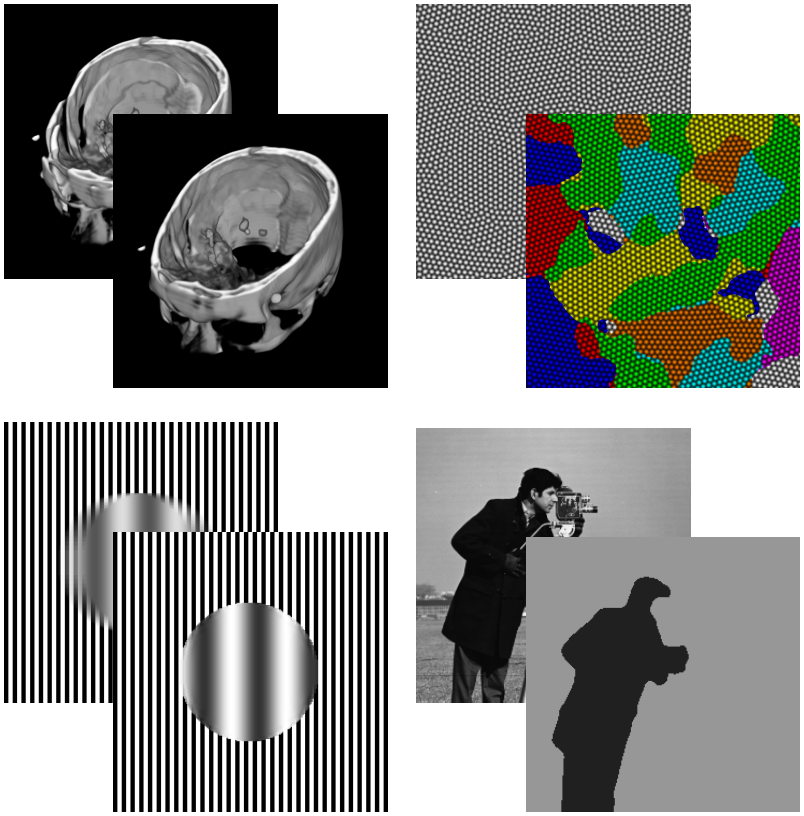
Joint methods in imaging based on diffuse image representations



Dissertation

zur Erlangung des Doktorgrades (Dr. rer. nat.)
der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von Benjamin Berkels
aus Duisburg

Bonn, Februar 2010

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
am Institut für Numerische Simulation

Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn
`http://hss.ulb.uni-bonn.de/diss_online` elektronisch publiziert.

Erscheinungsjahr 2010

1. Referent: Prof. Dr. Martin Rumpf
2. Referent: Prof. Dr. Sören Bartels

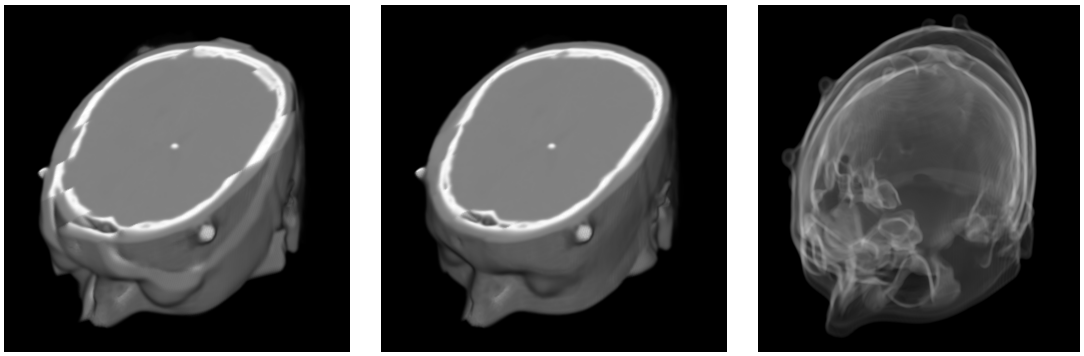
Tag der Promotion: 9. Juli 2010

Zusammenfassung

Diese Dissertation befasst sich mit der Anwendung und der Analyse des Mumford-Shah-Modells im Kontext der Bildverarbeitung.

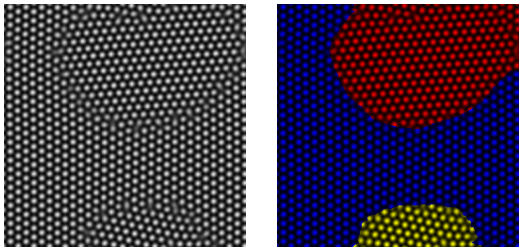
Zunächst wird das Mumford-Shah-Modell selbst in verschiedenen Varianten vorgestellt. Bei dieser Art von Modellen wird eine gegebene Funktion stückweise glatt oder stückweise konstant approximiert, wobei eine besondere Schwierigkeit dabei die Behandlung der Menge der Diskontinuitäten darstellt. Insbesondere für die Numerik sind dazu weitere Modelle notwendig, wobei die grundlegenden Modelle, auf die wir uns in dieser Arbeit stützen, ebenfalls hier erläutert werden. Der Hauptteil der Arbeit befasst sich mit den folgenden vier Verfahren.

Gleichzeitige Kantenerkennung und Registrierung zweier Bilder. Die Registrierung basiert dabei auf den erkannten Kanten. Die Kantenerkennung wiederum wird, basierend auf dem Mumford-Shah-Modell, mit dem Ambrosio-Tortorelli-Modell durchgeführt, welches die Menge der Diskontinuitäten durch ein Phasenfeld approximiert. Die Registrierung durch unser Modell ist vollständig symmetrisch in dem Sinne, dass das Modell die gleiche Registrierung liefert, wenn die Rollen der beiden zu registrierenden Bilder vertauscht werden.



3D CT-CT Registrierung mit symmetrischem Kantenmatching: Initialer Mismatch, berechnete Segmentierung (zur Verdeutlichung mit einer Schnittebene) und gefundene Kanten (von links nach rechts).

Erkennung von Körnern aus atomar aufgelösten Bildern von Metallen oder Metall-Legierungen. Hierbei handelt es sich um ein Bildverarbeitungsproblem aus den Materialwissenschaften, wobei solche Bilder experimentell durch Transmissionselektronenmikroskopie oder durch numerische Simulationen erhalten werden können. Als *Körner* bezeichnet man Materialregionen, in denen die Orientierung des Atomgitters von der Umgebung abweicht. Basierend auf einem Mumford-Shah-artigen Funktional werden die Korngrenzen als Sprungmenge aufgefasst, an der die Orientierung des Atomgitters springt. Neben den Korngrenzen erlaubt das Modell noch die Extraktion einer globalen elastischen Deformation des Atomgitters. Numerisch wird die Sprungmenge hier dem Chan-Vese-Modell folgend mit einer Niveaumengenfunktion modelliert.



Erkennung von Körnern aus Simulationsbildern: Eingangsbild (links) und berechnete Segmentierung (rechts).

Simultane Bewegungsschätzung und Restauration von Bewegungsunschärfe. Zunächst entwickeln wir ein neues Bewegungsunschärfe-Modell, das die Bewegungsunschärfe auch am Rand eines sich bewegenden Objektes korrekt darstellt. Basierend darauf entwickeln wir ein Variationsmodel, das gleichzeitige Bewegungsschätzung und Restauration von Bewegungsunschärfe aus aufeinanderfolgenden Einzelbildern einer Videosequenz erlaubt. Hierbei wird angenommen, dass die Videosequenz ein Objekt zeigt, das sich vor einem statischen Hintergrund bewegt. Die Segmentierung in Objekt und Hintergrund wird durch einen Mumford-Shah-artigen Teil in dem Variationsmodel ermöglicht.

Restauration von Bewegungsunschärfe: Eingangsbilder (links) und restaurierte Bilder (rechts).

Konvexifizierung des binären Mumford-Shah-Segmentierungsproblems. Nachdem sich die übrigen Themen mit der Anwendung von Mumford-Shah-artigen Modellen zur Lösung spezieller Problemstellungen befasst haben, wird das Mumford-Shah-Funktional selbst eingehender studiert. Inspiriert durch die Methode von Nikolova-Esedoḡlu-Chan entwickeln wir einen Ansatz, der es erlaubt, globale Minimierer des binären Mumford-Shah-Segmentierungsproblems durch das Lösen eines konvexen, unrestringierten Minimierungsproblems zu finden. Anschließend stellen wir eine in Entwicklung befindliche Anwendung des Verfahrens zur globalen Optimierung vor, nämlich die Segmentierung von Flussfeldern in stückweise affine Regionen.

Global optimale Segmentierung: Zu segmentierendes Bild, Lösung des zugehörigen konvexen Problems und binäre Segmentierung (von links nach rechts).

Contents

Basic terminology and notation

$\mathcal{H}^{d-1}$	$d-1$ dimensional Hausdorff measure
χ_A	characteristic function of the set A
$\lvert Du\rvert(\Omega)$	total variation of u in Ω
$\mathrm{Per}(\Sigma,\Omega)$	perimeter of the set $\Sigma \subset \mathbb{R}^d$ in Ω, i. e. $\mathrm{Per}(\Sigma,\Omega) = \lvert D\chi_\Sigma\rvert(\Omega)$
$\mathrm{Per}(\Sigma)$	simplified notation of $\mathrm{Per}(\Sigma,\Omega)$
$\lvert\Sigma\rvert$	volume of the set Σ
$\{\phi > a\}$	a-super level set of ϕ, i. e. $\{x \in \Omega : \phi(x) > a\}$
$\{\phi < a\}$	a-sub level set of ϕ, i. e. $\{x \in \Omega : \phi(x) < a\}$
$\{\phi = a\}$	a-level set of ϕ, i. e. $\{x \in \Omega : \phi(x) = a\}$
H	Heaviside function, defined as $H(z) = 1$ for $z > 0$ and 0 elsewhere
$\langle E'[x], y\rangle$	first variation of E at x in direction y, i. e. $\frac{\mathrm{d}}{\mathrm{d}\epsilon}\left(E[x+\epsilon y]\right)\rvert_{\epsilon=0}$
$A \Delta B$	symmetric difference of two sets A and B, i. e. $(A \setminus B) \cup (B \setminus A)$
$x \cdot y$	scalar product of two vectors x and y
$\lvert x\rvert$	Euclidean norm of a vector x, i. e. $\sqrt{x \cdot x}$
$A : B$	scalar product of two matrices A and B interpreted as vectors, i. e. $\mathrm{tr}(A^T B)$
$\lVert A\rVert$	Frobenius norm of a matrix A, i. e. $\sqrt{A : A}$
$[a, b]$	interval from a to b, including a and b
$[a, b)$	interval from a to b, including a but excluding b
δ_{ij}	Kronecker delta, defined as $\delta_{ij} = 1$ for $i = j$ and 0 for $i \neq j$
e_i	i-th canonical basis vector of $\mathbb{R}^d$, i. e. $e_i = (\delta_{ij})_{j=1}^d$
$\mathbb{1}$	identity matrix, i. e. $\mathbb{1} = (\delta_{ij})_{ij}$
id	identity mapping, i. e. $\mathrm{id}(x) = x$
$\overline{1}$	one-vector, i. e. $\overline{1} = (1, \ldots, 1)^T$
$D\psi$	Jacobian matrix, i. e. $(D\psi)_{ij} = \partial_j \psi_i$

Conventions

- Unless otherwise stated, Ω denotes an open subset of $\mathbb{R}^d$.
- Elements of $\mathbb{R}^d$ are interpreted as column vectors.

1 Introduction

1.1 The Mumford–Shah model

THE nowadays well-known and widely used Mumford–Shah model, first proposed in the literature in 1989 [100], will be the starting point of all models we present here. In this model, a given image is approximated by a cartoon (u, K), consisting of a piecewise smooth image u with sharp edges on K, the discontinuity set in the image domain. This model has been extensively studied for numerous applications, e. g. segmentation, image denoising or shape modeling, cf. [99, 45, 46, 70] and the references therein.

For an image, i. e. a function $u_0 : \Omega \to \mathbb{R}$ on an image domain $\Omega \subset \mathbb{R}^d$ and nonnegative constants α, β and ν, the Mumford–Shah functional E_{MS} is given by

$$E_{\mathrm{MS}}[u, K] = \frac{\alpha}{2} \int_\Omega (u - u_0)^2 \,\mathrm{d}x + \frac{\beta}{2} \int_{\Omega \setminus K} |\nabla u|^2 \,\mathrm{d}x + \nu \mathcal{H}^{d-1}(K). \tag{1.1}$$

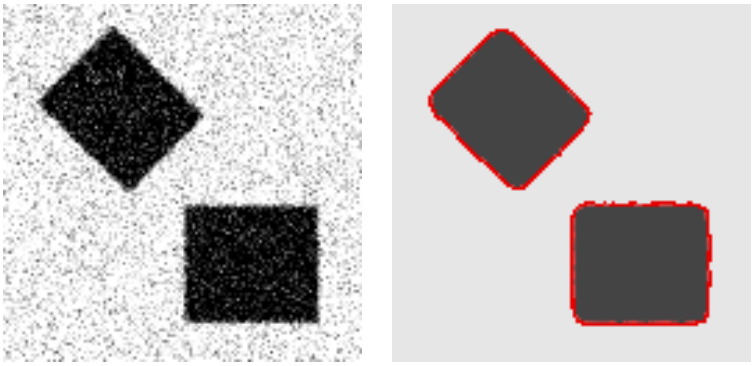
The first term, often called fidelity term, measures how well the piecewise smooth image u approximates the input image u_0. The second term acts as a kind of "edge-preserving smoother" in the sense that it penalizes large gradients of u in the homogeneous regions while not smoothing the image in the edge set K. The last term $\mathcal{H}^{d-1}$ denotes the $(d-1)$ dimensional Hausdorff measure and is used to control the length of the edge set. In particular it ensures that K is at most $(d-1)$ dimensional. Existence of pairs (u, K) minimizing (1.1) under mild conditions can be shown using SBV, the space of special functions of bounded variation [5, Theorem 7.15 + Theorem 7.22]. The key to the existence theory is a reformulation of the problem proposed by De Giorgi, Carriero, and Leaci [56] that only depends on $u \in SBV(\Omega)$. Here, the measure theoretic discontinuity set of u takes the role of K.

A variant of this model is the piecewise constant Mumford–Shah model. Here, we are looking for a piecewise constant image u (instead of a piecewise smooth one) to approximate the input image u_0. Let S_u denote the jump set of u, then the piecewise constant Mumford–Shah functional $E_{\mathrm{MS,pwc}}$ is defined as

$$E_{\mathrm{MS,pwc}}[u] = \frac{\alpha}{2} \int_\Omega (u - u_0)^2 \,\mathrm{d}x + \nu \mathcal{H}^{d-1}(S_u) \tag{1.2}$$

and to be minimized over the set of piecewise constant functions. Figure 1.1 shows how a noisy image is denoised with this model. Note that the functionals E_{MS} and $E_{\mathrm{MS,pwc}}$ coincide in the following sense: For any piecewise constant function u, it holds that $E_{\mathrm{MS,pwc}}[u] = E_{\mathrm{MS}}[u, S_u]$.

In this work, we give a glimpse at the flexibility offered by the Mumford–Shah model as we introduce several models based on it, each tackling a very different application. Chapter 2 presents a model for simultaneous edge detection of two images and joint estimation of a consistent pair of dense, nonlinear deformations (one in each direction) to match the two images based on the detected edges. Hereby, the edge detection is done in the spirit of the piecewise smooth Mumford–Shah model using the Ambrosio–Tortorelli approximation (cf.



u_0 (u, S_u)

Figure 1.1: A noisy input image (left) and the corresponding minimizer of the piecewise constant Mumford–Shah model (right).

Section 1.2.1) to handle the discontinuity set. The treatment of the two input images in this model is fully symmetric, i. e. the same matching is attained if the roles of the input images are swapped. Unlike many of the current asymmetric matching methods in the literature, this symmetric handling allows to establish one-to-one correspondences between the edge features of the two input images. The numerical implementation uses Finite Elements for the spatial discretization in combination with an expectation-maximization (EM) type algorithm involving a step size controlled, regularized gradient descent to update the deformations. Furthermore, the minimization algorithm uses a cascadic approach in a "coarse to fine" manner to avoid local minima. The influence of the various parameters of the symmetric matching model is investigated in a parameter study on a T1 and T2 magnetic resonance image (MRI) data pair. Finally, the performance of the proposed algorithm is illustrated on four different applications: intersubject monomodal registration, retinal image registration, matching a neurosurgical photograph with its projected volume data and motion estimation for frame interpolation.

Afterwards, in Chapter 3, we turn to a segmentation problem arising in materials science: Modern image acquisition techniques in materials science allow to resolve images at atomic scale and thus also to resolve so-called *grains*. Grains are material regions with different atomic lattice orientation which in addition are frequently elastically stressed compared to the reference configuration of the atomic lattice one would observe in the ideal case. Likewise, new microscopic simulation tools allow to study the dynamics of such grain structures. Single atoms are resolved experimentally as well as in simulation results on the data microscale, whereas lattice orientation and elastic deformation describe corresponding physical structures mesoscopically. A quantitative study of experimental images and simulation results and the comparison of simulation and experiment requires the robust and reliable extraction of mesoscopic properties from the microscopic image data, making this a two-scale problem. Based on a Mumford–Shah type functional, grain boundaries are described as free discontinuity sets at which the orientation parameter for the lattice jumps. The lattice structure itself is encoded by an indicator function depending on a local lattice orientation and an elastic displacement. This indicator function is built upon the fact that atoms are described by dots in the input images and upon the spatial relation of these dots to adjacent atomic dots. One global elastic displacement function, as well as a lattice orientation for each grain are considered as unknowns implicitly described by the image microstructure. To handle the deformation extraction, the Mumford–Shah type

functional is supplemented with an elastic energy for the deformation acting as a prior and a constraint on the deformation that separates the lattice orientation from the deformation. In addition to grain boundaries, the proposed approach incorporates interfaces between solid and liquid material regions. The resulting Mumford–Shah functional is approximated by a level set active contour model following the approach by Chan and Vese (cf. Section 1.2.2). Similar to Chapter 2, the numerical implementation is based on a Finite Element discretization in space and uses an EM type algorithm involving a step size controlled, regularized gradient descent. Finally, the results shown in this chapter illustrate that the proposed algorithm works equally well on simulated (phase field crystal simulations) and experimental data (transmission electron microscopy images).

In Chapter 4, we turn to the problem of motion estimation and restoration of objects in a video sequence affected by motion blur. This kind of blur results from fast movement of objects in combination with the aperture time of the camera used for the recording. Due to the motion blur, the direct velocity estimation from such videos is inaccurate. On the other hand, an accurate estimation of the velocity of the moving objects is crucial for restoration of motion-blurred video. In other words, restoration needs accurate motion estimation and vice versa, and a joint handling of restoration and motion estimation is called for. To address this, we first derive a novel model of the blurring process that is accurate also close to the boundary of the moving object, a key property missing in existing blurring models in the literature. Based on the blurring model, we propose a variational framework acting on consecutive video frames for joint object detection, deblurring and velocity estimation. Here, the video is assumed to consist of a moving object and a static background, and the automatic distinction between the moving object and the background is handled by a Mumford–Shah type aspect of the proposed model. The importance of this joint estimation and its superior performance when compared to the independent estimation of motion and restoration is outlined by experimental results both on simulated and real video data.

After developing several models based on the Mumford–Shah functional to solve specific image processing tasks in the previous chapters, Chapter 5 approaches the Mumford–Shah functional itself and provides a way to obtain global minimizers of the two-phase Mumford–Shah segmentation model, despite the fact that this is a non-convex optimization problem. Inspired by the work of Nikolova, Esedoḡlu and Chan (cf. Section 1.2.4) and similar to their approach, this is accomplished by deriving a convex minimization problem whose minimizers can be directly converted to minimizers of the initial non-convex problem by thresholding. The key difference to the Nikolova–Esedoḡlu–Chan (NEC) model is that the model we propose here does not need to impose any constraint in the convex formulation, neither explicitly nor implicitly by an additional, artificial penalty term in the convex objective functional. The unconstrained approach is related to recent results by Chambolle derived in the context of total variation minimization. Due to the simplicity of the resulting convex optimization problem, even a straightforward gradient descent allows for a reliable computation of the global minimizer. Moreover, the two-phase model can be combined with the multiphase idea of Vese and Chan (cf. Section 1.2.3) and is extended to multiphase segmentation, though the convexity is lost when moving to multiphase segmentation. Numerically, we apply the proposed approach to the classical piecewise constant Mumford–Shah problem and show results for two, four and eight phase segmentation. Furthermore, we compare the numerical binary segmentation quality of the proposed method with the one of the NEC model.

1.2 Handling the discontinuity set

One of the key challenges when it comes to effectively using the Mumford–Shah model in numerical applications is the proper treatment of the discontinuity set K. In this work, we mainly use two different approaches: Diffuse interface representations (cf. Section 1.2.1) and sharp interface representations (cf. Section 1.2.2). Each of these representations has its own benefits and shortcomings, and it depends on the application which one is more appropriate.

1.2.1 The Ambrosio–Tortorelli model

The basic idea behind the Ambrosio–Tortorelli (AT) approximation [4] is to replace the discontinuity set K by a scalar-valued function, here denoted by v. This so-called *phase field function* is essentially determined by two properties: First, it shall approximate $(1 - \chi_K)$, the characteristic function of the complement of K, i. e. $v(x) \approx 0$ for $x \in K$ and $v(x) \approx 1$ otherwise. Second, it is supposed to be smooth (in the H^1 sense). Unlike the approach by Chan and Vese [46] (cf. Section 1.2.2), the discontinuity set is only approximated here in a diffuse manner. Therefore, this approach is referred to as *diffuse interface model*.

The entire approximation functional designed to fulfill these goals is defined as follows:

$$E^{\epsilon}_{\mathrm{AT}}[u, v] = \frac{\alpha}{2} \int_{\Omega} (u - u_0)^2 \,\mathrm{d}x + \frac{\beta}{2} \int_{\Omega} v^2 \left|\nabla u\right|^2 \mathrm{d}x + \frac{\nu}{2} \int_{\Omega} \epsilon \left|\nabla v\right|^2 + \frac{1}{4\epsilon}(v - 1)^2 \,\mathrm{d}x. \tag{1.3}$$

The three terms the energy consists of approximate the corresponding terms of the Mumford–Shah functional (1.1). The first term is the same as the first one of E_{MS}. The second term, working as an "edge-preserving smoother" like the second term of E_{MS}, couples zero-regions of v with regions where the gradient of u is large. The last term approximates $\mathcal{H}^{d-1}(K)$, i. e. the edge length of K. Due to the second term and the second part of the third term (a so-called *single well potential*) the following "coupling" between u and v is energetically preferable:

$$v(x) \approx \begin{cases} 0 & \text{where } |\nabla u| \gg 0, \\ 1 & \text{where } |\nabla u| \approx 0. \end{cases} \tag{1.4}$$

The additional parameter ϵ, not used in E_{MS}, controls the "width" of the diffusive edge set, cf. Figure 1.2. In particular, the transition profile of the phase field at an edge is characterized by the following Lemma (cf. [40]):

1.2.1 Lemma. *$v_\epsilon : [0, \infty) \to \mathbb{R}, x \mapsto 1 - \exp\left(-\frac{x}{2\epsilon}\right)$ minimizes*

$$\int_0^{\infty} \epsilon \left|v'\right|^2 + \frac{1}{4\epsilon}(v - 1)^2 \,\mathrm{d}x$$

under the boundary conditions $v(0) = 0$ and $\lim_{x\to\infty} v(x) = 1$.

Proof. The corresponding Euler–Lagrange equation is

$$-2\epsilon v'' + \frac{1}{2\epsilon}(v - 1) = 0 \text{ in } (0, \infty).$$

$v_\epsilon(0) = 0$ and $\lim_{x\to\infty} v_\epsilon(x) = 1$ obviously hold. Furthermore, because of

$$v_\epsilon''(x) = -\frac{1}{4\epsilon^2} \exp\left(-\frac{x}{2\epsilon}\right),$$

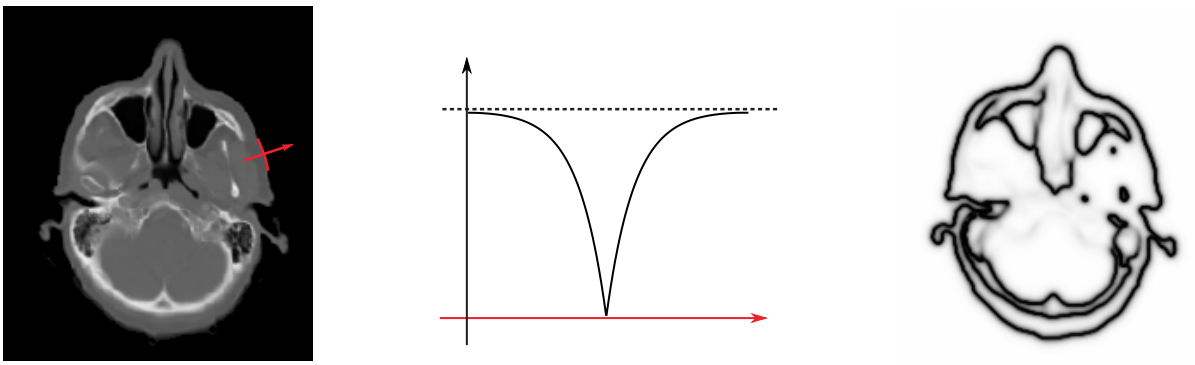


Figure 1.2: Structure of phase fields: Input image u_0 (left), phase field function plotted across an edge (middle) and phase field function v corresponding to u_0 scaled to gray values.

u_0 u v

Figure 1.3: Results of Ambrosio–Tortorelli segmentation on the well-known Lena image.

v_ϵ apparently solves the Euler–Lagrange equation. Combined with the fact that the value of the functional evaluated at v_ϵ is finite and the convexity of the functional, we can conclude that v_ϵ is a minimizer. □

The performance of the Ambrosio–Tortorelli model is illustrated on the well-known Lena image [1, image 4.2.04] in Figure 1.3. The connection between the Ambrosio–Tortorelli model and the Mumford–Shah model can be specified as follows: The sequence of functionals E^ϵ_{AT} $\Gamma-$converges to E_{MS}, i. e.

$$\Gamma\text{-}\lim_{\epsilon\to 0} E^\epsilon_{\mathrm{AT}} = E_{\mathrm{MS}}.$$

To establish this result, v^2 in the second term of $E^\epsilon_{\mathrm{AT}}[u,v]$ has to be replaced by v^2+k_ϵ, where k_ϵ is a positive parameter fulfilling $\lim_{\epsilon\downarrow 0} k_\epsilon/\epsilon = 0$. This change ensures the coercivity of E^ϵ_{AT} in $H^1(\Omega)\times H^1(\Omega)$. For a detailed discussion including a rigorous proof, we refer to [4, 28].

1.2.2 The Chan–Vese model

For an image u_0, the well-known piecewise constant Mumford–Shah functional for two-phase segmentation is given by

$$E_{\text{MS-2,pwc}}[\Sigma, c_1, c_2] = \int_\Sigma (u_0-c_1)^2\,\mathrm{d}x + \int_{\Omega\setminus\Sigma} (u_0-c_2)^2\,\mathrm{d}x + \nu\,\mathrm{Per}(\Sigma). \tag{1.5}$$

Here, $\mathrm{Per}(\Sigma)$ is a simplified notation of $\mathrm{Per}(\Sigma, \Omega)$ and denotes the perimeter of the set $\Sigma \subset \mathbb{R}^d$ in Ω, cf. [5, Definition 3.35]. It is obtained directly from (1.1): Set $\alpha = 2$ and restrict u to be piecewise constant and only allowed to take two different values, i. e. $u = c_1\chi_\Sigma + c_2\chi_{\Omega\setminus\Sigma}$. Then $\partial\Sigma$ is the jump set of u, thus we have $K = \partial\Sigma$ and $\nabla u = 0$ in $\Omega \setminus K$, hence the second term of (1.1) vanishes. Furthermore, the first term of (1.1) coincides with the sum of the first two terms of (1.5) and (under mild regularity assumptions) the third term of both equations is equal.

Replacing $(u_0 - c_1)^2$ and $(u_0 - c_2)^2$ by general indicator functions $f_1, f_2 \in L^1(\Omega)$ such that $f_1, f_2 \geq 0$ a. e., we get the prototype Mumford–Shah energy

$$E_{\text{MS-2}}[\Sigma] := \int_\Sigma f_1 \,\mathrm{d}x + \int_{\Omega\setminus\Sigma} f_2 \,\mathrm{d}x + \nu \,\mathrm{Per}(\Sigma). \tag{1.6}$$

To obtain (local) minimizers of $E_{\text{MS-2}}$, Chan and Vese [46] proposed to parametrize the unknown set Σ by a function, i. e. the unknown set Σ is represented by the zero super level set $\{\phi > 0\} := \{x \in \Omega : \phi(x) > 0\}$ of a so-called *level set function* $\phi : \Omega \to \mathbb{R}$, building upon the level set methods of Osher and Sethian [105]. In other words, we have $\Sigma = \{\phi > 0\}$ and in particular $\partial\Sigma = \{\phi = 0\}$. Hence, the model exactly localizes the discontinuity set and is referred to as *sharp interface model*.

The domain splitting into Σ and its complement in the different energy terms then can easily be expressed in terms of ϕ via the *Heaviside function*

$$H : \mathbb{R} \to \{0, 1\}, s \mapsto \begin{cases} 1 & s > 0 \\ 0 & \text{else} \end{cases}, \tag{1.7}$$

because $\chi_\Sigma = H(\phi)$ and $\chi_{\Omega\setminus\Sigma} = (1 - H(\phi))$. For $u \in BV(\Omega)$, let $|Du|(\Omega)$ denote the total variation of u in Ω. Then, by [5, Proposition 3.6], we have

$$|Du|(\Omega) = \sup\left\{ \int_\Omega u \,\mathrm{div} p \,\mathrm{d}x \,:\, p \in C_0^1(\Omega)^d, \max_{x\in\Omega} |p(x)| \leq 1 \right\} =: \sup_{|p|\leq 1} \int_\Omega u \,\mathrm{div} p \,\mathrm{d}x \tag{1.8}$$

and the perimeter of the unknown set can be rewritten as the total variation of $H \circ \phi$:

1.2.2 Lemma. *If $\phi : \Omega \to \mathbb{R}$ is such that $\{\phi > 0\}$ has finite perimeter, it holds that*

$$|D(H \circ \phi)|(\Omega) = \mathrm{Per}(\{\phi > 0\}, \Omega).$$

Proof Any $u \in BV(\Omega)$ fulfills

$$|Du|(\Omega) = \int_{-\infty}^{\infty} \left|D\chi_{\{u>s\}}(\Omega)\right| \mathrm{d}s = \int_{-\infty}^{\infty} \mathrm{Per}(\{u > s\}, \Omega) \,\mathrm{d}s\,. \tag{1.9}$$

The first equality holds because of [5, Theorem 3.40], the second one because of [5, Theorem 3.36]. Since $\{\phi > 0\}$ has finite perimeter, we have $H \circ \phi \in BV(\Omega)$. Therefore, we get

$$\begin{aligned} |D(H \circ \phi)|(\Omega) &= \int_{-\infty}^{\infty} \mathrm{Per}(\{H \circ \phi > s\}, \Omega) \,\mathrm{d}s = \int_0^1 \mathrm{Per}(\{H \circ \phi > s\}, \Omega) \,\mathrm{d}s \\ &= \mathrm{Per}(\{H \circ \phi > 0\}, \Omega) = \mathrm{Per}(\{\phi > 0\}, \Omega). \end{aligned}$$

The first equality uses (1.9), for the second equality we used that

$$\{H \circ \phi > s\} = \begin{cases} \Omega & s < 0 \\ \emptyset & s > 1 \end{cases},$$

while the third one holds because of $\{H \circ \phi > s\} = \{H \circ \phi > 0\}$ for $s \in (0,1)$. Finally, in the last equality we used that $\{H \circ \phi > 0\} = \{\phi > 0\}$. □

Collecting what we observed so far leads to the following Chan–Vese energy

$$E_{\mathrm{CV}}[\phi] := \int_\Omega H(\phi) f_1 + (1 - H(\phi)) f_2 \,\mathrm{d}x + \nu \,|D(H \circ \phi)|\,(\Omega). \tag{1.10}$$

It is a reformulation of $E_{\text{MS-2}}$ in the sense that $E_{\text{MS-2}}[\{\phi > 0\}] = E_{\mathrm{CV}}[\phi]$.

To numerically minimize this energy we want to use a gradient flow and therefore have to derive E_{CV}. Since H is not continuous, we replace it by a smeared out Heaviside function. As in [46] we use

$$H_\delta(s) := \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{s}{\delta}\right), \tag{1.11}$$

for a scale parameter $\delta > 0$ that controls the strength of the regularization. While the specific choice is not important, it is important to use a function whose derivative does not have compact support, because the desired guidance of the initial zero contour to the actual targeted segmentation boundary relies on this property (cf. [46]). Furthermore, we will make use of the fact that $H'_\delta(s) > 0$ for all $s \in \mathbb{R}$ when calculating the variation of the regularized perimeter term (see below). Also note that

$$H'_\delta(s) = \frac{\delta}{\pi(\delta^2 + s^2)}$$

converges to H' in the sense of distributions for $\delta \to 0$.

Note that this kind of regularization of the Heaviside function can be interpreted as phase field type approach. Here, the transition profile between interior and exterior of the unknown set is explicitly modeled by the regularization instead of being implicitly encoded in the energy functional (cf. (5.2)).

Additionally, the length term of (1.10) needs to be regularized. First, we note that

$$|D(H_\delta \circ \phi)|\,(\Omega) = \int_\Omega |\nabla(H_\delta \circ \phi)| \,\mathrm{d}x$$

holds for $\phi \in H^1(\Omega)$. Since the absolute value $|\cdot|$ is not differentiable in 0, we regularize it by $|z|_\varrho = \sqrt{z^2 + \varrho^2}$. In total, we get the regularized energy

$$E^{\delta,\varrho}_{\mathrm{CV}}[\phi] := \int_\Omega H_\delta(\phi) f_1 + (1 - H_\delta(\phi)) f_2 + \nu |\nabla(H_\delta \circ \phi)|_\varrho \,\mathrm{d}x\,. \tag{1.12}$$

Unless otherwise noted, $\varrho = 0.1$ is used in the numerics.

While this approach is widely used and suitable for a number of problems (cf. Chapters 3 and 4), one drawback of the energy (1.12) is its non-convexity in ϕ. In Chapter 5, we turn to this problem and present an alternative approach to find (even global) minimizers of (1.6) by solving a strictly convex, unconstrained optimization problem.

The only term of (1.12), whose derivative needs special treatment, is the length term. Thus we look into this first: Let

$$L[\phi] := \int_\Omega |\nabla(H_\delta \circ \phi)|_\varrho \,\mathrm{d}x\,.$$

Using $\nabla(H_\delta \circ \phi) = H'_\delta(\phi)\nabla\phi$, $H'_\delta(\phi) > 0$ and

$$\frac{\mathrm{d}}{\mathrm{d}z}\,|z|_\varrho = \frac{z}{|z|_\varrho}$$

we get the variation (with test function $\vartheta \in C_0^\infty(\Omega)$)

$$\begin{aligned}\left\langle L'[\phi], \vartheta \right\rangle &= \int_\Omega \frac{\nabla\phi}{|\nabla\phi|_\varrho} \cdot \left(H''_\delta(\phi)\nabla\phi\vartheta + H'_\delta(\phi)\nabla\vartheta\right) \mathrm{d}x \\ &= \int_\Omega \nabla(H'_\delta(\phi)\vartheta) \cdot \frac{\nabla\phi}{|\nabla\phi|_\varrho}\,\mathrm{d}x \\ &= -\int_\Omega H'_\delta(\phi)\vartheta \,\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|_\varrho}\right)\mathrm{d}x\,.\end{aligned} \tag{1.13}$$

Note that we have carefully rewritten the variation using integration by parts to get rid of the second derivative of H_δ. The advantage of this particular reformulation will become apparent in (1.14) and (1.15).

With (1.13) we easily derive the variation of $E_{\mathrm{CV}}^{\delta,\varrho}$:

$$\begin{aligned}\left\langle \partial_\phi E_{\mathrm{CV}}^{\delta,\varrho}[\phi], \vartheta \right\rangle &= \int_\Omega H'_\delta(\phi) f_1 \vartheta \,\mathrm{d}x - \int_\Omega H'_\delta(\phi) f_2 \vartheta \,\mathrm{d}x - \nu \int_\Omega H'_\delta(\phi)\vartheta \,\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|_\varrho}\right)\mathrm{d}x \\ &= -\int_\Omega H'_\delta(\phi)\left[(f_2 - f_1) + \nu\,\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|_\varrho}\right)\right]\vartheta\,\mathrm{d}x\,.\end{aligned} \tag{1.14}$$

By definition, the weak formulation of the L^2-gradient flow for $E_{\mathrm{CV}}^{\delta,\varrho}$ (cf. Section 6.2) is

$$\underset{\vartheta\in C_0^\infty(\Omega)}{\forall} \quad \int_\Omega \partial_t\phi\vartheta\,\mathrm{d}x = -\left\langle \partial_\phi E_{\mathrm{CV}}^{\delta,\varrho}[\phi], \vartheta \right\rangle.$$

Hence, by (1.14) and the fundamental lemma of the calculus of variations the strong formulation of the L^2-gradient flow is

$$\partial_t\phi = H'_\delta(\phi)\left[(f_2 - f_1) + \nu\,\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|_\varrho}\right)\right]. \tag{1.15}$$

Division by $H'_\delta(\phi)$ yields

$$\frac{\partial_t\phi}{H'_\delta(\phi)} = (f_2 - f_1) + \nu\,\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|_\varrho}\right)$$

and with integration by parts we get the weak formulation of the gradient flow (1.15)

$$\underset{\vartheta\in C_0^\infty(\Omega)}{\forall} \quad \int_\Omega \frac{\partial_t\phi}{H'_\delta(\phi)}\vartheta\,\mathrm{d}x = \int_\Omega (f_2 - f_1)\vartheta\,\mathrm{d}x - \nu\int_\Omega \nabla\vartheta \cdot \frac{\nabla\phi}{|\nabla\phi|_\varrho}\,\mathrm{d}x\,. \tag{1.16}$$

1.2.3 The Vese–Chan model for multiphase segmentation

In [125], Vese and Chan proposed an extension of their binary segmentation model to multiphase segmentation. The basic idea is to use additional level set functions to encode additional segments while encoding as many segments per level set function as possible: Given n level set functions $\phi_1, \dots, \phi_n$, one can encode 2^n segments by considering all possible sign combinations of the level set functions. Precisely, the segments are

$$\Sigma_{\boldsymbol{k}} := \left\{ x \,:\, (-1)^{k_i}\phi_i > 0,\, i = 1, \dots, n \right\} \text{ for } \boldsymbol{k} = (k_1, \dots, k_n) \in \{0,1\}^n.$$

Furthermore, we assume that the segmentation is based on a scalar value and an indicator function $f : \Omega \times \mathbb{R} \to \mathbb{R}$. Then, for each segment $\Sigma_{\boldsymbol{k}}$ we also look for a $c_{\boldsymbol{k}} \in \mathbb{R}$, and the multiphase Vese–Chan energy for multiple segments is defined as follows:

$$E_{\mathrm{VC}}[(\phi_i)_i, (c_{\boldsymbol{k}})_{\boldsymbol{k}}] = \underbrace{\sum_{\boldsymbol{k}} \int_\Omega \prod_i H((-1)^{k_i}\phi_i) f(x, c_{\boldsymbol{k}}) \,\mathrm{d}x}_{=:E_{\mathrm{VC,fid}}[(\phi_i)_i, (c_{\boldsymbol{k}})_{\boldsymbol{k}}]} + \nu \sum_i |D(H \circ \phi_i)|\,(\Omega).$$

Upon closer inspection, the fidelity term $E_{\mathrm{VC,fid}}$ is a reformulation of the corresponding term form the piecewise constant Mumford–Shah functional (1.2): By construction of $\Sigma_{\boldsymbol{k}}$, we have

$$\prod_i H((-1)^{k_i}\phi_i(x)) = \chi_{\Sigma_{\boldsymbol{k}}}(x) \text{ for } x \in \Omega.$$

Hence, defining $c = \sum_{\boldsymbol{k}} c_{\boldsymbol{k}} \chi_{\Sigma_{\boldsymbol{k}}}$ and noting that $\Sigma_{\boldsymbol{k}} \cap \Sigma_{\boldsymbol{l}} = \emptyset$ for $\boldsymbol{k} \neq \boldsymbol{l}$, we have

$$E_{\mathrm{VC,fid}}[(\phi_i)_i, (c_{\boldsymbol{k}})_{\boldsymbol{k}}] = \sum_{\boldsymbol{k}} \int_{\Sigma_{\boldsymbol{k}}} f(x, c_{\boldsymbol{k}}) \,\mathrm{d}x = \int_\Omega f(x, c) \,\mathrm{d}x\,.$$

Note that for the latter equality we need to assume that the 0-level sets $\{\phi_i = 0\}$ are Lebesgue null sets. In this case, $E_{\mathrm{VC,fid}}$ coincides with the first term of (1.2), if we choose the indicator function accordingly, i. e. $f(x, c) := (u_0(x) - c)^2$ for an image u_0. Here, the scalar quantity $c_{\boldsymbol{k}}$ plays the role of the average gray value in $\Sigma_{\boldsymbol{k}}$.

For the same choice of f and $n = 1$, we deduce from $H(-s) = 1 - H(s)$ for $s \neq 0$ that (as long as $\{\phi_1 = 0\}$ is a Lebesgue null sets) the Vese–Chan energy coincides with the Chan–Vese reformulation of the piecewise constant Mumford–Shah functional for two-phase segmentation (1.5).

Let us point out one important drawback of this approach: For $n \geq 2$, E_{VC} is not an exact reformulation of the corresponding Mumford–Shah model, because the perimeter of the segments $\Sigma_{\boldsymbol{k}}$ is not measured uniformly. To demonstrate this problem we exemplarily consider the case $n = 2$: Then, we have following four segments

$$\begin{aligned}
\Sigma_{(0,0)} &= \{\phi_1 > 0\} \cap \{\phi_2 > 0\},\\
\Sigma_{(1,0)} &= \{\phi_1 < 0\} \cap \{\phi_2 > 0\},\\
\Sigma_{(0,1)} &= \{\phi_1 > 0\} \cap \{\phi_2 < 0\},\\
\Sigma_{(1,1)} &= \{\phi_1 < 0\} \cap \{\phi_2 < 0\}.
\end{aligned}$$

Now, $\sum_i |D(H \circ \phi_i)|\,(\Omega)$ measures the boundary between $\Sigma_{(0,0)}$ and $\Sigma_{(1,0)}$ once because ϕ_1 changes its sign in this region while ϕ_2 does not, whereas it measures the boundary between

$\Sigma_{(0,0)}$ and $\Sigma_{(1,1)}$ twice because both ϕ_1 and ϕ_2 change their sign here. This nonuniform perimeter measurement is an undesired side effect of the Vese–Chan model that does not happen in the original Mumford–Shah model. Nevertheless, we use the Vese–Chan model because this effect is of no consequence for the applications considered in this work.

Applying the same regularization used for (1.12), we get the regularized Vese–Chan energy

$$E_{\mathrm{VC}}^{\delta,\varrho}[(\phi_i)_i,(c_{\boldsymbol{k}})_{\boldsymbol{k}}] = \sum_{\boldsymbol{k}} \int_\Omega \prod_i H_\delta((-1)^{k_i}\phi_i) f(x,c_{\boldsymbol{k}})\,\mathrm{d}x + \nu \sum_i \int_\Omega |\nabla(H_\delta\circ\phi_i)|_\varrho\,\mathrm{d}x\,. \quad (1.17)$$

Noting that

$$H_\delta(-s) = 1 - H_\delta(s) \text{ and } H'_\delta(s) = H'_\delta(-s),$$

the variations of the regularized energy are obtained straightforwardly:

$$\begin{aligned}
\left\langle \partial_{\phi_j} E_{\mathrm{VC}}^{\delta,\varrho}[(\phi_i)_i,(c_{\boldsymbol{k}})_{\boldsymbol{k}}], \vartheta \right\rangle &= \sum_{\boldsymbol{k}} \int_\Omega (-1)^{k_j} H'_\delta((-1)^{k_j}\phi_j) \prod_{i\neq j} H_\delta((-1)^{k_i}\phi_i) f(x,c_{\boldsymbol{k}})\vartheta\,\mathrm{d}x \\
&\quad + \int_\Omega \nabla(H'_\delta(\phi_j)\vartheta)\cdot\frac{\nabla\phi_j}{|\nabla\phi_j|_\varrho}\,\mathrm{d}x, \\
&= \sum_{\boldsymbol{k}} \int_\Omega (-1)^{k_j} H'_\delta(\phi_j) \prod_{i\neq j} H_\delta((-1)^{k_i}\phi_i) f(x,c_{\boldsymbol{k}})\vartheta\,\mathrm{d}x \\
&\quad + \int_\Omega \nabla(H'_\delta(\phi_j)\vartheta)\cdot\frac{\nabla\phi_j}{|\nabla\phi_j|_\varrho}\,\mathrm{d}x, \\
\partial_{c_{\boldsymbol{l}}} E_{\mathrm{VC}}^{\delta,\varrho}[(\phi_i)_i,(c_{\boldsymbol{k}})_{\boldsymbol{k}}] &= \int_\Omega \prod_i H_\delta((-1)^{l_i}\phi_i)\partial_c f(x,c_{\boldsymbol{l}})\,\mathrm{d}x\,.
\end{aligned}$$

Though the regularization of the Heaviside function makes this approach phase field like, the Vese–Chan model is still conceptually different from the Ambrosio–Tortorelli model (cf. Section 1.2.1), even if the indicator function is chosen to resemble the fidelity term of the AT model, i. e. $f(x,c) := (u_0(x)-c)^2$. The main difference here is that the AT model only separates the domain Ω into edges and smooth regions, while the VC model further separates the smooth part of the domain into distinct segments.

1.2.4 The Nikolova–Esedoḡlu–Chan model

Using a formal calculation (assuming $\nabla\phi \neq 0$ a. e.) analogously to (1.15), one obtains the L^2-gradient flow of the regularized Chan–Vese functional without regularization of the absolute value (i. e. $\varrho = 0$):

$$\partial_t\phi = H'_\delta(\phi)\left[(f_2-f_1) + \nu\,\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right].$$

The starting point of Nikolova et al. [102] is the observation that, due to $H'_\delta(\phi) > 0$, this gradient flow and

$$\partial_t\phi = \left[(f_2-f_1) + \nu\,\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right]$$

have the same stationary points. Obviously the latter is the L^2-gradient flow of the energy

$$E_{\text{NEC}}[\phi] = \int_\Omega (f_1 - f_2)\phi \,\mathrm{d}x + \nu \,|D\phi|\,(\Omega),$$

serving as motivation to study the properties of this energy.

Note that this energy can be considered as a reformulation of $E_{\text{MS-2}}$ from sets to binary images, i. e. images only taking values of 0 or 1. Consider $u \in BV(\Omega, \{0,1\})$. Then, $u = \chi_{\{u=1\}}$ and, using [5, Theorem 3.36], one gets $|Du|\,(\Omega) = \operatorname{Per}(\{u=1\}, \Omega)$. Therefore,

$$\begin{aligned} E_{\text{MS-2}}[\{u=1\}] &= \int_\Omega uf_1 + (1-u)f_2 \,\mathrm{d}x + \nu \,|Du|\,(\Omega) \\ &= E_{\text{NEC}}[u] + \int_\Omega f_2 \,\mathrm{d}x\,. \end{aligned}$$

Hence, $E_{\text{MS-2}}$ and E_{NEC} are the same up to a constant if we interpret sets as binary images.

In general, $f_1 - f_2$ takes positive and negative values, therefore E_{NEC} is not bounded (neither from below nor from above). In other words, it does not necessarily have a minimizer. However, this is easily fixed by restricting the minimization to $0 \le \phi(x) \le 1$ for all $x \in \Omega$. Based on this, the following theorem holds:

1.2.3 Theorem. *For given indicator functions $f_1, f_2 \in L^1(\Omega)$ such that $f_1, f_2 \ge 0$ a. e., let*

$$u := \operatorname*{argmin}_{0\le\tilde{u}\le1} \int_\Omega (f_1 - f_2)\tilde{u} \,\mathrm{d}x + \nu \,|D\tilde{u}|\,(\Omega) = \operatorname*{argmin}_{0\le\tilde{u}\le1} E_{\text{NEC}}[\tilde{u}]$$

and $\Sigma_c := \{u > c\}$. Then Σ_c is a minimizer of the binary Mumford–Shah energy (1.6) *for a. e. $c \in [0,1]$.*

Proof This theorem has been proven by Nikolova et al. in [102]. Nevertheless, we show their proof here, reformulated to fit into our context, since we will reuse some of its ideas later on.

