Masterthesis

Image Segmentation Methods and an Application to Brain Images

Christoph Nicolai

Technische Universität Hamburg-Harburg

Betreuer	Prof. Dr. Marko Lindner
Mitgutachter	Dr. Mijail Guillemard

Universitätsklinikum Hamburg-Eppendorf

Ansprechpartner für medizinische Aspekte	Dr. Siawoosh Mohammadi

Eigenständigkeitserklärung

Die vorliegende Arbeit habe ich selbständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt. Die Arbeit habe ich vorher nicht in einem anderen Prüfungsverfahren eingereicht. Die eingereichte schriftliche Fassung entspricht genau der auf dem elektronischen Speichermedium.

Ort und Datum	Christoph Nicolai

Contents

Chapter 1.

Introduction

At the beginning of this thesis about *image segmentation*, we first have to define what an *image* is in a mathematical sense. We want to describe two-dimensional images, like photos or scans, as well as three-dimensional images, created by computer tomography or magnetic resonance imaging for example. These images may be described by one brightness value per image point (grayscale image) or several brightness values (several color channels).

1.1. Mathematical Description of Images

We will describe images as a mapping from a domain Ω to a color space C:

Definition 1.1 (Image) An *image* f is a mapping from the *image domain* Ω to the *color space* C. In this thesis and the accompanying software toolbox, the input image to an image processing method is typically denoted by f and the method's output image by u.

The domain Ω could either be a discrete set $\Omega \subset \mathbb{N}^d$, typically

$$\Omega = \prod_{i=1}^{d} \{1, \dots, n_i\},$$

or a continuous set $\Omega \subset \mathbb{R}^d$, typically

$$\Omega = \prod_{i=1}^{d} [s_i, t_i],$$

with $d = 2$ for two-dimensional images and $d = 3$ for three-dimensional images.

The color space C as well may be either a discrete or a continuous set. In the discrete setting, we will only consider k-bit images with 2^k distinct brightness values for each of the c color channels, $C = \left\{0, \dots, 2^k - 1\right\}^c$. In the continuous setting, we will consider $C = I \subset \mathbb{R}$ or $C = I^c \subset \mathbb{R}^c$ for an interval I, typically $I = [0, 1]$.

Whether we will model the image as a function on a discrete or continuous set Ω will depend on the respective image segmentation method.

Regarding the color space, we will almost always start with a k-bit image (as this is the way images are stored on a computer) and then migrate to a continuous color space since this simplifies many computations. The result will then be stored as a k-bit image again, mapping the darkest value of the image to 0, the brightest value to $2^k - 1$ and the intermediate values proportionally.

1.2. Image Segmentation

Image segmentation is the decomposition of an image (strictly speaking of the domain of an image) into coherent regions. This could be the differentiation between an object and the background, but also the separation of several objects inside an image.

For human beings, segmentation is typically easy. The human eye is well capable of telling different image regions apart, even if their borders are blurred, inarticulate or even partially obscured, see figure 1.1. The segmentation under human survey is called *supervised segmentation*, in contrast to the automatic *unsupervised segmentation* requiring no human input. Due to the respective image capturing techniques, real-world images inevitably contain noise, artifacts or other quality flaws. Therefore, the unsupervised segmentation of an image is a challenging task.

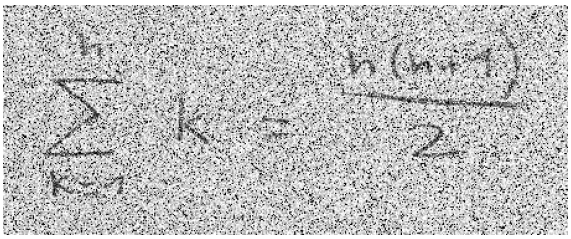


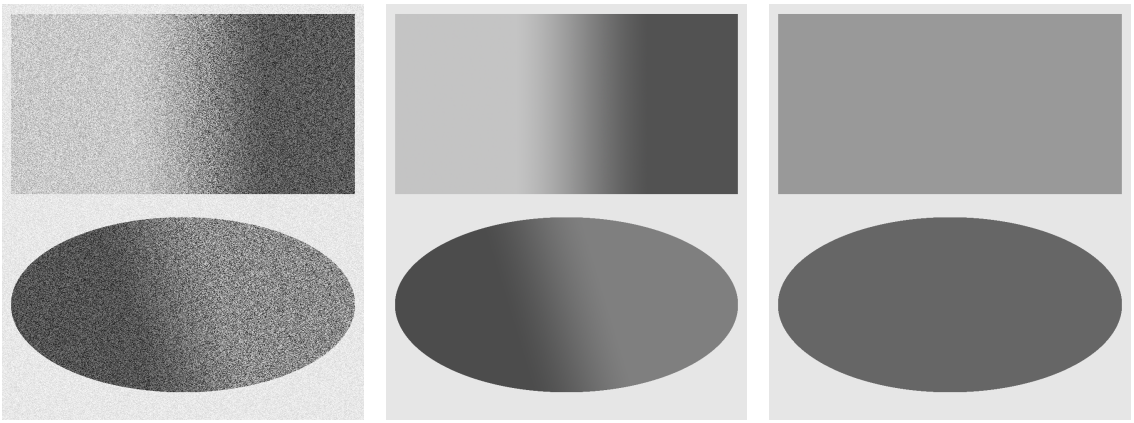
Figure 1.1.: Highly degraded image, yet still readable

In the medical context, segmentation means the differentiation between diagnostically or therapeutically relevant regions like tissue, tumors or vessels for example. For brains, this could mean the separation of white matter, gray matter, and cerebrospinal fluid.

Definition 1.2 (Segmentation) Let $f : \Omega \to C$ be an image, $u : \Omega \to C$ be an image and $S \subset \mathcal{P}(\Omega)$ be a partition of Ω, that is

$$\bigcup_{A \in S} A = \Omega \qquad \text{and} \qquad A \cap B = \emptyset \; \forall A, B \in S \text{ with } A \neq B.$$

The tuple (u, S) is called *segmentation* of f and the sets $A \in S$ are called *regions*. If there is a $k \geq 1$ such that u is k-times continuously differentiable on every $A \in S$, the segmentation is called *piecewise smooth*. If u is constant on A for every $A \in S$, the segmentation is called *piecewise constant*. See also figure 1.2.



(a) Input image (b) Piecewise smooth (c) Piecewise constant

Figure 1.2.: Image segmentation example

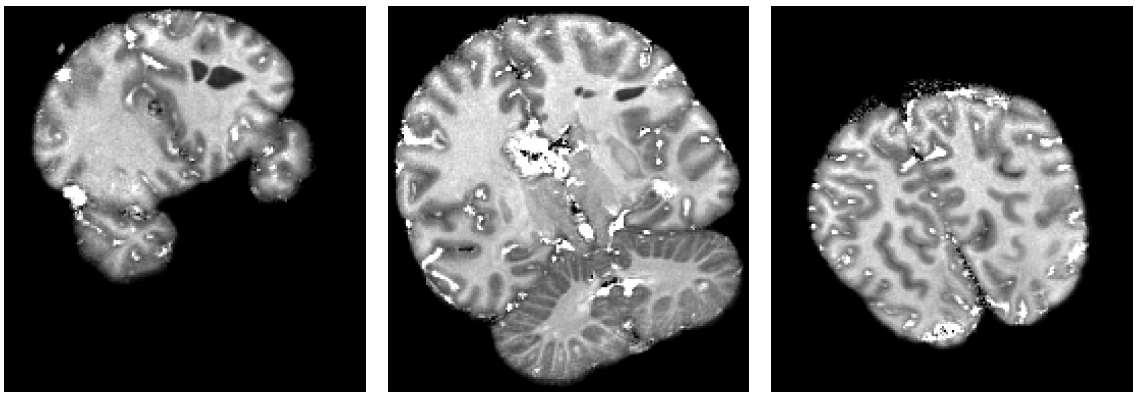
Remark 1.3 Up to this point, a segmentation in the sense of definition 1.2 is a purely formal concept with limited practical use. The following properties are desirable for a "good", practically relevant segmentation (u, S) of f:

1. u should be a segmented version of f, so in some sense f and u should be similar.
2. The system S should be a "fine enough" partition of Ω, such that there is little variation of u on $A \in S$ and there is no need to further subdivide A.
3. The system S should be a "coarse enough" partition of Ω, such that small artifacts, noise or little dirt particles are not assigned an own region.

1.3. Images of the Brain

The presented segmentation methods will be examined on two types of brain images: three-dimensional magnetic resonance images, like the image seen in figure 1.3, and two-dimensional photographs of histological sections, like the image seen in figure 1.4.

Only marginal parts of a three-dimensional image can be shown in a two-dimensional document, which makes a credible performance estimate and comparison impossible. Our focus, therefore, will be the histological image in figure 1.4, but the results apply to the three-dimensional data in a similar way.



(a) $d_3 = 63$ (b) $d_3 = 119$ (c) $d_3 = 187$

Figure 1.3.: Three slices of three-dimensional (size $d_1 \times d_2 \times d_3$) magnetic resonance image [20]

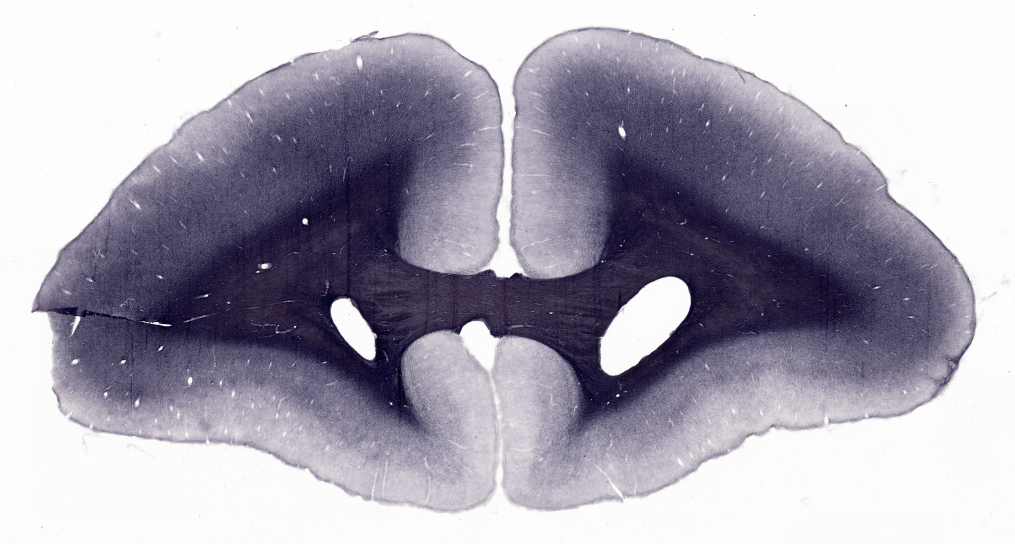


Figure 1.4.: Histological section of the brain [21]

1.4. Overview of the Upcoming Chapters

Chapters 2 and 3 describe two basic segmentation methods, *thresholding* and *graph-based segmentation*. Both will turn out to work well for special kinds of images, but to be inapplicable to general problems.

In chapter 4, a sophisticated variational approach to segmentation is developed. The core part of this approach is the *Mumford-Shah functional*, which is well suited to model a segmentation in the sense of remark 1.3. Unfortunately, the numerical implementation of this method will turn out to be very difficult and subsequent chapters will present alternative approximations to the Mumford-Shah functional.

Chapter 5 deals with the task to recognize an object in front of a background and how this special case of *Active Contours* can be tackled numerically.

Chapter 6 is dedicated to the optimization of variational problems in the context of image processing. The theory of *primal-dual optimization* and the associated numerical algorithms are described. The results of chapter chapter 6 will then be used in chapter 7, where the *Fast Mumford-Shah* approach is presented. This approach is very flexible as it allows an arbitrary number of regions and both piecewise smooth and piecewise constant segmentations. It can be practically implemented using the primal-dual optimization framework.

Chapter 8 will show how the combination of the presented methods delivers interesting results and opens up new perspectives.

An essential part of this thesis is the associated software toolbox. Appendix A contains the documentation of the user-relevant functions in this toolbox.

Chapter 2.

Thresholding

Thresholding is a histogram-based segmentation method primarily useful for grayscale[1] images. In thresholding, only the brightness values and no spatial properties of the input image are considered.

Definition 2.1 (Histogram) Let $f : \Omega \to C = \{0, \dots, 2^k - 1\}$ be a k-bit image on the discrete set Ω. The mapping

$$\begin{aligned} H_f &: C \to \mathbb{N}_0 \\ H_f(t) &\coloneqq \left|\{x \in \Omega : f(x) = t\}\right| \end{aligned}$$

is called the *histogram* of f.

2.1. Definition of Thresholding

Suppose that f is a k-bit grayscale image on Ω. Let the color space $C = \{0, \dots, 2^k - 1\}$ be split into n regions by the $(n-1)$ thresholds $t_1, \dots, t_{n-1}$:

$$\begin{aligned} L_1 &= \{k \in F : k \leq t_1\} \\ L_i &= \{k \in F : t_{i-1} < k \leq t_i\} \qquad i \in \{2, \dots, n\} \\ L_n &= \{k \in F : t_{n-1} < k\} \end{aligned}$$

See also figure 2.1. This partitioning of C may be used to obtain a segmentation u as

$$\begin{aligned} u &: \Omega \to \{1, \dots, n\} \\ u(x) &= i \quad \text{if } x \in L_i \end{aligned}$$

This order of segmentation steps, partitioning the color space first and using this partition to segment the domain of the image, demonstrates that thresholding only utilizes the brightness distribution and no spatial or geometric features, see also figure 2.2.

[1] There are extensions to color images, but these typically consider every color channel separately or transform the color space such that the brightness is described by only one channel (e.g. into HSV or LAB).

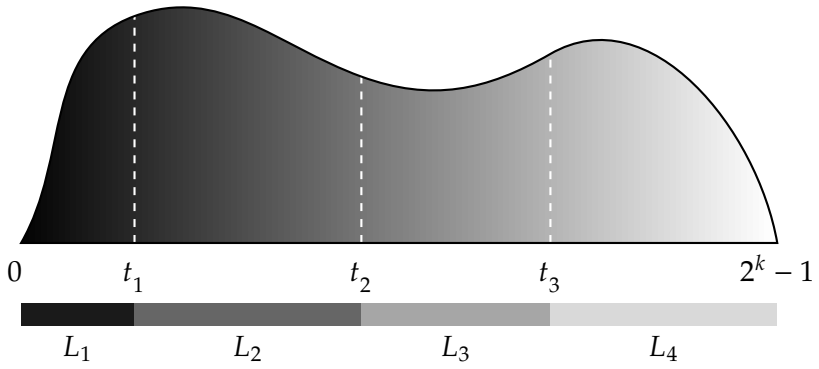
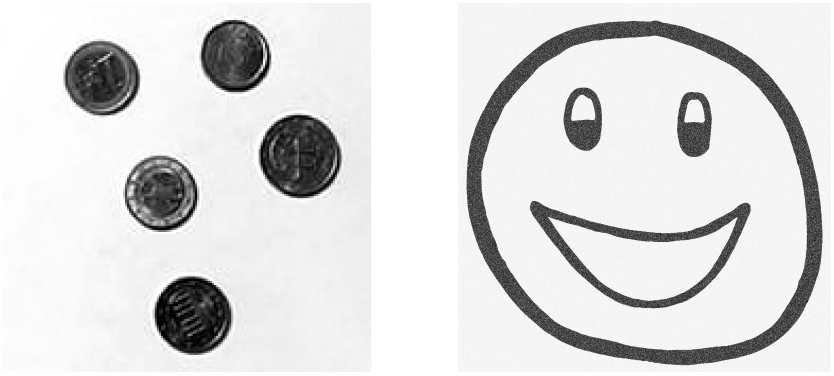


Figure 2.1.: The thresholds t_1, t_2, t_3 shown in the histogram above partition the color space in the four classes L_1, L_2, L_3, L_4



(a) (b)

Figure 2.2.: Rearranging the pixels of image (a) yields image (b), therefore these two images have the same histogram. Although not impossible, it is unlikely that there is a threshold t_1 that could be used to separate object from background in both images.

2.2. Automatic Threshold Computation with Otsu's Method

If the thresholds $t_1, \ldots, t_n$ are computed automatically, thresholding may be used as an unsupervised segmentation method. This computation can be done using *Otsu's method* ([13]):

Let $f : \Omega \to C = \{0, \ldots, 2^k - 1\}$ be a k-bit image. For every class L_i, we define the following properties:

$$
\begin{aligned}
m_i &:= \sum_{h \in L_i} H_u(h) && \text{probability of class } L_i \\
\mu_i &:= \frac{1}{m_i} \sum_{h \in L_i} h \cdot H_u(h) && \text{expexted value of class } L_i \\
\sigma_i^2 &:= \sum_{h \in L_i} \left(h - \mu_i\right)^2 \cdot H_u(h) && \text{variance of class } L_i
\end{aligned}
$$

Now choose $t_1, \ldots, t_{n-1}$ such that the sum of the brightness variances is minimized:

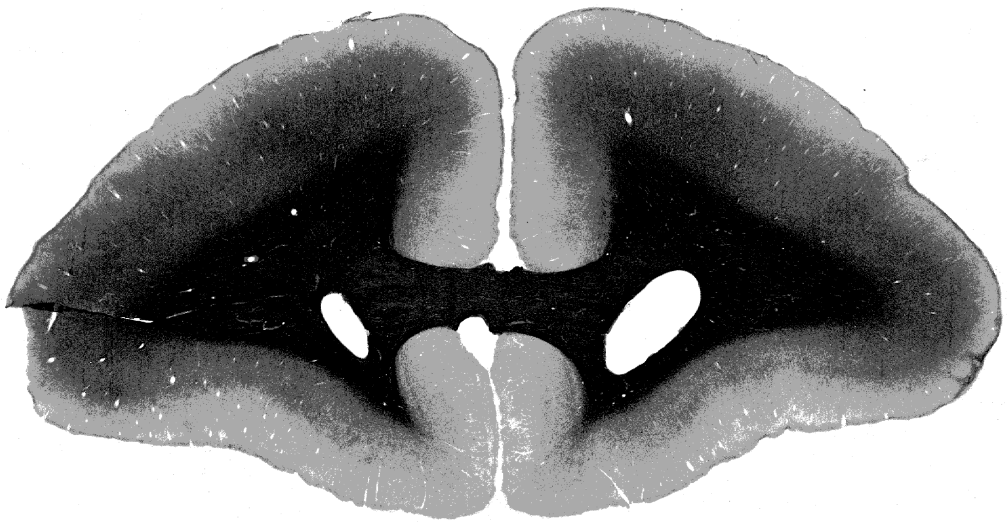
$$
\min_{(t_1, \ldots, t_{n-1})} \sum_{i=1}^{n} \sigma_i^2
$$

For a visualization, see figure 2.4.

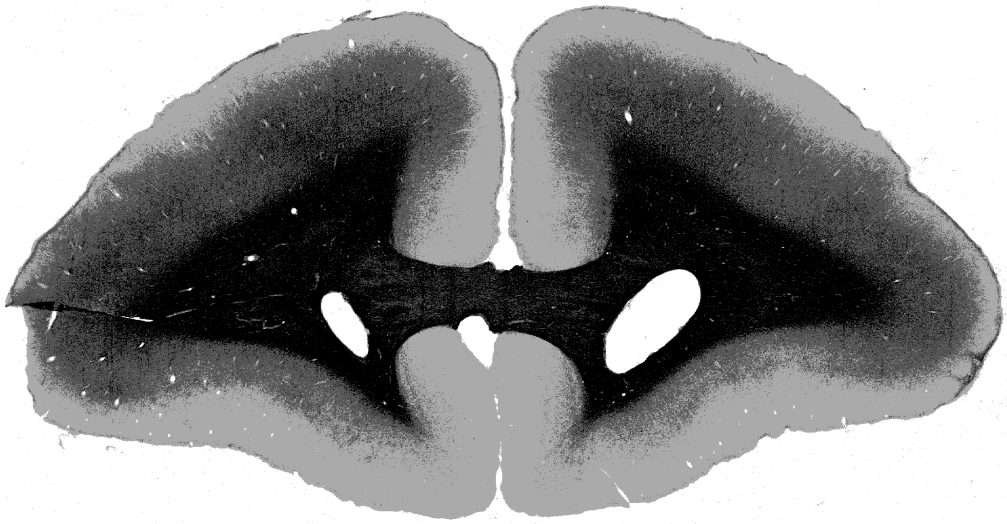
2.3. Application to Medical Data

When applying thresholding to the histology image 1.4, the results are dissatisfying, see figure 2.3. Thresholding works well if the histogram of the input image contains prominent peaks, so that the image is nearly segmented and there are only slight variations around each region's average brightness left.

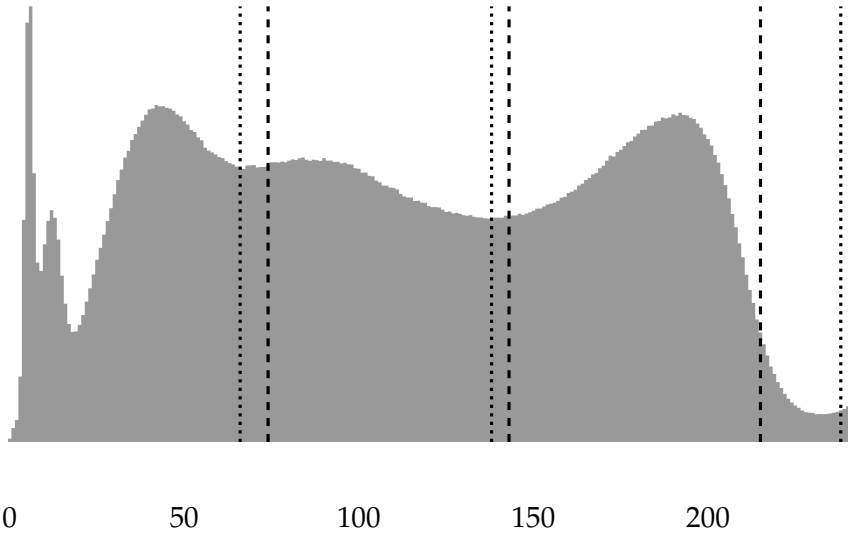
Another use case is the coarsening of a piecewise constant segmentation, that is the transition of a segmentation with N regions to a segmentation with n regions, where $n < N$. This works well if the original segmentation contains many regions with similar brightness. We will see and use this in chapter 8.



(a) Thresholding with Otsu thresholds $t = (74, 143, 215)$

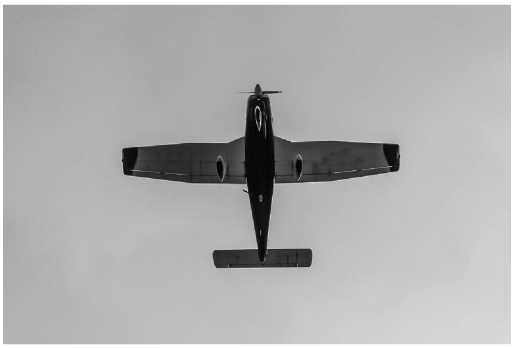


(b) Thresholding with manual thresholds $t = (66, 138, 238)$



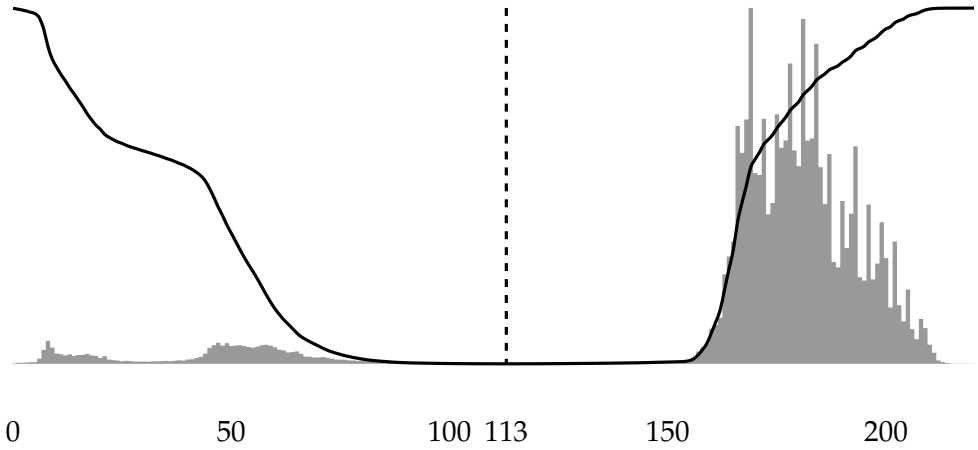
(c) Excerpt of the histogram with dashed Otsu thresholds and dotted manual thresholds

Figure 2.3.: Thresholding with $t = (t_1, t_2, t_3)$ applied to the histology image 1.4. While the manual thresholds work slightly better (see for example the light gray area at the bottom of the images), both results are dissatisfying: the regions are sprinkled and there is no sharp border between them.



(a) Input image

(b) Thresholding with $t = 113$



(c) Histogram in gray, sum of variances $\sigma_1^2 + \sigma_2^2$ as the solid line and optimal threshold $t = 113$ as dashed line

Figure 2.4.: Otsu's method: computing the threshold that minimizes the sum of the class variances

Chapter 3.

Graph-based Segmentation

Definition 3.1 A *graph* $G = (V, E)$ consists of the non-empty, finite set of *vertices* V and the set of *edges* $E \subset V \times V$.

With the *weight function* $w : E \to \mathbb{R}_+$ assigning a weight to every edge, (V, E, w) is called *weighted graph*.

An image $f : \Omega \to C = \mathbb{R}^c$ on a discrete set Ω can be modeled as a weighted graph: Every point in Ω relates to a vertex in V. All vertices $v_i, v_j \in V$ are pairwise connected ($E = V \times V$) and the weight function w quantifies the correspondence between v_i and v_j regarding yet to be discussed criteria. It is convenient to consider $w(v_i, v_j) = 1$ for maximum correspondence between v_i and v_j (especially $w(v_i, v_i) = 1$) and $w(v_i, v_j) = 0$ for no correspondence at all. In this case, the weight could also be understood as the probability that v_i and v_j belong to the same region.

A piecewise constant segmentation of f in n regions then corresponds to a partition of V in $V_1, \ldots, V_n$. To construct this partition, in some way the weighting function w can be used.

This already highlights an advantage of the graph-based segmentation methods: their flexibility. The weight function w that specifies which vertices are considered to be similar and the technique to partition the set of vertices can be chosen independently.

3.1. Partitioning by Minimal Normalized Cuts

The very popular technique of *minimal normalized cuts* is presented in [17].

Let $V = \{v_1, \ldots, v_n\}$ be the set of vertices and $E = V \times V$. The weight function $w : E \to [0, 1]$ is made up of two terms, the brightness term w^b and the spatial term w^s with

$$w(v_i, v_j) := w^b(v_i, v_j) \cdot w^s(v_i, v_j),$$

where w^b measures the correspondence with respect to the brightness or color values and w^s the spatial distance of v_i, v_j. Possible choices could be

$$w^b(v_i, v_j) := \exp(-\|f(v_i) - f(v_j)\|^2)$$

and

$$w^s(v_i, v_j) := \begin{cases} \exp(-\|v_i - v_j\|^2) & \text{if } \|v_i - v_j\| < r \\ 0 & \text{otherwise,} \end{cases}$$

with a radius $r > 0$, where $f(v_i)$ is the color value at the point in Ω that corresponds to v_i and $\|v_i - v_j\|$ is the distance of the points corresponding to v_i, v_j in Ω.

