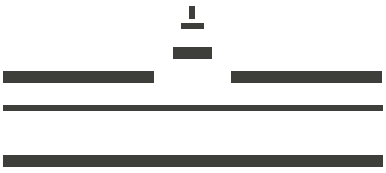


Westfälische
Wilhelms-Universität
Münster

Variational Methods for Joint Motion Estimation and Image Reconstruction

Hendrik Meinert Dirks

- 2015 -

wissen.leben
WWU Münster

Angewandte
Mathematik
Münster

Fach: Mathematik

Variational Methods for Joint Motion Estimation and Image Reconstruction

Inaugural-Dissertation

zur Erlangung des Doktorgrades der Naturwissenschaften
- Dr. rer. nat. -
im Fachbereich Mathematik und Informatik
der Mathematisch-Naturwissenschaftlichen Fakultät
der Westfälischen Wilhelms-Universität Münster

eingereicht von
Hendrik Meinert Dirks
aus Wilhelmshaven
- 2015 -

Dekan	Prof. Dr. Martin Stein
Erster Gutachter:	Prof. Dr. Martin Burger (Universität Münster)
Zweiter Gutachter:	Dr. Carola-Bibiane Schönlieb (University of Cambridge)
Tag der mündlichen Prüfung:	03.06.2015
Tag der Promotion:	03.06.2015

Abstract

In this thesis we present different techniques for motion estimation from image sequences and combine them with image reconstruction.

The main body of this work is divided in two modeling and one application part. The first model is a variational approach for motion estimation from image sequences. Mathematical background as well as different models for motion estimation are presented. We illustrate the numerical realization based on a primal-dual framework and evaluate our model towards different types of motion.
In the second main part we connect the field of motion estimation to the task of image reconstruction. Here, we deduce variational models for joint motion estimation and image reconstruction, prove existence of minimizers and present primal-dual schemes for the numerical implementation. The later numerical evaluation illustrates benefits of the joint framework and contains promising results for temporal inpainting.
The application part divides into chapters about image segmentation, where Geodesic Active Contour and Chan-Vese models are applied to images of vertebra and intracellular flows. In the context of intracellular flow, we later apply our joint models to denoise image sequences and estimate their underlying motion simultaneously.

Keywords: Image Processing, Image Reconstruction, Motion Estimation, Joint Image Reconstruction and Motion Estimation, Total Variation, Optical Flow, Microscopy Imaging, Biomedical Imaging, Temporal Inpainting, Variational Methods, Primal-Dual Methods

Acknowledgements

I would like to thank

Martin Burger for giving me all the opportunities within the last six years. Starting with a job as a student assistant followed by an interesting diploma thesis and the offer to continue my research as a PhD student. Moreover for introducing me to Carola-Bibiane Schönlieb and supporting my research at the University of Cambridge, for always patiently answering my questions and being the fastest and shortest answerer to emails one can imagine.

Carola-Bibiane Schönlieb for a great semester at the University of Cambridge and introducing me to so many people there. Furthermore for co-reviewing my thesis and for always having time and answering my questions, for inviting me to breakfast, lunch and a great barbecue last summer.

My colleagues and ex-colleagues from the work groups in Münster and Cambridge and the whole Institute for Computational and Applied Mathematics, in particular

- *Martin Benning* for being a good friend and such a nice housemate, for my rememberable first pub crawl in Cambridge and many nice evenings together. Moreover for patiently answering my questions and having a great kind of humor. Finally, of course, for introducing me to a magical drink called *Ratzeputz*.

- *Pia Heins* for being a good friend and great office mate for years, for mathematical, emotional and silly discussions and watering all my tomatoes.

- *Frank Wübbeling* for many interesting discussions, for answering my questions and organizing the skiing seminar.

- *Christoph Brune* for many interesting discussions and sharing his code for colored motion fields.
- *Lena Frerking* for proofreading my thesis and sharing our research of motion estimation.
- *Claudia Giesbert* and *Carolin Gietz* for being such nice and helpful in any kind of organizational problem.
- *Jan Hegemann* for his CV template and for sharing the addiction to strategy computer games.
- *Jahn Müller* for his great LaTeXtemplate
- *Jan Lellmann* and *Tuomo Valkonen* for daily lunches in Cambridge.

My parents *Anke Dirks* and *Meinert Dirks* for being always there for me, for putting their own interests aside when it comes to us children, for building me up when I was down and for always believing in me.

My brother *Malte Dirks* for being always there for me, for giving useful advices in difficult situations and for many informatory discussions.

Carolin Roßmanith for supporting me more than I can tell, for always believing in me, for building me up so many times, for being together even when living thousands of kilometers apart and for going through all good and bad times together.

The whole Portugal crew, especially *Christoph Lindemann* and *Robert Wolff*, for spending many enjoyable holidays together in Vale do Lobo.

All my friends, for being there for me.

Everyone that should be mentioned here, but whom I forgot. Sorry!

Contents

$\mathbb{N}$	The set of natural numbers
$\mathbb{R}$	The set of real numbers
$\bar{\mathbb{R}}$	The set $\mathbb{R} \cup \{\infty\}$
$\|\cdot\|$	Norm
$\|\cdot\|$	Semi-norm
$\mathcal{X}$	A real Banach space
$\mathcal{X}^*$	The dual space of $\mathcal{X}$
$L^p(\Omega)$	The space of Lebesgue measurable functions
$L^p(0,T;\mathcal{X})$	For $1 \leq p < \infty$ and a Banach space $\mathcal{X}$ with norm $\|\cdot\|_{\mathcal{X}}$ the space of functions which are dt–measurable, i.e. the function $t \mapsto u(\cdot,t)$ is dt–measurable. This is a Banach space with norm $$\|u\|_{L^p(0,T;\mathcal{X})} = \left(\int_0^T \left\|u(\cdot,t)\right\|_{\mathcal{X}}^p dt \right)^{\frac{1}{p}}$$
$\mathcal{D}'([0,T] \times \Omega)$	The space of distributions on the set $[0,T] \times \Omega$
$W^{k,p}(\Omega)$	The space of Sobolev functions
$H^k(\Omega)$	The Hilbert space $H^k = W^{k,2}$
$BV(\Omega)$	The space of functions with bounded variation
$C_0^\infty(\Omega)$	The space of infinitely often differentiable functions with compact support on Ω
$\rightarrow$	Convergence in the strong (norm) topology
$\rightharpoonup$	Convergence in the weak topology
$\rightharpoonup^*$	Convergence in the weak* topology
G^*	The convex conjugate of G
G^{**}	The biconjugate of G
$\|u\|_{BV}$	The total variation of u, defined as $\sup_{\phi \in C_0^\infty(\Omega;\mathbb{R}^N), \|\phi\|_\infty \leq 1} \int_\Omega u \nabla \cdot \phi dx$
$TV(u)$	The total variation of u
$\nabla_x u$	For a function $u(x_1,\ldots,x_n,t)$ depending of space and time, this is the vector $(u_{x_1},\ldots,u_{x_n})$ consisting of partial derivatives with respect to the spatial dimension
$\mathcal{X} \hookrightarrow \mathcal{Y}$	$\mathcal{X}$ embeds continuously into $\mathcal{Y}$
$\mathcal{X} \subset\subset \mathcal{Y}$	$\mathcal{X}$ embeds compact into $\mathcal{Y}$
$\{\Phi < 0\}$	Short notation for the set $\left\{x \in \Omega : \Phi(x) < 0\right\}$
$\mathcal{X} \cong \mathcal{Y}$	The spaces $\mathcal{X}$ and $\mathcal{Y}$ are isomorph.

1 Introduction

1.1. Motivation

The central point of interest in this thesis is the estimation of motion in image sequences. From a mathematical point of view, motion is defined as the change of spatial location of an object with respect to time. There exist different ways to quantify motion between two subsequent images. The most common parameter is the so-called *velocity*, which is defined as the difference between the starting point of a moving object in the first image and its terminal point in the following image. The velocity couples speed and direction of motion at the same time. Estimating motion is a current challenge in many fields of research. Consequently, let us start with four completely different examples that have the aspect of motion in common.

Self-driving cars, what many people have dreamed of for decades, are about to become reality. Google's *self-driving car* [40] is already being tested on US streets. While driving, the internal software has to estimate the motion of other moving cars as well as pedestrians, animals or even soccer balls being kicked out onto the street. The motion has to be estimated in real time, which is crucial for the safety of passengers and environment.

In modern medicine, biology, chemistry, etc. even the smallest objects are observed by high resolution microscopy. Betzig [16], Moerner [62] and Hell [42] were awarded with the Nobel Price in Chemistry for *"bringing optical microscopy into the nanodimension"*. However, in every aspect of these subjects, the scientists were mainly interested in the dynamics of the observed object. For example, in biology, intracellular flows are a current field of science. In medicine, understanding the pattern formation of cancer

cells might help to develop new drugs. For both applications, estimating motion from recorded image data is the key task.
For weather forecasting, thousands of small stations around the Earth and dozens of satellites are continuously recording the current meteorological situation. The current weather state is always the initial value for the forward model that aims to predict our weather tomorrow or in one week. Analyzing the movement of clouds plays an important role here. Moreover, extreme weather situations, such as tornados or even hurricanes, are not entirely understood to this day. Estimating their flows using satellite data is the current challenge for scientists and researchers as yet.

Most of the imaging problems discussed in this thesis, such as motion estimation or image reconstruction, are so-called inverse problems. We seek to find an unknown cause u such as a velocity field or a denoised image, but only measure data f. For motion estimation, we measure two images containing some kind of object displacement. We are interested in the motion field generating this dynamic process. In the context of image denoising, we are generally interested in a clean image u, but often only able to measure a noisy or blurred variant f. From a mathematical point of view, u and f are connected by the equation

$$\mathcal{A}u = f + \sigma =: \tilde{f}, \tag{1.1}$$

where $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ denotes a linear and compact operator, $\mathcal{X}$ denotes the space of solutions, $\mathcal{Y}$ the space of measurements and σ random noise. Solving (1.1) for u, given f, constitutes the solution to an inverse problem. The solution to this inverse problem is often ill-posed in the sense of Hadamard (1902):

Definition 1.1.1. Ill-posed Problem
Hadamard defines a problem as *well-posed* if and only if

- There exists a solution to the problem. (Existence)
- The solution is unique. (Uniqueness)
- The solution continuously depends on the given data. (Stability)

If one of the conditions is violated, then the problem is called *ill-posed*

Due to the compactness, we cannot invert the operator continuously (cf. [32]) and consequently, the third condition is violated. In fact, the problem is more complex. If

we could invert the operator, Equation (1.1) would read

$$u = \mathcal{A}^{-1} f + \mathcal{A}^{-1} \sigma.$$

This looks good at a first glance, but the part $\mathcal{A}^{-1}\sigma$ is random, usually unpredictably large and completely disturbs the recovery of u. In the course of this thesis, we will see that for motion estimation, the measurements are usually in $\mathbb{R}^n$, but the unknown is in $\mathbb{R}^{2n}$. Also, the system is massively underdetermined and condition two is violated.

A way out is given by so-called variational approaches. Here, we define a functional such that the minimum of the functional coincides with the expected solution:

$$\hat{u} = \underset{u \in dom(\mathcal{A})}{\arg\min} \left\{ \mathcal{D}(\mathcal{A}u, f) + \alpha \mathcal{R}(u) \right\} \tag{1.2}$$

The first part is generally known as *data term* and measures the distance between measured data f and $\mathcal{A}u$. Hence, for the solution $\hat{u}$, we ensure that $\mathcal{D}(\mathcal{A}u, f)$ is small so that we are not too far from solving Equation (1.1). The energy in Equation (1.2) contains a second term, the so-called *regularizer*. This additional energy term allows us to promote certain solutions and model additional a-priori information. The most common regularizers penalize u or derivatives of u in a suitable norm.
In this thesis, we even go one step further. Consider the unknown u from Equation (1.2) as an image and let us denote $\boldsymbol{v}$ as an unknown velocity field. For the velocity field $\boldsymbol{v}$, we are able to use a variational approach similar to Equation (1.1), where the velocity field should fit to given image data u and certain a-priori information is modeled by a regularizer. A joint variational model for u and $\boldsymbol{v}$ can now be written as:

$$(\hat{u}, \hat{\boldsymbol{v}}) = \underset{u,v}{\arg\min} \left\{ \mathcal{D}(\mathcal{A}u, \tilde{f}) + \alpha \mathcal{R}(u) + \beta \mathcal{S}(\boldsymbol{v}) + \mathcal{C}(u, \boldsymbol{v}) \right\}.$$

Similar to the previous model, we consider measured data $\tilde{f}$, a data term $\mathcal{D}(\mathcal{A}u, \tilde{f})$ and an image regularizer $\alpha\mathcal{R}(u)$. Now, $\mathcal{C}(u, \boldsymbol{v})$ acts as a data term for the unknown velocity field $\boldsymbol{v}$ together with a regularizer $\beta\mathcal{S}(\boldsymbol{v})$ carrying additional a-priori information. Moreover, $\mathcal{C}(u, \boldsymbol{v})$ draws a connection between the image and underlying dynamics. As we will see in later chapters, this connection is beneficial for image reconstruction as well as motion estimation.

1.2. Contribution

Estimating the flow from image sequences has been discussed in literature for decades. Already in 1981, Horn and Schunck proposed a variational model for flow estimation [43]. This basic model used the L^2 norm for the optical flow term as well as for the regularizer and became very popular. Aubert et al. analyzed the L^1 norm for the optical flow constraint [7] in 1999 and demonstrated its advantages towards a quadratic L^2 norm. In 2006, Papenberg, Weickert et al. [56] introduced the total variation regularization, resp. the differentiable approximation, to the field of flow estimation. An efficient duality-based $L^1 - TV$ algorithm for flow estimation was proposed by Zach, Pock and Bischof in 2007 [77].
The topic of joint models for motion estimation and image reconstruction was already introduced by Tomasi and Kanade [70] in 1992. Instead of a variational approach, they use a matrix-based discrete formulation with constraints to the matrix rank to find a proper solution. In 2002, Gilland, Mair, Bowsher and Jaszczak published a joint variational model for gated cardiac CT [38]. For two images, they formulated a data term, based on the Kullback-Leibler divergence (cf. [20] for details) and incorporated the motion field via quadratic deformation term and regularizer. Bar, Berkels, Rumpf and Sapiro proposed a variational framework for joint motion estimation and image deblurring in 2007 [11]. The underlying flow was assumed to be a simple translation and coupled into a blurring model for the foreground and background. This resulted in a Mumford-Shah-type functional. Also in 2007, Shen, Zhang, Huang and Li proposed a statistical approach for joint motion estimation, segmentation and superresolution [65]. The model assumed an affine linear transformation of segmentation labels to incorporate the dynamics and was solved calculating the MAP solution. Another possible approach was given by Brune in 2010 [20]. The 4d (3d + time) variational model consists of an L^2 data term for image reconstruction and incorporates the underlying dynamics using a variational term, introduced by Benamou and Brenier [14, 13]. This variational term solves Monge's classical transport problem.

1.3. Organization of this Work

The previous section served as motivation for this thesis and illuminated the topics in which we are interested.
Chapter two is going to provide the necessary mathematical framework for the upcoming chapters. Starting with fundamental definitions of function spaces with a special

focus on functions with bounded variation, we also introduce Bochner spaces, required for the analyses in Chapter 5. In the context of variational calculus, we introduce the concept of Fenchel duality. We devote significant attention to duality in image processing and its application for operator decoupling. We end this chapter with algorithms for the minimization of variational models.

In chapter three, the focus lies on applied problems, and focuses on models for image segmentation. We introduce the Geodesic Active Contours model and the Chan-Vese model, discuss their benefits, their drawbacks and deduce numerical realizations. We end this chapter with interesting applications for feature extraction from vertebrae and analysis of intracellular flows of microscopy data.

In Chapter four, we introduce into the field of motion estimation from images based on variational models. We begin deriving the optical flow and mass preservation constraint, which act as data fidelity for our variational model. These data terms are penalized with $\|\cdot\|_1$ and $\|\cdot\|_2^2$ norm and coupled with different regularizers. This leads to a total of six variational models. Efficient numerical schemes based on the popular primal-dual framework of Chambolle and Pock [26] are deduced. The proposed models are evaluated in terms of flow-adapted error measures and runtime. The spectrum of observed datasets contains basic flow types, influence of noise and scenes from the IPOL optical flow database [44].

Based on the results in Chapter four, we introduce novel variational models for joint motion estimation and image reconstruction in Chapter five. Besides proofs for the existence of minimizers, we give a detailed derivation of numerical schemes. In the end, our models are evaluated for different data situations, from joint denoising and motion estimation up to temporal inpainting based on estimated motion. In the final Chapter six, we couple the introduced techniques for image segmentation, denoising and motion estimation and apply all of them to a microscopy dataset. The aim here is to enhance image quality, track the cell and quantify intracellular motion.

In the Appendix, we derive an alternative approach for image discretization for flow estimation purposes. Moreover a $O(n)$ fast marching algorithm, widely used for levelset recalculation used in Chapter three and six, is presented.

2 Mathematical Preliminaries

In this chapter we provide the mathematical background for this thesis. We start with some fundamental definitions about function spaces. Beginning with Lebesgue and Sobolev spaces, we introduce the space of functions with bounded variation. Moreover, the so-called Bochner spaces, required for the analysis in Section 5, are defined. In this context, we need a set of embeddings for the introduced spaces. We also discuss some aspects of variational calculus in a subsequent section. Starting with definitions required for the existence of minimizers we also discuss the concept of Fenchel duality. We end this chapter with a section about efficient minimization of variational models, concentrating on a summary of the primal-dual framework of Chambolle and Pock [26]. Let, if not explicitly defined differently, for this thesis $\Omega \subset \mathbb{R}^n$ be an open and bounded domain with Lipschitz boundary.

2.1. Function Spaces

Definition 2.1.1. Dual Space
Let $\mathcal{X}$ be a metric space. The dual space $\mathcal{X}^*$ is the space of all continuous linear functionals $l : \mathcal{X} \to \mathbb{R}$. The norm on $\mathcal{X}^*$ is given by

$$\|l\|_{\mathcal{X}^*} = \sup_{x \in \mathcal{X} \setminus \{0\}} \frac{|l(x)|}{\|x\|}$$

Definition 2.1.2. Reflexive Space
Let $\mathcal{X}$ be a metric space and $\mathcal{X}^*$ the corresponding dual space. Similar to the dual

space of $\mathcal{X}$, we can define the dual space $\mathcal{X}^{**}$ of $\mathcal{X}^*$, the so-called bidual of $\mathcal{X}$. If $\mathcal{X}$ coincides with its bidual $\mathcal{X}^{**}$, $\mathcal{X}$ is called *reflexive*.

Examples for reflexive Banach spaces are $L^p(\Omega)$ for $1 < p < \infty$.

Lemma 2.1.3. Dual Inclusion
Let $\mathcal{X}, \mathcal{Y}$ be Banach spaces and $\mathcal{X}^*, \mathcal{Y}^*$ be their corresponding dual spaces. If $\mathcal{X} \subset \mathcal{Y}$ we have $\mathcal{Y}^* \subset \mathcal{X}^*$.

Proof. Let $l(\cdot)$ be an element of $\mathcal{Y}^*$, e.g. a continuous linear functional on $\mathcal{Y}$. Now it is easy to see that l must also be a continuous linear functional on $\mathcal{X}$, because $\mathcal{X}$ forms a subspace of $\mathcal{Y}$. □

For the advanced analysis the general definition of (strong) norm-convergence is insufficient. We need a weaker definition, namely the weak and the weak-* topology.

Definition 2.1.4. Weak and Weak* Topology
Let $\mathcal{X}$ be a Banach space and $\mathcal{X}^*$ its corresponding dual space. We call the sequence $u_k \in \mathcal{X}$ *weakly convergent* to $u \in \mathcal{X}$ if the evaluations by bounded linear functionals v converge for every $v \in \mathcal{X}^*$,

$$u_k \rightharpoonup u \Leftrightarrow \langle v, u_k \rangle \to \langle v, u \rangle \, \forall v \in \mathcal{X}^*.$$

For the dual space $\mathcal{X}^*$ the weak-* convergence can be defined as

$$v_k \rightharpoonup^* v \Leftrightarrow \langle v_k, u \rangle \to \langle v, u \rangle \, \forall u \in \mathcal{X}.$$

By this definition we can easily see that for reflexive spaces ($X^{**} \equiv X$) the weak and the weak-* convergence coincide.

2.1.1. Lebesgue Spaces

Definition 2.1.5. Lebesgue Space
Let u be a measurable function and $1 \leq p < \infty$. The Lebesgue space $L^p(\Omega)$ contains all functions whose absolute value raised to the p-th power has a finite integral

$$L^p(\Omega) = \left\{ u : \Omega \to \mathbb{R} : \int_\Omega |u|^p \, dx < \infty \right\}.$$

The norm in $L^p(\Omega)$ is given by the p-th root of the integral:

$$\|u\|_{L^p(\Omega)} := \left(\int_\Omega |u|^p\right)^{\frac{1}{p}}.$$

We usually write $\|u\|_p$ instead of $\|u\|_{L^p(\Omega)}$.

The dual space of $L^p(\Omega)$ can be identified with $L^q(\Omega)$, where q is chosen such that $\frac{1}{p}+\frac{1}{q}=1 \Leftrightarrow p=\frac{q}{q-1}$. For $L^1(\Omega)$ we obtain the corresponding dual space $L^\infty(\Omega)$ from the formula above, which is defined as

$$L^\infty(\Omega) = \left\{u : \Omega \to \bar{\mathbb{R}} : \text{u measurable}, ess\,sup\,|u| < \infty\right\},$$

where the essential supremum of u is the supremum except for a set $\mathcal{N}$ of measure zero

$$ess\,sup\,|u| = \inf_{\mathcal{N}} \sup_{x\in\Omega\setminus\mathcal{N}} \left|u(x)\right|.$$

Remark 1.
Let $1 \le p < +\infty$. Then the space $L^p(\Omega)$ is a Banach space with norm $\|u\|_p$. Moreover, $L^\infty(\Omega)$ is a Banach space with norm $\|u\|_\infty := ess\,sup\,|u|$.
In the special case $p=2$ the Lebesgue space $L^2(\Omega)$ is a Hilbert space equipped with the scalar product

$$\langle u, v\rangle_{L^2(\Omega)} := \int_\Omega u(x)v(x)\,\mathrm{d}x.$$

Lemma 2.1.6. Coarea Formula
Let $u : \Omega \to \mathbb{R}$ be a Lipschitz continuous function. Then, for $g \in L^1(\Omega)$ we have

$$\int_\Omega g(x)\left|\nabla u(x)\right|\mathrm{d}x = \int_{-\infty}^{\infty}\left(\int_{u^{-1}(t)} g(x)\,\mathrm{d}H_{n-1}(x)\right)\mathrm{d}t,$$

where H_{n-1} represents the $(n-1)$ dimensional Hausdorff measure and

$$u^{-1}(t) := \{x \in \Omega : u(x) = t\}.$$

This implies

$$\int_\Omega \left|\nabla u(x)\right|\mathrm{d}x = \int_{-\infty}^{\infty} H_{n-1}(u^{-1}(t))\,\mathrm{d}t.$$

Proof. The proof can be found in [48]. □

2.1.2. Sobolev Spaces

Definition 2.1.7. Sobolev Space
Let u be a measurable function and $1 \leq k, p < \infty$. The Sobolev space $W^{k,p}(\Omega)$ contains all functions whose weak derivatives up to order k lie in $L^p(\Omega)$.

$$W^{k,p}(\Omega) = \left\{u : L^p(\Omega) : \partial^\alpha u \in L^p(\Omega), \forall\, |\alpha| \leq k\right\}$$

The norm in $W^{k,p}$ is given by the p-th root of the sum of the derivative norms

$$\|u\|_{W^{k,p}(\Omega)} := \left(\sum_{|\alpha|\leq k} \|\partial^\alpha u\|^p_{L^p(\Omega)}\right)^{\frac{1}{p}},$$

with short notation $\|u\|_{k,p}$. The most common space in the subsequent analysis is $W^{1,p}(\Omega)$,

$$W^{1,p}(\Omega) = \left\{u : L^p(\Omega) : \partial_{x_i} u \in L^p(\Omega), i = 1, \ldots, n\right\}$$
$$\|u\|_{1,p} = \left(\|u\|^p_p + \sum_{i=1}^{d} \left\|\partial_{x_i} u\right\|^p_p\right)^{\frac{1}{p}}.$$

Remark 2.
For $k = 1$ and $p = 2$ we obtain the Hilbert space $H^1(\Omega) := W^{1,2}(\Omega)$, whose norm is induced by the scalar product

$$\langle u, v\rangle_1 := \int_\Omega \left(uv + \sum_{j=1}^{d} \partial_{x_j} u \partial_{x_j} v\right) dx.$$

The Hilbert space H^1 is very common in the analysis of linear equations. In an analogue way, Hilbert spaces of higher order $H^k(\Omega) := W^{k,2}(\Omega)$ are defined.

2.1.3. BV Space

In this section the space of functions with bounded variation is introduced. These functions are used in many results of this thesis. In the context of image analysis,

this space has certain desirable properties on the one hand, and on the other hand naturally contains what an observer would call an *image*. Starting with the space $L^1(\Omega)$, it can be shown that this space contains images but also the noise, which is unwanted. Considering the smaller space $W^{1,1}(\Omega)$ instead, we see that the space no longer contains noise, but unfortunately $W^{1,1}(\Omega)$ does not allow discontinuities in its elements. But these discontinuities are required for image analysis, because objects are usually formed by an intensity jump against the background, which coincides with a discontinuity in the image function u. The proper space for such u is the space of functions with bounded variation, which is somewhere between $L^1(\Omega)$ and $W^{1,1}(\Omega)$. We will only give a short summary of these functions and refer the reader to [21] for further details.
For this section we assume $\Omega \in \mathbb{R}^N$ to be open and sufficiently regular and begin with the definition of the *total variation*:

Definition 2.1.8. Total Variation
Let Ω be a domain and $u \in L^1(\Omega)$. The total variation of u is defined as

$$TV(u) = |u|_{BV} := \sup_{\phi \in C_0^\infty(\Omega;\mathbb{R}^N), \|\phi\|_\infty \leq 1} \int_\Omega u \nabla \cdot \phi dx.$$

Definition 2.1.9. BV Space
Let Ω be a domain. We define the space of functions with bounded total variation as follows:

$$BV(\Omega) := \left\{ u \in L^1(\Omega) : |u|_{BV} < \infty \right\}$$

The total variation defines a semi-norm on $BV(\Omega)$. By setting $u = c \neq 0$ we can easily prove that $|u|_{BV}$ is only a semi-norm.
Finally the norm on $BV(\Omega)$ is given by the sum of L^1 and BV-semi-norm

$$\|u\|_{BV} := |u|_{BV} + \|u\|_{L^1}$$

Remark 3. If the considered function u is sufficiently smooth, $u \in W^{1,1}(\Omega)$, then the total variation simplifies to

$$TV(u) = \int_\Omega |\nabla u| \, \mathrm{d}x.$$

As we will later see in Theorem 2.1.22, the space $W^{1,1}(\Omega)$ is a subspace of $BV(\Omega)$. This is why, in an informal setting, we often write $\int_\Omega |\nabla u| \, \mathrm{d}x$, neglecting that $BV(\Omega)$ contains a broader set of functions.

For the following theorem of Banach-Alaoglu (see 2.2.16), required for the variational analysis, we need a space of which $BV(\Omega)$ is the dual space.

Lemma 2.1.10. BV is a Dual Space
The space $BV(\Omega)$ can be identified with the dual space of $\mathcal{Y}$, where

$$\mathcal{Y} := \left\{ (c, \nabla \cdot \varphi) : c \in \mathbb{R}, \varphi \in C(\bar{\Omega}; \mathbb{R}^d), \varphi|_{\partial\Omega} \cdot n = 0 \right\}$$

Proof. A proof for $BV_0(\Omega)$ (space of functions with bounded variation and mean value 0) can be found in [21]. This proof can be modified to functions with mean value different from 0. □

For the later analysis we need a few essential properties

Lemma 2.1.11. Properties of the Total Variation:

- $|\cdot|_{BV}$ is convex.
- $|\cdot|_{BV}$ is lower semicontinuous with respect to the weak topology of L^1.
- Any uniformly bounded sequence in BV is relatively compact in L^1.

Proof. Can be found in [2]. □

2.1.4. Bochner Spaces

Definition 2.1.12. Measurable space
A set $\mathcal{X}$ with a distinguished σ-algebra of measurable subsets is called *measurable space*. One example is the $\mathbb{R}^n$ with the Lebesgue σ-algebra.

Definition 2.1.13. Bochner space
Let $\mathcal{X}$ denote a measurable Banach space with norm $\|\cdot\|_{\mathcal{X}}$. The space $L^p(0, T, \mathcal{X})$ is

called *Bochner space* and consists of all $\mathcal{X}$-measurable functions $u : [0,T] \to \mathcal{X}$ with norm

$$\|u\|_{L^p(0,T;\mathcal{X})} := \left(\int_0^T \left\|u(t)\right\|_{\mathcal{X}}^p \mathrm{d}t \right)^{\frac{1}{p}} < \infty \quad \text{for } 1 \leq p < \infty$$

and

$$\|u\|_{L^\infty(0,T;\mathcal{X})} := ess\, sup_{0\leq t\leq T} \left\|u(t)\right\| < \infty$$

In Chapter 5 we need the concept of Bochner spaces. Those spaces can be defined in a more general context, but for our applications we restrict to $L^p(0,T,\mathcal{X})$. This space consists of all functions u such that $u(\cdot,t) \in \mathcal{X}$ for every $t \in [0,T]$. In addition to this property, the time integral of the p-th power of u must exist.

Theorem 2.1.14. Dual of a Bochner Space
Let $\mathcal{X}$ be a Banach space and $\mathcal{X}^*$ be a separable measurable space. Then for $1 < p < \infty$, $\frac{1}{p} + \frac{1}{q} = 1$ we have

$$L^p(0,T;\mathcal{X})^* \cong L^q(0,T;\mathcal{X}^*).$$

For $p = 1$ the dual space can be identified with $L^\infty(0,T;\mathcal{X}^*)$, hence

$$L^1(0,T;\mathcal{X})^* \cong L^\infty(0,T;\mathcal{X}^*)$$

Proof. Can be found in [24]. □

Corollary 2.1.15.
Let $\mathcal{X}$ be a reflexive Banach space, then $L^p(0,T;\mathcal{X})$ is a reflexive space.

In the context of Bochner spaces and variational analysis we face a huge problem. Consider the spaces $L^p(0,T;\mathcal{X})$ and $L^q(0,T;\mathcal{Y})$ with $p < q$. As we will see later (Section 2.1.5), there exists the natural compact embedding $L^p \subset\subset L^q$. Let us moreover assume that $\mathcal{X} \subset\subset \mathcal{Y}$. Unfortunately, we are not able to deduce the compact embedding $L^p(0,T;\mathcal{X}) \subset\subset L^q(0,T;\mathcal{Y})$. A way out yields the following lemma:

Lemma 2.1.16. Aubin-Lions
Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be Banach spaces with $\mathcal{X} \subset\subset \mathcal{Y}$ and $\mathcal{Y} \hookrightarrow \mathcal{Z}$. Let u_n be a sequence of

bounded functions in $L^p(0,T;\mathcal{X})$ and $\partial_t u_n$ be bounded in $L^q(0,T;\mathcal{Z})$ (for $q=1$ and $1 \leq p < \infty$ or $q > 1$ and $1 \leq p \leq \infty$).
Then u_n is relatively compact in $L^p(0,T;\mathcal{Y})$.

Proof. See [9, 46, 66] □

In other words: If we are able to prove time regularity for functions $u \in L^p(0,T;\mathcal{X})$, we do get a compact embedding.

2.1.5. Embeddings

Definition 2.1.17. Embedding
Let $\mathcal{X} \subset \mathcal{Y}$ be Banach spaces with respective norms $\|\cdot\|_{\mathcal{X}}$ and $\|\cdot\|_{\mathcal{Y}}$. We say $\mathcal{X}$ is continuously embedded in $\mathcal{Y}$, written $\mathcal{X} \hookrightarrow \mathcal{Y}$, if the inclusion map $\mathcal{X} \to \mathcal{Y} : x \mapsto x$ is continuous, i.e. $\|x\|_{\mathcal{Y}} \leq C \|x\|_{\mathcal{X}}, \forall x \in \mathcal{X}$ for some constant $C > 0$.
We furthermore speak of a compact embedding, written $\mathcal{X} \subset\subset \mathcal{Y}$, if the inclusion is a compact operator, i.e. any bounded sequence has a subsequence that is Cauchy in the norm $\|\cdot\|_{\mathcal{Y}}$.

There is a natural embedding $L^p \hookrightarrow L^q$ if $p > q$, which is proven by the next Lemma.

Lemma 2.1.18. Lebesgue Embedding
Let $1 \leq p < q \leq \infty$ then $L^p(\Omega) \hookrightarrow L^q(\Omega)$.

Proof. Let $u \in L^p(\Omega)$ and $\frac{1}{p} = \frac{1}{q} + \frac{1}{r}$, then using Hölders inequality (Lemma A.1.9)

$$\|f\|_p = \|1 \cdot f\|_p \leq \|1\|_r \|f\|_q = |\Omega| \|f\|_q .$$

□

Theorem 2.1.19. Sobolev Embedding Theorem
Let Ω be a domain in $\mathbb{R}^n$ and, for $1 \leq k \leq n$, let Ω_k be the intersection of Ω with a plane of dimension k in $\mathbb{R}^n$. (If $k = n$, then $\Omega_k = \Omega$). Let $j \geq 0$ and $m \geq 1$ be integers and let $1 \leq p < \infty$.
Case 1: If either $mp > n$ or $m = n$, $p = 1$ and $1 \leq k \leq n$, then

$$W^{j+m,p}(\Omega) \hookrightarrow W^{j,q}(\Omega_k) \quad \text{for } p \leq q \leq \infty,$$

and, in particular,

$$W^{m,p}(\Omega) \hookrightarrow L^q(\Omega) \quad \text{for } p \leq q \leq \infty$$

Case 2: If $1 \leq k \leq n$ and $mp = n$, then

$$W^{j+m,p}(\Omega) \hookrightarrow W^{j,q}(\Omega_k) \quad \text{for } p \leq q < \infty,$$

and, in particular,

$$W^{m,p}(\Omega) \hookrightarrow L^q(\Omega) \quad \text{for } p \leq q < \infty$$

Case 3: If $mp < n$, and either $n - mp < k \leq n$ or $p = 1$ and $n - m \leq k \leq n$, then

$$W^{j+m,p}(\Omega) \hookrightarrow W^{j,q}(\Omega_k) \quad \text{for } p \leq q \leq p^* = \frac{kp}{n - mp}.$$

In particular,

$$W^{m,p}(\Omega) \hookrightarrow L^q(\Omega) \quad \text{for } p \leq q \leq p^* = \frac{np}{n - mp}.$$

Proof. A proof can be found in [3]. □

Theorem 2.1.20. Rellich-Kondrachov Compactness Theorem
Consider a bounded open subset $\Omega \subset \mathbb{R}^n$ with $\partial\Omega$ C^1. Suppose $1 \leq p < n$. Then

$$W^{1,p}(\Omega) \subset\subset L^q(\Omega)$$

for each $1 \leq q < p^*$, where p^* is the Sobolev conjugate.

Proof. The proof can be found in [34]. □

This theorem can be applied as follows. Consider for example a set $\Omega \subset \mathbb{R}^3$ and $W^{1,2}(\Omega)$. Then we get for the Sobolev conjugate $p^* = 6$. Now the theorem yields a compact embedding of $W^{1,2}(\Omega)$ into $L^q(\Omega)$ for $q = 1, \ldots, 5$.

The former result can be generalized to arbitrary $W^{k,p}$ by the following result:

Theorem 2.1.21. General Compact Sobolev Embeddings
Suppose a bounded open subset $\Omega \subset \mathbb{R}^n$ with $\partial\Omega \in C^1, p \in [1, \infty), k, l \in \mathbb{N}$ with $l \geq 1$ and $q < \frac{np}{n-pl}$. Then we obtain a compact embedding

$$W^{k,p}(\Omega) \subset\subset W^{k-l,q}(\Omega)$$

Proof. The proof can be found in [3]. □

Theorem 2.1.22. Embedding for BV
Suppose a bounded open subset $\Omega \subset \mathbb{R}^n$ and $1 \leq p \leq \frac{n}{n-1}$. Then there exists a continuous embedding $BV(\Omega) \hookrightarrow L^p(\Omega)$. For $p < \frac{n}{n-1}$ the embedding is compact, thus $BV(\Omega) \subset\subset L^p(\Omega)$.

Proof. Can be found in [6]. □

Lemma 2.1.23. Embedding for Bochner Spaces

1. Consider the Bochner spaces $L^p(0,T;\mathcal{X})$ and $L^p(0,T;\mathcal{Y})$ with $\mathcal{X} \hookrightarrow \mathcal{Y}$. Then
$$L^p(0,T;\mathcal{X}) \hookrightarrow L^p(0,T;\mathcal{Y})$$
2. Let $1 \leq s < t \leq \infty$ and consider the Bochner spaces $L^s(0,T;\mathcal{X})$ and $L^t(0,T;\mathcal{X})$. Then we have the continuous embedding
$$L^t(0,T;\mathcal{X}) \hookrightarrow L^s(0,T;\mathcal{X}).$$

Proof. We start with 1: Let $u \in L^p(0,T;\mathcal{X})$. Due to the continuous embedding $\mathcal{X} \hookrightarrow \mathcal{Y}$ we have
$$\left\|u(t)\right\|_{\mathcal{Y}} \leq C_t \left\|u(t)\right\|_{\mathcal{X}} \quad \forall t \in [0,T]\,.$$
The constant C_t might be different for every $t \in [0,T]$, but it is not a function of t and we define
$$C_m := \max_{t\in[0,T]} C_t.$$
Hence,
$$\left\|u(t)\right\|_{\mathcal{Y}} \leq C_m \left\|u(t)\right\|_{\mathcal{X}} \quad \forall t \in [0,T]\,.$$
This leads to the estimate
$$\begin{aligned}\|u\|^p_{L^p(0,T;\mathcal{Y})} &= \int_0^T \left\|u(t)\right\|^p_{\mathcal{Y}} \,\mathrm{d}t \leq \int_0^T (C_m \left\|u(t)\right\|_{\mathcal{X}})^p \,\mathrm{d}t \\ &= C_m^p \int_0^T \left\|u(t)\right\|^p_{\mathcal{X}} \,\mathrm{d}t = C_m^p \|u\|^p_{L^p(0,T;\mathcal{X})}\end{aligned}$$

Taking the p–th root on both sides we end up with the required

$$\|u\|^p_{L^p(0,T;\mathcal{Y})} \leq C \|u\|^p_{L^p(0,T;\mathcal{X})} .$$

Part two follows from the embedding theory for Lebesgue spaces (see Lemma 2.1.18). □

2.1.6. Convergence Results

Lemma 2.1.24. Strong and Weak Convergence
Let $\mathcal{X}$ be a Banach space and $u_k \to u$ in $L^p(\mathcal{X})$, then $u_k \rightharpoonup u$ in $L^p(\mathcal{X})$.

Proof. Let $v \in \mathcal{X}^*$ be an arbitrary element from the dual space, then we write down for the weak convergence

$$\big|\langle v, u_k\rangle - \langle v, u\rangle\big| = \big|\langle v, u_k - u\rangle\big| = \left|\int_{\mathcal{X}} v(u_k - u)dx\right| \underbrace{\leq}_{\text{Hölder}} \|v\|_{L^q} \underbrace{\|u - u_k\|_{L^p}}_{\to 0}$$

□

Note that weak convergence does not imply strong convergence in general!

Lemma 2.1.25. $W^{1,p}$ Convergence Implies L^p Convergence
Let $\mathcal{X}$ be a Banach space and $u_k \to u$ in $W^{1,p}(\Omega)$, then $u_k \to u$ in $L^p(\Omega)$ and $\partial_{x_i} u_k \to v_i$ in $L^p(\Omega)$ where v_i is the i-th distributional derivative of u.

Proof. The proof follows directly from the definition of the $W^{1,p}$–Norm, which is the sum of the L^p norm of u_k and $\partial_{x_i} u_k$. Convergence of the sum implies convergence of all elements, since they have positive value. □

Corollary 2.1.26. Weak Compactness for L^p
Let $1 < p < \infty$ and u_k a bounded sequence in L^p, then there exists a subsequence u_{k_n} such that $u_{k_n} \rightharpoonup u$ in L^p weakly.

Proof. The proof is a corollary of Theorem 2.2.16. □

Remark 4. Weak-* Convergence in $L^\infty(\Omega)$
Due to the fact that $L^\infty(\Omega)$ is the dual space to $L^1(\Omega)$ ($L^1(\Omega)$ is not the dual space of $L^\infty(\Omega)$) we can define weak-* convergence in $L^\infty(\Omega)$. Let therefore be $\Omega \subset \mathbb{R}^n$ an open set and $u_n \in L^\infty(\Omega)$. We say that $u_n \in L^\infty(\Omega)$ converges weak-* to $u \in L^\infty(\Omega)$ if

$$\int_\Omega u_n v \to \int_\Omega uv \quad \forall v \in L^1(\Omega).$$

Thus for bounded subsequences in $L^\infty(\Omega)$ we have weak-* convergence.

2.2. Variational Calculus

In this section we will give a coarse introduction to the calculus of variations. We begin with basic definitions of operators and functionals and continue with the concept of variational derivatives for energies. Afterwards we introduce convexity and lower semi-continuity, which then lead to the fundamental theorem of optimization. Then we recall the concept of Fenchel duality, which the connects primal and dual variational problem. Finally, we present examples for a set of concrete variational problems and deduce solutions.

Definition 2.2.1. Operator and Functional
Let $\mathcal{X}, \mathcal{Y}$ denote two Banach spaces with topology τ_1, τ_2. A mapping $J : \mathcal{X} \to \mathcal{Y}$ between these spaces is called an *operator*. In case $\mathcal{Y}$ is a scalar field, J is called a *functional*.

In our observations we usually consider $\mathcal{Y} = \bar{\mathbb{R}}$ and define a *proper* functional:

Definition 2.2.2. Proper Functional
Let $\mathcal{X}$ denote a Banach space. A functional $J : \mathcal{X} \to \bar{\mathbb{R}}$ is called *proper* if $J(u) \neq -\infty$ for all $u \in \mathcal{X}$ and if there exists at least one $u \in \mathcal{X}$ with $J(u) \neq \infty$. We furthermore define the *effective domain* of J as $\{u \in \mathcal{X} : J(u) < \infty\}$.

2.2.1. Derivatives

We start by defining, in analogy to the classical difference quotient, a difference quotient for functionals, the so-called *directional derivative*.

Definition 2.2.3. Directional Derivative
Let $\mathcal{X}$ denote a Banach space, and $J : \mathcal{X} \to \bar{\mathbb{R}}$ be a proper functional. For $u \in \mathcal{X}$ we define the *directional derivative* in u in the direction v as

$$d_v J(u) := \lim_{t \to 0} \frac{J(u + tv) - J(u)}{t}.$$

Definition 2.2.4. Gâteaux Derivative
Let $\mathcal{X}$ be a Banach space, and $J : \mathcal{X} \to \bar{\mathbb{R}}$ be a proper functional. The set of directional derivatives

$$dJ(u) := \left\{ d_v J(u) : d_v J(u) < \infty \text{ for } v \in \mathcal{X} \right\}$$

is called the *Gâteaux derivative* in $u \in \mathcal{X}$. If $dJ(u) \neq \left\{\emptyset\right\}$, then J is called *Gâteaux differentiable* in u.

Definition 2.2.5. Fréchet Derivative
Let $\mathcal{X}$ be a Banach space, and $J : \mathcal{X} \to \bar{\mathbb{R}}$ be a proper functional. Let furthermore $dJ(u)$ exist for all $v \in \mathcal{X}$. If there exists a continuous linear functional $F : \mathcal{X} \to \bar{\mathbb{R}}$ such that

$$Fv = d_v J(u) \quad \forall v \in \mathcal{X},$$

and

$$\lim_{\|v\|_{\mathcal{X}} \to 0} \frac{\left| J(u + v) - J(u) - Fv \right|}{\|v\|_{\mathcal{X}}} = 0 \quad \forall v \in \mathcal{X},$$

then J is called Fréchet differentiable in $u \in \mathcal{X}$ and F is called the *Fréchet derivative*.

Since we are often interested in the optimality condition of convex functionals, which are not Frechét differentiable, we want to further generalize this definition. Therefore we extend the concept of differentiation to include the points where no Frechét derivative exists.

Definition 2.2.6. Subdifferential
Let $\mathcal{X}$ be a Banach space, and $J : \mathcal{X} \to \bar{\mathbb{R}}$ be a proper and convex functional. For

$u \in \mathcal{X}$ we call J subdifferentiable if there exists $p \in \mathcal{X}^*$ such that

$$J(v) - J(u) - \{p, v - u\} \geq 0 \quad \forall v \in \mathcal{X}.$$

Then p is called a *subgradient* of J in u and

$$\partial J(u) := \left\{p \in \mathcal{X}^* : J(v) - J(u) - \{p, v - u\} \geq 0 \quad \forall v \in \mathcal{X}\right\}$$

is called the *subdifferential* of J in u.

2.2.2. Convexity

The concept of convexity in variational calculus is very useful. Finding a minimizer of a strictly convex model always implies its uniqueness. This is also important when numerically minimizing an energy since convexity prevents getting trapped in local minima. For further details about convex optimization we refer the reader to [57, 58, 18, 31]

Definition 2.2.7. Convex Set
Let $\mathcal{X}$ be a Banach space. A subset $\Omega \in \mathcal{X}$ is called *convex* if for all $u, v \in \mathcal{X}$

$$\alpha u + (1 - \alpha)v \in \Omega, \quad \forall \alpha \in [0, 1]$$

Definition 2.2.8. Convex Functional
Let $\mathcal{X}$ be a Banach space, $\Omega \subset \mathcal{X}$ a convex subset and $J : \Omega \to \mathbb{R} \cup \{\infty\}$ a functional. J is said to be *convex* if the inequality

$$J(\alpha u + (1 - \alpha)v) \leq \alpha J(u) + (1 - \alpha)J(v)$$

is satisfied for all $u, v \in \Omega$ and $\alpha \in [0, 1]$. We furthermore call J strictly convex if the equality above is satisfied with $<$ strict and $\alpha \in (0, 1)$.

Lemma 2.2.9. Examples for Convex Functionals

1. Any norm $J(u) = \|u\|$ is a convex function.
2. Any affine norm $J(u) = \|Ku - f\|$ is a convex function for an arbitrary operator K.

3. Let $p \geq 1$, then $J(u) = \|u\|^p$ is convex.

Proof. Can be found in [57]. □

It follows in particular that the later discussed data-fidelity terms $\|Ku - f\|_2^2$ are convex.

Lemma 2.2.10. Convexity of the Total Variation
The Total Variation (see Definition 2.1.8) is convex.

Proof.

$$\begin{aligned} TV(\alpha u + (1-\alpha v)) &= \sup_{\|p\|_{L^\infty}(\Omega)\leq 1} \int_\Omega (\alpha u + (1-\alpha)v)\nabla \cdot p \,\mathrm{d}x \\ &= \sup_{\|p\|_{L^\infty}(\Omega)\leq 1} \int_\Omega (\alpha u \nabla \cdot p + (1-\alpha)v\nabla \cdot p) \,\mathrm{d}x \\ &\leq \alpha \sup_{\|p\|_{L^\infty}(\Omega)\leq 1} \int_\Omega u\nabla \cdot p \,\mathrm{d}x + (1-\alpha) \sup_{\|p\|_{L^\infty}(\Omega)\leq 1} \int_\Omega v\nabla \cdot p \,\mathrm{d}x \\ &= \alpha TV(u) + (1 - \alpha TV(v))) \end{aligned}$$

□

2.2.3. Lower Semi-Continuity

Definition 2.2.11. Lower Semi-Continuity
Let $(\mathcal{X}, \tau)$ be a measurable topological space and $J : (\mathcal{X}, \tau) \to \bar{\mathbb{R}}$ a functional on $\mathcal{X}$. J is said to be *lower semi-continuous* with respect to τ in u if for every sequence $u_k \xrightarrow{\tau} u$

$$J(u) \leq \liminf_k J(u_k)$$

Corollary 2.2.12. Lower Semi-Continuity for Sums
Let $(\mathcal{X}, \tau)$ be a topological space and $J, \hat{J} : (\mathcal{X}, \tau) \to \mathbb{R} \cup \{\infty\}$ lower semi-continuous functionals on $\mathcal{X}$ with respect to τ. Then $J + \hat{J}$ is also lower semi-continuous with respect to τ.

The following Lemma will give us an easy criterion for weak lower semi-continuity on reflexive Banach spaces (for example L^p spaces).

Lemma 2.2.13.
Let $J : \mathcal{X} \to \mathbb{R}$ be a convex functional on a reflexive Banach space $\mathcal{X}$, then J is weakly lower semicontinuous.

Proof. The proof is done by contradiction and we assume that J is not weakly lower semi-continuous, which implies that there exists a sequence $u_k \rightharpoonup u$ with $J(u) > \lim_k J(u_k)$. From convexity of J follows directly the convexity of the epigraph, i.e. the set

$$epi(J) := \left\{(u, a) \in \mathcal{X} \times \mathbb{R} : a \geq J(u)\right\}$$

is convex. Therefore with $J(u) > \lim_k J(u_k)$ there exists $b \in \mathbb{R}$ with $J(u) > b > \lim_k J(u_k)$ and consequently $(u, b) \notin epi(J)$. With Hahn-Banach's theorem (see [61]) the convex set $epi(J)$ can be split by a linear function, i.e. there exist $c, d \in \mathbb{R}$ and $p \in \mathcal{X}^*$ with

$$cb + \langle p, u\rangle + d \leq 0$$

and

$$ca + \langle p, v\rangle + d \geq 0$$

for all $(v, a), (u, b) \in epi(J)$. Choosing $a = J(u)$ and combining both inequalities gives us

$$cb \leq cJ(u),$$

it follows that $c > 0$ and we divide by c. It follows that for $a = J(u_k), v = u_k$

$$b + \frac{1}{c}\langle p, u\rangle \leq J(u_k) + \frac{1}{c}\langle p, u_k\rangle .$$

Due to the weak convergence we have $\langle p, u\rangle = \lim \langle p, u_k\rangle$, thus $b \leq \lim_k J(u_k)$ which is a contradiction. □

Lemma 2.2.14. The total variation is weak$-^*$ lower semicontinuous on $BV(\Omega)$.

Proof. Let $u_n \rightharpoonup^* u$ and let $\varphi_k \in C_0^\infty(\Omega, \mathbb{R}^d)$ with $\|\varphi_k\|_\infty \leq 1$ such that

$$|u|_{BV} = \lim_k \int_\Omega u \nabla \cdot \varphi_k \, \mathrm{d}x.$$

Then we have due to the weak$-^*$ convergence

$$\begin{aligned}\int_\Omega u\nabla\cdot\varphi_k\,\mathrm{d}x &= \lim_n \int_\Omega u_n\nabla\cdot\varphi_k\,\mathrm{d}x\\ &= \liminf_n \int_\Omega u_n\nabla\cdot\varphi_k\,\mathrm{d}x\\ &\leq \liminf_n \sup_{\varphi\in C_0^\infty(\Omega;\mathbb{R}^d),\|\varphi\|_\infty\leq 1}\int_\Omega u_n\nabla\cdot\varphi\,\mathrm{d}x\\ &= \liminf_n |u_n|_{BV}\,.\end{aligned}$$

No taking the limit over k yields

$$|u|_{BV}\leq \liminf_n |u_n|_{BV}\,.$$

□

2.2.4. Existence of Minimizers

Definition 2.2.15. Coercivity
Let $(\mathcal{X},\tau)$ be a topological space and $J:\mathcal{X}\to\mathbb{R}\cup\{\infty\}$ a functional on $\mathcal{X}$. We call J *coercive* if it has compact sub-level sets. This means there exists an $\alpha\in\mathbb{R}$ such that the set

$$S(\alpha):=\big\{u\in\mathcal{X}|\quad J(u)\leq\alpha\big\}$$

is not empty and compact in τ.

Theorem 2.2.16. Theorem of Banach-Alaoglu
Let $\mathcal{X}$ be the dual of a Banach space and $C>0$. Then the set

$$\big\{u\in\mathcal{X}:\|u\|_{\mathcal{X}}\leq C\big\}$$

is compact in the weak-* topology.

Proof. A proof can be found in [61] p. 66-68, Chapter 3, Theorem 3.15. □

Let us mention here that this theorem is fundamental in the later analysis. Due to the L^p and Sobolev theory the spaces in our analysis are usually dual spaces and proving boundedness of a set is a property that can be shown relatively easy.

The definitions of lower semi-continuity and coercivity lead us directly to a fundamental result from the optimization theory. We refer to [8] for further details.

Theorem 2.2.17. Fundamental Theorem of Optimization
Let $(\mathcal{X}, \tau)$ be a metric space and $J : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ a functional on $\mathcal{X}$. Moreover let J be lower semi-continuous (2.2.11) and coercive (2.2.15).
Then there exists a global minimum $\bar{u} \in \mathcal{X}$, i.e.

$$J(\bar{u}) = \inf_{u \in \mathcal{X}} J(u)$$

Proof. Let u_k be a minimizing sequence such that $\lim_{k\to\infty} J(u_k) = \inf_{u\in\mathcal{X}} J(u)$. Let k_0 be large enough such that $u_{k_0} \in S(\alpha)$ (2.2.15). Then the set $\{u_k : k \geq k_0\}$ is contained in a compact set and it has a convergent subsequence $\hat{u}_k$ with $\lim_{k\to\infty} J(\hat{u}_k) = J(\hat{u})$. Then we have

$$\inf_{u\in\mathcal{X}} J(u) \leq J(\hat{u}) \underbrace{\leq}_{2.2.11} \lim_{k\to\infty} J(\hat{u}_k) = \inf_{u\in\mathcal{X}} J(u),$$

and it follows that $\hat{u}$ is a global minimum of J. □

2.2.5. Legendre-Fenchel Duality

For the analysis and later implementation of variational problems the concept of duality is essential. For the scope of this thesis we are only able to give a short overview of some basic definitions and properties. For a complete discussion of this subject we refer the reader to [57, 58, 18].

Definition 2.2.18.
Let $\mathcal{X}$ be a Banach space with dual space $\mathcal{X}^*$. The *convex conjugate* $J^* : \mathcal{X}^* \to \bar{\mathbb{R}}$ of a proper functional $J : \mathcal{X} \to \bar{\mathbb{R}}$ is defined by

$$J^*(p) := \sup_{u\in\mathcal{X}} \left\{ \langle p, u \rangle_{\mathcal{X}} - J(u) \right\} \quad \text{for } p \in \mathcal{X}^*.$$

The *biconjugate* $J^{**} : \mathcal{X}^{**} \to \bar{\mathbb{R}}$ (in case of reflexive spaces we have $\mathcal{X} = \mathcal{X}^{**}$) of J is defined as

$$J^{**}(u) := \sup_{p\in\mathcal{X}^*} \left\{ \langle u, p \rangle_{\mathcal{X}^*} - J^*(p) \right\} \quad \text{for } u \in \mathcal{X}.$$

The convex conjugate is also known as Legendre-Fenchel transform, named after Adrien-Marie Legendre and Werner Fenchel.

We have special interest in the biconjugate J^{**}. Graphically speaking, the biconjugate is the largest convex functional below J. The following theorem makes this statement more precise.

Theorem 2.2.19. Fenchel-Morau-Rockafellar
Let $\mathcal{X}$ be a Banach space and $J : \mathcal{X} \to \bar{\mathbb{R}}$. Then we have

1. $J^{**} \leq J$,

2. $J^{**} = J$ iff J is convex and lower-semicontinuous.

Proof. Can be found in [57]. □

The following rules are helpful when calculating the convex conjugate:

Remark 5. Duality Correspondence
Let $J : \mathbb{R}^n \to \bar{\mathbb{R}}$ be proper, lower-semicontinuous and convex. Then we have the following correspondence:

$$\begin{aligned}(J(\cdot + a))^* &= J^*(\cdot) - \langle \cdot, a \rangle, \\ (J(\cdot) + a)^* &= J^*(\cdot) - a, \\ (\lambda J(\cdot))^* &= \lambda J^*(\frac{\cdot}{\lambda}), \quad \lambda > 0\end{aligned}$$

Proof. Follows from the definition of the convex conjugate. □

The following theorem is very important in the context of variational minimization. The application of Fenchel's duality theorem connects the primal variational problem to the corresponding dual problem, which is in many cases easier to solve. Moreover, the Legendre-Fenchel transform is the basis for many efficient numerical methods for variational problems (see Section 2.3.1).

Theorem 2.2.20. Fenchel's Duality Theorem
Let $J_1 : \mathcal{X} \to \bar{\mathbb{R}}$ and $J_2 : \mathcal{Y} \to \bar{\mathbb{R}}$ be proper, lower semi-continuous and convex functionals on Banach spaces $\mathcal{X}$ and $\mathcal{Y}$, such that $domJ_1 \cap domJ_2 \neq \emptyset$. Furthermore,

let $K : \mathcal{X} \to \mathcal{Y}$ be a bounded linear operator.
Then we have the following equality:

$$\inf_{u \in \mathcal{X}} \left\{ J_1(Ku) + J_2(u) \right\} = \sup_{p \in \mathcal{X}^*} \left\{ J_1^*(p) + J_2^*(K^* p) \right\}. \tag{2.1}$$

Proof. A proof can be found in [31]. □

2.2.6. Duality in Image Processing

The aim of this section is to connect the concept of duality from the previous section with the variational models for image processing used in this thesis. As we will see later (cf. Section 2.3.1), duality offers an efficient way of minimizing variational models. This section has been inspired by the work of Chambolle and Pock [26], but we want to give a more detailed overview here. As a start, let us consider two finite dimensional vector spaces $\mathcal{X}$ and $\mathcal{Y}$ equipped with a scalar product $\langle \cdot, \cdot \rangle$ and a norm $\|\cdot\|_2$. We furthermore consider a continuous linear operator $K : \mathcal{X} \to \mathcal{Y}$. Now, the general class of problems in this thesis can be written as

$$\min_{x \in \mathcal{X}} G(x) + F(Kx), \tag{2.2}$$

with $F, G : \mathcal{X} \to \mathbb{R}$ are proper, convex and lower semi-continuous functionals. In the following, we denote Equation (2.2) as the *primal problem*.

Example 2.2.21. Rudin-Osher-Fatemi Model
Setting $G(u) := \frac{1}{2} \|u - f\|_2^2$ and $F(Ku) := \alpha \|\nabla u\|_1$ we obtain

$$J(u) = \frac{1}{2} \|u - f\|_2^2 + \alpha \|\nabla u\|_1 .$$

In this context, the operator K is simply the gradient $K := \nabla$. This variational model is known as *Rudin-Osher-Fatemi (ROF)* model for image denoising (cf. [60]).

The primal problem is often hard to minimize or yields very slow algorithms. Instead of the primal problem, we can equivalently solve the so-called *primal-dual problem*.

Lemma 2.2.22. Equivalence of Primal and Primal-Dual Problem

Let $\hat{x}_p$ be a minimizer of the primal problem

$$\min_{x \in \mathcal{X}} G(x) + F(Kx),$$

where G and F fulfill the assumptions above. Let furthermore the tuple $(\hat{x}_{pd}, \hat{y}_{pd})$ be a minimizer of the saddle-point problem

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

Then we have $\hat{x}_p = \hat{x}_{pd}$.

Proof. Let us start with the primal problem. Since F was assumed to be a convex, semi-continuous functional we can replace $F(Kx)$ by its biconjugate $F^{**}(Kx)$, and get for the primal problem

$$\min_{x \in \mathcal{X}} G(x) + F^{**}(Kx). \tag{2.3}$$

Now, using the definition of the convex conjugate, we have

$$F^{**}(Kx) = \max_{y \in \mathcal{Y}} \langle Kx, y \rangle - F^*(y) = \max_{y \in \mathcal{Y}} \langle x, K^* y \rangle - F^*(y).$$

Inserting this into Equation (2.3) we get

$$\min_{x \in \mathcal{X}} G(x) + \left\{ \max_{y \in \mathcal{Y}} \langle x, K^* y \rangle - F^*(y) \right\}. \tag{2.4}$$

Since $G(x)$ is simply a scalar addition, we can move the max outwards and arrive at the required

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} G(x) + \langle x, K^* y \rangle - F^*(y). \tag{2.5}$$

□

Example 2.2.23. Convex Conjugate of Squared L^2 Norm
Let be $\mathcal{X} = L^2(\Omega)$ and $J(x) = \frac{\alpha}{2} \|x\|_2^2$ for $\alpha > 0$. Then we have $J^*(x^*) = \frac{1}{2\alpha} \|x^*\|_2^2$

Proof. We begin with the case $\alpha = 1$. Since $\mathcal{X}$ is a Hilbert space, it can be identified

with its dual space $\mathcal{X}^*$. Using the definition, we get for the convex conjugate:

$$J^*(x^*) = \sup_{x \in L^2(\Omega)} \langle x, x^* \rangle - \frac{1}{2} \|x\|_2^2 .$$

We can directly see that J^* is Frechet-differentiable and get for the optimal point $\hat{x} = x^*$. Consequently, we have

$$J^*(x^*) = \langle x^*, x^* \rangle - \frac{1}{2} \|x^*\|_2^2 = \frac{1}{2} \|x^*\|_2^2 .$$

The case for general α can be directly deduced from Lemma 5, since $(\lambda J(\cdot))^* = \lambda J^*(\frac{\cdot}{\lambda})$. □

Example 2.2.24. Convex Conjugate of General Norm
Let $\mathcal{X}$ be a measurable space and let $B(\mathcal{X})$ denote the unit ball in $\mathcal{X}$. Consider $J(x) = \|x\|_{\mathcal{X}}$, then we have $J^*(x^*) = \delta_{B(\mathcal{X}^*)}(x^*)$. Here, $\delta_{B(\mathcal{X}^*)}$ denotes the indicator function of the unit ball in $\mathcal{X}^*$ defined as

$$\delta_{B(\mathcal{X}^*)}(x^*) := \begin{cases} 0 & \text{if } x^* \in B(\mathcal{X}^*) \\ \infty & \text{else} \end{cases} .$$

Proof. We distinguish between two cases for the input argument x^*:

1. Let be $\|x^*\|_{\mathcal{X}^*} \leq 1$. Let us observe first that the following estimate for the dual pairing holds:

   $$\langle x^*, x \rangle \leq \|x^*\|_{\mathcal{X}^*} \|x\|_{\mathcal{X}} .$$

   Consequently, we have $\langle x^*, x \rangle - \|x\|_{\mathcal{X}} \leq 0$. With $\langle x^*, 0 \rangle = 0 = \|0\|_{\mathcal{X}}$ we get

   $$J^*(x^*) = \sup_{x^* \in \mathcal{X}^*} \langle x^*, x \rangle - \|x\|_{\mathcal{X}} = 0.$$

2. For the case $\|x^*\|_{\mathcal{X}^*} > 1$ we remind ourselves of the definition of the norm in $\mathcal{X}^*$:

   $$\|x^*\|_{\mathcal{X}^*} = \sup_{x \in \mathcal{X}} \frac{\langle x^*, x \rangle}{\|x\|_{\mathcal{X}}}.$$

Consequently there exists $x_0 \in \mathcal{X}$ such that

$$\frac{\langle x^*, x_0 \rangle}{\|x_0\|_{\mathcal{X}}} > 1 \Leftrightarrow \langle x^*, x_0 \rangle > \|x_0\|_{\mathcal{X}} \Leftrightarrow \langle x^*, x_0 \rangle - \|x_0\|_{\mathcal{X}} > 0.$$

Now, multiplying with some constant $c > 0$ we observe that

$$0 < c \underbrace{\left(\langle x^*, x_0 \rangle - \|x_0\|_{\mathcal{X}} \right)}_{>0} = \langle x^*, cx_0 \rangle - \|cx_0\|_{\mathcal{X}} \leq J^*(x^*).$$

Going to the limit $c \to \infty$ yields the required $J^*(x^*) = \infty$

□

Example 2.2.25. Convex Conjugate of Affine Linear L^1 Norm
Consider $J(x) = \|x + a\|_{L^1}$ for some $a \in \mathbb{R}$. Then we have $J^*(x^*) = \delta_{B(L^\infty)}(x^*) - \langle x^*, a \rangle$.

Proof. We simply combine the result from the previous example with Lemma 5 where $(J(\cdot + a))^* = J^*(\cdot) - \langle \cdot, a \rangle$. □

Example 2.2.26. From [51, 2012, Section 2.1, Example 2.1.6]
Consider $\mathcal{X} = \mathbb{R}^n, \mathcal{Y} = \mathbb{R}^m$ and $K \in \mathbb{R}^{m \times n}$. In the following table we wrote down some primal problems and their corresponding dual problems.

Primal Problem			Dual Problem		
$\frac{1}{2}\|Ku - f\|_2^2$	$+$	$\alpha \|u\|_1$	$\frac{1}{2}\|p - f\|_2^2$	$+$	$\delta_{B(L^\infty)}(\frac{K^T p}{\alpha})$
$\frac{1}{2}\|Ku - f\|_2^2$	$+$	$\alpha \|u\|_\infty$	$\frac{1}{2}\|p - f\|_2^2$	$+$	$\delta_{B(L^1)}(\frac{K^T p}{\alpha})$
$\frac{1}{2}\|Ku - f\|_2^2$	$+$	$\delta_{B(L^1)}(\frac{u}{\alpha})$	$\frac{1}{2}\|p - f\|_2^2$	$+$	$\alpha \left\|K^T p\right\|_\infty$
$\frac{1}{2}\|Ku - f\|_2^2$	$+$	$\delta_{B(L^\infty)}(\frac{u}{\alpha})$	$\frac{1}{2}\|p - f\|_2^2$	$+$	$\alpha \left\|K^T p\right\|_1$
$\frac{1}{2}\|u - f\|_2^2$	$+$	$\alpha \|Ku\|_1$	$\frac{1}{2}\left\|K^T p - f\right\|_2^2$	$+$	$\delta_{B(L^\infty)}(\frac{p}{\alpha})$

2.2.7. Examples for Concrete Problems

In this section we provide a set of concrete examples of variational problems. Problems of this structure will arise especially in Chapters 4 and 5 and we are able to directly apply the presented solutions to these models.

Lemma 2.2.27. L^2 Optimization
Let $\boldsymbol{x} := (x_1, \ldots, x_n), \boldsymbol{b} := (b_1, \ldots, b_n)$ and consider the discrete variational problem

$$\arg\min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\lambda}{2} \|K\boldsymbol{x}\|_2^2 ,$$

which consists of an L^2 data term and an additional L^2 regularization term with a linear operator K (e.g. the gradient).
Then the solution can be restricted to solving a system of linear equations $A\boldsymbol{x} = \boldsymbol{b}$.

Proof. We simply write down the optimality condition

$$\boldsymbol{x} - \boldsymbol{b} + \lambda K^* K\boldsymbol{x} \stackrel{!}{=} 0 \Leftrightarrow \boldsymbol{x} + \lambda K^* K\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow (I + K^* K)\boldsymbol{x} = \boldsymbol{b}$$

now, defining $A := I + K^* K$, we obtain the required result. □

Lemma 2.2.28. L^1 Optimization
Let $\boldsymbol{x} := (x_1, \ldots, x_n), \boldsymbol{a} := (a_1, \ldots, a_n)$ and consider the L^1 optimization problem

$$\arg\min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{a}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 .$$

Depending on the context we denote $\|\boldsymbol{x}\|_1 := \sum_{i=1}^n |x_i|$ as the *anisotropic*, and $\|\boldsymbol{x}\|_1 := \sqrt{\sum_{i=1}^n x_i^2}$ as the *isotropic* L^1 *optimization* problem.
The solution $\hat{\boldsymbol{x}}$ is given by

$$\hat{\boldsymbol{x}} = sgn(\boldsymbol{a}) \cdot \max(|\boldsymbol{a}| - \lambda, 0) =: S_{aniso}(\boldsymbol{a}, \lambda)$$

in the anisotropic case and by

$$\hat{\boldsymbol{x}} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_2} \cdot \max(\|\boldsymbol{a}\|_2 - \lambda, 0) =: S_{iso}(\boldsymbol{a}, \lambda)$$

in the isotropic case. Inspired by the fact that the input argument $\boldsymbol{a}$ shrinks towards zero, the solution is often called **shrinkage**.