Using (1.9) and $0 \le u \le 1$ a. e., we deduce

$$|Du|\,(\Omega) = \int_{-\infty}^{\infty} \operatorname{Per}(\{u > c\}, \Omega) \,\mathrm{d}c = \int_0^1 \operatorname{Per}(\Sigma_c) \,\mathrm{d}c\,.$$

Further, we get

$$\begin{aligned} \int_\Omega f_1(x)u(x) \,\mathrm{d}x &= \int_\Omega f_1(x) \int_0^1 \chi_{[0,u(x)]}(c) \,\mathrm{d}c \,\mathrm{d}x = \int_0^1 \int_\Omega f_1(x)\chi_{[0,u(x)]}(c) \,\mathrm{d}x \,\mathrm{d}c \\ &= \int_0^1 \int_\Omega f_1(x)\chi_{\Sigma_c}(x) \,\mathrm{d}x \,\mathrm{d}c = \int_0^1 \int_{\Sigma_c} f_1(x) \,\mathrm{d}x \,\mathrm{d}c\,. \end{aligned}$$

Similarly,

$$\begin{aligned} \int_\Omega f_2(x)u(x) \,\mathrm{d}x &= \int_0^1 \int_{\Sigma_c} f_2(x) \,\mathrm{d}x \,\mathrm{d}c = \int_0^1 \int_\Omega f_2(x) \,\mathrm{d}x \,\mathrm{d}c - \int_0^1 \int_{\Omega\setminus\Sigma_c} f_2(x) \,\mathrm{d}x \,\mathrm{d}c \\ &= \underbrace{\int_\Omega f_2(x) \,\mathrm{d}x}_{=:C} - \int_0^1 \int_{\Omega\setminus\Sigma_c} f_2(x) \,\mathrm{d}x \,\mathrm{d}c, \end{aligned}$$

where C is a constant independent of u. This leads to

$$\begin{aligned} E_{\text{NEC}}[u] &= \int_0^1 \left[\int_{\Sigma_c} f_1(x)\,\mathrm{d}x + \int_{\Omega\setminus\Sigma_c} f_2(x)\,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_c) \right] \mathrm{d}c - C \\ &= \int_0^1 E_{\text{MS-2}}[\Sigma_c]\,\mathrm{d}c - C. \end{aligned}$$

Let $\Sigma_* \subset \Omega$ be a minimizer of $E_{\text{MS-2}}$. The existence of such minimizers using convergence in measure (the distance of two sets being the Lebesgue measure of their symmetric difference) follows from standard arguments: The perimeter is lower semicontinuous (cf. [5]) and the lower semicontinuity of the integral terms follows from Fatou's lemma using $f_1, f_2 \in L^1(\Omega)$ to get an integrable lower bound.

Let $M := \{c \in [0,1] : E_{\text{MS-2}}[\Sigma_c] > E_{\text{MS-2}}[\Sigma_*]\}$ and assume $\mu(M) > 0$. This gives

$$E_{\text{NEC}}[\chi_{\Sigma_*}] = \int_0^1 E_{\text{MS-2}}[\Sigma_*]\,\mathrm{d}c - C < \int_0^1 E_{\text{MS-2}}[\Sigma_c]\,\mathrm{d}c - C = E_{\text{NEC}}[u].$$

This is a contradiction to the fact that u minimizes E_{NEC}. Therefore $\mu(M) = 0$ holds and the proposition is proven. □

Theorem 1.2.3 is the be-all and end-all of the Nikolova–Esedoḡlu–Chan model because of the connection it establishes between E_{NEC} and $E_{\text{MS-2}}$: By minimizing the convex energy E_{NEC} under a convex constraint, followed by simple thresholding, one obtains a global minimizer of the binary Mumford–Shah energy (1.6).

To solve the constrained optimization problem Nikolova et al. introduce an exact penalty to transform the constrained optimization problem into an unconstrained one:

1.2.4 Proposition. *Let $s \in L^\infty(\Omega)$. Then*

$$\min_{0\le u\le 1} \int_\Omega s(x)u(x)\,\mathrm{d}x + \nu\,|Du|\,(\Omega)$$

and

$$\min_u \int_\Omega s(x)u(x) + \alpha p(u(x))\,\mathrm{d}x + \nu\,|Du|\,(\Omega),$$

where $p(s) = \max\{0, 2\left|s - \frac{1}{2}\right| - 1\}$, have the same set of minimizer, provided that $\alpha > \frac{1}{2}\|p\|_{L^\infty(\Omega)}$.

Proof See [102, Claim 1]. □

1.3 Publications and collaborations

A number of publications emerged during the development of this work, listed at the end of this section. Most of the publications are covered here, but some of them are beyond the scope of this thesis. These are anisotropic total variation methods for right-angled corner preserving cartoon extraction of aerial images [BBD+06] and image guided motion inpainting [BKGR09] as well as a shape median based on symmetric area differences that uses a Mumford–Shah type variational formulation to handle the median shape as a free discontinuity set.

Moreover, let us point out that several of the methods presented in this work are the result of close collaborations. In particular, the one-to-one edge matching approach (cf. Chapter 2) and the motion deblurring model (cf. Chapter 4) should be mentioned here. The one-to-one edge matching approach was developed with Jingfeng Han and Joachim Hornegger from the Institute of Pattern Recognition, University of Erlangen-Nuremberg. Here, Jingfeng Han focussed on the implementation of the functionals and variations, and the numerical evaluation of the method, while I concentrated on the mathematical modeling and the implementation of a general, step size controlled gradient flow framework. The one-to-one edge matching model is also discussed in the thesis of Jingfeng Han [72]. Furthermore, the underlying idea to symmetrize the model of Droske et al. was brought up by Jingfeng Han.

The motion deblurring model was devised in cooperation with Leah Bar and Guillermo Sapiro from the Department of Electrical and Computer Engineering, University of Minnesota. Here, Leah Bar took care of the numerical implementation and the experiments, while I concentrated on the modeling aspects, in particular the development of an accurate motion blur model and numerically usable representations of the variations of the functional.

Publications

[BBD+06] Benjamin Berkels, Martin Burger, Marc Droske, Oliver Nemitz, and Martin Rumpf. Cartoon extraction based on anisotropic image classification. In *Vision, Modeling, and Visualization Proceedings*, pages 293–300, 2006.

[BBRS07] Leah Bar, Benjamin Berkels, Martin Rumpf, and Guillermo Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *Eleventh IEEE International Conference on Computer Vision (ICCV 2007)*, 2007.

[Ber09] Benjamin Berkels. An unconstrained multiphase thresholding approach for image segmentation. In *Proceedings of the Second International Conference on Scale Space Methods and Variational Methods in Computer Vision (SSVM 2009)*, volume 5567 of *Lecture Notes in Computer Science*, pages 26–37. Springer, 2009.

[BKGR09] Benjamin Berkels, Claudia Kondermann, Christoph Garbe, and Martin Rumpf. Reconstructing optical flow fields by motion inpainting. In *Seventh International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR 2009)*, volume 5681 of *Lecture Notes in Computer Science*, pages 388–400. Springer, 2009.

[BLR08] Benjamin Berkels, Gina Linkmann, and Martin Rumpf. A shape median based on symmetric area differences. In Oliver Deussen, Daniel Keim, and Dietmar Saupe, editors, *Vision, Modeling, and Visualization Proceedings*, pages 399–407, 2008.

[BLR10] Benjamin Berkels, Gina Linkmann, and Martin Rumpf. An $SL(2)$ invariant shape median. *Journal of Mathematical Imaging and Vision*, 37(2):85–97, June 2010.

[BRRV07] Benjamin Berkels, Andreas Rätz, Martin Rumpf, and Axel Voigt. Identification of grain boundary contours at atomic scale. In *Proceedings of the First International Conference on Scale Space Methods and Variational Methods in Computer Vision (SSVM 2007)*, volume 4485 of *Lecture Notes in Computer Science*, pages 765–776. Springer, 2007.

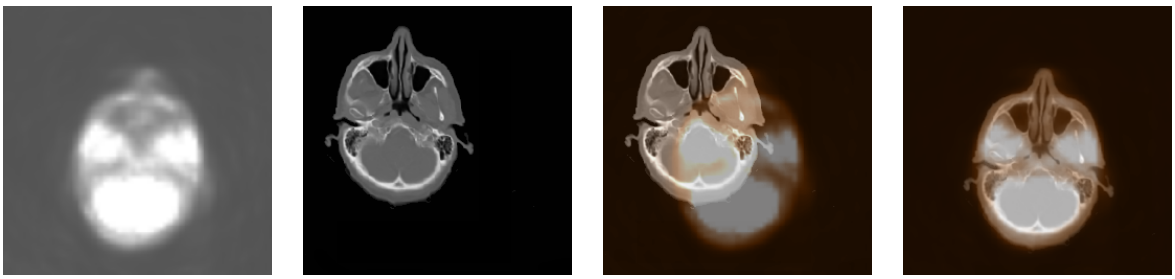
[BRRV08] Benjamin Berkels, Andreas Rätz, Martin Rumpf, and Axel Voigt. Extracting grain boundaries and macroscopic deformations from images on atomic scale. *Journal of Scientific Computing*, 35(1):1–23, 2008.

[HBD+07] Jingfeng Han, Benjamin Berkels, Marc Droske, Joachim Hornegger, Martin Rumpf, Carlo Schaller, Jasmin Scorzin, and Horst Urbach. Mumford-Shah model for one-to-one edge matching. *IEEE Transactions on Image Processing*, 16(11):2720–2732, 2007.

[HBR+06] Jingfeng Han, Benjamin Berkels, Martin Rumpf, Joachim Hornegger, Marc Droske, Michael Fried, Jasmin Scorzin, and Carlo Schaller. A variational framework for joint image registration, denoising and edge detection. In *Bildverarbeitung für die Medizin 2006*, pages 246–250. Springer, March 2006.

2 A Mumford–Shah model for one-to-one edge matching

LET $g_\mathrm{R}, g_\mathrm{T} : \Omega \to \mathbb{R}$ be a reference and a template image respectively. The task of finding a deformation $\phi : \Omega \to \Omega$ such that $g_\mathrm{T} \circ \phi$ corresponds to g_R is called image registration or matching, and ϕ is called the deformation from g_T to g_R. The easiest registration problem is monomodal image registration. Here, the aforementioned correspondence means $g_\mathrm{T} \circ \phi = g_\mathrm{R}$. In many applications, for example if the images were acquired using different modalities, e. g. *X-ray computed tomography* (CT) and *positron emission tomography* (PET), one has to look for a more general correspondence, cf. Figure 2.1.



g_R g_T $g_\mathrm{R}/g_\mathrm{T}$ $g_\mathrm{R}/g_\mathrm{T} \circ \phi$

Figure 2.1: Multimodal matching example of a PET image (g_R) and a CT image (g_T). Here, the deformation ϕ was obtained using the one-to-one edge matching approach we present in this chapter. Due to the joint handling of the edge detection and the edge-based matching, the proposed method is able to perform the matching even though the PET image does not have any discernable edges.

In [58], Droske et al. proposed to use the Mumford–Shah model in the context of image registration. The main idea of this approach is to simultaneously segment two images with a shared edge set. It is modeled by the functional

$$\begin{aligned} E[u_\mathrm{R}, u_\mathrm{T}, K_\mathrm{T}, \phi] = {} & \int_\Omega (u_\mathrm{T} - g_\mathrm{T})^2 \,\mathrm{d}x + \mu \int_{\Omega \setminus K_\mathrm{T}} |\nabla u_\mathrm{T}|^2 \,\mathrm{d}x + \nu \mathcal{H}^{d-1}(K_\mathrm{T}) + E_\mathrm{reg}[\phi] \\ & + \int_\Omega (u_\mathrm{R} - g_\mathrm{R})^2 \,\mathrm{d}x + \mu \int_{\Omega \setminus \phi^{-1}(K_\mathrm{T})} |\nabla u_\mathrm{R}|^2 \,\mathrm{d}x, \end{aligned}$$

where u_R and u_T are piecewise smooth functions, i. e. cartoon approximations of g_R and g_T, and $K_\mathrm{T} \subset \Omega$ acts as the edge set of u_T and $\phi^{-1}(K_\mathrm{T})$ as the edge set of u_R. Because of $\chi_{\phi^{-1}(K_\mathrm{T})} = \chi_{K_\mathrm{T}} \circ \phi$, we have

$$\int_{\Omega \setminus \phi^{-1}(K_\mathrm{T})} |\nabla u_\mathrm{R}|^2 \,\mathrm{d}x = \int_\Omega (1 - (\chi_{K_\mathrm{T}} \circ \phi)) \, |\nabla u_\mathrm{R}|^2 \,\mathrm{d}x = \int_\Omega (\chi_{\Omega \setminus K_\mathrm{T}} \circ \phi) \, |\nabla u_\mathrm{R}|^2 \,\mathrm{d}x \,.$$

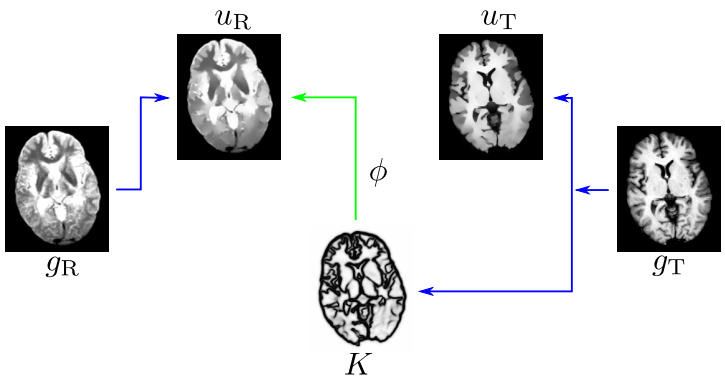


Figure 2.2: Schematic view of the non-symmetric Mumford–Shah model for edge matching. g_R and g_T denote reference and template image. u_R and u_T are the piecewise smooth approximations of g_R and g_T respectively. K is the combined discontinuity set of both images. ϕ represents the deformation matching u_T to u_R.

Therefore, this model makes the edges of $g_T \circ \phi$ correspond to the edges of g_R, hence the smooth deformation ϕ establishes a mapping between the edge features of the input images. Here, the modified Mumford–Shah model simultaneously handles edge segmentation and non-rigid registration, two highly interdependent tasks. The main benefit and motivation to use such kind of joint models is that any knowledge on the solution of one task can be used to improve the solution of the other task. This benefit of joint approaches in the context of segmentation and registration has already been exploited by Yezzi, Zöllei and Kapur [81]: They use an active contour model, similar to the one proposed by Chan and Vese [46] (cf. Section 1.2.2), to simultaneously segment and register multiple images, by evolving a single contour as well as affine deformations of this contour to the edge features of each of the images.

In tasks where g_R and g_T have roughly the same (albeit deformed) edges, a major drawback of the above Mumford–Shah based matching is its asymmetric nature with respect to edge features and the spatial mapping between them. Figure 2.2 depicts the underlying scheme of the model. The similarity measure is not symmetric because a single discontinuity set K is used to represent two edge sets, the edges of the restored template image u_T and the edges of the restored deformed reference image $u_R \circ \phi^{-1}$. Furthermore, as illustrated in Figure 2.2, the deformation ϕ between the two images is only defined in one direction, from g_T to g_R. As pointed out in [108], an asymmetric similarity measure and a single directional deformation are not enough to ensure the consistency of the method, i. e., if it is used to compute the deformation ϕ from g_T to g_R and then the roles of g_T and g_R are switched to compute the deformation ψ from g_R to g_T with the same method, the obtained deformations are not necessarily inverse to each other.

To resolve the asymmetric nature we propose a symmetric edge matching model [74, 73, 72] again based on the Mumford–Shah model. Figure 2.3 depicts the underlying scheme of this symmetric model. The symmetric model uses two separate discontinuity sets, denoted by K_R and K_T in Figure 2.3, that explicitly represent the edge sets of u_R and u_T respectively. To account for the correspondence ambiguity, we pick up the idea of consistent registration by Christensen and Johnson [48, 80]: The deformations in both directions are simultaneously estimated while a penalty term constrains each deformation to be the inverse of the other one.

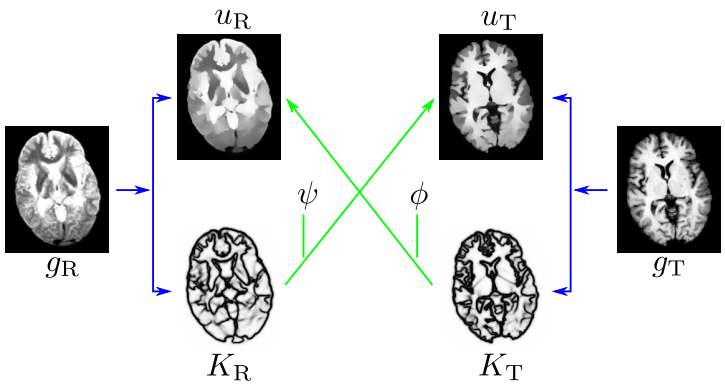


Figure 2.3: Schematic view of the symmetric Mumford–Shah model for one-to-one edge matching. g_R and g_T denote reference and template image. u_R and u_T are the piecewise smooth approximations of g_R and g_T respectively. K_R and K_T are the discontinuity sets of u_R and u_T respectively. ϕ represents the transformation from u_T to u_R, ψ represents the transformation from u_R to u_T.

Hereby, both edge sets K_R and K_T have equal influence on the edge matching and switching the roles of g_T and g_R just switches the roles of ϕ and ψ. In this sense, the proposed symmetric matching approach determines one-to-one correspondences between the edge features of the two input images. Therefore, it is suitable for a broad range of applications where the correspondence of the same structure in different images needs to be determined (e. g. non-rigid registration for atlas construction [110, 94], biological images [119, 6] or motion estimation [101]).

Before we start to develop the functionals for the symmetric model, one should note that it is difficult to minimize the original Mumford–Shah functional (1.1) because of its explicit handling of the discontinuity set K. Various approximations have been proposed during the last two decades. For the registration model we focus on the phase field based Ambrosio–Tortorelli approximation with elliptic functionals [4] (cf. Section 1.2.1). Another very important approach was proposed by Chan and Vese [46] (cf. Section 1.2.2) and plays an important role in the other chapters of this work. For a comparison of these two methods in the context of edge-based image registration we refer to [61]. A different way to extend the registration model proposed by Droske et al. [58] is to match the regular morphology in addition to the singular morphology, i. e. the edges, cf. [60].

This chapter is organized as follows: In Section 2.1, we introduce our one-to-one edge matching model by constructing the necessary functionals and the corresponding variational formulation. Afterwards, the minimization algorithm and the numerical implementation are discussed in Section 2.2. Finally, in Section 2.3, we study the influence of the parameters on the algorithm and show experimental results for various applications.

2.1 One-to-one edge matching

As already mentioned in the beginning of this chapter, image registration is the following task: Given a reference and a template image denoted by g_R and g_T respectively, find a suitable deformation ϕ such that the deformed template image $g_T \circ \phi$ becomes as similar to the reference image g_R as possible, cf. [98]. The key point here is to choose a way to measure this similarity

(or dissimilarity) that is appropriate for the class of registration problems that one wants to solve. There is a multitude of different similarity measures, usually involving the gray values g_R and g_T directly or certain features such as edges derived from the gray values.

Building on the Mumford–Shah based registration model by Droske et al. [58], we consider an edge-based matching method. First, the method needs to extract the edge features from the gray values of the images and simply employs the Mumford–Shah model to tackle this task. In practice, the discontinuity sets that encode the edges are approximated by phase field functions, cf. Section 1.2.1. Since we need to extract the edges of both images, this involves four unknowns $\{u_R, u_T, v_R, v_T\}$, one pair (u_R, v_R) for g_R and one pair (u_T, v_T) for g_T. Second, the method needs to do the actual registration by using the aforementioned edges. We denote the deformation from g_T to g_R by ϕ and the deformation from g_R to g_T by ψ. Basically, the model from [58] is used twice to obtain both of these deformations, i. e. ϕ and ψ are supposed to match the two feature representations (u_T, v_T) and (u_R, v_R) to u_R and u_T respectively, but some special precautions need to be taken instead of handling the deformations more or less independently. Otherwise, we may end up with a ψ that considerably differs from the inverse of ϕ. In order to overcome such correspondence ambiguities, we follow the method of consistent registration [48] and jointly estimate the deformations in both directions. This involves using a consistency energy term ensuring that the deformations are approximately inverse to each other. Finally, the full model is supposed to do both, the edge extraction and the edge registration, simultaneously making use of the fact that both tasks are highly interdependent.

For reasons of practicability, we allow ϕ and ψ to map from Ω to $\mathbb{R}^d$ instead of restricting their range to Ω. This is accompanied by an extension of all unknowns from Ω to $\mathbb{R}^d$ by zero, e. g. $v_T(\phi(x)) = 0$ if $\phi(x) \notin \Omega$.

2.1.1 Construction of the energy

To encode the aforementioned desired properties of the six unknowns $\{u_R, u_T, v_R, v_T, \phi, \psi\}$ we construct an energy such that the unknowns can be obtained by minimizing a joint energy functional E_{SYM}. This functional consists of different terms responsible for the different desired properties and is of the following structure:

$$E_{SYM} = E_{ED} + \mu E_{MA} + \lambda E_{REG} + \kappa E_{CON}. \tag{2.1}$$

μ, λ and κ are nonnegative constant parameters that allow to control the contributions of the associated functionals. In the following, we give a detailed definition of these functionals.

Edge-detection functional

As already pointed out, the edge detection is based on E^ϵ_{AT}, the Ambrosio–Tortorelli functional defined in equation (1.3). To express the dependence of E^ϵ_{AT} on the input image u_0, we write E^{ϵ,u_0}_{AT} instead of just E^ϵ_{AT}. This notation allows us to define the edge-detection functional as follows:

$$E_{ED}[u_R, v_R, u_T, v_T] := E^{\epsilon,g_R}_{AT}[u_R, v_R] + E^{\epsilon,g_T}_{AT}[u_T, v_T]. \tag{2.2}$$

Either of the two E^ϵ_{AT} instances uses the mechanisms of the Ambrosio–Tortorelli approximation to obtain the feature representation (u_R, v_R) or (u_T, v_T) of the input image g_R or g_T respectively, such that the piecewise smooth function u_R or u_T couples with the corresponding phase field

function v_R or v_T as described by equation (1.4). Roughly speaking, E_{ED} handles the detecting of the edge features of both images and establishes the relationship between the phase field function v_R or v_T and the corresponding piecewise smooth function u_R or u_T.

Note that the segmented edge features of the two images, (u_R, v_R) and (u_T, v_T), are totally independent of each other in E_{ED}, i. e. changing either g_R or g_T has no influence on (u_T, v_T) or (u_R, v_R) respectively.

Matching functional

E_{MA} is responsible for matching the edge features of the two images. It is defined picking up the ideas of [58]:

$$\begin{aligned} E_{MA}[u_R, v_R, u_T, v_T, \phi, \psi] &:= C_{MA}[u_R, v_T, \phi] + C_{MA}[u_T, v_R, \psi] \\ &:= \frac{1}{2}\int_\Omega (v_T \circ \phi)^2 \left|\nabla u_R\right|^2 \mathrm{d}x + \frac{1}{2}\int_\Omega (v_R \circ \psi)^2 \left|\nabla u_T\right|^2 \mathrm{d}x\,. \end{aligned} \tag{2.3}$$

It favors deformations ϕ and ψ which couple the feature representations (u_R, v_R) and (u_T, v_T) such that $v_T \circ \phi \approx 0$ where $|\nabla u_R| \gg 0$ and $v_R \circ \psi \approx 0$ where $|\nabla u_T| \gg 0$. Combined with the phase field length terms for v_R and v_T from E_{ED}, the following coupling is induced (similar to equation (1.4)):

$$v_T \circ \phi \approx \begin{cases} 0 & \text{where } |\nabla u_R| \gg 0, \\ 1 & \text{where } |\nabla u_R| \approx 0. \end{cases}$$

$$v_R \circ \psi \approx \begin{cases} 0 & \text{where } |\nabla u_T| \gg 0, \\ 1 & \text{where } |\nabla u_T| \approx 0. \end{cases}$$

By construction, the matching functional treats segmentation and registration in a joint manner: The registration is taken care of since the functional acts as a similarity measure based on the correspondence of the edge features of the images to each other. Instead of the naive approach to directly match the phase fields ($v_R \leftrightarrow v_T$) and the smooth functions ($u_R \leftrightarrow u_T$) to each other, E_{MA} aims at bringing the gradient of each of the smoothed images in correspondence to the phase field of the respective other image ($v_R \leftrightarrow \nabla u_T, v_T \leftrightarrow \nabla u_R$). This frees the functional from relying on a direct relationship between the gray values of the images and enables the method to handle certain kinds of multimodal registration problems. Furthermore, the coupling of the edge features segmented from one image to the other image introduced by E_{MA} gives the functional a direct influence on the segmentation.

Note that this functional does not guarantee any local correspondence of edge features: Without any further constraints on the transformations ϕ and ψ, E_{MA} allows ϕ, for instance, to couple all edges of u_R to a single point where v_T vanishes.

Deformation regularization functional

To establish a local edge feature correspondence we introduce a spatial regularization for both transformations:

$$\begin{aligned} E_{REG}[\phi, \psi] &:= C_{REG}[\phi] + C_{REG}[\psi] \\ &:= \frac{1}{2}\int_\Omega \|D(\phi - \mathrm{id})\|^2 \mathrm{d}x + \frac{1}{2}\int_\Omega \|D(\psi - \mathrm{id})\|^2 \mathrm{d}x\,. \end{aligned} \tag{2.4}$$

Here, $\mathrm{id} : \Omega \to \mathbb{R}^d, x \mapsto x$ denotes the identity mapping and $\|A\|$ denotes the Frobenius norm on matrices. Therefore, $\phi - \mathrm{id}$, $\psi - \mathrm{id}$ are the displacement fields corresponding to the deformations ϕ and ψ. Establishing a local edge feature correspondence is not the only task of E_{REG}. It is also supposed to prevent deformations with singularities like cracks, foldings, or other undesired properties. For the sake of simplicity, we confine in this work to a simple regularizer in form of the sum of the norm of the Jacobian of both displacement fields (cf. [9] for further information on this kind of regularizations).

Various more sophisticated regularizers that can be used in the context of non-rigid registration have been proposed in the literature, e. g. linear elastic [32, 49] and viscous fluid [31, 49] regularizations. Both make use of the corresponding continuous mechanical models [71]. Another alternative is the nonlinear hyperelastic, polyconvex regularization used in [59]. It separately cares about length, area and volume deformation and especially penalizes volume shrinkage. A major advantage of this approach is that it already ensures a homeomorphism property of the regularized deformation [59, 15, 16].

Deformation consistency functional

With the energy functionals defined so far, there is almost no coupling between ϕ and ψ. With respect to E_{ED} and E_{REG}, the two transformations are completely independent of each other. Only E_{MA} imposes an implicit correlation via the matching of the two image and phase field pairs, i. e. $(u_{\mathrm{R}}, v_{\mathrm{T}} \circ \phi) \leftrightarrow (u_{\mathrm{T}}, v_{\mathrm{R}} \circ \psi)$. The missing explicit relationship between ϕ and ψ is encoded in the consistency functional E_{CON}:

$$\begin{aligned} E_{\mathrm{CON}} &:= C_{\mathrm{CON}}[\phi, \psi] + C_{\mathrm{CON}}[\psi, \phi] \\ &:= \frac{1}{2} \int_\Omega |\phi \circ \psi(x) - x|^2 \,\mathrm{d}x + \frac{1}{2} \int_\Omega |\psi \circ \phi(x) - x|^2 \,\mathrm{d}x \,. \end{aligned} \tag{2.5}$$

Unlike E_{REG}, E_{CON} is a classical penalty term: Ideally each deformation should be the inverse of the respective deformation in the other direction, i. e. the deformations should fulfill $\phi = \psi^{-1}$ and $\psi = \phi^{-1}$. This can be expressed as the pointwise property $\phi \circ \psi(x) = x = \psi \circ \phi(x)$ for all $x \in \Omega$. Instead of explicitly enforcing this property, E_{CON} penalizes deviations from it, introducing a soft constraint controlled by the penalty parameter κ in (2.1). Therefore, this penalty functional implicitly encourages a bijective edge feature correspondence.

2.1.2 Variational formulation

To find a minimizer of the entire energy E_{SYM} we look for a zero crossing of its variation with respect to all the unknowns $\{u_{\mathrm{R}}, u_{\mathrm{T}}, v_{\mathrm{R}}, v_{\mathrm{T}}, \phi, \psi\}$. The definition of the entire functional E_{SYM} as well as its components E_{ED}, E_{MA}, E_{REG} and E_{CON} is completely symmetric with respect to the two groups of unknowns, each corresponding to one of the input images: $\{u_{\mathrm{R}}, v_{\mathrm{R}}, \phi\}$ and $\{u_{\mathrm{T}}, v_{\mathrm{T}}, \psi\}$. Thus, we can confine to discussing the variations with respect to $\{u_{\mathrm{R}}, v_{\mathrm{R}}, \phi\}$. The variations with respect the other group are then obtained analogously.

For an arbitrary scalar test function $\vartheta \in C_0^\infty(\Omega)$, we get

$$\begin{aligned} \langle \partial_{u_{\mathrm{R}}} E_{\mathrm{SYM}}, \vartheta \rangle &= \langle \partial_{u_{\mathrm{R}}} E_{\mathrm{AT}}, \vartheta \rangle + \langle \partial_{u_{\mathrm{R}}} E_{\mathrm{MA}}, \vartheta \rangle \\ &= \int_\Omega \alpha(u_{\mathrm{R}} - g_{\mathrm{R}})\vartheta + \beta v_{\mathrm{R}}^2 \nabla u_{\mathrm{R}} \cdot \nabla \vartheta + \mu (v_{\mathrm{T}} \circ \phi)^2 \nabla u_{\mathrm{R}} \cdot \nabla \vartheta \;\mathrm{d}x, \end{aligned} \tag{2.6}$$

$$\begin{aligned}\langle \partial_{v_\mathrm{R}} E_{\mathrm{SYM}}, \vartheta \rangle &= \langle \partial_{v_\mathrm{R}} E_{\mathrm{AT}}, \vartheta \rangle + \langle \partial_{v_\mathrm{R}} E_{\mathrm{MA}}, \vartheta \rangle \\ &= \int_\Omega \beta \left|\nabla u_\mathrm{R}\right|^2 v_\mathrm{R} \vartheta \; + \frac{\nu}{4\epsilon}(v_\mathrm{R} - 1)\vartheta \; + \nu\epsilon \nabla v_\mathrm{R} \cdot \nabla \vartheta \,\mathrm{d}x \\ &\quad + \int_\Omega \mu \left|\nabla u_\mathrm{T}\right|^2 (v_\mathrm{R} \circ \psi)(\vartheta \circ \psi) \,\mathrm{d}x \,.\end{aligned}$$

In view of the Finite Element method we are going to use for the spatial discretization (cf. Section 2.2), the above formulation of the variation with respect to v_R is not optimal. The deformed test function $\vartheta \circ \psi$ in the integrand of the last term alters the support of the test function, nullifying some of the advantages of the FE method. The following lemma allows us to get rid of the need to treat deformed test functions:

2.1.1 Lemma (Transformation rule for zero extensions)**.** *Let $f, g \in L^2(\Omega)$ and $\psi : \Omega \to \psi(\Omega)$ be a C^1-diffeomorphism. Extend $(f \circ \psi^{-1}) \left|\det D\psi^{-1}\right| : \psi(\Omega) \to \mathbb{R}$ and $g : \Omega \to \mathbb{R}$ to $\mathbb{R}^d$ by zero, i. e. $(f \circ \psi^{-1})(x) \left|\det D\psi^{-1}(x)\right| = 0$ for $x \notin \psi(\Omega)$ and $g(x) = 0$ for $x \notin \Omega$. Then*

$$\int_\Omega f(x)\,(g \circ \psi)(x) \,\mathrm{d}x = \int_\Omega (f \circ \psi^{-1})(x)\, g(x) \left|\det D\psi^{-1}(x)\right| \mathrm{d}x \,.$$

Proof. Denote $(f \circ \psi^{-1}) \left|\det D\psi^{-1}\right|$ by h. Using the standard transformation rule, one obtains

$$\begin{aligned}\int_\Omega f(x)\,(g \circ \psi)(x) \,\mathrm{d}x &= \int_{\psi(\Omega)} (f \circ \psi^{-1})(x)\, g(x) \left|\det D\psi^{-1}(x)\right| \mathrm{d}x = \int_{\psi(\Omega)} hg \,\mathrm{d}x \\ &= \int_{\psi(\Omega)\cap\Omega} hg \,\mathrm{d}x + \underbrace{\int_{\psi(\Omega)\cap(\mathbb{R}^d\setminus\Omega)} hg \,\mathrm{d}x}_{=0\,(g\equiv 0 \text{ in } \mathbb{R}^d\setminus\Omega)} = \int_{\psi(\Omega)\cap\Omega} hg \,\mathrm{d}x \\ &= \int_{\psi(\Omega)\cap\Omega} hg \,\mathrm{d}x + \underbrace{\int_{(\mathbb{R}^d\setminus\psi(\Omega))\cap\Omega} hg \,\mathrm{d}x}_{=0\,(h\equiv 0 \text{ in } \mathbb{R}^d\setminus\psi(\Omega)} = \int_\Omega hg \,\mathrm{d}x \,.\end{aligned}$$

□

Using the zero extension of v_R to $\mathbb{R}^d$ and Lemma 2.1.1, we get

$$\begin{aligned}\langle \partial_{v_\mathrm{R}} E_{\mathrm{SYM}}, \vartheta \rangle &= \int_\Omega \beta \left|\nabla u_\mathrm{R}\right|^2 v_\mathrm{R} \vartheta \; + \frac{\nu}{4\epsilon}(v_\mathrm{R} - 1)\vartheta \; + \nu\epsilon \nabla v_\mathrm{R} \cdot \nabla \vartheta \,\mathrm{d}x \\ &\quad + \int_\Omega \mu \left|\nabla u_\mathrm{T} \circ \psi^{-1}\right|^2 v_\mathrm{R} \vartheta \left|\det D\psi^{-1}\right| \mathrm{d}x \,.\end{aligned} \tag{2.7}$$

Here, Lemma 2.1.1 also gives a meaning to the integrand of the last term where ψ^{-1} is not defined, i. e. in $(\mathbb{R}^d \setminus \psi(\Omega)) \cap \Omega$.

For an arbitrary vector-valued test function $\zeta \in C_0^\infty(\Omega, \mathbb{R}^d)$, using Lemma 2.1.1 and

$$(\psi \circ \phi - \mathrm{id})^T ((D\psi) \circ \phi)\zeta = ((D\psi) \circ \phi)^T (\psi \circ \phi - \mathrm{id}) \cdot \zeta,$$

we get

$$\begin{aligned}\langle \partial_\phi E_{\mathrm{SYM}}, \zeta\rangle &= \langle \partial_\phi E_{\mathrm{MA}}, \zeta\rangle + \langle \partial_\phi E_{\mathrm{REG}}, \zeta\rangle + \langle \partial_\phi E_{\mathrm{CON}}, \zeta\rangle \\ &= \int_\Omega \mu \left|\nabla u_{\mathrm{R}}\right|^2 (v_{\mathrm{T}} \circ \phi)((\nabla v_{\mathrm{T}}) \circ \phi) \cdot \zeta + \lambda D(\phi - \mathrm{id}) : D\zeta \,\mathrm{d}x \\ &\quad + \int_\Omega \kappa \left|\det D\psi^{-1}\right| (\phi - \psi^{-1}) \cdot \zeta \,\mathrm{d}x \\ &\quad + \int_\Omega \kappa((D\psi) \circ \phi)^T (\psi \circ \phi - \mathrm{id}) \cdot \zeta \,\mathrm{d}x\,.\end{aligned} \tag{2.8}$$

Here, $A : B = \mathrm{tr}(A^T B)$ for $A, B \in \mathbb{R}^{d\times d}$.

2.2 Minimization algorithm

To deal with the high complexity of the minimization problem (six unknown functions, two of them vector-valued) the unknowns are estimated in an expectation-maximization (EM) like manner, also known as alternating minimization. For a general energy E depending on m unknown functions $f_1, \ldots, f_m$ and a given estimate of the unknown functions, one step of the generic EM procedure replaces f_i by $\operatorname{argmin}_f E[f_1, \ldots, f_{i-1}, f, f_{i+1}, \ldots, f_m]$ for $1, \ldots, m$.

This approach not only reduces the computational complexity, but also allows us to take advantage of the fact that the variations with respect to the images and the phase fields are linear in the corresponding unknown.

2.2.1 Solution of the linear part

Noting that $\partial_{u_{\mathrm{R}}} E_{\mathrm{SYM}}$ and $\partial_{v_{\mathrm{R}}} E_{\mathrm{SYM}}$ are linear in u_{R} and v_{R} respectively (cf. equations (2.6) and (2.7)), after spatial discretization their zero-crossings can simply be calculated by solving the corresponding systems of linear equations. Following the FE method (cf. Section 6.1) all continuous functions are replaced by their FE approximations, e. g. g_{R} and u_{R} by $G_{\mathrm{R}}(x) = \sum_{i=1}^n \overline{G_{\mathrm{R}}}^i \Lambda_i(x)$ and $U_{\mathrm{R}}(x) = \sum_{i=1}^n \overline{U_{\mathrm{R}}}^i \Lambda_i(x)$. Finding a zero crossing of (2.6) in the FE space is equivalent to solving

$$\begin{aligned}&\alpha \sum_{i=1}^n \overline{U_{\mathrm{R}}}^i \int_\Omega \Lambda_i(x)\Lambda_j(x)\,\mathrm{d}x \\ &+ \beta \sum_{i=1}^n \overline{U_{\mathrm{R}}}^i \int_\Omega V_{\mathrm{R}}^2(x) \nabla\Lambda_i(x) \cdot \nabla\Lambda_j(x)\,\mathrm{d}x \\ &+ \mu \sum_{i=1}^n \overline{U_{\mathrm{R}}}^i \int_\Omega (V_{\mathrm{T}} \circ \Phi)^2(x) \nabla\Lambda_i(x) \cdot \nabla\Lambda_j(x)\,\mathrm{d}x \\ &= \alpha \sum_{i=1}^n \overline{G_{\mathrm{R}}}^i \int_\Omega \Lambda_i(x)\Lambda_j(x)\,\mathrm{d}x \qquad \text{for all } 1 \le j \le n.\end{aligned} \tag{2.9}$$

Using the definitions of generalized mass (6.1) and stiffness matrices (6.2), equation (2.9) is equivalent to

$$\left(\alpha M + \beta L\left[V_{\mathrm{R}}^2\right] + \mu L\left[(V_{\mathrm{T}} \circ \Phi)^2\right]\right) \overline{U_{\mathrm{R}}} = \alpha M \overline{G_{\mathrm{R}}}. \tag{2.10}$$

Similarly, (2.7) leads to

$$\left(\mu M\left[\left|\nabla U_{\mathrm{T}}\circ\Psi^{-1}\right|^2\left|\det D\Psi^{-1}\right|\right]+\beta M\left[\left|\nabla U_{\mathrm{R}}\right|^2\right]+\frac{\nu}{4\epsilon}M+\nu\epsilon L\right)\overline{V_{\mathrm{R}}}=\frac{\nu}{4\epsilon}M\,\overline{1}\,. \tag{2.11}$$

Here $\overline{1}$ denotes the one-vector, i. e. $(1,\dots,1)^T$. Analogously, one obtains the linear systems for $\overline{U_{\mathrm{T}}}$ and $\overline{V_{\mathrm{T}}}$

$$\left(\alpha M+\beta L\left[V_{\mathrm{T}}^2\right]+\mu L\left[(V_{\mathrm{R}}\circ\Psi)^2\right]\right)\overline{U_{\mathrm{T}}}=\alpha M\overline{G_{\mathrm{T}}}. \tag{2.12}$$

$$\left(\mu M\left[\left|\nabla U_{\mathrm{R}}\circ\Phi^{-1}\right|^2\left|\det D\Phi^{-1}\right|\right]+\beta M\left[\left|\nabla U_{\mathrm{T}}\right|^2\right]+\frac{\nu}{4\epsilon}M+\nu\epsilon L\right)\overline{V_{\mathrm{T}}}=\frac{\nu}{4\epsilon}M\,\overline{1}\,. \tag{2.13}$$

In the implementation, the linear systems (2.10) to (2.13) are solved with a conjugate gradient method using SSOR preconditioning.

2.2.2 Solution of the nonlinear part

Unlike $\partial_{u_{\mathrm{R}}}E_{\mathrm{SYM}}$ and $\partial_{v_{\mathrm{R}}}E_{\mathrm{SYM}}$, $\partial_\phi E_{\mathrm{SYM}}$ is not linear (cf. (2.8)). Thus we cannot find a zero-crossing of (2.8) by just solving a linear system. Instead, we employ the following explicit gradient flow scheme (cf. Section 6.2) to approximate the zero-crossing iteratively:

$$\Phi^{k+1}=\Phi^k-\tau^k\cdot\operatorname{grad}_\Phi^{g_\sigma}E_{\mathrm{SYM}}[\Phi^k]. \tag{2.14}$$

Here, $\operatorname{grad}_\Phi^{g_\sigma}E_{\mathrm{SYM}}[\Phi^k]$ denotes the gradient of E_{SYM} with respect to the deformation Φ and a metric g_σ, i. e. it is characterized by fulfilling

$$g_\sigma\left(\operatorname{grad}_\Phi^{g_\sigma}E_{\mathrm{SYM}}[\Phi^k],\zeta\right)=\langle\partial_\phi E_{\mathrm{SYM}},\zeta\rangle\text{ for all }\zeta\in C_0^\infty(\Omega,\mathbb{R}^d),$$

and τ^k is a step size yet to be determined.

We choose the metric, inspired by the Sobolev active contour approach [120], to be a scaled version of the H^1 metric, i. e.

$$g_\sigma(\Phi_1,\Phi_2)=(\Phi_1,\Phi_2)_{L^2}+\frac{\sigma^2}{2}(D\Phi_1,D\Phi_2)_{L^2}.$$

σ represents a filter width of the corresponding time discrete and implicit heat equation filter kernel. In Section 6.2 we give further explanations on the influence of the metric and the regularizing effects of g_σ. Still we want to emphasize here that the choice of the metric does not alter the energy landscape itself in any way, but solely the descent path towards the set of minimizers.

The actual computation of $\operatorname{grad}_\Phi^{g_\sigma}E_{\mathrm{SYM}}[\Phi^k]$ in our implementation is done in two steps:

- Compute the discrete variation, given by

$$\partial_\Phi E_{\mathrm{SYM}}[\Phi^k]=\left(\left\langle\partial_\Phi E_{\mathrm{SYM}}[\Phi^k],\Lambda_i e_j\right\rangle\right)_{1\le i\le n,1\le j\le d}\in\mathbb{R}^{nd}.$$

- A straightforward calculation shows that the representation of g_σ in FE-terms is

$$g_\sigma(\Phi_1,\Phi_2)=\left(M_{\mathrm{bl}}+\tfrac{\sigma^2}{2}L_{\mathrm{bl}}\right)\overline{\Phi_1}\cdot\overline{\Phi_2}, \tag{2.15}$$

therfore

$$\operatorname{grad}_{\Phi}^{g_\sigma} E_{\mathrm{SYM}}[\Phi^k] = \left(M_{\mathrm{bl}} + \tfrac{\sigma^2}{2} L_{\mathrm{bl}}\right)^{-1} \partial_\Phi E_{\mathrm{SYM}}[\Phi^k]$$

holds. Here, M_{bl} and L_{bl} denote $d \times d$ block matrices with the standard mass and stiffness matrices, respectively, on the diagonal block positions and zero matrices on the remaining block positions. For all results presented in this chapter, we use $\sigma = \sqrt{10h}$, where h denotes the mesh resolution, cf. Section 6.1. The solution of the linear system is approximated by a single V-cycle of a multigrid solver [29, 130]. We do not need the exact solution, but only the regularizing effect of the inverse of the metric representation. Thus one V-cycle is sufficient.

We employ the Armijo rule [25] to determine the step size of the gradient flow and give a detailed description of this approach in Section 6.3. The parameters are chosen as $\sigma = \frac{1}{4}$ and $\beta = \frac{1}{2}$.

The natural way to handle the deformations indicated by the EM procedure is to update each deformation individually and particularly to determine the step size for each deformation separately, i. e. estimating τ_Φ for Φ and then estimating τ_Ψ for Ψ after updating Φ. However, if Φ and Ψ are treated sequentially, the consistency energy (2.5) significantly limits τ_Φ and τ_Ψ, because large individual step sizes result in a significant enlargement of the consistency energy. Fortunately, this shortcoming can easily be avoided: Instead of treating Φ and Ψ separately, we treat both combined as a single unknown in our EM procedure. Hence we use the gradient flow scheme

$$\begin{bmatrix} \Phi^{k+1} \\ \Psi^{k+1} \end{bmatrix} = \begin{bmatrix} \Phi^{k} \\ \Psi^{k} \end{bmatrix} - \tau^k \begin{bmatrix} \operatorname{grad}_{\Phi}^{g_\sigma} E[\Phi^k, \Psi^k] \\ \operatorname{grad}_{\Psi}^{g_\sigma} E[\Phi^k, \Psi^k] \end{bmatrix} \tag{2.16}$$

to update the transformations. Since Φ and Ψ are updated simultaneously, the consistency energy does not necessarily forbid large step sizes.

Compared to a classical gradient descent with fixed step size, the regularized gradient flow with adaptive step size control performs significantly better. The step size control noticeably reduces the amount of necessary descent steps and at least experimentally ensures convergence. In the experiments in this chapter, we use five gradient flow steps to update the deformations in each iteration of the EM procedure.

2.2.3 Cascadic descent approach

One drawback of our joint energy functional E_{SYM}, typical for non-rigid registration functionals, is that it has many local minimizers. Furthermore, the whole EM procedure and the gradient flow used to update the transformations both are attracted to the "nearest" local minimizer. In order to avoid being dragged into undesirable local minima, we employ a spatially cascadic scheme. In a nutshell, we start by calculating a minimizer with the EM procedure on a very coarse spatial resolution, prolongate the minimizer to a finer resolution and then repeat the minimizing and prolongation steps till we reach the resolution of the input data. The coarser the resolution, the fewer local structures prevail in the input data. Hence, the cascadic scheme segments and matches global structures before local ones.

To conveniently handle the prolongation and restriction needed for the cascadic scheme, we use a special case of meshes to discretize the image domain $\Omega := [0, 1]^d$ described in Section 6.1.

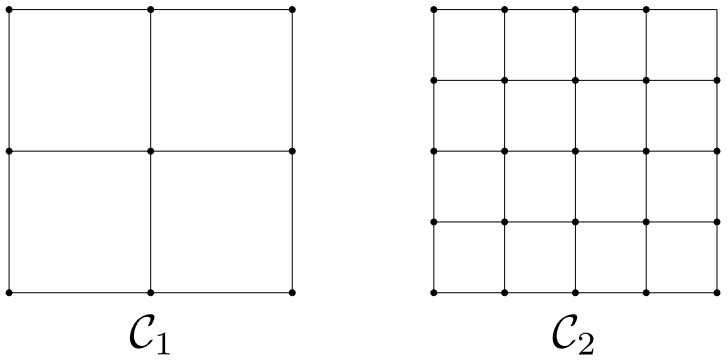


Figure 2.4: 2D nested mesh hierarchy on a uniform rectangular mesh: All nodes of the coarse mesh $\mathcal{C}_1$ are also nodes of the fine mesh $\mathcal{C}_2$.

The uniform rectangular mesh is chosen such that it has $2^m + 1$ equidistant nodes along each axis, hence $n := (2^m + 1)^d$ nodes in total. The mesh is denoted by $\mathcal{C}_m$ and m is called the level of the mesh. By construction, these meshes are nested in each other in an ascending manner, i. e. $\mathcal{N}_{m-1} \subset \mathcal{N}_m$ where $\mathcal{N}_m$ denotes the set of nodes of $\mathcal{C}_m$, cf. Figure 2.4. Although such a nested mesh hierarchy is not natural for finite difference methods, where commonly discrete images with 2^m pixels or voxels along each axis are used, it is the commonly used, canonical hierarchy in the Finite Element context. Due to the nestedness of the meshes, the prolongation from one level to the next higher level can be done in a simple and straightforward manner: To prolongate a discrete function from level $m-1$ to m we just need to determine its value on the nodes $\mathcal{N}_m$. This is done by evaluating the function on these nodes. Based on the construction of our FE spaces, this means that the function values on the nodes $\mathcal{N}_m \cap \mathcal{N}_{m-1} = \mathcal{N}_{m-1}$ are directly transferred and the values on the nodes $\mathcal{N}_m \setminus \mathcal{N}_{m-1}$ are determined by multi-linear interpolation from the values on the neighboring nodes in $\mathcal{N}_{m-1}$.

The prolongation from level $m-1$ to m is a linear mapping and can be represented by a matrix P. Before we can start the minimization on the coarsest desired grid level, we have to restrict the input data, which is given on the finest grid level, to the coarsest one. For this we use the restriction from level m to level $m-1$ given by the matrix P^T after normalizing the rows of P^T to have a row sum of one and successively apply the restriction to get the input data on all necessary levels.

The last thing we need to take into account for the cascadic procedure is the dependency of the parameters on the mesh level. Fortunately, all but one of the parameters are independent of the level and therefore do not need to be adapted throughout the cascadic algorithm. The only exception is ϵ, the "width" of the diffusive edge sets, as it is naturally linked to the mesh resolution h. To properly resolve the smooth transition from "edge" to "no edge" in the phase fields, ϵ needs to be chosen of the order of h, i. e. $\epsilon = ch$ for $c > 0$ arbitrary but fixed. In other words, on level m the algorithm automatically sets ϵ to ch_m, where h_m denotes the mesh resolution of the mesh corresponding to level m.

Combined with the EM procedure, this leads to Algorithm 2.1, the complete registration algorithm. Numerically, the energy E_{SYM} and its variations as well as the matrices are approximated using a Gauss quadrature scheme of order 3 (cf. Section 6.1).