For a partition $V = A \cup B$, we define

$$\text{cut}(A, B) := \sum_{u \in A, v \in B} w(u, v),$$

the sum of the weight of all edges that are removed when the graph is split in the two parts. Naturally, we want to minimize this cut, so that we separate vertices with a low correspondence. Apart from removing edges with low weight only, minimizing the cut can be also achieved by removing as few edges as possible. This means that minimizing the cut tends to create one big subset $A \subset V$ and one small subset $B \subset V$, where the cut is minimal because of the small number of edges between A and B.

To prevent this from happening, the authors in [17] suggest normalizing the cut in the following way: They define the *association* of a subset $S \subset V$ and V as the sum of the weights of all edges from S to V as

$$\text{assoc}(S, V) := \sum_{s \in S, v \in V} w(s, v),$$

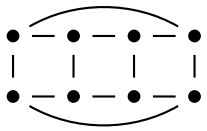
and then define the *normalized cut* of A, B as

$$\text{ncut}(A, B) := \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}. \tag{3.1}$$

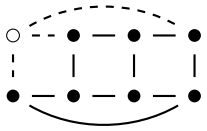
Minimizing this normalized cut instead of the original cut reduces the problem of small sets with isolated vertices, see figure 3.1 for an example.

Minimizing the normalized cut exactly turns out to be NP-complete and the authors develop an approximation as a computable alternative. They make the following definitions:

- The weight matrix W with
$$W_{ij} = w(v_i, v_j), \tag{3.2}$$
- the diagonal matrix D, where D_{ii} is the sum of the weights of all edges leaving v_i:
$$D_{ii} = \sum_{j=1}^{n} W_{ij}, \tag{3.3}$$

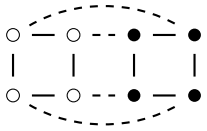


For simplicity, we assume every edge in this graph to have weight one.



Cut and normalized cut for the partition displayed left:

$$\operatorname{cut}(A,B) = 3 \qquad \operatorname{ncut}(A,B) = \frac{3}{3} + \frac{3}{12} = \frac{5}{4}$$



Equally sized A, B result in a higher cut, but a lower ncut:

$$\operatorname{cut}(A,B) = 4 \qquad \operatorname{ncut}(A,B) = \frac{4}{8} + \frac{4}{8} = 1$$

Figure 3.1.: Comparison of cut and normalized cut when partitioning $V = A \cup B$. The rings are the vertices in A and the filled circles are the vertices in B.

- the n-dimensional index vector x with $x_i = 1$ if $v_i \in A$ and $x_i = -1$ otherwise,
- the n-dimensional vector e with $e_i = 1$ for every $i \in \{1, \dots, n\}$,
- the ratio of the association of A, V to the association of V with itself,

$$k = \frac{\sum\limits_{\{1 \le i \le n : x_i > 0\}} D_{ii}}{\sum\limits_{i=1}^{n} D_{ii}} = \frac{\operatorname{assoc}(A,V)}{\operatorname{assoc}(V,V)} \qquad \text{and} \qquad b = \frac{k}{1-k}.$$

With these definitions, they show the following equivalence to minimizing the normalized cut:

$$\min_x \operatorname{ncut}(x) = \min_y \frac{y^T(D-W)y}{y^T D y}, \tag{3.4}$$

with the constraints on y that $y_i \in \{1, -b\}$ and $y^T De = 1$ and where $\operatorname{ncut}(x) \coloneqq \operatorname{ncut}(A,B)$ with A, B defined by x. Up to here, equation (3.4) is only a reformulation of the original problem and not easier to solve. But rejecting the constraint $y_i \in \{1, -b\}$ and instead considering $y \in \mathbb{R}^n$ and equation (3.4) as a generalized Rayleigh quotient, y may be computed as the eigenvector to the second smallest eigenvalue in the eigenvector problem

$$(D - W)\, y = \lambda D y. \tag{3.5}$$

The authors show that such an eigenvector y always fulfills the second constraint $y^T De = 1$.

The remaining question is how to partition $V = A \cup B$ given y. One could choose to

just consider the sign of y, that is

$$v_i \in \begin{cases} A & \text{if } y_i > 0, \\ B & \text{otherwise.} \end{cases}$$

The suggestion in [17] is to not only consider the threshold 0, but to take several thresholds distributed between the minimal and maximal entry of y and of these take the one that gives the smallest normalized cut.

This procedure is summarized in algorithm 1. If the goal is a segmentation in more than two regions, the algorithm may be used repeatedly. To give an example, to partition $G = (V, E)$ in four regions one would apply the algorithm, obtain $V = A \cup B$ and could then apply the algorithm to $G_A = (A, E_A)$ and $G_B = (B, E_B)$. If three regions are desired, the algorithm would be only applied to either G_A or G_B.

If the decision which regions should be further subdivided could be made automatically or is known from the beginning (or especially if only two regions are desired), this graph-based approach is an unsupervised segmentation method. Otherwise, it is a supervised method.

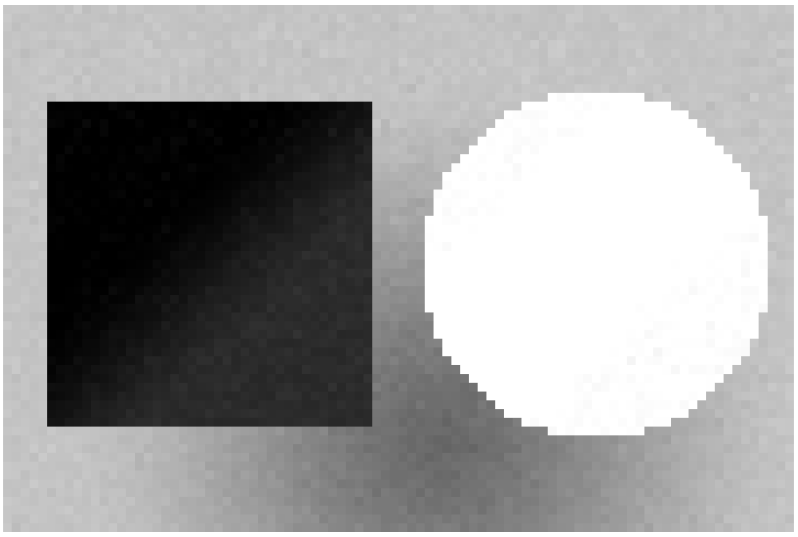
An example can be seen in figure 3.2.

Algorithm 1

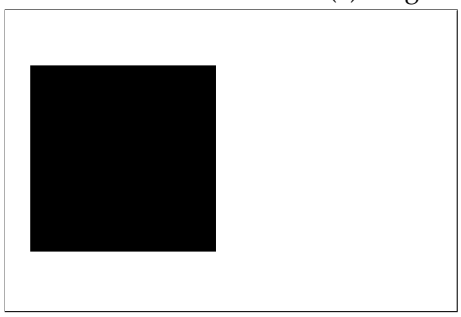
1: **Initialize:** $G = (V, E)$, weight function w that measures similarity between vertices, weight matrix W as in equation (3.2), diagonal matrix D as in equation (3.3), $k \geq 1$ as number of thresholds plus one .
2: Compute the eigenvector corresponding to the second smallest eigenvalue in equation (3.5).
3: $m \leftarrow \min_i y_i$
4: $M \leftarrow \max_i y_i$
5: **for** i=1,. . . ,k-1 **do**
6: $\quad t_i \leftarrow m + i \cdot \frac{M-m}{k}$
7: $\quad A_i \leftarrow \{v_j \in V : y_j > t\}$
8: $\quad B_i \leftarrow \{v_j \in V : y_j \leq t\}$
9: $\quad c_i \leftarrow \text{ncut}(A_i, B_i)$
10: **end for**
11: Select $j \in \{1, \ldots, k-1\}$ such that $c_j = \underset{i \in \{1,\ldots,k-1\}}{\arg\min}\ c_i$

A big disadvantage of algorithm 1 is its computational cost. Even for a small image with a size of 100×100 pixels, an eigenvector with a length of $100 \cdot 100 = 10000$ has to be computed. For slightly larger (but still relatively small) images, the length of the

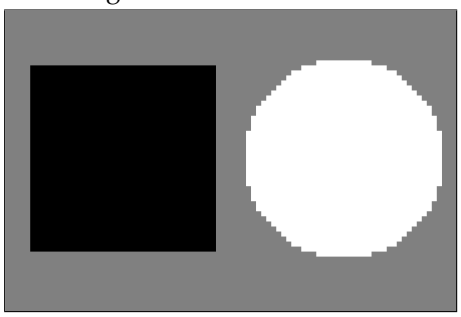
eigenvector quickly goes beyond the scope of computational possibility. This algorithm could still turn out to be interesting in combination with another method, as we will see in section 8.2.



(a) Original, artificial image



(b) First segmentation step yields A (black) and B (white)



(c) Second segmentation step, algorithm 1 applied to $G_B = (V_B, E_B)$, the white area in (b)

Figure 3.2.: Segmentation using algorithm 1

Chapter 4.

Variational Segmentation and Definition of the Mumford-Shah Functional

Variational methods are a popular technique in mathematical image processing. For a given image processing task, each desired property of the processed image $\bar{u}$ is formulated as a functional that quantifies how well $\bar{u}$ fulfills that respective property. Typically, the lower the value of the functional at $\bar{u}$, the better does $\bar{u}$ fulfill the property. These properties could for example be similarity to the input image f, exact accordance to the input on a subset of the image domain or smoothness. Then one tries to find the "optimal" image $\bar{u}$ that minimizes the sum of these functionals.

The desired properties of a segmentation in remark 1.3 can be modeled well by a variational approach. If $f : \Omega \to C$ is the input image, then we are looking for an image u (in a yet to be defined function space X) and an *edge set* $K \subset \Omega$ dividing Ω in distinct regions such that

1. u and f are similar, which could be achieved by demanding that $\mathcal{D}_f(u) = \|u - f\|_X^2$ is small,
2. there is no intense variation in brightness and color of u on every region, so in some sense the gradient of u is small on $\Omega \setminus K$,
3. the set K is small (in a sense that is yet to be defined), such that small artefacts or dirt particles are not dedicated an own region.

The sum of these three requirements will lead to the *Mumford-Shah functional*, first described in [9], *the* classical variational approach to image segmentation. Before we can formulate this functional, we have to address the two remaining questions above: in what sense is the set K small and what is a suitable function space X for segmented images?

The set $K \subset \Omega \subset \mathbb{R}^d$ represents the border between different regions, therefore has a dimension that is smaller than d and in consequence is $\mathcal{L}^d$ negligible.

On the one hand, the space X should contain functions with discontinuities, since those represent edges in an image. On the other hand, the functions in X should be differentiable, such that in some sense a gradient exists. These seemingly inconsistent requirements are neither fulfilled by the classical function spaces C^k nor by the Sobolev spaces.

In the following section 4.1, we will define the *Hausdorff measure* as an answer to the first question and lay the groundwork for the definition of the space of *Special Functions of Bounded Variation*, which turns out to be a function space well suited for our needs.

The next sections follow the demonstrations in [19, 8, 16].

In the following let $\Omega \subset \mathbb{R}^d$ be a non-empty and open subset, $\mathcal{P}(\Omega)$ be the power set of Ω and let

- $W = [0, \infty]$,
- $W = [-\infty, \infty)$ or $W = (-\infty, \infty]$,
- or $W = \mathbb{R}^d$.

4.1. Preliminaries from Measure Theory

Definition 4.1 The family of subsets $\mathcal{A} \subset \mathcal{P}(\Omega)$ is called *Sigma-algebra*, short σ-algebra, if

- $\emptyset \in \mathcal{A}$
- for every $A \in \mathcal{A}$ it holds $(\Omega \setminus A) \in \mathcal{A}$
- if $A_i \in \mathcal{A}$, $i \in \mathbb{N}$, then also $\bigcup_{i \in \mathbb{N}} A_i \in \mathcal{A}$

The pair $(\Omega, \mathcal{A})$ is called *measurable space* and the sets $A \in \mathcal{A}$ are called *measurable*.

Definition 4.2 Let $(\Omega, \mathcal{A})$ be a measurable space. A function $\nu : \mathcal{A} \to W$ fulfilling $\nu(\emptyset) = 0$ is called σ-additive, if for all pairwise disjoint $(A_i)_{i=1}^{\infty} \in \mathcal{A}$

$$\nu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \nu\left(A_i\right).$$

Definition 4.3 (Measure) Let $(\Omega, \mathcal{A})$ be a measurable space. Let $\nu : \mathcal{A} \to W$ be σ-additive. For

$W = [0, \infty]$	ν is called	*non-negative measure*
$W = [-\infty, \infty)$ or $W = (-\infty, \infty]$	ν is called	*signed measure*
$W = \mathbb{R}^d$	ν is called	*vector measure*

on $(\Omega, \mathcal{A})$. In all three cases we call ν a *measure* on $(\Omega, \mathcal{A})$ and the tuple $(\Omega, \mathcal{A}, \nu)$ is called *measure space.*

Definition 4.4 Let $(\Omega, \mathcal{A}, \nu)$ be a measure space. A set $A \in \mathcal{A}$ is called ν-negligible, if $\nu(A) = 0$. A proposition is said to hold *ν-almost everywhere* on Ω or for *ν-almost every* $x \in \Omega$, short ν-a.e., if the set $N \subset \Omega$ where it does not hold is ν-negligible.

Definition 4.5 Let $(\Omega, \mathcal{A})$ be a measurable space and $\nu, \mu : \mathcal{A} \to W$ be two measures.
ν is called *absolutely continuous* with respect two μ, denoted

$$\nu \ll \mu,$$

if every μ-negligible set is also ν-negligible.
ν, μ are called *singular* on $(\Omega, \mathcal{A})$, denoted

$$\nu \perp \mu,$$

if there are $A_1, A_2 \in \mathcal{A}$ such that $\Omega = A_1 \cup A_2$, A_1 is μ-negligible and A_2 is ν-negligible.

Definition 4.6 (Restriction of a measure) Let $(\Omega, \mathcal{A}, \nu)$ be a measure space and $A \in \mathcal{A}$. The restriction of ν to A is a measure defined as

$$(\nu \llcorner A)(B) = \nu(A \cap B).$$

Theorem 4.7 (Jordan decomposition) Let ν be a signed measure on the measurable space $(\Omega, \mathcal{A})$. There is a unique decomposition

$$\nu = \nu_+ - \nu_-$$

such that ν_+ and ν_- are non-negative measures on $(\Omega, \mathcal{A})$, at least one of them is finite ($\nu_+(\Omega) < \infty$ or $\nu_-(\Omega) < \infty$) and ν_+, ν_- are singular.

Proof:
For a proof see [7, chapter VII.1]. □

For the definition of measurable functions, the integration of real or vector valued functions with respect to a non-negative measure, the associated properties and function spaces see for example [7] and [4, chapter 2.2].

The concept of integration may be naturally extended to signed and vector measures:

Definition 4.8 Let $(\Omega, \mathcal{A})$ be a measurable space and $f : \Omega \to W$. The integration of f over $A \in \mathcal{A}$ with respect to a signed or vector measure ν is defined in terms of the integration with respect to non-negative measures.

- Let ν be a signed measure on $(\Omega, \mathcal{A})$ and $W = [-\infty, \infty]$. Then

$$\int_A f \, d\nu := \int_A f \, d\nu_+ - \int_A f \, d\nu_-,$$

defined if the case "$\infty - \infty$" does not occur and with $\nu = \nu_+ - \nu_-$ as in theorem 4.7.

- Let $\nu = (\nu_1, \dots, \nu_d)$ be a vector measure and $W = [-\infty, \infty]$. Then

$$\int_A f \, d\nu := \left(\int_A f \, d\nu_1, \int_A f \, d\nu_2, \dots, \int_A f \, d\nu_d \right) \in \mathbb{R}^d,$$

  defined if all integrals on the right-hand side are finite.
- Let $\nu = (\nu_1, \dots, \nu_d)$ be a vector measure and $W = \mathbb{R}^d$. Then

$$\int_A f \bullet d\nu := \sum_{i=1}^{d} \int_A f_i \, d\nu_i \in \mathbb{R},$$

  defined if all integrals on the right-hand side are finite.

Since the integrals on the right-hand sides in the previous definition are σ-additive with respect to $A \in \mathcal{A}$, the expressions on the left-hand are measures, and we further define:

Definition 4.9 Let $(\Omega, \mathcal{A})$ be a measurable space, $f : \Omega \to W$ and ν be a measure. We define the following measure for $A \in \mathcal{A}$ if the integrals on the right-hand side exist:

- For $W = \mathbb{R}_+$ and a non-negative measure ν, the non-negative measure $f\nu$ is defined as

$$f\nu(A) := \int_A f \, d\nu.$$

- For $W = [-\infty, \infty]$ and a vector measure ν or $W = \mathbb{R}^d$ and a non-negative measure ν, the vector measure $f\nu$ is defined as

$$f\nu(A) := \int_A f \, d\nu.$$

- For $W = \mathbb{R}^d$, the signed measure $f \bullet \nu$ is defined as

$$f \bullet \nu(A) := \int_A f \bullet d\nu.$$

In the respective case, $f\nu$ or $f \bullet \nu$ is called *weighting* of ν with *density* f.

4.1.1. Vector Radon Measures and their Variation

Definition 4.10 (Radon measure) Let $\mathfrak{B}(\Omega)$ be the Borel σ-algebra over Ω.

1. A non-negative measure $\mu : \mathfrak{B}(\Omega) \to [0, \infty]$ with $\mu(K) < \infty$ for every compact $K \in \mathfrak{B}(\Omega)$ is called *(non-negative) Radon measure*.
2. $\nu : \mathfrak{B}(\Omega) \to \mathbb{R}^d$ is called *(vector) Radon measure*, if there exists a non-negative Radon measure μ on $(\Omega, \mathfrak{B}(\Omega))$ and a density $f \in \mathrm{L}^1_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d; \mu\right)$ such that

$$\nu = f\mu.$$

The linear space of Radon measures on Ω is denoted as $\mathrm{RM}_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d\right)$.

Remark 4.11 Strictly speaking, the definition of vector Radon measures requires an intermediate step. In the definition of a non-negative Radon measure μ, the case of an Ω with $\mu(\Omega) = \infty$ is covered. For vector Radon measures, this case is more difficult, since there is no clear understanding of "∞" in $\mathbb{R}^d$.

A possible solution is to define vector Radon measures initially on the relatively compact Borel sets

$$\mathfrak{K}(\Omega) := \left\{A \in \mathfrak{B}(\Omega) : \overline{A} \text{ compact in } \Omega\right\}.$$

Because the vector Radon measure $\mu : \mathfrak{K}(\Omega) \to \mathbb{R}^d$ may be uniquely extended to $\mathfrak{B}(\Omega)$ (see [16]) and because the situation of a domain Ω with an infinite mass might be inappropriate in the context of image processing from the beginning, we omitted this intermediate step in definition 4.10.

Definition 4.12 (Variation of a measure) Let $\nu = f\mu \in \mathrm{RM}_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d\right)$ as in definition 4.10. The non-negative Radon measure

$$|\nu| := |f|\mu$$

on Ω is called the *variation* of ν.

The variation of a vector measure is a non-negative measure and can be interpreted as an absolute value for measures, see the following lemma:

Lemma 4.13 (Polar decomposition) Let $\nu \in \mathrm{RM}_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d\right)$. There is a $g \in \mathrm{L}^1_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d; |\nu|\right)$ such that

$$\nu = g|\nu|$$

and

$$\|g(x)\| = 1 \quad \text{for } |\nu|\text{-almost every } x \in \Omega.$$

g is $|\nu|$-almost everywhere uniquely determined by ν.

Proof:
Let $\nu = f\mu$ as in definition 4.10 and let $\mathcal{N} := \left\{x \in \Omega \mid f(x) = 0\right\}$, then

$$|\nu|(\mathcal{N}) = \left(|f|\mu\right)(\mathcal{N}) = \int_{\mathcal{N}} |f|\, d\mu = 0$$

and we may for $|\nu|$-almost every $x \in \Omega$ define

$$g := \frac{f}{|f|} \qquad \text{with} \qquad \|g(x)\| = 1.$$

It further holds

$$\nu = f\mu = g|f|\mu = g|\nu|.$$

For the $|\nu|$-almost everywhere uniqueness, let

$$\nu = g_1|\nu| = g_2|\nu|.$$

Then

$$\int_\Omega g_1 \, d|v| = \int_\Omega g_2 \, d|v|$$

and $g_1 = g_2$ $|\nu|$-almost everywhere. See [16] for further properties and details. □

Definition 4.14 The linear space of finite Radon measures on Ω is defined as

$$\mathrm{RM}\left(\Omega, \mathbb{R}^d\right) := \left\{\nu \in \mathrm{RM}_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d\right) : |\nu|(\Omega) < \infty\right\}.$$

Lemma 4.15 $\mathrm{RM}\left(\Omega, \mathbb{R}^d\right)$ equipped with the *total variation*

$$\|\cdot\|_{\mathrm{RM}\left(\Omega, \mathbb{R}^d\right)} : \nu \mapsto |\nu|(\Omega)$$

as a norm is a Banach space.

Proof:
For a proof see [16, theorem 0.22]. □

4.1.2. Hausdorff Measure

Suppose that $A \in \mathfrak{B}\left(\mathbb{R}^d\right)$ is k-dimensional with $k < d$. We want to develop a method to quantify the volume of A. The Lebesgue measure $\mathcal{L}^d$ is not suitable, since $\mathcal{L}^d(A) = 0$. Instead, we will use the following idea: We cover A with countably many balls $B^d_{r_i}(x_i) \subset \mathbb{R}^d$ with finite radius $0 \le r_i < \delta$ around certain points $x_i \in \mathbb{R}^d$,

$$A \subset \bigcup_{i=1}^{\infty} B^d_{r_i}(x_i),$$

and then sum up the volume of the k-dimensional equatorial planes inside $B^d_{r_i}(x_i)$:

$$\sum_{i=1}^{\infty} \alpha(k) r_i^k \;:\; A \subset \bigcup_{i=1}^{\infty} B^d_{r_i}(x_i).$$

$\alpha(k)$ is the volume of the unit sphere in $\mathbb{R}^k$, $\alpha(1) = 2, \alpha(2) = \pi, \alpha(3) = 4\pi/3, \ldots$.
We then take the infimum of all covers of A and finally consider the limit $\delta \to 0$. This construction yields the Hausdorff measure, see also definition 4.16 and figure 4.1.

Definition 4.16 (Hausdorff measure) The k-dimensional *Hausdorff measure* on $\mathfrak{B}\left(\mathbb{R}^d\right)$ is defined as

$$\mathcal{H}^k(A) \coloneqq \lim_{\delta \to 0} \inf_{(B_i)} \left\{ \sum_{i=1}^{\infty} \alpha(k) r_i^k \ : \ A \subset \bigcup_{i=1}^{\infty} B_{r_i}^d(x_i), 0 \leq r_i < \delta \right\}$$

where $B_0^n(x_i) = \emptyset$ and $\alpha(k)$ is the volume of the unit sphere in $\mathbb{R}^k$.

Theorem 4.17 The function $\mathcal{H}^k$ in definition 4.16 is a measure on $\mathfrak{B}\left(\mathbb{R}^d\right)$.

Proof:
For a proof see [8], theorem 1 in section 2.1. □

Theorem 4.18 (Properties of the Hausdorff measure) Using the Hausdorff measure on lower-dimensional subsets is reasonable in the following sense:

1. $\mathcal{H}^0$ is the counting measure.
2. $\mathcal{H}^d = c\mathcal{L}^d$ with a constant factor $c \in \mathbb{R}$. This factor may be applied in the definition of the Hausdorff measure, such that $\mathcal{H}^d = \mathcal{L}^d$, but this makes the definition less intuitive.
3. $\mathcal{H}^k \equiv 0$ if $k > d$.
4. Scaling a subset with a factor respects the dimension: $\mathcal{H}^k(\lambda A) = \lambda^k \mathcal{H}^k(A)$ for all $\lambda > 0, A \subset \mathbb{R}^d$.
5. The position of a subset is irrelevant for its size: $\mathcal{H}^k(L(A)) = \mathcal{H}^k(A)$ for each affine isometry $L : \mathbb{R}^d \to \mathbb{R}^d, A \subset \mathbb{R}^d$.

Proof:
For a proof see [8], theorem 2 in section 2.1 and the section 2.2. □

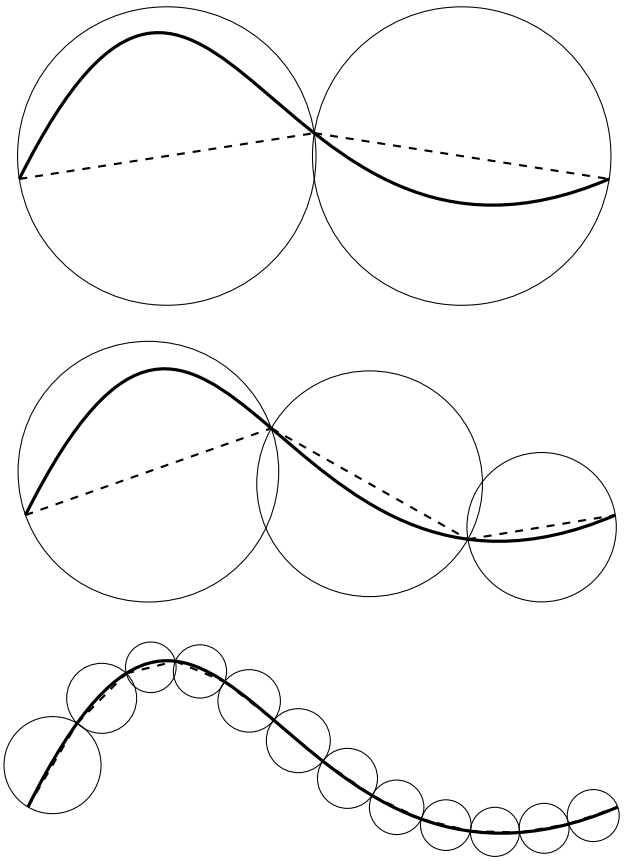


Figure 4.1.: Construction of the Hausdorff measure for a $k = 1$ dimensional curve in $n = 2$ dimensions. From top to bottom, the allowed maximal radius δ is decreased. The smaller the maximal radius, the better the approximation of the curve length.

4.2. The Spaces BV and SBV

For the rest of this chapter, we will consider the color space $C = \mathbb{R}$. The treatment of the general case $C = \mathbb{R}^n$ can be done analogously, but requires more technical work.

Let $u \in C^1(\Omega)$ and let $\varphi \in C^1\left(\Omega, \mathbb{R}^d\right)$ be compactly supported in Ω. The formula for integration by parts (Gauß) states

$$\int_\Omega u \operatorname{div} \varphi \, d\mathcal{L}^d = -\int_\Omega \varphi \bullet \nabla u \, d\mathcal{L}^d, \tag{4.1}$$

The idea for functions that are not differentiable in a classical sense is to replace $\nabla u \mathcal{L}^d$ on the right-hand side of equation (4.1) by a general Radon measure ν.

Definition 4.19 (Weak derivative) Let $u \in \mathrm{L}^1_{\mathrm{loc}}(\Omega)$. ν in $\mathrm{RM}_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d\right)$ is called *weak (total) derivative* of u and u is called *weakly differentiable* on Ω, if

$$\int_\Omega u \operatorname{div} \varphi \, d\mathcal{L}^d = -\int_\Omega \varphi \bullet d\nu \qquad \forall \varphi \in \mathcal{D}\left(\Omega, \mathbb{R}^d\right). \tag{4.2}$$

In this case, we will write $Du \coloneqq \nu$.