Proof. **Anisotropic case:**
We basically follow the arguments from [58] and start with the anisotropic case. The

problem now reads

$$\arg\min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{a}\|_2^2 + \lambda \sum_{i=1}^{n} |x_i| \,.$$

Due to the fact that the x_i are independent we can restrict to the 1-dimensional case. Using the definition of the dual norm we can rewrite the problem to

$$\arg\min_{x} \frac{1}{2\lambda} \|x - a\|_2^2 + \arg\max_{y \in Y} \langle x, y \rangle \,.$$

Since the 1-norm is the dual of the infinity norm we have $Y := \left\{ y : |y| \leq 1 \right\}$. Both, arg min and arg max exist and we can interchange them:

$$\arg\max_{y \in Y} \arg\min_{x} \frac{1}{2\lambda} \|x - a\|_2^2 + \langle x, y \rangle \,. \tag{2.6}$$

With this structure we can now solve the inner problem directly and get as optimality condition

$$\frac{1}{\lambda}(x - a) + y \stackrel{!}{=} 0 \leftrightarrow x = a - \lambda y$$

Inserting this in Equation (2.6) reads

$$\begin{aligned}
&\arg\max_{y \in Y} \frac{1}{2\lambda} \|a - \lambda y - a\|_2^2 + \langle a - \lambda y, y \rangle \\
\Leftrightarrow &\arg\max_{y \in Y} \frac{\lambda}{2} \|y\|_2^2 + \langle a, y \rangle - \lambda \langle y, y \rangle \\
\Leftrightarrow &\arg\max_{y \in Y} -\frac{\lambda}{2} \|y\|_2^2 + \langle a, y \rangle \\
\Leftrightarrow &\arg\max_{y \in Y} -\frac{\lambda}{2} \left\| y - \frac{a}{\lambda} \right\|_2^2
\end{aligned}$$

The solution can be found by case analysis and using the definition of Y as

$$\hat{y} = \begin{cases} \frac{a}{\lambda} & \text{if } \left|\frac{a}{\lambda}\right| \leq 1 \quad (\Leftrightarrow |a| \leq \lambda \Leftrightarrow |a| - \lambda \leq 0) \\ sgn(\frac{a}{\lambda}) & \text{if } \left|\frac{a}{\lambda}\right| > 1 \quad (\Leftrightarrow |a| > \lambda \Leftrightarrow |a| - \lambda > 0) \end{cases}$$

Going back to the primal problem we have

$$\arg\min_{x} \frac{1}{2\lambda} \|x - a\|_2^2 + \langle x, \hat{y} \rangle \,,$$

with the optimality condition

$$\frac{1}{\lambda}(\hat{x}-a)+\hat{y}\stackrel{!}{=}0 \Leftrightarrow \hat{x}=a-\lambda\hat{y}=\begin{cases}0 & \text{if } |a|-\lambda\leq 0\\ a-\lambda sgn(a) & \text{if } |a|-\lambda>0\end{cases}$$

This is equivalent to the proposed formula

$$\hat{\boldsymbol{x}}=sgn(\boldsymbol{a})\cdot\max(|a|-\lambda,0).$$

Isotropic case:
The previous proof needs only small adjustments to cover the isotropic case. The minimization problem here reads

$$\arg\min_{\boldsymbol{x}}\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{a}\|_2^2+\lambda\sqrt{\sum_{i=1}^{n}x_i^2}.$$

We can no longer restrict to the 1-dimensional case, because the components are now connected through the squareroot. The first change in the proof appears within the set Y. Since the 2-norm is dual to itself, the set changes to $Y:=\left\{\boldsymbol{y}:\|\boldsymbol{y}\|_2\leq 1\right\}$. This requires a change in the solution for the minimization problem

$$\arg\max_{\boldsymbol{y}\in Y}-\frac{\lambda}{2}\left\|\boldsymbol{y}-\frac{\boldsymbol{a}}{\lambda}\right\|_2^2$$

The solution consists of the same structure as before. If $\frac{\boldsymbol{a}}{\lambda}$ lies within the L^2 unit ball we choose $\hat{\boldsymbol{y}}=\frac{\boldsymbol{a}}{\lambda}$ otherwise we choose the element in the L^2 ball closest (in the L^2 sense) to $\frac{\boldsymbol{a}}{\lambda}$, which is the projection of $\frac{\boldsymbol{a}}{\lambda}$ onto the ball, hence $\hat{\boldsymbol{y}}=\frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_2}$. Together we have

$$\hat{\boldsymbol{y}}=\begin{cases}\frac{\boldsymbol{a}}{\lambda} & \text{if } \|\boldsymbol{a}\|_2-\lambda\leq 0\\ \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_2} & \text{if } \|\boldsymbol{a}\|_2-\lambda>0\end{cases}$$

This transfers to the solution $\hat{\boldsymbol{x}}$ as follows

$$\hat{\boldsymbol{x}}=\begin{cases}0 & \text{if } \|\boldsymbol{a}\|_2-\lambda\leq 0\\ a-\lambda\frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_2} & \text{if } \|\boldsymbol{a}\|_2-\lambda>0\end{cases},$$

which is equivalent to the short formula

$$\hat{\boldsymbol{x}} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_2} \cdot \max(\|a\|_2 - \lambda, 0).$$

□

Lemma 2.2.29. Affine Linear Soft Thresholding
Consider the affine linear L^1 optimization problem for given $\tilde{u}$ and g

$$\arg\min_u \frac{1}{2} \|u - \tilde{u}\|_2^2 + \lambda \|\beta u - g\|_1, \quad \lambda, \beta > 0.$$

Let us denote the affine linear function as follows:

$$f(u) := \beta u - g.$$

Then the L^1 minimization problem can be solved directly via the soft thresholding formula:

$$u_{min} = \tilde{u} + \begin{cases} \lambda\beta & \text{if } f(\tilde{u}) < -\lambda\beta^2 \\ -\lambda\beta & \text{if } f(\tilde{u}) > \lambda\beta^2 \\ -\frac{f(\tilde{u})}{\beta} & \left|f(\tilde{u})\right| \leq \lambda\beta^2 \end{cases}$$

Proof. The proof is quite similar to the classical shrinkage formula. Let us observe that, for calculating the optimality condition, the critical point is given by $u = \frac{g}{\beta}$, because the L^1 norm is not differentiable here. That is why we distinguish between the three cases $u < \frac{g}{\beta}, u > \frac{g}{\beta}$ and $u = \frac{g}{\beta}$.

1. $u < \frac{g}{\beta}$: The optimality condition can be directly calculated as

$$u - \tilde{u} + \lambda(-1)\beta \stackrel{!}{=} 0.$$

Consequently, we obtain the solution as

$$u = \tilde{u} + \lambda\beta.$$

From $u < \frac{g}{\beta}$ we deduce the equivalent condition

$$\tilde{u} + \lambda\beta < \frac{g}{\beta} \Leftrightarrow \beta\tilde{u} - g < -\lambda\beta^2 \Leftrightarrow f(\tilde{u}) < -\lambda\beta^2,$$

which completes the first case.

2. $u > \frac{g}{\beta}$: The second case can be done in analogy to the first one.

3. $u = \frac{g}{\beta}$: For the critical case, the optimality condition reads

$$u - \tilde{u} + \lambda p \stackrel{!}{=} 0 \Leftrightarrow u = \tilde{u} - \lambda p,$$

where p is an element of the subdifferential of $\|\beta u - g\|_1$. Now, the subdifferential of a standard L^1 term $\|u\|_1$ is given by $\{p \in [-1, 1]\}$. In our case we have βu which lifts the subdifferential to $\{p \in [-\beta, \beta]\}$. The affine term g is a constant shift of all function values and has no influence to the subdifferential.
To find the right p, we use the optimality condition above and insert the known $u = \frac{g}{\beta}$:

$$\frac{g}{\beta} - \tilde{u} + \lambda p = 0 \Leftrightarrow p = -\frac{g}{\lambda\beta} + \frac{\tilde{u}}{\lambda}.$$

Inserting this into the update for u, we have

$$u = \tilde{u} - \lambda\left(-\frac{g}{\lambda\beta} + \frac{\tilde{u}}{\lambda}\right) = \tilde{u} - \left(\tilde{u} - \frac{g}{\beta}\right) = \tilde{u} - \frac{f(\tilde{u})}{\beta}.$$

Finally, to obtain the correct condition, we already know that $p \in [-\beta, \beta]$ and consequently $|p| \leq \beta$. Inserting this yields

$$\left|-\frac{g}{\lambda\beta} + \frac{\tilde{u}}{\lambda}\right| \leq \beta \Leftrightarrow |\beta\tilde{u} - g| \leq \lambda\beta^2 \Leftrightarrow \left|f(\tilde{u})\right| \leq \lambda\beta^2.$$

□

Lemma 2.2.30. Vectorial Affine Linear Soft Thresholding
Consider for $\boldsymbol{u} := (u_1, \ldots, u_n)^T, \boldsymbol{\beta} := (\beta_1, \ldots, \beta_n)^T$ the affine linear function

$$f(\boldsymbol{u}) := \boldsymbol{\beta} \cdot \boldsymbol{u} - g.$$

Then for some $\tilde{u}$ the L^1 minimization problem

$$\arg\min_{\boldsymbol{u}} \frac{1}{2} \|\boldsymbol{u} - \tilde{u}\|_2^2 + \lambda \left\| f(\boldsymbol{u}) \right\|_1$$

can be solved directly via the soft thresholding formula

$$u_{min} = \tilde{u} + \begin{cases} \lambda\boldsymbol{\beta} & \text{if } f(\tilde{u}) < -\lambda \|\boldsymbol{\beta}\|^2 \\ -\lambda\boldsymbol{\beta} & \text{if } f(\tilde{u}) > \lambda \|\boldsymbol{\beta}\|^2 \\ -\frac{f(\tilde{u})\beta}{\|\boldsymbol{\beta}\|^2} & \text{else} \end{cases}$$

Proof. Can be done with slight modifications to the one-dimensional case. □

2.3. Variational Minimization

The aim of this section is to introduce a framework for the minimization of variational models in image processing. The previously introduced concept of duality can be used to derive efficient schemes for the minimization of variational models. In the following, we introduce so-called *primal-dual* methods. They offer a way to express the (often numerically expensive) inversion of an operator by application of the operator and the respective adjoint operator. Similar primal-dual frameworks have been introduced by Esser, Zhang and Chan in 2010 [33], Chambolle and Pock in 2011 [26] and Zhang, Burger and Osher in 2011 [78]. Our choice is a primal-dual algorithm proposed by Chambolle and Pock. We will give a coarse introduction to the general framework and will explain how to apply this to our class of problems.

2.3.1. Primal-Dual Framework

As a start consider two finite dimensional vector spaces $\mathcal{X}$ and $\mathcal{Y}$ equipped with a scalar product $\langle \cdot, \cdot \rangle$ and a norm $\|\cdot\|_2^{\frac{1}{2}}$. We furthermore consider a continuous linear operator $K : \mathcal{X} \to \mathcal{Y}$. Now, the general class of problems in this thesis can be written as

$$\min_{x \in \mathcal{X}} G(x) + F(Kx), \tag{2.7}$$

with $F, G : \mathcal{X} \to \mathbb{R}$ are proper, convex and lower semi-continuous functionals. In the following we denote Equation (2.2) as the *primal problem*. Consider for example the

well-known Rudin-Osher-Fatemi (ROF) functional

$$J(u) = \frac{1}{2} \|u - f\|_2^2 + \alpha \|\nabla u\|_1 ,$$

here we simply set $G(u) := \frac{1}{2} \|u - f\|_2^2$ and $F(Ku) := \alpha \|\nabla u\|_1$ with $K := \nabla$ and the requirements are met.
Now, it can be shown that solving the minimization problem in Equation 2.2 is equivalent to solving the saddle-point problem

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \langle Kx, y \rangle + G(x) - F^*(y), \tag{2.8}$$

where F^* is the convex conjugate of F (see Section 2.2.6 for details). The minimization problem in Equation 2.8 is called the *primal-dual problem*.
Let us now return to the general problem in Equation (2.7) with $K = I$

$$\min_x F(x) + G(x).$$

It can be shown (see [30]) that the solution is given by the fixed point equation

$$x = prox_{\tau F}(x - \tau \nabla G(x)), \tag{2.9}$$

with some problem-dependent parameter τ. The expression $prox_F$ stands for the so-called *proximity* or *resolvent* operator, defined as follows:

Definition 2.3.1. Resolvent Operator
Let F be a proper functional and $\mathcal{X}$ a Banach space. For some input argument y and constant $\tau > 0$ we define the resolvent operator as

$$x = (I + \tau \partial F)^{-1}(y) := \arg\min_x \left\{ \frac{\|x - y\|_2^2}{2} + \tau F(x) \right\} = prox_{\tau F}(y)$$

The proximity operator can be interpreted as a compromise between minimizing F and being close to the input argument y. We are looking for a fixed point in Equation (2.9). A simple numerical scheme for doing so is the following iteration with some initial value x^0:

$$x^{k+1} = prox_{\tau F}(x^k - \tau \nabla G(x^k)).$$

Now, the Chambolle-Pock algorithm aims at solving the saddle-point Problem (2.8) by performing alternating fixed point iterations for x and y and adding an intermediate overrelaxation-step to x. This leads to the following 3-step scheme

Definition 2.3.2. Chambolle-Pock Algorithm
Choose $\tau, \sigma > 0, \theta \in [0,1], (x^0, y^0) \in \mathcal{X} \times \mathcal{Y}$ and set $\bar{x}^0 = 0$. Then the Chambolle-Pock iterations for Problem (2.8) are defined as

$$\begin{aligned} y^{k+1} &= prox_{\sigma F^*}(y^k - \sigma \nabla_y \langle K\hat{x}^k, -y \rangle) \\ &= prox_{\sigma F^*}(y^k + \sigma K\hat{x}^k) \end{aligned} \tag{2.10}$$

$$\begin{aligned} x^{k+1} &= prox_{\tau G}(x^k - \tau \nabla_x \langle Kx, -y^{k+1} \rangle) \\ &= prox_{\tau G}(x^k - \tau K^* y^{k+1}) \end{aligned} \tag{2.11}$$

$$\hat{x}^{k+1} = x^{k+1} + \theta(x^{k+1} - x^k) \tag{2.12}$$

For some applications the dual functional F^* in Equation (2.10) is unknown, hard to evaluate or can only be calculated at very high computational costs. A way out yields the Moreau identity, which gives an easy correspondence between primal and dual functional.

Theorem 2.3.3. Moreau's Identity
Let F be a proper functional, $\mathcal{X}$ a Banach space and F^* the corresponding dual functional. Then primal and dual resolvent are related through the following identity:

$$\begin{aligned} x &= prox_{\tau F}(x) + \tau prox_{\frac{1}{\tau}F^*}(\frac{x}{\tau}) \\ \Leftrightarrow prox_{\tau F}(x) &= x - \tau prox_{\frac{1}{\tau}F^*}(\frac{x}{\tau}). \end{aligned}$$

This is very useful, because this identity enables a modified Chambolle-Pock variant where we can restrict ourselves to the purely primal functionals F and G.

Corollary 2.3.4. Modified Chambolle-Pock Algorithm
Using Moreau's identity (Theorem 2.3.3) the dual update in Definition 2.3.2 simplifies to an evaluation of the primal functional F. Then the scheme reads

$$\begin{aligned} y^{k+1} &= y^k + \sigma K\hat{x}^k - \sigma prox_{\frac{1}{\sigma}F}(\frac{y^k}{\sigma} + K\hat{x}^k) \\ x^{k+1} &= prox_{\tau G}(x^k - \tau K^* y^{k+1}) \end{aligned}$$

$$\hat{x}^{k+1} = x^{k+1} + \theta(x^{k+1} - x^k).$$

Going back to our initial ROF example we obtain the following scheme

$$\begin{aligned}
\tilde{y}^{k+1} &= y^k + \sigma \nabla \hat{x}^k \\
y^{k+1} &= \tilde{y}^{k+1} - \sigma \arg\min_y \left\{ \frac{1}{2} \left\| y - \frac{\tilde{y}^{k+1}}{\sigma} \right\|_2^2 + \frac{\alpha}{\sigma} |y|_1 \right\} \\
\tilde{x}^{k+1} &= x^k - \tau(-\nabla \cdot y^{k+1}) \\
x^{k+1} &= \arg\min_x \left\{ \frac{1}{2} \left\| x - \tilde{x}^{k+1} \right\|_2^2 + \frac{\tau}{2} \|x - f\|_2^2 \right\} \\
\hat{x}^{k+1} &= x^{k+1} + \theta(x^{k+1} - x^k).
\end{aligned}$$

The subproblem in y is a simple $L^1 - L^2$ problem, which can be directly solved by soft thresholding $S(\cdot, \cdot)$ (see Lemma 2.2.28). We obtain

$$y = S(\frac{\alpha}{\sigma}, \frac{\tilde{y}^{k+1}}{\sigma}).$$

For the linear subproblem in x we calculate the optimality condition as

$$x - \tilde{x}^{k+1} + \tau(x - f) \stackrel{!}{=} 0 \Leftrightarrow x = \frac{1}{1+\tau}(\tilde{x} + \tau f).$$

We end up with the following fairly simple iterative scheme

$$\begin{aligned}
\tilde{y}^{k+1} &= y^k + \sigma \nabla \hat{x}^k \\
y^{k+1} &= \tilde{y}^{k+1} - \sigma S(\frac{\alpha}{\sigma}, \frac{\tilde{y}^{k+1}}{\sigma}) \\
\tilde{x}^{k+1} &= x^k + \tau \nabla \cdot y^{k+1} \\
x^{k+1} &= \frac{1}{1+\tau}(\tilde{x} + \tau f) \\
\hat{x}^{k+1} &= x^{k+1} + \theta(x^{k+1} - x^k).
\end{aligned}$$

3 Image Segmentation

This chapter is dedicated to image segmentation models. We begin by introducing the mathematical context of image segmentation and proceed with two concrete models, namely the Geodesic Active Contours model in Section 3.2.1 and the Chan-Vese model in Section 3.2.2. The chapter ends with two applications for image segmentation, one concerning feature extraction from vertebra and another concerning intracellular flow. The latter application will be discussed again in Chapter 6 in the context of mathematical models for motion estimation and joint image reconstruction.

3.1. Background

3.1.1. Introduction

If we should describe the word *segmentation* to a non-mathematician, we would most likely speak of dividing an image into different regions or objects. Usually, we seek for a simplified version of the underlying image, which consists of homogeneous regions that are separated by sharp edges. In this context, depending on their position in the image, some contours may be more important than others in the classification process. However, the important image-feature in the segmentation process is usually an edge. From a general point of view, there exist two very different strategies in the field of segmentation. The first strategy uses the variational Mumford and Shah image model for decoupling the image into sharp edges and smooth parts [52]. The functional is

defined by

$$J(u,\Gamma) = \int_{\Omega} \|u-f\|_2^2 + \int_{\Omega\setminus\Gamma} \|\nabla u\|_2^2 + \mathcal{H}^{d-1}(\Gamma). \tag{3.1}$$

Here u is the reconstructed image we are interested in. The variable f denotes the given data, and $\mathcal{H}^{d-1}(\Gamma)$ denotes the $(d-1)$-dimensional Hausdorff-measure of the set of discontinuities Γ. The first part keeps the approximation u close to the data f in a least squares sense, the second term ensures that u consists of smooth regions and the third term minimizes the length of the boundary set Γ. Having constructed the pair (u,Γ), the segmentation can directly be extracted as the connected components in u. Unfortunately, from a numerical point of view, the classical Mumford-Shah model is hard to minimize due to the geometrical variable Γ. In Section 3.2.2 we present the Chan-Vese model [27], that is a simplification of the Mumford-Shah energy, and that can be computed at low computational costs.

The second segmentation strategy aims at finding certain objects in images, where the structure inside the object is usually neglected. The basic idea is to place a curve around an image, which contracts as the algorithm runs. The curve then stops at *boundaries* (however they may be defined) and the image can be divided into two regions - i.e. **inside** and **outside** of the curve. This strategy was proposed by Kass, Witkin and Terzopoulos [45] and named as *active contours*. We want to underline here that in classical context of active contours the algorithms needs the set of boundaries as an external input.

Since this thesis is dedicated to biological applications, we want to give a short example for the classes of images to be segmented. Figure 3.1 gives an example of the data we are interested in - moving cells. Since we are unable to record a moving cell directly, certain components of the cell are labeled with fluorophores, which emit light at a certain frequency. To create the left image in Figure 3.1, proteins concentrating in the membrane (hull) of the cell have been labeled to represent the cell as a whole. Because we are not only interested in the movement of the whole cell, but also in the dynamics within the cell, the right image in Figure 3.1 shows the labeled mitochondria, which are organelles inside of a cell. This is a dynamic process and we visualize the state at only one time step. The point of interest for biologists is the correlation between cell-movement and behavior of the organelles inside.

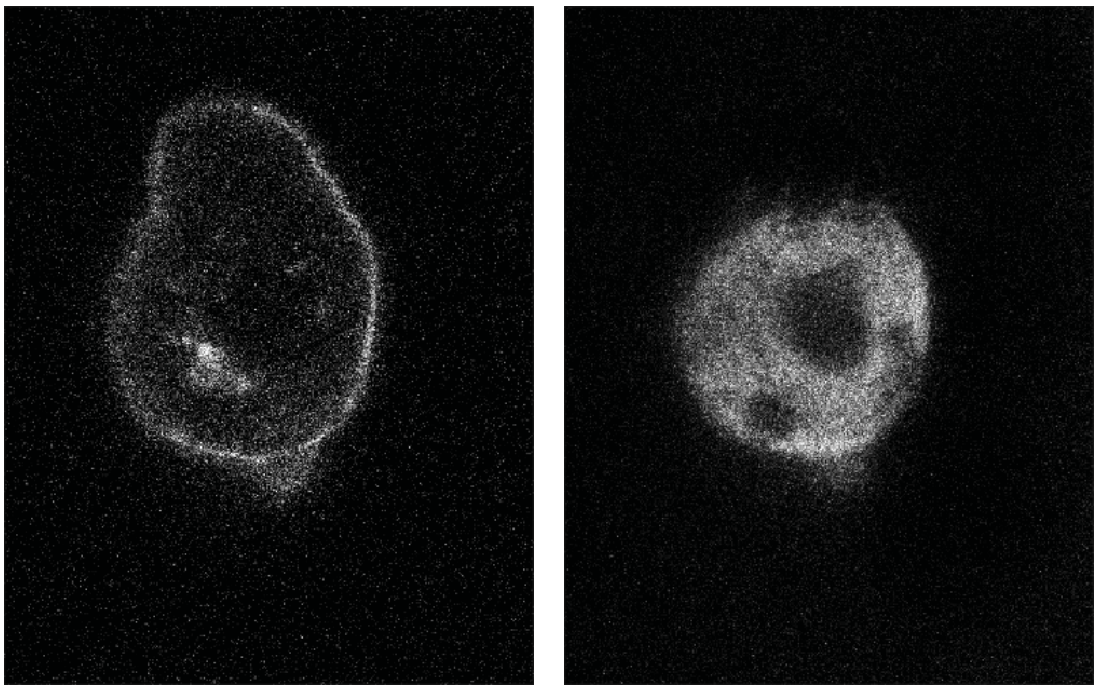


Figure 3.1.: Image data from microscopy. Left: Image of the whole cell with coloured actin filaments. Right: Coloured mitrochondia. Images are property of Raz group from ZMBE [79].

3.1.2. Mathematical Framework

Segmentation

In the following section, we give a short definition of segmentation in image processing and explain how an image and its segmentation correlate.

General Definition:

From a mathematical point of view a segmentation is a decomposition of an image into different regions or categories, which correspond to different objects or parts of objects. For a continuous image u on the domain $\Omega \subset \mathbb{R}^n$, we would like to find subsets $S_1, \ldots, S_m \subset \Omega$, such that:

$$\Omega = S_1 \cup \ldots \cup S_m$$

with $S_i \cap S_j = \emptyset$ for $i \neq j$.

The criteria used to identify a region S_i are strongly algorithm-dependent. We can formulate two general rules:

- The image varies smoothly inside the region.

- The boundary represents a discontinuity or has at least a high intensity gradient.

For the following work, let

$$S = \{R_1, \ldots, R_m\}$$

be a segmentation of an image u in m different regions.

Representation:
In general there are two ways to represent a segmentation. The first and intuitive one uses a region association function. For m regions $R_1, \ldots, R_m \subset \Omega$ we define a function

$$RF : \Omega \to \{1, \ldots, m\},$$

which holds

$$x \in R_i \iff RF(x) = i,$$

or in other words a segmentation of a discrete image u is just another matrix (same size), which contains the region number on each grid point.

If the segmentation consists of only two regions R_1 and R_2 we can use a so-called *level-set representation*. To divide the image, we use a distance function:

$$\Phi : \Omega \times [-T, T] \longrightarrow \mathbb{R},$$

which moves in normal direction of the level lines with speed v. Φ is obtained as the solution of the Hamilton-Jacobi equation

$$\frac{\Phi}{\partial t} = v \left\| \nabla \Phi \right\|, \quad \Phi(x, t) = 0 \text{ for } x \in \Gamma, \tag{3.2}$$

where Γ denotes the boundary between R_1 and R_2. We would like to interpret Φ as a distance function and set $|\nabla \Phi| = 1, v = 1$. In this context, Γ is often mentioned as the *zero level-set*. We can imagine Φ as a curve Γ that moves with constant speed 1 in time. It blows up in time respectively shrinks in negative timesteps. We simply denote for every point $x \in \Omega$ the time until the curve hits the point. It is clear that this is the shortest distance to Γ.

Now we can divide the image in two regions by the following rules:

$$x \in R_1 \Longleftrightarrow \Phi(x) > 0$$
$$x \in R_2 \Longleftrightarrow \Phi(x) < 0$$

The great advantage of the level-set representation is a sub-pixel contour precision. This is achieved due to the fact that in every pixel we just denote the distance to the boundary. In that way, the boundary is defined implicitly. Thus, in contrary to a region function, the boundary does not have to lie in the middle between two pixels.

Both representations can be converted to each other (with some loss of precision). This is a non-trivial problem, which is discussed in Appendix $A.4$.

3.2. Algorithms

In this section we introduce two well-known frameworks for image segmentation. On the one hand, the geodesic active contour model performs an image segmentation by evolution of a parabolic PDE. On the other hand, the Chan-Vese model is a simplification of the Mumford Shah model that can be explicitly formulated by a variational energy. We focus at the formulation of the respective models and their numerical implementation. The interested reader is refered to the original papers by Mumford and Shah [52] resp. Chan and Vese [28].

3.2.1. Geodesic Active Contours

Edge Detection:

As mentioned before, a segmentation consists of different regions R_i. Let moreover Γ_i denote the boundary set of R_i. Active contour algorithms, often called *snakes*, represent a certain class of segmentation algorithms that evolve a curve across the image domain which stops at high intensity gradients. In this context **edge detectors** come into play. We inherit the definition from [8] where

$$g : [0, \infty) \to [0, \infty)], \quad s \mapsto g(s)$$

is a valid edge detector function if g satisfies the following properties:

- g(s) is a monotonically decreasing function for $s \to \infty$,

- $g(0) = c, c \in \mathbb{R}^+$,
- $\lim_{s\to\infty} g(s) = 0$.

Consequently, we expect g to be 0 at edges and c in homogeneous regions, with a continuous transition between both values. A well-known example for g is given by

$$g(s) = \frac{1}{1+s^2}.$$

Until now, the input argument for g was kept general, but there are several choices possible. Here, as a first-order method, the magnitude of the gradient directly occurs

$$|\nabla u| := \sqrt{u_x^2 + u_y^2},$$

as a possible choice for s. To be less sensitive towards noise, a convolution with a Gaussian kernel G_σ is often performed as a preprocessing step, hence

$$|\nabla u|_\sigma := \sqrt{(u * G_\sigma)_x^2 + (u * G_\sigma)_y^2},$$

where $G_\sigma := \frac{1}{2\pi}\exp(-\frac{x^2}{2\sigma^2})$.
Besides continuous functions for edge detection, there exist several methods that produce a binary feedback, where the output is *true* for edge and *false* for background. One of these advanced and more robust first-order methods is given by the *Canny detector*, introduced by John F. Canny in 1986 [22]. Here, also a smoothed image is used to calculate the magnitude of the gradient in u. Additionally, several postprocessing steps are applied to enhance $|\nabla u|_\sigma$, such as local filtering of maximum edges, thresholding of noise-caused edges and removal of edges not belonging to a larger connected component. Finally, the edge-enhanced image is binarized.
Nowadays, often replaced by the Canny detector, the *Marr-Hildreth* detector, introduced by David Marr and Ellen Catherine Hildreth in 1980 [49], is a second order edge-detector. The idea here is to calculate the Laplacian

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2},$$

of a Gaussian itself. The resulting filter is used in a convolution with the image u to create an edge-map. Usually, the result of Marr-Hildreth is thresholded to create a binary output, but can be used as an input for g in its absolute valued version. The main drawback towards the Canny detector are the missing postprocessing steps. An

example can be found in Figure 3.2.

Model:

A well-known active contour model based on an edge function g is given by the **geodesic active contours**, introduced by Caselles, Kimmel and Sapiro in 1997 [23]. Here, the evolution of a level-set function Φ is given by the parabolic partial differential equation

$$\Phi_t = |\nabla\Phi| \, \nabla \cdot \left(g \frac{\nabla\Phi}{|\nabla\Phi|} \right), \tag{3.3}$$

with some initial boundary condition $\Phi(\cdot, 0) = \Phi_0$. The geometrical interpretation of Equation (3.3) relies on the classical Eikonal equation,

$$\Phi_t = |\nabla\Phi| \, v,$$

where the zero level-set evolves in normal direction to the level lines with speed v. In the context of geodesic active contours the speed v is given by

$$v = \nabla \cdot \left(g \frac{\nabla\Phi}{|\nabla\Phi|} \right).$$

For $g \equiv 1$ this boils down to the definition of the local curvature. Consequently, the equation

$$\Phi_t = |\nabla\Phi| \, \nabla \cdot \left(\frac{\nabla\Phi}{|\nabla\Phi|} \right), \tag{3.4}$$

describes an evolution of Φ towards the normal direction with speed equal to the local curvature. This is generally known as *mean curvature motion*. Now, adding a g to the inner term acts as a stopping criterion, because g tends to zero near the edges and hence the speed v becomes zero as well.

Numerical Realization:

Following the suggestion in [28], the numerical realization for Equation (3.3) can be done in a standard explicit finite difference scheme. Let therefore D^+, D^- and D^c denote forward, backward and central difference operators. Then we obtain:

$$\frac{\Phi_{i,j}^{n+1} - \Phi_{i,j}^{n}}{\delta_t} = \sqrt{(D_x^c \Phi_{i,j}^n)^2 + (D_y^c \Phi_{i,j}^n)^2} \left[\nabla \cdot \left(g \frac{\nabla\Phi^n}{|\nabla\Phi^n|} \right) \right]_{i,j},$$

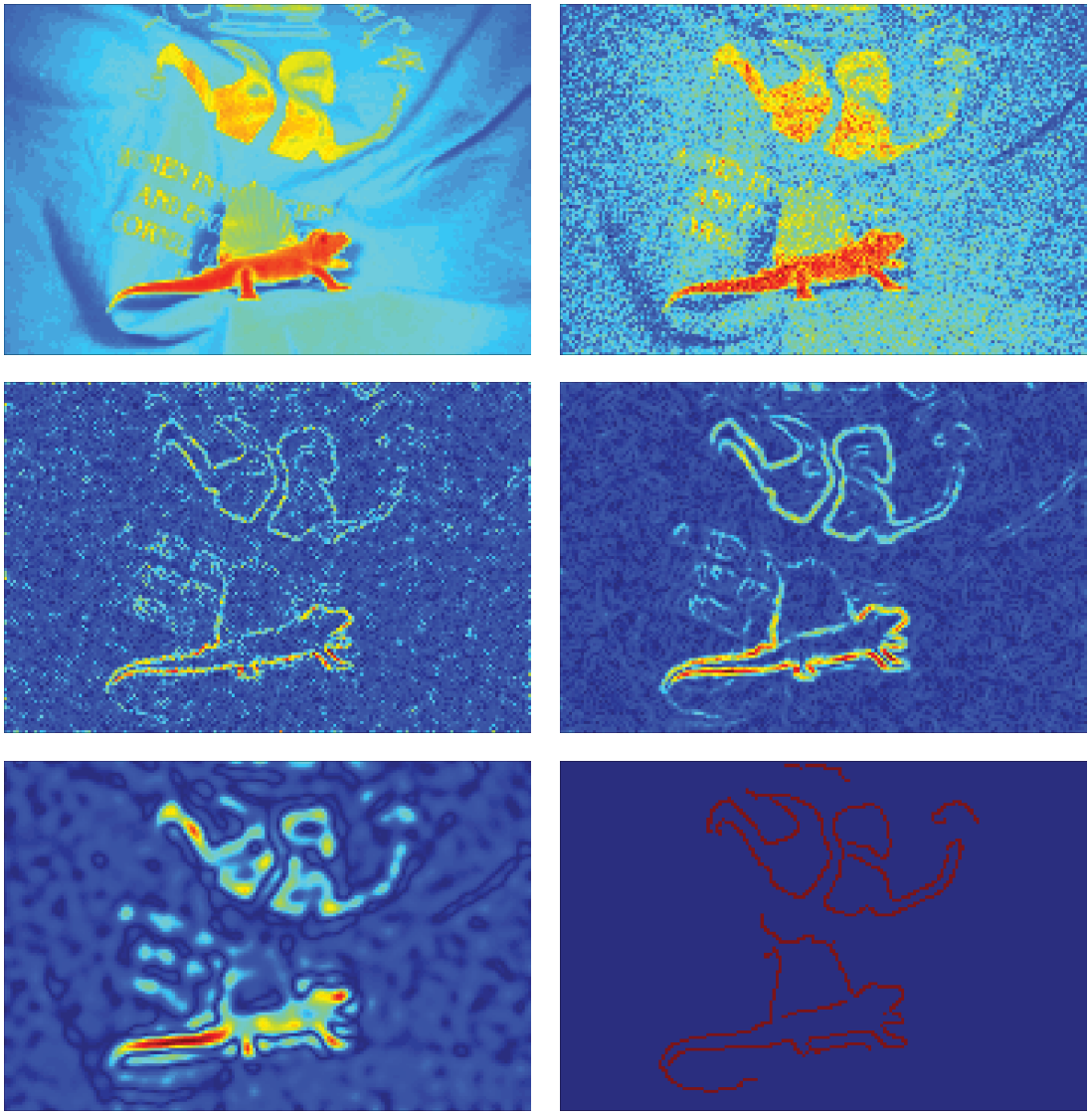


Figure 3.2.: Overview of different edge detection algorithms under the influence of noise. Top-left: Image from [44], Top-right: Image with noise (Gaussian with $\sigma = 0.005$), Middle-left: $|\nabla u|$, Middle-right: $|\nabla u|_\sigma$, Bottom-left: Marr-Hildreth, Bottom-right: Canny edge detector applied to the image.

$$\Leftrightarrow \quad \Phi_{i,j}^{n+1} = \Phi_{i,j}^{n} + \delta_t \sqrt{(D_x^c \Phi_{i,j}^n)^2 + (D_y^c \Phi_{i,j}^n)^2} \left[\nabla \cdot \left(g \frac{\nabla \Phi^n}{|\nabla \Phi^n|} \right) \right]_{i,j},$$

with

$$\nabla \cdot \left(g \frac{\nabla \Phi^n}{|\nabla \Phi^n|} \right) = D_x^- \left(g \frac{D_x^+ \Phi^n}{\sqrt{(D_x^+ \Phi^n)^2 + (D_y^c \Phi^n)^2}} \right) + D_y^- \left(g \frac{D_y^+ \Phi^n}{\sqrt{(D_x^c \Phi^n)^2 + (D_y^+ \Phi^n)^2}} \right).$$

Due to the fact that our weighting function g is non-negative, the zero level-set $\Phi^{-1}(0)$ is only able to evolve inwards. Consequently, the initial condition $\Phi(\cdot, 0)$ is usually chosen such that the zero level-set lies outside of the object of interest. Now, the evolution in time will let the contour shrink and stop at edges of the object.

The convergence speed can be improved by an iterative multiscale method. Let

$$0 < s_1 < \ldots < s_n = 1$$

define scale levels and let further u^{s_i} represent an in every dimension by a factor of s_i downscaled version of the underlying image u. The idea is now to run the algorithm on the coarsest scale u^{s_1} first, and use the result as an initialization for the next finer scale. To avoid discretization issues, the result of the ith scale is pushed slightly outwards by

$$\Phi^{i+1}(\cdot, 0) := \text{rescale}(\Phi^i(\cdot, c)), \quad c \in \mathbb{R}^+.$$

Speaking figural, we set the initial zero level-set for scale $i+1$ to the isocontour level c for a small $c \in R^+$. We recall in this context that the inner part is characterized by $\{\Phi < 0\}$ and the outside part by $\{\Phi > 0\}$.

We end up with the following pseudocode:

Algorithm 1 Geodesic Active Contours

GeodesicActiveContoursg
$\Phi \leftarrow$ initializeLevelset()
$iteration \leftarrow 0$ $iteration < maxIteration$
$divTerm \leftarrow$ calculateDivergence(Φ,g)
$gradTerm \leftarrow$ calculateNormgradient(Φ)
$\Phi \leftarrow \Phi + \delta_t \cdot gradTerm \cdot divTerm$
$iteration \leftarrow iteration + 1$
Φ

3.2.2. Chan-Vese

Model:

A well known simplified version of the Mumford-Shah functional (Equation (3.1), resp. [52]) was given by Chan and Vese in 2001 [28]. Assuming that the observed image consists of only two regions R_1 and R_2, each of them with a constant brightness c_1 resp. r_2, they split up the first data term in the Mumford-Shah functional into

$$\int_{\Omega=R_1\cup R_2} (u-f)^2 = \int_{R_1} (u-c_1)^2 + \int_{R_2} (u-c_2)^2.$$

We mention here that R_1 and R_2 do not necessarily consist of only one connected component, which means that each of the regions may split up into various components. Since the image intensity is assumed to be constant in both regions R_1 and R_2, we have $\nabla u = 0$ Besides the data fidelity terms above, the Chan-Vese model consists of a penalty term for the length of the contour Γ, which separates R_1 and R_2, and an area term for R_1. The whole model then reads:

$$J(u, c_1, c_2) = \lambda_1 \int_{R_1} (u-c_1)^2 + \lambda_2 \int_{R_2} (u-c_2)^2 + \mu\mathcal{H}(\Gamma) + \nu\,|R_1|\,, \tag{3.5}$$

where $\mathcal{H}(\Gamma)$ is the Hausdorff measure of Γ (i.e. if Γ is smooth this is the d-dimensional length of Γ). Of course, the Chan-Vese models fits for more general images. The values c_1 and c_2 can then be seen as the mean intensity values for R_1 resp. R_2. The length term $\mathcal{H}(\Gamma)$ acts as a regularizer to be more robust towards noise. For the area term $|R_1|$, in most applications ν is set to zero, but it can theoretically be used to emphasize R_1 representing the smaller or larger region in the image.
We underline here that the Chan-Vese model is intensity based, hence it does not rely on a boundary detector.

Numerical Realization:

The direct minimization of Energy $J(u, c_1, c_2)$ in Equation (3.5) is challenging due to the dependence on the geometric variable Γ. But since the model consists of only two regions, it can be rewritten in equivalent form using the Heaviside function (see Definition $A.1$).

Lemma 3.2.1. Equivalent Chan-Vese Energy
Using a level-set function Φ, the Chan-Vese energy (Equation (3.5)) can be rewritten as the equivalent minimization problem

$$J(\Phi, c_1, c_2) = \lambda_1 \int_\Omega (u - c_1)^2 H(\Phi) + \lambda_2 \int_\Omega (u - c_2)^2 (1 - H(\Phi)) + \mu \int_\Omega \left|\nabla H(\Phi)\right| + \nu \int_\Omega H(\Phi)$$

Proof. It is directly clear that instead of integrating over R_1 we can equivalently integrate over the whole domain $\Omega = R_1 \cup R_2$ using the characteristic function $H(\Phi)$ inside the integrator. Hence we get:

$$\int_\Omega (u - c_1)^2 H(\Phi) = \int_{R_1} (u - c_1)^2.$$

The same argument holds for the integral over R_2 and the area term $|R_1|$.
The length $|\Gamma|$ needs slightly more theory. We start writing down the length using the zero level-set $\Phi^{-1}(0)$, which is the boundary $\partial \{\Phi > 0\}$

$$\begin{aligned} |\Gamma| &= \int_{\Phi^{-1}(0)} 1 \,\mathrm{d}s = \int_{\partial\{\Phi>0\}} 1 \,\mathrm{d}s = \int_{\mathbb{R}} \delta(x) \int_{\partial\{\Phi>x\}} 1 \,\mathrm{d}s \,\mathrm{d}x \\ &= \int_\Omega \delta(\Phi) \left|\nabla \Phi\right| \mathrm{d}\delta = \int_\Omega \left|\nabla H(\Phi)\right| \mathrm{d}\delta \end{aligned} \tag{3.6}$$

In the third step, the co-area formula (see Lemma 2.1.6) was used to transform the double integral to one integral over Ω. Furthermore, $\mathrm{d}\delta$ denotes the Dirac measure. □

The following Euler-Lagrange system can be deduced for the equivalent Chan-Vese energy from Lemma 3.2.1

$$\begin{aligned} 0 &\overset{!}{=} \lambda_1 \int_\Omega (u - c_1) H(\Phi), \text{ for } c_1, \\ 0 &\overset{!}{=} \lambda_2 \int_\Omega (u - c_2)(1 - H(\Phi)), \text{ for } c_2, \\ 0 &\overset{!}{=} \delta_\epsilon(\Phi) \left(-\mu C_\Phi \nabla \cdot \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) + \lambda_1 (u - c_1)^2 - \lambda_2 (u - c_2)^2 \right), \text{ for } \Phi, \end{aligned}$$

with $C_\Phi := \int_\Omega \left|\nabla H_\epsilon(\Phi)\right| \mathrm{d}x$ and $\delta_\epsilon, H_\epsilon$ smoothed Dirac delta respectively Heaviside functions. We refer to [28] for details of the derivation. The equations for c_1 and c_2 can be solved directly and we get

$$c_1 = \frac{\int_\Omega uH(\Phi)}{\int_\Omega H(\Phi)},$$
$$c_2 = \frac{\int_\Omega u(1-H(\Phi))}{\int_\Omega (1-H(\Phi))}.$$

The minimization for Φ is done using a gradient descent scheme, and we get

$$\Phi_t = -\delta_\epsilon(\Phi)\left(-\mu C_\Phi \nabla \cdot \left(\frac{\nabla\Phi}{|\nabla\Phi|}\right) + \lambda_1(u-c_1)^2 - \lambda_2(u-c_2)^2\right).$$

We want to have a closer look at the gradient flow here, which consists basically of two parts. First, we have the mean curvature flow $\nabla \cdot \left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)$, which pushes the contour inwards and acts as a penalizer for the length of Γ (or equivalently $\partial\{\Phi > 0\}$). Secondly, the flow is controlled by the two data parts $\lambda_1(u-c_1)^2 - \lambda_2(u-c_2)^2$, which aim to push the contour between both regions R_1 and R_2 and move the contour towards the required direction.

A further point of interest lies in the factor $\delta_\epsilon(\Phi)$, which causes the problem to be strongly dependent on the initial data, because the gradient flow only acts around the zero level-set $\Phi^{-1}(0)$ and the energy will often get stuck in local minima. A way out was directly proposed in the original paper [28] and consists on replacing $\delta_\epsilon(\Phi)$ by $|\nabla\Phi|$ to transform the gradient flow into the Eikonal-type partial differential equation

$$\Phi_t = -|\nabla\Phi|\left(-\mu C_\Phi \nabla \cdot \left(\frac{\nabla\Phi}{|\nabla\Phi|}\right) + \lambda_1(u-c_1)^2 - \lambda_2(u-c_2)^2\right).$$

From a segmentation point of view, the difference between both models lies in the detection of local structures. Let us consider for now the cell example from Figure 3.1. The Dirac version is usually initialized around an object and then evolves inwards, until the whole cell is enclosed. Unfortunately, the inner part of the cell is not assigned to the outer region, although the intensities fit better to the background, because the zero level-set stops at the contour and is not able to evolve inwards. On the other hand, in the Eikonal-like model, the evolution is not restricted to the zero level-set and the inner part is assigned correctly.

Proceeding to the discrete scheme, the constants c_1 and c_2 can be updated as the mean

intensities inside both regions and we get for the $(n+1)$th iteration

$$c_1^{n+1} = \frac{\sum_{i\in\{\Phi^n>0\}} u(i)}{\sum_{i\in\{\Phi^n>0\}} 1}, \qquad c_2^{n+1} = \frac{\sum_{i\in\{\Phi^n<0\}} u(i)}{\sum_{i\in\{\Phi^n<0\}} 1}.$$

The gradient flow is discretized, similar to the geodesic active contour flow, in an explicit scheme. Let therefore again D^+, D^-, D^c denote forward, backward and central difference operators. Then we have

$$\begin{aligned}
&\frac{\Phi_{i,j}^{n+1} - \Phi_{i,j}^n}{\delta_t} = \delta_\epsilon(\Phi_{i,j}^n)\left(\mu C_{\Phi^n}\left[\nabla\cdot\left(\frac{\nabla\Phi^n}{|\nabla\Phi^n|}\right)\right]_{i,j} - \lambda_1(u_{i,j}-c_1)^2 + \lambda_2(u_{i,j}-c_2)^2\right),\\
\Leftrightarrow&\Phi_{i,j}^{n+1} = \Phi_{i,j}^n + \delta_t\delta_\epsilon(\Phi_{i,j}^n)\left(\mu C_{\Phi^n}\left[\nabla\cdot\left(\frac{\nabla\Phi^n}{|\nabla\Phi^n|}\right)\right]_{i,j} - \lambda_1(u_{i,j}-c_1)^2 + \lambda_2(u_{i,j}-c_2)^2\right),
\end{aligned}$$

with

$$\nabla\cdot\left(\frac{\nabla\Phi^n}{|\nabla\Phi^n|}\right) = D_x^-\left(\frac{D_x^+\Phi^n}{\sqrt{(D_x^+\Phi^n)^2 + (D_y^c\Phi^n)^2}}\right) + D_y^-\left(\frac{D_y^+\Phi^n}{\sqrt{(D_x^c\Phi^n)^2 + (D_y^+\Phi^n)^2}}\right).$$

For the Eikonal-type model we simply replace $\delta_\epsilon(\Phi_{i,j}^n)$ with $\sqrt{(D_x^c\Phi_{i,j}^n)^2 + (D_y^c\Phi_{i,j}^n)^2}$. Finally, we need to calculate C_Φ, where the previous equivalence is used

$$C_{\Phi^n} = \int_\Omega \left|\nabla H_\epsilon(\Phi^n)\right| \mathrm{d}x = \int_\Omega \delta_\epsilon(\Phi^n)\,|\nabla\Phi^n|\,\mathrm{d}x.$$

For the smoothed Heaviside and Dirac functions, we inherit the approximation from [28], which reads

$$H_\epsilon(x) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan(\frac{x}{\epsilon})\right), \quad \delta_\epsilon(x) = H_\epsilon'(x) = \frac{1}{x}\frac{\epsilon}{\epsilon^2 + x^2}.$$

For the sake of implementation, we have to recalculate Φ in order to keep a correct distance function. For our implementation, we perform a recalculation every ten iterations. Finally, we end up with the following algorithm:

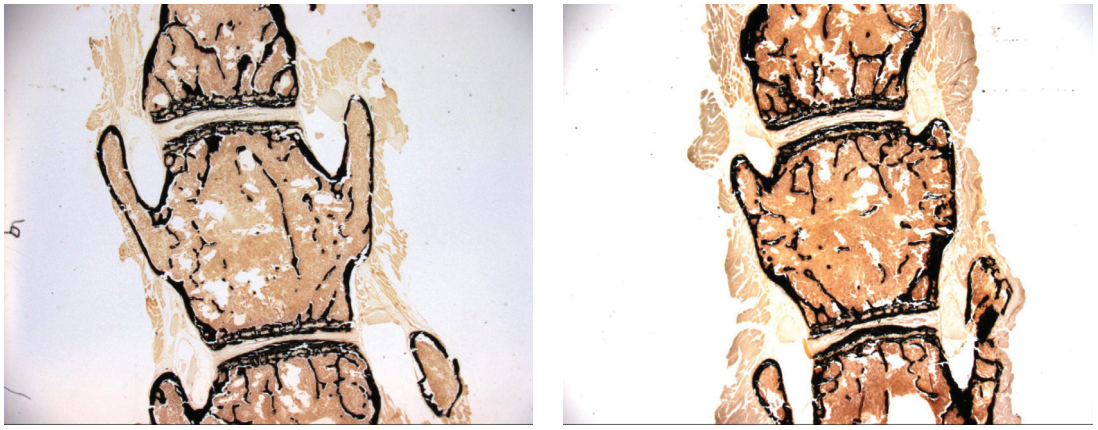


Figure 3.3.: Slices of spines from genetically altered mice, property of University Hospital Münster [71].

Algorithm 2 Chan Vese

ChanVese$\mu, \lambda_1, \lambda_2$
$\Phi \leftarrow$ initializeLevelset()
$iteration \leftarrow 0$ $iteration < maxIteration$
$meanI \leftarrow \text{mean}(\Phi < 0)$
$meanO \leftarrow \text{mean}(\Phi > 0)$
$divTerm \leftarrow$ calculateDivergence(Φ,g)
$deltaTerm \leftarrow$ calculateDiracDelta(Φ)
$areaTerm \leftarrow$ calculateArea(Φ)
$\Phi \leftarrow \Phi + \delta_t \cdot deltaTerm(\mu \cdot areaTerm \cdot divTerm + \lambda_1 \cdot meanI + \lambda_2 \cdot meanO)$
$iteration \leftarrow iteration + 1 \bmod(iteration,10) == 1$
$\Phi \leftarrow$ recorrectDistanceFunction(Φ)
Φ

3.3. Application

3.3.1. Histomorphometry

Introduction:

Histomorphometry is defined as the analysis of bone material in terms of shape, form or other parameters. In a joint project with the University Hospital Münster (UKM) we combined different image processing tools to create a semi-automatic framework for histomorphometric analysis of vertebral bodies from genetically altered mice. Figure 3.3

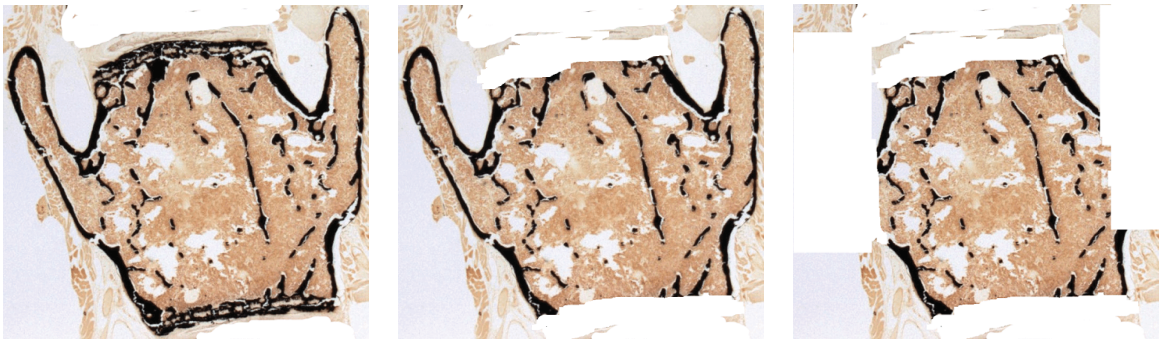


Figure 3.4.: Preprocessed vertebra image. Left: Whole vertebra, middle: Erased endplates, Right: Additionally erased transverse processes. Images are property of University Hospital Münster [71].

gives an example of typical images to be analyzed. The central point in this framework was an automated segmentation of the vertebral body in given images, which was done by hand so far.
In the following section, we will coarsely describe the developed framework and will explain how to automatically calculate a set of features from the given data.

Preprocessing:

The preprocessing is the part, which is not done automatically. First, the user has to cut out the targeted vertebral body from the image given in Figure 3.3. Next, the user has to specify endplates (bony structure at top and bottom of the vertebral body) and transverse processes (arm-like outgrowth on the left and right side). In our framework, this is done by simply adding two more images of the vertebral body to the framework, one with erased endplates, u^e, and another with additionally erased transverse processes, u^t, compare Figure 3.4. This is the starting point for our automated algorithm.

Segmentation:

The most important part in an automated histomorphometric analysis is an appropriate segmentation of the complete vertebral body. Since this body is outlined by clear and sharp edges, a **Geodesic Active Contour** algorithm, introduced in Section 3.2.1, is used. The edgemap is chosen as

$$g(x) = \frac{1}{1 + \left\|\nabla u(x)\right\|_2^2}.$$

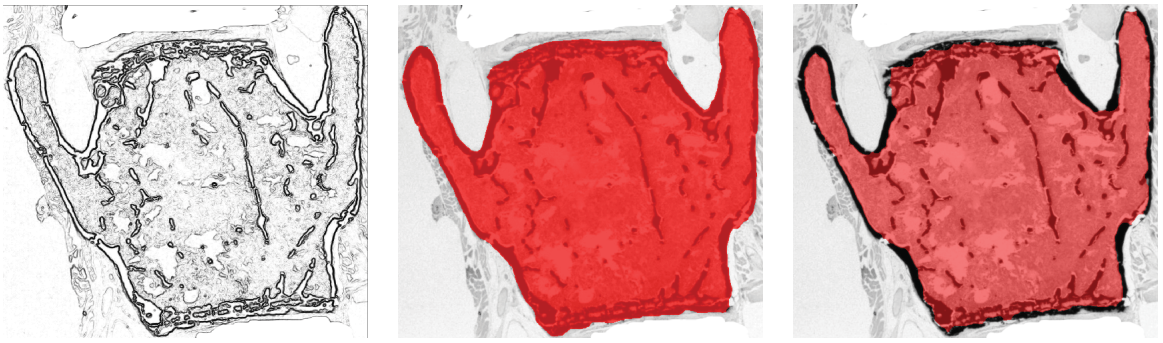


Figure 3.5.: Segmentation of the vertebral body using the geodesic active contour algorithm. Left: Edgemap, middle: Segmentation of the complete vertebral body, Right: Segmentation of the inner part. Images are property of University Hospital Münster [71]

To improve convergence speed, we use the multiscale version of the active contour algorithm, presented in Section 3.2.1. Since we are only interested in the central object, all disconnected components of $\{\Phi < 0\}$ are deleted as a postprocessing step. Therefore, a list of connected components (in terms of 4-neighborhood) for the binary image $\{\Phi < 0\}$ was created and only the largest is kept.
Figure 3.5 shows edgemap, segmentation shape and overlay segmentation for the previously introduced vertebra image.

Besides a segmentation of the whole vertebral body, also a segmentation of the inner part is required. Here, the thick layer of bone material is useful, because it creates fronts of high gradients at the outer and inner side. Consequently, we can use the slightly shifted former segmentation as an initial value for this second step.
Let Φ^{outer} be the level-set representation of the complete vertebral body. Then we set

$$\Phi_0^{inner} = \Phi^{inner}(\cdot, 0) := \Phi^{outer}(\cdot, d), \quad d \in R^-, \tag{3.7}$$

for a small value d. By this choice, the zero level-set of the outer segmentation is shifted inwards by a distance of d and is then used as an initial value for the second segmentation. An example for the result can be found in Figure 3.5.

Feature Extraction:

In this section we want to shortly summarize how different features can be extracted from the images resp. segmentation.

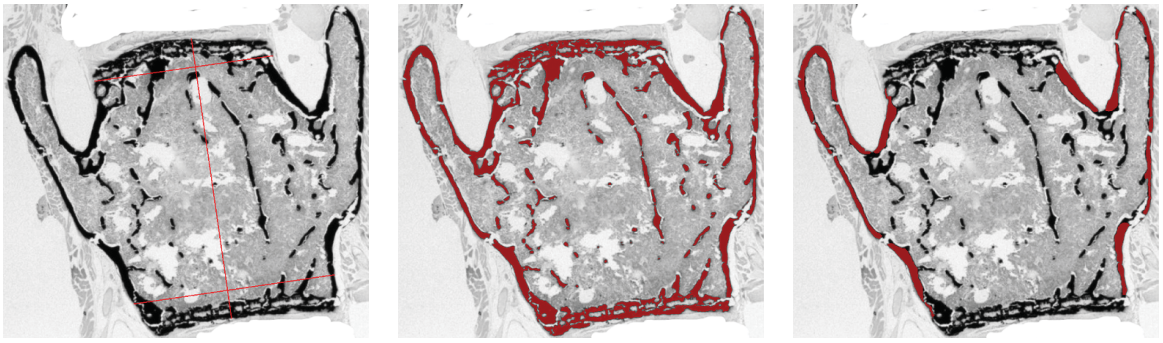


Figure 3.6.: Feature extraction from vertebral body. Left: Width and height, middle: Calcified area and width, Right: Corticalis area. Images are property of University Hospital Münster [71].

Area:

The area can be calculated as

$$area = \int_{\Omega} \mathbb{1}_{\{\Phi<0\}} \, \mathrm{d}x.$$

Perimeter:

Since the perimeter is the length of the zero level-set $\Gamma = \Phi^{-1}(0)$, we inherit the previous results (see Equation (3.6)) and have

$$per = |\Gamma| = \int_{\Omega} |\nabla \Phi| \, \mathrm{d}\delta(\Phi).$$

For the discrete setting, the Dirac delta is approximated by

$$\delta_\epsilon(x) = \frac{1}{x} \frac{\epsilon}{\epsilon^2 + x^2},$$

where $\epsilon = 3$ led to good results in our experiments. Central differences were used to approximate the gradient. This leads to the discrete formula

$$per = |\Gamma| \approx \sum_{(i,j)} \delta_\epsilon(\Phi_{i,j}) \sqrt{(D_x^c \Phi_{i,j})^2 + (D_y^c \Phi_{i,j})^2}.$$

Height and Width:

The calculation of the height of the vertebral body is complicated by the fact that the object does not lie straight in the image, but is tilted by some degrees. Let us assume that we approximately know the angle $\alpha \in [\theta_1, \theta_2]$.

We want to underline here that the definition of **height** is not directly clear from the images. Hence, we need to construct a robust and repeatable method to calculate this feature. For this sake, we took the center of mass of the segmentation and defined the height as the length of the line, going through the center of mass and following the orientation of the vertebral body.
As a start, we have to calculate the center of mass $\boldsymbol{m} = (m_1, m_2)$ of the binary image $\{\Phi < 0\}$. The center of mass can be calculated in each dimension independently. For the first dimension we have

$$m_1 = \frac{1}{area} \int_\Omega x_1 \cdot \mathbb{1}_{\Phi(x_1)<0} \, \mathrm{d}x_1.$$

One way to calculate the objects height is to take the Radon transform (with zero attenuation) of the binary image $\{\Phi < 0\}$, restricted to the interval $[\theta_1, \theta_2]$, denoted by $R(\{\Phi < 0\})_{[\theta_1,\theta_2]}$. This Radon transform now consists of all line integrals (or equivalently diameters) and we extract those, which go through the center of mass $\boldsymbol{m}$ and choose the minimal value of these integrals. The result can be found in Figure 3.6.
The width of a vertebral body at top and bottom is of interest. Therefore the length of orthogonal lines ten percent above the starting point and below the endpoint of the height is measured.

Calcified Area:

The calcified area is characterized by the dark color (i.e. lower intensity) in the observed image. The calcified area can then be extracted as an integral over all intensities inside the vertebral body that are below some threshold t_c, hence

$$calcifiedArea = \int_\Omega \mathbb{1}_{\Phi(x)<0} \cdot \mathbb{1}_{I(x)<t_c} \, \mathrm{d}x.$$

Cortical Area and Thickness:

The cortical area is defined as the part of the calcified area that represents the outer hull of the vertebral body. At this point, we have to treat the endplates separately because they are not part of the cortical area.
For the calculation, we need the outer hull first, which can be calculated as the intersection of the complete vertebral segmentation $\{\Phi^o < 0\}$ and the outside of the inner segmentation $\{\Phi^i > 0\}$. As already mentioned in the beginning, the user should make a second image available that contains no endplates, u^e. This image is used in the

thresholding step, hence

$$corticalArea = \int_{\Omega} \mathbb{1}_{\Phi^i(x)>0} \cdot \mathbb{1}_{\Phi^o(x)<0} \cdot \mathbb{1}_{I_e(x)<t_c} \, \mathrm{d}x.$$

Besides the cortical area, the average cortical thickness was a feature requested by our collaborators from the university hospital. Here, the level-set representation of the outer segmentation is useful again. First, we extract the zero level-set of the outer segmentation (see Equation (3.7)), which is part of the cortical area

$$\Gamma_{cort} = \left\{ x : x \in \Phi^{-1}(0), I_e(x) < t_c \right\}.$$

On this part of the contour, we are now able to calculate the negative normal $-\vec{n}$ (the normal on a level-set is usually given by $\nabla\Phi(x)$). Both, point x and $-\vec{n}$ form a linear function, whose intersection with the **Cortical Area** measures the thickness at this point. Averaging over all those lines gives the requested information.

Trabeculae:

Trabeculae are the small bone-structures inside the vertebral body. Different information concerning them is of interest, starting with the total number of trabeculae. Since trabeculae are structures inside, we use the inner segmentation Φ^i and identify the calcified structure using the previous threshold t_c. This yields the set

$$trabeculaeArea = \left\{ x : \Phi^i(x) < 0, I_e(x) < t_c \right\}.$$

This set is automatically divided into connected components $t_1, \ldots t_n$, using a 8-neighborhood (the 8 adjacent pixels in the 3d space). For the n trabeculae, we can now calculate their average size. Of further interest is the neighborhood of each trabecula. For a distance d_n, we calculated the average number of neighbors, which have a distance less than d_n and are in the line sight, for each trabecula. This information gives insight into the distribution of trabeculae within the vertebral body.

HistoGUI

The HistoGUI semi-automatic framework for Matlab® can be downloaded for free at http://wwwmath.uni-muenster.de/u/hendrik.dirks/downloads/HistoGUI.zip. Figure 3.7 gives an insight about the general structure of the HistoGUI. The program consists of four steps, beginning with a loading option for the image to be analyzed. Afterwards, the user has to create segmentations of the vertebral body with and with-

out calcified hull. For this purpose, all parameters of the algorithm can be adjusted to fit a broad set of images. In a third step, different characteristics of the underlying vertebra can be calculated. To create accurate results, all measurements can be adjusted. To finally analyze the trabecular structure, a respective tool constitutes the last step. The trabeculae are semi-automatically detected. Afterwards, the user can remove and/or combine the them and calculate different characteristics.

3.3.2. Intracellular Flow

Nowadays, cell-tracking has become very popular in molecular biology and medicine. Here, image sequences of moving cells are captured with microscopes and afterwards, ideally automatically, analyzed. The goal is to gather information about the behavior of cell systems or individual cells. But the focus goes already one level deeper. Using high resolution microscopy in combination with staining techniques, we can look inside a cell during its movement. A special type of movement is the so-called *blebbing*, where intracellular pressure causes the membrane to bulge outwards [17, 29, 35]. Afterwards, cytoplasma flows through the membrane forming the **bleb**. The process of blebbing is not very well understood, especially how different cell-components interact during the bleb. There also exists the hypothesis that certain cell-components generate pressure to force the bleb. Aiming to answer these questions, biologists generate series of high resolution images, where usually the cytoplasma and one other cell-component are colored In the analysis of these sequences, image processing tools come into play. We created an automatic framework for series of images, starting with preprocessing, segmentation, motion estimation and ending with the analysis of the resulting data (evaluation of motion field, cell-shape etc.).

Preprocessing

The recorded raw data comes as one *.lsm* file, which can be imported into Matlab using an improved tiffread function (developed by Francois Nedelec, see [69]). The file contains images, usually in two channels, recorded at timesteps $t_1 \ldots, t_n$. One of the channel contains images of the colored cytoplasma, whereas the other carries information about a certain cell-component. Due to the dynamic character of the experiments, there is only a short illumination time for each image, which causes the recorded images to be noisy. Consequently, the images have to be preprocessed to reduce background noise, enhance the contours and make them suitable for segmentation and motion estimation. In this application, the images have been smoothed by convolution with

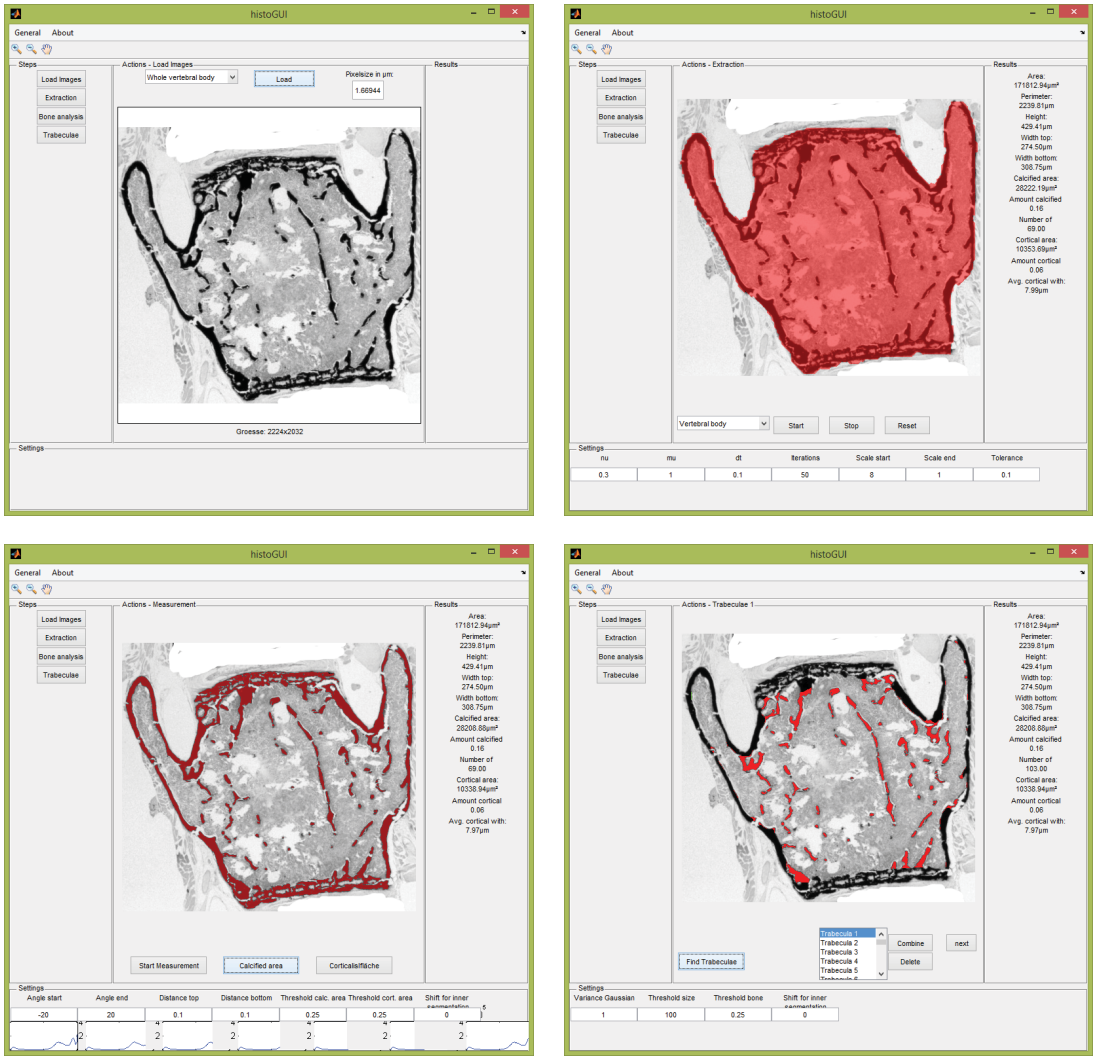


Figure 3.7.: Overview of the basic HistoGUI features. Top-Left: Load images, Top-Right: Segmentation window, Bottom-left: Feature extraction, Bottom-right: Trabeculae analysis. Images are property of University Hospital Münster [71].

a Gaussian filter. An example for images from both channels, before and after the preprocessing step, can be found in Figure 3.8.

Segmentation

To obtain a reliable segmentation of the cell and the cell-component inside, we combine a set of different segmentation algorithms and postprocessing steps. To generate a coarse segmentation, we use a thresholding algorithm on the green channel:

$$cell^{tr}_{inner} = \mathbb{1}_{I^n(x)<t},$$

with $t = 0.1$ in our framework. The boundary set $\Gamma(cell^{tr}_{inner})$ is then used as initial contour for a Chan-Vese algorithm (see Section 3.2.2 for details)

$$cell^{cv}_{inner} = chanVese(\Gamma(cell^{tr}_{inner}), \lambda_1 = 400, \lambda_1 = 200, \mu = 0.1, \nu = 0).$$

Since the Chan-Vese algorithm does not incorporate any local information, the result might contain additional unwanted objects or regions. Moreover, the inside of the cell is often assigned to the background region. To improve the segmentation, we apply a size-filter that removes all inner connected components, except for the largest (which should be the cell in our case), followed by a closing tool which removes all holes inside the cell. Afterwards, a Gaussian filter is applied to smooth the segmentation and achieve a nicer looking shape. Figure 3.9 shows an example of an image of each step in the framework. Now that the complete cell (green channel) has been segmented, the segmentation of the cell-component (red channel) is still remaining. Therefore, we use the result of the green channel segmentation as an initialization for the red channel. We start directly with an application of the Chan-Vese algorithm in combination with the postprocessing steps from before. The result can be found in Figure 3.9.
For what follows, let us denote Φ^{outer} as the level-set representation of the green-channel segmentation and Φ^{inner} as the level-set representation of the red-channel segmentation.

