the color information for the whole image in (b).

of all finite linear combinations of $K_x v$ for a kernel K of positive type. Let for $g := \sum_{i=1}^n a_i K_{x_i} v_i$ and $\tilde{g} := \sum_{i=1}^n b_i K_{y_i} w_i$

$$\langle g, \tilde{g} \rangle_{\mathcal{H}} := \sum_{i,j=1}^n a_i b_j \langle v_i, K(x_i, y_j) w_j \rangle, \quad \|g\|_{\mathcal{H}}^2 = \sum_{i,j=1}^n a_i a_j \langle v_i, K(x_i, x_j) v_j \rangle. \tag{1.28}$$

Then the closure $\mathcal{H} = \overline{\mathcal{H}^{\text{pre}}}$ with respect to $\|\cdot\|_{\mathcal{H}}$ is the RKHS associated with K.

In the following, we consider a regularized least squares problem in a RKHS. Let $f : \Omega \to \mathbb{R}$ with $\Omega \in \mathbb{R}^d$ denote a given gray scale image. With $c = 3$ according to three color channels, Ha Quang et al. [71] propose to use

$$\hat{g} := \operatorname*{argmin}_{g \in \mathcal{H}} \sum_{j \in L} \|y(j) - g(x_j)\|_2^2 + \lambda \|g\|_{\mathcal{H}}^2, \quad \lambda > 0 \tag{1.29}$$

for image colorization where $j \in L \subset \Omega$ are finitely many pixels of given color and $y(j) \in \mathbb{R}^3$ the corresponding color values, see Figure 1.4. The minimizer $\hat{g} : \Omega \to \mathbb{R}^3$ is then a colored version of the gray scale image f. By the Representer Theorem [109, 71] we obtain

$$\hat{g} = \sum_{j \in L} K(\cdot, x_j) \alpha(j), \quad \alpha(j) \in \mathbb{R}^c. \tag{1.30}$$

This means, we need only to determine $\alpha(j)$ for $j \in L$ to get the whole function $\hat{g} \in \mathcal{H}$.

In their experiments, Ha Quang et al. use a diagonal kernel of the form $K(x_i, x_j) = w_{x_i, x_j} I$ where w_{x_i, x_j} represents the similarity between pixels x_i and x_j, e.g., $w_{x_i, x_j} = \exp(-|f(x_i) - f(x_j)|^2 / \sigma^2)$. With this simplification, problem (3.12) can be equivalently expressed in terms of scalar-valued reproducing kernel Hilbert spaces independently for each color channel. The theory for the scalar-valued case goes back to the work of Aronszajn [7] in 1950 and the scalar-valued version of the Representer Theorem has been established by Kimeldorf and Wahba [91] in 1971.

In machine learning, there exist various methods using RKHSs such as classification with *Support Vector Machines*. More details can be found, e.g., in the textbooks [135, 76] and the references therein. There also exist combinations of scalar-valued as well as vector-valued RKHSs with graph p-Laplacians, see, e.g., [142, 13, 111, 103], see also Section 3.3. For supervised image segmentation, methods related to scalar-valued RKHSs are widely used and investigated such as in [15, 93, 127, 162, 169] just to mention a few.

1.2 Contributions

The Fuzzy c-Means Model as an Approximation of Hard Clustering in Terms of p-Means Motivated by the work of Teboulle [156], we derive the optimization problem tackled by the fuzzy c-means algorithm (FCM) [20, 77] as an approximation of the c-means model [106, 102, 137] in terms of p-means (power means, Hölder means). To this end, we pursue a different direction than Teboulle and state a novel theorem on the representation of p-means as maxima of linear functions over specific sets. In the limiting case $p = -\infty$ corresponding to hard clustering, these sets coincide with the unit simplex.

Convergence of a General Biconvex Segmentation Algorithm We state the general biconvex multi-class segmentation model in finite dimensional spaces. Similarly to many other segmentation models, our model consists of a data term incorporating segment centers and a regularizer. We pose some general requirements to the dissimilarity function in the data term while the regularizer only has to fulfill continuity. Further, we propose the general algorithm *regularized c-centers* (RcC) that alternatingly optimizes the segment membership of pixels and the codebook of segment centroids. This algorithm contains many known algorithms as particular implementations such as c-means, fuzzy c-means, moving-average FCM [84], and the algorithm of Chan and Vese [42] with global optimization for fixed cluster centers [115]. We prove convergence of RcC in the sense of Zangwill [172]. More precisely, a sequence generated by RcC either converges to a partial minimizer (see Section 2.1) or – in the worst case – contains a subsequence that does. Moreover, every convergent subsequence has a partial minimizer as limit point. Our motivation is the convergence analysis for the simple fuzzy c-means algorithm stated by Hathaway et al. [77] that is also based on the work of Zangwill.

An Efficient Implementation and a Model Adaption for Real-Life Applications Based on the work of [171, 98, 11, 120], we present a total variation regularized multi-class segmentation model. Additionally to the previous work, we optimize also over the segment centers. To minimize this model, we contribute the alternating algorithm *total variation c-means* (TVcM) that is a particular implementation of RcC with the related convergence theory. While the optimization of the centers can be done in a single step, the segment membership of each pixel has to be optimized iteratively. Our algorithm iterates merely until a certain amount of decrease of the objective has been achieved before updating the cluster centers. This results in faster convergence in our experiments compared to a full optimization of the segment membership in every iteration.

We show promising results for the segmentation of simulated brain tissue images. Moreover, we apply our methods successfully to real-life problems, namely the segmentation of volumetric data of materials. We provide an efficient 3d implementation incorporating the discrete cosine transform for the fast solution of a linear system occurring in every iteration. In addition, we adjust the optimization problem to make the minimizer fulfill a separation property of a particular material called carbon fiber reinforced silicon carbide (C/SiC). More precisely, one of the components of C/SiC always separates the other two. This is exploited in a two-step segmentation approach where we penalize neighboring pixels in an initial segmentation obtained in a first step. We obtain favorably results compared to the method of [97] that incorporates a distance between labels into the regularizer for a unified one-step segmentation.

Investigation of Supervised Learning Methods for Segmentation Related to p-Laplacians and RKHSs For supervised image segmentation, we investigate a family of models related to graph p-Laplacians that have been used for $p = 2$ in segmentation [96] and also recently for $p = 1$ in learning [28]. For $p = 2$ we state that the simplex constraints are fulfilled automatically with a novel proof. This has been shown in [96] in 2012 for an equivalent model without mentioning any relations to 2-Laplacian and with a completely different proof. We study the effect of different parameters $p \geq 1$ and we discuss the effect of the choice of the weights used to construct the graph.

Motivated by the work in image colorization [71], we apply a well known regularized least squares model class-wise in reproducing kernel Hilbert spaces to multi-class image segmentation. For the p-Laplacian and the RKHS model we obtain promising results for appropriate images. The performance of the different methods depending on the image type and on the expected segmentation is investigated. More precisely, the RKHS model tends to provide more detailed and accurate solutions while the p-Laplacian model is suitable if one is interested in smooth segmentations.

A Novel Model Combining p-Laplacian and RKHSs Methods To benefit from the advantages of both approaches, we mention a model where the RKHS part acts as data term and the p-Laplacian part operates as regularizer. For $c = p = 2$ this model has been used in machine learning before [142, 13] and this model has already attracted attention for $c \geq 2$ and $p = 1$ very recently in [27]. However, since it is not suitable for image segmentation due to tractability issues, we propose a different model incorporating a projection. We obtain smaller linear systems in our algorithms that can be solved on usual desktop computers for reasonably sized images.

1.3 Outline

Chapter 2 This chapter copes with unsupervised segmentation. Unsupervised segmentation is the application of clustering to image pixels. We review center-based partitioning clustering and the c-means algorithm in **Section 2.1**.

Based on [156], we demonstrate how to derive the fuzzy c-means model as an approximation of the hard c-means model in terms of power means.

Section 2.3 introduces a biconvex regularized multi-class segmentation model and the corresponding alternating algorithm called regularized c-centers (RcC) that contains some well known methods such as the model of Chan and Vese [42] with global optimization for fixed cluster centers [115] as special cases. Motivated by the convergence theory of fuzzy c-means [77], we state a convergence theorem in the sense of Zangwill [172].

In **Section 2.4** we present two closely related algorithms, namely total variation c-means (TVcM) and single step total variation c-means (STVcM). Based on the work of [171, 98, 11, 120] we use the total variation as a regularizer and optimize alternatingly with respect to the segment membership of the pixels and the codebook of segment centroids. The main difference between the two methods is that TVcM is a particular implementation of RcC which is not true for STVcM that we have published first in [79].

The good performance of our algorithms is shown in the numerical experiments of **Section 2.5**. First, we provide promising results for STVcM in the segmentation of simulated brain tissue data. Then, we show that TVcM compares similarly with a suitable parameter choice and we present examples where STVcM fails to converge while TVcM still does.

Section 2.6 is concerned with the segmentation of real-life data of materials. We see that our algorithms are well suited for these tasks. Further, we present a model adaption to one of the material's properties, namely the separation of two segments by a third one. This has been published by us in [140].

Chapter 3 The topic of this chapter is supervised segmentation, i.e., some image pixels are labeled according to their segment beforehand in contrast to Chapter 2. Another difference to Chapter 2 is that all models are convex rather than biconvex. Further, we leave the center based approach behind us and represent our images as undirected graphs. We have published this chapter in [88].

Section 3.1 copes with a model related to graph p-Laplacians. We show that simplex constraints stemming from the relaxation of the minimal graph cut are fulfilled automatically for a solution of the 2-Laplacian functional. Further, we derive in the non-trivial case $p \neq 2$ optimization algorithms in terms of a first order primal dual approach due to Chambolle and Pock [39]. We present numerical examples indicating the influence of p and the chosen weights.

In **Section 3.2** we state a well known least squares problem in a reproducing kernel Hilbert space that has been used for image colorization before by Ha Quang et al. [71]. Fortunately, this variational problem reduces to a finite dimensional optimization problem due to the Representer Theorem of Kimeldorf and Wahba [91] from 1971 in the scalar-valued case and [109, 71] in the vector-valued case. In numerical experiments we show promising results for this simple and efficient method and compare to the p-Laplacian model. While the p-Laplacian approach provides smooth solutions, the RKHS approach reveals more details.

To benefit from both smoothness and high accuracy we propose a combined model in **Section 3.3**. Thereby, the p-Laplacian term is used as regularizer while the RKHS term can be understood as data term. To make this model feasible for segmentation tasks that naturally involve a large number of data points, we present a projection model.

Section 3.4 provides segmentation results for medical image data. We see that the different models perform favorably on different types of images and expected segmentations.

Chapter 4 In the final chapter we draw conclusions for the two previous chapters and point out directions of possible future research.

Chapter 2

Unsupervised Segmentation

Image segmentation is the task of partitioning an image into meaningful regions (segments, phases). Figure 2.1 shows an example where a slice of carbon fiber reinforced silicon carbide (C/SiC) is segmented according to its components. For more details about this application see Section 2.6.1 and [140].

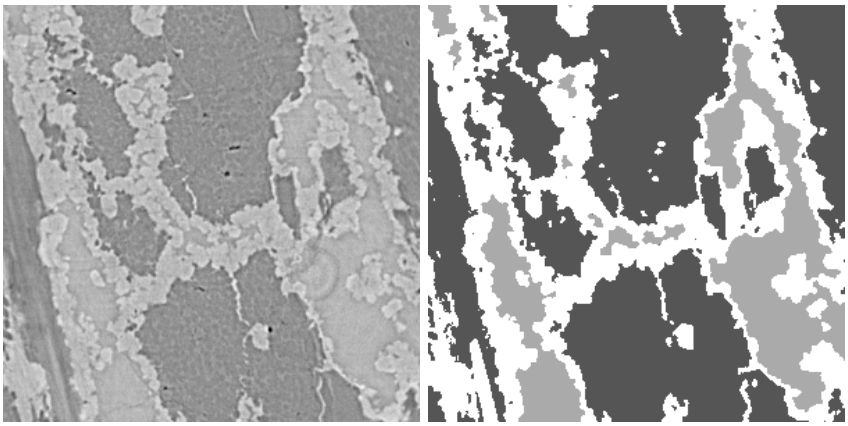
Contrary to supervised segmentation tackled in Chapter 3, in unsupervised segmentation no labeled pixels (also called markers) are given before the segmentation process starts. In other words, we do not have any information about the segment membership of a subset of the pixels.

All the models considered in this work consist of the minimization of some functional. In case images are represented by functions the resulting problems belong to the field of calculus of variations. Models of this kind are often referred to as variational segmentation models. However, at some point these models have to be discretized for numerical computations. Further, for our models the advantages of a variational representation such as circumventing discretization artifacts or technical details of a particular discretization are negligible. Therefore, we prefer to represent images as discrete matrices often reshaped to vectors leading to finite dimensional optimization problems allowing a simpler algorithmic treatment.

Contributions and Outline Since segmentation is clustering applied to image pixels, Section 2.1 introduces center-based clustering with a focus on *hard* clustering basically following the lines of Teboulle [156]. This includes a discussion of the widespread c-means algorithm [106, 102] and its convergence due to Selim and Ismail [137].

Section 2.2 presents an example of *soft* clustering, namely the fuzzy c-means method (FCM) [20, 18]. Thereby, we derive the algorithm in terms of p-means similar to [156]. To this end, we give an introduction of means, describe some of their basic properties, and present novel results on the representation of power means in terms of maxima of linear functions over specific sets. This novel theorem is used in the derivation of the fuzzy c-means model as an approximation of the hard clustering model in terms of power means.

In Section 2.3 we review the convergence theory for monotonic algorithms of Zangwill [172]. Our main theoretical contribution of this chapter is the application of Zangwill's Theorem to our general segmentation algorithm *regularized c-centers* (RcC) to develop its convergence theory. RcC consists of the alternating minimization over the cluster membership for fixed centers and vice versa. Further, RcC contains many methods as particular implementations. Examples are



(a) Input (b) Result

Figure 2.1.: Unsupervised segmentation of C/SiC ceramics [124, 140].

c-means [106, 102], fuzzy c-means [20, 18], moving-average FCM [84], the method of Chan and Vese [42] with global optimization [115, 41], and total variation c-means proposed in Section 2.4 of our work.

Section 2.4 proposes the segmentation method *single step total variation c-means* (STVcM) published in [79] by us without any proof of convergence. Further, we introduce its novel variant *total variation c-means* (TVcM). In contrast to STVcM, Zangwill's convergence theory is applicable to TVcM. While the update of the segment centers is straightforward, we apply convex optimization in terms of the alternating direction method of multipliers (ADMM, see [62]) to update the segment membership. STVcM performs exactly one step of ADMM. In contrast, TVcM waits until a certain amount of decrease of the objective has been achieved before updating the cluster centers.

In Section 2.5 we show the good performance of STVcM. We compare it to moving-average FCM [84] designed for brain tissue segmentation and obtain significantly better results especially for higher noise levels in segmentation of phantom brain tissue data. Additionally, we show that the performance of TVcM and STVcM is similar in numerical experiments. We show that both approaches outperform an alternate convex search algorithm that waits until ADMM converges before updating the cluster centers and is again a particular implementation of RcC.

Section 2.6 shows experiments with real world image data from materials science. For a specific application, namely the 3d segmentation of carbon fiber reinforced silicon carbide ceramics, we present a way of adjusting the model to separate the layers. This separating property is due to the material's structure but not always visible in the image data.

2.1 Center-Based Partitioning Clustering by the c-Means Algorithm

As unsupervised segmentation is the application of clustering to image pixels, we introduce the general concept of clustering following the lines of Teboulle [156]. Thereby, we focus on partitioning clustering were the N given data points are partitioned into c pairwise disjoint

clusters. Of course, the interesting situation in clustering is to have more than c distinct data points and $1 < c < N$. We restrict ourselves to this case throughout the whole work. In our applications the number of clusters c is known beforehand. To determine a suitable c is a topic of ongoing research in itself, see, e.g., [74, 153, 159]. Additionally, there exist clustering algorithms that determine the number of clusters during the clustering process such as proposed in [57, 6, 87].

Given the number of clusters c, many partitioning clustering methods update iteratively the so called cluster centers or centroids. These centers do not necessarily have to be an element of the data set. Further, the centers represent the clusters. For instance, the cluster k can be defined by the set of points with minimal distance to the k^{th} center.

Depending on the way data points are assigned to such a center and the corresponding cluster, the model is either a *hard* clustering or a *soft* clustering model. In hard strategies each data point is assigned to exactly one cluster. On the contrary, this is relaxed to the *degree of cluster membership* of a data point to a cluster by soft clustering methods. The well known soft clustering method fuzzy c-means will be introduced in Section 2.2. The remaining part of this section copes with hard clustering and the c-means algorithm.

For the statement of the general clustering problem we use a *dissimilarity function* $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$. A dissimilarity function in our definition such as $d(x, y) := \|x - y\|_2^2$ does not have to fulfill the properties of a metric. More precisely, we require a dissimilarity function d to be

- definite, i.e., $d(x, y) = 0$ if and only if $x = y$,
- coercive in the first variable, i.e., $d(x, y) \to \infty$ as $\|x\| \to \infty$ for fixed $y \in \mathbb{R}^m$, and
- continuous for fixed $y \in \mathbb{R}^m$.

With this notion of distance at hand we can introduce the general clustering problem. Let a dissimilarity function d, data vectors $f(j) \in \mathbb{R}^m,\ j = 1, \ldots, N$, and the number of clusters c be given. Then, the *general clustering problem* is denoted by

$$\text{(GCP)} \qquad \operatorname*{argmin}_{r_1,\ldots,r_c \in \mathbb{R}^m} \sum_{j=1}^{N} \min_{k \in \{1,\ldots,c\}} d(r_k, f(j)).$$

A minimizer $\hat{r} = (\hat{r}_k)_{k=1}^c$ is a vector of c cluster centers such that the sum of distances of every data vector $f(j)$ to its closest center is as small as possible. Obviously, a meaningful definition of a cluster k is given by

$$\mathbf{C}_k := \left\{ j : k = \min \operatorname*{argmin}_{\tilde{k}} d(r_{\tilde{k}}, f(j)) \right\}. \tag{2.1}$$

In case $|\operatorname{argmin}_{\tilde{k}} d(r_{\tilde{k}}, f(j))| > 1$, we have to decide to which cluster j belongs. To take the cluster with the smallest index k as in (2.1) is only one possible choice. Further, the membership of a data point j to a cluster k can be expressed differently.

To this end, we reformulate (GCP) in terms of characteristic vectors u_k of $\mathbf{C}_k$. For a component $u_k(j)$ of the vector

$$u = (u_k)_{k=1}^c \in \mathbb{R}^{cN}, \quad u_k = (u_k(j))_{j=1}^N$$

we require $u_k(j) = 1$ if data point j is contained in the k^{th} cluster $\mathbf{C}_k$ and $u_k(j) = 0$ otherwise. Since the clusters are disjoint, u has to fulfill $\sum_{k=1}^{c} u_k(j) = 1$ for all $j = 1, \dots, N$. The vector of characteristic functions u is also called *cluster (or segment) membership vector*.

To reformulate (GCP) using cluster membership vectors we consider

$$\begin{aligned} \min_{k\in\{1,\dots,c\}} d(r_k, f(j)) &= \min_{u(1),\dots,u(N)\in\{e_1,\dots,e_c\}} \langle d_j(r), u(j)\rangle \\ &= \min_{u(1),\dots,u(N)\in S_c} \langle d_j(r), u(j)\rangle \end{aligned} \tag{2.2}$$

where $d_j(r) = (d(r_1, f(j)), \dots, d(r_c, f(j)))^\top$, $e_k \in \mathbb{R}^c$ is the k^{th} standard basis vector, and

$$S_c = \left\{ u \in \mathbb{R}^c : \sum_{k=1}^{c} u_k = 1,\ u \geq 0 \right\} = \operatorname{conv}\{e_1, \dots, e_c\}$$

denotes the unit simplex in $\mathbb{R}^c$. To see the latter equality in (2.2), let $\hat{k} \in \operatorname{argmin}_{k\in\{1,\dots,c\}} x_k$ and note that

$$x_{\hat{k}} = \sum_{k=1}^{c} s_k x_{\hat{k}} \leq \sum_{k=1}^{c} s_k x_k = \langle s, x\rangle \text{ for all } s \in S_c \tag{2.3}$$

where $x \in \mathbb{R}^c$. Consequently, we can state (GCP) as

$$\begin{aligned} &\operatorname*{argmin}_{r\in\mathbb{R}^{cm}, u\in S_c^N} E_{\text{GC}}(u, r) \\ &\text{where } E_{\text{GC}}(u, r) := \sum_{j=1}^{N} \langle d_j(r), u(j)\rangle. \end{aligned}$$

Note that if there exist k_1, k_2, j with $d(r_{k_1}, f(j)) = d(r_{k_2}, f(j))$ for fixed r the set $\operatorname{argmin}_u E_{\text{GC}}(u, r)$ does not only contain binary vectors. If such k_1, k_2, j do not exist, the set $\operatorname{argmin}_u E_{\text{GC}}(u, r)$ contains exactly one element which must be binary due to (2.3).

If d is convex, (GCP) is a convex function for fixed u as well as for fixed r. However, the minimization over both variables is biconvex. This means that not every local minimizer or saddle point is a global minimizer. We call $(\hat{u}, \hat{r})$ a *partial minimizer* (see, e.g., [137, 69]) if

$$E(\hat{u}, \hat{r}) \leq E(u, \hat{r}) \text{ for all feasible } u \quad \text{and} \quad E(\hat{u}, \hat{r}) \leq E(\hat{u}, r) \text{ for all feasible } r \tag{2.4}$$

holds true where E is some objective functional. We call $E(\hat{u}, \hat{r})$ *partial minimum*.

Remark 2.1. *In case E is differentiable, every partial minimizer $(\hat{u}, \hat{r})$ of E contained in $\operatorname{int}\operatorname{dom} E$ is stationary, i.e., $\nabla E(\hat{u}, \hat{r}) = 0$. In general, the converse is not true (see [69, Example 4.1] for a counter example).*

That a partial minimizer does not need to be a local minimizer is depicted in Figure 2.2. Concededly, the function

$$g(x) : \mathbb{R}^2 \to \mathbb{R},\ x \mapsto x_1^4 - 6(x_1 x_2)^2 + x_2^4$$

shown in Figure 2.2 is in contrast to E_{GC} not biconvex. However, there also exist – at least in higher dimension – biconvex functions with partial minimizers that are not local minimizers, see, e.g., [18].

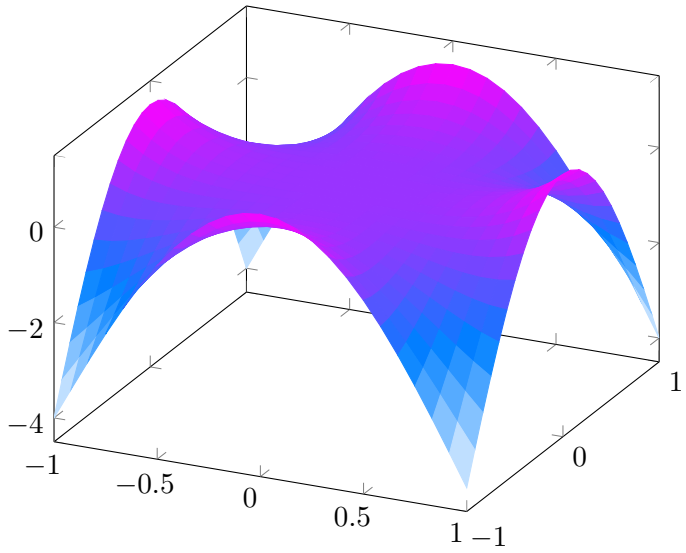


Figure 2.2.: Example of a partial minimizer that is not a local minimizer.

To find partial minimizers of the biconvex problem (GCP), we use a straightforward approach also called nonlinear Gauß-Seidel minimization algorithm [16, 156] or alternate convex search [69] in the literature. We alternatingly minimize over u and r while the other variable is fixed as stated in Algorithm 1.

Algorithm 1: Hard clustering algorithm

Input: data set f, number of clusters c
Output: cluster membership vector u and codebook r
initialization of $r^{(0)} \in \mathbb{R}^{cm}$, $\varepsilon_{\text{emp}} > 0$
repeat for $n := 0, 1, \ldots$

```
// Assign points to centers
```
$u^{(n)} \in \operatorname{argmin}_{u \in S_c^N} \sum_{j=1}^{N} \langle d_j(r^{(n)}), u(j) \rangle$

```
// Check for empty cluster
```
for $k := 1, \ldots, c$ **do**
  if $\sum_{j=1}^{N} u_k^{(n)}(j) < \varepsilon_{\text{emp}}$ **then**
    Reduce c by 1 and restart with $(r_l^{(n)})_{l \neq k}$ as initial solution
  end
end

```
// Update centers
```
$r^{(n+1)} \in \operatorname{argmin}_{r \in \mathbb{R}^{cm}} \sum_{j=1}^{N} \langle d_j(r), u^{(n)}(j) \rangle$

until $(u^{(n)}, r^{(n)})$ *is a partial minimizer*

The first obvious convergence result for Algorithm 1 is that the sequence of objective values

$(E_{\mathrm{GC}}(u^{(n)}, r^{(n)}))_{n\in\mathbb{N}}$ converges to some limit E^*_{GC}, since it is decreasing and bounded from below. Note that monotonicity and boundedness is not sufficient for the convergence of the sequence of iterates $(u^{(n)}, r^{(n)})_{n\in\mathbb{N}}$ to a partial minimizer. A counter example will be shown later (Example 2.11).

Regarding a convergence statement for the iterates $(r^{(n)}, u^{(n)})_{n\in\mathbb{N}}$ we require that

$$\hat{r} := \operatorname*{argmin}_{r\in\mathbb{R}^{cm}} E_{\mathrm{GC}}(u, r)$$

is unique for all $u \in S_c^N \setminus \{v \in S_c^N : \text{ there exists } k \in \{1, \dots, c\} \text{ with } \sum_{j=1}^N v_k(j) = 0\}$.

Remark 2.2. *The choice $d(r_k, f(j)) := \|r_k - f(j)\|_2$ results in the weighted median*

$$\hat{r}_k \in \operatorname*{argmin}_{r\in\mathbb{R}^{cm}} \sum_{j=1}^{N} u_k(j) \|r_k - f(j)\|_2 . \tag{2.5}$$

The problem in (2.5) *has a unique solution, if there exist two indices $j_1, j_2 \in \{1, \dots, N\}$ such that $f(j_1)$ and $f(j_2)$ are not collinear [139]. For vector-valued (e.g., colored) images $f : \{1, \dots, N\} \to \mathbb{R}^m$ this is fulfilled in most cases, especially for real world applications. On the other hand, gray-valued images $f : \{1, \dots, N\} \to \mathbb{R}$, of course, contain always exclusively collinear data points.*

Then, we obtain convergence of Algorithm 1 in the sense of Zangwill's Theory as shown in more detail in Section 2.3.1. More precisely, every sequence $(r^{(n)}, u^{(n)})_{n\in\mathbb{N}}$ generated by Algorithm 1 converges to a partial minimizer or contains a subsequence converging to a partial minimizer and every convergent subsequence of $(r^{(n)}, u^{(n)})_{n\in\mathbb{N}}$ has a partial minimizer as limit point.

Moreover, with one further restriction we obtain a stronger convergence result. Assume that

$$u^{(n)} \in \{e_1, \dots, e_c\}^N \text{ for all } n \in \mathbb{N}.$$

This means that in case $d(r_{k_1}^{(n)}, f(j)) = d(r_{k_2}^{(n)}, f(j))$ and $\{k_1, k_2\} \subseteq \operatorname{argmin}_k d(r_k^{(n)}, f(j))$ we choose either $u^{(n)}(j) = e_{k_1}$ or $u^{(n)}(j) = e_{k_2}$. Then, Algorithm 1 terminates after finitely many steps due to [137]. For the proof, see Corollary 2.18.

As stated in [156], several popular hard clustering algorithms can be obtained as special cases of Algorithm 1. As one of the most popular clustering algorithms, we obtain c-means from Algorithm 1 by setting $d(x, y) := \|x - y\|_2^2$. It is simple to implement and efficient in practical applications, see, e.g., [106, 102, 21, 76, 73]. However, Aloise et al. [2] show that (GCP) with $d(x, y) = \|x - y\|_2^2$ is an NP-hard problem. The c-means method is stated in Algorithm 2.

Algorithm 2: c-Means

Input: data set f, number of clusters c
Output: cluster membership vector u and codebook of cluster centers r
initialization of $r^{(0)} \in \mathbb{R}^{cm}$
repeat for $n = 0, 1, \ldots$

```
// Assign points to centers
```
for $j := 1, \ldots, N$ **do**
  $\{1, \ldots, c\} \supseteq \hat{K} = \operatorname{argmin}_k \left\| r_k^{(n)} - f(j) \right\|_2^2$
  Choose arbitrary $\hat{k} \in \hat{K}$ with $u_{\hat{k}}^{(n)}(j) = 1$ and $u_k^{(n)}(j) = 0$ for $k \neq \hat{k}$
end

```
// Check for empty cluster
```
for $k := 1, \ldots, c$ **do**
  if $\sum_{j=1}^{N} u_k^{(n)}(j) < 1$ **then**
    Reduce c by 1 and restart with $(r_l^{(n)})_{l \neq k}$ as initial solution
  end
end

```
// Update centers
```
for $k := 1, \ldots, c$ **do**
  $r_k^{(n+1)} = \frac{\sum_{j=1}^{N} u_k^{(n)}(j) f(j)}{\sum_{j=1}^{N} u_k^{(n)}(j)}$
end

until $r^{(n+1)} = r^{(n)}$

Note that $(u^{(n)}, r^{(n)})$ is a partial minimizer if and only if the stopping criterion $r^{(n+1)} = r^{(n)}$ of Algorithm 2 holds true. The "if" direction follows from

$$\begin{aligned} E_{\mathrm{GC}}(u^{(n)}, r^{(n)}) &\leq E_{\mathrm{GC}}(u, r^{(n)}) && \text{for all } u \in \{e_1, \ldots, e_c\}^N, \\ E_{\mathrm{GC}}(u^{(n)}, r^{(n)}) = E_{\mathrm{GC}}(u^{(n)}, r^{(n+1)}) &\leq E_{\mathrm{GC}}(u^{(n)}, r) && \text{for all } r \in \mathbb{R}^{cm}. \end{aligned}$$

For the "only if" direction, let $(u^{(n)}, r^{(n)})$ be a partial minimizer. This especially implies

$$E_{\mathrm{GC}}(u^{(n)}, r^{(n)}) \leq E_{\mathrm{GC}}(u^{(n)}, r) \text{ for all feasible } r$$

and hence $r^{(n)} = r^{(n+1)}$ since $r^{(n+1)}$ is the unique solution of $\operatorname{argmin}_r E_{\mathrm{GC}}(u^{(n)}, r)$.

2.2 From Means to Fuzzy c-Means

In this section we review the well known fuzzy c-means algorithm [53, 20, 77, 18]. Fuzzy c-means is a *soft* (or fuzzy) center-based partitioning clustering algorithm. The difference to *hard*

center-based partitioning clustering algorithms introduced in the previous section is the non-binary assignment of data points to clusters. This means that a data point has a certain grade of membership to each cluster. This grade of membership is represented by a real number between 0 and 1. A higher number means a stronger association between data point and cluster.

We derive the popular soft clustering algorithm fuzzy c-means similar to Teboulle [156]. Therefore, we introduce in Section 2.2.1 means and some of their properties. Further, we present novel results on the representation of p-means as maxima of linear functions over specific sets.

In Section 2.2.2, we approximate the general clustering problem (GCP) using power means and obtain fuzzy c-means as alternating minimization algorithm. In contrast to Teboulle [156], we use the aforementioned novel result on the representation of p-means for this approximation. More precisely, we show that the clustering model obtained by the approximation in terms of p-means is equivalent to the fuzzy c-means model proposed in [20]. In contrast, Teboulle [156] shows that the resulting algorithms are equivalent disregarding the case where cluster centers r_k and data points $f(j)$ coincide.

2.2.1 Means

In the following we assume that $\phi : (0, +\infty) \to \mathbb{R}$ is a continuous, and strictly increasing or decreasing function. Then ϕ^{-1} exists on the image of ϕ and is again continuous, and strictly increasing or decreasing, respectively. For $\gamma \in \mathbb{R}^n$ with

$$\sum_{i=1}^{n} \gamma_i = 1, \quad \gamma > 0$$

we define the mean $m_{\phi,\gamma} : \mathbb{R}^n_{>0} \to \mathbb{R}$ with respect to γ and ϕ by

$$m_{\phi,\gamma}(x) := \phi^{-1}\left(\sum_{i=1}^{n} \gamma_i \phi(x_i)\right), \quad x > 0.$$

We extend the definition of $m_{\phi,\gamma}$ to $\mathbb{R}^n_{\geq 0}$ if possible. The following simple property states that means are bounded by the minimal and maximal component of $x = (x_i)_{i\in\mathcal{I}}$ where $\mathcal{I} := \{1, \ldots, n\}$, cf. [75, Theorem 82].

Lemma 2.3. *The function $m_{\phi,\gamma}$ fulfills*

$$\min_{i\in\mathcal{I}} x_i \leq m_{\phi,\gamma}(x) \leq \max_{i\in\mathcal{I}} x_i.$$

If ϕ is increasing and convex or decreasing and concave, then $\sum_{i=1}^n \gamma_i x_i \leq m_{\phi,\gamma}(x)$. If ϕ is increasing and concave or decreasing and convex, then $\sum_{i=1}^n \gamma_i x_i \geq m_{\phi,\gamma}(x)$.

Proof. Let $x_a := \min_{i\in\mathcal{I}} x_i$ and $x_b := \max_{i\in\mathcal{I}} x_i$. If ϕ is increasing, we obtain by definition of γ that $\phi(x_a) \leq \sum_{i=1}^n \gamma_i \phi(x_i) \leq \phi(x_b)$ and thus

$$x_a \leq \phi^{-1}\left(\sum_{i=1}^{n} \gamma_i \phi(x_i)\right) \leq x_b.$$

Similarly, we can reason for decreasing ϕ.

If ϕ is convex, we have by Jensen's inequality that $\phi\left(\sum_{i=1}^n \gamma_i x_i\right) \leq \sum_{i=1}^n \gamma_i \phi(x_i)$ which implies for increasing ϕ^{-1} that $\sum_{i=1}^n \gamma_i x_i \leq m_{\phi,\gamma}(x)$. The other cases can be proven analogously. □

Well known examples of means are the p-means for $p \in [-\infty, \infty]$ where often $\gamma_i = \frac{1}{n}$, $i \in \mathcal{I}$, see, e.g., [75]. These means follow for $x > 0$ from the above definition by setting

$$\phi(x) := \begin{cases} x^p & \text{if } p \neq 0, \\ \log x & \text{if } p = 0, \end{cases} \qquad \phi^{-1}(x) = \begin{cases} x^{\frac{1}{p}} & \text{if } p \neq 0, \\ e^x & \text{if } p = 0. \end{cases}$$

Then, for $x \geq 0$, the *p-means (power means, Hölder means)* are defined by

$$m_{p,\gamma} := \begin{cases} \left(\sum_{i=1}^n \gamma_i x_i^p \right)^{1/p} & \text{if } p < 0 \text{ and there does not exist a } j \in \mathcal{I} \text{ with } x_j = 0 \text{ or } p > 0, \\ \prod_{i=1}^n x_i^{\gamma_i} & \text{if } p = 0, \\ 0 & \text{if } p < 0 \text{ and there exists a } j \in \mathcal{I} \text{ with } x_j = 0. \end{cases}$$

For $\gamma = 1/n$ we use the notation $m_p := m_{p,\gamma}$.

Regarding the limiting cases of p, let $p > 0$ and $b \in \operatorname{argmax}_{i \in \mathcal{I}} x_i$. This yields

$$\gamma_b^{\frac{1}{p}} x_b \leq m_{p,\gamma}(x) \leq x_b$$

so that

$$\lim_{p \to +\infty} m_{p,\gamma}(x) = \max_{i \in \mathcal{I}} x_i =: m_{\infty,\gamma}(x).$$

Further,

$$\lim_{p \to -\infty} m_{p,\gamma}(x) = \min_{i \in \mathcal{I}} x_i =: m_{-\infty,\gamma}(x)$$

is trivial if any x_i is zero and follows otherwise by

$$\lim_{p \to -\infty} m_{p,\gamma}(x) = \frac{1}{\lim_{p \to \infty} m_{p,\gamma}\left(\frac{1}{x}\right)} = \frac{1}{\max_{i \in \mathcal{I}} \frac{1}{x_i}} = \min_{i \in \mathcal{I}} x_i, \quad x > 0.$$

To compute $\lim_{p \to 0} m_{p,\gamma}$ let $x > 0$ and observe that by the rule of l'Hopital

$$\lim_{p \to 0} \frac{\log(\sum_{i=1}^n \gamma_i x_i^p)}{p} = \lim_{p \to 0} \frac{\sum_{i=1}^n \gamma_i x_i^p \log(x_i)}{\sum_{i=1}^n \gamma_i x_i^p} = \frac{\sum_{i=1}^n \gamma_i \log(x_i)}{\sum_{i=1}^n \gamma_i} = \sum_{i=1}^n \log(x_i^{\gamma_i}) = \log(\prod_{i=1}^n x_i^{\gamma_i}).$$

Hence, we obtain

$$\lim_{p \to 0} m_{p,\gamma} = \lim_{p \to 0} \exp\left(\frac{\log(\sum_{i=1}^n \gamma_i x_i^p)}{p} \right) = \prod_{i=1}^n x_i^{\gamma_i}.$$

In case there exists i with $x_i = 0$ we have $\lim_{p \to -0} m_{p,\gamma} = 0 = m_{0,\gamma}$ by definition. Further, let $\mathcal{I}_0 := \{i : x_i = 0\}$. Then,

$$\lim_{p \to +0} m_{p,\gamma} = \lim_{p \to +0} \Big(\sum_{i \in \mathcal{I} \setminus \mathcal{I}_0} \gamma_i \Big)^{\frac{1}{p}} \Big(\sum_{i \in \mathcal{I} \setminus \mathcal{I}_0} \underbrace{\frac{\gamma_i}{\sum_{i \in \mathcal{I} \setminus \mathcal{I}_0} \gamma_i}}_{\tilde{\gamma}_i} x_i^p \Big)^{\frac{1}{p}} = \lim_{p \to +0} \Big(\sum_{i \in \mathcal{I} \setminus \mathcal{I}_0} \gamma_i \Big)^{\frac{1}{p}} m_{p,\tilde{\gamma}}(x) = 0$$

due to the boundedness of $m_{p,\tilde{\gamma}}$. Table 2.1 shows some specifically named p-means.

We collect further known properties of means in the subsequent Theorem 2.5. For the proof of its first part we utilize the following technical lemma.

$p = -\infty$	minimum	$m_{-\infty} = \min_{i \in \mathcal{I}} x_i$
$p = -1$	harmonic mean	$m_{-1}(x) = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$
$p = 0$	geometric mean	$m_0(x) = (\prod_{i=1}^n x_i)^{\frac{1}{n}}$
$p = 1$	arithmetic mean	$m_1(x) = \frac{1}{n} \sum_{i=1}^n x_i$
$p = 2$	quadratic mean	$m_2(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$
$p = \infty$	maximum	$m_\infty = \max_{i \in \mathcal{I}} x_i$

Table 2.1.: Special p-means $m_p := m_{p,\gamma}$ for $\gamma_i = \frac{1}{n}$, $i \in \mathcal{I}$.

Lemma 2.4. *Let $\psi \in \mathcal{C}^3((0,\infty))$ be strictly increasing and strictly concave, i.e., $\psi'(t) > 0$ and $\psi''(t) < 0$ for all $t \in (0,\infty)$, respectively. Then, $h(t_1,t_2) := (\psi(t_1) - \psi(t_2))/\psi'(t_2)$ is concave if and only if $1/\rho(t)$ is convex where $\rho(t) := -\phi''(t)/\phi'(t)$ and $\phi := \psi^{-1}$.*

Proof. This proof works analogously to the proof of [14, Lemma 1, p. 1449]. We show that the Hessian $\nabla^2 h(t_1,t_2)$ is negative semi-definite if and only if $1/\rho(t)$ is convex. We have

$$\nabla^2 h(x) = \begin{pmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{pmatrix}$$

where $h_{ij} = \frac{\partial h(t_1,t_2)}{\partial x_i \partial x_j}$, that is,

$$h_{11} = \frac{\psi''(t_1)}{\psi'(t_2)}, \qquad h_{12} = -\frac{\psi'(t_1)\psi''(t_2)}{\psi'^2(t_2)},$$
$$h_{22} = \frac{\psi''(t_2)}{\psi'(t_2)} + h(t_1,t_2)\left(\frac{2\psi''^2(t_2) - \psi'''(t_2)\psi'(t_2)}{\psi'^2(t_2)}\right).$$

Since $h_{11} < 0$ the matrix ∇^2 is negative semi-definite if and only if

$$h_{11}h_{22} - h_{12}^2 \geq 0 \quad \text{or equivalently} \quad h_{22} - \frac{h_{12}^2}{h_{11}} \leq 0. \tag{2.6}$$

Further, by the inverse function theorem we obtain

$$(\psi^{-1}(t))' = \frac{1}{\psi'(\psi^{-1}(t))}, \qquad (\psi^{-1}(t))'' = -\frac{\psi''(\psi^{-1}(t))}{\psi'^3(\psi^{-1}(t))},$$
$$(\psi^{-1}(t))''' = \frac{3\psi''^2(\psi^{-1}(t)) - \psi'''(\psi^{-1}(t))\psi'(\psi^{-1}(t))}{\psi'^5(\psi^{-1}(t))}$$

and hence

$$\rho(\psi(t)) = \frac{\psi''(t)}{\psi'^2(t)}, \qquad \rho'(\psi(t)) = \frac{\psi'(t)\psi'''(t) - 2\psi''^2(t)}{\psi'^4(t)} \tag{2.7}$$

where $\rho'(\psi(t))$ denotes the value of $\rho'(\cdot)$ at $\psi(t)$. Rewriting (2.6) yields

$$\begin{aligned} & \frac{\psi''^2(t_2)\psi'^2(t_1)\psi'(t_2)}{\psi'^4(t_2)\psi''(t_1)} \geq \frac{\psi''(t_2)}{\psi'(t_2)} - \frac{(\psi(t_1)-\psi(t_2))}{\psi'(t_2)} \frac{\psi'(t_2)\psi'''(t_2) - 2\psi''^2(t_2)}{\psi'^2(t_2)} \\ \Leftrightarrow \quad & \frac{\psi''^2(t_2)}{\psi'^4(t_2)} \leq \frac{\psi''(t_2)\psi''(t_1)}{\psi'^2(t_2)\psi'^2(t_1)} \\ & \qquad - (\psi(t_1)-\psi(t_2)) \frac{\psi''(t_1)}{\psi'^2(t_1)} \frac{\psi'(t_2)\psi'''(t_2) - 2\psi''^2(t_2)}{\psi'^4(t_2)} \end{aligned} \tag{2.8}$$

where the change of the inequality sign is due to the negativity of ψ''. We plug (2.7) into (2.8) and obtain

$$\rho^2(\psi(t_2)) \leq \rho(\psi(t_2))r(\psi(t_1)) - (\psi(t_1)-\psi(t_2))\rho(\psi(t_1))\rho'(\psi(t_2)).$$

Setting $s_i = \psi(t_i),\ i = 1, 2$ and dividing by $\rho(s_1)\rho^2(s_2) < 0$ yields

$$\frac{1}{\rho(s_1)} \geq \frac{1}{\rho(s_2)} - (s_1 - s_2)\frac{\rho'(s_2)}{\rho^2(s_2)}$$

which is a gradient inequality equivalent to convexity of $1/\rho$. □

Let us turn to the properties of means announced above.

Theorem 2.5 (Properties of Means)**.** *Let $\phi, \psi : (0, +\infty) \to \mathbb{R}$ be continuous and strictly monotonic functions. Then, for $x > 0$ the following relations hold true:*

i) *If $\phi \in \mathcal{C}^3(0, \infty)$ is strictly convex and strictly increasing, then $m_{\phi,\gamma}$ is convex if and only if $\rho(t) = -\frac{\phi''(t)}{\phi'(t)}$ is convex.*

ii) *Let ϕ be strictly increasing. Then, we have*

$$m_{\psi,\gamma}(x) \leq m_{\phi,\gamma}(x) \text{ for all } x$$

if and only if $\phi \circ \psi^{-1}$ is convex. Moreover, equality holds true if and only if $\phi = \alpha\psi + \beta$ for some $\alpha, \beta \in \mathbb{R}$, $\alpha \neq 0$.

iii) *If $m_{\phi,\gamma}(kx) = k\, m_{\phi,\gamma}(x)$ for $k > 0$, then there exists $p \in \mathbb{R}$ such that $m_{\phi,\gamma} = m_{p,\gamma}$. In other words, the only positively homogeneous means are the p-means.*

Proof. i) See [156, Lemma 10]. Let $\xi(z) := \sum_{i=1}^n \gamma_i \phi(z_i)$. Then, the mean $m_{\phi,\gamma}(z) = \phi^{-1}(\xi(z))$ is convex if and only if

$$\frac{\phi^{-1}(\xi(y)) - \phi^{-1}(\xi(x))}{(\phi^{-1})'(\xi(x))} \geq \sum_{i=1}^n \gamma_i (y_i - x_i)\phi'(x_i). \tag{2.9}$$

To show that (2.9) is fulfilled let

$$H(s,t) := \frac{\phi^{-1}(s) - \phi^{-1}(t)}{(\phi^{-1})'(t)}.$$

By Lemma 2.4 the function H is concave if and only if $t \mapsto 1/\rho(t)$ is convex. Thus, it remains to show that the concavity of H is equivalent to (2.9). By Jensen's Inequality we obtain

$$H(\xi(y), \xi(x)) \geq \sum_{i=1}^{n} \gamma_i H(\phi(y_i), \phi(x_i)) = \sum_{i=1}^{n} \gamma_i \frac{y_i - x_i}{(\phi^{-1})'(\phi(x_i))}.$$

Using $1/\phi^{-1}(t_0) = \phi'(\phi^{-1}(t_0))$ at $t_0 := \phi(x_i)$ concludes the proof.

ii) For the first part, see [75, Theorems 83 and 92]. Let $\vartheta(x) := \phi(\psi^{-1}(x))$ and $a = \psi(x)$ for fixed x. Consequently, $m_{\phi,\gamma}(x) \leq m_{\psi,\gamma}(x)$ if and only if

$$\vartheta(\sum_{i=1}^{n} \gamma_i a_i) \leq \sum_{i=1}^{n} \gamma_i \vartheta(a_i).$$

if ϕ is strictly increasing.

The second part is stated in [75, Theorem 83]. "$\Leftarrow$": Assume $\phi = \alpha\psi + \beta$. Then,

$$\phi(m_{\phi,\gamma}(x)) = \sum_{i=1}^{n} \gamma_i \phi(x_i) = \alpha \sum_{i=1}^{n} \gamma_i \psi(x_i) + \beta = \alpha\psi(m_{\psi,\gamma}) + \beta = \phi(m_{\psi,\gamma}(x)).$$

"$\Rightarrow$": Assume $\phi(m_{\phi,\gamma}(x)) = \phi(m_{\psi,\gamma}(x))$ for all $x \in \mathbb{R}^2$. For some fixed $x \in \mathbb{R}^2$, let $\gamma_1 = \frac{x_2 - t}{x_2 - x_1}$ and $\gamma_2 = \frac{t - x_1}{x_2 - x_1}$ where $x_1 < t < x_2$. Then,

$$a := \psi^{-1}\left(\frac{x_2 - t}{x_2 - x_1}\psi(x_1) + \frac{t - x_1}{x_2 - x_1}\psi(x_2)\right) = \phi^{-1}\left(\frac{x_2 - t}{x_2 - x_1}\phi(x_1) + \frac{t - x_1}{x_2 - x_1}\phi(x_2)\right)$$

for $x_1 \leq t \leq x_2$. This implies

$$\begin{aligned} \phi(a) &= \frac{x_2 - t}{x_2 - x_1}\phi(x_1) + \frac{t - x_1}{x_2 - x_1}\phi(x_2) \\ &= \frac{\psi(x_2) - \psi(a)}{\psi(x_2) - \psi(x_1)}\phi(x_1) + \frac{\psi(a) - \psi(x_1)}{\psi(x_2) - \psi(x_1)}\phi(x_2) = \alpha\psi(a) + \beta \end{aligned}$$

for all $a \in [x_1, x_2]$.

iii) See [75, Theorems 84]. We suppose without loss of generality that $\phi(0) = 1$, since one can replace $\phi(x)$ by $\phi(x) - \phi(1)$. Further, we obtain by assumption

$$m_{\phi,\gamma}(x) = \frac{1}{k} m_{\phi,\gamma}(kx) = \frac{1}{k}\phi^{-1}\left(\sum_{i=1}^{n} \gamma_i \phi(kx)\right) = m_{\psi,\gamma}(x)$$

for $k > 0$ where $\psi(t) = \phi(kt)$. With part ii we have

$$\phi(kt) = \alpha(k)\phi(t) + \beta(k) \tag{2.10}$$

where α and β depend on k and $\alpha(k) \neq 0$. Moreover, $\phi(k) = \beta(k)$ follows from setting $t = 1$ in (2.10). After substituting $k = s$, this implies

$$\phi(st) = \alpha(s)\phi(t) + \phi(s), \quad t, s > 0 \tag{2.11}$$

and analogously

$$\phi(ts) = \alpha(t)\phi(s) + \phi(t), \quad t, s > 0.$$

Hence,

$$\frac{\alpha(t) - 1}{\phi(t)} = \frac{\alpha(s) - 1}{\phi(s)} =: c$$

for $t, s \neq 1$ which is not a restriction as we will see in (2.12). Hence with (2.11),

$$\phi(st) = c\phi(x)\phi(y) + \phi(x) + \phi(y). \tag{2.12}$$

If $c = 0$ (2.12) can be written as $\phi(xy) = \phi(x) + \phi(y)$ and the continuous solution is given by $\phi = C\log(\cdot)$. If $c \neq 0$ we set $c\phi(x) + 1 = f(x)$ and (2.12) can be written as $f(st) = f(s)f(t)$ yielding $f(t) = t^p$ and hence

$$\phi(t) = \frac{x^p - 1}{c}.$$

□

Concerning the p-means, Theorem 2.5 ii implies for $x \geq 0$ and $t \leq s$ in particular

$$m_{t,\gamma}(x) \leq m_{s,\gamma}(x).$$

Moreover, the next corollary follows from Theorem 2.5.