Definition 4.20 The space of weakly differentiable functions is denoted by

$$\mathrm{BV}_{\mathrm{loc}}(\Omega) \coloneqq \left\{u \in \mathrm{L}^1_{\mathrm{loc}}(\Omega) : Du \in \mathrm{RM}_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d\right)\right\}.$$

Definition 4.21 (Space of functions of bounded variation) The linear space of functions of bounded variation is

$$\begin{aligned}\mathrm{BV}(\Omega) &= \left\{u \in \mathrm{BV}_{\mathrm{loc}}(\Omega) \cap \mathrm{L}^1(\Omega) : Du \in \mathrm{RM}\left(\Omega, \mathbb{R}^{n\times d}\right)\right\} \\ &= \left\{u \in \mathrm{BV}_{\mathrm{loc}}(\Omega) : \int_\Omega |u| \, d\mathcal{L}^d + |Du|(\Omega) < \infty\right\}.\end{aligned}$$

Lemma 4.22 With the norm

$$\|\cdot\| : u \mapsto \int_\Omega |u| \, d\mathcal{L}^d + |Du|(\Omega) \tag{4.3}$$

$\mathrm{BV}(\Omega)$ is a Banach space.

Proof:
See for example [4, Lemma 6.105]. In [16, Theorem 1.9] it is also proved that $\mathrm{BV}(\Omega)$ is not separable for $\Omega \neq \emptyset$. □

Remark 4.23 Equation (4.1) is often used to define the *Sobolev spaces* by calling a function that behaves like ∇u for all test functions the weak derivative of u and then define the Sobolev space as the functions with these weak derivatives.

In our situation, we could define Sobolev spaces from the other direction as those BV functions whose weak derivatives in the sense of definition 4.19 are a weighting of the Lebesgue measure:

$$W^{1,1}_{\mathrm{loc}}(\Omega) \coloneqq \left\{ u \in \mathrm{BV}_{\mathrm{loc}}(\Omega) : \exists f \in \mathrm{L}^1_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d\right) : Du = f\mathcal{L}^d \right\}.$$

Remark 4.24 The following function is a canonical example for a weakly differentiable function: Let $u : (a,b) \to \mathbb{R}$, $u \in C^1\big((a,x_0) \cup (x_0,b)\big)$ with $\mathcal{L}^1$-almost everywhere defined classical derivative $u' \in \mathrm{L}^1_{\mathrm{loc}}((a,b))$ and jump in $x_0 \in (a,b)$ (figure 4.2). Assume that the one-sided limits $u(x_{0+})$ and $u(x_{0-})$ exist. For $\varphi \in \mathcal{D}((a,b),\mathbb{R})$ it holds

$$\begin{aligned}
\int_a^b u\varphi'\, d\mathcal{L}^1 &= \int_a^{x_0} u\varphi'\, d\mathcal{L}^1 + \int_{x_0}^b u\varphi'\, d\mathcal{L}^1 \\
&= \left[u\varphi\right]_a^{x_{0-}} - \int_a^{x_0} \varphi u'\, d\mathcal{L}^1 + \left[u\varphi\right]_{x_{0+}}^b - \int_{x_0}^b \varphi u'\, d\mathcal{L}^1 \\
&= -\left(\int_a^b \varphi u'\, d\mathcal{L}^1 + u(x_{0+})\varphi(x_0) - u(x_{0-})\varphi(x_0) \right) \\
&= -\int_a^b \varphi\, d\left(u'\mathcal{L}^1 + \left(u(x_{0+}) - u(x_{0-})\right)\delta_{x_0}\right)
\end{aligned}$$

and therefore $u \in \mathrm{BV}_{\mathrm{loc}}((a,b),\mathbb{R})$ with weak derivative

$$Du = u'\mathcal{L}^1 + \big(u(x_{0+}) - u(x_{0-})\big)\delta_{x_0}.$$

Remark 4.25 In many applications, the regularizer in variational methods demands that solutions don't "vary too much" on Ω or on certain subsets of Ω, that means the brightness is more or less homogeneous on the respective set.

For a continuously differentiable $u : \Omega \to \mathbb{R}$, this "variation" may be quantified using the derivative u':

$$\mathrm{Var}_\Omega(u) = \int_\Omega |u'|\, d\mathcal{L}^1.$$

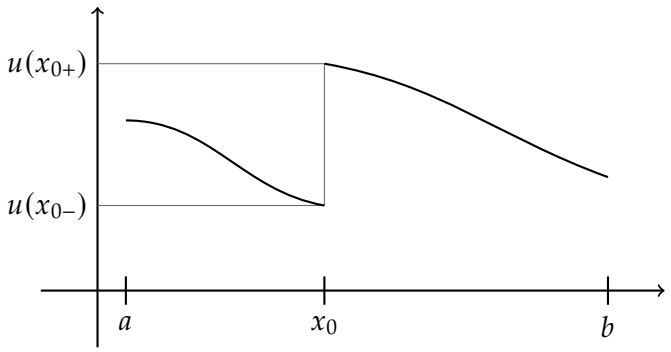


Figure 4.2.: Weakly differentiable function with discontinuity

To give an example, if we consider

$$u(x) = -x^3 + \frac{1}{2}x^2 + x$$

on $\Omega = (-1, 1)$, we can calculate the variation as

$$\begin{aligned}\mathrm{Var}_{(-1,1)}(u) &= \int_{-1}^{1} |-3x^2 + x|\, dx \\ &= \int_{-1}^{0} 3x^2 - x\, dx + \int_{0}^{1/3} -3x^2 + x\, dx + \int_{1/3}^{1} 3x^2 - x\, dx = \frac{55}{27}.\end{aligned}$$

If we define the not continuous $\hat{u}$ (see figure 4.3) with two "jumps" of length $1/2$ as

$$\hat{u}(x) = \begin{cases} u(x) + \frac{1}{2} & \text{if } -\frac{1}{2} \le x \le \frac{1}{2} \\ u(x) & \text{otherwise,} \end{cases}$$

once again thinking of discontinuities as edges in an image, we would now expect the variation of $\hat{u}$ to be

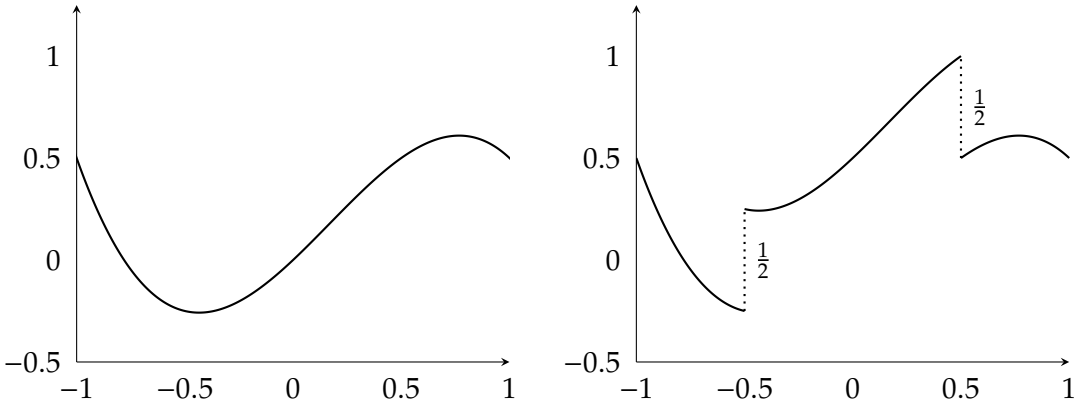
$$\mathrm{Var}_{(-1,1)}(\hat{u}) = \mathrm{Var}_{(-1,1)}(u) + 1,$$

and indeed the total variation of the weak derivative of $\hat{u}$ gives this result: As we have seen in remark 4.24, we can compute the weak derivative of $\hat{u}$ as

$$\begin{aligned}D\hat{u} &= u'\mathcal{L}^1 + \frac{1}{2}\delta_{-\frac{1}{2}} - \frac{1}{2}\delta_{\frac{1}{2}} \\ &= f\left(\mathcal{L}^1 + \delta_{-\frac{1}{2}} + \delta_{\frac{1}{2}}\right) \quad \text{with } f(x) = \begin{cases} u'(x) & \text{if } x \in (-1,1) \setminus \left\{-\frac{1}{2}, \frac{1}{2}\right\}, \\ \frac{1}{2} & \text{if } x = -\frac{1}{2}, \\ -\frac{1}{2} & \text{if } x = \frac{1}{2}, \end{cases}\end{aligned}$$

where $f \neq u'$ only on the $\mathcal{L}^1$-negligible subset $\{-1/2, 1/2\}$. Now

$$\begin{aligned} |D\hat{u}|(\Omega) &= |f| \left(\mathcal{L}^1 + \delta_{-\frac{1}{2}} + \delta_{\frac{1}{2}} \right)(\Omega) \\ &= \int_{-1}^{1} |u'(x)|\, dx + \frac{1}{2}\delta_{-\frac{1}{2}}(\Omega) + \frac{1}{2}\delta_{\frac{1}{2}}(\Omega) \\ &= \mathrm{Var}_{(-1,1)}(u) + \frac{1}{2} + \frac{1}{2} = \mathrm{Var}_{(-1,1)}(u) + 1. \end{aligned}$$



(a) $u(x) = -x^3 + \frac{1}{2}x^2 + x$

(b) $\hat{u}(x) = \begin{cases} u(x) + \frac{1}{2} & \text{if } -\frac{1}{2} \leq x \leq \frac{1}{2} \\ u(x) & \text{otherwise} \end{cases}$

Figure 4.3.: $\hat{u}$ is in principle u but has two jumps, each with "length" 1/2. We would therefore expect $\hat{u}$ to have the variation of u plus the length of the jumps.

4.2.1. Structure of Weak Derivatives

For proofs and details regarding the statements in this subsection, see [3, 2, 8].

Since BV allows a concept of derivatives for functions with discontinuities, it is reasonable to expect the segmented images are contained in that space. Unfortunately, BV contains functions which are highly inappropriate in the context of image segmentation as well: Functions whose weak derivative may be decomposed into a measure $D_{\mathcal{L}} \ll \mathcal{L}^d$ and a measure with a density depending on jumps as described in remark 4.24 are not the only kind of functions with weak derivatives.

We define a sequence $(f_n)_{n\in\mathbb{N}}$ of continuous functions $f_n : [0,1] \to [0,1]$ as

$$f_1(x) = x \qquad f_{n+1}(x) = \begin{cases} \frac{1}{2} f_n(3x) & \text{for } 0 \le x \le \frac{1}{3}, \\ \frac{1}{2} & \text{for } \frac{1}{3} < x < \frac{2}{3}, \\ \frac{1}{2}\left(1 + f_n\left(3\left(x - \frac{2}{3}\right)\right)\right) & \text{for } \frac{2}{3} \le x \le 1, \end{cases}$$

see also figure 4.4. $\left(f_n\right)$ uniformly converges to a function $f \in \mathrm{BV}\big((0,1)\big)$, the *Cantor-Vitali* function. This function is $\mathcal{L}^1$-a.e. constant on $[0,1]$ (since f is not constant only on the $\mathcal{L}^1$-negligible Cantor set) and especially $f' = 0$ $\mathcal{L}^1$-almost everywhere. Yet still, f is continuous and monotonically increasing on $[0,1]$ with $f(0) = 1$ and $f(1) = 1$.

To analyze the structure of weak derivatives, we need the following terms and definitions:

Definition 4.26 Let $u : \Omega \to \mathbb{R}$. $l \in \mathbb{R}$ is called the *approximate limit* of f in $x \in \Omega$, denoted

$$\operatorname*{ap\,lim}_{y\to x} f(y) = l,$$

if for all $\varepsilon > 0$

$$\lim_{r\to 0} \frac{\mathcal{L}^d\big(B_r(x) \cap \{|f - l| \ge \varepsilon\}\big)}{\mathcal{L}^d\big(B_r(x)\big)} = 0.$$

In essence this means l is the approximate limit of f in x, if f evaluates to a value close to l almost everywhere in the vicinity of x.

The approximate limit may be used to define the *approximate derivate*:

Definition 4.27 Let $u : \Omega \to \mathbb{R}$. u is called *approximately differentiable* at $x \in \Omega$, if there exists a linear mapping $L : \Omega \to \mathbb{R}$ such that

$$\operatorname*{ap\,lim}_{y\to x} \frac{|f(y) - f(x) - L(y - x)|}{|y - x|} = 0.$$

L is called the *approximate derivate* of u in x.

Definition 4.28 The *jump set* S_u of u is the set of all points where u has no approximate limit.

It turns out that the weak derivative of every $u \in \mathrm{BV}(\Omega)$ may be decomposed into three mutually singular measures:

$$Du = D_l u + D_j u + D_c u \tag{4.4}$$

where

- $D_l u = \nabla u \mathcal{L}^d$ is a weighting of the Lebesgue measure, where $\nabla u(x)$ is the approximate derivative for $\mathcal{L}^d$-a.e. $x \in \Omega$ and it is called the *Lebesgue part*,
- $D_j u$ is a weighting of the restricted Hausdorff-measure $\mathcal{H}^{d-1} \llcorner S_u$, called the *Jump part*,
- $D_c u$ is the so called *Cantor part*.

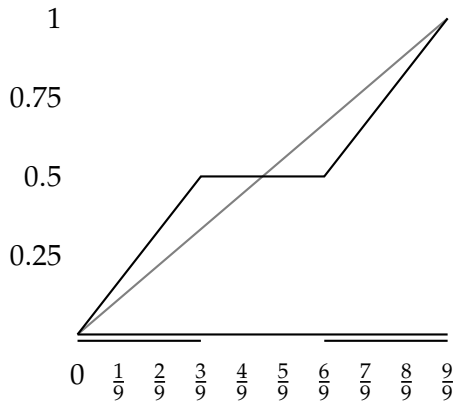
For functions like the Cantor-Vitali function f it holds $D_l f = D_j f = 0$. These kinds of functions are dense in $\mathrm{L}^2(\Omega)$, which means that (in anticipation of the functional we will define in section 4.3) for any input $f \in \mathrm{L}^2(\Omega)$ and for all $\alpha, \lambda > 0$

$$\inf_{u \in \mathrm{BV}(\Omega)} \int_\Omega (u - f)^2 \, d\mathcal{L}^d + \alpha \int_{\Omega \setminus K} |\nabla u|^2 \, d\mathcal{L}^d + \lambda \int_K d(\mathcal{H}^{d-1} \llcorner S_u) = 0.$$

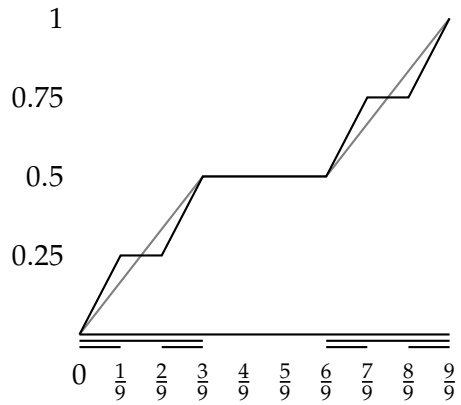
After all, functions whose weak derivatives have a Cantor part are not well suited to represent images, and we will restrict the function space to those functions whose weak derivatives don't have a Cantor part.

Definition 4.29 The space of *special functions of bounded variation* is defined as

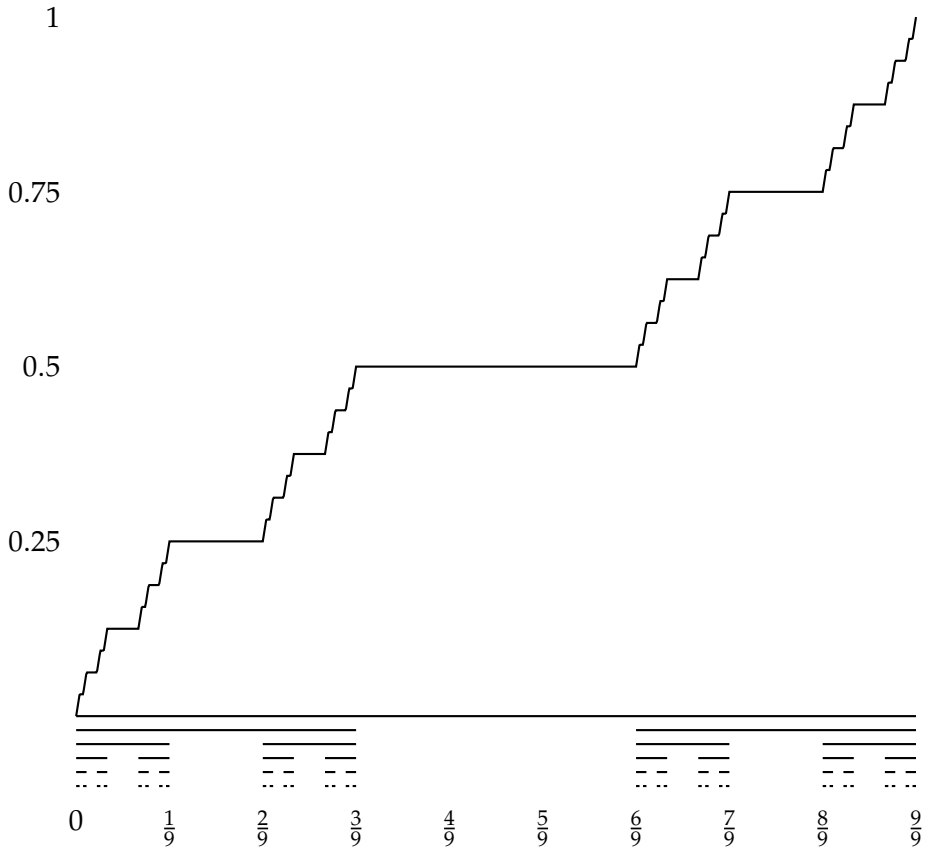
$$\mathrm{SBV}(\Omega) = \left\{ u \in \mathrm{BV}(\Omega) : D_c u = 0 \text{ in equation (4.4)} \right\}.$$



(a) Constructing f_2 from f_1



(b) Constructing f_3 from f_2



(c) f_8

Figure 4.4.: The Cantor-Vitali function, also known as *the devil's staircase*

4.3. The Mumford-Shah Functional

We are finally able to formulate a segmentation in the sense of remark 1.3 as a variational problem on SBV (Ω). We can fulfill the "wishes" in remark 1.3 in the following way:

1. u and f should be similar, which we can model by requiring

$$\int_{\Omega} (u - f)^2 \, d\mathcal{L}^d \tag{4.5}$$

   to be small.
2. Inside each region, u should be more or less homogeneous, which we can model by requiring

$$\int_{\Omega \setminus K} |\nabla u|^2 \, d\mathcal{L}^d \tag{4.6}$$

   to be small. Here, $K \subset \Omega$ is the set of the borders that are separating the different regions and ∇u is the density of the Lebesgue part in Du.
3. Small artifacts, noise or little dirt particles should be ignored and not assigned an own region. In other words, the set K should be small, so we require

$$\mathcal{H}^{d-1}(K) = \int_{K} d\mathcal{H}^{d-1} \tag{4.7}$$

   to be small.

The combination of equations (4.5) to (4.7) with parameters $\alpha, \lambda > 0$ forms the Mumford-Shah functional:

Definition 4.30 (Mumford-Shah functional) Let $f : \Omega \to \mathbb{R}$, $\alpha, \lambda > 0$ and $K \subset \Omega$ be a closed subset. The *Mumford-Shah functional* $\mathcal{F}_{\mathrm{MS}}$ is defined as

$$\mathcal{F}_{\mathrm{MS}}(u, K) = \int_{\Omega} (u - f)^2 \, d\mathcal{L}^d + \alpha \int_{\Omega \setminus K} |\nabla u|^2 \, d\mathcal{L}^d + \lambda \int_{K} d\mathcal{H}^{d-1}.$$

Theorem 4.31 Let $\Omega \subset \mathbb{R}^d$ be a bounded, open set, $\alpha, \lambda > 0$ and $f \in \mathrm{L}^\infty(\Omega)$. Then there exists a solution of

$$\min_{\substack{u \in \mathrm{SBV}(\Omega) \\ K \subset \Omega \text{ closed}}} \mathcal{F}_{\mathrm{MS}}(u, K).$$

Proof:
Luigi Ambrosio proved the existence in a series of publications he then summarized in [3]. The proof utilizes the *Direct Method of Variational Calculus* (in anticipation of remark 6.10):

The Mumford-Shah functional consists of the integration of non-negative functions with respect to non-negative measures, is consequently non-negative itself and bounded from below, so the first requirement in the direct method is fulfilled.

Ambrosio defines

$$\hat{\mathcal{F}}(u) := \int_\Omega (u-f)^2 \, d\mathcal{L}^d + \alpha \int_\Omega |\nabla u|^2 \, d\mathcal{L}^d + \lambda \int_{S_u} d\mathcal{H}^{d-1},$$

using $\mathcal{L}^d(\Omega \setminus K) = \mathcal{L}^d(\Omega)$ and where S_u is the jump set of u, and argues that

$$\inf_{u \in \mathrm{SBV}(\Omega)} \hat{\mathcal{F}}(u) \leq \inf_{\substack{u \in \mathrm{SBV}(\Omega) \\ K \subset \Omega}} \mathcal{F}_{\mathrm{MS}}(u, K).$$

Given a minimizer u of $\hat{\mathcal{F}}$ and setting $K = \Omega \cap \bar{S}_u$, the pair (u, K) will minimize $\mathcal{F}_{\mathrm{MS}}$.

Ambrosio then generalizes the problem to functionals on SBV (Ω) of the type

$$\mathcal{F}(u) = \int_\Omega f(x, u, \nabla u) \, d\mathcal{L}^d + \int_{S_u} \varphi \, d\mathcal{H}^{d-1}, \tag{4.8}$$

where $u : x \mapsto u(x)$ and with a quite general (here not further explained) density φ.

In chapter 3 of [3], a compactness theorem on SBV (Ω) which ensures the existence of a converging subsequence is proved and the second requirement of the direct method is fulfilled.

Finally, in chapter 4 of [3], the lower semi-continuity of functionals of the type (4.8) is proved, the third requirement is fulfilled and the direct method can be applied. □

While the existence of solutions certainly is desirable, we are equally interested in finding a way to compute these solutions. A great difficulty in the minimization of the Mumford-Shah functional is the very different nature of the two parameters u, K in $\mathcal{F}_{MS}(u, K)$: u is a function on Ω and K is a subset of Ω. As we have seen in the proof of theorem 4.31, the optimal set may be described in terms of the optimal image $\bar{u}$ as $\bar{K} = \Omega \cap S_{\bar{u}}$. But recalling definition 4.28, we realize that it is difficult to compute the jump set explicitly.

In chapter 5 we will consider images of objects in front of a background and how in this special case the Mumford-Shah functional can be minimized. In chapter 7, we will describe the set K in terms of the discrete gradient (obtained by finite differences) and define the *Fast-Mumford-Shah* functional which may be used in a more general setting than the special case above.

Chapter 5.

Active Contours: Separating Object and Background

Active contours ([6]) are a well-studied method to minimize a special case of the Mumford-Shah functional. We assume to have a two-dimensional image of an object in front of some background, separated by a smooth curve $C \subset \Omega$ with length $l(C)$. The segmentation task is to compute a piecewise constant segmentation with two regions Ω_o (object) and Ω_b (background) with $\Omega = \Omega_o \dot\cup \Omega_b$, such that Ω_o is open and $C = \partial\Omega_o$.

With the above and assuming that u evaluates to c_o on Ω_o and c_b on Ω_b, the Mumford-Shah functional reduces to

$$\mathcal{F}_{\mathrm{AC}}(c_o, c_b, C) \coloneqq \lambda_1 \int_{\Omega_o} \big(f(x) - c_o\big)^2 \, d\mathcal{L}^2(x) + \lambda_2 \int_{\Omega_b} \big(f(x) - c_b\big)^2 \, d\mathcal{L}^2(x) + l(C) \tag{5.1}$$

with parameters $\lambda_1, \lambda_2 > 0$.

For a fixed curve, the optimization with respect to c_o and c_b can be done analytically by differentiation:

$$\frac{\partial}{\partial c_o}\mathcal{F}_{\mathrm{AC}}(c_o, c_b, C) = 0 \qquad \Rightarrow \qquad c_o = \frac{\int_{\Omega_o} f \, d\mathcal{L}^2}{\int_{\Omega_o} d\mathcal{L}^2},$$

so c_o is the average of f on Ω_o and likewise c_b is the average of f on Ω_b.

The curve C might be represented as the zero level set of a level set function ϕ as follows:

$$\begin{aligned} C &= \big\{x \in \Omega : \phi(x) = 0\big\} \\ \Omega_o &= \big\{x \in \Omega : \phi(x) > 0\big\} \\ \Omega_b &= \big\{x \in \Omega : \phi(x) < 0\big\} \end{aligned}$$

Using ϕ and the Heaviside function H, defined by

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

we can obtain the characteristic function of the subset Ω_o as

$$\mathbb{1}_{\Omega_o} = H \circ \phi.$$

Now, we may rewrite the length of C as

$$l(C) = \mathcal{H}^1(\underbrace{\Omega_o \cap \Omega}_{=\Omega_o}) = P(\Omega_o, \Omega) = \int_\Omega d|D\mathbb{1}_{\Omega_o}|,$$

where $P(\Omega_o, \Omega)$ is the *perimeter* of Ω_o in Ω, defined as the variation of the weak derivative of the characteristic function of Ω_o, evaluated at the whole set Ω. See [19, section 5.4] and [8, chapter 5] for details and the theory of *sets of finite perimeter*.

Using the above, equation (5.1) can be reformulated as

$$\begin{aligned}\mathcal{F}_{\mathrm{AC}}(c_o, c_b, \phi) = \; & \lambda_1 \int_\Omega \left(f(x) - c_o\right)^2 H\left(\phi(x)\right)\, d\mathcal{L}^2(x) \\ & + \lambda_2 \int_\Omega \Big(f(x) - c_b\Big)^2 \Big(1 - H\left(\phi(x)\right)\Big)\, d\mathcal{L}^2(x) \\ & + \int_\Omega d|D\mathbb{1}_{\Omega_o}|,\end{aligned}$$

where all the integrals are over the whole set Ω now. The authors in [6] then compute the Euler-Lagrange equation (with respect to ϕ) of a regularized version of the above functional and introduce an artificial time $t \geq 0$ in $\Phi(t, x)$, where $\Phi(0, x) = \phi_o(x)$ is an initial curve. They further specify algorithm 2 to iteratively evolve this initial curve ϕ_0 and argue that ϕ^{k+1} is obtained from ϕ^k by propagation of the set Ω_o parallel to normal direction. See figure 5.1 for a visual demonstration and figure 5.2 for an example.