Motion Estimation

The next step in the framework, characterizing intracellular flow, is the estimation of underlying motion. Our goal is to estimate motion of the entire cell on the one hand, and of the intracellular component on the other hand. For both estimation processes, we use a standard L^2-L^2 optical flow model, which consists of minimizing the following

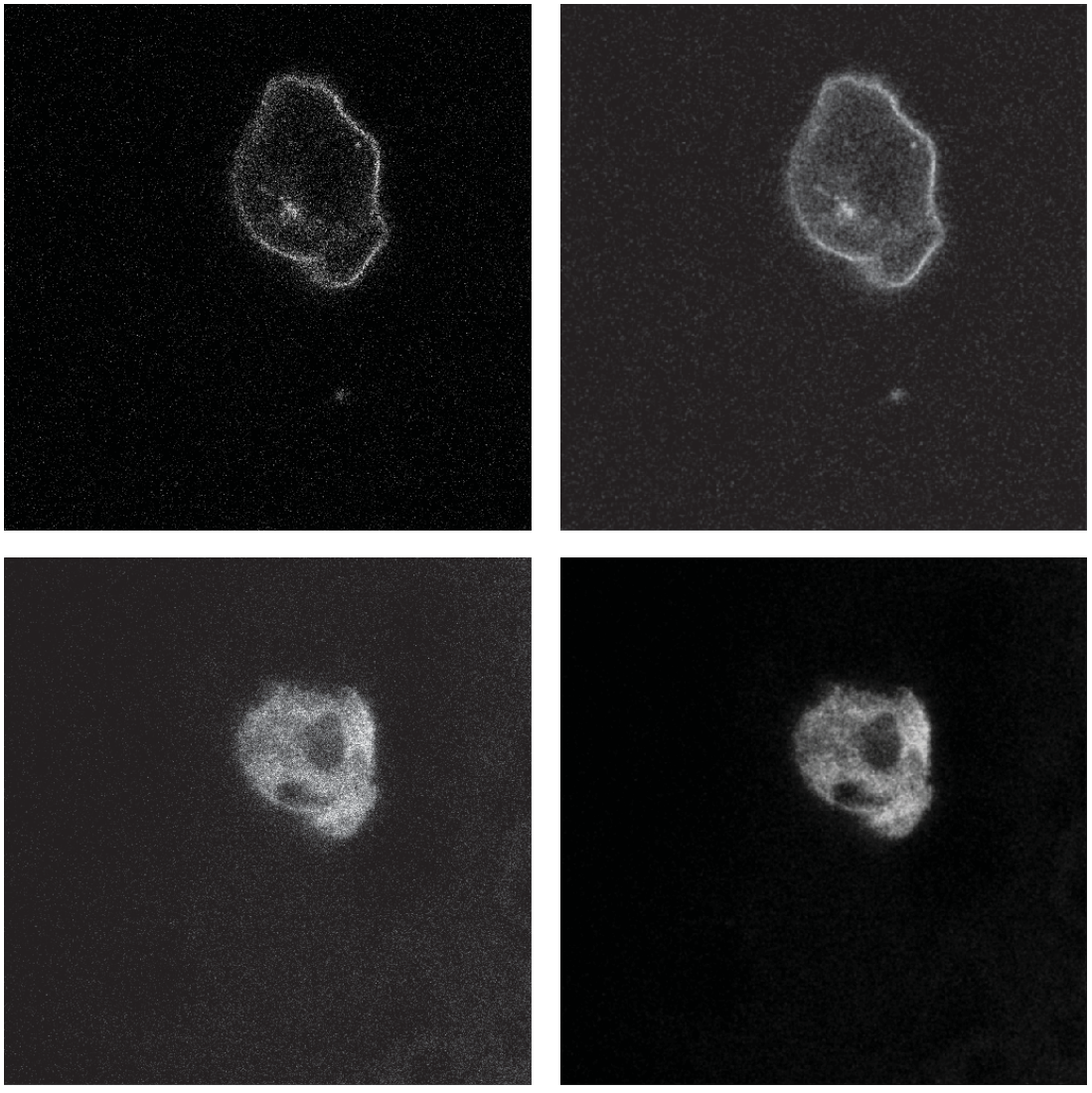


Figure 3.8.: Preprocessing using a Gaussian filter. Top-Left: Green channel raw, Top-Right: Green channel smoothed, Bottom-left: Red channel raw, Bottom-right: Red channel smoothed. Images are property of Raz group from ZMBE [79].

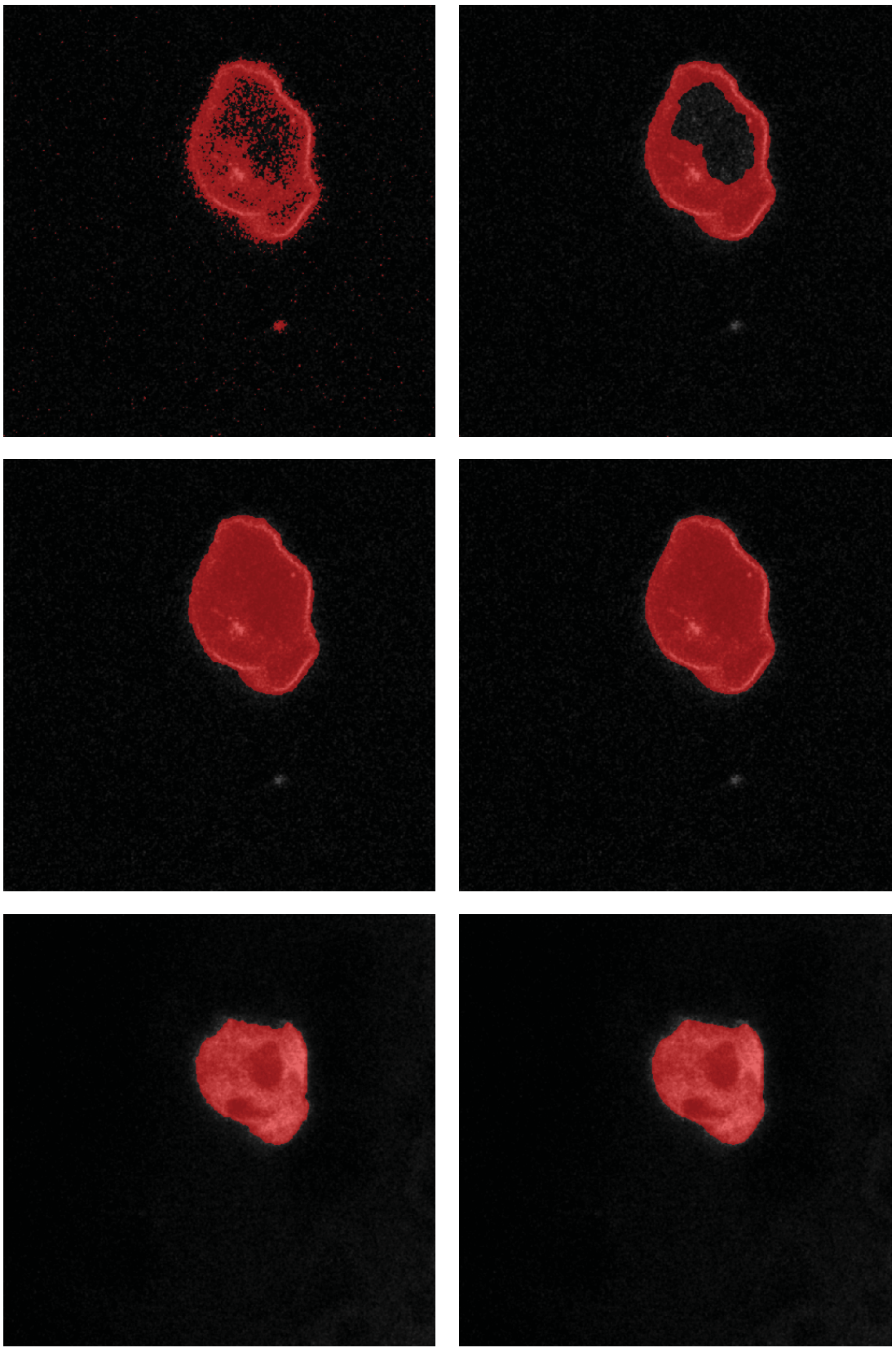


Figure 3.9.: Step-by-step segmentation. Top-Left: After thresholding, Top-Right: After Chan-Vese, Middle-left: Holes closed, Middle-right: Smoothed contour, Bottom-left: Red channel after Chan-Vese, Bottom-right: Red channel after smoothing. Images are property of Raz group from ZMBE [79].

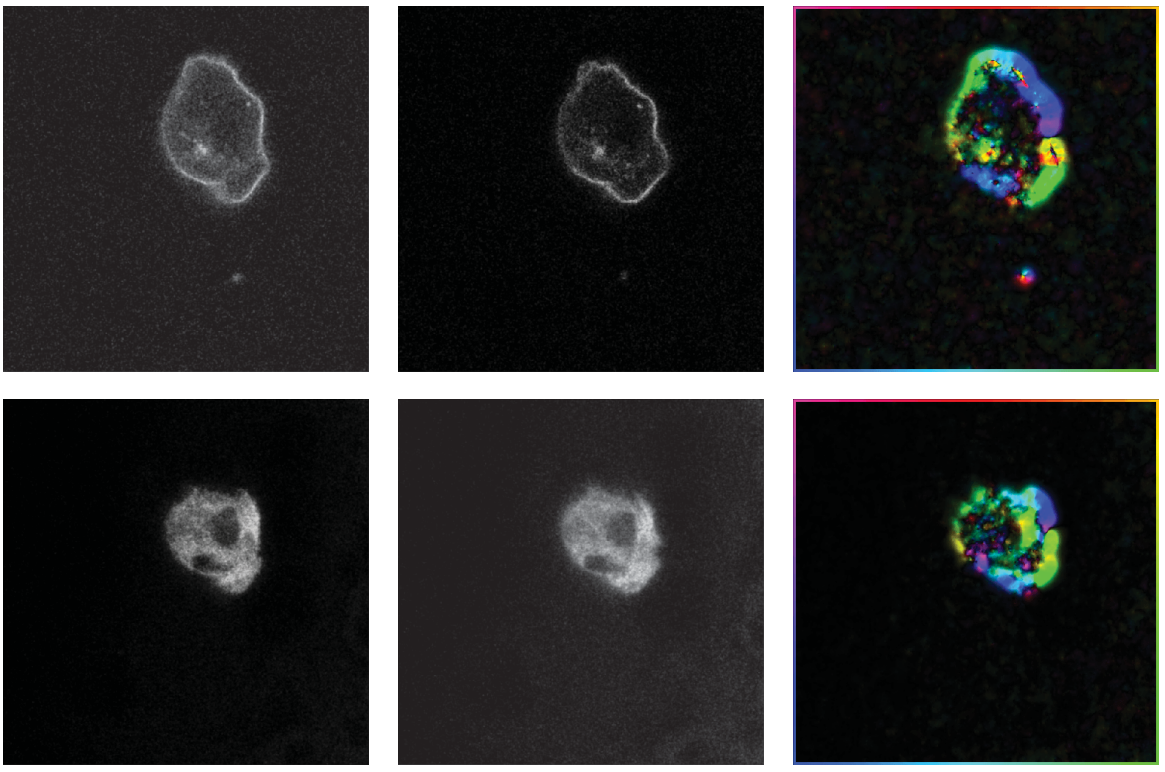


Figure 3.10.: Consecutive images and estimated motion between them. Top-Left: Green channel in image 1, Top-Middle: Green channel in image 2, Top-Right: Estimated motion field, Bottom-Left: Red channel in image 1, Bottom-Middle: Red channel in image 2, Bottom-Right: Estimated motion field. Images are property of Raz group from ZMBE [79].

variational model

$$\arg\min_{\boldsymbol{v}} \frac{1}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 + \frac{\alpha}{2} \sum_{i=1}^{d} \|\nabla v_i\|_2^2 ,$$

where u represents the underlying image sequence and $\boldsymbol{v}$ the, yet unknown, velocity field. We refer to Section 4.5.2 for a detailed explanation of the model, data term and regularizer. Since noise strongly influences motion estimation, the smoothed images (of both channels) from before are used. The result can be found in Figure 3.10. Unfortunately, the algorithm still detects motion in the background and, moreover, it is unclear whether the preprocessing step destroys useful motion information. This is why we propose a joint model for motion estimation and image reconstruction in Chapter 5.

Bleb Detection

Having calculated reliable segmentations and motion fields, we are able to extract at which timesteps blebs occur and of which size they are. We begin with an algorithm for automated bleb detection. A bleb can coarsely be characterized by two phases. Blebbing begins when the membrane bulges outwards, which creates a gap between membrane and the inside of the cell. A fully formed bleb is then characterized by a hump on the, usually roundish, cell shape. These criteria can be used to automatically detect blebs.

Phase 1:

To detect the gap between membrane and inner part, we can use the segmentation of cell and inner component. The region between both segmentations can be specified as:

$$\Omega_{gap} := \left\{ x : \Phi^{outer}(x) < 0, \Phi^{inner}(x) > 0 \right\}$$

We can measure the distance by solving the Eikonal equation (speed 1) with initial value equal to the zero level-set of both segmentations

$$\Phi_t = \|\nabla \Phi\|, \quad \Phi(x,t) = 0 \text{ for } x \in \Gamma^i \cup \Gamma^o,$$

where Γ^i and Γ^o represent the zero level-set of the segmentation of the green resp. the red channel. The solution Φ^{gap} measures the distance between the membrane and the inner part of the cell. To detect a bleb, we have to specify a critical threshold value d^{gap} and define the set of possible blebs as

$$\Omega_{bleb1} := \left\{ x : x \in \Omega_{gap}, \Phi^{gap}(x) > d^{gap} \right\}.$$

By this choice, we will detect each point which has a distance above the critical distance d^{gap}. Let us now divide Ω_{bleb1} into connected components (4-neighborhood) $\Omega^1_{bleb1}, \ldots, \Omega^n_{bleb1}$. Due to errors in the segmentation and discontinuities in the distribution of the colored medium, we can justify the deletion of components below a certain size and end up with the set of blebs. The result can be found in Figure 3.11.

Phase 2:

Due to the fact that the evolution of blebs happens in a very short timescale, the microscope sometimes records an image when the bleb has fully formed. To detect these blebs of a more evolved state, we can use the roundness of the cell which is violated due to the hump on the surface of the cell. To detect those humps, we start with the binary segmentation of the green channel $\left\{ \Phi^{outer} < 0 \right\}$ and calculate the maximum inscribed

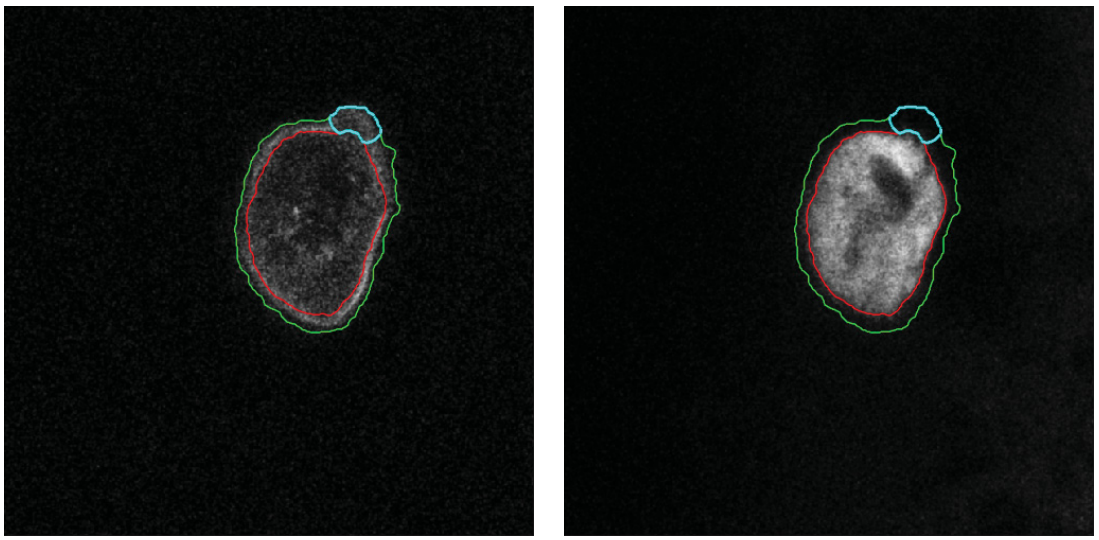


Figure 3.11.: Detected bleb in phase 1, overlay of the zero level set of the complete cell (green), the inner part (red) and the bleb (cyan). Left: Green channel, Right: Red channel. Images are property of Raz group from ZMBE [79].

circle. This is the circle with the maximal diameter inside an object, characterized by position (c_x, c_y) and diameter r. An efficient $n\log(n)$ algorithm can be found in [5]. We found that extending the diameter r by a factor of 1.2 ($r^* := r \cdot 1.2$) usually covers the whole cell. Hence, detecting a larger part of the cell outside the circle, we might have found a further formed bleb and define:

$$\Omega^2_{bleb2} := \left\{ x = (x_1, x_2) : \sqrt{(x_1 - c_x)^2 + (x_2 - c_y)^2} > r^* \right\}.$$

In some cases, the cell is not perfectly round or the segmentation is not perfectly aligned with the cells outer hull. For these cases, even the extended maximum inscribed circle might not cover a *normal* cell and Ω^2_{bleb2} contains a thin slice of the outer hull. To overcome this problem we start shrinking Ω^2_{bleb2} by a diameter $d_{shrink} \in \mathbb{R}^+$, which removes everything with a diameter below d_{shrink}. We start by first solving

$$\Phi_t = \|\nabla\Phi\|, \quad \Phi(x,t) = 0 \text{ for } x \in \Gamma(\Omega^2_{bleb2}),$$

and then extracting from the solution $\hat{\Phi}$

$$\hat{\Omega}^2_{bleb2} := \left\{ x : \hat{\Phi}(x) < -d_{shrink} \right\}.$$

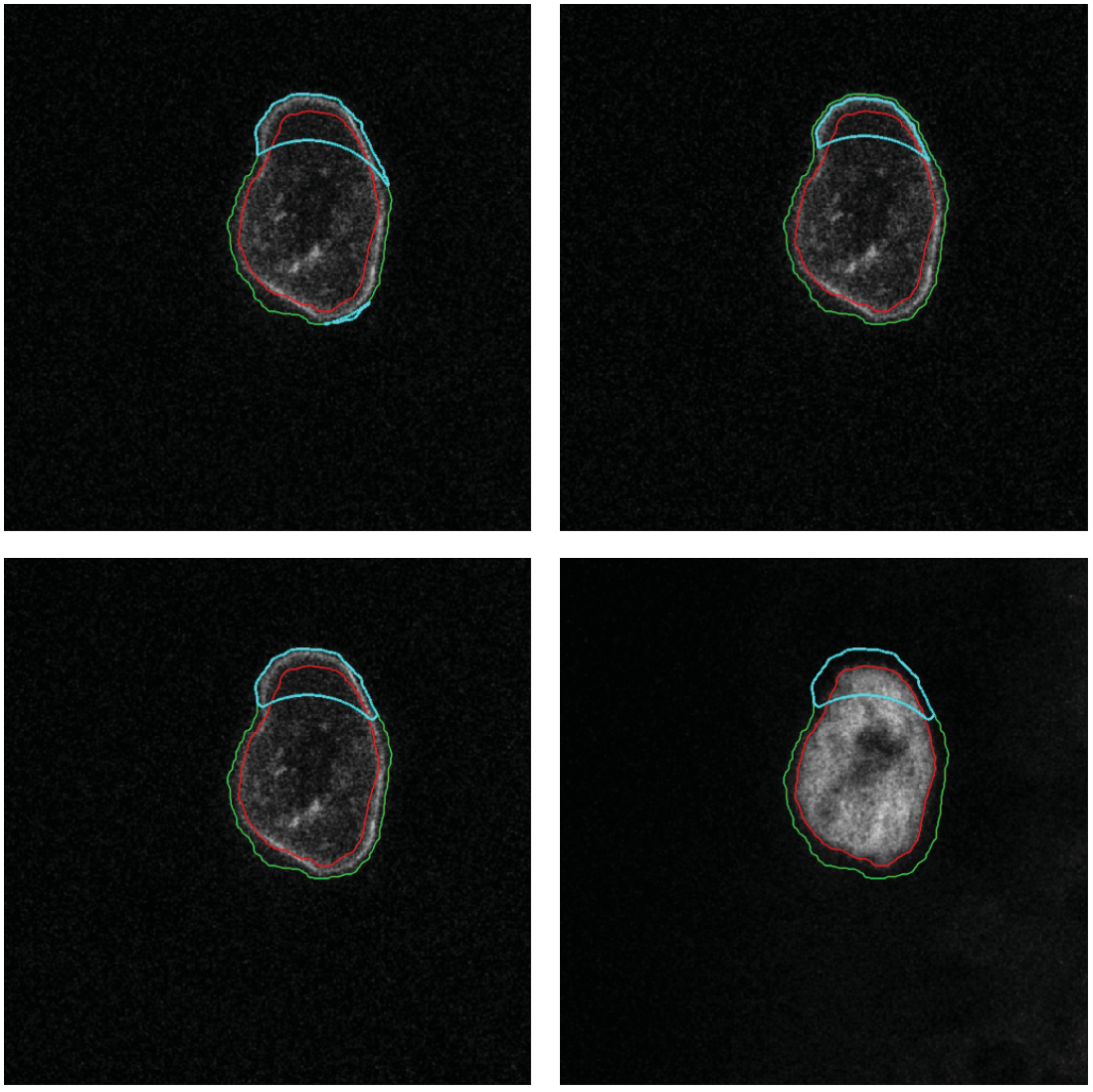


Figure 3.12.: Detected bleb in phase 2, overlay of the complete cell (green), the inner part (red) and the bleb (cyan). Top-Left: Blebbs found with maximum inscribed circle criterion, Top-Right: Blebbs after shrinking by diameter $d_{shrink} = 5$, Bottom-Left: Result with green channel in background, Bottom-Right: Result with red channel in background. Images are property of Raz group from ZMBE [79].

This region is afterwards blown up by first solving

$$\Phi_t = \|\nabla\Phi\|, \quad \Phi(x,t) = 0 \text{ for } x \in \Gamma(\hat{\Omega}^2_{bleb2}),$$

and then extracting from the solution $\bar{\Phi}$

$$\bar{\Omega}^2_{bleb2} := \left\{x : \bar{\Phi}(x) < d_{shrink}\right\}.$$

Similar to phase one we now remove all components below a certain size and end up with the set of blebs. A step-by-step application of the phase 2 detection can be found in Figure 3.12.

Conclusion

Together with the automatically detected blebs, the calculated data, consisting of cell-segmentation (which can be used for tracking), and local motion information of both channels can now be analyzed by biologists to better understand the process of blebbing. Again, we want to point the readers attention towards an alternative approach for joint image denoising and motion estimation introduced in Chapter 5 and applied to a concrete dataset in Chapter 6.

4 Motion Estimation

In this chapter we will introduce different models for motion estimation in image processing. Having in mind image sequences in general and biological data recorded by a microscope in particular, we will discuss drawbacks and advantages of different variational models.
In the beginning of this chapter we give a short introduction to the topic of motion estimation from image data. From different assumptions for the underlying flow field we derive the optical flow constraint and the mass preservation constraint which are then used to define variational motion estimation models. Different a-priori information is integrated in the model via a weighted regularization term. We present efficient strategies for the minimization of the underlying energy, based on the popular primal-dual framework of Chambolle and Pock (see Section 2.3.1 resp. [26]).
The proposed models are evaluated in terms of flow-adapted error measures, runtime and number of iterations. The spectrum of observed datasets contains basic flow types, influence of noise and scenes from the IPOL optical flow database [44].

4.1. Introduction

Motion estimation generally arises in the context of image sequences $u(x,t)$, depending on a spatial position $x \in \mathbb{R}^d$ and a time $t \in [0,T]$. For real applications there exists only the discrete counterpart of u, which is a set of images recorded at time steps $t_0, t_0+\delta_t, t_0+2\delta_t, \ldots$ (see Figure 4.1). There also exist a variety of characteristics we have to consider when estimating motion:

- A digital image can suffer from low resolution, low contrast, different gray levels

and noise.

- The temporal resolution δ_t is strongly connected to the underlying motion. For too large time steps we might loose correspondence between consecutive images (e.g. a very fast car might only be visible in one image).

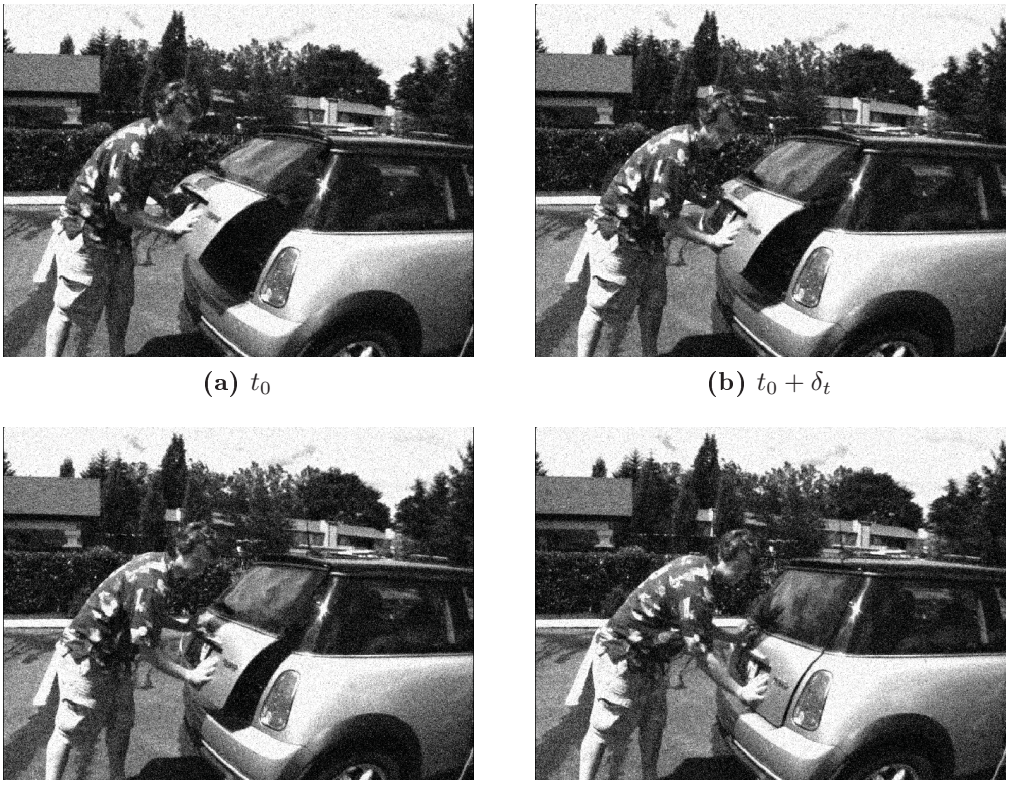
- A natural image often contains a set of moving objects with different speeds. A sufficient model should simultaneously be able to detect small and large movements in the same sequence. On the other hand, for the static background no motion should be detected.

- For many biological applications, we have to consider the fact that the illumination is constant, but fluorescence of the observed object can be inhomogeneous in space, or might even underlie changes over time.

- Finally, we have to consider that a camera only detects 2-dimensional images of a 3-dimensional world. Reconstructing the missing depth information from images is also a current task in image processing [64].

4.1.1. Optical Flow and Real Motion

When looking at image sequences and moving objects we directly speak of motion. This is a false implication since only projections of the real 3-dimensional motion fields are recorded by an image recording device (and in particular the human eye).
To emphasize this fact, we have a closer look at Figure 4.2. From the optical center O we are only able to follow a 2-dimensional path $\boldsymbol{v}_p = (v^1, v^2)$ on the image domain Ω, which is the projection of the real 3-dimensional path $\boldsymbol{v}_r = (v^1, v^2, v^3)$. Thus, already one degree of information gets lost here. This problem is even worse since we are not able to measure the 2-dimensional motion field directly. On images only the apparent motion (or *optical flow*) is visible, that is displacements of intensities. Unfortunately the apparent motion and the 2-dimensional motion field are two fundamentally different properties (a detailed discussion of this problem can be found in [72]). Barbert's pole example (Figure 4.3) is often used to underline this difference. The pole simply rotates counterclockwise and the real motion field basically points to the right side (a). Unfortunately optical devices (e.g. camera, eye) can only detect gray values tending upwards and consequently the optical flow points upwards (b).



(a) t_0 **(b)** $t_0 + \delta_t$

(c) $t_0 + 2\delta_t$ **(d)** $t_0 + 3\delta_t$

Figure 4.1.: The image sequence "Mini Cooper" from the Middlebury optical flow database [10] shows a sequence of digital images. The main characteristics are different ranges of motion, noise and a static background.

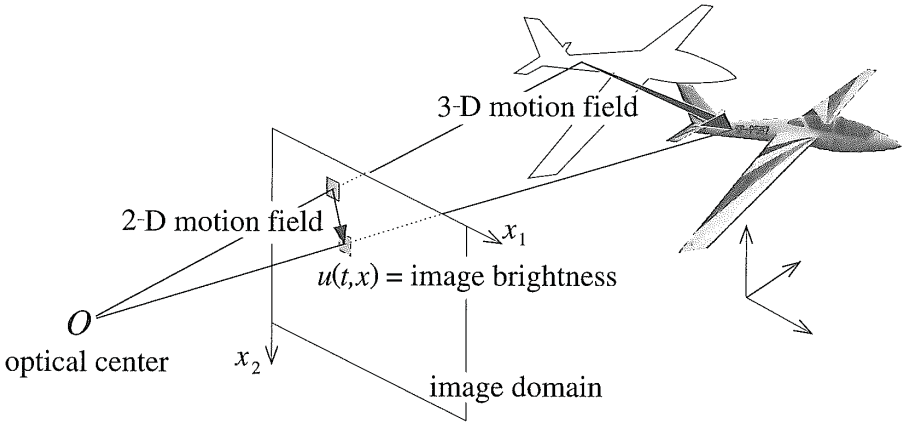


Figure 4.2.: A simplified illustration of a 2D and 3D motion field to emphasize the difference between real motion and recorded motion by a camera (from [8]).

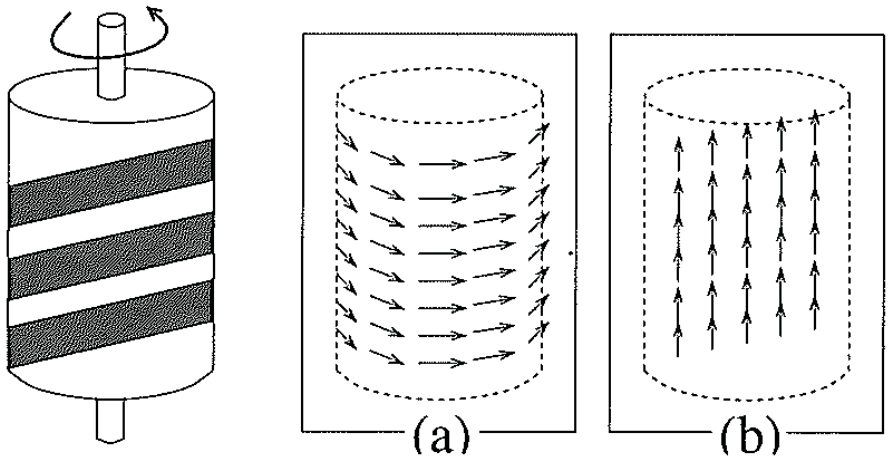


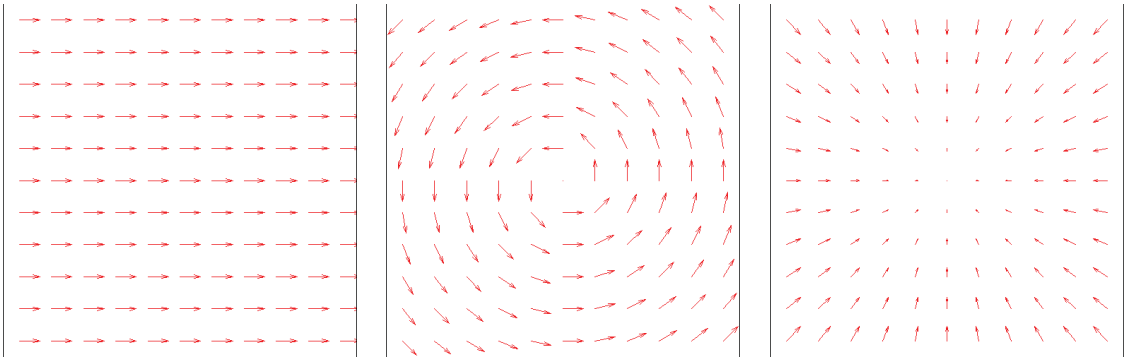
Figure 4.3.: Barbert's pole example to underline the difference between apparent motion and real motion (from [8]).

4.1.2. Mathematical Context

In this Section let Ω bet the image domain and $[0,T]$ the observed time interval. Let furthermore $u(x,t) : \Omega \times [0,T] \to [0,1]$ be a measurable image function depending on spatial position and time. For most applications we have $\Omega \subset \mathbb{R}^2$. Then the resulting velocity field is $\boldsymbol{v} = (v^1, v^2) : \mathbb{R}^2 \to \mathbb{R}^2$. The velocity field $\boldsymbol{v}$ has two components and therefore assigns a vector to every point (x,t) in the space/time domain.
The minimization process of the models introduced in the following is always performed in the finite dimensional setting. This is a natural assumption because our goal is to extract motion from image sequences which are defined on regular Cartesian grids.
We furthermore introduce three basic types of motion — translation, rotation and scaling (see Figure 4.4). Most of the resulting velocity fields are combinations of these three basic types.

4.2. Optical Flow Constraint

One of the most common techniques to formally link intensity variations and motion is the optical flow constraint. We consider images at time steps t and $t + \delta_t$ (δ_t small) and follow the common assumption that constant intensities follow a displacement field $\boldsymbol{v}(x,t)$ (brightness constancy assumption) from the first image to the second. This can



(a) Translation **(b)** Rotation **(c)** Scaling

Figure 4.4.: The three basic types of motion.

be formalized as

$$u(x,t) = u(x + \delta_t \boldsymbol{v}(x,t), t + \delta_t)$$
$$\underbrace{=}_{Taylor} u(x,t) + \delta_t \boldsymbol{v}(x,t) \cdot \nabla u(x,t) + \delta_t u_t(x,t) + O(\delta_t^2).$$

Now subtracting $u(x,t)$ on both sides and dividing by δ_t we get

$$0 = \boldsymbol{v}(x,t) \cdot \nabla u(x,t) + u_t(x,t) + O(\delta_t).$$

For small time steps δ_t we can ignore the remainder and conclude with the so-called **optical flow constraint**

$$\boldsymbol{v} \cdot \nabla u + u_t = 0. \tag{4.1}$$

Unfortunately this scalar equation is insufficient to find both unknown components v^1, v^2 of the velocity field. Hence, an infinite number of solutions exist and the problem is ill-posed [36]. This problem is called the **aperture problem**.

However, there is still some information hidden in (4.1). All possible solutions are components of the direction of ∇u (see Figure 4.5) and are normal to the isophotes (levels of constant brightness) of the image. Calculating the absolute value of the normal velocity $\|\boldsymbol{v}\|_n$ is well defined for $\|\nabla u\| \neq 0$ and we get

$$\|\boldsymbol{v}\|_n = \frac{\|u_t\|}{\|\nabla u\|}. \tag{4.2}$$

Concluding this Section we want to mention that the aperture problem mainly arises

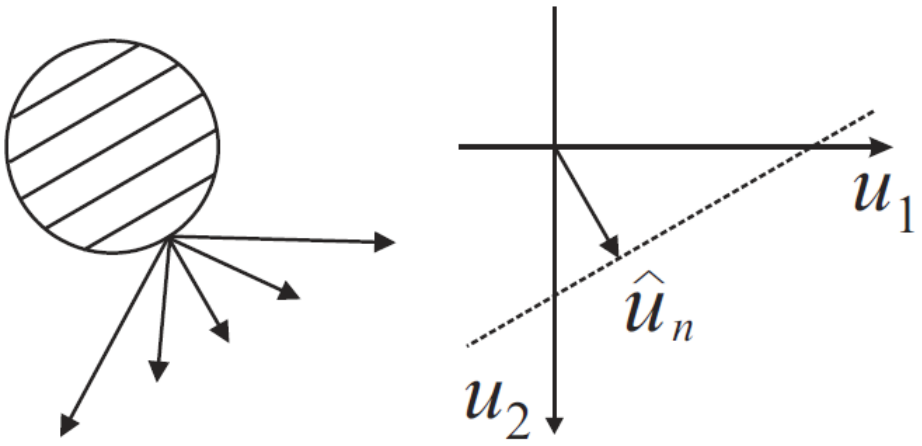


Figure 4.5.: When looking at a crate through a circular aperture, all velocities along a line in the velocity space are consistent with the optical flow (from [4]).

due to the locality of the optical flow equation. In the variational motion estimation in Section 4.5 we aim to solve the underdetermined system by adding additional a-priori information.

4.3. Mass Preservation Constraint

Our aim in this part is to link the physical idea of a preserving quantity to images and underlying motion. The resulting continuity equation represents, in contrast to the optical flow equation, an alternative approach to derive flow information from local intensity variations.

We start with the observation that mass, energy and momentum are preserved physical quantities. Carried to the field of image processing we assume that total brightness in the image is constant for every time step, thus

$$\int_\Omega u(x,t)\,\mathrm{d}x = C \quad \forall t \in [0,T]\,.$$

For real applications this assumption makes absolutely sense. For example in photon emission tomography, a certain amount of tracer liquid is injected into the patient at time $t_0 = 0$. This tracer distributes through the body in time, but the total amount is constant in each time step.

Besides this global conservation law, we furthermore enforce a local form of conservation. Mass cannot be created, destroyed or teleported from one place to another. Mass can only be moved by a continuous flow $\boldsymbol{v}$. Transfered to formula, consider a subset $\mathcal{S} \subset \Omega$.

The variation of mass is given by the integral of the time derivative and we have

$$\frac{d}{dt}\int_{\mathcal{S}} u(x,t)\,\mathrm{d}x = \int_{\mathcal{S}} \frac{\partial}{\partial t}u(x,t)\,\mathrm{d}x. \tag{4.3}$$

Going back to transport by a continuous flow $\boldsymbol{v}$ we already know that for every $x \in \Omega$ the total flow in this point is given by $u\boldsymbol{v}$. It follows that the flow of mass towards the boundary $\partial\mathcal{S}$ is given by $u\boldsymbol{v}\cdot n$ (with normal vector n to $\mathrm{d}S$). Hence integrating over $\partial\mathcal{S}$ yields the total mass flow towards the boundary:

$$\int_{\partial\mathcal{S}} u\boldsymbol{v}\cdot n\,\mathrm{d}S. \tag{4.4}$$

Since we already argued that mass can only be moved by a continuous flow and the time-variation of mass in $\mathcal{S}$ (4.3) must equal the total mass flow $\partial\mathcal{S}$ (4.4), we get

$$\int_{\mathcal{S}} \frac{\partial}{\partial t}u\,\mathrm{d}x \overset{!}{=} -\int_{\partial\mathcal{S}} u\boldsymbol{v}\cdot n\,\mathrm{d}S \overset{Gauss}{=} -\int_{\mathcal{S}} \boldsymbol{v}\nabla u - \int_{\mathcal{S}} u\nabla\cdot\boldsymbol{v}\,\mathrm{d}x = -\int_{\mathcal{S}} \nabla\cdot(u\boldsymbol{v})\,\mathrm{d}x. \tag{4.5}$$

Now combining both sides yields

$$\int_{\mathcal{S}} \frac{\partial}{\partial t}u + \nabla\cdot(u\boldsymbol{v})\,\mathrm{d}x = 0. \tag{4.6}$$

Since (4.6) holds for arbitrary $\mathcal{S} \in \Omega$ we obtain the **mass preservation constraint**

$$\frac{\partial}{\partial t}u + \nabla\cdot(u\boldsymbol{v}) = 0. \tag{4.7}$$

This equation is generally known as **continuity equation** in the literature.

4.4. Comparison Between Optical Flow and Mass Preservation

We would like to emphasize the difference between the **optical flow constraint** $\partial_t u + \nabla u\cdot\boldsymbol{v} = 0$ and the **mass preservation constraint** $\partial_t u + \nabla\cdot(u\boldsymbol{v}) = 0$. Writing out the latter one

$$0 = \partial_t u + \nabla\cdot(u\boldsymbol{v}) = \underbrace{\partial_t u + \nabla u\cdot\boldsymbol{v}}_{Optical\,Flow} + u\nabla\cdot\boldsymbol{v},$$

we see that the mass preservation equation is basically an extension of the optical flow equation. We no longer claim

$$\partial_t u + \nabla u \cdot \boldsymbol{v} = 0,$$

but now allow

$$\partial_t u + \nabla u \cdot \boldsymbol{v} = -u \nabla \cdot \boldsymbol{v}.$$

We are furthermore able to directly see that divergence-free flow fields (i.e. $\nabla \cdot \boldsymbol{v} = 0$) that fulfill the mass preservation constraint also fulfill the optical flow constraint and vice versa. This means that the divergence-free mass preservation set

$$S^0_{MP} = \left\{(u, \boldsymbol{v}) : \partial_t u + \nabla \cdot (u\boldsymbol{v}) = 0, \nabla \cdot \boldsymbol{v} = 0\right\}.$$

is included in the set fulfilling the optical flow constraint

$$S_{OF} = \left\{(u, \boldsymbol{v}) : \partial_t u + \nabla u \cdot \boldsymbol{v} = 0\right\}$$

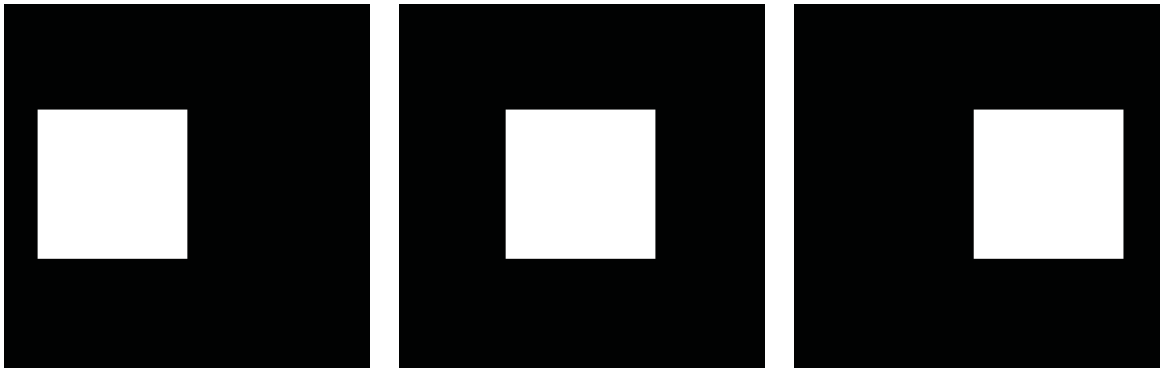
It follows that, in terms of extracting apparent motion from image sequences, the mass preservation constraint fits to a set of data situations with intersections to the set of optical flow datasets. Especially for situations where the brightness constancy assumption is violated, the mass preservation constraint offers a new way to estimate the underlying motion field.
Figure (4.6) shows examples of an image sequence fulfilling both equations, namely optical flow (Section 4.2) and mass preservation (Section 4.3), and an image sequence fulfilling only the later one. The result Section (4.6) compares both constraints applied to a translation example, which coincides with the first sequence. Moreover, an example for scaling is evaluated, which coincides with the melting block.

4.5. Variational Motion Estimation

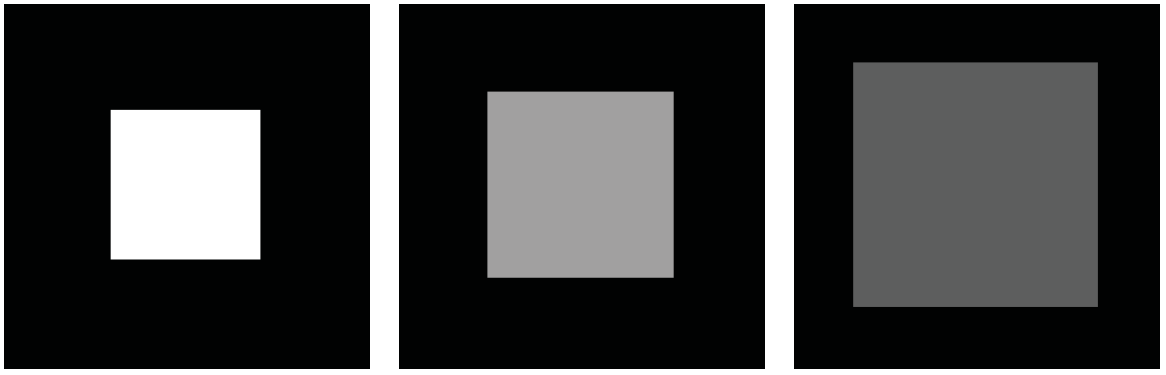
4.5.1. Mathematical Context

In the previous Section we deduced the optical flow constraint assuming a constancy of brightness

$$\partial_t u + \nabla u \cdot \boldsymbol{v} = 0$$



(a) t_0 **(b)** t_1 **(c)** t_2



(d) t_0 **(e)** t_1 **(f)** t_2

Figure 4.6.: Images (a) - (c) show a block of constant brightness moving from left to right. This image sequence fulfills the optical flow- and mass preservation constraint. Images (d) - (f) show a melting block. The sum of brightness is the same in all images, but spreads over a larger area. The optical flow constraint is no longer fulfilled.

and assuming a constant mass, which is being transported by a continuous flow, the mass preservation constraint

$$\partial_t u + \nabla \cdot (u\boldsymbol{v}) = 0.$$

One or the other has to be satisfied by the unknown velocity field $\boldsymbol{v}$ for the underlying image data u. As a matter of fact there is only one equation for two unknowns (or three in case of 3-dimensional images), and we have an underdetermined equation in every point $x \in \Omega$. A possible way out is to consider a variational problem of the following type

$$\arg\min_{\boldsymbol{v}} = C(u, \boldsymbol{v}) + \alpha R(\boldsymbol{v}),$$

where $C(u, \boldsymbol{v})$ is a placeholder for the underlying flow equation mentioned above. With the second term $R(\boldsymbol{v})$ we are able to model additional a-priori information about the solution $\boldsymbol{v}$ that renders the problem well-posed. We usually set

$$C(u, \boldsymbol{v}) = \|\partial_t u + \nabla u \cdot \boldsymbol{v}\|_{L^r}^s, \quad r, s \geq 1$$

for the optical flow constraint or

$$C(u, \boldsymbol{v}) = \left\|\partial_t u + \nabla \cdot (u\boldsymbol{v})\right\|_{L^r}^s, \quad r, s \geq 1$$

for the mass preservation constraint. We want to emphasize that $\boldsymbol{v}$ is the objective variable and u, u_t and ∇u are given. The choice of the norm and the exponent of $C(u, \boldsymbol{v})$ are dependent on the problem at hand. The quadratic L^2-norm minimizes deviations from the constraint in a least squares sense. The L^1 norm matches the constraint in a linear way. In Section 4.5.6 a more detailed motivation for the L^1 norm is deduced.
Besides the data term $C(u, \boldsymbol{v})$ we can incorporate additional a-priori information in form of a regularization term $R(\boldsymbol{v})$. In many situations a reasonable assumption is that the velocity field is smooth. From a mathematical point of view we want to penalize high gradients and therefore choose

$$R(\boldsymbol{v}) = \sum_{i=1}^{d} \|\nabla v_i\|_2^2,$$

with spatial dimension d. Another possible regularizer that aims for regions of constant movement split by jumps is the total variation (see Section 2.1.3 for details) and we set

$$R(\boldsymbol{v}) = \sum_{i=1}^{d} |v_i|_{BV} \,,$$

We want to mention here that for the minimization part always discrete images u and velocity fields $\boldsymbol{v}$ are assumed.

4.5.2. $L^2 - L^2$ Optical Flow Model

Model

Assuming that the underlying image data satisfies the assumptions discussed in Section 4.2 we use the quadratic L^2 optical flow constraint as data fidelity and also the quadratic L^2 norm of the gradient as a regularization. This leads to the following model:

Model 4.5.1. $L^2 - L^2$ Optical Flow Model

$$\arg\min_{\boldsymbol{v}} \frac{1}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 + \frac{\alpha}{2} \sum_{i=1}^{d} \|\nabla v_i\|_2^2 \,,$$

which is the optical flow model by Horn and Schunck [43]. It has been published in 1981 and can be seen as the grandfather of variational motion estimation. We expect from a minimizer to fulfill the optical flow constraint in a least squares sense, which is reasonable because this counteracts against the influence of noise. Due to the quadratic L^2 regularization of the gradient possible discontinuities of the flow field are not preserved and we expect a smooth flow field.

Computational Minimization

We consider the finite dimensional setting for the minimization process . The minimization of this model could be done in a straight forward way because we have a simple linear L^2 problem and are able to apply Lemma 2.2.27 (solving L^2 optimization problems using a system of linear equations). Here we want to go a different way and use the well known primal-dual Chambolle-Pock algorithm [26]. This is due to the fact that we will consider more difficult problems later on and want to use a consistent

framework and notation for all our models. In Section 2.3.1 we gave a brief introduction to the primal-dual minimization theory.
For transferring our model to the notation of [26] we decouple the regularization term in terms of the dual functional and define

$$F(K\boldsymbol{v}) := \sum_{i=1}^{2} \frac{\alpha}{2} \|\nabla v_i\|_2^2 ,$$

with

$$K\boldsymbol{v} = \begin{pmatrix} \nabla & 0 \\ 0 & \nabla \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \end{pmatrix}, \quad K^*\boldsymbol{y} = -\begin{pmatrix} \nabla\cdot & 0 \\ 0 & \nabla\cdot \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix},$$

where $\boldsymbol{y} = (y_1, y_2)^t$ represents the newly introduced dual variables. Due to the fact that we are in a finite dimensional setting, K is a bounded linear operator. According to Example 2.2.23, we have for the squared L^2 norm

$$F^*(\boldsymbol{y}) := \sum_{i=1}^{2} \frac{1}{2\alpha} \|y_i\|_2^2 .$$

The optical flow part is linear in $\boldsymbol{v}$ and is incorporated in the primal functional as follows:

$$G(\boldsymbol{v}) := \frac{1}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 .$$

Consequently, the corresponding primal-dual problem to Model 4.5.1 reads

$$\arg\min_{\boldsymbol{v}} \arg\max_{\boldsymbol{y}} \frac{1}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 + \langle \boldsymbol{v}, K^*\boldsymbol{y} \rangle - \frac{1}{2\alpha} \sum_{i=1}^{2} \|y_i\|_2^2 .$$

Now, using the Chambolle-Pock algorithm (Section 2.3.1) we end up with the following iterative scheme:

$$\tilde{\boldsymbol{y}}_{k+1} = \boldsymbol{y}_k + \sigma K \bar{\boldsymbol{v}}_{\boldsymbol{k}} \tag{4.8}$$

$$\boldsymbol{y}_{k+1} = \arg\min_{\boldsymbol{y}} \left\{ \frac{1}{2} \left\| \boldsymbol{y} - \tilde{\boldsymbol{y}}_{k+1} \right\|_2^2 + \frac{\sigma}{2\alpha} \|\boldsymbol{y}\|_2^2 \right\} \tag{4.9}$$

$$\tilde{\boldsymbol{v}}_{k+1} = \boldsymbol{v}_k + \tau K^* \boldsymbol{y}_{k+1} \tag{4.10}$$

$$\boldsymbol{v}_{k+1} = \arg\min_{\boldsymbol{v}} \left\{ \frac{1}{2} \|\boldsymbol{v} - \tilde{\boldsymbol{v}}_{k+1}\|_2^2 + \frac{\tau}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 \right\} \tag{4.11}$$

$$\bar{\boldsymbol{v}}_{k+1} = \boldsymbol{v}_{k+1} + \theta(\boldsymbol{v}_{k+1} - \boldsymbol{v}_k). \tag{4.12}$$

The subproblem in $\boldsymbol{y}$ (in Equation (4.9)) is a linear L^2 problem. Exemplarily for this section we will derive optimality conditions. We start with the written-out energy for $\lambda := \frac{\sigma}{\alpha}$:

$$\begin{aligned} J(\boldsymbol{y}) = J(y_1, y_2) &= \frac{1}{2} \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_2^2 + \frac{\lambda}{2} \|\boldsymbol{y}\|_2^2 \\ &= \frac{1}{2} \int_\Omega (y_1 - \tilde{y}_1)^2 + (y_2 - \tilde{y}_2)^2 \, \mathrm{d}\boldsymbol{y} + \frac{\lambda}{2} \int_\Omega y_1^2 + y_2^2 \, \mathrm{d}\boldsymbol{y} \end{aligned}$$

Let us now calculate the Gateâux-derivative for y_1 (see Definition 2.2.4 for details). Let therefore ψ be an arbitrary direction

$$\begin{aligned} d_\psi J(y_1, y_2) &= \lim_{\tau \to 0} \frac{J(y_1 + \tau\psi, y_2) - J(y_1, y_2)}{\tau} = \frac{d}{d\tau} J(y_1 + \tau\psi, y_2)\Big|_{\tau=0} \\ &= \frac{d}{d\tau} \frac{1}{2} \int_\Omega (y_1 + \tau\psi - \tilde{y}_1)^2 + (y_2 - \tilde{y}_2)^2 + \frac{\lambda}{2} \left((y_1 + \tau\psi)^2 + y_2^2 \right) \mathrm{d}\boldsymbol{y} \Big|_{\tau=0} \\ &= \frac{1}{2} \int_\Omega 2\psi(y_1 + \tau\psi - \tilde{y}_1) + \frac{\lambda}{2} \cdot 2\psi(y_1 + \tau\psi) \, \mathrm{d}\boldsymbol{y} \Big|_{\tau=0} = \int_\Omega \psi(y_1 - \tilde{y}_1) + \lambda\psi y_1 \, \mathrm{d}\boldsymbol{y} \\ &= \int_\Omega \psi \left((y_1 - \tilde{y}_1) + \lambda y_1 \right) \mathrm{d}\boldsymbol{y} \end{aligned}$$

Due to the fact that for an optimal solution all derivatives should be zero we end up with the optimality condition

$$y_1 - \tilde{y}_1 + \lambda y_1 \stackrel{!}{=} 0.$$

This can be done in an analogue way for y_2 and we obtain the optimality system

$$\begin{aligned} y_1 - \tilde{y}_1 + \lambda y_1 &\stackrel{!}{=} 0, \\ y_2 - \tilde{y}_2 + \lambda y_2 &\stackrel{!}{=} 0, \end{aligned}$$

or alternatively in a shorter vectorial notation (with $\frac{\sigma}{\alpha} := \lambda$)

$$\boldsymbol{y} - \tilde{\boldsymbol{y}}_{k+1} + \frac{\sigma}{\alpha} \boldsymbol{y} \stackrel{!}{=} 0.$$

Because there is no operator involved we can simply solve the equation for $\boldsymbol{y}$ and see that the solution is given by

$$\boldsymbol{y}_{k+1} = \frac{\alpha}{\alpha + \sigma}\tilde{\boldsymbol{y}}_{k+1}.$$

The problem for the primal variables $\boldsymbol{v} = \begin{pmatrix} v^1 \\ v^2 \end{pmatrix}$ is a quadratic optimization problem and we can directly calculate the optimality conditions as

$$\begin{aligned} 0 &\overset{!}{=} (v^1 - \tilde{v}^1_{k+1}) + \tau u_x(u_t + v^1 u_x + v^2 u_y) \\ 0 &\overset{!}{=} (v^1 - \tilde{v}^2_{k+1}) + \tau u_y(u_t + v^1 u_x + v^2 u_y). \end{aligned}$$

A few simple calculations yield the rearrangement

$$\begin{aligned} (1 + \tau u_x u_x)v^1 + \tau u_x u_y v^2 &= \tilde{v}^1_{k+1} - \tau u_x u_t \\ \tau u_x u_y v^1 + (1 + \tau u_y u_y)v^2 &= \tilde{v}^2_{k+1} - \tau u_y u_t. \end{aligned} \tag{4.13}$$

It is easy to see that we have a system of two equations with two unknowns. This can be solved directly. For this sake we define

$$\begin{aligned} c_1 := 1 + \tau u_x u_x, \quad c_2 = \tau u_x u_y, \quad c_3 := 1 + \tau u_y u_y \\ b_1 := \tilde{v}^1_{k+1} - \tau u_x u_t \quad b_2 := \tilde{v}^2_{k+1} - \tau u_y u_t. \end{aligned}$$

We see that system (4.13) simplifies to

$$\begin{aligned} c_1 v^1 + c_2 v^2 &= b_1 \\ c_2 v^1 + c_3 v^2 &= b_2, \end{aligned}$$

and we can directly calculate the solution as

$$v^1_{k+1} = \frac{b_1 c_3 - c_2 b_2}{c_1 c_3 - c_2^2}, \quad v^2_{k+1} = \frac{b_2 c_1 - c_2 b_1}{c_1 c_3 - c_2^2}.$$

We end this Section with the final iteration scheme for $L^2 - L^2$ optical flow estimation:

$$\tilde{\boldsymbol{y}}_{k+1} = \boldsymbol{y}_k + \sigma K \bar{\boldsymbol{v}}_{\boldsymbol{k}} \tag{4.14}$$

$$\boldsymbol{y}_{k+1} = \frac{\alpha}{\alpha + \sigma}\tilde{\boldsymbol{y}}_{k+1} \tag{4.15}$$

$$\tilde{\boldsymbol{v}}_{k+1} = \boldsymbol{v}_k - \tau K^* \boldsymbol{y}_{k+1} \tag{4.16}$$

$$\boldsymbol{v}_{k+1} = \left(\frac{b_1 c_3 - c_2 b_2}{c_1 c_3 - c_2^2}, \frac{b_2 c_1 - c_2 b_1}{c_1 c_3 - c_2^2} \right)^T \tag{4.17}$$

$$\bar{\boldsymbol{v}}_{k+1} = \boldsymbol{v}_{k+1} + \theta(\boldsymbol{v}_{k+1} - \boldsymbol{v}_k) \tag{4.18}$$

Discretization and Algorithm

Operator Discretization:

To finally implement the given iterative scheme we have to specify the discretization for the involved operators. We recall the given data term:

$$u_t + \nabla u \cdot \boldsymbol{v}.$$

From the viewpoint of u this is a transport equation. Since we are only minimizing with respect to $\boldsymbol{v}$ and the partial derivatives of u not necessarily need to be a stable scheme for the transport equation.

Nevertheless we *will* introduce a scheme that is stable in terms of transport because in the subsequent Chapter 5, joint models for motion estimation and image reconstruction are derived. Here the minimization is done with respect to u as well, which requires the scheme to be stable.

We already mentioned that

$$u_t + \nabla_x u \cdot \boldsymbol{v} = 0$$

has the form of a transport equation. Discretizing the derivatives with forward differences is impractical, because this requires a very small timestep δ_t. For most applications the underlying timestep between recorded images is far beyond the allowed value. For transport equations, we usually use an upwind scheme (see for example [41] Chapter **XVIII**) where depending on the sign of $\boldsymbol{v}$ a forward or backward difference approximation is used. It can be shown that this yields a stable discretization.

Taking now the perspective of $\boldsymbol{v}$ we are solving the optical flow equation. Unfortunately we observed that an upwind scheme for u generates artifacts in the resulting velocity field and therefore should not be used for motion estimation. A way out is given by a mix of forward and central differences. Using forward differences for the time derivative u_t and central differences for the spatial derivatives $\nabla_x u$ yields a stable discretization, if implicitly solved, of the transport equation and produces reasonable velocity fields. The adjoint operator then consists of a backward difference for the time derivative and a central difference for the spatial part. We mention here that the central difference is self-adjoint.

We denote the discrete version of u with U which is given at

$$\{(i,j,t) : i = 0,\dots,n_x, j = 0,\dots,n_y, t = 0,1\}$$

For the flow estimation the image derivatives have to be evaluated in every pixel at one timestep (which is omitted in the notation)

$$U_t(i,j) = \frac{U(i,j,t+1) - U(i,j,t)}{\delta_t}$$

$$U_x(i,j) = \begin{cases} \frac{U(i+1,j,t)-U(i-1,j,t)}{2\delta_x} & \text{if } i > 0 \text{ and } i < n_x \\ 0 & \text{if } i = 0 \text{ or } i = n_x \end{cases}$$

$$U_y(i,j) = \begin{cases} \frac{U(i,j+1,t)-U(i,j-1,t)}{2\delta_y} & \text{if } j > 0 \text{ and } j < n_y \\ 0 & \text{if } j = 0 \text{ or } j = n_y \end{cases}$$

In terms of the velocity field $\boldsymbol{v}$, we consider the discrete version V on

$$\{(i,j) : i = 0,\dots,n_x, j = 0,\dots,n_y\}.$$

A gradient operator is involved in the regularization part, hence partial derivatives ∂_x, ∂_y are needed. Here we use forward differences for the forward operator applied to v and backward differences for the adjoint operator applied to the dual variables $\boldsymbol{y}$ with Neumann boundary conditions. The resulting scheme reads

$$v_x(i,j) = \begin{cases} v(i+1,j) - v(i,j) & \text{if } i < n_x \\ 0 & \text{if } i = n_x \end{cases}$$

$$v_y(i,j) = \begin{cases} v(i,j+1) - v(i,j) & \text{if } j < n_y \\ 0 & \text{if } j = n_y \end{cases}$$

$$\nabla \cdot \boldsymbol{y}(i,j) = \begin{cases} y_1(i,j) - y_1(i-1,j) & \text{if } i > 0 \\ y_1(i,j) & \text{if } i = 0 \\ -y_1(i-1,j) & \text{if } i = n_x \end{cases} + \begin{cases} y_2(i,j) - y_2(i,j-1) & \text{if } j > 0 \\ y_2(i,j) & \text{if } j = 0 \\ -y_2(i,j-1) & \text{if } j = n_y. \end{cases}$$

In appendix $A.2$ an alternative image discretization is explained. The original paper of Horn and Schunck contains so-called cell-centered differences to discretize the gradient operator. They form consistent derivatives for u (proof can be found in Appendix $A.2$, Lemma $A.2.1$), but are not a stable scheme for the transport equation (proof can be found in Appendix $A.2$, Lemma $A.2.2$).

Stopping criterion:
Besides the operator discretization a stopping criterion is needed for the iterative scheme. For our algorithms a combination of the so-called primal and dual residual is used. This criterion has been introduced by Goldstein, Esser and Baraniuk [39] and offers an easy to adapt and numerically cheap convergence criterion.

Definition 4.5.2. Primal and dual residual
Let $x^k, x^{k+1}, y^k, y^{k+1}$ be subsequent primal resp. dual iterations and $K : \Omega \to \mathbb{R}^d$ the forward operator from the primal-dual algorithm (see Section 2.3.1). Then we define

$$p^k := \left| \frac{x^k - x^{k+1}}{\tau^k} - K^*(y^k - y^{k+1}) \right|$$
$$d^k := \left| \frac{y^k - y^{k+1}}{\sigma^k} - K(x^k - x^{k+1}) \right|,$$

with $|\cdot|$ being the sum of absolute values. We call p^k the primal residual and d^k the dual residual.

It has been shown in [39] that, as the algorithm converges to a minimizer, we have $p^k \to 0$ and $d^k \to 0$. Consequently we can use the residuals as a stopping criterion and define an error

$$\epsilon^k := \frac{p^k + d^k}{|\Omega|}.$$

The error is normalized through division by $|\Omega|$, to be invariant of the underlying problem size. The algorithm is terminated once ϵ^k is below some error threshold η.

Initial values and parameters:
We initialize the variables $(\boldsymbol{v}, \bar{\boldsymbol{v}}, \boldsymbol{y})$ with zero. It has been proven in [26] that convergence of the algorithm is guaranteed for $\tau\sigma\,|K|^2 < 1$ (and $\theta = 1$). For the chosen forward discretization of the gradient it has been shown in [25] that $\sigma = \tau = \frac{1}{\sqrt{8}}$ satis-

fies the required bound. As stopping criterion for the algorithm we set η to 1.0^{-4}.

Algorithm:
From a computational point of view the algorithm requires very few basic operations per iteration. Application of K and K^* requires the relatively highest effort as we have to evaluate forward and backward differences. Furthermore the algorithm can be directly parallelized where both lines 7-9 and lines 10-12 form independent blocks.

Algorithm 3 $L^2 - L^2$ Optical Flow

L2L2OpticalFlowU_1, U_2, α, η
1: $U_t, U_x, U_y \leftarrow$ calculateImageDerivatives(U_1,U_2)
2: $K \leftarrow$ buildOperator()
3: $\sigma, \tau \leftarrow \frac{1}{\sqrt{8}}$
4: $v, \bar{v}, y \leftarrow 0$ $\epsilon < \eta$
5: $vOld \leftarrow v$
6: $\tilde{y} \leftarrow y + \sigma K \bar{v}$
7: $y \leftarrow \frac{\alpha}{\alpha+\sigma}\tilde{y}$
8: $\tilde{v} \leftarrow v - \sigma K^* y$
9: $v \leftarrow solveMatrix(\tilde{v}, U_t, U_x, U_y)$
10: $\bar{v} \leftarrow 2 \cdot v - vOld$
11: v

4.5.3. $L^2 - L^2$ Mass Preservation Model

Model

In contrast to an optical flow model we assume a mass preservation in time and additionally the local continuity, which means that mass can only be moved by a continuous flow (see (4.3) for details). The resulting mass preservation constraint is used as data fidelity besides a quadratic L^2 norm of the gradient as regularization for the velocity field. Then we have the following model:

Model 4.5.3. $L^2 - L^2$ Mass Preservation Model

$$\arg\min_{\boldsymbol{v}} \frac{1}{2} \left\| u_t + \nabla \cdot (u\boldsymbol{v}) \right\|_2^2 + \frac{\alpha}{2} \sum_{i=1}^{d} \|\nabla v_i\|_2^2 . \tag{4.19}$$

Choosing this model for motion estimation assumes the underlying flow field to fulfill the continuity equation in a least squares sense. Moreover smooth components v_i are supposed and consequently a L^2 regularization of the gradient is used.

Computational Minimization

For the sake of minimization the primal-dual algorithm from Section 2.3.1 is used again. Due to the fact that the target variable $\boldsymbol{v}$ now occurs in combination with an operator in the data term and the regularization term we represent the whole functional in terms of the dual F. Consequently we have

$$F(K\boldsymbol{v}) := \frac{1}{2}\left\|u_t + \nabla\cdot(u\boldsymbol{v})\right\|_2^2 + \frac{\alpha}{2}\sum_{i=1}^{d}\|\nabla v_i\|_2^2\,,$$

with the linear operator K given by

$$K\boldsymbol{v} = \begin{pmatrix} \nabla & 0 \\ 0 & \nabla \\ \partial_x u & \partial_y u \end{pmatrix}\begin{pmatrix} v^1 \\ v^2 \end{pmatrix}.$$

For the sake of clarity we want to emphasize here that $\partial_x u\cdot v^1 := \partial_x(u\cdot v^1)$. The adjoint operator for the dual $\boldsymbol{y} = (y_1, y_2, y_3)^t$ is given by

$$K^*\boldsymbol{y} = -\begin{pmatrix} \nabla\cdot & 0 & u\partial_x \\ 0 & \nabla\cdot & u\partial_y \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}.$$

The corresponding dual functional $F^*(\boldsymbol{y})$ to F can be calculated using Examples 2.2.23 and 2.2.25 as:

$$F^*(\boldsymbol{y}) := \frac{1}{2}\|y_3\|_2^2 - \langle u_t, y\rangle + \sum_{i=1}^{2}\frac{1}{2\alpha}\|y_i\|_2^2\,.$$

Consequently, the corresponding primal-dual problem to Model 4.5.3 reads

$$\arg\min_{\boldsymbol{v}}\arg\max_{\boldsymbol{y}}\langle\boldsymbol{v}, K^*\boldsymbol{y}\rangle - \frac{1}{2\alpha}\sum_{i=1}^{2}\|y_i\|_2^2 - \frac{1}{2}\|y_3\|_2^2 + \langle u_t, y_3\rangle\,.$$

Incorporating this into our algorithm, we have the iterative scheme

$$\tilde{\boldsymbol{y}}^{k+1} = \boldsymbol{y}^k + \sigma K \bar{\boldsymbol{v}} \tag{4.20}$$

$$y_{1,2}^{k+1} = \arg\min_{\boldsymbol{y}} \left\{ \frac{1}{2} \left\| \boldsymbol{y} - \tilde{y}_{1,2}^{k+1} \right\|_2^2 + \frac{\sigma}{2\alpha} \left\| \boldsymbol{y} \right\|_2^2 \right\} \tag{4.21}$$

$$y_3^{k+1} = \arg\min_{y} \left\{ \frac{1}{2} \left\| y - \tilde{y}_3^{k+1} \right\|_2^2 + \frac{\sigma}{2} \left\| y \right\|_2 - \sigma \left\langle u_t, y \right\rangle \right\} \tag{4.22}$$

$$\boldsymbol{v}^{k+1} = \arg\min_{\boldsymbol{v}} \left\{ \int_0^T \frac{1}{2} \left\| \boldsymbol{v} - (\boldsymbol{v}^k - \tau K^* \boldsymbol{y}) \right\|_2^2 \right\} \tag{4.23}$$

$$\bar{\boldsymbol{v}}^{k+1} = \boldsymbol{v}^{k+1} + \theta(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k). \tag{4.24}$$

The update step given in Equation (4.21) does not differ from the optical flow case. The solution can be found in Section 4.5.2 and is given by

$$y_{1,2}^{k+1} = \frac{\alpha}{\alpha + \sigma} \tilde{y}_{1,2}^{k+1}.$$

For the subproblem in Equation (4.22) we calculate the optimality condition as

$$0 \stackrel{!}{=} y - \tilde{y}_3^{k+1} + \sigma y - \sigma u_t.$$

Now, we solve the equation for y and get the update formula

$$y_3^{k+1} = \frac{1}{\sigma + 1} \tilde{y}_3^{k+1} + \frac{\sigma}{\sigma + 1} u_t,$$

where the latter part is just a constant. The solution for the primal update can be derived directly as

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k - \tau K^* \boldsymbol{y}.$$

Hence we get the following iterative scheme for $L^2 - L^2$ mass preservation:

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{k+1} &= \boldsymbol{y}^k + \sigma K \bar{\boldsymbol{v}}^{\boldsymbol{k}} \\
\boldsymbol{y}_{1,2}^{k+1} &= \frac{\alpha}{\alpha + \sigma} \tilde{\boldsymbol{y}}_{1,2}^{k+1} \\
y_3^{k+1} &= \frac{1}{\sigma + 1} \tilde{y}_3^{k+1} + \frac{\sigma}{\sigma + 1} u_t \\
\boldsymbol{v}^{k+1} &= \boldsymbol{v}^k - \tau K^* \boldsymbol{y} \\
\bar{\boldsymbol{v}}^{k+1} &= \boldsymbol{v}^{k+1} + \theta(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k)
\end{aligned}$$

Discretization and Algorithm

The discretization can be done similar to the $L^2 - L^2$ optical flow model (see Section 4.5.2) with minor changes for the transport part $u_t + \nabla_x \cdot (u\boldsymbol{v})$. We still use forward differences for the discrete time-derivative u_t and central differences for the spatial part $\nabla_x \cdot$. But in contrary to the optical flow discretization we can no longer precalculate the spatial part because this operator is now applied to the objective variable $\boldsymbol{v}$. Moreover, the gradient has to be applied to a product of u and $\boldsymbol{v}$. The adjusted discretization scheme reads

$$\partial_x(v^1 \cdot U)(i,j) = \begin{cases} \frac{1}{2\delta_x}\left((v^1 \cdot u)(i+1,j) - (v^1 \cdot u)(i-1,j)\right) & \text{if } i > 0 \text{ and } i < n_x \\ 0 & \text{if } i = 0 \text{ or } i = n_x \end{cases}$$

$$\partial_y(v^2 \cdot u)(i,j) = \begin{cases} \frac{1}{2\delta_y}\left((v^2 \cdot u)(i,j+1) - (v^2 \cdot u)(i,j-1)\right) & \text{if } j > 0 \text{ and } j < n_y \\ 0 & \text{if } j = 0 \text{ or } j = n_y \end{cases}.$$

No changes are needed in the discretization of the regularization operator.