Corollary 2.6. i) *For $p \geq 1$, the function $m_{p,\gamma}$ is convex on $\mathbb{R}^n_{\geq 0}$.*

ii) *For $p < 1$, the function $-m_{p,\gamma}$ is convex on $\mathbb{R}^n_{\geq 0}$.*

Before we state our main result of this section, let us recall *Hölder's inequality* [75] and its *reverse* version. To this end, we define

$$\|x\|_p := \left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}} \text{ for } p \in (-\infty, 1) \quad \text{and} \quad \|x\|_{-\infty} := \min_{i \in \mathcal{I}} |x_i|.$$

Of course, $\|x\|_p$ for $p < 1$ is not a norm. Further, let p and q denote conjugated exponents, i.e., $1/p + 1/q = 1$ where $1/\infty = 0$.

Theorem 2.7. i) (Hölder's inequality) *Let $p \geq 1$. Then, for any $x, u \in \mathbb{R}^n$ the relation*

$$\sum_{i=1}^n |x_i u_i| \leq \|x\|_p \|u\|_q$$

holds true where equality is attained if and only if there exists $\alpha \neq 0$ with $|x_i| = \alpha |u_i|^{\frac{1}{p-1}}$ for all $i \in \mathcal{I}$.

ii) (Reverse Hölder's inequality) *Let $p \in (0,1]$, i.e. $q = \frac{p}{p-1} \in [-\infty, 0)$. Here, $p = 1$ corresponds to $q = -\infty$. Then, for any $x, u \in \mathbb{R}^n$ the relation*

$$\sum_{i=1}^{n} |x_i u_i| \geq \|x\|_p \|u\|_q$$

holds true, where $\|u\|_q := 0$ if u has zero coefficients. Equality is attained in the case $p \neq 1$ if and only if either

- *$u \neq 0$ and there exists $\alpha \neq 0$ with $|x_i| = \alpha |u_i|^{\frac{1}{p-1}}$ for all $i \in \mathcal{I}$ or*
- *$|u_i| = 0$ for $i \in \mathcal{I}_0 \subset \mathcal{I}$ and $|x_j| = 0$ for $j \in \mathcal{I} \backslash \mathcal{I}_0$.*

If $p = 1$, then equality is attained if $\|x\|_1 = 1$ and $x_j = 0$ for all $j \in \mathcal{I}$ with $u_j \neq \min_{i \in \mathcal{I}} |u_i|$.

Subsequently, we state the main result of this section. To obtain the fuzzy c-means model as an approximation of hard clustering in the next section, part iii of the following novel theorem will be of interest.

Theorem 2.8. *Let*

$$\mathcal{S}_{p,\gamma} := \left\{ u \in \mathbb{R}^n_{\geq 0} : \left\| \gamma^{-\frac{1}{p}} u \right\|_q = m_{q,\gamma}(u/\gamma) = 1 \right\}.$$

i) *If $p \geq 1$ and $x \geq 0$, then*

$$m_{p,\gamma}(x) = \max_{u \in \mathcal{S}_{p,\gamma}} \langle x, u \rangle$$

where the maximum is attained at $u = \gamma x^{p-1} / \left\| \gamma^{\frac{1}{q}} x^{p-1} \right\|_q$.

ii) *If $p \in (0,1)$ and $x \geq 0$, then*

$$-m_{p,\gamma}(x) = \sup_{u \in \mathcal{S}_{p,\gamma}} \langle x, -u \rangle.$$

iii) *If $p \in [-\infty, 0)$ and $x \geq 0$, then*

$$-m_{p,\gamma}(x) = \max_{u \in \mathcal{S}_{p,\gamma}} \langle x, -u \rangle,$$

where the maximum is attained for $u = \gamma x^{p-1} / \left\| \gamma^{\frac{1}{q}} x^{p-1} \right\|_q$ in case $x > 0$, and for $u_i = 0$ for all $i \in \mathcal{I} \backslash \mathcal{I}_0$ and $\left\| \gamma^{-\frac{1}{p}} u \right\|_q = 1$ in case $x_i = 0$ for all $i \in \mathcal{I}_0 \subset \mathcal{I}$.

Proof. i) Let $p \geq 1$. Then, for all $x \in \mathbb{R}^n$ we know by the dual norm relation that

$$\begin{aligned} \left\| \gamma^{\frac{1}{p}} x \right\|_p &= \max \left\{ |\langle \gamma^{\frac{1}{p}} x, w \rangle| : \|w\|_q \leq 1 \right\} \\ &= \max \left\{ |\langle x, u \rangle| : \left\| \gamma^{-\frac{1}{p}} u \right\|_q \leq 1 \right\}. \end{aligned}$$

Hence, for $x \geq 0$ we obtain

$$m_{p,\gamma}(x) = \left\|\gamma^{\frac{1}{p}} x\right\|_p = \max\{\langle x, u\rangle : \left\|\gamma^{-\frac{1}{p}} u\right\|_q \leq 1,\ u \geq 0\},$$

where by Theorem 2.7 i equality can be attained for some $u = \gamma x^{p-1} / \left\|\gamma^{\frac{1}{q}} x^{p-1}\right\|_q$. This proves part i.

ii) Let $p \in (0,1)$. Then we obtain for $x \geq 0$ by the reverse Hölder inequality

$$\langle \gamma^{\frac{1}{p}} x, \gamma^{-\frac{1}{p}} (|u_i|)_{i=1}^n \rangle \geq m_{p,\gamma}(x) \left\|\gamma^{-\frac{1}{p}} u\right\|_q \quad \text{for all } u \in \mathbb{R}^n.$$

This implies

$$\langle x, -u\rangle \leq -m_{p,\gamma}(x) \left\|\gamma^{-\frac{1}{p}} u\right\|_q \quad \text{for all } u \geq 0$$

where equality holds true for $x > 0$ and $\gamma^{-\frac{1}{p}} u = \alpha(\gamma^{\frac{1}{p}} x)^{p-1}$, i.e., $u = \alpha\gamma x^{p-1}$, $\alpha \neq 0$. Consequently, if $\|\gamma^{-\frac{1}{p}} u\|_q \geq 1$, then

$$\langle x, -u\rangle \leq -m_{p,\gamma}(x) \tag{2.13}$$

with equality in the case $x > 0$ for $u = \alpha\gamma x^{p-1}$ with $\|\gamma^{-\frac{1}{p}} \alpha\gamma x^{p-1}\|_q = 1$, i.e., $\alpha = 1/\|\gamma^{\frac{1}{p}} x^{p-1}\|_q$. If $x_i = 0$ for $i \in \mathcal{I}_0 \subset \mathcal{I}$, we can choose, e.g., a sequence $u^{(j)} \in \mathbb{R}^n$ satisfying $u \geq 0$ for sufficiently large j and $\|\gamma^{-\frac{1}{p}} u^{(j)}\|_q = 1$ for all j as follows. Set $u_i^{(j)} := \beta^{(j)} \gamma_i x_i^{p-1}$, $i \in \mathcal{I}\backslash\mathcal{I}_0$ and $u_i^{(j)} := \gamma_i^{\frac{1}{q}} j$, $i \in \mathcal{I}_0$ where $\beta^{(j)} := \big((1 - |\mathcal{I}_0| j^q)/(\sum_{i\in\mathcal{I}\backslash\mathcal{I}_0} \gamma_i x_i^p)\big)^{\frac{1}{q}}$. Then $-m_{p,\gamma}(x) = \lim_{j\to\infty} \langle x, -u^{(j)}\rangle$.

iii) Let $p \in [-\infty, 0)$. Then (2.13) can be derived for $x > 0$ by applying the reverse Hölder inequality with flipped p and q. If $x_i = 0$ for $i \in \mathcal{I}_0 \subset \mathcal{I}$, we obtain the equality

$$-m_{p,\gamma}(x) = 0 = \langle x, -u\rangle, \quad u \in \mathcal{S}_{p,\gamma}$$

exactly by choosing $u_i = 0$ for all $i \in \mathcal{I}\backslash\mathcal{I}_0$ and $u_i \geq 0$, $i \in \mathcal{I}_0$ such that $\left\|\gamma^{-\frac{1}{p}} u\right\|_q = 1$.

□

For a visualization of $\mathcal{S}_{p,\gamma} \subset \mathbb{R}^2$, see Figure 2.3.

2.2.2 Fuzzy c-Means

The work of Teboulle [156] introduces an optimization framework for clustering methods that contains the fuzzy c-means algorithm as a special case. Similarly to [156], we derive in this subsection the fuzzy c-means as the alternating minimization procedure of an approximation of the general clustering problem as stated in (2.2) in terms of p-means.

In the general clustering problem (2.2) we are given vectors $f(j) \in \mathbb{R}^m, j = 1, \ldots, N$, and the number of clusters c. Further, we set $d_j(r) := \big(d(r_1, f(j)), \ldots, d(r_c, f(j))\big)^\top$ where $r = (r_k)_{k=1}^c \in$

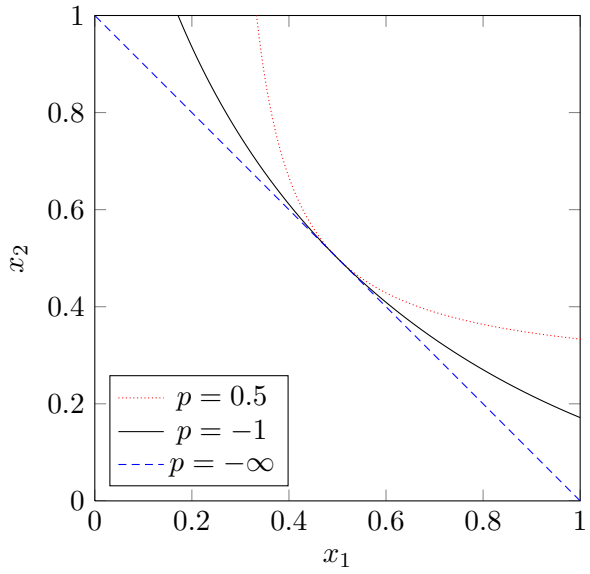


Figure 2.3.: The closed set $\mathcal{S}_{p,\gamma} \subset \mathbb{R}^2$ according to different values of $p < 1$. $n = 2$, $\gamma_1 = \gamma_2 = 0.5$.

$\mathbb{R}^{cm}$, and $\gamma := 1/c$ as well as $m_p := m_{p,\gamma}$. We want to find centers $r_k \in \mathbb{R}^m, k = 1, \ldots, c$ by solving

$$\text{(GCP)} \qquad \operatorname*{argmin}_{r_1,\ldots,r_c} \sum_{j=1}^{N} \min_{k\in\{1,\ldots,c\}} d(r_k, f(j)) = \operatorname*{argmin}_{r_1,\ldots,r_c} \sum_{j=1}^{N} m_{-\infty}(d_j(r)).$$

To approximate (GCP), we simply replace $-\infty$ by some finite $p < 0$ and obtain the *approximate clustering problem* given by

$$\text{(ACP)} \qquad \operatorname*{argmin}_{r\in\mathbb{R}^{cm}} E_{\mathrm{AC}}(r)$$
$$\text{where } E_{\mathrm{AC}}(r) := \sum_{j=1}^{N} m_p(d_j(r)).$$

Similarly to (2.2), we can reformulate E_{AC} with Theorem 2.8 iii and obtain

$$E_{\mathrm{AC}}(r) = \sum_{j=1}^{N} m_p(d_j(r)) = \sum_{j=1}^{N} \min_{w(j)\in\mathcal{S}_p} \langle w(j), d_j(r)\rangle = \min_{w\in\mathcal{S}_p^N} \sum_{j=1}^{N} \langle w(j), d_j(r)\rangle.$$

For the original formulation of the fuzzy c-means functional [20] we substitute $u(j) = (w(j))^q$ yielding

$$E_{\mathrm{AC}}(r) = \min_{u\in S_c^N} \sum_{j=1}^{N} \langle (u(j))^{\frac{1}{q}}, d_j(r)\rangle = \min_{u\in S_c^N} \sum_{j=1}^{N} \langle (u(j))^{\beta}, d_j(r)\rangle \tag{2.14}$$

where we fix $d(r_k, f(j)) := \|r_k - f(j)\|_2^2$ for the remaining part of this Section 2.2.2 and set $\beta := 1/q > 1$. The fuzzy c-means algorithm consists of alternating optimization with respect

to u and r as stated in Algorithm 3 for $\beta = 2$, i.e., $p = -1/2$. The case $\beta = 1$, i.e., $p = -\infty$, resembles the reformulation of (GCP) stated in (2.2).

Due to Teboulle [156], one can derive fuzzy c-means also as Weiszfeld's algorithm [139, 160, 164, 165] of (ACP) as long as $d_j(r) > 0$, i.e., $f(j) \neq r_k$ for all j, k. As stated in [156], Weiszfeld's algorithm aims to find a codebook r that solves

$$\nabla \left(\sum_{j=1}^{N} m_p(d_j(r)) \right) = 0$$

with a corresponding fixed point iteration. More precisely, we obtain with straightforward calculation

$$\frac{\partial(m_p(d_j(r)))}{\partial r_k} = \frac{d(r_k, f(j))^{p-1} 2(r_k - f(j))}{\left(\sum_{\bar{k}=1}^{c} d(r_{\bar{k}}, f(j))^p\right)^{1/q}} = 2(r_k - f(j)) \frac{d(r_k, f(j))^{\frac{\beta}{(1-\beta)}}}{\left(\sum_{\bar{k}=1}^{c} d(r_{\bar{k}}, f(j))^{\frac{1}{(1-\beta)}}\right)^{\beta}}$$

where we have fixed $d(r_k, f(j)) = \|r_k - f(j)\|_2^2$. This yields

$$\sum_{j=1}^{N} 2(r_k - f(j)) \Bigg(\underbrace{\frac{d(r_k, f(j))^{\frac{1}{(1-\beta)}}}{\sum_{\bar{k}=1}^{c} d(r_{\bar{k}}, f(j))^{\frac{1}{(1-\beta)}}}}_{=:u_k(j)} \Bigg)^{\beta} = 0$$

and hence

$$r_k = \frac{\sum_{j=1}^{N} (u_k(j))^{\beta} f(j)}{\sum_{j=1}^{N} (u_k(j))^{\beta}}.$$

This results in Algorithm 3 disregarding the case $d(r_k^{(n)}, f(j)) = 0$ for any k, j, n. However, Teboulle [156] does not show the equivalence of the objective functionals in (ACP) and (2.14) directly.

One advantage of fuzzy c-means compared to c-means (Algorithm 2) is that special treatment of empty clusters to avoid division of 0 by 0, i.e., $\sum_{j=1}^{N} u_k(j) = 0$ for any k, is superfluous. More precisely, Hathaway et al. [77, Theorem 3] prove that $\sum_{j=1}^{N} u_k(j) \geq \varepsilon$ for all k where $\varepsilon \in \mathbb{R}_{>0}$.

Another advantage of soft clustering algorithms is that the degree of membership provides more information compared to binary assignment. By thresholding it is always possible to obtain binary cluster assignment from a soft cluster membership vector while the other direction obviously does not work.

Regarding the convergence of Algorithm 3, Bezdek et al. [18] show that fuzzy c-means converges to a partial minimizer or contains a subsequence converging to a partial minimizer. Thereby, the algorithm is represented by a point-to-set-map to leave the iterate unspecified in case there exists $\mathcal{I} \subseteq \{1, \ldots, c\}$ with $\#\mathcal{I} > 1$ and $d(r_{\bar{l}}, f(j)) = 0$ for all $\bar{l} \in \mathcal{I}$. In fact, the work of Bezdek et al. [18] motivated the convergence analysis of our segmentation algorithm in the

following Section 2.3.

Algorithm 3: Fuzzy c-Means

Input: data set f, number of clusters c
Output: cluster membership vector u and codebook r
initialization of $r^{(0)} \in \mathbb{R}^{cm}$
repeat for $n = 0, 1, \ldots$

```
// Assign points to centers
```

for $k := 1, \ldots, c$ *and* $j := 1, \ldots, N$ **do**
  if $d_j(r^{(n)}) > 0$ **then**
    $u_k^{(n)}(j) := \dfrac{d(r_k^{(n)}, f(j))^{\frac{1}{1-\beta}}}{\sum_{\bar{k}=1}^{c} d(r_{\bar{k}}^{(n)}, f(j))^{\frac{1}{1-\beta}}}$
  else
    if $d(r_k^{(n)}, f(j)) > 0$ **then**
      $u_k^{(n)}(j) := 0$
    else
      Choose $\kappa_k \geq 0$ with $\sum_{k \in \mathcal{I}} \kappa_k = 1$ where $\mathcal{I} := \{k : d(r_k^{(n)}, f(j)) > 0\}$ and set
      $u_k^{(n)}(j) := \kappa_k$
    end
  end
end

```
// Update centers
```

for $k := 1, \ldots, c$ **do**
  $r_k^{(n+1)} = \dfrac{\sum_{j=1}^{N} (u_k^{(n)}(j))^{\beta} f(j)}{\sum_{j=1}^{N} (u_k^{(n)}(j))^{\beta}}$
end

until $r^{(n+1)} = r^{(n)}$

2.3 The Regularized c-Centers Algorithm

After the introduction of hard and soft clustering in the previous sections, we focus in this section on unsupervised segmentation of images. Since noise is an ubiquitous problem for image data, the application of the previously mentioned unregularized clustering algorithms to images can lead to poor results as shown in Figure 2.4 where we want to separate bird and background. Without regularization we obtain a lot of undesired parts of the image such as the wood or the corners as foreground (black). Moreover, small isolated parts within the bird are assigned to the background (white). See also Figure 1.1 in the introduction for another example.

In this section, we propose a general biconvex segmentation model. Thereby, we extend a

(a) Input (b) Without regularization (c) With TV-regularization

Figure 2.4.: Segmentation of a bird, input taken from [107].

data approximating clustering objective by addition of a regularizer to enforce smoothness of the result. We restrict our considerations to continuous regularizers on $\mathbb{R}^N$.

For the data term we use a dissimilarity function $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ as defined in Section 2.1 to formulate our optimization problem. We consider the *regularized segmentation problem*

$$\text{(RSP)} \qquad \operatorname*{argmin}_{u,r} E_{\mathrm{RS}}(u,r) \quad \text{subject to} \quad u \in C$$
$$\text{where } E_{\mathrm{RS}}(u,r) = \sum_{j=1}^{N} \sum_{k=1}^{c} (u_k(j))^\beta d(r_k, f(j)) + \lambda \Psi(u)$$

with the parameters $\lambda \geq 0$, $\beta \geq 1$, an image $f : \{1, \dots, N\} \to \mathbb{R}^m$, the codebook of segment centers $r = (r_k)_{k=1}^c \in \mathbb{R}^{cm}$, $r_k \in \mathbb{R}^m$, a continuous function $\Psi : \mathbb{R}^{cN} \to \mathbb{R}$, and a compact set $C \subset \mathbb{R}^{cN}$. Further, we require that

$$\begin{aligned} &\hat{r} := \operatorname*{argmin}_{r \in \mathbb{R}^{cm}} E_{\mathrm{RS}}(u,r) \text{ is unique for all} \\ &u \in C \setminus \{v \in C : \text{ there exists } k \in \{1, \dots, c\} \text{ with } \textstyle\sum_{j=1}^{N} v_k(j) = 0\}. \end{aligned} \tag{2.15}$$

Remark 2.9. *The theoretical results of this section hold for arbitrary compact sets C. However, in the previous sections we have already seen two of reasonable choices of C:*

1. *the set of standard basis vectors in $\mathbb{R}^c$ to the power N denoted by $\{e_1, \dots, e_c\}^N$ and*
2. *the unit simplex $S_c = \operatorname{conv}\{e_1, \dots, e_c\}$ in $\mathbb{R}^c$ to the power N denoted by S_c^N.*

In fact, these are the only two sets of segment membership vectors that occur for particular implementations of (RSP) in our work.

Further, let us point out that a restriction of r to a closed subset of $\mathbb{R}^{cm}$ does not violate the following convergence theory. However, we stick to $r \in \mathbb{R}^{cm}$ due to better readability.

To minimize $E_{\mathrm{RS}}(u,r)$ we use *regularized c-centers* (RcC) stated in Algorithm 4 where the *level set* of a function $g : \mathbb{R}^{cN} \to \mathbb{R}$ is defined by $\operatorname{lev}_b g := \{u \in \mathbb{R}^{cN} : g(u) \leq b\}$ and

$$\mathcal{H} : C \times \mathbb{R}^{cm} \rightrightarrows C, \tag{2.16}$$
$$\mathcal{H}(u,r) := \begin{cases} \{v \in C : E_{\mathrm{RS}}(v,r) + \delta \leq E_{\mathrm{RS}}(u,r)\} & \text{if } u \notin \operatorname{lev}_{b(r)} E_{\mathrm{RS}}(\cdot, r), \\ \operatorname{argmin}_{v \in C} E_{\mathrm{RS}}(v,r) & \text{otherwise} \end{cases}$$

with $b(r) := \min_{u \in C} E_{\mathrm{RS}}(u, r) + \delta$. Figure 2.5 explains the two different cases in the definition of $\mathcal{H}$. $\mathcal{H}(u, r)$ is either the set of all $v \in C$ that reduce the value of $E_{\mathrm{RS}}(u, r)$ by at least δ or – if a decrease of δ is not possible – the set of minimizers $\operatorname{argmin}_v E_{\mathrm{RS}}(v, r)$.

Note that $b(r)$ does not have to be computed explicitly in our algorithm, as we will see later in Section 2.4 for a particular implementation of RcC.

Algorithm 4: Regularized c-Centers (RcC)

Input: image f, number of clusters c, parameters $\lambda \geq 0$, $\varepsilon_{\mathrm{emp}} > 0, \delta > 0$, $\beta \geq 1$
Output: cluster membership matrix u and codebook r
initialization of $r^{(0)} \in \mathbb{R}^{cm}$ and $u^{(-1)} \in S_c^N$ with $\sum_{j=1}^N u_k^{(-1)}(j) \geq \varepsilon_{\mathrm{emp}}$ for $k = 1, \ldots, c$
for $n = 0, 1, 2, \ldots$ **do**

`// Assign points to centers`
$u^{(n)} \in \mathcal{H}(u^{(n-1)}, r^{(n)})$ where $\mathcal{H}$ is defined in (2.16)

`// Check for empty clusters`
for $k := 1, \ldots, c$ **do**
  if $\sum_{j=1}^N u_k^{(n)}(j) < \varepsilon_{\mathrm{emp}}$ **then**
    Reduce c by 1 and restart with $(r_l^{(n)})_{l \neq k}$ as initial solution
  end
end

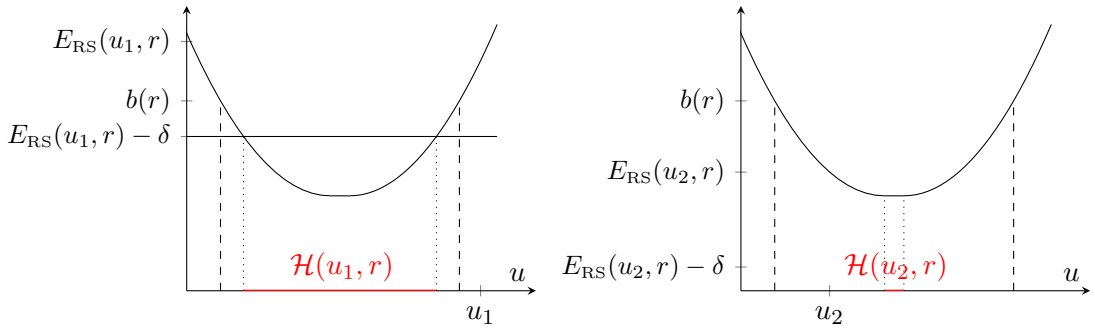
`// Update centers`
$$r^{(n+1)} \in \operatorname{argmin}_{r \in \mathbb{R}^{cm}} \left\{ \sum_{j=1}^{N} \sum_{k=1}^{c} (u_k^{(n)}(j))^{\beta} d(r_k, f(j)) \right\}$$

end

In general, the existence of global minimizers for the continuous function $E_{\mathrm{RS}}(u, \cdot)$ is guaranteed since C is compact. We do not explicitly demand biconvexity of E_{RS} for the theoretical results of this section. Nevertheless, we only consider particular implementations of RcC where we are able to compute the global optima for fixed r and for fixed u due to convexity.

The following subsections deal with the convergence of the sequence $(u^{(n)}, r^{(n)})_{n \in \mathbb{N}}$ generated by Algorithm 4. Our approach is motivated by the convergence proof for fuzzy c-means [77]. Likewise, we use Zangwill's theory [172] to state a convergence theorem for RcC. Section 2.3.1 reviews Zangwill's convergence theory for monotonic algorithms.

In Section 2.3.2 Zangwill's theory is applied to RcC. More precisely, we show that the objective $(E_{\mathrm{RS}}(u^{(n)}, r^{(n)}))_{n \in \mathbb{N}}$ decreases strictly as long as $(u^{(n)}, r^{(n)})$ is not a partial minimizer of E_{RS}. Additionally, every sequence generated by RcC converges to a partial minimizer of E_{RS} or contains a subsequence converging to a partial minimizer of E_{RS} and every convergent subsequence has a partial minimizer as limit point. In case C is finite we can be sure that after finitely many iterations $(u^{(n)}, r^{(n)})$ is a partial minimizer as shown later in Corollary 2.18. Popular examples for methods using the finite set $C = \{e_1, \ldots, e_c\}^N$ are c-means (see Section 2.1) or the level-set segmentation method of Chan and Vese (see [42] and Section 2.4.3) with global optimization for fixed codebook r [115, 41].



(a) $u_1 \notin \mathrm{lev}_{b(r)} E_{\mathrm{RS}}(\cdot, r)$ (b) $u_2 \in \mathrm{lev}_{b(r)} E_{\mathrm{RS}}(\cdot, r)$

Figure 2.5.: A sketch of the two different cases in the definition of $\mathcal{H}$ for fixed r. The resulting sets are colored in red and delimited by dotted (not dashed) lines. a) Decrease of E_{RS} by δ is possible since $E_{\mathrm{RS}}(u_1, r) - \min_{u \in C} E_{\mathrm{RS}}(u, r) > \delta$. b) Contrary to (a), we have $E_{\mathrm{RS}}(u_2, r) - \min_{u \in C} E_{\mathrm{RS}}(u, r) < \delta$ and cannot achieve a decrease of δ. Hence, we set $\mathcal{H}(u_2, r) = \operatorname{argmin}_{u \in C} E_{\mathrm{RS}}(u, r)$.

For biconvex E_{RS} and a full optimization in the u-step, i.e., $u^{(n+1)} \in \operatorname{argmin}_{v \in C} E_{\mathrm{RS}}(v, r^{(n)})$ for all $u^{(n)}$, a convergence result is cited in the following remark.

Remark 2.10. *Gorski et al. [69] consider general biconvex problems and investigate the convergence of a minimization algorithm called* alternate convex search *(ACS) described in the following. Let*

$$\operatorname*{argmin}_{x,y} E(x, y) \text{ subject to } (x, y) \in B \tag{2.17}$$

be a biconvex optimization problem. ACS consists of alternatingly minimizing (2.17) *for fixed y and x. The proof of convergence given in [69] was motivated by [49, 108] and the work of Zangwill [172]. Note that ACS really needs to obtain a global minimizer of both convex subproblems in every iteration. In contrast, in RcC we require only a certain amount of decrease of the objective after updating the segment membership vector leading to a faster algorithm.*

2.3.1 Zangwill's Convergence Theory for Monotonic Algorithms

To discuss the convergence of RcC we review Zangwill's Theory [172] in this section. We are given

- a so called *solution set* X_{sol} contained in a metric space X,
- an iterative algorithm defined by a *point-to-set-map* $\mathcal{T} : X \rightrightarrows X$, i.e., $\mathcal{T} : X \to 2^X$, and
- a sequence $(x^{(i)})_{i \in \mathbb{N}}$ generated by $\mathcal{T}$, i.e.,

$$\begin{aligned} x^{(i+1)} &\in \mathcal{T}(x^{(i)}) && \text{if } \mathcal{T}(x^{(i)}) \neq \emptyset, \\ x^{(i+1)} &= x^{(i)} && \text{if } \mathcal{T}(x^{(i)}) = \emptyset. \end{aligned}$$

We would like to guarantee that the limit point of every convergent subsequence of $(x^{(i)})_{i\in\mathbb{N}}$ is contained in the solution set X_{sol}, i.e., the set of all accumulation points of $(x^{(i)})_{i\in\mathbb{N}}$ denoted by $A_{(x^{(i)})_{i\in\mathbb{N}}}$ is contained in X_{sol}. Therefore, we need two ingredients:

1. a so called *descending merit function* $M \in \mathcal{C}(X, \mathbb{R})$ to measure the progress of the algorithm for given $\mathcal{T}$ and X_{sol} and
2. the closeness of $\mathcal{T}$.

We call a merit function $M \in \mathcal{C}(X, \mathbb{R})$ *descending* if

i) for all $x \in X$ and all $\bar{x} \in \mathcal{T}(x)$ the inequality $M(\bar{x}) \leq M(x)$ holds true and

ii) for all $x \in X \backslash X_{\text{sol}}$ with $\mathcal{T}(x) \neq \emptyset$ and all $\bar{x} \in \mathcal{T}(x)$ the strict inequality $M(\bar{x}) < M(x)$ holds true.

The second ingredient, i.e., the closeness of $\mathcal{T}$, is defined as follows. Let X be a metric space, $x \in X$ and $\bigcup_{i\in\mathbb{N}}\{x^{(i)}\} \subset X$. A point-to-set-map $\mathcal{T} : X \rightrightarrows X$ is *closed at* x if

$$\left.\begin{array}{l} x^{(i)} \to x \\ y^{(i)} \in \mathcal{T}(x^{(i)}) \\ y^{(i)} \to y \end{array}\right\} \text{ imply } y \in \mathcal{T}(x).$$

A point-to-set-map is closed on $\bar{X} \subset X$ if it is closed at every $\bar{x} \in \bar{X}$.

We will see that for closed $\mathcal{T}$ the existence of a descending merit function can help to prove convergence. However, the following example shows that the existence of a descending merit function is not sufficient for $A_{(x^{(i)})_{i\in\mathbb{N}}} \subseteq X_{\text{sol}}$ if $\mathcal{T}$ is not closed.

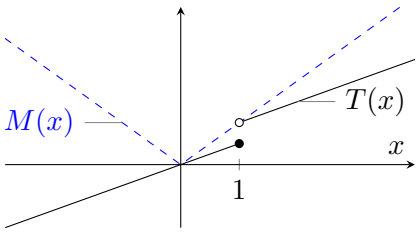


Figure 2.6.: Function T and the descending merit function M from Example 2.11.

Example 2.11. *Consider the function* $T : \mathbb{R} \to \mathbb{R}$ *as a special case of a point-to-set-map with*

$$T(x) := \begin{cases} \frac{1}{2}x & \text{if } x \leq 1, \\ \frac{1}{2}x + \frac{1}{2} & \text{if } x > 1 \end{cases}$$

shown in Figure 2.6 and let $X_{\text{sol}} = \{0\}$. *Then a descending merit function is defined by* $M(x) := |x|$. *For* $x^{(0)} > 1$ *we have with* $(x^{(i)})_{i\in\mathbb{N}}$ *generated by* T *that* $x^{(i)} \to 1 \notin X_{\text{sol}}$. *The problem is that* T *is not closed at* $x = 1$ *and hence* $1 \notin T(1)$. *If we consider the closed point-to-set-map*

$$\mathcal{T}(x) := \begin{cases} \frac{1}{2}x & \text{if } x < 1, \\ \{\frac{1}{2}, 1\} & \text{if } x = 1, \\ \frac{1}{2}x + \frac{1}{2} & \text{if } x > 1 \end{cases}$$

instead, the merit function $M(x) = |x|$ *is not descending.*

Zangwill's Theorem

For the proof of Zangwill's Theorem, we use the subsequent lemma.

Lemma 2.12. *Let $M \in \mathcal{C}(X)$ and $(x^{(i)})_{i\in\mathbb{N}} \subset X$. Further,*

i) *let $M(x^{(i+1)}) \leq M(x^{(i)})$ for all $i \in \mathbb{N}$ and*

ii) *let there exist a subsequence $(x^{(i_j)})_{j\in\mathbb{N}}$ of $(x^{(i)})_{i\in\mathbb{N}}$ that convergences to $\hat{x} \in X$.*

Then

$$\lim_{i\to\infty} M(x^{(i)}) = \lim_{j\to\infty} M(x^{(i_j)}) = M(\hat{x}).$$

Proof. Continuity of M implies

$$\lim_{j\to\infty} M(x^{(i_j)}) = M(\hat{x}). \tag{2.18}$$

Assume that there exists an i with $M(x^{(i)}) < M(\hat{x})$. Then, due to i) we have

$$M(\hat{x}) > M(x^{(i)}) \geq \lim_{j\to\infty} M(x^{(i_j)}) = M(\hat{x})$$

which yields $M(x^{(i)}) \geq M(\hat{x})$ by contradiction. Thus, $M(x^{(i)})$ is bounded from below by $M(\hat{x})$ and any monotonically decreasing sequence which is bounded from below converges. By (2.18) we have that $\lim_{i\to\infty} M(x^{(i)}) = M(\hat{x})$. □

Now we can prove Zangwill's Theorem A, see [172, p. 91].

Theorem 2.13 (Zangwill's Convergence Theorem)**.** *Let the metric space X, the point-to-set-map $\mathcal{T} : X \rightrightarrows X$ and a solution set $X_{\text{sol}} \subset X$ be given. Assume that the following holds true:*

i) *There exists a descending merit function $M \in \mathcal{C}(X, \mathbb{R})$ and*

ii) *$\mathcal{T}$ is closed at every $x \in X \backslash X_{\text{sol}}$.*

Then for any sequence $(x^{(i)})_{i\in\mathbb{N}}$ generated by $\mathcal{T}$ and contained in a compact subset $K \subseteq X$, the set of accumulation points $A_{(x^{(i)})_{i\in\mathbb{N}}}$ is contained in X_{sol}, i.e., every convergent subsequence of $(x^{(i)})_{i\in\mathbb{N}}$ converges to a solution $\hat{x} \in X_{\text{sol}}$.

Proof. Let $x^{(i_j)} \to \hat{x}$ as $j \to \infty$ such that $\hat{x} \in A_{(x^{(i)})_{i\in\mathbb{N}}}$. Note that $A_{(x^{(i)})_{i\in\mathbb{N}}} \neq \emptyset$, because $(x^{(i)})_{i\in\mathbb{N}}$ is contained in a compact subset of X. We show that $\hat{x} \in X_{\text{sol}}$.
Case 1: There exists $n \in \mathbb{N}$ with $\mathcal{T}(x^{(n)}) = \emptyset$. Then $\hat{x} = x^{(n)} = x^{(n+1)} = x^{(n+2)} = \cdots \in X_{\text{sol}}$ by assumption on M.
Case 2: We have $\mathcal{T}(x^{(i)}) \neq \emptyset$ for all $i \in \mathbb{N}$. Then $M(x^{(i+1)}) \leq M(x^{(i)})$ and further

$$\lim_{i\to\infty} M(x^{(i)}) = \lim_{j\to\infty} M(x^{(i_j)}) = M(\hat{x})$$

by Lemma 2.12. Hence, for arbitrary $\hat{y} \in A_{(x^{(i)})_{i\in\mathbb{N}}}$ we have

$$M(\hat{x}) = \lim_{i\to\infty} M(x^{(i)}) = M(\hat{y}). \tag{2.19}$$

Assume that $\hat{x} \notin X_{\text{sol}}$. Consider $x^{(i_j+1)} \in \mathcal{T}(x^{(i_j)})$. The sequence $(x^{(i_j+1)})_{j\in\mathbb{N}}$ is contained in the compact set K. This implies the existence of a subsequence $(x^{(i_{k_l})})_{l\in\mathbb{N}}$ of $(x^{(i_j+1)})_{j\in\mathbb{N}}$ with $x^{(i_{k_l})} \to \hat{y} \in K$ as $l \to \infty$. Furthermore, with $x^{(i_{k_l})} \in \mathcal{T}(x^{(i_{j_l})})$, $x^{(i_{j_l})} \to \hat{x}$ as $l \to \infty$ and the closeness of $\mathcal{T}$ on $X \setminus X_{\text{sol}}$ we have $\hat{y} \in \mathcal{T}(\hat{x})$. Since $\hat{x} \notin X_{\text{sol}}$, we see by definition of M that $M(\hat{y}) < M(\hat{x})$ in contradiction to (2.19). Thus, $\hat{x} \in X_{\text{sol}}$ and we are done. □

Closed Compositions of Point-to-Set-Maps

The rest of this section deals with the closeness of the composition of point to set maps. In the following, a definition of the *composition* of two point-to-set-maps is given. Let X, Y, Z be metric spaces and $\mathcal{T}_1 : X \rightrightarrows Y$, $\mathcal{T}_2 : Y \rightrightarrows Z$. Then

$$\mathcal{T}_2 \circ \mathcal{T}_1 : X \rightrightarrows Z,\ (\mathcal{T}_2 \circ \mathcal{T}_1)(x) := \bigcup_{y\in\mathcal{T}_1(x)} \mathcal{T}_2(y).$$

We will see that the results of this section can help us to prove the closeness of the point-to-set-map $\mathcal{T}_{\text{RcC}}$ in (2.22), which defines the RcC algorithm. For the next lemma see [172, p. 95].

Lemma 2.14. *Let $\mathcal{T}_1 : X \rightrightarrows Y$ be closed at $x \in X$ and $\mathcal{T}_2 : Y \rightrightarrows Z$ be closed at $y \in \mathcal{T}_1(x)$. Furthermore, assume that $x^{(i)} \to x$ and $y^{(i)} \in \mathcal{T}_1(x^{(i)})$ imply that there exists a subsequence $(y^{(i_j)})_{j\in\mathbb{N}}$ of $(y^{(i)})_{i\in\mathbb{N}}$ with $y^{(i_j)} \to y \in Y$ as $j \to \infty$. Then $\mathcal{T} := \mathcal{T}_2 \circ \mathcal{T}_1$ is closed at x.*

Proof. Let

$$x^{(i)} \to x,\ z^{(i)} \in (\mathcal{T}_2 \circ \mathcal{T}_1)(x^{(i)}) \text{ and } z^{(i)} \to z. \tag{2.20}$$

We show that $z \in \mathcal{T}(x)$. We choose $y^{(i)} \in \mathcal{T}_1(x^{(i)})$ such that $z^{(i)} \in \mathcal{T}_2(y^{(i)})$ for all $i \in \mathbb{N}$. By (2.20) and the last assumption of the lemma we have that there exists a subsequence $(y^{(i_j)})_{j\in\mathbb{N}}$ of $(y^{(i)})_{i\in\mathbb{N}}$ such that

$$x^{(i_j)} \underset{j\to\infty}{\longrightarrow} x,\ y^{(i_j)} \in \mathcal{T}_1(x^{(i_j)}) \text{ and } y^{(i_j)} \underset{j\to\infty}{\longrightarrow} y.$$

By the closeness of $\mathcal{T}_1$ at x this implies $y \in \mathcal{T}_1(x)$. Therefore, we have by

$$y^{(i_j)} \underset{j\to\infty}{\longrightarrow} y,\ z^{(i_j)} \in \mathcal{T}_2(y^{(i_j)}) \text{ and } z^{(i_j)} \underset{j\to\infty}{\longrightarrow} z$$

and the closeness of $\mathcal{T}_2$ at y that $z \in \mathcal{T}_2(y)$ and hence $z \in (\mathcal{T}_2 \circ \mathcal{T}_1)(x)$.

□

The first corollary of Lemma 2.14 deals with the composition of closed point-to-set-maps, see [172, p. 96].

Corollary 2.15. *Let $\mathcal{T}_1 : X \rightrightarrows Y$ be closed at $x \in X$ and $\mathcal{T}_2 : Y \rightrightarrows Z$ be closed at $y \in \mathcal{T}_1(x)$ and assume Y is compact. Then $\mathcal{T} : X \rightrightarrows Z$, $\mathcal{T} := \mathcal{T}_2 \circ \mathcal{T}_1$ is closed at x.*

Proof. Assume $x^{(i)} \to x$ and $y^{(i)} \in \mathcal{T}_1(x^{(i)}) \subseteq Y$. Since Y is a compact metric space there exists a convergent subsequence $(y^{(i_j)})_{j\in\mathbb{N}}$ of $(y^{(i)})_{i\in\mathbb{N}}$ with limit $y \in Y$. Hence, the closeness of $\mathcal{T}$ is provided by Lemma 2.14. □

The second corollary considers the composition of a closed points-to-set-map with a continuous function, see [172, p. 96].

Corollary 2.16. *Let the function $T_1 : X \to Y$ be continuous at $x \in X$ and the point-to-set-map $\mathcal{T}_2 : Y \rightrightarrows Z$ be closed at $y = T_1(x)$. Then $\mathcal{T} : X \rightrightarrows Z,\ \mathcal{T} = \mathcal{T}_2 \circ T_1$ is closed at $x \in X$.*

Proof. Assume $x^{(i)} \to x$ and $y_i = T_1(x^{(i)})$. Then by the continuity of T_1 we have $y^{(i)} = T_1(x^{(i)}) \to T_1(x) = y$. Application of Lemma 2.14 concludes the proof. □

2.3.2 Convergence of Regularized c-Centers

In this section we show that the RcC algorithm fulfills all requirements of Theorem 2.13.

Firstly, we define the solution set by

$$X_{\text{sol}} := \Big\{(\hat{u}, \hat{r}) \in C \times \mathbb{R}^{cm} : \quad E_{\text{RS}}(\hat{u}, \hat{r}) \le E_{\text{RS}}(u, \hat{r})\ \forall u \in C \text{ and } E_{\text{RS}}(\hat{u}, \hat{r}) \le E_{\text{RS}}(\hat{u}, r)\ \forall r \in \mathbb{R}^{cm}\Big\}. \tag{2.21}$$

In fact, X_{sol} is the set of partial minimizers defined in (2.4).

Next, we define the operator $\mathcal{T}_{\text{RcC}}$ of the RcC algorithm. Let

$$\mathcal{G} : C \rightrightarrows \mathbb{R}^{cm},\ \mathcal{G}(u) := \operatorname*{argmin}_{r \in \mathbb{R}^{cm}} \sum_{j=1}^{N} \sum_{k=1}^{c} (u_k(j))^{\beta} d(r_k, f(j))$$

where $\beta \ge 1$. We use the set-valued notation since $\mathcal{G}(u)$ is only single-valued due to (2.15), if $\sum_{k=1}^{c} u_k(j) \neq 0$ for all $k \in \{1, \ldots, c\}$. Moreover, we have

$$\mathcal{H} : C \times \mathbb{R}^{cm} \rightrightarrows C,\ \mathcal{H}(u, r) = \begin{cases} \{v : E_{\text{RS}}(v, r) + \delta \le E_{\text{RS}}(u, r)\} & \text{if } u \notin \operatorname{lev}_{b(r)} E_{\text{RS}}(\cdot, r) \\ \operatorname{argmin}_{v \in C} E_{\text{RS}}(v, r) & \text{otherwise} \end{cases}$$

where $b(r) = \min_{u \in C} E_{\text{RS}}(u, r) + \delta$ as defined in (2.16). The algorithmic map is given by

$$\mathcal{T}_{\text{RcC}} : C \times \mathbb{R}^{cm} \rightrightarrows C \times \mathbb{R}^{cm}, \mathcal{T}_{\text{RcC}} := \mathcal{T}_2 \circ \mathcal{T}_1 \tag{2.22}$$

with

$$\mathcal{T}_1 : C \times \mathbb{R}^{cm} \rightrightarrows C \times \mathbb{R}^{cm},\ \mathcal{T}_1(u, r) := (u, \mathcal{G}(u))$$

and

$$\mathcal{T}_2 : C \times \mathbb{R}^{cm} \rightrightarrows C \times \mathbb{R}^{cm},\ \mathcal{T}_2(u, r) := (\mathcal{H}(u, r), r).$$

As merit function we choose the objective function E_{RS}. Indeed, E_{RS} is a descending merit function for $\mathcal{T}_{\text{RcC}}$ and X_{sol} as we show in the subsequent theorem.

Theorem 2.17. *The objective E_{RS} is a descending merit function for $\mathcal{T}_{\text{RcC}}$ and X_{sol}, i.e., $E_{\text{RS}}(u^{(n+1)}, r^{(n+1)}) \le E_{\text{RS}}(u^{(n)}, r^{(n)})$ for all $(u^{(n+1)}, r^{(n+1)}) \in \mathcal{T}_{\text{RcC}}(u^{(n)}, r^{(n)})$ with strict inequality if $(u^{(n)}, r^{(n)}) \notin X_{\text{sol}}$.*

Proof. Let $(u^{(n+1)}, r^{(n+1)}) \in \mathcal{T}_{\mathrm{RcC}}(u^{(n)}, r^{(n)})$. The definitions of $\mathcal{G}$ and $\mathcal{H}$ imply that

$$E_{\mathrm{RS}}(u^{(n+1)}, r^{(n+1)}) \leq E_{\mathrm{RS}}(u^{(n)}, r^{(n+1)}) \leq E_{\mathrm{RS}}(u^{(n)}, r^{(n)}).$$

It remains to prove that $E_{\mathrm{RS}}(u^{(n+1)}, r^{(n+1)}) < E_{\mathrm{RS}}(u^{(n)}, r^{(n)})$ for $(u^{(n)}, r^{(n)}) \notin X_{\mathrm{sol}}$. Let $(u^{(n)}, r^{(n)}) \notin X_{\mathrm{sol}}$. Then, the second inequality in the definition of X_{sol} in (2.21) is violated or fulfilled. Of course, if the second inequality holds true, the first inequality of (2.21) is violated for $(u^{(n)}, r^{(n)}) \notin X_{\mathrm{sol}}$. Accordingly we distinguish two cases.

Case 1: If the second inequality in (2.21) is violated by $(u^{(n)}, r^{(n)}) \notin X_{\mathrm{sol}}$ we have that there exists $\tilde{r} \in \mathbb{R}^{cm}$ with $E_{\mathrm{RS}}(u^{(n)}, r^{(n)}) > E_{\mathrm{RS}}(u^{(n)}, \tilde{r})$. The definition of $\mathcal{T}_{\mathrm{RcC}}$ yields

$$r^{(n+1)} \in \mathcal{G}(u^{(n)}) = \operatorname*{argmin}_{r \in \mathbb{R}^{cm}} E_{\mathrm{RS}}(u^{(n)}, r).$$

Hence,

$$E_{\mathrm{RS}}(u^{(n)}, r^{(n)}) > E_{\mathrm{RS}}(u^{(n)}, \tilde{r}) \geq E_{\mathrm{RS}}(u^{(n)}, r^{(n+1)}) \geq E_{\mathrm{RS}}(u^{(n+1)}, r^{(n+1)}).$$

Case 2: Let the second inequality in (2.21) hold true for $(u^{(n)}, r^{(n)}) \notin X_{\mathrm{sol}}$, i.e., $E_{\mathrm{RS}}(u^{(n)}, r^{(n)}) \leq E_{\mathrm{RS}}(u^{(n)}, r)\ \forall r \in \mathbb{R}^{cm}$. Thus,

$$r^{(n)} \in \operatorname*{argmin}_{r \in \mathbb{R}^{cm}} E_{\mathrm{RS}}(u^{(n)}, r) \tag{2.23}$$

and $r^{(n)} = r^{(n+1)}$ due to uniqueness[1] of (2.23). If $u^{(n)} \notin \mathrm{lev}_{b(r^{(n+1)})} E_{\mathrm{RS}}(\cdot, r^{(n+1)})$ the definitions of $\mathcal{G}$ and $\mathcal{H}$ conclude the second case since

$$E_{\mathrm{RS}}(u^{(n)}, r^{(n)}) \geq E_{\mathrm{RS}}(u^{(n)}, r^{(n+1)}) \geq E_{\mathrm{RS}}(u^{(n+1)}, r^{(n+1)}) + \delta > E_{\mathrm{RS}}(u^{(n+1)}, r^{(n+1)}).$$

If $u^{(n)} \in \mathrm{lev}_{b(r^{(n+1)})} E_{\mathrm{RS}}(\cdot, r^{(n+1)})$ we have due to violation of the first inequality in (2.21) that there exists an $\tilde{u} \in C$ with $E_{\mathrm{RS}}(u^{(n)}, r^{(n)}) > E_{\mathrm{RS}}(\tilde{u}, r^{(n)})$. Thus, we obtain

$$E_{\mathrm{RS}}(u^{(n)}, r^{(n)}) > E_{\mathrm{RS}}(\tilde{u}, r^{(n)}) = E_{\mathrm{RS}}(\tilde{u}, r^{(n+1)}) \geq E_{\mathrm{RS}}(u^{(n+1)}, r^{(n+1)}).$$

□

The next corollary handles the special case where C is finite. It states that after finitely many iterations the iterate of RcC is a partial minimizer. We proceed according to the convergence proof of c-means by Selim and Ismail [137].