Algorithm 2 Active-Contour Iteration

1: **Initialize:** $\phi^0 = \phi_0$
2: **for** $k = 0, 1, 2 \ldots$ **do**
3: Compute the averages c_o, c_b for the partition defined by ϕ^k
4: Compute ϕ^{k+1} by solving the partial differential equation given by the Euler-Lagrange equation
5: **if** Termination criterion fulfilled **then**
6: break
7: **end if**
8: **end for**

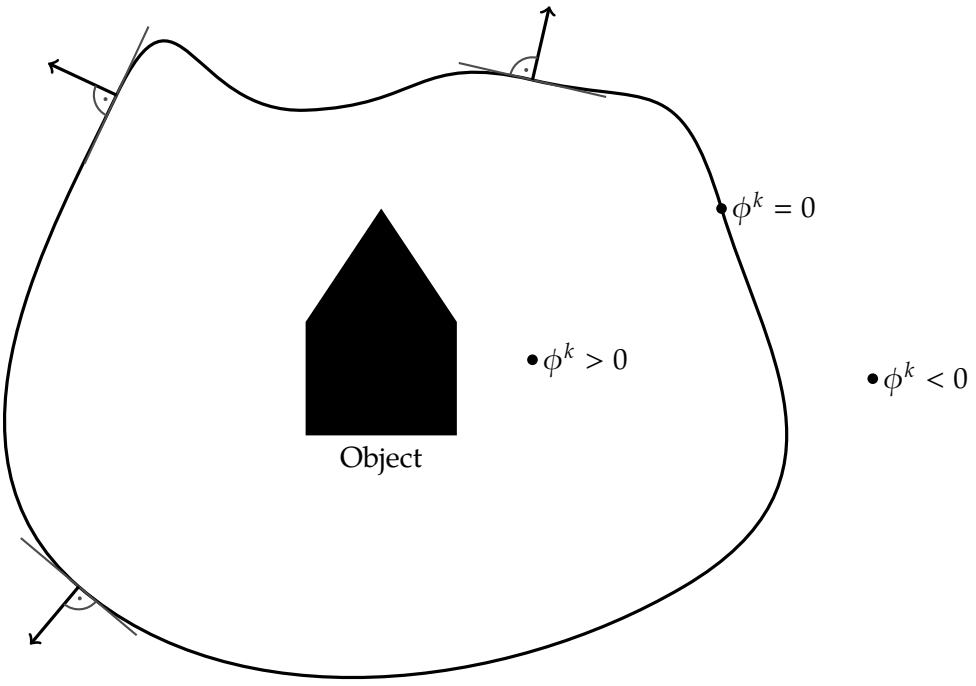
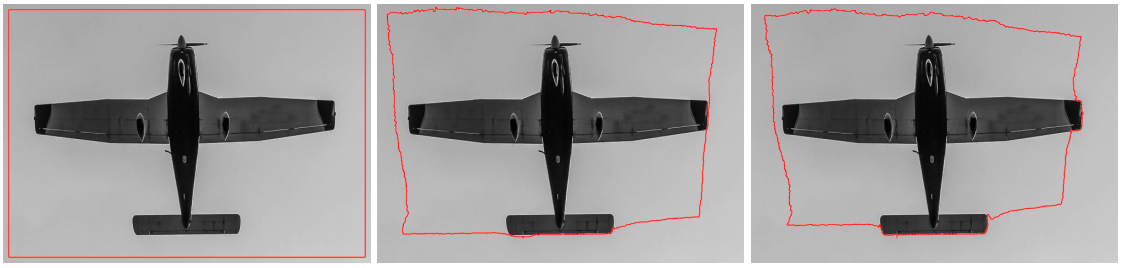
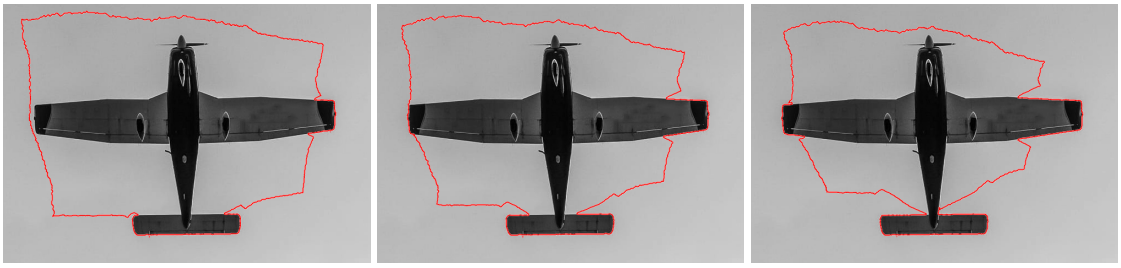


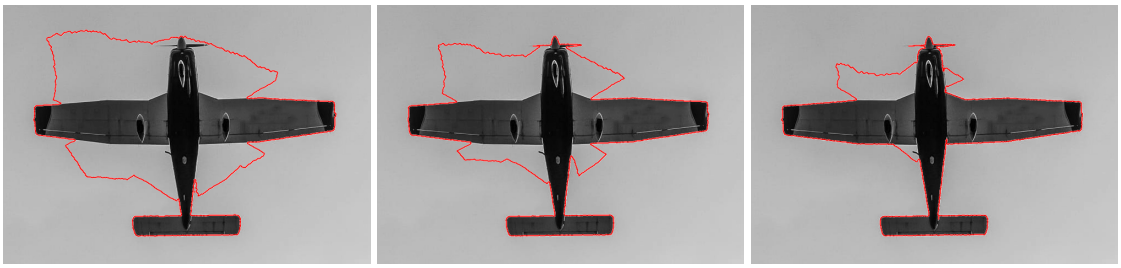
Figure 5.1.: Computing ϕ^{k+1} from ϕ^k corresponds to a propagation of the curve C parallel to normal direction.



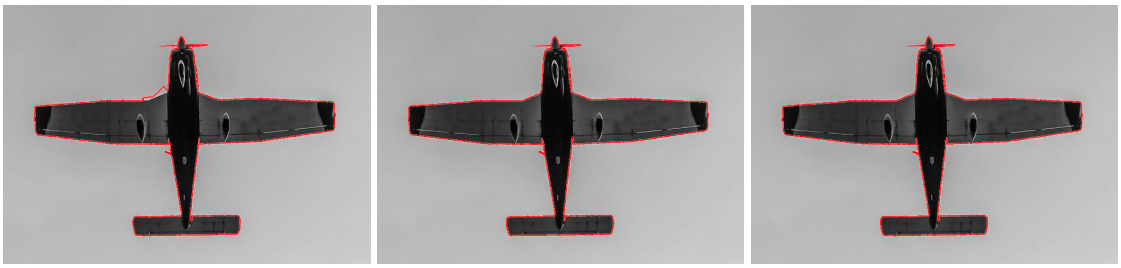
(a) Initial contour (b) 100 iterations (c) 200 iterations



(d) 300 iterations (e) 400 iterations (f) 500 iterations



(g) 600 iterations (h) 700 iterations (i) 800 iterations



(j) 900 iterations (k) 1000 iterations (l) 1100 iterations

Figure 5.2.: Iteratively updating the initial contour until it separates object and background after around 1000 iterations.

Chapter 6.

Optimization of Variational Problems in Image Processing

In this chapter, we will develop a general framework to analyze and (numerically) optimize variational problems of the type

$$\bar{u} = \arg\min_{u \in X} F(u) + G(Au)$$

where F, G are functionals on spaces X, Y and $A : X \to Y$ is a linear operator between these spaces. While there are certain requirements on F, G, they are not required to be differentiable. This framework will then be used in the upcoming chapter 7.

This chapter follows [10, chapter 2] and [4, chapter 6]. If not explicitly stated otherwise, in this chapter let X and Y be Banach spaces over $\mathbb{R}$.

6.1. Convex Analysis

Definition 6.1 (Domain and proper functional) A functional $f : X \to \mathbb{R}_\infty$ is called *proper*, if its *domain*

$$\operatorname{dom} F := \left\{x \in X : f(x) < \infty\right\}$$

is non-empty.

Definition 6.2 (Convex set) A subset $C \subset X$ is called *convex*, if for every $x, y \in C$ and for every $\lambda \in [0, 1]$ it holds

$$\lambda x + (1 - \lambda) y \in C.$$

Definition 6.3 (Convex function) Let $C \subset X$ be convex and let $f : C \to \mathbb{R}_\infty$ be a function on C.

f is called *convex*, if for all $x, y \in C$ and for every $\lambda \in [0, 1]$

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y).$$

f is called *strictly convex*, if for all $x, y \in C$ with $x \neq y$ and for every $\lambda \in (0,1)$

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y).$$

f is called *strongly convex* with parameter $\gamma > 0$, if for all $x, y \in C$ and for every $\lambda \in [0,1]$

$$f(\lambda x + (1-\lambda)y) + \frac{\gamma}{2}\lambda(1-\lambda)\|x-y\|^2 \leq \lambda f(x) + (1-\lambda) f(y).$$

Remark 6.4 By definition 6.3, every strictly convex function is also convex. And since for $\gamma > 0$, $x, y \in C$ with $x \neq y$ and $\lambda \in (0,1)$

$$\frac{\gamma}{2}\lambda(1-\lambda)\|x-y\|^2 > 0,$$

every strongly convex function is also strictly convex.

Lemma 6.5 Let $C \subset X$ be convex and let $f, g : C \to \mathbb{R}_\infty$ be convex.

1. $(f+g)$ is convex. If g is strictly convex, then $(f+g)$ is strictly convex.
2. If $\alpha > 0$, then (αf) is convex. If f is strictly convex, then (αf) is strictly convex.
3. If $g : \mathbb{R}_\infty \to \mathbb{R}_\infty$ is convex and monotonically increasing, then $g \circ f$ is convex. If f is strictly convex and g strictly monotonically increasing, then $g \circ f$ is strictly convex.

Proof:
Let $f, g : C \to \mathbb{R}_\infty$ convex, $x, y \in C$, $\alpha > 0$ and $\lambda \in (0,1)$. Then

1.
$$\begin{aligned}
&(f+g)\big(\lambda x + (1-\lambda)y\big) \\
&= f\big(\lambda x + (1-\lambda)y\big) + g\big(\lambda x + (1-\lambda)y\big) \\
&\leq \lambda f(x) + (1-\lambda) f(y) + \lambda g(x) + (1-\lambda) g(y) \\
&= \lambda (f+g)(x) + (1-\lambda)(f+g)(y)
\end{aligned}$$

2. Multiplication with $\alpha > 0$ does not affect the "$\leq$" and "$<$" relations in definition 6.3.
3.
$$g\Big(f\big(\lambda x + (1-\lambda)y\big)\Big) \leq g\big(\lambda f(x) + (1-\lambda) f(y)\big) \leq \lambda g\big(f(x)\big) + (1-\lambda) g\big(f(y)\big)$$

The proofs for strict convexity may be carried out analogously. □

Lemma 6.6 Let $f : X \to \mathbb{R}_\infty$, $\hat{x} \in X$ and $g : X \to \mathbb{R}_\infty$ with $g(x) \coloneqq f(x + \hat{x})$.

$$\text{If } f \text{ is } \begin{cases}\text{convex,}\\ \text{strictly convex,}\\ \text{strongly convex,}\end{cases} \quad \text{then } g \text{ is also } \begin{cases}\text{convex.}\\ \text{strictly convex.}\\ \text{strongly convex.}\end{cases}$$

Proof:
Let $x, y \in X$, $\lambda \in [0,1]$ and assume f is convex. Then

$$\begin{aligned} & g\big(\lambda x + (1-\lambda)y\big) && (6.1)\\ &= f\big(\lambda x + (1-\lambda)y + \hat{x}\big) && (6.2)\\ &= f\big(\lambda x + \lambda\hat{x} + (1-\lambda)y + \hat{x} - \lambda\hat{x}\big) && (6.3)\\ &= f\big(\lambda(x+\hat{x}) + (1-\lambda)(y+\hat{x})\big) && (6.4)\\ &\leq \lambda f(x+\hat{x}) + (1-\lambda)f(y+\hat{x}) && (6.5)\\ &= \lambda g(x) + (1-\lambda)g(y) && (6.6)\end{aligned}$$

and g is also convex. For $x \neq y$, $\lambda \in (0,1)$ and strictly convex f, the proof may be carried out the same way, replacing the "$\leq$" between equations (6.4) and (6.5) by "$<$". If f is strongly convex with $\gamma > 0$, the proof may be carried out by adding

$$\frac{\gamma}{2}\lambda(1-\lambda)\|x-y\|^2$$

to equations (6.1) to (6.4). □

Lemma 6.7 $\|\cdot\|_X^p$ is convex for $p \geq 1$.

Proof:
Let $x, y \in X$ and $\lambda \in [0,1]$. Since $\|\cdot\|_X$ is absolutely homogeneous and subadditive,

$$\|\lambda x + (1-\lambda)y\| \leq \lambda\|x\| + (1-\lambda)\|y\|.$$

For $p \geq 1$, convexity is guaranteed by lemma 6.5. □

Lemma 6.8 In a Hilbert space X, $\|\cdot\|_X^2$ defined as $\|x\|_X \coloneqq \sqrt{(x,x)}$ is strongly convex with parameter $\gamma = 2$.

Proof:
We show that $\|\cdot\|_X^2$ fulfills the criterion for strong convexity in definition 6.3 for $\gamma = 2$.

Let $x, y \in X$ and $\lambda \in [0,1]$ be arbitrary. Then

$$\begin{aligned}
&\|\lambda x + (1-\lambda)y\|_X^2 + \lambda(1-\lambda)\|x-y\|_X^2 \\
=\ & \lambda^2(x,x) + 2\lambda(1-\lambda)(x,y) + (1-\lambda)^2(y,y) \\
&+ \lambda(1-\lambda)\big((x,x) - 2(x,y) + (y,y)\big) \\
=\ & \lambda^2(x,x) + \lambda(x,x) - \lambda^2(x,x) \\
&+ 2\lambda(1-\lambda)(x,y) - 2\lambda(1-\lambda)(x,y) \\
&+ (1-\lambda)^2(y,y) + \lambda(y,y) - \lambda^2(y,y) \\
=\ & \lambda(x,x) + (1-\lambda)(y,y) \\
=\ & \lambda\|x\|_X^2 + (1-\lambda)\|y\|_X^2.
\end{aligned}$$

□

6.2. Existence and Uniqueness of Solutions

Definition 6.9 Let $F : X \to \mathbb{R}_\infty$. If for all sequences $(u_n)_{n\in\mathbb{N}}$ it holds

- $\lim_{n\to\infty} \|u_n\| = \infty \Rightarrow \lim_{n\to\infty} F(u_n) = \infty$, then F is called *coercive*.
- $u_n \to u \Rightarrow F(u) \le \liminf_{n\to\infty} F(u_n)$, then F is called *lower semi-continuous*.

Remark 6.10 (Direct Method of Variational Calculus) Let X be a topological space and $F : X \to \mathbb{R}$. To prove the existence of a minimizer, conduct the following steps:

1. Verify that F is bounded from below, which means there is a sequence (u_n) with
$$\lim_{n\to\infty} F(u_n) = \inf_{u\in X} F(u).$$
2. Choose a topology on X such that (u_n) has a subsequence u_{n_k} which converges to $\bar{u} \in X$.
3. Show that F is lower semi-continuous with respect to the topology from step 2 (see definition 6.9). Then
$$\inf_{u\in X} F(u) \le F(\bar{u}) \le \liminf_{k\to\infty} F(u_{n_k}) = \inf_{u\in X} F(u)$$
and $\bar{u}$ is a minimizer of F.

Note that the steps 2 and 3 are competing: when a stronger topology is chosen, on the one hand the lower semi-continuity of F has to be shown for fewer sequences, but on the other hand there are also fewer sequences with a converging subsequence. See also [4, section 6.2.1].

Theorem 6.11 (Existence of solutions) Let X be a reflexive Banach space and $F : X \to \mathbb{R}_\infty$ be bounded from below, coercive, convex and lower semi-continuous. Then there exists a solution to the minimization problem

$$\min_{u \in X} F(u)$$

Proof:
This theorem is a combination of theorem 6.17 and corollary 6.28 in [4]. □

Theorem 6.12 (Uniqueness of Solutions) If $F : X \to \mathbb{R}_\infty$ is strictly convex, the minimization problem

$$\min_{u \in X} F(u)$$

has at most one solution.

Proof:
Assume there exists $u, v \in X, u \neq v$ such that $F(u) = F(v) = \inf_{u \in X} F(u)$. Then, for an arbitrary $\lambda \in (0, 1)$,

$$F\big(\lambda u + (1-\lambda)v\big) < \lambda F(u) + (1-\lambda)F(v) = \inf_{u \in X} F(u),$$

which is a contradiction. □

6.3. Subdifferential Calculus

Definition 6.13 (Gâteaux-derivative) Let $F : X \to \mathbb{R}_\infty$. For $x \in X$, $h \in X$ let

$$\delta F(x; h) := \lim_{t \to 0} \frac{F(x + th) - F(x)}{t}.$$

If for $x \in X$ and for every $h \in X$ the limit $\delta F(x; h)$ exists such that there is a linear operator $F'(x) \in X^*$ with

$$\delta F(x; h) = (F'(x))\, h,$$

then $F'(x)$ is called *Gâteaux-derivative* of F in x. If the Gâteaux-derivative exists for every $x \in X$, F is called *Gâteaux-differentiable*.

Remark 6.14 (Theorem 6.33 in [4]) Let $F : X \to \mathbb{R}_\infty$ be convex and Gâteaux-differentiable and u be an interior point of X. Then $w = F'(u)$ is the unique operator in X^* such that

$$F(u) + \langle w, v - u \rangle \leq F(v) \qquad \forall v \in X. \tag{6.7}$$

The concept of derivatives is essential in numerical optimization. Unfortunately, many interesting functionals in image processing are not Gâteaux-differentiable. In order to broaden the concept of differentiation, we use equation (6.7) to define subgradients and the subdifferential:

Definition 6.15 (Subgradient and Subdifferential) Let $F : X \to \mathbb{R}_\infty$ be convex. $w \in X^*$ is called *subgradient* in u if it fulfills the *subgradient inequality*

$$F(u) + \langle w, v - u \rangle \leq F(v) \qquad \forall v \in X. \tag{6.8}$$

Given $u \in X$, the *subdifferential* of F in u is the set of all subgradients:

$$\partial F(u) \coloneqq \left\{ w \in X^* \mid u \text{ and } w \text{ fulfill the subgradient inequality (6.8)} \right\}$$

Because for $u \in X$ there might be several subgradients $w \in X^*$, the subdifferential is not a mapping $\partial F : X \to X^*$, it is a multi-valued operator:

Definition 6.16 (Multi-valued operator) A *multi-valued* operator $F : X \rightrightarrows Y$ is a subset $F \subset X \times Y$. We will write

$$F(x) = \left\{ y \in Y \mid (x, y) \in F \right\} \qquad \text{and} \qquad y \in F(x) :\Leftrightarrow (x, y) \in F.$$

For $F, G : X \rightrightarrows Y$ and $\lambda \in \mathbb{R}$ we define

$$\begin{aligned} (F + G)(x) &\coloneqq \left\{ y_f + y_g \mid y_f \in F(x), y_g \in G(x) \right\}, \\ (\lambda F)(x) &\coloneqq \left\{ \lambda y \mid y \in F(x) \right\}. \end{aligned}$$

For $F : X \rightrightarrows Y$ the inverse $F^{-1} : Y \rightrightarrows X$ is defined as

$$F^{-1} \coloneqq \left\{ (y, x) \in Y \times X \mid (x, y) \in F \right\}.$$

The identity $\mathrm{id} : X \rightrightarrows X$ is defined as

$$\mathrm{id} \coloneqq \{ (x, x) \in X \times X \}.$$

Remark 6.17 If $F : X \rightrightarrows X$, but for every $x \in X$ there exists exactly one $y \in Y$ such that $(x, y) \in Y$, we will implicitly treat F also as a mapping $F : X \to X$ and write $y = F(x)$.

Lemma 6.18 Let $F, G : X \to \mathbb{R}_\infty$ be convex and proper and $\lambda > 0$. Assume it exists $\hat{u} \in \operatorname{dom} F \cap \operatorname{dom} G$ such that F is continuous in $\hat{u}$. Then

1. $\partial(\lambda F) = \lambda \partial F$,
2. $\partial (F + G) = \partial F + \partial G$.

Proof:
For a proof see [4, theorem 6.51]. □

Theorem 6.19 (Generalization of remark 2.17 and theorem 6.43 in [4]) Let $F : X \to \mathbb{R}_\infty$ be convex. It holds

$$\bar{u} = \arg\min_{u \in X} F(u) \quad \Leftrightarrow \quad 0 \in \partial F(\bar{u}).$$

Proof:

$$\begin{aligned} & \bar{u} = \arg\min_{u \in X} F(u) \\ \Leftrightarrow \quad & F(\bar{u}) \leq F(u) && \forall u \in X \\ \Leftrightarrow \quad & F(\bar{u}) + \langle 0, u - \bar{u} \rangle \leq F(u) && \forall u \in X \\ \Leftrightarrow \quad & 0 \in \partial F(\bar{u}) \end{aligned}$$

□

Definition 6.20 (Proximal operator) Let X be a Hilbert space and $F : X \to \mathbb{R}_\infty$ be proper, convex, lower semi-continuous and let $\sigma > 0$. The *proximal operator* of F with respect to σ is defined as

$$\text{prox}_{\sigma,F} := \begin{cases} X & \to X \\ u & \mapsto \arg\min_{v \in X} \frac{\|v-u\|^2}{2\sigma} + F(v). \end{cases}$$

By lemmata 6.5 and 6.8, $\frac{\|v-u\|^2}{2\sigma} + F(v)$ is strictly convex and by section 6.2, there is a unique minimizer.

In section 6.5, for a given $F : X \rightrightarrows X$ and $\sigma > 0$, we want to derive multi-valued operators like $(\text{id} + \sigma\partial F)^{-1} : X \rightrightarrows X$. It turns out that for certain f, these types of multi-valued operators reduce to the proximal operator.

Definition 6.21 (Resolvent operator) If $F : X \rightrightarrows X$ is a multi-valued operator and $\sigma > 0$, then $(\text{id} + \sigma\partial F)^{-1} : X \rightrightarrows X$ is called *resolvent operator* of F with respect to σ.

Lemma 6.22 Let X be a real Hilbert space, $F : X \to \mathbb{R}_\infty$ be proper, convex, lower semi-continuous, and let $\sigma > 0$. Then the resolvent operator and proximal operator of F with respect to sigma coincide:

$$(\text{id} + \sigma\partial F)^{-1} = \text{prox}_{\sigma,F}$$

Proof:
For a proof see [4, lemma 6.134]. □

Lemma 6.23 Let X be a real Hilbert space, $F : X \to \mathbb{R}_\infty$ be proper, convex, lower semi-continuous, and let $\sigma > 0$.

$$G(u) = F(u) + \alpha, \alpha \in \mathbb{R} \quad \Rightarrow \quad \text{prox}_{\sigma,G} = \text{prox}_{\sigma,F} \tag{6.9}$$

$$G(u) = F(u) + (u, w), w \in X \quad \Rightarrow \quad \text{prox}_{\sigma,G}(u) = \text{prox}_{\sigma,F}(u - \sigma w) \tag{6.10}$$

Proof:

- Equation (6.9) holds, because addition of a real constant only changes the value, but not the argument of the minimization problem in definition 6.20.
- Equation (6.10): Let $G(u) = F(u) + (u, w)$.

$$\begin{aligned}
&\text{prox}_{\sigma,G}(u - \sigma w) \\
= \ &\arg\min_{v \in X} \frac{\|v - u + \sigma w\|^2}{2\sigma} + F(v) \\
= \ &\arg\min_{v \in X} \frac{(v - u, v - u) + 2\,(v - u, \sigma w) + (\sigma w, \sigma w)}{2\sigma} + F(v) \\
= \ &\arg\min_{v \in X} \frac{(v - u, v - u) + 2\sigma\big((v, w) - (u, w)\big) + (\sigma w, \sigma w)}{2\sigma} + F(v) \\
= \ &\arg\min_{v \in X} \frac{\|v - u\|^2}{2\sigma} + F(v) + (v, w) = \text{prox}_{\sigma,G}(u),
\end{aligned}$$

  where we used that terms independent of v don't change the argument of the minimization problem. □

6.4. Duality in Optimization

Many variational methods in image processing may be written in the following way: Find $\bar{u} \in X$ such that

$$\bar{u} = \arg\min_{u \in X} F(u) + G(Au), \tag{6.11}$$

where $F : X \to \mathbb{R}_\infty$, $G : Y \to \mathbb{R}_\infty$ are proper, convex and lower semi-continuous and $A : X \to Y$ is a linear operator. Note that F, G are not required to be differentiable. Equation (6.11) is called the *primal problem*.

Definition 6.24 Let $F : X \to \mathbb{R}_\infty$ be proper. The *Fenchel-conjugate* $F^* : X^* \to \mathbb{R}_\infty$ (also called *convex-conjugate*) of F is defined as

$$F^*(w) := \sup_{u \in X} (\langle w, u \rangle - F(u)).$$

For every $(u, w) \in \operatorname{dom} F \times \operatorname{dom} F^*$ definition 6.24 directly gives the *Fenchel-inequality*

$$\langle w, u \rangle \leq F(u) + F^*(w). \tag{6.12}$$

For a proper $F : X \to \mathbb{R}_\infty$ we also define the Fenchel-biconjugate $F^{**} : X \to \mathbb{R}_\infty$ as

$$F^{**}(u) := \sup_{w \in X^*} (\langle w, u \rangle - F^*(w)).$$

If F is convex and lower semi-continuous, then $F^{**} = F$ (see [4, remark 6.62 and lemma 6.63]).

Lemma 6.25 Let $F : X \to \mathbb{R}_\infty$. Then F^* is convex.