Stopping criterion, initial values and parameters:
As a stopping criterion we inherit the sum of the primal and dual residual ϵ^k (see Definition 4.5.2) and initialize the variables $(\boldsymbol{v}, \bar{\boldsymbol{v}}, \boldsymbol{y})$ with zero. For the choice of τ and σ we have to be careful because the operator K cannot be estimated directly. We now have image information involved within K. The numerical realization shows that $\tau = \sigma = \frac{1}{\sqrt{8}\max|u|}$ yields a stable scheme.

Algorithm:
We denote the following pseudocode for the $L^2 - L^2$ mass preservation model. Similar to the previous $L^2 - L^2$ optical flow section the algorithm can be evaluated easily from a numerical point of view. We have to apply a forward- resp. backward difference in line 8. resp. line 11. The remaining operations only consist of addition and multiplication.

4.5.4. $L^2 - TV$ Optical Flow Model

Model

The common characteristic of the the previous models was the regularization term. Penalizing the squared gradients of the v_i's results in a smooth velocity field. However, in many applications this is not what we expect to see. In certain applications we

Algorithm 4 $L^2 - L^2$ Mass Preservation

L2L2MassPreservationU_1, U_2, α, η
$U_t \leftarrow$ calculateImageDerivatives(U_1,U_2)
$K \leftarrow$ buildOperator(U_1, U_2)
$\sigma, \tau \leftarrow \frac{1}{\sqrt{8}\max|u|}$
$v, \bar{v}, y \leftarrow 0$ $\epsilon < \eta$
$vOld \leftarrow v$
$\tilde{y} \leftarrow y + \sigma K \bar{v}$
$y_{1,2} \leftarrow \frac{\alpha}{\alpha+\sigma} \tilde{y_{1,2}}$
$y_3 \leftarrow \frac{1}{\sigma+1} \tilde{y_3} + \frac{\sigma}{\sigma+1} U_t$
$v \leftarrow v - \sigma K^* y$
$\bar{v} \leftarrow 2 \cdot v - vOld$
v

are interested in recovering the movement of objects (e.g. cells, cars, people), which are separated from the background by sharp edges. In these cases we also expect the velocity field to have sharp edges against the background. Furthermore we expect a constant magnitude inside the subdomain that represents the object.
One possible solution to tackle these properties is the use of the total variation of v_i instead of the squared L^2-norm of the gradient. The total variation allows discontinuities and tends to recover cartoon-like motion components. The resulting model then reads

Model 4.5.4. $L^2 - TV$ Optical Flow Model

$$\arg\min_{\boldsymbol{v}} \frac{1}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 + \alpha \sum_{i=1}^{d} |v_i|_{BV} . \tag{4.25}$$

Computational Minimization

We proceed here in the finite dimensional setting where Model 4.5.4 simplifies to

$$\arg\min_{\boldsymbol{v}} \frac{1}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 + \alpha \sum_{i=1}^{d} \|\nabla v_i\|_1 .$$

Similar to the $L^2 - L^2$ optical flow model we use the primal-dual algorithm proposed by Chambolle and Pock. We write the regularization terms in form of the dual functional

and define therefore

$$F(K\boldsymbol{v}) := \alpha \sum_{i=1}^{d} \|\nabla v_i\|_1 ,$$

with

$$K\boldsymbol{v} = \begin{pmatrix} \nabla & 0 \\ 0 & \nabla \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \end{pmatrix},$$

for the forward operator. By introducing dual variables $\boldsymbol{y} = (y_1, y_2)^t$ we get the adjoint operator

$$K^*\boldsymbol{y} = - \begin{pmatrix} \nabla\cdot & 0 \\ 0 & \nabla\cdot \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

According to Example 2.2.25, we have for the L^1 norm

$$F^*(\boldsymbol{y}) := \alpha \sum_{i=1}^{2} \delta_{B(L^\infty)}(y_i/\alpha).$$

The optical flow part is again inserted in the primal functional as follows

$$G(\boldsymbol{v}) := \frac{1}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 .$$

The primal-dual formulation of the $L^2 - TV$ optical flow model reads:

$$\arg\min_{\boldsymbol{v}} \arg\max_{\boldsymbol{y}} \frac{1}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 + \langle \boldsymbol{v}, K^*\boldsymbol{y} \rangle - \alpha \sum_{i=1}^{2} \delta_{B(L^\infty)}(y_i/\alpha).$$

For the total variation regularization, only the underlying norm in the dual part has changed and we are able to use results from the $L^2 - L^2$ optical flow Section 4.5.2. This leads to the iterative scheme

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{k+1} &= \boldsymbol{y}^k + \sigma K \bar{\boldsymbol{v}}^{\boldsymbol{k}} \\
\boldsymbol{y}^{k+1} &= \arg\min_{\boldsymbol{y}=(y^1,y^2)} \left\{ \frac{1}{2} \left\| \boldsymbol{y} - \tilde{\boldsymbol{y}}^{k+1} \right\|_2^2 + \alpha\sigma \delta_{B(L^\infty)}(\boldsymbol{y}/\alpha) \right\} \\
\tilde{\boldsymbol{v}}^{k+1} &= \boldsymbol{v}^k + \tau K^* \boldsymbol{y}^{k+1} \\
\boldsymbol{v}^{k+1} &= \arg\min_{\boldsymbol{v}} \left\{ \frac{1}{2} \left\| \boldsymbol{v} - \tilde{\boldsymbol{v}}^{k+1} \right\|_2^2 + \frac{\tau}{2} \|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 \right\}
\end{aligned}$$

$$\bar{\boldsymbol{v}}^{k+1} = \boldsymbol{v}^{k+1} + \theta(\boldsymbol{v}^{k+1} - \boldsymbol{v}^{k}).$$

Except for the dual subproblem in $\boldsymbol{y}^{k+1}$ this scheme coincides with the $L^2 - L^2$ optical flow scheme (see Section 4.5.2).

Let us begin by writing down the convex set corresponding to $\delta_{B(L^\infty)}(y^1/\alpha)$:

$$\left\{ y^1 : \left\| y^1/\alpha \right\|_\infty \leq 1 \right\} \Leftrightarrow \left\{ y^1 : \left\| y^1 \right\|_\infty \leq \alpha \right\}.$$

Here, $\left\| y^1 \right\|_\infty$ denotes the discrete maximum over all elements (i, j) in the image domain:

$$\left\| y^1 \right\|_\infty = \max_{(i,j)} \left| y^1_{i,j} \right|.$$

We only consider the anisotropic total variation which corresponds to the vectorial L^1 norm for each element (i, j)

$$\left| \nabla v^1_{i,j} \right| = \left| \partial_x v^1_{i,j} \right| + \left| \partial_y v^1_{i,j} \right|.$$

Since the convex conjugate corresponds to a projection into the unit ball of the associated dual norm, we obtain the maximum norm for the dual variable:

$$\left| y^1_{i,j} \right| = \max \left\{ \left| y^{1,1}_{i,j} \right|, \left| y^{1,2}_{i,j} \right| \right\}.$$

Now, the solution to the minimization problem is directly given as the point-wise projection of $\tilde{\boldsymbol{y}}^{k+1}$ onto $[-\alpha, \alpha]$:

$$\boldsymbol{y}^{k+1} = \min(\alpha, \max(-\alpha, \tilde{\boldsymbol{y}}^{k+1})) =: \pi_\alpha(\tilde{\boldsymbol{y}}^{k+1})$$

Together with the preliminary results from the $L^2 - L^2$ optical flow part this leads to the following scheme:

$$\begin{aligned}
\tilde{\boldsymbol{y}}_{k+1} &= \boldsymbol{y}_k + \sigma K \bar{\boldsymbol{v}}_{\boldsymbol{k}} \\
\boldsymbol{y}_{k+1} &= \pi_\alpha(\tilde{\boldsymbol{y}}_{k+1}) \\
\tilde{\boldsymbol{v}}_{k+1} &= \boldsymbol{v}_k - \tau K^* \boldsymbol{y}_{k+1} \\
\boldsymbol{v}_{k+1} &= \left(\frac{b_1 c_3 - c_2 b_2}{c_1 c_3 - c_2^2}, \frac{b_2 c_1 - c_2 b_1}{c_1 c_3 - c_2^2} \right)^T \\
\bar{\boldsymbol{v}}_{k+1} &= \boldsymbol{v}_{k+1} + \theta(\boldsymbol{v}_{k+1} - \boldsymbol{v}_k).
\end{aligned}$$

Discretization and Algorithm

The discretization does not differ from the $L^2 - L^2$ optical flow model because the only difference of the two models lies in the norm of the regularization term. The discretization of the involved operators can be found in Section 4.5.2. Moreover, we are able to use a similar stopping criteria and variable initializations as in Section 4.5.2. This leads to the following algorithm. From the numerical point of view lines 8 and 10

Algorithm 5 $L^2 - TV$ Optical Flow

L2TVOpticalFlowU_1, U_2, α, η
1: $U_t, U_x, U_y \leftarrow$ calculateImageDerivatives(U_1,U_2)
2: $K \leftarrow$ buildOperator()
3: $\sigma, \tau \leftarrow \frac{1}{\sqrt{8}}$
4: $v, \bar{v}, y \leftarrow 0$ $\epsilon < \eta$
5: $vOld \leftarrow v$
6: $\tilde{y} \leftarrow y + \sigma K \bar{v}$
7: $y \leftarrow \tilde{y} - \sigma S(\frac{\tilde{y}}{\sigma}, \frac{\alpha}{\sigma})$
8: $\tilde{v} \leftarrow v - \sigma K^* y$
9: $v \leftarrow solveMatrix(\tilde{v}, U_t, U_x, U_y)$
10: $\bar{v} \leftarrow 2 \cdot v - vOld$
11: v

are the most expensive operations. The solution for solving the matrix in line 11 can be deduced directly as shown in the $L^2 - L^2$ optical flow section. Concerning to the newly introduced thresholding operator, also this operation requires only a few additions and multiplications besides a maximum evaluation of two elements and consequently is numerically cheap.

4.5.5. $L^2 - TV$ Mass Preservation Model

Model

In analogy to the $L^2 - TV$ optical flow model from the previous Section 4.5.4 we want to transfer the same ideas to the mass preservation case. We begin writing down the model.

Model 4.5.5. $L^2 - TV$ Mass Preservation Model

$$\arg\min_{\boldsymbol{v}} \frac{1}{2} \left\| u_t + \nabla \cdot (u\boldsymbol{v}) \right\|_2^2 + \alpha \sum_{i=1}^{d} |v_i|_{BV} .$$

Here, we do not assume the displacement of constant intensities in the image data, but moreover expect the flow to fulfill the continuity equation $u_t + \nabla \cdot (u\boldsymbol{v})$ in a least squares sense. In contrast to the linear $L^2 - L^2$ mass preservation model, we choose total variation regularization for motion fields that are expected to consist of constant magnitude and are separated by sharp edges.

Computational Minimization

The total variation regularizer in the finite dimensional setting again becomes $\|\nabla v_i\|_1$ and we minimize

$$\arg\min_{\boldsymbol{v}} \frac{1}{2} \left\|u_t + \nabla \cdot (u\boldsymbol{v})\right\|_2^2 + \alpha \sum_{i=1}^{2} \|\nabla v_i\|_1 .$$

Similar to the $L^2 - TV$ optical flow energy we have a non-differentiability of the regularization part $\|\nabla v_i\|_1$. We therefore return to the Chambolle–Pock method and mainly combine the results from the $L^2 - L^2$ mass preservation Section 4.5.3 and the previous Section. Starting with the notation we inherit the definition of F as follows:

$$F(K\boldsymbol{v}) := \frac{1}{2} \left\|u_t + \nabla \cdot (u\boldsymbol{v})\right\|_2^2 + \alpha \sum_{i=1}^{2} \|\nabla v_i\|_1 ,$$

with the linear operator K given by

$$K\boldsymbol{v} = \begin{pmatrix} \nabla & 0 \\ 0 & \nabla \\ \partial_x u & \partial_y u \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \end{pmatrix} .$$

The adjoint operator for the dual $\boldsymbol{y} = (y_1, y_2, y_3)^t$ is now given by

$$K^* \boldsymbol{y} = - \begin{pmatrix} \nabla\cdot & 0 & u\partial_x \\ 0 & \nabla\cdot & u\partial_y \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} .$$

The corresponding dual functional $F^*(\boldsymbol{y})$ to F can be calculated using Examples 2.2.23 and 2.2.25 as:

$$F^*(\boldsymbol{y}) := \frac{1}{2}\|y_3\|_2^2 - \langle u_t, y\rangle + \alpha \sum_{i=1}^{2} \delta_{B(L^\infty)}(y_i/\alpha).$$

Consequently the primal part $G(\boldsymbol{v})$ is zero and we get for the primal-dual formulation:

$$\arg\min_{\boldsymbol{v}} \arg\max_{\boldsymbol{y}} \langle \boldsymbol{v}, K^*\boldsymbol{y}\rangle - \alpha \sum_{i=1}^{2} \delta_{B(L^\infty)}(y_i/\alpha) - \frac{1}{2}\|y_3\|_2^2 + \langle u_t, y_3\rangle .$$

Incorporating this into the Chambolle–Pock framework leads to the scheme

$$\tilde{\boldsymbol{y}}^{k+1} = \boldsymbol{y}^k + \sigma K \bar{\boldsymbol{v}} \tag{4.26}$$

$$\boldsymbol{y}_{1,2}^{k+1} = \arg\min_{\boldsymbol{y}=(y^1,y^2)} \left\{ \frac{1}{2}\left\|\boldsymbol{y} - \tilde{\boldsymbol{y}}_{1,2}^{k+1}\right\|_2^2 + \alpha\sigma\delta_{B(L^\infty)}(\boldsymbol{y}/\alpha) \right\} \tag{4.27}$$

$$y_3^{k+1} = \arg\min_{y} \left\{ \frac{1}{2}\left\|y - \tilde{y}_3^{k+1}\right\|_2^2 + \frac{\sigma}{2}\|y\|_2^2 - \sigma\langle u_t, y\rangle \right\} \tag{4.28}$$

$$\boldsymbol{v}^{k+1} = \arg\min_{\boldsymbol{v}} \left\{ \int_0^T \frac{1}{2}\left\|\boldsymbol{v} - (\boldsymbol{v}^k - \tau K^*\boldsymbol{y})\right\|_2^2 \right\} \tag{4.29}$$

$$\bar{\boldsymbol{v}}^{k+1} = \boldsymbol{v}^{k+1} + \theta(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k). \tag{4.30}$$

The subproblem for $y_{1,2}^{k+1}$ in Equation (4.27) has been solved for the $TV - L^2$ optical flow model 4.5.4. The solution is given by point-wise projection onto $[-\alpha, \alpha]$. The other problems do not differ from the $L^2 - L^2$ mass preservation scheme. Hence we obtain the following iterations which converge to a solution of Model 4.5.5:

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{k+1} &= \boldsymbol{y}^k + \sigma K \bar{\boldsymbol{v}}^{\boldsymbol{k}} \\
\boldsymbol{y}_{k+1} &= \pi_\alpha(\tilde{y}_{1,2}^{k+1}) \\
y_3^{k+1} &= \frac{1}{\sigma+1}\tilde{y}_3^{k+1} + \frac{\sigma}{\sigma+1}u_t \\
\boldsymbol{v}^{k+1} &= \boldsymbol{v}^k - \tau K^*\boldsymbol{y} \\
\bar{\boldsymbol{v}}^{k+1} &= \boldsymbol{v}^{k+1} + \theta(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k)
\end{aligned}$$

Discretization and Algorithm

The discretization does not differ from the $L^2 - L^2$ mass preservation and we inherit all proposed strategies from Section 4.5.3. Hence we obtain the following algorithm: The

Algorithm 6 $L^2 - TV$ Mass Preservation

L2TVMassPreservationU_1, U_2, α, η
1: $U_t \leftarrow$ calculateImageDerivatives(U_1,U_2)
2: $K \leftarrow$ buildOperator(U_1, U_2)
3: $\sigma, \tau \leftarrow \frac{1}{\sqrt{8}\max|u|}$
4: $v, \bar{v}, y \leftarrow 0$ $\epsilon < \eta$
5: $vOld \leftarrow v$
6: $\tilde{y} \leftarrow y + \sigma K \bar{v}$
7: $y_{1,2} \leftarrow \tilde{y}_{1,2} - \sigma S(\frac{\tilde{y}_{1,2}}{\sigma}, \frac{\alpha}{\sigma})$
8: $y_3 \leftarrow \frac{1}{\sigma+1}\tilde{y_3} + \frac{\sigma}{\sigma+1}U_t$
9: $v \leftarrow v - \sigma K^* y$
10: $\bar{v} \leftarrow 2 \cdot v - vOld$
11: v

algorithm runs efficiently, because the dual updates can be calculated directly and the matrix K is sparse and can be preassembled. The update in line 9 requires point-wise multiplication, addition and a maximum evaluation of two numbers, thus is fast to evaluate.

4.5.6. $L^1 - TV$ Optical Flow Model

Model

The quadratic L^2 norm for incorporating the optical flow equation in the variational model in general has the major drawback that outliners of the optical flow constraint are not handled robustly. To overcome this limitation Aubert, Deriche and Kornprobst analyzed different variational models and finally proved the advantages of an L^1 data fidelity term using rigorous theory of BV-functions [7]. Transfered to the concept of sparsity, the L^1 data term allows the optical flow constraint to be violated on a least sparse set. To overcome the other limitations of the classical $L^2 - L^2$ optical flow approach we adopt the ideas of the previous sections and propose a total variation regularization, which leads to the following model

Model 4.5.6. $L^1 - TV$ Optical Flow Model

$$\arg\min_{\boldsymbol{v}} \|u_t + \nabla u \cdot \boldsymbol{v}\|_1 + \alpha \sum_{i=1}^{d} |v_i|_{BV} .$$

Here, we expect a result that may violate the optical flow constraint in a sparse set which is admitted by the L^1 norm. Moreover we expect a piecewise constant motion field due to the total variation regularization.

Computational Minimization

For the finite dimensional setting the $L^1 - TV$ optical flow model reads

$$\arg\min_{\boldsymbol{v}} \|u_t + \nabla u \cdot \boldsymbol{v}\|_1 + \alpha \sum_{i=1}^{2} \|\nabla v_i\|_1 .$$

In order to minimize this energy we again propose a direct application of the previously introduced Chambolle–Pock method. As a first step the definitions are adjusted as follows:

$$F(K\boldsymbol{v}) := \alpha \sum_{i=1}^{d} \|\nabla v_i\|_1 ,$$

with

$$K\boldsymbol{v} = \begin{pmatrix} \nabla & 0 \\ 0 & \nabla \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \end{pmatrix}, \qquad K^*\boldsymbol{y} = - \begin{pmatrix} \nabla\cdot & 0 \\ 0 & \nabla\cdot \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} .$$

Similar to the $TV - L^2$ optical flow model we get the dual as:

$$F^*(\boldsymbol{y}) := \alpha \sum_{i=1}^{2} \delta_{B(L^\infty)}(y_i/\alpha).$$

Here, the norm for the data model has changed to L^1, but still no operator is applied to $\boldsymbol{v}$. Consequently, the optical flow part is put into the primal functional as follows:

$$G(\boldsymbol{v}) := \|u_t + \nabla_x u \cdot \boldsymbol{v}\|_1 .$$

We receive the following primal-dual formulation:

$$\arg\min_{\boldsymbol{v}} \arg\max_{\boldsymbol{y}} \|u_t + \nabla_x u \cdot \boldsymbol{v}\|_1 + \langle \boldsymbol{v}, K^*\boldsymbol{y} \rangle - \alpha \sum_{i=1}^{2} \delta_{B(L^\infty)}(y_i/\alpha).$$

Incorporating this in the algorithm yields the modified iterations

$$\tilde{\boldsymbol{y}}^{k+1} = \boldsymbol{y}^k + \sigma K \bar{\boldsymbol{v}}^{\boldsymbol{k}}$$

$$
\begin{aligned}
\boldsymbol{y}^{k+1} &= \arg\min_{\boldsymbol{y}=(y^1,y^2)} \left\{ \frac{1}{2} \left\| \boldsymbol{y} - \tilde{\boldsymbol{y}}^{k+1} \right\|_2^2 + \alpha\sigma\delta_{B(L^\infty)}(\boldsymbol{y}/\alpha) \right\} \\
\tilde{\boldsymbol{v}}^{k+1} &= \boldsymbol{v}^k + \tau K^* \boldsymbol{y}^{k+1} \\
\boldsymbol{v}^{k+1} &= \arg\min_{\boldsymbol{v}} \left\{ \frac{1}{2} \left\| \boldsymbol{v} - \tilde{\boldsymbol{v}}^{k+1} \right\|_2^2 + \tau \left\| u_t + \nabla u \cdot \boldsymbol{v} \right\|_1 \right\} \\
\bar{\boldsymbol{v}}^{k+1} &= \boldsymbol{v}^{k+1} + \theta(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k).
\end{aligned}
$$

Only the subproblem for $\boldsymbol{v}^{k+1}$ differs from the $L^2 - TV$ optical flow model and has to be handled differently. Here, we have an $L^1 - L^2$ optimization problem with an affine linear transformation of $\boldsymbol{v}$. The L^1 part $u_t + \nabla u \cdot \boldsymbol{v}$ contains for $\boldsymbol{v}$ only a pointwise multiplication with u_x resp. u_y, which is then shifted pointwise by u_t. Thus we can apply Lemma 2.2.29 (direct solution for affine linear L^1 problems). We set

$$
\rho(\boldsymbol{v}) := u_t + \nabla u \cdot \boldsymbol{v}, \quad \boldsymbol{\beta} := (u_x, u_y).
$$

Then the solution is given by

$$
\boldsymbol{v} = \tilde{\boldsymbol{v}}^{k+1} + \begin{cases} \tau\boldsymbol{\beta} & \text{if } \rho(\tilde{\boldsymbol{v}}^{k+1}) < -\tau \left\|\boldsymbol{\beta}\right\|_2^2 \\ -\tau\boldsymbol{\beta} & \text{if } \rho(\tilde{\boldsymbol{v}}^{k+1}) > \tau \left\|\boldsymbol{\beta}\right\|_2^2 \\ -\frac{\rho(\tilde{\boldsymbol{v}}^{k+1})}{\left\|\boldsymbol{\beta}\right\|_2^2}\boldsymbol{\beta} & \text{else} \end{cases},
$$

and we are able to solve all subproblems directly and we obtain the following scheme:

$$
\begin{aligned}
\tilde{\boldsymbol{y}}^{k+1} &= \boldsymbol{y}^k + \sigma K \bar{\boldsymbol{v}}^{\boldsymbol{k}} \\
\boldsymbol{y}_{k+1} &= \pi_\alpha(\tilde{\boldsymbol{y}}^{k+1}) \\
\tilde{\boldsymbol{v}}^{k+1} &= \boldsymbol{v}^k - \tau K^* \boldsymbol{y}^{k+1} \\
\boldsymbol{v}^{k+1} &= \tilde{\boldsymbol{v}}^{k+1} + \begin{cases} \tau\boldsymbol{\beta} & \text{if } \rho(\tilde{\boldsymbol{v}}^{k+1}) < -\tau \left\|\boldsymbol{\beta}\right\|_2^2 \\ -\tau\boldsymbol{\beta} & \text{if } \rho(\tilde{\boldsymbol{v}}^{k+1}) > \tau \left\|\boldsymbol{\beta}\right\|_2^2 \\ -\frac{\rho(\tilde{\boldsymbol{v}}^{k+1})}{\left\|\boldsymbol{\beta}\right\|_2^2}\boldsymbol{\beta} & \text{else} \end{cases} \\
\bar{\boldsymbol{v}}^{k+1} &= \boldsymbol{v}^{k+1} + \theta(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k)
\end{aligned}
$$

Implementation and Algorithm

Due to the fact that the resulting minimization differs only in minor aspects from the $L^2 - TV$ optical flow model, we inherit the same discretization techniques. This leads

to the following algorithm. The computational complexity for solving the $L^1 - TV$

Algorithm 7 $L^1 - TV$ Optical Flow

L1TVOpticalFlowU_1, U_2, α, η
1: $U_t, U_x, U_y \leftarrow$ calculateImageDerivatives(U_1,U_2)
2: $K \leftarrow$ buildOperator()
3: $\sigma, \tau \leftarrow \frac{1}{\sqrt{8}}$
4: $v, \bar{v}, y \leftarrow 0$ $\epsilon < \eta$
5: $vOld \leftarrow v$
6: $\tilde{y} \leftarrow y + \sigma K \bar{v}$
7: $y \leftarrow \tilde{y} - \sigma S(\frac{\tilde{y}}{\sigma}, \frac{\alpha}{\sigma})$
8: $\tilde{v} \leftarrow v - \sigma K^* y$
9: $v \leftarrow affineShrink(\tilde{v}, \tau)$
10: $\bar{v} \leftarrow 2 \cdot v - vOld$
11: v

optical flow problem only lies slightly above the $L^2 - L^2$ optical flow resp. $L^2 - TV$ optical flow model. Still evaluations of the forward operator K and backwards operator K^* are numerically expensive. The shrinkage formula can be evaluated by only a few operations. For the affine shrinkage the quadratic L^2-norm is used, so no square root has to be calculated. The rest consists of case analysis and a few multiplications resp. additions. The algorithm can furthermore be parallelized to gain additional speed. Here, lines 7-9 and 10-12 form independent blocks.

4.5.7. $L^1 - TV$ Mass Preservation Model

Model

The final model introduced in this section is the $TV - L^1$ mass preservation model. We use the mass preservation constraint $u_t + \nabla \cdot (u\boldsymbol{v})$ as a data fidelity part for the unknown velocity field $\boldsymbol{v} = (v^1, v^2)$, now penalized in the L^1 norm (instead of the former L^2 norm) to better handle outliners in the equation. The total variation is again used as a regularizer and we obtain the following model.

Model 4.5.7. $L^1 - TV$ Mass Preservation Model

$$\arg\min_{\boldsymbol{v}} \left\| u_t + \nabla \cdot (u\boldsymbol{v}) \right\|_1 + \alpha \sum_{i=1}^{d} |v_i|_{BV} .$$

This $L^1 - TV$ mass preservation model starts with the L^1 mass preservation data term, which allows the constraint to be vulnerated in a sparse set. Moreover total variation is used as a regularizer to emphasize velocity fields consisting of constant areas with a sparse set of jumps between them.

Computational Minimization

We minimize the finite dimensional counterpart of the previous model, which reads

$$\arg\min_{\boldsymbol{v}} \left\| u_t + \nabla \cdot (u\boldsymbol{v}) \right\|_1 + \alpha \sum_{i=1}^{d} \left\| \nabla v_i \right\|_1 .$$

The complete energy is now put into the dual operator as follows:

$$F(K\boldsymbol{v}) := \left\| u_t + \nabla \cdot (u\boldsymbol{v}) \right\|_1 + \alpha \sum_{i=1}^{d} \left\| \nabla v_i \right\|_1 ,$$

with the linear operator K given by

$$K\boldsymbol{v} = \begin{pmatrix} \nabla & 0 \\ 0 & \nabla \\ \partial_x u & \partial_y u \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \end{pmatrix} .$$

The adjoint operator for the dual $\boldsymbol{y} = (y_1, y_2, y_3)^t$ is now given by

$$K^* \boldsymbol{y} = - \begin{pmatrix} \nabla\cdot & 0 & u\partial_x \\ 0 & \nabla\cdot & u\partial_y \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} .$$

The corresponding dual functional $F^*(\boldsymbol{y})$ to F can be calculated using Examples 2.2.23 and 2.2.25 as:

$$F^*(\boldsymbol{y}) := \delta_{B(L^\infty)}(y_3) - \langle u_t, y_3 \rangle + \alpha \sum_{i=1}^{2} \delta_{B(L^\infty)}(y_i/\alpha).$$

Incorporating this into the Chambolle–Pock framework leads to the scheme

$$\begin{aligned} \tilde{\boldsymbol{y}}^{k+1} &= \boldsymbol{y}^k + \sigma K \bar{\boldsymbol{v}} \\ \boldsymbol{y}_{1,2}^{k+1} &= \arg\min_{\boldsymbol{y}=(y^1,y^2)} \left\{ \frac{1}{2} \left\| \boldsymbol{y} - \tilde{\boldsymbol{y}}_{1,2}^{k+1} \right\|_2^2 + \alpha\sigma \delta_{B(L^\infty)}(\boldsymbol{y}/\alpha) \right\} \end{aligned}$$

$$
\begin{aligned}
y_3^{k+1} &= \arg\min_y \left\{ \frac{1}{2} \left\| y - \tilde{y}_3^{k+1} \right\|_2^2 + \sigma \delta_{B(L^\infty)}(y) - \sigma \langle u_t, y \rangle \right\} \\
\boldsymbol{v}^{k+1} &= \arg\min_{\boldsymbol{v}} \left\{ \int_0^T \frac{1}{2} \left\| \boldsymbol{v} - (\boldsymbol{v}^k - \tau K^* \boldsymbol{y}) \right\|_2^2 \right\} \\
\bar{\boldsymbol{v}}^{k+1} &= \boldsymbol{v}^{k+1} + \theta(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k).
\end{aligned}
$$

Except for the subproblem in y_3 all results from the $L^2 - TV$ mass preservation section can be inherited. Let us first rewrite the problem for y_3^{k+1}:

$$
\begin{aligned}
&\arg\min_y \left\{ \frac{1}{2} \left\| y - \tilde{y}_3^{k+1} \right\|_2^2 + \sigma \delta_{B(L^\infty)}(y_3) - \sigma \langle u_t, y \rangle \right\} \\
\Leftrightarrow &\arg\min_y \left\{ \frac{1}{2} \langle y - \tilde{y}_3^{k+1}, y - \tilde{y}_3^{k+1} \rangle + \sigma \delta_{B(L^\infty)}(y_3) - \langle \sigma u_t, y \rangle \right\} \\
\Leftrightarrow &\arg\min_y \left\{ \frac{1}{2} \langle y, y \rangle - \langle y, \tilde{y}_3^{k+1} \rangle + \frac{1}{2} \langle \tilde{y}_3^{k+1}, \tilde{y}_3^{k+1} \rangle + \sigma \delta_{B(L^\infty)}(y_3) - \langle \sigma u_t, y \rangle \right\} \\
\Leftrightarrow &\arg\min_y \left\{ \frac{1}{2} \langle y, y \rangle - \langle y, \tilde{y}_3^{k+1} + \sigma u_t \rangle + \frac{1}{2} \langle \tilde{y}_3^{k+1} + \sigma u_t, \tilde{y}_3^{k+1} + \sigma u_t \rangle + \sigma \delta_{B(L^\infty)}(y_3) \right\} \\
\Leftrightarrow &\arg\min_y \left\{ \frac{1}{2} \langle y - (\tilde{y}_3^{k+1} + \sigma u_t), y - (\tilde{y}_3^{k+1} + \sigma u_t) \rangle + \sigma \delta_{B(L^\infty)}(y_3) \right\} \\
\Leftrightarrow &\arg\min_y \left\{ \frac{1}{2} \left\| y - (\tilde{y}_3^{k+1} + \sigma u_t) \right\|_2^2 + \sigma \delta_{B(L^\infty)}(y_3) \right\}
\end{aligned}
$$

Now, the solution is simply given by the projection of $\tilde{y}_3^{k+1} + \sigma u_t$ onto $[-1, 1]$:

$$
y_3^{k+1} = \pi_1(\tilde{y}_3^{k+1} + \sigma u_t)
$$

This leads to the following scheme:

$$
\begin{aligned}
\tilde{\boldsymbol{y}}^{k+1} &= \boldsymbol{y}^k + \sigma K \bar{\boldsymbol{v}}^{\boldsymbol{k}} \\
y_{1,2}^{k+1} &= \pi_\alpha(\tilde{y}_{1,2}^{k+1}) \\
y_3^{k+1} &= \pi_1(\tilde{y}_3^{k+1} + \sigma u_t) \\
\boldsymbol{v}^{k+1} &= \boldsymbol{v}^k - \tau K^* \boldsymbol{y} \\
\bar{\boldsymbol{v}}^{k+1} &= \boldsymbol{v}^{k+1} + \theta(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k)
\end{aligned}
$$

and we are able to solve this problem directly.

Implementation and Algorithm

The discretization is done similar to the $L^1 - TV$ mass preservation model, because only slight changes to the norms were made. We end up with the following algorithm: We inherit the runtime argumentation from the $TV - L^1$ optical flow part. The only

Algorithm 8 $L^1 - TV$ Mass Preservation

L1TVMassPreservationU_1, U_2, α, η
$U_t \leftarrow$ calculateImageDerivatives(U_1,U_2)
$K \leftarrow$ buildOperator(U_1, U_2)
$\sigma, \tau \leftarrow \frac{1}{\sqrt{8}\max|u|}$
$v, \bar{v}, y \leftarrow 0$ $\epsilon < \eta$
$vOld \leftarrow v$
$\tilde{y} \leftarrow y + \sigma K \bar{v}$
$y_{1,2} \leftarrow \tilde{y}_{1,2} - \sigma S(\frac{\tilde{y}_{1,2}}{\sigma}, \frac{\alpha}{\sigma})$
$y_3 \leftarrow affineShrink(\tilde{y}_3, \frac{\alpha}{\sigma})$
$v \leftarrow v - \sigma K^* y$
$\bar{v} \leftarrow 2 \cdot v - vOld$
v

difference lies in the extended affine shrinkage which has more variables involved now. The numerical effort increases linear with the number of variables involved in the shrinkage, thus is still very efficient.

4.6. Numerical Evaluation

The following section is dedicated to the evaluation of the previously introduced motion estimation models. In the first part, we want to elucidate that quantifying results from motion estimation is a non-trivial problem. Afterwards, we introduce several error measures for motion fields and apply our models to different types of image sequences. Finally, our models are evaluated in terms of influence of noise and runtime.

4.6.1. Error Measures for Velocity Fields

Finding error measures for velocity fields is a delicate problem which can be motivated by the following simple example. Consider two images, each consisting of only 3×3

pixels of the following form:

$$u_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad u_2 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{4.31}$$

From our point of view we seek for a velocity field $\boldsymbol{v} = (v^1, v^2)$ that transforms u_1 into u_2. Unfortunately it is unclear which velocity field underlies this motion. Examples of possible solutions are

1. Just move the pixels at position $(1,1)$ and $(1,2)$ by 1 to the bottom, hence

$$v^1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad v^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

2. Move the pixels at position $(1,1)$ and $(1,2)$ by 1 to the bottom and change their position:

$$v^1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad v^2 = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

3. Move the whole image by 1 to the bottom which gives

$$v^1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad v^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

4. Move the pixels at position $(1,1)$ and $(1,2)$ by 1 to the bottom and exchange some pixels in the third column

$$v^1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & -2 \end{pmatrix}, \quad v^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

All these velocity fields produce the same result, that is they all fulfill the optical flow constraint $u_t + \nabla u \cdot \boldsymbol{v} = 0$ resp. the mass preservation constraint $u_t + \nabla \cdot (u\boldsymbol{v}) = 0$.

This results from the fact that for given images the optical flow constraint (similar to the mass preservation constraint) states for every point $x \in \Omega$ only one equation for a 2-dimensional velocity field $\boldsymbol{v}$. The solution is unique iff there exists a unique bijection between u_1 and u_2. This requires both images to consist of the same intensity values, which furthermore have to be unique. Unfortunately this assumption is far from reality, since regions of constant intensity are characteristic for background or objects. Hence, for practical problems we have a highly underdetermined system and consequently a huge variety of possible underlying velocity fields $\boldsymbol{v}$.
As we will see later an error measure always explicitly or implicitly favors one of these possible results over the others. Those error measures, which explicitly prefer one result, expect a given ground truth field $\boldsymbol{v}_{GT} = (v^1_{GT}, v^2_{GT})$ and measure a distance $d(\boldsymbol{v}_{GT}, \boldsymbol{v})$ from the calculated velocity field to the given ground truth field. This strategy is questionable because in real world examples we usually do not have a ground truth velocity field, and artificial examples can usually be generated by several different ground truth fields. Hence, setting the ground truth $\boldsymbol{v}_{GT}$ directly favors a subjective result. This is at least from the mathematical viewpoint questionable.

Absolute Endpoint Error

Despite the fact that explicit error measures might be problematic they are often used in the literature and we will also use them to evaluate our algorithms. In [10] two explicit error measures have been presented. The most intuitive one is the average endpoint error (aee), proposed in [55], which is the vector-wise Euclidean norm of the difference vector $\boldsymbol{v} - \boldsymbol{v}_{GT}$. The difference is divided by $|\Omega|$ and we have

$$aee := \frac{1}{|\Omega|} \int_\Omega \sqrt{(v^1(x) - v^1_{GT}(x))^2 + (v^2(x) - v^2_{GT}(x))^2}\, \mathrm{d}x,$$

or in a discrete formulation

$$AEE := \frac{1}{nPx} \sum_{i=1}^{nPx} \sqrt{(v^1(i) - v^1_{GT}(i))^2 + (v^2(i) - v^2_{GT}(i))^2}.$$

Here, nPx denotes the number of pixels.

Angular Error

A second measure states the angular error (ae) which goes back to the work of Fleet and Jepson [37] and a survey of Barron *et al.* [12]. Here $\boldsymbol{v}$ and $\boldsymbol{v}_{GT}$ are projected into

the 3-D space (to avoid division by zero) and normalized by

$$\hat{\boldsymbol{v}} := \frac{(v^1, v^2, 1)}{\sqrt{\|\boldsymbol{v}\|^2 + 1}}, \qquad \hat{\boldsymbol{v}}_{GT} := \frac{(v^1_{GT}, v^2_{GT}, 1)}{\sqrt{\|\boldsymbol{v}_{GT}\|^2 + 1}}.$$

The error is then calculated measuring the angle between $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{v}}_{GT}$ in the continuous setting as

$$ae := \frac{1}{|\Omega|} \int_\Omega \arccos(\hat{\boldsymbol{v}}(x) \cdot \hat{\boldsymbol{v}}_{GT}(x)) \, \mathrm{d}x,$$

and in a discrete setting as

$$AE := \frac{1}{nPx} \sum_{i=1}^{nPx} \arccos(\hat{\boldsymbol{v}}(i) \cdot \hat{\boldsymbol{v}}_{GT}(i)).$$

Besides the fact that the ae is very popular, we want to emphasize a serious drawback. To illustrate this drawback we took a vector $\boldsymbol{v}_{GT} = (v^1, v^2)$ and denote the Euclidean norm on the x-axis in Figure 4.7. This vector has been disturbed by an absolute and relative error of 0.01 to create $\boldsymbol{v}_{abs}$ and $\boldsymbol{v}_{rel}$:

$$\begin{aligned} \boldsymbol{v}_{abs} &= \boldsymbol{v}_{GT} - (0.99, 0.99)^T \\ \boldsymbol{v}_{rel} &= \frac{\boldsymbol{v}_{GT}}{\|\boldsymbol{v}_{GT}\|_2} \cdot 0.99. \end{aligned}$$

Afterwards we calculated the angular error between $\boldsymbol{v}$ and $\boldsymbol{v}_{abs}$ resp. $\boldsymbol{v}$ and $\boldsymbol{v}_{rel}$ and denoted this error on the $y - axis$ of Figure 4.7. The first obvious property is that the error shrinks for larger velocities. This is critical since larger velocities are usually coupled with the object we are interested in. In the contrary absolute errors in small velocities (e.g. background) are over-penalized. To make this clear, let us consider a simple example: Take an algorithm producing a all-zero velocity field. This result perfectly recovers the background, but causes errors in the region of a moving object. Consider on the other hand an algorithm that perfectly recovers the movement in the background, but introduces slight errors in parts of the object. Due to the fact that errors in the background are over-penalized, the first result might be preferred although it yields a worse result.

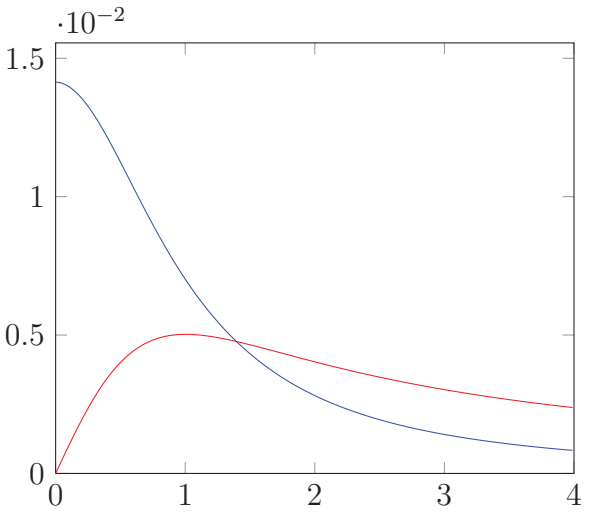


Figure 4.7.: Plot of the angular error ae on the y-axis with increasing $\|\boldsymbol{v}\|$ on the x-axis.
Blue: Absolute error $\|\boldsymbol{v} - \boldsymbol{v}_{GT}\|_2$ of 0.01
Red: Relative error $\|\boldsymbol{v} - \boldsymbol{v}_{GT}\|_2 / \|\boldsymbol{v}_{GT}\|$ of 0.01

SSD Interpolation Error

We already mentioned that it is questionable to predetermine a certain velocity field as ground truth. For given images u_1 and u_2 it is reasonable using the calculated motion field to $\boldsymbol{v}$ interpolate u_1 and compare the difference to u_2. Therefore we define the morphed image $\tilde{u}_2$ as

$$\tilde{u}_2 := u_1(x + \boldsymbol{v}),$$

which requires an interpolation scheme for which we use a **bicubic** one. Then we can measure the distance between the *morphed* image $\tilde{u}_2$ and the ground truth u_2 in a continuous setting by

$$ie := \sqrt{\frac{1}{|\Omega|} \int_\Omega (\tilde{u}_2(x) - u_2(x))^2 \, \mathrm{d}x},$$

or discrete as

$$IE := \sqrt{\frac{1}{nPx} \sum_{i=1}^{nPx} (\tilde{u}_2(i) - u_2(i))^2}.$$

This root-mean-squared difference is a simplified version of the interpolation error proposed by Baker *et al.* in their flow evaluation survey [10]. Szeliski proposed a weighted interpolation error, to compensate numerical effects [67]. He claims gradient approximation of images in the optical flow context to be noisy itself and introduces a weight corresponding to the norm of the image gradient. The resulting error measure is called the *normalized interpolation error*

$$nie := \sqrt{\frac{1}{|\Omega|}\int_{\Omega} \frac{(\tilde{u}_2(x) - u_2(x))^2}{\left\|\nabla u_2(x)\right\|^2 + 1}\,\mathrm{d}x},$$

resp.

$$NIE := \sqrt{\frac{1}{nPx}\sum_{i=1}^{nPx} \frac{(\tilde{u}_2(i) - u_2(i))^2}{\left\|\nabla u_2(i)\right\|^2 + 1}}.$$

By this choice the error is weighted less close to edges, because $\left\|\nabla u_2(x)\right\|^2$ increases and the difference is divided by a larger value then. As the distance from the edge increases, $\left\|\nabla u_2(i)\right\|^2 + 1$ tends to 1 and no weight is applied to the distance. But evaluating our algorithms with this measure might be problematic too because it makes no assumptions on the velocity field, except for the fact that it should transform u_1 into u_2. If for example a white dot on a black ground changes its position from u_1 to u_2, we would expect one block of constant movement, but this is not considered by the *ie* resp. the *nie*. The motion is allowed to be absolutely chaotic and might still have an error of zero, as long as it transforms u_1 into u_2.

Interpolated Level Line Error

In this thesis we propose a slightly advanced error measure based on interpolated images. Let therefore Γ_1, Γ_2 be the set of boundaries of u_1 and u_2. Based on this we want to calculate level set functions φ_1 and φ_2 as the solution of the parabolic Eikonal equation:

$$\begin{aligned}
\frac{\partial \varphi_1}{\partial t} &= \left\|\nabla \varphi_1\right\|, \quad \varphi_1(x,t) = 0 \text{ for } x \in \Gamma_1, \\
\frac{\partial \varphi_2}{\partial t} &= \left\|\nabla \varphi_2\right\|, \quad \varphi_2(x,t) = 0 \text{ for } x \in \Gamma_2.
\end{aligned}$$

Intuitively speaking, the curve Γ_i moves with speed 1 towards normal direction and $\varphi_i(x)$ is the arrival time of Γ_i at point x. Consequently, the level set function $\varphi_i : \Omega \to \mathbb{R}$

denotes the distance of the closest boundary point. This concept has been introduced by Sethian and Osher [54]. Having calculated the level set representations of u_1 and u_2 we proceed with the idea of interpolation, but apply this to the level set function φ_1. Therefore we define

$$\tilde{\varphi_2} = \varphi_1(x + \boldsymbol{v}),$$

which is calculated using spline interpolation. The function $\tilde{\varphi_2}$ can be interpreted as an interpolated level set function of φ_1 based on the estimated velocity field $\boldsymbol{v}$. The level set functions $\tilde{\varphi_2}$ and φ_2 are now compared using a weighted root-mean-squared difference, as introduced in Section 4.6.1:

$$ille := \sqrt{\frac{1}{|\Omega|}\int_\Omega (\tilde{\varphi_2}(x) - \varphi_2(x))^2 w(x)\,\mathrm{d}x},$$

or discrete as

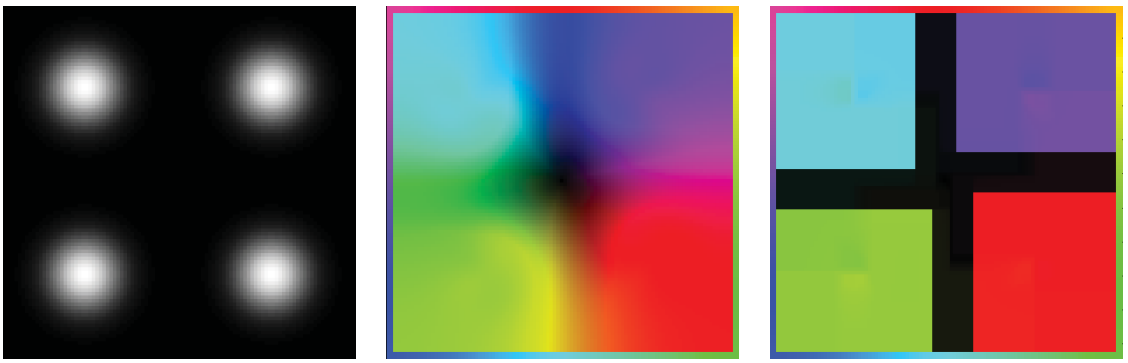
$$ILLE := \sqrt{\frac{1}{nPx}\sum_{i=1}^{nPx} (\tilde{\varphi_2}(i) - \varphi_2(i))^2 W(i)}.$$

Since the evaluation of the level set function φ denotes the distance to the boundary, we use $w(x) := \frac{1}{|\varphi_2(x)|+\epsilon}$ as a weighting function. By this choice we emphasize $\boldsymbol{v}$ to correctly transport the boundaries of u_1 onto u_2, but moreover to transport the surrounding image information in a reasonable way.
The proposed measure satisfies this requirement because the level lines of φ_1 have to be shifted correctly to achieve a small *ille*. The parameter ϵ can be adjusted, to amplify the measure close to the boundary. In our tests we choose $\epsilon = 1$.

4.6.2. Comparison of Regularizers

We introduced a total of six different models for motion estimation, which differ in the choice of the constraint in the data term (optical flow and mass preservation), the norm of the data term and the regularization term (L^2 resp. L^1 norm of the gradient). Figure 4.8 shows a motion field generated by the $L^2 - L^2$ optical flow model (Section 4.5.2), and a field generated by the $L^1 - TV$ optical flow model (Section 4.5.6).

In the middle image we see the expected smooth result from the L^2 regularization of the gradient. The movement has the right scale in the center of the object, but is



(a) $L^2 - L^2$ Motion Field (b) $L^2 - L^2$ X-Component (c) $L^2 - L^2$ Y-Component

Figure 4.8.: Typical motion fields generated by different regularizers. Left: Image to calculate motion on; Middle: typical velocity field generated by L^2 regularization; Right: typical velocity field generated by TV regularization.

also smoothed out to the sides. No motion is detected for the background. The bottom row is a typical representative for a total variation regularization. Each component has a constant block of motion that is separated from the background by a sharp edge. Besides this we see that the movement reaches also the sides of the images. This is due to the static background which fulfills any motion equation, and the total variation (which is equal to the length of the edge multiplied with the jump height) that is smaller if the edge is drawn directly to the sides.

4.6.3. Test Data

As a start, the six introduced algorithms have been tested on the basic flow types introduced in Figure 4.4, namely translation, rotation and scaling.
This basic evaluation gives some insight into the general quality of the different types of algorithms, however the basic flow types do not represent scenes close to the reality. Therefore we extend the set of evaluation scenes by the IPOL database [44]. An overview of the evaluation scenes from the IPOL database can be found in Figure 4.13.

4.6.4. Evaluation for Basic Flow Types

As mentioned before we want to begin our evaluation with the three basic types of flow fields. Figure 4.9 contains MATLABs **Lena** image, which was always used as the first image u_1 for our algorithm. Then we created the second image u_2 by cubic interpolation of $u_1(\boldsymbol{x} + \boldsymbol{v})$, where $\boldsymbol{v}$ represents one of the ground truth flows also shown in

Algorithm	AEE	AE	IE	NIE	ILLE
$L^2 - L^2$ optical flow	0.02	0.017	0.002	9.5e-05	0.079
$L^2 - L^2$ mass preservation	0.025	0.021	0.0028	9.6e-05	0.082
$L^2 - TV$ optical flow	**0.004**	**0.003**	**0.0008**	0.000101	0.077
$L^2 - TV$ mass preservation	0.004	0.003	0.0008	0.0001	0.076
$L^1 - TV$ optical flow	0.02	0.02	0.0026	**7.6e-05**	**0.067**
$L^1 - TV$ mass preservation	0.086	0.076	0.0087	0.000159	0.086

Table 4.1.: Evaluation for translation. Table shows minimal possible errors that could be found for the example in Figure 4.10.

Figure 4.9. Due to the chosen discretization we are only able to exactly detect motion with a magnitude of one pixel at most, consequently the ground truth flow fields are chosen accordingly. We want to underline that for every example each of the available algorithms was run with a broad set of regularization weights to find the lowest possible error.

Translation:
For the translation flow the whole **Lena** image was shifted by one pixel to the right. Whereby the ground truth flow consists of a block of constant area of magnitude one. The detailed evaluation results can be found in Table 4.1. Furthermore Figure 4.10 shows the best result (in terms of the AE) each of the algorithms was able to generate. We see that the $L^2 - TV$ models minimize AEE, AE and IE. This was expected, since the ground truth flow consists of one large block of constant intensity 1 and the AEE and AE use this ground truth flow for determining the error. The IE is also minimized because the $L^2 - TV$ optical flow model recovers one block of constant magnitude that shifts the first image by one pixel to the right. This gives a very good interpolation. The $L^2 - TV$ mass preservation model produces slightly worse results than the optical flow model, followed by the $L^2 - L^2$ optical flow and the $L^1 - TV$ optical flow model.
Rotation:
As a ground truth flow for the rotation example we chose the rotation angle such that the magnitude of the flow lies below 1 everywhere. The color-coded ground truth can again be found on Figure 4.9. We see that the color-intensity decreases towards the center, which comes from the fact that an even rotation around one central point requires less magnitude of the flow field the closer the center is. Table 4.2 gives an overview of the respectively lowest possible errors each of the algorithms was able to generate. Figure 4.11 shows, exemplarily for the AE, the best results our algorithms were able to generate. Here, the $L^2 - L^2$ optical flow model is able to produce the smallest values

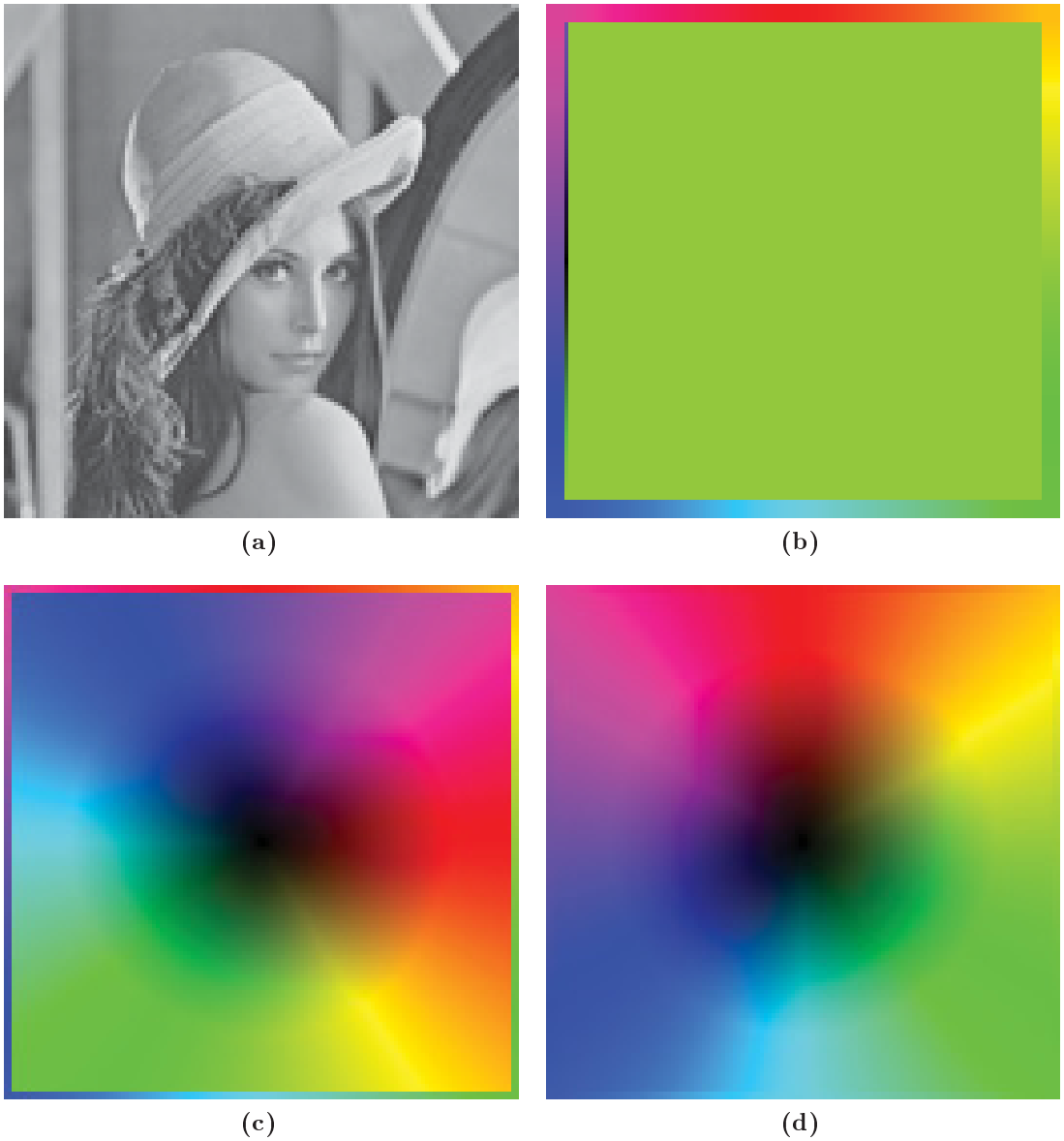


Figure 4.9.: (a) Lena image from MATLAB, (b) Translation ground truth, (c) Rotation ground truth, (d) Scaling ground truth

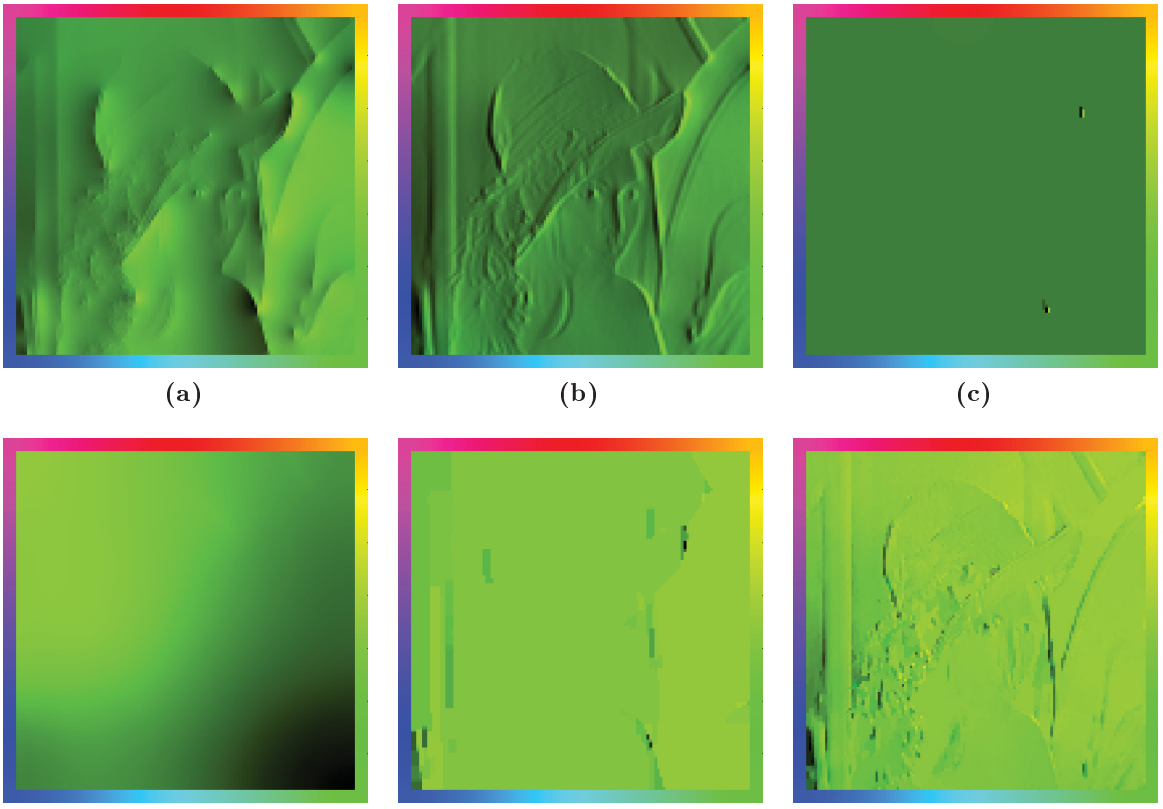


Figure 4.10.: Overview of the best results (in terms of the AE) our algorithms were able to generate for the translation flow. (a) $L^2 - L^2$ optical flow, (b) $L^2 - L^2$ mass preservation, (c) $L^2 - TV$ optical flow, (d) $L^2 - TV$ mass preservation, (e) $L^1 - TV$ optical flow, (f) $L^1 - TV$ mass preservation

Algorithm	AEE	AE	IE	NIE	ILLE
$L^2 - L^2$ optical flow	**0.059**	**0.038**	**0.0044**	**7.6e-05**	**0.249**
$L^2 - L^2$ mass preservation	0.073	0.047	0.0046	7.8e-05	0.252
$L^2 - TV$ optical flow	0.104	0.067	0.0053	9e-05	0.251
$L^2 - TV$ mass preservation	0.109	0.069	0.0059	0.000102	0.255
$L^1 - TV$ optical flow	0.087	0.056	0.0051	8.3e-05	0.249
$L^1 - TV$ mass preservation	0.1	0.064	0.0054	8.7e-05	0.251

Table 4.2.: Evaluation for rotation. Table shows minimal possible errors that could be found for the example in Figure 4.11.

Algorithm	AEE	AE	IE	NIE	ILLE
$L^2 - L^2$ optical flow	**0.057**	**0.037**	**0.0041**	**4.9e-05**	0.257
$L^2 - L^2$ mass preservation	0.158	0.096	0.0053	6.5e-05	0.269
$L^2 - TV$ optical flow	0.095	0.06	0.0051	6.9e-05	0.258
$L^2 - TV$ mass preservation	0.212	0.127	0.0069	9.4e-05	0.274
$L^1 - TV$ optical flow	0.086	0.054	0.0053	7.9e-05	**0.256**
$L^1 - TV$ mass preservation	0.329	0.188	0.0088	0.000118	0.285

Table 4.3.: Evaluation for scaling. Table shows minimal possible errors that could be found for the example in Figure 4.12.

in terms of all evaluated error measures. The $L^2 - L^2$ mass preservation model follows the optical flow model, and afterwards an L^2-data term coupled with an L^2 regularizer fits best to this situation. This is not surprising when looking at the ground truth field in Figure 4.9, which is completely smooth.

Scaling:

The ground truth flow field for scaling was chosen, similar to the rotation, such that the overall magnitude is bounded by 1. At first glance rotation and scaling look very similar but they describe a completely different phenomenon. For scaling the direction of the flow is always away from the central point towards the boundary. This becomes clear when looking at Figure 4.9. We see from Table 4.3 that the $L^2 - L^2$ optical flow model again produces the overall best result. Similar to the previous flow types we plotted the best results in terms of the AE in Figure 4.12. The reason for the outstanding result of the $L^2 - L^2$ model is again the completely smooth ground truth field.

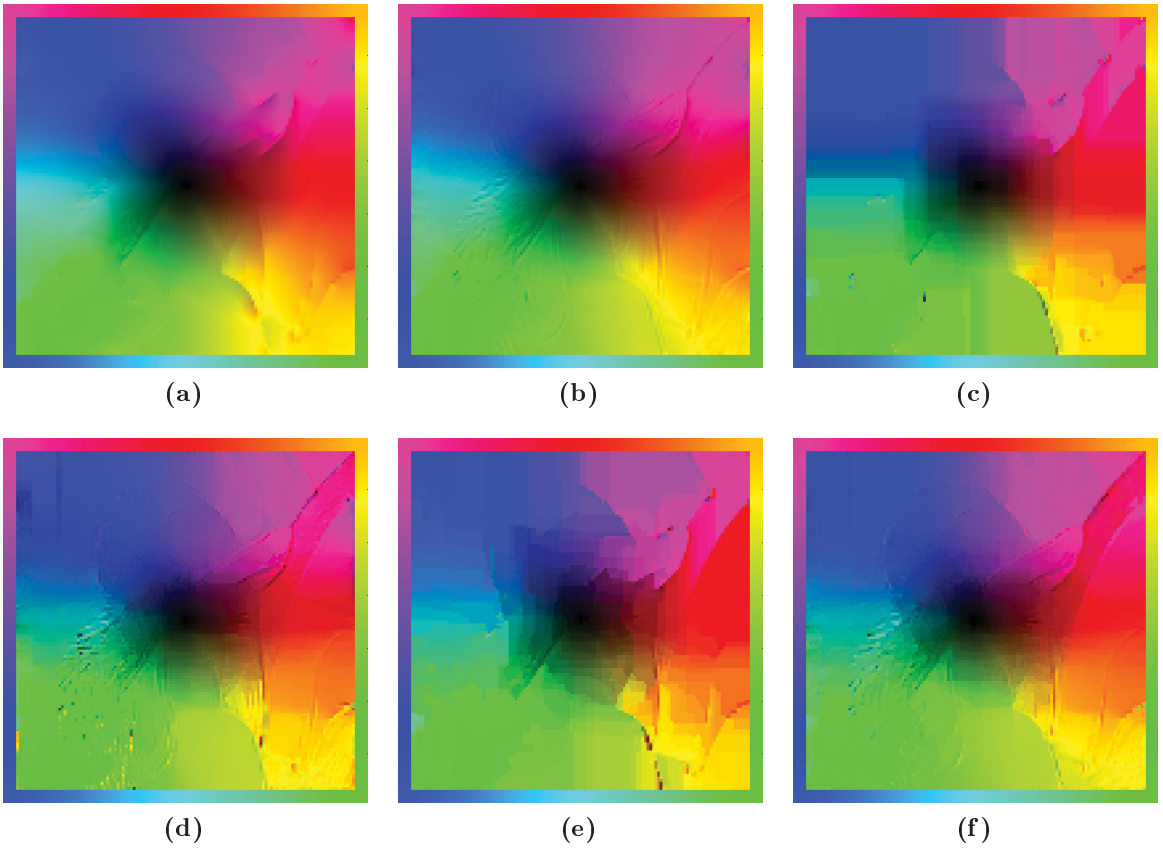
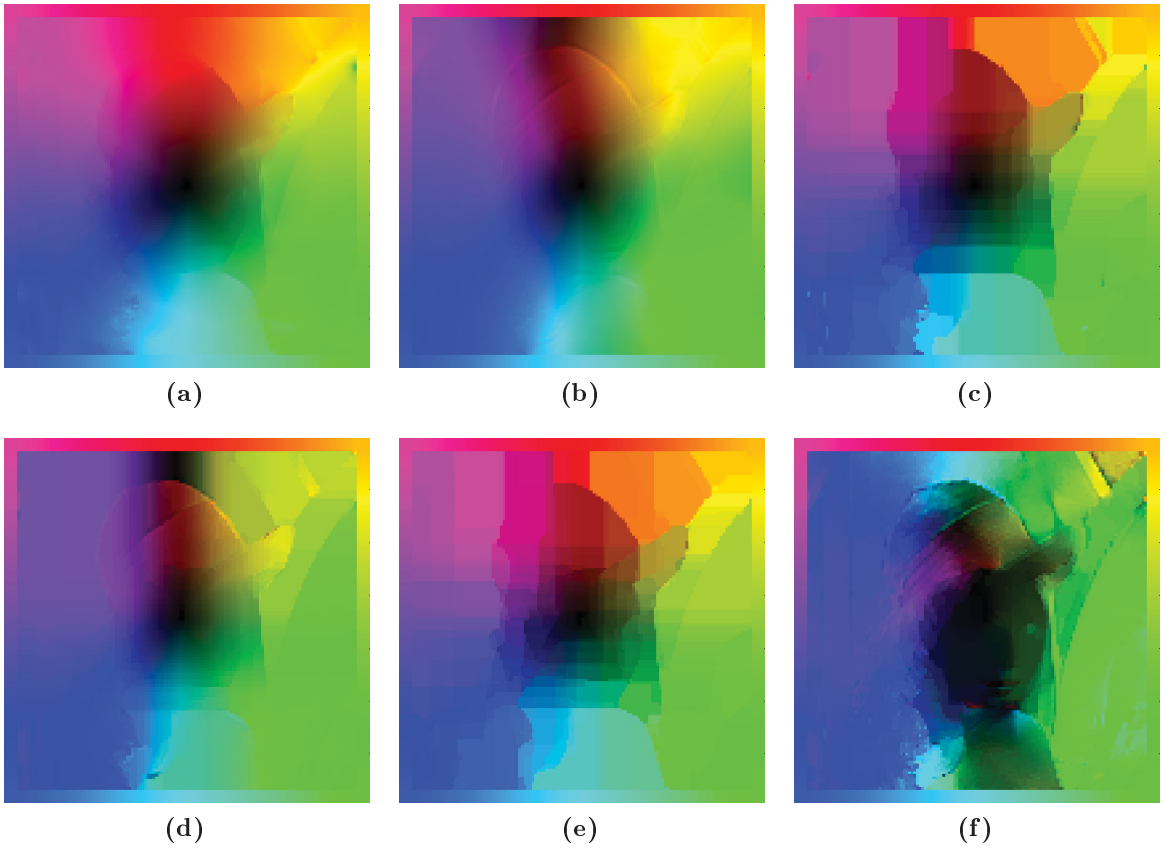


Figure 4.11.: Overview of the best results (in terms of the AE) our algorithms were able to generate for the rotation flow. (a) $L^2 - L^2$ optical flow, (b) $L^2 - L^2$ mass preservation, (c) $L^2 - TV$ optical flow, (d) $L^2 - TV$ mass preservation, (e) $L^1 - TV$ optical flow, (f) $L^1 - TV$ mass preservation



(a) (b) (c)

(d) (e) (f)

Figure 4.12.: Overview of the best results (in terms of the AE) our algorithms were able to generate for the scaling flow. (a) $L^2 - L^2$ optical flow, (b) $L^2 - L^2$ mass preservation, (c) $L^2 - TV$ optical flow, (d) $L^2 - TV$ mass preservation, (e) $L^1 - TV$ optical flow, (f) $L^1 - TV$ mass preservation

Algorithm	AEE	AE	IE	NIE	ILLE
$L^2 - L^2$ optical flow	**0.07**	**0.042**	**0.0025**	**2e-05**	**0.177**
$L^2 - L^2$ mass preservation	0.092	0.054	0.0028	2.3e-05	0.179
$L^2 - TV$ optical flow	0.1	0.062	0.0028	2.1e-05	0.179
$L^2 - TV$ mass preservation	0.137	0.081	0.0034	2.9e-05	0.182
$L^1 - TV$ optical flow	0.076	0.045	0.0025	2.1e-05	0.177
$L^1 - TV$ mass preservation	0.105	0.063	0.0031	2.7e-05	0.179

Table 4.4.: Dimetrodon dataset. Table shows minimal possible errors that could be found.

4.6.5. Evaluation for Real Data

Coming from the basic flow types we want to go over to more challenging evaluation data. Therefore we used slightly modified image sequences from the IPOL [44] database. Since our motion estimation algorithms are limited to a discrete length of one pixel we scaled down the IPOL ground truth fields to a length of 1 and used this new ground truth field to generate a second image $u_2 = I(\boldsymbol{x} - \boldsymbol{v})$ by cubic interpolation.
We want to underline that these modified sequences still cover a variety of difficulties such as several independently moving objects, nonrigid motion, thin structures, movement in regions with little contrast and a variety of photometric effects. Figure 4.13 shows an overview of the image scenes.

Dimetrodon:
Starting with the results from the Dimetrodon scene, we see from Table 4.4 that the $L^2 - L^2$ optical flow model performs slightly better than the $L^1 - TV$ model for the AE, AEE and NIE. Both models perform the same in terms of IE and ILLE. Both models differ in data term and regularizer from each other, but are able to minimize the errors in the same way. This can be explained by the ground truth flow in Figure 4.13, which consists of partly smooth areas and some small discontinuities. Figure 4.14 contains the calculated flows and we see that the $L^2 - L^2$ optical flow model better reconstructs the smooth areas whereas the $TV - L^1$ model is able to better find the discontinuities in the flow.