Corollary 2.18. *Let $(u^{(n)}, r^{(n)})$ be a sequence generated by $\mathcal{T}_{\mathrm{RcC}}$ and let C be finite. Then, there exists $n_0 \in \mathbb{N}$ such that $(u^{(n_0)}, r^{(n_0)})$ is a partial minimizer.*

Proof. Let $(u^{(n_1)}, r^{(n_1)})$ and $(u^{(n_2)}, r^{(n_2)})$ not be partial minimizers and let us assume that $u^{(n_1)} = u^{(n_2)}$ for some $n_1 < n_2$. This implies

$$E_{\mathrm{RS}}(u^{(n_1)}, r^{(n_1)}) = E_{\mathrm{RS}}(u^{(n_2)}, r^{(n_2)}). \tag{2.24}$$

Equation (2.24) is a contradiction, since $(E_{\mathrm{RS}}(u^{(n)}, r^{(n)}))_{n \in \mathbb{N}}$ is strictly decreasing by Theorem 2.17. Hence, we achieve every possible value of u at most once. By assumption that C is finite there are only finitely many different possible values of u. □

[1] In fact, this is the only place in this section where we use the uniqueness of (2.23).

In the next step we show that the elements of every sequence generated by $\mathcal{T}_{\mathrm{RcC}}$ are contained in a compact set. This ensures the existence of a convergent subsequence. Furthermore, this property is necessary

- in the proof of the closeness of $\mathcal{H}$ in Lemma 2.23 and
- for the application of Corollary 2.15 in the proof of the closeness of $\mathcal{T}_{\mathrm{RcC}}$ in Theorem 2.24.

Lemma 2.19. *Let $y_j \in \mathbb{R}^m$ and $w_j > 0$ be fixed for all $j \in \{1, \dots, N\}$ and some $N \in \mathbb{N}$. Then, the minimizer $\hat{x}$ of $\phi(x) := \sum_{j=1}^{N} w_j d(x, y_j)$ is contained in a compact set $K_{y_1,\dots,y_N}$ which only depends on the fixed vectors y_j.*

Proof. Let the closed ball of radius ρ around y_j with respect to d be denoted by

$$B_d(y_j, \rho) := \{x \in \mathbb{R}^m : d(x, y_j) \leq \rho\}$$

and let

$$\rho := \max_{j_1, j_2 \in \{1,\dots,N\}} d(y_{j_1}, y_{j_2}).$$

Note that for fixed y, ρ the set B_d is closed and bounded since d is lsc and coercive, respectively. Assume that the minimizer of ϕ denoted by

$$\hat{x} \in \operatorname*{argmin}_{x \in \mathbb{R}^m} \phi(x) = \sum_{j=1}^{N} w_j d(x, y_j) \tag{2.25}$$

is not contained in $K_{y_1,\dots,y_N} := \bigcup_{j \in \{1,\dots,N\}} B_d(y_j, \rho)$, see Figure 2.7. It follows with $j, j_0 \in \{1, \dots, N\}$ that

$$w_j d(\hat{x}, y_j) > w_j d(y_{j_0}, y_j)$$

which is a contradiction to (2.25). Hence, $\hat{x}$ is contained in the compact set $K_{y_1,\dots,y_N}$. □

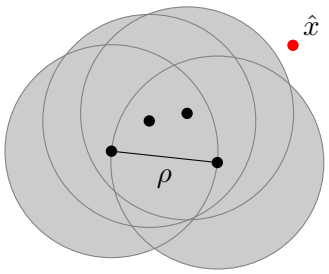


Figure 2.7.: Setting in the proof of Lemma 2.19. The set $\bigcup_{j \in \{1,\dots,N\}} B_d(y_j, \rho)$ is filled with gray background and every black point represents an $y_j \in \mathbb{R}^m$.

The following theorem guarantees that the sequence generated by $\mathcal{T}_{\mathrm{RcC}}$ is contained in a compact set.

Theorem 2.20. *The elements of a sequence $(u^{(n)}, r^{(n)})_{n \in \mathbb{N}}$ generated by $\mathcal{T}_{\mathrm{RcC}}$ are contained in a compact subset K of $C \times \mathbb{R}^{cm}$ which has the form $K = C \times R$.*

Proof. The set C is compact so that we only have to show that $\bigcup_{n\in\mathbb{N}}\{r^{(n)}\}$ is contained in a compact set. This follows directly from Lemma 2.19 and we are done with

$$R := \left(\bigcup_{j\in\{1,\dots,N\}} B_d(f(j),\rho) \right)^c \text{ and } K = C \times R$$

where $\rho = \max_{j_1,j_2\in\{1,\dots,N\}} d(f(j_1),f(j_2))$ and X^c denotes the c-fold Cartesian product of a set X. □

Remark 2.21. *For special choices of d one can prove Theorem 2.20 in another way and show that*

$$\bigcup_{n\in\mathbb{N}}\{r^{(n)}\} \subset \left(\operatorname{conv} \bigcup_{j=1}^{N}\{f(j)\} \right)^c. \tag{2.26}$$

In the case $d(x,y) = \|x-y\|_2^p$ the idea of the proof goes back to [166]. Let $r \in \bigcup_{n\in\mathbb{N}}\{r^{(n)}\}$. We show that r_k is contained in the compact set $\operatorname{conv}\bigcup_{j=1}^{N}\{f(j)\}$. Assume that, $r_k \notin \operatorname{conv}\bigcup_{j=1}^{N}\{f(j)\}$. Then we can separate r_k and $\operatorname{conv}\bigcup_{j=1}^{N}\{f(j)\}$ by a hyperplane h. Let $P(r_k)$ be the orthogonal projection of r_k onto h (see Figure 2.8). Then we have $\langle r_k - P(r_k), P(r_k) - f(j)\rangle > 0$ for all j. It follows with $w_j := (u_k(j))^\beta$ that

$$\begin{aligned}
&w_j^2 \|r_k - f(j)\|_2^2 - w_j^2 \|P(r_k) - f(j)\|_2^2 \\
&= w_j^2 \left(\langle r_k - f(j), r_k - f(j)\rangle - \langle P(r_k) - f(j), P(r_k) - f(j)\rangle\right) \\
&= w_j^2 \left(\langle r_k, r_k\rangle - 2\langle r_k, f(j)\rangle - \langle P(r_k), P(r_k)\rangle + 2\langle P(r_k), f(j)\rangle\right) \\
&= w_j^2 \left(2\langle r_k - P(r_k), P(r_k) - f(j)\rangle + \langle r_k - P(r_k), r_k - P(r_k)\rangle\right) > 0.
\end{aligned}$$

Consequently we get

$$w_j \|r_k - f(j)\|_2^p > w_j \|P(r_k) - f(j)\|_2^p,\ j = 1,\dots,N.$$

Hence, we have a contradiction to $r_k \in \operatorname{argmin}_{x\in\mathbb{R}^d} \sum_{j=1}^{N} (u_k^{(n)}(j))^\beta \|f(j) - x\|_2^p$.

For $d(x,y) = \|x-y\|_2^2$ and $C = S_c^N$, Bezdek [19] stated the following alternative and simple proof of (2.26). Let $\alpha_k(j) := u_k(j)^\beta / (\sum_{l=1}^{N} u_k(l)^\beta)$, $j \in \{1,\dots,N\}$ with $u \in C$. Then we have $\alpha_k(j) \in [0,1]$ for all k, j and

$$\sum_{j=1}^{N} \alpha_k(j) = \sum_{j=1}^{N} \frac{u_k(j)^\beta}{\sum_{l=1}^{N} u_k(l)^\beta} = 1,$$

such that r_k is a convex combination of the $f(j)$, $j = 1,\dots,N$.

It remains to show that $\mathcal{T}_{\mathrm{RcC}}$ is closed. Therefore, we use the following lemma.

Lemma 2.22. *Let X, Y, Z be metric spaces, let $F : X \times Y \to Z$ be lsc with $\operatorname{argmin}_{x\in X} F(x,y) \neq \emptyset$ for all $y \in Y$ and let $F(x,\cdot)$ be continuous for all $x \in X$. Then the point-to-set-map $\mathcal{F} : Y \rightrightarrows X$ defined by $\mathcal{F}(y) := \operatorname{argmin}_{x\in X} F(x,y)$ is closed.*

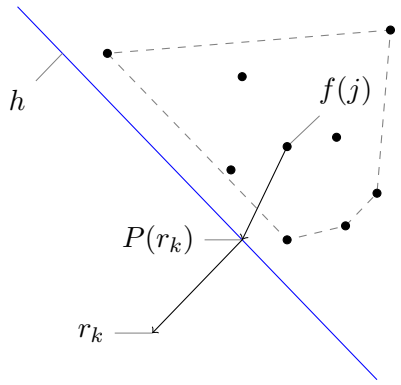


Figure 2.8.: Setting in the proof of (2.26) for $d(x,y) = \|x-y\|_2^p$.

Proof. Closeness of $\mathcal{F}$ means

$$\left.\begin{array}{l} y^{(i)} \to y^* \\ x^{(i)} \in \mathcal{F}(y^{(i)}) \\ x^{(i)} \to x^* \end{array}\right\} \text{ imply } x^* \in \mathcal{F}(x, y^*).$$

Thus, we have to show that

$$F(x^*, y^*) \leq F(x, y^*) \text{ for all } x \in X.$$

Using that F is lsc, as well as $x^{(i)} \in \mathcal{F}(y^{(i)}) = \operatorname{argmin}_{\tilde{x} \in X} F(\tilde{x}, y^{(i)})$, and the continuity of $F(x, \cdot)$ we obtain

$$F(x^*, y^*) \leq \lim_{i \to \infty} F(x^{(i)}, y^{(i)}) \leq \lim_{i \to \infty} F(x, y^{(i)}) = F(x, y^*)$$

for all $x \in X$. □

Using Lemma 2.22 we show the closeness of $\mathcal{G}$, and we prove the closeness of $\mathcal{H}$ in the subsequent theorem.

Lemma 2.23. *i) The mapping $\mathcal{G}$ is closed.*

ii) The mapping $\mathcal{H}$ is closed.

Proof. i) This follows directly from Lemma 2.22.

ii) First we show the continuity of $b_0 : r \mapsto \min_{u \in C} E_{\mathrm{RS}}(u, r)$ with respect to an arbitrary norm in $\mathbb{R}^{cm}$. To this end, we use without loss of generality the Euclidean norm $\|\cdot\|_2$. Since E_{RS} is uniformly continuous on the compact set $K = C \times R$ defined in Theorem 2.20, for every $\varepsilon > 0$ there exists a $\rho(\varepsilon) > 0$ such that for all $(u, r), (\tilde{u}, \tilde{r}) \in K$ with $\|(u, r) - (\tilde{u}, \tilde{r})\|_2 < \rho(\varepsilon)$ we have $|E_{\mathrm{RS}}(u, r) - E_{\mathrm{RS}}(\tilde{u}, \tilde{r})| < \varepsilon$. Particularly, this implies for all $\varepsilon > 0$ that there exists

a $\rho(\varepsilon) > 0$ such that for all $u \in C$ and $r, \tilde{r} \in R$ with $\|(u,r) - (u,\tilde{r})\|_2 = \|r - \tilde{r}\|_2 < \rho(\varepsilon)$ we have

$$E_{\mathrm{RS}}(u,r) - \varepsilon < E_{\mathrm{RS}}(u,\tilde{r}) < E_{\mathrm{RS}}(u,r) + \varepsilon.$$

For all $r \in R$ there exists $\tilde{u}(r) \in C$ with $b_0(r) = E_{\mathrm{RS}}(\tilde{u}(r), r)$. Hence,

$$b_0(r) + \varepsilon = E_{\mathrm{RS}}(\tilde{u}(r), r) + \varepsilon > E_{\mathrm{RS}}(\tilde{u}(r), \tilde{r}) \geq b_0(\tilde{r}).$$

On the other hand, for $\hat{u}(\tilde{r}) \in \operatorname{argmin}_{u \in C} E_{\mathrm{RS}}(u, \tilde{r})$ we obtain

$$b_0(r) - \varepsilon \leq E_{\mathrm{RS}}(\hat{u}(\tilde{r}), r) - \varepsilon < E_{\mathrm{RS}}(\hat{u}(\tilde{r}), \tilde{r}) = b_0(\tilde{r}).$$

Thus, b_0 and b are continuous functions.

Let us turn to the closeness of $\mathcal{H}$, i.e., we have to show that

$$\left.\begin{array}{l} (u^{(i)}, r^{(i)}) \to (u^*, r^*) \\ v^{(i)} \in \mathcal{H}(u^{(i)}, r^{(i)}) \\ v^{(i)} \to v^* \end{array}\right\} \text{ imply } v^* \in \mathcal{H}(u^*, r^*).$$

Now there exists a subsequence $(u^{(i_j)})_{j \in \mathbb{N}}$ such that either

$$u^{(i_j)} \notin \operatorname{lev}_{b(r^{(i_j)})} E_{\mathrm{RS}}(\cdot, r^{(i_j)}) = \{v : E_{\mathrm{RS}}(v, r^{(i_j)}) \leq b(r^{(i_j)}) = \min_{\tilde{u}} E_{\mathrm{RS}}(\tilde{u}, r^{(i_j)}) + \delta\} \quad (2.27)$$

for all $j \in \mathbb{N}$, or $u^{(i_j)} \in \operatorname{lev}_{b(r^{(i_j)})} E_{\mathrm{RS}}(\cdot, r^{(i_j)})$ for all $j \in \mathbb{N}$. If $u^{(i_j)} \notin \operatorname{lev}_{b(r^{(i_j)})} E_{\mathrm{RS}}(\cdot, r^{(i_j)})$ for all $j \in \mathbb{N}$ we have

$$E_{\mathrm{RS}}(v^*, r^*) + \delta = \lim_{j \to \infty} E_{\mathrm{RS}}(v^{(i_j)}, r^{(i_j)}) + \delta \leq \lim_{j \to \infty} E_{\mathrm{RS}}(u^{(i_j)}, r^{(i_j)}) = E_{\mathrm{RS}}(u^*, r^*)$$

Further, $u^* \notin \operatorname{int} \operatorname{lev}_{b(r^*)}$, since

$$\begin{gathered} E_{\mathrm{RS}}(u^{(i_j)}, r^{(i_j)}) > b(r^{(i_j)}) \text{ for all } j \in \mathbb{N} \text{ implies} \\ E_{\mathrm{RS}}(u^*, r^*) = \lim_{j \to \infty} E_{\mathrm{RS}}(u^{(i_j)}, r^{(i_j)}) \geq \lim_{j \to \infty} b(r^{(i_j)}) = b(r^*) \end{gathered}$$

due to (2.27) and continuity of E_{RS} and b. Thus, $v^* \in \mathcal{H}(u^*, r^*)$ since both cases in the definition of $\mathcal{H}(u^*, r^*)$ coincide for $u^* \in \partial \operatorname{lev}_{b(r^*)} E_{\mathrm{RS}}(\cdot, r^*)$. If $u^{(i_j)} \in \operatorname{lev}_{b(r^{(i_j)})} E_{\mathrm{RS}}(\cdot, r^{(i_j)})$ for all $j \in \mathbb{N}$ we have

$$E_{\mathrm{RS}}(v^*, r^*) = \lim_{j \to \infty} E_{\mathrm{RS}}(v^{(i_j)}, r^{(i_j)}) \leq \lim_{j \to \infty} E_{\mathrm{RS}}(u, r^{(i_j)}) = E_{\mathrm{RS}}(u, r^*)$$

for all $u \in C$ and $v^* \in \mathcal{H}$.

□

In the next theorem we can utilize Corollary 2.15 about the composition of closed maps to show the closeness of $\mathcal{T}_{\mathrm{RcC}}$.

Theorem 2.24. *The mapping $\mathcal{T}_{\mathrm{RcC}}$ is closed.*

Proof. We have that $\mathcal{T}_{\mathrm{RcC}}(u,r) = (\mathcal{H}(u,\mathcal{G}(u)),\mathcal{G}(u))$. The closeness of $\mathcal{T}_{\mathrm{RcC}}$ follows from Corollary 2.15 since $\bigcup_{n\in\mathbb{N}}\{r^{(n)}\}$ is contained in a compact set by Theorem 2.20 and $\mathcal{H}$ and $\mathcal{G}$ are closed by Lemma 2.23. □

We have shown that our algorithm fulfills all requirements of Zangwill's Theorem.

Theorem 2.25. *The set of accumulation points of the sequence $(u^{(n)},r^{(n)})_{n\in\mathbb{N}}$ generated by $\mathcal{T}_{\mathrm{RcC}}$ is contained in X_{sol}.*

In other words, Theorem 2.25 states that $(u^{(n)},r^{(n)})_{n\in\mathbb{N}}$ either converges to a partial minimizer or – in the worst case – contains a subsequence that converges to a partial minimizer. Moreover, every convergent subsequence of $(u^{(n)},r^{(n)})_{n\in\mathbb{N}}$ converges to a partial minimizer.

2.4 Total Variation Regularized c-Means

In this section we introduce the *discrete total variation* (TV) as regularizer for our segmentation model. The total variation for image processing has been proposed by Rudin, Osher, and Fatemi [129] in 1992. For the case that cluster centers r are known in advance, TV has been used for segmentation or labeling, e.g., by [171, 11, 97, 120]. TV-related multi-class segmentation models also optimizing the cluster centers have been proposed, e.g., in [158, 44, 10, 31] and by us in [79].

Recall that for functions g in the Sobolev space $W^{1,1}(\Omega)$ where $\Omega\subset\mathbb{R}^d$, the *total variation* defined in (1.2) can be written as

$$\mathcal{TV}(g) = \int_\Omega \|\mathcal{D}g(x)\|_2\,dx = \int_\Omega \sqrt{g_{x_1}(x)^2+\cdots+g_{x_d}(x)^2}\,dx$$

with weak partial derivatives g_{x_i}, $i=1,\dots,d$. Descriptively, this is the amount of change or variation measured by the modulus of the gradient summed over the whole domain, i.e., the total amount of variation. For more details see, e.g., [4, 65, 132].

In the following definition of the *discrete total variation*, we consider 2-dimensional images. For images of dimension $d\geq 3$, see Chapter A.1 in the appendix. To obtain a discrete representation of the total variation, we replace the weak partial derivatives g_{x_1} and g_{x_2} by finite differences. More precisely, for discrete, gray-valued, 2-dimensional images $F:\{1,\dots,n_1\}\times\{1,\dots,n_2\}\to\mathbb{R}$ column-wise reshaped to vectors $f:\{1,\dots,N\}\to\mathbb{R}$ with $N=n_1n_2$ we have the subsequent definitions. The finite differences of f in terms of the discrete gradient $\boldsymbol{D}$ are given by

$$\begin{pmatrix} f_{x_1}\\ f_{x_2}\end{pmatrix} := \underbrace{\begin{pmatrix} I_{n_2}\otimes D_{n_1}\\ D_{n_1}\otimes I_{n_2}\end{pmatrix}}_{=:\boldsymbol{D}} f$$

where $\otimes$ denotes the Kronecker product. Thereby,

$$D_n = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ 0 & & \dots & & 0 \end{pmatrix} \in \mathbb{R}^{n \times n} \tag{2.28}$$

denotes the *forward difference matrix* with mirrored boundary conditions. Eventually, the *discrete total variation* of a vector $f : \{1, \dots, N\} \to \mathbb{R}$ is given by

$$\mathrm{TV}(f) := \| \, |\boldsymbol{D} f| \, \|_1 = \sum_{j=1}^{N} |\boldsymbol{D} f|(j) = \sum_{j=1}^{n} \sqrt{f_{x_1}(j)^2 + f_{x_2}(j)^2}. \tag{2.29}$$

With the discrete total variation at hand, let us turn to a TV-regularized segmentation model. In [79] we proposed an image segmentation method that consists of tackling

$$\text{(TVS)} \qquad \operatorname*{argmin}_{u,r} E_{\mathrm{TV}}(u, r) \quad \text{subject to} \quad u \in S_c^N, \; S_c = \mathrm{conv}\{e_1, \dots, e_c\}$$

$$\text{where } E_{\mathrm{TV}} = \sum_{k=1}^{c} \sum_{j=1}^{N} u_k(j) \, \|r_k - f(j)\|_2^2 \; + \; \lambda \sum_{k=1}^{c} \mathrm{TV}(u_k)$$

by alternating optimization with only a single step in the computation of a new iterate $u^{(n)}$ for fixed $r^{(n)}$. Here, we call our method from [79] *single step total variation c-means* (STVcM). We obtained promising results in practical applications on medical images with STVcM. However, a proof of convergence was not given. Indeed, there exist parameter configurations for which STVcM does not converge in practice. To overcome this drawback, we propose a slight modification of STVcM, namely *total variation c-means* (TVcM), which is a particular implementation of RcC with the according convergence results of Section 2.3. In numerical experiments we show that the performance of TVcM is similar in case STVcM converges numerically. Additionally, we will see that in those cases where STVcM denies to converge in applications, TVcM still does converge.

The remaining part of this section is organized as follows. Section 2.4.1 describes how to solve (TVS) with fixed cluster centers r via convex optimization. Section 2.4.2 proposes the alternating algorithms TVcM and STVcM for a solution of (TVS) in both variables u and r. Thereby, we utilize the algorithms reviewed in Section 2.4.1 for the optimization step in u. Section 2.4.3 describes the relations of our algorithm to the famous models of Mumford and Shah [113], and Chan and Vese [42].

Numerical experiments that show the performance of our algorithms can be found in the following Section 2.5.

2.4.1 Convex Optimization Algorithms for Segmentation with Fixed Codebook

In this Subsection we describe how to solve (TVS) for fixed codebook r, see [140, 79]. We review two efficient algorithms for the solution of convex and non-smooth problems, namely

the *alternating direction method of multipliers* (ADMM) and the *primal dual hybrid gradient algorithm with modified dual variable* (PDHGM*b*). Note that there also exist other methods suitable for our optimization problems such as the algorithm of Nesterov [114] and the fast iterative shrinkage-thresholding algorithm (FISTA) of Beck and Teboulle [12].

In general, both algorithms ADMM and PDHGM*b* are suitable for convex optimization problems of the form

$$\begin{aligned} &\underset{x\in\mathbb{R}^{N_1},y\in\mathbb{R}^{N_2}}{\operatorname{argmin}} \{G_1(x)+G_2(y)\} \\ &\text{subject to } Ax=y \end{aligned} \tag{2.30}$$

with proper, lsc, convex functions $G_1 : \mathbb{R}^{N_1} \to \mathbb{R}\cup\{+\infty\}$, $G_2 : \mathbb{R}^{N_2} \to \mathbb{R}\cup\{+\infty\}$ and $A \in \mathbb{R}^{N_2\times N_1}$. Further, both algorithms split the problem into smaller sub-problems that have to be solved alternatingly as stated in Algorithm 5 for ADMM.

Algorithm 5: ADMM

Initialization: $y^{(0)},\ b^{(0)} \in \mathbb{R}^{N_2}$, $\gamma > 0$.

for $n = 0, 1, \ldots$ *until convergence* **do**

$$x^{(n+1)} := \underset{x\in\mathbb{R}^{N_1}}{\operatorname{argmin}} \left\{ G_1(x) + \frac{1}{2\gamma}\left\| b^{(n)} + Ax - y^{(n)} \right\|_2^2 \right\} \tag{2.31}$$

$$y^{(n+1)} := \underset{y\in\mathbb{R}^{N_2}}{\operatorname{argmin}} \left\{ G_2(y) + \frac{1}{2\gamma}\left\| b^{(n)} + Ax^{(n+1)} - y \right\|_2^2 \right\}$$

$$b^{(n+1)} := b^{(n)} + Ax^{(n+1)} - y^{(n+1)}$$

end

For problems of the form (2.30) the ADMM coincides with the alternating split Bregman method [67] and with the Douglas-Rachford splitting method applied to the dual problem. The following assumptions guarantee convergence of the sequence $(x^{(n)}, y^{(n)})_{n\in\mathbb{N}}$ generated by Algorithm 5 to a solution of (2.30):

1. The Lagrangian

   $$L(x, y, p) = G_1(x) + G_2(y) + \langle p, Ax - y\rangle$$

   has a saddle point. A point $(\hat{x}, \hat{y}, \hat{p})$ is called saddle point of L if $L(\hat{x}, \hat{y}, p) \le L(\hat{x}, \hat{y}, \hat{p}) \le L(x, y, \hat{p})$ for all $(x^\top, y^\top)^\top \in \mathbb{R}^{N_1} \times \mathbb{R}^{N_2}$ and $p \in \mathbb{R}^{N_2}$.

2. The solution of the first step (2.31) is unique, see [138, Theorem 5].

All these assumptions are fulfilled by our problems. For more details see, e.g., [24, 54, 56, 62, 112, 138].

The PDHGM*b* algorithm can be formulated such that its only difference to ADMM is a distinct first step given by

$$x^{(n+1)} := \operatorname*{argmin}_{x \in \mathbb{R}^{N_1}} \left\{ G_1(x) + \frac{1}{2\tau} \left\| x - (x^{(n)} + \frac{\tau}{\gamma} A^\top (2b^{(n)} - b^{(n-1)}) \right\|_2^2 \right\}.$$

To use PDHGM*b* instead of ADMM can be useful if there occurs a linear system in the first ADMM step that cannot be solved efficiently. Regarding the convergence, let us assume that a saddle point of the Lagrangian exists and that $\frac{\tau}{\gamma} < \|A\|^2$. Then, PDHGM*b* converges as shown by Chambolle and Pock [39]. See also [56, 176].

To be able to apply the mentioned convex optimization algorithms we reformulate the objective of (TVS) for fixed r to

$$\begin{aligned} &\operatorname*{argmin}_{u,v,w} \sum_{j=1}^{N} \sum_{k=1}^{c} u_k(j) s_k(j) + \lambda \sum_{k=1}^{c} \| \, |v_k| \, \|_1 + \iota_{S_c^N}(w) \\ &\text{subject to} \quad \underbrace{\begin{pmatrix} I_c \otimes \boldsymbol{D} \\ I_{cN} \end{pmatrix}}_{A} u = \begin{pmatrix} v \\ w \end{pmatrix} \end{aligned} \tag{2.32}$$

where $u, w \in \mathbb{R}^{cN}$, $v \in \mathbb{R}^{2cN}$. The choice $s_k(j) := \|r_k - f(j)\|_2^2$ resembles (TVS) for fixed codebook. Of course, one can use any other constant vector for s. Further,

$$v_k = (v_{k,x}^\top, v_{k,y}^\top)^\top, \; v_{k,x}, v_{k,y} \in \mathbb{R}^N, \; |v_k| = \left(\sqrt{v_{k,x}(j)^2 + v_{k,y}(j)^2} \right)_{j=1}^{N} \in \mathbb{R}^N$$

according to the definition of the TV based regularizer of (TVS). Problem (2.32) is a non-smooth and convex optimization problem. Its *global* minimizer can be computed, e.g., by ADMM as stated in Algorithm 6.

Algorithm 6: ADMM for (2.32)

Input: image f, number of clusters c, constant $s_k(j)$ for all $k = 1, \dots, c$ and $j = 1, \dots, N$, parameters $\lambda > 0$, $\gamma > 0$

Output: segment membership vector u

Initialization of $u^{(0)} \in S_c^N$, $v^{(0)} \in \mathbb{R}^{2cN}$, $w^{(0)} \in \mathbb{R}^{cN}$, $b^{(0)} \in \mathbb{R}^{3cN}$

for $n = 0, 1, 2, \dots$ *until convergence* **do**

$$u^{(n+1)} := \underset{u \in S_c^N}{\operatorname{argmin}} \sum_{k=1}^{c} \sum_{j=1}^{N} u_k(j) s_k(j) + \frac{1}{2\gamma} \left\| b^{(n)} + Au - \left((v^{(n)})^\top, (w^{(n)})^\top \right)^\top \right\|_2^2 \tag{2.33}$$

$$\begin{pmatrix} v^{(n+1)} \\ w^{(n+1)} \end{pmatrix} := \underset{v,w}{\operatorname{argmin}} \lambda \sum_{k=1}^{c} \| \, |v_k| \, \|_1 + \iota_{S_c^N}(w) + \frac{1}{2\gamma} \left\| b^{(n)} + Au^{(n+1)} - \begin{pmatrix} v \\ w \end{pmatrix} \right\|_2^2 \tag{2.34}$$

$$b^{(n+1)} := b^{(n)} + Au^{(n+1)} - \left((v^{(n)})^\top, (w^{(n)})^\top \right)^\top$$

end

In the remaining part of this section, we explain how to solve steps (2.33) and (2.34) efficiently. Note that in step (2.34) we can compute $v^{(n+1)}$ and $w^{(n+1)}$ independently.

Computing $u^{(n+1)}$

Here, we focus on the case $d = 2$. For the case $d \geq 3$ see Chapter A.1 in the appendix.

Since

$$\sum_{k=1}^{c} \sum_{j=1}^{N} u_k(j) s_k(j) + \frac{1}{2\gamma} \left\| b^{(n)} + Au - \left((v^{(n)})^\top, (w^{(n)})^\top \right)^\top \right\|_2^2$$

is differentiable in u we can compute its minimizer by setting the gradient to zero. Thus, the minimizer is given by the solution of the linear system

$$A^\top A u^{(n+1)} = A^\top \left(\begin{pmatrix} v^{(n)} \\ w^{(n)} \end{pmatrix} - \underbrace{\begin{pmatrix} b_v^{(n)} \\ b_w^{(n)} \end{pmatrix}}_{b^{(n)}} \right) - \gamma s. \tag{2.35}$$

Algorithm 6 works fast, if this linear system can be solved efficiently. Indeed, this is the case as we can see in the following.

With the rules of the Kronecker product

$$(A \otimes B)^\top = A^\top \otimes B^\top \text{ and } (A \otimes B)(C \otimes D) = AC \otimes BD \tag{2.36}$$

we have

$$A^\top A = \begin{pmatrix} I_c \otimes \boldsymbol{D}^\top \,,\, I_{cN} \end{pmatrix} \begin{pmatrix} I_c \otimes \boldsymbol{D} \\ I_{cN} \end{pmatrix} = (I_c \otimes \boldsymbol{D}^\top \boldsymbol{D}) + I_{cN}.$$

Hence, we solve for every $k = 1, \ldots, c$ the linear system of equations

$$(\boldsymbol{D}^\top \boldsymbol{D} + I_N) u_k = a_k \quad \text{with} \quad a_k := \boldsymbol{D}^\top \left(v_k^{(n)} - b_{v,k}^{(n)} \right) + \left(w_k^{(n)} - b_{w,k}^{(n)} \right) - \gamma s \tag{2.37}$$

and obtain the solution of (2.35). To this end, we diagonalize $D_n^\top D_n$ by the *discrete cosine II transformation matrix* (DCT-II matrix) C_n, i.e.,

$$D_n^\top D_n = C_n^\top \operatorname{diag}(q_n)\, C_n, \quad q_n := \left(4 \sin^2 \frac{\pi j}{2n} \right)_{j=0}^{n-1} \tag{2.38}$$

with the orthogonal *DCT-II matrix*

$$C_n := \sqrt{\frac{2}{n}} \left(\epsilon_j \cos \frac{j(2k+1)\pi}{2n} \right)_{j,k=0}^{n-1}, \quad \epsilon_j := \begin{cases} 1/\sqrt{2} & j = 0, \\ 1 & \text{otherwise}, \end{cases} \tag{2.39}$$

see [122, 126]. After straightforward computation we obtain

$$(\boldsymbol{D}^\top \boldsymbol{D} + I_N) = (C_{n_2}^\top \otimes C_{n_1}^\top)(\Lambda + I_N)(C_{n_2} \otimes C_{n_1})$$

where $\Lambda = I_{n_2} \otimes \operatorname{diag}(q_{n_1}) + \operatorname{diag}(q_{n_2}) \otimes I_{n_1}$, see Lemma A.1 in the appendix. Finally, we obtain

$$u_k = (C_{n_2}^\top \otimes C_{n_1}^\top)(\Lambda + I_N)^{-1}(C_{n_2} \otimes C_{n_1}) a_k. \tag{2.40}$$

Firstly, note that the discrete cosine transform of a vector, i.e., its multiplication with the cosine matrix C_n, can be computed with $\mathcal{O}(N \log N)$ arithmetic operations. Secondly, we do not utilize Kronecker products in our numerical computations. Instead, we work with reformulations to matrix – matrix operations based on another rule of the Kronecker product, namely

$$(A \otimes B)\operatorname{vec}(F) = \operatorname{vec}(BFA^\top), \tag{2.41}$$

see (A.2) in the appendix.

Computing $v^{(n+1)}$

The iterate $v^{(n+1)}$ can be computed for each k separately, i.e.,

$$v_k^{(n+1)} = \operatorname*{argmin}_{v_k \in \mathbb{R}^{2N}} \left\{ \lambda \parallel |v_k| \parallel_1 + \frac{1}{2\gamma} \big|\big| v_k - \underbrace{(b_{v,k}^{(n)} + \boldsymbol{D} u_k^{(n+1)})}_{=: g_{v,k}} \big|\big|_2^2 \right\}. \tag{2.42}$$

The minimizer $v_k^{(n+1)}$ can be obtained now in parallel for every image pixel j by the so-called *coupled shrinkage* of $g_{v,k}$ with *threshold* $\lambda\gamma$ given by

$$v_k(j)^{(n+1)} = \begin{cases} 0 & \text{if } |g_{v,k}|(j) \le \lambda\gamma, \\ g_{v,k}(j) \left(1 - \dfrac{\lambda\gamma}{|g_{v,k}|(j)} \right) & \text{otherwise}, \end{cases} \tag{2.43}$$

see, e.g., Chapter A.2 in the appendix or [138].

Computing $w^{(n+1)}$

To determine $w^{(n+1)}$ we have to solve

$$w^{(n+1)} = \operatorname*{argmin}_{w \in \mathbb{R}^{cN}} \left\{ \iota_{S_c^N}(w) + \frac{1}{2\gamma} \big|\big| w - \underbrace{(b_w^{(n)} + u^{(n+1)})}_{=:g_w} \big|\big|_2^2 \right\}.$$

In other words, we have to find the orthogonal projection of g_w onto the simplex S_c^N. We can project separately for every $j = 1, \dots, N$. This means that we have to solve N independent sub-problems of the form

$$\operatorname*{argmin}_{x \in \mathbb{R}^c} \frac{1}{2} \|x - g\|_2^2 + \iota_{S_c}.$$

This projection onto the unit simplex $S_c \in \mathbb{R}^c$ is explained in the following remark, see [110, 52].

Remark 2.26. *The orthogonal projection of $y \in \mathbb{R}^c$ onto the hyperplane*

$$\left\{ x \in \mathbb{R}^c : \sum_{k=1}^{c} x_k = \langle x, \mathbf{1}_c \rangle = 1 \right\}$$

containing the unit simplex is given by

$$\hat{w} = y - \frac{1}{c}(\sum_{k=1}^{c} y_k - 1)\mathbf{1}_c,$$

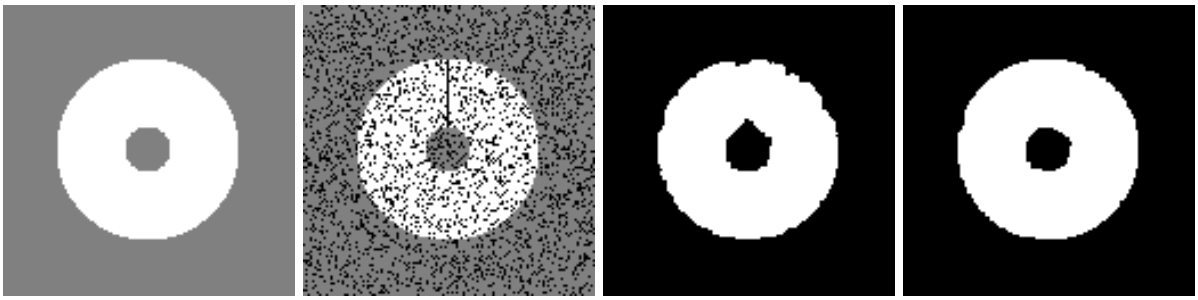
where $\mathbf{1}_c := (1, \dots, 1)^\top \in \mathbb{R}^c$. Further, we compute the projection onto S_c by the following steps:

1. *If $\hat{w} \geq 0$ then $\hat{w}$ is contained in S_c and we are done. Otherwise, we denote by $\mathcal{I}$ the index set of all negative components and set $\hat{w}_k = 0$ for all $k \in \mathcal{I}$.*
2. *We compute the projection of the reduced vector without zeros $(\hat{w}_k)_{k \in \mathcal{I}} \in \mathbb{R}^{c-|\mathcal{I}|}$ onto the reduced hyperplane $\{x \in \mathbb{R}^{c-|\mathcal{I}|} : \langle x, 1_{c-|\mathcal{I}|} \rangle = 1\}$ and go back to the previous step.*

One possibility to tackle the multi-class segmentation is to determine the codebook beforehand (e.g., by fuzzy c-means) and fix it for the optimization over u. This can result in sufficiently good segmentations in terms of the application's demands, see, e.g., Section 2.6.1 and the corresponding publication [140]. However, there exist images where alternating optimization over both variables r and u improves results significantly. For an artificial example, see Figure 2.9 and also [79].

2.4.2 The TVcM Algorithm

In this subsection we discard the assumption of a fixed codebook from the previous Subsection 2.4.1 and propose an algorithm for the solution of (TVS). Of course, our problem is in contrast to Subsection 2.4.1 not convex but biconvex. This means, we cannot hope to find a global minimizer. Instead, we seek a partial minimizer defined in (2.4).



(a) Ground truth (b) Corrupted input (c) Segmentation using a fixed codebook r (d) Optimizing codebook r during segmentation

Figure 2.9.: Segmentation with fixed and optimized codebook. The black part in the segmentation results corresponds to the background in the ground truth. a) Original image. b) Corrupted input with about 20% data missing. c) Codebook r determined with fuzzy c-means before optimizing over u with Algorithm 6. d) Result of Algorithm 8 including optimization over r. 500 iterations and $\lambda = 0.4$.

To find a partial minimizer we alternatingly fix one variable and update the other variable. To this end, we propose Algorithm 7 in [79]. The optimization of r is the computation of a weighted mean and can be done in a single step.

To update u our suggestion in [79] is to perform one single step of Algorithm 6. We call this method *single step total variation c-means* (STVcM). Since ADMM and thus STVcM does not converge monotonically, the theory of Zangwill [172] for monotonic algorithms cannot be applied to STVcM. And even in case STVcM converges experimentally, we cannot be sure that the limit is a partial minimizer. As we will see in the numerical experiments of Section 2.5, there exist parameter configurations such that STVcM does not converge in practice.

To overcome these drawbacks, we propose the slight modification *total variation c-means* (TVcM) as stated in Algorithm 8. To update u we now propose to iterate with Algorithm 6 for fixed r until we observe a decrease in E_{TV} of $\delta > 0$, see the inner for loop of TVcM. Thus, we obtain a particular implementation of RcC and the according convergence results. The price we have to pay for the convergence theory is that we must evaluate the objective functional in every iteration of TVcM. For an evaluation of the small cost differences between both Algorithms 7 and 8, see Section 2.5.

Algorithm 7: Our proposal in [79] (STVcM)

Input: image f, number of clusters c, parameters $\lambda > 0$, $\gamma > 0$

Output: segment membership vector u and codebook r

Initialization of $u^{(0)} \in S_c^N, v^{(0)} \in \mathbb{R}^{2cN}, w^{(0)} \in \mathbb{R}^{cN}, b^{(0)} \in \mathbb{R}^{3cN}$

for $n = 0, 1, 2, \ldots$ *until convergence* **do**

$$r_k^{(n+1)} := \frac{\sum_{j=1}^N u_k^{(n)}(j) f(j)}{\sum_{j=1}^N u_k^{(n)}(j)}$$

$$u^{(n+1)} :=$$

$$A^\top A^{-1}\Big(A^\top(((v^{(n)})^\top, (w^{(n)})^\top)^\top - \overbrace{((b_v^{(n)})^\top, (b_w^{(n)})^\top)^\top}^{b^{(n)}}\Big) - \gamma((((\|r_k - f(j)\|_2^2)_{j=1}^N)_{k=1}^c)$$

$$g_{v,k} := b_{v,k}^{(n)} + \boldsymbol{D} u_k^{(n+1)} \text{ for } k = 1 \ldots, c$$

Compute for $j = 1, \ldots, N$ and $k = 1, \ldots, c$ the coupled shrinkage

$$v_k^{(n+1)}(j) := \begin{cases} 0 & \text{if } |g_{v,k}(j)| \leq \gamma \\ g_{v,k}(j)\left(1 - \frac{\gamma}{|g_{v,k}(j)|}\right) & \text{otherwise} \end{cases}$$

For $j = 1, \ldots, N$ compute $w^{(n+1)}(j)$ as the orthogonal projection of $b_w^{(n)}(j) + u^{(n+1)} \in \mathbb{R}^c$ onto the unit simplex $S_c \subset \mathbb{R}^c$

$$b^{(n+1)} := b^{(n)} + A u^{(n+1)} - \left((v^{(n)})^\top, (w^{(n)})^\top\right)^\top$$

end

Algorithm 8: TVcM

Input: image f, number of clusters c, parameters $\lambda > 0$, $\varepsilon_{\text{emp}} > 0, \delta > 0$, $\gamma > 0$

Output: segment membership vector u and codebook r

Initialization of $r^{(0)} \in \mathbb{R}^{cm}$, $E_{\text{old}} = \infty, v^{(0)} \in \mathbb{R}^{2cN}, w^{(0)} \in \mathbb{R}^{cN}$, and $b^{(0)} \in \mathbb{R}^{3cN}$

for $n = 0, 1, 2, \ldots$ *until convergence* **do**

  for $l = 0, 1, 2, \ldots$ *until* $E_{\text{TV}}(w^{(l+1)}, r^{(n)}) + \delta \leq E_{\text{old}}$ *or convergence* **do**

    $\tilde{u}^{(l+1)} :=$

    $$A^\top A^{-1}\Big(A^\top\Big(((v^{(l)})^\top, (w^{(l)})^\top)^\top - \overbrace{((b_v^{(l)})^\top, (b_w^{(l)})^\top)^\top}^{b^{(l)}}\Big)\Big) - \gamma(((\Big\|r_k^{(n)} - f(j)\Big\|_2^2)_{j=1}^N)_{k=1}^c)$$

    $g_{v,k} := b_{v,k}^{(l)} + \boldsymbol{D}\tilde{u}_k^{(l+1)}$ for $k = 1 \ldots, c$

    Compute for $j = 1, \ldots, N$ and $k = 1, \ldots, c$ the coupled shrinkage

    $$v_k^{(l+1)}(j) := \begin{cases} 0 & \text{if } |g_{v,k}(j)| \leq \gamma \\ g_{v,k}(j)\left(1 - \frac{\gamma}{|g_{v,k}(j)|}\right) & \text{otherwise} \end{cases}$$

    For $j = 1, \ldots, N$ compute $w^{(l+1)}(j)$ as the orthogonal projection of $b_w^{(l)}(j) + \tilde{u}^{(l+1)} \in \mathbb{R}^c$ onto the unit simplex $S_c \subset \mathbb{R}^c$

    $b^{(l+1)} := b^{(l)} + A\tilde{u}^{(n+1)} - ((v^{(l)})^\top, (w^{(l)})^\top)^\top$

  end

  $u^{(n)} := w^{(l+1)}$

  `// Check for empty cluster`

  for $k := 1, \ldots, c$ **do**

    if $\sum_{j=1}^N u_k^{(n)}(j) < \varepsilon_{\text{emp}}$ **then**

      Reduce c by 1 and restart with $(r_l^{(n)})_{l \neq k}$ as initial solution

    end

  end

  $$r_k^{(n+1)} := \frac{\sum_{j=1}^N u_k^{(n)}(j) f(j)}{\sum_{j=1}^N u_k^{(n)}(j)}$$

  $E_{\text{old}} := E_{\text{TV}}(u^{(n)}, r^{(n+1)})$

end

2.4.3 Relations to the Models of Chan and Vese, and Mumford and Shah

In this subsection, we describe the relations of (TVS) to the models of Chan and Vese [42], and Mumford and Shah [113]. These models where introduced in the continuous setting where images are represented by functions and not by vectors of finite dimension. We stick to this

continuous formulation in this section. The reason is that Mumford and Shah minimize over curves separating segments. To this end, the continuous setting is more intuitive.

Recall from Section 1.1.1 that Mumford and Shah [113] propose to solve

$$\text{(MS)} \qquad \operatorname*{argmin}_{v\in W^{1,2}(\Omega\setminus K),K\subseteq\Omega} E_{\mathrm{MS}}(v,K)$$
$$\text{where } E_{\mathrm{MS}}(v,K)=\int_\Omega (f(x)-v(x))^2dx+\mu\int_{\Omega\setminus K}\|\nabla v(x)\|_2^2\,dx+\nu\mathcal{H}^{d-1}(K)$$

to obtain a segmentation of $f\in L^\infty(\Omega)$. $\mathcal{H}^{d-1}(K)$ denotes the $d-1$-dimensional *Hausdorff measure* of the discontinuity set K as defined in (1.1) and $\Omega\subset\mathbb{R}^d$ is open and bounded.

For two classes $c=2$ and with restriction of (MS) to piecewise constant v, i.e., $v_k|_{\Omega_k}=r_k$ where $\Omega_1=\Sigma$ and $\Omega_2=\Omega\setminus\bar{\Sigma}$, the model of Chan and Vese [42] is given by

$$\text{(CV)} \qquad \operatorname*{argmin}_{r_1,r_2,\Sigma} E_{\mathrm{CV}}(r_1,r_2,\Sigma)$$
$$\text{where } E_{\mathrm{CV}}(r_1,r_2,\Sigma)=\int_\Omega (f(x)-r_1)^2dx+\int_\Omega (f(x)-r_2)^2dx+\nu\,\mathrm{Per}(\Sigma,\Omega).$$

The perimeter Per of a set is defined in terms of the total variation of the characteristic function $\chi_\Sigma:\Omega\to\mathbb{R}$ by $\mathrm{Per}(\Sigma,\Omega)=\mathcal{TV}(\chi_\Sigma)$. The perimeter coincides with the boundary length ($d=2$) for sufficiently smooth sets, see Section 1.1.1.

For fixed Σ the minimizer is given segment-wise by

$$\hat{r}_k=\frac{\int_{\Omega_k}f(x)dx}{\mathcal{L}^d(\Omega_k)}$$

where $\mathcal{L}^d$ denotes the d-dimensional Lebesgue measure corresponding to the area of Ω_k. To minimize E_{CV} over Σ for fixed r, Chan and Vese [42] introduce a level-set formulation of the model and optimize this variational problem by solving the resulting Euler-Lagrange differential equation numerically. The algorithm of Chan and Vese consists similarly to TVcM of alternating minimization for fixed r and fixed Σ.