Proof:
Let $v, w \in X^*$ and $\lambda \in [0, 1]$.

$$\begin{aligned}
F^*\big(\lambda v + (1-\lambda)w\big) &= \sup_{u \in X} \big(\langle \lambda v + (1-\lambda)w, u \rangle - F(u)\big) \\
&= \sup_{u \in X} \big(\lambda \langle v, u \rangle + (1-\lambda) \langle w, u \rangle - F(u)\big) \\
&= \sup_{u \in X} \big(\lambda \langle v, u \rangle - \lambda F(u) + (1-\lambda) \langle w, u \rangle - (1-\lambda) F(u)\big) \\
&\leq \sup_{u, \hat{u} \in X} \big(\lambda \langle v, u \rangle - \lambda F(u) + (1-\lambda) \langle w, \hat{u} \rangle - (1-\lambda) F(\hat{u})\big) \\
&= \lambda \sup_{u \in X} \big(\langle v, u \rangle - F(u)\big) + (1-\lambda) \sup_{\hat{u} \in X} \big(\langle w, \hat{u} \rangle - F(\hat{u})\big) \\
&= \lambda F^*(v) + (1-\lambda) F^*(w)
\end{aligned}$$

□

Lemma 6.26 Let $F : X \to \mathbb{R}_\infty$ be proper.

$$G(u) = \lambda F(u) \quad \text{with } \lambda > 0 \quad \Rightarrow \quad G^*(w) = \lambda F^*\left(\frac{w}{\lambda}\right) \tag{6.13}$$

$$G(u) = F(u - \hat{u}) \text{ with } \hat{u} \in X \quad \Rightarrow \quad G^*(w) = F^*(w) - \langle w, \hat{u} \rangle \tag{6.14}$$

Proof:
These are special cases of lemma 6.65 in [4]. □

Lemma 6.27 Let $F : X \to \mathbb{R}_\infty$ be proper, convex and lower semi-continuous. For $x \in X$, $x^* \in X^*$ it holds

$$x^* \in \partial F(x) \quad \Leftrightarrow \quad x \in \partial F^*(x).$$

Proof:
For a proof see [14]. □

Lemma 6.28 (Moreau decomposition) Let $F : X \to \mathbb{R}_\infty$ be proper, convex, lower semi-continuous and let $\sigma > 0$. For every $u \in X$ it holds

$$u = \operatorname{prox}_{\sigma,F}(u) + \sigma \cdot \operatorname{prox}_{\frac{1}{\sigma},F^*}\left(\frac{u}{\sigma}\right). \tag{6.15}$$

Proof:
The proof for the case $X = \mathbb{R}^d$ and $\sigma = 1$ can be found in [15]. For the sake of completeness, we prove the lemma for a Banach space X and also start with the case $\sigma = 1$.

$$\begin{aligned}
& x = \operatorname{prox}_{1,F}(u) \\
\Leftrightarrow\ & x = \arg\min_{v \in X} \frac{\|v-u\|^2}{2} + F(v) \\
\Leftrightarrow\ & 0 \in (x-u) + \partial F(x) && \text{(“Classical” optimization and theorem 6.19)} \\
\Leftrightarrow\ & (u-x) \in \partial F(x) \\
\Leftrightarrow\ & x \in \partial F^*(u-x) && \text{(Lemma 6.27)} \\
\Leftrightarrow\ & u - (u-x) \in \partial F^*(u-x) \\
\Leftrightarrow\ & u - x = \operatorname{prox}_{1,F^*}(u) && \text{(Same argumentation using theorem 6.19 as above)}
\end{aligned}$$

And inserting the first in the last equation shows

$$u = \operatorname{prox}_{1,F}(u) + \operatorname{prox}_{1,F^*}(u). \tag{6.16}$$

We obtain the general case with $\sigma > 0$ by inserting σF in equation (6.16):

$$\begin{aligned}
& u = \operatorname{prox}_{1,\sigma F}(u) + \operatorname{prox}_{1,(\sigma F)^*}(u) \\
\Leftrightarrow\ & u = \operatorname{prox}_{\sigma,F}(u) \ + \operatorname{prox}_{1,(\sigma F)^*}(u) \\
\Leftrightarrow\ & u = \operatorname{prox}_{\sigma,F}(u) \ + \arg\min_{v \in X} \frac{\|v-u\|^2}{2} + (\sigma F)^*(v) \\
\Leftrightarrow\ & u = \operatorname{prox}_{\sigma,F}(u) \ + \arg\min_{v \in X} \frac{\|v-u\|^2}{2} + \sigma F^*\left(\frac{v}{\sigma}\right) && \text{(Equation (6.13))} \\
\Leftrightarrow\ & u = \operatorname{prox}_{\sigma,F}(u) \ + \sigma \cdot \arg\min_{v \in X} \frac{\|\sigma v-u\|^2}{2} + \sigma F^*(v) \\
\Leftrightarrow\ & u = \operatorname{prox}_{\sigma,F}(u) \ + \sigma \cdot \arg\min_{v \in X} \frac{\sigma^2\|v-\frac{u}{\sigma}\|^2}{2} + \sigma F^*(v)
\end{aligned}$$

(Factor $\sigma > 0$ does not change minimizing argument)

$$\begin{aligned}
\Leftrightarrow\ & u = \operatorname{prox}_{\sigma,F}(u) \ + \sigma \cdot \arg\min_{v \in X} \frac{\sigma\|v-\frac{u}{\sigma}\|^2}{2} + F^*(v) \\
\Leftrightarrow\ & u = \operatorname{prox}_{\sigma,F}(u) \ + \sigma \cdot \operatorname{prox}_{\frac{1}{\sigma},F^*}\left(\frac{u}{\sigma}\right)
\end{aligned}$$

□

Remark 6.29 If $F = F^{**}$, equation (6.15) also holds with F and F^{**} swapped:

$$u = \text{prox}_{\sigma,F^*}(u) + \sigma \text{prox}_{\frac{1}{\sigma},F}\left(\frac{u}{\sigma}\right)$$

Theorem 6.30 (Fenchel-Rockafellar duality, theorem 6.68 in [4]) Let $F : X \to \mathbb{R}_\infty$ and $G : Y \to \mathbb{R}_\infty$ be proper, convex, lower semi-continuous and let $A : X \to Y$ be a linear operator. Further, assume there is $\bar{u} \in X$ such that

$$\bar{u} = \arg\min_{u \in X} F(u) + G(Au).$$

If there is an $u_0 \in X$ such that $F(u_0) < \infty$, $G(Au_0) < \infty$ and such that G is continuous in Au_0. Then

$$\max_{w \in Y^*} -F^*(-A^*w) - G^*(w) = \min_{u \in X} F(u) + G(Au). \tag{6.17}$$

Definition 6.31 (Dual problem) The optimization problem to find $\bar{w} \in Y^*$ such that

$$\bar{w} = \arg\max_{w \in Y^*} -F^*(-A^*w) - G^*(w) \tag{6.18}$$

is called *dual problem* to the primal problem (6.11). The task to find $(\bar{u}, \bar{w}) \in X \times Y^*$ such that $\bar{u}$ solves the primal problem (6.11) and $\bar{w}$ solves the dual problem (6.18) is called *primal-dual problem*.

The dual problem has nice properties. By lemma 6.25, we know that F^* and G^* are convex even if F and G are not. We call a functional J *concave* if $-J$ is convex and since $-F^*$ and $-G^*$ are concave, the dual problem (6.18) is always concave. The theory of convex minimization may be applied to concave maximization analogously and therefore it might be easier to solve the dual problem than the primal problem. In addition, theorem 6.30 describes a correspondence between the primal problem we intend to solve and the related dual problem.

Unfortunately, equation (6.17) describes a correspondence between the *optimal values*, not between the *optimal arguments* we are interested in. The duality concept will still turn out to be useful.

Definition 6.32 The functional $L : \text{dom}\, F \times \text{dom}\, G^* \to \mathbb{R}$ with

$$L(u, w) \coloneqq \langle w, Au \rangle + F(u) - G^*(w)$$

is called the *Lagrange-functional* of F, G and A.
$(\hat{u}, \hat{w}) \in \text{dom}\, F \times \text{dom}\, G^*$ is called *saddle-point* of L, if for all $(u, w) \in \text{dom}\, F \times \text{dom}\, G^*$

$$L(\hat{u}, w) \leq L(\hat{u}, \hat{w}) \leq L(u, \hat{w}).$$

Theorem 6.33 $(\bar{u}, \bar{w}) \in X \times Y^*$ is a solution to the primal-dual problem if and only if it is a saddle-point of the Lagrange-functional of F and G.

Proof:
Let $(\bar{u}, \bar{w}) \in X \times Y^*$ solve the primal-dual problem. Then $\bar{u} \in \operatorname{dom} F$ and $\bar{w} \in \operatorname{dom} G^*$ because otherwise the minimum in the primal problem and respectively the maximum in the dual problem would not be attained. It further holds

$$\begin{aligned}
L(\bar{u}, \bar{w}) &\leq \sup_{w \in Y^*} L(\bar{u}, w) \\
&= \sup_{w \in Y^*} \big(\langle w, A\bar{u} \rangle + F(\bar{u}) - G^*(w) \big) \\
&\leq \sup_{w \in Y^*} \big(G(A\bar{u}) + G^*(w) + F(\bar{u}) - G^*(w) \big) && \text{(Fenchel-inequality (6.12))} \\
&= F(\bar{u}) + G(A\bar{u}) \\
&= -F^*(-A^*\bar{w}) - G^*(\bar{w}) && \text{(Assumption)} \\
&= -\big(F^*(-A^*\bar{w}) + G^*(\bar{w}) \big) \\
&= -\sup_{u \in X} \big(\langle -A^*\bar{w}, u \rangle - F(u) + G^*(\bar{w}) \big) && \text{(Definition 6.24 Fenchel-conjugate)} \\
&= \inf_{u \in X} \big(\langle \bar{w}, Au \rangle + F(u) - G^*(\bar{w}) \big) \\
&= \inf_{u \in X} L(u, \bar{w}) \leq L(\bar{u}, \bar{w})
\end{aligned}$$

And, comparing the first and last line, in summary

$$\sup_{w \in Y^*} L(\bar{u}, w) = L(\bar{u}, \bar{w}) = \inf_{u \in X} L(u, \bar{w})$$

and $(\bar{u}, \bar{w})$ is a saddle-point of the Lagrange-functional.

Now let $(\bar{u}, \bar{w})$ be a saddle-point of the Lagrange-functional. Then

$$\begin{aligned}
L(\bar{u}, \bar{w}) &= \sup_{w \in Y^*} L(\bar{u}, w) \\
&= F(\bar{u}) + \sup_{w \in Y^*} \big(\langle w, A\bar{u} \rangle - G^*(w) \big) \\
&= F(\bar{u}) + G^{**}(A\bar{u}) = F(\bar{u}) + G(A\bar{u}) \\
&= \inf_{u \in X} L(u, \bar{w}) && \text{(because } (\bar{u}, \bar{w}) \text{ is a saddle-point)} \\
&= -F^*(-A^*\bar{w}) - G^*(\bar{w}) && \square
\end{aligned}$$

By theorem 6.33, instead of finding a minimizer of the primal-problem we may instead compute a saddle-point of the Lagrange-functional. At first glance, this might seem

unprofitable: instead of solving the primal problem with respect to u, we define a second problem with respect to w and couple both problems via the Lagrange-functional.

This concept turns out to be very useful, as the saddle-point may be derived by two fixed-point iterations. These iterations are coupled: when computing u^{n+1}, besides u^n also w^n is used and respectively for computing w^{n+1} also u^n is used.

6.5. Primal-Dual Algorithms

From now on

- let X, Y be finite Hilbert spaces over $\mathbb{R}$,
- let $F : X \to \mathbb{R}_\infty$ and $G : Y \to \mathbb{R}_\infty$ be proper, convex and lower semi-continuous,
- assume there exists a solution $\bar{u} \in X$ to the primal problem (6.11),
- assume given $\sigma, \tau > 0$, the proximal operators $\text{prox}_{\tau,F}$ and prox_{σ,G^*} are computable, that is the optimization problem in definition 6.20 may be solved analytically,
- for the linear operator $A : X \to Y$ let $K := \|A\|$ and assume the adjoint operator A^* is computable,
- assume there is a $u_0 \in \text{dom}\, F$ such that $Au_0 \in \text{dom}\, G$ and G is continuous in Au_0.

The Fenchel-conjugate for a proper $F : X \to \mathbb{R}_\infty$ then becomes

$$\begin{aligned} F^* &: X \to \mathbb{R}_\infty \\ F^*(w) &= \sup_{u \in X} (w, u) - F(u). \end{aligned}$$

To compute a saddle-point of the Lagrange functional $L : \text{dom}\, F \times \text{dom}\, G^* \to \mathbb{R}$, we have to minimize L with respect to $u \in X$ and maximize L with respect to $w \in Y$. For the sake of clarity let

$$\begin{aligned} L_{\hat{w}} &: \text{dom}\, F \to \mathbb{R} & L_{\hat{u}} &: \text{dom}\, G^* \to \mathbb{R} \\ L_{\hat{w}}(u) &:= L(u, \hat{w}) & L_{\hat{u}}(w) &:= L(\hat{u}, w) \end{aligned}$$

If F and G^* are convex, then also $L_{\hat{w}}$ and $-L_{\hat{u}}$ are convex. Let $\sigma, \tau > 0$.
Minimization of $L_{\hat{w}}$:

$$\begin{aligned} \bar{u} = \arg\min_{u \in X} L_{\hat{w}}(u) &\Leftrightarrow 0 \in \partial L_{\hat{w}}(\bar{u}) \\ &\Leftrightarrow \bar{u} \in \bar{u} + \tau \partial L_{\hat{w}}(\bar{u}) \\ &\Leftrightarrow \bar{u} \in (\text{id} + \tau \partial L_{\hat{w}})(\bar{u}) \\ &\Leftrightarrow \bar{u} \in (\text{id} + \tau \partial L_{\hat{w}})^{-1}(\bar{u}) \\ &\Leftrightarrow \bar{u} = \text{prox}_{\tau, L_{\hat{w}}}(\bar{u}) \qquad \text{(Lemma 6.22)} \end{aligned}$$

$\text{prox}_{\tau,L_{\hat{w}}}$ may be expressed in terms of $\text{prox}_{\tau,F}$ by using lemma 6.23:

$$\begin{aligned} & \text{prox}_{\tau,L_{\hat{w}}}(\bar{u}) & \\ = \; & \text{prox}_{\tau,H_1}(\bar{u}) & \text{with } H_1 = \left(\hat{w}, A(\cdot)\right) + F(\cdot) - G^*(\hat{w}) \\ = \; & \text{prox}_{\tau,H_2}(\bar{u}) & \text{with } H_2 = (A^*\hat{w}, \cdot) + F(\cdot) \\ = \; & \text{prox}_{\tau,F}(u - \tau A^*\hat{w}) & \end{aligned}$$

Maximizing $L_{\hat{u}}$ is equivalent to minimizing $-L_{\hat{u}}$ and analog transformations yield

$$\bar{w} = \arg\min_{w \in Y} -L_{\hat{u}}(w) \quad \Leftrightarrow \quad \bar{w} = \text{prox}_{\sigma,-L_{\hat{u}}}(\bar{w})$$

and by application of lemma 6.23

$$\text{prox}_{\sigma,-L_{\hat{u}}}(w) = \text{prox}_{\sigma,G^*}(w + \sigma A\hat{u}).$$

In summary, we derive the coupled fixed-point equations

$$\begin{cases} \bar{u} = \text{prox}_{\tau,F}(\bar{u} - \tau A^*\hat{w}) \\ \bar{w} = \text{prox}_{\sigma,G^*}(\bar{w} + \sigma A\hat{u}) \end{cases} \tag{6.19}$$

In [5], Antonin Chambolle and Thomas Pock present and analyze algorithms 3 to 5. These algorithms to solve the fixed-point equations (6.19) are called *primal-dual algorithms*. Algorithm 3 may be used for problems fulfilling the requirements listed at the beginning of this section. If in addition F is strongly convex with parameter $\gamma > 0$, the accelerated algorithm 4 may be used. If F is strongly convex with parameter $\gamma > 0$ and G^* is strongly convex with parameter $\delta > 0$, the even faster algorithm 5 may be used.

We will stop the iteration when the stopping criterion $\mathcal{G}\left(u^{k+1}, u^k, w^{k+1}, w^k\right) < \varepsilon$ is fulfilled. Possible criteria are

- the difference between the values of the primal and dual problem, the so called *duality gap*:

$$\mathcal{G}\left(u^{k+1}, u^k, w^{k+1}, w^k\right) = F(u^{k+1}) + G(Au^{k+1}) + F^*(-A^*w^{k+1}) + G^*(w^{k+1}),$$

- the difference of the primal iteratives:

$$\mathcal{G}\left(u^{k+1}, u^k, w^{k+1}, w^k\right) = \|u^{k+1} - u^k\|,$$

- a combination of the differences of the primal and dual iteratives:

$$\mathcal{G}\left(u^{k+1}, u^k, w^{k+1}, w^k\right) = \eta_1\|u^{k+1} - u^k\| + \eta_2\|w^{k+1} - w^k\|,$$

  with $\eta_1, \eta_2 > 0$.

A proof for the convergence of these algorithms and further analysis can be found in [5].

Algorithm 3

Initialize: $\sigma, \tau > 0, \Theta \in [0,1]$, $\left(u^0, w^0\right) \in X \times Y$, $\hat{u}^0 = u^0$, $\varepsilon > 0$
for $k = 0, 1, 2 \ldots$ **do**
  $w^{k+1} \leftarrow \text{prox}_{\sigma, G^*}\left(w^k + \sigma A \hat{u}^k\right)$
  $u^{k+1} \leftarrow \text{prox}_{\tau, F}\left(u^k - \tau A^* w^{k+1}\right)$
  $\hat{u}^{k+1} \leftarrow u^{k+1} + \Theta\left(u^{k+1} - u^k\right)$
  if $\mathcal{G}\left(u^{k+1}, u^k, w^{k+1}, w^k\right) < \varepsilon$ **then**
    break
  end if
end for

Algorithm 4

Initialize: $\sigma_0, \tau_0 > 0$, $\sigma_0 \tau_0 K^2 \leq 1$, $\left(u^0, w^0\right) \in X \times Y$, $\hat{u}^0 = u^0$, $\varepsilon > 0$
for $k = 0, 1, 2 \ldots$ **do**
  $w^{k+1} \leftarrow \text{prox}_{\sigma_n, G^*}\left(w^k + \sigma_n A \hat{u}^k\right)$
  $u^{k+1} \leftarrow \text{prox}_{\tau_n, F}\left(u^k - \tau_n A^* w^{k+1}\right)$
  $\Theta_n \leftarrow 1/\sqrt{1 + 2\gamma\tau_n}$
  $\tau_{n+1} \leftarrow \Theta_n \tau_n$
  $\sigma_{n+1} \leftarrow \sigma_n / \Theta_n$
  $\hat{u}^{k+1} \leftarrow u^{k+1} + \Theta_n\left(u^{k+1} - u^k\right)$
  if $\mathcal{G}\left(u^{k+1}, u^k, w^{k+1}, w^k\right) < \varepsilon$ **then**
    break
  end if
end for

Algorithm 5

Initialize: Choose $\mu \leq 2\sqrt{\gamma\delta}/K$, let $\tau = \mu(2\gamma)$, $\sigma = \mu/(2\delta)$, $\Theta \in \left[\left(1+\mu\right)^{-1}, 1\right]$, $\left(u^0, w^0\right) \in X \times Y$, $\hat{u} = u^0$, $\varepsilon > 0$
for $k = 0, 1, 2 \ldots$ **do**
  $w^{k+1} \leftarrow \text{prox}_{\sigma, G^*}\left(w^k + \sigma A \hat{u}^k\right)$
  $u^{k+1} \leftarrow \text{prox}_{\tau, F}\left(u^k - \tau A^* w^{k+1}\right)$
  $\hat{u}^{k+1} \leftarrow u^{k+1} + \Theta\left(u^{k+1} - u^k\right)$
  if $\mathcal{G}\left(u^{k+1}, u^k, w^{k+1}, w^k\right) < \varepsilon$ **then**
    break
  end if
end for

Chapter 7.

The Fast Mumford-Shah Approach

The idea of the *Fast Mumford-Shah* approach (presented in [18]) is to describe the edge set K in terms of the gradient of u: When the norm of the gradient is small at $x \in \Omega$, then around x there is probably a small brightness or color variation inside a region. If on the other hand the norm of the gradient at x is large, there is a significant brightness and/or color change around x, it probably lies on the border between different regions and consequently $x \in K$.

To define a functional using this pointwise properties is easier in the discrete than in the continuous setting, and since we would have to the discretize the functional for the numerical implementation anyways, we follow the original [18] and use a discrete setting from the beginning.

7.1. Discrete Setting

The most general case we will consider are images on a three-dimensional domain into a c-dimensional color space, and we might represent these images as elements of $X = \mathbb{R}^{d_1 \times d_2 \times d_3 \times c}$ with $d_1, d_2 \geq 2$, $d_3, c \geq 1$. The image domain is three-dimensional, if $d_3 > 1$ and two-dimensional, if $d_3 = 1$. To be able to define partial derivatives for all dimensions, we also define the space $Y = \mathbb{R}^{d_1 \times d_2 \times d_3 \times c \times n_d}$, with $n_d = 2$ for two-dimensional and $n_d = 3$ for three-dimensional images.

For the sake of clarity, let

$$\mathcal{S} \coloneqq \left\{(i,j,k) : i \in \{1,\dots,d_1\}, j \in \{1,\dots,d_2\}, k \in \{1,\dots,d_3\}\right\}$$

and for $s \in \mathcal{S}$ let

$$\begin{aligned} X_s &\coloneqq \left\{u_{i,j,k,l} \in X : s = (i,j,k), l \in \{1,\dots,c\}\right\} \\ Y_s &\coloneqq \left\{w_{i,j,k,l,m} \in Y : s = (i,j,k), l \in \{1,\dots,c\}, m \in \{1,\dots,n_d\}\right\}. \end{aligned}$$

For the rest of this chapter, all norms on X, Y, X_s and Y_s are the respective Frobenius norm, and we identify $X \cong R^{d_1 \cdot d_2 \cdot d_3 \cdot c}$, $Y \cong R^{d_1 \cdot d_2 \cdot d_3 \cdot c \cdot n_d}$, $X_s \cong \mathbb{R}^c$ and $Y_s \cong \mathbb{R}^{c \cdot n_d}$.

Definition 7.1 We define the following finite difference operators:

For $u \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c}$ and for $t = 1, 2, 3$ we define the forward finite difference operators ∂_t with $\partial_t u \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c}$ as

$$(\partial_1 u)_{i,j,k,l} := \begin{cases} u_{i+1,j,k,l} - u_{i,j,k,l} & i < d_1 \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

$$(\partial_2 u)_{i,j,k,l} := \begin{cases} u_{i,j+1,k,l} - u_{i,j,k,l} & j < d_2 \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

$$(\partial_3 u)_{i,j,k,l} := \begin{cases} u_{i,j,k+1,l} - u_{i,j,k,l} & k < d_3 \\ 0 & \text{otherwise.} \end{cases} \tag{7.3}$$

For $w \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c \times n_d}$ and for $t = 1, 2, 3$ we define the backward finite difference operators ∂_t^- with $\partial_t^- w \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c}$ as

$$\left(\partial_1^- w\right)_{i,j,k,l} := \begin{cases} w_{1,j,k,l,1} & i = 1 \\ w_{i,j,k,l,1} - w_{i-1,j,k,l,1} & 1 < i < d_1 \\ -w_{d_1-1,j,k,l,1} & i = d_1 \end{cases} \tag{7.4}$$

$$\left(\partial_2^- w\right)_{i,j,k,l} := \begin{cases} w_{i,1,k,l,2} & j = 1 \\ w_{i,j,k,l,2} - w_{i,j-1,k,l,2} & 1 < j < d_2 \\ -w_{i,d_2-1,k,l,2} & j = d_2 \end{cases} \tag{7.5}$$

$$\left(\partial_3^- w\right)_{i,j,k,l} := \begin{cases} w_{i,j,1,l,3} & k = 1 \\ w_{i,j,k,l,3} - w_{i,j,k-1,l,3} & 1 < k < d_3 \\ -w_{i,j,d_3-1,l,3} & k = d_3. \end{cases} \tag{7.6}$$

Definition 7.2 (Discrete Gradient) For $u \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c}$, the discrete gradient is defined as

$$\nabla u := \begin{cases} (\partial_t u)_{1 \le t \le 2} \in \mathbb{R}^{d_1 \times d_2 \times 1 \times c \times 2} & d_3 = 1 \\ (\partial_t u)_{1 \le t \le 3} \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c \times 3} & d_3 > 1. \end{cases}$$

Definition 7.3 (Discrete Divergence) For $w \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c \times n_d}$, the discrete divergence is defined as

$$\operatorname{div} w := \sum_{t=1}^{n_d} \partial_t^- w \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c}.$$

Lemma 7.4 (Generalization of lemma 6.142 in [4]) The linear operator

$$\nabla : \mathbb{R}^{d_1 \times d_2 \times d_3 \times c} \longrightarrow \mathbb{R}^{d_1 \times d_2 \times d_3 \times c \times n_d}$$

has the following properties:

$$\nabla^* = -\mathrm{div} \qquad \text{and} \qquad \|\nabla\|^2 \leq 4n_d.$$

Proof:
We start with the adjoint operator. Let $u \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c}$, $w \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c \times n_d}$ and let $w^t = \left(w_{ijklm}\right) \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c}$ where $m = t$ is fixed. We first observe

$$\begin{aligned}
&\left(\partial_1 u, w^1\right)_{\mathbb{R}^{d_1 \times d_2 \times d_3 \times c}} \\
&= \sum_{i=1}^{d_1-1} \sum_{j=1}^{d_2} \sum_{k=1}^{d_3} \sum_{l=1}^{c} \left(u_{i+1,j,k,l} - u_{i,j,k,l}\right) w_{i,j,k,l,1} \\
&= \sum_{j=1}^{d_2} \sum_{k=1}^{d_3} \sum_{l=1}^{c} \left(\sum_{i=1}^{d_1-1} \left(u_{i+1,j,k,l} w_{i,j,k,l,1}\right) - \sum_{i=1}^{d_1-1} \left(u_{i,j,k,l} w_{i,j,k,l,1}\right) \right) \\
&= \sum_{j=1}^{d_2} \sum_{k=1}^{d_3} \sum_{l=1}^{c} \left(\sum_{i=2}^{d_1} \left(u_{i,j,k,l} w_{i-1,j,k,l,1}\right) - \sum_{i=1}^{d_1-1} \left(u_{i,j,k,l} w_{i,j,k,l,1}\right) \right) \\
&= \sum_{j=1}^{d_2} \sum_{k=1}^{d_3} \sum_{l=1}^{c} \left(-w_{1,j,k,l,1} u_{1,j,k,l} + \sum_{i=2}^{d_1-1} \left(\left(w_{i-1,j,k,l,1} - w_{i,j,k,l,1}\right) u_{i,j,k,l} \right) + w_{d_1-1,j,k,l,1} u_{d_1,j,k,l} \right) \\
&= \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \sum_{k=1}^{d_3} \sum_{l=1}^{c} -\left(\partial_1^- w\right)_{i,j,k,l} u_{i,j,k,l}.
\end{aligned}$$

Analog transformations on j and k yield the following expressions for $\left(\partial_2 u, w^2\right)$ and $\left(\partial_3 u, w^3\right)$:

$$\left(\partial_2 u, w^2\right) = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \sum_{k=1}^{d_3} \sum_{l=1}^{c} -\left(\partial_2^- w\right)_{i,j,k,l} u_{i,j,k,l}$$

$$\left(\partial_3 u, w^3\right) = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \sum_{k=1}^{d_3} \sum_{l=1}^{c} -\left(\partial_3^- w\right)_{i,j,k,l} u_{i,j,k,l}$$

Now we may conclude

$$\begin{aligned}(\nabla u, w)_{\mathbb{R}^{d_1\times d_2\times d_3\times c\times n_d}} &= \sum_{t=1}^{n_d} \left(\partial_t u, w^t\right)_{\mathbb{R}^{d_1\times d_2\times d_3\times c}} \\ &= \sum_{t=1}^{n_d}\sum_{i=1}^{d_1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3}\sum_{l=1}^{c} -\left(\partial_t^- w\right)_{i,j,k,l} u_{i,j,k,l} \\ &= (u, -\operatorname{div} w)_{\mathbb{R}^{d_1\times d_2\times d_3\times c}}\end{aligned}$$

and consequently $\nabla^* = -\operatorname{div}$.

Next, we prove that $\|\nabla\|^2 \leq 4n_d$ by showing that $\|\nabla u\|^2 \leq 4n_d$ if $\|u\| = 1$. For $u \in \mathbb{R}^{d_1\times d_2\times d_3\times c}$ with $\|u\| = 1$ it holds

$$-\sum_{i=1}^{d_1-1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3}\sum_{l=1}^{c} u_{i+1,j,k,l}u_{i,j,k,l} < 1, \tag{7.7}$$

$$-\sum_{i=1}^{d_1}\sum_{j=1}^{d_2-1}\sum_{k=1}^{d_3}\sum_{l=1}^{c} u_{,j+1,k,l}u_{i,j,k,l} < 1, \tag{7.8}$$

$$-\sum_{i=1}^{d_1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3-1}\sum_{l=1}^{c} u_{,j,k+1,l}u_{i,j,k,l} < 1. \tag{7.9}$$

Assume the first of these inequalities does not hold and let $w_{i,j,k,l} = -u_{i+1,j,k,l}$ for $i < d_1$, $j \leq d_2$, $k \leq d_3$, $l \leq c$ and $w_{d_1,j,k,l} = 0$. Then, by assumption,

$$\sum_{i=1}^{d_1-1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3}\sum_{l=1}^{c} w_{i,j,k,l}u_{i,j,k,l} = (w, u) \geq 1.$$

On the other hand, by construction $\|w\| \leq \|u\| = 1$ and with the Cauchy-Schwarz inequality

$$(w, u) \leq \|w\|\|u\| \leq 1, \tag{7.10}$$

and in consequence $(w, u) = 1$. Now (7.10) holds with equality and therefore either $w = u$ or $w = -u$, where the latter can be rejected because of $(w, u) = 1$. It further holds by recursion along $i < d_1$ and for $j \leq d_2$, $k \leq d_3$ and $l \leq c$

$$0 = w_{d_1,j,k,l} = u_{d_1,j,k,l} = -u_{d_1-1,j,k,l} = (-1)^i u_{i,j,k,l}.$$

The above means $u = 0$, a contradiction to $\|u\| = 1$ and inequality (7.7) holds. The inequalities (7.8) and (7.9) can be proved under analogous argumentation.