Algorithm 2.1: Complete registration procedure

given starting level m_0 and ending level m_1;
given images $G_R = G_R^{m_1}$ and $G_T = G_T^{m_1}$;
given number of iterations on each level k_1;
for $m = m_1 - 1$ **to** m_0 **do**
  initialize $[G_R^m, G_T^m]$ by restricting $[G_R^{m+1}, G_T^{m+1}]$;
end
initialize $[U_R^{m_0}, V_R^{m_0}, U_T^{m_0}, V_T^{m_0}]$ with 0;
initialize $[\Phi^{m_0}, \Psi^{m_0}]$ with id;
for $m = m_0$ **to** m_1 **do**
  for $k = 1$ **to** k_1 **do**
    update U_R^m by solving (2.10);
    update V_R^m by solving (2.11);
    update U_T^m by solving (2.12);
    update V_T^m by solving (2.13);
    update $[\Phi^m, \Psi^m]$ with 5 gradient flow steps using (2.16);
  end
  if $m \neq m_1$ **then**
    initialize $[U_R^{m+1}, V_R^{m+1}, U_T^{m+1}, V_T^{m+1}, \Phi^{m+1}, \Psi^{m+1}]$
    by prolongating $[U_R^m, V_R^m, U_T^m, V_T^m, \Phi^m, \Psi^m]$;
  end
end

2.3 Numerical results

To investigate the performance of the proposed one-to-one edge matching algorithm, we performed various numerical experiments. We start with a study on the influence of the various parameters involved in the model using T1- and T2-MRI volumes of the same patient as input data. Afterwards, we use the method for 3D intersubject monomodal registrations, useful to establish anatomical correspondences between different individuals. We continue with the registration of retinal images and then apply the method to match 2D photographs arising in neurosurgery to the corresponding projections of 3D MRI volume data. Finally, we show that the method is not limited to medical data and can also go beyond the scope of classical image matching by using it for video frame interpolation.

To be able to apply the cascadic approach to any kind of given input data, we need to make sure that the data complies with the restrictions introduced by the mesh hierarchy, cf. Section 2.2.3. A simple resampling of the input data to the finest mesh of the hierarchy is sufficient for this, potentially preceded by cropping in case the input data does not have the same amount of pixels or voxels in each coordinate direction. For our experiments, the cropping was only necessary in the frame interpolation example. The resampling is done by multi-linear interpolation, i. e. bilinear for 2D data and trilinear for 3D data. This kind of interpolation is the canonical approach implied by our Finite Element framework (cf. Section 6.1) and gives sufficient accuracy. Note that the method does neither depend on the concrete choice of the resampling approach nor on the concrete construction of the cascadic framework.

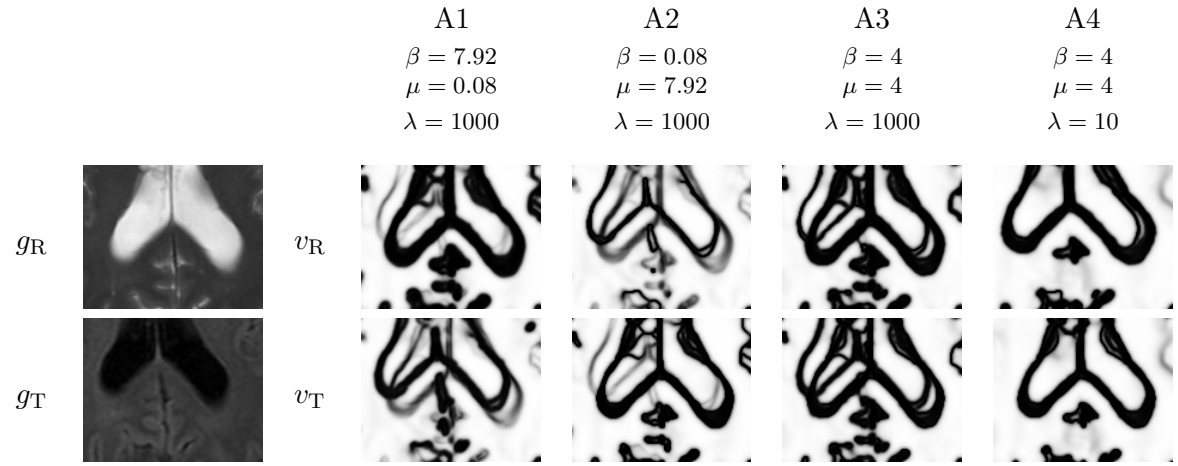


Figure 2.5: Influence of the parameters β, μ and λ on the phase fields.

2.3.1 Parameter study for 3D data

The basis of our parameter study is a pair of 3D MRI data sets, both acquired from the same person and with the same MRI scanner but with different scan parameters (T1/T2). Due to this acquisition setup, the original T1-MRI and T2-MRI volumes are already almost perfectly matched to each other. In order to show the effect of the registration process, the T2-MRI volume is deformed by an artificial elastic deformation. The original T1 volume is used as reference image g_R, the artificially deformed T2 volume is used as template image g_T. Both input volumes are of size $512 \times 512 \times 101$ and were resampled (keeping the aspect ratio and using extension by zero) to 129^3 voxels to apply the cascadic scheme. Each experiment used 10 EM-iterations on the 129^3 mesh. The runtime per experiment was about two hours on a standard PC with Intel Pentium 4 2.26 GHz processor and 2.0 GB RAM and using a non-optimized implementation. The necessary computational time is expected to decrease significantly if the employed general purpose implementation is optimized for the particular model. Even though the parameter influence is only studied for T1-/T2-MRI edge matching here, the effects of the parameter ratios seen in these experiments are also valid for the edge matching of other image modalities.

Figure 2.5 illustrates how the parameters β, μ and λ balance edge detection against edge matching. The parameters not mentioned in the figure are fixed at $\alpha = 2550$, $\nu = 0.1$, $\kappa = 100$, $\epsilon = 0.5h$. The images shown are details of a single slice of the volume data. Since we only want to investigate the influence of the parameters, it is sufficient to confine to these details. In experiments A1-A3, the overwhelmingly large regularization parameter λ essentially disables the matching part of the approach and hence allows only edge detections. These three experiments show that the ratio between β and μ controls whether a phase field represents the edges of its "own" image, the edges of its counterpart or a mixture of both. In experiment A1, where β is much larger than μ, E_{ED} has more influence than E_{MA}. Thus the resulting phase fields mainly resemble their "own" edges. The opposite case is depicted in experiment A2. Here, β is much smaller than μ and hence the phase fields prefer to resemble the edges of their counterparts, i. e. v_R shows the edges of g_T while v_T shows the edges of g_R.

Proper edge detection is a necessity to achieve good edge-based matching. Therefore, the parameters β and μ need to be set in such a way that the resulting phase fields v_T and v_R clearly

resemble the edges of both images, as shown in experiment A3. In the case of the T1-/T2-MRI data considered here, experiment A3 shows that it is reasonable to use equal values for β and μ. However, this ratio needs to be customized to the specific application considered. For example if the intensity patterns of the input images are vastly different, like in Section 2.3.4, the parameters β and μ have to be chosen differently. Finally, experiment A4 enables edge matching in addition to edge detection due to the smaller value of the regularization parameter λ. Compared to experiment A3, each phase field resembles the edges of both images, but the edge matching merges the edges in the phase fields.

We performed further experiments to study the effects of the parameters λ and κ, cf. [73]. Unsurprisingly, a certain level of regularization induced by λ is necessary and sufficient to obtain topology-preserving deformations in practice. Furthermore, κ may not be chosen too small in order to attain adequate consistency between ϕ and ψ. In conclusion, it seems to be safe to roughly fix five of the parameters in most 2D and 3D applications, i. e. choosing $\lambda = 10, \kappa = 100, \alpha = 2550, \nu = 0.1 \sim 1, \epsilon = 0.5h$ usually achieves good matching results.

2.3.2 Intersubject monomodal registration

The following two experiments show the performance of the proposed method for intersubject monomodal registration: We apply our method to a pair of MRI data sets (MRI-to-MRI) and a pair of CT data sets (CT-to-CT). Both of the MRI data sets picture a healthy brain, each of a different subject. The CT data sets show the heads of two other subjects, one healthy and the other one abnormal. All data sets were acquired by the same MRI and CT scanners respectively and with the same scanning parameters. Furthermore, the MRI data sets were preprocessed using MRIcro (`http://www.sph.sc.edu/comd/rorden/mricro.html`) to extract the brain from the scan of the head.

To apply the cascadic scheme, all input data sets were resampled (keeping the aspect ratio and using extension by zero) to 257^3 voxels from originally $512 \times 512 \times 58$ and $512 \times 512 \times 61$ (CT) and $256 \times 256 \times 160$ and $256 \times 256 \times 170$ (MRI) respectively. Both experiments used 10 EM-iterations each on the levels 33^3, 65^3, 129^3 and 257^3 with the following parameter settings: $\alpha = 2550$, $\beta = 1$, $\nu = 0.1$, $\mu = 1$, $\lambda = 10$, $\kappa = 100$, $\epsilon = 0.5h$. Respective runtime (on the same machine specified in the parameter study) for each level was approximately 1 minute, 10 minutes, 90 minutes and 5 hours.

The matching results are visualized by an "interlace-stripe" pattern, showing two data sets in an alternating manner within a single volume. Such a stipe view of two data sets A and B is denoted by $A\|B$. Therefore, recalling the notation from the previous sections, $G_\mathrm{R}\|G_\mathrm{T}$ and $G_\mathrm{T}\|G_\mathrm{R}$ show the interlace-stripe volumes of the original data sets G_R and G_T, while $G_\mathrm{R}\|G_\mathrm{T} \circ \Phi$ and $G_\mathrm{T}\|G_\mathrm{R} \circ \Psi$ show the interlace-stripe volumes of a matched data pair, i. e. one input data set with the corresponding deformed data set.

From visual inspection, the proposed algorithm successfully registers MRI-to-MRI and CT-to-CT volume data sets of different subjects in both directions. Figure 2.6 shows that the prominent edges such as the brain's outer contour, the hemispheric gap and the ventricular system are precisely aligned in the MRI-to-MRI matching experiment. The same holds true for the CT-to-CT experiment: The prominent edges such as the bones and the outer contour of the head are properly aligned, cf. Figures 2.8 and 2.10. The edges that were taken into account for the registration by the algorithm are depicted by Figures 2.7 and 2.9.

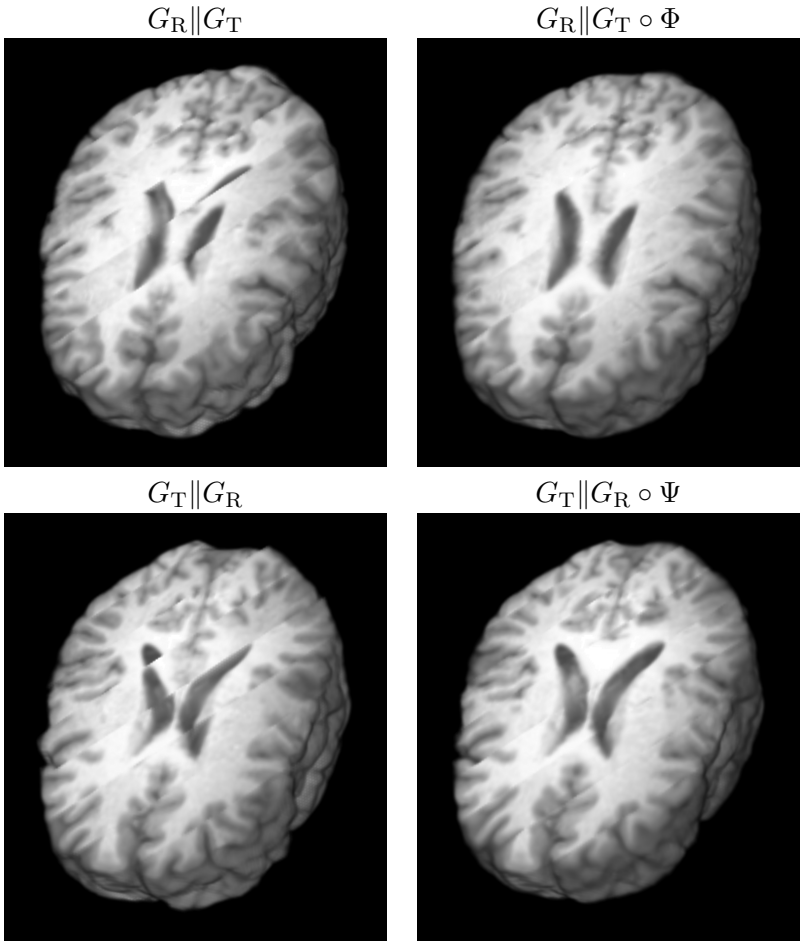


Figure 2.6: Intersubject MRI-to-MRI registration using one-to-one edge matching: Volume rendering of the 3D data sets with a cut plane to illustrate the quality of the registration by the alignment of important structures, here, outer brain contour, hemispheric gap and ventricular system.

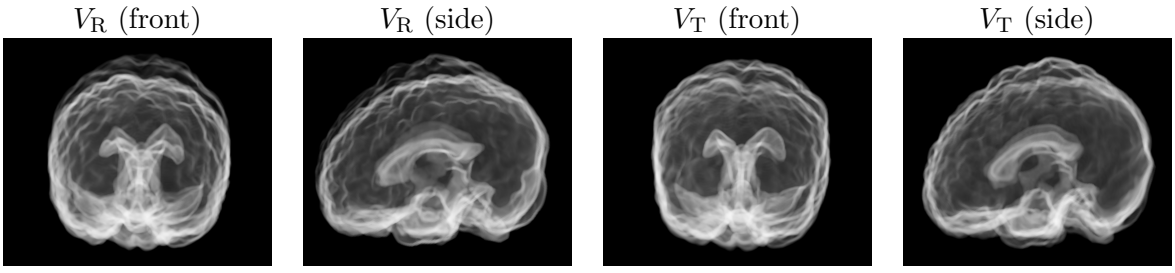


Figure 2.7: Volume rendering of the final phase fields from the intersubject MRI-to-MRI registration (cf. Figure 2.6) in a front and a side view. The window function was chosen to display regions were the phase fields are zero, e. g. detected edges, as transparent white.

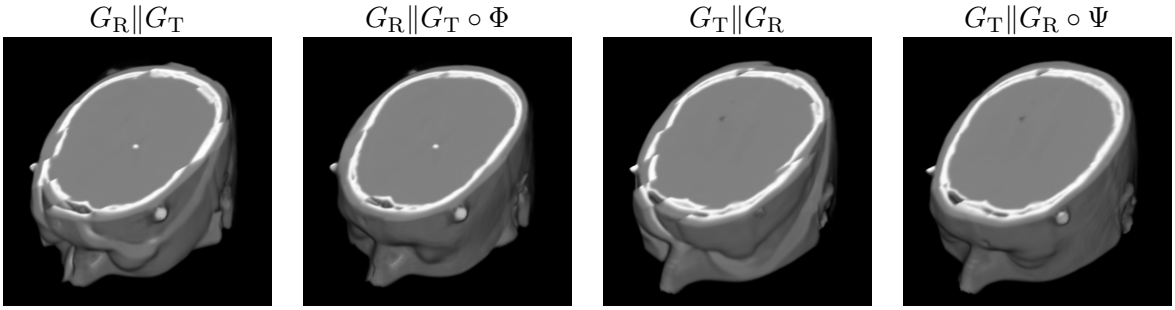


Figure 2.8: Intersubject CT-to-CT registration using one-to-one edge matching: Volume rendering of the 3D data sets with a cut plane to illustrate the quality of the registration by the alignment of important structures, here, outer head contour and bones.

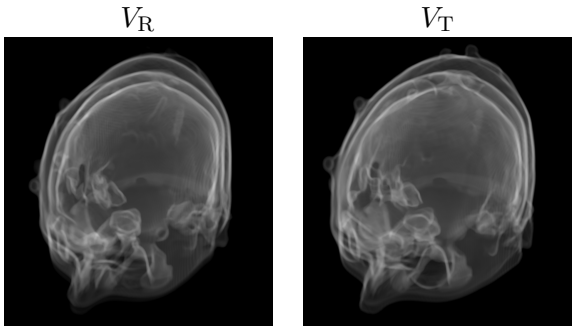


Figure 2.9: Volume rendering of the final phase fields from the intersubject CT-to-CT registration (cf. Figure 2.8). The window function was chosen to display regions were the phase fields are zero, e. g. detected edges, as transparent white.

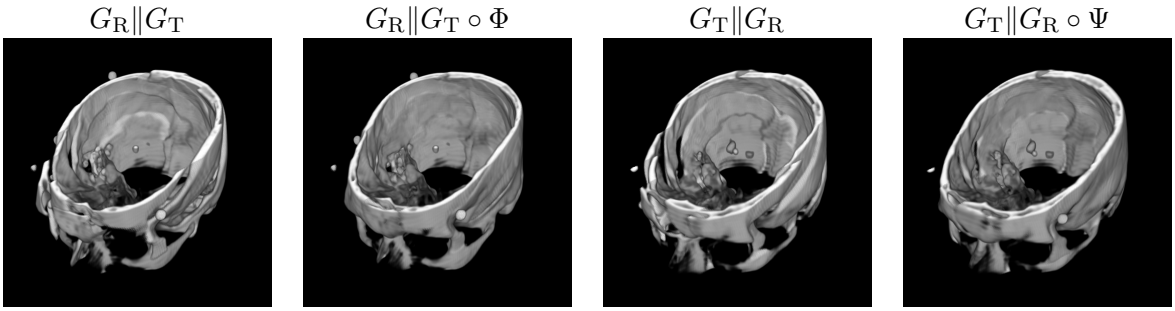


Figure 2.10: Alignment of the skulls in intersubject CT-to-CT registration: Volume rendering of the 3D data sets with a cut plane and a bone window function.

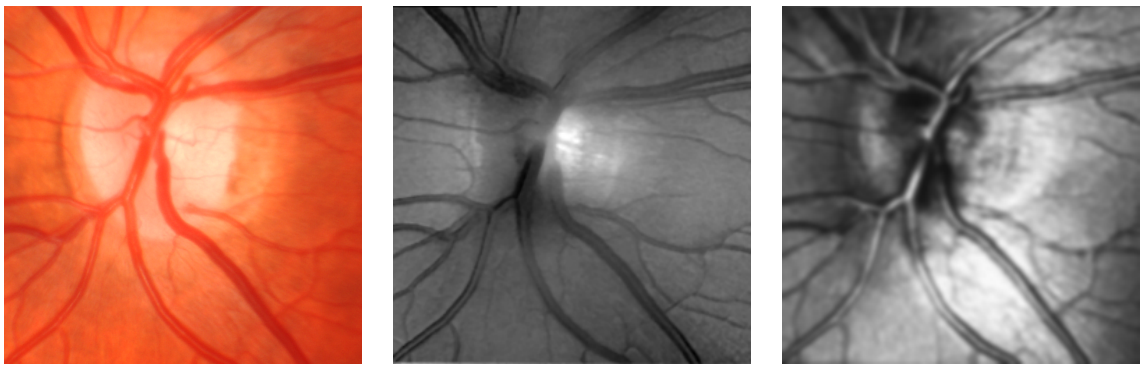


Figure 2.11: Multimodal retinal images of one subject: photograph (left), depth image (middle) and reflectance image (right). All images are courtesy of Rüdiger Bock (Chair of Pattern Recognition, Erlangen-Nürnberg University).

2.3.3 Retinal images

The reliability of glaucoma diagnosis can be increased significantly by a concurrent representation of the optic nerve header and the neuroretinal rim in various retinal image modalities. Several modalities of retinal images are used in this context: reflection-free photographs with electronic flash illumination, and depth as well as reflectance retinal images acquired by a scanning-laser-tomograph. Figure 2.11 depicts one retinal image of each kind, all acquired from the same patient. Due to the acquisition setup of a scanning-laser-tomograph, the depth and reflectance images are normally already almost perfectly matched to each other right after acquisition. Thus the task is the matching of reflectance and depth images with corresponding photographs, i. e. a multimodal registration problem. We do not consider monomodal retinal image registration here and refer to [35, 36] for this. Recently, an affine transformation model combined with an extended mutual information similarity [88] were applied to the multimodal registration problem we just introduced. However, as shown in the first column of Figure 2.12, this method cannot recover the minor deviations in the vessels and neuroretinal rims due to the limitations imposed by the employed affine transformation model. We employ the proposed method in this experiment as a post-registration step to compensate for the small, nonlinear deviations that are not captured by [88].

The original images are preprocessed as follows: The green channel of the photograph is extracted and used as input for the affine pre-registration of the photograph to the reflectance and depth images using the automatic tool described in [88]. The necessary software for the pre-registration was kindly provided by Rüdiger Bock (Chair of Pattern Recognition, Erlangen-Nürnberg University). After the pre-registration step, all pre-registered images are resampled to 257^2 to comply with the cascadic strategy. The algorithm was run using 10 EM-iterations each on the levels 65^2, 129^2 and 257^2 with the following parameter settings: $\alpha = 2550$, $\beta = 1$, $\nu = 0.1$, $\mu = 0.5$, $\lambda = 10$, $\kappa = 100$, $\epsilon = 0.5h$. The full runtime was less than three minutes. Figure 2.12 depicts the results of this experiment and shows that most of the minor deviations in the vessels are found and corrected. Note that the affine pre-registration used here to compensate for the large initial mismatch also helped our method to avoid getting stuck in a local minimum.

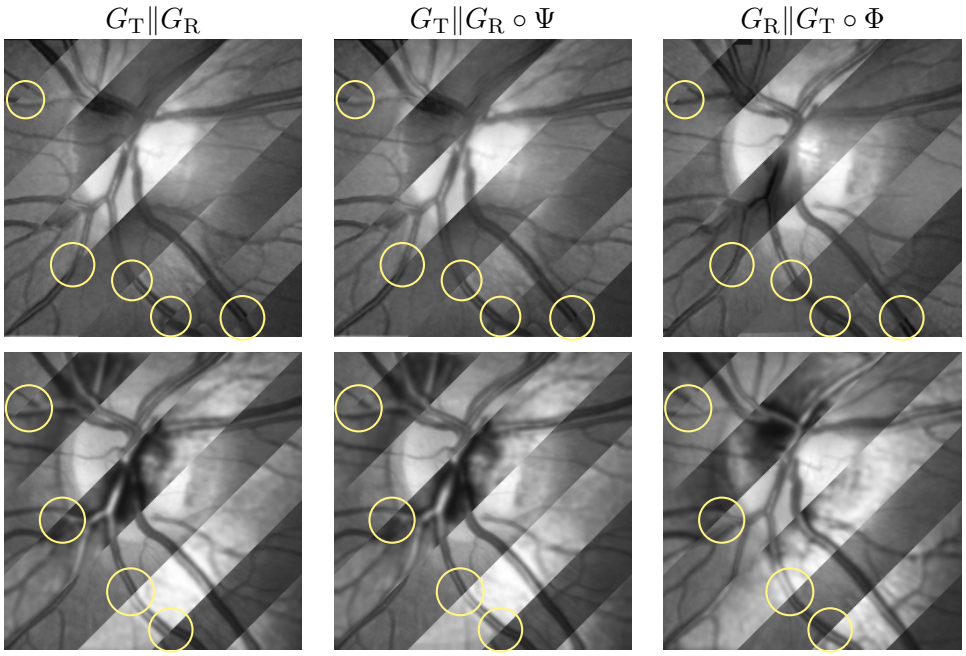


Figure 2.12: Post-registration of multimodal retinal images using one-to-one edge matching: The photograph is registered with the depth image (top) and the reflectance image (bottom). The result of the affine pre-registration [88] (first column) is used as input for our algorithm. The circles highlight the regions where the initial nonlinear deviations not captured by [88] are most noticeable.

2.3.4 Photographs of neurosurgery

In neocortical epilepsy surgery, the lesion causing the epileptic seizures of a patient may be located next to, or even partly within, functionally very relevant, so-called *eloquent*, cortical brain regions. For instance, the motor cortex (involved in the control of voluntary motor functions) is such an eloquent region. A safe neurosurgical planning requires the appearance of the exposed cortex to be mapped to the underlying functionality by the physician. The treatment usually requires two surgeries: The first surgery enables an electrophysiological examination of the underlying brain functionalities by placing an electrode array on the surface of the brain and taking photographs of the exposed cortex with and without the electrode array. After the first surgery, the cortex regions on the photographs are manually colored according to their functionality that is revealed by the electrode contacts. Additionally a 3D MRI scan, also containing information about the underlying lesion and healthy tissue, is acquired before the first surgery. In the second surgery, the pre-operative 3D MRI scan and the map of the brain functionalities on the photograph are used together to facilitate the surgical removal of the lesion without harming functionally relevant areas. At the moment, the fusion of the information from the 3D MRI scan with the photographs requires a manual affine registration and region of interest selection by a neocortical expert. In particular, the manual registration requires the expert to find the appropriate 2D projection of the 3D MRI data set to the cortex region shown on the photographs. However, the photograph and the MRI projection cannot be accurately aligned by just an affine transformation due to the so-called *brain shift* that is affecting the

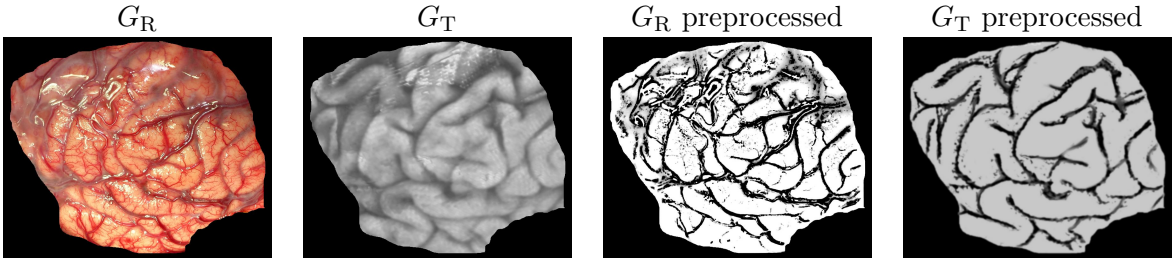


Figure 2.13: Matching a neurosurgery photograph with its MRI projection using one-to-one edge matching: Original images and the corresponding preprocessed versions that are fed to our algorithm.

photograph but not the MRI scan. Brain shift is the tissue deformation that occurs during neurosurgical procedures, for instance, due to loss of cerebrospinal fluid and cortical swelling. In this experiment, we make use of the proposed method to refine the manual registration between a 2D digital photograph of the cortex and the projection of the corresponding pre-operative 3D MRI scan. Regarding the details of the equipment used to acquire the photographs and the MRI scan, we refer to [73].

Figure 2.13 shows the input images used in this experiment, obtained using manual affine registration by a physician. G_R denotes the digital photograph and shows the exposed left hemisphere from an intraoperative viewpoint. In particular, the gyri and sulci as well as the overlying vessels are clearly visible. G_T denotes the 2D projection of the MRI data set and displays the left-sided view of the rendered MRI volume in the corresponding parts. Comparing G_R and G_T, one notices that the photograph clearly displays surface vessels and a reflectance flash, whereas these features are not present in the MRI projection image. Thus the registration should base the matching on the gyri and sulci, present in both images. In order to ease the corresponding edge detection that needs to be done by our algorithm, both images were preprocessed by appropriate GIMP filter chains for edge enhancement, see "G_R preprocessed" and "G_T preprocessed" respectively in Figure 2.13. In particular, the filter chain for G_R involves successive erosion and dilation to remove very small vessels from the photograph.

To apply the cascadic scheme, both input images were resampled (keeping the aspect ratio and using extension by zero) to 2049^2 pixels. The algorithm was run using 10 EM-iterations each on all levels from 9^2 to 2049^2 with the following parameter settings: $\alpha = 2550$, $\beta = 100$, $\nu = 0.1$, $\mu = 0.1$, $\lambda = 10$, $\kappa = 100$, $\epsilon = 0.5h$. Note that the values of the parameters β and μ differ considerably from the values used in the other examples. This stems from the largely different image modalities of the photograph and the corresponding MRI projection. β and μ are chosen such that both phase fields V_T and V_R clearly represent the edge features of the cortex. Thus each phase field has similar influence on the registration. Figure 2.14 shows the results of this experiment by depicting interlace-stripe views of the unregistered and registered images. The registered views show that the proposed method considerably refines the initial manual affine matching done by the physician. Particularly the gyri and sulci are accurately aligned to each other by our method.

To compare the quality of our method to a well established method, we have implemented a mutual information based registration algorithm within the same Finite Element framework (including the step sized controlled, regularized, cascadic descent). Note that Viola and Wells [127] pioneered the use of mutual information in the context of image registration. Overall,

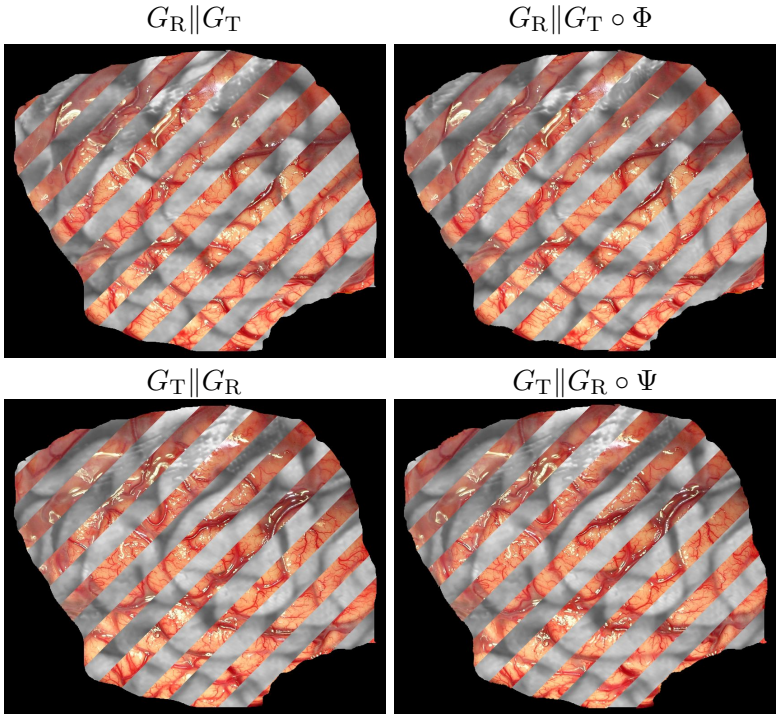


Figure 2.14: Matching a neurosurgery photograph with its MRI projection using one-to-one edge matching: Interlace-stripe views before and after the registration to illustrate the quality of the registration by the alignment of important structures, here, gyri and sulci.

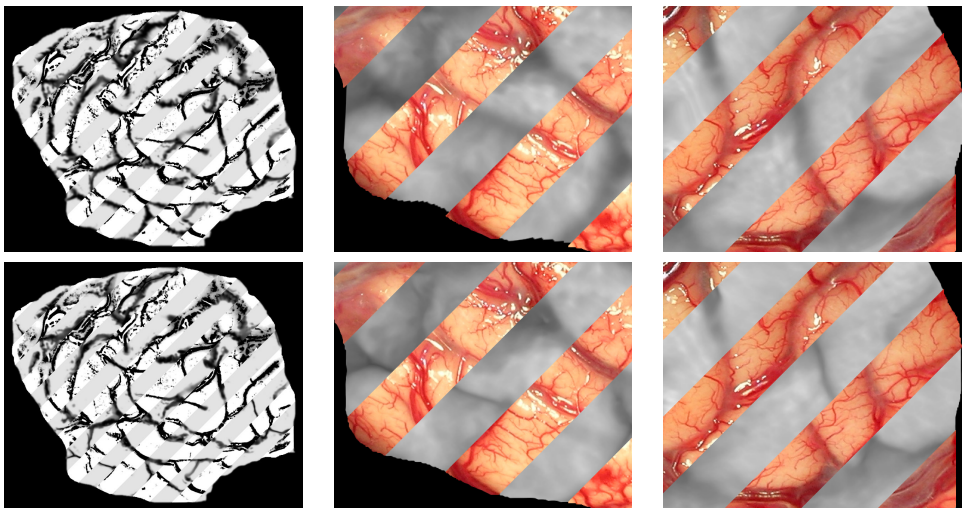


Figure 2.15: Comparison of one-to-one edge matching (top row) and mutual information based matching (bottom row) in the same Finite Element framework.

the results obtained by the proposed method are mostly comparable to mutual information, especially when dealing with coarse structures. However, as shown by the zoom views depicted in Figure 2.15, the alignment achieved by the edge-matching method is better than the one obtained by the mutual information based registration in case the input data sets contain a large number of lower dimensional structures.

2.3.5 Motion estimation based video frame interpolation

Temporal interpolation of video frames in order to either increase the frame rate or to restore corrupted frames requires the estimation of a motion field. This motion field can be seen as a transformation and encodes how a point moves from one frame to another. Given a motion field between two frames, intermediate frames can be obtained by interpolation of the pixels along the path given by the motion field. For a review of frame interpolation techniques, we refer to [87, 82]. Here, we confine to giving a proof of concept showing that our proposed matching method can be used for this kind of application. Later, in Chapter 4, we will return to the problem of motion estimation and combine it with the restoration of motion-blurred objects from video sequences.

In this experiment, the goal is to interpolate frame 58 of the "Susie sequence" from frames 57 and 59, cf. Figure 2.16. To apply the cascadic scheme, the original frames are cropped to 257^2 pixels. We denote the cropped frames 57, 58 and 59 by F_{57}, F_{58} and F_{59} respectively. F_{57} is used as template image G_{T}, F_{59} is used as reference image G_{R}. Hence Φ denotes the transformation from F_{57} to F_{59}, Ψ denotes the one from F_{59} to F_{57}. The algorithm was run using 10 EM-iterations with the following parameter settings: $\alpha = 2550$, $\beta = 1$, $\nu = 0.1$, $\mu = 1$,

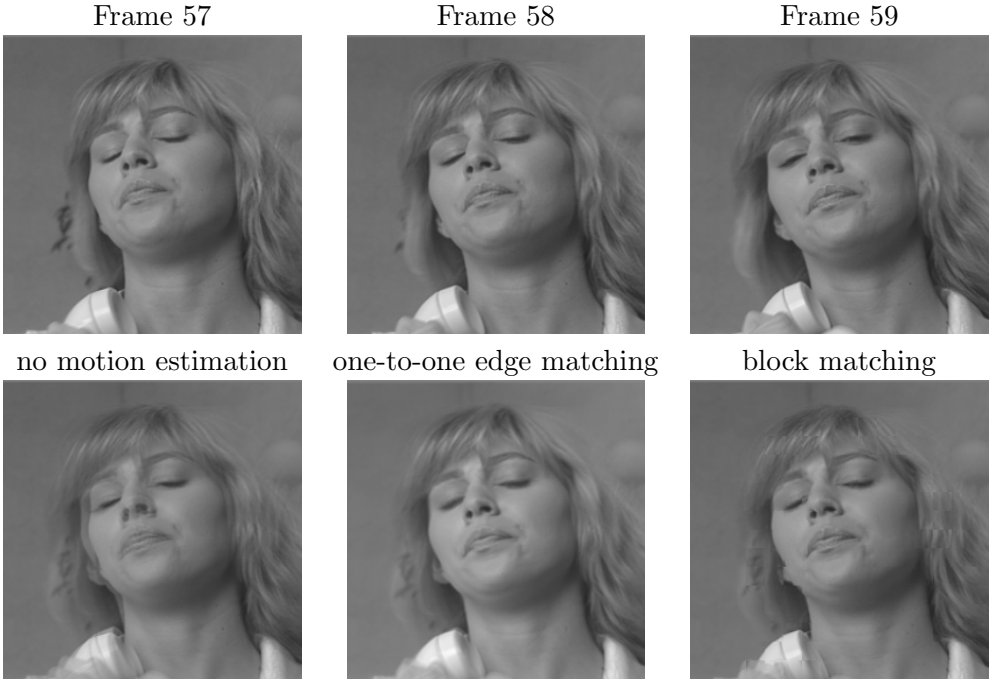


Figure 2.16: Video frame interpolation of the "Susie" sequence using one-to-one edge matching compared to simple averaging and a standard block matching algorithm.

$\lambda = 10$, $\kappa = 100$, $\epsilon = 0.5h$. After the registration of F_{57} and F_{59}, F_{58} is interpolated using the average of the frame predictions indicated by Φ and Ψ, i. e.

$$F_{58}(x) = \frac{1}{2}\left(F_{57}\left(\frac{1}{2}(\Phi(x) - x) + x\right) + F_{59}\left(\frac{1}{2}(\Psi(x) - x) + x\right)\right).$$

Note that $\Phi(x) - x$ is the displacement of the transformation Φ, hence $\frac{1}{2}(\Phi(x) - x) + x$ halves the displacement done by Φ, the same holds for Ψ. Figure 2.16 compares the interpolation obtained by our algorithm with simple averaging (no frame interpolation) and a standard block matching algorithm, so-called *adaptive rood pattern search* [101], using blocks of size 16×16 and a horizontal and vertical search range of $[-16, 16]$. In the results shown in Figure 2.16, the adaptive rood pattern search leads to unpleasant block artifacts, whereas the interpolated frame obtained by the one-to-one edge matching approach is visually convincing and without apparent artifacts.

3 Grain boundaries and macroscopic deformations on atomic scale

THE goal of this chapter is to present a method to extract certain mesoscopic quantities from microscopic image data in materials science. This task is of practical relevance because the image data acquired in this context often lives on the microscale (e. g. the underlying atomic structure), whereas the material properties (electrical, optical, mechanical, etc.) one is looking for are determined by mesoscopic properties.

In particular, the actual material properties are usually determined on a mesoscopic length scale. There, non-equilibrium structures exist which form and evolve during material processing. For example, local variations of the inter-atomic distance can be understood as material deformation on the mesoscale. Another example is the yield strength of a polycrystal that varies with the inverse square of the average grain size. Here, *grains* are material regions with different lattice orientation which are typically not in equilibrium. In additional they are frequently observed in an elastically deformed state. Experimental tools such as TEM (transmission electron microscopy) [84] nowadays allow measurements down to an atomic resolution (cf. Figure 3.1). A reliable extraction of elastic deformations and grains (characterized by their boundaries) from these TEM images is essential for an efficient material characterization. Likewise, recent numerical simulation tools have been developed for physical models of grain

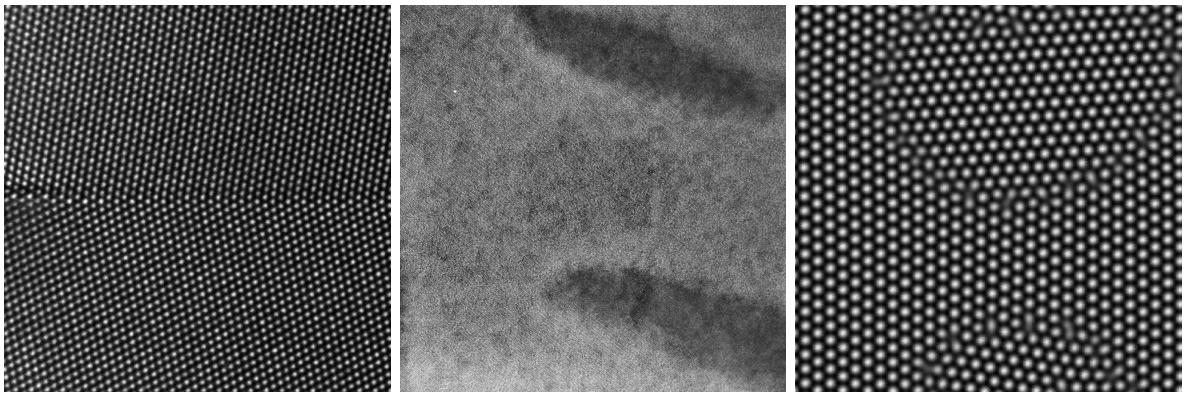


Figure 3.1: In a TEM image (left), light dots render atoms from a single atom layer of aluminum (exemplary shown by a $\Sigma 11(113)/[\bar{1}00]$ grain boundary [84], courtesy of Geoffrey H. Campbell, Lawrence Livermore National Laboratory). The second TEM image (middle) illustrates that on a mesoscale elastic displacements are implicitly encoded in a spatially varying inter atom distance (courtesy of Nick Schryvers, Antwerp University). The last image (right) is a time step from a numerical PFC simulation showing a similar atomic layer as the first image. In both images (left and right), one can observe that grain boundaries are characterized by jumps in the lattice orientation.

formation and grain dynamics on the atomic scale. Concerning such simulations, we refer to numerical results obtained from a phase field crystal (PFC) model [64] derived from the density function theory of freezing [118]. Its methodology describes the evolution of the atomic density of a system according to dissipative dynamics driven by free energy minimization. The resulting highly nonlinear partial differential equation of sixth order can be solved applying a finite element discretization and a semi-implicit time discretization [14]. As an example, Figure 3.2 shows several time steps of a PFC simulation, depicting the simulated nucleation and growth of grains. Such simulations in particular will allow a validation of the physical models based on the comparison of mesoscopic properties such as the propagation speed of grain boundaries. Note that the formation of grains from an undercooled melt happens on a much faster time scale than their subsequent growing and coalescence, whereas the evolution of grain boundaries at later stages of the process is of particular interest. Figure 3.1 shows two experimental (TEM) and one numerically simulated (PFC) image at atomic scale side by side and outlines the similar structure of both image types.

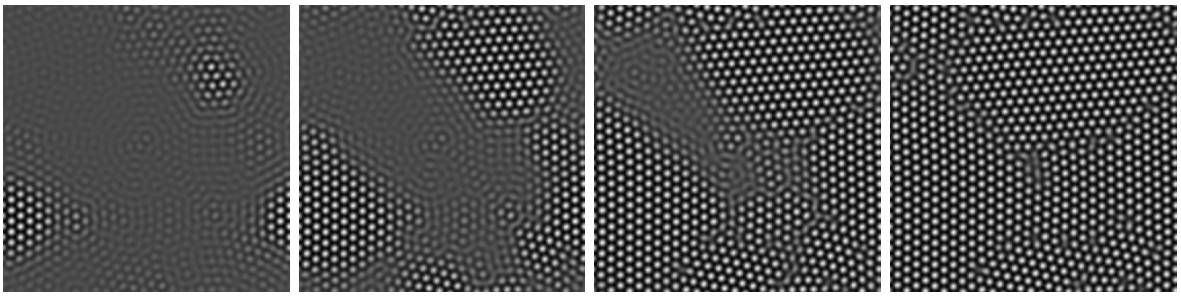


Figure 3.2: Nucleation of grains in a phase field crystal simulation.

As mesoscopic material properties can be deduced from observations of microstructures, their robust and reliable extraction via image processing methodology is expected to provide physical insight in the underlying materials. Here, we are treating the reliable extraction of grain boundaries [22] and elastic grain lattice deformations [23], as well as the detection of the interfaces between the liquid and the solid phase of a material. For this, we apply a variational approach based on the description of the interfaces by level sets. Furthermore, we generalize the variational approach for the extraction of an elastic deformation and a full coupling of orientation and deformation classification. We demonstrate the applicability of our approach by experiments on phase field crystal simulation results and and also on experimental TEM images.

The chapter is organized as follows: We begin with a review of related work in Section 3.1. Then, in Section 3.2, we discuss the case of how to extract the orientation and the elastic deformation from a single grain. Afterwards, the segmentation of grain boundaries in the non-deformed case is discussed in Section 3.3. In both sections, we first introduce a variational problem involving sharp interfaces on the microscale (describing atomic dot patterns) and on the macroscale (representing grain boundaries), then discuss a suitable smooth approximation, and derive a minimization algorithm based on a regularized gradient descent (cf. Section 6.2). The particular case of a liquid phase beside the solid phase is treated in Section 3.4. Combining the different approaches in Section 3.5, we derive a joint approach for the simultaneous extraction of grain domains classified by local lattice orientations and the computation of an underlying elastic deformation. Finally, we give a short outlook in Section 3.6.

3.1 Related work

To the best of our knowledge, our method [22, 23] differs significantly from other variational approaches in the literature. This is due to fact that our focus is not to develop a general purpose texture classification and segmentation tool but a model tailored to the specific application in materials science.

In general, texture segmentation can be regarded as a two-scale problem, where the microscale is represented by the local structure of the texture and the macroscale by the geometric structure of interfaces between regions with different texture. In the materials science problem we are focussing on, we have strong a priori knowledge on the local structure of the texture on the microscale and incorporate this directly into the variational approach on the macroscale. Thus, at the cost of general applicability, our scale separation is more direct than in other approaches based on a local, direction-sensitive frequency analysis. Currently, the postprocessing for experimental images and the pattern analysis used in the literature is mostly based on local, discrete Fourier filtering [114].

As one of the basic problems in image processing, general image classification, i. e. assigning a label to each point in the image domain, has been studied extensively in the last decades. Classification can be based on geometric and on texture information. Many models have been developed either based on region growing [124, 106, 38], on statistical approaches [24, 26, 89, 93] or in particular recently on variational approaches [12, 11, 53, 97, 126].

Bridging the gap between texture classification and the Mumford–Shah approach [100] to image segmentation and denoising, the boundaries of the classified regions can be considered as free discontinuity sets of classification parameters. As already discussed in Section 1.2.2, a robust and efficient approximation of the Mumford–Shah functional has been presented by Chan and Vese [46] for piecewise constant image segmentation in two phases and extended to multiphase segmentation in [125]. There, the decomposition of the image domain is implicitly described by a single or by multiple level set functions (for a review on level sets we refer to [104, 115]). In [111], their approach has been further generalized for the texture segmentation using a directional sensitive frequency analysis based on Gabor filtering. Texture classification based on the energy represented by selected wavelet coefficients is investigated in [10]. Inspired by the work of Meyer [97] on cartoon and texture decomposition, the classification of geometric and texture information has been investigated further in [12]. There, a logic classification framework from [111] has been considered to combine texture classification and geometry segmentation. A combination of level set segmentation and filter response statistics has been considered for texture segmentation in [76]. For a variational texture segmentation approach in image sequences based on level sets, we refer to [57].

3.2 Macroscopic elastic deformations from deformed lattices

First, let us introduce some basic notation. As input data, we always consider a real-valued image of a single atom layer resolved at atomic scale. In our applications, this image either is an intensity map acquired experimentally by transmission electron microscopy or originates from a phase field crystal simulation (in this case the image is the PFC function). For both sources, single atoms are represented by blurry, dot-like structures in this image. Let us denote this intensity or phase field crystal function by $u : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$ is the image domain or the computational domain of the PFC simulation, respectively. In this work, we always assume

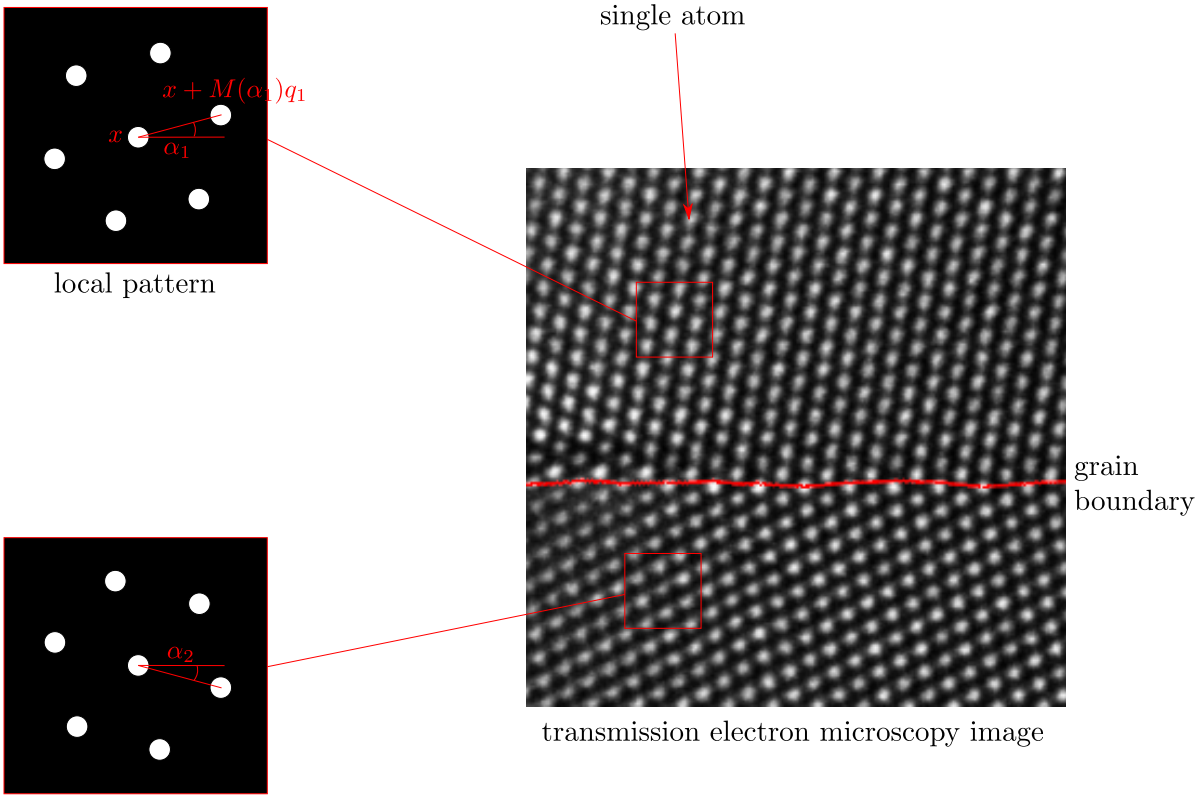


Figure 3.3: Schematic view of a typical grain boundary in a TEM image alongside the rotated reference cells corresponding to the two grains in the TEM image.

Ω to be a bounded domain (i. e. an open and connected set) with Lipschitz boundary. From a practical point of view, this is no restriction as it is always fulfilled by the input data. We further introduce the local atomic lattice orientation as a piecewise constant function $\alpha : \Omega \to \mathbb{R}$. Grains are characterized by a homogeneous lattice orientation, thus each region of constant lattice orientation characterizes a grain and α corresponds to a decomposition of the domain Ω into grains. Conversely, the grain boundaries form the jump set of the orientation function α. Additionally, we assume a global deformation $\psi : \Omega \to \mathbb{R}^2$ acting on all grains and reflecting the physical response for instance due to an external loading.

At first, let us suppose that there is no liquid phase and focus on a single grain.