In the following, let us explain the connection between our model (TVS) for $c=2$ and (CV) formulated in the continuous setting. The continuous 2-class version of (TVS) reads

$$\operatorname*{argmin}_{\substack{u_1,u_2\in\\ BV(\Omega)}}\int_\Omega u_1(x)(r_1-f(x))^2+u_2(x)(r_2-f(x))^2dx+\lambda\left(\mathcal{TV}(u_1)+\mathcal{TV}(u_2)\right)$$
$$\text{subject to}\quad (u_1(x),u_2(x))^\top\in S_2=\mathrm{conv}\{e_1,e_2\}\text{ for all }x\in\Omega$$

for gray-valued images $f:L^\infty(\Omega)$. Due to the constraint $u_1(x)+u_2(x)=1$ we have

$$\begin{aligned}&\int_\Omega u_1(x)(r_1-f(x))^2+(1-u_1(x))(r_2-f(x))^2dx+\lambda\left(\mathcal{TV}(u_1)+\mathcal{TV}(1-u_1)\right)\\&=\int_\Omega u_1(x)\left((r_1-f(x))^2-(r_2-f(x))^2\right)+(r_2-f(x))^2dx+2\lambda\,\mathcal{TV}(u_1)dx.\end{aligned}\tag{2.44}$$

Of course, a minimizer of (2.44) over $u_1 \in BV(\Omega)$ is contained in

$$\operatorname*{argmin}_{u \in BV(\Omega)} \int_\Omega u(x) \left((r_1 - f(x))^2 - (r_2 - f(x))^2 \right) dx + \nu \, \mathcal{TV}(u) dx$$

where $2\lambda = \nu$. Moreover, there exists the subsequent theorem, see [115, 41] and the references in [41]. It mainly states that a global minimizer of $E_{\mathrm{CV}}(r_1, r_2, \cdot)$ can be found by solving a convex optimization problem and performing a thresholding with an arbitrary[2] threshold $\alpha \in (0, 1]$ afterwards. The proof uses the co-area formula for functions of bounded variation [59].

Theorem 2.27. *Let $r_1, r_2 \in \mathbb{R}$ be given and fixed. Then, a global minimizer of $E_{\mathrm{CV}}(r_1, r_2, \cdot)$ is given by $\hat{\Sigma} = \{x : \hat{u}(x) \geq \alpha\}$ for arbitrary $\alpha \in (0, 1]$ where*

$$\hat{u} \in \operatorname*{argmin}_{u \in BV(\Omega)} \int_\Omega u(x) \left((r_1 - f(x))^2 - (r_2 - f(x))^2 \right) dx + \nu \, \mathcal{TV}(u). \tag{2.45}$$

Remark 2.28. *A minimizer $\hat{u}$ itself does not have to be binary. To see this, consider Figure 2.10. Figure 2.10a shows an artificial example consisting of 3 different image regions. Solving* (2.45) *with Figure 2.10a as input results in a minimizer $\hat{u}$ far from being binary as revealed by Figure 2.10b. The solutions after thresholds* 0.3 *and* 0.7 *are depicted in Figures 2.10c and 2.10d, respectively. The two white regions are both global minimizers of $E_{\mathrm{CV}}(r_1, r_2, \cdot)$. However, the thresholded versions of $\hat{u}$ both also minimize* (2.45)*. In practical applications $\hat{u}$ is often close to binary as observed in [115].*

Finally, the convex problem for two classes (2.44) is tackled for fixed codebook r in every iteration of TVcM. This is equivalent to the convex problem (2.45) in terms of minimizers. However, in TVcM we do not propose a full optimization for fixed r. Moreover, the update of r is different. More precisely, we cope with the relaxed problem where $(u_1(x), u_2(x))^\top \in S_2$. This results in the weighted mean given in continuous notation by

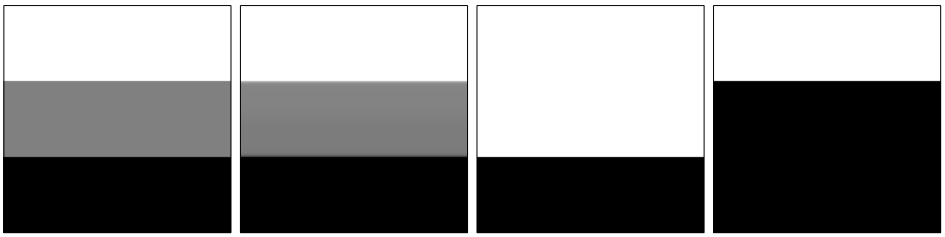
$$r_k = \frac{\int_\Omega u_k(x) f(x) dx}{\int_\Omega u_k(x) dx}.$$

In contrast, Chan and Vese [42] use the standard mean

$$r_k = \frac{\int_{\Omega_k} f(x) dx}{\mathcal{L}^d(\Omega_k)}$$

and solve the unrelaxed problem $u(x) \in \{0, 1\}$. Hence, TVcM can be interpreted as a relaxed multi-class Chan and Vese algorithm. Furthermore, note that the algorithm of Chan and Vese in the discrete case with global optimization for fixed r is a particular implementation of RcC stated in Algorithm 4 with the corresponding convergence theory developed in this work.

[2] In their formulation Nikolova et al. [115] stated "for almost all $\alpha \in [0, 1]$" instead of "for arbitrary $\alpha \in (0, 1]$". However, Chambolle et al. [41] show that if the theorem holds true for almost all $\alpha \in [0, 1]$ it also holds true for arbitrary $\alpha \in (0, 1]$.



(a) Input (b) $\hat{u}$ (c) Threshold of $\hat{u}$ at 0.3 (d) Threshold of $\hat{u}$ at 0.7

Figure 2.10.: Example of non-binary solutions $\hat{u}$ of (2.45) and two different minimizers of E_{CV} with fixed codebook $r_1 = 1, r_2 = 0$ and $\nu = 1$.

2.5 Numerical Experiments

The purpose of this section is twofold. Firstly, we review some of the most promising results obtained in [79] with Algorithm 7. To this end, we introduce briefly the moving-average FCM [84] for comparison. For more experiments on the performance of STVcM we refer to [79].

Secondly, we compare the performance of TVcM and STVcM in terms of speed and objective value. Additionally, we use the alternate convex search algorithm (ACS) described in Remark 2.10 applied to (TVS) for comparison. We show that a full minimization in every iteration with ACS provides the worst results of all algorithms, while the other two perform similarly in case of a suitable parameter choice. In case of an unsuitable parameter choice we show that STVcM can deny to converge.

In all our experiments in this section we rescale the images to $[0, 1]$. Further, we use the result of fuzzy c-means as initial solution for the regularized algorithms. Fuzzy c-means, on the other hand, was initialized either randomly or by $r_k = ((k-1)/c + 1/(2c))_{i=1}^{m}$. We use Matlab on an Intel i7 2.93Ghz machine.

2.5.1 Segmentation of Simulated Brain Tissue Images

In [79] we obtained good segmentation results with STVcM[3]. Here, we will review the most convincing experiments including brain tissue segmentation and refer to [79] for additional evaluation of the method. To compare the performance we review briefly the *moving-average FCM* [84] designed for brain tissue segmentation in the subsequent paragraph.

[3]There is one negligible difference between our version in [79] and (TVS). In [79] the regularization parameter λ appearing in E_{TV} is multiplied with the data term and not with the regularizer.

Moving-Average FCM Moving-average FCM is a regularized version of fuzzy c-means introduced in Section 2.2. As regularizer, Hou et al. [84] use

$$\sum_{k=1}^{c}\sum_{j=1}^{N}(u_k(j)-\bar{u}_k(j))^2$$

where $\bar{u}_k(j) = 1/9\sum_{i\in\mathcal{N}_j^8} u_k(i)$ for all $j \in \{1,\dots,N\}$. Further, $\mathcal{N}_j^8$ denotes the neighborhood of the 9 closest pixels in terms of position including the pixel itself. For gray-valued images $f : \{1,\dots,N\} \to \mathbb{R}$, this results in the optimization problem

$$\text{(MA)} \qquad \operatorname*{argmin}_{u,r} E_{\mathrm{MA}}(u,r) \quad \text{subject to} \quad u \in S_c^N = \operatorname{conv}\{e_1,\dots,e_c\}^N$$
$$\text{where } E_{\mathrm{MA}}(u,r) = \sum_{k=1}^{c}\sum_{j=1}^{N} u_k^2(j)(r_k - f(j))^2 + \beta\sum_{k=1}^{c}\sum_{j=1}^{N}(u_k(j)-\bar{u}_k(j))^2.$$

and $\beta > 0$. To solve (MA), again alternating minimization over r and u is performed as stated in Algorithm 9. In every iteration and for each variable necessary first order optimality conditions are solved. More precisely, for fixed u we solve the problem in r by setting the gradient to zero. For fixed r we minimize the problem in u by solving the Karush-Kuhn-Tucker conditions

$$\nabla_u\left(E_{\mathrm{MA}}(u,r) + \langle\mu, 1-\sum_{k=1}^{c}u_k\rangle + \langle\lambda,u\rangle\right) = 0,\ \langle u,\lambda\rangle = 0,\ \lambda\geq 0,\ 1-\sum_{k=1}^{c}u_k = 0,\ u\geq 0.$$

Regarding the convergence of Algorithm 9, note that moving-average FCM is a particular implementation of RcC with the corresponding convergence theory. Hence, we know that the sequence of iterates either converges to a partial minimizer of E_{MA} or contains a subsequence that does.

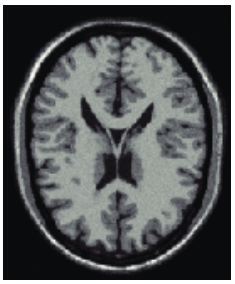


Figure 2.11.: One slice of phantom data with 3% noise.

Algorithm 9: Moving-Average FCM

Input: data set f, number of clusters c, parameter $\varepsilon_{\text{emp}} > 0$
Output: segment membership vector u and codebook r
Initialization of $r^{(0)} \in \mathbb{R}^{cm}$ and $u^{(-1)} \in S_c^N$
for $n := 0, \ldots$ **do**
  for $k := 1, \ldots, c$ **do**

$$u_k^{(n)}(j) := \frac{1 - \sum_{l=1}^{c} \frac{\beta \bar{u}_l^{(n-1)}(j)}{(f(j)-r_l)^2+\beta} + \beta \bar{u}_k^{(n-1)}(j) \sum_{l=1}^{c} \frac{1}{(f(j)-r_l)^2+\beta}}{((f(j)-r_k)^2+\beta) \sum_{l=1}^{c} \frac{1}{(f(j)-r_l)^2+\beta}}$$

  end

  `// Check for empty cluster`
  for $k := 1, \ldots, c$ **do**
    if $\sum_{j=1}^{N} u_k^{(n)}(j) < \varepsilon_{\text{emp}}$ **then**
      Reduce c by 1 and restart with $(r_l^{(n)})_{l \neq k}$ as initial solution
    end
  end

  for $k := 1, \ldots, c$ **do**
    $r_k^{(n+1)} := \sum_{j=1}^{N} (u_k^{(n)}(j))^2 f(j) / \sum_{j=1}^{N} (u_k^{(n)}(j))^2$
  end

end

Evaluation and Results For the evaluation and comparison of fuzzy c-means [20, 18], moving-average FCM [84], and STVcM we use in [79] simulated brain tissue data [47, 94, 95] generated with the online tool of Cocosco et al. [46]. We select one slice in the x-y-plane corresponding to $z = 90$ of simulated 3d data of size $181 \times 217 \times 181$ ($x \times y \times z$). The modality of the phantom

(a) Target area (b) Ground truth

Figure 2.12.: Target pixels and ground truth for the segmentation of phantom brain data.

data is T1 and the thickness of the slices is 1mm, see Figure 2.11. We tested levels of additive Gaussian noise between 1% and 49% added only to the foreground. The noise level is given by $\sigma / \max_j f_{\mathrm{gt}}(j)$ where f_{gt} denotes the ground truth given in Figure 2.12b and σ is the standard deviation of the Gaussian noise. As performance measure we use the target accuracy

$$\mathrm{TA} = \frac{\text{number of correctly classified target pixels}}{\text{number of target pixels}}.$$

Figure 2.12a shows the target pixels. To obtain the target pixels we thresholded the ground truth in Figure 2.12b such that all background pixels are 0 and all foreground pixels are 1. Then we applied morphological dilation with a square structuring element of size 4 to the thresholded image to enlarge the target area a little (see, e.g., [145] for an introduction to mathematical morphology). Note that for the segmentation we removed the outer parts of Figure 2.11 and obtain a 4 class segmentation problem (white matter, gray matter, cerebrospinal fluid, and background).

Table 2.2 and the corresponding plot in Figure 2.13 reveal that for low noise levels the performance of the algorithms in terms of target accuracy is similar. In contrast, for high noise levels our STVcM clearly outperforms fuzzy c-means and moving-average FCM. This effect can also be seen in the resulting segmentations as shown in Figures 2.14-2.17.

Noise level	TA of fuzzy c-means	TA of moving-average FCM	TA of STVcM
1 %	0.9830	0.9830	0.9825
3 %	0.9712	0.9724	0.9731
5 %	0.9513	0.9596	0.9594
7 %	0.9159	0.9445	0.9465
9 %	0.8584	0.9324	0.9391
11 %	0.7764	0.9089	0.9231
13 %	0.7081	0.8726	0.9092
15 %	0.6533	0.7878	0.8981
17 %	0.6064	0.7229	0.8888
19 %	0.5694	0.6905	0.8722
21 %	0.5497	0.6594	0.8653
23 %	0.5247	0.6349	0.8509
25 %	0.4978	0.6132	0.8402
27 %	0.4800	0.5966	0.8136
29 %	0.4652	0.5746	0.7875
31 %	0.4587	0.5684	0.7811
33 %	0.4458	0.5678	0.7700
35 %	0.4320	0.5638	0.7280
37 %	0.4295	0.5443	0.7327
39 %	0.4263	0.5506	0.7245
41 %	0.4110	0.5305	0.7258
43 %	0.4042	0.5242	0.7168
45 %	0.4046	0.5222	0.6827
47 %	0.3943	0.5136	0.6543
49 %	0.3990	0.5057	0.6690

Table 2.2.: Comparison of the target accuracy of the algorithms for different noise levels.

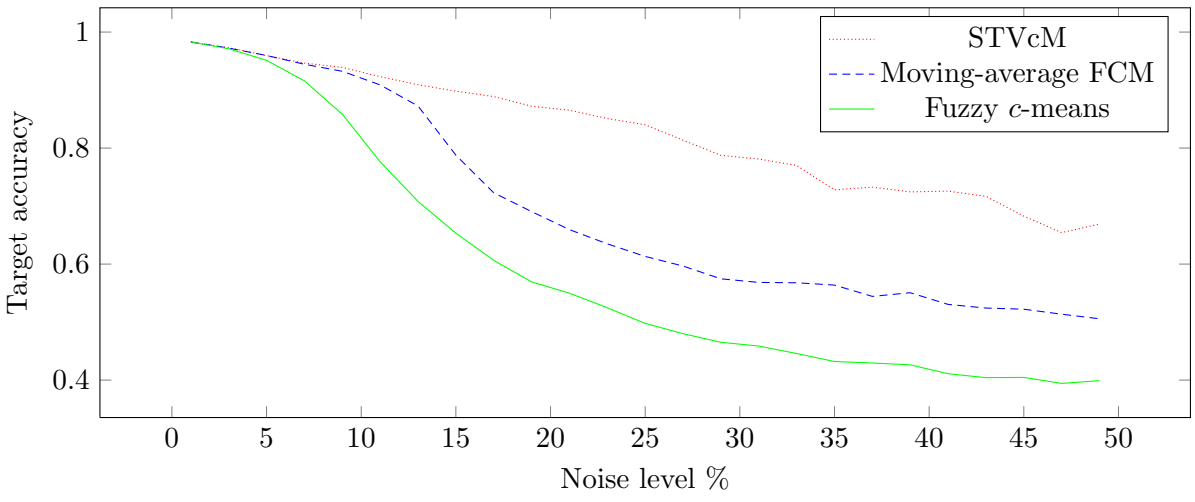
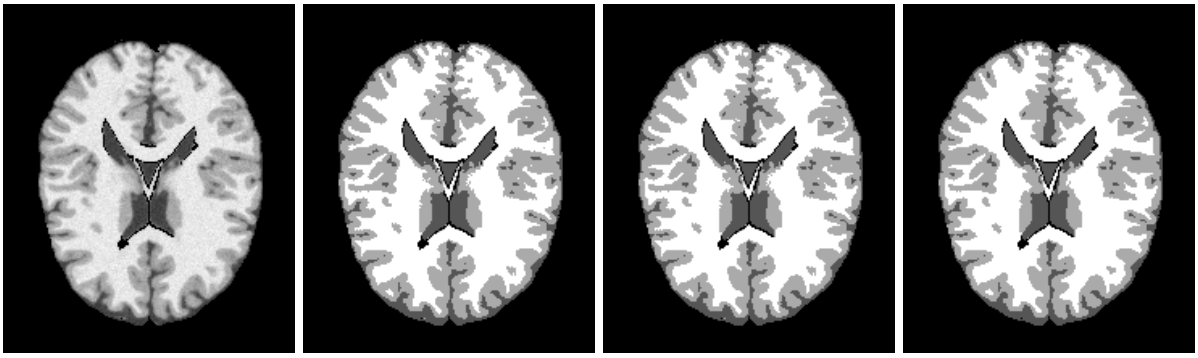
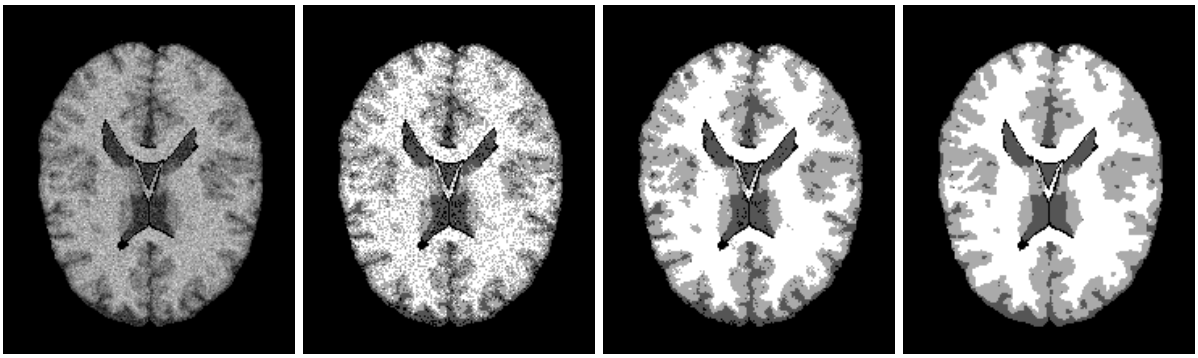


Figure 2.13.: Target accuracy for distinct noise levels of the three compared algorithms.



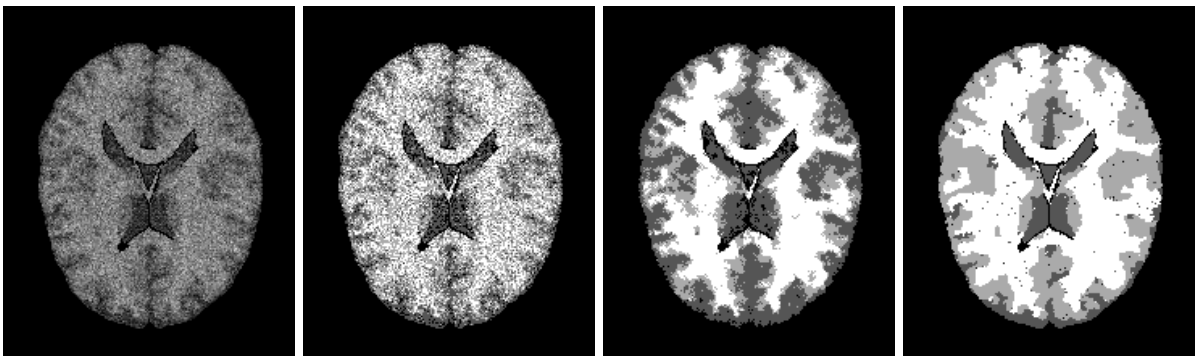
(a) Input (b) Fuzzy c-means, TA = 0.9712 (c) Moving-average FCM, TA = 0.9724 (d) STVcM, TA = 0.9731

Figure 2.14.: Segmentation results for 3% noise.



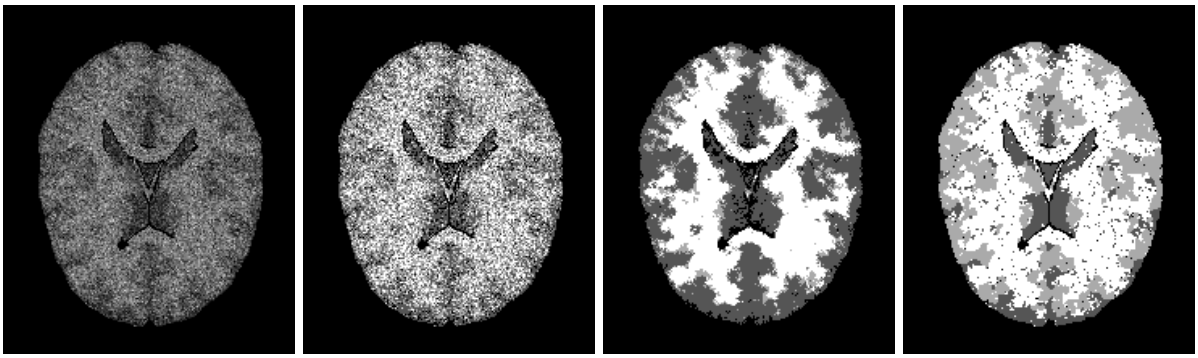
(a) Input (b) Fuzzy c-means, TA = 0.7081 (c) Moving-average FCM, TA = 0.8726 (d) STVcM, TA = 0.9092

Figure 2.15.: Segmentation results for 13% noise.



(a) Input (b) Fuzzy c-means, TA = 0.5247 (c) Moving-average FCM, TA = 0.6349 (d) STVcM, TA = 0.8509

Figure 2.16.: Segmentation results for 23% noise.



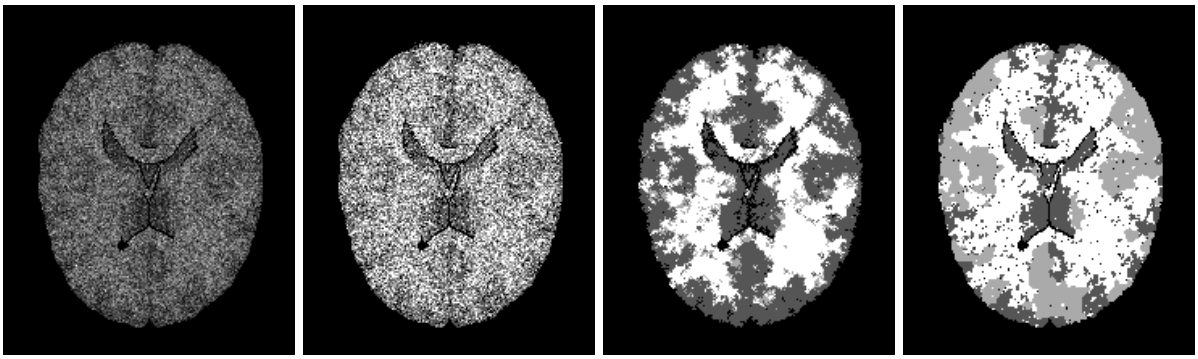
(a) Input

(b) Fuzzy c-means, TA = 0.4458

(c) Moving-average FCM, TA = 0.5678

(d) STVcM, TA = 0.77

Figure 2.17.: Segmentation results for 33% noise.



(a) Input

(b) Fuzzy c-means, TA = 0.4042

(c) Moving-average FCM, TA = 0.5242

(d) STVcM, TA = 0.7168

Figure 2.18.: Segmentation results for 43% noise.

2.5.2 Comparing TVcM and STVcM

In this section we show in practical applications for a large variety of different parameters and images that the performance of TVcM is comparable to STVcM [79]. The experiments contain some artificial images used in [79] as well as natural images.

As third method the alternate convex search (ACS) applied to (TVS) is added to the comparison, see [69] and Remark 2.10. More precisely, ACS is given by Algorithm 8 without the stopping condition

$$E_{\mathrm{TV}}(w^{(l+1)}, r^{(n+1)}) + \delta \leq E_{\mathrm{TV}}(u^{(n)}, r^{(n+1)}) = E_{\mathrm{old}}$$

in the inner loop. This means ACS repeats the inner iterations until convergence in every case.

Further, we show some examples where STVcM fails to converge if the parameters γ and λ are not suitable. Contrary to λ, the parameter γ does not influence the model. However, the convergence speed of ADMM depends on γ and one can end up in different partial minimizers for different values of γ.

As a stopping criterion we choose $\left\|u^{k+1} - u^k\right\|_2 / \left\|u^{k+1}\right\|_2 < 10^{-4}$ for inner and outer iterations as often used in practice (e.g., [39, 79]). The parameter $\delta = 0.001$ steers the amount of demanded decrease of E_{TV} in TVcM.

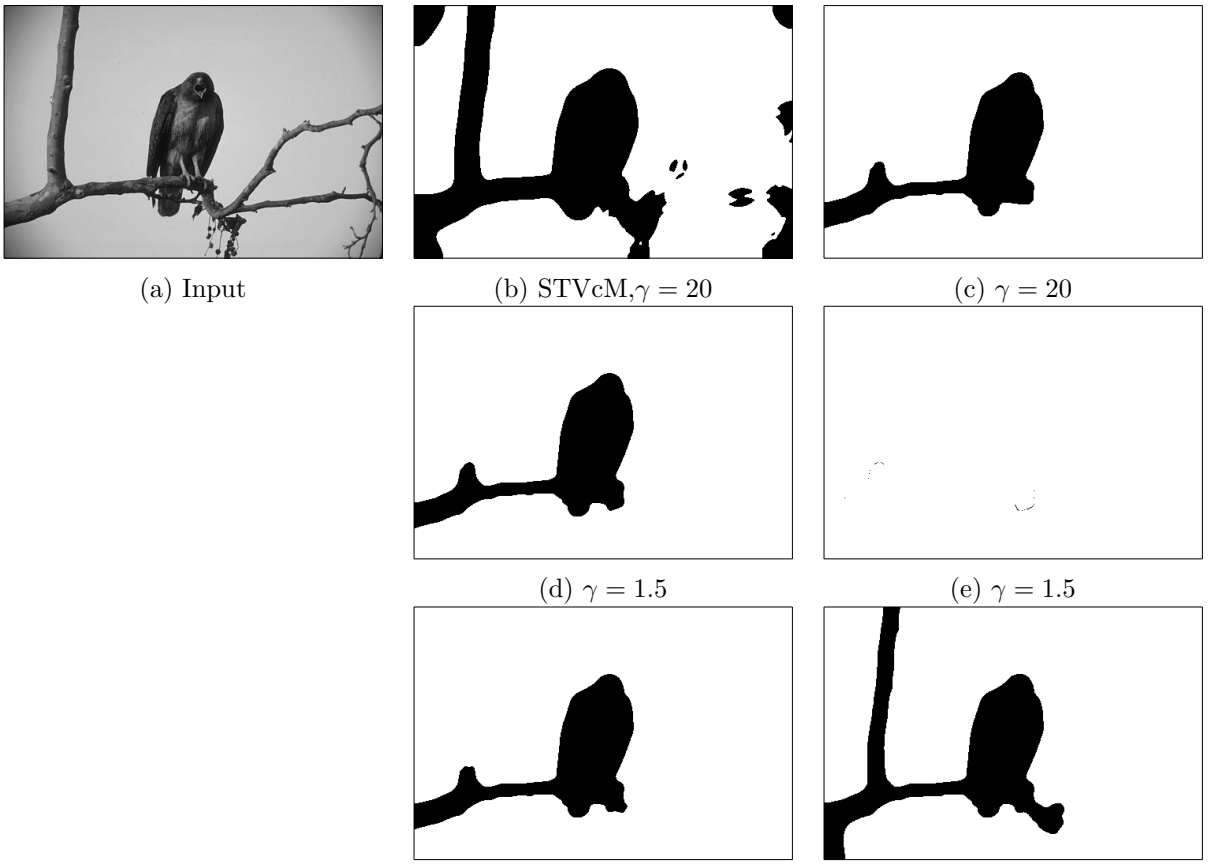
Segmentation of a Natural Bird Image Figure 2.19 shows segmentations for different values of the algorithm parameter γ.

For $\gamma = 20$ and $\lambda = 1$, STVcM does not converge and provides results with artifacts (Figure 2.19b). In fact, the artifacts change depending at which iteration the algorithm is stopped. The objective values of the *inner iterates*[4] are plotted in Figure 2.20a. We have stopped STVcM after 3000 inner iterations without convergence.

For $\gamma = 1.5$ and $\lambda = 1$, ACS converges faster at the beginning as depicted in Figure 2.20b. Nevertheless, ACS needs longer to obtain a partial minimizer. This is also clearly visible in the segmentation result after 100 iterations. The segmentation difference between STVcM and TVcM (Figure 2.19f shows only the result of TVcM) is only 3 pixels and they are both close to the converged result in Figure 2.19d. On the other hand, the ACS segmentation after 100 iterations provides a segmentation (Figure 2.19g) that is still far away from the converged result. The segmentation results after convergence, however, are nearly identical as depicted in Figures 2.19d and 2.19e.

In Figure 2.21 a comparison of the execution time for $\gamma = 1.5$ and $\lambda = 1$ is shown. The time of the algorithm is averaged over 10 executions and the iteration column states the number of inner iterations. The small speed difference between STVcM and TVcM is due to the fact that TVcM needs the evaluation of the energy functional in every iteration. If one evaluates the energy functional anyway, e.g., to use it as stopping criterion, STVcM will lose its little speed advantage completely as long as the number of iterations is identical. Table 2.21b shows the results after 100 iterations. This might be relevant for practical applications where one does not have the time to wait until the algorithm converges.

[4] Since we consider the values at the inner iterations, the convergence of ACS and TVcM does not look monotonic in all plots of of this section. To obtain monotonic plots, one has to consider the sequence of objective values only at *outer iterates* given by $(E_{\mathrm{TV}}(u^{(n)}))_n$.



(a) Input (b) STVcM,$\gamma = 20$ (c) $\gamma = 20$

(d) $\gamma = 1.5$ (e) $\gamma = 1.5$

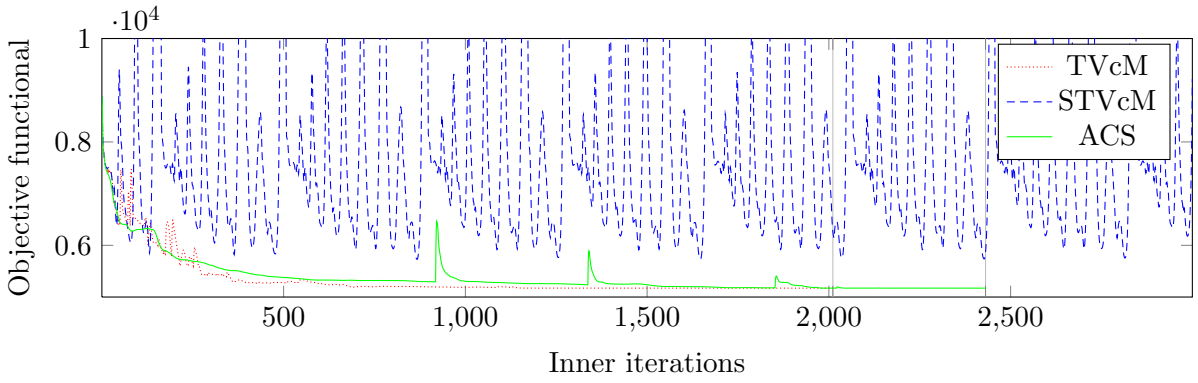
(f) $\gamma = 1.5$ (g) $\gamma = 1.5$

Figure 2.19.: Segmentation of a bird taken from [107] with $\lambda = 1$. a) Input of size 481×321 pixels. b) Result of STVcM after 3000 iterations. c) Result of TVcM after convergence. d) Result of STVcM and TVcM after convergence. e) Difference between the result of ACS after convergence and (d). f) TVcM after 100 iteration. g) ACS after 100 iterations.

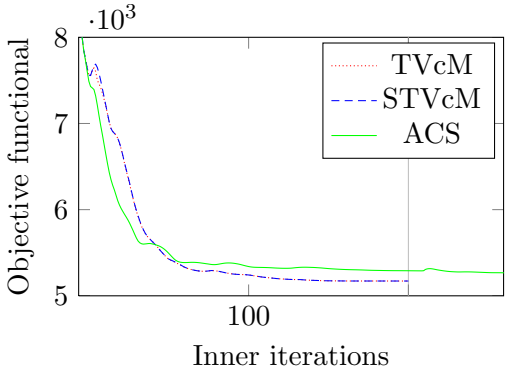
Figures 2.22 and 2.23 illustrate the comparison for other values of the regularization parameter λ. We use $\gamma = 1.5$ as the most efficient parameter. In both figures we show the two cases which are most relevant for practical applications. For $\lambda = 0.5$, TVcM and STVcM are more efficient compared to ACS. Figure 2.22a shows results after convergence and Figure 2.22b shows results after 100 iterations.

Analogous tables are shown in Figures 2.23a and 2.23b for $\lambda = 1.5$. Here, ACS performs better than STVcM after 100 iteration. In fact, for none of the algorithms 100 iterations are sufficient to obtain a solution close to the optimum. In the corresponding convergence plot in Figure 2.20c one can see that ACS has the highest convergence speed at the beginning and needs significantly more iterations than TVcM and STVcM to fulfill the convergence criterion. TVcM performs in this example as the best compromise.

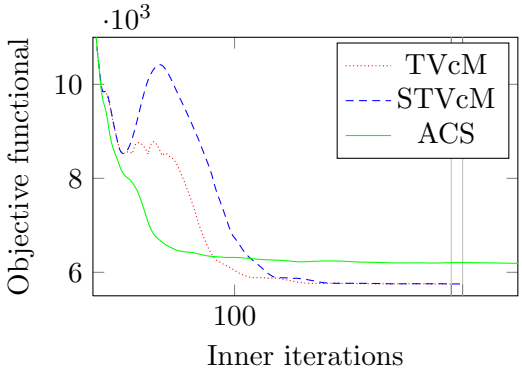
For certain examples (e.g. $\lambda = 1.5$) a speed up of TVcM or STVcM can be obtained by forcing a few iterations (e.g. 10) of the ADMM when updating u for fixed r. The idea is to exploit the



(a) Objective functional for $\gamma = 20$ and $\lambda = 1$. TVcM and ACS converge after 2011 and 2431 iterations.



(b) Objective for $\gamma = 1.5$ and $\lambda = 1$. TVcM and STVcM converge after 194 iterations while ACS needs 826 iterations.



(c) Objective for $\gamma = 1.5$ and $\lambda = 0.5$. TVcM and STVcM converge after 253 and 261 iterations, respectively, while ACS needs 523 iterations.

Figure 2.20.: Objective functionals with Fig. 2.19a as input for different values of γ and λ. Vertical bars mark the iterates that fulfill the convergence criterion.

algorithm	objective	iterations	time [s]
TVcM	5171.69	194	21.4276
STVcM	5171.69	194	20.4516
ACS	5171.65	826	82.9734

(a) Iteration until convergence, $\gamma = 1.5$.

algorithm	objective	iterations	time [s]
TVcM	5242.16	100	11.9202
STVcM	5241.73	100	10.7301
ACS	5339.74	100	10.1917

(b) 100 iterations, $\gamma = 1.5$.

Figure 2.21.: Results for the segmentation of Figure 2.19a with $\lambda = 1$.

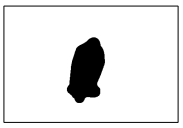
algorithm	obj.	it.	segmentation
TVcM	3803.87	153	
STVcM	3803.86	154	
ACS	3803.71	511	

(a) Iteration until convergence.

algorithm	obj.	it.	segmentation
TVcM	3805.24	100	
STVcM	3805.25	100	
ACS	3826.35	100	

(b) 100 iterations.

Figure 2.22.: Results for the segmentation of Figure 2.19a with $\gamma = 1.5, \lambda = 0.5$.

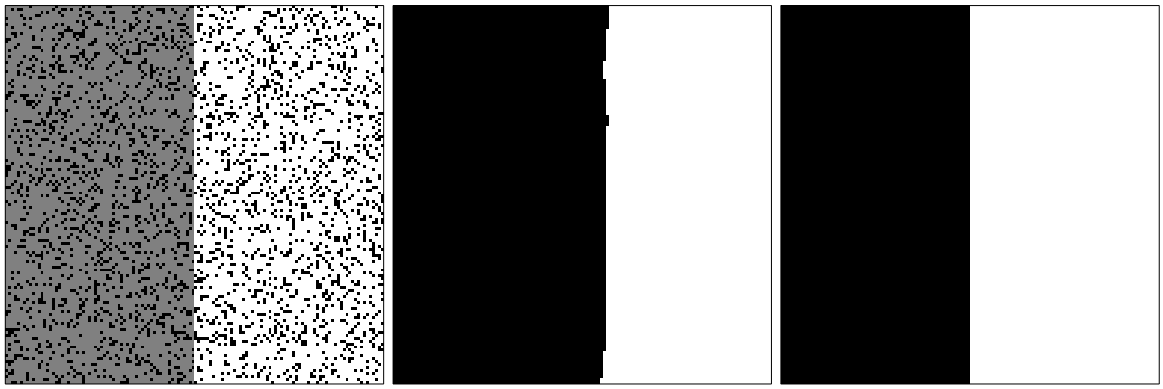
algorithm	obj.	it.	segmentation
TVcM	5752.27	253	
STVcM	5752.28	261	
ACS	5752.13	523	

(a) Iteration until convergence.

algorithm	obj.	it.	segmentation
TVcM	6042.92	100	
STVcM	6734.05	100	
ACS	6316.58	100	

(b) 100 iterations.

Figure 2.23.: Results for the segmentation of Figure 2.19a with $\gamma = 1.5, \lambda = 1.5$.



(a) Input (b) (c)

Figure 2.24.: Segmentation of a synthetic image with 20% data missing from [79]. $\gamma = 1.5$, $\lambda = 3$. a) Input image with 20% data missing. b) Result of STVcM. c) Result of TVcM and ACS.

fast convergence of ADMM indicated by the plots of ACS at the beginning (Figures and 2.20b and 2.20c). However, for other examples (e.g. $\lambda = 0.5$) this enlarged the number of iterations until convergence.

Segmentation of a Synthetic Two-Strip Image Figure 2.24 shows the segmentation of a synthetic image with missing data. For the chosen parameters $\lambda = 3$ and $\gamma = 1.5$, STVcM provides a bad result (note that the black area is significantly larger than the white area) while TVcM and ACS reconstruct the original image perfectly. The reason is that STVcM does not converge as is shown in Figure 2.25. A similar behavior can be observed for $\gamma = 0.25$ (Figure 2.26). Of course, also STVcM is able to reconstruct the image perfectly, e.g., for $\gamma = 0.5$, see Tables 2.27a and 2.27b. Since $1.5 > 0.5 > 0.25$, we see that STVcM can also fail if γ is chosen too small instead of too large. As in most cases of this section, STVcM and TVcM perform on the same level while ACS is clearly outperformed when one has found good parameters as shown in Table 2.27b.

Segmentation of Phantom Data In this section we show results of a simulated 4-class image. As described in the first paragraph of this section, we took a sample of the phantom data with 7% noise from [47, 94, 95, 46] also used in [79]. The segmentation results with the best parameters $\gamma = 300$ and $\lambda = 0.005$ after 40 iterations are shown in Figure 2.28. The corresponding Tables are shown in Figure 2.29. Table 2.29a reveals that STVcM needs significantly less iterations compared to TVcM. However, if we stop the algorithms after 40 iterations, for both STVcM and TVcM the objective is quite close to the optimal value obtained after convergence. In fact, 25 iterations would suffice as visible in Figure 2.30a where the objective values of TVcM and STVcM both are close to their partial minimum. ACS is outperformed by both STVcM and TVcM.

Figure 2.30b plots the objective values for $\gamma = 2000$ and $\lambda = 0.005$. In this case STVcM does not converge in contrast to the other methods.

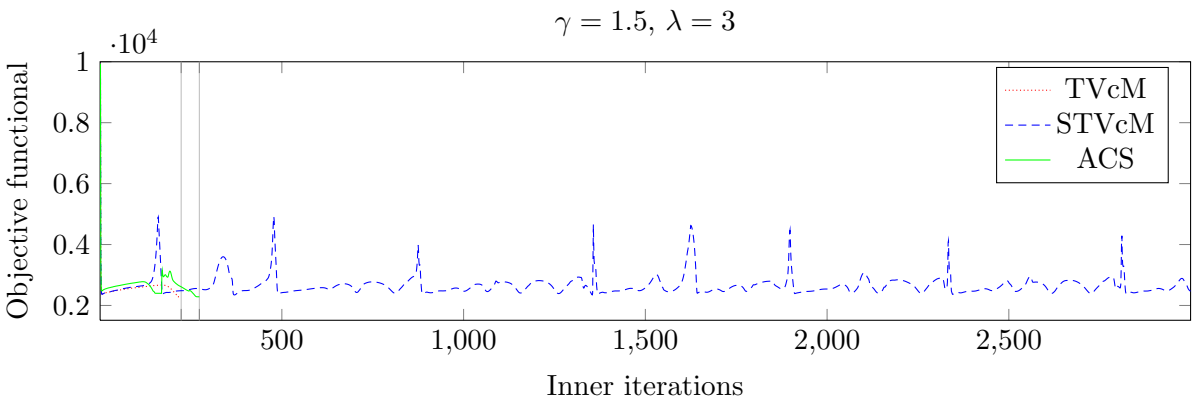


Figure 2.25.: Objective functional with Figure 2.24a as input. TVcM and ACS converge after 223 and 273 iterations, respectively (marked by vertical bars). STVcM is stopped without convergence after 3000 iterations.

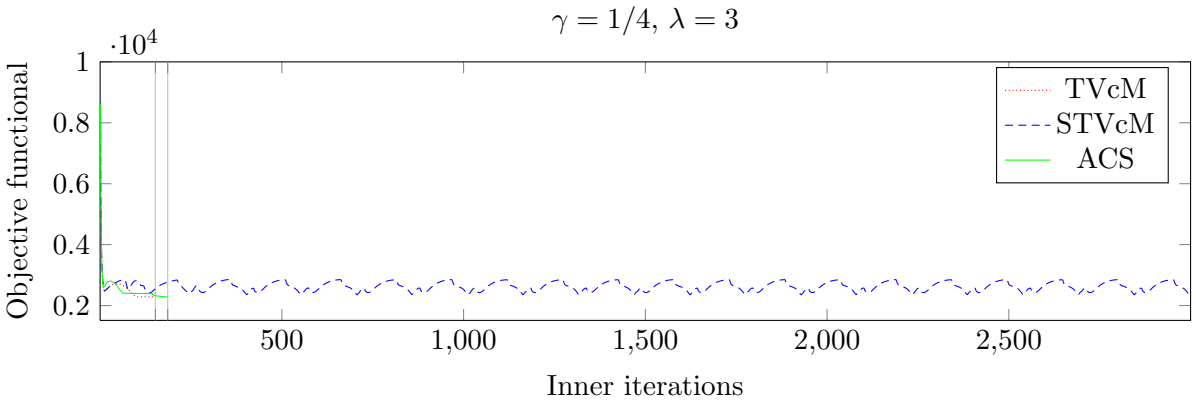


Figure 2.26.: Objective functional with Figure 2.24a as input. TVcM and ACS converge after 152 and 186 iterations, respectively (marked by vertical bars). STVcM is stopped without convergence after 3000 iterations

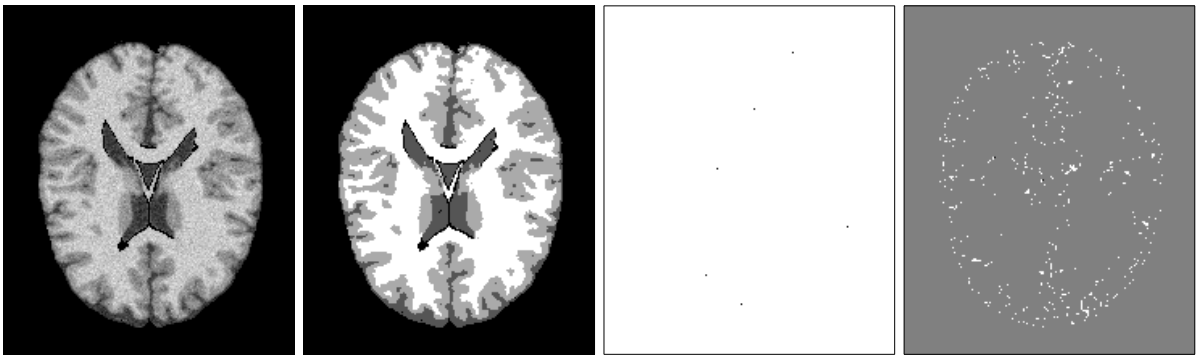
algorithm	obj.	it.
TVcM	2285.02	131
STVcM	2285.02	494
ACS	2285.02	197

(a) $\gamma = 0.5$, $\lambda = 3$

algorithm	obj.	it.	time [s]	γ
TVcM	2029.02	108	1.04551	1
STVcM	2029.02	118	1.07204	0.5
ACS	2029.02	173	1.497	1

(b) γ see table, $\lambda = 2$

Figure 2.27.: Results for the segmentation of Figure 2.24a. Iteration until convergence.



(a) Input (b) (c) (d)

Figure 2.28.: Segmentation of phantom brain tissue with 7% noise. $\gamma = 300$, $\lambda = 0.005$, 40 iterations. a) Input image with 7% noise. b) Result of STVcM. c) Difference between the results of STVcM and TVcM. d) Difference between the results of STVcM and ACS.

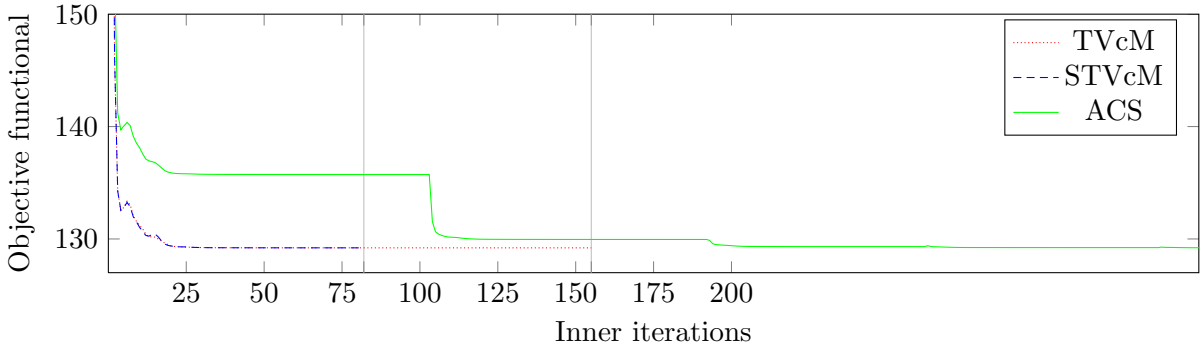
algorithm	obj.	it.
TVcM	129.21	155
STVcM	129.211	82
ACS	129.21	733

(a) It. until convergence.

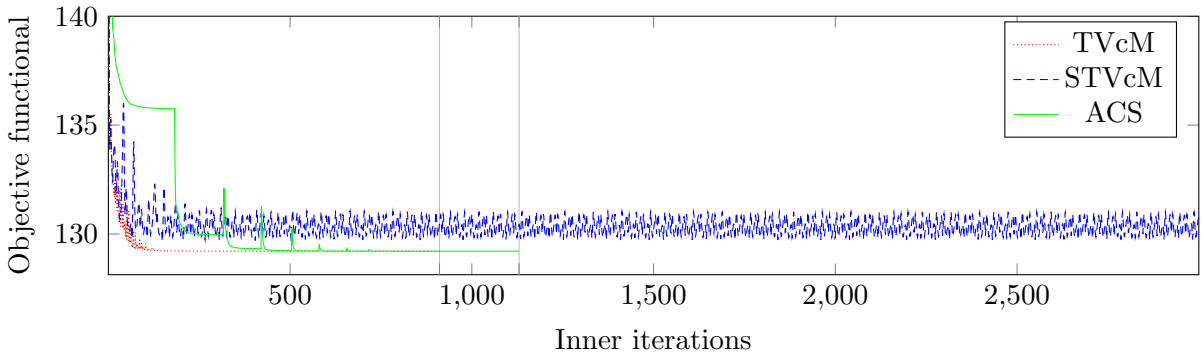
algorithm	obj.	it.	time [s]
TVcM	129.217	40	2.31643
STVcM	129.218	40	2.0823
ACS	135.74	40	2.02225

(b) 40 iterations.

Figure 2.29.: Results for the segmentation of Figure 2.28a. $\gamma = 300$, $\lambda = 0.005$.



(a) Objective for $\gamma = 300$ and $\lambda = 0.005$. TVcM and STVcM converge after 155 and 82 iterations, respectively, while ACS needs 733 iterations.



(b) Objective functional for $\gamma = 2000$ and $\lambda = 0.005$. TVcM and ACS converge after 911 and 1130 iterations, respectively. STVcM is stopped without convergence after 3000 iterations.

Figure 2.30.: Objective with Figure 2.28a as input for different values of λ and γ. Vertical bars mark the iterates that fulfill the convergence criterion.