Rubber Whale:
The Rubber Whale dataset highly differs from the Dimetrodon dataset. Here we have a set of moving objects equipped with some rotation compared to a smooth flow in the Dimetrodon dataset. Additionally the motion occurs towards different directions and

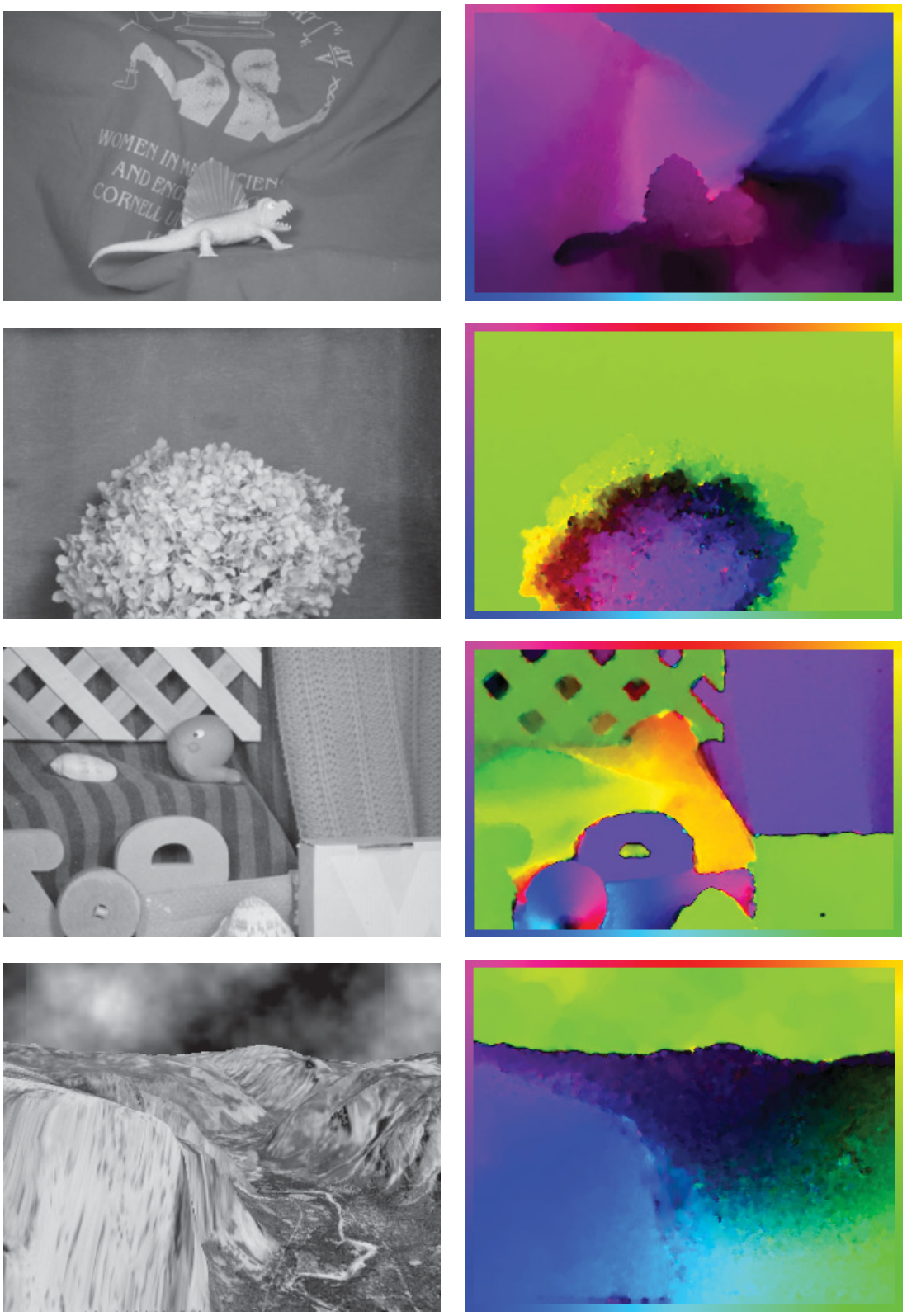


Figure 4.13.: Evaluated real data scenes (from [44]); left: image, right: ground truth flow.
First row: *Dimetrodon*, second row: *Hydrangea*, third row: *Rubber Whale*, fourth row: *Yosemite*

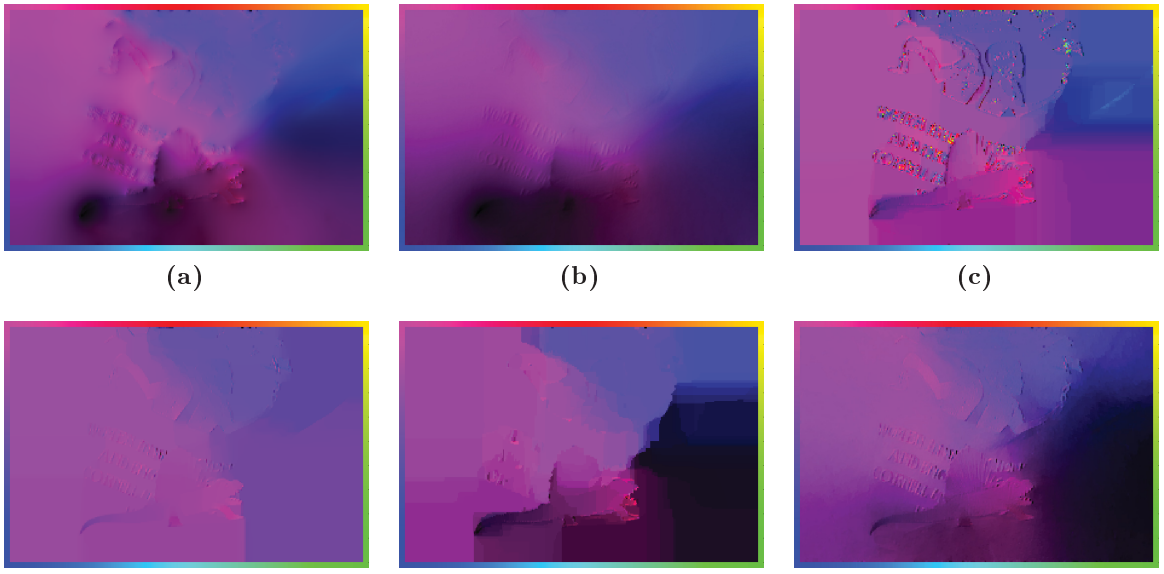


Figure 4.14.: Overview of the best results (in terms of the AE) our algorithms were able to generate for the Dimetrodon example. (a) L^2-L^2 optical flow, (b) L^2-L^2 mass preservation, (c) L^2-TV optical flow, (d) L^2-TV mass preservation, (e) L^1-TV optical flow, (f) L^1-TV mass preservation

Algorithm	AEE	AE	IE	NIE	ILLE
L^2-L^2 optical flow	0.075	0.04	0.0027	2.2e-05	**0.291**
L^2-L^2 mass preservation	0.105	0.056	0.0034	2.8e-05	0.293
L^2-TV optical flow	**0.054**	**0.029**	0.0028	2.1e-05	0.292
L^2-TV mass preservation	0.111	0.059	0.0037	3.2e-05	0.293
L^1-TV optical flow	0.062	0.033	**0.0025**	**2e-05**	0.292
L^1-TV mass preservation	0.095	0.05	0.0033	3e-05	0.293

Table 4.5.: Rubber Whale dataset. Table shows minimal possible errors that could be found.

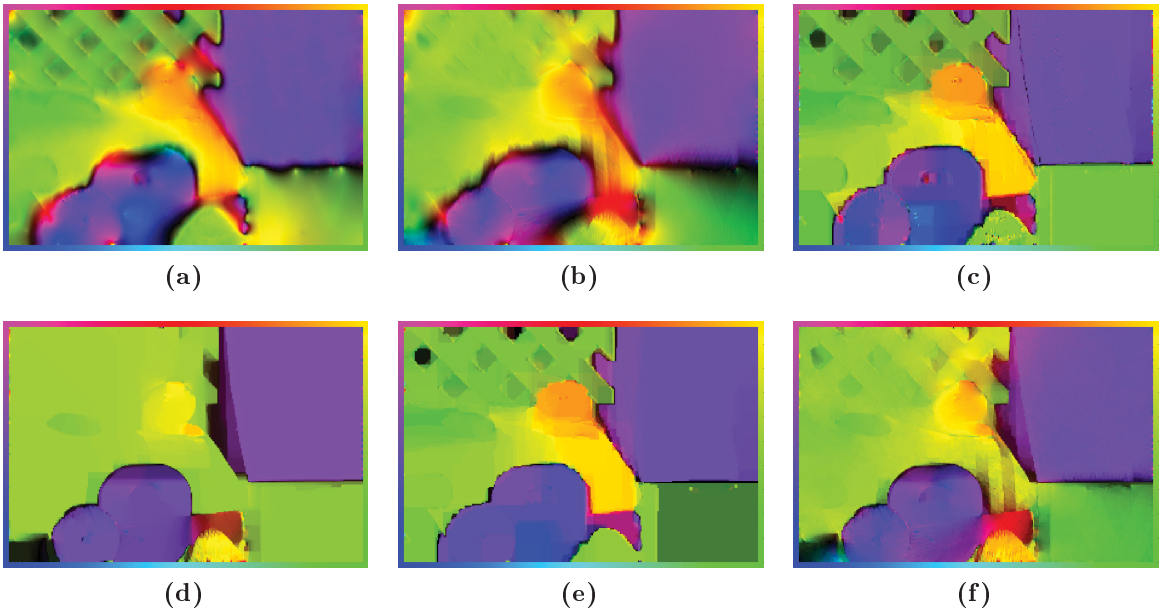


Figure 4.15.: Overview of the best results (in terms of the AE) our algorithms were able to generate for the Rubber Whale example. (a) $L^2 - L^2$ optical flow, (b) $L^2 - L^2$ mass preservation, (c) $L^2 - TV$ optical flow, (d) $L^2 - TV$ mass preservation, (e) $L^1 - TV$ optical flow, (f) $L^1 - TV$ mass preservation

different magnitudes. The evaluation results can be found in Table 4.5. The $TV - L^1$ optical flow can be seen as optimal because it minimizes AEE, IE, NIE and performs only slightly worse than the $TV - L^2$ model in terms of the AE. The color-coded best results can be found in Figure 4.15. The Total Variation regularization is advantageous for this class of motion due to many discontinuities in the flow field which cannot be recovered by the L^2-regularized models.

Hydrangea and Yosemite:

At first glance both sequences look different but from the perspective of motion estimation they share many aspects. We find many small structures which move independently but similar to the neighboring objects. The ground truth contains slight discontinuities which are not as large as in the Rubber Whale example. We listed the results in Tables 4.6 and 4.7. Again, the $TV - L^1$ optical flow model outperforms all other models and is able to generate the smallest AEE, AE, IE and NIE. From Figure 4.16 we see that the discontinuities are recovered very well and the static background-movement is also detected.

Algorithm	AEE	AE	IE	NIE	ILLE
$L^2 - L^2$ optical flow	0.092	0.054	0.0033	**2.4e-05**	2.175
$L^2 - L^2$ mass preservation	0.117	0.066	0.0039	2.9e-05	**2.173**
$L^2 - TV$ optical flow	0.067	0.037	0.0036	2.5e-05	2.174
$L^2 - TV$ mass preservation	0.073	0.04	0.0036	3.1e-05	2.177
$L^1 - TV$ optical flow	**0.047**	**0.027**	**0.0029**	2.4e-05	2.178
$L^1 - TV$ mass preservation	0.083	0.046	0.0042	3.5e-05	2.175

Table 4.6.: Hydrangea dataset. Table shows minimal possible errors that could be found.

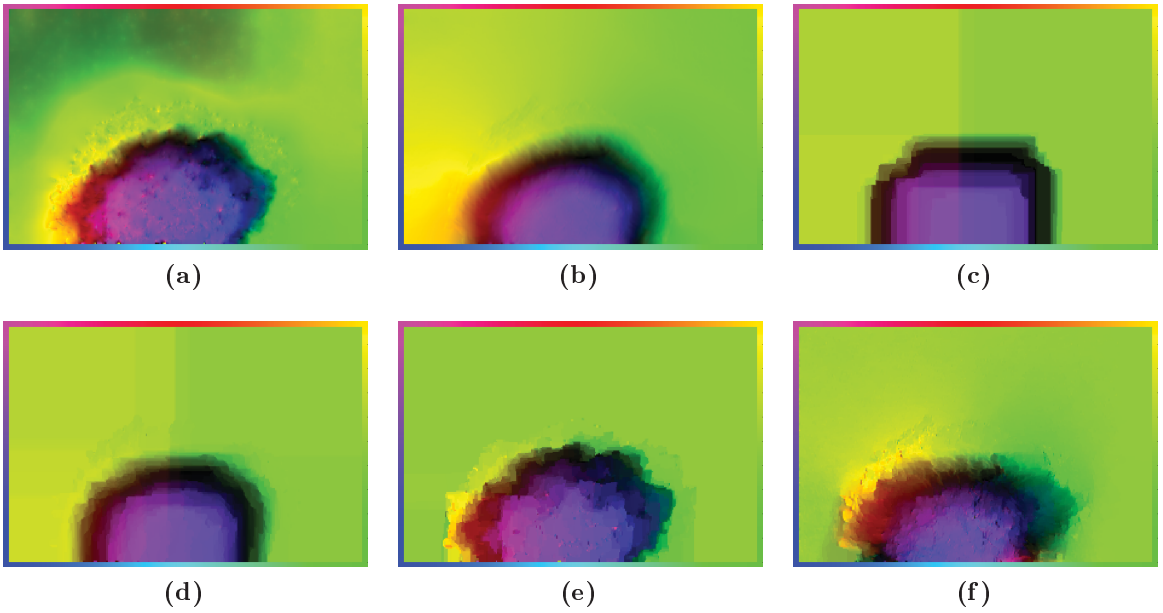


Figure 4.16.: Overview of the best results (in terms of the AE) our algorithms were able to generate for the Hydrangea example. (a) $L^2 - L^2$ optical flow, (b) $L^2 - L^2$ mass preservation, (c) $L^2 - TV$ optical flow, (d) $L^2 - TV$ mass preservation, (e) $L^1 - TV$ optical flow, (f) $L^1 - TV$ mass preservation

Algorithm	AEE	AE	IE	NIE	ILLE
$L^2 - L^2$ optical flow	0.075	0.04	0.0027	2.2e-05	**0.291**
$L^2 - L^2$ mass preservation	0.105	0.056	0.0034	2.8e-05	0.293
$L^2 - TV$ optical flow	**0.054**	**0.029**	0.0028	2.1e-05	0.292
$L^2 - TV$ mass preservation	0.111	0.059	0.0037	3.2e-05	0.293
$L^1 - TV$ optical flow	0.062	0.033	**0.0025**	**2e-05**	0.292
$L^1 - TV$ mass preservation	0.095	0.05	0.0033	3e-05	0.293

Table 4.7.: Yosemite Clouds dataset. Table shows minimal possible errors that could be found.

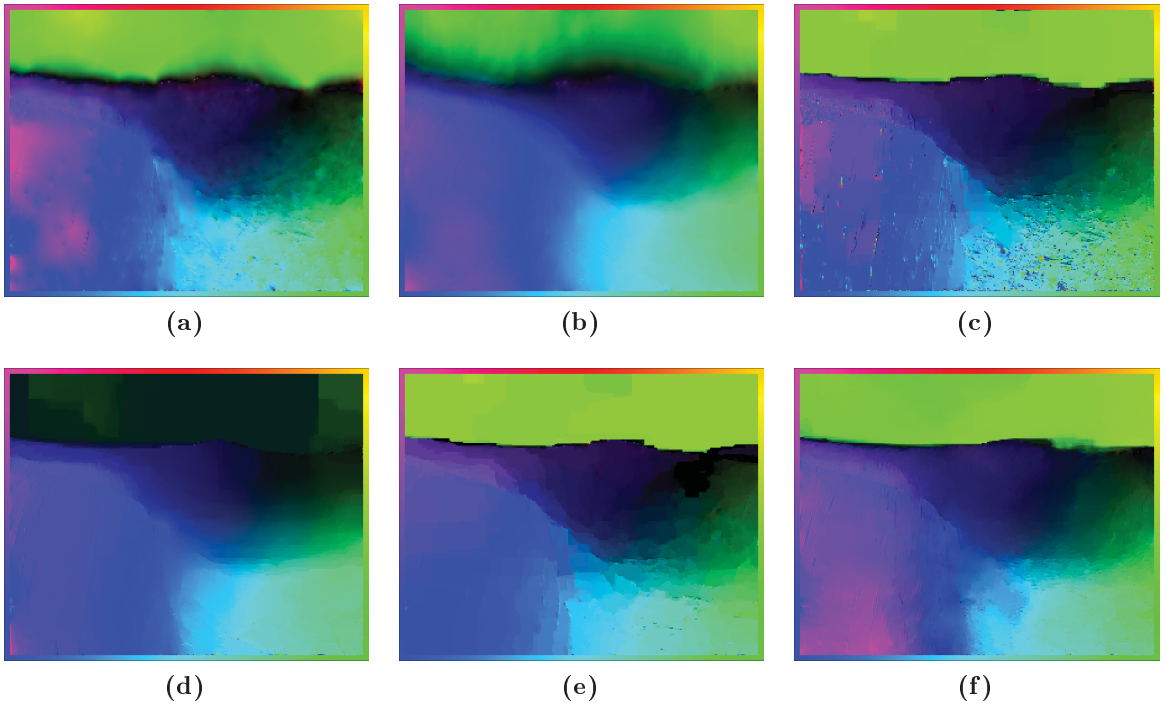


Figure 4.17.: Overview of the best results (in terms of the AE) our algorithms were able to generate for the Yosemite Clouds example. (a) L^2-L^2 optical flow, (b) L^2-L^2 mass preservation, (c) L^2-TV optical flow, (d) L^2-TV mass preservation, (e) L^1-TV optical flow, (f) L^1-TV mass preservation

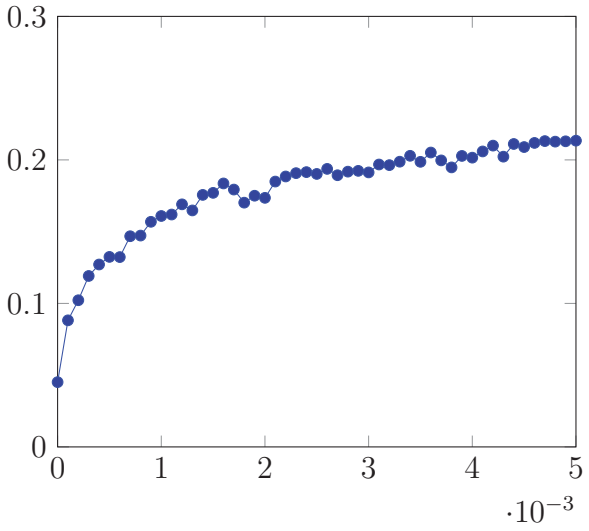


Figure 4.18.: Plot of the variance σ of noise (x-axis) versus ae of the reconstruction (y-axis).

4.6.6. Influence of Noise

In practical applications (e.g. microscopy) the recorded images often come with a lack of image quality which is caused by low acquisition times. This leads to another very interesting aspect in motion estimation - how does the noise-level on the image data correspond to the quality of the estimated velocity field $\boldsymbol{v}$. To answer this question we created a series of noisy *Dimetrodon* images, where Gaussian noise with different variance σ was added to the images (for $u : \Omega \to [0, 1]$). Then we estimated the motion using the $L^1 - TV$ optical flow algorithm with a static regularization parameter. In Figure 4.18 we plotted the variance of noise on the x-axis versus the **ae** of the reconstruction on the y-axis. We that already small levels of noise have massive influence to the motion estimation process. In Figure 4.19 we can see a the Dimetrodon image with a relative noise-level of $\sigma = 0.0002$ which can be described as *slightly noisy* but increases the absolute error in the motion field from ≈ 0.045 to ≈ 0.1. This high sensitivity towards noise can be explained by the optical flow constraint (see Section 4.2) which has been deduced from the brightness-constancy assumption. This assumption is violated by the Gaussian noise because very many local intensity variations are introduced and the constraint is broken.

Consequently, before estimating the motion a preprocessing step has to be applied to remove the noise. A more advanced technique is presented in Chapter 5 where we present variational models that are able to simultaneously denoise images and estimate the underlying motion, while both tasks improve each other.

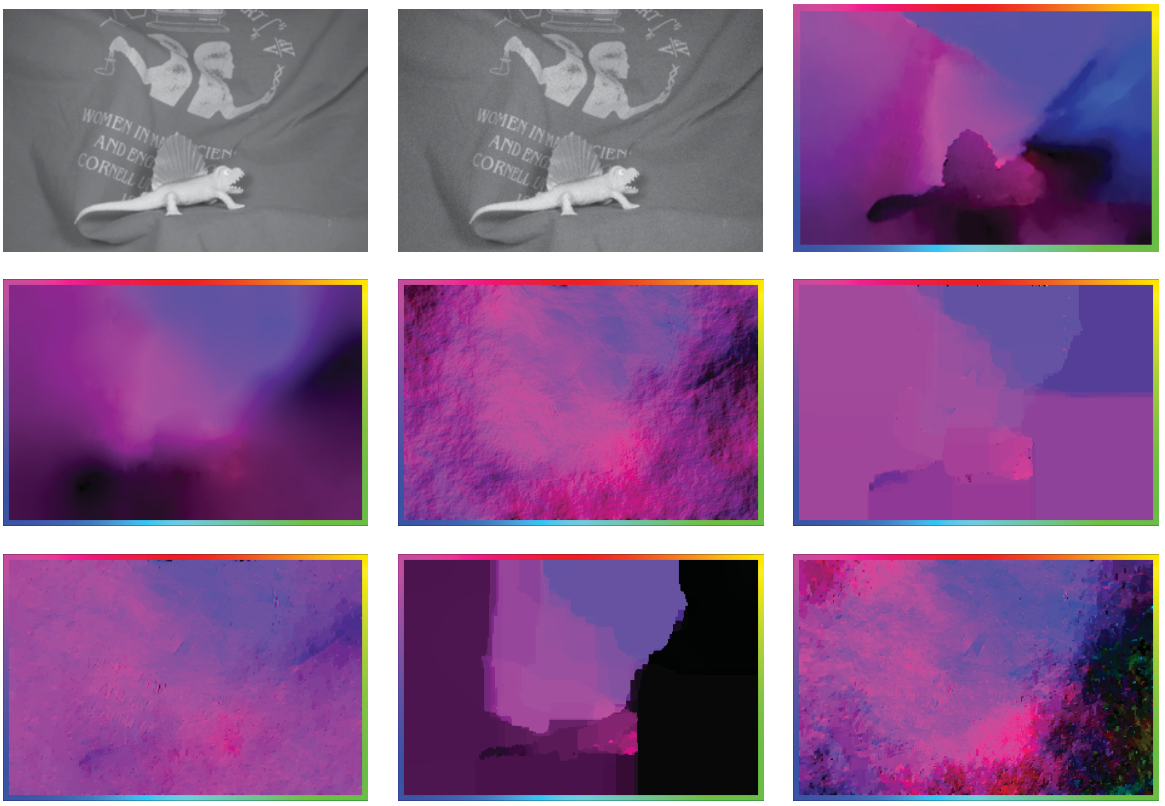


Figure 4.19.: Dimetrodon scene (from [44]) used to evaluate the influence of noise to the discussed motion estimation algorithms. Top row: Image, noisy counterpart, ground truth field; Middle row: L^2-L^2 optical flow,L^2-L^2 mass preservation, L^2-TV optical flow; Bottom row: L^2-TV mass preservation, L^1-TV optical flow, L^1-TV mass preservation.

To evaluate the influence of noise to the different algorithms we added a moderate level of noise (additive Gaussian with variance $\sigma = 0.0002$) to the Dimetrodon sequence. Figure 4.19 shows the ground truth image and the noisy counterpart.

4.6.7. Runtime

Finally we want to compare runtimes of the presented algorithms. For this sake we take the modified Dimetrodon example (see Figure 4.13) and scale it down up to a factor of one percent of the original size. All proposed algorithms are then applied to the intermediate images, where the regularization parameter is fixed and the tolerance chosen as $1e-4$. Figure 4.20 shows the image size in number of pixels on the x-axis versus the runtime in seconds on the y-axis.

The first observation we want to mention here is that the mass preservation models are always slower than the respective optical flow models. This can be explained by

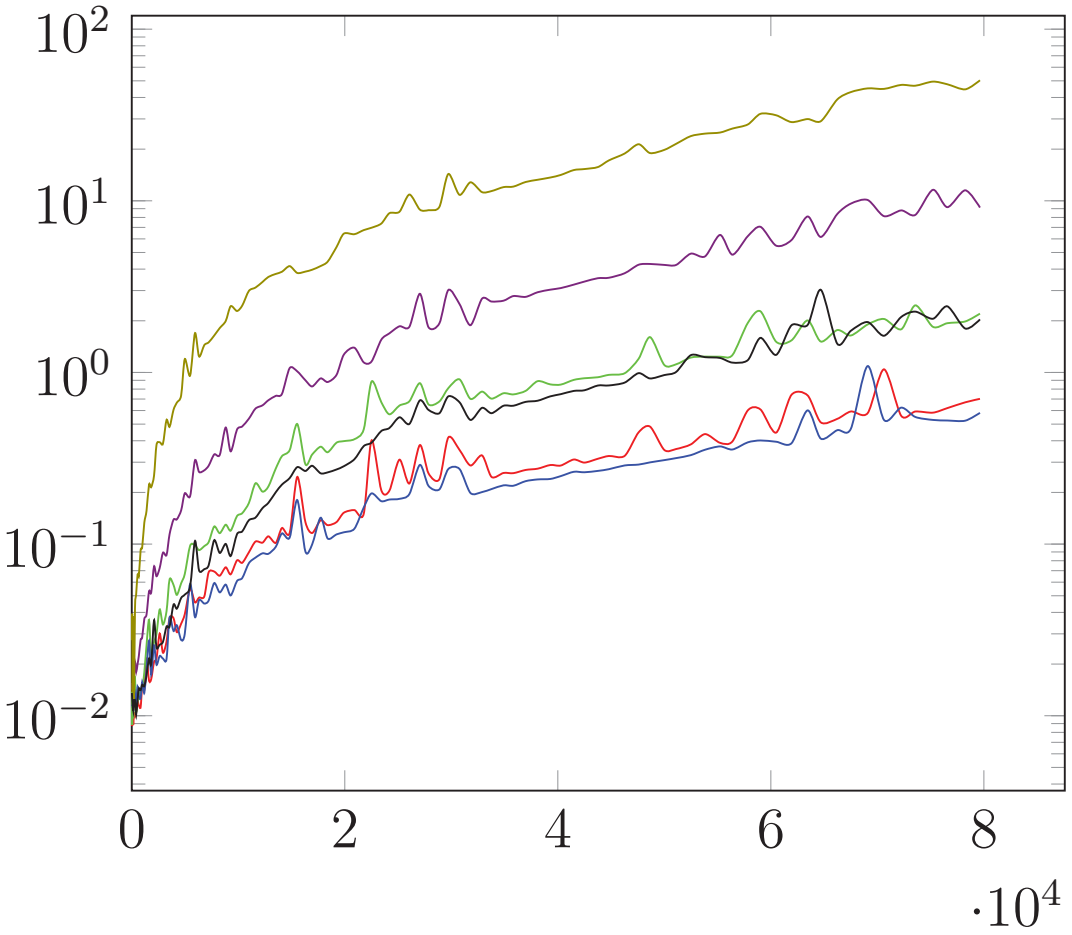


Figure 4.20.: Plot of the runtime (x-axis) versus the problem size (y-axis). $L^2 - L^2$ optical flow (red), $L^2 - L^2$ mass preservation (light green), $L^2 - TV$ optical flow (blue), $L^2 - TV$ mass preservation (light black, $L^1 - TV$ optical flow (violet), $L^1 - TV$ mass preservation (olive)

Algorithm	AEE	AE	IE	NIE	ILLE
$L^2 - L^2$ optical flow	0.107	0.064	**0.0126**	**8e-05**	1.86
$L^2 - L^2$ mass preservation	**0.097**	**0.058**	0.0156	8.2e-05	1.858
$L^2 - TV$ optical flow	0.122	0.072	0.0146	8.1e-05	1.86
$L^2 - TV$ mass preservation	0.132	0.079	0.0158	8.3e-05	1.861
$L^1 - TV$ optical flow	0.154	0.091	0.0149	8.3e-05	**1.857**
$L^1 - TV$ mass preservation	0.205	0.123	0.0161	8.5e-05	1.859

Algorithm	AEE	AE	IE	NIE	ILLE
$L^2 - L^2$ optical flow	**0.07**	**0.042**	**0.0025**	**2e-05**	**0.177**
$L^2 - L^2$ mass preservation	0.092	0.054	0.0028	2.3e-05	0.179
$L^2 - TV$ optical flow	0.1	0.062	0.0028	2.1e-05	0.179
$L^2 - TV$ mass preservation	0.137	0.081	0.0034	2.9e-05	0.182
$L^1 - TV$ optical flow	0.076	0.045	0.0025	2.1e-05	0.177
$L^1 - TV$ mass preservation	0.105	0.063	0.0031	2.7e-05	0.179

Table 4.8.: Dimetrodon dataset with (top) and without (bottom) noise. Table shows minimal possible errors that could be found

the fact that the mass preservation term contains an additional operator applied to $\boldsymbol{v}$, which increases the runtime. Next, the $L^2 - L^2$ models and the $L^2 - TV$ models differ only slightly in their execution times, where the $L^2 - TV$ model has the overall best runtime. Moreover, both $L^1 - TV$ models perform significantly slower where the $L^1 - TV$ mass preservation model is by far slowest. Finally, between fastest model ($TV - L^2$ optical flow) and slowest model ($TV - L^2$ optical flow) we denote a difference in runtime of factor 100.

5 Combined Image Reconstruction and Flow Estimation

In this chapter we will deduce a total of four models for simultaneous image reconstruction and motion estimation. The introduced models differ from each other in the choice of data term (optical flow and mass preservation) and regularization term for the velocity field (L^2 and total variation). Starting with a short introduction and derivation of the model we dedicate some time to a detailed proof of existence of minimizers. Here, an application of the Aubin-Lions Lemma (see 2.1.16) is required to obtain time-regularity of the solution. An efficient numerical realization based on the primal-dual framework from Section 2.3.1 is presented and we explain the discretization of the involved operators. All models are evaluated in terms of image reconstruction, motion estimation and inpainting. Finally, we compare runtimes and check for parameter dependencies.

5.1. General Model

Inspired by well-known models for image inpainting, image denoising [60] and optical flow [43, 77, 74] we present a model that couples the tasks of image reconstruction and flow estimation.

In the following work we consider noisy, time-dependent image data u on a space-time domain $\Omega \times [0,T] \subset \mathbb{R}^d \times \mathbb{R}^+$. This data is captured by a recording device (e.g.

microscope) and we arrive at measured data f, which can be modeled as

$$f = Ku + \delta_\sigma. \tag{5.1}$$

Here δ_σ represents a Gaussian-distributed noise with variance σ, which usually occurs due to bad illumination or low acquisition times. The (usually linear) operator K represents the forward operator modeling the relation of the image sequence u on the measured data f. The operator K allows us to model more complex situations. In our general application K represents an inpainting-operator, that is $K = P_\Sigma$ which projects the image u onto the known domain Ω, to model the problem of undersampled image data. For example in time-dependent 3-dimensional microscopy data we often have a good X/Y resolution, but only a few slices in Z direction. Moreover, K can be used to model a blurring effect, which often occurs in high-resolution microscopy.
Simultaneously to the reconstruction of the image sequence u we seek for the velocity field $\boldsymbol{v} : \Omega \times [0,T] \to \mathbb{R}^d$ describing the motion in the underlying image data.
Our motivation to recover both unknowns simultaneously is justified by their temporal coupling through equations that arise directly from a physical context, namely from the optical flow equation (see Section 4.2) and the continuity equation (see Section 4.3). On the one hand this additional information improves the image reconstruction, but on the other hand the improved image quality enhances the motion estimation quality (see Section 4.6.6). Consequently, in a joint model both processes endorse each other.
The image reconstruction is in general ill-posed. For some applications the inverse operator K^{-1} does not even exist. For others we can apply K^{-1} to both sides of Equation (5.1) and get

$$K^{-1}f = u + K^{-1}\delta_\sigma \leftrightarrow u = K^{-1}f - K^{-1}\delta_\sigma.$$

Unfortunately for the noise we have $K^{-1}\delta_\sigma \ggg K^{-1}f$ in the norm of the space of the unknown u and cannot recover u due to the dominating noise. For the motion estimation task we have to consider problems like the aperture problem (see 4.1.1) and underdetermination (see 4.1.2). Consequently, we complete our model with regularizers for both terms and weighting parameters α and β. This leads directly to the following model for joint motion estimation and image reconstruction

Model 5.1.1. Joint Reconstruction and Motion Estimation

$$\arg\min_{u,\boldsymbol{v}} \int_0^T \frac{1}{2} \|Ku - f\|_2^2 \,\mathrm{d}t + \alpha \int_0^T R(u(\cdot,t)) \,\mathrm{d}t + \beta \int_0^T S(\boldsymbol{v}(\cdot,t)) \,\mathrm{d}t \tag{5.2}$$

$$\text{s.t.} \quad C(u, \boldsymbol{v}) = 0 \quad \text{in } \mathcal{D}'([0,T] \times \Omega).$$

The first term in this functional acts as a data fidelity between the measured data f and the objective function u. Here we assume the noise to be additive Gaussian-distributed, which naturally leads to the squared L^2-norm (see for example [15]) for the distance $Ku - f$ integrated over the time domain:

$$\int_0^T \frac{1}{2} \|Ku - f\|_2^2 \, \mathrm{d}t.$$

We may think of data fidelities for other noise models like the L^1-distance for salt and pepper noise

$$\int_0^T \|Ku - f\|_1 \, \mathrm{d}t,$$

or the Kullback-Leibler divergence for Poisson noise

$$\int_0^T \int_\Omega Ku - f + f \log \frac{f}{Ku} \, \mathrm{d}x \, \mathrm{d}t,$$

but this is beyond the scope of this thesis. The underlying Bayesian modeling for these data fidelities can be found in [20].
The second term in Equation (5.2) is a regularization term for the underlying image sequence $u(x,t)$. We mention that R only acts on single frames $u(\cdot,t)$ and consequently, we do not have any time-dependence. If we just want to reconstruct smooth images, it would be obvious to use quadratic L^2-regularization on the gradient and to set

$$R(u(\cdot,t)) := \left\|\nabla u(\cdot,t)\right\|_{L^2}^2 . \tag{5.3}$$

From the context of image inpainting, this choice leads directly to solving a Poisson equation inside the unknown region $\Sigma \subset \Omega \subset \mathbb{R}^d$ with boundary conditions coming from the known areas.
A more natural regularization in the context of images is the total variation, which preserves edges and favors constant regions. We refer to Section 2.1.3 for more details about TV. In this case we set

$$R(u(\cdot,t)) := \left|u(\cdot,t)\right|_{BV} . \tag{5.4}$$

From the image inpainting perspective, a TV-based regularizer will fill up the unknown

region Σ with areas of constant color and might also connect edges.
Regularizers for the velocity field $\boldsymbol{v}$ can be motivated very similar to those for images. The regularizer in Model 5.2 acts independently on every timestep and we have no time-correspondence. If we expect smooth velocity fields, then a quadratic L^2-regularization should be the method of choice and we define

$$S(\boldsymbol{v}(\cdot,t)) := \left\|\nabla \boldsymbol{v}(\cdot,t)\right\|_{L^2}^2 . \tag{5.5}$$

This regularizer has been brought to the field of motion estimation by Horn and Schunck [43] (see Section 4.5.2).
For the motion field, a TV-based regularizer should in general fit better to the recorded data. From the perspective of motion estimation a moving object is a region of constant movement similar to the size of the object. We define analogously

$$S(\boldsymbol{v}(\cdot,t)) := \left|\boldsymbol{v}(\cdot,t)\right|_{BV} . \tag{5.6}$$

The final ingredient to define is the constraint $C(u,\boldsymbol{v})$. We do not require the constraint to be fulfilled strongly, but in a weak sense, and consequently choose the space of distributions $\mathcal{D}'([0,T]\times\Omega)$. The constraint should connect image data u and velocity field $\boldsymbol{v}$ in a reasonable way. Since motion is always time-dependent, the constraint will, in contrary to $R(u)$ and $R'(u)$, always incorporate the time t. The **continuity equation**, which has been deduced in Section 4.3, is a connection that arises from the natural assumption that mass keeps constant in every time $t \in [0,T]$. In this case we define

$$C(u,\boldsymbol{v}) := \partial_t u + \nabla\cdot(u\boldsymbol{v}). \tag{5.7}$$

A different approach is to assume that the images contain objects of **constant brightness**, which are only allowed to change their position within the image domain Ω. Some simple calculations lead to the so-called **optical flow constraint** (see Section 4.2 for details) and we define

$$C(u,\boldsymbol{v}) := \partial_t u + \nabla u\cdot\boldsymbol{v}. \tag{5.8}$$

5.2. Simultaneous $TV - L^2$ Image Reconstruction and Motion Estimation

5.2.1. Definition and Motivation

The first concrete realization of model (5.2) to be discussed here is equipped with a TV-regularizer for the image data u (see Equation 5.4) and a quadratic L^2-regularizer on the gradients of the motion field $\boldsymbol{v}$ (see Equation 5.5). Let us begin with the mass preserving variant:

Model 5.2.1. Mass-preserving $TV - L^2$ model

$$\begin{aligned} \arg\min_{u,\boldsymbol{v}} & \int_0^T \frac{1}{2} \|Ku - f\|_2^2 \,\mathrm{d}t + \alpha \int_0^T \left|u(\cdot,t)\right|_{BV} \mathrm{d}t + \frac{\beta}{2} \int_0^T \left\|\nabla \boldsymbol{v}(\cdot,t)\right\|_2^2 \mathrm{d}t \\ \text{s.t.} \quad & \partial_t u + \nabla \cdot (u\boldsymbol{v}) = 0 \quad \text{in } \mathcal{D}'([0,T]\times\Omega), \end{aligned}$$

and secondly proceed with the optical flow variant:

Model 5.2.2. Optical flow $TV - L^2$ model

$$\begin{aligned} \arg\min_{u,\boldsymbol{v}} & \int_0^T \frac{1}{2} \|Ku - f\|_2^2 \,\mathrm{d}t + \alpha \int_0^T \left|u(\cdot,t)\right|_{BV} \mathrm{d}t + \frac{\beta}{2} \int_0^T \left\|\nabla \boldsymbol{v}(\cdot,t)\right\|_2^2 \mathrm{d}t \\ \text{s.t.} \quad & \partial_t u + \nabla u \cdot \boldsymbol{v} = 0 \quad \text{in } \mathcal{D}'([0,T]\times\Omega). \end{aligned}$$

We mention that the latter model is basically an extension of the classical motion estimation model of Horn and Schunck [43].
Both models reconstruct the image sequence u and an underlying motion field $\boldsymbol{v}$. Furthermore, an additive Gaussian noise model for the data f is assumed. From the TV-based image regularizer we expect a cartoon-like image reconstruction in every timestep. Besides that, the quadratic L^2-regularization should give us smooth velocity fields. Since in this thesis we concentrate on applications for real data (e.g. cell movement), we expect only finite speeds. This gives us a useful natural bound

$$\|\boldsymbol{v}\|_\infty \leq c_v < \infty \quad \text{a.e. in } \Omega \times [0,T]\,. \tag{5.9}$$

Besides this we need a bound on the divergence of $\boldsymbol{v}$ in L^p for the existence of a minimizer for the optical flow model. From the physical point of view the divergence measures the magnitude of the source or sink of $\boldsymbol{v}$ at a given point $x \in [0,T]$. Consequently, having $\nabla \cdot \boldsymbol{v} \in L^p$ means an overall boundedness of sources and sinks, which is however not necessarily pointwise. Moreover, for the flow $\boldsymbol{v}$ the divergence is a measure for compressibility. We speak of an incompressible flow if $\nabla \cdot \boldsymbol{v} = 0$, so bounding the divergence means bounding the compressibility of $\boldsymbol{v}$.

5.2.2. Existence and Uniqueness

For this section we consider a more general energy, which now contains powers of both regularizers:

$$J(u,\boldsymbol{v}) = \int_0^T \frac{1}{2}\left\|Ku - f\right\|_2^2 \mathrm{d}t + \alpha \int_0^T \left|u(\cdot,t)\right|_{BV}^p \mathrm{d}t + \beta \int_0^T \left(\left\|\nabla \boldsymbol{v}(\cdot,t)\right\|_2^2\right)^q \mathrm{d}t, \quad (5.10)$$

for $1 < p, q \in \mathbb{R}$.

Existence

In this context we will apply the fundamental theorem of optimization (see 2.2.17 for details), which requires:

1. Compactness of sublevel sets (coercivity) in the topology τ (see 2.2.15)
2. Lower semicontinuity with respect to τ (see 2.2.11)
3. Convergence of the constraint, $D(u_k, \boldsymbol{v_k}) \to D(u,\boldsymbol{v})$ in a distributional sense

Since compactness of sublevel sets and lower semicontinuity are independent of the constraint $D(u,\boldsymbol{v})$, it can be shown for both models at the same time:

Lemma 5.2.3. Compactness of sublevel sets of the general TV-L^2 model
Let

$$u \in L^p(0,T;BV(\Omega)), \quad \boldsymbol{v} \in L^q(0,T;W^{1,2}(\Omega)), \quad J(u,\boldsymbol{v}) \leq \alpha$$

and furthermore

$$\left\|\boldsymbol{v}\right\|_\infty \leq c_v < \infty \quad \text{a.e. in } \Omega \times [0,T] \text{ and } K\mathbf{1}_t \neq 0 \quad \forall t \in [0,T]\,.$$

Now, for $(u, \boldsymbol{v})$ in the set

$$\mathcal{S}(c) = \left\{(u, \boldsymbol{v}) \in L^{\hat{p}}(0, T; BV(\Omega)) \times L^q(0, T; W^{1,2}(\Omega)) : J(u, \boldsymbol{v}) \leq c\right\}$$

we have

$$\|u\|_{L^{\hat{p}}(0,T;BV(\Omega))} \leq c, \quad \|v\|_{L^q(0,T;W^{1,2}(\Omega))} \leq c \quad \text{with } \hat{p} = \min\{p, 2\},$$

and consequently, $\mathcal{S}(c)$ is not empty and compact in the weak-* topology of $L^{\hat{p}}(0, T; BV(\Omega)) \times L^q(0, T; W^{1,2}(\Omega))$.

Proof. We start with the bound for u and have to prove that for arbitrary $u \in L^p(0, T; BV(\Omega))$ with $J(u, \cdot) \leq \alpha$ we have

$$\begin{aligned}\|u\|_{L^p(0,T;BV(\Omega))} \leq \|u\|^p_{L^p(0,T;BV(\Omega))} &= \int_0^T \|u\|^p_{BV(\Omega)} \,\mathrm{d}t \\ &\leq \int_0^T \|u\|^p_{L^1(\Omega)} \,\mathrm{d}t + \int_0^T |u|^p_{BV(\Omega)} \,\mathrm{d}t \leq c.\end{aligned}$$

To deduce this bound we need to estimate each of the two terms in the last line of the inequality. Later on we will see that this bound cannot be shown for arbitrary $1 < p \leq 2$.

Since all three terms in the functional (5.10) are positive, from $J(u, \boldsymbol{v}) \leq \alpha$ we directly get a bound on each of the three parts. It follows that

$$\|Ku - f\|_{L^2(0,T;L^2(\Omega))} \leq \alpha,$$

which naturally implies

$$\big(Ku(\cdot, t) - f(\cdot, t)\big) \in L^2(\Omega) \quad \text{ a.e. in } [0, T].$$

Consequently, $\big\|Ku(\cdot, t) - f(\cdot, t)\big\|_{L^2(\Omega)}$ is bounded almost everywhere in $t \in [0, T]$ and we define

$$c_K(t) := \big\|Ku(\cdot, t) - f(\cdot, t)\big\|_{L^2(\Omega)}.$$

We want to emphasize here that $c_K(t)$ gives a constant for every time step $t \in [0, T]$, but the integral $\int_0^T c_K^p \,\mathrm{d}t$ is only bounded for $1 < p \leq 2$ due to the L^2-regularity in time.

Besides this constant, for the TV-part, we directly also obtain

$$\int_0^T |u|_{BV(\Omega)}^p \,\mathrm{d}t = \int_0^T TV(u)^p \,\mathrm{d}t \leq \alpha.$$

The crucial point in this proof is to find a bound for $\|u\|_{L^p(0,T;L^1(\Omega))}$. Let $t \in [0,T]$ be an arbitrary time step. First, we deduce a bound for this single time step $\left\|u(\cdot,t)\right\|_{L^1(\Omega)}$ and start with a decomposition for u:

$$\bar{u} = \frac{1}{|\Omega|}\int_\Omega u(x,t)\,\mathrm{d}x, \quad u_0 = u(\cdot,t) - \bar{u} \leftrightarrow u(\cdot,t) = u_0 + \bar{u}.$$

From this definition it follows directly that u_0 fulfills

$$\int_\Omega u_0 \,\mathrm{d}x = 0 \quad \text{(mean value zero)},$$

and $TV(u(\cdot,t)) = TV(u_0) \leq \alpha$. Using the Poincaré-Wirtinger inequality (see [50] and Lemma A.1.10) we obtain an L^2-bound for u_0:

$$\|u_0\|_{L^2(\Omega)} \leq c_1 \|\nabla u_0\|_{L^2(\Omega)} \leq c_1 c_2 \|\nabla u_0\|_{L^1(\Omega)} = c_3 TV(u_0) \leq c_3\alpha,$$

where c_1, c_2 and c_3 are positive constants. Moreover, we need a bound for $\|K\bar{u}\|_{L^2(\Omega)}$, which we get by calculating

$$\begin{aligned}
\|K\bar{u}\|_{L^2}^2 - 2\|K\bar{u}\|_{L^2}\left(\|K\|\,\|u_0\|_{L^2} + \|f\|_{L^2}\right) &= \|K\bar{u}\|_{L^2}\left(\|K\bar{u}\|_{L^2} - 2\left(\|K\|\,\|u_0\|_{L^2} + \|f\|_{L^2}\right)\right) \\
&\leq \|K\bar{u}\|_{L^2}\left(\|K\bar{u}\|_{L^2} - 2\|Ku_0 - f\|_{L^2}\right) \\
&= \|K\bar{u}\|_{L^2}^2 - 2\|Ku_0 - f\|_{L^2}\|K\bar{u}\|_{L^2} \\
&\leq \|Ku_0 - f\|_{L^2}^2 + \|K\bar{u}\|_{L^2}^2 - 2\|Ku_0 - f\|_{L^2}\|K\bar{u}\|_{L^2} \\
&= \left(\|Ku_0 - f\|_{L^2} - \|K\bar{u}\|_{L^2}\right)^2 \leq \|Ku_0 + K\bar{u} - f\|_{L^2}^2 \\
&= \left\|Ku(\cdot,t) - f(\cdot,t)\right\|_{L^2}^2 \leq c_K(t)^2.
\end{aligned}$$

Defining $x := \|K\bar{u}\|_{L^2(\Omega)}, a := \|K\|\,\|u_0\|_{L^2(\Omega)} + \|f\|_{L^2(\Omega)}$, we get the simple quadratic inequality

$$x^2 - 2xa \leq c_K(t)^2 \tag{5.11}$$

and furthermore know

$$0 \leq a \leq \|K\|\,c_3\alpha + \|f\|_{L^2(\Omega)} =: c_4.$$

Plugging this into the quadratic inequality (5.11) yields the solution

$$0 \leq x \leq c_4 + \sqrt{\alpha + c_4^2} \leq c_4 + c_7(c_K(t) + c_4).$$

Assuming $K\mathbf{1}_t \neq 0$, this leads to an estimate for the operator

$$\begin{aligned} \|K\bar{u}\|_{L^2(\Omega)} &= \left| \frac{1}{|\Omega|} \int_\Omega u \, \mathrm{d}x \right| \|K\mathbf{1}\|_{L^2(\Omega)} \leq c_4 + \sqrt{\alpha + c_4^2} \\ \Leftrightarrow \left| \frac{1}{|\Omega|} \int_\Omega u \, \mathrm{d}x \right| &\leq \frac{c_4 + c_7(c_K(t) + c_4)}{\|K\mathbf{1}\|_{L^2(\Omega)}} =: c_5(t). \end{aligned}$$

We are now able to bound the L^1-norm of a single timestep $t \in [0, T]$ by a constant $c_u(t)$ as follows:

$$\begin{aligned} 0 \leq \big\|u(\cdot,t)\big\|_{L^1(\Omega)} &\leq c_6 \big\|u(\cdot,t)\big\|_{L^2(\Omega)} = c_6 \left\| u_0 + \frac{1}{|\Omega|} \int_\Omega u(x,t) \, \mathrm{d}x \right\|_{L^2(\Omega)} \\ &\leq c_6 \left(\|u_0\|_{L^2(\Omega)} + \left| \frac{1}{|\Omega|} \int_\Omega u(x,t) \, \mathrm{d}x \right| \right) \\ &\leq c_6 \big(c_3\alpha + c_5(t) \big) =: c_u(t). \end{aligned}$$

This leads to the crucial point, since we are integrating over all these constants $c_u(t)$ and the integral is only bounded for $1 < p \leq 2$. Consequently, we define $\hat{p} := \min\{p, 2\}$ and get

$$\int_0^T \big\|u(\cdot,t)\big\|_{L^1(\Omega)}^{\hat{p}} \, \mathrm{d}t \leq \int_0^T c_u(t)^{\hat{p}} \, \mathrm{d}t \leq c_M.$$

Combining both estimations we conclude with the required bound for arbitrary $u \in L^p(0, T; BV(\Omega))$:

$$\begin{aligned} \|u\|_{L^{\hat{p}}(0,T;BV(\Omega))} \leq \|u\|^{\hat{p}}_{L^{\hat{p}}(0,T;BV(\Omega))} &= \int_0^T \|u\|^{\hat{p}}_{BV(\Omega)} \, \mathrm{d}t \\ &\leq \int_0^T \|u\|^{\hat{p}}_{L^1(\Omega)} \, \mathrm{d}t + \int_0^T |u|^{\hat{p}}_{BV(\Omega)} \, \mathrm{d}t \leq c_M T + \alpha. \end{aligned}$$

A bound for $\boldsymbol{v}$ is easier to establish, since we have $\|\boldsymbol{v}\|_{L^\infty(\Omega)} \leq c_v < \infty$ (see Equation 5.9) almost everywhere. Similar to u, from $J(u, \boldsymbol{v}) \leq \alpha$ we obtain the a-priori bound

$$\int_0^T \Big(\big\|\nabla \boldsymbol{v}(\cdot,t)\big\|^2_{L^2(\Omega)} \Big)^q \, \mathrm{d}t \leq \alpha$$

for $\boldsymbol{v}$ from Equation (5.10). We calculate the bound for $\boldsymbol{v}$ directly:

$$\begin{aligned}\|\boldsymbol{v}\|_{L^q(0,T;W^{1,2}(\Omega))} &\leq \|\boldsymbol{v}\|^q_{L^q(0,T;W^{1,2}(\Omega))} = \int_0^T \left\|\nabla \boldsymbol{v}(\cdot,t)\right\|^q_{W^{1,2}(\Omega)} \mathrm{d}t \\ &\leq \int_0^T \left\|\boldsymbol{v}(\cdot,t)\right\|^q_{L^2(\Omega)} \mathrm{d}t + \int_0^T \left\|\nabla \boldsymbol{v}(\cdot,t)\right\|^{2q}_{L^2(\Omega)} \mathrm{d}t \\ &\leq |\Omega|^q \int_0^T \left\|\boldsymbol{v}(\cdot,t)\right\|^q_{L^\infty(\Omega)} \mathrm{d}t + \alpha \leq |\Omega|^q \, c_v^q T + \alpha.\end{aligned}$$

In Lemma 2.1.10 we have shown that $BV(\Omega)$ is the dual space of a Banach space $\mathcal{Y}$. The application of Theorem 2.1.14 yields

$$L^p(0,T;BV(\Omega)) = L^{p^*}(0,T;\mathcal{Y}(\Omega))^*, \qquad \text{with } \frac{1}{p} + \frac{1}{p^*} = 1,$$

and with the same argumentation we get

$$L^q(0,T;W^{1,2}(\Omega)) = L^{q^*}(0,T;W^{-1,2}(\Omega))^*, \qquad \text{with } \frac{1}{q} + \frac{1}{q^*} = 1.$$

Since both spaces are duals, an application of the Banach-Alaoglu Theorem 2.2.16 yields the compactness in the weak-* topology. □

Lemma 5.2.4. Lower semicontinuity of the general TV-L^2 model
The TV-L^2 energy (5.10) is weak-* lower semicontinuous.

Proof. It has been shown that norms and affine norms are always convex (see 2.2.9). Convexity holds for arbitrary exponentials larger than one, thus $\|Ku - f\|^2_{L^2(\Omega)}$ and $\|\nabla \boldsymbol{v}\|^{2q}_{L^2(\Omega)}$ are convex for $q \geq 1$ (see 2.2.9). Convex functionals on Banach spaces have been proven to be weakly lower semicontinuous (see Lemma 2.2.13). Due to the reflexivity of L^2 we directly obtain weak-* lower semicontinuity.
Furthermore, we have shown that TV is weak-* lower semicontinuous (see Lemma 2.2.14). This property holds for exponentials p of TV satisfying $p > 1$.
The lower semicontinuity also holds for sums of lower semicontinuous functionals (2.2.12), which finishes the proof. □

Finally, the most challenging point is to show the convergence of the constraint

$$D(u_k, \boldsymbol{v_k}) \xrightarrow{\tau} D(u, \boldsymbol{v}).$$

The major problem arises from the product $\nabla \cdot (u\boldsymbol{v})$ (continuity) resp. $\nabla u \cdot \boldsymbol{v}$ (optical flow) in the constraint. Since both factors are bounded (in L^p resp. BV space) we can only derive weak convergence

$$u_k \rightharpoonup u, \quad \boldsymbol{v}_k \rightharpoonup \boldsymbol{v},$$

but a product of weakly convergent sequences does not necessarily converge to the product of their individual limits:

$$\nabla \cdot (u_k \boldsymbol{v}_k) \nrightarrow \nabla \cdot (u\boldsymbol{v}), \quad \nabla u_k \cdot \boldsymbol{v}_k \nrightarrow \nabla u \cdot \boldsymbol{v}.$$

A counterexample can be found in [68]. To achieve convergence we need at least one of the factors to converge strongly, but this can not be deduced from boundedness directly. A way out gives the Aubin-Lions Lemma 2.1.16, which yields a compact embedding

$$L^p(0,T;\mathcal{X}) \subset\subset L^p(0,T;\mathcal{Y}),$$

and hence strong convergence in $\mathcal{Y}$, if u_k is bounded in $L^p(0,T;\mathcal{X})$ and $\partial_t u_k$ is bounded in $L^q(0,T;\mathcal{Z})$ for Banach spaces $\mathcal{X} \subset\subset \mathcal{Y} \hookrightarrow \mathcal{Z}$. Applied to our case we set $\mathcal{X} = BV(\Omega)$ and $\mathcal{Y} = L^r(\Omega)$. The first goal is to derive a bound for $\partial_t u_k$ in some Lebesgue space $L^r(0,T;\mathcal{Z})$, which is given by the following lemma:

Lemma 5.2.5. Bound for $\partial_t u$
Let $\Omega \subset \mathbb{R}^2, u \in L^p(0,T;BV(\Omega))$ and $\boldsymbol{v} \in L^q(0,T;W^{1,2}(\Omega))$. Let furthermore be

$$\|\boldsymbol{v}\|_\infty \leq c_v < \infty \quad \text{a.e. in } \Omega \times [0,T]\,.$$

1. **Mass Preservation:**
   For $u, \boldsymbol{v}$ solving the continuity equation

   $$\partial_t u + \nabla \cdot (u\boldsymbol{v}) = 0$$

   we have

   $$\partial_t u \in L^p(0,T;W^{-1,2}(\Omega)).$$

2. **Optical Flow:**
   Let furthermore $\nabla \cdot \boldsymbol{v} \in L^{p^* s}(0,T;L^{2k}(\Omega))$ with $\frac{1}{k} + \frac{1}{k^*} = 1, k^* > 1$, $\frac{1}{s} + \frac{1}{t} =$

$1, \frac{1}{p} + \frac{1}{p^*} = 1$ and let $u, \boldsymbol{v}$ solve the optical flow equation

$$\partial_t u + \nabla u \cdot \boldsymbol{v} = 0.$$

Then we have

$$\partial_t u \in L^{\frac{p^* t}{p^* t - 1}}(0, T; L^{\frac{2k^*}{2k^* - 1}}(\Omega)).$$

Proof.

Mass preservation:

Our goal is to show that $\partial_t u$, solving the continuity equation, acts as a bounded linear functional on $L^{p^*}(0, T; W^{1,2}(\Omega))$, thus being an element of the dual space $L^p(0, T; W^{-1,2}(\Omega))$ (see Theorem 2.1.14 for details about duality in Bochner spaces).

We write down the weak form of the continuity equation with some test function $\varphi \in L^{p^*}(0, T; W^{1,2}(\Omega))$

$$\int_0^T \int_\Omega \partial_t u \varphi \, \mathrm{d}x \, \mathrm{d}t = - \int_0^T \int_\Omega \nabla \cdot (u\boldsymbol{v}) \varphi \, \mathrm{d}x \, \mathrm{d}t = \int_0^T \int_\Omega u\boldsymbol{v} \cdot \nabla \varphi \, \mathrm{d}x \, \mathrm{d}t.$$

Here the boundary term vanishes, because we have zero flow towards the boundary of Ω. This leads to:

$$\begin{aligned}
\left| \int_0^T \int_\Omega \partial_t u \varphi \, \mathrm{d}x \, \mathrm{d}t \right| &= \left| \int_0^T \int_\Omega u\boldsymbol{v} \cdot \nabla \varphi \, \mathrm{d}x \, \mathrm{d}t \right| \\
&\leq \int_0^T \int_\Omega |u\boldsymbol{v} \cdot \nabla \varphi| \, \mathrm{d}x \, \mathrm{d}t \\
&\underbrace{\leq}_{\text{Hölder}} \int_0^T \left(\int_\Omega u^2 \, \mathrm{d}x \right)^{\frac{1}{2}} \left(\int_\Omega (\boldsymbol{v} \cdot \nabla \varphi)^2 \, \mathrm{d}x \right)^{\frac{1}{2}} \mathrm{d}t \qquad (5.12) \\
&\underbrace{\leq}_{\text{CS}} \int_0^T \|u\|_{L^2} \left(\int_\Omega \langle \boldsymbol{v}, \boldsymbol{v} \rangle \langle \nabla \varphi, \nabla \varphi \rangle \, \mathrm{d}x \right)^{\frac{1}{2}} \mathrm{d}t \\
&\underbrace{\leq}_{\text{Hölder}} \int_0^T \|u\|_{L^2} \|\boldsymbol{v}\|_{L^\infty} \left(\int_\Omega \langle \nabla \varphi, \nabla \varphi \rangle \, \mathrm{d}x \right)^{\frac{1}{2}} \mathrm{d}t \\
&= \int_0^T \|u\|_{L^2} \|\boldsymbol{v}\|_{L^\infty} \|\nabla \varphi\|_{L^2} \, \mathrm{d}t \\
&\leq c_{\boldsymbol{v}} \int_0^T \|u\|_{L^2} \|\varphi\|_{W^{1,2}} \, \mathrm{d}t
\end{aligned}$$

$$\underbrace{\leq}_{\text{Hölder}} c_{\boldsymbol{v}} \left(\int_0^T (\|u\|_{L^2})^p \, \mathrm{d}t \right)^{\frac{1}{p}} \left(\int_0^T (\|\varphi\|_{W^{1,2}})^{p^*} \, \mathrm{d}t \right)^{\frac{1}{p^*}}$$
$$\leq c_{\boldsymbol{v}} \|u\|_{L^p(0,T;L^2)} \|\varphi\|_{L^{p^*}(0,T;W^{1,2})}$$

Due to the continuous embedding of $BV(\Omega)$ into $L^2(\Omega)$ (for $\Omega \subset \mathbb{R}^2$, see Theorem 2.1.22), with Lemma 2.1.23, we get the continuous embedding

$$L^p(0,T;BV(\Omega)) \hookrightarrow L^p(0,T;L^2(\Omega)).$$

Thus,

$$\langle \partial_t u, \varphi \rangle := \int_0^T \int_\Omega \partial_t u \varphi \, \mathrm{d}x \, \mathrm{d}t \leq C \|\varphi\|_{L^{p^*}(0,T;W^{1,2})},$$

and $\partial_t u$ is a bounded linear functional on $L^{p^*}(0,T;W^{1,2}(\Omega))$. Consequently we get

$$\partial_t u \in (L^{p^*}(0,T;W^{1,2}(\Omega)))^* = L^p(0,T;W^{-1,2}(\Omega)).$$

Optical Flow:
For the optical flow equation we proceed in an analogous way and write down the constraint

$$\partial_t u + \nabla u \cdot \boldsymbol{v} = 0,$$

resp. its weak form

$$\int_0^T \int_\Omega \partial_t u \varphi \, \mathrm{d}x \, \mathrm{d}t = \int_0^T \int_\Omega u \nabla \cdot (\boldsymbol{v}\varphi) \, \mathrm{d}x \, \mathrm{d}t.$$

The estimates are quite similar to those for the continuity equation:

$$\left| \int_0^T \int_\Omega \partial_t u \varphi \, \mathrm{d}x \, \mathrm{d}t \right| = \left| \int_0^T \int_\Omega u \nabla \cdot (\boldsymbol{v}\varphi) \, \mathrm{d}x \, \mathrm{d}t \right|$$
$$\leq \int_0^T \int_\Omega \left| u \nabla \cdot (\boldsymbol{v}\varphi) \right| \mathrm{d}x \, \mathrm{d}t$$
$$\underbrace{\leq}_{\text{Hölder}} \int_0^T \left(\int_\Omega u^2 \, \mathrm{d}x \right)^{\frac{1}{2}} \left(\int_\Omega (\nabla \cdot (\boldsymbol{v}\varphi))^2 \, \mathrm{d}x \right)^{\frac{1}{2}} \mathrm{d}t$$
$$= \int_0^T \|u\|_{L^2} \left(\int_\Omega (\varphi \nabla \cdot \boldsymbol{v} + \boldsymbol{v} \cdot \nabla \varphi)^2 \, \mathrm{d}x \right)^{\frac{1}{2}} \mathrm{d}t$$

$$\underbrace{\leq}_{\text{Mink.}} \int_0^T \|u\|_{L^2} \left[\left(\int_\Omega (\varphi \nabla \cdot \boldsymbol{v})^2 \,\mathrm{d}x \right)^{\frac{1}{2}} + \left(\int_\Omega (\boldsymbol{v} \cdot \nabla \varphi)^2 \,\mathrm{d}x \right)^{\frac{1}{2}} \right] \mathrm{d}t$$
$$= \underbrace{\int_0^T \|u\|_{L^2} \left(\int_\Omega (\varphi \nabla \cdot \boldsymbol{v})^2 \,\mathrm{d}x \right)^{\frac{1}{2}} \mathrm{d}t}_{(i)} + \underbrace{\int_0^T \|u\|_{L^2} \left(\int_\Omega (\boldsymbol{v} \cdot \nabla \varphi)^2 \,\mathrm{d}x \right)^{\frac{1}{2}} \mathrm{d}t}_{(ii)}$$

An estimation for part (ii) has been shown in the former part (see 5.12), so we proceed with an estimation for part (i):

$$\int_0^T \|u\|_{L^2} \left(\int_\Omega (\varphi \nabla \cdot \boldsymbol{v})^2 \,\mathrm{d}x \right)^{\frac{1}{2}} \mathrm{d}t = \int_0^T \|u\|_{L^2} \left(\int_\Omega (\nabla \cdot \boldsymbol{v})^2 \varphi^2 \,\mathrm{d}x \right)^{\frac{1}{2}} \mathrm{d}t$$
$$\underbrace{\leq}_{\text{Hölder}} \int_0^T \|u\|_{L^2} \|\nabla \cdot \boldsymbol{v}\|_{L^{2k}} \|\varphi\|_{L^{2k^*}} \,\mathrm{d}t$$
$$\underbrace{\leq}_{\text{Hölder}} \left(\int_0^T (\|u\|_{L^2})^p \,\mathrm{d}t \right)^{\frac{1}{p}} \left(\int_0^T (\|\nabla \cdot \boldsymbol{v}\|_{L^{2k}} \|\varphi\|_{L^{2k^*}})^{p^*} \,\mathrm{d}t \right)^{\frac{1}{p^*}}$$
$$\underbrace{\leq}_{\text{Hölder}} \|u\|_{L^p(0,T;L^2)} \left(\int_0^T (\|\nabla \cdot \boldsymbol{v}\|_{L^{2k}})^{p^* s} \,\mathrm{d}t \right)^{\frac{1}{p^* s}} \left(\int_0^T (\|\varphi\|_{L^{2k^*}})^{p^* s^*} \,\mathrm{d}t \right)^{\frac{1}{p^* s^*}}$$
$$\leq \|u\|_{L^p(0,T;L^2)} \|\nabla \cdot \boldsymbol{v}\|_{L^{p^* s}(0,T;L^{2k})} \|\varphi\|_{L^{p^* s^*}(0,T;L^{2k^*})}$$

Combining the estimations for both parts we obtain

$$\left| \int_0^T \int_\Omega \partial_t u \varphi \,\mathrm{d}x \,\mathrm{d}t \right| \leq \|u\|_{L^p(0,T;L^2)} \|\nabla \cdot \boldsymbol{v}\|_{L^{p^* s}(0,T;L^{2k})} \|\varphi\|_{L^{p^* s^*}(0,T;L^{2k^*})}$$
$$+ \|u\|_{L^p(0,T;L^2)} \|\boldsymbol{v}\|_{L^{p^* s}(0,T;L^2)} \|\varphi\|_{L^{p^* s^*}(0,T;W^{1,2})}$$
$$\leq \|u\|_{L^p(0,T;L^2)} \|\nabla \cdot \boldsymbol{v}\|_{L^{p^* s}(0,T;L^{2k})} \|\varphi\|_{L^{p^* s^*}(0,T;L^{2k^*})}$$
$$+ \|u\|_{L^p(0,T;L^2)} \|\boldsymbol{v}\|_{L^{p^* s}(0,T;L^2)} \|\varphi\|_{L^{p^* s^*}(0,T;L^{2k^*})}$$
$$\leq (\|\nabla \cdot \boldsymbol{v}\|_{L^{p^* s}(0,T;L^{2k})} + \|\boldsymbol{v}\|_{L^{p^* s}(0,T;L^2)}) \|u\|_{L^p(0,T;L^2)} \|\varphi\|_{L^{p^* s^*}(0,T;L^{2k^*})} .$$

In the first inequality, we used the embedding

$$W^{1,2}(\Omega) \hookrightarrow L^{2k^*}(\Omega), \quad \forall k^* < \infty.$$

The sum of the first terms is bounded because of the assumptions made above. A

bound for u follows again from the $BV(\Omega)$ embedding (2.1.22) and we conclude

$$\langle \partial_t u, \varphi \rangle := \int_0^T \int_\Omega \partial_t u \varphi \, \mathrm{d}x \, \mathrm{d}t \leq C \, \|\varphi\|_{L^{p^* s^*}(0,T;L^{2k^*})} .$$

Thus, $\partial_t u$ forms a bounded linear functional on $L^{p^* s^*}(0,T;L^{2k^*}(\Omega))$ and we end up with

$$\partial_t u \in (L^{p^* s^*}(0,T;L^{2k^*}(\Omega)))^* = L^{\frac{p^* s^*}{p^* s^* - 1}}(0,T;L^{\frac{2k^*}{2k^*-1}}(\Omega)).$$

□

Having deduced bounds for $\partial_t u$ we are now able to apply the Aubin-Lions Lemma 2.1.16 with the following theorem:

Theorem 5.2.6. Compact embedding for u
Let be $\Omega \subset \mathbb{R}^2$, $\boldsymbol{v} \in L^q(0,T;W^{1,2}(\Omega))$ and furthermore for $c_v \in \mathbb{R}^+$

$$\|\boldsymbol{v}\|_\infty \leq c_v < \infty \quad \text{a.e. in } \Omega \times [0,T] .$$

1. **Mass Preservation:**
   The set

   $$\left\{ u : u \in L^p(0,T;BV(\Omega)), u_t + \nabla \cdot (u\boldsymbol{v}) = 0 \right\}$$

   can be compactly embedded into

   $$L^p(0,T;L^r(\Omega)), \quad \text{for } 1 < r < 2.$$

2. **Optical Flow:**
   Let furthermore $\nabla \cdot \boldsymbol{v} \in L^{p^* s}(0,T;L^{2k}(\Omega))$ with $\frac{1}{k} + \frac{1}{k^*} = 1 = \frac{1}{p} + \frac{1}{p^*}$ and $s, k^* > 1$.
   Then the set

   $$\left\{ u : u \in L^p(0,T;BV(\Omega)), u_t + \nabla u \cdot \boldsymbol{v} = 0 \right\}$$

   can be compactly embedded into

   $$L^p(0,T;L^r(\Omega)), \quad \text{for } \frac{2k^*}{2k^* - 1} \leq r < 2.$$

Proof.

$$\begin{array}{ccc} L^r(\Omega) & \hookrightarrow & W^{-1,2}(\Omega) \\ {\scriptstyle dual}\updownarrow & & \uparrow{\scriptstyle dual} \\ L^{r^*}(\Omega) & \hookleftarrow & W^{1,2}(\Omega) \end{array}$$

Figure 5.1.: Dual embedding for $L^r(\Omega)$

Mass Preservation:
We have a natural a-priori estimate for u in $L^p(0,T;BV(\Omega))$. Using the weak formulation of the continuity equation we previously deduced a bound for $\partial_t u$ in $L^p(0,T;W^{-1,2}(\Omega))$ (see Lemma 5.2.5). Speaking in terms of the Aubin-Lions lemma (see 2.1.16) we set $\mathcal{X} = BV(\Omega)$ and $\mathcal{Z} = W^{-1,2}(\Omega)$. The crucial point is to find a proper r such that $\mathcal{Y} = L^r(\Omega)$ and

$$BV(\Omega) \subset\subset \mathcal{Y} \hookrightarrow W^{-1,2}(\Omega).$$

Embeddings of $BV(\Omega)$ into $L^r(\Omega)$ are compact for $r < \frac{n}{n-1}$ (see Theorem 2.1.22), where n is the spatial dimension. Applied to our case of dimension 2 we obtain $r < 2$ as a first constraint. Now we have to ensure that $L^r(\Omega)$ embeds continuously into $W^{-1,2}(\Omega)$. This can be shown by a dual argument. L^r is the dual space of $L^{r^*}(\Omega)$, satisfying $\frac{1}{r} + \frac{1}{r*} = 1$, and $W^{-1,2}(\Omega)$ is the dual space of $W^{1,2}(\Omega)$. Finding an embedding of $W^{1,2}(\Omega)$ into $L^{r^*}(\Omega)$ implies the opposite for the dual spaces (see Figure 5.1). Using the embedding theory for Sobolev spaces, $W^{1,2}(\Omega)$ can be continuously embedded into $L^{r^*}(\Omega)$ for $2 \leq r^* < \infty$ (see Theorem 2.1.19). Translated to the dual exponent r it requires $1 < r \leq 2$. Combining both restrictions to r we obtain

$$BV(\Omega) \subset\subset L^r(\Omega) \hookrightarrow W^{-1,2}(\Omega), \quad \text{for } 1 < r < 2.$$

Applying the Aubin-Lions lemma yields the compact embedding

$$\left\{u : u \in L^p(0,T;BV(\Omega)), u_t + \nabla \cdot (u\boldsymbol{v}) = 0\right\} \subset\subset L^p(0,T;L^r(\Omega)), \quad \text{for } 1 < r < 2.$$

Optical Flow:
We recall the arguments above and directly get an a-priori estimate for u in $L^p(0,T;BV(\Omega))$. From the optical flow equation we moreover deduced a bound for $\partial_t u$ in $L^{\frac{p^* t}{p^* t - 1}}(0,T;L^{\frac{2k^*}{2k^*-1}}(\Omega))$. Using the BV-embedding we obtain $\mathcal{Y} = L^r(\Omega)$ for $r < 2$ (see Theorem 2.1.22). For

the embedding $L^r(\Omega) \hookrightarrow L^{\frac{2k^*}{2k^*-1}}(\Omega)$ we obviously need $r \geq \frac{2k^*}{2k^*-1}$. It follows

$$BV(\Omega) \subset\subset L^r(\Omega) \hookrightarrow L^{\frac{2k^*}{2k^*-1}}(\Omega), \quad \text{for } \frac{2k^*}{2k^*-1} \leq r < 2.$$

Another application of the Aubin-Lions lemma yields the compact embedding

$$\left\{u : u \in L^p(0,T;BV(\Omega)), u_t + \nabla u \cdot \boldsymbol{v} = 0\right\} \subset\subset L^p(0,T;L^r(\Omega)), \quad \text{for } \frac{2k^*}{2k^*-1} \leq r < 2.$$

□

Let us also mention that $\lim_{k^*\to\infty} \frac{2k^*}{2k^*-1} = 1$, so choosing a larger k^* is favorable

Example:
Let us give a concrete example for all the constants we previously defined. We start setting $p = q = 2$. What is a suitable embedding for functions solving the continuity equation or the optical flow equation? This is fairly simple to answer for the continuity case. We get i.e. a compact embedding $L^2(0,T;BV(\Omega))$ to the fractional Bochner space $L^2(0,T;L^{\frac{3}{2}}(\Omega))$. This means that for every $t \in [0,T]$ u lies in the fractional Lebesgue space $L^{\frac{3}{2}}(\Omega)$.
For the optical flow case we additionally need a bound for the divergence of $\boldsymbol{v}$. This bound should be as low as possible, meaning that the exponentials in $L^{p^*s}(0,T;L^{2k}(\Omega))$ should be as low as possible. We directly get $p^* = 2$, because p^* is the conjugate of $p = 2$. The variable s is connected to the bound for $\partial_t u$, so we are fairly free here, but set $s = \frac{3}{2}$. The last variable states k, which is set to $\frac{3}{2}$. This implies $k^* = 3$. This results in the assumption $\nabla \cdot \boldsymbol{v}$ in $L^3(0,T;L^3(\Omega))$ as a bound. Consequently, we have $\frac{2k^*}{2k^*-1} = \frac{6}{5} \leq r < 2$. Therefore $r = \frac{3}{2}$ is a valid choice and we again get a compact embedding $L^2(0,T;BV(\Omega))$ to the fractional Bochner space $L^2(0,T;L^{\frac{3}{2}}(\Omega))$.