Now, for $\|u\| = 1$, we consider

$$\begin{aligned}
\|\nabla u\|^2 = & \sum_{i=1}^{d_1-1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3}\sum_{l=1}^{c}\left(u_{i+1,j,k,l}-u_{i,j,k,l}\right)^2 \\
& + \sum_{i=1}^{d_1}\sum_{j=1}^{d_2-1}\sum_{k=1}^{d_3}\sum_{l=1}^{c}\left(u_{i,j+1,k,l}-u_{i,j,k,l}\right)^2 \\
& + \underbrace{\sum_{i=1}^{d_1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3-1}\sum_{l=1}^{c}\left(u_{i,j,k+1,l}-u_{i,j,k,l}\right)^2}_{=0 \text{ if } d_3=1 \Leftrightarrow n_d=2} \\
= & \sum_{i=1}^{d_1-1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3}\sum_{l=1}^{c} u_{i+1,j,k,l}^2 - 2u_{i+1,j,k,l}u_{i,j,k,l} + u_{i,j,k,l}^2 \\
& + \sum_{i=1}^{d_1}\sum_{j=1}^{d_2-1}\sum_{k=1}^{d_3}\sum_{l=1}^{c} u_{i,j+1,k,l}^2 - 2u_{i,j+1,k,l}u_{i,j,k,l} + u_{i,j,k,l}^2 \\
& + \underbrace{\sum_{i=1}^{d_1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3-1}\sum_{l=1}^{c} u_{i,j,k+1,l}^2 - 2u_{i,j,k+1,l}u_{i,j,k,l} + u_{i,j,k,l}^2}_{=0 \text{ if } n_d=2}.
\end{aligned}$$

For now let $n_d = 2$. We split the sums and group the quadratic and non-quadratic terms, where for the quadratic terms we add the summands for $i = d_1$ and $j = d_2$ respectively:

$$\begin{aligned}
\|\nabla u\|^2 \leq & \underbrace{4\left(\sum_{i=1}^{d_1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3}\sum_{l_1}^{c} u_{i,j,k,l}^2\right)}_{=1} \\
& + \underbrace{2\left(-\sum_{i=1}^{d_1-1}\sum_{j=1}^{d_2}\sum_{k=1}^{d_3}\sum_{l=1}^{c} u_{i+1,j,k,l}u_{i,j,k,l}\right)}_{<1 \text{ by inequality (7.7)}} \\
& + \underbrace{2\left(-\sum_{i=1}^{d_1-1}\sum_{j=1}^{d_2-1}\sum_{k=1}^{d_3}\sum_{l=1}^{c} u_{i,j+1,k,l}u_{i,j,k,l}\right)}_{<1 \text{ by inequality (7.8)}} \\
< & \; 4+2+2 = 8 = 4n_d.
\end{aligned}$$

For $n_d = 3$ with the additional terms from above, analogous argumentation results in

$$\|\nabla u\|^2 \leq \ 6 + 2 + 2 + 2 = 12 = 4n_d. \qquad \square$$

7.2. The Fast Mumford-Shah Functional

Similar to the original Mumford-Shah functional, we define the data-term $F : X \to \mathbb{R}$ as

$$F(u) = \|u - f\|^2.$$

Instead of using two regularizers, we use the combination $G : Y \to \mathbb{R}$ with parameters $\alpha, \lambda > 0$,

$$G(w) = \sum_{s \in \mathcal{S}} G_s(w_s) \qquad\qquad G_s(w_s) = \min\left(\alpha \|w_s\|^2, \lambda\right).$$

With $A = \nabla : X \to Y$, we define the *Fast Mumford-Shah* functional $\mathcal{F}_{\text{FMS}} : X \to \mathbb{R}$ as

$$\mathcal{F}_{\text{FMS}}(u) \coloneqq \|u - f\|^2 + G(\nabla u). \tag{7.11}$$

This means that if $\|(\nabla u)_s\|$ at a point $s \in \mathcal{S}$ is smaller than $\sqrt{\lambda/\alpha}$, we assume that at s there is a small brightness or color variation within a region, and we penalize this variation in equation (7.11) with the factor $\alpha > 0$.

If on the other hand $\|(\nabla u)_s\| \geq \sqrt{\lambda/\alpha}$, we assume that at s is a border between different regions which we penalize with $\lambda > 0$. We especially obtain the set K as

$$K \coloneqq \left\{ s \in \mathcal{S} : \|(\nabla u)_s\| \geq \sqrt{\frac{\lambda}{\alpha}} \right\}.$$

Increasing λ will result in a higher penalty for the border between regions and consequently in the optimal solution this border will get shorter and the number of regions decreases.

Increasing α will result in a higher penalty for variation inside a region and consequently the optimal solution will get smoother inside every region. Note that K is defined in terms of α, so changing α will also have an effect on K.

In the limit case $\alpha \to \infty$ we get

$$\lim_{\alpha \to \infty} G_s(w_s) = \lim_{\alpha \to \infty} \min\left(\alpha \|w_s\|^2, \lambda\right) = \begin{cases} \lambda & \text{if } \|w_s\| \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

This means that every point at which the norm of the gradient is positive is a point in K, the slightest brightness or color variation is considered to define an edge. Or, in other

words, the resulting image is constant on every region, so for $\alpha \to \infty$ we obtain a piecewise constant segmentation and for $0 < \alpha < \infty$ a piecewise smooth segmentation.

One apparent characteristic of this approach is the lacking possibility to fix the number of regions in the segmentation. While α, λ have an influence on this number (the higher λ/α the lower the number of regions), it is almost impossible to select them in such a way that the output will contain exactly k regions. This drawback can be evened out by the combination of Fast Mumford-Shah with other segmentation methods, see chapter 8.

7.3. Primal-dual Optimization Algorithm

Requirements

To minimize the Fast Mumford-Shah functional (7.11), we use the framework described in chapter 6.

According to lemmata 6.6 and 6.8, the data term

$$F(u) = \|u - f\|^2$$

is strongly convex with parameter $\gamma = 2$, so we are able to use the accelerated algorithm 4.

The combined regularizer G on the other hand is not strongly convex. In fact, it is not even convex, see figure 7.1. The theory in chapter 6 is only *guaranteed* to work

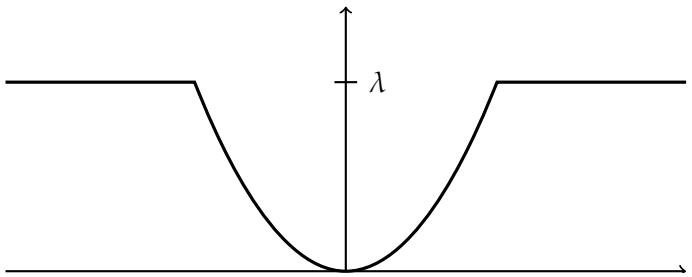


Figure 7.1.: $G(u) = \min\left(\alpha|u|^2, \lambda\right)$ is not convex

in the case where F, G both are convex, but we did not state that it would fail for a non-convex G. It turns out that it works well and changes to α and λ result in the expected effects (see section 7.5). Note that in algorithms 3 to 5, G itself does not appear, only its Fenchel-conjugate G^*, which according to lemma 6.25 *is* convex.

We check the remaining requirements of section 6.5:

- X, Y are finite Hilbert spaces over $\mathbb{R}$.
- F, G are proper, $\operatorname{dom} F = X$, $\operatorname{dom} G = Y$ and continuous on X and Y respectively.
- The *direct method* (remark 6.10) shows that there exists a solution:

1. $\mathcal{F}_{\text{FMS}}$ is non-negative and therefore bounded from below, so there exists a minimizing sequence.
2. For $\|u_n\| \to \infty$ it holds $F(u_n) \to \infty$ and $G(u_n) \geq 0$, so consequently $\mathcal{F}_{\text{FMS}} \to \infty$ and $\mathcal{F}_{\text{FMS}}$ is coercive. This means that the minimizing sequence from step 1 has to be bounded and by the Bolzano-Weierstraß theorem, there is a converging subsequence.
3. $\mathcal{F}_{\text{FMS}}$ is continuous and especially lower semi-continuous.

- We will see below that we are able to compute the proximal operators $\text{prox}_{\tau,F}$ and prox_{σ,G^*}.
- For $A = \nabla$, by lemma 7.4 we have an estimate $\|\nabla\| \leq 4n_d$.

The proximal operators

The proximal operator of $F : X = \mathbb{R}^{d_1 \times d_2 \times d_3 \times c} \to \mathbb{R}$ with $F(u) = \|u - f\|^2$ can be derived as

$$\begin{aligned}\text{prox}_{\tau,F}(u) &= \underset{v \in X}{\arg\min} \left(\frac{\|v-u\|^2}{2\tau} + F(v) \right) \\ &= \underset{v \in X}{\arg\min} \left(\frac{\|v-u\|^2}{2\tau} + \|v-f\|^2 \right).\end{aligned}$$

This expression is strictly convex with respect to v and setting the derivative to zero yields

$$0 = \frac{(\bar{v}-u)}{\tau} + 2(\bar{v}-f) \Rightarrow \bar{v} = \frac{u + 2\tau f}{1 + 2\tau}$$

and in conclusion

$$\text{prox}_{\tau,F}(u) = \frac{u + 2\tau f}{1 + 2\tau}. \tag{7.12}$$

Deriving the proximal operator prox_{σ,G^*} is more involved. We will determine the proximal operator of $G : Y = \mathbb{R}^{d_1 \times d_2 \times d_3 \times c \times n_d} \to \mathbb{R}$ with

$$G(w) = \sum_{s \in \mathcal{S}} G_s(w_s), \qquad G_s(w_s) = \min\left(\alpha \|p_s\|^2, \lambda\right),$$

and then use the Moreau decomposition to get the proximal operator of G^*. We at first

observe that

$$\begin{aligned}
\operatorname{prox}_{\sigma,G}(w) &= \arg\min_{y\in Y} \frac{\|y-w\|^2}{2\sigma} + G(y) \\
&= \arg\min_{y\in Y} \frac{\|y-w\|^2}{2\sigma} + \sum_{s\in\mathcal{S}} \min\left(\alpha\|y_s\|^2, \lambda\right) \\
&= \arg\min_{y\in Y} \sum_{s\in\mathcal{S}} \left(\frac{\|y_s-w_s\|^2}{2\sigma} + \min\left(\alpha\|y_s\|^2, \lambda\right)\right) \\
&= \arg\min_{y\in Y} \sum_{s\in\mathcal{S}} \operatorname{prox}_{\sigma,G_s}(w_s),
\end{aligned}$$

which enables us to derive y_s for every $s \in \mathcal{S}$ and then reconstruct y by combining the y_s. To avoid indices, let $q := w_s$.

$$\operatorname{prox}_{\sigma,G_s}(q) = \arg\min_{y_s\in\mathbb{R}^{c\times d}} \left(\frac{\|y_s-q\|^2}{2\sigma} + \min\left(\alpha\|y_s\|^2, \lambda\right)\right)$$

This minimization problem can be transformed into an one-dimensional problem, see figure 7.2.

$$\begin{aligned}
&\min_{y_s\in\mathbb{R}^{c\times d}} \left(\frac{\|y_s-q\|^2}{2\sigma} + \min\left(\alpha\|y_s\|^2, \lambda\right)\right) \qquad (7.13)\\
&= \min_{\xi\in\mathbb{R}} \left(\frac{\left\|\xi\frac{q}{\|q\|}-q\right\|^2}{2\sigma} + \min\left(\alpha\left\|\xi\frac{q}{\|q\|}\right\|^2, \lambda\right)\right) \\
&= \min_{\xi\in\mathbb{R}} \left(\frac{\left(\frac{\xi}{\|q\|}-1\right)^2\|q\|^2}{2\sigma} + \min\left(\alpha\xi^2, \lambda\right)\right) \\
&= \min_{\xi\in\mathbb{R}} \left(\frac{1}{2\sigma}\xi^2 - \frac{\|q\|}{\sigma}\xi + \frac{\|q\|^2}{2\sigma} + \min\left(\alpha\xi^2, \lambda\right)\right) \\
&= \min_{\xi\in\mathbb{R}} \left(f_1(\xi), f_2(\xi)\right),
\end{aligned}$$

with

$$f_1(\xi) = \left(\frac{1}{2\sigma} + \alpha\right)\xi^2 - \frac{\|q\|}{\sigma}\xi + \frac{\|q\|^2}{2\sigma} \qquad (7.14)$$

$$f_2(\xi) = \frac{1}{2\sigma}\xi^2 - \frac{\|q\|}{\sigma}\xi + \frac{\|q\|^2}{2\sigma} + \lambda. \qquad (7.15)$$

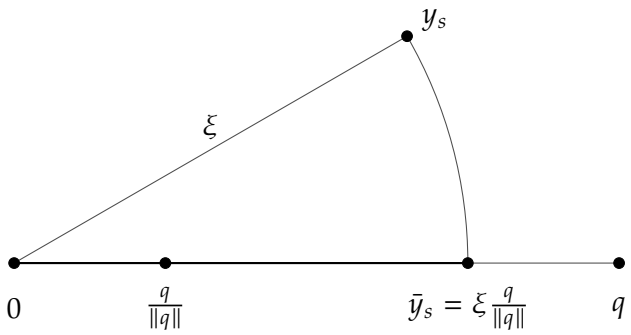


Figure 7.2.: Given any $y_s \in \mathbb{R}^{c\times d}$ with norm $\delta = \|y_s\|$, all points $\hat{y}_s$ on the sphere with radius ξ will result in the same value of $\min\left(\alpha\|\hat{y}_s\|, \lambda\right)$. But the point $\bar{y}_s$ on that sphere that also lies on the line through 0 and q will minimize the first summand in equation (7.13). It is therefore sufficient to search the optimal y_s along the line through 0 and q.

Both f_1, f_2 are quadratic and strictly convex in ξ and the respective minimizers ξ_1, ξ_2 may be computed as

$$\frac{\partial}{\partial\xi} f_1(\xi) = \left(\frac{1}{\sigma} + 2\alpha\right)\xi - \frac{\|q\|}{\sigma} \stackrel{!}{=} 0 \quad \Rightarrow \quad \xi_1 = \frac{\|q\|}{1+2\sigma\alpha} \tag{7.16}$$

$$\frac{\partial}{\partial\xi} f_2(\xi) = \frac{1}{\sigma}\xi - \frac{\|q\|}{\sigma} \stackrel{!}{=} 0 \quad \Rightarrow \quad \xi_2 = \|q\|. \tag{7.17}$$

Inserting ξ_1, ξ_2 in equations (7.14) and (7.15) yields

$$\begin{aligned}\min_{\xi\in\mathbb{R}} f_1(\xi) &= \left(\frac{1}{2\sigma} + \alpha\right)\frac{\|q\|^2}{(1+2\sigma\alpha)^2} - \frac{\|q\|^2}{\sigma\,(1+2\sigma\alpha)} + \frac{\|q\|^2}{2\sigma} \\ &= \frac{(1+2\sigma\alpha)\,\|q\|^2}{2\sigma(1+2\sigma\alpha)^2} - \frac{2\|q\|^2}{2\sigma(1+2\sigma\alpha)} + \frac{(1+2\sigma\alpha)\|q\|^2}{2\sigma(1+2\sigma\alpha)} \\ &= \frac{\|q\|^2 - 2\|q\|^2 + \|q\|^2 + 2\sigma\alpha\|q\|^2}{2\sigma(1+2\sigma\alpha)} = \frac{\alpha\|q\|^2}{1+2\sigma\alpha} \end{aligned} \tag{7.18}$$

$$\min_{\xi\in\mathbb{R}} f_2(\xi) = \frac{\|q\|^2}{2\sigma} - \frac{\|q\|^2}{\sigma} + \frac{\|q\|^2}{2\sigma} + \lambda = \lambda. \tag{7.19}$$

It remains to compute

$$\min\left(\min_{\xi\in\mathbb{R}} f_1(\xi), \min_{\xi\in\mathbb{R}} f_2(\xi)\right)$$

by comparing equations (7.18) and (7.19). σ, α and λ are fixed, but whether the expression in equation (7.18) or in equation (7.19) is smaller depends on $\|q\| = \|p_s\|$:

$$\frac{\alpha\|q\|^2}{1+2\sigma\alpha} \leq \lambda \quad \Leftrightarrow \quad \|q\| \leq \sqrt{\frac{\lambda}{\alpha}(1+2\sigma\alpha)}. \tag{7.20}$$

Using $y_s = \xi \cdot {}^{q}/_{\|q\|}$, ξ_1, ξ_2 from equations (7.16) and (7.17) and the result from equation (7.20), we conclude

$$\text{prox}_{\sigma,G}(w) = \begin{cases} \frac{1}{1+2\sigma\alpha} w & \text{if } \|w\| \leq \sqrt{\frac{\lambda}{\alpha}(1+2\sigma\alpha)}, \\ w & \text{otherwise.} \end{cases} \tag{7.21}$$

Now we are able to determine $\text{prox}_{\sigma,G_s^*}$. Let $\sigma' = \sigma^{-1}$, $w' = \sigma' w$. Then, by Moreau decomposition (lemma 6.28 and remark 6.29),

$$\begin{aligned}
\text{prox}_{\sigma,G_s^*}(w) &= w - \sigma \text{prox}_{\sigma',G_s}(w') \\
&= w - \sigma \cdot \begin{cases} \frac{1}{1+2\sigma'\alpha} w' & \|w'\| \leq \sqrt{\frac{\lambda}{\alpha}(1+2\sigma'\alpha)}, \\ w' & \text{otherwise,} \end{cases} \\
&= w - \sigma \cdot \begin{cases} \frac{1}{1+2\sigma'\alpha} \sigma' w & \|\sigma' w\| \leq \sqrt{\frac{\lambda}{\alpha}(1+2\sigma'\alpha)}, \\ \sigma' w & \text{otherwise,} \end{cases} \\
&= \begin{cases} \left(1 - \frac{1}{1+2\sigma'\alpha}\right) w & \|w\| \leq \sqrt{\frac{\lambda}{\sigma'^2\alpha}(1+2\sigma'\alpha)}, \\ 0 & \text{otherwise,} \end{cases} \\
&= \begin{cases} \frac{1+2\sigma'\alpha-1}{1+2\sigma'\alpha} w & \|w\| \leq \sqrt{\frac{\lambda}{\sigma'\alpha}(\frac{1}{\sigma'}+2\alpha)}, \\ 0 & \text{otherwise,} \end{cases} \\
&= \begin{cases} \frac{2\alpha}{\sigma+2\alpha} w & \|w\| \leq \sqrt{\frac{\lambda\sigma}{\alpha}(\sigma+2\alpha)}, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{7.22}$$

The Algorithm

Using the proximal operators above, $\|\nabla\| \leq 4n_d$ and that F is strongly convex with parameter $\gamma = 2$, we are now able to reformulate algorithm 4 to minimize the Fast Mumford-Shah functional and obtain algorithm 6. In [18], the authors show that algorithm 6 always terminates, that is

$$\|u^{k+1} - u^k\| \to 0 \quad \text{as} \quad k \to \infty,$$

although with the methods of chapter 6 we can not guarantee that u^k converges to a minimizer. However, practical experiments show that the results depend on α, λ as one would expect.

Algorithm 6 Fast Mumford-Shah Segmentation

Initialize: Input image $f \in X$, parameters $0 < \alpha \leq \infty$ and $0 < \lambda < \infty$, $\sigma_0 = 1/2n_d$ and $\tau_0 = 1/2$, $u^0 = f$, $w^0 = 0$, $\hat{u}^0 = u^0$, $\varepsilon > 0$
for $k = 0, 1, 2 \ldots$ **do**
  $\tilde{w}^{k+1} \leftarrow w^k + \sigma_n \nabla \hat{u}^k$
  for $s \in \mathcal{S}$ **do**
    $$w_s^{k+1} \leftarrow \begin{cases} \frac{2\alpha}{\sigma_n + 2\alpha} \tilde{w}_s^{k+1} & \text{if } \|\tilde{w}_s^{k+1}\| \leq \sqrt{\frac{\lambda}{\alpha}\sigma_n(\sigma_n + 2\alpha)} \\ 0 & \text{otherwise} \end{cases}$$
  end for
  $\tilde{u} = u^k + \tau_n \operatorname{div} w^{n+1}$
  $u^{k+1} = \left(\tilde{u} + 2\tau_n f\right) / (1 + 2\tau_n)$
  $\Theta_n \leftarrow 1/\sqrt{1 + 4\tau_n}$
  $\tau_{n+1} \leftarrow \Theta_n \tau_n$
  $\sigma_{n+1} \leftarrow \sigma_n / \Theta_n$
  $\hat{u}^{k+1} \leftarrow u^{k+1} + \Theta_n \left(u^{k+1} - u^k\right)$
  if $\|u^{k+1} - u^k\| < \varepsilon$ **then**
    break
  end if
end for

7.4. Parallel Implementation using OpenACC

When implementing algorithm 6, one will quickly run into the problem of long runtimes. An implementation in MATLAB is only applicable when the input image is small, and for larger images even an efficient implementation in C could result in a runtime of several days[1].

However, algorithm 6 is efficiently parallelizable: As explicitly stated in the algorithm, for every $s \in \mathcal{S}$ the local component w_s is updated and w is obtained by the combination of the w_s. For every $s \in \mathcal{S}$, the computation of w_s can be done in parallel. Although not explicitly stated in the algorithm, the primal variable u may be updated in the same way.

A relatively new, but already very powerful parallel programming model is OpenACC[2].

[1] RGB image with 7132×3808 pixels, serial execution on a 3,4 GHz processor with a tolerance $\varepsilon = 10^{-4}$.
[2] `https://www.openacc.org`

In this model, sections in the code are marked with the so called *directives*. An OpenACC capable compiler (for example the `pgcc`[3] compiler) will then interpret and use these directives, while a not capable compiler will ignore them. So even with the OpenACC directives, the program still consists of valid C code that requires no special compiler or parallel computing environment.

When using the `pgcc` compiler, the execution target can be chosen flexibly. It is possible to compile for either the central processing unit (short CPU, also called the *host*) or for a graphics processing unit (short GPU, also called the *device*) by using the following flags in the compilation command:

`-ta=host` will compile for serial execution on the host,
`-ta=multicore` will compile for parallel execution on the host,
`-acc -ta=tesla` will compile for parallel execution on a Tesla device.

First, we consider a simple loop that can be parallelized using the `kernels` construct, the simplest OpenACC parallelization directive:

```
1 #pragma acc kernels
2 for(i=0;i<n;i++)
3 {
4         u[i] = i;
5 }
```

This directive (*`#pragma acc kernels`*) instructs the compiler to parallelize the following loop, if that is possible. Compiling this loop with the `-ta=multicore` flag gives the following result:

```
2, Loop is parallelizable
Generating Multicore code
2, #pragma acc loop gang
```

And compilation with the `-acc -ta=tesla` flag gives:

```
2, Loop is parallelizable
Accelerator kernel generated
Generating Tesla code
2, #pragma acc loop gang, vector(128) /* blockIdx.x threadIdx.x */
```

[3] `https://www.pgroup.com`

The compiler automatically recognized that the loop is parallelizable and created multicore code for execution on the host or Tesla code for execution on the device respectively.

Now we consider the following, not parallelizable example:

```
#pragma acc kernels
for(i=1;i<n;i++)
{
    u[i] = 2*u[i-1];
}
```

Compilation for multicore execution results in

```
2, Loop carried dependence of u-> prevents parallelization
   Loop carried backward dependence of u-> prevents vectorization
```

and the compiler recognized that the loop is not parallelizable and consequently generated no parallel code.

The loops arising when implementing algorithm 6 are more complicated. We have to iterate over up to three dimensions and over the color channels. For an image $u \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times c}$, this results in loops of the following form:

```
for(l=0;l<c;l++)
{
        for(k=0;k<d3;k++)
        {
                for(j=0;j<d2;j++)
                {
                        for(i=0;i<d1;i++)
                        {
                                idx = l*d1*d2*d3 + k*d1*d2 + j*d1 + i;
                                u[idx] = /* ... */
                        }
                }
        }
}
```

When the loops are this deeply nested and the array indices are this complex, the compiler is unable to recognize whether the loops are parallelizable or not. In this case, the kernels construct is not applicable. Instead, we have to use the `acc parallel loop independent` construct. With this construct, it lies in the responsibility of the programmer to ensure

that the respective loop is parallelizable. If it is not, the compiler won't recognize and the program might not even crash during execution. Instead, the consequence will most likely be false, time-dependent and not easily reproduceable program state.

The implementation of algorithm 6 can be found in `fastms.c` and consists essentially of the following components:

- An outer loop, often called *convergence loop*, that is executed as long as the termination criterion $\|u^{k+1} - u^k\| < \varepsilon$ is not fulfilled.
- The update of the dual variable. Here, for every dimension, every color channel and every $s \in \mathcal{S}$ the discrete partial derivative has to be computed. This can be done in parallel. Note that we do not store the gradient $\nabla \hat{u}^k$ but combine its computation and the update of w in one step.
- The update of the primal variable, which again can be done in parallel for every dimension, color channel and $s \in \mathcal{S}$ and update of the scalar values τ, σ and Θ.
- The computation of $\|u^{k+1} - u^k\|$, which also can be done in parallel. To reduce computational cost, we only check this condition every 20 iteration steps.

When using OpenACC directive for the steps above, the following things have to be kept in mind:

- OpenACC does not work with pointer arithmetic, so expressions like

```
1 for(i=0;i<n;i++)
2 {
3         *(u++) = i;
4 }
5 u -= n;
```

  have to be replaced by

```
1 for(i=0;i<n;i++)
2 {
3         u[i] = i;
4 }
```

- When compiling for execution on the GPU device, memory management is important. Consider the case of an outer, convergence loop with an inner loop working on some array, just marked with the `#pragma acc parallel loop` construct. The computation time on the device might take longer than the serial execution on the host. The reason is the data transfer between the host and the device. In every

iteration, the whole array is copied from the host to the device, the computation is done, and the updated array is copied back from the device to the host.
To avoid these time consuming transfers, OpenACC provides the `data` directives. With these, the programmer can specify which data should be kept on the host and which on the device. It is typically sufficient to copy the data to the device once, perform all operations on the device and copy the data back after the convergence loop is done.