3.2.1 Local identification of lattice parameters

As already stated, grains are characterized by a homogeneous lattice orientation, but before we can employ this fact, we need a precise characterization of the underlying atomic lattice and the lattice orientation in particular. For our approach, we assume that the underlying atomic lattice is a so-called *Bravais lattice* [8], i. e. the lattice looks the same when viewed centered at any atom position. Thus the lattice is uniquely characterized by the local neighborhood of a single atom in this lattice. Consider a reference frame with an atom at the origin and let m denote the number of direct neighbors of an atom in the lattice. Then, there exist positions $q_1, \dots, q_m \in \mathbb{R}^2$ such that the m neighboring atoms of the atom at the origin are placed at these positions. Now consider a grain whose lattice is rotated by an angle α compared to the

reference configuration we just introduced. Then, if there is an atom at position $x \in \mathbb{R}^2$, the neighboring atoms are located at the positions $x + M(\alpha)q_i$, cf. Figure 3.3. Here, $M(\alpha)$ is a rotation by α, i. e.

$$M(\alpha) := \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}.$$

The fact that atoms are represented by blurry, dot-like structures in the input image u, can be formalized as follows: The higher the image intensity at any position, the likelier the presence of an atom at this position. Thus there exists a suitable threshold θ for the identification of atom dots and we consider the indicator function

$$\chi_{\{u>\theta\}}(x) = \begin{cases} 1 & u(x) > \theta \\ 0 & \text{else.} \end{cases} \tag{3.1}$$

The threshold θ has to be chosen in accordance with the input image and our algorithm will require the user to supply the threshold along with the input image. Without loss of generality, we can assume $\theta > 0$ (in case it is not, replace u by $u - \theta + 1$ and θ by 1). Hence, for a given lattice orientation α and a point x with $\chi_{\{u>\theta\}}(x) = 1$, we assume an atom to be present at position x and thus expect atoms to be located at $x + M(\alpha)q_i$ for $i = 1, \dots, m$ as well, which in turn translates to $\chi_{\{u>\theta\}}(x + M(\alpha)q_i) = 1$ for $i = 1, \dots, m$. Let r denote the average radius of a single atom dot and define the maximal lattice spacing $d := \max_{i=1,\dots,m} |q_i|$. Based on these considerations, we construct the following indicator function f:

$$f[\alpha](x) = \frac{d^2}{r^2}\chi_{\{u>\theta\}}(x)\Upsilon\left(\left(\chi_{\{u>\theta\}}(x + M(\alpha)q_i)\right)_{i=1,\dots,m}\right) \tag{3.2}$$

Here, $\Upsilon : [0,1]^m \to \mathbb{R}$ is a function with a unique global minimum at $\overline{1} = (1, \dots, 1)$ and $\Upsilon(\overline{1}) = 0$. The scaling $\frac{d^2}{r^2}$ is used for normalization purposes and ensures a uniform upper bound of order 1 for the integral of f over the domain of the grain. In particular, the upper bound is independent of d and r.

Two natural choices for Υ are for instance

$$\Upsilon(\chi_1, \dots, \chi_m) := \frac{1}{m}\sum_{i=1}^{m}(1 - \chi_i), \quad \text{or} \tag{3.3}$$

$$\Upsilon(\chi_1, \dots, \chi_m) := 1 - \prod_{i=1}^{m}\chi_i. \tag{3.4}$$

In the course of our numerical experiments, (3.3) turned out to be better suited than (3.4), probably due to the fact that (3.3) is smoother than (3.4) in the following sense: Consider $\chi \in \{0,1\}^m \setminus \{\overline{1}\}$ and let n denote the number of components of χ that are equal to zero. For (3.4) we have $\Upsilon(\chi) = 0$, whereas for (3.3) it holds that $\Upsilon(\chi) = \frac{n}{m}$. In other words, as long as one atom is missing, (3.4) imposes a penalty that is independent of the number of missing atoms, while the penalty imposed by (3.3) is proportional to the number of missing atoms.

Finally, if the grain is deformed by an elastic deformation ψ and there is an atom at $\psi(x)$, we observe atoms at positions $\psi(x + M(\alpha)q_i)$. Analogously to (3.2), we can construct an indicator function f that takes the deformation ψ into account in addition to the lattice orientation α:

$$f[\alpha, \psi](x) = \frac{d^2}{r^2}\chi_{\{u>\theta\}}(\psi(x))\Upsilon\left(\left(\chi_{\{u>\theta\}}(\psi(x + M(\alpha)q_i))\right)_{i=1,\dots,m}\right) \tag{3.5}$$

If ψ is the identical deformation id, i. e. $\mathrm{id}(y) = y$ for all y, the two indicators agree, i. e. $f[\alpha](x) = f[\alpha, \mathrm{id}](x)$.

By construction, we have $f[\alpha, \psi](x) = 0$ if x is inside a grain with perfect lattice structure, orientation α, deformed by ψ and the distance of x to the grain boundary in the undeformed state is at least d.

While our approach is applicable to any Bravais lattice, for the sake of simplicity we confine to the case of a hexagonal packing. Henceforth, each atom has six direct neighbors at equal distances and we obtain

$$q_i := d\left(\cos\left(i\frac{\pi}{3}\right), \sin\left(i\frac{\pi}{3}\right)\right) \qquad i = 1, \dots, 6\,,$$

where $d > 0$ denotes the distance between two atoms.

3.2.2 Lattice deformation and lattice orientation on a single grain

At first, we assume our input image u to show a single grain with an unknown orientation α and affected by an unknown deformation ψ and our proclaimed task is to estimate α and ψ from the given image u. In line with the previous approaches in this work, we want to use a variational approach and thus phrase the estimation as a minimization problem on the class of deformations ψ and constant orientation angles α. The corresponding objective functional will consist of a fidelity term that measures how well a given pair of α and ψ fits to the actually observed configuration u and a regularizing prior that encodes assumptions on the deformation ψ. Unlike ψ, α does not depend on the location, but just a scalar constant. Thus, there is no need for any regularization of α.

Our indicator function f defined in (3.5) already lays the foundation of the fidelity term, i. e. this term is given by the integral over the indicator function and hence depends on both unknowns, α and ψ:

$$E_{\mathrm{fid}}[\alpha, \psi] = \int_\Omega f[\alpha, \psi](x)\,\mathrm{d}x \tag{3.6}$$

When closely examining the integrand f in combination with the integration domain Ω, one notices that the integral requires ψ to be evaluated outside of Ω (for x close enough to the boundary of Ω). Furthermore, due to the definition of $\chi_{\{u>\theta\}}$ (3.1), the integral requires the evaluation of u on the range of ψ. The latter can easily be overcome by extending u from Ω to $\mathbb{R}^d$ by zero, e. g. $u(x) = 0$ for $x \notin \Omega$. To treat the former, note that evaluating the integral for all α requires the evaluation of ψ on the set

$$\mathcal{D} := \mathcal{D}[\Omega] := \left\{x \in \mathbb{R}^2 \,:\, \mathrm{dist}(x, \Omega) < d\right\}.$$

Thus, we can remedy this by looking for ψ in the class of mappings from $\mathcal{D}$ to $\mathbb{R}^2$ instead of assuming ψ just to be defined on Ω. To account for this extension, the yet to be determined regularity term has to ensure the smoothness of the deformation ψ on $\mathcal{D}$. Because of the discontinuities of our integrand (originating from the concatenation of a characteristic function with the deformation ψ), classical results based on the direct method in the calculus of variations [54] to ensure the existence of minimizers cannot be applied to E_{fid}. One possibility to guarantee the existence of minimizing deformations is to use a suitable nonlinear elastic regularization energy for ψ, cf. [15, 16]. However, to simplify matters we confine here to

a quadratic regularization term on the Jacobian of ψ that results in a linearized elastic regularization in the Euler–Lagrange equations.

Close inspection of the model we introduced so far reveals that there are effectively two deformations involved. Obviously, there is the physical deformation ψ, but alongside of it, the orientation of the lattice also is a deformation (albeit restricted to a special class of deformations, namely rotations). Without putting any restrictions on ψ, it could contain a global rotational component conflicting with estimation of the lattice orientation given by α. Fortunately in our case the axiom of frame indifference applies, because we model a physically stressed material. Therefore, we are actually only interested in the non-rotational part of the physical deformation and can chose the class of deformations we consider for ψ accordingly. Thus, we assume ψ to have no angular momentum, i. e.

$$\int_{\mathcal{D}} \psi_2(x)x_1 - \psi_1(x)x_2 \,\mathrm{d}x = 0. \tag{3.7}$$

With this constraint we get a proper decoupling of a global rotation $M(\alpha)$ describing the lattice orientation and the elastic deformation ψ that (because of the constraint) has a vanishing global, linearized rotational component. Alternatively we have taken into account a constraint on the mean value of the skew symmetric part of $D\psi$,

$$\int_{\mathcal{D}} D\psi(x) - (D\psi)^T(x) \,\mathrm{d}x = 0. \tag{3.8}$$

This constraint rules out infinitesimal rotations and in particular turned out to be favorable for the numerical implementation.

Now that we have settled in which class of deformations we want to search for ψ, we have to chose a regularizing energy term suited for this class. Hence, taking into account a linearized deformation model we consider the symmetric part of the Jacobian of the displacement $\psi - \mathrm{id}$, i. e.

$$\frac{1}{2}\left(D(\psi - \mathrm{id}) + D(\psi - \mathrm{id})^T\right) = \frac{1}{2}\left(D\psi + D\psi^T - 2\,\mathbb{1}\right),$$

where $\mathbb{1}$ denotes the identity matrix, and define the elastic regularization term

$$E_{\mathrm{elast}}[\psi] = \frac{1}{2}\int_{\mathcal{D}} \left\|D\psi(x) + D\psi(x)^T - 2\,\mathbb{1}\right\|^2 \mathrm{d}x\,. \tag{3.9}$$

Here, $\|A\| := \sqrt{A : A}$ denotes the Frobenius norm, where $A : B = \mathrm{tr}(A^T B)$.

Note, using the displacement $\boldsymbol{u} = \psi - \mathrm{id}$ corresponding to the deformation ψ and the so-called *infinitesimal strain tensor* $\varepsilon(\boldsymbol{u}) = \frac{1}{2}\left(D\boldsymbol{u} + D\boldsymbol{u}^T\right)$, we have

$$E_{\mathrm{elast}}[\psi] = 2\int_{\mathcal{D}} \varepsilon(\boldsymbol{u}) : \varepsilon(\boldsymbol{u}) \,\mathrm{d}x\,.$$

Therefore, E_{elast} is a special case of the general elastic energy

$$\frac{1}{2}\int_{\mathcal{D}} C\varepsilon(\boldsymbol{u}) : \varepsilon(\boldsymbol{u}) \,\mathrm{d}x, \tag{3.10}$$

where C is a suitable fourth-order tensor. With the choice of C, the underlying properties of the material can be modeled. For the sake of simplicity, we confine to the isotropic elastic

regularization term E_{elast} here. This corresponds to an isotropic linearly elastic material with stiffness (Young's modulus) 4 and no bulging (Poisson's ratio 0), i. e. $C_{ijkl} = 2\,(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$.

Finally, combining the fidelity and the regularization term for the single grain case

$$E_{\text{single}}[\alpha, \psi] = E_{\text{fid}}[\alpha, \psi] + \mu E_{\text{elast}}[\psi]\,, \tag{3.11}$$

where $\mu > 0$ denotes a weighting parameter for the regularization, and ask for a minimizer (α, ψ) of E_{single} over all rotation angles $\alpha \in \mathbb{R}$ and deformations with vanishing angular momentum, i. e.

$$\psi \in \left\{ \tilde{\psi} \in H^{1,2}(\mathcal{D}, \mathbb{R}^2) \,:\, \int_{\mathcal{D}} \tilde{\psi}_2(x) x_1 - \tilde{\psi}_1(x) x_2 \,\mathrm{d}x = 0 \right\}.$$

As usual, $H^{1,2}(\mathcal{D}, \mathbb{R}^2)$ denotes the Sobolev space of vector-valued L^2-functions on $\mathcal{D}$ whose weak first derivatives are in L^2. Note, that E_{elast} is translation invariant, i. e.

$$E_{\text{elast}}[\psi] = E_{\text{elast}}[\psi + c] \text{ for any } c \in \mathbb{R}^2.$$

Thus, one should expect minimizers only up to a translation. However, because of the special structure of E_{fid}, the translation can be expected to be unique up to a multiple of the lattice spacing in each lattice direction.

3.2.3 Euler–Lagrange equations for the single grain case

Now that we formulated the variational problem for the single grain case, we can derive the corresponding Euler–Lagrange equations. At first, we derive the variations of the fidelity term, which is a little less straightforward to handle than the regularity term. Using the Heaviside function (cf. (1.7)), we can conveniently rewrite the characteristic function $\chi_{\{u>\theta\}}$ (3.1) of a super level set $\{u > \theta\}$, i. e.

$$\chi_{\{u>\theta\}}(x) = H\left(u(x) - \theta\right)$$

and obtain the following formulation of the indicator function f (cf. (3.5) and (3.3)):

$$f[\alpha, \psi](x) = \frac{d^2}{mr^2} H(u(\psi(x)) - \theta) \sum_{i=1}^{m} \left(1 - \left(H(u(\psi(x + M(\alpha)q_i)) - \theta)\right)\right). \tag{3.12}$$

Using

$$\frac{d}{d\epsilon}\left(\int_\Omega H(u((\psi + \epsilon\zeta)(x)) - \theta) \,\mathrm{d}x \right)\Big|_{\epsilon=0} = \int\limits_{\{u\circ\psi=\theta\}} \nabla u(\psi(x)) \cdot \zeta(x)$$

we get the variation of E_{fid} with respect to the deformation in a test direction ζ

$$\begin{aligned}
&\langle \partial_\psi E_{\text{fid}}[\alpha, \psi], \zeta \rangle \\
&= \frac{d^2}{mr^2} \int\limits_{\{u\circ\psi=\theta\}} \nabla u(\psi(x)) \cdot \zeta(x) \sum_{i=1}^{m} \left(1 - H(u(\psi(x + M(\alpha)q_i)) - \theta)\right) \mathrm{d}\mathcal{H}^1 \\
&\quad - \frac{d^2}{mr^2} \sum_{i=1}^{m} \int\limits_{\{u\circ\psi(\cdot + M(\alpha)q_i))=\theta\}} H(u(\psi(x)) - \theta) \nabla u(\psi(x + M(\alpha)q_i)) \cdot \zeta(x + M(\alpha)q_i) \,\mathrm{d}\mathcal{H}^1.
\end{aligned}$$

The second row can be simplified by an integral substitution ($y = x + M(\alpha)q_i$), leading to

$$\begin{aligned}\langle \partial_\psi E_{\text{fid}}[\alpha,\psi],\zeta\rangle &= \frac{d^2}{mr^2}\int\limits_{\{u\circ\psi=\theta\}} \nabla u(\psi(x))\cdot\zeta(x)\sum_{i=1}^{m}(1-H(u(\psi(x+M(\alpha)q_i))-\theta))\mathrm{d}\mathcal{H}^1\\ &\quad-\frac{d^2}{mr^2}\sum_{i=1}^{m}\int\limits_{\{u\circ\psi=\theta\}} H(u(\psi(x-M(\alpha)q_i))-\theta)\nabla u(\psi(x))\cdot\zeta(x)\,\mathrm{d}\mathcal{H}^1\\ &= \frac{d^2}{mr^2}\int\limits_{\{u\circ\psi=\theta\}} \nabla u(\psi(x))\cdot\zeta(x)\sum_{i=1}^{m}\Big(1-H(u(\psi(x+M(\alpha)q_i))-\theta)\\ &\qquad\qquad -H(u(\psi(x-M(\alpha)q_i))-\theta)\Big)\,\mathrm{d}\mathcal{H}^1.\end{aligned}$$

In a straightforward manner one obtains the variation of E_{elast}

$$\begin{aligned}\langle \partial_\psi E_{\text{elast}}[\psi],\zeta\rangle &= \int_{\mathcal{D}}\left(D\psi(x)+D\psi(x)^T-2\,\mathbb{1}\right):\left(D\zeta(x)+D\zeta(x)^T\right)\mathrm{d}x\\ &= 2\int_{\mathcal{D}}\left(D\psi(x)+D\psi(x)^T-2\,\mathbb{1}\right):D\zeta(x)\,\mathrm{d}x,\end{aligned}$$

where we used the symmetry relation $A^T : B = A : B^T$.

To get from the weak formulation of the variations to the strong Euler–Lagrange equations, we take the usual route: Integration by parts yields

$$\begin{aligned}\langle \partial_\psi E_{\text{elast}}[\psi],\zeta\rangle = &-2\int_{\mathcal{D}}\operatorname{div}\left(D\psi(x)+D\psi(x)^T-2\,\mathbb{1}\right)\cdot\zeta(x)\,\mathrm{d}x\\ &+2\int_{\partial\mathcal{D}}\left(D\psi(x)+D\psi(x)^T-2\,\mathbb{1}\right)\nu(x)\cdot\zeta(x)\,\mathrm{d}x,\end{aligned}$$

where ν denotes the outer normal to $\partial\mathcal{D}$. Combined with

$$\begin{aligned}\operatorname{div}\left(D\psi(x)+D\psi(x)^T-2\,\mathbb{1}\right) &= \operatorname{div}\left(D\psi(x)+D\psi(x)^T\right) = \left(\sum_{i=1}^{2}\partial_i\left(\partial_i\psi_j+\partial_j\psi_i\right)\right)_j\\ &= \Delta\psi+\nabla\operatorname{div}\psi\end{aligned}$$

we end up with

$$\begin{aligned}\langle \partial_\psi E_{\text{elast}}[\psi],\zeta\rangle = &-2\int_{\mathcal{D}}\left(\Delta\psi(x)+\nabla\operatorname{div}\psi(x)\right)\cdot\zeta(x)\,\mathrm{d}x\\ &+2\int_{\partial\mathcal{D}}\left(D\psi(x)+D\psi(x)^T-2\,\mathbb{1}\right)\nu(x)\cdot\zeta(x)\,\mathrm{d}x\,.\end{aligned}$$

At first, we only consider test functions ζ that vanish on the boundary of the atomic dots, i. e. $\{u\circ\psi=\theta\}$, because in that case $\langle\partial_\psi E_{\text{fid}}[\alpha,\psi],\zeta\rangle$ vanishes. For reasons of simplification, we also assume $\{u\circ\psi=\theta\}$ and $\partial\mathcal{D}$ to be disjoint. Due to the fact that $u=0$ in $\mathcal{D}\setminus\Omega$ and $\theta>0$, this is not an unreasonable assumption for a minimizing deformation.

If we do not impose the constraint (3.7), using the fundamental lemma of the calculus of variations we can deduce from the necessary condition of a minimizer, i. e. $\langle\partial_\psi E_{\text{single}}[\alpha,\psi],\zeta\rangle = 0$,

the following system of partial differential equations

$$\begin{aligned} -2\mu(\Delta\psi + \nabla \operatorname{div}\psi) &= 0 \text{ on } \mathcal{D} \setminus \{u \circ \psi = \theta\}, \\ (D\psi^T + D\psi - 2\,\mathbb{1}) \cdot \nu &= 0 \text{ on } \partial\mathcal{D}. \end{aligned}$$

If the vanishing angular momentum constraint (3.7) is imposed, the natural Neumann boundary conditions still hold, but the first equation is different. To handle the constraint, we use the theory of Lagrange multipliers [131] and define the Lagrangian function

$$L[\alpha, \psi, \lambda] := E_{\text{single}}[\alpha, \psi] + \lambda \int_{\mathcal{D}} \psi_2(x)x_1 - \psi_1(x)x_2 \,\mathrm{d}x$$

that encodes the constraint. Its first variation with respect to ψ is

$$\begin{aligned} \langle \partial_\psi L[\alpha, \psi, \lambda], \zeta\rangle &= \langle \partial_\psi E_{\text{single}}[\alpha, \psi], \zeta\rangle + \lambda \int_{\mathcal{D}} \zeta_2(x)x_1 - \zeta_1(x)x_2 \,\mathrm{d}x \\ &= \langle \partial_\psi E_{\text{single}}[\alpha, \psi], \zeta\rangle + \lambda \int_{\mathcal{D}} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} x \cdot \zeta \,\mathrm{d}x, \end{aligned}$$

and it yields the following optimality condition (including a Lagrange multiplier $\lambda \in \mathbb{R}$ to handle the constraint), well-known from linearized elasticity:

$$-2\mu(\Delta\psi(x) + \nabla \operatorname{div}\psi(x)) = \lambda \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} x \text{ for } x \in \mathcal{D} \setminus \{u \circ \psi = \theta\}$$

Now that we derived the equations for ψ everywhere but in $\{u \circ \psi = \theta\}$, we turn to this by considering test functions whose support intersects $\{u \circ \psi = \theta\}$. Since $\{u \circ \psi = \theta\}$ separates $\mathcal{D}$ into two disjoint parts, i. e. $\{u \circ \psi > \theta\}$ and $\{u \circ \psi < \theta\}$, we can split the integral in $\langle \partial_\psi E_{\text{elast}}[\psi], \zeta\rangle$ into two integrals. Separate integration by parts then leads to the jump term $2\left[(D\psi^T + D\psi - 2\,\mathbb{1})\nu\right](x)$ for $x \in \{u \circ \psi = \theta\}$, where $\nu(x)$ denotes the outer normal to $\{u \circ \psi = \theta\}$ at x and

$$[\xi](x) := \lim_{\epsilon \to 0} \xi(x + \epsilon\nu(x)) - \xi(x + \epsilon\nu(x))$$

denotes the jump of a function ξ at position x. Thus, combining this with what we know about $\langle \partial_\psi E_{\text{fid}}[\alpha, \psi], \zeta\rangle$, for x in $\{u \circ \psi = \theta\}$ we end up with the following jump condition:

$$\begin{aligned} \left[(D\psi^T(x) + D\psi(x) - 2\,\mathbb{1})\nu(x)\right] = \frac{d^2}{2m\mu r^2} \nabla u(\psi(x)) \sum_{i=1}^{m} \Big(& H(u(\psi(x + M(\alpha)q_i)) - \theta) - 1 \\ & + H(u(\psi(x - M(\alpha)q_i)) - \theta) \Big). \end{aligned}$$

Using the same integral substitution we already applied when calculating $\langle \partial_\psi E_{\text{fid}}[\alpha, \psi], \zeta\rangle$, i. e. $y = x + M(\alpha)q_i$, we obtain the variation with respect to the scalar quantity α:

$$\begin{aligned} &\partial_\alpha E_{\text{fid}}[\alpha, \psi] \\ &= -\frac{d^2}{mr^2} \sum_{i=1}^{m} \int_{\{u \circ \psi = \theta\}} H(u(\psi(x - M(\alpha)q_i)) - \theta) \nabla u(\psi(x)) \cdot \left(D\psi(x) M'(\alpha) q_i\right) \,\mathrm{d}\mathcal{H}^1 \\ &= -\frac{d^2}{mr^2} \sum_{i=1}^{m} M'(\alpha)q_i \cdot \int_{\{u \circ \psi = \theta\}} H(u(\psi(x - M(\alpha)q_i)) - \theta) D\psi(x)^T \nabla u(\psi(x)) \,\mathrm{d}\mathcal{H}^1 \end{aligned} \tag{3.13}$$

where

$$M'(\alpha) = \begin{pmatrix} -\sin\alpha & -\cos\alpha \\ \cos\alpha & -\sin\alpha \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} M(\alpha)\,.$$

Finally, collecting everything we derived so far leads to the Euler–Lagrange equations subsumed as follows:

3.2.1 Proposition (Euler–Lagrange equations in the single grain case)**.** *Suppose the input image u to be sufficiently smooth and (α, ψ) to be a minimizer of* (3.11) *under the vanishing angular momentum constraint* (3.7)*. Furthermore, assume the minimizing deformation ψ to be sufficiently smooth as well, the interface $\{u \circ \psi = \theta\}$ to be a set of piecewise smooth curves and assume $\{u \circ \psi = \theta\}$ and $\partial\mathcal{D}$ not to intersect.*

Then, there exists a Lagrange multiplier $\lambda \in \mathbb{R}$ such that the deformation ψ solves the system of partial differential equations

$$-2\mu(\Delta\psi(x) + \nabla\,\mathrm{div}\psi(x)) = \lambda \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} x \text{ for } x \in \mathcal{D} \setminus \{u \circ \psi = \theta\},$$

the jump condition for the elastic stresses

$$\begin{aligned}\left[(D\psi^T + D\psi - 2\,\mathbb{1}) \cdot \nu\right] = \frac{d^2}{2m\mu r^2}(\nabla u) \circ \psi \sum_{i=1}^{m} \Big(& H(u \circ \psi(\cdot + M(\alpha)q_i) - \theta) - 1 \\ & + H(u \circ \psi(\cdot - M(\alpha)q_i) - \theta)\Big) \\ & \text{on } \{u \circ \psi = \theta\},\end{aligned}$$

and the natural boundary conditions

$$(D\psi^T + D\psi - 2\,\mathbb{1}) \cdot \nu = 0 \text{ on } \partial\mathcal{D}.$$

Here, ν denotes the outer normal to $\{u \circ \psi = \theta\}$ and $\partial\mathcal{D}$ respectively. Finally, the grain angle α fulfills

$$0 = \sum_{i=1}^{m} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} M(\alpha)q_i \cdot \int\limits_{\{u\circ\psi=\theta\}} H(u(\psi(x - M(\alpha)q_i)) - \theta) D\psi(x)^T \nabla u(\psi(x))\,\mathrm{d}\mathcal{H}^1.$$

3.2.4 Regularization and numerical approximation

The Euler–Lagrange equations above show that the discontinuous integrant of the fidelity energy (3.6) leads to concentration on the interfaces $\{u \circ \psi = \theta\}$. To get a robust minimization algorithm and an effective numerical approximation, we avoid any explicit handling of the level line interfaces by a suitable regularization of the discontinuous aspect of the fidelity functional. We replace the discontinuous Heaviside function H used in the indicator function (3.12) by the smeared out approximation $H_\epsilon(s) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{s}{\epsilon}\right)$ defined in (1.11) that was already used in the regularized Mumford–Shah model proposed by Chan and Vese (discussed in Section 1.2.2). Here, $\epsilon > 0$ controls the strength of the regularization. Thus, we get the regularized indicator function

$$f_\epsilon[\alpha, \psi](x) = \frac{d^2}{mr^2} H_\epsilon(u(\psi(x)) - \theta) \sum_{i=1}^{m} (1 - (H_\epsilon(u(\psi(x + M(\alpha)q_i)) - \theta))) \tag{3.14}$$

which in turn leads to the regularized fidelity energy

$$E^{\epsilon}_{\mathrm{fid}}[\alpha,\psi] = \int_{\Omega} f_{\epsilon}[\alpha,\psi](x)\,\mathrm{d}x$$

and the total regularized energy

$$E^{\epsilon}_{\mathrm{single}}[\alpha,\psi] = E^{\epsilon}_{\mathrm{fid}}[\alpha,\psi] + \mu E_{\mathrm{elast}}[\psi].$$

In the numerics, we choose the regularization scale parameter to equal the mesh resolution h (cf. Section 6.1), i. e. $\epsilon = h$. Thus ϵ represents the data resolution on the microscale of the atomic dot pattern.

Analogously to the calculation of $\langle \partial_{\psi} E_{\mathrm{fid}}[\alpha,\psi],\zeta\rangle$ in Section 3.2.3, we compute the variation of the regularized energy with respect to the deformation using the integral substitution $y = x + M(\alpha)q_i$ and the zero property of u outside Ω. This leads to

$$\begin{aligned}\left\langle \partial_{\psi} E^{\epsilon}_{\mathrm{single}}[\alpha,\psi],\zeta\right\rangle = 2\mu \int_{\mathcal{D}} \left(D\psi(x) + D\psi(x)^T - 2\,\mathbb{1}\right) : D\zeta(x)\,\mathrm{d}x \\ + \frac{d^2}{mr^2}\int_{\Omega} \Big[H'_{\epsilon}(u(\psi(x)) - \theta)\nabla u(\psi(x))\cdot\zeta(x) \\ \sum_{i=1}^{m}\big(1 - H_{\epsilon}(u(\psi(x + M(\alpha)q_i)) - \theta) \\ - H_{\epsilon}(u(\psi(x - M(\alpha)q_i)) - \theta)\big)\Big]\,\mathrm{d}x\,.\end{aligned}$$

The variation of the regularized energy with respect to the lattice orientation α is

$$\partial_{\alpha} E^{\epsilon}_{\mathrm{single}}[\alpha,\psi] = \int_{\Omega} \partial_{\alpha} f_{\epsilon}[\alpha,\psi](x)\,\mathrm{d}x,$$

where the integrand is the straightforward α-derivative of the regularized indicator function f_{ϵ}

$$\begin{aligned}\partial_{\alpha} f_{\epsilon}[\alpha,\psi](x) = -\frac{d^2}{mr^2}\sum_{i=1}^{m}\big(\nabla u(\psi(x + M(\alpha)q_i))\cdot(D\psi(x + M(\alpha)q_i)M'(\alpha)q_i) \\ H_{\epsilon}(u(\psi(x)) - \theta)H'_{\epsilon}(u(\psi(x + M(\alpha)q_i)) - \theta)\big).\end{aligned} \tag{3.15}$$

We minimize the energy $E^{\epsilon}_{\mathrm{single}}$ alternatingly with respect to the deformation ψ and the scalar orientation variable α, i. e. each minimization step consists of a minimization step in ψ followed by a minimization step in α, analogous to the generic EM procedure discussed in Section 2.1.2. The actual minimization steps with respect to the individual unknowns are performed using gradient flow techniques (cf. Section 6.2). The descent in α is done with a standard gradient descent, while the descent in ψ uses a regularized gradient descent with the regularizing metric (6.5)

$$g_{\sigma}(\zeta_1,\zeta_2) = \int_{\mathcal{D}} \zeta_1(x)\cdot\zeta_2(x) + \frac{\sigma^2}{2} D\zeta_1(x) : D\zeta_2(x)\,\mathrm{d}x \tag{3.16}$$

on variations ζ_1, ζ_2 of the deformation. As discussed in Section 6.2, σ represents the filter width of the time discrete and implicit heat equation filter kernel corresponding to the inverse of the metric which is denoted by A_{σ}^{-1}. In both cases, we use the Armijo rule [25] separately to determine the step size used in the descents (cf. Section 6.3).

Handling the constraint on the deformation

The constraint on the deformation is integrated in the gradient descent as follows: In each iteration, we first perform a regularized gradient descent step for ψ without respecting the constraint. Then we project the new estimate for ψ back onto the space of deformations that fulfill the constraint. Because the constraint (3.8) turned out to be favorable for the numerical implementation compared to (3.7), we confine to describing the back projection with respect to (3.8). Furthermore, all results involving deformations shown in this chapter were obtained using (3.8). The projection itself is established by the following lemma:

3.2.2 Lemma. *Let* $\psi \in H^1(\mathcal{D}, \mathbb{R}^2)$ *and*

$$S[\psi] := \frac{1}{2|\mathcal{D}|} \int_{\mathcal{D}} D\psi(x) - (D\psi)^T(x) \, \mathrm{d}x \in \mathbb{R}^{2\times 2}.$$

Then $\psi - S[\psi](\cdot - x_\Omega) : \mathcal{D} \to \mathbb{R}^2, x \mapsto \psi(x) - S[\psi]x - S[\psi]x_\Omega$ *fulfills the constraint* (3.8)*. Here,* x_Ω *denotes the center of mass of* Ω*, i. e.*

$$x_\Omega = \frac{1}{|\Omega|} \int_\Omega \mathrm{d}x \, .$$

Proof

$$\begin{aligned}
&\int_{\mathcal{D}} D(\psi - S[\psi](\cdot - x_\Omega))(x) - (D(\psi - S[\psi](\cdot - x_\Omega)))^T(x) \, \mathrm{d}x \\
&\quad = \int_{\mathcal{D}} D\psi(x) - S[\psi] - (D\psi)^T(x) + S[\psi]^T \, \mathrm{d}x \\
&\quad = \int_{\mathcal{D}} D\psi(x) - (D\psi)^T(x) \, \mathrm{d}x - |\mathcal{D}|(S[\psi] - S[\psi]^T) = 0,
\end{aligned}$$

because

$$S[\psi] - S[\psi]^T = 2S[\psi] = \frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} D\psi(x) - (D\psi)^T(x) \, \mathrm{d}x \, .$$

□

Note that, instead of subtracting $S[\psi](\cdot - x_\Omega)$ from ψ to achieve the back projection, we could subtract $S[\psi](\cdot) + b$ for any $b \in \mathbb{R}^2$ instead. We settled for $b = -S[\psi]x_\Omega$ because this way the value of ψ at the center of mass x_Ω is unchanged under the projection.

Spatial discretization and multiscale algorithm

The spatial discretization is carried out with bilinear Finite Elements on a uniform rectangular mesh as described in Section 6.1, i. e. each pixel of our input image u corresponds to a node of the Finite Element mesh. The non-convexity of E^ϵ_{fid} in ψ combined with the high number of unknown values for ψ (two for each pixel of the input image as ψ is allowed to freely deform any node of our FE mesh) leads to a high number of local minimizers of E^ϵ_{fid} with respect to ψ. To account for this, we apply a multiscale minimization strategy to find the minimizing deformation. Unlike the cascadic approach we employed for the registration problem in Section 2.2.3), the strategy used here is not based on a coarse to fine mesh hierarchy, but on the scale parameter σ inherent in the regularizing metric (3.16) utilized in the regularized gradient descent for ψ.

Recall that applying A_σ^{-1} is equivalent to convolving each scalar component of the target with the Gaussian kernel of filter width σ (cf. Section 6.2). Hence, the bigger σ, the more features are removed from the descent direction $\partial_\psi E^\epsilon_{\text{single}}[\alpha^k, \psi^k]$ by the application of A_σ^{-1} (starting with the smallest features for a small value of σ). Thus, we start the algorithm with $\sigma = 1$, i. e. the filter width is equal to the size of the computational domain, and perform the descent until the energy decay per gradient descent step falls below a user selectable threshold $\iota > 0$. Whenever the threshold is hit, σ is halved and the minimization procedure is continued until a relaxation of the energy is achieved (using a tolerance parameter ς) for a filter width of less than the mesh resolution h, i. e. the spatial resolution of a single image pixel. Moreover, recalling (2.15), the Finite Element representation of A_σ is $M_{\text{bl}} + \frac{\sigma^2}{2} L_{\text{bl}}$.

Let us again point out that the regularization induced by A_σ^{-1} does not affect the energy landscape in any way, but solely the descent path towards minima. Therefore, this kind of regularization is conceptually different from the relaxation of the energy itself induced by the smoothing of the Heaviside function controlled by the parameter ϵ.

Finally, Algorithm 3.1 is the full algorithm (in pseudocode notation for the FE case). Numerically, the energy $E^\epsilon_{\text{single}}$ and its variations as well as the matrices are approximated using a Gauss quadrature scheme of order 3 (cf. Section 6.1).

Algorithm 3.1: Minimization of $E^\epsilon_{\text{single}}$

given input image u;
initialize: $\Psi^0 = \text{id}$;
initialize: $\alpha^0 = 0$;
initialize: $\sigma = 1$;
initialize: $k = 0$;
repeat
  $d^\Psi = [M_{\text{bl}} + \frac{\sigma^2}{2} L_{\text{bl}}]^{-1} \partial_\Psi E^\epsilon_{\text{single}}[\alpha^k, \Psi^k]$;
  $\tilde{\Psi}^{k+1} = \Psi^k - \text{ArmijoStepSize}[E^\epsilon_{\text{single}}, \Psi^k, d^\Psi] d^\Psi$;
  if $(E^\epsilon_{\text{single}}[\alpha^k, \Psi^k] - E^\epsilon_{\text{single}}[\alpha^k, \tilde{\Psi}^{k+1}] \leq \iota) \wedge (\sigma \geq h)$ **then**
    $\sigma \leftarrow \frac{\sigma}{2}$;
  end
  $\Psi^{k+1} = \tilde{\Psi}^{k+1} - S(\cdot - x_\Omega)$;
  $d^\alpha = \partial_\alpha E^\epsilon_{\text{single}}[\alpha^k, \Psi^{k+1}]$;
  $\alpha^{k+1} = \alpha^k - \text{ArmijoStepSize}[E^\epsilon_{\text{single}}, \alpha^k, d^\alpha] d^\alpha$;
  $k \leftarrow k + 1$;
until $(|\Psi^{k+1} - \Psi^k|, |\alpha^{k+1} - \alpha^k| \leq \varsigma) \wedge (\sigma < h)$;

Results for the single grain case

To demonstrate the performance of the proposed method, we applied the method to artificial test data and experimental images (Figures 3.5, 3.6 and 3.7). In these applications, the resolution of the input images (and thus the computational domain) ranges from 129×129 to 513×513. Hence, the corresponding grid spacing ranges from $h = 0.0078125$ to $h = 0.001953125$. The size (in grid cells) of the transition layer for typical input data is illustrated by Figure 3.4. For the artificial test cases the lattice spacing is known by construction ($d = 0.072552$), in case of the experimental images it obviously depends on the concrete type of image and is manually

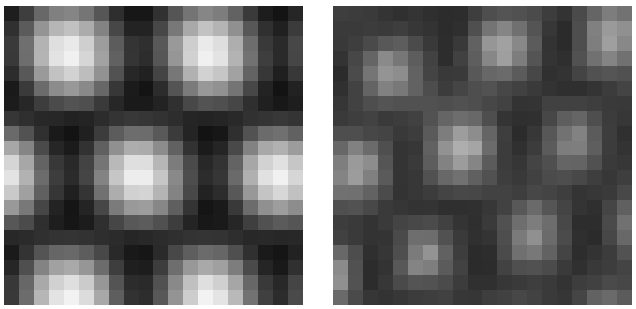


Figure 3.4: Close-ups of typical input images, PFC (left) and TEM (right), illustrating the resolution of the underlying grid.

estimated based on a small image sample in advance. Furthermore, for all applications we used $r = d$, $\epsilon = h$ and $\theta = 0.5$ (except for the image used in the third column of Figure 3.6 and the second and third row of Figure 3.7, there it was necessary to use the smaller threshold $\theta = 0.41$). Due to the vastly different underlying deformations in the artificial test cases and the experimental data, the elastic energy weighting varies from $\mu = 0.1$ (test data) to $\mu = 1.0$ (experimental data). In the second row in Figure 3.6, an even stronger elastic regularization turned out to be appropriate ($\mu = 10.0$).

As first test, we applied a simplified version of the algorithm on artificial test data. The simplification consists of fixing the orientation parameter $\alpha = 0$ and ignoring the constraint on the deformation (the corresponding parts in the algorithm, i. e. the α update and the constraint back projection steps, are skipped). With this simplification, the deformation ψ is supposed to recover not only the non-rotational part of the deformation from the reference frame to the given input image, but the full deformation. Results using this simplification are shown in Figures 3.5 and 3.6. Figure 3.5 shows the recovery of different types of deformations for artificial test cases (each of the test cases was generated by applying a deformation to a perfect reference frame), while Figure 3.6 depicts the deformation obtained for experimental data. Subsequently, results of the full algorithm (simultaneous detection of α and ψ with the back projection, cf. Lemma 3.2.2, to fulfill the constraint) are shown in Figure 3.7. Finally, the energy decay in a particular application of the minimization algorithm is depicted in Figure 3.8.

3.3 Segmenting grain boundaries

As stated earlier, grains are identified by a homogeneous lattice orientation. Combined with the characterization of the underlying atomic lattice derived in Section 3.2.1, this knowledge serves as basis for the Mumford–Shah type model for the segmentation of grain boundaries that we introduce next. For this first segmentation model we assume that there is no elastic deformation on the lattice, whereas the combined grain segmentation and elastic deformation retrieval will be tackled later in Section 3.5. Thereby, the computation of lattice deformation and orientation and the grain segmentation is combined in a joint approach.

3.3.1 A Mumford Shah type model for grain segmentation

As hinted by Figure 3.1, the images we are working with partition the image domain Ω into a number of disjoint grains. Let m denote the number of grains. Then this means that there exists

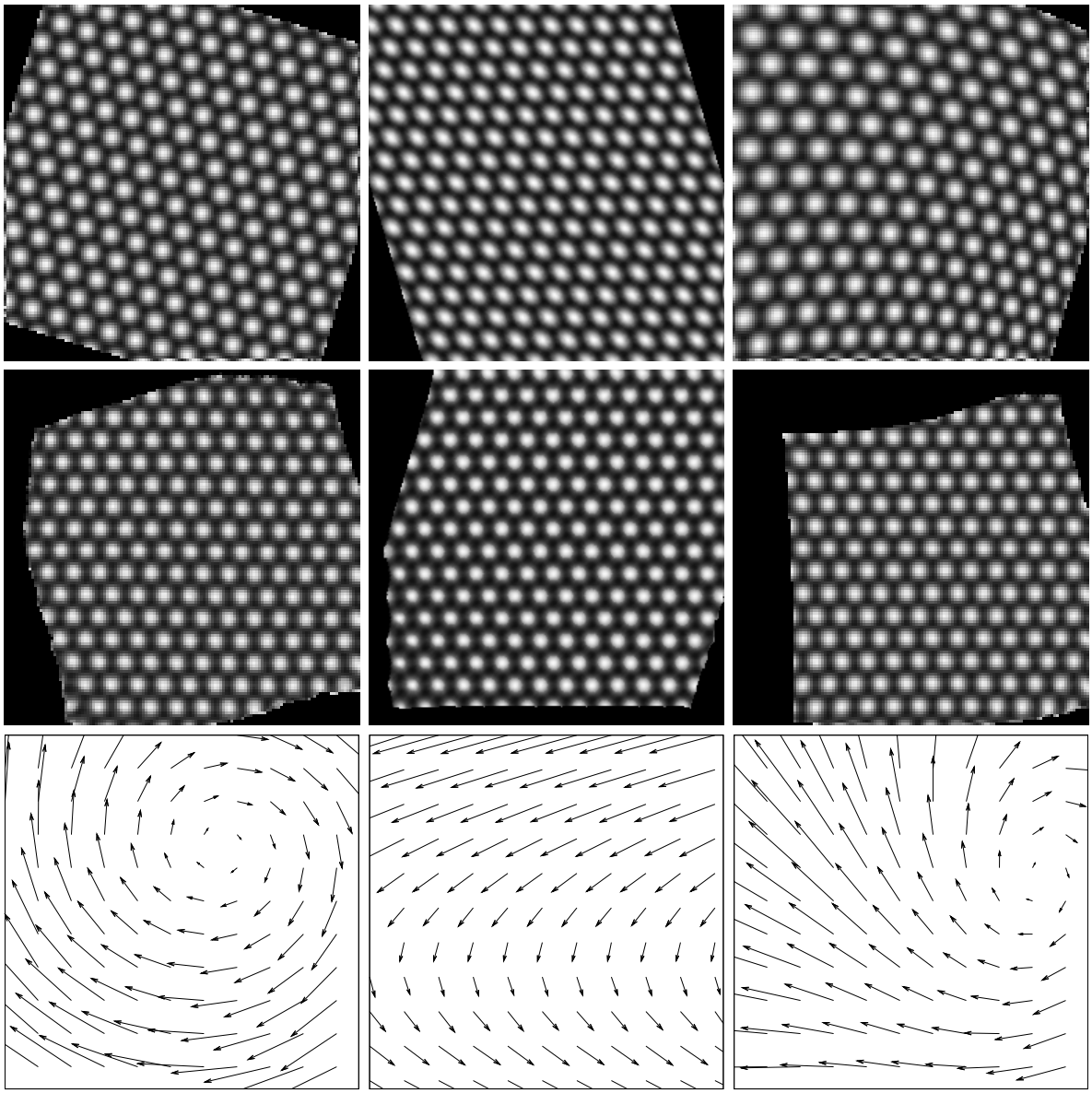


Figure 3.5: Recovery of ψ with the simplified algorithm ($\alpha = 0$ fixed, no constraint for ψ) using artificially deformed lattices as input data. Each of the input images u (top row) reflects a different deformation type: global rotation, global shear and non-homogeneous and nonlinear deformation (from left to right). Furthermore, the deformed images $u \circ \psi^{-1}$ (middle row) and deformations ψ (bottom row) are shown.

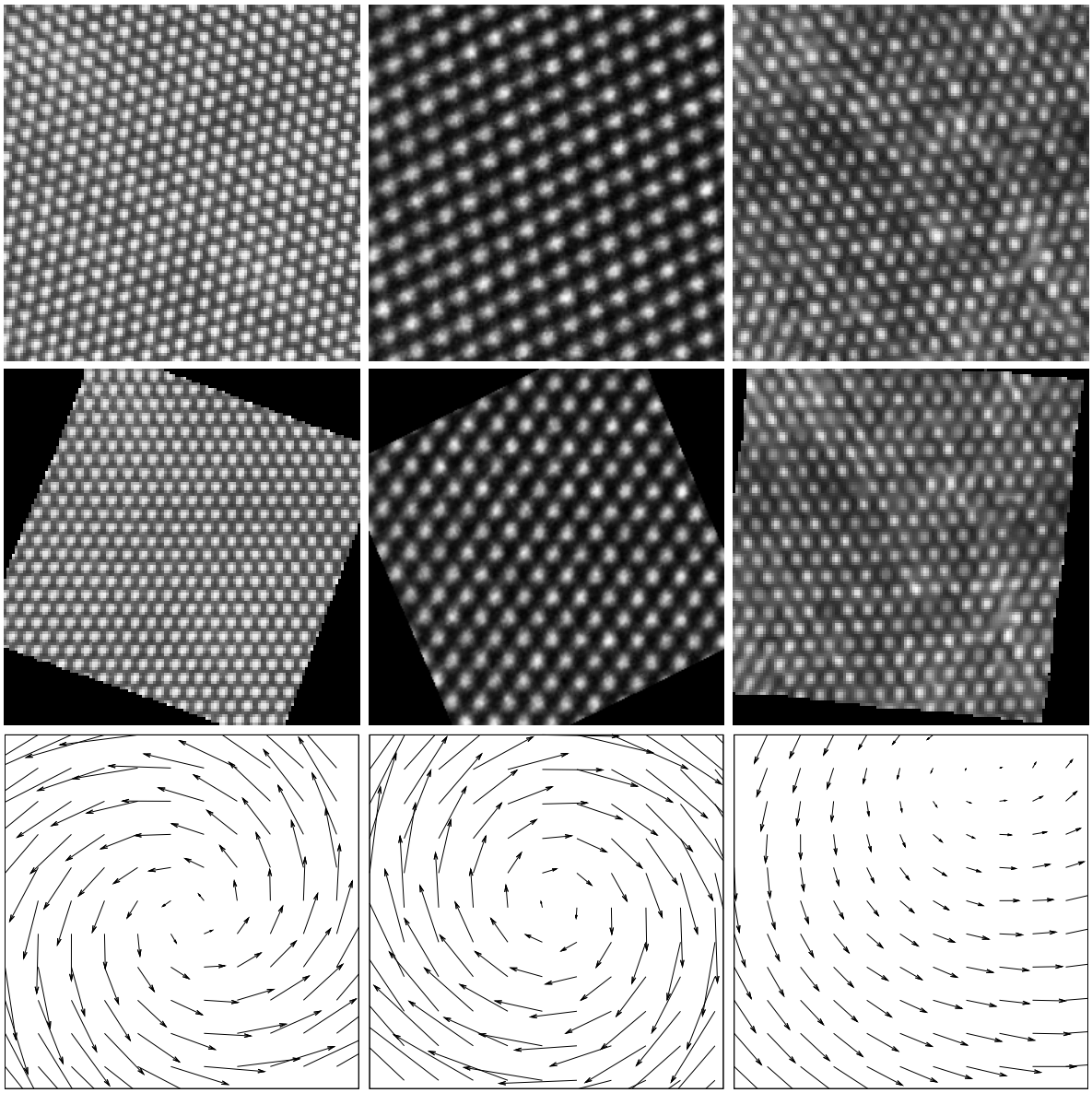


Figure 3.6: Recovery of ψ with the simplified algorithm ($\alpha = 0$ fixed, no constraint for ψ) from experimental data. The input images u (top row) show single atom layers of different metals/metal alloys: GaN, Al and NiTi (from left to right). Furthermore, the deformed images $u \circ \psi^{-1}$ (middle row) and deformations ψ (bottom row) are shown. The TEM image in the first column is courtesy of David M. Tricker (Department of Materials Science and Metallurgy, University of Cambridge), the TEM image in the second column is courtesy of Geoffrey H. Campbell (Lawrence Livermore National Laboratory), the image in the third column is courtesy of Nick Schryvers (Antwerp University)

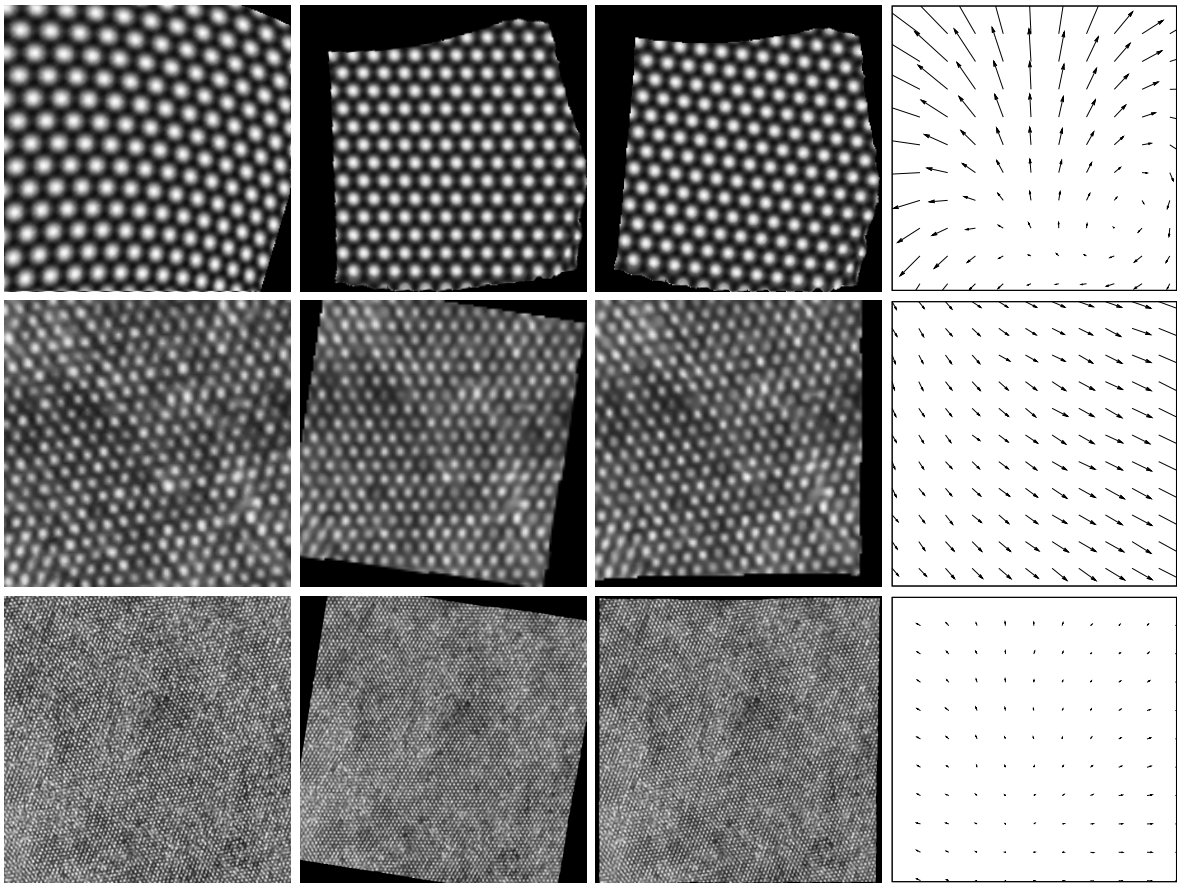


Figure 3.7: Recovery of α and ψ with the $E^{\epsilon}_{\text{single}}$ minimization algorithm from a test image deformed by a nonlinear deformation (top) and experimental images (middle and bottom). The input images u, the deformed and rotated images $u \circ M(-\alpha) \circ \psi^{-1}$, the deformed images $u \circ \psi^{-1}$ and the computed deformations ψ are depicted (from left to right). The recovered angles are $\alpha = 0.119545$ (top row), $\alpha = -0.152956$ (middle row) and $\alpha = -0.159562$ (bottom row).