2.6 Real World Applications in Materials Science

2.6.1 3d Carbon Fiber Reinforced Silicon Carbide

This subsection, based on our publication [140], copes with the segmentation of 2d and 3d image data of carbon fiber reinforced silicon carbide (C/SiC) ceramics acquired by synchrotron micro-tomography (μCT). Since C/SiC can withstand extremely high temperatures and is tough with respect to fractures, these lightweight long lasting materials are used, for instance, in jet engines, thermal protection systems of space-crafts, brakes in passenger cars, friction bearings and lightweight gears.

To produce C/SiC, a porous carbon preform is reinforced with bundled carbon fibers and infiltrated with liquid silicon at about 1500° C. While the liquid silicon vanishes a layer of silicon carbide emerges due to the chemical reaction between silicon and carbon. Due to this production process, a layer of silicon carbide is separating the porous carbon preform from the bundles of carbon fiber everywhere. The production process is time consuming and costly. Segmentation of the corresponding image data is a crucial step to monitor the quality of the material.

The segmentation of the C/SiC-ceramics data is a challenging task. Simple thresholding techniques fail due to the facts that different segments can still contain similar gray values and the imaging process can cause the appearance of ring artifacts. Further, the separation by the SiC-layer is not clearly visible everywhere. Figure 2.31 shows a part of one slice of the μCT data of C/SiC ceramics acquired at the BAMline BESSY II, Helmholtz Zentrum, Berlin.

In this subsection we do not use TVcM. Instead, we compute a codebook of segment prototypes or centroids r with fuzzy c-means as stated in Algorithm 3. To obtain the segment membership, we fix r and apply ADMM (Algorithm 5). Note that we did not observe significant improvements with alternating minimization for these examples. Further, we describe a rather practical approach respecting the layer structure and we describe the technicalities that are necessary to obtain a satisfactory solution of this real world problem.

In the literature, there exist different models for the separating layer problem. Lellmann et al. [97] include a distance between labels into the regularizer. The results of our method proposed in the following paragraphs is compared to this related approach of [97] in the following experiments. Further away from our approach, separation of layers can also be stated as overlapping layers [40, 85]. Another method in [9] works with a continuous max-flow/min-cut formulation and for an anisotropic total variation approach using directed graph cut algorithms, see [50].

Inclusion of Separating Layers Due to historical reasons, we use the slightly distinct regularizer of [98] instead of TV, namely

$$\mathrm{TV}_c := \| \, |(I_c \otimes \boldsymbol{D})u| \, \|_1 = \sum_{j=1}^{N} |(I_c \otimes \boldsymbol{D})u|(j) = \sum_{j=1}^{N} \sqrt{|\boldsymbol{D}u_1(j)|^2 + \ldots + |\boldsymbol{D}u_c(j)|^2}. \quad (2.46)$$

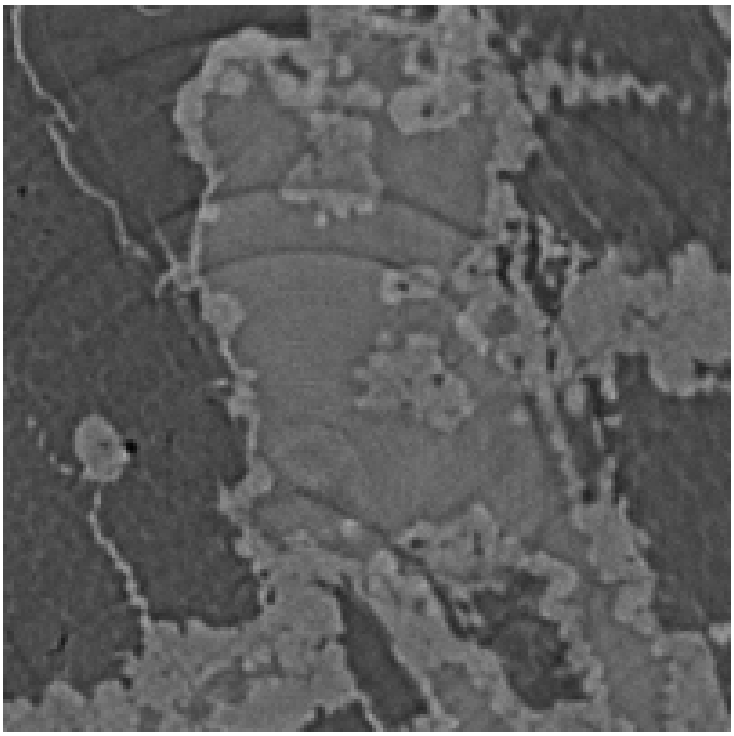


Figure 2.31.: Part of one slice of 3d micro-computed tomography data of C/SiC-ceramics [124]. Due to the imaging process there appear ring artifacts and similar gray values make it very difficult to distinguish between the different layers.

However, the differences between TV and TV_c in the results of our experiments are negligible. Our optimization problem reads

$$\text{(TVSfix)} \qquad \operatorname*{argmin}_u E_{\mathrm{FIX}}(u) \quad \text{subject to} \quad u \in S_c^N$$
$$\text{where } E_{\mathrm{FIX}} := \sum_{k=1}^{c} \sum_{j=1}^{N} u_k(j) s_k(j) \; + \; \lambda \, \mathrm{TV}_c(u)$$

and $s_k(j)$ is a constant describing some kind of distance between pixel j and segment k. A solution of (TVSfix) can be computed by applying ADMM (Algorithm 5). The resulting optimization method is quite close to Algorithm 6. Merely, the different regularizer yields the replacement of (2.42) by

$$v^{(n+1)} = \operatorname*{argmin}_{v \in \mathbb{R}^{2cN}} \left\{ \lambda \parallel |v| \parallel_1 + \frac{1}{2\gamma} \big|\big| v_k - \underbrace{(b_v^{(n)} + (I \otimes \boldsymbol{D}) u^{(n+1)})}_{=: g_v} \big|\big|_2^2 \right\}. \tag{2.47}$$

Of course, analogous changes have to be applied to step (2.34). Similar to the minimizer of (2.42), the solution of (2.47) can be computed in parallel for every image pixel j by the coupled

shrinkage of g_v with threshold $\lambda\gamma$ given by

$$v(j)^{(n+1)} = \begin{cases} 0 & \text{if } |g_v|(j) \leq \lambda\gamma, \\ g_v(j)\left(1 - \dfrac{\lambda\gamma}{|g_v|(j)}\right) & \text{otherwise,} \end{cases} \tag{2.48}$$

see, e.g., Chapter A.2 in the appendix or [138]. The difference to Algorithm 6 is that the coupled shrinkage is not computed for each k independently.

Let us turn to the problem of separating layers. The basic idea we pursue is to penalize the objective functional E_{FIX} by changing s such that pixels belonging to non-touching layers cannot be neighbors in our segmented image. For every a label $k \in \{1, \dots, c\}$ the binary distance function

$$b_k : \{1, \dots, c\} \to \{0, 1\}, \quad b_k(k') := \begin{cases} 1 & \text{if layer } k' \text{ must not touch layer } k, \\ 0 & \text{else} \end{cases}$$

determines which layers are allowed neighbors and which are not. Naturally, we always set $b_k(k) = 0$. With an initial segmentation $l_0 : \{1, \dots, N\} \to \{1, \dots, c\}$ we define the penalizing function P_k for $k = 1, \dots, c$ by

$$P_k(j) := \sum_{i \in \mathcal{N}(j)} b_k(l_0(i)), \quad j = 1, \dots, N$$

where $\mathcal{N}_j$ denotes the (four- or eight-) neighborhood of pixel j. Finally, we set s in our data term to

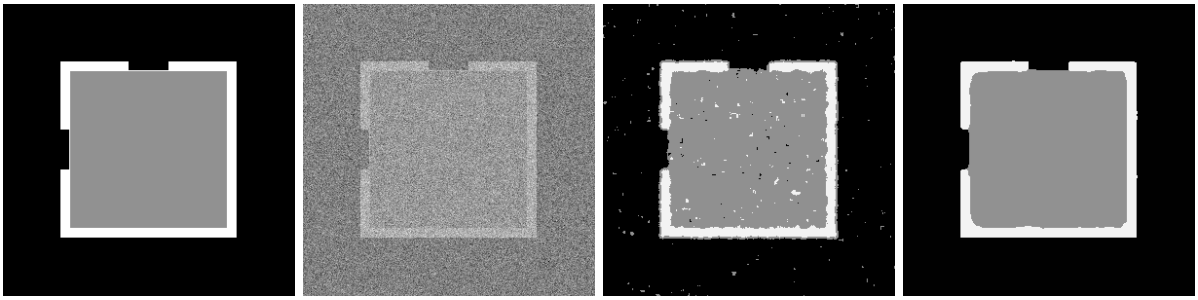
$$s_k(j) := |f(j) - r_k| + \mu P_k(j), \quad \mu \geq 0. \tag{2.49}$$

Summarizing, we apply a two-step approach:

1. For fixed r (computed with fuzzy c-means beforehand) and $s_k(j) := |r_k - f(j)|$ solve (TVSfix) with ADMM (Algorithm 5) as described above.

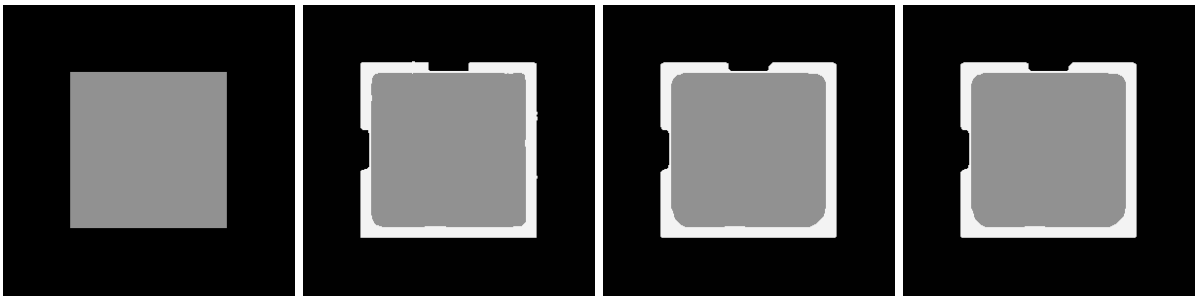
2. Change s according to the layer penalization (2.49) with the segmentation result of the previous step as initial segmentation l_0 and rerun ADMM for (TVSfix).

In the following, we present an artificial experiment where layer penalization improves the result. Firstly, let us consider the results without layer penalization in Figure 2.32. The ground truth is depicted in Figure 2.32a. It consists of three layers with gray values $r_{\text{gt}} = (0, 127, 255)$. The white part separates the black from the gray part of the image. Observe that the separating part is very thin at the notches. Of course, we also want to preserve the thin lines. We added white Gaussian noise of standard deviation 120 to the ground truth as shown in Figure 2.32b. To obtain a codebook we smoothed the input image by convolution with a Gaussian of standard deviation $\sigma = 5$ and applied fuzzy c-means (Algorithm 3) afterwards. For the resulting segmentation see Figure 2.32c. Although we applied a pre-smoothing step, the image is still noisy due to the lack of regularization. As corresponding codebook we obtain $r \approx (-1, 125, 240)$. This is used to construct s in (TVSfix). Figure 2.32d shows the result of the ADMM segmentation (Algorithm 5) with $r \approx (-1, 125, 240)$ obtained from the fuzzy c-means algorithm in the previous



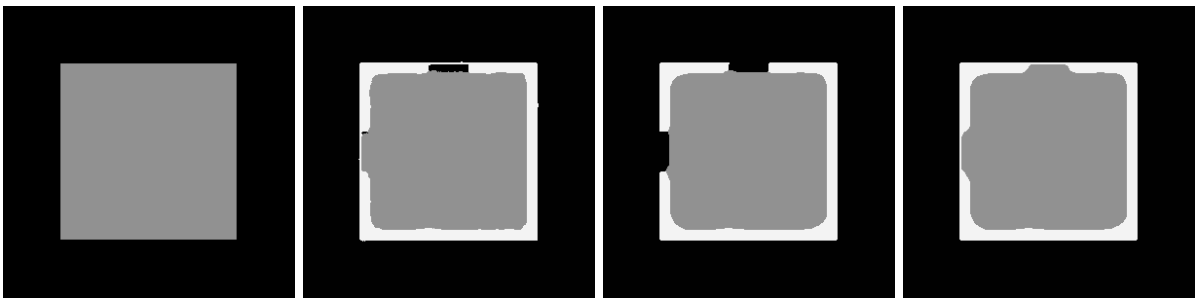
(a) (b) (c) (d)

Figure 2.32.: Results without layer penalization. a) Ground truth with white separating white lines which are thin at the notches. b) Noisy image with Gaussian noise of standard deviation $\sigma = 120$, scaled to $[0, 255]$ for visualization. c) Result of the fuzzy c-means algorithm applied to the noisy input after Gaussian filtering. d) Solution of Algorithm 6 applied to the noise input without layer penalization using $\lambda = 250$.



(a) Initial segmentation (b) $\lambda = 250, \mu = 1500$ (c) $\lambda = 500, \mu = 500$ (d) $\lambda = 500, \mu = 1500$

Figure 2.33.: The effect of layer penalization with respect to the inner square of Figure 2.32a as initial segmentation for different parameters λ and μ.



(a) Initial segmentation (b) $\lambda = 250, \mu = 1500$ (c) $\lambda = 500, \mu = 500$ (d) $\lambda = 500, \mu = 1500$

Figure 2.34.: The effect of layer penalization with respect to the outer square of Figure 2.32a as initial segmentation for different parameters λ and μ.

step. Contrary to Figure 2.32c, the noise disappears. However, the thin lines at the notches are not preserved.

In the following experiment, we show how the layer penalization (2.49) facilitates preservation of the thin lines at the notches. More precisely, we use

$$b_1(3) = b_3(1) = b_3(2) = b_2(3) = 0 \text{ and } b_1(2) = b_2(1) = 1 \tag{2.50}$$

where $\{j : l_0(j) = 1\}$, $\{j : l_0(j) = 2\}$, and $\{j : l_0(j) = 3\}$ denote the black, the gray, and the white segment, respectively. This implies that the black and the gray segment must not have a piece of common boundary.

For purpose of demonstration, we do not use a result of ADMM as initial segmentation l_0 in our two-step approach for the next experiment. Instead, in Figures 2.33 and 2.34 the inner square of Figure 2.32a as good initial segmentation and the outer square as bad initial segmentation, respectively, are compared. The result for the good initial segmentation is shown in Figure 2.33 for different parameters μ and λ. The pixels at the notches are penalized due to the choice the inner square as initial segmentation.

In contrast, the pixels at the notches are not penalized at all in Figure 2.34. Observe that on the boundary of the outer square we obtain only valid configurations in the resulting Figures 2.34b, 2.34c and 2.34d. That is, on the boundary of the outer square appears no black pixel with a gray neighbor or vice versa. Of course, all results still have roughly the shape of the image because the segmentation does not only depend on the penalization of the boundary but also on the original image data.

Remark 2.29 (Thickness of the Separating Layer)**.** *Figure 2.35 shows the left notch in large. Let us assume that k and k' are not allowed to be labels of neighboring pixels. From a theoretical point of view, it is enough to change only one label, say k', of invalid configurations in the initial segmentations. However, we penalize in our experiments both the pixels labeled with k neighboring a pixel labeled with k' and vice versa as depicted in Figure 2.35b yielding more reliable separations. Figure 2.35b is a part of our initial segmentation computed with ADMM, see Figure 2.32d. Of course, with this procedure one obtains too thick separating structures as visible in Figure 2.35c. Nevertheless, we resolve this inaccuracy by simply superimposing with the segment of the initial segmentation that corresponds to the darkest gray values in the input image, see Figure 2.35d. The choice of the superimposing segment depends on the particular application.*

Of course, one might argue that incorporating the distances of the labels into an one step optimization approach is more elegant. This has been done, for instance, in [97]. Lellmann et al. [97] consider a more general class of regularizers incorporating a weights matrix M and replace TV_c in (TVSfix) by

$$\mathrm{TV}_M(u) := \| \, |(M \otimes \boldsymbol{D})u| \, \|_1 \, . \tag{2.51}$$

For our applications, (2.51) can be used to modify the regularization corresponding to a distance $d(k, k')$ between labels k and k'. In the *Euclidean metric method* of [97] the weight matrix $M = (m_1|m_2|m_3)$ in the regularizer corresponds to the label distances $d(i, j) = \|m_i - m_j\|_2$. However, Figure 2.32d shows that the method of [97] is not able to separate the segments properly. The incorporation of the separation into the regularizer has the drawback that a higher regularization parameter λ tends to let the separating part vanish as shown in Figure 2.36. One problem may be the nontrivial parameter adjustment.

(a) (b) (c) (d)

Figure 2.35.: Extracts for $\lambda = 250$ and $\mu = 1500$. a) Ground truth. b) Penalized pixels of Figure 2.32d corresponding to (2.50) marked in red. c) Segmentation with layer penalization using (b) as initial segmentation. d) Result after superimposing with the corresponding segment of the initial segmentation.

Segmentation of C/SiC As mentioned in the beginning of this section, the C/SiC image data consists of a carbon fiber layer, a carbon preform layer and a silicon carbide layer which separates the carbon fiber layer from the carbon preform layer. The carbon fiber layer has the darkest gray, the carbon preform layer is of a medium gray and the separating silicon carbide has the lightest gray. Accordingly, the materials are represented in the segmented results by the labels 1, 2, and 3, respectively. There are artifacts in form of circles visible in the input images. These should be eliminated by the segmentation process. They do not really exist in the material but arise due to the μCT imaging method.

For the implementations we used Matlab and executed our experiments on an Intel Xeon 2.5 GHz processor. Further, we apply the superimposition step described in Remark 2.29 with the darkest segment to all our results of the penalized model. This corrects the thickness of the separating layer.

2d segmentation of a slice of the C/SiC image data In this experiment, we segment a 2d slice of the size 1251×1251 ignoring any 3d information. Therefore, the results of this paragraph are not relevant for the real application. Nevertheless, we show them for demonstration purposes.

Due to the small amount of noise it was not necessary to apply a smoothing filter before computing the codebook. We use as codebook the result of fuzzy c-means (Algorithm 3) given by $r \approx (71, 100, 120)$. The initial segmentation was the result of ADMM (Algorithm 5) for (TVSfix) with $s := |r_k - f(j)|$, only 5 iterations, and $\gamma = 0.01$. Figure 2.37 shows the corresponding results. ADMM without layer penalization eliminates the ring artifacts but cannot enforce the separation of the layers everywhere regardless of the number of iterations. To this end, invoking

(a) [97] (b) [97] (c) [97] (d) Our method

Figure 2.36.: Comparison between our method and the approach of [97] for different parameters. a) Method of [97] $\lambda = 40$, $M = (3e_1|3e_2|e_3)$. b) Method of [97] $\lambda = 80$, $M = (3e_1|3e_2|e_3)$. c) Method of [97] $\lambda = 80$, $M = (3e_1|3e_2|0.01e_3)$. d) Our method with Figure 2.32d as initial segmentation and $\lambda = 250$, $\mu = 1500$.

the layer penalization improves the result and again separates the layers better than the unified approach of [97]. We use 40 ADMM iterations for the layer penalization model. The overall algorithm including the computation time for the fuzzy c-means and the initial segmentation requires about 630 seconds, but the code can further be optimized. We are able to separate the carbon fibers from the carbon preform.

However, one has to admit that minimizing model (TVSfix) does not provide satisfactory results in the 2d case. The thickness of the SiC layer is far from the true solution. This changes significantly in the next paragraph that is relevant for the real application where we segment volumetric 3d image data.

Segmenting 3d Volume Data of C/SiC In this paragraph we deal with volumetric image data of size of $300 \times 300 \times 100$ voxels. The discrete 3d gradient and some 3d modifications required in the first step of the ADMM algorithm are described in Section A.1 in the appendix. Our codebook is obtained again by fuzzy c-means (Algorithm 3) which results in $r \approx (126, 151, 175)$. Figures 2.38 and 2.39 show our 3d segmentation results for 4 and 6 arbitrarily chosen subsequent slices in the x-y- and x-z-plane, respectively. As initial segmentation we use again ADMM results with 5 iterations and $\gamma := 0.1$ shown in the second rows of the figures. The markers in the figures indicate that not everywhere the separation property was fulfilled. With the layer penalization model we executed 30 ADMM iterations. The improvement is visible in the third row of the figures. The overall computation took approximately 1900 seconds. Increasing the number of iterations of the initial segmentation did not change our final results significantly.

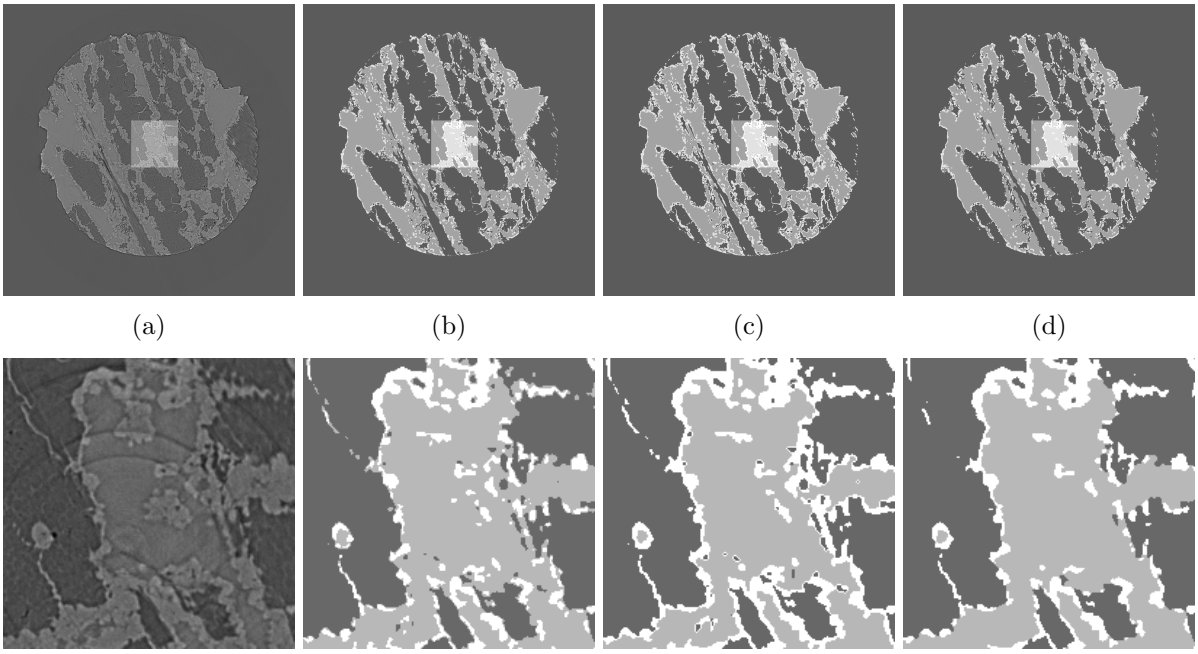


Figure 2.37.: 2d segmentation of a C/SiC slice. The first row depicts the whole image and its segmentations. The second row shows the marked areas of the first row in large. a) Input image. b) Initial segmentation with 5 ADMM iterations without layer penalization and $\lambda = 10$. c) Final segmentation with 40 ADMM iterations applied to our model with layer penalization and $\lambda = 10$, $\mu = 200$. d) 40 ADMM iterations, $\lambda = 5$ and weight matrix $M = (3e_1|3e_2|e_3)$ in the regularizer as proposed by [97] corresponding to the label distances $d(1,2) = \|3e_1 - 3e_2\| = \sqrt{18}$ and $d(1,3) = d(2,3) = \sqrt{10}$.

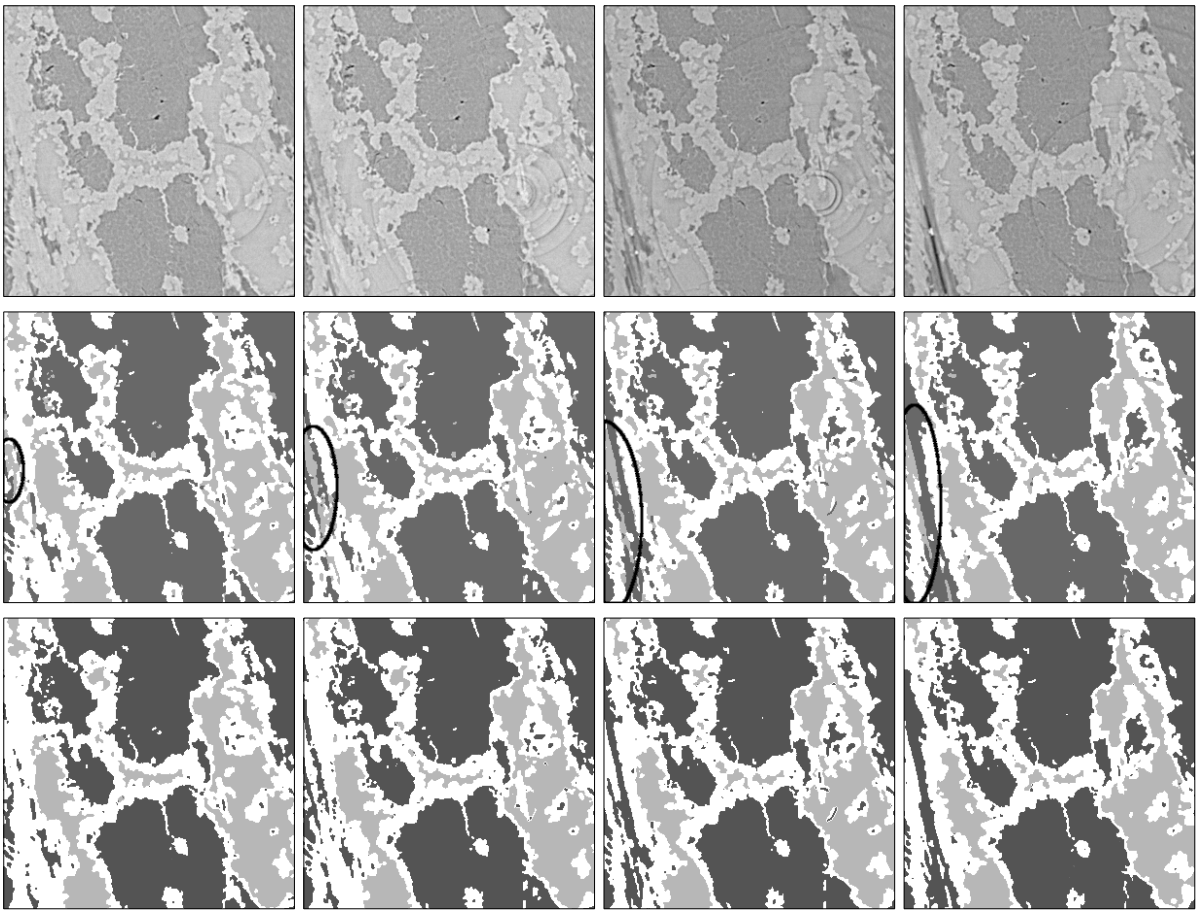


Figure 2.38.: First row: x-y-slices of 3d input image data. Second row: 3d initial segmentation with 5 ADMM iterations without layer penalization and $\lambda = 15$. Third row: Final segmentation with 30 ADMM iterations and $\lambda = 15$, $\mu = 500$.

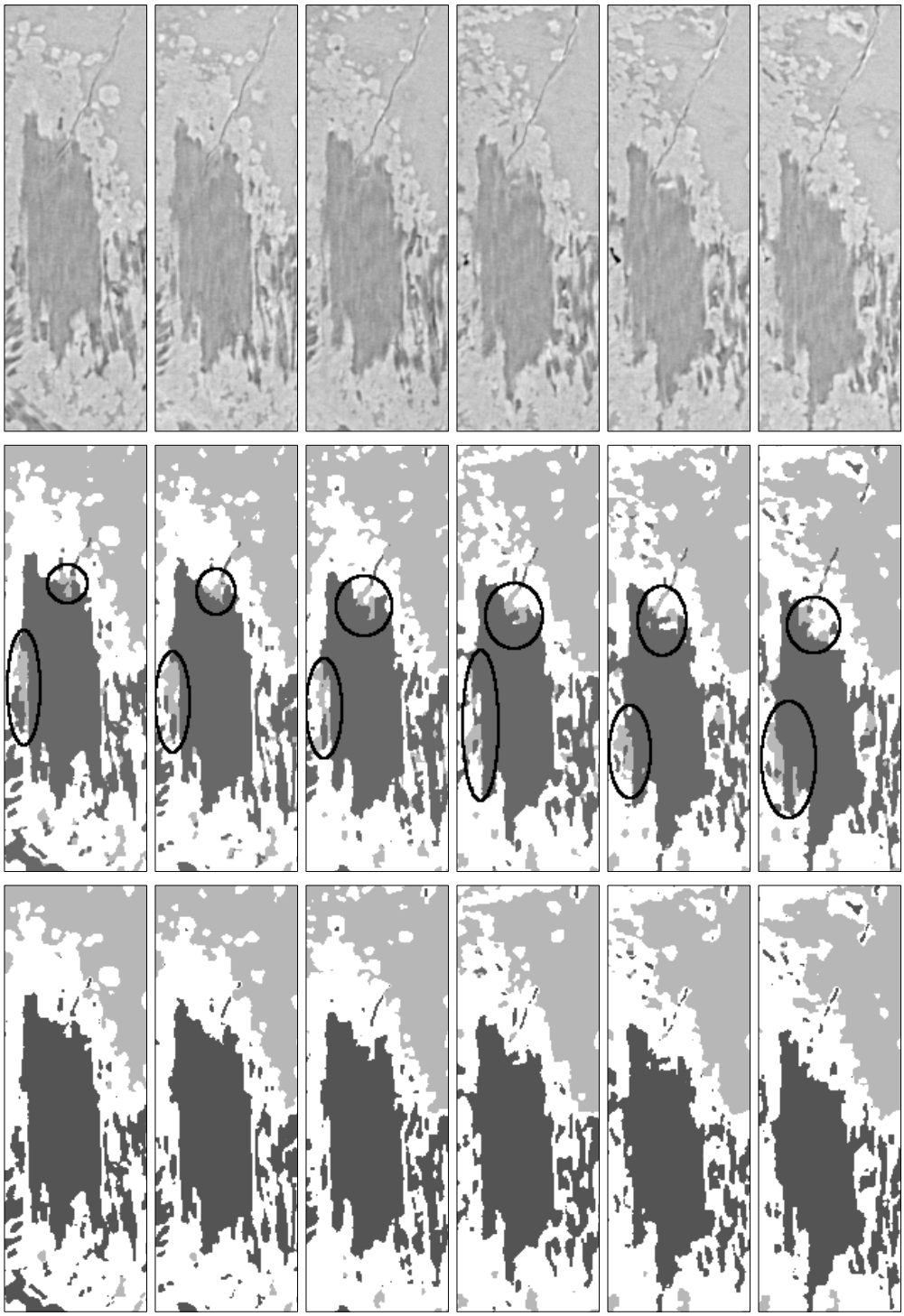
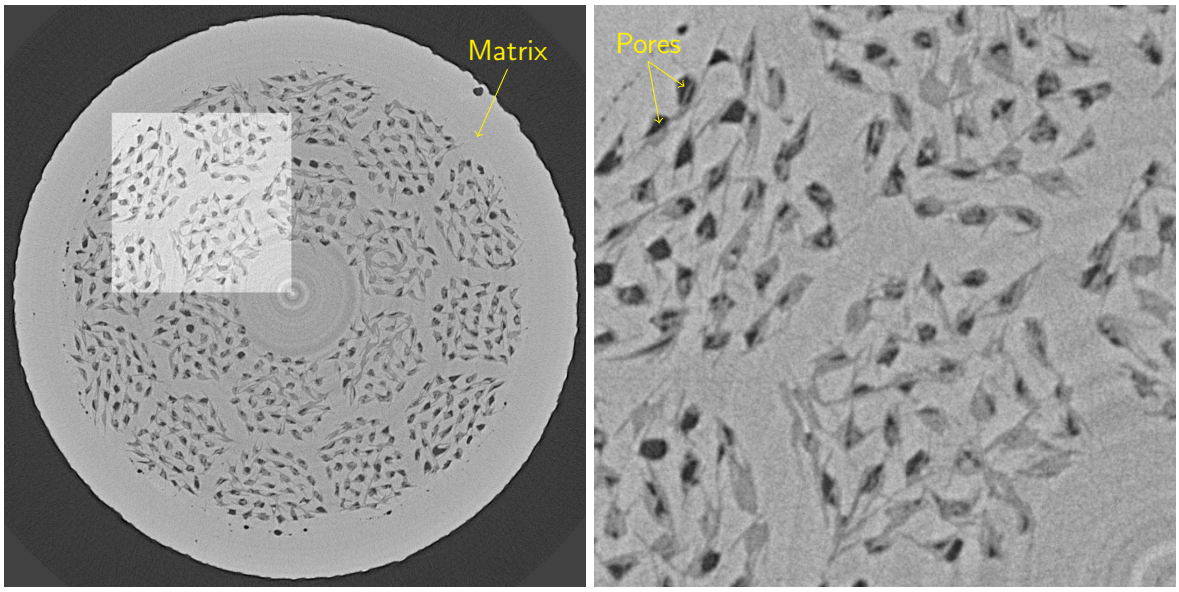


Figure 2.39.: x-z-slices of our 3d segmentation results as in Fig. 2.38.

2.6.2 Volumetric Data of a Multifilament Superconductor



(a) (b)

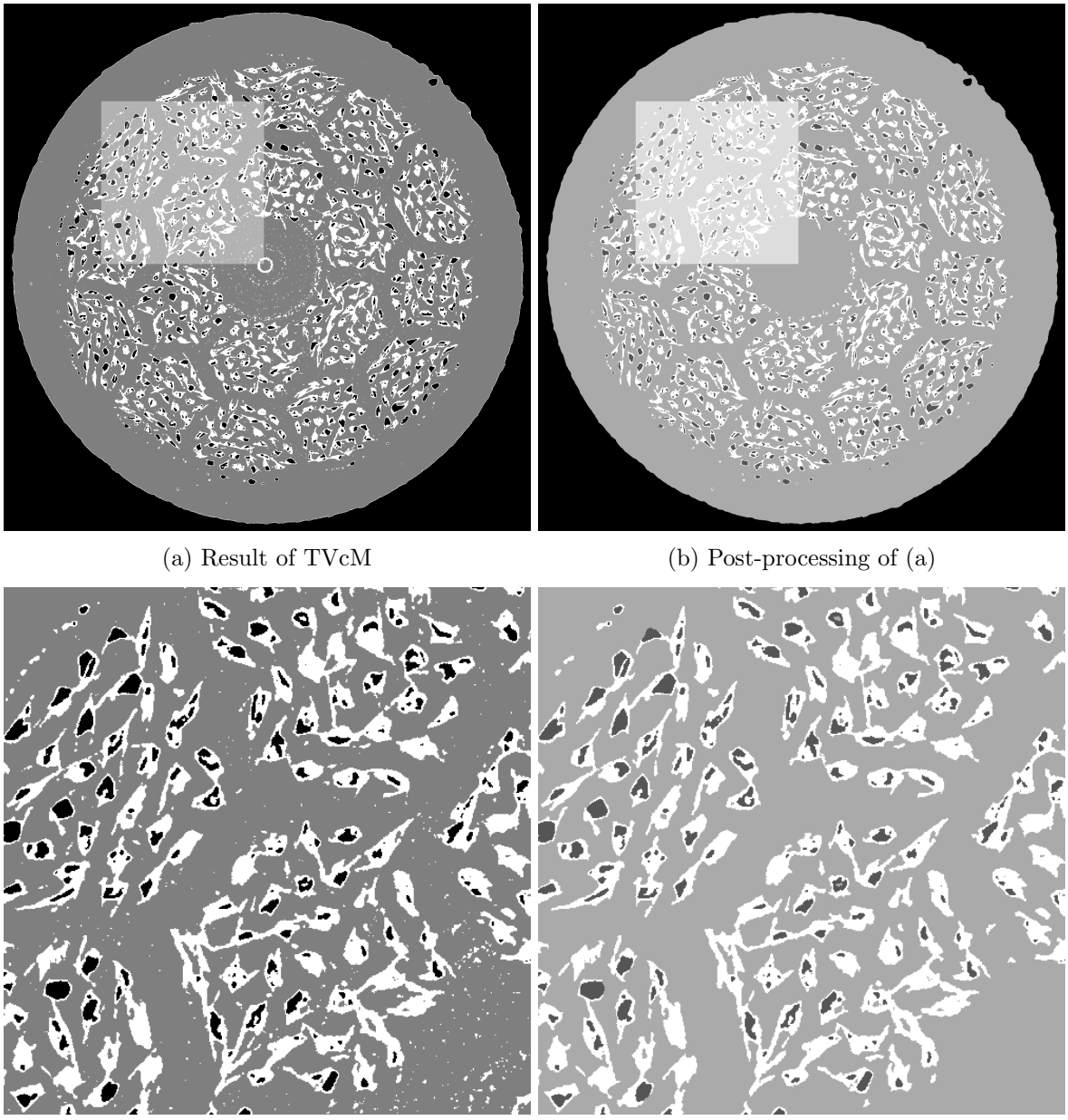
Figure 2.40.: One slice of multifilament composite superconductor data acquired with μCT at the European Synchrotron Radiation Facility (ESRF ID 19) in Grenoble, France. a) One complete slice of the volumetric data. b) Enlargement of the region marked in (a).

While the electrical resistance of usual metallic conductors, e.g., made of copper, gradually decreases in temperature, a superconductors' electrical resistance drops to zero precipitously when cooled below its critical temperature. Superconductivity was discovered in 1911 by the Dutch physicist Heike Kamerlingh Onnes and still is a topic of ongoing research. For an introduction to solid state physics and superconductors we refer to textbooks such as [8].

This chapter is concerned with the segmentation of 3d data showing a multifilament composite wire. A slice of this volumetric data is depicted in Figure 2.40. More details about the material can be found in the article of Scheuerlein et al. [133]. All specimens of this material were provided by the Applied Superconductivity Center of the National High Magnetic Field Laboratory, Florida State University, Tallahassee, USA.

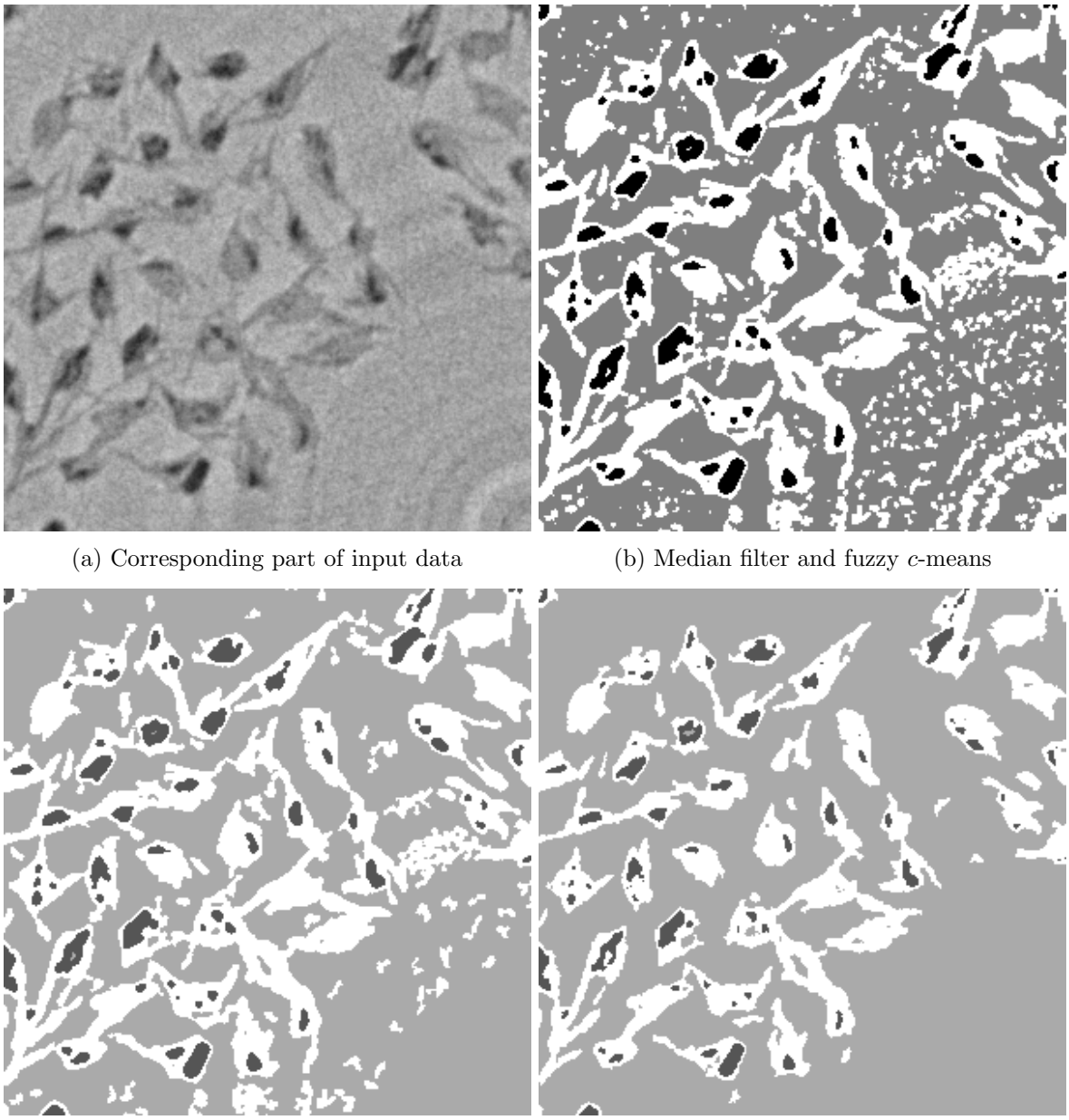
The whole data set of Figure 2.40 is of $1620 \times 1620 \times 2048$ voxels and consumes about 5.4GB disk space. To be able to keep the data in memory we divided the volume into slightly overlapping blocks. The overlap was introduced to avoid boundary artifacts in the computations of the discrete gradient. As block size we used $1620 \times 1620 \times (8 + 1)$ where the $+1$ indicates the overlap. We executed our C++ implementation of TVcM (Algorithm 8) with 4 parallel blocks on a compute server in about 11 hours. In this setting the random access memory consumption achieved its maximum at around 50GB.

The slice of the result corresponding to Figure 2.40 is shown in Figure 2.41. Figure 2.41a shows the 3-class segmentation with TVcM. Note that the inner pores have the same label as



(a) Result of TVcM (b) Post-processing of (a)

Figure 2.41.: Segmentation with TVcM. The first row shows the complete slice. The second row shows respective enlargements of the marked regions of the first row.



(a) Corresponding part of input data

(b) Median filter and fuzzy c-means

(c) Post-processing of (b)

(d) TVcM and post-processing

Figure 2.42.: Comparison between TVcM and fuzzy c-means after a median filter.

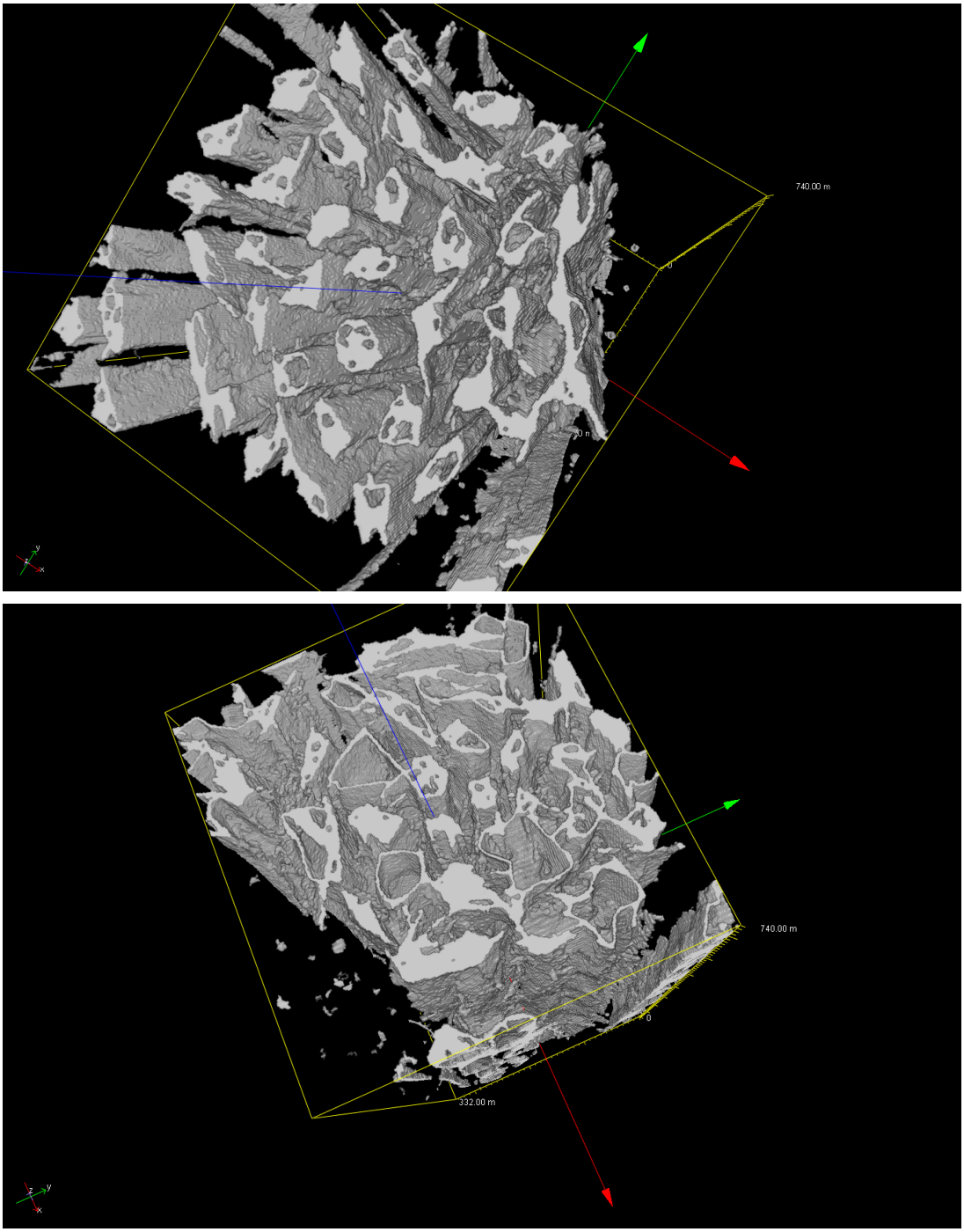


Figure 2.43.: Volume rendering of two different data sets. The first volume rendering corresponds to the data shown in Figures 2.40, 2.41, and 2.42. Slices for the second volume rendering are not shown.

the background. We changed this using the pixels' distance to the center of the image. Further, we still have isolated spots of noise in Figure 2.41a that can be reduced significantly in a post-processing step. Thereby, we erase all connected components below a specific size of the white segment.

Figure 2.42 shows a comparison of TVcM to a much simpler (and faster) approach. Figure 2.42b depicts the result of fuzzy c-means (Algorithm 3) applied to the median filtered[5] input data with filter size $3 \times 3 \times 3$. Figure 2.42c shows the result after the post-processing described above. The application of TVcM and the post-processing provides a segmentation with less noise and artifacts as shown in Figure 2.42d. On the other hand, more of the thin connecting parts of the white segment disappear after application of TVcM.

In Figure 2.43 we see volume renderings of two different data sets of this material. The volume renderings show the segment separating the pores from the matrix. To create the volume renderings from the segmented data sets we used the software MAVI [61].

[5]The result of applying the median filter to an image $f : \{1, \dots, N\} \to \mathbb{R}$ is the mapping from $\{1, \dots, N\}$ to $\mathbb{R}$ defined by $j \mapsto \text{median}\{f(i) : i \in \mathcal{N}_j\}$ where $\mathcal{N}_j$ denotes a neighborhood of j. In this application we use a block of size $3 \times 3 \times 3$ centered in j as neighborhood $\mathcal{N}_j$.

Chapter 3

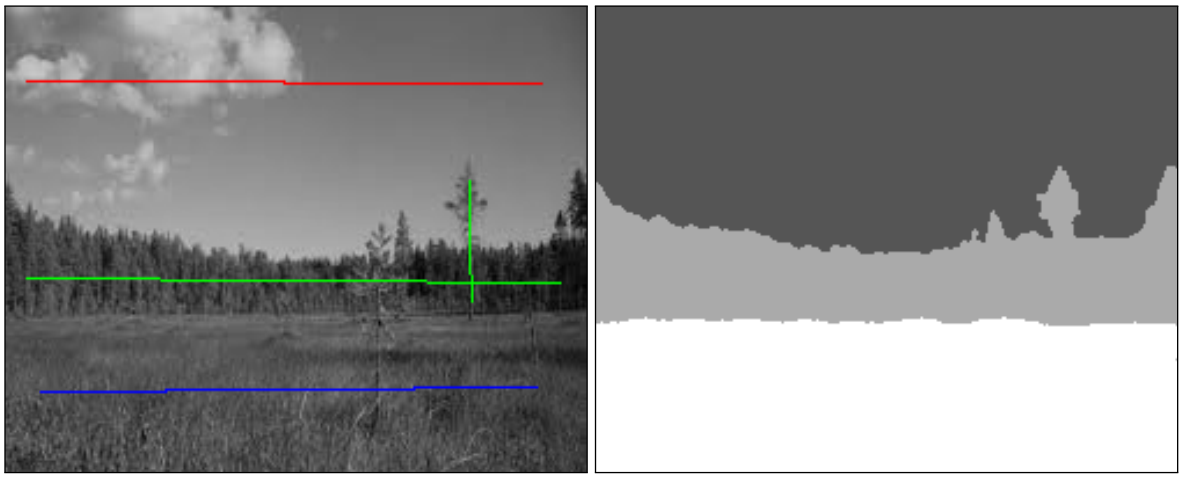
Supervised Segmentation

This chapter is based on our publication [88]. The aim of image segmentation is to partition all image pixels into meaningful non-overlapping segments (also called phases) of the image. In contrast to unsupervised segmentation handled in Chapter 2, in supervised segmentation we are given a labeled subset of image pixels indicating the segment each pixel belongs to. The task is to determine the segment membership of the unlabeled pixels as shown in Figure 3.1.