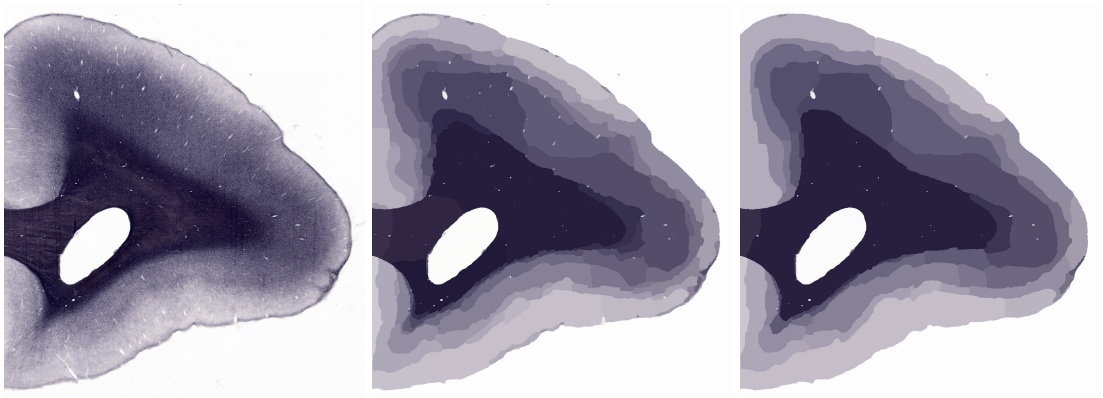
- A *reduction loop* (a loop computing the sum or the maximum of all array entries, for example) has to be marked explicitly with the `reduction` construct. This construct takes the operator and the local variable that stores intermediate state as an argument. The computation of $\|u^{k+1} - u^k\|$, for example, is done as

```
1 #pragma acc parallel loop reduction(+:sum)
2 for(i=0;i<lmnc;i++)
3 {
4         sum += (u[i]-uOld[i])*(u[i]-uOld[i]);
5 }
6 normDiff = sqrt(sum);
```

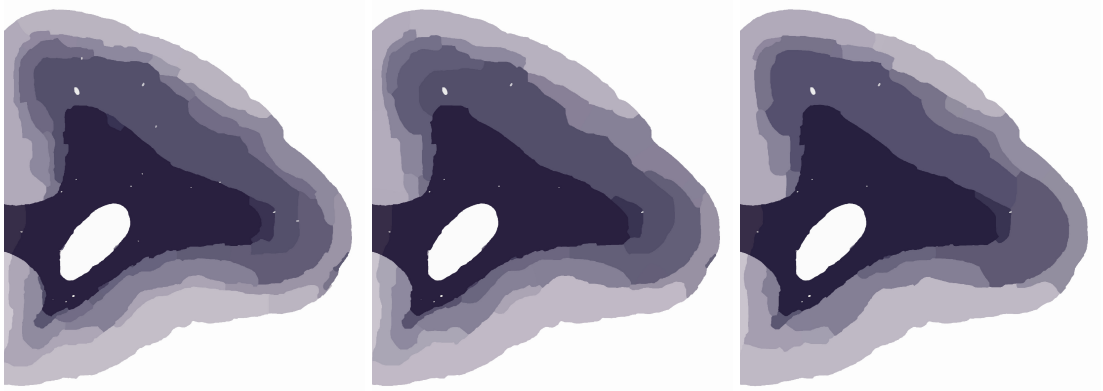
Further information on OpenACC, its possibilities and "best practices" can be found in [1, 12, 11].

7.5. Results

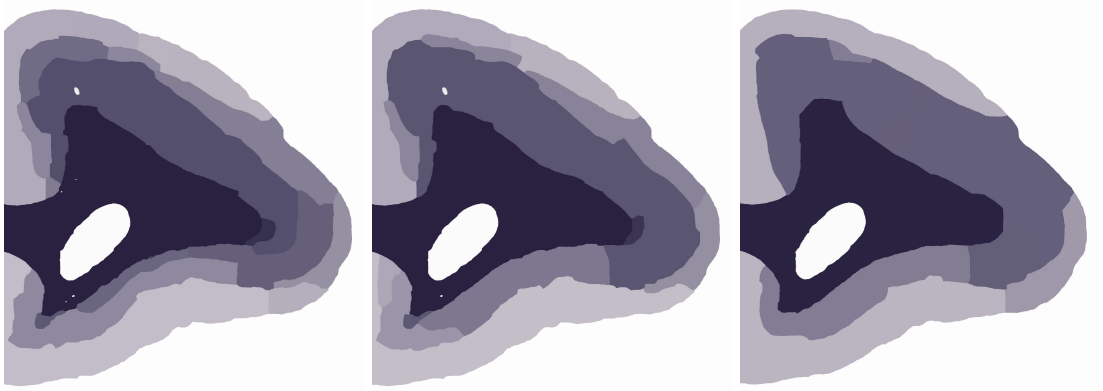
The results of the Fast Mumford-Shah method when applied to the medical data can be seen in figures 7.3 to 7.6, which for the sake of comparability are positioned on the following, opposing pages.



(a) Original (b) $\lambda = 0.4$ (c) $\lambda = 0.8$

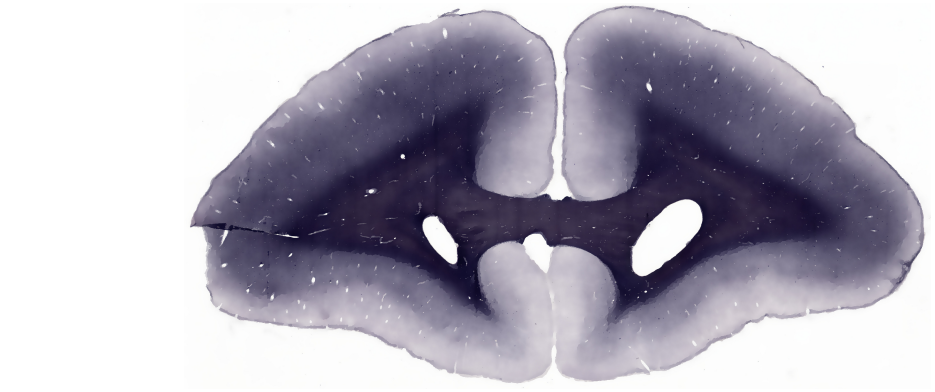


(d) $\lambda = 1.2$ (e) $\lambda = 1.6$ (f) $\lambda = 2.0$

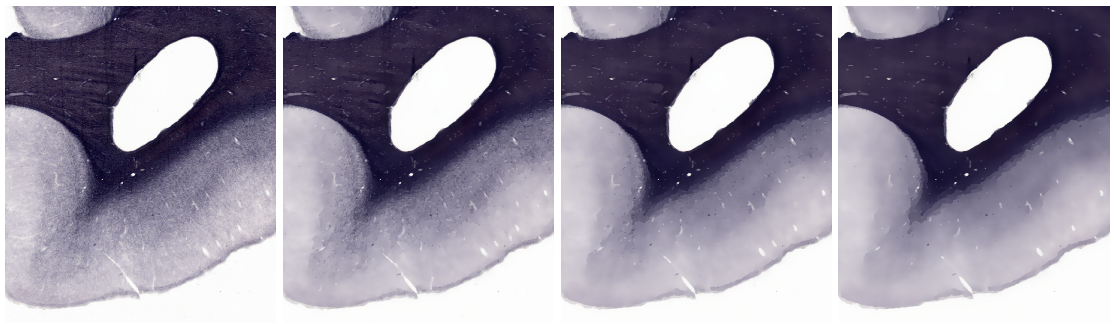


(g) $\lambda = 2.4$ (h) $\lambda = 2.8$ (i) $\lambda = 8$

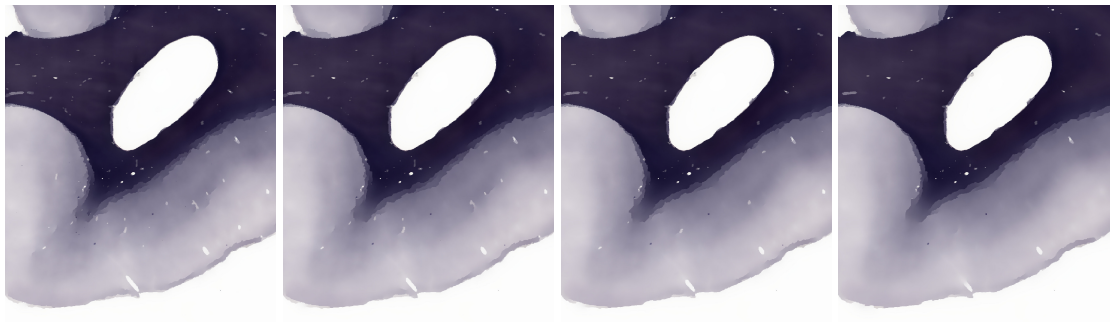
Figure 7.3.: Piecewise constant segmentation of the histological image 1.4 with varying parameter λ. Because the left and right half of the image are very similar, we concentrate on the right half to ease the comparison. As one would expect, with increasing λ the number of regions decreases.



(a) $\alpha = 90$

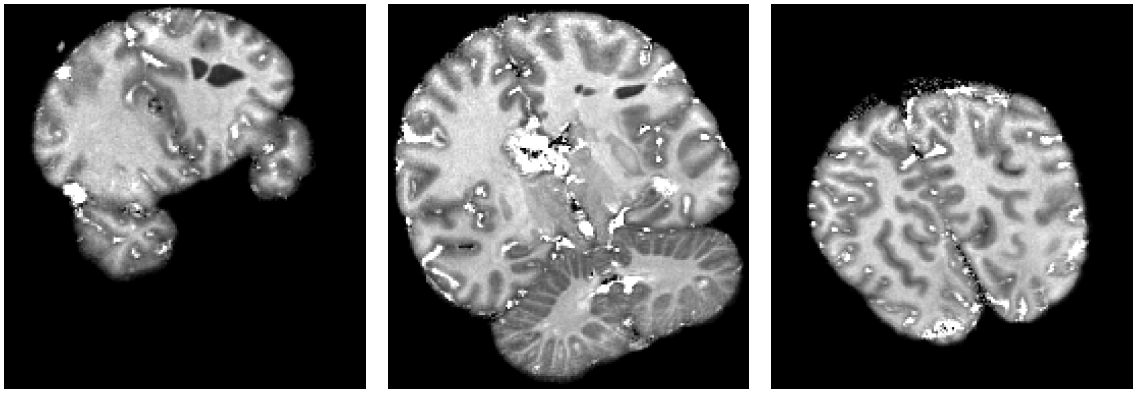


(b) Original (c) $\alpha = 30$ (d) $\alpha = 60$ (e) $\alpha = 90$

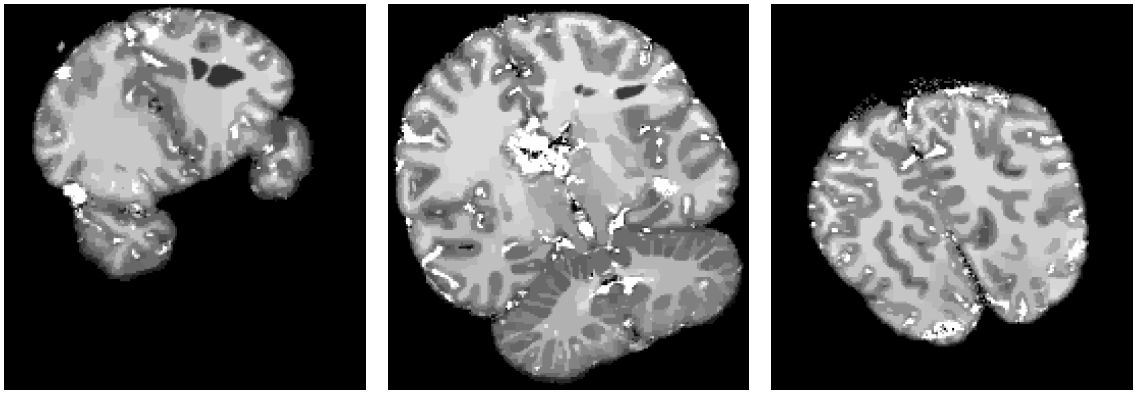


(f) $\alpha = 120$ (g) $\alpha = 150$ (h) $\alpha = 180$ (i) $\alpha = 210$

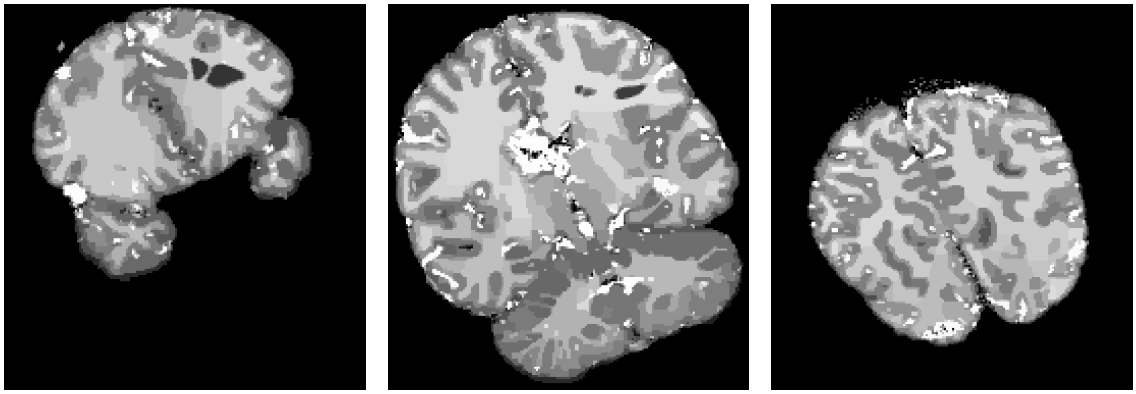
Figure 7.4.: Piecewise smooth segmentation of the histological image 1.4 with varying parameter α. We choose $\lambda = 0.001\alpha$, such that λ/α and in consequence the set K is constant. As one would expect, with increasing α the variation inside each region decreases.



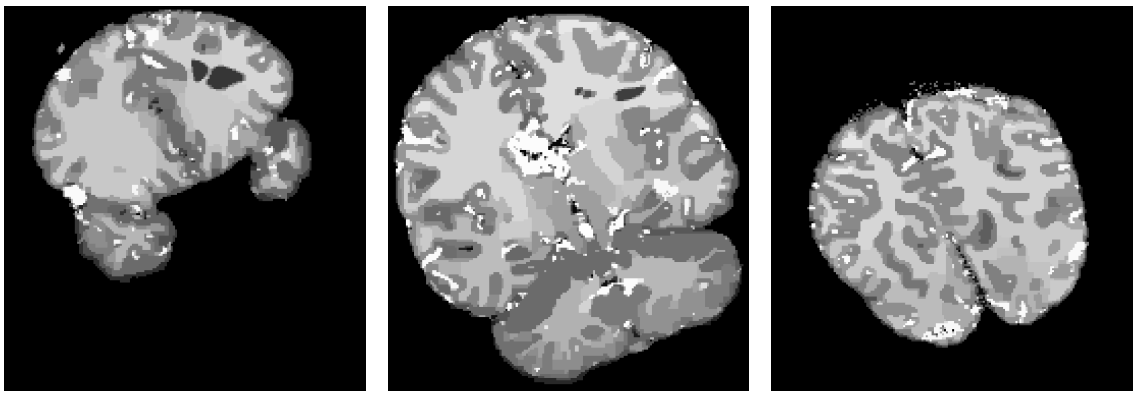
(a) Original (b) Original (c) Original



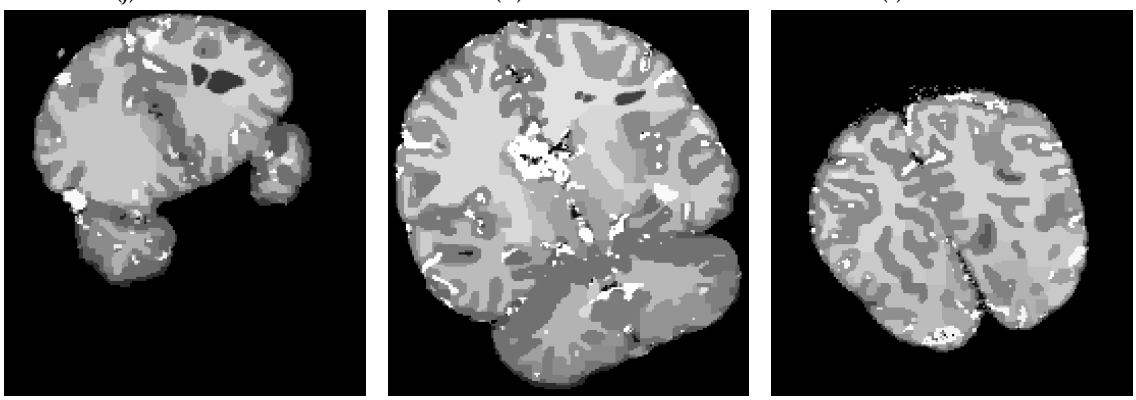
(d) $\lambda = 0.004$ (e) $\lambda = 0.004$ (f) $\lambda = 0.004$



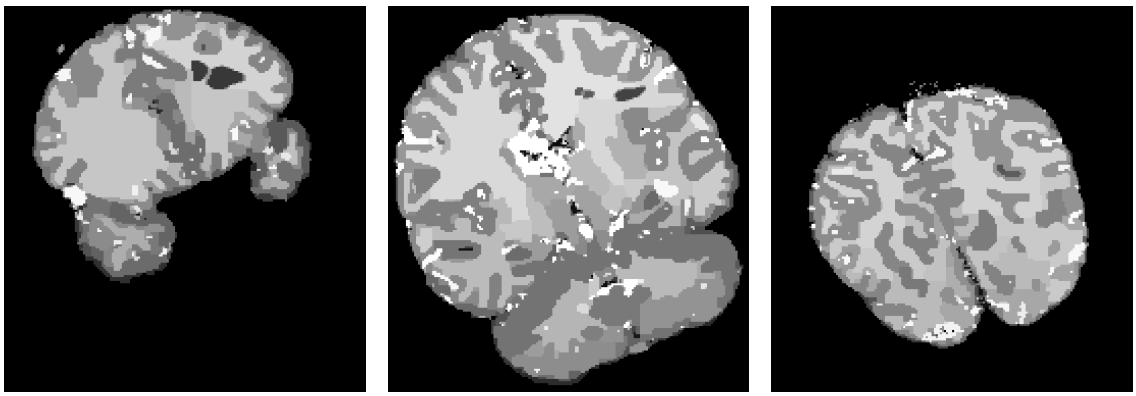
(g) $\lambda = 0.008$ (h) $\lambda = 0.008$ (i) $\lambda = 0.008$



(j) $\lambda = 0.012$ (k) $\lambda = 0.012$ (l) $\lambda = 0.012$

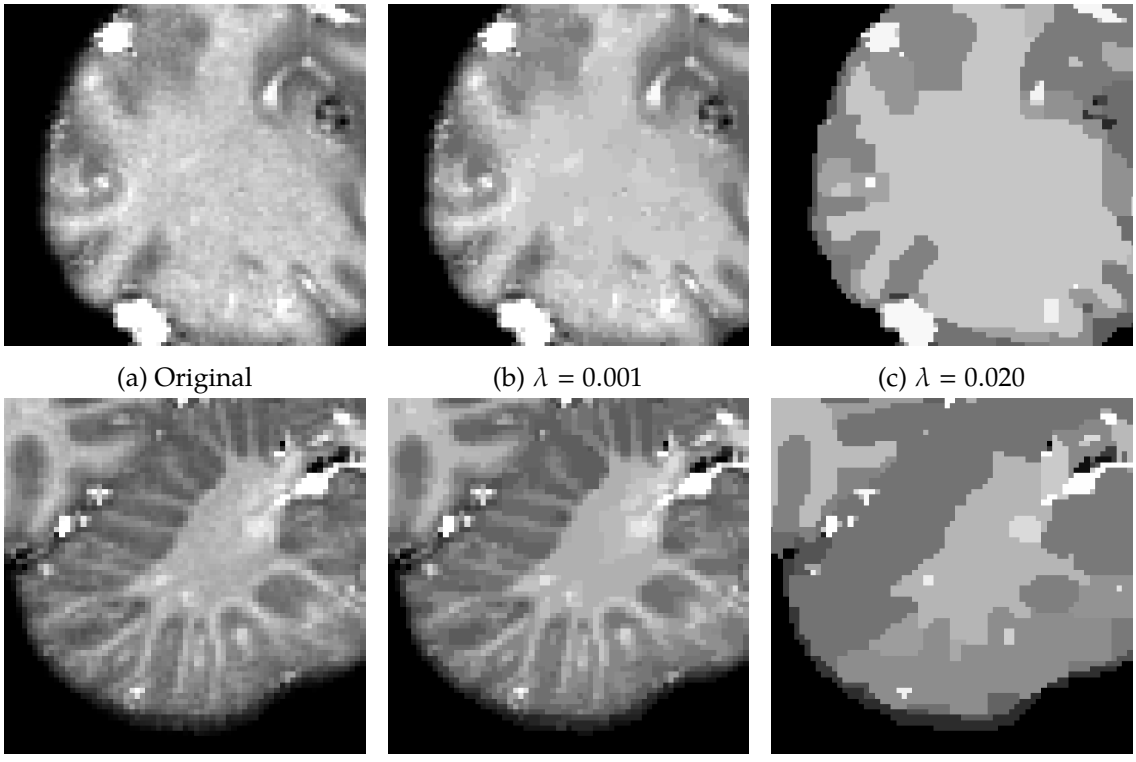


(m) $\lambda = 0.016$ (n) $\lambda = 0.016$ (o) $\lambda = 0.016$



(p) $\lambda = 0.020$ (q) $\lambda = 0.020$ (r) $\lambda = 0.020$

Figure 7.5.: Piecewise constant segmentation of the magnetic resonance image 1.3



(a) Original (b) $\lambda = 0.001$ (c) $\lambda = 0.020$

(d) Original (e) $\lambda = 0.001$ (f) $\lambda = 0.020$

Figure 7.6.: In the Mumford-Shah functional and the derived approximations, the parameters α, λ are typically set globally for the whole image. If an image contains both fine and coarse structures, a difficulty to select the optimal parameters may arise: $\lambda = 0.001$ is too small for (b), the segmentation is too coarse. For this region of the image, $\lambda = 0.02$ works better, see (c). On the other hand, for finer structures $\lambda = 0.001$ works better (see (e)) and the higher value of $\lambda = 0.02$ results in a loss of information, see (f). It could be promising to decompose the image into several levels (for example by wavelet transformation) and segment those levels independently or to use local parameters $\alpha(x)$ and $\lambda(x)$.

Chapter 8.

Combination of Segmentation Methods and Perspective

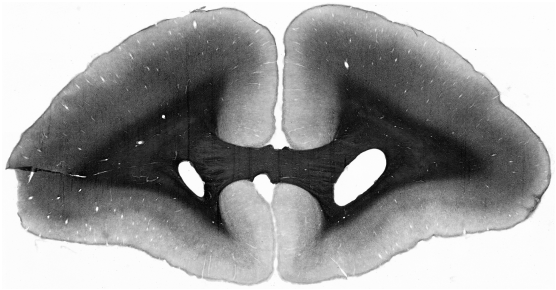
8.1. Fast Mumford-Shah and Thresholding

We already discussed that in the Fast Mumford-Shah method the number of regions can only be controlled in a limited way. But the output contains many regions that have a similar brightness, see also figure 8.2. This segmentation may be coarsened using thresholding, where the number of regions can be explicitly specified. Two possible results can be seen in figure 8.1.

(a) Three regions (including background), $\lambda = 2.51$

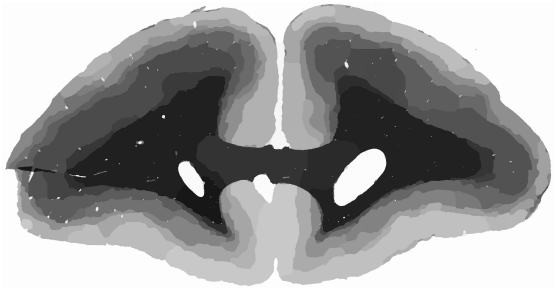
(b) Four regions (including background), $\lambda = 1.35$

Figure 8.1.: Thresholding after piecewise constant Fast Mumford-Shah Segmentation

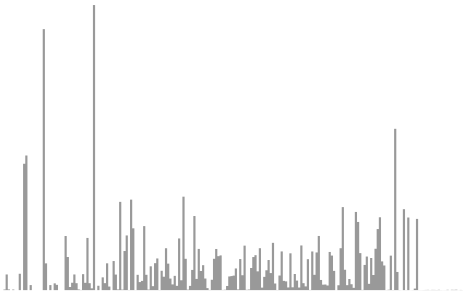


(a) Original

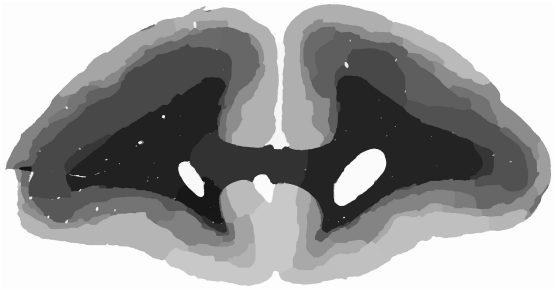
(b) Original



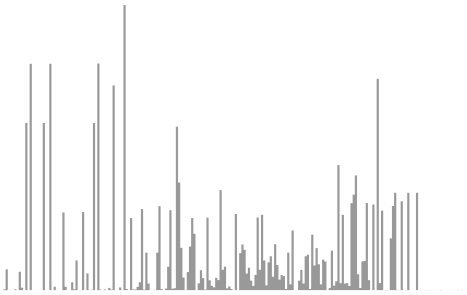
(c) $\lambda = 0.5$



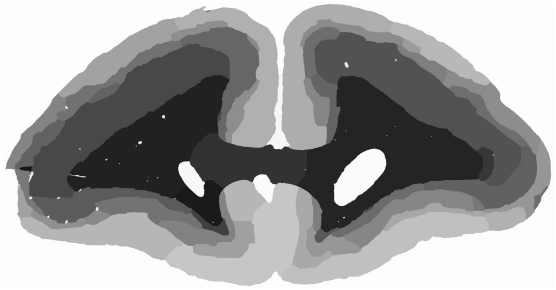
(d) $\lambda = 0.5$



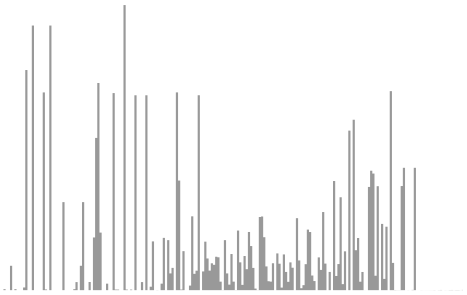
(e) $\lambda = 1.0$



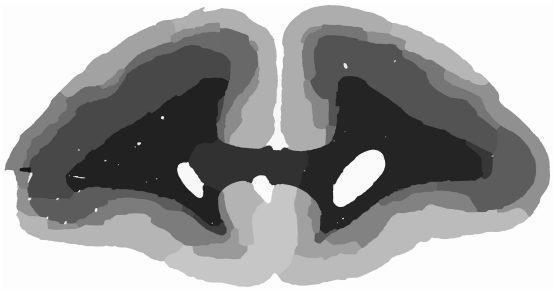
(f) $\lambda = 1.0$



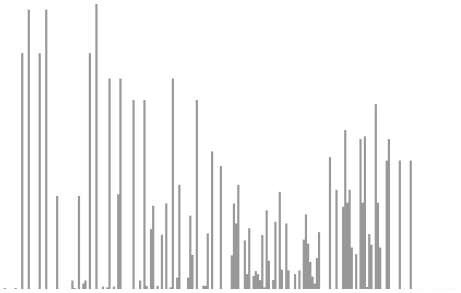
(g) $\lambda = 1.5$



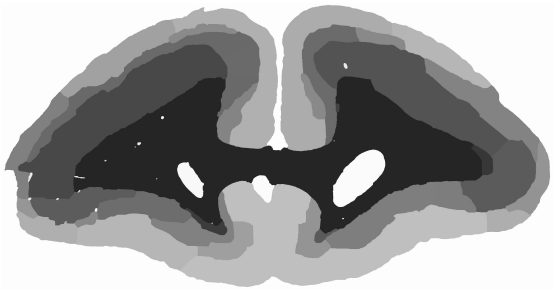
(h) $\lambda = 1.5$



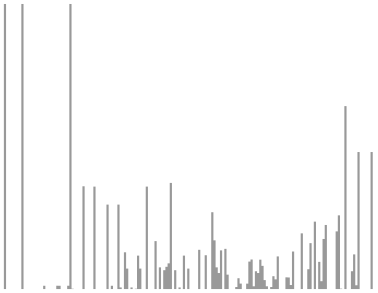
(i) $\lambda = 2.0$



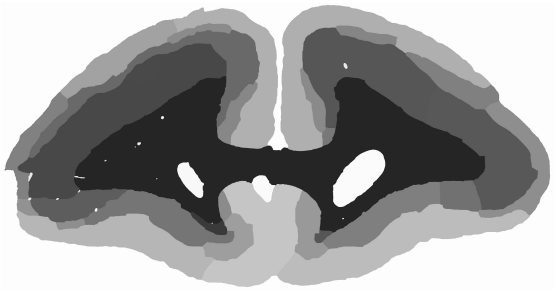
(j) $\lambda = 2.0$



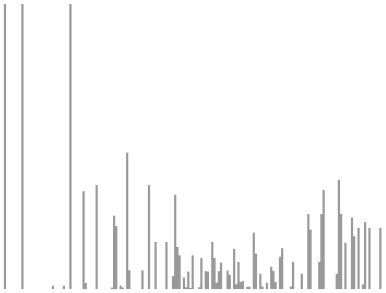
(k) $\lambda = 2.5$



(l) $\lambda = 2.5$



(m) $\lambda = 3.0$



(n) $\lambda = 3.0$

Figure 8.2.: On the left-hand side piecewise constant segmentations of the histological image using the Fast Mumford-Shah method with different parameters λ. On the right-hand side the respective histogram, where the bar corresponding to the white background pixels is omitted and the maximum is normalized to one. With increasing λ, the histograms show the development of gaps with increasing size, so thresholding could be a promising next processing step.