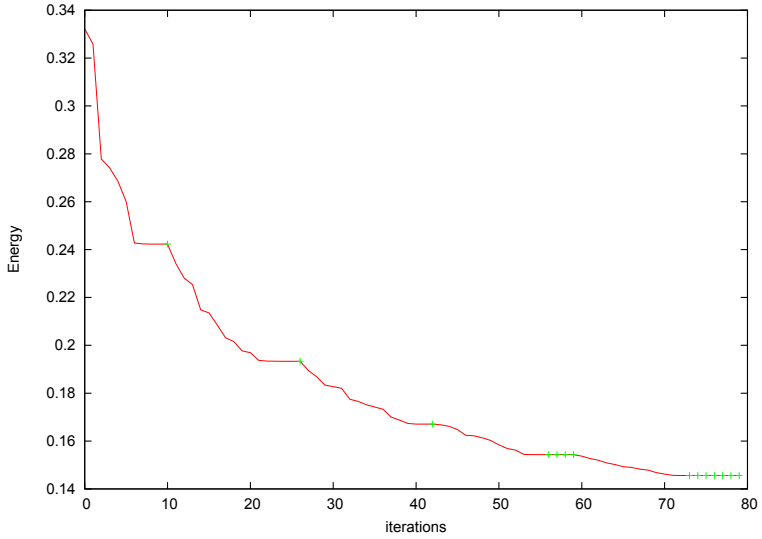


Figure 3.8: Decay of the energy in the simplified algorithm ($\alpha = 0$ fixed, no constraint for ψ) corresponding to the computation of the first column of Figure 3.5. The crosses mark the steps in which the scale parameter σ was automatically refined.

a so-called *partition of* Ω, i. e. a set of open sets $\Omega_1, \dots, \Omega_m$ with finite perimeter such that $\Omega_j \cap \Omega_k = \emptyset$ for $j \neq k$ and $\bigcup_{j=1}^m \overline{\Omega}_j = \overline{\Omega}$. Each of the sets is supposed to represent a distinct grain and hence characterized by a homogeneous lattice orientation. Henceforth, let $\alpha_j \in \mathbb{R}$ denote the lattice orientation corresponding to the grain Ω_j and define

$$\alpha = \sum_{j=1}^{m} \alpha_j \chi_{\Omega_j}.$$

By construction, α is a piecewise constant function that maps a position in the image domain to the lattice orientation corresponding to the grain in the given region of Ω. Note that α vanishes on

$$\Gamma[(\Omega_j)_{j=1}^m] = \bigcup_{\substack{j,k=1,\dots,m \\ j \neq k}} (\overline{\Omega}_j \cap \overline{\Omega}_k \cap \Omega) \subset \Omega,$$

the set of interfaces between the grains. Treating the grain domains Ω_j and the corresponding lattice orientations α_j as unknowns, we define the grain segmentation functional E_{grain} in the spirit of the Mumford–Shah model:

$$E_{\text{grain}}[(\alpha_j, \Omega_j)_{j=1}^m] = \sum_{j=1}^{m} \left(\int_{\Omega_j} f[\alpha_j](x)\, \mathrm{d}x + \frac{\nu}{2} \operatorname{Per}(\Omega_j) \right). \tag{3.17}$$

Due to the construction of the lattice indicator function $f[\alpha_j]$ (3.2), we can expect a minimizer of this energy to be a reliable identification of the grains and the corresponding lattice orientations from our input image.

Under mild regularity assumptions on the boundaries of the sets Ω_j, it holds that

$$\sum_{j=1}^{m} \operatorname{Per}(\Omega_j) = \sum_{j=1}^{m} \operatorname{Per}(\Omega_j, \Omega) = 2\mathcal{H}^1(\Gamma[(\Omega_j)_{j=1}^m])$$

and

$$\sum_{j=1}^{m} \int_{\Omega_j} f[\alpha_j](x) = \int_{\Omega} f[\alpha](x).$$

The latter is obtained from the definition of α and the fact that $\Gamma[(\Omega_j)_{j=1}^m]$ is a Lebesgue null set. Furthermore, denoting the jump set of α by S_α, we have $S_\alpha = \Gamma[(\Omega_j)_{j=1}^m]$ and hence

$$E_{\text{grain}}[(\alpha_j, \Omega_j)_{j=1}^m] = \int_{\Omega} f[\alpha](x) + \nu \mathcal{H}^1(S_\alpha).$$

This shows the close similarity of the functional E_{grain} to the piecewise constant Mumford–Shah functional (1.2). The only discernible difference is that the lattice indicator function $f[\alpha_j]$ replaces the squared difference to the input image in (1.2) as the segmentation criterion.

At first, let us the consider the binary segmentation case, multiphase segmentation will be discussed in Section 3.3.3. In the binary case, i. e. $m = 2$, there are only two different lattice orientations α_1 and α_2 and the corresponding grain domains Ω_1 and Ω_2. Note that neither of the domains needs to be connected. Because of the partition property constraint of Ω_1 and Ω_2, we know that $\Omega_2 = \Omega \setminus \overline{\Omega_1}$ and henceforth denote Ω_1 by Σ. Taking into account that Σ is open, that $\partial\Sigma \cap \Omega$ is a Lebesgue null set and $\Omega \setminus \Sigma = (\Omega \setminus \overline{\Sigma}) \cup (\partial\Sigma \cap \Omega)$, we have

$$\sum_{j=1}^{2} \int_{\Omega_j} f[\alpha_j](x)\,\mathrm{d}x = \int_{\Sigma} f[\alpha_1](x)\,\mathrm{d}x + \int_{\Omega\setminus\Sigma} f[\alpha_2](x)\,\mathrm{d}x$$

and

$$\sum_{j=1}^{2} \operatorname{Per}(\Omega_j) = \operatorname{Per}(\Sigma, \Omega) + \operatorname{Per}(\Omega \setminus \overline{\Sigma}, \Omega) = \operatorname{Per}(\Sigma, \Omega) + \operatorname{Per}(\overline{\Sigma}) = 2\operatorname{Per}(\Sigma).$$

For the latter, we additionally used [5, Proposition 3.38 (c)+(d)] and

$$\left|\Omega \cap (\Sigma \Delta \overline{\Sigma})\right| = |\Omega \cap \partial\Sigma| = 0.$$

Here, Δ denotes the symmetric difference of two sets, i. e. $A\Delta B = (A\setminus B)\cup(B\setminus A)$. Summarizing what we collected above leads to

$$E_{\text{grain}}[(\alpha_j, \Omega_j)_{j=1}^2] = \int_{\Sigma} f[\alpha_1](x)\,\mathrm{d}x + \int_{\Omega\setminus\Sigma} f[\alpha_2](x)\,\mathrm{d}x + \nu \operatorname{Per}(\Sigma).$$

Hence, we can equivalently formulate the grain segmentation problem in the two-phase case as a problem on the lattice orientations α_1 and α_2 and the set Σ using the energy

$$E_{\text{grain-2}}[\alpha_1, \alpha_2, \Sigma] = \int_{\Sigma} f[\alpha_1](x)\,\mathrm{d}x + \int_{\Omega\setminus\Sigma} f[\alpha_2](x)\,\mathrm{d}x + \nu \operatorname{Per}(\Sigma). \tag{3.18}$$

3.3.2 Binary grain segmentation

The two-phase functional $E_{\text{grain-2}}$ introduced above is a representative of the prototype Mumford–Shah energy (1.6) and we can apply the Chan–Vese approach discussed in Section 1.2.2. Thus,

we represent Σ by the zero super level set of level set function ϕ, i. e. $\Sigma = \{\phi > 0\}$. Analogously to the derivation of (1.10), we obtain the level set formulation of $E_{\text{grain-2}}$

$$E_{\text{grain,CV}}[\alpha_1, \alpha_2, \phi] := \int_\Omega H(\phi) f[\alpha_1] + (1 - H(\phi)) f[\alpha_2] \,\mathrm{d}x + \nu \left| D(H \circ \phi) \right| (\Omega) \tag{3.19}$$

depending on the level set function ϕ and the two grain orientations α_1 and α_2.

Instead of the Chan–Vese model, we could also use a globally optimal binary segmentation method like the unconstrained thresholding technique we introduce in Chapter 5. Due to the relatively simple topological structure of the grains in the TEM images and the PFC simulation data considered here, it is not necessary to use such a sophisticated method, though.

Now, following our derivation of (1.12) (using the regularized Heaviside function H_δ (1.11) for a scale parameter $\delta > 0$ and the regularization of the absolute value) together with the regularization of the indicator function $f_\epsilon[\alpha, \psi]$ (3.14) from Section 3.2.4, we obtain a regularized Chan–Vese type energy for binary grain segmentation

$$E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}[\alpha_1, \alpha_2, \phi] = \int_\Omega H_\delta(\phi) f_\epsilon[\alpha_1] + (1 - H_\delta(\phi)) f_\epsilon[\alpha_2] + \nu \left| \nabla H_\delta(\phi) \right|_\varrho \mathrm{d}x \,. \tag{3.20}$$

Similar to the minimization strategy for $E^{\epsilon}_{\text{single}}$ discussed in Section 3.2.4, the minimization of $E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}$ is done in an alternating manner with respect to the level set function ϕ and the two orientation values α_1 and α_2. The minimization with respect to any one of the unknowns is again based on a gradient descent, because, unlike the original Chan–Vese gray value segmentation energy, our energy is not quadratic in the segmentation parameters α_1 and α_2 and thus minimization over these parameters is already a non-linear problem.

As basic prerequisite for a gradient descent, we need to compute the variations of the energy. Using (1.13) to handle the perimeter length term, we obtain the variation of the energy with respect to the level set function ϕ:

$$\left\langle \partial_\phi E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}[\alpha_1, \alpha_2, \phi], \vartheta \right\rangle = \int_\Omega H'_\delta(\phi)\vartheta(f_\epsilon[\alpha_1] - f_\epsilon[\alpha_2]) + \nu \int_\Omega \frac{\nabla\phi}{|\nabla\phi|_\varrho} \cdot \nabla(H'_\delta(\phi)\vartheta) \,\mathrm{d}x \,.$$

Note that this variation reflects the sensitivity with respect to modifications of the implicit description of the grain interface $\{\phi = 0\}$. Furthermore, the variations of the energy with respect to the grain orientations are

$$\begin{aligned}
\partial_{\alpha_1} E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}[\alpha_1, \alpha_2, \phi] &= \int_\Omega H_\delta(\phi) \partial_\alpha f_\epsilon[\alpha_1] \,\mathrm{d}x, \\
\partial_{\alpha_2} E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}[\alpha_1, \alpha_2, \phi] &= \int_\Omega (1 - H_\delta(\phi)) \partial_\alpha f_\epsilon[\alpha_2] \,\mathrm{d}x,
\end{aligned}$$

where the α-derivative of the regularized indicator function f_ϵ is given by (3.15).

In analogy to (1.16), the weak formulation of the L^2-gradient flow for $E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}$ is

$$\forall_{\vartheta \in C_0^\infty(\Omega)} \quad \int_\Omega \frac{\partial_t \phi}{H'_\delta(\phi)} \vartheta \,\mathrm{d}x = \int_\Omega (f_\epsilon[\alpha_2] - f_\epsilon[\alpha_1])\vartheta \,\mathrm{d}x - \nu \int_\Omega \nabla\vartheta \cdot \frac{\nabla\phi}{|\nabla\phi|_\varrho} \,\mathrm{d}x \,. \tag{3.21}$$

Note that the right hand side of the equation is equal to $-\langle \partial_\phi F[\alpha_1, \alpha_2, \phi], \vartheta \rangle$, where

$$F[\alpha_1, \alpha_2, \phi] = \int_\Omega (f_\epsilon[\alpha_1] - f_\epsilon[\alpha_2])\phi \,\mathrm{d}x + \nu \left| \nabla\phi \right|_\varrho \mathrm{d}x \,.$$

As in Section 3.2.4 for the deformation ψ, the spatial discretization of the level set function ϕ is done with bilinear Finite Elements on a uniform rectangular mesh. Henceforth, the Finite Element approximation of ϕ is denoted by Φ, whereas $\Lambda_1, \ldots, \Lambda_n$ are a basis of the underlying Finite Element space $\mathcal{V}$ (using the notation from Section 6.1). This leads to the spatially discrete version of (3.21):

$$\forall_{j\in\{1,\ldots,n\}} \quad \int_\Omega \frac{\partial_t \Phi}{H'_\delta(\Phi)} \Lambda_j \,\mathrm{d}x = -\left\langle \partial_\Phi F[\alpha_1, \alpha_2, \Phi], \Lambda_j \right\rangle .$$

Applying an explicit time discretization (i. e. the Euler forward method) with step size τ, we get

$$\forall_{j\in\{1,\ldots,n\}} \quad \int_\Omega \frac{\Phi^{k+1}}{H'_\delta(\Phi^k)} \Lambda_j \,\mathrm{d}x = \int_\Omega \frac{\Phi^{k}}{H'_\delta(\Phi^k)} \Lambda_j \,\mathrm{d}x - \tau \left\langle \partial_\Phi F[\alpha_1, \alpha_2, \Phi^k], \Lambda_j \right\rangle .$$

Using

$$\partial_\Phi F[\alpha_1, \alpha_2, \Phi] = \left(\langle \partial_\Phi F[\alpha_1, \alpha_2, \Phi], \Lambda_i \rangle\right)_i$$

we obtain the following matrix-vector formulation:

$$\begin{aligned} & M[(H'_\delta(\Phi^k))^{-1}]\overline{\Phi}^{k+1} = M[(H'_\delta(\Phi^k))^{-1}]\overline{\Phi}^{k} - \tau \partial_\Phi F[\alpha_1, \alpha_2, \Phi^k] \\ \Leftrightarrow\ & \overline{\Phi}^{k+1} = \overline{\Phi}^{k} - \tau M[(H'_\delta(\Phi^k))^{-1}]^{-1} \partial_\Phi F[\alpha_1, \alpha_2, \Phi^k] \end{aligned}$$

To numerically calculate $M[(H'_\delta(\Phi^k))^{-1}]$, instead of the Gauss quadrature scheme, we use the so-called *method of lumped masses* (cf. [121, Section 15]). Here, essentially the mass of each row is lumped into the corresponding diagonal entry. More precisely, the weight $(H'_\delta(\Phi^k))^{-1}$ is approximated by a piecewise constant function using its value at the center of the elements, and the product of the basis functions $\Lambda_i \Lambda_j$ is approximated by a piecewise multi-linear function using nodal interpolation. This approximation of $\Lambda_i \Lambda_j$ vanishes for $i \neq j$ and thus the weighted lumped mass matrix is diagonal and its inverse can be calculated easily.

Due to the non-convexity of the energy with respect to Φ and in analogy to (3.16), we also want to use a regularized descent for Φ. Noting that the Finite Element representation of (6.3) is $M + \frac{\sigma^2}{2} L$, we apply the inverse to the descent direction and end of with the following update formula for Φ

$$\overline{\Phi}^{k+1} = \overline{\Phi}^{k} - \tau [M + \tfrac{\sigma^2}{2} L]^{-1} [M (H'_\delta(\Phi^k))^{-1}]^{-1} \partial_\Phi F[\alpha_1, \alpha_2, \Phi^k].$$

This regularization is combined with the same kind of multiscale minimization strategy we already employed for the deformation in Algorithm 3.1, steered by the scale parameter σ: Due to the regularizing properties of $[M + \frac{\sigma^2}{2} L]^{-1}$, for relatively big values of σ only coarse scale adjustments of ϕ are allowed, whereas for successively smaller values of σ (i. e. finer scales) more and more details of the grain boundary structures can be resolved by ϕ. Let us point out that the expected spatial accuracy of the grain boundaries is limited by the lattice spacing d. The reason for this is that the atom neighborhood structure our lattice indicator function is build upon is possibly violated for those atoms that are directly at a grain boundary. Nevertheless, we expect sub-lattice accuracy in regions of smooth grain boundaries due to the symmetric treatment of the lattice indicator function in both grains and the overlapping pattern consistency measurement encoded in the fidelity term of the energy.

Finally, Algorithm 3.2 is the full algorithm (in pseudocode notation for the FE case). Numerically, the energy $E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}$ and its variations as well as the matrices (except for $M[(H'_\delta(\Phi^k))^{-1}]$) are approximated using a Gauss quadrature scheme of order 3 (cf. Section 6.1). $M[(H'_\delta(\Phi^k))^{-1}]$ is approximated using mass lumping as described above.

Algorithm 3.2: Minimization of $E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}$

given input image u;
initialize: Φ^0;
initialize: $\alpha_1^0 = 0, \alpha_2^0 = \frac{\pi}{4}$;
initialize: $\sigma = 1$;
initialize: $k = 0$;
repeat
  $d^\Phi = [M + \frac{\sigma^2}{2}L]^{-1}[M(H'_\delta(\Phi^k))^{-1}]^{-1}\partial_\Phi F[\alpha_1^k, \alpha_2^k, \Phi^k]$;
  $\Phi^{k+1} = \Phi^k - \text{ArmijoStepSize}[E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}, \Phi^k, d^\Phi]d^\Phi$;
  if $(E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}[\alpha_1^k, \alpha_2^k, \Phi^k] - E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}[\alpha_1^k, \alpha_2^k, \Phi^{k+1}] \leq \iota) \wedge (\sigma \geq h)$ **then**
    $\sigma \leftarrow \frac{\sigma}{2}$;
  end
  $(d^{\alpha_i})_{i=1,2} = (\partial_{\alpha_i} E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}[\alpha_1^k, \alpha_2^k, \Phi^{k+1}])_{i=1,2}$;
  $\tau^\alpha = \text{ArmijoStepSize}[E^{\delta,\epsilon,\varrho}_{\text{grain,CV}}, (\alpha_1^k, \alpha_2^k), (d^{\alpha_1}, d^{\alpha_2})]$;
  $(\alpha_1^{k+1}, \alpha_2^{k+1}) = (\alpha_1^k, \alpha_2^k) - \tau^\alpha(d^{\alpha_1}, d^{\alpha_2})$;
  $k \leftarrow k + 1$;
until $(|\Phi^{k+1} - \Phi^k|, |(\alpha_1^{k+1}, \alpha_2^{k+1}) - (\alpha_1^k, \alpha_2^k)| \leq \varsigma) \wedge (\sigma < h)$;

Results for the two grain case

To evaluate the segmentation quality of the proposed method, we applied the method to artificial test data (Figure 3.9), PFC simulation data (Figure 3.10) and experimental images (Figure 3.11). The resolution of the input images (and thus the computational domain) in these applications ranges from 129×129 to 513×513. Hence, the corresponding grid spacing ranges from $h = 0.0078125$ to $h = 0.001953125$. As in the previous section, the lattice spacing for the artificial test cases is known by construction ($d = 0.072552$), whereas in case of the PFC simulation image and the experimental images it is manually estimated based on a small image sample in advance. The other parameters are chosen as follows for all applications in this section: $r = d$, $\theta = 0.5$, $\epsilon = \delta = 0.01$, $\nu = 0.05$.

We start the evaluation of the proposed binary grain segmentation algorithm on artificial test data. The test data was created by dividing the image domain in two regions, separated by a sine wave of a certain amplitude, and then filling each region with a homogeneous dot pattern. In one region the dot pattern is given by the reference lattice configuration, in the other region the dot pattern is rotated by a fixed angle. The segmentation results for two such test images with different boundary amplitude are shown in Figure 3.9. The grain boundary is recovered almost perfectly, only at the amplitude peak of the boundary in the second example, the boundary estimation is off by one atom. This is in line with the fact that the expected spatial accuracy of the grain boundaries is limited by the lattice spacing d due to the construction of our lattice indicator function as mentioned earlier. Subsequently, Figure 3.10 shows the results of our algorithm when applied to PFC simulation data. Finally, Figure 3.11 shows results on experimental images acquired by transmission electron microscopy. In contrast to the previous input images used for the segmentation, the TEM images are affected by noise and natural fluctuations in the shape of the atom dots as well as the lattice spacing. Nevertheless, the proposed method accurately detects the grain boundaries demonstrating the robustness

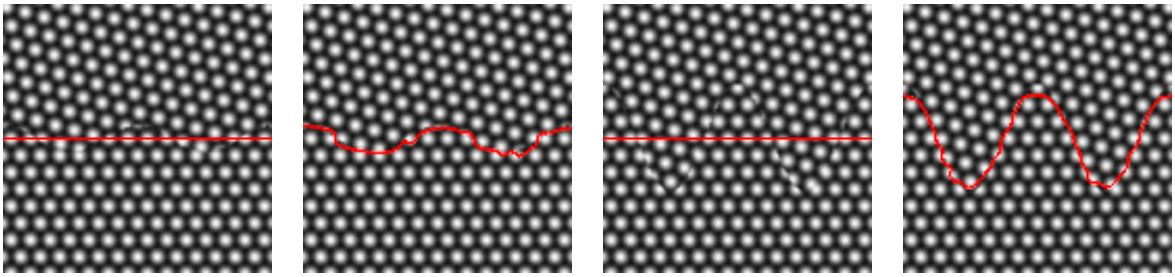


Figure 3.9: Binary grain segmentation of artificial data: Input images overlayed with the grain boundary initialization fed into the algorithm (first and third picture) and the computed grain boundaries (second and fourth picture).

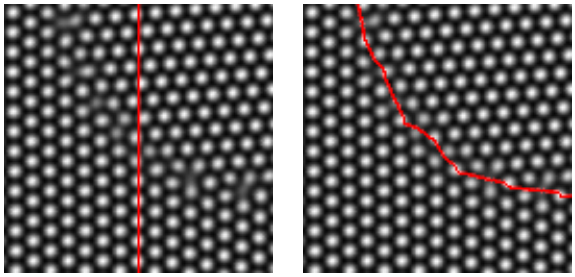


Figure 3.10: Binary grain segmentation of PFC simulation data: Input image overlayed with the grain boundary initialization (left) and the computed grain boundary (right).

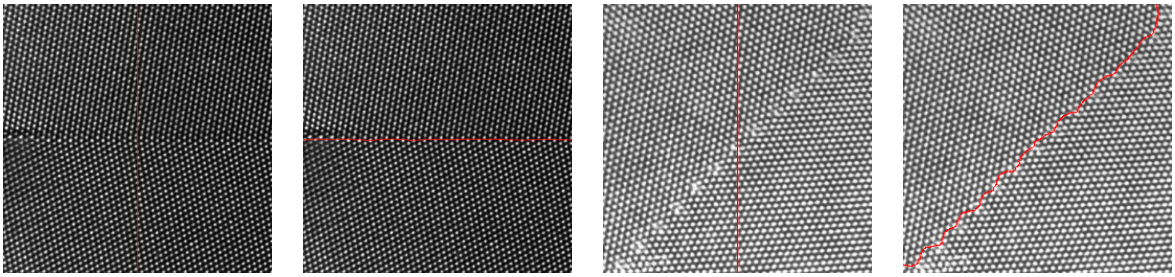


Figure 3.11: Binary grain segmentation of TEM images: Input images overlayed with the grain boundary initialization fed into the algorithm (first and third picture) and the computed grain boundaries (second and fourth picture). The TEM image in the first picture pair is courtesy of Geoffrey H. Campbell, Lawrence Livermore National Laboratory (compare Figure 3.1), the image used in the second picture pair is courtesy of David M. Tricker (Department of Materials Science and Metallurgy, University of Cambridge) showing a $\Sigma 19$ grain boundary in aluminum.

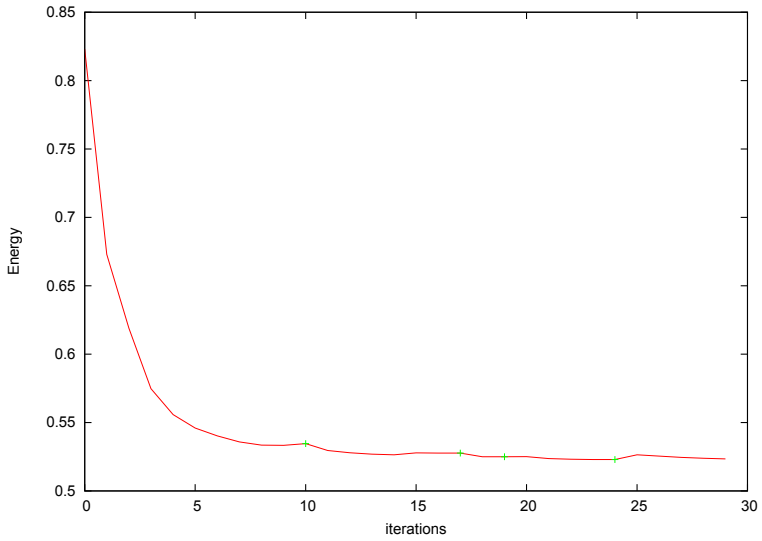


Figure 3.12: Decay of the energy in the binary grain segmentation algorithm corresponding to the computation of the third/fourth row of figure 3.11. The crosses mark the steps in which the scale parameter σ was automatically refined as outlined in Algorithm 3.2. Furthermore, redistancing of ϕ is done every five iterations, which slightly increases the energy in the corresponding steps.

of our approach with respect to these kind of natural effects inherent to experimental image acquisition. Furthermore, the second picture pair shows that the method is capable of detecting effects on an intermediate scale like the oscillating boundary pattern in this input image. In addition, Figure 3.12 exemplarily depicts the energy decay for a specific application of the binary segmentation algorithm.

3.3.3 Multiphase binary grain segmentation

The multiphase extension of the Chan–Vese model developed by Vese and Chan [125] we discussed in Section 1.2.3 can be applied to our grain segmentation model in a straightforward manner and allows us to extend our model to cover the segmentation of more than two grain orientations. Doing so leads us to the following regularized energy for multiphase grain segmentation (cf. (1.17))

$$E^{\delta,\epsilon,\varrho}_{\text{grain,VC}}[(\phi_i)_i,(\alpha_{\boldsymbol{k}})_{\boldsymbol{k}}] = \sum_{\boldsymbol{k}} \int_\Omega \prod_i H_\delta((-1)^{k_i}\phi_i) f_\epsilon[\alpha_{\boldsymbol{k}}]\,\mathrm{d}x + \nu \sum_i \int_\Omega |\nabla(H_\delta \circ \phi_i)|_\varrho \,\mathrm{d}x\,.$$

The minimization of this energy is done like in Algorithm 3.2. The only exception is that the alternating minimization strategy is extended to cover n level set functions instead of only one, and 2^n lattice orientation instead of 2. For this purpose the vector $(\alpha_{\boldsymbol{k}})_{\boldsymbol{k}}$ is treated as one vectorial unknown, whereas each of the level set functions is treated separately.

Figures 3.13 and 3.14 show the performance of the multiphase segmentation model on PFC simulation data [14]. In Figure 3.13, we used two level set functions for the segmentation, allowing for a total of four segments. Since the input data only consists of three different grains, the algorithm only uses three of the four available segments. In particular this shows that the

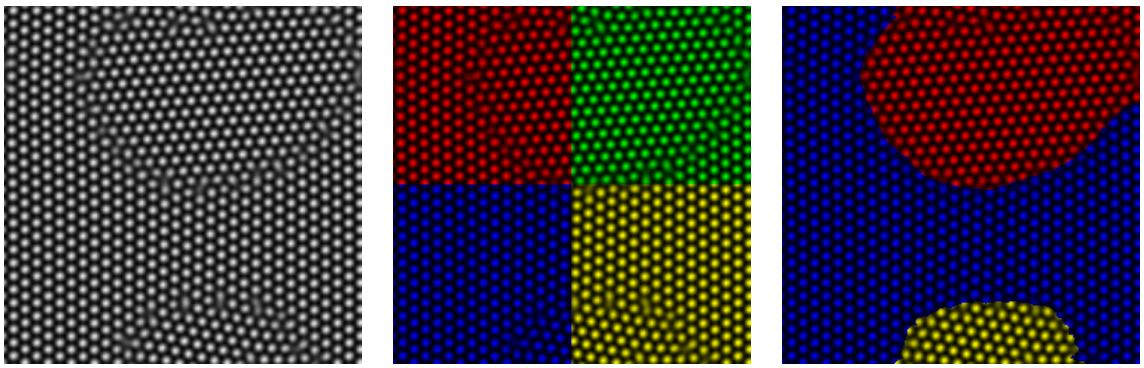


Figure 3.13: Multiphase grain segmentation of PFC simulation data with two level set functions: Original PFC input image (left), input image colored according to the initialization of the segmentation (middle) and the computed segmentation (right).

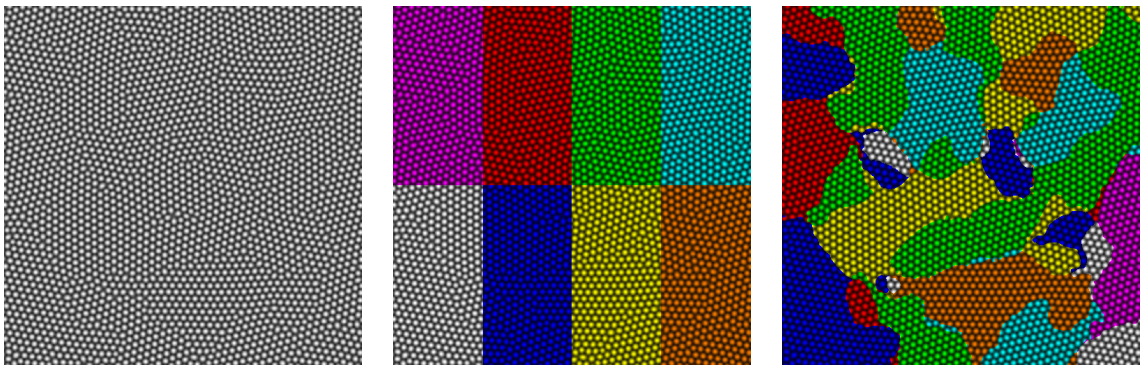


Figure 3.14: Multiphase grain segmentation of PFC simulation data with three level set functions: Original PFC input image (left), input image colored according to the initialization of the segmentation (middle) and the computed segmentation (right).

algorithm is not forced to use all available segments and thus can be used on input data where the number of grains is not a power of two. Subsequently, Figure 3.14 shows a segmentation result obtained with three level set function. Here, all eight available segments are used.

3.4 Detecting a liquid-solid interface

Apart from grain boundaries, there is another type of interface that is of interest in our applications: The interface between the liquid and the solid phase. This kind of interface in particular appears in phase field crystal simulation data of homogeneous nucleation (cf. Figure 3.2). In this section, we will first introduce a Mumford–Shah type model that can handle the segmentation of these two phases. Afterwards, this model is combined with the grain segmentation model to simultaneously handle the detecting of grains boundaries and a liquid-solid interface.

At first we need to derive a way to locally distinguish between the solid and the liquid phase in our input data. For this, we note that the solid state is characterized by the presence of prominent atoms, whereas prominent atom dots are absent in the liquid state. Recalling that

atoms are characterized by high values of u, i. e. if $u(x) > \theta_2$ for a suitable, user selectable threshold θ_2, it is reasonable to assume that there is an atom at position x. Conversely, regions between atoms are characterized by particularly low values of u, i. e. if $u(x) < \theta_1$ for another suitable, user selectable threshold θ_1, x can be assumed to be in a solid region between two atoms. On the other hand, there are neither very high nor low values of u in the liquid state, i. e. if x belongs to the liquid part of the domain, we can suppose $u(x) \in [\theta_1, \theta_2]$. Unfortunately the converse does not hold, because u attains values between θ_1 and θ_2 in solid regions that are in between an atom and a hole. Fortunately, there still is a distinction between such a solid transition region and the liquid phase: The norm of the gradient of u exceeds a certain threshold θ_3 in these transition regions of the solid phase because u has to change from a high value (atom) to a low value (hole) which is not the case in the liquid state.

Therefore, we define the indicator function

$$\begin{aligned} q(x) &:= 1 - \chi_{\{u>\theta_1\}}(x)\chi_{\{u<\theta_2\}}(x)\chi_{\{|\nabla u|<\theta_3\}}(x) \\ &= \begin{cases} 0 & \text{for } u \in (\theta_1, \theta_2) \wedge |\nabla u| < \theta_3, \\ 1 & \text{else.} \end{cases} \end{aligned}$$

and can assume x to be in the liquid phase if $q(x) = 0$ and to be in the solid phase if $q(x) = 1$. Thus, setting $f_1 = q$ and $f_2 = (1 - q)$ in the prototype Mumford–Shah energy (1.6), we obtain a Mumford–Shah type energy for the domain splitting into a liquid phase Ω_L and a solid phase $\Omega \setminus \Omega_L$

$$E_{\text{phase}}[\Omega_L] = \int_{\Omega_L} q(x)\,\mathrm{d}x + \int_{\Omega\setminus\Omega_L} (1 - q(x))\,\mathrm{d}x + \nu_L \operatorname{Per}(\Omega_L). \tag{3.22}$$

Structurally, this energy is very similar to the binary grain segmentation energy (3.18), there is one important difference though: Unlike the grain case, where there is an unknown lattice orientation corresponding to each grain, there are no additional scalar unknowns associated to the two phases. Thus the energy only has to be minimized with respect to the unknown domain Ω_L. This not only simplifies the minimization algorithm (alternating minimization is not necessary), but also frees us of the need to regularize the indicator function q. Following the approach by Chan and Vese as done in Section 3.3.2, we obtain the regularized Chan–Vese type liquid–solid interface detection energy

$$E^{\delta,\varrho}_{\text{phase,CV}}[\phi] = \int_\Omega H_\delta(\phi)q + (1 - H_\delta(\phi))(1 - q) + \nu\,|\nabla H_\delta(\phi)|_\varrho\,\mathrm{d}x\,.$$

This energy can be minimized using Algorithm 3.2, except that the part handling the update of the scalar values is skipped. Figure 3.15 shows a result of this method applied to test data using the thresholds $\theta_1 = 0.3$, $\theta_2 = 0.5$ and $\theta_3 = 1$.

3.4.1 Simultaneously detecting grain boundaries and a liquid-solid interface

Now that we have constructed a functional for the grain segmentation (3.17) and a functional for the liquid-solid separation (3.22), we can combine these functionals to derive a joint model that simultaneously detects the liquid phase Ω_L and finds a partition $\Omega_1, \ldots, \Omega_m$ of the solid

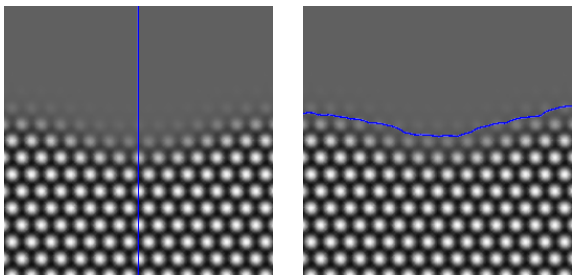


Figure 3.15: Liquid–solid interface detection of artificial test data: Input image overlayed with the interface initialization fed into the algorithm (left) and the computed interface (right).

phase $\Omega \setminus \Omega_L$ into m grains. The joint energy is given by the following weighted sum of both energies:

$$\begin{aligned} E_{\text{p+g,MS}}[\Omega_L, (\alpha_j, \Omega_j)_{j=1}^m] &= \mu E_{\text{phase}}[\Omega_L] + E_{\text{grain}}[(\alpha_j, \Omega_j)_{j=1}^m] \\ &= \mu \int_{\Omega_L} q(x)\,\mathrm{d}x + \mu \int_{\Omega\setminus\Omega_L} (1 - q(x))\,\mathrm{d}x + \nu_L \operatorname{Per}(\Omega_L) \\ &\quad + \sum_{j=1}^{m} \left(\int_{\Omega_j} f[\alpha_j](x)\,\mathrm{d}x + \frac{\nu_G}{2} \operatorname{Per}(\Omega_j) \right). \end{aligned} \tag{3.23}$$

Note that the perimeter weighting parameter in the grain energy is called ν_G instead of ν as in (3.17). This allows to easily distinguish between the weighting parameters corresponding to the grain and the liquid-solid part. The newly introduced weighting parameter μ is used to ensure that the primal decomposition done by the variational formulation is the liquid-solid separating. Thus we will use a relatively large value of μ in the numerical experiments.

In the binary grain case, i. e. $m = 2$, following the derivation of (3.18), the energy simplifies to

$$\begin{aligned} E_{\text{p+g,MS-2}}[\Omega_L, \alpha_1, \alpha_2, \Sigma] &= \mu \int_{\Omega_L} q(x)\,\mathrm{d}x + \mu \int_{\Omega\setminus\Omega_L} 1 - q(x)\,\mathrm{d}x + \nu_L \operatorname{Per}(\Omega_L) \\ &\quad + \int_{\Sigma} f[\alpha_1](x)\,\mathrm{d}x + \int_{(\Omega\setminus\Omega_L)\setminus\Sigma} f[\alpha_2](x)\,\mathrm{d}x + \nu_G \operatorname{Per}(\Sigma) \end{aligned}$$

and the $\Omega \setminus \Omega_L$-partition property of $\Omega_1, \ldots, \Omega_m$ simplifies to the constraint $\Sigma \subset \Omega \setminus \Omega_L$.

3.4.2 Numerical approximation

The numerical handling of the combined grain boundary detecting and liquid-solid phase separation poses a few additional problems compared to the separate handling of these tasks. Since the additional problems of the joint model already occur in case of only two grains, we confine to discussing this case here, but this case can be generalized to handle more than two grains following the multiphase grain model discussed in Section 3.3.3.

The basic approach follows the line of the separate models, i. e. the unknown sets are represented by level set functions. ϕ_L is the representation of the liquid phase, i. e. $\Omega_L = \{\phi_L >$

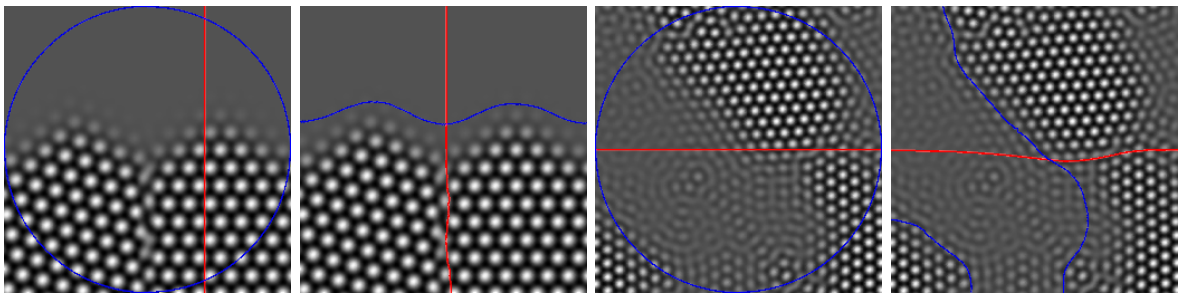


Figure 3.16: Joint liquid-solid separation and grain boundary detection on artificial test data (first picture pair) and on PFC simulation data (second picture pair): Input images overlayed with the initial grain boundary in red and liquid-solid interface in blue (first and third picture) and the computed grain boundary and liquid-solid interface (second and fourth picture).

$0\}$, whereas ϕ_G will be used to represent the set Σ handling the grain segmentation. To simplify the formulation we do not handle the constraint $\Sigma \subset \Omega \setminus \Omega_L$ explicitly by defining ϕ_G only on $\{\phi_L \leq 0\} = \Omega \setminus \Omega_L$. Instead we use a level set function ϕ_G defined on Ω and employ the fact that $\{\phi_L \leq 0\}$ already models the solid phase. Thus we have

$$\Sigma = \{\phi_L \leq 0\} \cap \{\phi_G > 0\} \text{ and } (\Omega \setminus \Omega_L) \setminus \Sigma = \{\phi_L \leq 0\} \cap \{\phi_G \leq 0\}.$$

Using this and following Chan–Vese approach from Section 3.3.2 we obtain the regularized Chan–Vese type energy for combined liquid–solid interface and grain boundary detection

$$\begin{aligned} E^{\delta,\epsilon,\varrho}_{\text{p+g,CV}}[\phi_L, \alpha_1, \alpha_2, \phi_G] = \int_\Omega & \mu[H_\delta(\phi_L)q + (1 - H_\delta(\phi_L))(1-q)] + 2\nu_L \left|\nabla H_\delta(\phi_L)\right|_\varrho \\ & + (1 - H_\delta(\phi_L))\, H_\delta(\phi_G) f_\epsilon[\alpha_1] \\ & + (1 - H_\delta(\phi_L))\, (1 - H_\delta(\phi_G)) f_\epsilon[\alpha_2] \\ & + \nu_G(1 - \beta H_\delta(\phi_L))\, \left|\nabla H_\delta(\phi_G)\right|_\varrho \mathrm{d}x\,. \end{aligned}$$

Here we introduced a new parameter β that allows us to control the regularity of the interface $\{\phi_L > 0\} \cap \{\phi_G = 0\}$. In case $\beta = 1$, this interface is not controlled by the energy at all which corresponds to the behavior of $E_{\text{p+g,MS-2}}$, but also introduces numerical problems. For $0 \leq \beta < 1$ the interface handling differs from the one of the original energy, because the energy sees the length of $\{\phi_L > 0\} \cap \{\phi_G = 0\}$, scaled by $(1 - \beta)\nu_G$. Therefore, in case $0 \leq \beta < 1$, the method extends the grain boundaries through the liquid domain to the boundary of Ω by shortest paths. In our experiments, this simplification did not introduce noticeable artifacts in the numerically obtained results.

The energy derived in this section is minimized like in Algorithm 3.2. The only extension necessary is that the alternating minimization strategy is extended to cover ϕ_L in addition to ϕ_G and (α_1, α_2). Figure 3.16 shows results of the combined liquid-solid interface and grain boundary identification on an artificial test data set and on PFC simulation data. In both applications, the following parameters were used $\beta = 0$, $\mu = 10$, $\nu_L = \nu_G = 0.05$, $r = d$, $\theta = 0.5$, $\theta_1 = 0.3$, $\theta_2 = 0.5$, $\epsilon = \delta = 0.01$. The lattice spacing d was determined like in the previous sections. Apart from d, only the gradient threshold θ_3 needed to be adapted for the particular application: We used $\theta_3 = 1$ for the artificial test data and $\theta_3 = 20$ for the PFC data, respectively.

3.5 Joint deformation and grain geometry extraction

As final model in this chapter, we consider a joint model that combines the grain segmentation model from Section 3.3 with the elastic deformation extraction model from Section 3.2. Thus, we are looking for a partition of Ω into grains $\Omega_1, \dots, \Omega_m$ and the corresponding lattice orientations $\alpha_1, \dots, \alpha_m$ together with a global elastic deformation ψ affecting the lattice spacing in the grains. As basis we a variational formulation use the grain segmentation energy (3.17) but endow it with the local lattice indicator function (3.5) that incorporates a deformation instead of the simpler indicator function (3.2) used in (3.17). Combining this with the elastic regularization term (3.9) and the constraint (3.8) on the mean value of the skew symmetric part of $D\psi$, we obtain the following Mumford–Shah type functional:

$$E_{\text{joint,MS}}[(\alpha_j, \Omega_j)_{j=1}^m, \psi] = \sum_{j=1}^{m} \left(\int_{\Omega_j} f[\alpha_j, \psi] \, \mathrm{d}x + \frac{\nu}{2} \operatorname{Per}(\Omega_j) \right) + \mu E_{\text{elast}}[\psi].$$

In the two grain case, following the arguments from Section 3.3.1, the energy simplifies to

$$E_{\text{joint,MS-2}}[\alpha_1, \alpha_2, \Sigma, \psi] = \int_{\Sigma} f[\alpha_1, \psi] \, \mathrm{d}x + \int_{\Omega \setminus \Sigma} f[\alpha_2, \psi] \, \mathrm{d}x + \nu \operatorname{Per}(\Sigma) + \mu E_{\text{elast}}[\psi].$$

For the sake of simplicity, we confine to numerically investigating the joint model for the case of two grains. The multiphase generalization can be done as described in Section 3.3.3.

Combining the regularization and numerical approximation for the deformation extraction from Section 3.2.4 and the binary grain segmentation from Section 3.3.2, we derive a joint Chan–Vese type model using the following regularized energy functional

$$\begin{aligned} E^{\delta,\epsilon,\varrho}_{\text{joint,CV}}[\alpha_1, \alpha_2, \phi, \psi] = \int_{\Omega} & H_\delta(\phi) f_\epsilon[\alpha_1, \psi] + (1 - H_\delta(\phi)) f_\epsilon[\alpha_2, \psi] + \nu \left| \nabla H_\delta(\phi) \right|_\varrho \mathrm{d}x \\ & + \mu E_{\text{elast}}[\psi]. \end{aligned}$$

that has to be minimized respecting the constraint (3.8) on the skew symmetric part of $D\psi$.

As before ψ and ϕ are discretized using finite elements. The minimization strategy we use for the joint model is a natural combination of Algorithms 3.1 and 3.2. The complete algorithm in pseudocode notation for the FE case is shown in Algorithm 3.3, where

$$F[\alpha_1, \alpha_2, \phi, \psi] = \int_{\Omega} (f_\epsilon[\alpha_1, \psi] - f_\epsilon[\alpha_2, \psi]) \phi \, \mathrm{d}x + \nu \left| \nabla \phi \right|_\varrho \mathrm{d}x \, .$$

The same quadrature is utilized here in the numerical implementation that was already used for the Algorithms 3.1 and 3.2.

Finally, two results of the joint grain segmentation and deformation extraction approach are depicted in Figure 3.17.

3.6 Outlook

While the grain segmentation of still images could be done manually instead of by using an automatic segmentation method like the one we proposed here, this task already becomes very tedious once an image shows a relatively large number of grains (cf. Figure 3.14). At the latest when analyzing time dependent input data, a large number of images has to be segmented,

Algorithm 3.3: Minimization of $E^{\delta,\epsilon,\varrho}_{\text{joint,CV}}$

given input image u;
initialize: Φ^0;
initialize: $\Psi^0 = \text{id}$;
initialize: $\alpha_1^0 = 0, \alpha_2^0 = \frac{\pi}{4}$;
initialize: $\sigma = 1$;
initialize: $k = 0$;
repeat
  $d^\Phi = [M + \frac{\sigma^2}{2}L]^{-1}[M(H'_\delta(\Phi^k))^{-1}]^{-1}\partial_\Phi F[\alpha_1^k, \alpha_2^k, \Phi^k, \Psi^k]$;
  $\Phi^{k+1} = \Phi^k - \text{ArmijoStepSize}[E^{\delta,\epsilon,\varrho}_{\text{joint,CV}}, \Phi^k, d^\Phi]d^\Phi$;
  $d^\Psi = [M_{\text{bl}} + \frac{\sigma^2}{2}L_{\text{bl}}]^{-1}\partial_\Psi E^{\delta,\epsilon,\varrho}_{\text{joint,CV}}[\alpha_1^k, \alpha_2^k, \Phi^{k+1}, \Psi^k]$;
  $\tilde{\Psi}^{k+1} = \Psi^k - \text{ArmijoStepSize}[E^{\delta,\epsilon,\varrho}_{\text{joint,CV}}, \Psi^k, d^\Psi]d^\Psi$;
  if $(E^{\delta,\epsilon,\varrho}_{\text{joint,CV}}[\alpha_1^k, \alpha_2^k, \Phi^k, \Psi^k] - E^{\delta,\epsilon,\varrho}_{\text{joint,CV}}[\alpha_1^k, \alpha_2^k, \Phi^{k+1}, \tilde{\Psi}^{k+1}] \leq \iota) \wedge (\sigma \geq h)$ **then**
    $\sigma \leftarrow \frac{\sigma}{2}$;
  end
  $\Psi^{k+1} = \tilde{\Psi}^{k+1} - S(\cdot - x_\Omega)$;
  $(d^{\alpha_i})_{i=1,2} = (\partial_{\alpha_i} E^{\delta,\epsilon,\varrho}_{\text{joint,CV}}[\alpha_1^k, \alpha_2^k, \Phi^{k+1}, \Psi^{k+1}])_{i=1,2}$;
  $\tau^\alpha = \text{ArmijoStepSize}[E^{\delta,\epsilon,\varrho}_{\text{joint,CV}}, (\alpha_1^k, \alpha_2^k), (d^{\alpha_1}, d^{\alpha_2})]$;
  $(\alpha_1^{k+1}, \alpha_2^{k+1}) = (\alpha_1^k, \alpha_2^k) - \tau^\alpha(d^{\alpha_1}, d^{\alpha_2})$;
  $k \leftarrow k + 1$;
until $(|\Phi^{k+1} - \Phi^k|, |\Psi^{k+1} - \Psi^k|, |(\alpha_1^{k+1}, \alpha_2^{k+1}) - (\alpha_1^k, \alpha_2^k)| \leq \varsigma) \wedge (\sigma < h)$;

making a manual segmentation infeasible. Such time dependent data arises for example when validating physical models with experimental data. In this case, the evolution of the grain boundaries over time is of particular importance and thus is is necessary to have an automatic method that offers a robust and accurate way to extract interface velocities. An extension of our model to time dependent input data could be developed to fulfill these requirements.