Further, let us try to clarify the term *transductive* which describes a certain property of supervised problems. As in Chapter 2, all our segmentation models consist of solving an optimization problem. Let all labeled or unlabeled pixels be involved in the optimization that determines the segmentation. Additionally, assume that the complete model has to be optimized again if we want to determine the segment membership of an arbitrary number of new unlabeled pixels added to the previously segmented image. Then, we call such a supervised segmentation model *transductive*. On the other hand, if our minimizer can be used to determine the segment membership of possibly added unknown pixels without re-solving our optimization problem, we call the approach *inductive*. In other words, optimizing an *inductive* model generates some kind of rule that determines the segment membership of unknown pixels, e.g., in a new image. For an *inductive* model it is not important whether the unlabeled pixels are involved in the optimization process or not. In the literature, inductive models that involve all labeled and unlabeled pixels in the optimization problem are also called *(truly) semi-supervised*. However, note that these terms are not very precise as exhaustively discussed in [43, Chapter 25]. In general, the terms *transductive* and *inductive* are also used for methods that do not consist in solving an optimization problem, if they have analogous properties to those described here.

Contributions and Outline Section 3.1 presents a transductive segmentation approach using graph p-Laplacians. Our main contribution is the evaluation of p-Laplacians for $p \in \{1, 3/2, 2\}$ for transductive multi-class image segmentation including the choice of the weights. In learning, related models have been investigated before such as in the recent preprint of Bresson et al. [28] for $p = 1$. Further, we show that in case $p = 2$ the solution is automatically contained in the unit simplex. Although this has been proved before by [96] without mentioning relations to the 2-Laplacian, we give a novel different proof.

Section 3.2 presents a regularized least squares model in vector-valued reproducing kernel Hilbert spaces for inductive image segmentation that has been used for image colorization before



(a) (b)

Figure 3.1.: The segment membership of the gray unlabeled pixels in (a) is determined by the segmentation (b).

[71]. In fact, due to our choice of a block diagonal kernel matrix, this approach can also be understood as the application of a regularized least squares model in scalar-valued reproducing kernel Hilbert spaces to every class separately. The obtained segmentation results are promising. We compare the results to those obtained by the p-Laplacian models. In general, the RKHS approach provides more accurate solutions while the p-Laplacian results are smoother.

The main novelty of this chapter is proposed in Section 3.3. In this section a combination of the models in Sections 3.1 and 3.2 is presented to benefit from the advantages of both approaches. Firstly, we review an inductive combination of the functionals by addition that has been used for two-class learning with $p = 2$ before [142, 13]. This approach is not suitable for image segmentation on usual desktop computers since the number data points, i.e., pixels, is very large for reasonably sized images. However, based on our preprint [88], this model has been investigated very recently for $p = 1$, number of segments $c \geq 2$, and learning tasks with fewer data points by Bresson and Zhang [27]. To make a combination applicable to image segmentation we propose a novel transductive model incorporating a projection.

Section 3.4 states results for medical images and shows that distinct models are suitable for different kinds of images.

Notation Before introducing the first transductive model in the following section, we fix some notation that will be used throughout the whole chapter. Let L_k denote the index set of pixels labeled to be in segment k, let $L := \bigcup_{k=1}^{c} L_k$ denote the index set of all labeled pixels, and let U denote the index set of unlabeled pixels. Naturally, we assume $L_k \neq \emptyset$ and $L_{k_1} \cap L_{k_2} = \emptyset$ for all $k, k_1, k_2 \in \{1, \dots, c\}$ as well as $L \cap U = \emptyset$ and $L \cup U = \{1, \dots, N\}$. The given labels are

collected in vectors of characteristic functions of the sets $L_1, \dots, L_c$ denoted by

$$\ell(j) := \begin{pmatrix} \ell_1(j) \\ \vdots \\ \ell_c(j) \end{pmatrix} \in \mathbb{R}^c, \ \ell_k := \begin{pmatrix} \ell_k(1) \\ \vdots \\ \ell_k(N) \end{pmatrix} \in \mathbb{R}^N, \ \ell := \begin{pmatrix} \ell_1 \\ \vdots \\ \ell_c \end{pmatrix} \in \mathbb{R}^{cN} \tag{3.1}$$
$$\text{where } \ell_k(j) := \begin{cases} 1 & \text{if } j \in L_k, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we use $u = (u_k)_{k=1}^c$ where $u_k = (u_k(j))_{j=1}^N$ and $u(j) = (u_k(j))_{k=1}^c$ to denote the segment membership especially of unlabeled pixels. For index sets J, J' we work with the notation

$$u(J) = (u_k(J))_{k=1}^c, \ u_k(J) = (u_k(j))_{j\in J},$$
$$\ell(J) = ((\ell_k(J))_{j\in J})_{k=1}^c, \ell_k(j) = (\ell_k(j))_{j\in J}, \text{ and } A_{J,J'} = (a_{j,j'})_{j\in J, j'\in J'}$$

where A denotes an arbitrary matrix with components $a_{j,j'}$.

3.1 Transductive Multi-Class Segmentation with p-Laplacians

For transductive segmentation we restate the minimal cut problem (1.20) in terms of p-Laplacians (1.26) for $\infty > p \geq 1$ as

$$\text{(MinCut)} \qquad \operatorname*{argmin}_{u(U)\in\mathbb{R}^{c|U|}} E_{\Delta_p}\left(\begin{pmatrix} \ell(L) \\ u(U) \end{pmatrix}\right) \quad \text{subject to} \quad u(U) \in \{e_1, \dots, e_c\}^{|U|}$$
$$\text{where } E_{\Delta_p}(u) := \frac{2}{p}\sum_{k=1}^c \langle u_k, \Delta_p u_k\rangle = \frac{1}{p}\sum_{k=1}^c \sum_{i,j=1}^N w_{i,j}|u_k(i) - u_k(j)|^p.$$

Our segmentation model can be obtained by relaxing the binary constraints of the NP-hard problem (MinCut) yielding

$$(\Delta_p\text{P}) \qquad \operatorname*{argmin}_{u(U)\in\mathbb{R}^{c|U|}} E_{\Delta_p}\left(\begin{pmatrix} \ell(L) \\ u(U) \end{pmatrix}\right) \quad \text{subject to} \quad u(U) \in S_c^{|U|}.$$

In the literature, p-Laplacians have been considered for 2-class learning, see [101]. A p-Laplacian which differs from those used in this work was applied for learning in [174]. For $p \in \{1, 2\}$ and 2 classes (Δ_pP) was used for supervised segmentation by Sinop and Grady [144]. Additionally, in [144] the ∞-norm model

$$\operatorname*{argmin}_{v\in\mathbb{R}^N} \max_{i,j} w_{i,j}|v_i - v_j| = \operatorname*{argmin}_{v\in\mathbb{R}^N} \lim_{p\to\infty} \sqrt[p]{\sum_{i,j=1}^N (w_{i,j}|v(i) - v(j)|)^p}$$

was proposed.

Singaraju, Grady, and Vidal [143] introduced in 2009 different values $p_{i,j} \in [1, \infty)$ for every pair of pixels. The corresponding model reads

$$\begin{aligned} &\operatorname*{argmin}_{v \in \mathbb{R}^N} \sum_{i,j=1}^{N} w_{i,j} |v(i) - v(j)|^{p_{i,j}} \\ &\text{subject to} \quad v(j) = 1 \text{ if } j \in L_1 \text{ and } v(j) = 0 \text{ if } j \in L_2. \end{aligned} \tag{3.2}$$

Singaraju et al. show that a solution $\hat{v}$ of (3.2) fulfills $\hat{v}(j) \in [0,1]$ for all $j \in \{1, \dots, N\}$. Reformulating (3.2) into our multi-class notation yields

$$\begin{aligned} 2 \sum_{i,j=1}^{N} w_{i,j} |v(i) - v(j)|^{p_{i,j}} &= \sum_{i,j=1}^{N} w_{i,j} |v(i) - v(j)|^{p_{i,j}} + \sum_{i,j=1}^{N} w_{i,j} |1 - v(j) - (1 - v(i))|^{p_{i,j}} \\ &= \sum_{i,j=1}^{N} w_{i,j} |u_1(j) - u_1(i)|^{p_{i,j}} + \sum_{i,j=1}^{N} w_{i,j} |u_2(j) - u_2(i)|^{p_{i,j}} \end{aligned}$$

where we substitute $v(j)$ with $u_1(j)$ and $1 - v(j)$ with $u_1(j), u_2(j)$ for all $j \in \{1, \dots, N\}$. This implies $\hat{u}_2(j) \in [0,1]$ and $\hat{u}_1(j) + \hat{u}_2(j) = \hat{v}(j) + 1 - \hat{v}(j) = 1$, i.e., the simplex constraint $\hat{u} \in S_2^N$ is fulfilled automatically for a solution $\hat{u}$. If $p = 2$, we will see in the next section that this property is also fulfilled for $c > 2$.

3.1.1 The 2-Laplacian Model

Several variants of (Δ_2P) have been used in machine learning for a long time. For instance, transductive two-class learning is tackled with (Δ_2P) by Zhu et al. [178] in 2003. Very recently, (Δ_2P) has been used for supervised multi-class image segmentation by Law et al. [96] without mentioning any connections to the graph 2-Laplacian.

We sort Δ_2 defined in (1.22) according to labeled and unlabeled components yielding

$$W = (w_{i,j})_{i,j=1}^{N} = \begin{pmatrix} W_{LL} & W_{LU} \\ W_{UL} & W_{UU} \end{pmatrix} \quad \text{and} \quad D^{\mathcal{G}} = \operatorname{diag}\left(\sum_{j=1}^{N} w_{1,j}, \dots, \sum_{j=1}^{N} w_{N,j} \right).$$

Then E_{Δ_2} can be reformulated as

$$E_{\Delta_2}\left(\begin{pmatrix} \ell(L) \\ u(U) \end{pmatrix} \right) = \sum_{k=1}^{c} \begin{pmatrix} \ell_k(L) \\ u_k(U) \end{pmatrix}^{\top} \underbrace{\begin{pmatrix} D_{LL}^{\mathcal{G}} - W_{LL} & -W_{LU} \\ -W_{UL} & D_{UU}^{\mathcal{G}} - W_{UU} \end{pmatrix}}_{=:\Delta_2} \begin{pmatrix} \ell_k(L) \\ u_k(U) \end{pmatrix}. \tag{3.3}$$

Note that $W_{LU} = W_{UL}^{\top}$ due to the symmetry of our weights.

A unique solution of (Δ_2P) can be obtained by just minimizing the $E_{\Delta_2}((\ell(L)^{\top}, u(U)^{\top})^{\top})$ with respect to $u(U)$, if the weights are chosen appropriately, e.g., as for our experiments in Section 3.1.3. In particular, the minimizer of (3.3) automatically fulfills the simplex constraint. To prove the corresponding theorem we need the notion of an M-matrix. A matrix $A \in \mathbb{R}^{n \times n}$

is called an *M-matrix* if $a_{i,j} \leq 0$ for all $i \neq j$ and if it is *inverse isotonic*, i.e., A^{-1} exists and $A^{-1} \geq 0$ component-wise. In general it is hard to see if a matrix is inverse isotonic. However, it is well-known that any strictly diagonally dominant matrix A which fulfills $a_{i,i} > 0$ for all $i = 1, \ldots, n$ and $a_{i,j} \leq 0$ for all $i \neq j$ is an M-matrix, see, e.g., [83, p. 113ff] and [150, p. 303].

Theorem 3.1. *Assume that the weights in* $(\Delta_2\mathrm{P})$ *are chosen such that* $\mathcal{N}_i \cap L \neq \emptyset$ *for all* $i \in U$*, i.e., for every* $i \in U$ *there exists at least one* $j \in L$ *such that* $w_{i,j} > 0$*. Then,* $(\Delta_2\mathrm{P})$ *has a unique solution* $\hat{u}(U)$ *given by the solutions* $\hat{u}_k(U)$ *of the linear systems of equations*

$$(D^{\mathcal{G}}_{UU} - W_{UU})\hat{u}_k(U) = W_{UL}\ell_k(L), \quad k = 1, \ldots, c.$$

Proof. The function E_{Δ_2} can be rewritten as

$$E_{\Delta_2}\left(\begin{pmatrix}\ell(L)\\ u(U)\end{pmatrix}\right)$$

$$= \sum_{k=1}^{c} \left((u_k(U))^\top (D^{\mathcal{G}}_{UU} - W_{UU})u_k(U) - 2(u(U))^\top W_{UL}\ell_k(L) + (\ell_k(L))^\top (D^{\mathcal{G}}_{LL} - W_{LL})\ell_k(L)\right).$$

Further, $\hat{u}(U)$ is a minimizer of this convex function if and only if

$$(D^{\mathcal{G}}_{UU} - W_{UU})\hat{u}_k(U) = W_{UL}\ell_k(L) \tag{3.4}$$

for all $k = 1, \ldots, c$. By our assumption on the weights, the matrix $D^{\mathcal{G}}_{UU} - W_{UU}$ has positive diagonal entries and non-diagonal entries smaller than or equal to zero. Moreover, it is strictly diagonally dominant such that it is an M-matrix. Thus, $(D^{\mathcal{G}}_{UU} - W_{UU})^{-1}$ exists and the linear systems in (3.4) have unique solutions $\hat{u}_k(U)$, $k = 1, \ldots, c$. Further, since $\ell(L) \geq 0$, $W_{UL} \geq 0$, and $(D^{\mathcal{G}}_{UU} - W_{UU})^{-1} \geq 0$ we conclude that $\hat{u}(U) \geq 0$. To show that $\hat{u}(U) \in S_c^{|U|}$ it remains to prove that

$$\sum_{k=1}^{c} \hat{u}_k(U) = \mathbf{1}_{|U|}. \tag{3.5}$$

Summing up the equations in (3.4) we obtain

$$(D^{\mathcal{G}}_{UU} - W_{UU})\sum_{k=1}^{c} \hat{u}_k(U) - W_{UL}\sum_{k=1}^{c} \ell_k(L) = 0$$

and by the choice of the labeled components in (3.1) we have

$$(D^{\mathcal{G}}_{UU} - W_{UU})\sum_{k=1}^{c} \hat{u}_k(U) - W_{UL}\mathbf{1}_{|L|} = 0.$$

By definition of $D^{\mathcal{G}}$, we know that $(-W_{UL}|D^{\mathcal{G}}_{UU} - W_{UU})\begin{pmatrix}\mathbf{1}_{|L|}\\ \mathbf{1}_{|U|}\end{pmatrix} = 0$ so that

$$(D^{\mathcal{G}}_{UU} - W_{UU})\sum_{k=1}^{c} \hat{u}_k(U) - (D^{\mathcal{G}}_{UU} - W_{UU})\mathbf{1}_{|U|} + W_{UL}\mathbf{1}_{|L|} - W_{UL}\mathbf{1}_{|L|} = 0,$$

$$(D^{\mathcal{G}}_{UU} - W_{UU})\left(\sum_{k=1}^{c} \hat{u}_k(U) - \mathbf{1}_{|U|}\right) = 0.$$

Since $D^{\mathcal{G}}_{UU} - W_{UU}$ is invertible, this implies (3.5). □

This result, formulated without the notion of Δ_2, can also be found in [96] with a completely different proof. Their analysis uses duality considerations and Karush-Kuhn-Tucker conditions. Law et al. [96] have chosen sophisticated weights for their model which in particular meet the assumption of this theorem.

3.1.2 The p-Laplacian Model for $p \in [1, \infty)$

In this section we consider (Δ_pP) with $\infty > p \geq 1$. For transductive multi-class learning Bresson et al. investigate in [28] the problem (Δ_1P) and a supervised multi-class variant of the ratio Cheeger cut (1.27).

We apply PDHGM*b* [39, 56, 176] described in Subsection 2.4.1 to obtain a solution of (Δ_pP). To this end, let $A \in \mathbb{R}^{\nu N \times N}$ denote the matrix corresponding to the linear mapping

$$x \mapsto \left(\left(w_{i,j}^{1/p} (x(i) - x(j)) \right)_{j \in \mathcal{N}_i} \right)_{i=1}^{N}, \tag{3.6}$$

where we assume for the sake of simplicity that $|\mathcal{N}_i| = |\mathcal{N}_j| =: \nu$ for all $i, j \in \{1, \ldots, N\}$. Further, let $A_U \in \mathbb{R}^{\nu N, |U|}$ and $A_L \in \mathbb{R}^{\nu N, |L|}$ denote the matrices containing the columns of A corresponding to the indices in U and L, respectively. Then, we reformulate (Δ_pP) to

$$\operatorname*{argmin}_{u(U)} \frac{1}{p} \sum_{k=1}^{c} \|A_U u_k(U) + A_L \ell_k(L)\|_p^p \quad \text{subject to} \quad u(U) \in S_c^{|U|}.$$

Using the notation $M_U := I_c \otimes A_U$, $y := (A_L \ell_k(L))_{k=1}^c$ and the indicator function $\iota_{S_c^{|U|}}$ of $S_c^{|U|}$ this problem can be further rewritten as

$$\operatorname*{argmin}_{u(U)} \frac{1}{p} \|M_U u(U) + y\|_p^p + \iota_{S_c^{|U|}}(u(U))$$

and equivalently as

$$\operatorname*{argmin}_{u(U), v} \frac{1}{p} \|v + y\|_p^p + \iota_{S_c^{|U|}}(u(U)) \quad \text{subject to} \quad M_U u(U) = v. \tag{3.7}$$

PDHGM*b* applied to (3.7) is stated in Algorithm 10.

Algorithm 10: PDHGMb for (Δ_pP)

Input: $M_U = I_c \otimes A_U$, $y = (A_L \ell_k(L))_{k=1}^c$, and two parameters γ and τ.
Output: Segment membership vector u.
Initialization of $u(U)^{(1)}$, $v^{(1)}$, $b^{(1)}$ and $b^{(0)}$
for $n = 1, 2, \ldots$ *until a stopping criterion is reached* **do**

$$
\begin{aligned}
u(U)^{(n+1)} &= \operatorname*{argmin}_{u(U)} \iota_{S_c^{|U|}} + \frac{1}{2\tau} \left\| u(U) - (u(U)^{(n)} - \gamma\tau M_U^\top (2b^{(n)} - b^{(n-1)})) \right\|_2^2 \\
v^{(n+1)} &= \operatorname*{argmin}_{v} \frac{1}{p} \|v + y\|_p^p + \frac{1}{2\gamma} \left\| v - (b^{(n)} + M_U u(U)^{(n+1)}) \right\|_2^2 \\
b^{(n+1)} &= b^{(n)} + M_U u(U)^{(n+1)} - v^{(n+1)}
\end{aligned}
$$

end

The first minimization step requires a projection of $(u(U)^{(n)} - \gamma\tau M_U^\top (2b^{(n)} - b^{(n-1)}))$ onto the simplex $S_c^{|U|}$ which can be done separately for all $j \in U$, see Remark 2.26 and [110, 52].

The second minimization step can be rewritten by setting $z = v + y$ as

$$
\hat{z} = \operatorname*{argmin}_{z} \frac{1}{p} \|z\|_p^p + \frac{1}{2\gamma} \left\| z - (y + b^{(n)} + M_U u(U)^{(n+1)}) \right\|_2^2 .
$$

For $p = 1$, the minimizer $\hat{z}$ can be computed by the component-wise soft shrinkage of $a := y + b^{(n)} + M_U u(U)^{(n+1)}$ with threshold γ, see [163]. For $p > 1$, the minimizer can be computed for every component separately by

$$
\hat{z}_i = \operatorname*{argmin}_{x \in \mathbb{R}} \frac{1}{p}|x|^p + \frac{1}{2\gamma}(x - a_i)^2 \quad \text{which gives} \quad \operatorname{sign}(x)|x|^{p-1} + \frac{1}{\gamma}(x - a_i) = 0.
$$

This can be solved by a semi-implicit (Weiszfeld-like) method while for $p = \frac{3}{2}$ we obtain the analytical solution

$$
\hat{z}_i = \begin{cases} a_i + \frac{\gamma}{2} - \sqrt{(a_i + \frac{\gamma}{2}) - a_i^2} & \text{if } a_i > 0, \\ 0 & \text{if } a_i = 0, \\ a_i - \frac{\gamma}{2} + \sqrt{(a_i - \frac{\gamma}{2}) - a_i^2} & \text{if } a_i < 0. \end{cases}
$$

Remark 3.2. *Model ($\Delta_p P$) can be rewritten as*

$$
\operatorname*{argmin}_{u \in \mathbb{R}^{cN}} E_{\Delta_p}(u) \quad \textit{subject to} \quad u(L) = \ell(L) \textit{ and } u \in S_c^N .
$$

This reformulation makes the relation to the problem

$$
\operatorname*{argmin}_{u \in \mathbb{R}^{cN}} \frac{1}{2} \|\ell(L) - I_c \otimes Ju\|_2^2 + \lambda E_{\Delta_p}(u) \quad \textit{subject to} \quad u \in S_c^N \tag{3.8}
$$

clearly visible where $J := \big(I_{|L|} | \mathbf{0}_{|L|,|U|}\big)$. The minimizer of (3.8) *can be found similarly as above. For instance in case $p = 2$, the minimizer is given by the solutions of the linear systems of equations*

$$
(J^\top J + \lambda \Delta_2)\hat{u}_k = J^\top \ell_k(L), \quad k = 1, \ldots, c.
$$

Assuming that Δ_2 is strictly diagonally dominant, we see that $J^\top J + \lambda\Delta_2$ is again an M-matrix and following the lines of Theorem 3.1 we can conclude that the solution $\hat{u}$ automatically fulfills the simplex constraints.

This regularized least squares formulation will be useful, if one cannot be sure that all labels ℓ are correct, e.g., if ℓ is the result of some algorithm. In the literature, similar formulations have been used for 2-class learning with $p = 2$, e.g., by [175].

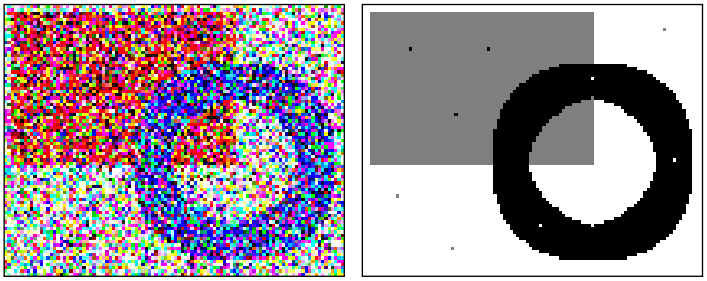
Remark 3.3. *The experiment in Figure 3.2 indicates that the simplex constraint of $(\Delta_1 P)$ and $(\Delta_{3/2} P)$ is not fulfilled automatically for $c > 2$ in contrast to the case $p = 2$. A formulation of $(\Delta_p P)$ without the simplex constraint is given by*

$$\operatorname*{argmin}_{u(U),v} \frac{1}{p} \|v + y\|_p^p \quad \text{subject to} \quad M_U u(U) = v$$

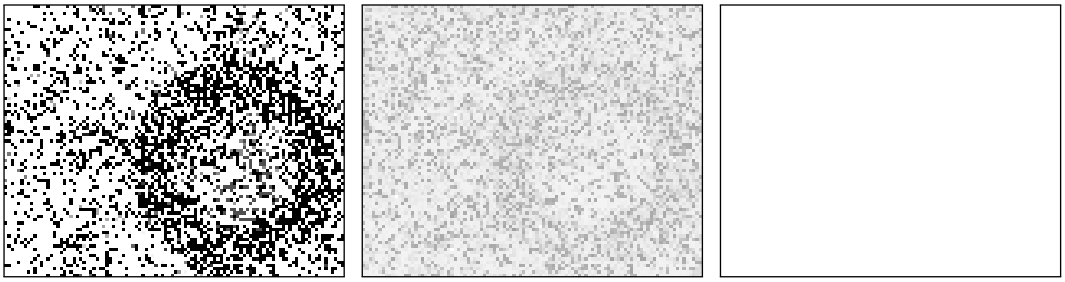
with v, y and M_U as in (3.7). To obtain the minimizing algorithm one only has to replace the first projection step in Algorithm 10 by the assignment

$$u(U)^{(n+1)} := (u(U)^{(n)} - \gamma\tau M_U^\top (2b^{(n)} - b^{(n-1)})).$$

Figures 3.2c and 3.2d show that the simplex constraint is not fulfilled for a solution $\hat{u}$ of this algorithm. Every pixel j in Figures 3.2c and 3.2d represents the value $\sum_{k=1}^c \hat{u}_k(j)$. A pixel j is white if it fulfills the simplex constraint, i.e., $\sum_{k=1}^c \hat{u}_k(j) = 1$, and a pixel j is black if $\sum_{k=1}^c \hat{u}_k(j) = 0$.



(a) Input (b) Truth superimposed by 9 given labels



(c) $p = 1$ (d) $p = 1.5$ (e) $p = 2$

Figure 3.2.: The simplex constraint is fulfilled automatically for $p = 2$ but in general this is not the case. Each pixel j in the second row depicts $\sum_{k=1}^c \hat{u}_k(j)$.

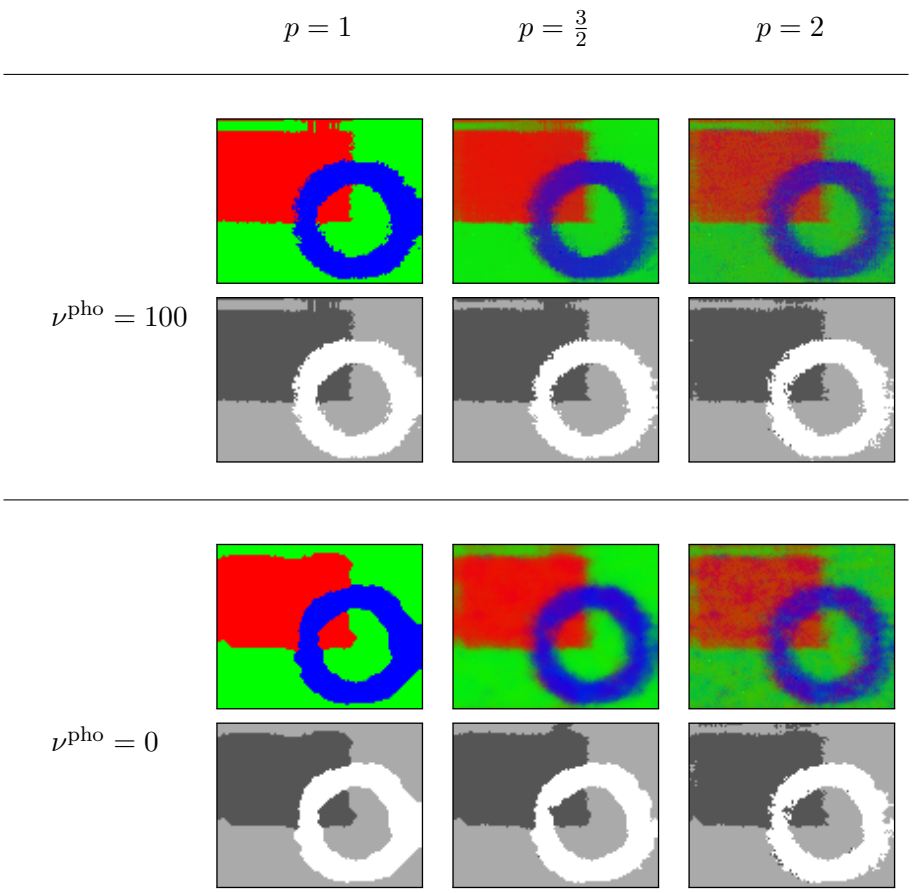


Figure 3.3.: Effect of different p and ν^{pho}. The other parameters are $\nu^{\text{lab}} = \frac{1}{2}, \vartheta = 5,\ \rho^2 = s = 363$.

3.1.3 The Impact of p and the Choice of the Weights in Numerical Examples

In this subsection we demonstrate the effect of p and the choice of the weights. The given image is depicted in Figure 3.2a. Figure 3.2b illustrates the true segmentation superimposed with the 9 labeled points. In all figures, the first row is the segment membership vector u reshaped to a matrix of dimension $n_2 \times n_1 \times 3$ since $c = 3$. The color red (vector $(1, 0, 0)^\top$) represents class 1, the color green class 2, and the color blue class 3. The second row shows the discretization, i.e., the final segmentation which is achieved by taking $\operatorname{argmax}_k u_k$ pixel-wise.

Typically, we observe in all experiments that a smaller value of p causes more regular results with smoother boundaries. Further, the results of the 1-Laplacian approach (Δ_1P) provide segment-membership-matrices close to binary while for larger p we obtain blurred boundaries and less contrast. However, the effect of the weights is quite strong and one can obtain similar segmentations after discretization for different p.

Based on [96], we describe in the following how the weights are chosen. The main idea is to gather geometrically and photometrically (also called tonally) similar pixels in neighborhoods.

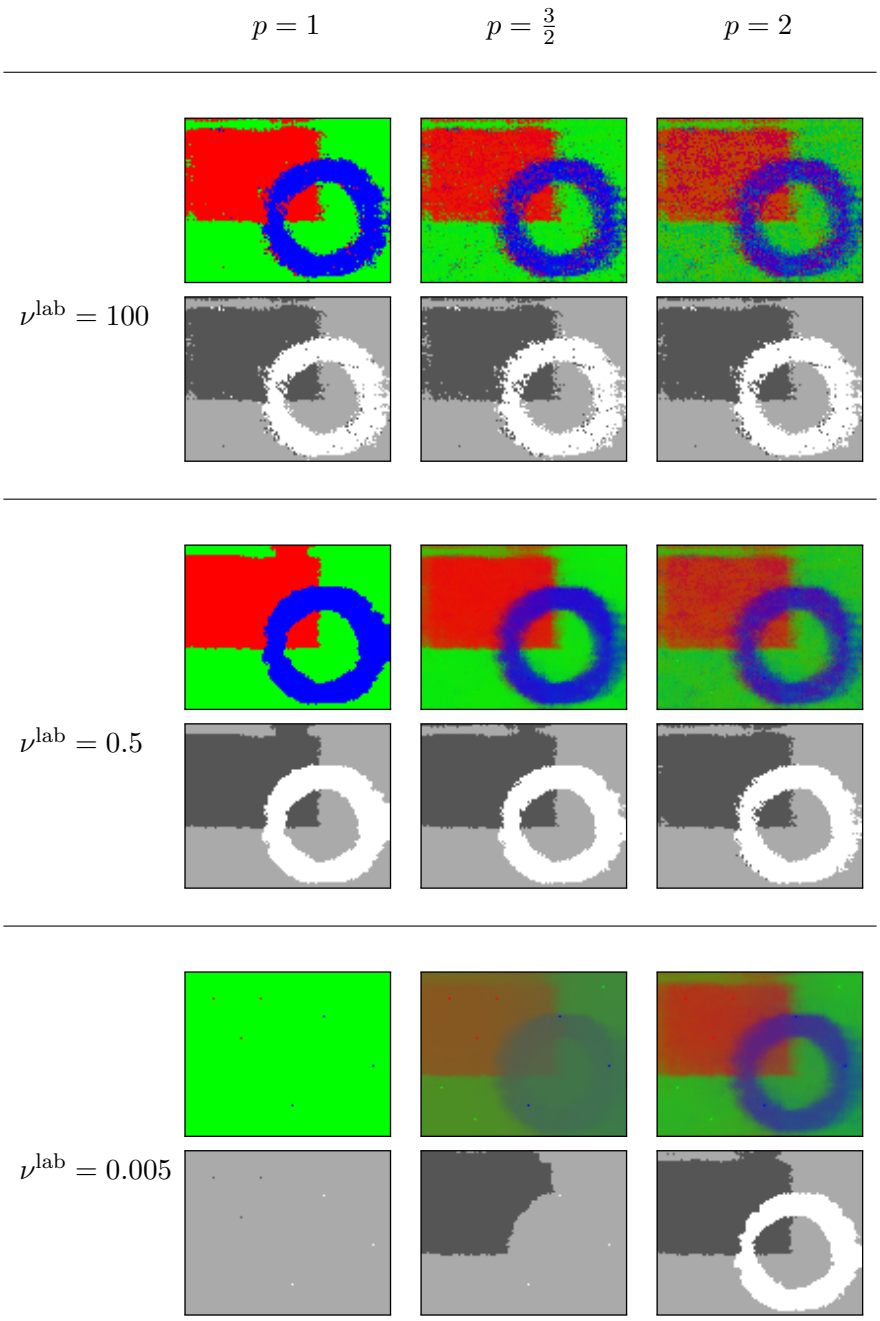


Figure 3.4.: Effect of ν^{lab}. The parameters $\nu^{\text{pho}} = 1$, $\vartheta = 5$, $\rho^2 = s = 363$ are fixed. With decreasing ν^{lab}, the influence of the labeled pixels decreases, and we obtain smoother results due to a stronger impact of weights between not necessarily labeled but similar pixels. For $p = 1$ in difference to $p = 2$ sharp edges are preserved. So in case $\nu^{\text{lab}} = 0.005$ we end up with a constant image with only the sharp edges to the labeled pixels left.

Geometric weights are based on pixel locations and photometric weights are based on color features. Further, we enforce every pixel to have labeled pixels in its neighborhood.

The geometric similarity between two pixels $i, j \in \{1, \dots, N\}$ is defined by

$$w_{i,j}^{\text{geo}} := \begin{cases} \dfrac{\exp\left(-\|(i_x - j_x, i_y - j_y)^\top\|_2^2\right)}{\sum\limits_{j \in \mathcal{N}_i^{\text{geo}}} \exp\left(-\|(i_x - j_x, i_y - j_y)^\top\|_2^2\right)} & \text{if } j \in \mathcal{N}_i^{\text{geo}}, \\ 0 & \text{otherwise} \end{cases}$$

where the $\mathcal{N}_i^{\text{geo}} := \{j \in \{1, \dots, N\} : \|(i_x - j_x, i_y - j_y)^\top\|_\infty \leq 1\}$ denotes the geometric neighborhood and $(i_x, i_y)^T$ denotes the pixel position of an image in matrix form, i.e., $i = n_1(i_y - 1) + i_x$.

The photometric neighborhood $\mathcal{N}_i^{\text{pho}}$ of the pixel i is defined as the set of the 4 most similar pixels j in a 17×17 window around pixel i with respect to the Euclidean norm of the feature vectors $\mathcal{F}(i)$ and $\mathcal{F}(j)$ obtained from the given image $f : \{1, \dots, N\} \to \mathbb{R}^m$. The feature vector $\mathcal{F}(i) \in \mathbb{R}^s$ is defined as the $(2\vartheta + 1) \times (2\vartheta + 1)$ window of radius ϑ around pixel i where $s = m(2\vartheta + 1)^2$. The weights are defined by

$$w_{i,j}^{\text{pho}} := \begin{cases} \kappa_i e^{-\frac{\|\mathcal{F}(i) - \mathcal{F}(j)\|_2^2}{\rho_i}} & \text{if } j \in \mathcal{N}_i^{\text{pho}}, \\ 0 & \text{otherwise} \end{cases}$$

where we normalize as in the geometric case with $\kappa_i := 1/\sum_{j \in \mathcal{N}_i^{\text{pho}}} e^{-\frac{\|\mathcal{F}(i) - \mathcal{F}(j)\|_2^2}{\rho_i}}$. For the parameter ρ_i we use either a constant ρ independent of i or we use the Euclidean norm of the component-wise sample variance of the color features $\mathcal{F}(j), j \in \mathcal{N}_i^{\text{pho}}$. The variance is defined by

$$\rho_i := \left\| \left(\frac{1}{|\mathcal{N}_i^{\text{pho}}| - 1} \sum_{j \in \mathcal{N}_i} (\mathcal{F}_m(j) - \overline{\mathcal{F}_m}) \right)_{m=1}^{s} \right\|_2 \tag{3.9}$$

where the mean is given by $\overline{\mathcal{F}_m} := \sum_{j \in \mathcal{N}_i^{\text{pho}}} \mathcal{F}_M(j)/|\mathcal{N}_i^{\text{pho}}|$.

We define the labeled neighborhood $\mathcal{N}_i^{\text{lab}}$ to be the 4 labeled pixels with smallest Euclidean distance to pixel i. The weights are defined analogously to the photometric weights. If there are too many pixels labeled, we choose a random sample of equal size for each segment. The neighborhood of a pixel i is given by $\mathcal{N}_i := \mathcal{N}_i^{\text{geo}} \cup \mathcal{N}_i^{\text{pho}} \cup \mathcal{N}_i^{\text{lab}}$.

The weight matrix is computed as follows. The geometric and photometric weights are added

$$W^* = \frac{1}{1 + \nu^{\text{pho}}} W^{\text{geo}} + \frac{\nu^{\text{pho}}}{1 + \nu^{\text{pho}}} W^{\text{pho}}. \tag{3.10}$$

Then, they are compared with the labeled weights via the element-wise maximum

$$\grave{W} = \max\left\{ \frac{\nu^{\text{lab}}}{1 + \nu^{\text{lab}}} W^{\text{lab}}, \frac{1}{1 + \nu^{\text{lab}}} W^* \right\}. \tag{3.11}$$

Finally, we use the symmetric weight matrix

$$W = \max\left\{\dot{W}, \dot{W}^T\right\}.$$

Figure 3.3 shows the effect of changing ν^{pho}. Decreasing ν^{pho} decreases the influence of the photometric neighborhood and increases the influence of the geometric neighborhood (3.10), resulting in smoother results.

A more severe effect can be shown by changing ν^{lab} in (3.11). We use the input image shown in Figures 3.2a and 3.2b. In Figure 3.4, we fix ν^{pho} and vary ν^{lab}. This comparison illustrates that decreasing ν^{lab} decreases the influence of the labeled pixels according to the definition of the weights (3.11). With decreasing ν^{lab} we obtain smoother results due to a stronger impact of the weights between not necessarily labeled but similar pixels.

Comparing among the same parameters, the resulting segment membership matrices u for $p = 1$ are close to binary and have smoother boundaries. By choosing different weight parameters, p-Laplacians with $p > 1$ can provide similar results to $p = 1$. In Figure 3.4 the segmentation result $\nu^{\text{lab}} = 0.5$ with $p = 1$ is similar to the segmentation result $\nu^{\text{lab}} = 0.005$ with $p = 2$. However, comparing the segment membership matrices $\hat{u}$ (shown in color) the one for $p = 1$ is sharper.

3.2 Supervised Segmentation in Reproducing Kernel Hilbert Spaces

In this section, we apply the image colorization method of [71] described in Section 1.1.3 to image segmentation. Instead of the segment membership vector u in the previous section, we consider a *segment membership function* $g = (g_k)_{k=1}^c : \mathbb{R}^2 \to \mathbb{R}^c$ belonging to a vector-valued RKHS $\mathcal{H}$ with kernel K as defined in 1.1.3.

3.2.1 The Segmentation Model

We find the segment membership function $\hat{g}$ as the solution of

$$\operatorname*{argmin}_{g\in\mathcal{H}} \sum_{i\in L} \|\ell(i) - g(x_i)\|_2^2 + \lambda \|g\|_{\mathcal{H}}^2, \quad \lambda > 0. \tag{3.12}$$

Alternatively, we can consider

$$\operatorname*{argmin}_{g\in\mathcal{H}} \|g\|_{\mathcal{H}} \quad \text{subject to} \quad g(x_i) = \ell(i),\ \forall i \in L. \tag{3.13}$$

Then, by the Representer Theorem [71, Proposition 1] the minimizers of (3.12) and (3.13) have the form

$$\hat{g} = \sum_{j\in L} K(\cdot, x_j)\alpha(j), \quad \alpha(j) \in \mathbb{R}^c. \tag{3.14}$$

We need only to determine $\alpha(j)$ for $j \in L$ to obtain the whole function $\hat{g} \in \mathcal{H}$. The difference to the colorization model of [71] is that we have replaced the color channels by segment membership functions, one for each class.

The kernel has the form $K(x_i, x_j) = \left(K^{\ell,k}(x_i, x_j)\right)_{\ell,k=1}^{c}$ and we use the notation

$$\mathbf{K}_{LL} := \begin{pmatrix} \left(K^{11}(x_i, x_j)\right)_{i,j\in L} & \dots & \left(K^{1c}(x_i, x_j)\right)_{i,j\in L} \\ \vdots & \ddots & \vdots \\ \left(K^{c1}(x_i, x_j)\right)_{i,j\in L} & \dots & \left(K^{cc}(x_i, x_j)\right)_{i,j\in L} \end{pmatrix} \in \mathbb{R}^{c|L|,c|L|},$$

and

$$\alpha := \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_c \end{pmatrix}, \quad \alpha_k = (\alpha_k(j))_{j\in L}.$$

To find the vectors $\alpha(j)$, we minimize the problem

$$(\mathcal{H}\mathrm{P}) \qquad \operatorname*{argmin}_{\alpha} \|\ell(L) - \mathbf{K}_{LL}\alpha\|_2^2 + \lambda \alpha^\top \mathbf{K}_{LL}\alpha$$

using (1.28), (3.12), and (3.14). The minimizer $\hat{\alpha}$ of $(\mathcal{H}\mathrm{P})$ is given by the solution of the linear system of equations

$$(\mathbf{K}_{LL} + \lambda I)\,\hat{\alpha} = \ell(L). \tag{3.15}$$

Note that the coefficient matrix is positive definite so that the solution is unique. Similarly, a minimizer of (3.13) can be computed by solving (3.15) with $\lambda = 0$, if $\mathbf{K}_{LL}$ is positive definite. Then, the segment membership function $\hat{g}$ can be evaluated at any $x \in \mathbb{R}^2$ by (3.14).

3.2.2 Numerical Experiments

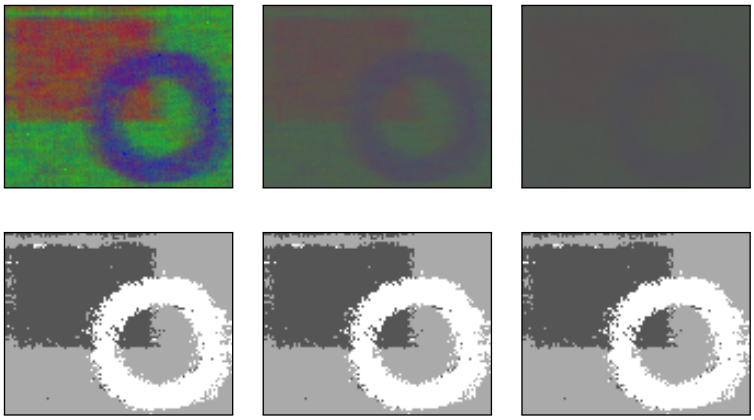
In the performed numerical experiments, $x_i = i \in \{1, \dots, N\}$ are image grid points. Similarly to [71], we decouple the classes in our applications and consider diagonal matrices $K(i, j) := \kappa(i, j)I_c$, $i, j \in L$ where the similarity between pixels i and j is denoted by $\kappa(i, j) \in \mathbb{R}_{>0}$. With these simplifications, $\mathbf{K}_{LL}$ becomes the block-diagonal matrix

$$\mathbf{K}_{LL} = \operatorname{diag}\left((\kappa(i, j))_{i,j\in L}, \dots, (\kappa(i, j))_{i,j\in L}\right).$$

For κ, we use

$$\kappa(i, j) := \exp\left(-\frac{\|\mathcal{F}(i) - \mathcal{F}(j)\|_2}{\sigma_1^2 d}\right) \exp\left(-\frac{\left\|\begin{pmatrix} i_x - j_x \\ i_y - j_y \end{pmatrix}\right\|_2}{\sigma_2^2\sqrt{n_1^2 + n_2^2}}\right)$$

where $1/\infty = 0$ and $(i_x, i_y)^T$ denotes the pixel position of an image in matrix form, i.e., $i = n_1(i_y - 1) + i_x$. Here $\mathcal{F}(i)$ denotes the same feature vector as in Subsection 3.1.3.



(a) (b) (c)

Figure 3.5.: Effect of λ in ($\mathcal{H}$P). Figures 3.2a and 3.2b are the input images. The top row (color) shows the segment membership function $\hat{g}$ and the second row the resulting segmentation achieved by taking $\operatorname{argmax}_k u_k$ component-wise. a) $\lambda = 0$, b) $\lambda = 0.1$ and c) $\lambda = 0.4$. The kernel parameters are $\vartheta = 5,\ \sigma_1^2 = 1,\ \sigma_2^2 = \infty$.

By (3.14), we obtain the values of our segment membership function $\hat{g}$ at the unlabeled points $x_i = i \in U$ by

$$\hat{g}_k = \begin{pmatrix} K_{LL} \\ K_{UL} \end{pmatrix} \hat{\alpha}_k, \tag{3.16}$$

where $K_{J_1 J_2} := (\kappa(i,j))_{i \in J_1, j \in J_2}$ for index sets J_1, J_2. Since $|L|$ is small, both the solution of the linear system (3.15) and the matrix-vector multiplications (3.16) can be computed efficiently.

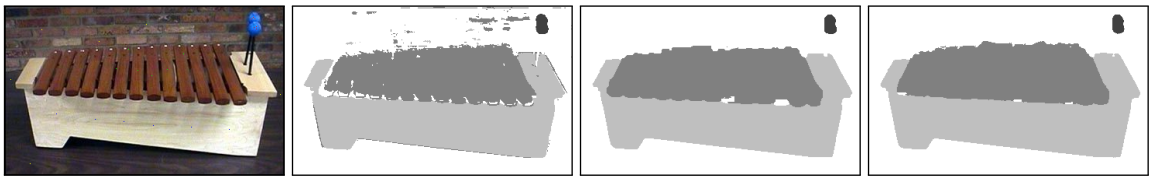
Remark 3.4. *Due to the choice of our kernel $K(i,j) = \kappa(i,j)I$, this approach can also be understood as regularized least squares in a scalar-valued RKHS applied for each k (also called least squares support vector machine (SVM) [154]). The Representer Theorem in the scalar valued case is due to Kimeldorf and Wahba [91].*

Remark 3.5. *The solution $\hat{\alpha}$ can be reused for different images. Given $\hat{\alpha}$, one can proceed as follows to obtain a segmentation of an unlabeled image:*

1. *Create $K_{\bar{U}L}$ where $\bar{U}$ contains all pixels of the unlabeled image and L are the labeled pixels of the image containing the labels.*
2. *Compute the segment membership function at the points in $\bar{U}$ by $K_{\bar{U}L}\hat{\alpha}$.*

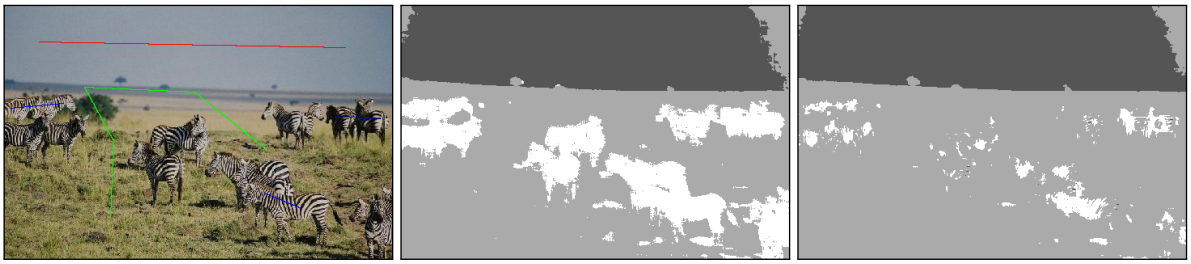
Therefore, we call this model inductive.

Let us turn to the obtained segmentations and compare them to the results of the p-Laplacian model. As a major difference to the p-Laplacian approach, the result of ($\mathcal{H}$P) depends only on the labeled points. This is also observable in the following experiments.



(a) Input (b) ($\mathcal{H}$P) (c) (Δ_1P) (d) (Δ_2P)

Figure 3.6.: a) Input image from [70], only 6 pixels per class are labeled. b) Result of ($\mathcal{H}$P) with $\sigma_1^2 = 1,\ \sigma_2^2 = 250,\ \vartheta = 3,\ \lambda = 0$. c) Result of ($\Delta_1$P) with $\nu^{\text{pho}} = 1,\ \nu^{\text{lab}} = 0.15,\ \rho^2 = 49,\ \vartheta = 3$. d) Result of ($\Delta_2$P) with $\nu^{\text{pho}} = 1,\ \nu^{\text{lab}} = 0.001,\ \rho^2 = 49,\ \vartheta = 3$.