With this compact embedding result we conclude with strong convergence for u_k to u and are now able to show convergence of the product $\nabla u_k \cdot \boldsymbol{v}_k$, resp. $\nabla \cdot (u_k \boldsymbol{v}_k)$, to the products of their individual limits $\nabla u \cdot \boldsymbol{v}$, resp. $\nabla \cdot (u\boldsymbol{v})$.

Lemma 5.2.7. Convergence of the constraints
Let be $\Omega \subset \mathbb{R}^2$, $u_k \in L^p(0,T;BV(\Omega))$ and $\boldsymbol{v} \in L^q(0,T;W^{1,2}(\Omega))$. Let furthermore be

$$\|\boldsymbol{v}\|_\infty \leq c_v < \infty \quad \text{a.e. in } \Omega \times [0,T].$$

1. **Mass Preservation:**
   Then

$$\partial_t u_k + \nabla \cdot (u_k \boldsymbol{v_k}) \rightharpoonup \partial_t u + \nabla \cdot (u\boldsymbol{v})$$

   in the sense of distributions.

2. **Optical Flow:**
   Let furthermore

$$\nabla \cdot \boldsymbol{v} \in L^{p^* s}(0, T; L^{2k}(\Omega)),$$

   with $k > 1, s > 1, \frac{1}{p} + \frac{1}{p^*} = 1$.
   Then

$$\partial_t u_k + \nabla u_k \cdot \boldsymbol{v_k} \rightharpoonup \partial_t u + \nabla u \cdot \boldsymbol{v}$$

   in the sense of distributions.

Proof. For the following proof let be $\varphi \in C_0^\infty(\Omega), u_k \in L^p(0, T; BV(\Omega))$ and $\boldsymbol{v}_k \in L^q(0, T; W^{1,2}(\Omega))$. For the time derivative ∂_t in both constraints we simply calculate

$$\int_0^T \int_\Omega (\partial_t u_k - \partial_t u)\, \varphi \,\mathrm{d}x \,\mathrm{d}t = -\int_0^T \int_\Omega (u_k - u)\, \partial_t \varphi \,\mathrm{d}x \,\mathrm{d}t \to 0.$$

Since test functions are dense in the dual space of u, we directly get convergence from the weak convergence $u_k \rightharpoonup u$.

Mass preservation:
We want to prove the convergence result

$$\nabla \cdot (u_k \boldsymbol{v_k}) \rightharpoonup \nabla \cdot (u\boldsymbol{v}),$$

and start by writing down the weak form of the equation as

$$\begin{aligned}
-\int_0^T \int_\Omega \big(\nabla \cdot (u_k \boldsymbol{v}_k) - \nabla \cdot (u\boldsymbol{v})\big)\, \varphi \,\mathrm{d}x \,\mathrm{d}t &= \int_0^T \int_\Omega (u_k \boldsymbol{v}_k - u\boldsymbol{v}) \cdot \nabla \varphi \,\mathrm{d}x \,\mathrm{d}t \\
&= \int_0^T \int_\Omega (u_k \boldsymbol{v}_k - u\boldsymbol{v}_k + u\boldsymbol{v}_k - u\boldsymbol{v}) \cdot \nabla \varphi \,\mathrm{d}x \,\mathrm{d}t \\
&= \int_0^T \int_\Omega \big((u_k - u)\boldsymbol{v}_k + u(\boldsymbol{v}_k - \boldsymbol{v})\big) \cdot \nabla \varphi \,\mathrm{d}x \,\mathrm{d}t
\end{aligned}$$

$$
\begin{aligned}
&\leq \int_0^T \|u_k - u\|_{L^r} \, \|\boldsymbol{v}_k \cdot \nabla\varphi\|_{L^{r^*}} \\
&+ \int_\Omega u(\boldsymbol{v}_k - \boldsymbol{v}) \cdot \nabla\varphi \, \mathrm{d}x \, \mathrm{d}t \\
&\leq \|u_k - u\|_{L^p(0,T;L^r)} \, \|\boldsymbol{v}_k \cdot \nabla\varphi\|_{L^{p^*}(0,T;L^{r^*})} \\
&+ \int_0^T \int_\Omega u(\boldsymbol{v}_k - \boldsymbol{v}) \cdot \nabla\varphi \, \mathrm{d}x \, \mathrm{d}t \\
&\leq \underbrace{C \, \|u_k - u\|_{L^p(0,T;L^r)}}_{(i)} \\
&+ \underbrace{\int_0^T \int_\Omega u(\boldsymbol{v}_k - \boldsymbol{v}) \cdot \nabla\varphi \, \mathrm{d}x \, \mathrm{d}t}_{(ii)} .
\end{aligned}
$$

We have proven the compact embedding

$$
\left\{ u : u \in L^p(0,T;BV(\Omega)), u_t + \nabla \cdot (u\boldsymbol{v}) = 0 \right\} \subset\subset L^p(0,T;L^r(\Omega)), \qquad \text{for } 1 < r < 2.
$$

Hence, for the bounded sequence $u_k \in L^p(0,T;BV(\Omega)$, we obtain $u_k \to u$ strongly in $L^p(0,T;L^r)$ and (i) tends to zero as $k \to \infty$.
For part (ii), the boundedness of $\boldsymbol{v}$ gives us $\boldsymbol{v} \in L^\infty([0,T] \times \Omega)$ and a-priori weak-* convergence (see Remark 4 for details). Consequently, for convergence of (ii) we need $u\nabla\varphi \in L^1([0,T] \times \Omega)$. Due to the compact embedding $BV(\Omega) \subset\subset L^1(\Omega)$ (see (2.1.22)) and $p > 1$ we get

$$
L^p(0,T;BV(\Omega)) \hookrightarrow L^p(0,T;L^1(\Omega)) \hookrightarrow L^1(0,T;L^1(\Omega)).
$$

This gives us $u \in L^1([0,T] \times \Omega)$ and since test functions are dense in L^1 we end up with the required $u\nabla\varphi \in L^1([0,T] \times \Omega)$. We conclude

$$
\begin{aligned}
\lim_{k\to\infty} \left| \int_0^T \int_\Omega u_k\boldsymbol{v}_k - u\boldsymbol{v} \, \mathrm{d}x \, \mathrm{d}t \right| \leq \lim_{k\to\infty} C \, \|u_k - u\|_{L^p(0,T;L^r)} + \\
\left| \int_0^T \int_\Omega u(\boldsymbol{v}_k - \boldsymbol{v}) \cdot \nabla\varphi \, \mathrm{d}x \, \mathrm{d}t \right| = 0.
\end{aligned}
$$

Optical flow:

We begin with an analogous argumentation and estimate

$$\begin{aligned}
-\int_0^T \int_\Omega (\nabla u_k \cdot \boldsymbol{v}_k - \nabla u \cdot \boldsymbol{v})\, \varphi \,\mathrm{d}x\,\mathrm{d}t &= \int_0^T \int_\Omega u_k \nabla \cdot (\varphi \boldsymbol{v}_k) - u \nabla \cdot (\varphi \boldsymbol{v})\,\mathrm{d}x\,\mathrm{d}t \\
&= \int_0^T \int_\Omega u_k \nabla \cdot (\varphi \boldsymbol{v}_k) - u \nabla \cdot (\varphi \boldsymbol{v}_k) \\
&\quad + u \nabla \cdot (\varphi \boldsymbol{v}_k) - u \nabla \cdot (\varphi \boldsymbol{v})\,\mathrm{d}x\,\mathrm{d}t \\
&= \int_0^T \int_\Omega (u_k - u) \nabla \cdot (\varphi \boldsymbol{v}_k) \\
&\quad + u \nabla \cdot (\varphi (\boldsymbol{v}_k - \boldsymbol{v}))\,\mathrm{d}x\,\mathrm{d}t \\
&= \underbrace{\int_0^T \int_\Omega (u_k - u) \nabla \cdot (\varphi \boldsymbol{v}_k)\,\mathrm{d}x\,\mathrm{d}t}_{(i)} \\
&\quad + \underbrace{\int_0^T \int_\Omega u \nabla \cdot (\varphi (\boldsymbol{v}_k - \boldsymbol{v}))\,\mathrm{d}x\,\mathrm{d}t}_{(ii)}.
\end{aligned}$$

Part (i) can be estimated as follows:

$$\begin{aligned}
\int_0^T \int_\Omega (u_k - u) \nabla \cdot (\varphi \boldsymbol{v}_k)\,\mathrm{d}x\,\mathrm{d}t &\leq \int_0^T \|u_k - u\|_{L^r} \left\|\nabla \cdot (\varphi \boldsymbol{v}_k)\right\|_{L^{r^*}} \mathrm{d}t \\
&\leq \|u_k - u\|_{L^p(0,T;L^r)} \left\|\nabla \cdot (\varphi \boldsymbol{v}_k)\right\|_{L^{p^*}(0,T;L^{r^*})} \\
&= \|u_k - u\|_{L^p(0,T;L^r)} \left\|\varphi \nabla \cdot \boldsymbol{v}_k + \boldsymbol{v}_k \cdot \nabla \varphi\right\|_{L^{p^*}(0,T;L^{r^*})} \\
&\leq \|u_k - u\|_{L^p(0,T;L^r(\Omega))} \cdot \\
&\quad \Big(\underbrace{\|\varphi \nabla \cdot \boldsymbol{v}_k\|_{L^{p^*}(0,T;L^{r^*}(\Omega))}}_{(i.1)} + \underbrace{\|\boldsymbol{v}_k \cdot \nabla \varphi\|_{L^{p^*}(0,T;L^{r^*}(\Omega))}}_{(i.2)}\Big)
\end{aligned}$$

To start with $(i.1)$, we have to show a bound for this term and therefore, have to argue with the boundedness of $\nabla \cdot \boldsymbol{v}$. From the assumptions above we a-priori get

$$\nabla \cdot \boldsymbol{v} \in L^{p^* s}(0,T;L^{2k}(\Omega)).$$

Consequently, we have to prove that

$$L^{p^* s}(0,T;L^{2k}(\Omega)) \hookrightarrow L^{p^*}(0,T;L^{r^*}(\Omega)).$$

Speaking in terms of the embedding theory of Lebesgue spaces we need $p^* s \geq p^*$ and $2k \geq r^*$. Since $s > 1$, the first inequality is obvious. At this point it is important

to keep in mind that r and r^* are Hölder-conjugated and the embedding theorem for optical flow allows only $\frac{2k^*}{2k^*-1} \leq r < 2$. We can translate the condition $2k \geq r^*$ as follows:

$$2k \geq r^* \Leftrightarrow 2k \geq \frac{r}{r-1} \Leftrightarrow 2kr - r \geq 2k \Leftrightarrow (2k-1)r \geq 2k \Leftrightarrow \frac{2k}{2k-1} \leq r \Leftrightarrow \frac{2k^*}{k^*+1} \leq r$$

Now for $k^* > 2$ we have $\frac{2k^*}{k^*+1} < \frac{2k^*}{k^*-1}$ and r can be chosen arbitrarily as long as $\frac{2k^*}{2k^*-1} \leq r < 2$. On the other hand, for $k^* \leq 2$ we have $\frac{2k^*}{k^*+1} > \frac{2k^*}{k^*-1}$. In this case we set $r = 3/2$ and obtain $\frac{2k^*}{k^*+1} \leq r$ as long as $1 < k^* \leq 2$. This yields the required bound for $\nabla \cdot \boldsymbol{v}_k$ in $L^{p^*}(0,T;L^{r^*}(\Omega))$.

Part $(i.2)$ has been discussed in the mass preservation case. Following the arguments for strong convergence of u from above, we conceive that (i) tends to zero.

Estimating part (ii) again requires Lebesgue embedding theory, since

$$\int_0^T \int_\Omega u\nabla \cdot (\varphi(v_k - v))\,\mathrm{d}x\,\mathrm{d}t = \int_0^T \int_\Omega \underbrace{u\varphi\nabla \cdot (v_k - v)}_{(ii).1} + \underbrace{u(v_k - v)\cdot\nabla\varphi}_{(ii).2}\,\mathrm{d}x\,\mathrm{d}t,$$

and we see that $(ii).2$ has already been discussed in the mass preservation part. In $(ii).1$ we have the bounded sequence

$$\nabla \cdot \boldsymbol{v}_k \in L^{p^*s}(0,T;L^{2k}(\Omega)),$$

thus, weak, resp. weak-*, convergence for

$$u\varphi \in L^{(p^*s)^*}(0,T;L^{(2k)^*}(\Omega)).$$

Since $k > 1$ we estimate

$$(2k)^* = \frac{2k}{2k-1} < 2 \quad \forall k > 1,$$

which gives, using the embedding theory (see Theorem 2.1.22), the continuous embedding

$$BV(\Omega) \hookrightarrow L^{\frac{2k}{2k-1}}(\Omega) = L^{(2k)^*}(\Omega) \quad \forall k > 1.$$

We moreover calculate

$$(p^*s)^* = \frac{p^*s}{p^*s-1} = \frac{\frac{p}{p-1}s}{\frac{p}{p-1}s-1} = \frac{ps}{ps-p+1},$$

and use this result for the inequality

$$\frac{ps}{ps-p+1} < p \Leftrightarrow ps < p^2s - p^2 + p \Leftrightarrow s < ps - p + 1 \Leftrightarrow s - 1 < p(s-1) \Leftrightarrow 1 < p.$$

Since $p > 1$ is one of our main assumptions we continue with the embedding

$$\begin{aligned} L^p(0,T;BV(\Omega)) &\hookrightarrow L^p(0,T;L^{\frac{2k}{2k-1}}(\Omega)) = L^p(0,T;L^{(2k)^*}(\Omega)) \\ &\hookrightarrow L^{\frac{ps}{ps-p+1}}(0,T;L^{(2k)^*}(\Omega)) = L^{(p^*s)^*}(0,T;L^{(2k)^*}(\Omega)). \end{aligned}$$

Consequently, we have $u \in L^{(p^*s)^*}(0,T;L^{(2k)^*}(\Omega))$, which is the dual of $L^{p^*s}(0,T;L^{2k}(\Omega))$. Due to the weak-* convergence of $\nabla \cdot \boldsymbol{v}_k$ part $(ii).1$ tends to 0 as $k \to \infty$.

Putting all arguments together we end up with convergence of the constraint

$$\begin{aligned} \lim_{k\to\infty} \left| \int_0^T \int_\Omega (\nabla u_k \cdot \boldsymbol{v}_k - \nabla u \cdot \boldsymbol{v})\, \varphi \,\mathrm{d}x\,\mathrm{d}t \right| &\leq \lim_{k\to\infty} \left| \int_0^T \int_\Omega (u_k - u) \nabla \cdot (\varphi v_k)\,\mathrm{d}x\,\mathrm{d}t \right| \\ &\quad + \lim_{k\to\infty} \left| \int_0^T \int_\Omega u \nabla \cdot (\varphi(v_k - v))\,\mathrm{d}x\,\mathrm{d}t \right| \\ &\leq C \lim_{k\to\infty} \|u_k - u\|_{L^p(0,T;L^r)} \\ &\quad + \lim_{k\to\infty} \left| \int_0^T \int_\Omega u\varphi \nabla \cdot (v_k - v)\,\mathrm{d}x\,\mathrm{d}t \right| \\ &\quad + \lim_{k\to\infty} \left| \int_0^T \int_\Omega u(v_k - v) \cdot \nabla\varphi \,\mathrm{d}x\,\mathrm{d}t \right| \\ &= 0. \end{aligned}$$

□

We are finally able to use the fundamental theorem of optimization and prove existence of a minimizer for Model (5.2.1) and Model (5.2.2).

Theorem 5.2.8. Existence of a minimizer for the $TV - L^2$ model
Let $\Omega \subset \mathbb{R}^2, p, q > 1, \hat{p} = \min\{p, 2\}, T \in \mathbb{R}^+$ and consider the $TV - L^2$ energy

$$J(u, \boldsymbol{v}) = \int_0^T \frac{1}{2} \|Ku - f\|_2^2 \, \mathrm{d}t + \alpha \int_0^T \left|u(\cdot, t)\right|_{BV}^p \mathrm{d}t + \frac{\beta}{2} \int_0^T \left\|\nabla \boldsymbol{v}(\cdot, t)\right\|_{L^2}^{2q} \mathrm{d}t.$$

Let furthermore be

$$\|\boldsymbol{v}\|_\infty \leq c_v < \infty \quad \text{a.e. in } \Omega \times [0, T] \text{ and } K\mathbf{1}_t \neq 0 \quad \forall t \in [0, T].$$

Mass Preservation:
Then there exists a minimizer $(\tilde{u}, \tilde{\boldsymbol{v}})$ in the set

$$\left\{(u, \boldsymbol{v}) : u \in L^{\hat{p}}(0, T; BV(\Omega)), \boldsymbol{v} \in L^q(0, T; W^{1,2}(\Omega)), \partial_t u + \nabla \cdot (u\boldsymbol{v}) = 0\right\}.$$

Optical Flow:
Let furthermore be $s, k > 1$ and p^* such that $\frac{1}{p} + \frac{1}{p^*} = 1$. Then there also exists a minimizer in the set

$$\begin{aligned}\{(u, \boldsymbol{v}) :& u \in L^{\hat{p}}(0, T; BV(\Omega)), \boldsymbol{v} \in L^q(0, T; W^{1,2}(\Omega)), \\ & \nabla \cdot \boldsymbol{v} \in L^{p^* s}(0, T; L^{2k}(\Omega)), \partial_t u + \nabla \cdot (u\boldsymbol{v}) = 0\}.\end{aligned}$$

Proof. We haven proven all requirements for the fundamental theorem of optimization (Theorem 2.2.17). Compactness of the sublevel sets with respect to the weak-* topology follows from Lemma 5.2.3 and the energy has been proven to be weak-* lower semicontinuous (Lemma 5.2.4).
Finally, for the sequence $(u_k, \boldsymbol{v}_k)$ both constraints converge in a distributional sense (Lemma 5.2.7). It follows that Model 5.2.1 and Model 5.2.2, equipped with the generalized energy (Equation (5.10)), have a minimizer. □

Uniqueness

The energy for mass preserving TV-L^2 model and optical flow TV-L^2 model is convex. However, we are not able to prove uniqueness, since the continuity equation

$$\partial_t u + \nabla \cdot (u\boldsymbol{v}) = 0$$

and the optical flow equation

$$\partial_t + \nabla u \cdot \boldsymbol{v} = 0$$

are both non-linear.

5.2.3. Numerical Realization

The numerical realization is divided in three steps: the primal-dual framework, the discretization and the algorithm. First, the general minimization problem with objective variables u and $\boldsymbol{v}$ is divided in two separate minimization problems. Starting with the problem in u, the primal-dual framework of Chambolle and Pock [26] is applied, which leads to a short and easy to implement numerical scheme. A similar scheme is then deduced for $\boldsymbol{v}$. After that we proceed to the discretized counterparts of u and $\boldsymbol{v}$ and to explain a discretization strategy for the involved gradient operators. The section ends with an algorithm that minimizes the proposed variational energies written in pseudocode.

As a first step we transform the constrained optimization problems in Models 5.2.2 and 5.2.1 into unconstrained ones using an L^1 penalty term with a weighting parameter γ. This gives us the unconstrained $TV - L^2$ optical flow model

$$\arg\min_{u,\boldsymbol{v}} \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \frac{\beta}{2} \|\nabla_x \boldsymbol{v}\|_2^2 + \gamma \|u_t + \nabla_x u \cdot \boldsymbol{v}\|_1 \, \mathrm{d}t, \tag{5.13}$$

resp. the unconstrained $TV - L^2$ mass preservation model

$$\arg\min_{u,\boldsymbol{v}} \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \frac{\beta}{2} \|\nabla_x \boldsymbol{v}\|_2^2 + \gamma \left\|u_t + \nabla_x \cdot (u\boldsymbol{v})\right\|_1 \mathrm{d}t. \tag{5.14}$$

Since the optical flow model is more easily to handle from a computational point of view, we start with this model and afterwards present an adjusted strategy for the mass preservation model afterwards.

The implementation has been done in MATLAB. In our application it turned out that the motion estimation part is always the bottleneck concerning costs of time. Consequently, this part is ported to a C^{++} mex file and parallelized using OpenMP, which turns out to be faster by a factor of approximately 50 compared to the MATLAB implementation.

Realization of the $TV - L^2$ Optical Flow Model

Primal-dual Framework

Due to the dependence of energy (5.13) on u and $\boldsymbol{v}$ we propose an alternating minimization technique. First we minimize with respect to u, then with respect to $\boldsymbol{v}$. Consequently, we get the scheme

$$u^{k+1} = \arg\min_u \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \gamma \left\| u_t + \nabla_x u \cdot \boldsymbol{v}^k \right\|_1 \mathrm{d}t, \tag{5.15}$$

$$\boldsymbol{v}^{k+1} = \arg\min_{\boldsymbol{v}=(v_1,v^2)} \int_0^T \left\| u_t^{k+1} + \nabla_x u^{k+1} \cdot \boldsymbol{v} \right\|_1 + \frac{\beta}{2\gamma} \|\nabla_x \boldsymbol{v}\|_2^2 \,\mathrm{d}t. \tag{5.16}$$

Problem in u:

For subproblem (5.15) we apply the primal-dual Chambolle Pock algorithm [26] (see Section 2.3.1 for a detailed explanation) to achieve a fast and robust algorithm that is able to tackle the involved discontinuities in the L^1-terms. We inherit the previously introduced notation from Section 2.2.6. Due to the fact that an operator is applied to u in every of the three terms we put the whole energy into the dual part and define

$$F(Cu) = \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \gamma \left\| (1, v_1^k, v_2^k) \cdot (u_t, u_x, u_y) \right\|_1 \mathrm{d}t,$$

with an underlying operator

$$Cu = \begin{pmatrix} K \\ \nabla_x \\ (\partial_t, \partial_x, \partial_y) \end{pmatrix} u$$

and

$$G(u) = 0.$$

The adjoint operator of C is directly given by

$$C^* \boldsymbol{y} = K^* y_1 - \nabla_x \cdot y_2 - (\partial_t(y_{3,1}) + \partial_x(y_{3,2}) + \partial_y(y_{3,3})).$$

We mention that for the sake of generality we do not specify the linear operator K (which can for instance be chosen as convolution or inpainting). Incorporating the

primal-dual formulation into the Chambolle-Pock scheme we obtain:

$$\tilde{\boldsymbol{y}}^{l+1} = \boldsymbol{y}^l + \sigma C \bar{u}^l \tag{5.17}$$

$$y_1^{l+1} = \arg\min_y \left\{ \int_0^T \frac{1}{2} \left\| y - \tilde{y_1}^{l+1} \right\|_2^2 + \frac{\sigma}{2} \|y\|_2^2 + \sigma \langle y, f \rangle \, \mathrm{d}t \right\} \tag{5.18}$$

$$y_2^{l+1} = \arg\min_y \left\{ \int_0^T \frac{1}{2} \left\| y - \tilde{y_2}^{l+1} \right\|_2^2 + \sigma\alpha \delta_{B(L^\infty)}(y/\alpha) \, \mathrm{d}t \right\} \tag{5.19}$$

$$y_3^{l+1} = \tilde{y_3}^{l+1} - \sigma \arg\min_y \left\{ \int_0^T \frac{1}{2} \left\| y - \frac{\tilde{y_3}^{l+1}}{\sigma} \right\|_2^2 + \frac{\gamma}{\sigma} \left\| (1, v_1^k, v_2^k) \cdot \boldsymbol{y} \right\|_1 \mathrm{d}t \right\} \tag{5.20}$$

$$u^{l+1} = \arg\min_u \left\{ \int_0^T \frac{1}{2} \left\| u - (u^l - \tau C^* \boldsymbol{y}^{l+1}) \right\|_2^2 \mathrm{d}t \right\} \tag{5.21}$$

$$\bar{u}^{l+1} = u^{l+1} + \theta(u^{l+1} - u^l). \tag{5.22}$$

The subproblem for y_1 in Equation (5.18) is equivalent to the following problem (see Section 4.5.7 for details):

$$\arg\min_y \left\{ \int_0^T \frac{1}{2} \left\| y - (\tilde{y_1}^{l+1} - \sigma f) \right\|_2^2 + \frac{\sigma}{2} \|y\|_2^2 \, \mathrm{d}t \right\}.$$

Since this is a linear L^2 problem, the solution can be directly found as

$$y_1^{l+1} = \frac{\tilde{y}_1^{l+1}}{\sigma + 1} - \frac{\sigma}{\sigma + 1} f.$$

The subproblem for y_2 has been extensively discussed in Section 4.5.4 and we get as a solution the point-wise projection of $\tilde{y_2}^l$ onto $[-\alpha, \alpha]$:

$$y_2^{l+1} = \pi_\alpha(\tilde{y_2}^{l+1}).$$

For the problem in y_3 we are able to apply the affine linear L^1 Lemma. Therefore, we define:

$$\boldsymbol{\beta} = (1, v_1^k, v_2^k), \quad \|\boldsymbol{\beta}\|^2 = \sqrt{1 + v_1^k v_1^k + v_2^k v_2^k}, \quad f(\boldsymbol{y}) = (1, v_1^k, v_2^k) \cdot \boldsymbol{y}.$$

Then the solution to the inner problem is given by

$$y = \frac{\tilde{y}_3^{l+1}}{\sigma} + \begin{cases} \frac{\gamma}{\sigma}\boldsymbol{\beta} & \text{if } f(\frac{\tilde{y}_3^{l+1}}{\sigma}) < -\frac{\gamma}{\sigma}\|\boldsymbol{\beta}\|^2 \\ -\frac{\gamma}{\sigma}\boldsymbol{\beta} & \text{if } f(\frac{\tilde{y}_3^{l+1}}{\sigma}) > \frac{\gamma}{\sigma}\|\boldsymbol{\beta}\|^2 \\ -\frac{f(\frac{\tilde{y}_3^{l+1}}{\sigma})\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|^2} & \text{else} \end{cases}$$

The solution for u^{l+1} can be calculated directly. The optimality condition reads:

$$u - (u^l - \tau C^* \boldsymbol{y}^{l+1}) \stackrel{!}{=} 0,$$

and for the solution we get

$$u = u^l - \tau C^* \boldsymbol{y}^{l+1}.$$

Summing up our results we obtain for static $\boldsymbol{v}^k$ the following scheme, which for $l \to \infty$ converges to the solution of problem (5.15), namely u^{k+1}:

$$\begin{aligned} \tilde{\boldsymbol{y}}^{l+1} &= \boldsymbol{y}^l + \sigma C \bar{u}^l \\ y_1^{l+1} &= \frac{\tilde{y}_1^{l+1} - \sigma f}{\sigma + 1} \\ y_2^{l+1} &= \pi_\alpha(\tilde{y_2}^{l+1}) \\ y_3^{l+1} &= \begin{cases} -\gamma\boldsymbol{\beta} & \text{if } f(\frac{\tilde{y}_3^{l+1}}{\sigma}) < -\frac{\gamma}{\sigma}\|\boldsymbol{\beta}\|^2 \\ \gamma\boldsymbol{\beta} & \text{if } f(\frac{\tilde{y}_3^{l+1}}{\sigma}) > \frac{\gamma}{\sigma}\|\boldsymbol{\beta}\|^2 \\ \frac{\sigma f(\frac{\tilde{y}_3^{l+1}}{\sigma})\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|^2} & \text{else} \end{cases} \\ u^{l+1} &= u^l - \tau C^* \boldsymbol{y}^{l+1} \\ \bar{u}^{l+1} &= u^{l+1} + \theta(u^{l+1} - u^l) \end{aligned}$$

Problem in $\boldsymbol{v}$:
A similar strategy can be used to solve problem (5.16), namely the update $\boldsymbol{v}^{k+1}$. We mention that, due to the alternating minimization scheme, u^{k+1} from the previous scheme is available. We decouple the regularization term into the dual functional and define

$$F(C\boldsymbol{v}) := \int_0^T \frac{\beta}{2\gamma} \|\nabla_x \boldsymbol{v}\|_2^2 \, \mathrm{d}t,$$

with

$$C\boldsymbol{v} = \begin{pmatrix} \nabla_x & 0 \\ 0 & \nabla_x \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \quad C^*\boldsymbol{y} = -\begin{pmatrix} \nabla_x \cdot & 0 \\ 0 & \nabla_x \cdot \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

The optical flow part is linear in $\boldsymbol{v}$ (the derivatives of u^{k+1} are static) and is put into the primal functional as follows:

$$G(\boldsymbol{v}) := \int_0^T \left\| u_t^{k+1} + \nabla_x u^{k+1} \cdot \boldsymbol{v} \right\|_1 \mathrm{d}t.$$

Calculating the dual functional F^* corresponding to F yields:

$$F^*(\boldsymbol{y}) := \frac{\gamma}{2\beta} \|\boldsymbol{y}\|_2^2.$$

Now, we are able to write down the primal-dual problem as:

$$\arg\min_{\boldsymbol{v}} \arg\max_{\boldsymbol{y}} \int_0^T \left\| u_t^{k+1} + \nabla_x u^{k+1} \cdot \boldsymbol{v} \right\|_1 + \langle \boldsymbol{v}, K^*\boldsymbol{y} \rangle - \frac{\gamma}{2\beta} \|\boldsymbol{y}\|_2^2 \mathrm{d}t.$$

We arrive at the following scheme:

$$\tilde{\boldsymbol{y}}^{l+1} = \boldsymbol{y}^l + \sigma C \bar{\boldsymbol{v}}^l \tag{5.23}$$

$$\boldsymbol{y}^{l+1} = \arg\min_{\boldsymbol{y}} \left\{ \int_0^T \frac{1}{2} \left\| \boldsymbol{y} - \tilde{\boldsymbol{y}}^{l+1} \right\|_2^2 + \frac{\sigma\gamma}{2\beta} \|\boldsymbol{y}\|_2^2 \mathrm{d}t \right\} \tag{5.24}$$

$$\boldsymbol{v}^{l+1} = \arg\min_{v} \left\{ \int_0^T \frac{1}{2} \left\| \boldsymbol{v} - (\boldsymbol{v}^l - \tau C^* \boldsymbol{y}^{l+1}) \right\|_2^2 + \tau \left\| u_t^k + \nabla_x u^k \cdot \boldsymbol{v} \right\|_1 \mathrm{d}t \right\} \tag{5.25}$$

$$\bar{\boldsymbol{v}}^{l+1} = \boldsymbol{v}^{l+1} + \theta(\boldsymbol{v}^{l+1} - \boldsymbol{v}^l) \tag{5.26}$$

The subproblem in $\boldsymbol{y}$ is a linear L^2-problem. We calculate the optimality condition

$$\boldsymbol{y} - \tilde{\boldsymbol{y}}^{l+1} + \frac{\sigma\gamma}{\beta} \boldsymbol{y} \stackrel{!}{=} 0,$$

and see that the solution is given by

$$\boldsymbol{y}^{l+1} = \frac{\beta}{\beta + \gamma\sigma} \tilde{\boldsymbol{y}}^{l+1}.$$

The optimization problem for $\boldsymbol{v}$ is a vectorial affine linear soft shrinkage (see Lemma

2.2.29). We define

$$\tilde{\boldsymbol{v}} := \boldsymbol{v}^l - \tau C^* \boldsymbol{y}^{l+1}, \quad f(\boldsymbol{v}) := u_t^k + \nabla_x u^k \cdot \boldsymbol{v}, \quad \boldsymbol{\beta} := \nabla_x u^k,$$

the solution is then given by

$$\boldsymbol{v} = \tilde{v} + \begin{cases} \tau\boldsymbol{\beta} & \text{if } f(\tilde{\boldsymbol{v}}) < -\tau \|\boldsymbol{\beta}\|^2 \\ -\tau\boldsymbol{\beta} & \text{if } f(\tilde{\boldsymbol{v}}) > \tau \|\boldsymbol{\beta}\|^2 \\ -\frac{f(\tilde{\boldsymbol{v}})\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|^2} & \text{else} \end{cases}.$$

Combining our efforts, this leads to the following scheme:

$$\begin{aligned} \tilde{\boldsymbol{y}}^{l+1} &= \boldsymbol{y}^l + \sigma C \bar{\boldsymbol{v}}^l \\ \boldsymbol{y}^{l+1} &= \frac{\beta}{\beta + \gamma\sigma} \tilde{\boldsymbol{y}}^{l+1} \\ \tilde{\boldsymbol{v}}^{l+1} &= \boldsymbol{v}^l - \tau C^* \boldsymbol{y}^{l+1} \\ \boldsymbol{v}^{l+1} &= \tilde{v} + \begin{cases} \tau\boldsymbol{\beta} & \text{if } f(\tilde{\boldsymbol{v}}) < -\tau \|\boldsymbol{\beta}\|^2 \\ -\tau\boldsymbol{\beta} & \text{if } f(\tilde{\boldsymbol{v}}) > \tau \|\boldsymbol{\beta}\|^2 \\ -\frac{f(\tilde{\boldsymbol{v}})\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|^2} & \text{else} \end{cases} \\ \bar{\boldsymbol{v}}^{l+1} &= \boldsymbol{v}^{l+1} + \theta(\boldsymbol{v}^{l+1} - \boldsymbol{v}^l) \end{aligned}$$

We are then able to extract the solution of Problem (5.16) as $\boldsymbol{v}^{k+1} = \lim_{l\to\infty} \boldsymbol{v}^l$.

Discretization and Algorithm

We recall the underlying energy:

$$\int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \frac{\beta}{2} \|\nabla_x \boldsymbol{v}\|_2^2 + \gamma \|u_t + \nabla_x u \cdot \boldsymbol{v}\|_1 \, \mathrm{d}t.$$

For the spatial regularization parts $\|\nabla_x u\|_1$ and $\|\nabla_x \boldsymbol{v}\|_2^2$ we use forward differences to discretize the involved operator ∇_x, respectively backward differences for the adjoint operator.

First, we assume the underlying grid to consist of the following set of discrete points:

$$\{(i, j, t) : i = 0, \ldots, n_x, j = 0, \ldots, n_y, t = 0, \ldots, n_t\}$$

The resulting discrete derivatives for v^i and the corresponding dual variable $\boldsymbol{y}$ are calculated as follows:

$$v_x^i(i,j) = \begin{cases} v(i+1,j) - v(i,j) & \text{if } i < n_x \\ 0 & \text{if } i = n_x \end{cases}$$

$$v_y^i(i,j) = \begin{cases} v(i,j+1) - v(i,j) & \text{if } j < n_y \\ 0 & \text{if } j = n_y \end{cases}$$

$$\nabla \cdot \boldsymbol{y}(i,j) = \begin{cases} y_1(i,j) - y_1(i-1,j) & \text{if } i > 0 \\ y_1(i,j) & \text{if } i = 0 \\ -y_1(i-1,j) & \text{if } i = n_x \end{cases}$$

$$+ \begin{cases} y_2(i,j) - y_2(i,j-1) & \text{if } j > 0 \\ y_2(i,j) & \text{if } j = 0 \\ -y_2(i,j-1) & \text{if } j = n_y. \end{cases}$$

The discrete derivatives for the regularizer of u have the same structure.

Besides the discretization for the spatial regularizers $\|\nabla_x u\|_1$ and $\|\nabla_x \boldsymbol{v}\|_2^2$ we have the coupling term $\|u_t + \nabla_x u \cdot \boldsymbol{v}\|_1$, which is the more challenging part. Fixing $\boldsymbol{v}$ as a constant, $u_t + \nabla_x u \cdot \boldsymbol{v}$ has the form of a 2-dimensional transport equation. It can be shown that forward differences are impractical, because they require very small timesteps δ_t. For transport equations we usually use an upwind scheme (see [41]), where depending on the sign of $\boldsymbol{v}$ a forward or backward difference is chosen. This yields a stable discretization for the transport equation.
Taking now the perspective of $\boldsymbol{v}$ with fixed u the coupling term $\|u_t + \nabla_x u \cdot \boldsymbol{v}\|_1$ acts as the data fidelity for flow estimation. Unfortunately, we observed that an upwind scheme for u generates artifacts in the resulting velocity field and therefore, should not be used for motion estimation. A common discretization technique are cell-centered gradients for u, as already proposed by Horn and Schunck in their original paper [43] (see appendix $A.2$ for a detailed discretization scheme). On the one hand they produce nice velocity fields, but on the other hand they do not yield a stable discretization for transport equations and are consequently not suitable for our problem.
A way out is given by a mix of forward and central differences. Using forward differences for the time derivative u_t and central differences for the spatial derivatives $\nabla_x u$ yields a stable discretization of the transport equation and produces reasonable velocity fields.

The adjoint operator then consists of a backward difference for the time derivative and a central difference for the spatial part. We mention here that the central difference is self-adjoint.
For discrete U given on the grid above we get the following discrete derivatives for the forward operator:

$$u_t(i,j,t) = \begin{cases} \frac{u(i,j,t+1)-u(i,j,t)}{\delta_t} & \text{if } t < n_t \\ 0 & \text{if } t = n_t \end{cases}$$

$$u_x(i,j,t) = \begin{cases} \frac{u(i+1,j,t)-u(i-1,j,t)}{2\delta_x} & \text{if } i > 0 \text{ and } i < n_x \\ 0 & \text{if } i = 0 \text{ or } i = n_x \end{cases}$$

$$u_y(i,j,t) = \begin{cases} \frac{u(i,j+1,t)-u(i,j-1,t)}{2\delta_y} & \text{if } j > 0 \text{ and } j < n_y \\ 0 & \text{if } j = 0 \text{ or } j = n_y \end{cases}$$

and for the adjoint operator

$$u_t(i,j,t) = \begin{cases} \frac{u(i,j,t)-u(i,j,t-1)}{\delta_t} & \text{if } t > 0 \\ 0 & \text{if } t = 0 \end{cases}$$

$$u_x(i,j,t) = \begin{cases} \frac{u(i+1,j,t)-u(i-1,j,t)}{2\delta_x} & \text{if } i > 0 \text{ and } i < n_x \\ 0 & \text{if } i = 0 \text{ or } i = n_x \end{cases}$$

$$u_y(i,j,t) = \begin{cases} \frac{u(i,j+1,t)-u(i,j-1,t)}{2\delta_y} & \text{if } j > 0 \text{ and } j < n_y \\ 0 & \text{if } j = 0 \text{ or } j = n_y \end{cases}$$

As a stopping criterion for both minimization subproblems we use, similar to the motion estimation algorithms, the primal-dual residual (see Definition 4.5.2) as a stopping criterion. Since the main algorithm only consists of the alternating minimization with respect to u and $\boldsymbol{v}$, we are unable to use the primal-dual residual here. Instead we measure the difference between two consecutive iterations k and $k+1$ by

$$err_{main} := \frac{\left|u^k - u^{k+1}\right| + \left|\boldsymbol{v}^k - \boldsymbol{v}^{k+1}\right|}{2\left|\Omega\right|}$$

and stop the main iteration if this difference falls below a threshold ϵ.
We end up with the following pseudocode:

Algorithm 9 Joint $TV - L^2$ Optical Flow based Motion Estimation and Image Reconstruction

JointTVL2OpticalFlow$f, \alpha, \beta, \gamma, K$
$v, u \leftarrow 0\ \epsilon < threshold$
$u_{OldM} \leftarrow u$
$\boldsymbol{v}_{OldM} \leftarrow \boldsymbol{v}$
$y, \bar{u} \leftarrow 0\ \epsilon_u < threshold$
$u_{Old} \leftarrow u$
$\tilde{\boldsymbol{y}} \leftarrow \boldsymbol{y} + \sigma C_u \bar{u}$
$y_1 \leftarrow \frac{\tilde{y_1}}{\sigma+1} - \sigma f$
$y_2 \leftarrow \tilde{y}_2 - \sigma S(\frac{\tilde{y}_2}{\sigma}, \frac{\alpha}{\sigma})$
$y_3 \leftarrow solveAffine(\tilde{y}_3, \boldsymbol{v})$
$u \leftarrow u - \tau C_u^* \boldsymbol{y}$
$\bar{u} \leftarrow 2u - u_{Old}$
$y, \bar{\boldsymbol{v}} \leftarrow 0\ \epsilon_v < threshold$
$\boldsymbol{v}_{Old} \leftarrow \boldsymbol{v}$
$\tilde{\boldsymbol{y}} \leftarrow \boldsymbol{y} + \sigma C_v \bar{\boldsymbol{v}}$
$\boldsymbol{y} \leftarrow \frac{\beta}{\beta+\gamma\sigma} \tilde{\boldsymbol{y}}$
$\tilde{\boldsymbol{v}} \leftarrow \boldsymbol{v} - \tau C_v^* \boldsymbol{y}$
$\boldsymbol{v} \leftarrow solveAffine(\tilde{\boldsymbol{v}})$
$\bar{\boldsymbol{v}} \leftarrow 2\boldsymbol{v} - \boldsymbol{v}_{Old}$
$\epsilon \leftarrow \frac{|u-u_{OldM}|+|\boldsymbol{v}-\boldsymbol{v}_{OldM}|}{2|\Omega|}$
v

Realization of the $TV - L^2$ Mass Preservation Model

Primal dual Framework

For the realization of the mass preserving variant we propose a similar strategy as in the optical flow case. The minimization is done alternatingly with respect to u and $\boldsymbol{v}$. Since constant terms can be neglected in the functional for given $u^k, \boldsymbol{v}^k$ this leads to the subproblems

$$u^{k+1} = \arg\min_u \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \gamma \left\| u_t + \nabla_x \cdot (u\boldsymbol{v}^k) \right\|_1 \mathrm{d}t, \tag{5.27}$$

$$\boldsymbol{v}^{k+1} = \arg\min_{\boldsymbol{v}} \int_0^T \left\| u_t^{k+1} + \nabla_x \cdot (u^{k+1}\boldsymbol{v}) \right\|_1 + \frac{\beta}{2\gamma} \|\nabla_x \boldsymbol{v}\|_2^2 \mathrm{d}t. \tag{5.28}$$

Problem in u:

We begin with the subproblem in Equation (5.27). Since every term in the subproblem for u contains an operator, we put the whole functional into F and arrive at

$$F(Cu) = \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \gamma \left\| u_t + \nabla_x \cdot (u\boldsymbol{v}^k) \right\|_1 \mathrm{d}t,$$

with

$$Cu = \begin{pmatrix} K \\ \nabla_x \\ (\partial_t, \nabla_x, I) \end{pmatrix} u$$

and

$$G(u) = 0.$$

The adjoint operator of C is now given by

$$C^*\boldsymbol{y} = K^* y_1 - \nabla_x \cdot y_2 - (\partial_t(y_{3,1}) + \partial_x(y_{3,2}) + \partial_y(y_{3,3}) - I(y_{3,4})).$$

Applying the Chambolle-Pock method gives us a new iteration counter l and the following scheme:

$$\tilde{\boldsymbol{y}}^{l+1} = \boldsymbol{y}^l + \sigma C \bar{u}^l \tag{5.29}$$

$$y_1^{l+1} = \arg\min_y \left\{ \int_0^T \frac{1}{2} \left\| y - \tilde{y_1}^{l+1} \right\|_2^2 + \frac{\sigma}{2} \|y\|_2^2 + \sigma \langle y, f \rangle \, \mathrm{d}t \right\} \tag{5.30}$$

$$y_2^{l+1} = \arg\min_y \left\{ \int_0^T \frac{1}{2} \left\| y - \tilde{y_2}^{l+1} \right\|_2^2 + \sigma\alpha\delta_{B(L^\infty)}(y/\alpha)\,\mathrm{d}t \right\} \tag{5.31}$$

$$y_3^{l+1} = \tilde{y}_3^{l+1} - \sigma \arg\min_y \left\{ \int_0^T \frac{1}{2} \left\| y - \frac{\tilde{y}_3^{l+1}}{\sigma} \right\|_2^2 + \frac{\gamma}{\sigma} \left\| y \cdot (1, v_1, v_2, \nabla_x \cdot \boldsymbol{v}) \right\|_1 \mathrm{d}t \right\} \tag{5.32}$$

$$u^{l+1} = \arg\min_u \left\{ \int_0^T \frac{1}{2} \left\| u - (u^l - \tau C^* \boldsymbol{y}^{l+1}) \right\|_2^2 \mathrm{d}t \right\} \tag{5.33}$$

$$\bar{u}^{l+1} = u^{l+1} + \theta(u^{l+1} - u^l) \tag{5.34}$$

Let us observe that this yields nearly the same scheme as in the $TV - L^2$ optical flow model from the previous section. Consequently, we inherit all solutions and finally obtain the following fairly simple iterations:

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{l+1} &= \boldsymbol{y}^l + \sigma C \bar{u}^l \\
y_1^{l+1} &= \frac{\tilde{y}_1^{l+1} - \sigma f}{\sigma + 1} \\
y_2^{l+1} &= \pi_\alpha(\tilde{y_2}^{l+1}) \\
y_3^{l+1} &= \begin{cases} -\gamma\boldsymbol{\beta} & \text{if } f(\frac{\tilde{y}_3^{l+1}}{\sigma}) < -\frac{\gamma}{\sigma} \|\boldsymbol{\beta}\|^2 \\ \gamma\boldsymbol{\beta} & \text{if } f(\frac{\tilde{y}_3^{l+1}}{\sigma}) > \frac{\gamma}{\sigma} \|\boldsymbol{\beta}\|^2 \\ \frac{\sigma f(\frac{\tilde{y}_3^{l+1}}{\sigma})\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|^2} & \text{else} \end{cases} \\
u^{l+1} &= u^l - \tau C^* \boldsymbol{y}^{l+1} \\
\bar{u}^{l+1} &= u^{l+1} + \theta(u^{l+1} - u^l).
\end{aligned}$$

The solution for (5.27) can be extracted as $u^{k+1} = \lim_{l\to\infty} u^l$.

Problem in $\boldsymbol{v}$:

For $\boldsymbol{v}^{k+1}$ we have to solve

$$\arg\min_{\boldsymbol{v}} \int_0^T \left\| u_t^{k+1} + \nabla_x \cdot (u^{k+1}\boldsymbol{v}) \right\|_1 + \frac{\beta}{2\gamma} \|\nabla_x \boldsymbol{v}\|_2^2 \,\mathrm{d}t, .$$

Speaking in the context of pimal-dual problems we shift the whole functional into the dual part F, since both terms contain operators applied to $\boldsymbol{v}$. Consequently,

$$F(C\boldsymbol{v}) := \int_0^T \left\| u_t^{k+1} + \nabla_x \cdot (u^{k+1}\boldsymbol{v}) \right\|_1 + \frac{\beta}{2\gamma} \|\nabla_x \boldsymbol{v}\|_2^2 \,\mathrm{d}t,$$

with the linear operator C given by

$$C\boldsymbol{v} = \begin{pmatrix} \nabla_x & 0 \\ 0 & \nabla_x \\ \partial_x u^{k+1} & \partial_y u^{k+1} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}.$$

We want to mention again that $(\partial_x u)(v_1) := \partial_x(v_1 u)$. Calculating the adjoint operator to C gives:

$$C^*\boldsymbol{y} = -\begin{pmatrix} \nabla_x \cdot & 0 & u^{k+1}\partial_x \\ 0 & \nabla_x \cdot & u^{k+1}\partial_y \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}.$$

Now, we get for the dual F^*:

$$F^*(\boldsymbol{y}) := \int_0^T \frac{\gamma}{2\beta} \left\|y_{1,2}\right\|_2^2 + \delta_{B(L^\infty)}(y_3) - \langle u_t^{k+1}, y_3 \rangle \, \mathrm{d}t.$$

This yields the primal-dual formulation:

$$\arg\min_{\boldsymbol{v}} \arg\max_{\boldsymbol{y}=(y_1,y_2,y_3)} \int_0^T \langle \boldsymbol{v}, C^*\boldsymbol{y} \rangle - \frac{\gamma}{2\beta} \left\|y_{1,2}\right\|_2^2 - \delta_{B(L^\infty)}(y_3) + \langle u_t^{k+1}, y_3 \rangle \, \mathrm{d}t.$$

Incorporating the primal-dual formulation into the Chambolle-Pock methods gives us:

$$\tilde{\boldsymbol{y}}^{l+1} = \boldsymbol{y}^l + \sigma C \bar{\boldsymbol{v}} \tag{5.35}$$

$$y_{1,2}^{l+1} = \arg\min_y \left\{ \int_0^T \frac{1}{2} \left\| y - \tilde{y}_{1,2}^{l+1} \right\|_2^2 + \frac{\sigma\gamma}{2\beta} \left\|y\right\|_2^2 \mathrm{d}t \right\} \tag{5.36}$$

$$y_3^{l+1} = \arg\min_y \left\{ \int_0^T \frac{1}{2} \left\| \boldsymbol{y} - \tilde{y}_3^{l+1} \right\|_2^2 + \delta_{B(L^\infty)}(y_3) - \langle u_t^{k+1}, y \rangle \, \mathrm{d}t \right\} \tag{5.37}$$

$$\boldsymbol{v}^{l+1} = \arg\min_v \left\{ \int_0^T \frac{1}{2} \left\| \boldsymbol{v} - (\boldsymbol{v}^l - \tau C^*\boldsymbol{y}) \right\|_2^2 \right\} \tag{5.38}$$

$$\bar{\boldsymbol{v}}^{l+1} = \boldsymbol{v}^{l+1} + \theta(\boldsymbol{v}^{l+1} - \boldsymbol{v}^l) \tag{5.39}$$

The subproblems for $y_{1,2}$ are linear and their solution can directly be calculated as

$$y_{1,2}^{l+1} = \frac{\beta}{\beta + \sigma\gamma} \tilde{y}_{1,2}^{l+1}.$$

For the third proximity problem we refer to Section 4.5.4 and get as a solution the point-wise projection of $\tilde{y_3}^{l+1}$ onto $[-1, 1]$:

$$y_3^{l+1} = \pi_1(\tilde{y_3}^{l+1}).$$

We end up with the following scheme:

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{l+1} &= \boldsymbol{y}^l + \sigma C\bar{\boldsymbol{v}}, \\
y_{1,2}^{l+1} &= \frac{\beta}{\beta + \sigma\gamma}\tilde{y}_{1,2}^{l+1}, \\
y_3^{l+1} &= \pi_1(\tilde{y_3}^{l+1}), \\
\boldsymbol{v}^{l+1} &= \boldsymbol{v}^l - \tau C^*\boldsymbol{y}, \\
\bar{\boldsymbol{v}}^{l+1} &= \boldsymbol{v}^{l+1} + \theta(\boldsymbol{v}^{l+1} - \boldsymbol{v}^l).
\end{aligned}$$

As before, we extract the solution for Equation (5.28) as $\boldsymbol{v}^{k+1} = \lim_{l\to\infty} \boldsymbol{v}^l$.

Discretization and Algorithm

For the discretization of the underlying operators, a similar strategy as for the L^2-TV optical flow model can be used for the $TV - L^2$ mass preservation energy

$$\int_0^T \frac{1}{2}\left\|Ku - f\right\|_2^2 + \alpha\left\|\nabla_x u\right\|_1 + \frac{\beta}{2}\left\|\nabla_x \boldsymbol{v}\right\|_2^2 + \gamma\left\|u_t + \nabla_x \cdot (u\boldsymbol{v})\right\|_1 \mathrm{d}t.$$

The spatial regularization terms $\|\nabla_x u\|_1$ and $\|\nabla_x \boldsymbol{v}\|_2^2$ are discretized with forward differences, which have been defined in the previous discretization section. The continuity equation $u_t + \nabla_x \cdot (u\boldsymbol{v})$ also represents a transport equation and we use the same combination of forward and central differences to discretize the involved derivatives. We refer to Section 5.2.3 for details. This leads to the following pseudocode:

5.3. Simultaneous $TV - TV$ Image Reconstruction and Motion Estimation

5.3.1. Definition and Motivation

A second model for joint motion estimation and image reconstruction is motivated by the fact that motion usually occurs as movement of whole objects. The underlying velocity field contains regions of constant movement with jumps between object and

Algorithm 10 Joint $TV - L^2$ Mass Preservation based Motion Estimation and Image Reconstruction

JointTVL2MassPreservation $f, \alpha, \beta, \gamma, K$

$v, u \leftarrow 0$ $\epsilon < threshold$
$u_{OldM} \leftarrow u$
$\boldsymbol{v}_{OldM} \leftarrow \boldsymbol{v}$
$y, \bar{u} \leftarrow 0$ $\epsilon_u < threshold$
$u_{Old} \leftarrow u$
$\tilde{\boldsymbol{y}} \leftarrow \boldsymbol{y} + \sigma C_u \bar{u}$
$y_1 \leftarrow \frac{\tilde{y_1}}{\sigma+1} - \sigma f$
$y_{2,3} \leftarrow \tilde{y}_{2,3} - \sigma S(\frac{\tilde{y}_{2,3}}{\sigma}, \frac{\alpha}{\sigma})$
$u \leftarrow u - \tau C_u^* \boldsymbol{y}$
$\bar{u} \leftarrow 2u - u_{Old}$
$y, \bar{\boldsymbol{v}} \leftarrow 0$ $\epsilon_v < threshold$
$\boldsymbol{v}_{Old} \leftarrow \boldsymbol{v}$
$\tilde{\boldsymbol{y}} \leftarrow \boldsymbol{y} + \sigma C_v \bar{\boldsymbol{v}}$
$y_{1,2} \leftarrow \frac{\beta}{\beta+\gamma\sigma} \tilde{y_{1,2}}$
$y_3 \leftarrow solveAffine(\tilde{y}_3, u)$
$\boldsymbol{v} \leftarrow \boldsymbol{v} - \tau C_v^* \boldsymbol{y}$
$\bar{\boldsymbol{v}} \leftarrow 2\boldsymbol{v} - \boldsymbol{v}_{Old}$
$\epsilon \leftarrow \frac{|u-u_{OldM}|+|\boldsymbol{v}-\boldsymbol{v}_{OldM}|}{2|\Omega|}$
v

background. This property favors the use of a total variation regularization for the motion field $\boldsymbol{v}$. For the image data u, we inherit the TV-regularization from the previous chapter. Similar to the TV-L^2 models we discuss a mass preservation and an optical flow model.

Model 5.3.1. Mass-preserving $TV - TV$ model

$$\begin{aligned} \arg\min_{u,\boldsymbol{v}} \int_0^T \frac{1}{2} \|Ku - f\|_2^2 \, \mathrm{d}t + \alpha \int_0^T \left|u(\cdot,t)\right|_{BV} \mathrm{d}t + \beta \int_0^T \left|\boldsymbol{v}(\cdot,t)\right|_{BV} \mathrm{d}t \qquad (5.40) \\ \text{s.t.} \quad \partial_t u + \nabla \cdot (u\boldsymbol{v}) = 0 \quad \text{ in } \mathcal{D}'([0,T] \times \Omega), \end{aligned}$$

Model 5.3.2. Optical flow $TV - TV$ model

$$\arg\min_{u,\boldsymbol{v}} \int_0^T \frac{1}{2} \|Ku - f\|_2^2 \, \mathrm{d}t + \alpha \int_0^T \left|u(\cdot,t)\right|_{BV} \mathrm{d}t + \beta \int_0^T \left|\boldsymbol{v}(\cdot,t)\right|_{BV} \mathrm{d}t \qquad (5.41)$$

$$\text{s.t.} \quad \partial_t u + \nabla u \cdot \boldsymbol{v} = 0 \quad \text{in } \mathcal{D}'([0,T] \times \Omega).$$

Similar to the previous chapter we assume corrupted image data f with additive Gaussian noise. Furthermore, the image u and the velocity field $\boldsymbol{v}$ are now regularized with the total variation. Mass preservation (Section 4.3) resp. optical flow constraint (Section 4.2) act as a connection between image data and velocity field.

5.3.2. Existence and Uniqueness

From the analytical point of view the analysis of the TV-TV model is slightly different from the TV-L^2 model. For the sake of universality we will analyze a more general energy. For the following, let $1 < p, q \in \mathbb{R}$. Then the general TV-TV energy is defined as

$$J(u, \boldsymbol{v}) = \int_0^T \frac{1}{2} \|Ku - f\|_2^2 \, \mathrm{d}t + \alpha \int_0^T \left|u(\cdot,t)\right|_{BV}^p \mathrm{d}t + \beta \int_0^T \left|\boldsymbol{v}(\cdot,t)\right|_{BV}^q \mathrm{d}t. \tag{5.42}$$

Existence

The first property to prove is again compactness of sublevel sets with respect to some topology τ, which can be shown similar to the previous chapter (see Lemma 5.2.3).

Lemma 5.3.3. Compactness of sub levelsets of the general TV-TV model
Let

$$u \in L^p(0,T;BV(\Omega)), \quad v \in L^q(0,T;BV(\Omega)), \quad J(u,\boldsymbol{v}) \leq \alpha$$

and furthermore

$$\|\boldsymbol{v}\|_\infty \leq c_v < \infty \quad \text{in } \Omega \times [0,T] \text{ and } K\mathbf{1}_t \neq 0 \quad \forall t \in [0,T].$$

Now, for $(u, \boldsymbol{v})$ in the set

$$\mathcal{S}(c) = \left\{(u,\boldsymbol{v}) \in L^{\hat{p}}(0,T;BV(\Omega)) \times L^q(0,T;BV(\Omega)) : J(u,\boldsymbol{v}) \leq c\right\}$$

we have

$$\|u\|_{L^{\hat{p}}(0,T;BV(\Omega))} \leq c, \quad \|v\|_{L^q(0,T;BV(\Omega))} \leq c \quad \text{with } \hat{p} = \min\{p,2\}.$$

Consequently, $\mathcal{S}(c)$ is not empty and compact in the weak-* topology.

Proof. Since the general TV-TV energy does not differ from the TV-L^2 energy in terms of u, the compactness can be shown similar to Lemma 5.2.3.
From the general energy (5.42) we can directly deduce an a-priori estimate for $\boldsymbol{v}$:

$$\int_0^T \left|\boldsymbol{v}(\cdot,t)\right|_{BV(\Omega)}^q \mathrm{d}t \leq \alpha$$

Now some simple calculations give an adjusted estimate for $\boldsymbol{v}$:

$$\begin{aligned}\|\boldsymbol{v}\|_{L^q(0,T;BV(\Omega))} \leq \|\boldsymbol{v}\|_{L^q(0,T;BV(\Omega))}^q &= \int_0^T \left\|\boldsymbol{v}(\cdot,t)\right\|_{BV(\Omega)}^q \mathrm{d}t \\ &\leq \int_0^T \left\|\boldsymbol{v}(\cdot,t)\right\|_{L^1(\Omega)}^q \mathrm{d}t + \int_0^T \left|\boldsymbol{v}(\cdot,t)\right|_{BV(\Omega)}^q \mathrm{d}t \\ &\leq \int_0^T c_{\boldsymbol{v}}^q \left|\Omega\right| \mathrm{d}t + \alpha = c_{\boldsymbol{v}}^q \left|\Omega\right| T + \alpha\end{aligned}$$

Combining the bounds for u and $\boldsymbol{v}$, an application of the Banach-Alaoglu Lemma 2.2.16 yields the required compactness result in the weak-* topology. □

Lemma 5.3.4. Lower semicontinuity of the general TV-TV model
The general TV-TV energy (5.42) is weak-* lower semicontinuous.

Proof. Semicontinuity of the first two terms directly follows from the lower semicontinuity of the $TV - L^2$ model (Lemma 5.2.4). Since the third term is simply another BV-term, the same argumentation holds. □

Again, the crucial part in the analysis is to prove convergence of the constraint. We use the same strategy as in the analysis of the TV-L^2 model and recall the necessary steps.

Lemma 5.3.5. Bound for $\partial_t u$
Let $\Omega \subset \mathbb{R}^2, u \in L^p(0,T;BV(\Omega))$ and $\boldsymbol{v} \in L^q(0,T;BV(\Omega))$. Let furthermore

$$\|\boldsymbol{v}\|_\infty \leq c_v < \infty \quad \text{a.e. in } \Omega \times [0,T]\,.$$

1. **Mass Preservation:**
   Then for $u, \boldsymbol{v}$ solving the continuity equation

$$\partial_t u + \nabla \cdot (u\boldsymbol{v}) = 0$$

we have

$$\partial_t u \in L^{\frac{p^* t}{p^* t - 1}}(0, T; W^{-1,2}(\Omega)).$$

2. **Optical Flow:**
   Let furthermore $\nabla \cdot \boldsymbol{v} \in L^{p^* s}(0, T; L^{2k}(\Omega))$ with $\frac{1}{k} + \frac{1}{k^*} = 1, k > 1$, $\frac{1}{s} + \frac{1}{t} = 1$ and let $u, \boldsymbol{v}$ solve the optical flow equation

$$\partial_t u + \nabla u \cdot \boldsymbol{v} = 0.$$

   Then we have:

$$\partial_t u \in L^{p^* t}(0, T; L^{\frac{2k^*}{2k^* - 1}}(\Omega)).$$

Proof. The proof can be done similar to the proof for the $TV - L^2$ model (Lemma 5.2.5), since the argumentation relies on the a-priori bounds for $\boldsymbol{v}$ and $\nabla \cdot \boldsymbol{v}$. Thus, for $\boldsymbol{v} \in L^q(0, T; BV(\Omega))$ we obtain the same boundedness result. □

Having deduced bounds for ∂_t, we are now able to apply the Aubin-Lions Lemma 2.1.16 with the following theorem:

Theorem 5.3.6. Compact embedding for u
Let be $\Omega \subset \mathbb{R}^2, u \in L^p(0, T; BV(\Omega))$ and $\boldsymbol{v} \in L^q(0, T; BV(\Omega))$. Let furthermore be

$$\|\boldsymbol{v}\|_\infty \leq c_v < \infty \quad \text{a.e. in } \Omega \times [0, T].$$

1. **Mass Preservation:**
   The set

$$\left\{u : u \in L^p(0, T; BV(\Omega)), u_t + \nabla \cdot (u\boldsymbol{v}) = 0\right\}$$

   can be compactly embedded into

$$L^p(0, T; L^r(\Omega)), \quad \text{for } 1 < r < 2.$$

2. **Optical Flow:**
   Let furthermore $\nabla \cdot \boldsymbol{v} \in L^{p^* s}(0, T; L^{2k}(\Omega))$ with $\frac{1}{k} + \frac{1}{k^*} = 1 = \frac{1}{p} + \frac{1}{p^*}$ and $s, k^* > 1$.

Then the set

$$\left\{u : u \in L^p(0,T;BV(\Omega)), u_t + \nabla u \cdot \boldsymbol{v} = 0\right\}$$

can be compactly embedded into

$$L^p(0,T;L^r(\Omega)), \qquad \text{for } \frac{2k^*}{2k^*-1} \leq r < 2.$$

Proof. The proof can be done similar to the proof of Theorem 5.2.6. □

Lemma 5.3.7. Convergence of the constraint
Let $\Omega \subset \mathbb{R}^2, u_k \in L^p(0,T;BV(\Omega))$ and $\boldsymbol{v}_k \in L^q(0,T;BV(\Omega))$. Let furthermore

$$\|\boldsymbol{v}\|_\infty \leq c_v < \infty \quad \text{ a.e. in } \Omega \times [0,T]\,.$$

1. **Mass Preservation:**
   Then

   $$\partial_t u_k + \nabla \cdot (u_k \boldsymbol{v_k}) \rightharpoonup \partial_t u + \nabla \cdot (u\boldsymbol{v})$$

   in sense of distributions.

2. **Optical Flow:**
   Let furthermore $\nabla \cdot \boldsymbol{v} \in L^{p^* s}(0,T;L^{2k}(\Omega))$ with $k, s > 1$. Then

   $$\partial_t u_k + \nabla u_k \cdot \boldsymbol{v_k} \rightharpoonup \partial_t u + \nabla u \cdot \boldsymbol{v}$$

   in the sense of distributions.

Proof. We will just follow the proof of Lemma 5.2.7 and will argue in a different way where necessary.
For the time derivative $\partial_t u$ no changes are needed for the proof and we directly proceed to the
Mass preservation:
Writing down the weak form and doing some calculations led us in Lemma 5.2.7 to the

equation

$$-\int_0^T \int_\Omega \big(\nabla \cdot (u_k \boldsymbol{v}_k) - \nabla \cdot (u\boldsymbol{v})\big)\,\varphi \,\mathrm{d}x\,\mathrm{d}t \leq \underbrace{C\,\|u_k - u\|_{L^p(0,T;L^r)}}_{(i)} + \underbrace{\int_0^T \int_\Omega u(\boldsymbol{v}_k - \boldsymbol{v}) \cdot \nabla\varphi \,\mathrm{d}x\,\mathrm{d}t}_{(ii)}$$

This estimate can be derived in the same way here. Part (i) has already been discussed. For part (ii) we can no longer use the $W^{1,2}$ embedding, since $\boldsymbol{v}$ is now a BV function. But for $\boldsymbol{v} \in L^\infty(\Omega)$ we have the useful weak-* convergence result (see Remark 4 for details). Due to the continuous embedding of BV into L^1 (see (2.1.22)) we have $u \in L^1$ and consequently $u\nabla\varphi \in L^1$. Thus,

$$\int_0^T \int_\Omega \boldsymbol{v}_k \cdot (u\nabla\varphi)\,\mathrm{d}x\,\mathrm{d}t \to \int_0^T \int_\Omega \boldsymbol{v} \cdot (u\nabla\varphi)\,\mathrm{d}x\,\mathrm{d}t$$

and part (ii) converges to zero. We conclude

$$\lim_{k\to\infty} \left| \int_0^T \int_\Omega u_k v_k - uv \,\mathrm{d}x\,\mathrm{d}t \right| \leq \lim_{k\to\infty} C\,\|u_k - u\|_{L^p(0,T;L^r)} + \left| \int_0^T \int_\Omega u(v_k - v) \cdot \nabla\varphi \,\mathrm{d}x\,\mathrm{d}t \right| = 0.$$

Optical flow: In the proof for the $TV - L^2$ model, the weak form of the optical flow constraint led us to

$$-\int_0^T \int_\Omega (\nabla u_k \cdot \boldsymbol{v}_k - \nabla u \cdot \boldsymbol{v})\,\varphi \,\mathrm{d}x\,\mathrm{d}t \leq \underbrace{\int_0^T \int_\Omega (u_k - u)\nabla \cdot (\varphi v_k)\,\mathrm{d}x\,\mathrm{d}t}_{(i)} + \underbrace{\int_0^T \int_\Omega u\nabla \cdot (\varphi(v_k - v))\,\mathrm{d}x\,\mathrm{d}t}_{(ii)}$$

The argumentation for part (i) holds, because we only used the boundedness of $\boldsymbol{v}$ resp. $\nabla \cdot \boldsymbol{v}$.