So far, the atomic lattice type (number and relative position of neighboring atoms) and its scale (maximal lattice spacing d) are prescribed in our model and hence have to be supplied by the user alongside the input data. If this should prove to limit the usability of our method in the field, one could generalize the variational approach to include the lattice type and scale as unknown parameters in addition to the local lattice orientation.

Furthermore, the isotropic elastic regularization term for the deformation (3.9), while convenient to handle, does not reflect the inherent anisotropic nature of the material induced by the underlying lattice. Thus, an anisotropic regularization term modeling the material properties could be developed to replace the current regularization and to increase the accuracy of the extracted deformation. This can be achieved by choosing a tensor $C(\alpha)$ in (3.10) such that it reflects the anisotropic material properties and takes into account the lattice orientation.

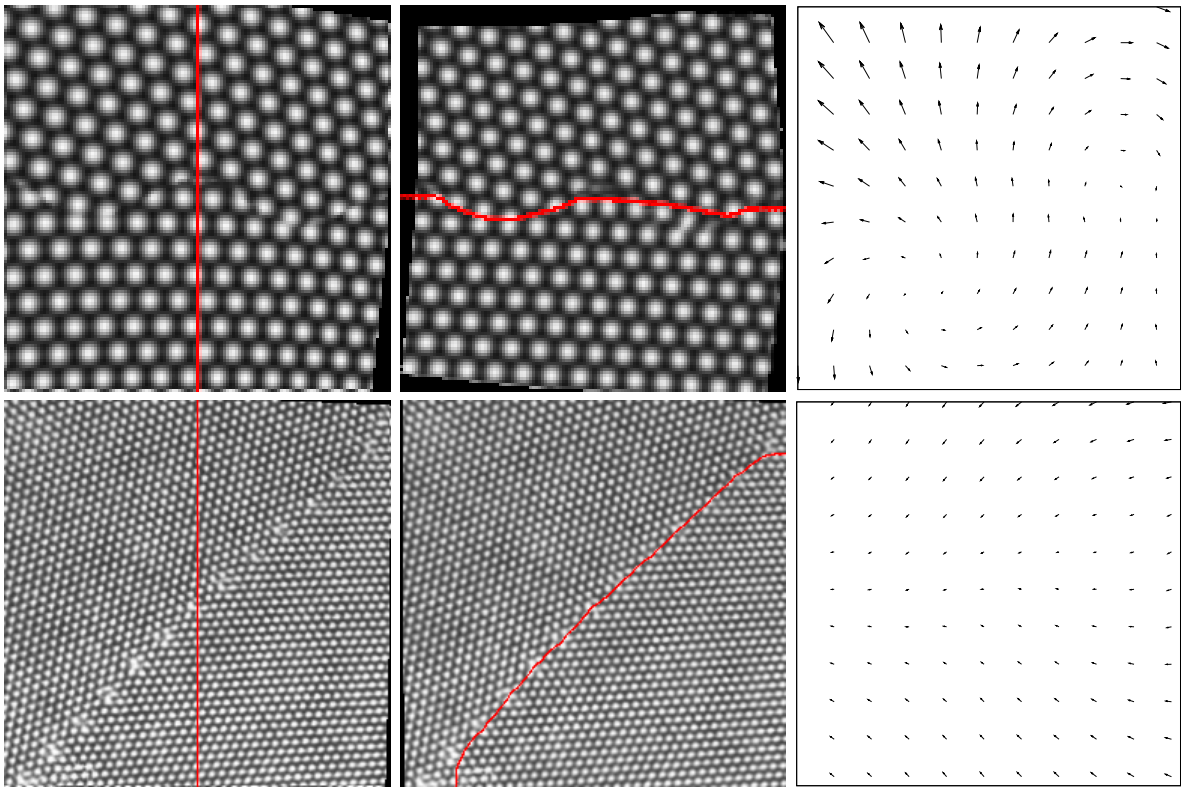


Figure 3.17: Joint grain segmentation and deformation extraction on an artificial test image (top row) and on a TEM image (bottom row), both artificially deformed: Input image u overlayed with the grain boundary initialization (left), input image deformed with the extracted deformation $u \circ \psi^{-1}$ overlayed with the computed grain boundary (middle) and extracted deformation ψ (right).

4 Joint motion estimation and restoration of motion-blurred video

THE computation of the velocity of moving objects in a given image sequence, shortly called motion estimation, is a well-known image processing problem and has been studied extensively in the past. One particularly popular approach to motion estimation is the computation of the optical flow. An abundance of methods has been developed to determine this flow, cf. [78, 129, 33], to name just a few. One commonly known fact is that the clearer the given image sequence is, the more reliable the motion can be estimated. While certain robustness has been addressed in motion estimation, e. g., under varying illumination [83], and contrast [37], a rough survey of the state-of-the-art literature in the subject reveals that the considered videos are quite sharp and in general of sufficiently high quality. In particular, blurred video is very rarely considered in motion estimation techniques.

Here, we investigate how to handle one typical kind of real world blur affecting video sequences: Considering video footage from a standard video camera, it is quite noticeable that relatively fast moving objects appear blurred (cf. Figure 4.1). This effect is called *motion blur*, and it is caused by the way a camera takes pictures, i. e. roughly integrating information in time over the aperture time of the camera. The longer the aperture is open, or the faster the motion, the blurrier moving objects appear (cf. Section 4.2 for a detailed explanation of motion blur).

To improve the accuracy of the motion estimation on a video suffering from motion blur, it would be helpful to remove the motion blur first. On the other hand, if the actual motion is known, the motion blur can be removed by "deconvolution," since the motion gives the velocity of the objects and therefore the exact kernel needed for deconvolution. Realizing that these two problems are intertwined suggests to develop a method to tackle both problems simultaneously.

Thus we introduce a variational method for *joint motion estimation, moving object detection, and motion blur deconvolution from multiple frames* [19]. The proposed framework combines a Mumford–Shah type approach to handle the segmentation of the moving object from the scene background with an explicit modeling of the motion-blur process and image regularization terms. The input to the variational formulation are two consecutive frames showing a moving object

Figure 4.1: Three consecutive frames from a blurred video recorded with an off-the-shelf video camera: The car is moving quickly enough to be affected by motion blur.

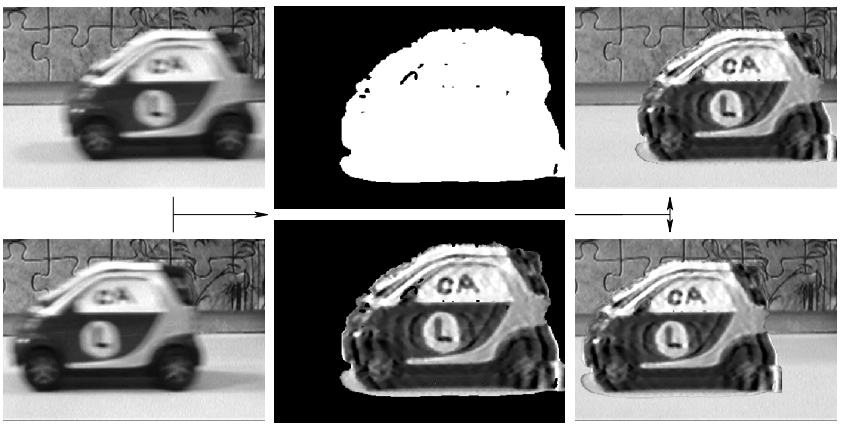


Figure 4.2: From two real blurred frames (left), we automatically and simultaneously estimate the motion region, the motion vector, and the image intensity of the foreground (middle). Based on this and the background intensity the unblurred frames are reconstructed (right).

in front of a static background, while the output are the corresponding reconstructed frames, the segmented moving object, and the actual motion velocity (cf. Figure 4.2). As illustrated in Section 4.5, the joint handling of all unknowns significantly outperforms techniques where each unknown is handled individually.

This chapter is organized as follows: After briefly reviewing related literature in Section 4.1, we develop a *motion blur model which is consistent at motion singularities* in Section 4.2 and derive our joint variational framework in Section 4.3. Section 4.4 is devoted to a detailed description of the energy minimization algorithm. Then, in Section 4.5, results of the joint approach are discussed. Finally, Section 4.6 gives a short outlook.

4.1 Review of related work

The literature on motion estimation is abundant, thus we concentrate only on works addressing blurred video. Similarly, there exist numerous methods to remove motion blur using a single frame, and these often introduce strong assumptions on the scene and/or blur [69]. As an example, let us mention the recent contribution on blind motion deblurring using image statistics presented in [91], where the author explains, as clear from the results, that while the image often well recovered, the actual motion and region of movement are often quite non-accurate. Another recent approach to motion deblurring [79] uses blending with the background but assumes the shift-invariant case, where shift-invariant means that the motion blur kernel does not depend on the spatial position. Furthermore, [18] tackles piecewise shift-variant deblurring, including a segmentation of the blurred regions. Of more interest to our approach are techniques that use multiple frames, and these (some of them hardware-based) are only very few, as summarized in [69]. More on the close connection between our work and [69] will be presented below.

Sequential motion estimation and then deblurring has been reported in [90] (see also [107]), while not addressing a truly joint estimation. The idea of developing joint methods for intertwined problems has become quite popular and successful recently, for example blind deconvolution and denoising [75], segmentation of moving objects in front of a still background and the

computation of the motion velocities [86], segmentation and registration using geodesic active contours [81, 122], anisotropic classification and cartoon extraction [21], optical flow computation and video denoising [103], and edge detection and matching as discussed in Chapter 2.

Motion deblurring can also be obtained using the so-called *super-resolution framework*, see [117] and references therein. The basic idea behind these approaches, which often assume that the blurring kernel is provided, is to obtain a higher resolution image from a collection of low-resolution frames. In addition, these techniques often assume that the whole frame suffers from motion blur (or attack this with robust norms), and do not explicitly separate the moving object from the background or estimate the motion velocity.

The pioneering work by Favaro and Soatto, [69], is the closest to ours, not only because of the use of multiple frames but also because of the joint estimation. In a separate paper [68], they also address the problem of simultaneously inferring the depth map, radiance and motion from motion blurred and defocused images via anisotropic diffusion. Thus, these works address the same challenges as we do here, which is the joint estimation of motion and scene deblurring from multiple frames. Some differences are that the authors of [69] approximate the motion blur with a Gaussian, rather than the more accurate rectangular filter we use, described in Section 4.2. This model leads them to an anisotropic diffusion flow, and inverting it is ill-posed. The model in [69] is designed to handle only very little blur (motion), while the proposed method, as illustrated by the examples below, can handle large velocities and blurs. We also model the crucial blending of the foreground and background, which happens in reality and significantly effects the blur as well as the reconstruction near the boundary of the moving object (see examples in Figures 4.5, 4.8, and 4.10). Finally, we note that while the proposed formulation could deal with multiple moving objects, here we provide examples with only one, whereas [69] develop their work for multiple moving objects— although they present no examples of this capability with real video data.

4.2 Modeling the blurring process

Let $f : [-T, T] \times \Omega, (t, x) \mapsto \mathbb{R}$ denote a continuous sequence of scene intensities over a time interval $[-T, T]$ and on a spatial image domain Ω. The scene is observed and recorded by a video camera resulting in a video sequence, i. e. a set of images $g_i : \Omega \to \mathbb{R}$ associated with times t_i, for $i = 1, \ldots, m$.

The recorded images are integrated measurements of the light intensity emitted from the scene over the aperture time interval of the camera. The following time integral realistically approximates the mechanical shutters of film cameras and the electronic readout of modern CCD video recorders:

$$g_i(x) = \frac{1}{\tau} \int_{t_i - \frac{1}{2}\tau}^{t_i + \frac{1}{2}\tau} f(s, x) \, \mathrm{d}s \, . \tag{4.1}$$

Here, τ denotes the aperture time of the camera. To derive our motion blur model, let us get started by considering the the simplest possible case: The sequence f shows a single object moving at constant velocity $v \in \mathbb{R}^2$ with no visible background, i. e. $f(x - sv) = f(s, x)$. Furthermore, we assume f to be given on $\mathbb{R}^2$, e. g. by extension by zero. Under these assumptions, we can transform the integration in time to an integration in space and obtain for the recorded

images

$$g_i(x) = \frac{1}{\tau}\int_{t_i-\frac{1}{2}\tau}^{t_i+\frac{1}{2}\tau} f(x-sv)\,\mathrm{d}s = \frac{1}{\tau}\int_{-\frac{1}{2}\tau}^{\frac{1}{2}\tau} f(x-t_iv-sv)\,\mathrm{d}s\,.$$

We are going to reformulate this as a convolution, so we can employ existing deconvolution techniques, e. g. [17]. If $v = 0$, there is no motion blur at all, thus we henceforth assume $v \neq 0$. Let $\tilde{x} \in \mathbb{R}^d$ be arbitrary but fixed, define the curve $c(s) = sv$, $\Gamma_v = c([-\frac{\tau}{2}, \frac{\tau}{2}])$ and $\Gamma_v^\infty = c\left((-\infty, \infty)\right)$. Then, we get $|\dot{c}| = |v|$ and

$$\begin{aligned}\frac{1}{\tau}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} f(\tilde{x}-sv)\,\mathrm{d}s &= \frac{1}{\tau\,|v|}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} f(\tilde{x}-c(s))\,|\dot{c}|\,\mathrm{d}s = \frac{1}{\tau\,|v|}\int_{\Gamma_v} f(\tilde{x}-y)\,\mathrm{d}S(y)\\ &= \frac{1}{\tau\,|v|}\int_{\Gamma_v^\infty} f(\tilde{x}-y)H(\tau\,|v|-2\,|y|)\,\mathrm{d}S(y).\end{aligned} \tag{4.2}$$

Here, H denotes the Heaviside function (cf. (1.7)) and we used

$$y \in \Gamma_v \Leftrightarrow y \in \Gamma_v^\infty \wedge \tau\,|v| - 2\,|y| \geq 0.$$

Let $v^\perp$ denote v rotated by 90 degrees, i. e. $v^\perp = (-v_2, v_1)$ for $v = (v_1, v_2)$, and define $\phi : \mathbb{R}^2 \to \mathbb{R}, y \mapsto y \cdot \frac{v^\perp}{|v|}$. Then it holds that $|\nabla\phi| = \left|v^\perp\right| / |v| = 1$ and the zero-level set of ϕ coincides with Γ_v^∞, i. e. $\{\phi = 0\} = \Gamma_v^\infty$. Let δ_0 denote the usual 1D Dirac delta distribution and $\delta_0(\phi)$ the pullback of δ_0 by ϕ (cf. [77, Theorem 6.1.2]). Then, by [77, Theorem 6.1.5] it holds that

$$\int_{\{\phi=0\}} h(y)\,\mathrm{d}S(y) = \int_{\{\phi=0\}} \frac{h(y)}{|\nabla\phi(y)|}\,\mathrm{d}S(y) = \int_{\mathbb{R}^2} h(y)\delta_0(\phi(y))\,\mathrm{d}y$$

for any function $h : \mathbb{R}^d \to \mathbb{R}$. Therefore,

$$\begin{aligned}&\frac{1}{\tau\,|v|}\int_{\Gamma_v^\infty} f(\tilde{x}-y)H(\tau\,|v|-2\,|y|)\,\mathrm{d}S(y)\\ &\quad= \frac{1}{\tau\,|v|}\int_{\mathbb{R}^d} f(\tilde{x}-y)H(\tau\,|v|-2\,|y|)\delta_0(\phi(y))\,\mathrm{d}y = \int_{\mathbb{R}^d} f(\tilde{x}-y)k_v(y)\,\mathrm{d}y,\end{aligned}$$

where we define

$$k_v(y) = \frac{1}{\tau\,|v|}\delta_0\left(y \cdot \frac{v^\perp}{|v|}\right) H(\tau\,|v|-2\,|y|). \tag{4.3}$$

Note that, by construction, k_v is a one dimensional filter kernel with filter width $\tau\,|v|$ in the direction of the motion trajectory Γ_v^∞. Combined with (4.2), this leads to

$$\frac{1}{\tau}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} f(\tilde{x}-sv)\,\mathrm{d}s = (f * k_v)(\tilde{x}). \tag{4.4}$$

Thus,

$$g_i(x) = (f * k_v)(x - t_iv). \tag{4.5}$$

Based on this motion blur model, the following, straightforward variational approach suggests itself to handle the deblurring in the basic setup where everything is moving

$$E[f] = \sum_{i=1}^{n} \int_{\Omega} ((f * k_v)(x - t_i v) - g_i(x))^2 \, \mathrm{d}x + \mu \int_{\Omega} |\nabla f(x)|^p, \tag{4.6}$$

where $p \geq 1$. For $p = 1$, this model is known as TV restoration (cf. [107, 75]) and we give a sample result in Figure 4.3.

Figure 4.3: Most basic motion deblurring: Original image f (left), an artificial input image generated by (4.5) (middle) and reconstruction obtained by minimizing the energy given in (4.6) (right).

Now that we know how to handle motion blur in the simplest case, let us move to the case we want to tackle with our model, a somewhat more complicated situation. Henceforth, we consider an object moving with speed $v \in \mathbb{R}^2$ in front of a static background $f_{\mathrm{bg}} : \Omega \to \mathbb{R}$. The object at time 0 is represented by an intensity function $f_{\mathrm{obj}} : \Omega_{\mathrm{obj}} \to \mathbb{R}$ defined on an object domain $\Omega_{\mathrm{obj}} \subset \Omega$. Based on these assumptions, the actual scene intensity function is given by

$$f(t, x) = f_{\mathrm{obj}}(x - tv)\chi_{\mathrm{obj}}(x - vt) + f_{\mathrm{bg}}(x)(1 - \chi_{\mathrm{obj}}(x - vt)), \tag{4.7}$$

where $\chi_{\mathrm{obj}} : \mathbb{R}^2 \to \mathbb{R}$ denotes the characteristic function of Ω_{obj}. Note that f_{obj} needs to be extended to $\mathbb{R}^2$ in order to evaluate f, but since f_{obj} is only evaluated multiplied by χ_{obj}, its values outside Ω_{obj} are irrelevant. At a point x close to the boundary of the object, the integration in time (4.1) of this scene intensity decomposes into a spatial integration of object intensities along the motion path for the sub-interval of the aperture interval where the object covers the background at position x, and a retrieval of the background intensity for the remaining opening time of the lens.

The usual approaches do not carefully model the observed intensities as the moving object occludes and uncovers the background, but just ignore the distinction close to the object boundary: Inside the object, the blur is applied to the scene function f, while the background remains untouched apart from the object, i. e.

$$g(x) = \begin{cases} (f(t, \cdot) * k_v) & x \in \Omega_{\mathrm{obj}} \\ f_{\mathrm{bg}} & \text{else} \end{cases} \tag{4.8}$$

(cf. the combination of equation (14) and equation (3) in [69]). Figure 4.4 shows a comparison between the (too) simple blur model (4.8) and the actually observed motion blur on a circular object, textured with back and white stripes, moving in horizontal direction in front of a similarly textured static background. While this comparison is already enough to question the validity of (4.8), Figure 4.5 clearly outlines the importance of a proper handling of the motion

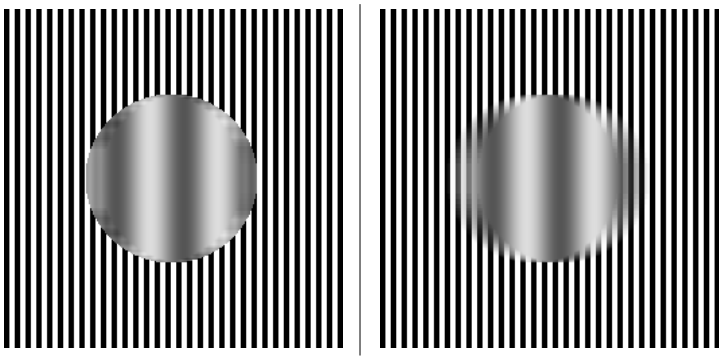


Figure 4.4: Comparison between the wrong motion blur model (4.8) which ignores the motion discontinuity at the boundary (left) and our realistic model (4.9) which carefully models the motion blur also in the vicinity of the boundary (right).

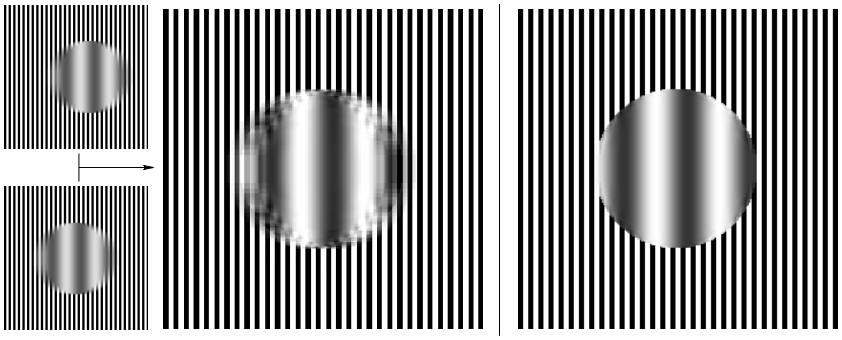


Figure 4.5: Given two frames with realistic motion blur (left), computational deblurring results based on the wrong motion blur model built into G_i (middle), and on our consistent model (right) are depicted.

discontinuity in the considered motion blur model. The proper blur handling is particularly important for the reliable recovery of boundaries of moving objects from recorded video frames g_i and subsequently for the proper restoration of image frames.

Now, inserting (4.7) in (4.1) and then using (4.4), we immediately deduce the correct formula for the theoretically observed motion blur at time t_i

$$\begin{aligned} G_i[\Omega_{\text{obj}}, v, f_{\text{obj}}, f_{\text{bg}}](x) := &((f_{\text{obj}}\chi_{\text{obj}}) * k_v)(x - t_i v) \\ &+ f_{\text{bg}}(x)(1 - (\chi_{\text{obj}} * k_v)(x - t_i v)), \end{aligned} \tag{4.9}$$

for given object domain Ω_{obj}, motion velocity v, and object and background intensity functions f_{obj} and f_{bg} respectively. Given the more precise motion blur model proposed here, we now proceed to derive a variational formulation to simultaneously estimate all parameters in this equation based on two consecutive frames.

4.3 A Mumford–Shah model

Given two frames g_1 and g_2 of a video sequence (showing a moving object in front of a static background) with motion blur recorded at times t_1 and t_2, respectively, we construct

a variational model to extract from these frames the object domain Ω_{obj}, the object image intensity f_{obj}, and the velocity v of the object. Here, we assume the background intensity f_{bg} to be known, as it can be relatively easily extracted from the video sequence in advance, e. g. by averaging pixels with stable values over a sequence of frames. The final formulation generalizes easily to include the estimation of the background intensity, though. Furthermore, the formulation can be extended to handle any number (≥ 2) of input frames. However, two frames are sufficient to obtain reasonable reconstruction results, hence we confine to two input frames. Having already pointed out the importance of a joint handling of the aforementioned degrees of freedom, of course we aim at formulating a joint energy to estimate them. At its core, the model is based upon the accurate motion blur model we developed in Section 4.2. Thus the energy includes the following fidelity term that measures the distance of the motion blurred frames G_i predicted by our forward model to the actually observed camera frames g_i in a least square sense, i. e.

$$\mathcal{F}[\Omega_{\text{obj}}, v, f_{\text{obj}}] := \sum_{i=1}^{2} \int_{\Omega} \left(G_i[\Omega_{\text{obj}}, v, f_{\text{obj}}, f_{\text{bg}}] - g_i\right)^2 \mathrm{d}x \,.$$

Obviously minimizing only this term is ill-posed, as the deconvolution therein by itself already is ill-posed (the motion blur increases the smoothness of any input image). Thus the need for additional regularity terms arises and our choice of the regularizers takes into account the following observations:

- Given v and f_{obj}, the problem of identifying the object Ω_{obj} is a task a piecewise constant Mumford–Shah model is well suited to handle, in particular since the unknown contour is significantly smeared out due to the motion blur. $\mathcal{F}[\cdot, v, f_{\text{obj}}]$ serves as the fidelity term, while we add the length of the boundary contour $\text{Per}(\Omega_{\text{obj}})$ as the corresponding prior from the piecewise constant Mumford–Shah model, cf. (1.6).

- Given v and Ω_{obj}, estimating f_{obj} by minimizing $\mathcal{F}[\Omega_{\text{obj}}, v, \cdot]$ is an almost classical deblurring problem for f_{obj}. Thus we can accompany the fidelity term by a typical deconvolution prior. Since f_{obj} is likely to have edges in motion blurred videos (cf. Figure 4.2, 4.8, and 4.10), the total variation functional $|Df_{\text{obj}}|(\Omega)$ [109] comes to mind. This functional has already been successfully applied to deconvolution problems (cf. [128, 47]), hence we chose it as prior for the deconvolution. Because the TV functional is considered on Ω and not only on Ω_{obj}, it also guarantees a suitable extension of f_{obj} onto the whole space. Furthermore, because the prior does not depend on Ω_{obj}, it does not adversely affect the estimation of Ω_{obj}.

- Finally, given Ω_{obj} and f_{obj}, the estimation of the velocity v is primarily an optical flow problem. Since the velocity is assumed to be constant inside the object, i. e. v is just a vector, not a vector-valued function, no regularizing prior is necessary for v. The transport of the object intensity f_{obj} from time t_1 to t_2 described in $\mathcal{F}[\cdot, \cdot, f_{\text{obj}}]$ provides us with information on v. In the case of limited intensity modulations on the moving object, i. e. if f_{obj} has little to no textural information, it is the comparison of the expected transition profile $\chi_{\text{obj}} * k_v$, encoded in G_i, with the observed profile in g_i that will act as a guidance for the estimation of v.

Thus we obtain the energy

$$\begin{aligned} E_{\mathrm{OD}}[\Omega_{\mathrm{obj}}, v, f_{\mathrm{obj}}] = &\sum_{i=1}^{2} \int_{\Omega} \left(G_i[\Omega_{\mathrm{obj}}, v, f_{\mathrm{obj}}, f_{\mathrm{bg}}] - g_i\right)^2 \mathrm{d}x \\ &+ \mu \left|Df_{\mathrm{obj}}\right|(\Omega) + \nu \operatorname{Per}(\Omega_{\mathrm{obj}}), \end{aligned} \tag{4.10}$$

and ask for a minimizer with respect to the unknowns Ω_{obj}, v, and f_{obj}. Once a minimizer is known, we can retrieve the deblurred images $f(t_1, \cdot)$ and $f(t_2, \cdot)$ simply by applying (4.7).

4.4 The minimization algorithm

To find minimizers of the energy (4.10), we follow the approach proposed by Chan and Vese [46] to handle the unknown object domain Ω_{obj}, i. e. it is represented by the zero super level set $\{\phi > 0\}$ of a level set function as discussed in Section 1.2.2. Analogously to Section 1.2.2, we replace χ_{obj} by $H_\delta(\phi)$ and thus get a regularized G_i^δ, representing the expected motion blur at time t_i:

$$\begin{aligned} G_i^\delta[\phi, v, f_{\mathrm{obj}}, f_{\mathrm{bg}}](x) = &\left((f_{\mathrm{obj}} H_\delta(\phi)) * k_v\right)(x - \tau_i v) \\ &+ f_{\mathrm{bg}}(x)\left(1 - (H_\delta(\phi) * k_v)(x - \tau_i v)\right). \end{aligned} \tag{4.11}$$

Finally, following our derivation of (1.12), we obtain an approximate global energy consisting of the fidelity term $\mathcal{F}^\delta$ and the regularizing prior $\mathcal{J}^{\delta,\varrho}$

$$\begin{aligned} E_{\mathrm{OD}}^{\delta,\varrho}[\phi, v, f_{\mathrm{obj}}] &= \mathcal{F}^\delta[\phi, v, f_{\mathrm{obj}}] + \mathcal{J}^{\delta,\varrho}[\phi, f_{\mathrm{obj}}] \\ &:= \sum_{i=1}^{2} \int_{\Omega} \left(G_i^\delta[\phi, v, f_{\mathrm{obj}}, f_{\mathrm{bg}}] - g_i\right)^2 \mathrm{d}x \\ &\quad + \int_{\Omega} \mu \left|\nabla f_{\mathrm{obj}}\right|_\varrho + \nu \left|\nabla H_\delta(\phi)\right|_\varrho \mathrm{d}x \ . \end{aligned} \tag{4.12}$$

This expression depends on the motion vector $v \in \mathbb{R}^2$ and two scalar, unknown functions, namely the level set description ϕ of the object domain Ω_{obj} and the object intensity f_{obj}.

Before we can start discussing our gradient flow based minimization algorithm, we first need to derive the variations of $E_{\mathrm{OD}}^{\delta,\varrho}$ with respect to the unknowns ϕ, v and f_{obj}. To shorten notation, we introduce the residual term

$$r_i(x) := 2\left[G_i^\delta[\phi, v, f_{\mathrm{obj}}, f_{\mathrm{bg}}](x) - g_i(x)\right].$$

First, we consider the derivative with respect to v_1 and v_2, the two components of the velocity vector v. Using (4.11) and

$$\partial_a(F(a,a)) = \nabla F(a,a) \cdot (1,1) = \partial_1 F(a,a) + \partial_2 F(a,a),$$

we get

$$\begin{aligned} &\partial_{v_j} G_i^\delta[\phi, v, f_{\mathrm{obj}}, f_{\mathrm{bg}}](x) \\ &= \partial_{v_j}\left[\left((f_{\mathrm{obj}} H_\delta(\phi)) * k_v\right)(x - \tau_i v)\right] - f_{\mathrm{bg}}(x)\partial_{v_j}\left[(H_\delta(\phi) * k_v)(x - \tau_i v)\right] \\ &= \left\{\partial_{v_j}\left[(f_{\mathrm{obj}} H_\delta(\phi)) * k_v\right]\right\}(x - \tau_i v) - \tau_i \left\{\partial_{x_j}\left[(f_{\mathrm{obj}} H_\delta(\phi)) * k_v\right]\right\}(x - \tau_i v) \\ &\quad - f_{\mathrm{bg}}(x)\left(\left\{\partial_{v_j}\left[H_\delta(\phi) * k_v\right]\right\}(x - \tau_i v) - \tau_i \left\{\partial_{x_j}\left[H_\delta(\phi) * k_v\right]\right\}(x - \tau_i v)\right) \end{aligned} \tag{4.13}$$

Now, instead of directly deriving k_v with respect to v, we take a different approach here, summarized in the following lemma:

4.4.1 Lemma. *For $f : \mathbb{R}^2 \to \mathbb{R}$ (sufficiently smooth) and the motion blur kernel k_v (4.3), it holds that*

$$\left\{\partial_{v_j}\left[f * k_v\right]\right\}(x) = (\partial_{x_j} f * h_v)(x),$$

where

$$h_v(y) = -\frac{y \cdot v}{|v|^2} k_v(y).$$

Proof Let

$$M_v f(x) := \frac{1}{T}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} f(x - sv)\,\mathrm{d}s\,.$$

From (4.4) we know $M_v f(x) = (f * k_v)(x)$ and thus

$$\begin{aligned}\left\{\partial_{v_j}\left[f * k_v\right]\right\}(x) = \partial_{v_j}\left(M_v f(x)\right) &= \partial_{v_j}\left(\frac{1}{\tau}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} f(x - sv)\,\mathrm{d}s\right)\\ &= -\frac{1}{\tau}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} s\partial_{x_j} f(x - sv)\,\mathrm{d}s =: N_v \partial_{x_j} f(x),\end{aligned}$$

because of

$$\begin{aligned}\partial_{v_j}(f(x - sv)) &= \nabla f(x - sv) \cdot \partial_{v_j}(x_1 - sv_1, x_2 - sv_2) = \nabla f(x - sv) \cdot (-s\delta_{1j}, -s\delta_{2j})\\ &= -s\partial_{x_j} f(x - sv).\end{aligned}$$

With the same curve c and Γ_v used in Section 4.2 ($c(s) = sv$), we get

$$\begin{aligned}N_v \partial_{x_j} f(x) &= -\frac{1}{\tau}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} s\partial_{x_j} f(x - sv)\,\mathrm{d}s = -\frac{1}{\tau\,|v|}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \frac{c(s) \cdot v}{|v|^2}\partial_{x_j} f(x - c(s))\,|\dot{c}|\,\mathrm{d}s\\ &= -\frac{1}{\tau\,|v|^3}\int_{\Gamma_v} y \cdot v\partial_{x_j} f(x - y)\,\mathrm{d}S(y).\end{aligned}$$

Analogously to (4.4), we deduce

$$N_v \partial_{x_j} f(x) = (\partial_{x_j} f * h_v)(x),$$

where

$$h_v(y) = -\frac{y \cdot v}{\tau\,|v|^3}\delta_0\left(y \cdot \frac{v^\perp}{|v|}\right) H(\tau\,|v| - 2\,|y|) = -\frac{y \cdot v}{|v|^2} k_v(y),$$

concluding the proof of the lemma. □

Using the just established lemma and (4.13), it holds that

$$\begin{aligned}&\partial_{v_j} G_i^\delta[\phi, v, f_{\mathrm{obj}}, f_{\mathrm{bg}}](x)\\ &= \left[\partial_{x_j}(f_{\mathrm{obj}} H_\delta(\phi)) * h_v\right](x - \tau_i v) - \tau_i\left\{\partial_{x_j}\left[(f_{\mathrm{obj}} H_\delta(\phi)) * k_v\right]\right\}(x - \tau_i v)\\ &\quad - f_{\mathrm{bg}}(x)\left(\left[\partial_{x_j} H_\delta(\phi) * h_v\right](x - \tau_i v) - \tau_i\left\{\partial_{x_j}\left[H_\delta(\phi) * k_v\right]\right\}(x - \tau_i v)\right)\\ &= \left[\partial_{x_j}(f_{\mathrm{obj}} H_\delta(\phi)) * (h_v - \tau_i k_v)\right](x - \tau_i v)\\ &\quad - f_{\mathrm{bg}}(x)\left[\partial_{x_j} H_\delta(\phi) * (h_v - \tau_i k_v)\right](x - \tau_i v)\end{aligned}$$

and we finally get the derivative of $\mathcal{F}^\delta$ with respect to v_j:

$$\partial_{v_j}\mathcal{F}^\delta[\phi, v, f_{\mathrm{obj}}] = \sum_{i=1}^{2}\int_\Omega \Big[\big[\partial_{x_j}(f_{\mathrm{obj}}H_\delta(\phi)) * (h_v - \tau_i k_v)\big](x-\tau_i v) \\ - f_{\mathrm{bg}}(x)\big[\partial_{x_j}H_\delta(\phi) * (h_v - \tau_i k_v)\big](x-\tau_i v)\Big] r_i(x)\,\mathrm{d}x\,.$$

Let us remark that by using Lemma 4.4.1 we do not need to regularize the block filter function k_v to be able to calculate the variation of $\mathcal{F}^\delta$ with respect to v. At the same time, this approach leads to significantly more stable numerical results compared to differentiating a regularized version of k_v directly.

The first variations of the prior functional $\mathcal{J}^{\delta,\varrho}$ follow directly from (1.13)

$$\Big\langle \partial_{f_{\mathrm{obj}}}\mathcal{J}^{\delta,\varrho}[\phi, f_{\mathrm{obj}}], \vartheta\Big\rangle = -\mu\int_\Omega \operatorname{div}\left(\frac{\nabla f_{\mathrm{obj}}}{|\nabla f_{\mathrm{obj}}|_\varrho}\right)\vartheta\,\mathrm{d}x,$$

$$\Big\langle \partial_{\phi}\mathcal{J}^{\delta,\varrho}[\phi, f_{\mathrm{obj}}], \psi\Big\rangle = -\nu\int_\Omega \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|_\varrho}\right)H'_\delta(\phi)\psi\,\mathrm{d}x\,.$$

Making use of the chain rule and the linearity of the convolution, the first variations of the fidelity functional $\mathcal{F}^\delta$ are

$$\Big\langle \partial_{f_{\mathrm{obj}}}\mathcal{F}^\delta[\phi, v, f_{\mathrm{obj}}], \vartheta\Big\rangle = \sum_{i=1}^{2}\int_\Omega r_i(x)((\vartheta H_\delta(\phi)) * k_v)(x-\tau_i v)\,\mathrm{d}x,$$

$$\Big\langle \partial_{\phi}\mathcal{F}^\delta[\phi, v, f_{\mathrm{obj}}], \psi\Big\rangle = \sum_{i=1}^{2}\int_\Omega r_i(x)\Big[((f_{\mathrm{obj}}H'_\delta(\phi)\psi) * k_v)(x-\tau_i v) \\ - f_{\mathrm{bg}}(x)((H'_\delta(\phi)\psi) * k_v)(x-\tau_i v)\Big]\,\mathrm{d}x\,.$$

Looking closely at the formulation of the variations of $\mathcal{F}^\delta$, one notes that the test functions appearing are convolved with the kernel k_v. We need to take care of this first, before we can move from here to the strong formulation of the variations. The tool we need is supplied by the next lemma:

4.4.2 Lemma (Convolution by parts)**.** *Given $f, g, h : \mathbb{R}^d \to \mathbb{R}$ (integrable) and $a, b \in \mathbb{R}^d$, it holds that*

$$\int_\Omega f(x+a)(g*h)(x+b)\,\mathrm{d}x = \int_\Omega (f^{a,+} * h^{b,-})(y)g(y)\,\mathrm{d}y, \tag{4.14}$$

where $q^{b,\pm}(x) := q(\pm x + b)$. For the special case $a = b = 0$ and $g^-(x) := g(-x)$, we have

$$\int_\Omega f(x)(g*h)(x)\,\mathrm{d}x = \int_\Omega (f * g^-)(y)h(y)\,\mathrm{d}y\,.$$

Proof The statement is proven by using the definition of the convolution and the following calculation:

$$\begin{aligned}\int_\Omega f(x+a)(g*h)(x+b)\,\mathrm{d}x &= \int_\Omega f(x+a)\left(\int_\Omega g(x+b-y)h(y)\,\mathrm{d}y\right)\mathrm{d}x\\ &= \int_\Omega \left(\int_\Omega f(x+a)g(x+b-y)\,\mathrm{d}x\right)h(y)\,\mathrm{d}y\\ &= \int_\Omega \left(\int_\Omega f^{a,+}(x)g^{b,-}(y-x)\,\mathrm{d}x\right)h(y)\,\mathrm{d}y\\ &= \int_\Omega (f^{a,+}*g^{b,-})(y)h(y)\,\mathrm{d}y\,.\end{aligned}$$

□

With the aforementioned lemma, we remove the convolution from the test functions and end up with

$$\left\langle \partial_{f_{\mathrm{obj}}}\mathcal{F}^\delta[\phi,v,f_{\mathrm{obj}}],\vartheta\right\rangle = \sum_{i=1}^{2}\int_\Omega (r_i * k_v^{-\tau_i v,-})(x)(\vartheta H_\delta(\phi))(x)\,\mathrm{d}x,$$

$$\begin{aligned}\left\langle \partial_{\phi}\mathcal{F}^\delta[\phi,v,f_{\mathrm{obj}}],\psi\right\rangle = \sum_{i=1}^{2}\int_\Omega &(r_i * k_v^{-\tau_i v,-})(x)(f_{\mathrm{obj}}H'_\delta(\phi)\psi)(x)\\ &- ((r_i f_{\mathrm{bg}}) * k_v^{-\tau_i v,-})(x)(H'_\delta(\phi)\psi)(x)\,\mathrm{d}x\,.\end{aligned}$$

Spatial discretization

Now, we take into account discrete intensities for a given video frame resolution of $n \times m$ pixels. We combine this with a finite difference approximation of the energy and denote by Φ and F_{obj} the vectors containing the nodal values Φ_k and $(F_{\mathrm{obj}})_k$, respectively, at the nodes x_k for $k = 1,\dots,nm$, i. e. $\Phi, F_{\mathrm{obj}} \in \mathbb{R}^{nm}$.

Using the already established continuous first variations of the fidelity functional $\mathcal{F}^\delta$ and the prior functional $\mathcal{J}^{\delta,\varrho}$, choosing test functions concentrated at nodes and evaluated for the spatially discretized energy and denoting the spatially discrete blurring residual by

$$R_i(x) := 2\left[G_i^\delta[\Phi,v,F_{\mathrm{obj}},F_{\mathrm{bg}}](x) - g_i(x)\right],$$

we finally obtain

$$\begin{aligned}\partial_{(F_{\mathrm{obj}})_k}\mathcal{F}^\delta[\Phi,v,F_{\mathrm{obj}}] &= \sum_{i=1}^{2}(R_i * k_v^{-\tau_i v,-})(x_k)H_\delta(\Phi(x_k)),\\ \partial_{\Phi_k}\mathcal{F}^\delta[\Phi,v,F_{\mathrm{obj}}] &= \sum_{i=1}^{2}(R_i * k_v^{(-\tau_i v),-})(x_k)(F_{\mathrm{obj}}H'_\delta(\Phi))(x_k)\\ &\qquad - ((R_i F_{\mathrm{bg}}) * k_v^{(-\tau_i v),-})(x_k)H'_\delta(\Phi(x_k)),\\ \partial_{(F_{\mathrm{obj}})_k}\mathcal{J}^{\delta,\varrho}[\Phi,F_{\mathrm{obj}}] &= -\mu\,\mathrm{div}\left(\frac{\nabla F_{\mathrm{obj}}(x_k)}{|\nabla F_{\mathrm{obj}}(x_k)|_\varrho}\right),\end{aligned}$$

$$\partial_{\Phi_k} \mathcal{J}^{\delta,\varrho}[\Phi, F_{\text{obj}}] = -\nu H'_\delta(\Phi(x_k)) \operatorname{div}\left(\frac{\nabla\Phi(x_k)}{|\nabla\Phi(x_k)|_\varrho}\right),$$

$$\begin{aligned}\partial_{v_j} \mathcal{F}^\delta[\Phi, v, F_{\text{obj}}] = \sum_{i=1}^{2} \int_\Omega \Big[\big[\partial_{x_j}(F_{\text{obj}} H_\delta(\Phi)) * (h_v - \tau_i k_v)\big](x - \tau_i v) \\ - f_{\text{bg}}(x) \big[\partial_{x_j} H_\delta(\Phi) * (h_v - \tau_i k_v)\big](x - \tau_i v)\Big] r_i(x)\,\mathrm{d}x\,.\end{aligned}$$

These form the entries of the Euclidean gradients $\operatorname{grad}_\Phi E_{\text{OD}}^{\delta,\varrho}$, $\operatorname{grad}_v E_{\text{OD}}^{\delta,\varrho}$, and $\operatorname{grad}_{F_{\text{obj}}} E_{\text{OD}}^{\delta,\varrho}$ required for the gradient descent algorithm. To numerically evaluate the derivatives appearing above, we use standard difference quotients. For the integral, we use a standard quadrature rule.

In what follows, we will outline an energy relaxation method in this already spatially discrete setting based on an operator splitting with step size control (cf. Section 6.3) and a regularized gradient descent (cf. Section 6.2) with respect to the level set function.

Initialization

At first, given an initial contour (usually a sufficiently large circle that contains part of the object and part of the background), we select and fix (in a very rough approximation step) F_{obj} as the intensity values of one of the images g_1 and g_2. Then, we relax the functional

$$E_{\text{MC}}^{\delta,\varrho}[\Phi, v] = \sum_{i=1}^{2} \int_\Omega (\tilde{G}_i^\delta[\Phi, v, F_{\text{obj}}, F_{\text{bg}}] - g_i)^2 + \nu \int_\Omega |\nabla H_\delta(\Phi)|_\varrho \,\mathrm{d}x,$$

where $\tilde{G}_i^\delta$ is obtained from G_i^δ by skipping the motion blur convolution, i. e.

$$\tilde{G}_i^\delta[\Phi, v, F_{\text{obj}}, F_{\text{bg}}](x) = (F_{\text{obj}} H_\delta(\Phi))(x - \tau_i v) + F_{\text{bg}}(x)(1 - H_\delta(\Phi)(x - \tau_i v)).$$

This initializing step can be regarded as a "motion competition approach" (as in the level set formulation of [52]), and we obtain an initial contour Φ^0 and an initial estimate v^0 for the motion velocity. Note that in this sense, our model can be seen as an extension of the motion competition approach. Then, fixing Φ^0 and v^0, a standard deblurring based on (4.4) is performed on g_1 and g_2 to obtain an initial estimate for F_{obj}^0. In total, this gives us initial estimates for all unknowns, i. e. Φ^0, v^0 and F_{obj}^0.

Gradient descent

In initial numerical experiments with the proposed model, we experienced a significantly different *roughness* (difference of gradient directions) of the energy landscape with respect to the different unknowns Φ, v, and F_{obj}. Hence, an operator splitting strategy which separates these directions and incorporates different time steps for all of them suggests itself to account for these observations. Thus, in any subsequent descent step we pick up the newly computed quantities from the same iteration in a Gauss-Seidel like manner. The descent in the level set description Φ of the object domain Ω_{obj} requires special treatment, while a standard gradient descent can be used to update v and F_{obj}. As step size control for the gradient descents, we employ the Armijo rule (cf. 6.3), evaluated separately for all three components.

A point-wise evaluation of the variation with respect to Φ possibly shows non-smoothness and concentration of the gradient, cf. Figure 4.6. Hence, we incorporate a regularized gradient

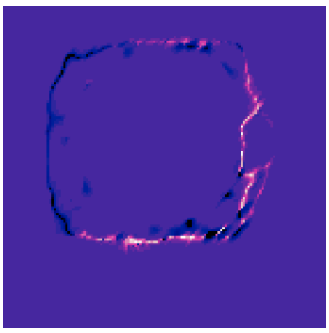


Figure 4.6: Color coded pointwise gradient $\operatorname{grad}_\Phi E_{\mathrm{OD}}^{\delta,\varrho}$ for one iteration from Figure 4.8.

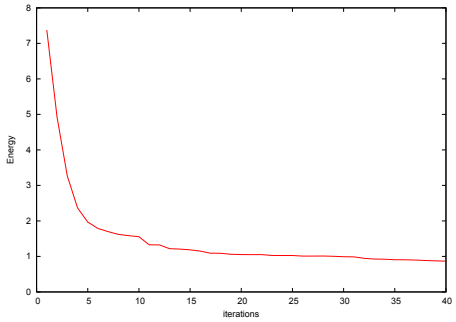


Figure 4.7: Plot of the energy decay in the descent algorithm for the experiment from Figure 4.8.

descent in the level set function inspired by the Sobolev active contour approach [120]. As outlined in Section 6.2, this approach turns out to result to a Gaussian filtering of the descent direction with a filter $\mathbf{G}_\sigma$ (cf. (6.4)). In the applications of this chapter, we use $\sigma = 0.005$ as filter width. Let us again emphasize that the resulting regularized descent does not affect the energy landscape itself, but solely the descent path towards the set of minimizers. Figure 4.7 shows a plot of the energy decay for the application in Figure 4.8.