(a) Input (b) ($\mathcal{H}$P) (c) (Δ_1P)

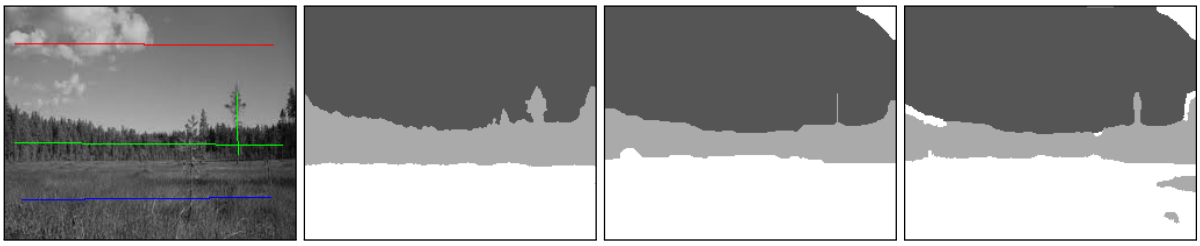
Figure 3.7.: a) Labeled input image. b) Segmentation result of ($\mathcal{H}$P) with $\sigma_1^2 = 1,\ \sigma_2^2 = \infty,\ \vartheta = 3,\ \lambda = 0$. c) Result of ($\Delta_1$P) with $\nu^{\text{pho}} = 1,\ \nu^{\text{lab}} = 1,\ \rho^2 = 49,\ \vartheta = 3$.

Figure 3.5 shows an experiment with different regularization parameters λ in the minimization problem ($\mathcal{H}$P). For larger λ the contrast of the segment membership function $\hat{g}$ decreases while the segmentation result stays quite similar. Hence, we will use $\lambda > 0$ only if a bad conditioning of the linear system forces us to.

Since the RKHS approach only uses the information from the labeled points L, the regularity of the result can be worse, as shown in Figure 3.6. The p-Laplacian models force unlabeled but similar points to be in the same segment. Compare also Figure 3.5 with Figures 3.3 and 3.4.

However, from ($\mathcal{H}$P) one can obtain more accurate results as illustrated in Figures 3.7 and 3.8. Figure 3.7 shows the results for a natural image taken from [70]. The results of ($\mathcal{H}$P) and (Δ_1P) are depicted in Figures 3.7b and 3.7c, respectively. Although Figure 3.7a has many details and the ground and zebras are hard to distinguish, the RKHS approach recovers the location of many different zebras correctly. Other values for the parameters $\rho, \nu^{\text{lab}}, \nu^{\text{pho}}, \vartheta$ did not improve the result of the (Δ_1P) significantly. Also with (Δ_2P) one does not obtain better results. In fact, for the parameters of Figure 3.7c the segmentations for $p = 1$ and $p = 2$ are quite similar.

Figure 3.8 shows an example of a landscape. Notice that sky, grass, and the small tree are better captured by the RKHS model. For (Δ_pP) the neighborhood of each pixel contains only a few labeled pixels, see Section 3.1.3 for more details. Increasing the number of labeled pixels in each neighborhood from 4 to 16 did not lead to significant improvements.



(a) Input (b) ($\mathcal{H}$P) (c) (Δ_1P) (d) (Δ_2P)

Figure 3.8.: a) Labeled input image. b) Segmentation result of ($\mathcal{H}$P) with $\sigma_1^2 = 1,\ \sigma_2^2 = 50,\ \vartheta = 5$. c) Result of ($\Delta_1$P) with $\nu^{\text{pho}} = 1,\ \nu^{\text{lab}} = 0.05,\ \rho^2 = 121,\ \vartheta = 5$. d) Result of ($\Delta_2$P) with $\nu^{\text{pho}} = 10,\ \nu^{\text{lab}} = 0.001,\ \rho^2 = 121,\ \vartheta = 5$.

3.3 Combining p-Laplacians and the RKHS model

As observed in the previous sections, the RKHS approach ($\mathcal{H}$P) uses only the information from labeled points L, while the p-Laplacian model (Δ_pP) also incorporates the similarity information between unlabeled points. Therefore (Δ_pP) can provide smoother results. On the other hand, ($\mathcal{H}$P) leads often to more accurate/detailed segmentation results as we have seen in the experiments of the previous section. To this end, we consider a combined model to benefit from both methods. We propose a projection approach such that a combination of ($\mathcal{H}$P) and (Δ_pP) becomes practicable for segmentation tasks with typically many data points, namely the image pixels.

First, we review a combination by addition as proposed in [13, 142] for two-class learning which couples ($\mathcal{H}$P) and (Δ_2P). In those papers ($\mathcal{H}$P) is addressed as RLP (regularized least squares). Since only two labels were considered, one can restrict the attention to a scalar-valued segment membership function $g : \mathbb{R}^2 \to \mathbb{R}$ and set a threshold on g to obtain the two classes. The optimal function $\hat{g}$ is obtained as the minimizer of

$$\operatorname*{argmin}_{g\in\mathcal{H}} \sum_{j\in L} \frac{1}{2} \|\ell(j) - g(x_j)\|_2^2 + \frac{\lambda}{2} \|g\|_{\mathcal{H}}^2 + \frac{\mu}{2} \sum_{i,j=1}^{N} w_{i,j} |g(x_i) - g(x_j)|^2. \tag{3.17}$$

Following the idea of the Representer Theorem, it is not hard to check that the minimizer $\hat{g}$ depends on all values x_j, $j \in \{1, \ldots, N\}$, i.e.,

$$\hat{g} = \sum_{j=1}^{N} K(\cdot, x_j)\beta(j). \tag{3.18}$$

Substituting this expression into (3.17) and using, the definition of the RKHS norm we conclude that the optimal $\hat{\beta}$ must be a solution of

$$\operatorname*{argmin}_{\beta} \frac{1}{2} \|\ell(L) - JK\beta\|_2^2 + \frac{\lambda}{2}\beta^\top K\beta + \frac{\mu}{2}\beta^\top K \triangle_2 K\beta$$

where

$$K := (\kappa(x_i, x_j))_{i,j \in L \cup U} = \begin{pmatrix} K_{LL} & K_{LU} \\ K_{LU} & K_{UU} \end{pmatrix}$$

and J as defined in Remark 3.2. Considering the gradient with respect to β and using that K is invertible, we obtain $\hat{\beta}$ by solving a linear system of equations

$$(J^\top J K + \lambda I_N + \mu \Delta_2 K)\hat{\beta} = J^\top \ell(L).$$

This means that one has to incorporate the whole fully populated matrix K, in particular K_{UU} into the computations. In segmentation tasks the number of unlabeled points is huge (nearly $N = n_1 n_2$ for images of size $n_1 \times n_2$) and the above model is not practicable both with respect to storage and computation time. However, this model with multiple classes and $p \geq 1$ is still applicable to learning tasks with fewer data points. This has been investigated in the very recent preprint [27] that has appeared during the review process of our corresponding publication [88].

3.3.1 A Projection Model

To avoid the computation with the huge matrix K_{UU}, we propose a combined model utilizing a projection idea. In contrast to the generalization ability (3.18), this model is again transductive with respect to the image grid points $x_i = i$, $i \in L \cup U$.

Let $\mathbf{K}_{LL} := I_c \otimes K_{LL}$ and $\mathbf{K}_{UL} := I_c \otimes (k(i,j))_{i \in U, j \in L}$. We consider the subspace H of $\mathbb{R}^{cN}$ defined by

$$H := \{ \underbrace{\begin{pmatrix} \mathbf{K}_{LL} \\ \mathbf{K}_{UL} \end{pmatrix}}_{\mathbf{K}} \alpha : \alpha \in \mathbb{R}^{c|L|} \}$$

with square norm

$$\|h\|_H^2 := \alpha^\top \mathbf{K}_{LL} \alpha \quad \text{for} \quad h := \mathbf{K}\alpha.$$

We are looking for vectors $\hat{h} \in H$ and $\hat{u} \in \mathbb{R}^{cN}$ solving the *combined model*

$$(\mathcal{H}\Delta_p\mathrm{P}) \qquad \begin{aligned} &\operatorname*{argmin}_{h \in H, u \in \mathbb{R}^{Nc}} \frac{1}{2} \sum_{i \in L} \|\ell(i) - h(i)\|_2^2 + \frac{\lambda}{2} \|h\|_H^2 + \frac{\mu}{p} \sum_{k=1}^{c} \sum_{i,j=1}^{N} w_{i,j} |u_k(i) - u_k(j)|^p, \\ &\text{subject to} \quad h = Pu. \end{aligned}$$

More precisely, we expect that $\hat{h}$ has similar properties as the solution $(\hat{g}(i))_{i \in \{1,\dots,N\}}$ of $(\mathcal{H}\mathrm{P})$. Further, $\hat{u}$ is the vector we are really looking for and $\hat{u}$ is smoothed by the Laplacian regularization.

Here $P : \mathbb{R}^{cN} \to H$ denotes the *orthogonal projector* from $\mathbb{R}^{cN}$ onto H. By definition of H, this orthogonal projector is given by $P = \mathbf{K}\mathbf{K}^\dagger$, where $\mathbf{K}^\dagger := (\mathbf{K}^\top \mathbf{K})^{-1} \mathbf{K}^\top$ is the *Moore-Penrose inverse* of $\mathbf{K}$. Note that $\mathbf{K}^\top \mathbf{K}$ is positive definite and thus invertible. Then, for $h := \mathbf{K}\alpha$ the constraint can be written as $\mathbf{K}\alpha = \mathbf{K}\mathbf{K}^\dagger u$. Since $\mathbf{K}$ has full column rank this is equivalent to

$$\alpha = \mathbf{K}^\dagger u.$$

Substituting this into ($\mathcal{H}\Delta_p$P) we obtain

$$\begin{aligned}\underset{u\in\mathbb{R}^{Nc}}{\operatorname{argmin}} \frac{1}{2}\left\|\ell(L)-\mathbf{K}_{LL}\mathbf{K}^{\dagger}u\right\|_2^2+\frac{\lambda}{2}u^{\top}(\mathbf{K}^{\dagger})^{\top}\mathbf{K}_{LL}\mathbf{K}^{\dagger}u\\ +\frac{\mu}{p}\sum_{k=1}^{c}\sum_{i,j=1}^{N}w_{i,j}|u_k(i)-u_k(j)|^p.\end{aligned} \tag{3.19}$$

For $p=2$, the solution $\hat{u}$ of (3.19) is given by the solution of the linear system of equations

$$\left((\mathbf{K}^{\dagger})^{\top}\mathbf{K}_{LL}(\mathbf{K}_{LL}+\lambda I_{c|l|})\mathbf{K}^{\dagger}+\mu\Delta_2\right)u=\mathbf{K}^{\dagger}\mathbf{K}_{LL}\ell(L). \tag{3.20}$$

We use the *conjugate gradient method* [82] to solve (3.20). Note that the huge matrix K_{UU} does not appear in the above linear system and we are able to implement the corresponding matrix multiplications efficiently.

For $p=1$, reformulation of (3.19) with $M:=I_c\otimes A$ and A as defined in (3.6) yields

$$\begin{aligned}&\underset{u\in\mathbb{R}^{Nc}}{\operatorname{argmin}} \frac{1}{2}\left\|\ell(L)-\mathbf{K}_{LL}\mathbf{K}^{\dagger}u\right\|_2^2+\frac{\lambda}{2}u^{\top}(\mathbf{K}^{\dagger})^{\top}\mathbf{K}_{LL}\mathbf{K}^{\dagger}u+\mu\left\|v\right\|_1\\ &\text{subject to } Mu=v.\end{aligned}$$

To solve this problem we apply the alternating direction method of multipliers (ADMM) as stated in Algorithm 11, see, e.g., [24, 54, 56, 138] and Section 2.4.1.

Algorithm 11: ADMM for ($\mathcal{H}\Delta_1$P).

Input: $M(=I_c\otimes A$ in (3.6)) and γ.
Output: segment membership vector u.
Initialization of $u^{(0)}\in\mathbb{R}^{Nc}$ and $b^{(0)},v^{(0)}\in\mathbb{R}^{N^2c}$
for $n=0,1,\ldots$ *until a stopping criterion is reached* **do**

$$u^{(n+1)}=\underset{u}{\operatorname{argmin}} \frac{1}{2}\left\|\ell(L)-\mathbf{K}_{LL}\mathbf{K}^{\dagger}u\right\|_2^2+\frac{\lambda}{2}u^{\top}(\mathbf{K}^{\dagger})^{\top}\mathbf{K}_{LL}\mathbf{K}^{\dagger}u+\frac{1}{2\gamma}\left\|b^{(n)}+Mu-v^{(n)}\right\|_2^2 \tag{3.21}$$

$$v^{(n+1)}=\underset{v}{\operatorname{argmin}}\,\mu\left\|v\right\|_1+\frac{1}{2\gamma}\left\|b^{(n)}+Mu^{(n+1)}-v\right\|_2^2 \tag{3.22}$$

$$b^{(n+1)}=b^{(n)}+Mu^{(n+1)}-v$$

end

The minimizer of (3.22) follows by soft-shrinkage of $b^{(n)}+Mg$ with the threshold $\gamma\mu$. The minimizer of (3.21) can be obtained by solving a system similar to (3.20), namely

$$\left((\mathbf{K}^{\dagger})^{\top}\mathbf{K}_{LL}\left(\mathbf{K}_{LL}+\lambda I_{c|L|}\right)\mathbf{K}^{\dagger}+\frac{1}{\gamma}M^{\top}M\right)u=(\mathbf{K}^{\dagger})^{\top}\mathbf{K}_{LL}\ell(L)+\frac{1}{\gamma}M^{\top}(v^{(n)}-b^{(n)}).$$

Alternatively, one could use PDHGM*b*, see Section 2.4.1. The only difference to ADMM is that the first step reads

$$u^{(n+1)} = \operatorname*{argmin}_u \frac{1}{2}\left\|\ell(L) - \mathbf{K}_{LL}\mathbf{K}^\dagger u\right\|_2^2 + \frac{\lambda}{2}u^\top (K^\dagger)^\top \mathbf{K}_{LL}\mathbf{K}^\dagger u + \frac{1}{2\tau}\left\|u - \left(\underbrace{u^{(n)} - \tau\gamma M^\top(2b^{(n)} - b^{(n-1)})}_{z}\right)\right\|_2^2$$

where $\tau > 0$ and $\tau\gamma \leq 1/\left\|M\right\|^2$.

Remark 3.6. *The various models we consider in this chapter can be summarized as follows. The general p-Laplacian model* (Δ_pP) *is affected by both labeled and unlabeled points as visible from its formulation*

$$\operatorname*{argmin}_{u(U)\in\mathbb{R}^{c|U|}} E_{\Delta_p}\left(\begin{pmatrix}\ell(L)\\ u(U)\end{pmatrix}\right) \quad \textit{subject to} \quad u(U) \in S_c^{|U|}$$

$$\textit{where } E_{\Delta_p}(u) := \frac{2}{p}\sum_{k=1}^{c}\langle u_k, \Delta_p u_k\rangle = \frac{1}{p}\sum_{k=1}^{c}\sum_{j=1}^{N} w_{i,j}|u(i) - u(j)|^p.$$

Compared to ($\mathcal{H}$P) *we obtain smoother results in numerical experiments. An optimal segment membership vector* $\hat{u}$ *is close to binary for* $p = 1$ *and rather blurry for* $p = 2$*. However, by adjusting the weights* W *one can obtain similar results for different* p *after discretization.*

In the RKHS model ($\mathcal{H}$P)*, the regularization appears from the reproducing kernel Hilbert space norm, i.e.,*

$$\operatorname*{argmin}_\alpha \left\|\ell(L) - \mathbf{K}_{LL}\alpha\right\|_2^2 + \lambda\alpha^\top \mathbf{K}_{LL}\alpha.$$

This regularization with $\lambda > 0$ *is only used to improve the conditioning of the resulting linear system if necessary. With* (3.16) *we know that the result of* ($\mathcal{H}$P) *is determined from the labeled points, i.e., no information of similar unlabeled points is used. By using few labeled points, the computation becomes very fast and numerical experiments show promising results for complicated segmentation problems.*

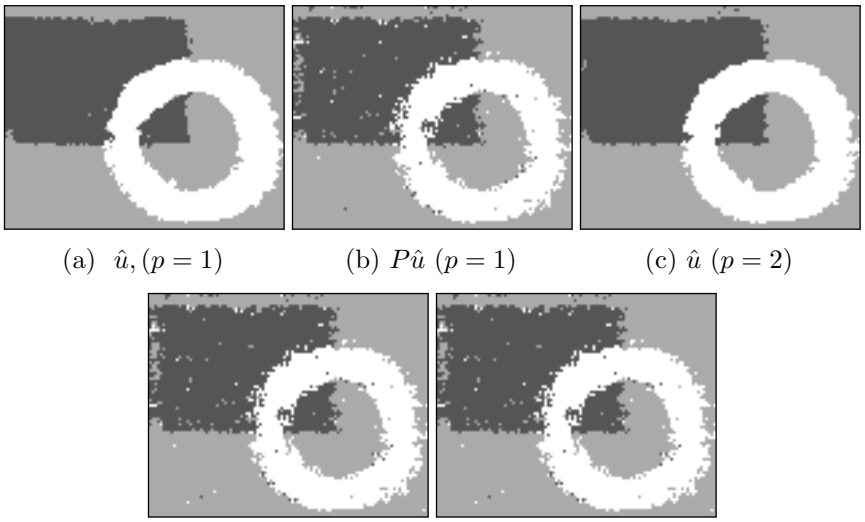
The combined model ($\mathcal{H}\Delta_p$P) *has the form*

$$\operatorname*{argmin}_{u,h} \mu E_{\Delta_p}(u) + \lambda\left\|h\right\|_{\mathcal{H}}^2 + \frac{1}{2}\|\ell(L) - h\|^2 \quad \textit{subject to} \quad Pu = h.$$

Here u *represents the segment membership vector regularized by the p-Laplacian and* h *is the segment-membership vector in the subspace* H*. The two vectors* u *and* h *are connected by an orthogonal projection* $P : \mathbb{R}^{cN} \to H$*, which has an explicit form given by the kernel* K*. The combined model regularizes the RKHS result and benefits from both approaches.*

3.3.2 Numerical Experiments

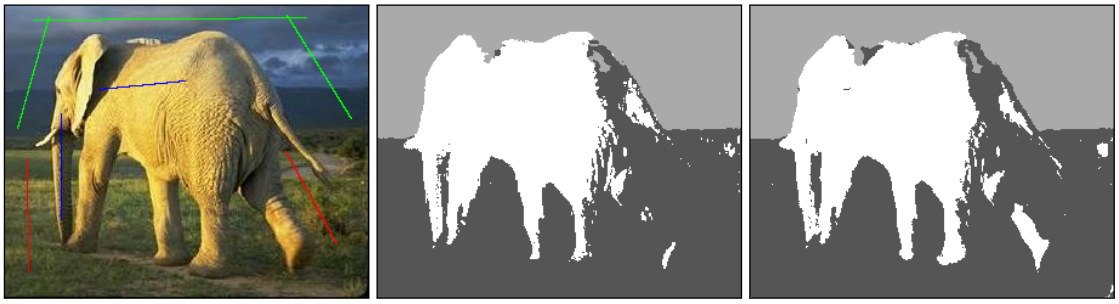
Figure 3.9 compares the results of the combined model ($\mathcal{H}\Delta_p$P) to the results of ($\mathcal{H}$P). The projection $\hat{h} = P\hat{u}$, computed for both $p = 1$ and $p = 2$, is quite similar to the result of the



(a) $\hat{u}, (p = 1)$ (b) $P\hat{u}$ $(p = 1)$ (c) $\hat{u}$ $(p = 2)$

(d) $P\hat{u}$ $(p = 2)$ (e) $\hat{g}$ from $(\mathcal{H}\mathrm{P})$

Figure 3.9.: The combined model $(\mathcal{H}\Delta_p\mathrm{P})$ in comparison to a solution of $(\mathcal{H}\mathrm{P})$. By adding the Laplacian regularization, the segmentation result $\hat{u}$ has smoother boundaries compared to $\hat{g}$ from the RKHS model. As expected $P\hat{u}$ is similar to the result $\hat{g}$ from the RKHS method. $\nu^{\mathrm{pho}} = 1,\ \nu^{\mathrm{lab}} = 0.2,\ \sigma_1^2 = 0.5,\ \sigma_2^2 = \infty,\ \vartheta = 4,\ \lambda = 0,\ \mu = 0.001$. a) $\hat{u}$ with $p = 1$. b) $\hat{h} = P\hat{u}$ with $p = 1$. c) $\hat{u}$ with $p = 2$. d) $\hat{h} = P\hat{u}$ with $p = 2$. e) $\hat{g}$ as solution of $(\mathcal{H}\mathrm{P})$.



(a) Input (b) $(\mathcal{H}\mathrm{P})$ (c) $(\mathcal{H}\Delta_1\mathrm{P})$, $\mu = 0.001$

Figure 3.10.: $(\mathcal{H}\Delta_1\mathrm{P})$ compared to $(\mathcal{H}\mathrm{P})$ for a natural image with $\nu^{\mathrm{pho}} = 0,\ \nu^{\mathrm{lab}} = 0.5,\ \sigma_1^2 = 1,\ \sigma_2^2 = 250,\ \vartheta = 2,\ \lambda = 0$.

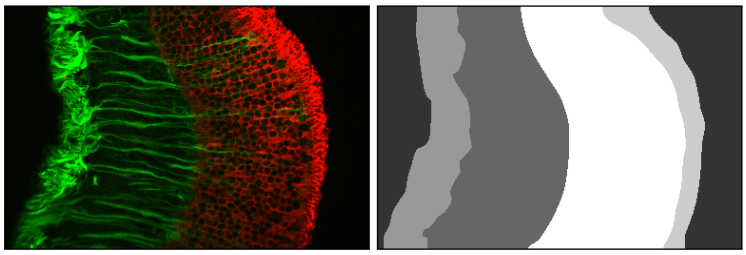


Figure 3.11.: Retina cross-section and ground truth.

RKHS approach ($\mathcal{H}$P) (not only the depicted segmentation, but also the segment membership function itself). On the other hand, $\hat{u}$ provides significantly smoother segmentations than $\hat{g}$ for both $p = 1$ and $p = 2$.

Figure 3.10 shows a comparison for a natural image between the result of ($\mathcal{H}\Delta_1$P) (discretization of $\hat{u}$) and the result of ($\mathcal{H}$P). The segmentation for $p = 2$ looks quite similar. Note that, besides smoother boundaries, a larger part of the leg is identified as part of the elephant. The combined model can improve the results, where the RKHS method is not smooth enough but more accurate than the p-Laplacian models, see also Figure 3.17 in the next section. The computation time for the combined model is a drawback of the current algorithm. However, there is a lot of potential to speed up computation, e.g., by parallelization.

3.4 Application to Medical Images

The segmentation of medical images is often very challenging. On the one hand, medical images suffer from noise and low contrast. On the other hand, many fine details have to be preserved. In contrast to our previous numerical experiments, we apply the segmentation models to a collection of similar images where only a single input image contains some points that are labeled in advance.

Figures 3.11 and 3.12 show a collection of cross-sections of the retina for layer segmentation, see [64, 63, 96]. The labels for the segmentation results in Figure 3.12 are randomly sampled from the ground truth in Figure 3.11. One can see that (Δ_pP) clearly outperforms ($\mathcal{H}$P). Beside the visual impression, this fact is substantiated by the segmentation accuracy of correctly classified pixels printed in the last column. This example indicates that (Δ_1P) can be an appropriate choice, if the ground truth of the segmentation has a relatively simple structure with smooth boundaries, while the input image contains a lot of inhomogeneous structure.

The next Figures 3.14 and 3.15 show cardiac MR heart images taken from [177]. The objective is to find the endocardial wall of both right and left ventricles (gray region), which is separated by epicardium walls (darker gray), while discarding the complicated background. This is a 3-phase segmentation: background, surrounding region and inner region. Only the first image is labeled as shown in Figure 3.13 and all the images in Figures 3.14 and 3.15 are segmented using the label information from the first image.

The images belong to a stack of the same heart. Hence, the segments are expected to be approximately in the same part of the image. This allows us to work with the spatial parameter

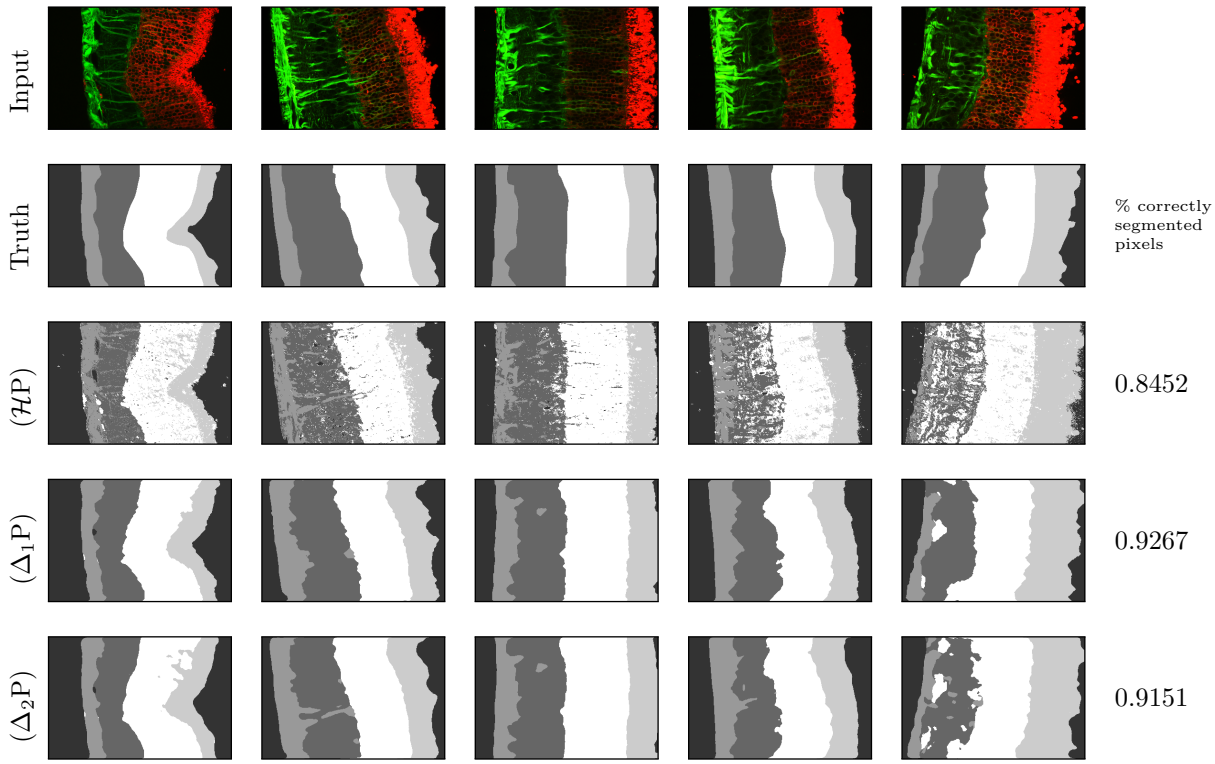


Figure 3.12.: Comparison of the investigated models applied to the layer segmentation of retina cross-section images. The percentage of correctly segmented pixels is averaged over all 5 results for each row. The parameter ρ_i was computed by (3.9). $\sigma_1^2 = \frac{1}{2}$, $\sigma_2^2 = \infty$, $\nu^{\text{pho}} = 0.001$, $\nu^{\text{lab}} = 0.01$, $\vartheta = 1$.

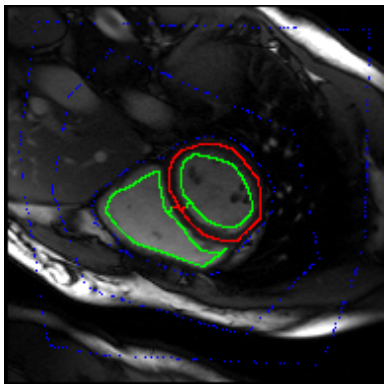
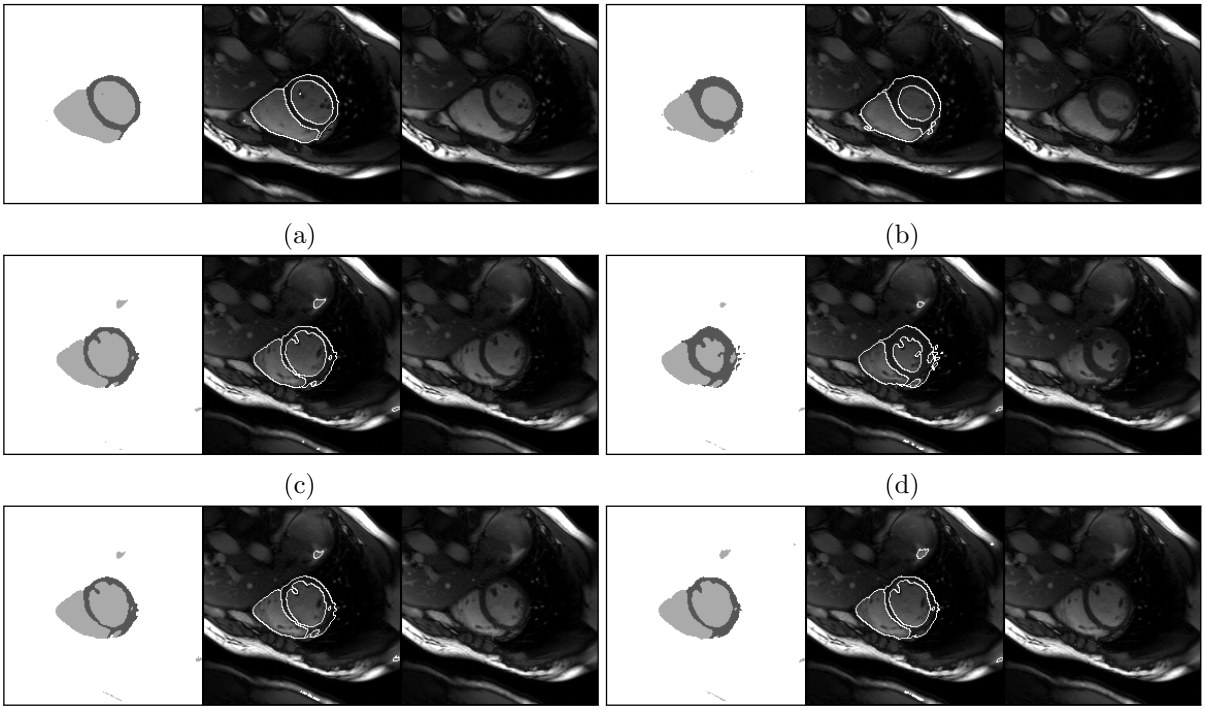
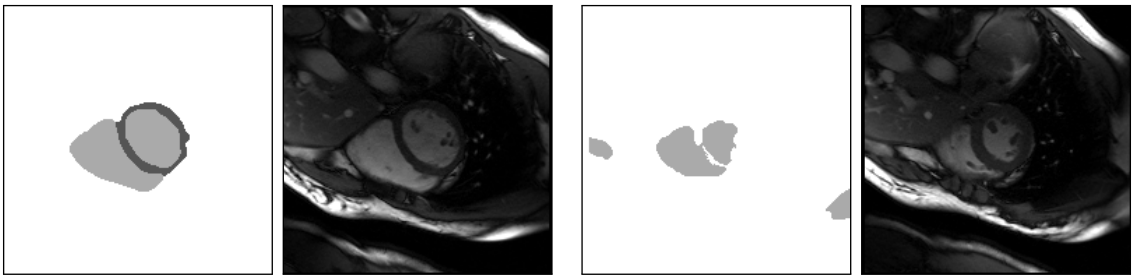


Figure 3.13.: Cardiac MR input image for the segmentation in Figures 3.14 and 3.15.



(a) (b) (c) (d) (e) (f)

Figure 3.14.: Solutions of the RKHS method applied to cardiac MR images with parameters $\sigma_1^2 = 1,\ \sigma_2^2 = 5,\ \vartheta = 3$ for six images. The left top image is the one from Figure 3.13. The segmentation result is shown in the two images next to the given image.



(a) (b)

Figure 3.15.: Solutions of the 1-Laplacian applied to cardiac MR images. $\rho^2 = s = 49,\ \nu^{\text{pho}} = 1\ \nu^{\text{lab}} = 0.1\ \vartheta = 3$.

$\sigma_2^2 < \infty$ for the kernel in ($\mathcal{H}$P). For this example, the parameter σ_2^2 is crucial to cast off background details since they are far away from the pixels labeled as foreground. Hence, we obtain satisfactory results with ($\mathcal{H}$P) as shown in Figure 3.14. Figure 3.15 shows results of (Δ_1P). We obtained similar results for $p = 2$. The labeled image itself is well segmented as depicted in Figure 3.15a. However, Figure 3.15b shows a similar image where (Δ_1P) fails to provide an adequate segmentation (the corresponding result of ($\mathcal{H}$P) is shown Figure 3.14d). This may be due to the fact that the unlabeled points U bring a lot of similarity details from the background into play which erroneously influences the resulting segmentation. Also introducing a spatial term in $\mathcal{N}^{\text{lab}}$ similar to the one activated by $\sigma_2^2 < \infty$ in ($\mathcal{H}$P) did not improve the results significantly. Note that in ($\mathcal{H}$P) the similarity of each pixel to all labeled pixels is used. In contrast, the p-Laplacian problems can computationally only handle a few weights to the labeled pixel resulting in less influence of such a spatial term.

	time for weight comp.	time algorithm
($\mathcal{H}$P)	24.7s	0.36s
(Δ_1P)	32.3s	6.7s
(Δ_2P)	28.7s	0.6s
($\mathcal{H}\Delta_1$P)	49s	196.5s
($\mathcal{H}\Delta_2$P)	45.5s	33.6s

Table 3.1.: Time comparison for single image segmentation using Figure 3.13 as input.

Table 3.1 shows time comparisons for the models of this chapter. The algorithms were implemented in Matlab and executed on an Intel Core i7 CPU with 2.93GHz. Our approaches require the computation of the weight or kernel matrix before executing the algorithms. We show time comparisons for both the weight computation and the algorithms. The first column states the approximate amount of time needed to create the weight or kernel matrices corresponding to the methods. The second column shows the approximate amount of time for the actual minimization process. For (Δ_1P) and ($\mathcal{H}\Delta_1$P), we have used the stopping criterion $\left\|(u^{(n)} - u^{(n-1)})\right\|_2 / \left\|u^{(n-1)}\right\|_2 < 0.001$. This table clearly shows that the RKHS approach is the most efficient with respect to computation time.

Figure 3.16 shows the input and the ground truth images of retina images for the results in Figure 3.17. As above, random elements from the ground truth in Figure 3.16 are used as labels for the segmentation of all similar input images shown in the top row of Figure 3.17. The second row shows the ground truths. The original images and the ground truth were taken from [148, 96]. The third row depicts the result of ($\mathcal{H}$P). Although the results are noisy, a lot of fine details are still visible. The fourth and fifth rows show that the results of the combined model ($\mathcal{H}\Delta_p$P) contain less noise than the result of ($\mathcal{H}$P) while keeping many details. The last two rows show the results of the p-Laplacian method (Δ_pP) for $p = 1, 2$. These segmentation results are the smoothest. Although very clean, many small details are missing. These visual effects are underpinned by the relative frequency of true and false positive pixels with respect to vessel pixels. More precisely, a pixel on the vessel classified as vessel pixel is true positive while a pixel classified as vessel pixel not lying on the vessel is false positive. Higher true positive values indicate the existence of fine details while higher false positive values come with more

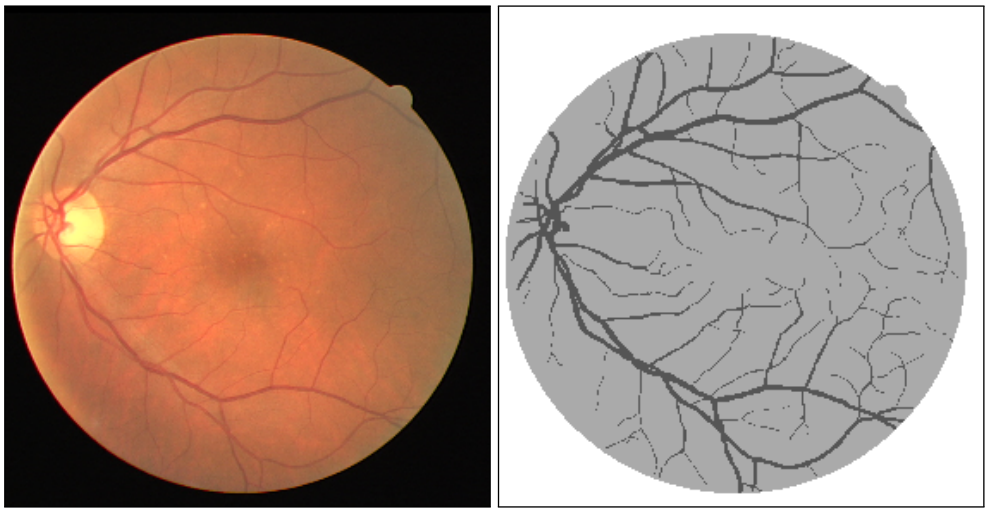


Figure 3.16.: Retina image and ground truth.

noise and artifacts in the image. Figure 3.18 shows enlargements of some of the images of the second column of Figure 3.17. Since the result of ($\mathcal{H}$P) keeps many of the fine details, one can also post-process these images for further denoising. To show the effect, we simply experimented with a median filter in Figure 3.19.

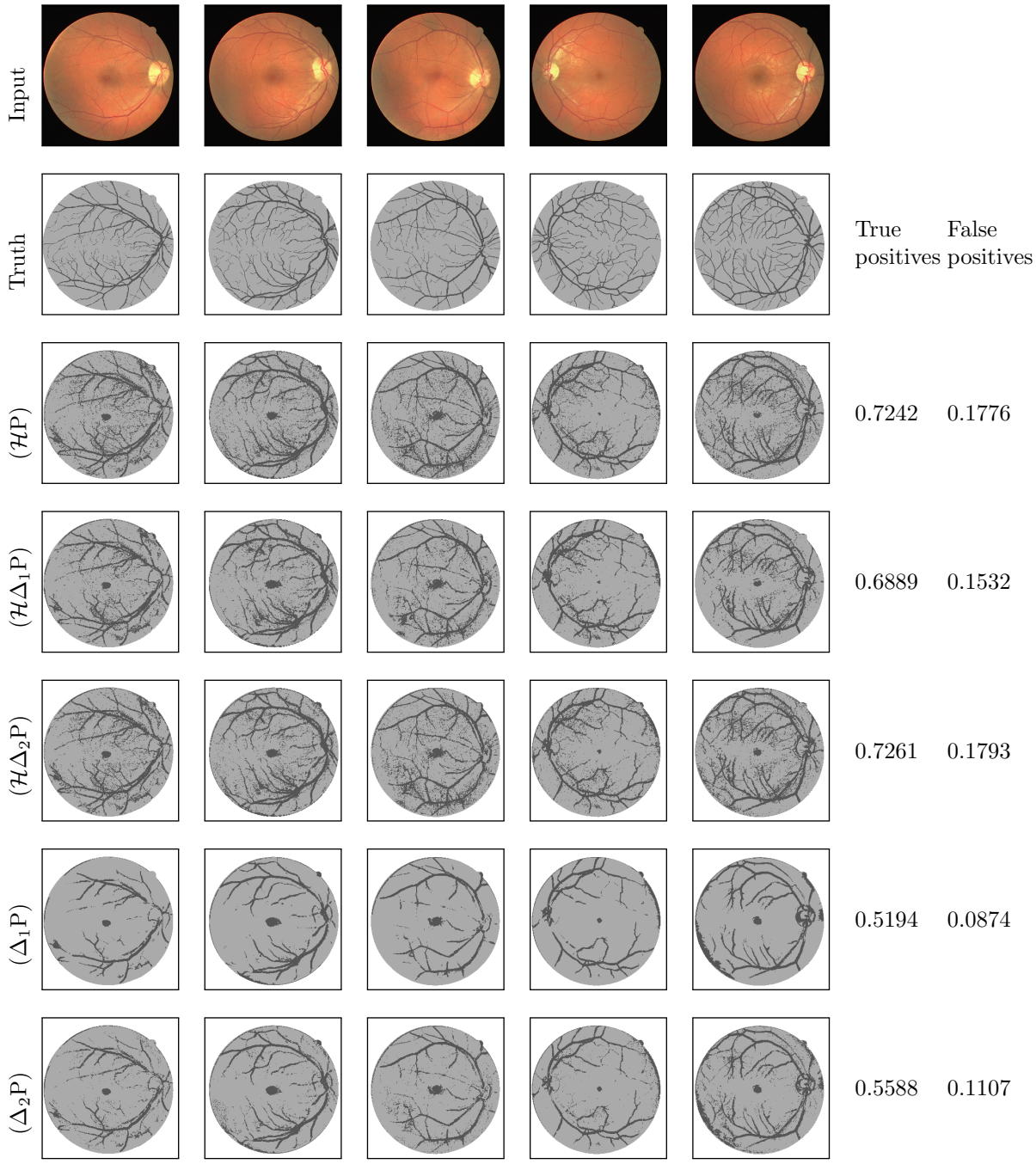
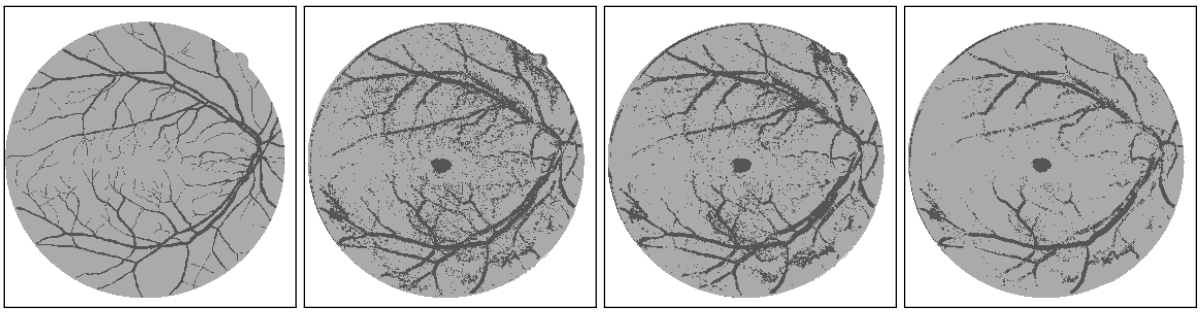
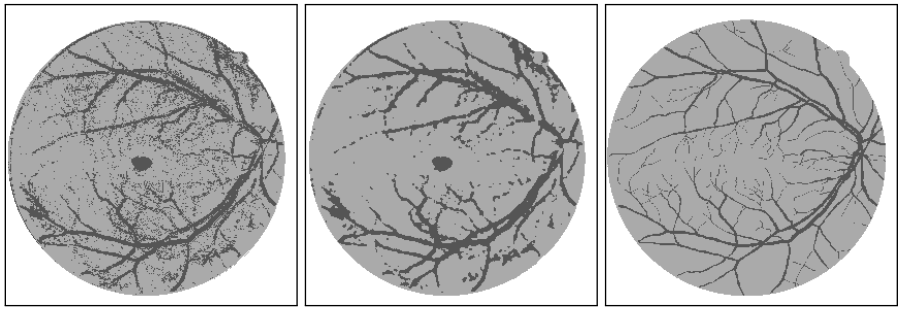


Figure 3.17.: Comparison of the investigated models applied to the vessel segmentation of retina images. The true and false positive values are determined with respect to the correct classification of vessel pixels and averaged over all five images in each row. For ($\mathcal{H}\Delta_1$P) we used $\mu = 0.0001$ and for ($\mathcal{H}\Delta_2$P) $\mu = 0.01$. The parameter ρ_i was computed by (3.9). $\sigma_1^2 = \frac{1}{2},\ \sigma_2^2 = \infty,\ \nu^{\text{pho}} = 1\ , \nu^{\text{lab}} = 0.8,\ \lambda = 0,\ \ \vartheta = 1.$



(a) Truth (b) ($\mathcal{H}$P) (c) ($\mathcal{H}\Delta_1$P) (d) (Δ_2P)

Figure 3.18.: These images show the results of ($\mathcal{H}$P) and ($\mathcal{H}\Delta_1$P) from the second column in Figure 3.17 in large. b) The result of ($\mathcal{H}$P) preserves more details. c) The result of ($\mathcal{H}\Delta_1$P) is less noisy compared to (b). d) (Δ_2P) is quite smooth but loses fine details. This effect is even stronger for (Δ_1P) for this image.



(a) ($\mathcal{H}$P) (b) Median filter (c) Truth

Figure 3.19.: Result of ($\mathcal{H}$P) and exemplary post-processing with a median filter of size 3×3.

Chapter 4

Conclusions and Perspectives

Unsupervised image segmentation is the application of clustering to image pixels. Therefore, we introduced center-based partitioning clustering in **Chapter 2**. We discussed the c-means algorithm for hard clustering and soft clustering with fuzzy c-means. Motivated by Teboulle [156], we derived the fuzzy c-means model as an approximation of hard clustering in terms of power means. Thereby, we presented a novel theorem on the representation of power means as linear functions over specific sets.

This approximation in terms of means was proposed by Teboulle, although with a different derivation. However, Teboulle did not show the equivalence to the fuzzy c-means model. Instead, he applied Weiszfeld's algorithm and showed the equivalence in terms of algorithms (not of models) disregarding the case where cluster centers r_k and data points $f(j)$ coincide. To fill the gap of coinciding data points and clusters in Weiszfeld's algorithm, one can consider the subdifferential instead of the gradient.

However, these simple clustering algorithms are not suitable for gray- or color-value based image segmentation since noise is ubiquitous in images. Hence, we added a smoothing term to our objective functional and obtained the general biconvex regularized segmentation model (RSP). For the solution of (RSP) we proposed the RcC algorithm that optimizes the assignment of pixels to segments in terms of the segment membership vector u and the codebook of segment centers r alternatingly. RcC contains many well known algorithms as particular implementations such as moving-average FCM [84], the method of Chan and Vese [42] with global optimization [115], and also the algorithms for unregularized clustering c-means and fuzzy c-means. Using Zangwill's Theory [172] for set-valued mappings, we developed a convergence statement for RcC.

Our convergence theorem says that a sequence of iterates of RcC either converges to a partial minimizer or – in the worst case – contains a convergent subsequence where the limit is again a partial minimizer. Moreover, the limit of every convergent subsequence is a partial minimizer. Although this seems good enough in practice[1], and this is the only convergence statement available for the much simpler fuzzy c-means algorithm [18] to our knowledge, it would be relieving to see that the sequence of iterates itself converges in every case.

Regarding theoretical future research, the generalization of the result of Nikolova et al. [115] stated in Theorem 2.27 to multiple segments is of great interest. This would imply that a

[1] We were not able to produce a single not convergent iteration in all experiments we actually executed with implementations of RcC.

discrete NP-hard problem can be solved by a continuous convex optimization problem. Hence it is not only interesting from a theoretical point of view. Also similar possibly weaker results could attract a lot of attention. Work in this direction can be found, e.g., in [100].

Further, we proposed the TVcM algorithm that aims to minimize the TV-regularized model (TVS) and is a particular implementation of RcC coming with the related convergence theory. The optimization step for a fixed codebook r is based on previous work to be found in [98, 171, 11, 120]. We demonstrated promising results in numerical examples. Our experiments contained real life 3d segmentation problems stemming from volumetric image data of different materials acquired by μCT. For one of these real life applications, namely the segmentation of C/SiC, we proposed an adaption of our model to make the results fulfill the material's property of separating layers.

Of course, our biconvex segmentation algorithm TVcM cannot guarantee global optimality. To this end, many approaches formulate equivalent optimization problems in higher dimensional spaces to obtain completely convex formulations of (TVS) or related models, see, e.g., [31, 10] described in Section 1.1.1. The main drawback of these approaches is the high computational effort. This effort can be reduced significantly by GPU implementations making completely convex models suitable for reasonably sized 2d images. Our applications arise from large volumetric data sets involving image sizes of several gigabytes. Therefore, we prefer our significantly faster biconvex approach. With these data sets even compute servers with large amounts of random access memory, e.g., over 50GB, reach their limits.

This problem already indicates two directions of future research. As mentioned in [31], one possible direction is the improvement of the completely convex models in terms of speed and memory consumption, e.g., by approaches similar to [66] described in Section 1.1.1. Another interesting direction of future research copes with the initialization of our biconvex model. So far, we have always used results of fuzzy c-means as initializations and in our experiments we obtained satisfying results. Of course, their still might be space for improvement. To find a good initialization in a cheap way could be the most practical approach, perhaps by applying a completely convex model on a very coarse scale.