8.2. Verdict and Perspective

Besides thresholding, also the graph-based segmentation is a possible second processing step after the Fast Mumford-Shah segmentation. As we saw in section 3.1, partitioning a graph with n vertices by minimizing the normalized cut requires the computation of an eigenvector of length n. If $n = n_p$ is the number of points in Ω (the number of pixels/voxels), with increasing image size this computation quickly becomes impossible.

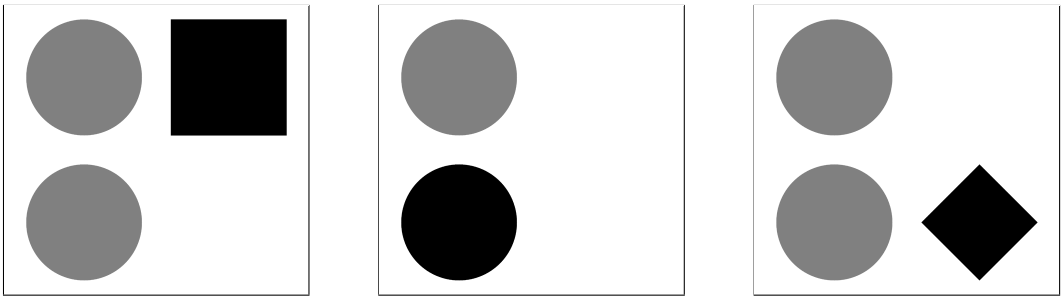
The application of the piecewise constant version of the Fast Mumford-Shah method yields a partition

$$\Omega = \Omega_1 \cup \ldots \cup \Omega_{n_r},$$

where n_r is much smaller than n_p. Now every region Ω_i instead of every point $x \in \Omega$ is considered to be a vertex in the graph, the resulting graph is much smaller and the eigenvector computation becomes possible.

Like thresholding, this approach can be used to coarsen a piecewise constant Fast Mumford-Shah segmentation. First experiments show that this idea is worth pursuing.

This combination could also be interesting for *multicontrast* (also called *multiband*) *segmentation*: Given are several images of the same scene or object, each highlighting a special area or emphasizing differences between certain areas, see also figure 8.3. The task is to create one segmented version of these images.



(a) (b) (c)

Figure 8.3.: Only (a) contains the square, only (b) highlights the difference between the two circles, only (c) contains the rhombus/rotated square. The task is to create one image, containing both squares within the same region, the lower circle also in that region and the upper circle in a different region.

Given those c images $f_i : \Omega \to \mathbb{R}, i \in \{1, \ldots, c\}$, let $f = (f_1, \ldots, f_c) : \Omega \to \mathbb{R}^c$ be their combination. Instead of c images mapping to a one-dimensional color space, we now have one image mapping to a c-dimensional color space. Using the piecewise constant version of the Fast Mumford-Shah method, we obtain $u : \Omega \to \mathbb{R}^c$, mapping into the same color space, and a partition of Ω. Then, as described above, a graph with respect

to the subsets in this partition may be constructed and the vertices of the graph can be assigned to k different regions using the minimal normalized cut method. Now, every point in Ω is uniquely assigned to one of that k regions, and we could construct a final image $\bar{u} : \Omega \to \{1, \ldots, k\}$ that is a segmented version of the original images (f_i). For this method, first experiments are promising as well.

In summary, the Fast Mumford-Shah approach is a sophisticated image segmentation method. It is capable of computing both piecewise constant and piecewise smooth segmentation and allows a flexible choice of the parameters α and λ. By combination with other segmentation methods, it is possible to explicitly set the desired number of regions. Because the parameters α, λ have to be set, Mumford-Shah and the derived approximations are supervised segmentation methods. Using a priori knowledge for special images, for example the expectation that the output image contains three circles in front of a background, it would be conceivable to tweak α, λ automatically such that this expectation is fulfilled. This could lead to an unsupervised segmentation method.

Appendix A.

Software Toolbox Documentation

Documentation of all user-relevant functions. For documentation of internal functions, see the respective code file.

A.1. MATLAB

A.1.1. MATLAB Functions

`bipartition` Partition the vertices V of a graph into two subsets $V = A \cup B$ using algorithm 1. If `f` is an image, then `u = bipartition(f)` is an image of the same size, where `u(i,j,k) = 1` if `u(i,j,k)` is in A, and `u(i,j,k) = -1` if `u(i,j,k)` is in B. If `f` is a cell array of length n, then `u = bipartition(f)` is a vector of length n, where `u(i) = 1` if the points corresponding to the brightness values in `f{i}` are in A and `u(i) = -1` otherwise. To use `bipartition`, the script `compileMex` has to be called first.

`callFms` Wrapper for the C function `fms.c` that has to compiled to `fms.out` first. For input image `f` and parameters `alpha`, `lambda` and `tolerance`, call

```
u = callFms(f,alpha,lambda,tolerance)
```

`f` will be exported as a binary file, the C program will be called, and the result will be imported to MATLAB. Note that `callFms` and `fmsMat` are called with the same parameters and might be interchanged.

`compileMex` Compiles the C-Mex functions.

`fmsMat` MATLAB implementation of algorithm 6. For input image `f` and parameters `alpha`, `lambda` and `tolerance`, call `u = fms(f,alpha,lambda,tolerance)`. Note that `callFms` and `fmsMat` are called with the same parameters and might be interchanged.

A.1.2. MATLAB GUI

The following functions are available with an accompanying graphical user interface:

guiThresh to perform thresholding as described in chapter 2. The number of thresholds can be set in the upper right of the user interface and the thresholds can be set using the sliders below. The button in the lower right sets the thresholds automatically using Otsu's method. The output image and the histogram with the thresholds can be seen on the left hand side. The histogram might be cropped using the sliders below. This is useful when the image contains many background pixels of the same color. See also figure A.1.

guiFms provides a graphical user interface to `fmsMat` and `callFms`. It allows to choose between the C and MATLAB version of algorithm 6, to choose between a piecewise constant and piecewise smooth segmentation, and to set the parameters α, λ and the tolerance ε. See also figure A.2.

guiFmsThresh is a combination of `guiThresh` and `guiFmsThresh` as described in chapter 8. It allows to compute a piecewise constant segmentation with parameters λ, ε and to use thresholding afterwards. The number of brightness classes (number of thresholds plus one) may be chosen and the thresholds can be set manually or automatically. See also figure A.3.

A.2. C Functions

`fms.c` is a C implementation of algorithm 6. It contains OpenACC directives and might be compiled for serial, multicore or GPU execution. If the executable is called `fms.out`, use the following syntax to call the program:

```
./fms.out $INPUT $OUTPUT $L $M $N $C $ALPHA $LAMBDA $PIECEWISE_CONSTANT
↪ $TOLERANCE
```

where

$INPUT is the binary file containing the input image,
$OUTPUT is a binary file that will contain the output after execution,
$L is equal to d_1, the size of the image in the first dimension,
$M is equal to d_2, the size of the image in the second dimension,
$N is equal to d_3, the size of the image in the third dimension,
$C is the number of colors,
$LAMBDA is the parameter λ in equation (7.11),
$ALPHA is the parameter α in equation (7.11),
$PIECEWISE_CONSTANT determines whether the segmentation will be piecewise constant (`PIECEWISE_CONSTANT=1`) or piecewise smooth (`PIECEWISE_CONSTANT=0`),
$TOLERANCE is the target tolerance ε in algorithm 6.

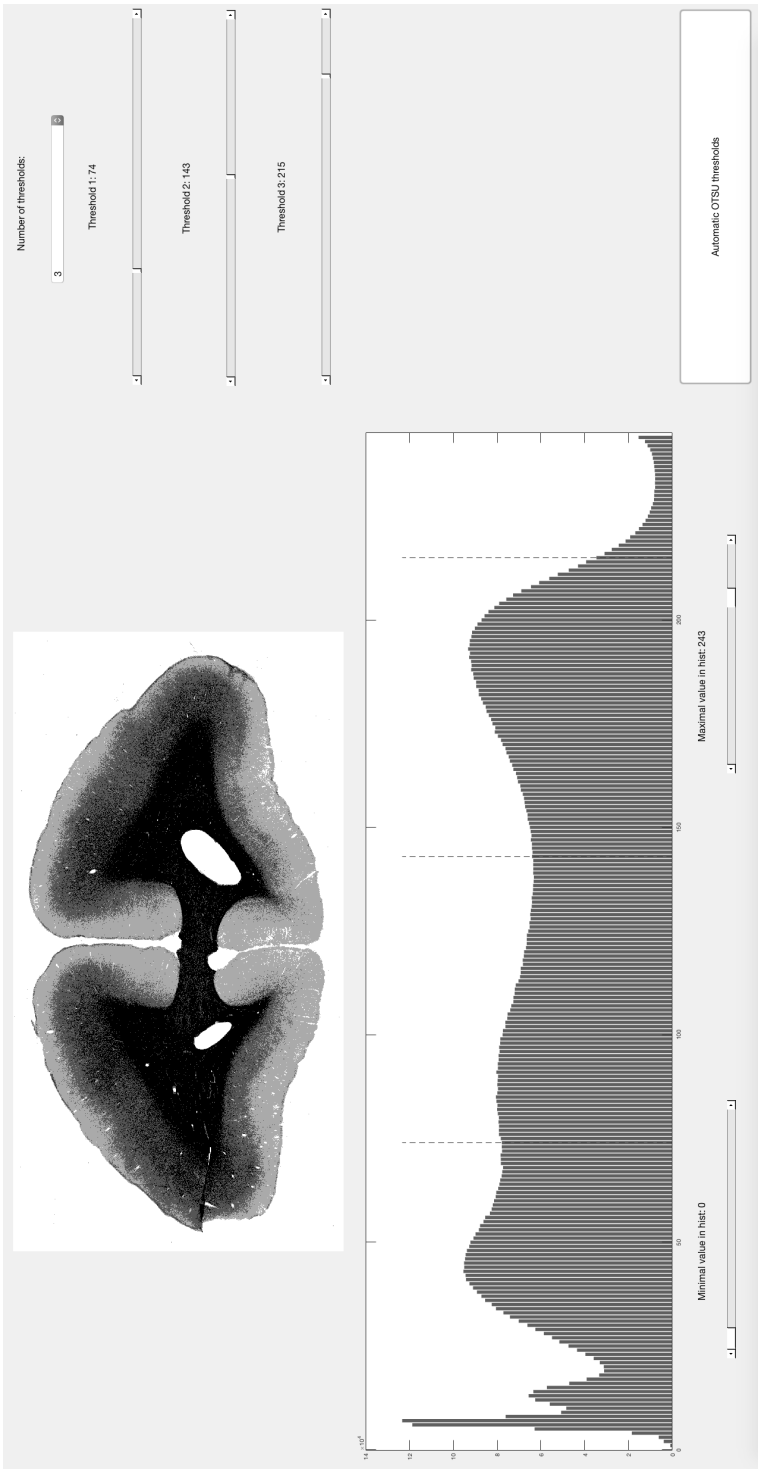


Figure A.1.: guiThresh

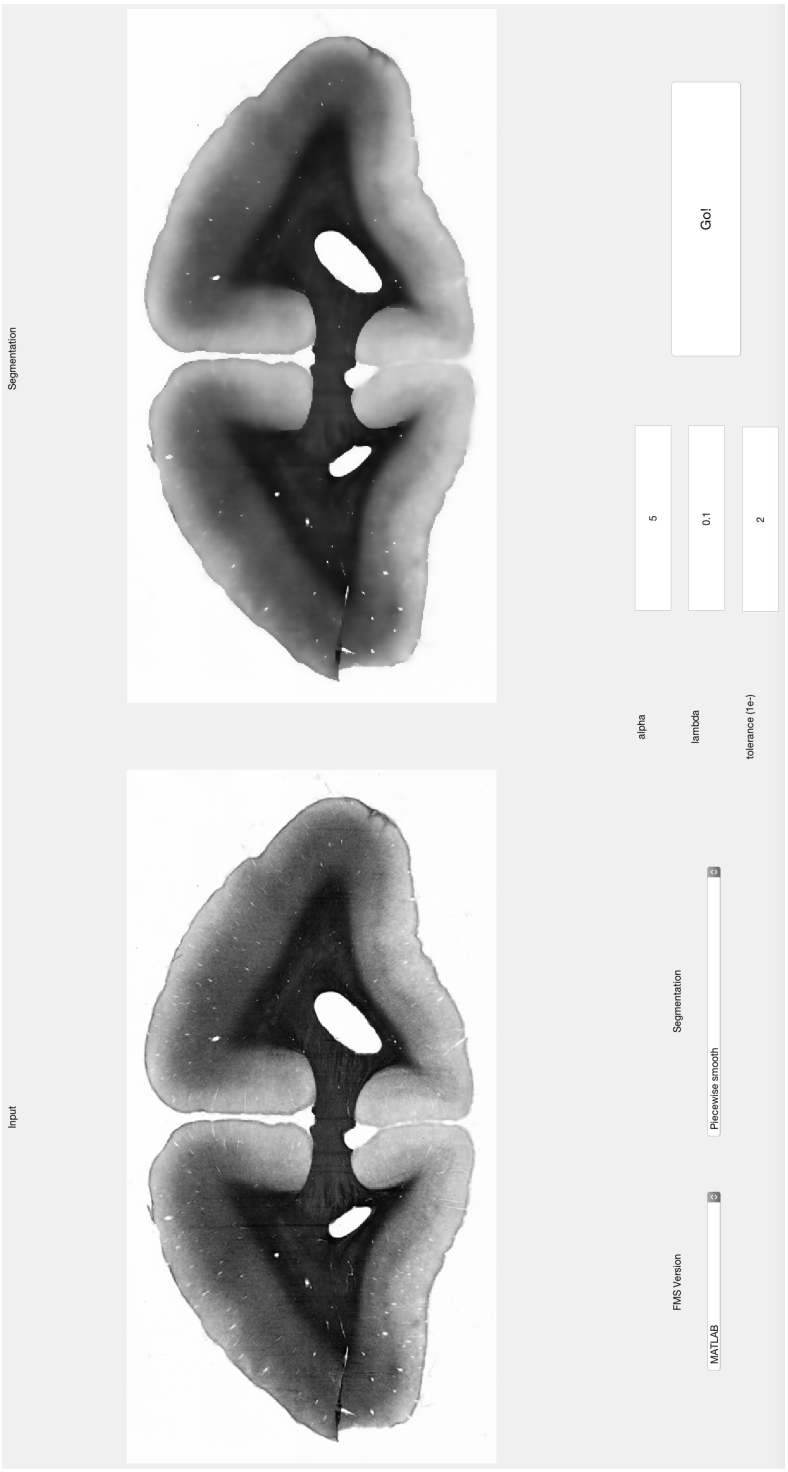


Figure A.2.: guiFms

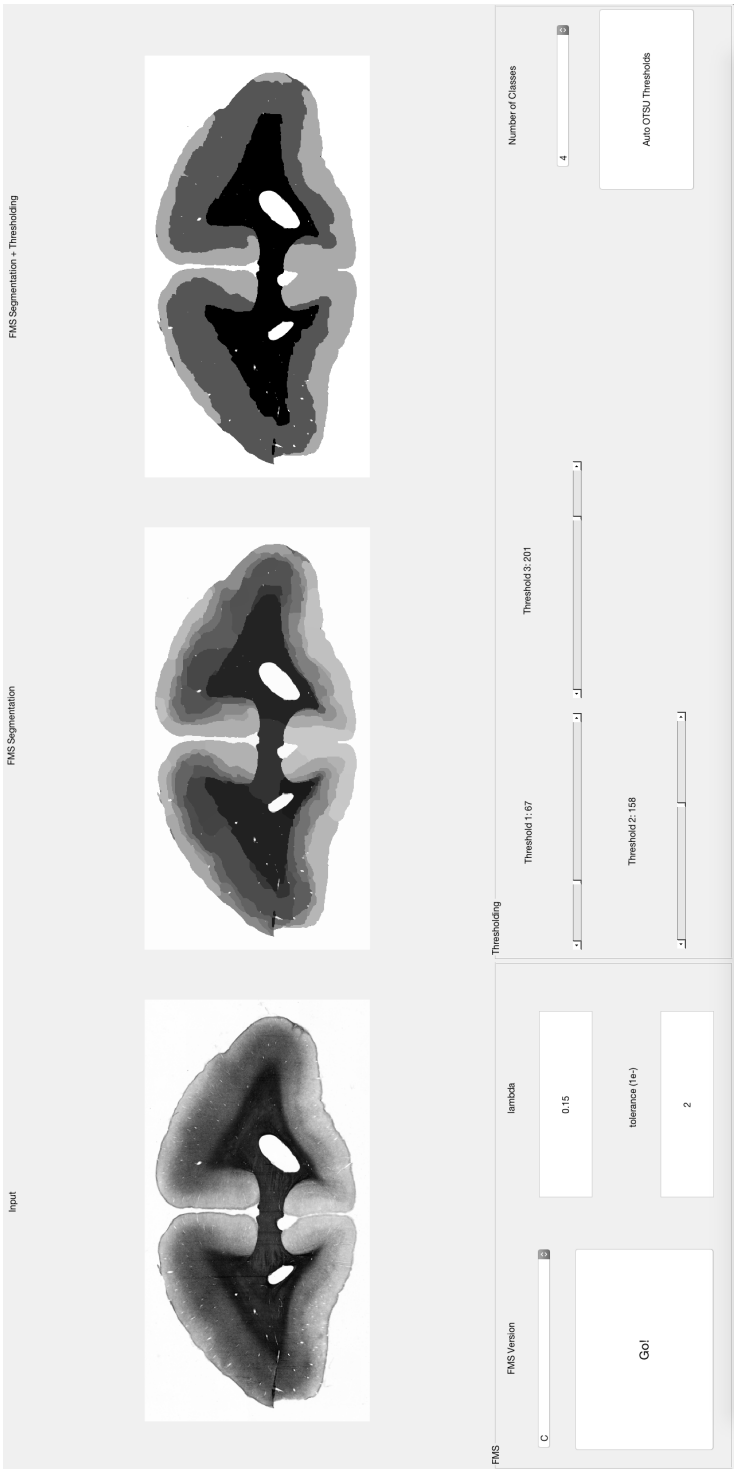


Figure A.3.: guiFmsThresh

A.3. Shell Scripts

`compileFms.sh` compiles `fms.c` and names the executable `fms.out`. If the `pgcc` compiler is available, it will be used to compile for a multicore target. The script also contains a compilation command for (Tesla) GPUs that is commented out and can be used instead of the multicore command. If the `pgcc` compiler is not available, the script falls back to the `clang` or `gcc` compiler, but *without using OpenACC*.

Bibliography

[1] *12 Tips for Maximum Performance with PGI Directives in C*. URL: http://www.nvidia.com/docs/IO/117377/directives-tips-for-c.pdf (visited on 01/02/2018).

[2] Giovanni Alberti and Carlo Mantegazza. 'A Note on the Theory of SBV Functions'. In: *Bollettino Unione Matematica Italiana* 11 (1997), pp. 375–382.

[3] Luigi Ambrosio. 'Variational Problems in SBV and Image Segmentation'. In: *Acta Applicandae Mathematicae* 17 (1989), pp. 1–40.

[4] Kristian Bredies and Dirk Lorenz. *Mathematische Bildverarbeitung. Einführung in Grundlagen und moderne Theorie*. 1. Auflage. Vieweg+Teubner, 2011.

[5] Antonin Chambolle and Thomas Pock. 'A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging'. In: *Journal of Mathematical Imaging and Vision* 40.1 (2011), pp. 120–145.

[6] Tony F. Chan and Luminita A. Vese. 'Active Contours Without Edges'. In: *IEEE Transactions on Image Processing* 10.2 (2001), pp. 266–277.

[7] Jürgen Elstrodt. *Maß- und Integrationstheorie*. 7. Auflage. Springer-Lehrbuch. Springer-Verlag Berlin Heidelberg, 2011.

[8] Lawrence C. Evans and Ronald F. Gariepy. *Measure Theory and Fine Properties of Functions*. Studies in Advanced Mathematics. CRC Press, 1992.

[9] David Mumford and Jayant Shah. 'Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems'. In: *Communications on Pure and Applied Mathematics* 42.5 (1989), pp. 577–685.

[10] Christoph Nicolai. 'Variationsmethoden in der Bildverarbeitung. Die Huber-Funktion im Regularitätsterm'. Bachelor thesis. Technische Universität Hamburg-Harburg, 2015.

[11] *OpenACC API Reference Guide*. Version 2.6. URL: https://www.openacc.org/sites/default/files/inline-files/OpenACC%20API%202.6%20Reference%20Guide.pdf.

[12] *OpenACC Programming and Best Practives Guide*. June 2015. URL: https://www.openacc.org/sites/default/files/inline-files/OpenACC_Programming_Guide_0.pdf.

[13] Nobuyuki Otsu. 'A Threshold Selection Method from Gray-Level Histograms'. In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66.

[14] Ralph Tyrrell Rockafellar. 'Characterization of the Subdifferentials of Convex Functions'. In: *Pacific Journal of Mathematics* 17.3 (1966), pp. 497–510.

[15] Ralph Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[16] Thomas Schmidt. *Variationsprobleme in BV*. July 2015. URL: `https://www.math.uni-hamburg.de/home/schmidt/lectures/BV.pdf`.

[17] Jianbo Shi and Jitendra Malik. 'Normalized Cuts and Image Segmentation'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905.

[18] Evgeny Strekalovskiy and Daniel Cremers. 'Real-Time Minimization of the Piecewise Smooth Mumford-Shah Functional'. In: *European Conference on Computer Vision* (2014), pp. 127–141.

[19] William P. Ziemer. *Weakly Differentiable Functions*. Springer, 1989.

Image Credits

All images not listed here are own images.

[20] Image was kindly provided by Siawoosh Mohammadi.

[21] Image was kindly provided by Harald Möller, Riccardo Metere, Afonso Silva, Stefan Geyer, Nicholas Bock and Katja Reimann.

List of symbols

$\mathfrak{B}(\Omega)$	Borel σ-algebra over Ω, the smallest σ-algebra containing all open subsets of Ω
$\langle \cdot, \cdot \rangle$	When X is a Banach space with dual space X^*, $u \in X$ and $w \in X^*$, then $\langle w, u \rangle \coloneqq w(u)$
$(\cdot, \cdot)$	Inner product in a Hilbert space
$\mathrm{BV}(\Omega)$	space of functions of bounded variation, definition 4.21 on page 27
$\mathrm{BV}_{\mathrm{loc}}(\Omega)$	space of weakly differentiable functions, definition 4.20 on page 27
$C^\infty(\Omega)$	space of arbitrarily often continuously differentiable functions
$C^k(\Omega)$	space of continuously differentiable functions up to order k
C	color space of an image, definition 1.1 on page 1
δ_x	Dirac measure: $\delta_x(A) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise} \end{cases}$
$\operatorname{dom} F$	domain of F, see definition 6.1 on page 41
$\mathcal{D}(\Omega, \mathbb{R}^d)$	$\mathcal{D}(\Omega, \mathbb{R}^d) \coloneqq \{u \in C^\infty(\Omega, \mathbb{R}^d) : \operatorname{supp} u \text{ compact in } \Omega\}$
∇u	In chapter 4, the classical derivative of u, if it exists. Otherwise, the approximate derivative as in definition 4.27 on page 31. In chapter 7, the discrete gradient as in definition 7.2 on page 58.
$\mathcal{H}^k$	$k-$dimensional Hausdorff measure, definition 4.16 on page 25
$\mathcal{L}^d$	Lebesgue measure on $\mathbb{R}^d$
$\nu \ll \mu$	ν is absolutely continuous with respect to μ, definition 4.5 on page 21
$\mathrm{L}^1_{\mathrm{loc}}(\Omega)$	space of locally integrable functions
$\mathrm{L}^1_{\mathrm{loc}}(\Omega, \mathbb{R}^n; \mu)$	space of locally integrable functions with integration with respect to the measure μ instead of the Lebesgue-measure

$L^p(\Omega)$	Lebesgue space
$L^p\left(\Omega, \mathbb{R}^n; \mu\right)$	Lebesgue space with integration with respect to the measure μ instead of the Lebesgue-measure
$\nu \perp \mu$	ν and μ are singular, definition 4.5 on page 21
$\mathbb{R}_\infty$	$\mathbb{R}_\infty := \mathbb{R} \cup \{\infty\}$
$\mathrm{RM}\left(\Omega, \mathbb{R}^d\right)$	space of finite vector Radon measures on Ω, definition 4.14 on page 24
$\mathrm{RM}_{\mathrm{loc}}\left(\Omega, \mathbb{R}^d\right)$	space of vector Radon measures on Ω, definition 4.10 on page 22
$\mathbb{R}_+$	$\mathbb{R}_+ := \{x \in \mathbb{R} \mid x \geq 0\}$
$\mathrm{SBV}(\Omega)$	space of special functions of bounded variation, definition 4.29 on page 32