We stop the descent algorithm once the changes in all three unknown vectors computed in the last time step and measured in the Euclidean norm are smaller than a given tolerance parameter ς. For the experiments presented here, we used $\varsigma = 0.01$. Finally, we have the following algorithm (in pseudocode notation):

Algorithm 4.1: Minimization of $E_{\mathrm{OD}}^{\delta,\varrho}$

given input frames g_1 and g_2;
initialize Φ^0, v^0 by "motion competition" (using g_1 or g_2 as F_{obj});
initialize F_{obj}^0 by standard deconvolution based on Φ^0, v^0;
initialize k with 0;
repeat
  $d^\Phi = \operatorname{grad}_\Phi E_{\mathrm{OD}}^{\delta,\varrho}[\Phi^k, v^k, F_{\mathrm{obj}}^k] * \mathbf{G}_\sigma$;
  $\Phi^{k+1} = \Phi^k - \mathrm{ArmijoStepSize}[E_{\mathrm{OD}}^{\delta,\varrho}, \Phi^k, d^\Phi] d^\Phi$;
  $d^v = \operatorname{grad}_v E_{\mathrm{OD}}^{\delta,\varrho}[\Phi^{k+1}, v^k, F_{\mathrm{obj}}^k]$;
  $v^{k+1} = v^k - \mathrm{ArmijoStepSize}[E_{\mathrm{OD}}^{\delta,\varrho}, v^k, d^v] d^v$;
  $d^F = \operatorname{grad}_{F_{\mathrm{obj}}} E_{\mathrm{OD}}^{\delta,\varrho}[\Phi^{k+1}, v^{k+1}, F_{\mathrm{obj}}^k]$;
  $F_{\mathrm{obj}}^{k+1} = F_{\mathrm{obj}}^k - \mathrm{ArmijoStepSize}[E_{\mathrm{OD}}^{\delta,\varrho}, F_{\mathrm{obj}}^k, d^F] d^F$;
  $k \leftarrow k + 1$;
until $\left(\left|\Phi^{k+1} - \Phi^k\right|, \left|v^{k+1} - v^k\right|, \left|F_{obj}^{k+1} - F_{obj}^k\right| \leq \varsigma\right)$;

4.5 Results

In this section, we investigate the performance of the proposed model based on results obtained for different applications. Figures 4.8 and 4.10 show numerical results for two different artificial test cases. In both, we see the proper identification of the moving object, estimation of the

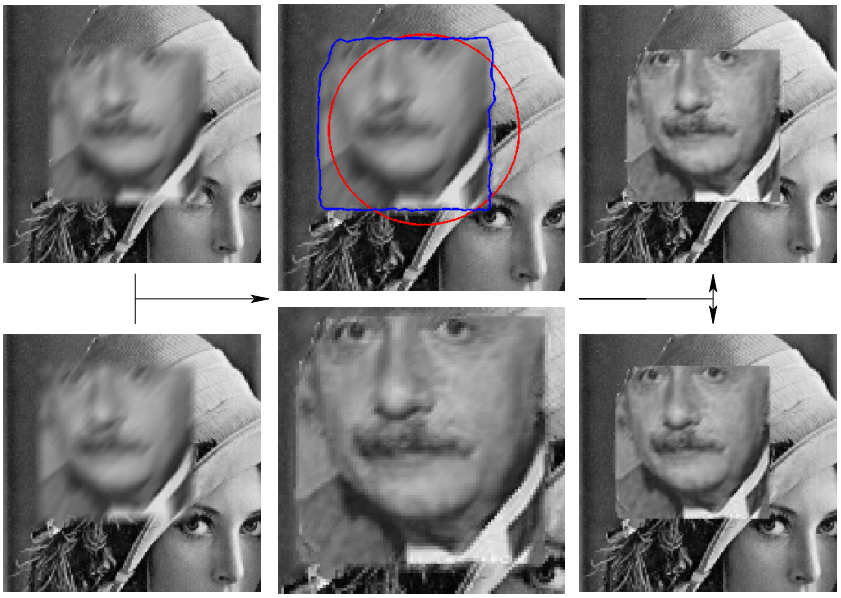


Figure 4.8: Results on an artificial image sequence showing the input images g_1 and g_2 (left), the recovered object intensity f_{obj} (middle bottom), the initial (red) and the computed (blue) boundary contour of the object (middle top), as well as the recovered frames f_1 and f_2 (right).

Figure 4.9: Intermediate results from the run of our algorithm on Figure 4.8 showing the object contour for three iterations from the motion competition initialization phase (top row) and three follow–up iterations from the full joint model (bottom row).

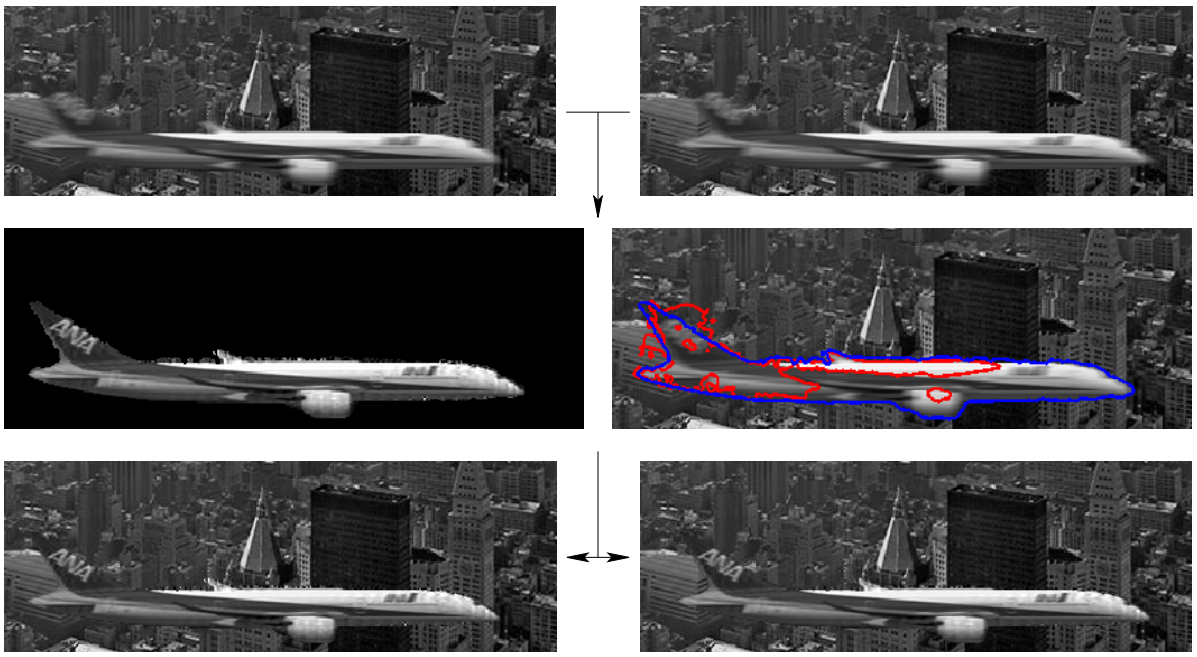
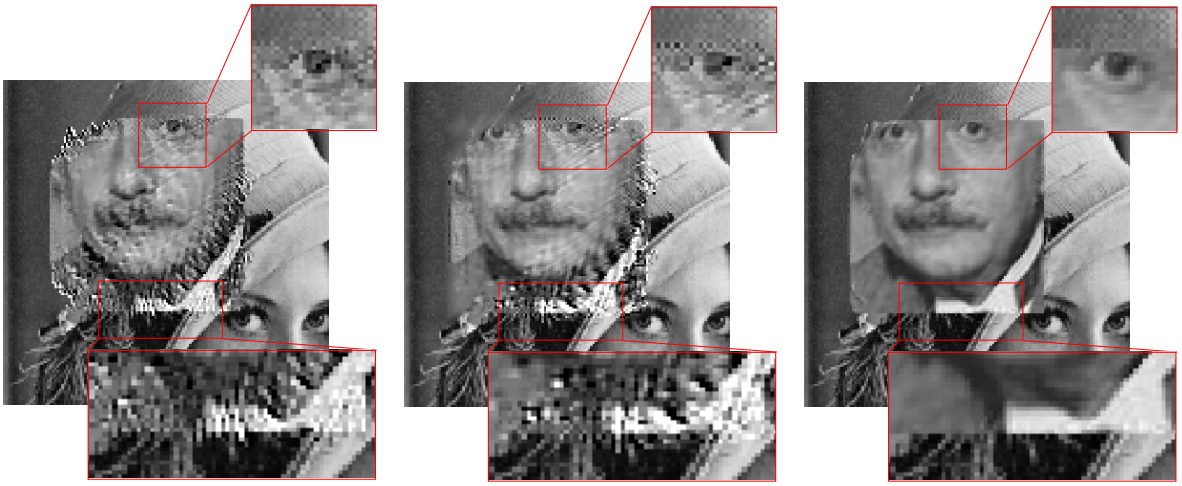


Figure 4.10: Results on another artificial image sequence showing the input images g_1 and g_2 (top), the recovered object intensity f_{obj} (middle left), an intermediate (red) and the final computed (blue) boundary contour of the object (middle right), as well as the recovered frames f_1 and f_2 (bottom).

motion velocity and deblurring of the object intensity. Although the setting in Figure 4.8 is artificial, it still is challenging and suitable to demonstrate the advantage of joint estimation of all unknowns. In this figure, a moving square (velocity vector $v = (6, 7)$) textured with an image of Einstein (f_{obj}) is considered in front of a static background textured with the Lena image (f_{bg}). The independently computed velocity (i. e. obtained using motion competition) from the blurred frames leads to an inaccurate estimate of $v = (5.78, 6.80)$ and of the moving object, which in turn results in a non-satisfactory restoration of the blurred frames (first image in the second row of Figure 4.9, see also Figure 4.11). With the proposed joint technique, on the other hand, the estimated velocity is $v = (5.98, 7.009)$ and both frames (last row of Figure 4.8) and the moving region (blue curve in the middle row of Figure 4.8) are accurately recovered. In Figure 4.10 we consider another artificial but more realistic example. Here, the object is an airplane moving with velocity vector $v = (10, 0)$. With the proposed method, we obtain $v = (9.47, -0.007)$ as estimated velocity of the plane and successfully remove the blur from it. Especially note that the letters on the vertical stabilizer of the plane become clearly readable because of the deblurring, whereas they are not readable at all in the blurred input frames.

In Figure 4.11, it becomes apparent that the joint approach for all three unknowns (motion velocity v, object intensity f_{obj} and the object domain Ω_{obj}) is crucial for a proper reconstruction of blurred video frames. In this figure, our joint approach is compared with a two step method which first tries to identify Ω_{obj} and v based on a motion competition algorithm [52], followed by a deblurring in a second step using the results from the first step to demonstrate the interdependence of the unknowns. The two step method is used in two flavors, once the deblurring is done with the non-consistent motion blur model (4.8) and once with the consistent



inaccurate blur, non-joint accurate blur, non-joint accurate blur, fully joint

Figure 4.11: A comparison of our joint method (right) with a non-joint method (middle) and a non-joint method that uses the non-consistent motion blur model (4.8) instead of the consistent one (4.9). The non-joint methods estimate the object contour and its velocity in advance with pure motion competition.

one (4.9). Thus, the results shown in this figure also outline the importance of the consistent motion blur model for a proper reconstruction in the vicinity of motion singularities.

Finally, we applied our model to a real word video sequence exhibiting motion blur and recorded with a hand-held, off-the-shelf video camera. The recorded sequence shows a small toy car moving in front of a puzzle (static background), cf. Figure 4.1. We have chosen a textured object moving in front of a textured background to demonstrate the interplay between the deblurring steered by the fidelity functional $\mathcal{F}^{\delta}$ and the reconstruction of sharp edges due to the total variation built into the prior $\mathcal{J}^{\delta,\varrho}$. The input data and the results for this application are shown in Figure 4.2, whereas Figure 4.12 displays a zoom onto the moving object, demonstrating the interplay of the deblurring and the edge reconstruction.

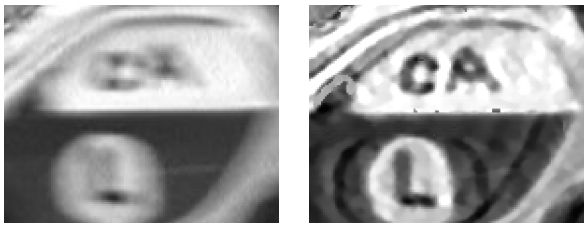


Figure 4.12: A zoom in of the moving car from Figure 4.2 for one of the original blurred frames (left) and of the corresponding restored frame (right) computed by our proposed model.

4.6 Outlook

Although the presented framework is generic, we particularly addressed single moving objects and static background. Handling multiple objects can be simply done by having multiple unknown regions $\Omega_{\text{obj},k}$ in the general formulation we introduced (cf. the approach by Vese and Chan [125] for the segmentation in more than two regions, described in Section 1.2.3). More elegantly, and thereby also permitting dynamic background, one could consider formulations that replace the assumption (4.7) by something along the lines of

$$f(t,\cdot)\chi(\cdot) = f(t+\tau,\cdot)\chi(\cdot - \tau v).$$

This constraint means that the function moves with the object and eliminates the need for having independent unknowns f_{obj} and f_{bg}.

5 Binary image segmentation by unconstrained thresholding

AFTER extensively using the Mumford–Shah functional in the preceding chapters to tackle concrete image processing problems, we approach the binary segmentation functional (1.6) itself and show how to efficiently find its minimizers in this chapter. So far we simply relied on the Chan–Vese model (cf. Section 1.2.2) to do the minimization process and even though it is suitable for the tasks handled in the previous chapters, it is not completely unproblematic and leaves room for fundamental improvements. The most striking issue of the Chan–Vese functional is the fact that it has local, non-global minima due to its non-convexity. This is not only a theoretical problem, since the commonly used numerical minimization techniques (like the gradient flow approach we used so far) can get stuck in local minima that differ considerably from a global minimum, hence possibly producing useless results. While the multiscale approach (build on the scale parameter of the regularizing metric in the gradient flow scheme) we introduced in Section 3.2.4 allows to avoid most of the problems associated to local minima, it still does not guarantee to find a global minimizer. Therefore, the goal of this chapter is to introduce a method to obtain a global minimizer of the Mumford–Shah functional for two-phase segmentation (1.6). Like the Nikolova–Esedoḡlu–Chan model (cf. Section 1.2.4), our model is based on a convex reformulation of the original problem. Unlike their model, ours does not need to impose any constraint in the convex formulation, neither explicitly nor implicitly by an additional, artificial penalty term [20]. Moreover, we extend our model to multiphase segmentation by the ideas of Vese and Chan (cf. Section 1.2.3) in a canonical way, albeit losing the global optimality property.

5.1 Related work

The problem of minimizing the Mumford–Shah segmentation functional has been extensively studied in the last decade leading to a wide range of existing methods, each with its own shortcomings. One of the first numerical feasible methods to obtain (local) minimizers of the functional was proposed by Chan and Vese [46], already discussed in Section 1.2.2 and used in Chapters 3 and 4.

Shen [116] developed a Γ-convergence formulation along with a simple implementation by the iterated integration of a linear Poisson equation. The unknown set is represented in a diffuse way by a phase field.

In [65], Esedoḡlu and Tsai tackle the minimization problem based on the threshold dynamics of Merriman, Bence and Osher [96] for evolving an interface by its mean curvature. Here the minimization is achieved by alternating solving a linear parabolic partial differential equation and simple thresholding.

Alvino and Yezzi [3] approximate Mumford–Shah segmentation using reduced image bases. According to them, the majority of the robustness of Mumford–Shah segmentation can be obtained without allowing each pixel to vary independently. Their approximative model has

comparable performance to Mumford–Shah segmentations where each pixel is allowed to vary freely.

El-Zehiry et al. [63] take a fully discrete approach: Picking up the ideas of Chan and Vese, they present a discrete graph-representable level set energy function and minimize it using graph cuts. This allows to find global minimizers of the discrete graph representation.

A way to obtain global minimizers was introduced by Nikolova, Esedoḡlu and Chan [102] (cf. Section 1.2.4). This method is closely related to the method we propose in this chapter, the key difference is that the Nikolova–Esedoḡlu–Chan model requires a constraint in the convex minimization while the model we propose here does not involve any constraint in the convex formulation [20].

On the other hand there are methods to solve a certain class of minimal surface problems by unconstrained convex optimization, cf. the work of Chambolle and Darbon [41, 44]. The two-phase Mumford–Shah functional belongs to this class, yet due to the best of our knowledge nobody else seems to have tapped the potential offered by these general insights for Mumford–Shah based image segmentation so far.

5.2 Constrained global two-phase minimization

Like in Section 1.2.2, we again consider the prototype Mumford–Shah energy (1.6), i. e.

$$E_{\text{MS-2}}[\Sigma] := \int_\Sigma f_1 \,\mathrm{d}x + \int_{\Omega\setminus\Sigma} f_2 \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma),$$

for given indicator functions $f_1, f_2 \in L^1(\Omega)$ such that $f_1, f_2 \geq 0$ a. e. in Ω.

5.2.1 Remark. Because of

$$E_{\text{MS-2}}[\Sigma] = \underbrace{\int_\Sigma (f_1 - f_2) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma)}_{=:\hat{E}_{\text{MS-2}}[\Sigma]} + \int_\Omega f_2 \,\mathrm{d}x,$$

$E_{\text{MS-2}}$ and $\hat{E}_{\text{MS-2}}$ share the same minimizers.

5.2.2 Remark. For $h(x) := e^{-|x|^2}$, we have

$$\begin{aligned} E_{\text{MS-2}}[\Sigma] &= \int_\Sigma f_1 \,\mathrm{d}x + \int_{\Omega\setminus\Sigma} f_2 \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma) + \int_\Omega h \,\mathrm{d}x - \int_\Omega h \,\mathrm{d}x \\ &= \int_\Sigma (f_1 + h) \,\mathrm{d}x + \int_{\Omega\setminus\Sigma} (f_2 + h) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma) - \underbrace{\int_\Omega h \,\mathrm{d}x}_{=C<\infty}, \end{aligned}$$

i. e. replacing f_1 and f_2 by $f_1 + h$ and $f_2 + h$ does not affect the minimizers of $E_{\text{MS-2}}$. This, combined with $f_1, f_2 \geq 0$ a. e., implies that we can assume $f_1, f_2 > 0$ a. e. in Ω without loss of generality.

Now, let us pick up the work of Nikolova et al. [102] we started to discuss in Section 1.2.4. At first, we remedy the last bit of "uncertainty" left in Theorem 1.2.3 by showing that it holds not only for almost every, but for every $c \in [0, 1)$:

5.2.3 Proposition. *Theorem 1.2.3 holds for all $c \in [0, 1)$.*

Proof. The proof of the statement's extension to all $c \in [0,1)$ is inspired by the proof of [2, Lemma 4 (iii)]: Let u be a minimizer of E_{NEC} under the constraint $0 \leq u \leq 1$ and denote its super level sets by Σ_c. Choose an arbitrary but fixed $\hat{c} \in [0,1)$. Because of Theorem 1.2.3, the statement holds for a. e. $c \in [0,1]$, so by Remark 5.2.1, there exists a sequence $(c_n) \in [0,1]^{\mathbb{N}}$ with $c_n \downarrow \hat{c}$ such that

$$\Sigma_{c_n} \in \operatorname*{argmin}_{\Sigma \subset \Omega} \hat{E}_{\text{MS-2}}[\Sigma].$$

Since the super level sets of a function are contained in each other, we have

$$\chi_{\Sigma_{c_n}} = \chi_{\bigcup_{k=1}^{n} \Sigma_{c_k}} \to \chi_{\Sigma^{\cup}} \text{ pointwise a. e.},$$

where $\Sigma^{\cup} := \bigcup_{n=1}^{\infty} \Sigma_{c_n}$. Setting $g := f_1 - f_2$ and using Lebesgue's dominated convergence theorem, we obtain

$$\int_{\Sigma^{\cup}} g \,\mathrm{d}x = \int_{\Omega} g\chi_{\Sigma^{\cup}} \,\mathrm{d}x = \lim_{n\to\infty} \int_{\Omega} g\chi_{\Sigma_{c_n}} \,\mathrm{d}x = \lim_{n\to\infty} \int_{\Sigma_{c_n}} g \,\mathrm{d}x\,.$$

Here we used $\left|g\chi_{\Sigma_{c_n}}\right| \leq |g| \leq |f_1| + |f_2|$ to provide the integrable upper bound. For each n and $\Sigma \subset \Omega$, due to the minimizing property of Σ_{c_n}, we have

$$\int_{\Sigma_{c_n}} g \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_{c_n}) \leq \int_{\Sigma} g \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma).$$

Using the continuity argument from above and the lower semicontinuity of the perimiter (cf. [5]), we get

$$\int_{\Sigma^{\cup}} g \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma^{\cup}) \leq \int_{\Sigma} g \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma),$$

i. e. $\Sigma^{\cup}$ is a minimizer of $E_{\text{MS-2}}$. Combining this with

$$\Sigma_c = \{x \in \Omega : u(x) > c\} = \bigcup_{n=1}^{\infty} \{x \in \Omega : u(x) > c_n\} = \bigcup_{n=1}^{\infty} \Sigma_{c_n} \tag{5.1}$$

concludes the proof. □

5.2.4 Remark. For any function u that fulfills the constraint, obviously $\{u > 1\} = \emptyset$. Therefore we cannot expect Theorem 1.2.3 to hold for $c = 1$.

As already mentioned in Section 1.2.4, Nikolova et al. showed that the constraint of their model can be handled by an exact penalty term (cf. Proposition 1.2.4).

While this result already gives a method to find global minimizers of $E_{\text{MS-2}}$ by solving a convex, unconstrained minimization problem, there is still room for improvement. Most numerical minimizations methods rely on the gradient of the functional, but the proposed penalty term is not differentiable, making a regularization necessary. But any smooth regularization of the penalty term will stop the minimizers of the convex, constrained functional from coinciding with those of the convex functional with penalty term. The stronger the regularization, the more the minimizers deviate.

Furthermore, the regularization imposes numerical difficulties. If an explicit gradient descent is used for the minimization (as proposed in [102]), a suitable step size control is needed to ensure

convergence. The step sizes allowed by such methods, e. g. the Armijo rule (cf. Section 6.3), typically correspond to the size of the region in which the linearization of the functional properly approximates the functional. Due to the nature of the penalty term p, the linearization at 0 and 1 of a regularized version of it only approximates the regularization properly in a region that is of the size of the regularization parameter. So, as soon as the current iterate of the gradient descent takes values near 0 or 1, the step size control only allows step sizes of the order of the regularization parameter, which, as mentioned above, cannot be chosen too big.

Instead of using a penalty term one could of course also approach the constrained convex optimization problem directly. This is done for example by Bresson et al. [30]. Their approach does not need a penalty term and gives an efficient algorithm to minimize E_{NEC}, but has to introduce an additional unknown v and a regularization parameter θ and needs to minimize for u and v alternatingly. Furthermore, the key idea to apply Chambolle's TV minimization algorithm [42] can also be directly applied to our model to obtain a simpler and faster minimization algorithm (cf. Section 5.3.1): There is no need to introduce v, θ and the alternating minimization. Therefore it is worth to investigate whether it is possible to simplify the problem by getting rid of the constraint altogether.

5.3 Unconstrained global two-phase minimization

Another alternative to the Chan–Vese model is a phase field approach [65, 116] with a typical double well term (in contrast to the single well term of the Ambrosio–Tortorelli model, cf. Section 1.2.1):

$$E^{\epsilon}_{\mathrm{PH}}[u] := \int_{\Omega} u^2 f_1 + (1-u)^2 f_2 + \nu \left(\frac{1}{\epsilon} u^2 (1-u)^2 + \epsilon \left| \nabla u \right|^2 \right) \mathrm{d}x \,. \tag{5.2}$$

A minimizer u_ϵ of this energy is a diffuse representation of the segmentation, i. e. $\{u_\epsilon \approx 0\}$ and $\{u_\epsilon \approx 1\}$ represent the two segments respectively with a smooth transition in between. $E^{\epsilon}_{\mathrm{PH}}[u]$ is known to Γ-converge to $E_{\text{MS-2}}$ [116], but unfortunately not convex and does not permit jumps in u for $\epsilon > 0$.

Knowing both E_{NEC} and $E^{\epsilon}_{\mathrm{PH}}$, the question arises whether it is possible to combine the advantages of both models while eliminating some of the disadvantages. Heuristically looking at both energies served as motivation to investigate the following energy:

$$E_{\mathrm{UC}}[u] := \int_{\Omega} u^2 f_1 + (1-u)^2 f_2 \, \mathrm{d}x + \nu \left| Du \right| (\Omega). \tag{5.3}$$

This energy is convex because it does not involve the non-convex double well term of $E^{\epsilon}_{\mathrm{PH}}$, and can be minimized without imposing constraints because it does not have the indicator term from E_{NEC} that is not bounded from below. Furthermore, it permits jumps in u.

5.3.1 Remark. Given a function u, obviously we have

$$E_{\mathrm{UC}}[\min\{\max\{0, u\}, 1\}] \leq E_{\mathrm{UC}}[u].$$

Therefore, a minimizer $u_{\min}$ fulfills $0 \leq u_{\min} \leq 1$.

While the proposed functional has some nice obvious properties, it is not obvious whether there is a relation between its minimizer and minimizers of $E_{\text{MS-2}}$. Before we tackle this question, let us remark a link between E_{NEC} and E_{UC}:

5.3.2 Remark. There is a direct relationship between E_{NEC} and E_{UC}: Using

$$\begin{aligned}
&u^2 f_1 + (1-u)^2 f_2 - (u f_1 + (1-u) f_2) \\
&= u^2 f_1 + f_2 - 2u f_2 + u^2 f_2 - u f_1 - f_2 + u f_2 \\
&= u^2 (f_1 + f_2) - u(f_1 + f_2) \\
&= (u^2 - u + \tfrac{1}{4})(f_1 + f_2) - \tfrac{1}{4}(f_1 + f_2) \\
&= (u - \tfrac{1}{2})^2 (f_1 + f_2) - \tfrac{1}{4}(f_1 + f_2),
\end{aligned}$$

we get

$$\begin{aligned}
u^2 f_1 + (1-u)^2 f_2 &= (u f_1 + (1-u) f_2) + (u - \tfrac{1}{2})^2 (f_1 + f_2) - \tfrac{1}{4}(f_1 + f_2) \\
&= (f_1 - f_2)\, u + (u - \tfrac{1}{2})^2 (f_1 + f_2) - \tfrac{1}{4}(f_1 + f_2) + f_2.
\end{aligned}$$

Therefore

$$\begin{aligned}
E_{\mathrm{UC}}[u] &= \int_\Omega u^2 f_1 + (1-u)^2 f_2 \,\mathrm{d}x + \nu \,|Du|\,(\Omega) \\
&= \int_\Omega (f_1 - f_2)\, u + (u - \tfrac{1}{2})^2 (f_1 + f_2) - \tfrac{1}{4}(f_1 + f_2) + f_2 \,\mathrm{d}x + \nu \,|Du|\,(\Omega) \\
&= E_{\mathrm{NEC}}[u] + \int_\Omega (f_1 + f_2)(u - \tfrac{1}{2})^2 \,\mathrm{d}x + C,
\end{aligned}$$

where

$$C = \int_\Omega f_2 - \tfrac{1}{4}(f_1 + f_2) \,\mathrm{d}x\,.$$

In other words, E_{UC} essentially equals E_{NEC} plus an additional quadratic penalty energy. The constant C is clearly irrelevant for the minimizers.

To investigate the relation between the minimizers of E_{UC} and minimizers of $E_{\text{MS-2}}$ we can make use of the theory derived in the context of the connection between minimal surface problems and total variation minimization.

The following general statement has been made by Chambolle [43], Chambolle and Darbon [44], in the continuous setting, its discrete counterpart is well-known.

5.3.3 Theorem. *Let $\Psi : \Omega \times \mathbb{R} \to \mathbb{R}, (x, s) \mapsto \Psi(x, s)$ such that $\Psi(x, \cdot)$ is C^1 and uniformly convex for all $x \in \Omega$ and*

$$u := \operatorname*{argmin}_{\tilde{u}} \int_\Omega \Psi(x, \tilde{u}(x)) \,\mathrm{d}x + \nu \,|D\tilde{u}|\,(\Omega).$$

Then $\Sigma_c := \{u > c\}$ for all $c \in \mathbb{R}$ is a minimizer of

$$\int_\Sigma \partial_s \Psi(x, c) \,\mathrm{d}x + \nu \,\mathrm{Per}(\Sigma).$$

Note that this general statement cannot be directly applied to the model of Nikolova et al. discussed in Section 1.2.4 because the integrand is neither uniformly (not even strictly) convex nor does the general statement incorporate the constraint.

As remarked in [44], the proof for a more specific statement given in [41] still applies to Theorem 5.3.3.

5.3.4 Theorem. *If u is a minimizer of (5.3), then $\Sigma := \{u > \frac{1}{2}\}$ minimizes*

$$E_{\text{MS-2}}[\Sigma] = \int_\Sigma f_1 \,\mathrm{d}x + \int_{\Omega\setminus\Sigma} f_2 \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma).$$

Proof. Let $\Psi(x,s) := s^2 f_1(x) + (1-s)^2 f_2(x)$. Obviously $\Psi(x,\cdot)$ is C^2 for all $x \in \Omega$ and we have

$$\partial_s \Psi(x,s) = 2s f_1(x) + 2(s-1) f_2(x)$$

and

$$\partial_s^2 \Psi(x,s) = 2(f_1(x) + f_2(x)).$$

From Remark 5.2.2, we know that $f_1, f_2 > 0$ a. e., therefore $\Psi(x,\cdot)$ is uniformly convex for a. e. $x \in \Omega$. Now just apply Theorem 5.3.3, noting $\partial_s \Psi(x, \frac{1}{2}) = f_1(x) - f_2(x)$ and Remark 5.2.1. □

In this sense, our theorem is a corollary of Theorem 5.3.3. Nevertheless, we will give an alternative proof for our theorem here that more clearly outlines the link of our model to the one of Nikolova et al..

Before we start with the proof, we first establish a maximum principle for the binary Mumford–Shah functional that we will need for the proof.

5.3.5 Proposition. *Let $g_1, g_2 \in L^1(\Omega)$ with $g_1 > g_2$ a. e. in Ω and*

$$\Sigma_i \in \operatorname*{argmin}_{\tilde\Sigma} E_{g_i}[\tilde\Sigma] \text{ for } i = 1,2, \text{ where } E_{g_i}[\Sigma] = \int_\Sigma g_i(x) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma).$$

Then $|\Sigma_1 \setminus \Sigma_2| = 0$, i. e. $\Sigma_1 \subset \Sigma_2$ up to a negligible set.

Proof. The statement was already made in [39, Lemma 3.2], for its proof we closely follow the proof of [2, Lemma 4 (i)]: Since Σ_i is a minimizer of E_{g_i}, we have

$$\int_{\Sigma_1} g_1(x) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_1) \le \int_{\Sigma_1\cap\Sigma_2} g_1(x) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_1 \cap \Sigma_2),$$
$$\int_{\Sigma_2} g_2(x) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_2) \le \int_{\Sigma_1\cup\Sigma_2} g_2(x) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_1 \cup \Sigma_2).$$

Adding both inequalities gives

$$\begin{aligned}&\int_{\Sigma_1} g_1(x) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_1) + \int_{\Sigma_2} g_2(x) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_2)\\ &\le \int_{\Sigma_1\cap\Sigma_2} g_1(x) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_1 \cap \Sigma_2) + \int_{\Sigma_1\cup\Sigma_2} g_2(x) \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma_1 \cup \Sigma_2).\end{aligned}$$

Noting that for all sets $A, B \subset \mathbb{R}^n$ with finite perimeter (cf. [5, Proposition 3.38 (d)])

$$\operatorname{Per}(A \cup B) + \operatorname{Per}(A \cap B) \le \operatorname{Per}(A) + \operatorname{Per}(B)$$

holds, we obtain

$$\begin{aligned}&\int_{\Sigma_1} g_1(x) \,\mathrm{d}x + \int_{\Sigma_2} g_2(x) \,\mathrm{d}x \le \int_{\Sigma_1\cap\Sigma_2} g_1(x) \,\mathrm{d}x + \int_{\Sigma_1\cup\Sigma_2} g_2(x) \,\mathrm{d}x\\ \Rightarrow &\int_{\Sigma_1} g_1(x) \,\mathrm{d}x - \int_{\Sigma_1\cap\Sigma_2} g_1(x) \,\mathrm{d}x \le \int_{\Sigma_1\cup\Sigma_2} g_2(x) \,\mathrm{d}x - \int_{\Sigma_2} g_2(x) \,\mathrm{d}x\\ \Rightarrow &\int_{\Sigma_1\setminus\Sigma_2} g_1(x) \,\mathrm{d}x \le \int_{\Sigma_1\setminus\Sigma_2} g_2(x) \,\mathrm{d}x\,.\end{aligned}$$

Because of $g_1 > g_2$ a. e., we can conclude $|\Sigma_1 \setminus \Sigma_2| = 0$. □

5.3.6 Theorem. *For given indicator functions $f_1, f_2 \in L^1(\Omega)$ such that $f_1, f_2 \geq 0$ a. e., let*

$$u := \operatorname*{argmin}_{\tilde{u}} \int_\Omega \tilde{u}^2 f_1 + (1-\tilde{u})^2 f_2 \,\mathrm{d}x + \nu \left|D\tilde{u}\right|(\Omega) = \operatorname*{argmin}_{\tilde{u}} E_{\mathrm{UC}}[\tilde{u}]$$

and $\Sigma_c := \{u > c\}$. Then Σ_c is a minimizer of the Mumford–Shah energy

$$E_{\text{MS-2}}[\Sigma, c] := 2\left\{\int_\Sigma (c f_1(x) - (1-c) f_2(x)) \,\mathrm{d}x\right\} + \nu \operatorname{Per}(\Sigma)$$

for a. e. $c \in [0,1]$.

Proof. Let $u \in BV(\Omega)$ with $0 \leq u \leq 1$ a. e. and denote its super level sets by Σ_c. Then, for any $0 \leq c \leq 1$ and a. e. $x \in \Omega$, we have

$$u^2(x) > c \Leftrightarrow u(x) > \sqrt{c} \Leftrightarrow x \in \Sigma_{\sqrt{c}}$$

and

$$(1-u(x))^2 \geq c \Leftrightarrow u(x) \leq 1 - \sqrt{c} \Leftrightarrow x \in (\Omega \setminus \Sigma_{1-\sqrt{c}}).$$

Hence, it holds that

$$\chi_{[0,u^2(x))}(c) = \chi_{\Sigma_{\sqrt{c}}}(x) \text{ and } \chi_{[0,(1-u(x))^2]}(c) = \chi_{\Omega\setminus\Sigma_{1-\sqrt{c}}}(x).$$

Using this we obtain

$$\begin{aligned}
\int_\Omega f_1(x) u^2(x) \,\mathrm{d}x &= \int_\Omega f_1(x) \int_0^1 \chi_{[0,u^2(x))}(c) \,\mathrm{d}c \,\mathrm{d}x = \int_0^1 \int_\Omega f_1(x) \chi_{[0,u^2(x))}(c) \,\mathrm{d}x \,\mathrm{d}c \\
&= \int_0^1 \int_{\Sigma_{\sqrt{c}}} f_1(x) \,\mathrm{d}x \,\mathrm{d}c \overset{\phi(c)=\sqrt{c}}{=} \int_0^1 2\phi(c)\phi'(c) \int_{\Sigma_{\phi(c)}} f_1(x) \,\mathrm{d}x \,\mathrm{d}c \\
&= \int_0^1 2c \int_{\Sigma_c} f_1(x) \,\mathrm{d}x \,\mathrm{d}c
\end{aligned}$$

and

$$\begin{aligned}
\int_\Omega f_2(x)(1-u(x))^2 \,\mathrm{d}x &= \int_0^1 \int_{\Omega\setminus\Sigma_{1-\sqrt{c}}} f_2(x) \,\mathrm{d}x \,\mathrm{d}c \\
&= \int_0^1 2(\phi(c)-1)\phi'(c) \int_{\Omega\setminus\Sigma_{\phi(c)}} f_2(x) \,\mathrm{d}x \,\mathrm{d}c \quad \big(\text{where } \phi(c) := 1-\sqrt{c}\big) \\
&= \int_1^0 2(c-1) \int_{\Omega\setminus\Sigma_c} f_2(x) \,\mathrm{d}x \,\mathrm{d}c = \int_0^1 2(1-c) \int_{\Omega\setminus\Sigma_c} f_2(x) \,\mathrm{d}x \,\mathrm{d}c \\
&= \int_0^1 2(1-c) \int_\Omega f_2(x) \,\mathrm{d}x \,\mathrm{d}c - \int_0^1 2(1-c) \int_{\Sigma_c} f_2(x) \,\mathrm{d}x \,\mathrm{d}c \\
&= C - \int_0^1 2(1-c) \int_{\Sigma_c} f_2(x) \,\mathrm{d}x \,\mathrm{d}c,
\end{aligned}$$

where C is a constant that does not depend on u. Combined with (1.9), this leads to

$$\begin{aligned} E_{\mathrm{UC}}[u] &= \int_0^1 \left[2 \left\{ \int_{\Sigma_c} (cf_1(x) - (1-c)f_2(x)) \,\mathrm{d}x \right\} + \nu \operatorname{Per}(\Sigma_c) \right] \mathrm{d}c + C \\ &= \int_0^1 E_{\text{MS-2}}[\Sigma_c, c] \,\mathrm{d}c + C. \end{aligned}$$

E_{UC} is convex and bounded from below, therefore a minimizer $u_{\min}$ exists and by Remark 5.3.1 $0 \leq u_{\min} \leq 1$ holds. From Remark 5.2.2 and the assumption $f_1, f_2 \geq 0$ a. e., we know that we can assume $f_1, f_2 > 0$ a. e. in Ω without loss of generality. From

$$g_c(x) := 2\left(cf_1(x) - (1-c)f_2(x)\right) = 2c(f_1(x) + f_2(x)) - 2f_2(x)$$

and $f_1, f_2 > 0$ a. e., we get

$$g_{c_1} = 2c_1(f_1 + f_2) - 2f_2 > 2c_2(f_1 + f_2) - 2f_2 = g_{c_2} \text{ a. e. for } c_1 > c_2. \tag{5.4}$$

For $c \in [0,1]$ let

$$A_c \in \operatorname*{argmin}_{\Sigma \subset \Omega} E_{\text{MS-2}}[\Sigma, c]. \tag{5.5}$$

Note that we already discussed the existence of such minimizers in the proof of Theorem 1.2.3. Using (5.4), Proposition 5.3.5 shows that

$$A_{c_1} \subset A_{c_2} \text{ up to a negligible set for } c_1 > c_2.$$

Therefore, $\hat{u}(x) := \sup\{\tilde{c} : x \in A_{\tilde{c}}\}$ defines a function $\hat{u}$ such that

$$\{\hat{u} > c\} = A_c \text{ up to a negligible set for all } c \in [0,1].$$

From

$$\begin{aligned} \int_{-\infty}^{\infty} \operatorname{Per}(\{\hat{u} > c\}) \,\mathrm{d}c = \int_0^1 \operatorname{Per}(\{\hat{u} > c\}) \,\mathrm{d}c = \int_0^1 \operatorname{Per}(A_c) \,\mathrm{d}c \leq \frac{1}{\nu} \int_0^1 E_{\text{MS-2}}[A_c, c] \,\mathrm{d}c \\ \leq \frac{1}{\nu} \int_0^1 E_{\text{MS-2}}[\{u_{\min} > c\}, c] \,\mathrm{d}c = \frac{1}{\nu} \left(E_{\mathrm{UC}}[u_{\min}] - C \right) < \infty \end{aligned}$$

we can deduce $\hat{u} \in BV(\Omega)$ with the coarea formula (cf. [67, Section 5.5, Theorem 1 (iii)]).

Finally, $\hat{u}$ allows to deduce that $\Sigma_c[u_{\min}]$ is a minimizer of $E_{\text{MS-2}}[\cdot, c]$ for a. e. $c \in [0,1]$. Otherwise $E_{\mathrm{UC}}[\hat{u}] < E_{\mathrm{UC}}[u_{\min}]$ would hold because of (5.5) and

$$E_{\mathrm{UC}}[\hat{u}] = \int_0^1 E_{\text{MS-2}}[A_c, c] \,\mathrm{d}c + C,$$

a contradiction to the fact that $u_{\min}$ minimizes E_{UC}. □

To extend Theorem 5.3.6 to all $c \in [0,1]$, we need another preparatory lemma:

5.3.7 Lemma. *Given*

$$A_{c_n} \in \operatorname*{argmin}_{\Sigma\subset\Omega} E_{\text{MS-2}}[\Sigma, c_n],$$

where $(c_n) \in [0,1]^{\mathbb{N}}$ *with* $c_n \downarrow c \in [0,1]$. *Then*

$$A^{\cup} := \bigcup_{n=1}^{\infty} A_{c_n} \in \operatorname*{argmin}_{\Sigma\subset\Omega} E_{\text{MS-2}}[\Sigma, c].$$

If $c_n \uparrow c \in [0,1]$, *then* $A^{\cap} := \bigcap_{n=1}^{\infty} A_{c_n}$ *is a minimizer of* $E_{\text{MS-2}}[\cdot, c]$.

Proof. This lemma is a modification of [2, Lemma 4 (iii)] and can be proven by a similar line of reasoning. Furthermore, the proof is similar to the proof of Proposition 5.2.3. The main difference to that proof is that there a sequence of sets that are all minimizers of the same energy is used, while we have here a sequence of sets where each set minimizes a slightly different energy.

For $k < n$, we have $c_n < c_k$. Combined with (5.4) and Proposition 5.3.5, this leads to

$$A_{c_k} \subset A_{c_n} \text{ up to a negligible set for } k < n.$$

Therefore, $\chi_{A_{c_n}} = \chi_{\bigcup_{k=1}^{n} A_{c_k}}$ a. e. and thus $\chi_{A_{c_n}} \to \chi_{A^{\cup}}$ pointwise a. e.. Using this and the fact that $g_{c_n} \to g_c$ pointwise a. e. and we get

$$\int_{A^{\cup}} g_c \,\mathrm{d}x = \int_{\Omega} g_c \chi_{A^{\cup}} \,\mathrm{d}x = \lim_{n\to\infty} \int_{\Omega} g_{c_n} \chi_{A_{c_n}} \,\mathrm{d}x = \lim_{n\to\infty} \int_{A_{c_n}} g_{c_n} \,\mathrm{d}x$$

and

$$\int_{\Sigma} g_c \,\mathrm{d}x = \lim_{n\to\infty} \int_{\Sigma} g_{c_n} \,\mathrm{d}x$$

by Lebesgue's dominated convergence theorem. Here, we used

$$\left|g_{c_n}\chi_{A_{c_n}}\right| \le |g_{c_n}| \le 2c_n\,|f_1| + 2(1-c_n)\,|f_2| \le 2\,(|f_1| + |f_2|)$$

to provide the integrable upper bound for both limits. For each n and $\Sigma \subset \Omega$, due to the minimizing property of A_{c_n}, we have

$$\int_{A_{c_n}} g_{c_n} \,\mathrm{d}x + \nu \operatorname{Per}(A_{c_n}) \le \int_{\Sigma} g_{c_n} \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma).$$

Using the continuity arguments from above and the lower semicontinuity of the perimiter, we get

$$\int_{A^{\cup}} g_c \,\mathrm{d}x + \nu \operatorname{Per}(A^{\cup}) \le \int_{\Sigma} g_c \,\mathrm{d}x + \nu \operatorname{Per}(\Sigma),$$

i. e. $A^{\cup}$ is a minimizer of $E_{\text{MS-2}}[\cdot, c]$. The second statement can be deduced analogously. □

5.3.8 Proposition. *Theorem 5.3.6 holds not only for a. e. c, but for all* $c \in [0,1]$.

Proof. Because $\Sigma_1 = \emptyset$ (up to a negligible set) and $g_1 = 2f_1 \geq 0$ a. e., the statement is true for $c = 1$. Now let $c \in [0, 1)$. Because of Theorem 5.3.6 there exists a sequence $(c_n) \in [0, 1]^{\mathbb{N}}$ with $c_n \downarrow c$ such that

$$\Sigma_{c_n} \in \operatorname*{argmin}_{\Sigma \subset \Omega} E_{\text{MS-2}}[\Sigma, c_n].$$

From Lemma 5.3.7 we know that $\bigcup \Sigma_{c_n}$ is a minimizer of $E_{\text{MS-2}}[\cdot, c]$. Combined with (5.1), this concludes the proof. □

The preceding proposition (or Theorem 5.3.4) finally tells us how to find a global minimizer of $E_{\text{MS-2}}[\cdot]$ given in (1.6): Minimize the convex energy (5.3) and threshold the minimizer to $\frac{1}{2}$. In case of the piecewise constant Mumford–Shah functional for two-phase segmentation, we obtain a global minimizer of the Mumford–Shah energy (1.5) with respect to Σ for fixed gray values c_1, c_2. We do not necessarily find a global minimizer with respect to Σ, c_1 and c_2.

Another link between E_{NEC} and E^{ϵ}_{PH} is the so-called *piecewise constant level set method* [92] for two-phase segmentation that constrains the level set function to only take two distinct values. If this constraint is approximated with a penalty energy, the method equals the phase field approach. If the constraint is relaxed to a certain boundedness constraint, the method equals [102]. In both cases the fidelity term has to be altered accordingly, making use of the fact that this term is the same in E_{NEC} and E^{ϵ}_{PH} if u only takes the values 0 and 1.

Since (5.3) is similar to the Rudin–Osher–Fatemi (ROF) energy [109], there is a wide variety of established minimization schemes to choose from, ranging from a straightforward gradient descent scheme with a differentiable approximation of the BV term over primal thresholding methods [55] to sophisticated methods based on the dual formulation of the BV norm, e. g. [42, 44]. In Section 5.3.1 we show how to use the minimization scheme from [42] in our context.

With $\Psi(x, s) = \frac{1}{2}(s - (f_2(x) - f_1(x)))^2$, another immediate consequence of Theorem 5.3.3 is that the zero superlevelset of a minimizer of the ROF energy

$$E_{\text{ROF}}[u] := \int_\Omega \frac{1}{2}(u - (f_2 - f_1))^2 \,\mathrm{d}x + \nu\,|Du|\,(\Omega) \tag{5.6}$$

is a global minimizer of $\hat{E}_{\text{MS-2}}$ and therefore by Remark 5.2.1 also of $E_{\text{MS-2}}$. This is another way to obtain a global minimizer of $E_{\text{MS-2}}$ by unconstrained convex optimization, but compared to (5.3) this method has a few shortcomings, cf. Sections 5.3.2 and 5.3.3. Furthermore, the boundedness mentioned in Remark 5.3.1 does not hold for minimizers of the ROF energy. Perhaps this is one of the reasons why, to the best of our knowledge, nobody seems to have used the classical ROF function for Mumford–Shah based image segmentation so far.