In **Chapter 3** various methods for supervised multi-class segmentation were developed and studied. The work was inspired by the colorization method based on RKHSs in [71]. Beside the application of the method of Ha Quang et al. [71] to multi-class image segmentation, we have explored approaches related to p-Laplacians. Our investigations contain a novel combination of both approaches. The performance of our different supervised models was demonstrated and compared by segmenting distinct types of (medical) images.

Since p-Laplacians utilize similarity information between the unlabeled points, their application results in more regular segmentations. Roughly speaking, the 1-Laplacian model provides the smoothest results compared to larger values of $p > 1$. However, by the choice of the weights $w_{i,j}$ similar results for different p are possible.

The RKHS approach in ($\mathcal{H}$P) is the most efficient method regarding computational time, if only a few labeled pixels are given. For some complicated images, e.g., Figures 3.7 and 3.14, ($\mathcal{H}$P) excelled the Laplacian methods. Nevertheless, since it utilizes only the information from the labeled pixels, the results are often noisy and less regular.

We proposed a combined method that benefits from advantages of both approaches. This

novel model incorporates a projection to make it feasible for segmentation tasks.

Further research regarding the algorithm for the combined method yields potential for improvement in terms of speed and memory consumption.

Another open issue is the influence of the least squares functional of ($\mathcal{H}$P) compared to a support vector machine functional, especially for the combined model. For learning applications this has been considered in the recent work of Bresson and Zhang [27] for the model without projection stated in (3.17) with multiple classes and $p = 1$.

For future work, it would be interesting to investigate how truly vector-valued reproducing kernel Hilbert spaces [35, 109, 136], i.e., with a kernel matrix that is not block diagonal, can be exploited to improve supervised image segmentation tasks, perhaps with some prior knowledge at hand.

Chapter A

Appendix

The purpose of this chapter is to state some rather technical explanations and results that are excluded from the main part due to better readability.

A.1 Computational Aspects of d-Dimensional Image Segmentation

As in [140, Chapter A], in this section we consider some rather technical aspects of the d-dimensional setting, $d \geq 3$. More precisely, while most generalizations to higher dimensions in the algorithms are straightforward, the efficient solution of the linear system of equations (2.37) in the first ADMM step requires some explanation.

To this end, we consider d-dimensional images $F : \{1, \dots, n_1\} \times \cdots \times \{1, \dots, n_d\} \to \mathbb{R}$ and reshape them column-wise to a vector $f \in \mathbb{R}^N$ with $N := \prod_{i=1}^{d} n_i$. Let $D_n \in \mathbb{R}^{n \times n}$ be the forward difference matrix defined in (2.28). We introduce the *d-dimensional discrete gradient*

$$\boldsymbol{D} := \begin{pmatrix} \boldsymbol{D}_1 \\ \boldsymbol{D}_2 \\ \vdots \\ \boldsymbol{D}_d \end{pmatrix} \text{ with } \boldsymbol{D}_i := \bigotimes_{j=d}^{i+1} I_{n_j} \otimes D_{n_i} \otimes \bigotimes_{j=i-1}^{1} I_{n_j} = I_{\beta_i} \otimes D_{n_i} \otimes I_{\alpha_i},$$

where $\alpha_i := \prod_{j=1}^{i-1} n_j$, $\beta_i := \prod_{j=i+1}^{d} n_j$ and

$$\bigotimes_{i=d}^{1} M_i := M_d \otimes M_{d-1} \otimes \cdots \otimes M_2 \otimes M_1.$$

To solve the d-dimensional version of the linear system of equations (2.37), we use the following lemma.

Lemma A.1. *The matrix $\boldsymbol{D}^T \boldsymbol{D}$ can be diagonalized by tensor products of DCT-II matrices C_n*

defined in (2.39) *by*

$$\boldsymbol{D}^T\boldsymbol{D} = \left(\bigotimes_{j=d}^{1} C_{n_j}^\top\right) \Lambda \left(\bigotimes_{j=d}^{1} C_{n_j}\right)$$

where

$$\Lambda := \sum_{i=1}^{d} \left(I_{\beta_i} \otimes \ \mathrm{diag}(q_{n_i}) \ \otimes I_{\alpha_i}\right), \quad q_n := \left(4\sin^2\frac{\pi j}{2n}\right)_{j=0}^{n-1}.$$

Proof. Using (2.36) and (2.38), we obtain

$$\boldsymbol{D}^\top\boldsymbol{D} = (\boldsymbol{D}_1^\top\boldsymbol{D}_1 + \ldots \ + \boldsymbol{D}_d^\top\boldsymbol{D}_d) = \sum_{i=1}^{d} \left(I_{\beta_i} \otimes D_{n_i}^\top D_{n_i} \otimes I_{\alpha_i}\right) = \sum_{i=1}^{d} \left(I_{\beta_j} \otimes C_{n_i}^\top \ \mathrm{diag}(q_{n_i}) \ C_{n_i} \otimes I_{\alpha_j}\right).$$

By the orthogonality of the cosine matrix $I_{n_i} = C_{n_i}^\top C_{n_i}$ and (2.36) we get further

$$\begin{aligned}\boldsymbol{D}^\top\boldsymbol{D} &= \sum_{i=1}^{d} \Big(\bigotimes_{j=d}^{i+1} C_{n_j}^\top C_{n_j}\Big) \otimes \left(C_{n_i}^\top \mathrm{diag}(q_{n_i}) C_{n_i}\right) \otimes \Big(\bigotimes_{j=i-1}^{1} C_{n_j}^\top C_{n_j}\Big) \\ &= \sum_{i=1}^{d} \underbrace{\left(C_{n_d}^\top \otimes C_{n_{d-1}}^\top\right)\left(C_{n_d} \otimes C_{n_{d-1}}\right)}_{C_{n_d}^\top C_{n_d} \otimes C_{n_{d-1}}^\top C_{n_{d-1}}} \otimes \Big(\bigotimes_{j=d-2}^{i+1} C_{n_j}^\top C_{n_j}\Big) \otimes \left((C_{n_i})^\top \ \mathrm{diag}(q_{n_i}) \ C_{n_i}\right) \otimes \Big(\bigotimes_{j=i-1}^{1} C_{n_j}^\top C_{n_j}\Big).\end{aligned}$$

Applying the previous step also to the last term and repeating it iteratively we obtain

$$\boldsymbol{D}^\top\boldsymbol{D} = \sum_{i=1}^{d} \Big(\bigotimes_{j=d}^{i+1} C_{n_j}^\top\Big)\Big(\bigotimes_{j=d}^{i+1} C_{n_j}\Big) \ \otimes \ \left(C_{n_i}^\top \mathrm{diag}(q_{n_i}) \ C_{n_i}\right) \ \otimes \Big(\bigotimes_{j=i-1}^{1} C_{n_j}^\top\Big)\Big(\bigotimes_{j=i-1}^{1} C_{n_j}\Big).$$

Using (2.36) results in

$$\begin{aligned}\boldsymbol{D}^\top\boldsymbol{D} &= \sum_{i=1}^{d} \underbrace{\Big(\bigotimes_{j=d}^{i} C_{n_j}^\top\Big)}_{A} \underbrace{\Big(\Big(\bigotimes_{j=d}^{i+1} C_{n_j}\Big) \otimes \ \mathrm{diag}(q_{n_i}) \ C_{n_i}\Big)}_{B} \otimes \underbrace{\Big(\bigotimes_{j=i-1}^{1} C_{n_j}^\top\Big)}_{C} \underbrace{\Big(\bigotimes_{j=i-1}^{1} C_{n_j}\Big)}_{D} \\ &= \sum_{i=1}^{d} \Big(\bigotimes_{j=d}^{1} C_{n_j}^\top\Big) \left(\Big(\bigotimes_{j=d}^{i+1} C_{n_j}\Big) \otimes \ \mathrm{diag}(q_{n_i}) \ C_{n_i} \otimes \Big(\bigotimes_{j=i-1}^{1} C_{n_j}\Big)\right) \\ &= \sum_{i=1}^{d} \Big(\bigotimes_{j=d}^{1} C_{n_j}^\top\Big) \left(I_{\beta_i}\Big(\bigotimes_{j=d}^{i+1} C_{n_j}\Big) \otimes \ \mathrm{diag}(q_{n_i}) \ C_{n_i} \otimes I_{\alpha_i}\Big(\bigotimes_{j=i-1}^{1} C_{n_j}\Big)\right) \\ &= \sum_{i=1}^{d} \Big(\bigotimes_{j=d}^{1} C_{n_j}^\top\Big) \left(I_{\beta_i} \otimes \ \mathrm{diag}(q_{n_i}) \ \otimes I_{\alpha_i}\right) \Big(\bigotimes_{j=d}^{1} C_{n_j}\Big).\end{aligned}$$

□

By the lemma, the d-dimensional version of (2.40) is given by

$$u_k = \left[\left(\bigotimes_{j=d}^{1} C_{n_j}^\top \right) (\Lambda + I_N)^{-1} \left(\bigotimes_{j=d}^{1} C_{n_j} \right) \right] a_k. \tag{A.1}$$

Tensor products of matrices are useful to write the models in a matrix-vector form. However, in our numerical computations we will not work with tensor products. In the following, we describe how this can be avoided. We start with the case $d = 2$. For 2-dimensional images we consider

$$(\boldsymbol{D}^\top \boldsymbol{D} + I_N)^{-1} f = \left(C_{n_2}^\top \otimes C_{n_1}^\top \right) (\Lambda + I_N)^{-1} \left(C_{n_2} \otimes C_{n_1} \right) f.$$

By (2.41) we have

$$\left(C_{n_2} \otimes C_{n_1} \right) f = \operatorname{vec} \left(C_{n_1} F C_{n_2}^\top \right).$$

Denoting the component-wise multiplication (Hadamard product) of two matrices by $\circ$ we obtain

$$(\Lambda + I_N)^{-1} \operatorname{vec} \left(C_{n_1} F C_{n_2}^\top \right) = \operatorname{vec} \left(L \circ \left(C_{n_1} F C_{n_2}^\top \right) \right)$$

where $L \in \mathbb{R}^{n_1 \times n_2}$ is the column-wise reshaping of the diagonal of $(\Lambda + I_N)^{-1}$ into a matrix. Applying (2.41) again leads to

$$\begin{aligned} (\boldsymbol{D}^\top \boldsymbol{D} + I_N)^{-1} f &= \left(C_{n_2}^\top \otimes C_{n_1}^\top \right) \operatorname{vec} \left(L \circ \left(C_{n_1} F C_{n_2}^\top \right) \right) \\ &= \operatorname{vec} \left(C_{n_1}^\top L \circ \left(C_{n_1} \cdot F \cdot C_{n_2}^\top \right) C_{n_2} \right). \end{aligned} \tag{A.2}$$

For the general d-dimensional case we have implemented the right-hand side of (A.1) without constructing the tensor products explicitly by applying the following lemma.

Lemma A.2. *Let $f \in \mathbb{R}^N$ with $N = n_1 \dots n_d$ be given. We set $N_k := n_1 \dots n_k$, $k = 1, \dots, d$. Starting with $f^{k_d} := f$, $k_d = 1$, we define successively:*

- $F^{k_d} := \operatorname{vec}^{-1}_{N_{d-1}, n_d}(f^{k_d}) \in \mathbb{R}^{N_{d-1} \times n_d}$,
- $f^{k_d, k_{d-1}}$ *is the k_{d-1}-th column of F^{k_d}, $k_{d-1} = 1, \dots, n_d$,*
- $F^{k_d, k_{d-1}} := \operatorname{vec}^{-1}_{N_{d-2}, n_{d-1}}(f^{k_d, k_{d-1}}) \in \mathbb{R}^{N_{d-2} \times n_{d-1}}$
- *and so on.*

Then, for $j = 2, \dots, d$ and $A_i \in \mathbb{R}^{n_i \times n_i}$ the following relation holds true:

$$\left(\bigotimes_{i=j}^{1} A_i \right) f^{k_d, \dots, k_j} = \operatorname{vec}_{N_{j-1}, n_j} \left(\left(\bigotimes_{i=j-1}^{1} A_i \right) F^{k_d, \dots, k_j} A_j^\top \right).$$

Proof. Apply (2.41) to every column k_j for every j. □

A.2 The Proximum of Positively Homogeneous Functions

In this Section we first introduce some preliminaries regarding support functions of positively homogeneous functions g, i.e., $g(\lambda x) = \lambda g(x)$ for $\lambda > 0$. Then, we briefly discuss the proximum operator and explain thereby how to derive the coupled shrinkage in Equation (2.48). For more details we refer to textbooks such as [128, 17, 23].

Preliminaries The *proximum* $\mathrm{Prox} : \mathbb{R}^n \to \mathbb{R}^n$ of a proper, convex, and lsc function $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is defined by

$$\mathrm{Prox}_{\lambda g}(b) := \operatorname*{argmin}_{x \in \mathbb{R}^n} \frac{1}{2} \|x - b\|_2^2 + \lambda g(x).$$

Further, the *conjugate function* of g is defined by

$$g^* : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\},\ g^*(p) = \sup_x (\langle x, p \rangle - g(x))$$

and $g^{**} = g$.

For the following computation, we need the subsequent simple rules. Let $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, $b \in \mathbb{R}^n$ and $\lambda \in \mathbb{R} \setminus \{0\}$. Then the following relations hold true:

- $(g(\lambda \cdot))^*(p) = g^*(\frac{1}{\lambda} p)$, (A.3)
- $(\lambda g)^*(p) = \lambda g^*(\frac{1}{\lambda} p)$. (A.4)

In the next step, we compute the conjugate function of a positively homogeneous function g. With $\lambda > 0$, (A.3), and (A.4) we obtain

$$g^*(p) := \sup_x \langle x, p \rangle - g(x) = \sup_x \langle x, p \rangle - \lambda g(x/\lambda) = (\lambda g(\cdot/\lambda))^*(p) \underset{\text{(A.3)}}{=} (\lambda g)^*(p\lambda) \underset{\text{(A.4)}}{=} \lambda g^*(p)$$

such that $g^*(p) = 0$ or $g^*(p) = \infty$. Let

$$C_g := \{p \in \mathbb{R}^n : \langle p, x \rangle \leq g(x) \text{ for all } x \in \mathbb{R}^n\}.$$

Of course, $p \in C_g$ implies

$$g^*(p) = \sup_x \underbrace{\langle x, p \rangle - g(x)}_{\leq 0} = 0,$$

since $g^*(p) \in \{0, \infty\}$. On the other hand, from $p \notin C_g$, i.e., $\langle p, x \rangle > g(x)$, follows for $x = \lambda p$ and $\lambda > 0$ that

$$\sup_\lambda \underbrace{\langle \lambda p, p \rangle - g(\lambda p)}_{>0} = \sup_\lambda \lambda (\underbrace{\langle p, p \rangle - g(p))}_{>0} = \infty = g^*(p).$$

Hence, $g^* = \iota_{C_g}$ where

$$\iota_S(x) := \begin{cases} 0 & \text{if } x \in S \\ \infty & \text{if } x \notin S \end{cases}$$

denotes the indicator of function of the set S. Further, one can see immediately that $\iota_S^*(p) = \sup_x \langle x, p\rangle - \iota_S(x) = \sup_{x\in S}\langle x, p\rangle =: \sigma_S(p)$ where σ is called *support function*. Hence, $g = \sigma_S$.

In case $g = \|\cdot\|$ is an arbitrary norm in $\mathbb{R}^n$ and

$$\|x\|_* := \sup_{\|y\|\leq 1} \langle x, y\rangle = \max_{\|y\|\leq 1} \langle x, y\rangle = \max_{\|y\|\neq 0} \langle x, \frac{y}{\|y\|}\rangle$$

denotes its dual norm we can compute $C_{\|\cdot\|}$ explicitly and obtain

$$\begin{aligned} C_{\|\cdot\|} &= \{p \in \mathbb{R}^n : \langle p, x\rangle \leq \|x\| \text{ for all } x\} = \{p \in \mathbb{R}^n : \langle p, \frac{x}{\|x\|}\rangle \leq 1 \text{ for all } x \neq 0\} \\ &= \{p \in \mathbb{R}^n : \max_{x\neq 0} \langle p, \frac{x}{\|x\|}\rangle \leq 1\} = \{p \in \mathbb{R}^n : \|p\|_* \leq 1\} =: B_{\|\cdot\|_*}(0, 1). \end{aligned}$$

For extensions of particular p-means to $\mathbb{R}^n$ the explicit form of C is stated in Section 2.2.1.

Solution via Conjugate Duality We are interested in the solution of

$$\operatorname{Prox}_{\lambda g}(b) := \operatorname*{argmin}_{x\in\mathbb{R}^n} \frac{1}{2}\|x - b\|_2^2 + \lambda g(x). \tag{A.5}$$

Since $\lambda g(x) = \sup_p \langle p, x\rangle - \lambda \iota_{C_{\lambda g}}(p) = \sup_p \langle p, x\rangle - \iota_{C_{\lambda g}}(p)$, we can rewrite (A.5) as equivalent primal (P) and dual (D) problems

$$\begin{aligned} (P) \quad & \operatorname*{argmin}_{x\in\mathbb{R}^n} \max_{p\in\mathbb{R}^n} \frac{1}{2}\|x - b\|_2^2 - \iota_{C_{\lambda g}}(p) + \langle p, x\rangle, \\ (D) \quad & \operatorname*{argmax}_{p\in\mathbb{R}^n} \min_{x\in\mathbb{R}^n} \frac{1}{2}\|x - b\|_2^2 - \iota_{C_{\lambda g}}(p) + \langle p, x\rangle. \end{aligned} \tag{A.6}$$

Computing

$$b - p = \hat{x} = \operatorname*{argmin}_{x\in\mathbb{R}^n} \frac{1}{2}\|x - b\|_2^2 - \iota_{C_{\lambda g}}(p) + \langle p, x\rangle$$

yields for the dual problem

$$(D) \quad \operatorname*{argmax}_{p\in\mathbb{R}^n} \langle p, b\rangle - \frac{1}{2}\|p\|_2^2 - \iota_{C_{\lambda g}}(p)$$

and equivalently

$$(D) \quad \operatorname*{argmin}_{p\in\mathbb{R}^n} \frac{1}{2}\|p\|_2^2 - \langle p, b\rangle + \iota_{C_{\lambda g}}(p).$$

Since the minimizers of $\frac{1}{2}\|\cdot\|_2^2 - \langle\cdot, b\rangle$ and $\frac{1}{2}\|\cdot - b\|$ for fixed b are equal, we obtain

$$(D) \quad \operatorname*{argmin}_{p\in\mathbb{R}^n} \frac{1}{2}\|p - b\|_2^2 + \iota_{C_{\lambda g}}(p).$$

The minimizer of the dual problem $\hat{p}$ can be obtained by an orthogonal projection onto $C_{\lambda g}$. For given $\hat{p}$ the solution of the primal problem (P) can be obtained by setting the gradient of (A.6) to zero yielding $\hat{x} = b - \hat{p}$.

In case of the coupled shrinkage in Equation (2.48), we just have to compute the projection $\hat{p}$ onto the ball $B_{\|\cdot\|_2}(0, \lambda) = C_{\lambda g} \subset \mathbb{R}^2$ for every pixel and take $b - \hat{p}$ for given b.

Bibliography

[1] G. Alberti, G. Bouchitté, and G. Dal Maso. The calibration method for the Mumford-Shah functional and free-discontinuity problems. *Calculus of Variations and Partial Differential Equations*, 16(3):299–333, 2003. (Cited on page 15.)

[2] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009. (Cited on page 39.)

[3] L. Ambrosio and V. Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. *Communications on Pure and Applied Mathematics*, 43(8): 999–1036, 1990. (Cited on page 15.)

[4] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of Bounded Variation and Free Discontinuity Problems*. Clarendon Press Oxford, 2000. (Cited on pages 14, 15, and 64.)

[5] S. Amghibech. Eigenvalues of the discrete p-Laplacian for graphs. *Ars Combinatoria*, 67: 283–302, 2003. (Cited on page 28.)

[6] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander. Optics: ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2):49–60, 1999. (Cited on page 36.)

[7] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950. (Cited on page 29.)

[8] N. W. Ashcroft and D. N. Mermin. *Solid State Physics*. Brooks Cole, 1st edition, 1976. (Cited on page 101.)

[9] E. Bae, J. Yuan, X.-C. Tai, and Y. Boykov. A study of continuous max-flow and min-cut approaches, part ii: Multiple linearly ordered labels. CAM Report 10-62, University of California, Los Angeles, 2010. (Cited on page 91.)

[10] E. Bae, J. Yuan, and X. Tai. Simultaneous convex optimization of regions and region parameters in image segmentation models. In *Dagstuhl Seminar Proceedings. Springer*, 2011. (Cited on pages 20, 21, 23, 64, and 134.)

[11] E. Bae, J. Yuan, and X.-C. Tai. Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision*, 92(1): 112–129, 2011. (Cited on pages 21, 22, 30, 32, 64, and 134.)

[12] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imgaging Science*, 2(1):183–202, 2009. (Cited on pages 24 and 66.)

[13] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled und unlabeled examples. *Journal of Macine Learning Research*, 7:2399–2434, 2006. (Cited on pages 30, 31, 107, and 121.)

[14] A. Ben-Tal and M. Teboulle. Expected utility, penalty functions, and duality in stochastic nonlinear programming. *Management Science*, 32(11):1445–1466, 1986. (Cited on page 43.)

[15] L. Bertelli, T. Yu, D. Vu, and B. Gokturk. Kernelized structural SVM learning for supervised object segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, 2011. (Cited on page 30.)

[16] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999. (Cited on page 38.)

[17] D. P. Bertsekas, A. Nedi, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003. (Cited on page 139.)

[18] J. Bezdek, R. Hathaway, M. Sabin, and W. Tucker. Convergence theory for fuzzy c-means: Counterexamples and repairs. *IEEE Transactions on Systems, Man, and Cybernetics*, 17 (5):873–877, 1987. (Cited on pages 34, 35, 37, 40, 50, 78, and 133.)

[19] J. C. Bezdek. A convergence theorem for the fuzzy ISODATA clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2:1–8, January 1980. (Cited on page 61.)

[20] J. C. Bezdek, R. Ehrlich, and W. Full. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203, 1984. (Cited on pages 30, 34, 35, 40, 41, 49, and 78.)

[21] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. (Cited on page 39.)

[22] A. Blake, P. Kohli, and C. Rother, editors. *Markov random fields for vision and image processing*. MIT Press, 2011. (Cited on page 26.)

[23] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer, 2006. (Cited on page 139.)

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011. (Cited on pages 24, 66, and 123.)

[25] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. (Cited on page 26.)

[26] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. (Cited on page 13.)

[27] X. Bresson and R. Zhang. Tv-svm: Total variation support vector machine for semi-supervised data classification. *arXiv:1210.0699*, 2012. (Cited on pages 31, 107, 122, and 135.)

[28] X. Bresson, X.-C. Tai, T. Chan, and A. Szlam. Multi-class transductive learning based on 1 relaxations of Cheeger cut and Mumford-Shah-Potts model. CAM Report 12-03, University of California, Los Angeles, 2012. (Cited on pages 31, 106, and 111.)

[29] E. S. Brown, T. F. Chan, and X. Bresson. A convex approach for multi-phase piecewise constant Mumford-Shah image segmentation. CAM Report 09-66, University of California, Los Angeles, 2009. (Cited on page 20.)

[30] E. S. Brown, T. F. Chan, and X. Bresson. A convex relaxation method for a class of vector-valued minimization problems with applications to Mumford-Shah segmentation. CAM Report 10-43, University of California, Los Angeles, 2010. (Cited on page 18.)

[31] E. S. Brown, T. F. Chan, and X. Bresson. Completely convex formulation of the Chan-Vese image segmentation model. *International Journal of Computer Vision*, 98(1):103–121, 2012. (Cited on pages 18, 19, 20, 21, 23, 64, and 134.)

[32] T. Brox and D. Cremers. On the statistical interpretation of the piecewise smooth Mumford-Shah functional. In *Proceedings of the* 1^{st} *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM '07)*, pages 203–213, 2007. (Cited on page 15.)

[33] T. Bühler and M. Hein. Spectral clustering based on the graph p-Laplacian. In *Proceedings of the* 26^{th} *International Conference on Machine Learning (ICML '09)*, 2009. (Cited on page 28.)

[34] X. Cai, R. Chan, and T. Zeng. A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding. *SIAM Journal on Imaging Sciences*, 6(1):368–390, 2013. (Cited on page 15.)

[35] C. Carmeli, E. De Vito, and A. Toigo. Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem. *Analysis and Applications*, 4(4):377, 2006. (Cited on pages 28 and 135.)

[36] A. Chambolle. Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations. *SIAM Journal on Applied Mathematics*, 55(3):827–863, 1995. (Cited on page 15.)

[37] A. Chambolle. Finite-differences discretizations of the Mumford-Shah functional. *Mathematical Modelling and Numerical Analysis*, 33(02):261–288, 1999. (Cited on page 15.)

[38] A. Chambolle. Total variation minimization and a class of binary MRF models. In *Proceedings of the* 5^{th} *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR '05)*, pages 136–152, 2005. (Cited on page 17.)

[39] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011. (Cited on pages 24, 32, 67, 83, and 111.)

[40] A. Chambolle, D. Cremers, and T. Pock. A convex approach for computing minimal partitions. Technical report, Ecole Polytechnique, Paris, 2008. (Cited on pages 23 and 91.)

[41] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, and T. Pock. An introduction to total variation for image analysis. Preprint hal-00437581, Centre pour la Communication Scientifique Directe, 2009. (Cited on pages 17, 21, 35, 53, and 75.)

[42] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. (Cited on pages 6, 16, 17, 30, 32, 35, 53, 65, 73, 74, 75, and 133.)

[43] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised learning*, volume 2. MIT press Cambridge, 2006. (Cited on page 106.)

[44] C. Chaux, A. Jezierska, J.-C. Pesquet, and H. Talbot. A spatial regularization approach for vector quantization. *Journal of Mathematical Imaging and Vision*, 41:23–38, 2011. (Cited on pages 26 and 64.)

[45] R. Ciak, B. Shafei, and G. Steidl. Homogeneous penalizers and constraints in convex image restoration. *Journal of Mathematical Imaging and Vision*, 47(3):210–230, 2013. (Cited on page 8.)

[46] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, G. B. Pike, and A. C. Evans. Brainweb: Online interface to a 3d MRI simulated brain database. *NeuroImage*, 5:425, 1997. URL `http://www.bic.mni.mcgill.ca/brainweb/`. (Cited on pages 78 and 87.)

[47] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans. Design and construction of a realistic digital brain phantom. *IEEE Transactions on Medical Imaging*, 17(3):463–468, 1998. (Cited on pages 78 and 87.)

[48] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal of Computer Vision*, 50:295–313, 2002. (Cited on page 13.)

[49] J. De Leeuw. Block relaxation algorithms in statistics. *Information systems and data analysis*, pages 308–325, 1994. (Cited on page 54.)

[50] A. Delong and Y. Boykov. Globally optimal segmentation of multi-region objects. In *Proceedings of the* 12^{th} *IEEE International Conference on Computer Vision (ICCV 09)*, pages 285–292, 2009. (Cited on page 91.)

[51] W. Donath and A. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973. (Cited on page 24.)

[52] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In *Proceedings of the* 25^{th} *International Conference on Machine Learning (ICML '08)*, ACM New York, 2008. (Cited on pages 70 and 112.)

[53] J. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973. (Cited on page 40.)

[54] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992. (Cited on pages 24, 66, and 123.)

[55] P. Elias, A. Feinstein, and C. Shannon. A note on the maximum flow through a network. *IRE Transactions on Information Theory*, 2(4):117–119, december 1956. (Cited on pages 24 and 25.)

[56] J. E. Esser. *Primal Dual Algorithms for Convex Models and Applications to Image Restoration, Registration and Nonlocal Inpainting.* PhD thesis, University of California, Los Angeles, 2010. (Cited on pages 24, 66, 67, 111, and 123.)

[57] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the* 2^{nd} *International Conference on Knowledge Discovery and Data mining*, volume 1996, pages 226–231. AAAI Press, 1996. (Cited on page 36.)

[58] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2): 298–305, 1973. (Cited on page 24.)

[59] W. H. Fleming and R. Rishel. An integral formula for total gradient variation. *Archiv der Mathematik*, 11:218–222, 1960. (Cited on pages 19 and 75.)

[60] L. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399–404, 1956. (Cited on pages 24 and 25.)

[61] Fraunhofer ITWM, Department of Image Processing. MAVI – modular algorithms for volume images, 2005. URL `http://www.mavi-3d.de`. (Cited on page 105.)

[62] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary Value Problems*, chapter IX, pages 299–340. North-Holland, Amsterdam, 1983. (Cited on pages 24, 35, and 66.)

[63] E. Gelasca, B. Obara, D. Fedorov, K. Kvilekval, and B. Manjunath. A biosegmentation benchmark for evaluation of bioimage analysis methods. *BMC Bioinformatics*, 10:368, 2009. (Cited on page 126.)

[64] E. D. Gelasca, J. Byun, B. Obara, and B. Manjunath. Evaluation and benchmark for biological image segmentation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP '08)*, 2008. (Cited on page 126.)

[65] E. Giusti. *Minimal surfaces and functions of bounded variation.* Birkhäuser, 1984. (Cited on pages 15 and 64.)

[66] B. Goldlücke and D. Cremers. Convex relaxation for multilabel problems with product label spaces. In *Proceedings of the* 11^{th} *European Conference on Computer Vision (ECCV '10)*, pages 225–238, 2010. (Cited on pages 23, 24, and 134.)

[67] T. Goldstein and S. Osher. The split Bregman method for L_1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009. (Cited on pages 24 and 66.)

[68] T. Goldstein, X. Bresson, and S. Osher. Global minimization of Markov random fields with applications to optical flow. CAM Report 09-77, University of California, Los Angeles, 2009. (Cited on page 18.)

[69] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions - a survey and extensions. *Mathematical Methods of Operations Research*, 66(3): 373–407, 2007. (Cited on pages 37, 38, 54, and 83.)

[70] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. URL `http://www.vision.caltech.edu/Image_Datasets/Caltech256/`. (Cited on page 120.)

[71] M. Ha Quang, S. H. Kang, and T. M. Le. Image and video colorization using vector-valued reproducing kernel hilbert spaces. *Journal of Mathematical Imaging and Vision*, 37:49–65, 2010. (Cited on pages 29, 31, 32, 107, 117, 118, and 134.)

[72] L. Hagen and A. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9): 1074–1085, 1992. (Cited on page 26.)

[73] S. Har-Peled and B. Sadri. How fast is the k-means method? *Algorithmica*, 41(3):185–202, 2005. (Cited on page 39.)

[74] A. Hardy. On the number of clusters. *Computational Statistics & Data Analysis*, 23(1): 83–96, 1996. (Cited on page 36.)

[75] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, 2nd edition, February 1988. (Cited on pages 41, 42, 45, and 46.)

[76] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009. (Cited on pages 27, 30, and 39.)

[77] R. Hathaway, J. Bezdek, and W. Tucker. An improved convergence theory for the fuzzy c-means clustering algorithms. *Analysis of Fuzzy Information*, 3:123–131, 1987. (Cited on pages 30, 32, 40, 50, and 53.)

[78] S. Häuser and G. Steidl. Convex multiclass segmentation with shearlet regularization. *International Journal of Computer Mathematics*, 90(1):62–81, 2013. (Cited on page 23.)

[79] Y. He, M. Y. Hussaini, J. Ma, B. Shafei, and G. Steidl. A fuzzy c-means method with total variation regularization for image segmentation. *Pattern Recognition*, 45(9):3463–3471, 2012. (Cited on pages 8, 32, 35, 64, 65, 70, 71, 72, 76, 78, 83, and 87.)

[80] M. Hein and T. Bühler. An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. In *Proceedings of Advances in Neural Information Processing Systems 23 (NIPS '10)*, 2010. (Cited on page 28.)

[81] M. Hein and S. Setzer. Beyond spectral clustering – tight relaxations of balanced graph cuts. In *Proceedings of Advances in Neural Information Processing Systems (NIPS '11)*, pages 2366–2374, 2011. (Cited on page 28.)

[82] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952. (Cited on page 123.)

[83] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, New York, 1991. (Cited on page 110.)

[84] Z. Hou, W. Qian, S. Huang, Q. Hu, and W. L. Nowinski. Regularized fuzzy c-means method for brain tissue clustering. *Pattern Recognition Letters*, 28:1788–1794, 2007. (Cited on pages 30, 35, 76, 77, 78, and 133.)

[85] H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1333–1336, 2003. (Cited on pages 18 and 91.)

[86] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31:253–258, 1925. (Cited on page 25.)

[87] S. H. Kang, B. Sandberg, and A. M. Yip. A regularized k-means and multiphase scale segmentation. *Inverse Problems and Imaging*, 5(2):407–429, 2011. (Cited on pages 18 and 36.)

[88] S. H. Kang, B. Shafei, and G. Steidl. Supervised and transductive multi-class segmentation using p-Laplacians and RKHS methods, 2012. Submitted. (Cited on pages 8, 32, 106, 107, and 122.)

[89] S. H. Kang, B. Shafei, and G. Steidl. Supervised and transductive multi-class segmentation using p-Laplacians and RKHS methods. Preprint, University of Kaiserslautern, 2012. (Cited on pages 8 and 28.)

[90] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. (Cited on page 16.)

[91] G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and its Applications.*, 33:82–95, 1971. (Cited on pages 29, 32, and 119.)

[92] M. Klodt, T. Schoenemann, K. Kolev, M. Schikora, and D. Cremers. An experimental comparison of discrete and continuous shape optimization methods. In *Proceedings of the* 10^{th} *European Conference on Computer Vision (ECCV '08)*, pages 332–345, 2008. (Cited on page 26.)

[93] C. Kotropoulos and I. Pitas. Segmentation of ultrasonic images using support vector machines. *Pattern Recognition Letters*, 24(4):715–727, 2003. (Cited on page 30.)

[94] R. Kwan, A. Evans, and G. Pike. An extensible MRI simulator for post-processing evaluation. In *Visualization in Biomedical Computing*, pages 135–140. Springer, 1996. (Cited on pages 78 and 87.)

[95] R. Kwan, A. Evans, and G. Pike. MRI simulation-based evaluation of image-processing and classification methods. *IEEE Transactions on Medical Imaging*, 18(11):1085–1097, 1999. (Cited on pages 78 and 87.)

[96] Y. N. Law, H. K. Lee, M. K. Ng, and A. M. Yip. A semi-supervised segmentation model for collections of images. *IEEE Transactions on Image Processing*, 21:2955–2968, 2012. (Cited on pages 31, 106, 109, 111, 114, 126, and 129.)

[97] J. Lellmann, F. Becker, and C. Schnörr. Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In *Proceedings of the* 12^{th} *IEEE International Conference on Computer Vision (ICCV 09)*, 2009. (Cited on pages 23, 31, 64, 91, 95, 97, and 98.)

[98] J. Lellmann, J. H. Kappes, J. Yuan, F. Becker, and C. Schnörr. Convex multi-class image labeling by simplex-constrained total variation. In *Proceedings of the* 2^{nd} *International Conference on Scale Space and Variational Methods (SSVM '09)*, pages 150–162, 2009. (Cited on pages 21, 22, 30, 32, 91, and 134.)

[99] J. Lellmann, D. Breitenreicher, and C. Schnörr. Fast and exact primal-dual iterations for variational problems in computer vision. In *Proceedings of the* 11^{th} *European Conference on Computer Vision (ECCV '10)*, pages 494–505, 2010. (Cited on page 21.)

[100] J. Lellmann, F. Lenzen, and C. Schnörr. Optimality bounds for a variational relaxation of the image partitioning problem. In *Proceedings of the* 8^{th} *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR '11)*, pages 132–146, 2011. (Cited on page 134.)

[101] O. Lezoray, V.-T. Ta, and A. Elmoataz. Partial differences as tools for filtering data on graphs. *Pattern Recognition Letters*, 31(14):2201–2213, 2010. (Cited on page 108.)

[102] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, Mar. 1982. (Cited on pages 27, 30, 34, 35, and 39.)

[103] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, and Y. Wen. Multiview vector-valued manifold regularization for multilabel image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2013. to appear. (Cited on page 30.)

[104] H. Lütkepohl. *Handbook of matrices*. Wiley, 1997. (Cited on page 27.)

[105] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. (Cited on pages 24, 26, 27, and 28.)

[106] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the* 5^{th} *Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967. (Cited on pages 27, 30, 34, 35, and 39.)

[107] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the* 8^{th} *IEEE International Conference on Computer Vision (ICCV '01)*, pages 416–423, 2001. URL `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/`. (Cited on pages 52 and 84.)

[108] R. Meyer. Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *Journal of Computer System Sciences*, 12:108–121, 1976. (Cited on page 54.)

[109] C. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17 (1):177–204, 2005. (Cited on pages 28, 29, 32, and 135.)

[110] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *Journal Optimization Theory and Applications*, 50(1):195–200, 1986. (Cited on pages 70 and 112.)

[111] H. Q. Minh and V. Sindhwani. Vector-valued manifold regularization. In *Proceedings of the* 28^{th} *International Conference on Machine Learning (ICML '11)*, pages 57–64, 2011. (Cited on page 30.)

[112] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel. A proof of convergence for the alternating direction method of multipliers applied to polyhedral-constrained functions. Preprint arXiv:1112.2295, Cornell University, 2011. (Cited on page 66.)

[113] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, XLII: 577–685, 1989. (Cited on pages 6, 14, 65, 73, and 74.)

[114] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005. (Cited on pages 24 and 66.)

[115] M. Nikolova, S. Esedoglu, and T. F. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5): 1632–1648, 2006. (Cited on pages 17, 21, 30, 32, 35, 53, 75, and 133.)

[116] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1): 12–49, 1988. (Cited on page 16.)

[117] G. Pedrick. *Theory of preproducing kernels for Hilbert spaces of vector-valued functions.* University of Kansas Department Of Math., Lawrence, Kansas, 1957. (Cited on page 28.)

[118] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proceedings of the* 13^{th} *IEEE International Conference on Computer Vision (ICCV '11)*, 2011. (Cited on page 24.)

[119] T. Pock, T. Schoenmann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *Proceedings of the* 10^{th} *European Conference on Computer Vision (ECCV '08)*, pages 792–805. Springer, 2008. (Cited on page 18.)

[120] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *Proceedings of the* 12^{th} *IEEE Conference on Computer Vision and Pattern Recognition (ICCV '09)*, pages 810–817, 2009. (Cited on pages 21, 22, 23, 30, 32, 64, and 134.)

[121] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *Proceedings of the* 12^{th} *IEEE International Conference on Computer Vision (ICCV '09)*, pages 1133–1140, 2009. (Cited on page 15.)

[122] D. Potts and G. Steidl. Optimal trigonometric preconditioners for nonsymmetric Toeplitz systems. *Linear Algebra and its Applications*, 281:265–292, 1998. (Cited on page 69.)

[123] R. B. Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952. (Cited on pages 21 and 25.)

[124] A. Rack, S. Zabler, B. R. Müller, H. Riesemeier, G. Weidemann, A. Lange, J. Goebbels, M. Hentschel, and W. Görner. High resolution synchrotron-based radiography and tomography using hard X-rays at the BAMline (BESSY II). *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 586(2):327–344, 2008. (Cited on pages 35 and 92.)

[125] S. Rangapuram and M. Hein. Constrained 1-spectral clustering. In *Proceedings of the* 15^{th} *International conference on Artificial Intelligence and Statistics*, pages 1143–1151, 2012. (Cited on page 28.)

[126] K. R. Rao and P. Yip. *Discrete Cosine Transform*. Academic Press, New York, 1990. (Cited on page 69.)

[127] E. Ricci and R. Perfetti. Retinal blood vessel segmentation using line operators and support vector classification. *IEEE Transactions on Medical Imaging*, 26(10):1357–1365, 2007. doi: 10.1109/TMI.2007.898551. (Cited on page 30.)

[128] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970. (Cited on page 139.)

[129] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992. (Cited on page 64.)

[130] B. Sandberg, S. Kang, and T. Chan. Unsupervised multiphase segmentation: A phase balancing model. *IEEE Transactions on Image Processing*, 19(1):119–130, 2010. (Cited on page 18.)

[131] A. Sawatzky, D. Tenbrinck, X. Jiang, and M. Burger. A variational framework for region-based segmentation incorporating physical noise models. *Journal of Mathematical Imaging and Vision*, 2013. (Cited on page 15.)

[132] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational methods in imaging*. Springer, 2009. (Cited on pages 15 and 64.)

[133] C. Scheuerlein, M. Di Michiel, M. Scheel, J. Jiang, F. Kametani, A. Malagoli, E. E. Hellstrom, and D. C. Larbalestier. Void and phase evolution during the processing of bi-2212 superconducting wires monitored by combined fast synchrotron micro-tomography and x-ray diffraction. *Superconductor Science and Technology*, 24(11):115004, 2011. (Cited on page 101.)

[134] B. Schmitzer and C. Schnörr. Modelling convex shape priors and matching based on the gromov-wasserstein distance. *Journal of Mathematical Imaging and Vision*, pages 1–17, 2012. (Cited on page 13.)

[135] B. Schölkopf and A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2001. (Cited on page 30.)

[136] L. Schwartz. Sous-espaces Hilbertiens d'espaces vectoriels topoligiques et noyaux associés (noyaux reproduisants). *Journal Analyse Mathematique*, pages 115–256, 1964. (Cited on pages 28 and 135.)

[137] S. Selim and M. Ismail. k-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6(1):81–87, 1984. (Cited on pages 30, 34, 37, 39, and 59.)

[138] S. Setzer. Operator splittings, Bregman methods and frame shrinkage in image processing. *International Journal of Computer Vision*, 92(3):265–280, 2011. (Cited on pages 24, 66, 69, 93, and 123.)

[139] S. Setzer, G. Steidl, and T. Teuber. On vector and matrix median computation. *Journal of Computational and Applied Mathematics*, 236(8):2200–2222, 2011. (Cited on pages 39 and 50.)

[140] B. Shafei and G. Steidl. Segmentation of images with separating layers by fuzzy c-means and convex optimization. *Journal of Visual Communication and Image Representation*, 23(4):611–621, 2012. (Cited on pages 8, 32, 34, 35, 65, 70, 91, and 136.)

[141] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. (Cited on page 26.)

[142] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the* 22^{nd} *international conference on Machine learning (ICML '05)*, pages 824–831, 2005. (Cited on pages 30, 31, 107, and 121.)

[143] D. Singaraju, L. Grady, and R. Vidal. P-brush: Continuous valued mrfs with normed pairwise distributions for image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pages 1303–1310, 2009. (Cited on page 109.)

[144] A. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Proceedings of the* 11^{th} *IEEE International Conference on Computer Vision (ICCV '07)*, pages 1–8, 2007. (Cited on page 108.)

[145] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer, 1st edition, 1999. ISBN 3540656715. (Cited on page 79.)

[146] C. Sommer, C. Straehle, U. Koethe, and F. A. Hamprecht. ilastik: Interactive Learning and Segmentation Toolkit. In 8^{th} *IEEE International Symposium on Biomedical Imaging (ISBI '11)*, 2011. (Cited on page 13.)

[147] B. Song and T. F. Chan. A fast algorithm for level set based optimization. CAM Report 02-68, University of California, Los Angeles, 2002. (Cited on page 18.)

[148] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23 (4):501–509, 2004. (Cited on page 129.)

[149] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 2012. (Cited on page 13.)

[150] J. Stoer and R. Bulirsch. *Numerische Mathematik 2*. Springer, Berlin-Heidelberg-New York, 1990. (Cited on page 110.)

[151] P. Strandmark, F. Kahl, and N. Overgaard. Optimizing parametric total variation models. In *Proceedings of the* 12^{th} *IEEE International Conference on Computer Vision (ICCV '09)*, pages 2240–2247, 2009. (Cited on page 17.)

[152] E. Strekalovskiy, A. Chambolle, and D. Cremers. A convex representation for the vectorial Mumford-Shah functional. In *Proccedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pages 1712–1719, 2012. (Cited on page 15.)

[153] C. Sugar and G. James. Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 98(463):750–763, 2003. (Cited on page 36.)

[154] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002. (Cited on page 119.)

[155] A. Szlam and X. Bresson. Total variation and cheeger cuts. In *Proceedings of the* 27^{th} *International Conference on Machine Learning (ICML '10)*, pages 1039–1046, 2010. (Cited on page 28.)

[156] M. Teboulle. A unified continuous optimization framework for center-based clustering methods. *Journal of Machine Learning Research*, 8:65–102, 2007. (Cited on pages 30, 31, 34, 35, 38, 39, 41, 44, 48, 50, and 133.)

[157] A. Tsai, A. Yezzi Jr, and A. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001. (Cited on page 15.)

[158] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50:271–293, 2002. (Cited on pages 16 and 64.)

[159] Z. Volkovich, Z. Barzily, and L. Morozensky. A statistical model of cluster stability. *Pattern Recognition*, 41(7):2174–2188, 2008. (Cited on page 36.)

[160] H. Voß and U. Eckhardt. Linear convergence of generalized Weiszfeld's method. *Computing*, 25:243–251, 1980. (Cited on page 50.)

[161] D. Wagner and F. Wagner. Between min cut and graph bisection. In *Proceedings of the* 18^{th} *International Symposium on Mathematical Foundations of Computer Science (MFCS '93)*, pages 744–750, 1993. (Cited on page 27.)

[162] X.-Y. Wang, T. Wang, and J. Bu. Color image segmentation using pixel wise support vector machine classification. *Pattern Recognition*, 44(4):777 – 787, 2011. (Cited on page 30.)

[163] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008. (Cited on page 112.)

[164] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tôhoku Mathematical Journal*, 43:355–386, 1937. (Cited on page 50.)

[165] E. Weiszfeld and F. Plastria. On the point for which the sum of the distances to n given points is minimum. *Annals of Operations Research*, 167:7–41, 2009. (Cited on page 50.)

[166] M. Welk, J. Weickert, F. Becker, C. Schnörr, C. Feddern, and B. Burgeth. Median and related local filters for tensor-valued images. *Signal Processing*, 87(2):291–308, 2007. (Cited on page 61.)

[167] M. Werlberger, M. Unger, T. Pock, and H. Bischof. Efficient minimization of the non-local Potts model. In *Proceedings of the* 3^{rd} *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM '11)*, pages 314–325, 2012. (Cited on page 23.)

[168] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993. (Cited on page 27.)

[169] H.-Y. Yang, X.-Y. Wang, Q.-Y. Wang, and X.-J. Zhang. Ls-svm based image segmentation using color and texture information. *Journal of Visual Communication and Image Representation*, 23(7):1095–1112, 2012. (Cited on page 30.)

[170] J. Yuan, E. Bae, X.-C. Tai, and Y. Boykov. A continuous max-flow approach to Potts model. In *Proceedings of the* 11^{th} *European Conference on Computer Vision (ECCV '10)*, volume 6316 of *Lecture Notes in Computer Science*, pages 379–392. Springer, 2010. (Cited on page 21.)

[171] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *Proceedings of the Vision, Modeling, and Visualization Conference (VMV '08)*, pages 243–252, 2008. (Cited on pages 21, 22, 23, 30, 32, 64, and 134.)

[172] W. I. Zangwill. *Non-Linear Programming: A Unified Approach*. Prentice-Hall, Englewood Cliffs, 1969. (Cited on pages 3, 4, 30, 32, 34, 53, 54, 56, 57, 58, 71, and 133.)

[173] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179–195, 1996. (Cited on page 16.)

[174] D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Proceedings of the 27th DAGM Symposium*, 2005. (Cited on page 108.)

[175] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16:321–328, 2004. (Cited on page 113.)

[176] M. Zhu and T. F. Chan. An efficient primal-dual hybrid gradient algorithm for total vatiation image restoration. CAM Report 08-34, University of California, Los Angeles, 2008. (Cited on pages 24, 67, and 111.)

[177] W. Zhu, S. H. Kang, and G. Biros. A geodesic active contour based variational model for short axis cardiac-MR image segmentation. *International Journal of Computer Mathematics*, 90:124–139, 2013. (Cited on page 126.)

[178] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the* 20^{th} *International Conference on Machine Learning (ICML '03)*, 2003. (Cited on page 109.)

Curriculum Vitae

Scientific Career

05/2013	Submission of doctoral thesis, University of Kaiserslautern, Germany, title: *Multi-Class Image Segmentation via Convex and Biconvex Optimization*
08/2010	Diploma (German degree "Diplom" similar to MSc.) in *Mathematics with Computer Science*, Technische Universität Darmstadt, Germany, diploma thesis title: *Energy-Optimal Control of Industrial Robots*, major fields of study: optimization, robotics, analysis

Wissenschaftlicher Werdegang

05/2013	Einreichen der Doktorarbeit, Technische Universität Kaiserslautern, Titel: *Multi-Class Image Segmentation via Convex and Biconvex Optimization*
08/2010	Diplom in *Mathematics with Computer Science*, Technische Universität Darmstadt, Diplomarbeitstitel: *Energieoptimale Steuerung von Industrierobotern*, Studienschwerpunkte: Optimierung, Robotik, Analysis