Estimating part (ii) can be done in a similar way. First, we write down the divergence using the product rule

$$\int_0^T \int_\Omega u\nabla \cdot (\varphi(v_k - v))\,\mathrm{d}x\,\mathrm{d}t = \int_0^T \int_\Omega \underbrace{u\varphi\nabla \cdot (v_k - v)}_{(ii).1} + \underbrace{u(v_k - v) \cdot \nabla\varphi}_{(ii.2)}\,\mathrm{d}x\,\mathrm{d}t,$$

and we see that the arguments from Lemma 5.2.7 still hold. Let us finally summarize

our estimates:

$$\begin{aligned}
\lim_{k\to\infty}\left|\int_0^T\int_\Omega (\nabla u_k\cdot \boldsymbol{v}_k - \nabla u\cdot \boldsymbol{v})\,\varphi\,\mathrm{d}x\,\mathrm{d}t\right| &\le \lim_{k\to\infty}\left|\int_0^T\int_\Omega (u_k-u)\nabla\cdot(\varphi v_k)\,\mathrm{d}x\,\mathrm{d}t\right| \\
&\quad + \lim_{k\to\infty}\left|\int_0^T\int_\Omega u\nabla\cdot(\varphi(v_k-v))\,\mathrm{d}x\,\mathrm{d}t\right| \\
&\le C\lim_{k\to\infty}\|u_k-u\|_{L^p(0,T;L^r)} \\
&\quad + \lim_{k\to\infty}\left|\int_0^T\int_\Omega u\varphi\nabla\cdot(v_k-v)\,\mathrm{d}x\,\mathrm{d}t\right| \\
&\quad + \lim_{k\to\infty}\left|\int_0^T\int_\Omega u(v_k-v)\cdot\nabla\varphi\,\mathrm{d}x\,\mathrm{d}t\right| \\
&= 0
\end{aligned}$$

□

With only small adjustments to the proof of the TV-L^2 model, we have proven all necessary requirements for the existence of a minimizer for the joint TV-TV image reconstruction and motion estimation model (5.40), (5.41) equipped with general energy (5.42).

Theorem 5.3.8. Existence of a minimizer for the $TV - TV$ model
Let $\Omega \subset \mathbb{R}^2, p, q > 1, \hat{p} = \min\{p, 2\}$ and consider the $TV - TV$ energy

$$J(u,\boldsymbol{v}) = \int_0^T \frac{1}{2}\|Ku-f\|_2^2\,\mathrm{d}t + \alpha\int_0^T \left|u(\cdot,t)\right|_{BV}\mathrm{d}t + \frac{\beta}{2}\int_0^T \left|\nabla\boldsymbol{v}(\cdot,t)\right|_{BV}\mathrm{d}t.$$

Let furthermore be

$$\|\boldsymbol{v}\|_\infty \le c_v < \infty \quad \text{a.e. in } \Omega\times[0,T] \text{ and } K\mathbf{1}_t \neq 0 \quad \forall t\in[0,T]\,.$$

Mass Preservation:
Then there exists a minimizer $(\tilde{u},\tilde{\boldsymbol{v}})$ in the set

$$\left\{(u,\boldsymbol{v}) : u\in L^{\hat{p}}(0,T;BV(\Omega)), \boldsymbol{v}\in L^q(0,T;BV(\Omega)), \partial_t u + \nabla\cdot(u\boldsymbol{v}) = 0\right\}.$$

Optical Flow:
Let furthermore $s, k > 1$ and p^* such that $\frac{1}{p}+\frac{1}{p^*} = 1$. Then there also exists a minimizer

in the set

$$\left\{(u,\boldsymbol{v}) : u \in L^{\hat{p}}(0,T;BV(\Omega)), \boldsymbol{v} \in L^{q}(0,T;BV(\Omega)), \nabla\cdot\boldsymbol{v} \in L^{p^*s}(0,T;L^{2k}(\Omega)), \partial_t u + \nabla\cdot(u\boldsymbol{v}) = 0\right\}.$$

Proof. Can be done in analogy to Theorem 5.3.8. □

Uniqueness

The energy for the mass preserving TV-TV model and the optical flow TV-TV model is convex. However, we are not able to prove uniqueness due to non-linearity of optical flow and continuity equation.

5.3.3. Numerical Realization

Realization of the $TV - TV$ Optical Flow Model

Primal-dual Framework

For the realization of the $TV - TV$ optical flow model, we propose a strategy similar to the $TV - L^2$ optical flow model (see Section 5.2.3). We recall the important steps and go into detail where changes are necessary.

The objective functional

$$\arg\min_{u,\boldsymbol{v}} \int_0^T \frac{1}{2}\|Ku - f\|_2^2 + \alpha\|\nabla_x u\|_1 + \beta\|\nabla_x \boldsymbol{v}\|_1 + \gamma\|u_t + \nabla_x u\cdot\boldsymbol{v}\|_1 \,\mathrm{d}t, \tag{5.43}$$

can be minimized with an alternating scheme. For given $u^k, \boldsymbol{v}^k$ we consider the following subproblems:

$$u^{k+1} = \arg\min_{u} \int_0^T \frac{1}{2}\|Ku - f\|_2^2 + \alpha\|\nabla_x u\|_1 + \gamma\left\|u_t + \nabla_x u\cdot\boldsymbol{v}^k\right\|_1 \mathrm{d}t \tag{5.44}$$

$$\boldsymbol{v}^{k+1} = \arg\min_{\boldsymbol{v}} \int_0^T \left\|u_t^{k+1} + \nabla_x u^{k+1}\cdot\boldsymbol{v}\right\|_1 + \frac{\beta}{\gamma}\|\nabla_x \boldsymbol{v}\|_1 \,\mathrm{d}t. \tag{5.45}$$

Problem in u:
It is directly clear that, since we only changed the regularization for $\boldsymbol{v}$, the subproblem for u (Equation 5.44) is the same as in the $TV - L^2$ case. Consequently we can inherit

the following iteration scheme and extract u^{k+1} as $\lim_{l\to\infty} u^l$:

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{l+1} &= \boldsymbol{y}^l + \sigma C\bar{u}^l \\
y_1^{l+1} &= \frac{\tilde{y}_1^{l+1} - \sigma f}{\sigma + 1} \\
y_2^{l+1} &= \pi_\alpha(\tilde{y_2}^{l+1}) \\
y_3^{l+1} &= \pi_\gamma(\tilde{y_3}^{l+1}), \\
u^{l+1} &= u^l - \tau C^* \boldsymbol{y}^{l+1} \\
\bar{u}^{l+1} &= u^{l+1} + \theta(u^{l+1} - u^l).
\end{aligned}$$

All necessary notations and definitions can be found within the $TV - L^2$ realization in Section 5.2.3.

Problem in v:
The subproblem in $\boldsymbol{v}$ (Equation 5.45) is exactly an $L^1 - TV$ optical flow problem. We refer to Section 4.5.6 for details and simply denote the final iteration scheme as:

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{l+1} &= \boldsymbol{y}^l + \sigma C\bar{\boldsymbol{v}}^l \\
\boldsymbol{y}^{l+1} &= \pi_{\frac{\beta}{\gamma}}(\tilde{\boldsymbol{y}}^{l+1}) \\
\tilde{\boldsymbol{v}}^{l+1} &= \boldsymbol{v}^l - \tau C^* \boldsymbol{y}^{l+1} \\
\boldsymbol{v}^{l+1} &= \tilde{v} + \begin{cases} \tau\boldsymbol{\beta} & \text{if } f(\tilde{\boldsymbol{v}}^{l+1}) < -\tau \left\|\boldsymbol{\beta}\right\|^2 \\ -\tau\boldsymbol{\beta} & \text{if } f(\tilde{\boldsymbol{v}}^{l+1}) > \tau \left\|\boldsymbol{\beta}\right\|^2 \\ -\frac{f(\tilde{\boldsymbol{v}}^{l+1})\boldsymbol{\beta}}{\left\|\boldsymbol{\beta}\right\|^2} & \text{else} \end{cases} \\
\bar{\boldsymbol{v}}^{l+1} &= \boldsymbol{v}^{l+1} + \theta(\boldsymbol{v}^{l+1} - \boldsymbol{v}^l).
\end{aligned}$$

The solution can be extracted as $\boldsymbol{v}^{k+1} = \lim_{l\to\infty} \boldsymbol{v}^l$.

Discretization and Algorithm

The difference of the $TV - L^2$ and the $TV - TV$ model lies in the regularization for the velocity field. Consequently, we are able to inherit a similar discretization strategy for the operators (see Section 5.2.3) and end up with the following algorithm:

Algorithm 11 Joint $TV - TV$ Optical Flow based Motion Estimation and Image Reconstruction

JointTVTVOpticalFlow$f, \alpha, \beta, \gamma, K$

$v, u \leftarrow 0$ $\epsilon < threshold$
$u_{OldM} \leftarrow u$
$\boldsymbol{v}_{OldM} \leftarrow \boldsymbol{v}$
$y, \bar{u} \leftarrow 0$ $\epsilon_u < threshold$
$u_{Old} \leftarrow u$
$\tilde{\boldsymbol{y}} \leftarrow \boldsymbol{y} + \sigma C_u \bar{u}$
$y_1 \leftarrow \frac{\tilde{y_1}}{\sigma+1} - \sigma f$
$y_2 \leftarrow \tilde{y}_2 - \sigma S(\frac{\tilde{y}_2}{\sigma}, \frac{\alpha}{\sigma})$
$y_3 \leftarrow solveAffine(\tilde{y}_3, \boldsymbol{v})$
$u \leftarrow u - \tau C_u^* \boldsymbol{y}$
$\bar{u} \leftarrow 2u - u_{Old}$
$y, \bar{\boldsymbol{v}} \leftarrow 0$ $\epsilon_v < threshold$
$\boldsymbol{v}_{Old} \leftarrow \boldsymbol{v}$
$\tilde{\boldsymbol{y}} \leftarrow \boldsymbol{y} + \sigma C_v \bar{\boldsymbol{v}}$
$\boldsymbol{y} \leftarrow y - \sigma S(\frac{\tilde{\boldsymbol{y}}}{\sigma}, \frac{\beta}{\gamma\sigma})$
$\tilde{\boldsymbol{v}} \leftarrow \boldsymbol{v} - \tau C_v^* \boldsymbol{y}$
$\boldsymbol{v} \leftarrow solveAffine(\tilde{\boldsymbol{v}})$
$\bar{\boldsymbol{v}} \leftarrow 2\boldsymbol{v} - \boldsymbol{v}_{Old}$
$\epsilon \leftarrow \frac{|u - u_{OldM}| + |\boldsymbol{v} - \boldsymbol{v}_{OldM}|}{2|\Omega|}$
v

Realization of the $TV - TV$ Mass Preservation Model

Primal-dual Framework

As before, we recall the objective functional

$$\arg\min_{u,\boldsymbol{v}=(v_1,v_2)} \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \beta \|\nabla_x \boldsymbol{v}\|_1 + \gamma \left\|u_t + \nabla_x \cdot (u\boldsymbol{v})\right\|_1 \mathrm{d}t.$$

We propose an alternating minimization scheme for given $u^k, \boldsymbol{v}^k$ and get the following subproblems:

$$u^{k+1} = \arg\min_u \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \gamma \left\|u_t + \nabla_x \cdot (u\boldsymbol{v}^k)\right\|_1 \mathrm{d}t$$
$$\boldsymbol{v}^{k+1} = \arg\min_{\boldsymbol{v}=(v_1,v_2)} \int_0^T \left\|u_t^{k+1} + \nabla_x \cdot (u^{k+1}\boldsymbol{v})\right\|_1 + \frac{\beta}{\gamma} \|\nabla_x \boldsymbol{v}\|_1 \mathrm{d}t$$

Problem in u:

A scheme for the subproblem in u has been deduced in the joint $TV - L^2$ Section (see Section 5.2.3).

Problem in v:

For the subproblem in $\boldsymbol{v}$, we simply have the $L^1 - TV$ mass preservation scheme and refer to Section 4.5.7 for details. Final iterations are given as

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{l+1} &= \boldsymbol{y}^l + \sigma C \bar{\boldsymbol{v}}^l \\
y_{1,2}^{l+1} &= \pi_{\frac{\beta}{\gamma}}(\tilde{y}_{1,2}^{l+1}) \\
y_3^{l+1} &= \pi_1(\tilde{y}_3^{l+1} + \sigma u_t^{k+1}) \\
\boldsymbol{v}^{l+1} &= \boldsymbol{v}^l - \tau C^* \boldsymbol{y}^{l+1} \\
\bar{\boldsymbol{v}}^{l+1} &= \boldsymbol{v}^{l+1} + \theta(\boldsymbol{v}^{l+1} - \boldsymbol{v}^l).
\end{aligned}$$

Discretization and Algorithm

Again, the difference to the $TV - L^2$ joint mass preservation model lies in the regularization term for the velocity field $\boldsymbol{v}$ and we inherit the discretization scheme from Section 5.2.3. Slight changes to the known algorithm yield

Algorithm 12 Joint $TV-TV$ Mass Preservation based Motion Estimation and Image Reconstruction

JointTVL2MassPreservation $f, \alpha, \beta, \gamma, K$

$v, u \leftarrow 0$ $\epsilon < threshold$
$u_{OldM} \leftarrow u$
$\boldsymbol{v}_{OldM} \leftarrow \boldsymbol{v}$
$y, \bar{u} \leftarrow 0$ $\epsilon_u < threshold$
$u_{Old} \leftarrow u$
$\tilde{\boldsymbol{y}} \leftarrow \boldsymbol{y} + \sigma C_u \bar{u}$
$y_1 \leftarrow \frac{\tilde{y_1}}{\sigma+1} - \sigma f$
$y_{2,3} \leftarrow \tilde{y}_{2,3} - \sigma S(\frac{\tilde{y}_{2,3}}{\sigma}, \frac{\alpha}{\sigma})$
$u \leftarrow u - \tau C_u^* \boldsymbol{y}$
$\bar{u} \leftarrow 2u - u_{Old}$
$y, \bar{\boldsymbol{v}} \leftarrow 0$ $\epsilon_v < threshold$
$\boldsymbol{v}_{Old} \leftarrow \boldsymbol{v}$
$\tilde{\boldsymbol{y}} \leftarrow \boldsymbol{y} + \sigma C_v \bar{\boldsymbol{v}}$
$y_{1,2} \leftarrow \tilde{y}_{1,2} - \sigma S(\frac{\tilde{y}_{1,2}}{\sigma}, \frac{\beta}{\gamma\sigma})$
$y_3 \leftarrow solveAffine(\tilde{y}_3, u)$
$\boldsymbol{v} \leftarrow \boldsymbol{v} - \tau C_v^* \boldsymbol{y}$
$\bar{\boldsymbol{v}} \leftarrow 2\boldsymbol{v} - \boldsymbol{v}_{Old}$
$\epsilon \leftarrow \frac{|u-u_{OldM}|+|\boldsymbol{v}-\boldsymbol{v}_{OldM}|}{2|\Omega|}$
$\boldsymbol{v}$

5.4. Results

5.4.1. Datasets

We created different dynamic datasets to evaluate our models. We took the Dimetrodon, Hydrangea and Rubber Whale dataset from the IPOL database [44] and used the given ground truth velocity field from [44] to create series of 4 consecutive images. We want to mention that our algorithms are limited to a motion with a velocity field of a maximum magnitude of 1 and, consequently, we scaled down the given ground truth fields accordingly. To create image u_{n+1} from image u_n and velocity field $\boldsymbol{v}$ we evaluated $u_n(x + \boldsymbol{v})$ using cubic interpolation. Afterwards, we added Gaussian noise with $\sigma = 0.0002$ to create the noisy counterparts. Figure 5.2 gives an overview of the datasets and ground truth flows.

5.4.2. Image Reconstruction

The first question that naturally arises in the context of this model is: Does the joint image reconstruction and motion estimation give a benefit towards a two-step model

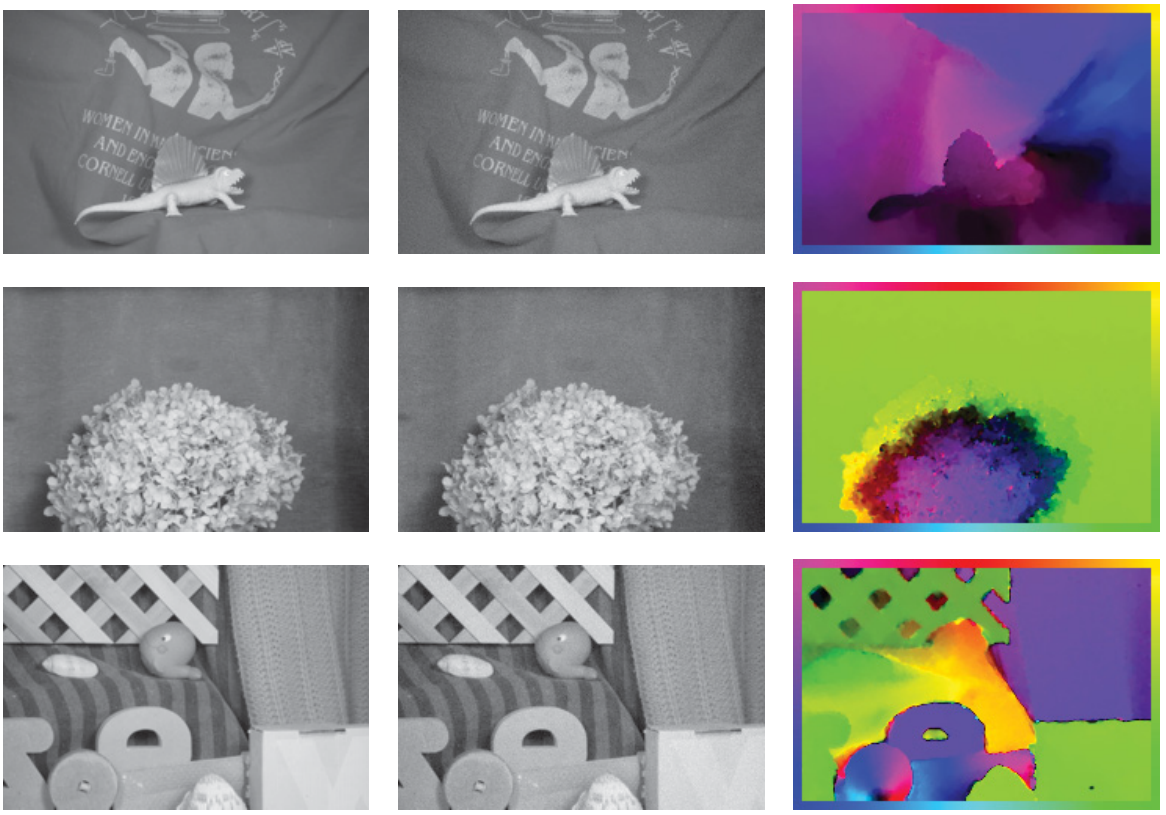


Figure 5.2.: Datasets (from [44]) used for evaluation of the joint models. Left: first image, middle: noisy counterpart (additive Gaussian noise with $\sigma = 0.0002$), right: ground truth velocity field used to create consecutive images.

of image processing and motion estimation? To answer this question concerning the benefit for image reconstruction first we took different series of noisy images (see Figure 5.2, Gaussian noise with $\sigma = 0.0002$) and used a standard ROF model (see [60] for details)

$$u = \arg\min_u \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 \, \mathrm{d}t$$

for noise removal on the one hand. On the other hand, assuming the ground truth velocity field $\boldsymbol{v}$ is known, we consider the subproblem for u from Section 5.2.3 and Section 5.3.3 and we minimize

$$u = \arg\min_u \int_0^T \frac{1}{2} \|Ku - f\|_2^2 + \alpha \|\nabla_x u\|_1 + \gamma \|u_t + \nabla_x u \cdot \boldsymbol{v}\|_1 \, \mathrm{d}t.$$

Both models have been tested with a broad range of regularization weights α. For the motion-including model we fixed $\gamma \equiv 1$.

To measure the quality we evaluated the structural similarity index, **SSIM**, [73] which measures the difference in luminance, contrast and structure of ground truth image and reconstruction:

$$SSIM := \frac{(2\mu_u \mu_{u_{rec}} + C_1)(2\sigma_{u,u_{rec}} + C_2)}{(\mu_u^2 + \mu_{u_{rec}}^2 + C_1)(\sigma_u^2 + \sigma_{u_{rec}}^2 + C_2)},$$

where $\mu_u, \mu_{u_{rec}}, \sigma_u, \sigma_{u_{rec}}$ and $\sigma_{u,u_{rec}}$ are local means, standard deviations and cross-covariances for ground truth image u and reconstruction u_{rec}. The constants are defined as $C_1 = 0.01^2$ and $C_2 = 0.03^2$. The index takes values between -1 and 1, whereas 1 stands for perfect similarity. Moreover we calculated the signal-to-noise ratio, **SNR**, and peak signal-to-noise ratio, **PSNR**, between noisy input and reconstruction.

$$SNR := 10 \log_{10}\left(\frac{\mathrm{mean}(u^2)}{\mathrm{mean}((u - u_{rec})^2)}\right),$$
$$PSNR := 10 \log_{10}\left(\frac{\max(u^2)}{\mathrm{mean}((u - u_{rec})^2)}\right).$$

Table 5.1 contains the results for both models. We deduce that our model outperforms the standard ROF model for each of the evaluated measures and datasets. This justifies the use of a joint model instead of a two-step algorithm, because we can take advantage of a significant measurable benefit.

From the perspective of motion estimation we refer to Section 4.6.6. Here we showed

Dataset	Algorithm	SSIM	SNR	PSNR
Dimetrodon	Joint	0.9764	33.3996	41.8472
	ROF	0.97406	32.7162	41.1637
Hydrangea	Joint	0.96496	33.2697	40.1215
	ROF	0.96086	32.343	39.1948
Rubber Whale	Joint	0.96957	35.1268	40.1419
	ROF	0.9628	34.1481	39.1633

Table 5.1.: Table comparing the quality of reconstruction for standard ROF and our joint model. The evaluated images have been corrupted with Gaussian noise of variance $\sigma = 0.0002$.

that the quality of the estimated motion field strongly depends on the quality of the images used. Consequently, the reliability of the estimation motion field increases with better image quality gained from the joint model.

5.4.3. Joint Image Reconstruction and Motion Estimation

Denoising and Motion Estimation

In the previous section, we measured the quality of the image reconstruction towards the influence of noise in a reduced model, where the exact velocity field was known. Proceeding now to the full model, where image sequence u and velocity field $\boldsymbol{v}$ are to be calculated, we evaluated our four models, namely TV-L^2 optical flow, TV-L^2 mass preservation, TV-TV optical flow and TV-TV mass preservation, with a broad set of regularization parameters α (weight for the image regularizer) and β (weight for the velocity field regularizer). The weighting parameter γ for the constraint was set to 1 in all experiments. The alternating minimization is stopped if the difference of two consecutive iterates $u^k, u^{k+1}, \boldsymbol{v}^k, \boldsymbol{v}^{k+1}$ reaches $1e-4$. For the subproblems in u and $\boldsymbol{v}$ the tolerance ϵ is set to $1e-6$.
To measure the quality of the reconstruction we used, similar to the previous section, the SSIM, SNR and PSNR. The quality of the reconstructed velocity field was measured in terms of the absolute endpoint error, AEE (see Section 4.6.1), and the angular error, AE (see Section 4.6.1). Table 5.2 contains the evaluation results. What becomes clear immediately is that the quality of the image reconstruction outperforms the standard ROF model for each of the joint algorithms. The overall best algorithm is given by the $TV-TV$ optical flow model, which performes best in terms of measures for image reconstruction as well as motion estimation. The visualized flows for the respective best reconstruction of the Dimetrodon sequence (see Figure 5.2) of our four models can

Dataset	Algorithm	SSIM	SNR	PSNR	AE	AEE
Dimetrodon	TV-L^2 o. f.	0.97664	33.3722	41.8198	0.11492	0.06754
	TV-L^2 m. p.	0.97405	32.721	41.1686	0.13659	0.080702
	TV-TV o. f.	0.9762	33.2453	41.6928	0.11881	0.070384
	TV-TV m. p.	0.9751	33.0121	41.4596	0.13208	0.078705
	ROF	0.97406	32.7162	41.1637	-	-
Hydrangea	TV-L^2 o. f.	0.96557	33.2038	40.0556	0.070458	0.038887
	TV-L^2 m. p.	0.96133	32.4387	39.2905	0.15986	0.086332
	TV-TV o. f.	0.96314	32.782	39.6338	0.042464	0.023671
	TV-TV m. p.	0.96399	32.9219	39.7737	0.092403	0.0505843
	ROF	0.96086	32.343	39.1948	-	-
Rubber Whale	TV-L^2 o. f.	0.96937	34.9829	39.998	0.1078	0.056923
	TV-L^2 m. p.	0.96397	34.2408	39.2559	0.1257	0.066107
	TV-TV o. f.	0.9655	34.4342	39.4493	0.10154	0.054059
	TV-TV m. p.	0.96507	34.4428	39.4579	0.12327	0.064973
	ROF	0.9628	34.1481	39.1633	-	-

Table 5.2.: Table containing evaluation results for the joint image reconstruction and motion estimation models. Additionally, we added the denoising results for the ROF model.

be found in Figure 5.3. Results for the Hydrangea and Rubber Whale examples have been transferred to Appendix $A.3$.

The reconstructed images are hard to distinguish from a visual point of view. However, Table 5.2 shows that SSIM, SNR and PSNR are minimized by the $TV - TV$ optical flow model.

Denoising, Inpainting and Motion Estimation

Until now, the operator K was equal to the identity. This corresponds to simple denoising and motion estimation. A more challenging task is given by inpainting of unknown regions of the image domain Ω. In this context we think of two branches of inpainting. First, considering a spatial image domain Ω for each timestep $t \in [0, T]$, we think of an unknown part Σ of the spatial domain Ω, which is to be reconstructed. Moreover, due to our dynamical models, we are also able to inpaint complete images in the sequence domain $\Omega \times [0, T]$.

Spatial Inpainting:

To cover the former mentioned spatial inpainting we choose the Dimetrodon sequence, corrupted with Gaussian noise of variance $\sigma = 0.0002$, and cut out every second pixel

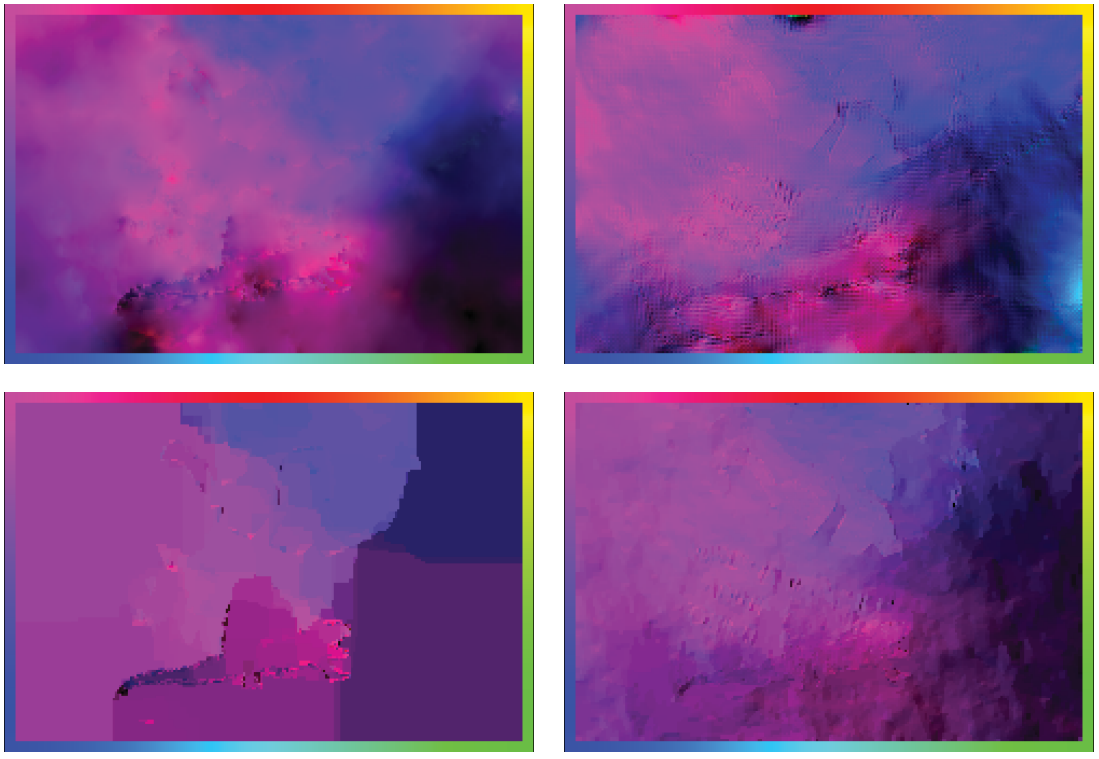


Figure 5.3.: Estimated motion field for the Dimetrodon example between timesteps t_1 and t_2. Top-left: TV-L^2 optical flow, top-right: TV-L^2 mass preservation, bottom-left: TV-TV optical flow, bottom-right: TV-TV mass preservation

Figure 5.4.: Reconstructed images generated by the joint models. Top-left: ground truth image, top-right: noisy image, middle-Left: $TV - L^2$ optical flow, middle-right: $TV - L^2$ mass preservation, bottom-left: $TV - TV$ optical flow, bottom-right: $TV - TV$ mass preservation.

Algorithm	SSIM	SNR	PSNR	AE	AEE
TV-L^2 optical flow	0.8885	21.6742	30.1218	0.4169	0.22869
TV-L^2 mass preservation	0.88663	21.6323	30.0799	0.43983	0.24101
TV-TV optical flow	0.88812	21.6765	30.124	0.42414	0.23239
TV-TV mass preservation	0.88783	21.6515	30.099	0.41176	0.22612

Table 5.3.: Table containing evaluation results for the spatial inpainting applied to the Dimetrodon sequence. The values for SSIM, SNR and PSNR correspond to reconstructed image and cut-out original image. The values for AE and AEE correspond to the velocity field estimated between reconstructed slice and consecutive image, and the ground truth field used for creating the sequence.

in X and Y direction. The reduced domain is denoted as $\Omega_{\frac{1}{2}}$ and for the input sequence we have

$$f : \Omega_{\frac{1}{2}} \times [0,T] \subset \Omega \times [0,T] \to \mathbb{R}.$$

Consequently, the operator K now acts as a projection / downsampling

$$K : \Omega \times [0,T] \to \Omega_{\frac{1}{2}} \times [0,T]\,.$$

The task for each model is to reconstruct the image on the complete domain $\Omega \times [0,T]$ in a meaningful way. For each model we evaluate a set of regularization parameters α, β and set $\gamma = 1$. The alternating minimization algorithms are terminated when two consecutive iterates $(u^k, \boldsymbol{v}^k)$ and $(u^{k+1}, \boldsymbol{v}^{k+1})$ differ less than $1e-4$. For the subproblems in u and $\boldsymbol{v}$ the tolerance ϵ is set to $1e-6$.
To measure the quality of the image reconstruction we calculate the SSIM, SNR and PSNR between reconstructed image sequence and ground truth image sequence. For the velocity field we calculate AE and AEE between estimated field and ground truth field. Table 5.3 and Figures 5.5 and 5.6 contain the results. Unfortunately none of the proposed models performs well here. On the one hand, the estimated velocity fields massively differ from the given ground truth. On the other hand, we cannot expect to obtain a perfect image reconstruction with incorrect motion fields. This is reflected in the low values for SSIM and consequently in an insufficient image reconstruction.

Slice Inpainting:
To evaluate our models in terms of inpainting of complete timesteps / slices, we again take the Dimetrodon sequence, corrupted with Gaussian noise of variance $\sigma = 0.0002$, and cut out one complete slice. Now, the proposed models are used to reconstruct that missing part. Parameters and stopping criterion are chosen similar to the spatial

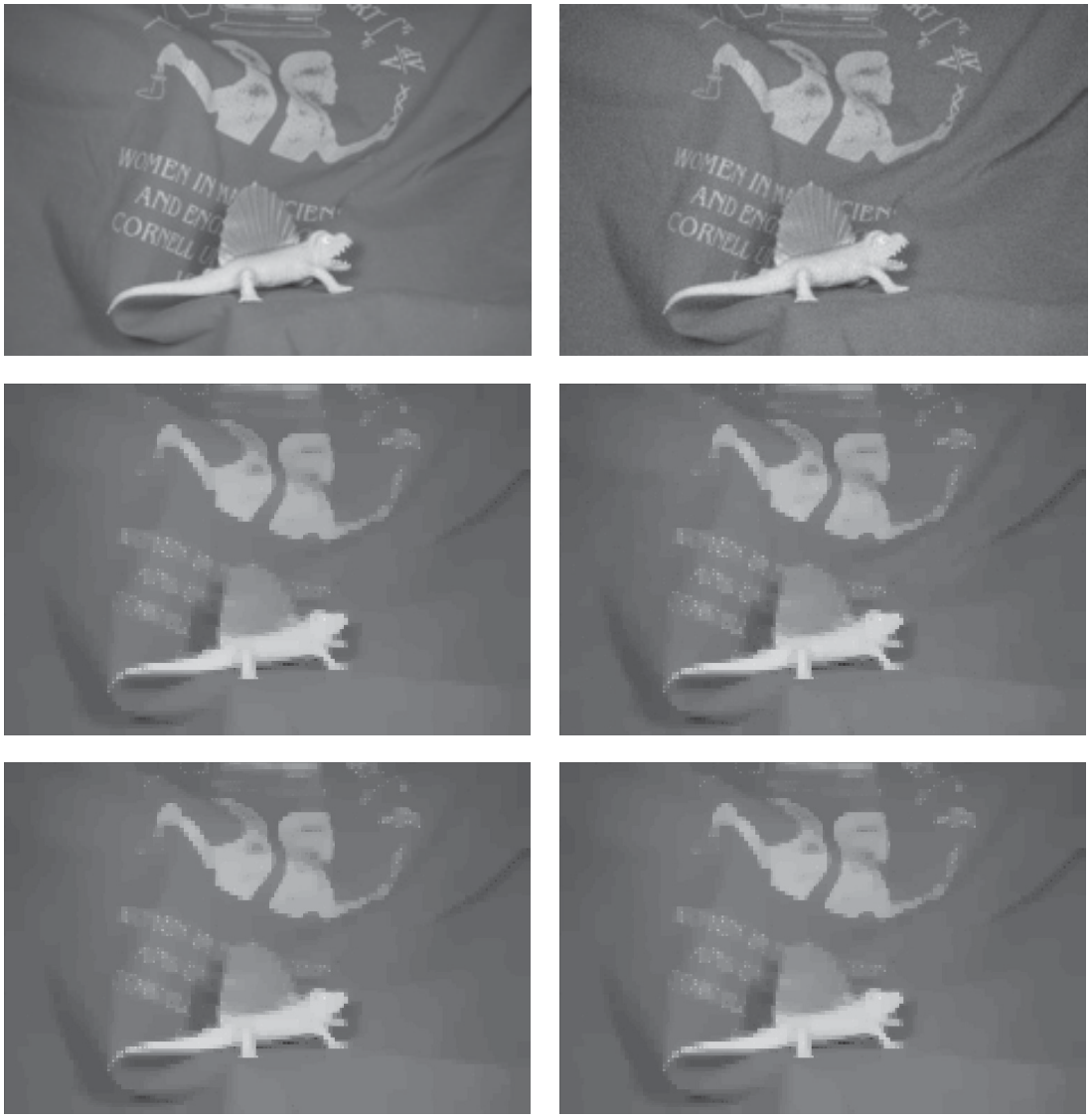


Figure 5.5.: Figure containing reconstruction results for the spatial inpainting applied to the Dimetrodon sequence. Top-left: ground truth image, top-right: noisy image, middle-left: $TV - L^2$ optical flow , middle-right: $TV - L^2$ mass preservation, bottom-left: $TV - TV$ optical flow, bottom-right: $TV - TV$ mass preservation.

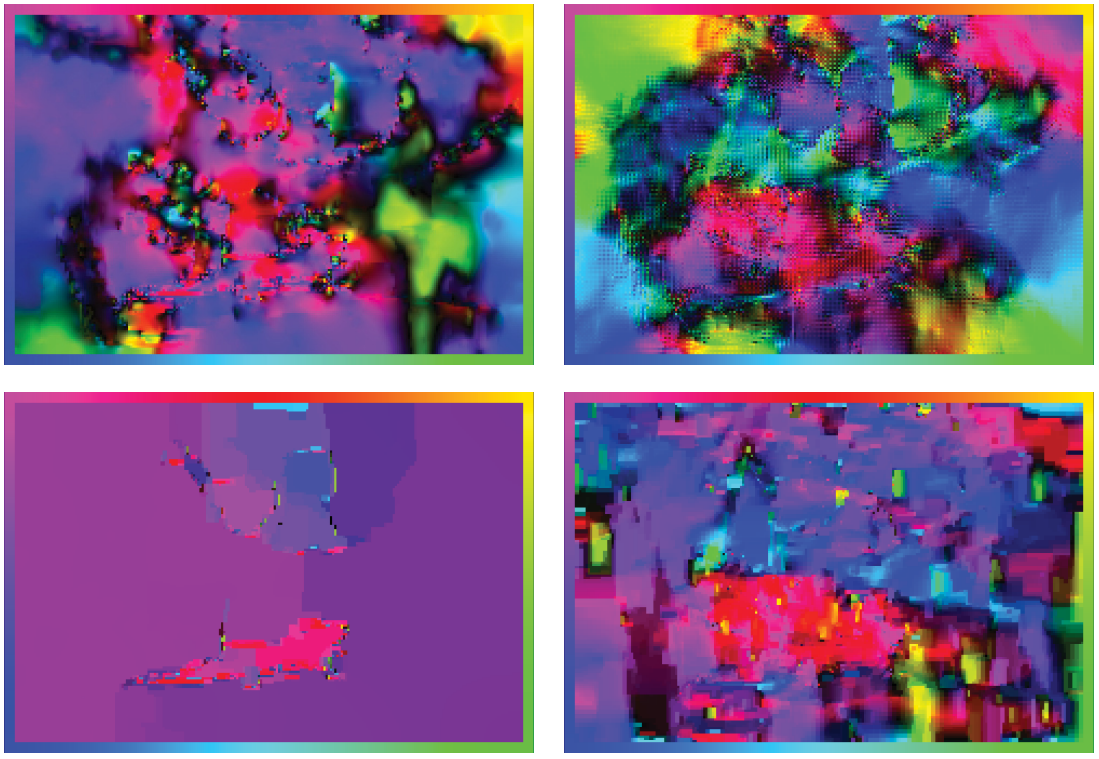


Figure 5.6.: Figure containing flow field results for the spatial inpainting applied to the Dimetrodon sequence. Top-left: ground truth field, middle-Left: $TV - L^2$ optical flow, middle-right: $TV - L^2$ mass preservation, bottom-left: $TV - TV$ optical flow, bottom-right: $TV - TV$ mass preservation.

Algorithm	SSIM	SNR	PSNR	AE	AEE
TV-L^2 optical flow	0.91557	25.1015	33.549	0.1797	0.10552
TV-L^2 mass preservation	0.83081	20.4868	28.9343	0.41777	0.2369
TV-TV optical flow	0.91167	24.8358	33.2833	0.18833	0.11078
TV-TV mass preservation	0.88901	23.3889	31.8364	0.23722	0.13613

Table 5.4.: Table containing evaluation results for the slice inpainting applied to the Dimetrodon sequence. The values for SSIM, SNR and PSNR correspond to reconstructed image and cut-out original image. The values for AE and AEE correspond to the velocity field estimated between reconstructed slice and consecutive image, and the ground truth field used for creating the sequence.

inpainting case.

Table 5.4 and Figures 5.7 and 5.8 contain the results. From a visual point of view we observe that all models yield a good result. The optical flow models perform slightly better. Here, small structures (i.e. eye of the Dimetrodon) are closer to the original. Taking the numbers we see that the $TV-L^2$ optical flow model yields the best result in terms of all error measures. The $TV-TV$ optical flow model performs slightly worse. We want to mention that the results for SSIM, SNR and PSNR cannot be compared with those from Table 5.1 (benefit from joint model), because applying just the ROF model to an unknown slice is not possible.

Runtime

Finally, we compare runtimes of the presented algorithms. Therefore, the Dimetrodon sequence is used and scaled down up to a factor of one percent. The four presented algorithms, namely $TV-L^2$ optical flow, $TV-L^2$ mass preservation, $TV-TV$ optical flow, $TV-TV$ mass preservation, are applied to the downscaled problems with fixed parameters α, β and γ. The algorithms are terminated when the difference of two consecutive iterates $(u^k, \boldsymbol{v}^k)$ and $(u^{k+1},, \boldsymbol{v}^{k+1})$ falls below $\epsilon < 1e-5$. The tolerance for the subproblems is set to $1e-6$.

Table 5.9 contains the results with increasing number of pixels in the image sequence on the x-axis and an increasing runtime on the y-axis. In general, the runtime of all models increase exponentially with increasing problem size. But we see huge differences between the $TV-TV$ mass preservation model (black) with a strong exponential character and the $TV-L^2$ optical flow model with a nearly linear increase. A second interesting fact is that the optical flow models (red and blue) perform much better than the mass preservation models (green and black). A slight increase would be normal, because for the mass preservation model there is a further divergence operator involved.

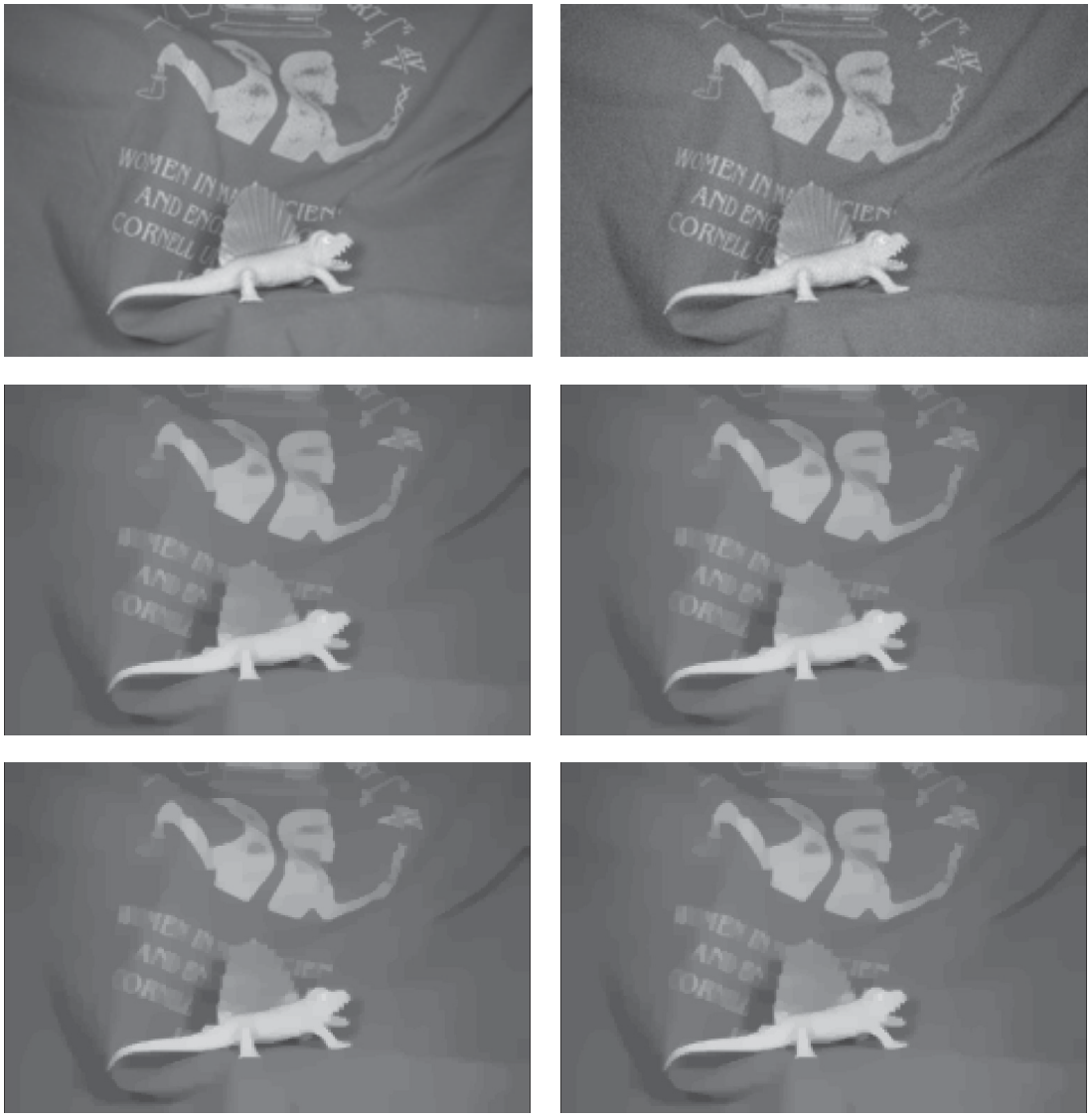


Figure 5.7.: Figure containing reconstruction results for the slice inpainting applied to the Dimetrodon sequence. Top-left: ground truth image, top-right: noisy image, middle-left: $TV - L^2$ optical flow, middle-right: $TV - L^2$ mass preservation, bottom-left: $TV - TV$ optical flow, bottom-right: $TV - TV$ mass preservation

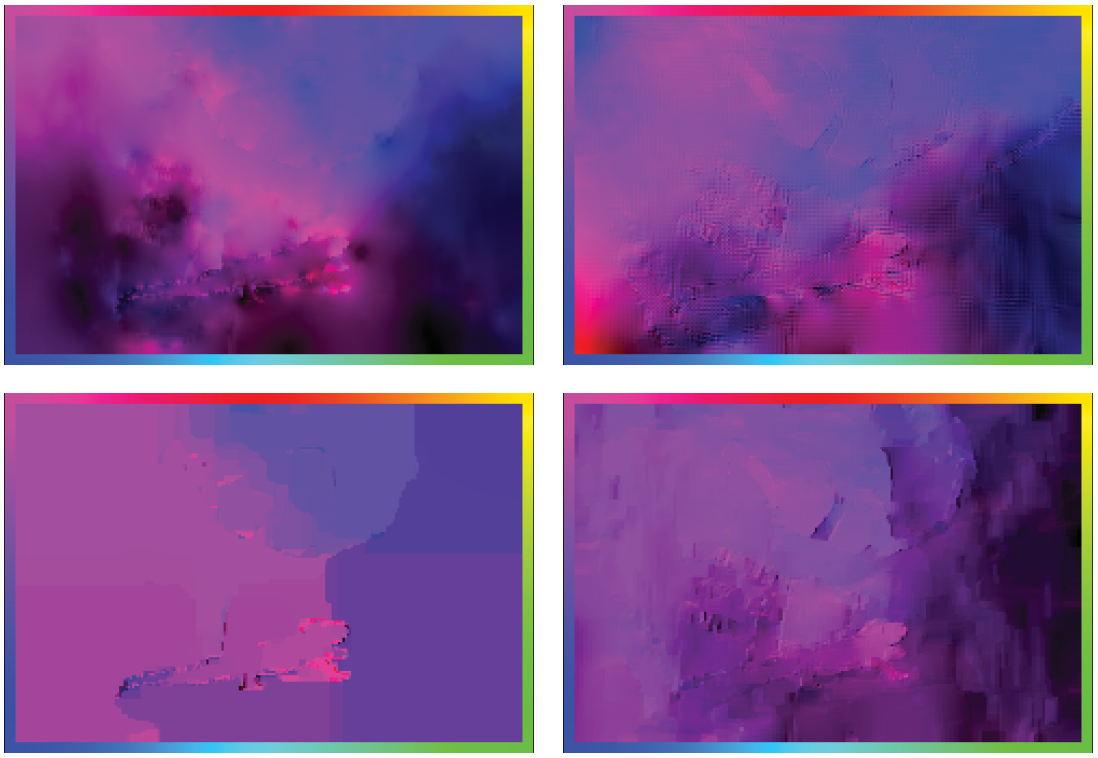


Figure 5.8.: Figure containing flow field results for the slice inpainting applied to the Dimetrodon sequence. Top-Left: $TV-L^2$ optical flow, top-right: $TV-L^2$ mass preservation, bottom-left: $TV-TV$ optical flow, bottom-right: $TV-TV$ mass preservation

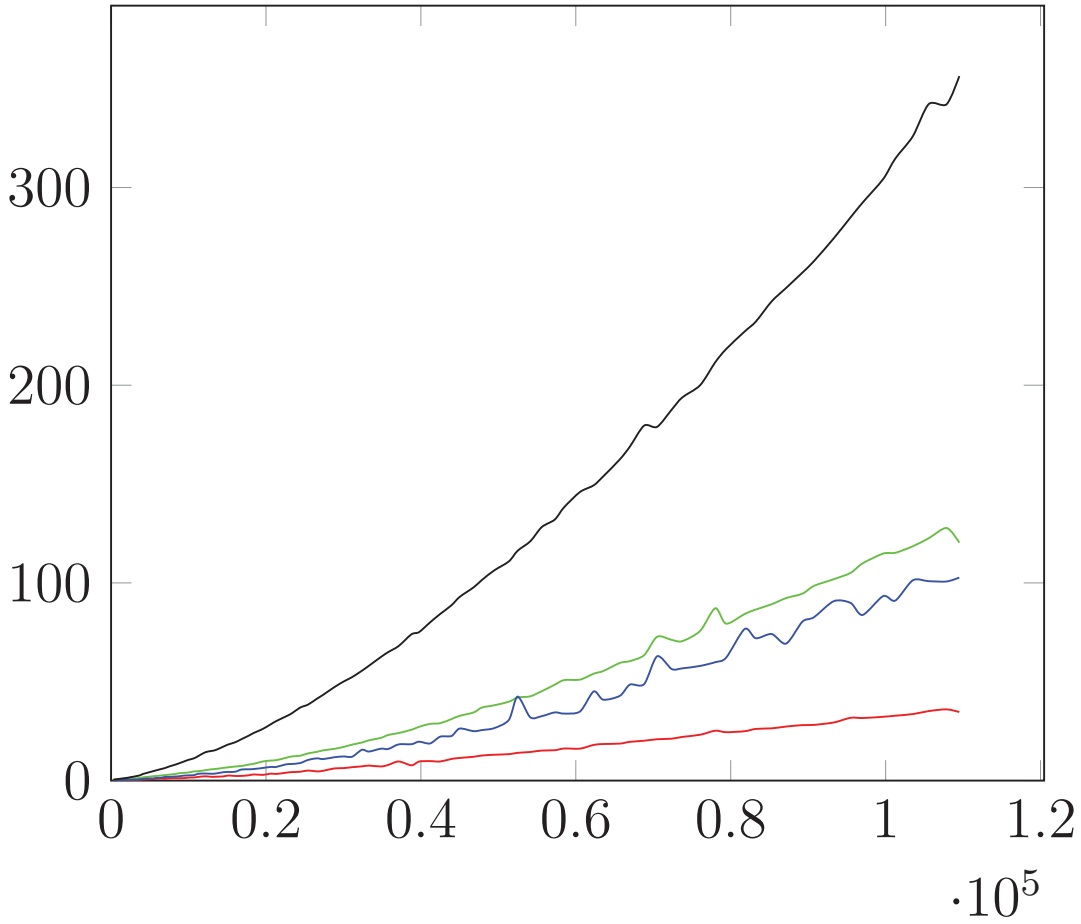


Figure 5.9.: Plot of problem size in number of pixels (x-axis) versus the runtime in seconds (y-axis).The colors specify $TV - L^2$ optical flow (red), $TV - L^2$ mass preservation (light green), $TV - TV$ optical flow (blue), $TV - TV$ mass preservation (light black)

Here, we determine an increase of approximately factor 4 between respective optical flow and mass preservation models, which only can origin from the mass preservation data term.
At the end we want to underline the very fast convergence speed of the $TV - L^2$ optical flow model. Not only does it converge fast, but it often generates the best result in the previous evaluations (see Section 5.4.3 and 5.4.3).

Parameter Dependence

Finally, we want verify the parameter dependence of our algorithms. We take, similar to the previous sections, the Dimetrodon sequence and run our algorithms, until the differences between consecutive iterations fall below a critical threshold $1e - 5$. The results can be found in Figure 5.10.

For the weighting parameter α, which acts as the weighting parameter for the image regularizer, we deduce a strong dependence of the image reconstruction. The optimal α lies at $\alpha = 0.008$. Already small changes in α strongly reduce the structural similarity of the reconstructed image. Interestingly, the quality of our resulting velocity field keeps being relatively good with changes in α. The AEE increases from ≈ 0.07 to ≈ 0.1 in the worst case, which coincides with a slight change of the quadratic error by only 0.03 per pixel. Moreover, we want to mention that all proposed algorithms react in the same way to changes in α i.e. the resulting plots have the same structure.
For changes in β, the weighting parameter for the velocity field, we want to underline that each algorithm requires a different optimal β. To respect this fact on the one hand, and to get meaningful results on the other hand we decided to check factors of the respective optimal β starting from 0.1 up to 10 for each algorithm. Starting with the influence towards the image reconstruction, we deduce that all algorithms are very robust for parameter changes of β. The absolute error of the velocity field on the other hand is vulnerable towards changes in β, where the parameter should be rather chosen too high than too low.

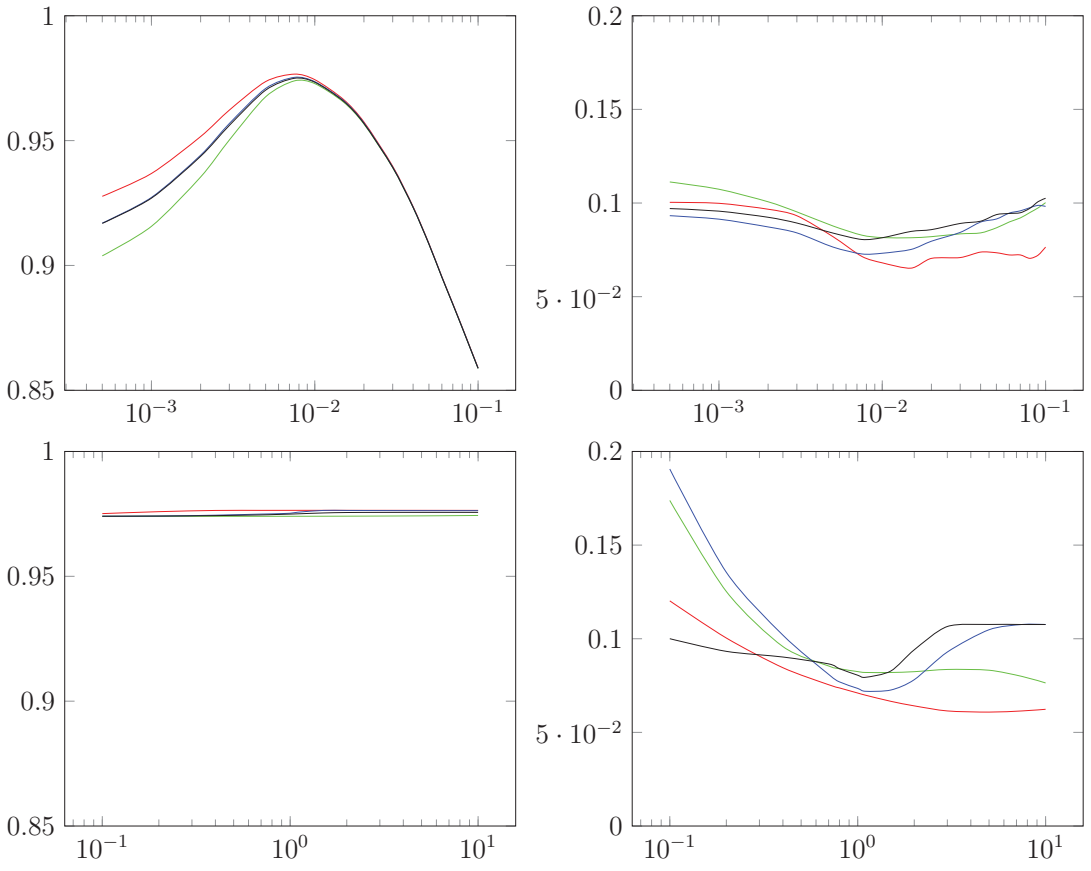


Figure 5.10.: Plots of parameter dependence. The plots on the left side contain SSIM of the reconstructed image, whereas the plots on the right side display AEE of the velocity field. The top row has different values of α on the x-axis, the bottom row has factors of the respective optimal β for each algorithm on the x-axis. The colors in the plots specify $TV - L^2$ optical flow (red), $TV - L^2$ mass preservation (light green), $TV - TV$ optical flow (blue), $TV - TV$ mass preservation (light black).

6

Application of the Joint Model

6.1. Cell-Tracking

6.1.1. Introduction

Of our special interest is a locomotion mechanism called blebbing. This mechanism is used by developing germ cells (e.g. cells that will rise to sexual organs) of all organisms that reproduce sexually. While the cell is blebbing, it's cytoskeleton breaks up and causes the membrane to bulge outwards. First, actin filaments are flowing inside the blebb, then other components follow. The blebbing process is not very well understood, especially how fluid mechanics of different interacting cell-components works during the blebb. This behavior of certain, and interaction of different cell-components is done by intracellular flow, a special case of intracellular fluid mechanics, the process we are interested in. Analysis of the underlying data consists of different tasks, for example segmentation and velocity field calculation (see Figure 1). There also exists the assumption that a cell builds up pressure to force a blebb into a certain direction. The experiments are performed with embryos of zebra fishes at the Center for Molecular Biology of Inflammation (ZMBE, WWU Münster). Some initial computations have already been done in Chapter 3, but we want to present more advanced results here.

6.1.2. Problem Formulation

The experimental pipeline starts with the biological preparation. Here, embryos of zebra fish are prepared for the later microscopy observations. The fish are genetically altered such that the actin filaments are already autofluorescent. To make another intracellular component fluorescent, a retrovirus is manually injected into the growing embryo. After hours of maturation at an optimal temperature the microscopy data is recorded. Now, for each timestep, an image at a certain z-direction is taken. The data is divided in two channels, one coincides with the actin filaments and another with the other cell component. This data is given to us. From a mathematical point of view our work divides in three fields of image processing. First, the recorded images are quite noisy, consequently the biologists are interested in image denoising. Secondly, the underlying motion is of interest. We want to underline here that motion of the cell as whole as well as motion information of the internal component is required. Finally we want to automatically detect the cell inside the image.
Proceeding to our models, we are able to cover two of the three tasks with our joint image reconstruction and motion estimation model at once. We are able to denoise and estimate motion of both channels and can expect to gain at better result than splitting both tasks (see Chapter 5 for details about benefits of the joint model). For the segmentation tasks we use a Chan-Vese based framework, already introduced in Chapter 3.3.2.

6.1.3. Image Processing and Motion Estimation

Due to the fact that in each evaluation the joint $TV - L^2$ optical flow model performed best, this model is used here. In this context we want to recall that a total variation regularizer is used for the image sequence u, but an L^2 regularization to the gradient of the velocity field $\boldsymbol{v}$.
Since we have two data channels, where different key features in the later analysis will be extracted, both channels are processed independently. The channel containing colored filaments, in the following called *green channel*, should be used to detect the cell shape and the cell movement as a whole. So in context of motion estimation we are not interested in the movement of small structures and choose a higher regularization parameter. For the channel containing image information about the intracellular component we are interested in image information on a very fine scale. Consequently a much lower regularization weight β is chosen.
Figures 6.1 and 6.2 contain raw (given) image data, reconstructed image and color-

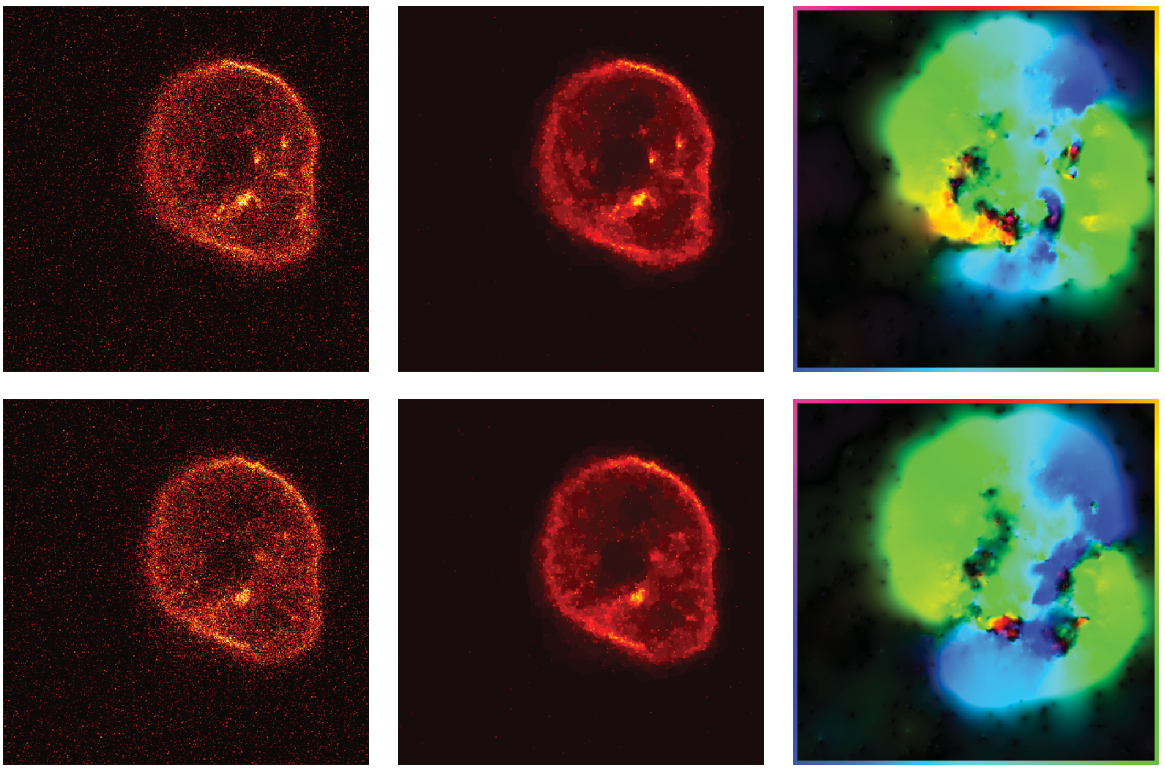


Figure 6.1.: Visualization of the green channel at two timesteps (top and bottom). Left: input image, middle: denoised image, right: estimated motion field. Images are property of Raz group from ZMBE [79].

coded flow information of both channels. In the denoised images, the cell separates from the background very good now, while sharp edges have been preserved. This makes the later segmentation faster and more accurate. The estimated motion field for the green channel contains heavy motion at the boundary of the cell, while the motion field of the red channel seems to have many independently moving parts.

6.1.4. Segmentation

For the segmentation part we are now able to use the denoised images from the previous part. A framework, similar to the one from Section 3.3.2 is now applied to each image in the green channel sequence. At this point the advantage coming from the image reconstruction in our joint model is beneficial. Initializing the segmentation with a simple thresholding algorithm we then apply the Chan-Vese [28] framework to each image in the sequence independently. The resulting segmentation is then smoothed and we close potential holes. A step-by-step application can be found in Figure 6.3. Compared to the afore used Gaussian smoothing, a total variation based image denoising

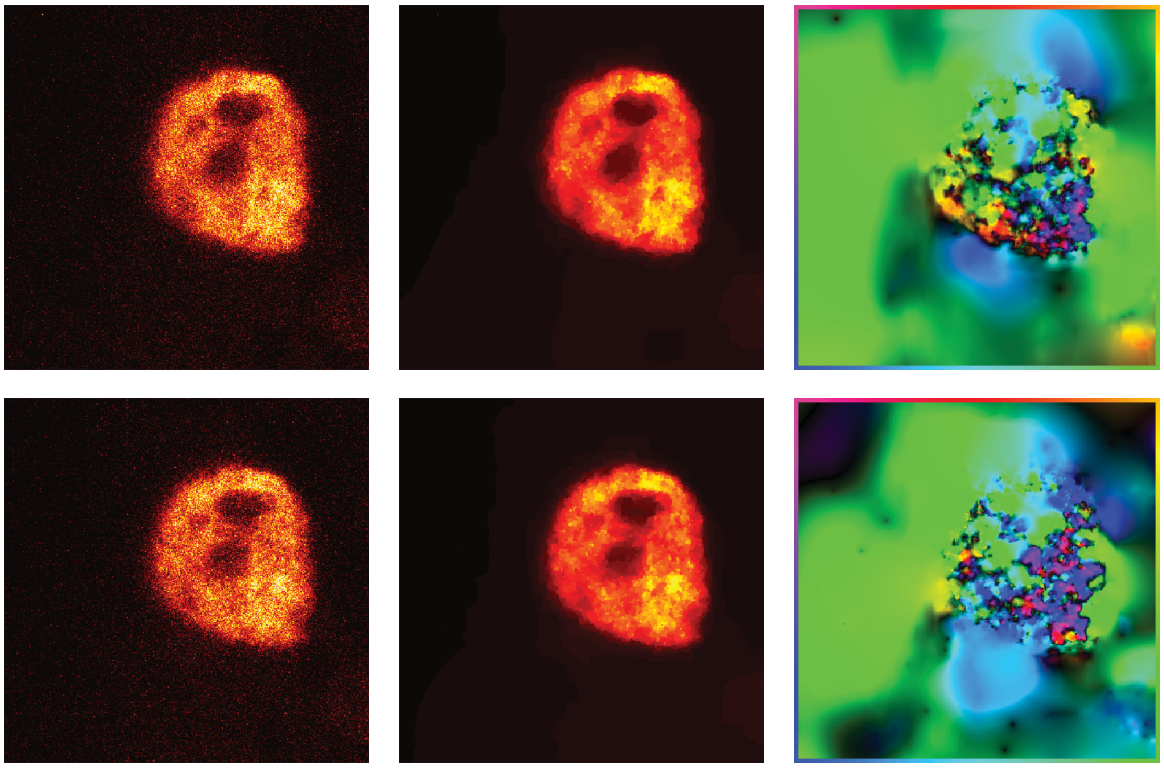


Figure 6.2.: Visualization of the red channel at two timesteps (top and bottom). Left: input image, middle: denoised image, right: estimated motion field. Images are property of Raz group from ZMBE [79].

enhances the quality of the reconstruction. Now, the enhanced images lead to a better segmentation result. We see that the cell is very well segmented. This segmentation can be used for further analysis of the underlying intracellular process.

6.2. Bacterial Pattern Formation

Pattern formation of self-organizing systems like bacteria on limited surfaces are not very well understood. However, they play an important role in many physical systems. In collaboration with Hugo Wioland and Raymond E. Goldstein from the University of Cambridge [53] we applied our joint model for image denoising and motion estimation to image data of a bacterial suspension on different geometries. The results of these experiments have already been published (see [75] and [47]), so our evaluation can be seen as a cross-check of the results.

For both experiments bacteria were placed on a bounded surface. Afterwards, movies were acquired at 125 fps with a high-speed camera. For our algorithm, we imported the recorded movies into MATLAB and applied the joint $TV - L^2$ optical flow model for joint motion estimation and image reconstruction algorithm (see 5.2.2). The operator K was chosen as identity. The weights for the regularizers were set to $\alpha = 0.04, \beta = 0.001$ and $\gamma = 0.01$ for both datasets. Since the motion is not homogeneous over time, we averaged the estimated motion field over all timesteps.

Figure 6.4 contains results for a limited roundish surface. We observe very good denoising results for the image data and the object structure is well recovered. Besidest that, we see a strong counterclockwise rotation inside the droplet. Moreover, a thin boundary layer that rotates clockwise is visible. This observation coincides with the published results from [75] and [47].

In Figure 6.5 we printed the result for the ellipsoid geometry. Similar to the round structure the image is well recovered. Due to a bad image resolution the bacteria are not visible as clear as in the other experiment. The resulting velocity field has a more complex structure. The inner motion divides into two vortices. On the left side we observe a round vertex of counterclockwise rotation, whereas the other vertex has an ellipsoid form but also rotates counterclockwise. Besides the motion inside, we see a thin boundary layer with clockwise rotation.

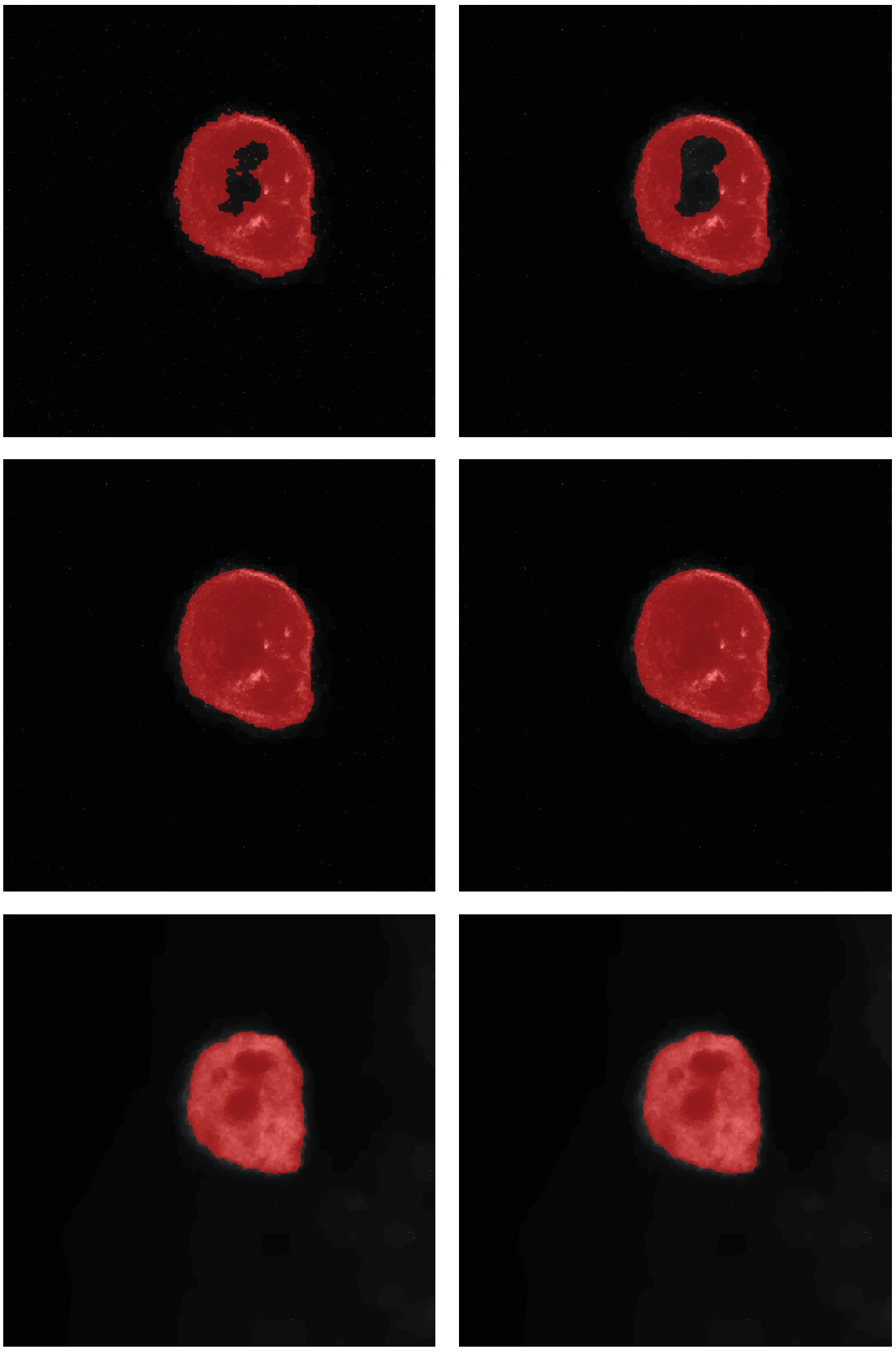


Figure 6.3.: Step-by-step segmentation. Top-Left: After thresholding, Top-Right: After Chan-Vese, Middle-left: Holes closed, Middle-right: Smoothed contour, Bottom-left: Red channel after Chan-Vese, Bottom-right: Red channel after smoothing. Images are property of Raz group from ZMBE [79].

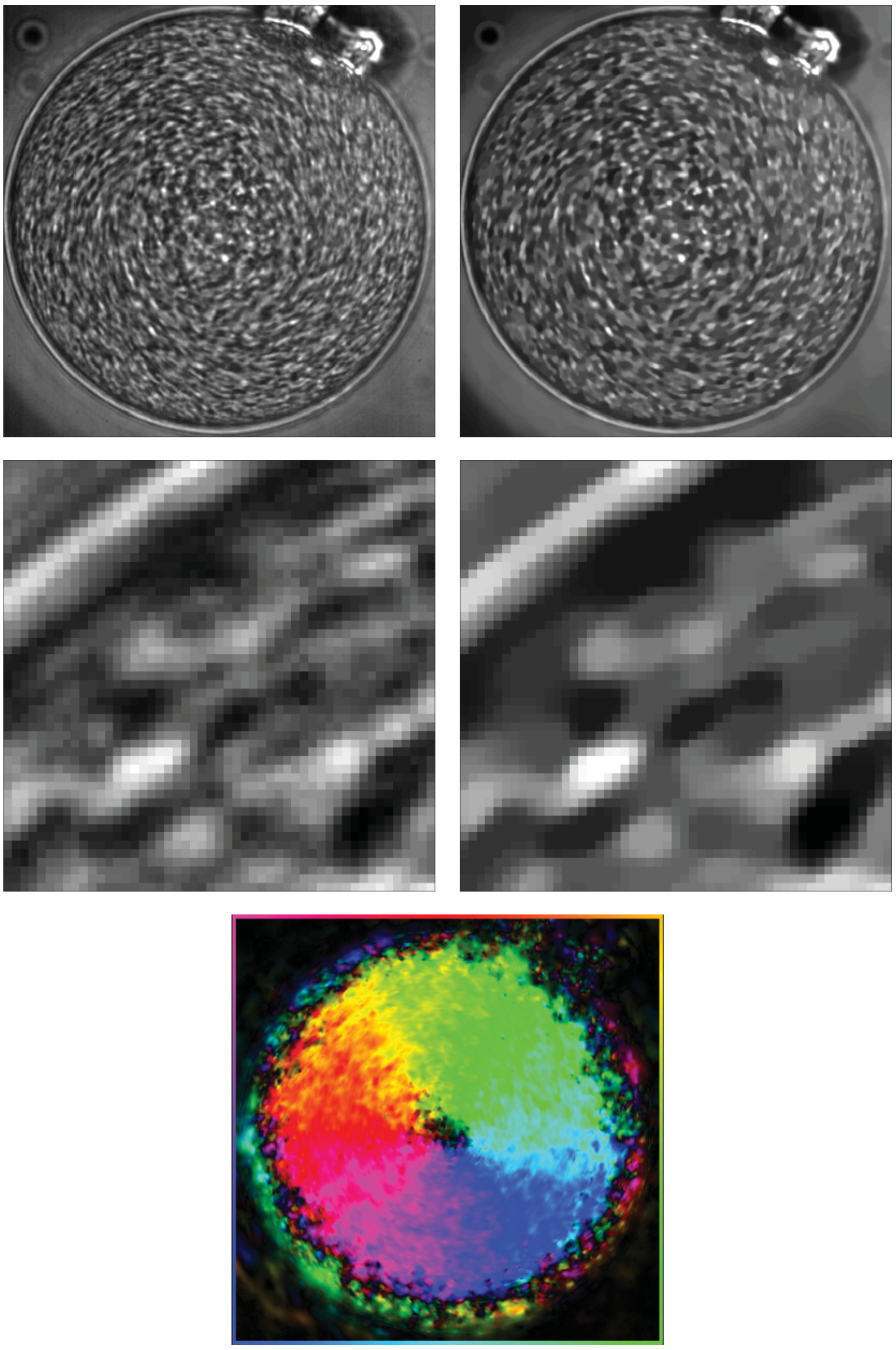


Figure 6.4.: Joint model applied to images (property of Goldstein group [53]) of bacterial formation in an round structure. Top: Raw image, Middle-Top: Denoised image, Middle-Bottom: Zoom into raw and TV denoised image. Bottom: Estimated mean flow field

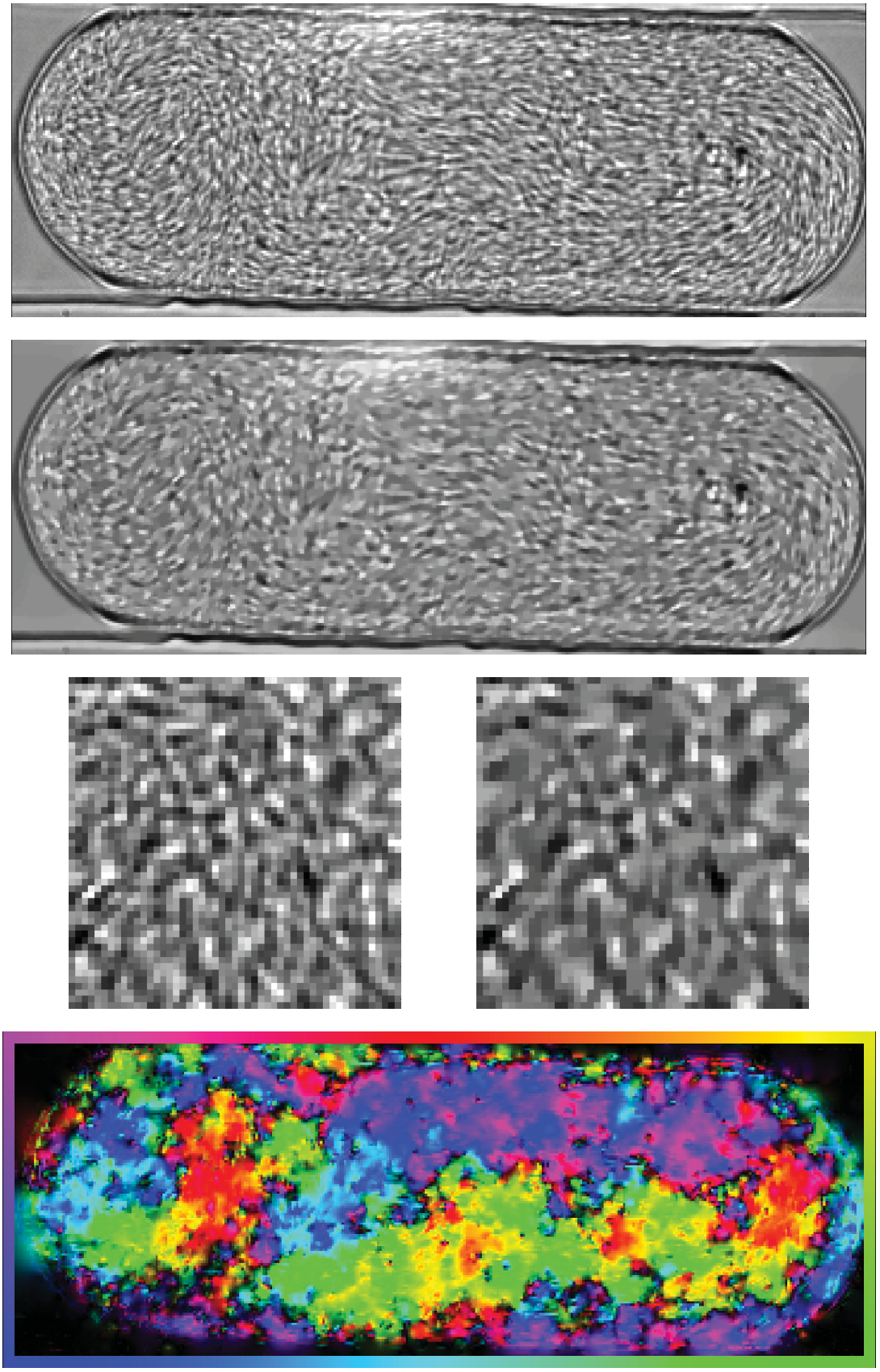


Figure 6.5.: Joint model applied to images (property of Goldstein group [53]) of bacterial formation in an ellipsoid structure. Top: Raw image, Middle-Top: Denoised image, Middle-Bottom: Zoom into raw and TV denoised image. Bottom: Estimated mean flow field

7 Conclusions and Outlook

This thesis consists of three main topics. In a general motion estimation chapter, we analyzed different models for motion extraction for image sequences. These motion estimation models were later coupled with the problem of image reconstruction in a second chapter. Finally, we presented different applications for image processing in the context of segmentation, denoising and motion estimation. In order to conclude, let us shortly summarize those three pillars of this thesis.

7.1. Motion Estimation

In the beginning of this chapter, we gave a short introduction into the problem of motion estimation from images. Here, we lay special interest onto the aperture problem and possible problems due to optical illusion. Then, we deduced the optical flow constraint and the mass preservation constraint from different assumptions to the underlying motion field. Both constraints were used as data fidelity in variational models where we added assumptions about smoothness (L^2) or sparsity (TV) of the gradient using a regularizing term. For the resulting variational models, we presented minimization strategies based on the primal-dual framework of Chambolle and Pock [26]. In an evaluation section, we began with a critical discussion on existing quality measures for velocity fields, and validated the proposed models towards different types of artificial and real flow. Moreover, we showed that already small levels of noise in the image data massively degrade the quality of the estimated motion field.
In terms of the regularizer, we did not evaluate the isotropic total variation. The

gradient was measured as

$$\left|\nabla v^i\right|_{aniso} = \left|v_x^i\right| + \left|v_y^i\right|,$$

so both derivatives are not coupled. The isotropic variant reads:

$$\left|\nabla v^i\right|_{iso} = \sqrt{(v_x^i)^2 + (v_y^i)^2}.$$

Here, both derivatives are coupled, which might lead to an even better motion estimation. Also from an intuitive point of view, the isotropic variant might be advantageous because jumps in the motion field do not occur in one component independently, but always in both.

Another possible drawback in our models is the restriction to small magnitudes in the motion field. This results from the linearization of the optical flow constraint. We assumed a small time step δ_t between both images, which is simply not true for displacements of larger magnitude. One way out could be the introduction of additional timesteps between our known images. From the image perspective, we would introduce temporal inpainting (see Section 5.4.3), which already gave promising results. From the perspective of motion estimation, the time step δ_t would be small between each of the timesteps such that the optical flow constraint is violated less.

Another approach to this problem is the following: Zack, Pock and Bishof [77] considered for consecutive images I_0, I_1 the nonlinear equation

$$I_0(\boldsymbol{x}) - I_1(\boldsymbol{x} + \boldsymbol{v}) = 0, \tag{7.1}$$

for the velocity field $\boldsymbol{v}$. For a given displacement field $\boldsymbol{v}_0$ they linearized the coupled part in Equation 7.1 near $\boldsymbol{x} + \boldsymbol{v}_0$, hence:

$$\begin{aligned} &I_0(\boldsymbol{x}) - (I_1(\boldsymbol{x} + \boldsymbol{v}_0) + \langle \nabla I_1, \boldsymbol{v} - \boldsymbol{v}_0 \rangle)) = 0 \\ \Leftrightarrow &I_1(\boldsymbol{x} + \boldsymbol{v}_0) + \langle \nabla I_1, \boldsymbol{v} - \boldsymbol{v}_0 \rangle - I_0(\boldsymbol{x}) = 0 \end{aligned} \tag{7.2}$$

This constraint is now taken as data fidelity in a variational model, together with an isotropic total variation regularizer. The problem can then be solved using an iterative warping scheme where the result from a lower resolution is used as $\boldsymbol{v}_0$ in the next step.

7.2. Joint Motion Estimation and Image Reconstruction

The techniques established in the motion estimation chapter have been extended in Chapter 5. Having in mind sequences of images with certain underlying dynamics, we took the Rudin-Osher-Fatemi model for image reconstruction and incorporated an optical flow constraint to combine image data with the dynamics beneath. Moreover, the joint reconstruction and motion estimation is motivated by the sensitivity towards noise discussed in Section 4.6.6. Besides the optical flow constraint, we added a regularizing term for the velocity field to the ROF model and were able to extract the estimated motion field and reconstructed image sequence as a minimizer of the variational model simultaneously. Not only the optical flow constraint has been used as a dynamic constraint, but also the mass preservation constraint. Besides this, different regularizers for the velocity fields were presented. We proved existence of minimizers for slightly generalized models using the Aubin–Lions Lemma [66] to obtain a compactness result. Numerical schemes, again based on the primal-dual framework, were established. Finally, we showed that our model outperforms the classical ROF model, consequently, incorporating the dynamics is justified. Another interesting feature we evaluated was temporal inpainting. Therefore, we used our models to inpaint complete timesteps in a given incomplete image sequence.
Similar to the proposed optical flow models, we restricted to motion fields of small magnitude. To cover larger velocities, we could use the linearization from Equation (7.2) also in the joint models. Moreover, the models could be equipped with more advanced regularizers both for the image sequence u and the velocity field $\boldsymbol{v}$.

7.3. Application

In the application part of this thesis, we covered the problem of image segmentation. Using level set approaches, we discussed the Geodesic Active Contour model as well as the Chan-Vese model and derived a scheme for the numerical realization. The Geodesic Active Contour model was later used for segmentation of vertebrae from images. For further data processing, we introduced a set of techniques for feature extraction from these images, most of them based on level set functions.
The Chan-Vese model was used for segmentation of cells from image sequences. Here, also the motion of the cells and interior components was estimated as well as automatic detection of membrane deformations, so-called blebbs.

In a subsequent chapter, we used our models for joint motion estimation and image reconstruction for the previously introduced image sequences of cells. The images were simultaneously denoised while underlying motion was estimated.

A Appendix

A.1. Additional Mathematical Preliminaries

A.1.1. Basic Definitions

Let, if not explicitly defined differently, for this thesis $\Omega \subset \mathbb{R}^n$ be an open and bounded domain with Lipschitz boundary.

Definition A.1.1. Multi-index Notation
For simplifying the notation of derivatives, we introduce the multi-index notation. A multi-index of degree $n \in \mathbb{N}$ is a tuple

$$\alpha = (\alpha_1, \ldots, \alpha_n), \alpha_i \in \mathbb{N}_0.$$

We define the sum and difference of two multi-indices α, β as the component-wise operation

$$\alpha \pm \beta = (\alpha_1 \pm \beta_1, \ldots, \alpha_n \pm \beta_n),$$

and the absolute value as the sum of all components

$$|\alpha| = \alpha_1 + \ldots + \alpha_n.$$

Using multi-index notation, we can easily formulate arbitrary partial derivatives

$$\partial^\alpha := \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}.$$

Definition A.1.2. Weak Derivative
The weak derivative is a generalization of the classical concept of differentiation, but only requires the underlying function to be integrable.
Let be a function operating on Ω and α be a multi-index. We denote v as the weak derivative with respect to α if

$$\int_\Omega u \partial^\alpha \varphi \, \mathrm{d}x = (-1)^{|\alpha|} \int_\Omega v \varphi \, \mathrm{d}x, \quad \forall \varphi \in C_0^\infty(\Omega).$$

Example A.1.3. Absolute Value Function
The most common example for weak derivatives is the absolute value function. Let $\Omega = (-1, 1) \in \mathbb{R}, f(x) = |x|, \varphi \in C_0^\infty$. We are now able to easily calculate the weak derivative

$$\begin{aligned}\int_{-1}^1 \varphi_x |x| \, \mathrm{d}x &= -\int_{-1}^0 \varphi_x x \, \mathrm{d}x + \int_0^1 \varphi_x x \, \mathrm{d}x = \int_{-1}^0 \varphi \, \mathrm{d}x - \int_0^1 \varphi \, \mathrm{d}x - x\varphi\Big|_{-1}^0 + x\varphi\Big|_0^1 \\ &= \int_{-1}^0 \varphi \, \mathrm{d}x - \int_0^1 \varphi \, \mathrm{d}x = -\int_{-1}^1 \varphi sgn(x) \, \mathrm{d}x\end{aligned}$$

Definition A.1.4. Test Function
Functions $\varphi : \Omega \to \mathbb{R}$ with compact support, which are indefinitely often differentiable are called *test functions*. The space of these test functions is given by:

$$\mathcal{C}_0^\infty(\Omega) := \left\{ \varphi \in \mathcal{C}^\infty(\Omega) : supp(\varphi) \subset \Omega \text{ is compact } \right\}$$

Definition A.1.5. Distribution
A distribution is a bounded linear functional l on the space of test functions $C_0^\infty(\Omega)$. We write $\langle l, \varphi \rangle$ for the functional l applied to $\varphi \in C_0^\infty(\Omega)$.

Example A.1.6.
Each function $u \in L^p(\Omega)$ creates a distribution by

$$\langle u, \varphi \rangle = \int_\Omega u\varphi dx.$$

Not only L^p functions form distributions, but also more general objects like the Dirac δ-distribution. It is represented by the formal properties

$$\delta(x) := \begin{cases} 0 & \text{if } x \neq 0 \\ \infty & \text{if } x = 0 \end{cases},$$

and $\int_\Omega \delta(x)dx = 1$. The corresponding linear functional is now defined as

$$\delta(\varphi) = \langle \delta, \varphi \rangle := \varphi(0)$$

Having generalized the definition of a function, the question of a derivative directly arises. This can be done analogously to the weak derivative.

Definition A.1.7. Distributional Derivative
Let η be a distribution and α be a multi-index. We call v the *distributional derivative* of order $|\alpha|$ if

$$\langle v, \varphi \rangle = (-1)^{|\alpha|} \langle \eta, \partial^\alpha \varphi \rangle, \quad \forall \varphi \in C_0^\infty(\Omega)$$

This definition is consistent with the classical derivative. Consider a distribution created by $u \in L^p$. Then we calculate the distributional derivative

$$\langle u', \varphi \rangle = -\langle u, \varphi' \rangle = -\int_\Omega u\varphi' dx = \int_\Omega u'\varphi dx,$$

and observe that $\langle u, \cdot \rangle$ coincides with the distribution created by u'.

Example A.1.8. Heaviside Function
Very interesting in this context is the fact that the distributional derivative yields a derivative for functions which do not have a derivative (neither classical, nor weak).

For example the Heaviside function

$$\Theta : \mathbb{R} \to \mathbb{R}, \quad \Theta(x) := \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}, \tag{A.1}$$

has no derivative in the classical sense. Now, calculating the distributional derivative

$$\begin{aligned} \langle \partial_x \Theta, \varphi \rangle &= - \langle \Theta, \partial_x \varphi \rangle = - \int_{\mathbb{R}} \Theta(x) \varphi(x) dx = \int_0^\infty \varphi(x) \\ &= \varphi(0) - \lim_{x \to \infty} \varphi(x) = \varphi(0) = \langle \delta, \varphi \rangle , \end{aligned}$$

yields the Dirac δ–distribution. Here, we used the compact support of test functions and the definition of the Dirac δ–distribution $A.1.5$.

A.1.2. Inequalities

Lemma A.1.9. Hölder's Inequality
Let Ω be a measurable set and $p, q \in \mathbb{N}$ satisfying $\frac{1}{p} + \frac{1}{q} = 1$. For functions $f \in L^p(\Omega)$ and $f \in L^q(\Omega)$ we have $fg \in L^1(\Omega)$ and we obtain the inequality

$$\|fg\|_1 \leq \|f\|_p \|g\|_q .$$

Proof. Let $\|f\| > 0, \|g\| > 0$ (otherwise the left-hand side has measure zero a.e.) and $\|f\| < \infty, \|g\| < \infty$ (otherwise the right-hand side is infinite). Let us furthermore assume $p, q \in (1, \infty)$ (otherwise the proof is trivial). Now, dividing f by $\|f\|$ and g by $\|g\|$ we can assume $\|f\| = 1 = \|g\|$.
For $a, b > 0$ Young's inequality states that

$$ab \leq \frac{a^p}{p} + \frac{b^p}{q},$$

applied to our problem we obtain

$$\left|f(s)g(s)\right| \leq \frac{\left|f(s)\right|^p}{p} + \frac{\left|g(s)\right|^q}{q}, s \in \Omega.$$

Now, integrating both sides yields the required

$$\|fg\|_1 \leq \frac{1}{p} + \frac{1}{p} = 1.$$

□

Lemma A.1.10. Poincaré Inequality
Let $\Omega \in \mathbb{R}^n$ be a measurable bounded set with Lipschitz boundary and let $1 \leq p \leq \infty$. Then there exists $C = C(\Omega, p)$ such that for $u \in W^{1,p}(\Omega)$

$$\|u - u_\Omega\|_{L^p(\Omega)} \leq C \|\nabla u\|_{L^p(\Omega)},$$

where

$$u_\Omega = \frac{1}{|\Omega|} \int_\Omega u(x) \, \mathrm{d}x.$$

Proof. Can be found in [3, Theorem 6.30]. □

Lemma A.1.11. L^1 and L^2 Inequality
Let Ω be a measurable set. Then we have the following inequality between L^1 and L^2 norm:

$$\|u\|_{L^1} \leq \sqrt{|\Omega|} \, \|u\|_{L^2}.$$

Proof. We can prove this inequality by simply writing down the norm and using the Cauchy-Schwartz inequality [1]:

$$\|u\|_{L^1} = \int_\Omega |u| = \int_\Omega |u| \cdot 1 = \langle u, 1 \rangle \leq \|u\|_{L^2} \, \|1\|_{L^2} = \sqrt{|\Omega|} \, \|u\|_{L^2}$$

□

A very useful estimation is given by the Minkowski inequality, which can be seen as the triangle inequality for L^p-spaces.

Lemma A.1.12. Minkowski Inequality
Let Ω be a measure space, $1 \leq p \leq \infty <$ and $f, g \in L^p(\Omega)$. Then $f + g$ fulfills the triangle inequality for L^p spaces, namely

$$\|f + g\|_p \leq \|f\|_p + \|g\|_p.$$

Proof. Let $1 < p < \infty$ (i): We first show that $f + g$ has a finite p–norm, if f and g both do. We use the fact that the function x^p is convex over $\mathbb{R}^+$ for $p > 1$, so it follows

$$\left|\frac{1}{2}f + \frac{1}{2}g\right|^p \leq \left|\frac{1}{2}|f| + \frac{1}{2}|g|\right|^p \leq \frac{1}{2}|f|^p + \frac{1}{2}|g|^p \Leftrightarrow |f+g|^p \leq 2^{p-1}(|f|^p + |g|^p).$$

Concerning $\|f+g\|_p$, we can now estimate this part using the general triangle inequality and the Hölder inequality with Hölder conjugates $\frac{1}{p}$ and $1 - \frac{1}{p}$:

$$\begin{aligned}
\|f+g\|_p^p &= \int |f+g|^p \,\mathrm{d}\mu = \int |f+g|\,|f+g|^{p-1}\,\mathrm{d}\mu \\
&\underbrace{\leq}_{\text{Triangle}} \int (|f|+|g|)\,|f+g|^{p-1}\,\mathrm{d}\mu = \int |f|\,|f+g|^{p-1}\,\mathrm{d}\mu + \int |g|\,|f+g|^{p-1}\,\mathrm{d}\mu \\
&\underbrace{\leq}_{\text{Hölder}} \left(\left(\int |f|^p\,\mathrm{d}\mu\right)^{\frac{1}{p}} + \left(\int |g|^p\,\mathrm{d}\mu\right)^{\frac{1}{p}}\right)\left(\int |f+g|^{(p-1)\frac{p}{p-1}}\,\mathrm{d}\mu\right)^{1-\frac{1}{p}} \\
&= (\|f\|_p + \|g\|_p)\frac{\|f+g\|_p^p}{\|f+g\|_p}.
\end{aligned}$$

Now, multiplying both sides with $\frac{\|f+g\|_p}{\|f+g\|_p^p}$ yields the required

$$\|f+g\|_p \leq \|f\|_p + \|g\|_p.$$

□

A.2. Horn-Schunck Discretization

In the following section we will address discretization scheme proposed by Horn and Schunck in 1981 [43]. For the discretization of the image derivatives they use so-called *cell-centered gradients*. We will prove that this discretization indeed gives a consistent scheme, but on the other hand must not be used for a transport equation.

Let us recall the $L^2 - L^2$ optical flow model for two dimensions from Section 4.5.2:

$$\arg\min_{\boldsymbol{v}} \frac{1}{2}\|u_t + \nabla u \cdot \boldsymbol{v}\|_2^2 + \frac{\alpha}{2}\sum_{i=1}^{2}\|\nabla v_i\|_2^2,$$

For the numerical implementation we need consistent approximations for u_t, u_x and u_y which refer to the same point in the discrete time/space domain. Since we usually es-

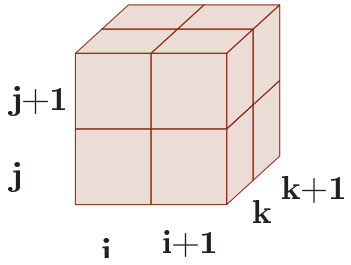


Figure A.1.: Derivatives of U are calculated in the center of a cube, using eight surrounding measurements (from [43])

timate motion only between two consecutive images, a standard forward- or backward difference will lead to inconsistencies. That is why we use so-called cell-centered differences, which were proposed by Horn and Schunck in their original paper [43]. Figure *A*.1 shows a cube formed by the eight measurements at points

$$u^{i,j,k}, u^{i+1,j,k}, u^{i,j+1,k}, u^{i,j,k+1}u^{i+1,j+1,k}, u^{i+1,j,k+1}, u^{i,j+1,k+1}, u^{i+1,j+1,k+1}.$$

which are used to approximate the derivatives u_x, u_y, u_t. Starting with the time-derivative u_t, we use linear interpolation for every image $u^{\cdot,\cdot,t}$ and $u^{\cdot,\cdot,t+1}$ to the center of the four points $(i,j),(i+1,j),(i,j+1),(i+1,j+1)$. To approximate u_t, we use linear interpolation to get

$$\begin{aligned} u^{x+\frac{1}{2},y+\frac{1}{2},z} &\approx \frac{1}{4}(u^{x,y,z} + u^{x+1,y,z} \\ &\quad + u^{x,y+1,z} + u^{x+1,y+1,z}) \\ u^{x+\frac{1}{2},y+\frac{1}{2},z+1} &\approx \frac{1}{4}(u^{x,y,z+1} + u^{x+1,y,z+1} \\ &\quad + u^{x,y+1,z+1} + u^{x+1,y+1,z+1}) \end{aligned}$$

and use a central difference to calculate the derivative

$$\begin{aligned} u_t^{x+\frac{1}{2},y+\frac{1}{2},z+\frac{1}{2}} &\approx u^{x+\frac{1}{2},y+\frac{1}{2},z} - u^{x+\frac{1}{2},y+\frac{1}{2},z+1} \\ &\approx \frac{1}{4}(u^{x,y,z} + u^{x+1,y,z} \\ &\quad + u^{x,y+1,z} + u^{x+1,y+1,z} \\ &\quad - u^{x,y,z+1} - u^{x+1,y,z+1} \\ &\quad - u^{x,y+1,z+1} - u^{x+1,y+1,z+1}). \end{aligned}$$

The derivatives U_x and U_y are being calculated in the same way and we get

$$\begin{aligned}
u_x^{x+\frac{1}{2},y+\frac{1}{2},z+\frac{1}{2}} \approx \frac{1}{4}(&u^{x+1,y,z} + u^{x+1,y+1,z} \\
&+ u^{x+1,y,z+1} + u^{x+1,y+1,z+1} \\
&- u^{x,y,z} - u^{x,y+1,z} \\
&- u^{x,y,z+1} - u^{x,y+1,z+1}) \\
u_y^{x+\frac{1}{2},y+\frac{1}{2},z+\frac{1}{2}} \approx \frac{1}{4}(&u^{x,y+1,z} + u^{x+1,y+1,z} \\
&+ u^{x,y+1,z+1} + u^{x+1,y+1,z+1} \\
&- u^{x,y,z} - u^{x+1,y,z} \\
&- u^{x,y,z+1} - u^{x+1,y,z+1})
\end{aligned}$$

Since the derivatives u_x, u_y, u_t only act as multipliers for v_i, they can be precalculated. This gives a consistent scheme, as the following lemma proves:

Lemma A.2.1. Consistency of cell-centered gradient
Let $u \in C^2(\Omega), u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ be an arbitrary function. Then the cell-centered gradient $\nabla^{cc} u := (\partial_x^{cc} u, \partial_y^{cc} u)^T$ with

$$\begin{aligned}
\partial_x^{cc} u &:= \frac{u(x+h,y+h) + u(x+h,y) - u(x,y+h) - u(x,y)}{2h} \\
\partial_y^{cc} u &:= \frac{u(x+h,y+h) + u(x,y+h) - u(x+h,y) - u(x,y)}{2h}
\end{aligned}$$

yields a consistent approximation for the gradient ∇u of order 1.

Proof. We will prove in the following that $\partial_x^{cc} u$ yields a consistent approximation of the x-derivative of u. The proof for u_y and a arbitrary dimension can be done in an analogue way.
Using Taylor expansion we get

$$\begin{aligned}
&\frac{u(x+h,y+h) + u(x+h,y) - u(x,y+h) - u(x,y)}{2h} \\
=&\frac{u(x+h,y) + hu_y(x+h,y) + u(x,y) + hu_x(x,y) - u(x,y) - hu_y(x,y) - u(x,y) + O(h^2)}{2h} \\
=&\frac{u(x,y) + hu_x(x,y) + hu_y(x,y) + u(x,y) + hu_x(x,y) - u(x,y) - hu_y(x,y) - u(x,y) + O(h^2)}{2h}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{2hu_x(x,y) + O(h^2)}{2h} \\
&= u_x(x,y) + O(h)
\end{aligned}
$$

□

Hence, this discretization can be used in the context of flow estimation from images. But in our joint models the requirement is even higher. The image discretization is not only used for motion estimation, but also as a transport equation for the image data u. Unfortunately the cell-centered gradient does not yield a stable discretization for the transport equation:

Lemma A.2.2. Transport instability of cell-centered approximation
Consider the one-dimensional transport equation

$$u_t + vu_x = 0, \tag{A.2}$$

then the cell-centered gradient does not yield a stable discretization for the numerical realization of Equation $(A.2)$.

Proof. We start with the discretized version of Equation $(A.2)$ using the cell-centered derivatives from Lemma $A.2.1$.

$$\frac{u_{j+1}^{k+1} + u_j^{k+1} - u_{j+1}^k - u_j^k}{2h_t} + v\frac{u_{j+1}^{k+1} + u_{j+1}^k - u_j^{k+1} - u_j^k}{2h_x} = 0, \tag{A.3}$$

with h_t, h_x time resp. spatial step sizes and v as the motion field. Defining $c := \frac{vh_t}{h_x}$ we have

$$u_{j+1}^{k+1}(1+c) + u_j^{k+1}(1-c) + u_{j+1}^k(c-1) + u_j^k(-1-c) = 0. \tag{A.4}$$

Let now $u_\cdot^k$ consist of some signal z^k with an arbitrary small sinus-like pertubation

$$u_j^k = z^k e^{ijh_x}, a := e^{ih_x}.$$

Then Equation $(A.4)$ reduces to

$$
\begin{aligned}
0 &= z^{k+1}a(1+c) + z^{k+1}(1-c) + z^k a(c-1) + z^k(-1-c) \\
\leftrightarrow z^{k+1} &= \frac{(a-1)(1-c)+2}{(a-1)(1+c)+2} z^k.
\end{aligned}
$$

Let us recall the definition of c and consider the three possible cases $c = 0, c > 0, c < 0$. We assume furthermore $a > 1$. For $c = 0$ we have $z^{k+1} = z^k$ and no instability occurs. But $c = 0$ already implies $v = 0$, thus we are not solving a transport equation anymore. The case $c > 0$ implies $v > 0$ and we have

$$\frac{(a-1)(1-c)+2}{(a-1)(1+c)+2} < 1,$$

which is ok because the pertubation does not increase. The problematic case is $c < 0$, which implies $v < 0$. Here we have

$$\frac{(a-1)(1-c)+2}{(a-1)(1+c)+2} > 1,$$

an the pertubation blows up over time. □

A.3. Joint Motion Estimation and Image Reconstruction

Figures $A.2$ and $A.4$ contain the motion estimation results for Rubber Whale and Hydrangea. Besides that, we find in Figures $A.3$ and $A.5$ results for the image denoising.

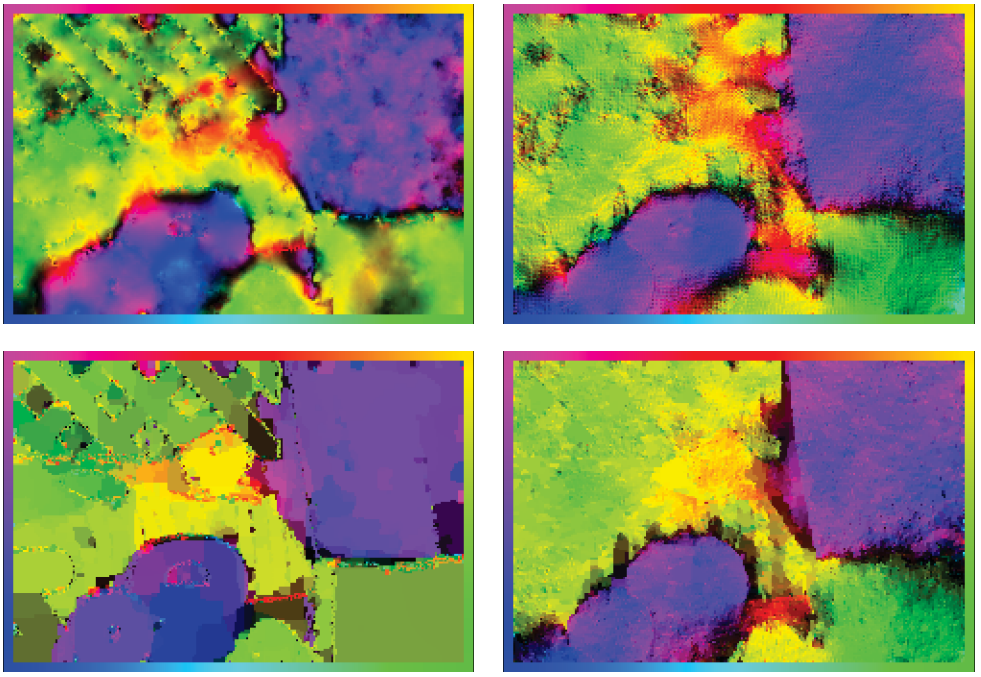


Figure A.2.: Estimated motion field for Rubber Whale example between timesteps t_1 and t_2 from TV-L^2 optical flow (top-left), TV-L^2 mass preservation (top-right), TV-TV optical flow (bottom-left), TV-TV mass preservation (bottom-right)

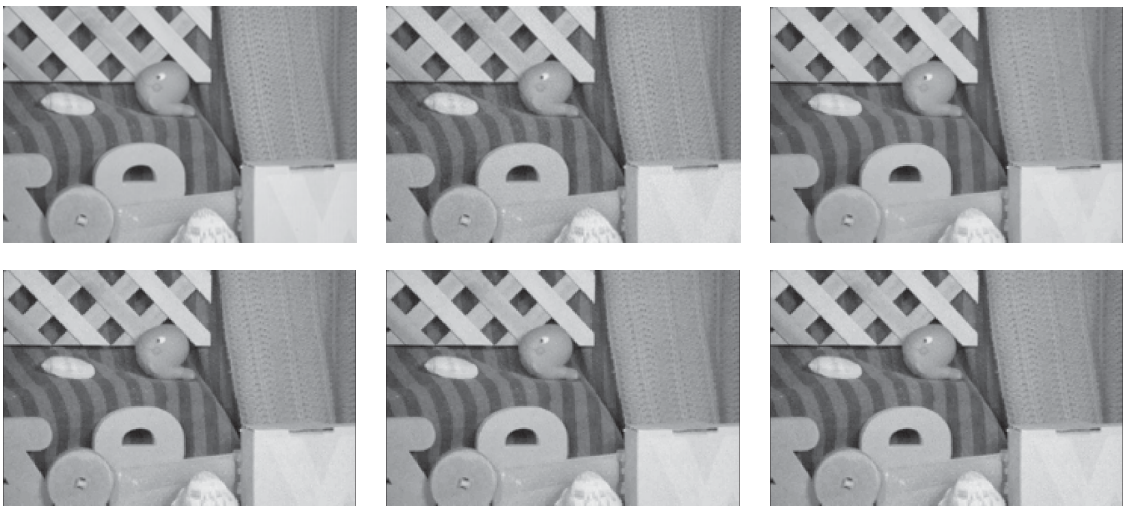


Figure A.3.: Reconstructed images for Rubber Whale example. Top-Left: Ground truth image, Top-Right: Noisy image, Middle-Left: $TV - L^2$ Optical Flow reconstruction, Middle-Right: $TV - L^2$ Mass Preservation reconstruction, Bottom-Left: $TV - TV$ Optical Flow reconstruction, Bottom-Right: $TV - TV$ Mass Preservation reconstruction

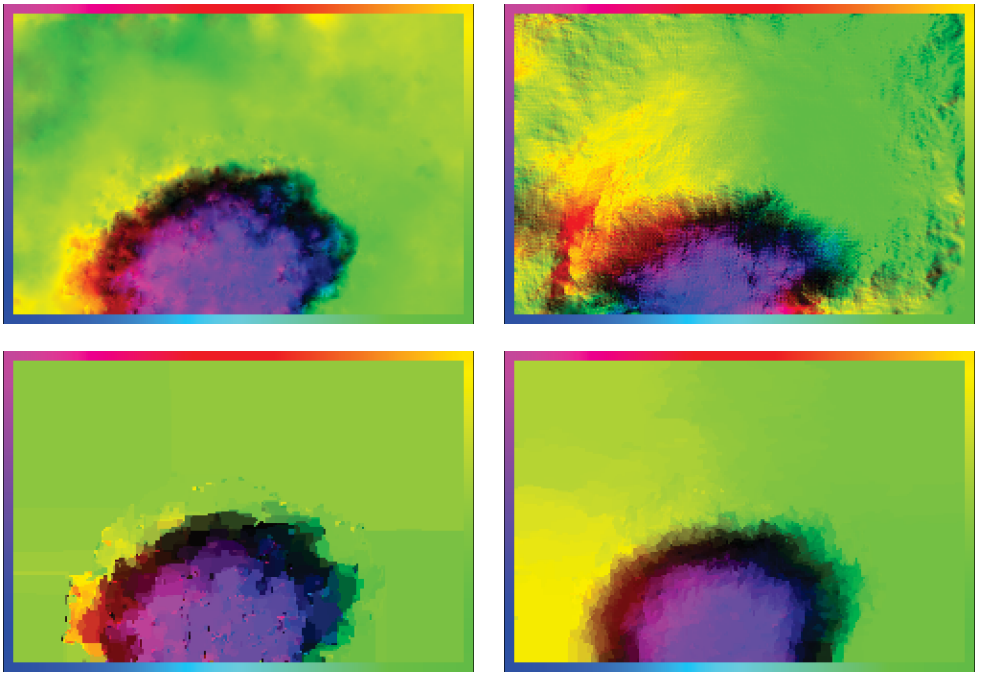


Figure A.4.: Estimated motion field for Hydrangea example between timesteps t_1 and t_2 from TV-L^2 optical flow (top-left), TV-L^2 mass preservation (top-right), TV-TV optical flow (bottom-left), TV-TV mass preservation (bottom-right)

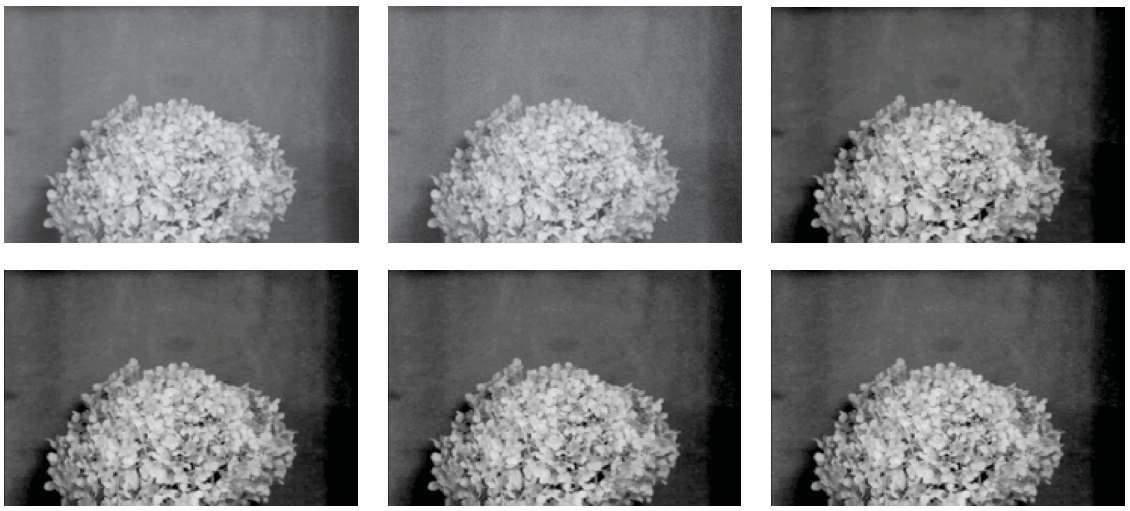


Figure A.5.: Reconstructed images for Hydrangea example. Top-Left: Ground truth image, Top-Right: Noisy image, Middle-Left: $TV - L^2$ Optical Flow reconstruction, Middle-Right: $TV - L^2$ Mass Preservation reconstruction, Bottom-Left: $TV - TV$ Optical Flow reconstruction, Bottom-Right: $TV - TV$ Mass Preservation reconstruction

A.4. Level-Set Converter

In Chapters 3 and 6, the level set formulation of a segmentation is widely used. Not only do we evolve a level set function in the Active Contour (see Section 3.2.1) and Chan-Vese (see Section 3.2.2) algorithms directly, we also use level set representations for later feature extraction of vertebra (see Section 3.3.1). In this context, the level set function often has to be calculated to a region-based segmentation and reversely. This section is dedicated to this problem and supplies an easy formula for the former problem and a fast $O(n)$ algorithm for the later one.

A.4.1. Motivation

As mentioned before, there are two ways to represent a segmentation. The first one works with a region-function RF. Its counterpart uses a signed distance function Φ, where the boundary is the zero level-set. Segmentation algorithms result in different representation types, so we would like to convert from region function to level-set or vice versa (this is only possible if we have a total number of 2 regions).

A.4.2. Level-Set to Region Function

We start with the trivial problem of converting a level-set segmentation to a region function. We remember that the signed distance function Φ divides the image into two regions, namely:

$$x \in R_1 \Longleftrightarrow \Phi(x) > 0$$
$$x \in R_2 \Longleftrightarrow \Phi(x) < 0$$

So we simply define the region function as:

$$RF(x) = \begin{cases} 1, & \text{if } \Phi(x) > 0 \\ 2, & \text{if } \Phi(x) \leq 0 \end{cases}$$

So this conversion can be performed by simply moving through the grid and changing the level-set function to one or two.

A.4.3. Region Function to Level-Set

The more problematic conversion is from region-function to level-set. Starting with the region function we have to calculate the distance to the initial boundary Γ on each grid-point. To overcome this problem we will make use of an idea introduced by Sethian in 1988 [63]. In the later part of this chapter we will present a faster $O(n)$ version of Sethian's algorithm.

Fast Marching

Introduction:
The fast marching algorithm is used for computing the position of monotonically advancing fronts. In detail this means that we start with an initial curve Γ which propagates with a constant speed F towards its normal direction. See Figure $A.6$ for an example. In our application we interpret Γ as a zero level-set of a function $\Phi(x, t = 0)$. So let $y \notin \Gamma$ be an arbitrary point and

$$\Phi(y, t = 0) = \pm d$$

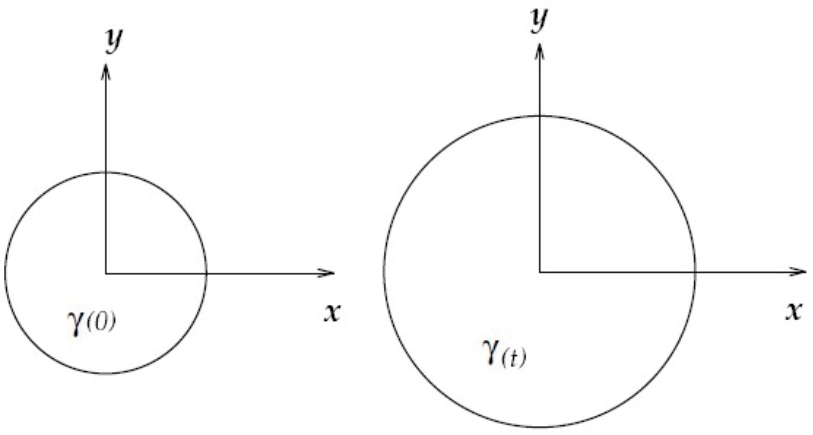


Figure A.6.: Curve moving in time (from [63])

where $\pm d$ is the Euclidean distance to the curve. Applying the chain rule leads to an equation for the time-dependent propagation of Φ [54]

$$\Phi_t + F\,|\nabla\Phi| = 0$$

with the given initial curve

$$\Gamma = \Phi(x, t = 0)$$

Thus for the calculation of Φ we have to solve an initial value problem for a partial differential equation. Further properties of the level set function can be found in [54]. Here we will name just two of them:

- The normal vector of Φ is given by: $\vec{n} = \frac{\nabla\Phi}{|\nabla\Phi|}$
- Calculation of Φ in three dimensions does not change the equations above

Of course in our concrete case, that is the conversion from a region function based on a level set representation, we set the speed $F \equiv 1$ because for every point $x \in \Omega$ we would like to calculate the shortest distance to Γ.

Numerical Approximation:
For the numerical calculation of Φ, we can use an upwind scheme [54]. For the 2-dimensional case this yields

$$\Phi_{ij}^{n+1} = \Phi_{ij}^{n} - \Delta t(max(D_{ij}^{-x}\Phi, 0)^2 + min(D_{ij}^{+x}\Phi, 0)^2 + max(D_{ij}^{-y}\Phi, 0)^2 + min(D_{ij}^{+y}\Phi, 0)^2)$$

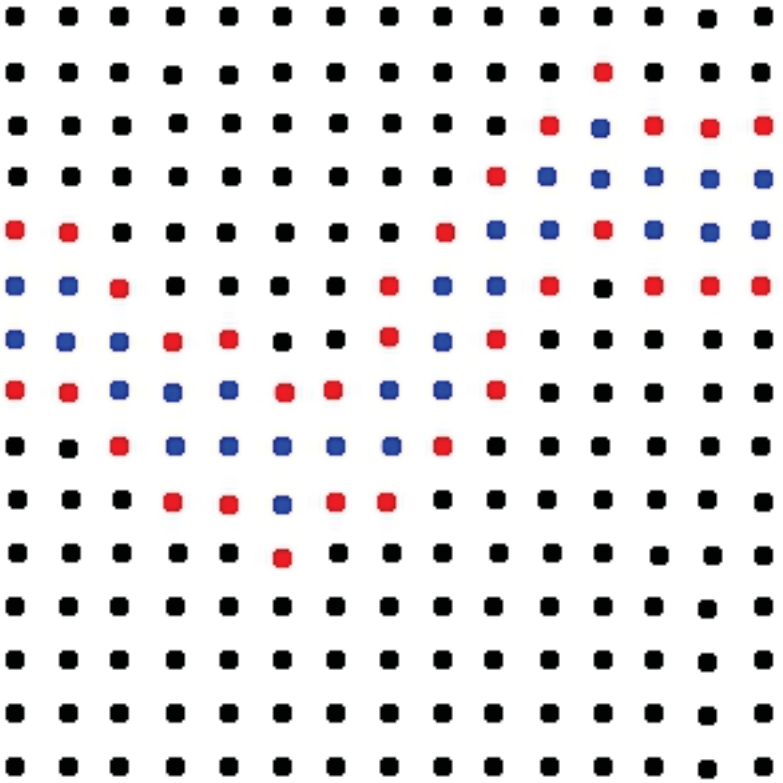


Figure A.7.: Narrow Band: Black far; Red alive; Blue done

where $D_{ij}^{+x}\Phi$ represents for the forward difference:

$$D_{ij}^{+x}\Phi = \frac{\Phi_{i+1,j} - \Phi_{i,j}}{\Delta x}$$

So for the calculation of Φ we would have to solve a large system of equations in every timestep.

Narrow Band:
An alternative idea is to limit the calculation to a small neighborhood of the zero level set. We call this small neighborhood a *narrow band*. The central idea is to divide the grid points into 3 categories:

- 'alive': Point is part of the current calculation
- 'done': Distance already calculated
- 'far': Point not part of the *narrow band* $\rightarrow$ no calculation

Using this type of calculation the operation count for a three dimensional matrix with

n^3 drops to $O(kn^2)$, where k depends on the size of the narrow band. Setting the size of the *narrow band* to just one cell leads us to the

Fast Marching Algorithm:
We assume a 2-dimensional curve that moves forward in time with speed F. Let $T(x, y)$ be the time until the curve hits the point (x, y). The correlation between T and F is clear. The faster the curve moves the smaller T is. They are inversely proportional to each other:

$$|\nabla T|\, F = 1$$

For a constant speed $F = 1$ we obtain:

$$|\nabla T| = 1 \tag{A.5}$$

Rouy and Tourin [59] introduced an iterative algorithm for the discretization of Equation $A.5$:

$$max(max(D_{ij}^{-x}T, 0), -min(D_{ij}^{+x}T, 0))^2 + max(max(D_{ij}^{-y}T, 0), -min(D_{ij}^{+y}T, 0))^2 = 1$$

We remember that T has a special property. The front propagates 'one-way', so the values of T increase monotonically. The algorithm calculates T in an increasing order. We initialize the algorithm with the zero-level-set and tag the surrounding points as *alive*. From the *alive* points we extract the point with the smallest T value and tag this point as *frozen*. The surrounding *far* points are added to the narrow band now. Afterwards we extract one point from the narrow band and so on ...
The detailed algorithm follows. We assume a special case where the front starts at the y-axis and moves upwards through a N by N grid with speed 1.

```
1. Initialize
(a) Let A be the set of all grid points {i,j=1}; set T(i,1) = 0.0
for all points in A
(b) Let NARROWBAND be the set of all grid points {i,j=2},
T(i,1) = dy for all points in NARROWBAND
(c) Let FARAWAY bet the set of all grid points {i,j>2},
```

```
set T(i,j) = Inf for all points in FARAWAY
2. MARCHING FORWARD
(a) Begin Loop: Let (iMIN,jMIN) be the point in NARROWBAND
with the smallest value for T
(b) Add the point (iMIN,jMIN) to A; remove it from NARROWBAND
(c) Tag as neighbors any points (iMIN-1,jMIN),(iMIN+1,jMIN),
(iMIN,jMIN-1),(iMIN,jMIN+1) that are either in NARROWBAND
or FARAWAY. If the neighbor is in FARAWAY, remove it from
that list and add it to the set NARROWBAND
(d) Recompute the values of T at all neighbors according to
the equation selecting the largest possible solution
to the quadratic equation
(e) Return to top of Loop:
```

A proof of work can be found in [54].

Conclusion:
We have found an algorithm that allows us to convert a region-based representation into a level-set representation. Our main problem is the runtime. A runtime of $O(N^2)$ for N grid points becomes problematic very fast. Especially for 3D-datasets (for example MRT dataset from human torso) the runtime is unacceptable large. So we dedicate ourselves to a faster $O(N)$ algorithm now.

O(n) Fast Marching Algorithm:

Motivation:
Yatziv, Bartesaghi and Sapiro introduced an $O(N)$-implementation of the fast marching algorithm [76]. Central idea is the use of special data structures. The classic fast marching algorithm computes the distances point by point. All alive-points are kept in a list and we extract that element from the list that has the overall shortest distance. The used list type is a so-called *Priority Queue*. This list type sorts the elements by their priority (e.g. distance). So in every iteration step we just have to extract the first element in the list.
So far the algorithm works very well. But now comes the crucial point: Every time we freeze a point and fix its distance, we have to recalculate the neighboring points. That means we have to remove the neighbors from the *Priority Queue* (complexity $O(N)$) and reinsert them $O(log(N))$.

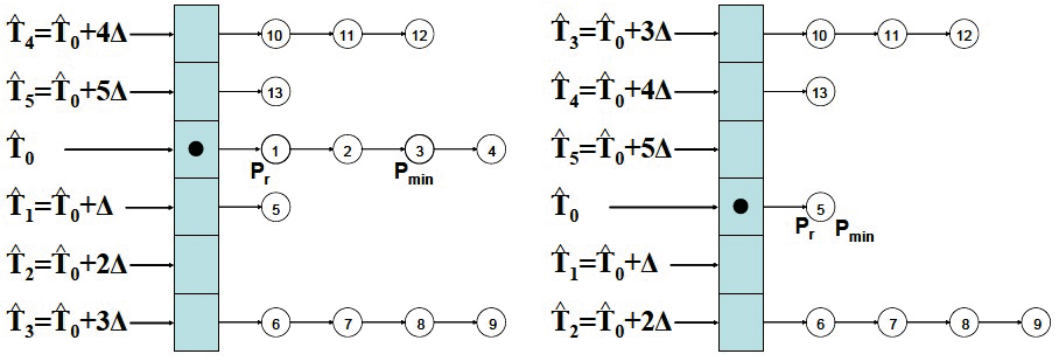


Figure A.8.: Circular Array at two different timesteps (from [76])

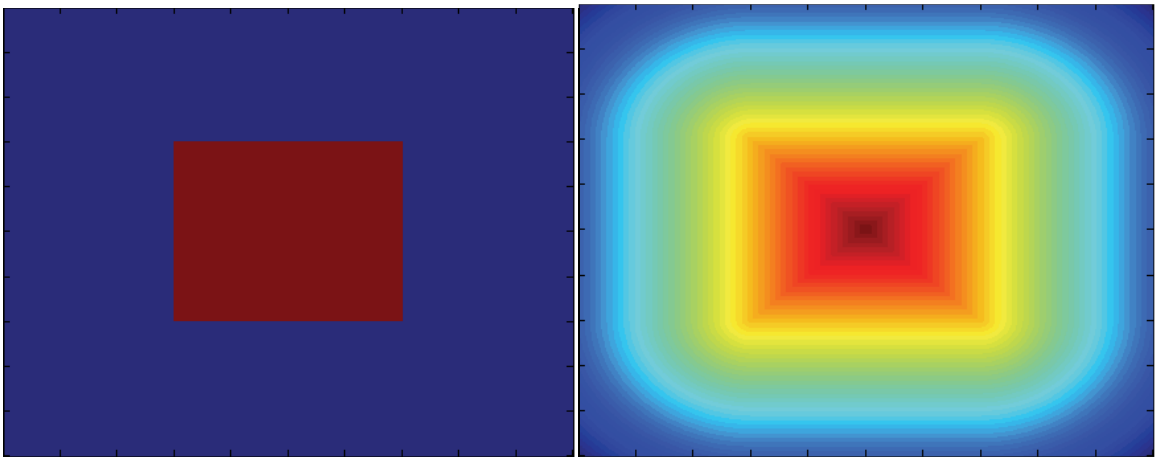
Untidy Priority Queue:
The improved algorithm now makes use of a special data structure which reduces the updating complexity to $O(1)$. This is due to the fact that the values of all points in the queue are equal or greater than the latest extracted point. Furthermore the speed of our propagating front is 1, so we have a maximal possible increment I_{max} over the point with the minimal T value currently in the queue. So all the T values in the queue have a fixed range, so we can use a data structure based on a *circular array* (See figure $A.8$). Each entry of the circular array contains a list of points with 'similar' T value. For a constant speed of 1 our possible range I_{max} is limited by 1, so any point in the narrow band has a distance

$$d \in [T_0, T_0 + 1]$$

where T_0 is the distance of the latest point. We divide our unit interval into K intervals. Every interval $i \in \{1, \ldots, K\}$ is represented by a FIFO list that contains T values in range $\left[T_0 + \frac{i-1}{K}, T_0 + \frac{i}{K}\right]$. If we like to insert or remove a specific element now, we will have to calculate the list number first. This can be done by a simple modulo operation. If we furthermore assume that every list contains an average number of d elements (can be achieve by dynamically resizing the number K of lists, see [19]), we assure a constant complexity of $O(1)$ for insert and removal.
We can show that the error, caused by the 'untidiness', is of the same order as the original fast marching algorithm, see [76] for further details.
Evaluation:
Table $A.1$ contains a few runtime tests and Figure $A.9$ shows a segmentation in region function and level set representation.

Image	Size	Pixels	Processor	Time
2D Box	100x100	10.000	I7 920: 8*2.6 ghz	0.55 sec
2D Box	1000x1000	1.000.000	I7 920: 8*2.6 ghz	5.41 sec
2D Box	2000x2000	4.000.000	I7 920: 8*2.6 ghz	25.06 sec
2D Box	3000x3000	9.000.000	I7 920: 8*2.6 ghz	63.02 sec
2D Walnut	261x326	85.086	I7 920: 8*2.6 ghz	1.06 sec
3D Walnut	131x163x131	2.797.243	I7 920: 8*2.6 ghz	400.14 sec

Table A.1.: Runtime of Different Examples



(a) Region Function **(b)** Level set

Figure A.9.: Different Representations of the 2D Box example

List of Figures

List of Tables

Bibliography

[1] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Number 55. Courier Corporation, 1964. 219

[2] Robert Acar and Curtis R Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse problems*, 10(6):1217, 1994. 24

[3] Robert A Adams and John JF Fournier. *Sobolev spaces*, volume 140. Academic press, 2003. 27, 28, 219

[4] Edward H Adelson and J Anthony Movshon. Phenomenal coherence of moving visual patterns. *Nature*, 300(5892):523–525, 1982. 86

[5] Alok Aggarwal, Leonidas J Guibas, James Saxe, and Peter W Shor. A linear-time algorithm for computing the voronoi diagram of a convex polygon. *Discrete & Computational Geometry*, 4(1):591–604, 1989. 77

[6] Luigi Ambrosio, Nicola Fusco, and Diego Pallara. *Functions of bounded variation and free discontinuity problems*, volume 254. Clarendon Press Oxford, 2000. 28

[7] G Aubert, R Deriche, and P Kornprobst. Computing optical flow via variational techniques. *SIAM Journal on Applied Mathematics*, 60(1):156–182, 1999. 16, 108

[8] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*, volume 147. Springer, 2006. 36, 55, 83, 84

[9] Jean-Pierre Aubin. Un théorème de compacité. *C. R. Acad. Sci. Paris*, 256:5042–5044, 1963. 26

[10] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. 83, 116, 119

[11] Leah Bar, Benjamin Berkels, Martin Rumpf, and Guillermo Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 16

[12] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994. 116

[13] J-D Benamou, Y Brenier, and K Guittet. The monge–kantorovitch mass transfer and its computational fluid mechanics formulation. *International Journal for Numerical methods in fluids*, 40(1-2):21–30, 2002. 16

[14] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000. 16

[15] M Benning. *Singular Regularization of Inverse Problems: Bregman Distances and their Applications to Variational Frameworks with Singular Regularization Energies*. PhD thesis, PhD thesis, Westfälische Wilhelms-Universität Münster, 2011. 37, 39, 2011. 141

[16] Eric Betzig, George H Patterson, Rachid Sougrat, O Wolf Lindwasser, Scott Olenych, Juan S Bonifacino, Michael W Davidson, Jennifer Lippincott-Schwartz, and Harald F Hess. Imaging intracellular fluorescent proteins at nanometer resolution. *Science*, 313(5793):1642–1645, 2006. 13

[17] Heiko Blaser, Michal Reichman-Fried, Irinka Castanon, Karin Dumstrei, Florence L Marlow, Koichi Kawakami, Lilianna Solnica-Krezel, Carl-Philipp Heisenberg, and Erez Raz. Migration of zebrafish primordial germ cells: a role for myosin contraction and cytoplasmic flow. *Developmental cell*, 11(5):613–627, 2006. 70

[18] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 32, 36

[19] Randy Brown. Calendar queues: a fast 0 (1) priority queue implementation for the simulation event set problem. *Communications of the ACM*, 31(10):1220–1227, 1988. 232

[20] Christoph Brune. *4d imaging in tomography and optical nanoscopy*. PhD thesis, PhD thesis, University of Münster, Germany, 2010. 16, 141

[21] Martin Burger, Andrea CG Mennucci, Stanley Osher, and Martin Rumpf. *Level Set and PDE Based Reconstruction Methods in Imaging*. Springer, 2008. 23, 24

[22] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986. 56

[23] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International journal of computer vision*, 22(1):61–79, 1997. 57

[24] B Cengiz. The dual of the Bochner space $L^p(\mu, E)$ for arbitrary μ. *Turkish Journal of Mathematics*, 22:343–348, 1998. 25

[25] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004. 97

[26] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. 17, 19, 38, 47, 81, 91, 92, 97, 162, 163, 211

[27] Tony F Chan, B Yezrielev Sandberg, and Luminita A Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11(2):130–141, 2000. 52

[28] Tony F Chan and Luminita A Vese. Active contours without edges. *Image processing, IEEE transactions on*, 10(2):266–277, 2001. 55, 57, 60, 62, 63, 205

[29] Guillaume Charras and Ewa Paluch. Blebs lead the way: how to migrate without lamellipodia. *Nature Reviews Molecular Cell Biology*, 9(9):730–736, 2008. 70

[30] Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005. 48

[31] Ivar Ekeland and Roger Temam. Convex analysis and 9 variational problems. 1976. 32, 38

[32] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996. 14

[33] Ernie Esser, Xiaoqun Zhang, and Tony F Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010. 47

[34] Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, 1998. 27

[35] Oliver T Fackler and Robert Grosse. Cell motility through plasma membrane blebbing. *The Journal of cell biology*, 181(6):879–884, 2008. 70

[36] Bernd Fischer and Jan Modersitzki. Ill-posed medicine-an introduction to image registration. *Inverse Problems*, 24(3):034008, 2008. 85

[37] David J Fleet and Allan D Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990. 116

[38] David R Gilland, Bernard A Mair, James E Bowsher, and Ronald J Jaszczak. Simultaneous reconstruction and motion estimation for gated cardiac ect. *Nuclear Science, IEEE Transactions on*, 49(5):2344–2349, 2002. 16

[39] Tom Goldstein, Ernie Esser, and Richard Baraniuk. Adaptive primal-dual hybrid gradient methods for saddle-point problems. *arXiv preprint arXiv:1305.0546*, 2013. 97

[40] Google. https://plus.google.com/+googleselfdrivingcars/videos, 2015. 13

[41] Martin Hanke-Bourgeois. *Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens*. Springer-Verlag, 2006. 95, 168

[42] Stefan W Hell, Steffen Lindek, Christoph Cremer, and Ernst HK Stelzer. Confocal microscopy with an increased detection aperture: type-b 4pi confocal microscopy. *Optics letters*, 19(3):222–224, 1994. 13

[43] Berthold K Horn and Brian G Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981. 16, 91, 139, 142, 143, 168, 220, 221

[44] Ipol. http://www.ipol.im, 2015. 17, 58, 81, 121, 128, 129, 135, 186, 187

[45] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. 52

[46] J.-L. Lions. *Quelques méthodes de résolution des problèmes aux limites non linéaires*. Dunod; Gauthier-Villars, Paris, 1969. 26

[47] Enkeleida Lushi, Hugo Wioland, and Raymond E. Goldstein. Fluid flows created by swimming bacteria drive self-organization in confined suspensions. *Proceedings of the National Academy of Sciences*, 111(27):9733–9738, 2014. 207

[48] Jan Malỳ, David Swanson, and William Ziemer. The co-area formula for sobolev mappings. *Transactions of the American Mathematical Society*, 355(2):477–492, 2003. 22

[49] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980. 56

[50] Norman G Meyers and William P Ziemer. Integral inequalities of poincare and wirtinger type for bv functions. *American Journal of Mathematics*, pages 1345–1360, 1977. 146

[51] Michael Möller. *Multiscale methods for (generalized) sparse recovery and applications in high dimensional imaging*. PhD thesis, 2012. 41

[52] D Mumford and J Shah. Boundary detection by minimizing functionals. *Image understanding*, pages 19–43, 1988. 51, 55, 60

[53] Website of Raymond E. Goldstein. http://www.damtp.cam.ac.uk/user/gold, 2015. 207, 209, 210

[54] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988. 120, 228, 231

[55] Michael Otte and H-H Nagel. Optical flow estimation: advances and comparisons. In *Computer Vision - ECCV'94*, pages 49–60. Springer, 1994. 116

[56] Nils Papenberg, Andrés Bruhn, Thomas Brox, Stephan Didas, and Joachim Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, 2006. 16

[57] R Tyrrell Rockafellar. *Convex Analysis (Princeton Mathematical Series)*, volume 46. Princeton University Press, 1970. 32, 33, 36, 37

[58] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009. 32, 36, 42

[59] Elisabeth Rouy and Agnès Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992. 230

[60] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992. 38, 139, 188

[61] Walter Rudin. *Functional analysis, 1973*. McGraw-Hill, New York, 1973. 34, 35

[62] PJ Schuck, DP Fromm, A Sundaramurthy, GS Kino, and WE Moerner. Improving the mismatch between light and nanoscale objects with gold bowtie nanoantennas. *Physical review letters*, 94(1):017402, 2005. 13

[63] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. 227, 228, 236

[64] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242. ACM, 1998. 82

[65] Huanfeng Shen, Liangpei Zhang, Bo Huang, and Pingxiang Li. A map approach for joint motion estimation, segmentation, and super resolution. *Image Processing, IEEE Transactions on*, 16(2):479–490, 2007. 16

[66] Jacques Simon. Compact sets in the space $L^p(0,T;B)$. *Annali di Matematica pura ed applicata*, 146(1):65–96, 1986. 26, 213

[67] Richard Szeliski. Prediction error as a quality metric for motion and stereo. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 781–788. IEEE, 1999. 119

[68] Luc Tartar. The compensated compactness method applied to systems of conservation laws. In *Systems of nonlinear partial differential equations*, pages 263–285. Springer, 1983. 149

[69] Tiffread. http://www.cytosim.org/other, 2015. 70

[70] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 16

[71] UKM. http://klinikum.uni-muenster.de/index.php?id=unfallchiurgie, 2015. 64, 65, 66, 67, 71

[72] Alessandro Verri and Tomaso Poggio. Motion field and optical flow: Qualitative properties. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(5):490–498, 1989. 82

[73] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004. 188

[74] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. An improved algorithm for tv-l 1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45. Springer, 2009. 139

[75] Hugo Wioland, Francis G. Woodhouse, Jörn Dunkel, John O. Kessler, and Raymond E. Goldstein. Confinement stabilizes a bacterial suspension into a spiral vortex. *Phys. Rev. Lett.*, 110:268102, Jun 2013. 207

[76] Liron Yatziv, Alberto Bartesaghi, and Guillermo Sapiro. O (n) implementation of the fast marching algorithm. *Journal of computational physics*, 212(2):393–399, 2006. 231, 232, 236

[77] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, pages 214–223. Springer, 2007. 16, 139, 212

[78] Xiaoqun Zhang, Martin Burger, and Stanley Osher. A unified primal-dual algorithm framework based on bregman iteration. *Journal of Scientific Computing*, 46(1):20–46, 2011. 47

[79] ZMBE. http://zmbe.uni-muenster.de, 2015. 53, 73, 74, 75, 77, 78, 205, 206, 208