5.3.1 Minimization using a dual formulation

Due to the similarity of our energy to the Rudin–Osher–Fatemi energy, we can adapt Chambolle's algorithm for total variation minimization [42] to find minimizers of our energy. The basic idea of the algorithm is to derive a dual formulation of the minimization problem that is easier to solve numerically than the initial minimization problem. This dual formulation is summarized in the following proposition:

5.3.9 Proposition. *A minimizer of* (5.3) *is given by*

$$u = \frac{2f_2 - \nu \operatorname{div} p}{2(f_1 + f_2)} \tag{5.7}$$

where p is a solution of

$$\nabla\left(\frac{\nu\operatorname{div}p - 2f_2}{2(f_1+f_2)}\right) - \left|\nabla\left(\frac{\nu\operatorname{div}p - 2f_2}{2(f_1+f_2)}\right)\right| p = 0. \tag{5.8}$$

Proof. The statement is structurally the same as [30, Proposition 1], allowing us to employ the same proof technique. Using the dual formulation of the total variation (1.8), we get

$$\min_u E_{\mathrm{UC}}[u] = \min_u \sup_{|p|\le 1}\left(\int_\Omega u^2 f_1 + (1-u)^2 f_2 + \nu u\operatorname{div}p\,\mathrm{d}x\right).$$

Using [62, Chapter IV, Proposition 2.3], a result from minimax theory, we may interchange the optimization with respect to u and p. Hence, the minimization of E_{UC} is equivalent to

$$\sup_{|p|\le 1}\inf_u\left(\int_\Omega u^2 f_1 + (1-u)^2 f_2 + \nu u\operatorname{div}p\,\mathrm{d}x\right).$$

The inner minimization problem with respect to u is a pointwise problem and can be solved explicitly. The corresponding optimality condition (zero crossing of the integrand's derivative with respect to u) is

$$0 = 2uf_1 + 2(u-1)f_2 + \nu\operatorname{div}p = 2u(f_1+f_2) - 2f_2 + \nu\operatorname{div}p$$

Solving for u gives (5.7). Because the second derivative of the integrand is $2(f_1+f_2)$, which is bigger than zero a. e. according to Remark 5.2.2, the obtained u is a minimizer. Hence, the infimum exists, i. e. "inf" turns to "min", and we can directly solve the minimization with respect to u by inserting the minimizer given by (5.7):

$$\begin{aligned}
&\min_u\left(\int_\Omega u^2 f_1 + (1-u)^2 f_2 + \nu u\operatorname{div}p\,\mathrm{d}x\right)\\
&= \int_\Omega \left(\frac{2f_2 - \nu\operatorname{div}p}{2(f_1+f_2)}\right)^2 f_1 + \left(1 - \frac{2f_2 - \nu\operatorname{div}p}{2(f_1+f_2)}\right)^2 f_2 + \nu\frac{2f_2-\nu\operatorname{div}p}{2(f_1+f_2)}\operatorname{div}p\,\mathrm{d}x\\
&= \int_\Omega \left(\frac{2f_2 - \nu\operatorname{div}p}{2(f_1+f_2)}\right)^2 f_1 + \left(\frac{2f_1 + \nu\operatorname{div}p}{2(f_1+f_2)}\right)^2 f_2 + \nu\frac{2f_2-\nu\operatorname{div}p}{2(f_1+f_2)}\operatorname{div}p\,\mathrm{d}x\\
&= \int_\Omega \frac{4f_2^2 - 4f_2\nu\operatorname{div}p + \nu^2(\operatorname{div}p)^2}{4(f_1+f_2)^2} f_1 + \frac{4f_1^2 + 4f_1\nu\operatorname{div}p + \nu^2(\operatorname{div}p)^2}{4(f_1+f_2)^2} f_2\\
&\qquad + \nu\frac{2f_2-\nu\operatorname{div}p}{2(f_1+f_2)}\operatorname{div}p\,\mathrm{d}x\\
&= \int_\Omega \frac{4f_1f_2^2 + 4f_1^2f_2}{4(f_1+f_2)^2} + \frac{f_1\nu^2(\operatorname{div}p)^2 + f_2\nu^2(\operatorname{div}p)^2}{4(f_1+f_2)^2} + \nu\frac{2f_2-\nu\operatorname{div}p}{2(f_1+f_2)}\operatorname{div}p\,\mathrm{d}x\\
&= \int_\Omega \frac{f_1f_2(f_2+f_1)}{(f_1+f_2)^2} + \frac{(f_1+f_2)\nu^2(\operatorname{div}p)^2}{4(f_1+f_2)^2} + \nu\frac{f_2\operatorname{div}p}{(f_1+f_2)} - \frac{\nu^2(\operatorname{div}p)^2}{2(f_1+f_2)}\,\mathrm{d}x\\
&= \int_\Omega \frac{f_1f_2}{(f_1+f_2)} - \frac{\nu^2(\operatorname{div}p)^2}{4(f_1+f_2)} + \nu\frac{f_2\operatorname{div}p}{(f_1+f_2)}\,\mathrm{d}x =: F[p]
\end{aligned}$$

For any $\zeta \in C_0^\infty(\Omega,\mathbb{R}^d)$, we get

$$\begin{aligned}
\big\langle F'[p],\zeta\big\rangle &= \int_\Omega -\frac{2\nu^2\operatorname{div}p\operatorname{div}\zeta}{4(f_1+f_2)} + \frac{\nu f_2\operatorname{div}\zeta}{(f_1+f_2)}\,\mathrm{d}x = \int_\Omega \frac{-\nu^2\operatorname{div}p + 2\nu f_2}{2(f_1+f_2)}\operatorname{div}\zeta\,\mathrm{d}x\\
&= \int_\Omega \nabla\left(\frac{\nu^2\operatorname{div}p - 2\nu f_2}{2(f_1+f_2)}\right)\zeta\,\mathrm{d}x
\end{aligned}$$

Then, by the general Kuhn-Tucker theorem for Banach spaces [95, Theorem 3.2], the optimality conditions for p under the pointwise constraint $\frac{1}{2}|p|^2 \leq \frac{1}{2}$ (equivalent to $|p| \leq 1$) are

$$\nabla\left(\frac{\nu^2 \operatorname{div} p - 2\nu f_2}{2(f_1+f_2)}\right) - \lambda p = 0 \tag{5.9}$$

$$\lambda \geq 0 \tag{5.10}$$

$$\lambda(|p|^2 - 1) = 0 \tag{5.11}$$

for a Lagrange multiplier $\lambda : \Omega \to \mathbb{R}$. Using an approach employed in [42] for a similar problem in the discrete setting, the Lagrange multiplier can be determined and subsequently eliminated: For the points where the constraint is active, i. e. $|p|^2 = 1$, scalar multiplying (5.9) by

$$\nabla\left(\frac{\nu^2 \operatorname{div} p - 2\nu f_2}{2(f_1+f_2)}\right) + \lambda p$$

shows that

$$\left|\nabla\left(\frac{\nu^2 \operatorname{div} p - 2\nu f_2}{2(f_1+f_2)}\right)\right|^2 - |\lambda p|^2 = 0.$$

Because of (5.10) and $|p|^2 = 1$, we have $|\lambda p| = |\lambda|\,|p| = \lambda$. This allows us to determine λ:

$$\lambda = \left|\nabla\left(\frac{\nu^2 \operatorname{div} p - 2\nu f_2}{2(f_1+f_2)}\right)\right| \tag{5.12}$$

Using the complementarity condition (5.11), we see that $\lambda = 0$ for the points where the constraint is inactive, i. e. where $|p|^2 < 1$. Hence, using (5.9), at these points

$$\nabla\left(\frac{\nu^2 \operatorname{div} p - 2\nu f_2}{2(f_1+f_2)}\right) = 0$$

holds. Thus, (5.12) also holds for the points where the constraint is inactive and we have determined the multiplier for all points. Now we can substitute λ in (5.9), and the optimality condition (without any multiplier left) for p is

$$\nabla\left(\frac{\nu^2 \operatorname{div} p - 2\nu f_2}{2(f_1+f_2)}\right) - \left|\nabla\left(\frac{\nu^2 \operatorname{div} p - 2\nu f_2}{2(f_1+f_2)}\right)\right| p = 0.$$

Scaling this by the positive constant $1/\nu$, we arrive at (5.8), concluding the proof. □

5.3.10 Remark. Following [42], (5.8) can be solved numerically by a semi implicit gradient descent scheme

$$\begin{aligned}
&\frac{p^{n+1} - p^n}{\tau} = \nabla\left(\frac{\operatorname{div} p^n - \frac{2}{\nu} f_2}{2(f_1+f_2)}\right) - \left|\nabla\left(\frac{\operatorname{div} p^n - \frac{2}{\nu} f_2}{2(f_1+f_2)}\right)\right| p^{n+1} \\
&\Rightarrow p^{n+1} = \left(p^n + \tau\nabla\left(\frac{\operatorname{div} p^n - \frac{2}{\nu} f_2}{2(f_1+f_2)}\right)\right) \Bigg/ \left(1 + \tau\left|\nabla\left(\frac{\operatorname{div} p^n - \frac{2}{\nu} f_2}{2(f_1+f_2)}\right)\right|\right).
\end{aligned}$$

Using the finite difference discretization from [42], first tests confirm the observation on the step size made in [42]: For practical purposes $\tau = \frac{1}{4}h^2$ seems to be the optimal step size.

5.3.2 Multiphase segmentation

Our functional can be extended to multiphase segmentation by applying the Vese–Chan approach to transform the binary segmentation functional into a multiphase one [125] (cf. Section 1.2.3). Because the general strategy to handle 2^n segments is already discussed in Section 1.2.3, we confine to discuss segmentation in 4 phases here. Already knowing the Vese–Chan approach, this is sufficient to see how to handle 2^n segments. Let $f_1, f_2, f_3, f_4 \in L^1(\Omega)$ such that $f_i \geq 0$ a. e., then the multiphase functional is given by

$$\begin{aligned} E_{\mathrm{UC}}[u_1, u_2] := \int_\Omega & u_1^2 u_2^2 f_1 + (1-u_1)^2 u_2^2 f_2 \\ & + u_1^2 (1-u_2)^2 f_3 + (1-u_1)^2 (1-u_2)^2 f_4 \,\mathrm{d}x \\ & + \nu \left(|Du_1|(\Omega) + |Du_2|(\Omega) \right). \end{aligned} \tag{5.13}$$

If we fix u_2, the reduced functional $E_{\mathrm{UC}}[\cdot, u_2]$ is the same as the two-phase functional (5.3) with the indicator functions $\tilde{f}_1 = u_2^2 f_1 + (1-u_2)^2 f_3$ and $\tilde{f}_2 = u_2^2 f_2 + (1-u_2)^2 f_4$. As in the two-phase case, we can assume $f_i > 0$ a. e. without loss of generality and because either $u_2^2 > 0$ or $(1-u_2)^2 > 0$ holds, we have $\tilde{f}_1, \tilde{f}_2 > 0$. Therefore, all statements proven for the two-phase functional can be applied to $E_{\mathrm{UC}}[\cdot, u_2]$, i. e. we can compute the global minimum (for fixed u_2). The same applies for fixed u_1, so as an optimization strategy, we propose to minimize with respect to u_1 and u_2 alternatingly (cf. the generic EM procedure from Section 2.2).

Even though it is easy to extend (5.3) to multiphase segmentation, the same does not apply to the ROF energy (5.6). There is no apparent extension in the sense of [125] to formulate the multiphase segmentation in a single functional.

5.3.3 Indicator parameters

In typical segmentation tasks, the indicator functions depend on unknown parameters, e. g. the gray values for each segment in case of the piecewise constant Mumford–Shah model. For the sake of simplicity, we discuss the latter model in its two-phase formulation here, i. e.

$$f_i(x) := (u_0(x) - c_i)^2 \text{ for } i = 1, 2,$$

but this discussion applies to other indicator functions and multiphase segmentation as well.

During the minimization we not only have to minimize (5.3) with respect to u, but also with respect to c_1 and c_2 as well. This is typically done in an alternating fashion, but there are two apparent possibilities to update the gray values: Minimize (5.3) with respect to c_1 and c_2 or do so for the energy in the set formulation (1.5). The two possible updating formulae for c_1 arising are

$$c_1 = \int_\Omega u^2 u_0 \,\mathrm{d}x \Big/ \int_\Omega u^2 \,\mathrm{d}x \quad \text{or} \quad c_1 = \int_{\{u > \frac{1}{2}\}} u_0 \,\mathrm{d}x \Big/ \int_{\{u > \frac{1}{2}\}} \mathrm{d}x \,.$$

The two possibilities only coincide if u is binary. The first formula not only averages u_0 in $\{u > \frac{1}{2}\}$, instead it takes into account the values of u_0 everywhere, but weights the values according to u^2. To a certain degree this is similar to the effect of the regularization of the Heaviside function in the model of Chan and Vese. From our experiments, this reduces the chance of getting stuck in local minima that can still occur when minimizing over u and the indicator parameters. Particulary in the case of multiphase segmentation it turned out to

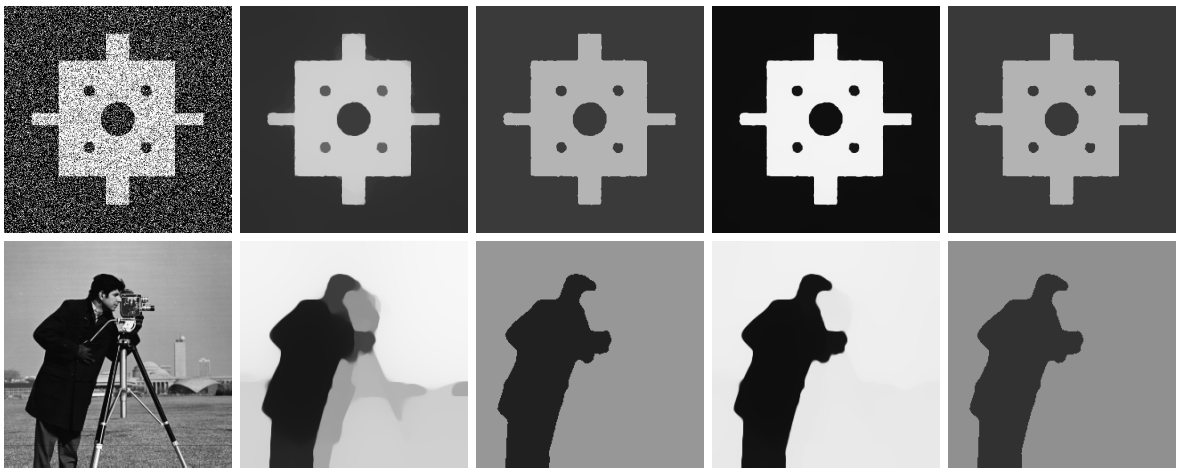


Figure 5.1: Segmentation of an artificial noisy structure ($\nu = 2 \cdot 10^{-3}$, top row) and the well-known Matlab cameraman image ($\nu = 4 \cdot 10^{-3}$, bottom row): Input image u_0 (left), segmentation function u and 0.5-superlevelset of u colored with the average gray values c_1, c_2 obtained by our model (middle) and by using E_{NEC} (right). The slight difference of the gray values is attributed to the employed update formula, cf. Section 5.3.3.

be beneficial: Using the second formula when segmenting the image shown in the top row of Figure 5.2 only three segments were identified by the algorithm, while all four segments were found with the first formula. The respective update formulae for c_2 are

$$c_2 = \int_\Omega (1-u)^2 u_0 \,\mathrm{d}x \Big/ \int_\Omega (1-u)^2 \,\mathrm{d}x \quad \text{or} \quad c_2 = \int_{\Omega\setminus\{u>\frac{1}{2}\}} u_0 \,\mathrm{d}x \Big/ \int_{\Omega\setminus\{u>\frac{1}{2}\}} \mathrm{d}x \,.$$

Due to the different way f_1 and f_2 are used in the ROF energy (5.6), it is not quadratic in c_1 and c_2. So this functional does not give a natural formula to update the gray values.

5.4 Numerical examples

In this section, we show the practical usability of the proposed model by applying it to the classical piecewise constant Mumford–Shah functional (1.5). For the sake of simplicity we do not use the dual formulation based minimization algorithm from Section 5.3.1 as minimization technique but an explicit gradient flow scheme (cf. Section 6.2) with the Armijo rule (cf. Section 6.3) as step size control combined with a spatial discretization by bilinear Finite Elements on a regular quadrilateral grid (cf. Section 6.1). The absolute value used to approximate the total variation is regularized as discussed in Section 1.2.2, i. e. $|z|_\varrho = \sqrt{z^2+\varrho^2}$ (in all examples presented here, $\varrho = 0.1$ is used). The gray values c_1 and c_2 are initialized with 0 and 1 respectively and updated occasionally during the gradient flow.

By choosing this minimization strategy instead of the dual algorithm, we could reuse most of the multiphase segmentation code already written for Chapter 3. If runtime is important, the dual algorithm should be used instead of the approach we used in this section.

Figure 5.1 shows results of our method and of the one proposed by Nikolova et al. [102] on one artificial image and one digital photo. In both examples, the minimizer u from our model

is far from being binary, but this is nothing to be expected from the theory we presented here. The 0.5-superlevelset gives an accurate segmentation that is not influenced by the presence of heavy noise (top row) and works on non-binary input images (bottom row). The minimizers u of the Nikolova et al. model look very different, but the segmentation obtained from the 0.5-superlevelsets is almost identical.

Upon closer inspection, the minimizer u of our model from the top row of Figure 5.1 looks very much like as obtained by minimizing the Rudin–Osher–Fatemi energy [109] with u_0 as input image. This is not surprising due to the following observation: If u_0 is binary, i. e. $u_0 = \chi_A$ for a set $A \subset \Omega$ and $c_1 = 1$, $c_2 = 0$ we have $f_1 = (\chi_A - 1)^2 = \chi_{\Omega\setminus A}$ and $f_2 = (\chi_A - 0)^2 = \chi_A$ and therefore

$$\begin{aligned} E_{\mathrm{UC}}[u] &= \int_\Omega u^2\chi_{\Omega\setminus A} + (1-u)^2\chi_A \,\mathrm{d}x + \nu\,|Du|\,(\Omega) \\ &= \int_\Omega u^2\chi_{\Omega\setminus A} + \chi_A - 2u\chi_A + u^2\chi_A \,\mathrm{d}x + \nu\,|Du|\,(\Omega) \\ &= \int_\Omega (u-\chi_A)^2 \,\mathrm{d}x + \nu\,|Du|\,(\Omega), \end{aligned}$$

i. e. E_{UC} equals the Rudin–Osher–Fatemi energy in this special case. This is not the case if u_0 is non-binary which can be seen from the bottom row of Figure 5.1.

Figure 5.2 shows four-phase segmentation results. Those indicate the tendency of the segmentation functions to become binary for small values of ν.

Finally, Figure 5.3 illustrates the behavior of the method for different numbers of segments and Figure 5.4 shows three iterates of the eight-phase segmentation.

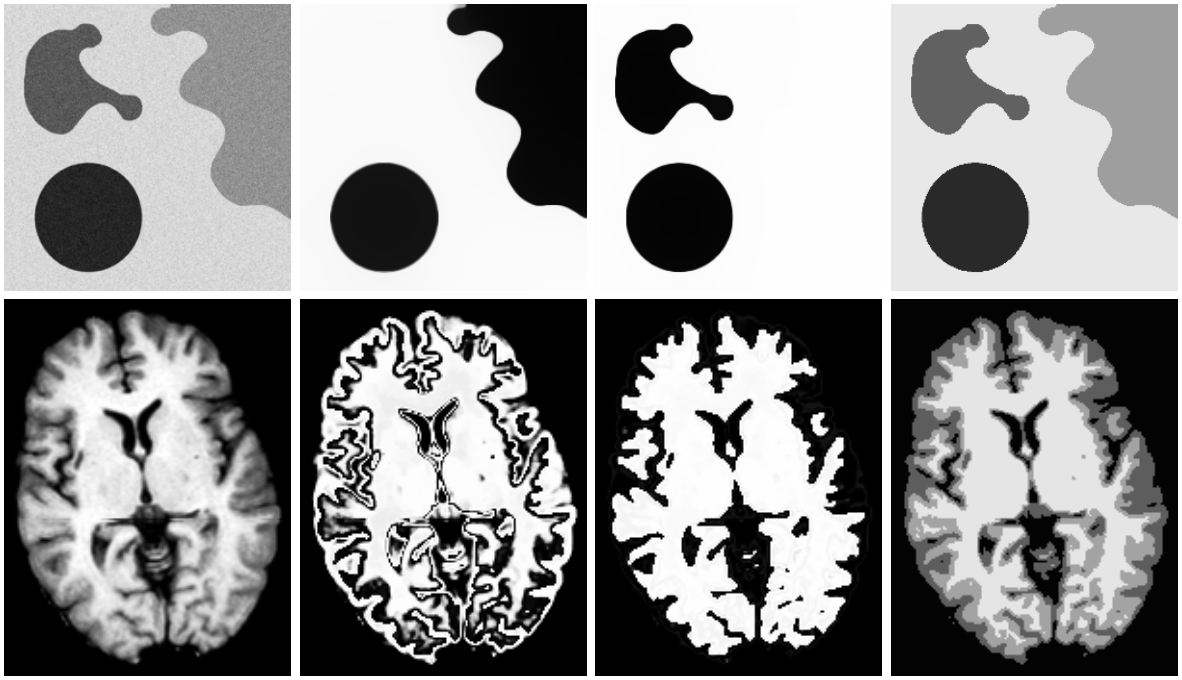


Figure 5.2: four-phase segmentation of an artificial noisy image (top row) and a MRI image (bottom row) ($\nu = 6 \cdot 10^{-4}$): Input image u_0 (left), segmentation functions u_1 and u_2 (middle), segmentation colored with the average gray values $c_1, ..., c_4$ (right).

Figure 5.3: Segmentation of a digital photo ($\nu = 2 \cdot 10^{-5}$). Input image u_0 (left), segmentation in four (middle) and eight (right) segments colored with the average gray values of the segments. Original image © bigmama / PIXELIO.

Figure 5.4: Intermediate results of the segmentation in eight segments shown in Figure 5.3 after 50 (left), 250 (middle) and 700 (right) gradient descent steps.

5.5 Outlook

Judging from the results shown in Chapters 3 and 4, one may get the impression that the Chan–Vese model is always sufficient to obtain suitable minimizers of the Mumford–Shah functional and that the extra machinery we build in this chapter to obtain global minimizers (instead of only local ones) for binary segmentation problems is not necessary. However, a quick look at the literature shows that global minimization techniques are getting more and more popular in particular in the context of binary segmentation, e. g. [85, 123].

In the following, we introduce a segmentation problem, where the Chan–Vese model often does not find usable minimizers. In particular, this problem motivated us to start researching on alternatives to Chan–Vese and eventually lead to the unconstrained global two-phase minimization approach presented in this chapter.

5.5.1 Flowfield segmentation

There is a wide range of methods for the visualization of vector fields, each of them with distinct assets and drawbacks. To give a glimpse at the range existing methods already cover, Figure 5.5 shows the same vector field visualized by three different methods.

Here, we are searching for a way to identify a number of regions that characterize a given vector field. When those regions are known, they can be used to enhance the visualization. Following the models discussed in the preceding chapters, we are going to construct a variational model to segment vector fields. A segmentation into piecewise constant parts is obviously not sufficient though, because vector fields commonly contain structures that cannot be described

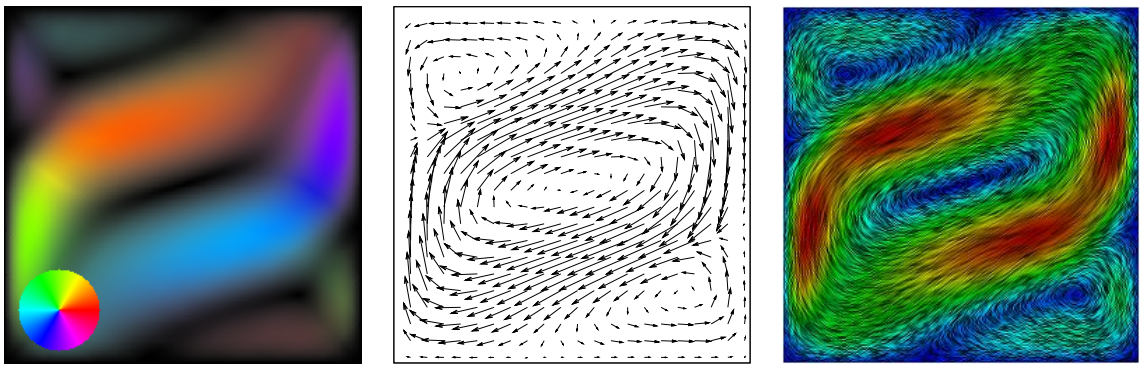


Figure 5.5: Three largely different ways to visualize a 2D vector field: Angle and lengths of the vectors encoded as color and brightness respectively (left), classical sparse arrow plot (middle) and a visualization using anisotropic diffusion and a color coding of the vector lenght [34] (right).

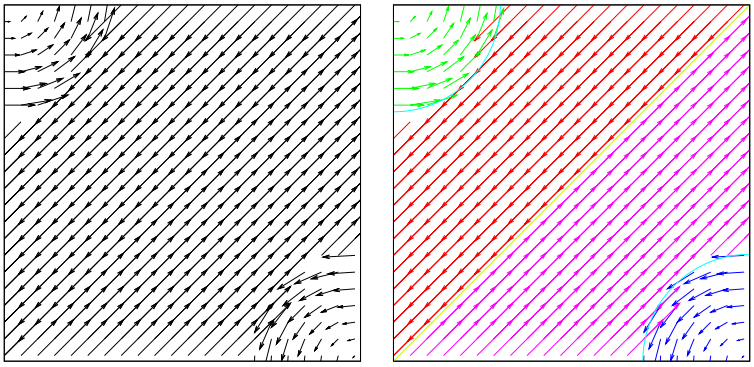


Figure 5.6: An artificial vector field, consisting of four affine regions, drawn as simple arrow plot (left) and as arrow plot colored according to its segments.

properly with a small number of constant segments. On the other hand, many of the common local structures encoded in vector fields (laminar flows, sheer flows, sources, sinks, vortices) can be approximated reasonably well with affine functions. Henceforth, we aim for a segmentation into affine regions (cf. Figure 5.6). For the sake of convenience, we only formalize the binary case here. The extension to multiphase segmentation is done exactly as described in Sections 1.2.3 and 5.3.2).

Given a vector field $v : \Omega \to \mathbb{R}^d$, the piecewise constant Mumford–Shah functional for two-phase segmentation (1.5) can be extended to handle the segmentation into piecewise affine regions by simply replacing the indicator functions. This leads to the following functional:

$$E_{\text{flow,MS}}[\Sigma, (A_i, b_i)_i] = \int_\Sigma |v(x) - A_1 x - b_1|^2 \,\mathrm{d}x + \int_{\Omega\setminus\Sigma} |v(x) - A_2 x - b_2|^2 \,\mathrm{d}x + \nu \,\mathrm{Per}(\Sigma).$$

Here, $A_1, A_2 \in \mathbb{R}^{d\times d}$ and $b_1, b_2 \in \mathbb{R}^d$ are the unknown affine parameters that replace the average gray values from (1.5).

Recalling (5.3), the corresponding unconstrained convex functional is

$$E_{\text{flow,UC}}[u,(A_i,b_i)_i] = \int_\Omega u^2\,|v(x)-A_1x-b_1|^2 + (1-u)^2\,|v(x)-A_2x-b_2|^2\,\mathrm{d}x + \nu\,|Du|\,(\Omega).$$

The optimization with respect to u has already been discussed in Chapter 5. Hence, we confine to study the optimization with respect to the affine parameters. The variation with respect to A_1 and b_1 respectively are

$$\partial_{A_1} E_{\text{flow,UC}}[u,(A_i,b_i)_i] = 2\int_\Omega u^2(A_1x+b_1-v(x))\otimes x\,\mathrm{d}x,$$
$$\partial_{b_1} E_{\text{flow,UC}}[u,(A_i,b_i)_i] = 2\int_\Omega u^2(A_1x+b_1-v(x))\,\mathrm{d}x\,.$$

Here we used that for $A=(A^{lm})_{lm}\in\mathbb{R}^{d\times d}$ and $b=(b^k)_k\in\mathbb{R}^d$ the following is fulfilled:

$$\partial_{A^{lm}}(Ax) = \partial_{A^{lm}}\left(\left(\sum\nolimits_j A^{ij}x_j\right)_{i=1}^d\right) = \left(\left(\sum\nolimits_j \delta_{il}\delta_{jm}x_j\right)_{i=1}^d\right) = \left((\delta_{il}x_m)_{i=1}^d\right) = x_m e_l,$$

and

$$\begin{aligned}\partial_{A^{lm}}\left[(v(x)-Ax-b)^2\right] &= 2\,(v(x)-Ax-b)\cdot(-x_m e_l) = 2(Ax+b-v(x))_l x_m\\ &= (2\,(Ax+b-v(x))\otimes x)_{lm}\,,\\ \partial_{b^k}\left[(v(x)-Ax-b)^2\right] &= 2\,(v(x)-Ax-b)\cdot(-e_k) = 2(Ax+b-v(x))_k.\end{aligned}$$

To calculate the optimal affine parameters A_1 and b_1 for a given segmentation u, we have to solve for zero crossings of the corresponding variations:

The variation with respect to A_1 leads to

$$\begin{aligned}0 &= \frac{1}{2}\partial_{A_1^{lm}} E_{\text{flow,UC}}[u,(A_i,b_i)_i]\\ &= \sum_j A_1^{lj}\underbrace{\int_\Omega u(x)^2x_jx_m\,\mathrm{d}x}_{=:H_{mj}} + b_1^l\underbrace{\int_\Omega u(x)^2x_m\,\mathrm{d}x}_{=:H_m} - \underbrace{\int_\Omega u(x)^2v_l(x)x_m\,\mathrm{d}x}_{=:V_{ml}}.\end{aligned}$$

Therefore, A_1 and b_1 have to solve

$$0 = \sum_j A_1^{lj}H_{mj} + b_1^l H_m - V_{ml} \quad \text{for } m,l=1,\dots,d. \tag{5.14}$$

Furthermore, the variation with respect to b_1 leads to

$$\begin{aligned}0 &= \frac{1}{2}\partial_{b_1^k} E_{\text{flow,UC}}[u,(A_i,b_i)_i]\\ &= \sum_j A_1^{kj}\underbrace{\int_\Omega u(x)^2x_j\,\mathrm{d}x}_{=H_j} + b_1^k\underbrace{\int_\Omega u(x)^2\,\mathrm{d}x}_{=:H} - \underbrace{\int_\Omega u(x)^2v_k(x)\,\mathrm{d}x}_{=:V_k}.\end{aligned}$$

Hence, in addition to (5.14), A_1 and b_1 have to solve

$$0 = \sum_j A_1^{kj} H_j + b_k H - V_k \quad \text{for } k = 1, \dots, d. \tag{5.15}$$

(5.14) and (5.15) show that d linear equation systems of size $(d+1) \times (d+1)$ have to be solved to find the optimal affine parameters A_1 and b_1.

For $d = 2$, these two linear equation systems are:

$$\begin{pmatrix} H_1 & H_2 & H \\ H_{11} & H_{12} & H_1 \\ H_{21} & H_{22} & H_2 \end{pmatrix} \begin{pmatrix} A_1^{11} \\ A_1^{12} \\ b_1^1 \end{pmatrix} = \begin{pmatrix} V_1 \\ V_{11} \\ V_{21} \end{pmatrix}$$

and

$$\begin{pmatrix} H_1 & H_2 & H \\ H_{11} & H_{12} & H_1 \\ H_{21} & H_{22} & H_2 \end{pmatrix} \begin{pmatrix} A_1^{21} \\ A_1^{22} \\ b_1^2 \end{pmatrix} = \begin{pmatrix} V_2 \\ V_{12} \\ V_{22} \end{pmatrix}.$$

The optimal affine parameters A_2 and b_2 can be calculated analogously.

5.5.2 Preliminary numerical results

Using the discretization and the minimization approach described in Section 5.4 combined with the linear equation systems to update the affine parameters (cf. (5.14) and (5.15)), we segment several vector fields in this section. At first, we compare the performance of the segmentation method using the Chan–Vese model (cf. Section 1.2.2) and the unconstrained convex model (cf. Section 5.3), both extended to multiphase segmentation with the Vese–Chan approach (cf. Section 1.2.3), in Figure 5.7. Here, the results obtained by the Chan–Vese model are clearly limited by the initial segmentation, whereas the unconstrained convex approach does not seem to be bound by the initialization.

Figure 5.8 shows the segmentation and approximation quality of the unconstrained convex model on another vector field. The segmentation behaves as expected: The more segments the model is allowed to use, the smaller the approximation error is. Unfortunately, an affine segmentation does not seem to be enough to structure this particular vector field into segments that are easy to interpret. More sophisticated segmentation criteria need to be developed to handle the visualization of this kind of vector fields.

Finally, Figure 5.9 shows a 3D segmentation result of the unconstrained convex model. The 3D segments are difficult to interpret, but the inflow or outflow region that generated the flow shown by the vector field are clearly visible.

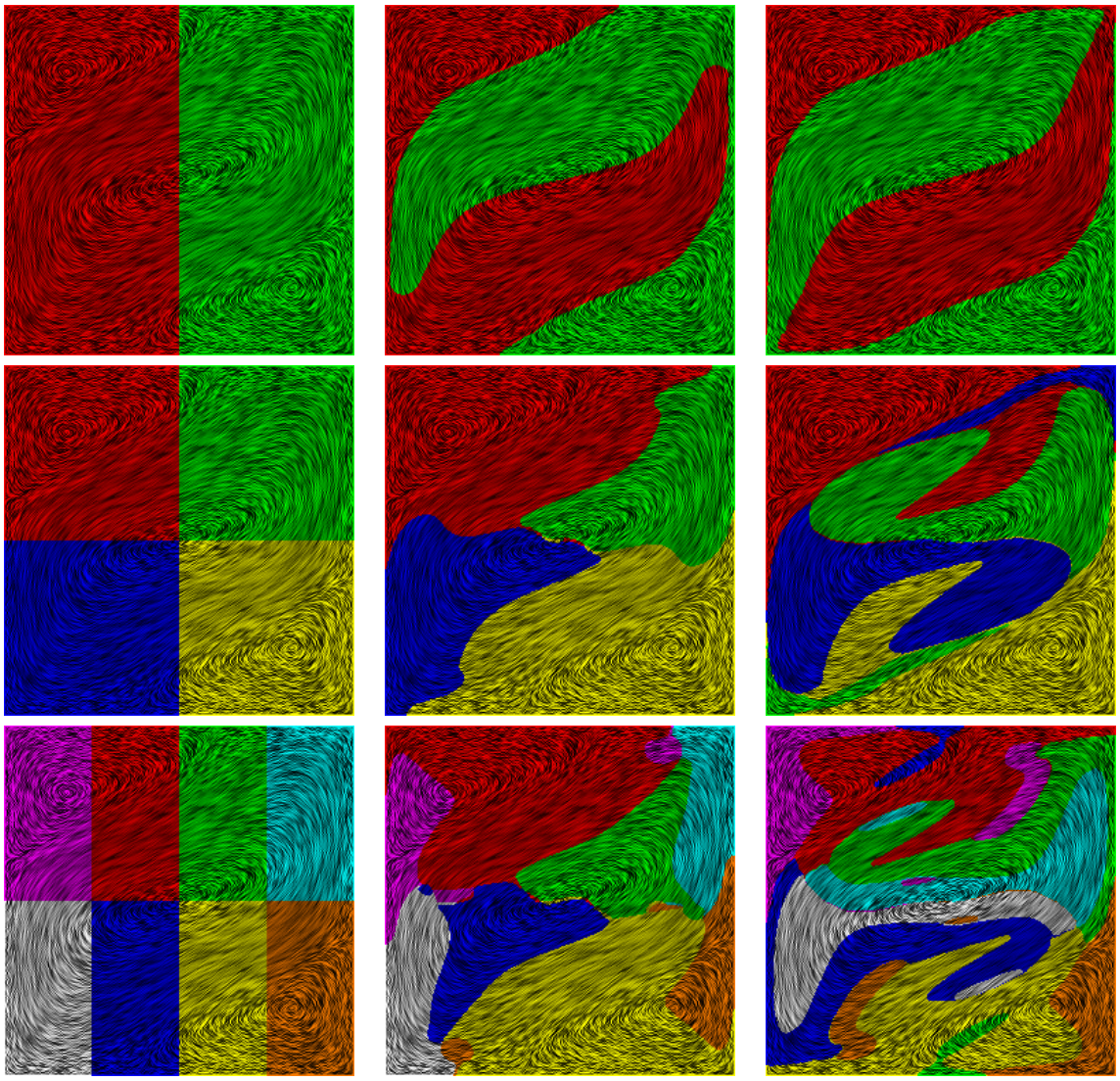
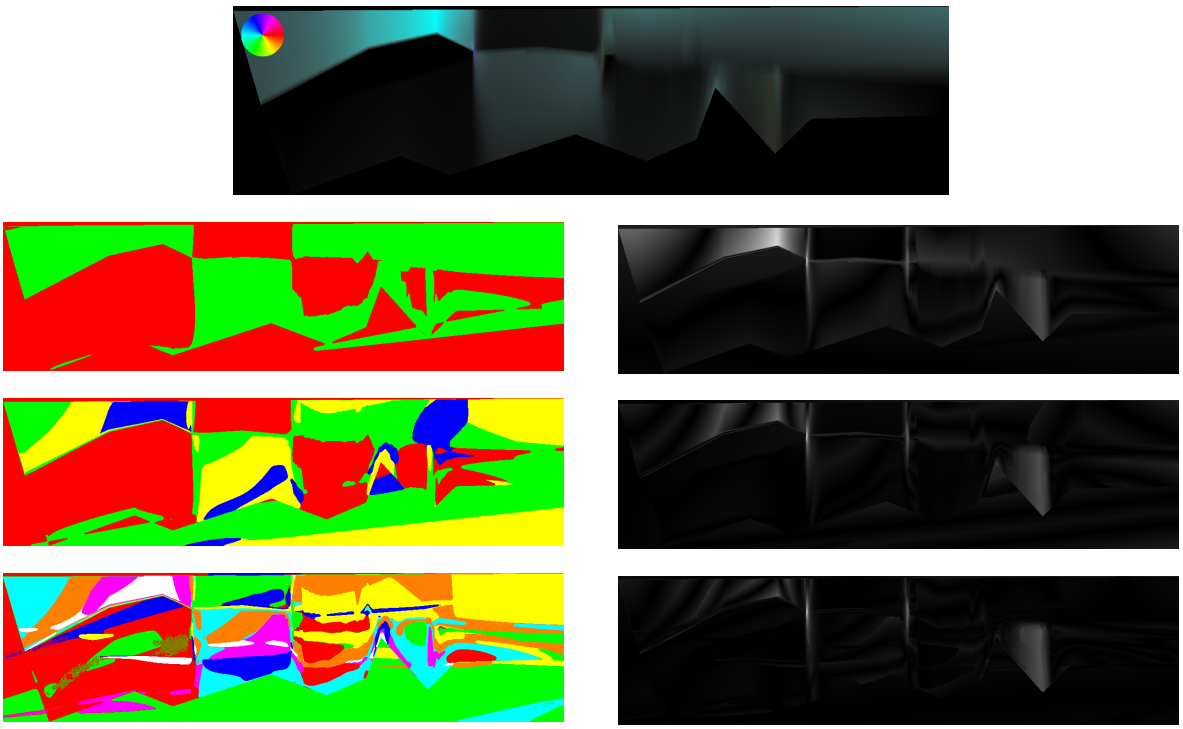


Figure 5.7: Comparison of the Chan–Vese model and the unconstrained convex model (extended to multiphase segmentation following Vese–Chan) on the vector field from Figure 5.5. From top to bottom, segmentation in two, four and eight segments is shown.



segments error

Figure 5.8: Segmentation of a vector field (top row) in two, four and eight segments. The left column shows the segments, while the right column shows the pointwise error, i. e. the absolute difference of the piecewise affine approximation to the original vector field. From top to bottom, the full L^2-error is $1.82 \cdot 10^{-4}$, $1.06 \cdot 10^{-4}$, and $8.39 \cdot 10^{-5}$, respectively.

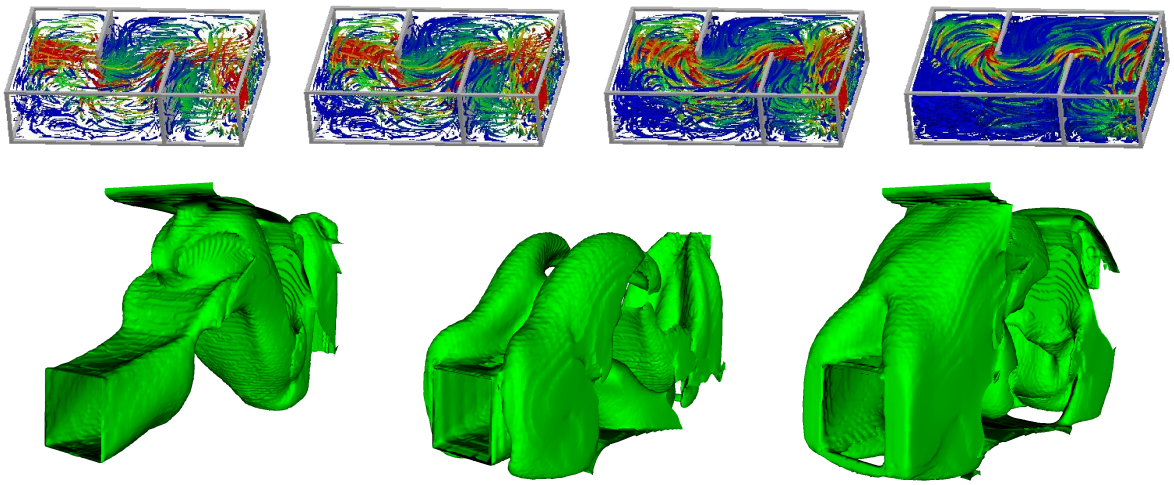


Figure 5.9: Segmentation of a 3D vector field in eight segments: Input vector field visualized by anisotropic diffusion (top row) and the 0.5-level set of the three segmentation functions (bottom row).

5.5.3 Future work

As we have seen in Figure 5.8, a simple piecewise affine segmentation is not sufficient to segment some vector fields in such a way that the segments can be used to ease the vector field visualization. A possibility to overcome these difficulties could be to incorporate prior knowledge about the vector fields we want to segment. For example, this can be done by adding a penalty term that encodes the prior knowledge. As very simple penalty, we could consider the penalty function

$$F[A] = \sum_{i=2}^{d} \sum_{j=1}^{i-1} \left(A^{ij} + A^{ji}\right)^2 .$$

Using the notation above, the variation of the aforementioned penalty function is

$$\partial_{A^{lm}} F[A] = 2(1 - \delta_{lm})(A^{lm} + A^{ml}).$$

If we now add $F[A_1]$ to our objective functional $E_{\text{flow,UC}}$, the segment corresponding to A_1 and b_1 is penalized by our penalty function.

As the penalty term is quadratic in A_1, we can still update the affine parameters by solving linear equation system. In concrete terms, (5.14) changes to

$$0 = \sum_j A_1^{lj} H_{mj} + b_1^l H_m - V_{ml} + (1 - \delta_{lm})(A^{lm} + A^{ml}) \quad \text{for } m, l = 1, \ldots, d.$$

The affine parameters are more interdependent with the penalty term, though, and we end up with one linear equation system of size $d(d+1) \times d(d+1)$ instead of d linear equation systems of size $(d+1) \times (d+1)$ as before.

6 Appendix

6.1 Multi-linear Finite Elements

In this section, we recall the multi-linear Finite Element (FE) method, used for the spatial discretization of most of the models presented in this work, and introduce the necessary notation. For a detailed introduction, we refer to [27].

The computational domain $\Omega \subset [0,1]^d$ is covered by a uniform rectangular mesh $\mathcal{C}$ such that the elements of the mesh are squares in 2D and cubes in 3D. h denotes the mesh resolution, i. e. the maximal edge length among all edges of all elements of the mesh. We consider all images as sets of pixels (2D) / voxels (3D), where each pixel / voxel corresponds to a grid node of $\mathcal{C}$. Let $\mathcal{N} = \{x_1, \dots, x_n\}$ denote the nodes of $\mathcal{C}$. The FE basis function of node x_i is defined as the continuous, piecewise multi-linear (mutli-linear on every element of the grid) function uniquely determined by the nodality property, i. e.

$$\Lambda_i(x_j) = \delta_{ij} := \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

Obviously these functions are linearly independent and therefore form a basis of their linear span. The FE-space $\mathcal{V}$ for the mesh $\mathcal{C}$ is defined as the linear span of the Λ_i, i. e.

$$\mathcal{V} := \operatorname{span}(\Lambda_1, \dots, \Lambda_n).$$

The FE-space of vector-valued functions is $\mathcal{V}^d$, the canonical basis of this space, is

$$\Lambda_1 e_1, \dots, \Lambda_n e_1, \dots, \Lambda_1 e_d, \dots, \Lambda_n e_d,$$

where $e_i = (\delta_{ij})_{j=1}^d$ is the i-th canonical basis vector of $\mathbb{R}^d$. Note that in the FE spaces $\mathcal{V}$ and $\mathcal{V}^d$, an element is uniquely determined by its values on the nodes $\mathcal{N}$. Thus in our FE context scalar and vector-valued functions, e. g. u and ψ, are approximated by their values at the mesh nodes, i. e.

$$u \approx U := \sum_{i=1}^{n} u(x_i)\Lambda_i(x) \text{ and } \psi = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_d \end{bmatrix} \approx \Psi := \begin{bmatrix} \sum_{i=1}^n \psi_1(x_i)\Lambda_i(x) \\ \vdots \\ \sum_{i=1}^n \psi_d(x_i)\Lambda_i(x) \end{bmatrix}.$$

The FE approximation of a function can also be represented by the vector of the function values on the nodes, e. g. $\overline{U} := (U(x_i), \dots, U(x_n))^T$ and $\overline{\Psi} := (\overline{\Psi_1}, \dots, \overline{\Psi_d})^T$ where $\overline{\Psi_j} = (\Psi_j(x_1), \dots, \Psi_j(x_n))^T$. To discern the three different objects directly from the notation, we denote continuous functions by lowercase letters (e. g. u or ψ), their FE representation by uppercase letters (e. g. U or Ψ) and their vector representation by "over-lined" uppercase letters (e. g. $\overline{U}$ or $\overline{\Psi}$).

Finally, let us introduce generalized mass and stiffness matrices, which are an important ingredient of the FE method: Given a function $f : \Omega \to \mathbb{R}$, the generalized mass matrix $M[f]$ and stiffness matrix $L[f]$ are defined as follows:

$$M[f] = \left(\int_\Omega f(x) \Lambda_i(x) \Lambda_j(x) \, \mathrm{d}x \right)_{i,j} \tag{6.1}$$

$$L[f] = \left(\int_\Omega f(x) \nabla \Lambda_i(x) \cdot \nabla \Lambda_j(x) \, \mathrm{d}x \right)_{i,j} \tag{6.2}$$

Both matrices are of size $n \times n$ and both are sparse, i. e. most of their entries are zero. Due to the special choice of our FE space $\mathcal{V}$ (piecewise multi-linear functions on a rectangular mesh) an entry of such a matrix is non-zero if and only if the nodes x_i and x_j are either adjacent in the mesh $\mathcal{C}$ or equal, i. e. $i = j$. To compute the integrals in these non-zero entries, we use a numerical Gauss quadrature scheme of order 3 (cf. [112]). The usual mass matrix M and stiffness matrix L are obviously just special cases of the generalized ones, i. e. $M := M[1]$ and $L := L[1]$.

6.2 Minimization by gradient flows

Gradient flows are a well-known and commonly used tool to find (local) minimizers of a given functional. While gradient flows are certainly not the definite answer to energy minimization, there are two main reasons making them very appealing. First, they do not put any special requirements on the objective functional (apart from mild regularity assumptions that can be fulfilled by regularization) and thus can be applied to almost any minimization problem. Second, they can be applied in a straightforward manner as they only need the variation of the objective functional. Due to these properties, we use gradient flows to solve most of the minimization problems we encounter in this work.

As small motivation, we consider the finite dimensional case to outline how a gradient flow works. Consider $F \in C^1(\mathbb{R}^n, \mathbb{R})$ and recall that $\operatorname{grad} F(X) = (\partial_{X_1} F(X), \dots, \partial_{X_n} F(X))$ by construction locally always points in the direction of steepest ascent, thus $-\operatorname{grad} F(X)$ is the direction of steepest descent. An obvious strategy to find a minimizer is to start at any position $X_0 \in \mathbb{R}^n$ and continuously move into the direction of steepest descent, described by the following ordinary differential equation:

$$\begin{aligned} \frac{d}{dt} X(t) &= -\operatorname{grad} F(X(t)), \\ X(0) &= X_0. \end{aligned}$$

This method is called *gradient descent*. For any solution $X : [0, T] \to \mathbb{R}^n$ of this ODE for a finite time $T > 0$, it holds that

$$F(X_0) \geq F(X(T)),$$

which follows from

$$\begin{aligned} F(X(T)) - F(X_0) = (F \circ X)(T) - (F \circ X)(0) &= \int_0^T (F \circ X)'(t) dt \\ &= \int_0^T \operatorname{grad} F(X(t)) \cdot \frac{d}{dt} X(t) dt = -\int_0^T \operatorname{grad} F(X(t)) \cdot \operatorname{grad} F(X(t)) dt \\ &= -\int_0^T |\operatorname{grad} F(X(t))|_2^2 \, dt \leq 0. \end{aligned}$$

Since we assumed $\operatorname{grad} F$ to be continuous, under the additional assumption $\operatorname{grad} F(X_0) \neq 0$, the inequality even holds strictly, i. e.

$$F(X_0) > F(X(T)).$$

Hence, given any non-critical point, the ODE leads us to a point at which F is strictly lower than what we started with. If X_0 is a critical point though, the solution of the ODE is the constant curve $X(t) = X_0$ and we cannot minimize F any further with the gradient descent.

Now we consider the more interesting infinite dimensional case: Let $\mathcal{H}$ be a real-valued function space and $E : \mathcal{H} \to \mathbb{R}$ a functional on $\mathcal{H}$. Furthermore, let $g : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ be a scalar product considered as a metric on $\mathcal{H}$ such that $\mathcal{H}$ is complete with respect to the norm induced by g, i. e. $(\mathcal{H}, g)$ is a Hilbert space. The gradient of E with respect to the metric g is characterized by the condition

$$\operatorname{grad}_g E[x] = v \in \mathcal{H} :\Leftrightarrow \underset{w \in \mathcal{H}}{\forall} \quad g(v, w) = \left\langle E'[x], w \right\rangle.$$

For any initial position $x_0 \in \mathcal{H}$, the *gradient flow* for E in $\mathcal{H}$ with respect to g is given by the following ODE

$$\begin{aligned} \partial_t x(t) &= -\operatorname{grad}_g E[x(t)], \\ x(0) &= x_0. \end{aligned}$$

With the definition of the gradient, this means

$$\underset{w \in \mathcal{H}}{\forall} \quad g\left(\partial_t x(t), w\right) = -\left\langle E'[x(t)], w \right\rangle.$$

By the Riesz representation theorem [13, Theorem 12.5], there is a bijective representation $A : \mathcal{H} \to \mathcal{H}'$ of g, i. e. it holds that

$$\underset{x,y \in \mathcal{H}}{\forall} \quad g(x, y) = \langle Ax, y \rangle.$$

Therefore,

$$\begin{aligned} &\underset{w \in \mathcal{H}}{\forall} \quad \langle A \partial_t x(t), w \rangle = g\left(\partial_t x(t), w\right) = -\left\langle E'[x], w \right\rangle \\ &\Leftrightarrow A \partial_t x = -E'[x] \\ &\Leftrightarrow \partial_t x = -A^{-1} E'[x] \end{aligned}$$

This reformulation of the gradient flow ODE outlines the effect the metric imposes on the evolution. By construction the gradient flow evolves from the starting point x_0 to the "nearest"

local minimum and that is where the metric comes into play: The notion of a distance on our space $\mathcal{H}$ is inherently dependent on the metric g we use to measure lengths and thus the choice of the metric influences which minimum the flow is attracted to. If we know that the solution sought fulfills certain properties, we can use a metric that favors these properties.

For example, if we want to select the smoothest minimizer from the set of nearby local minima, it is suitable to chose g (inspired by the Sobolev active contour approach [120]) to be a scaled version of the H^1 metric (and chose $\mathcal{H}$ accordingly, e. g. $\mathcal{H} = H^1(\Omega)$), i. e.

$$g_\sigma(\vartheta_1, \vartheta_2) = \int_\Omega \vartheta_1 \vartheta_2 + \frac{\sigma^2}{2} \nabla \vartheta_1 \cdot \nabla \vartheta_2 \, \mathrm{d}x \tag{6.3}$$

for variations $\vartheta_1, \vartheta_2 \in \mathcal{H}$. In this case, applying A^{-1} is equivalent to doing one time step of the heat equation discretized in time using the backward Euler scheme and step size $\frac{\sigma^2}{2}$. By [66, Section 2.3, Theorem 1], (continuously) solving the heat equation for time $\frac{\sigma^2}{2}$ in turn is the same as convolving with $\mathbf{G}_\sigma$, the Gaussian kernel of filter width σ, i. e.

$$\mathbf{G}_\sigma(x) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^d} \exp\left(\frac{-|x|^2}{2\sigma^2}\right). \tag{6.4}$$

The analogon of this metric for vector-valued problems (i. e. in case $\mathcal{H} = H^1(\Omega, \mathbb{R}^n)$) is

$$g_\sigma(\zeta_1, \zeta_2) = \int_\Omega \zeta_1(x) \cdot \zeta_2(x) + \frac{\sigma^2}{2} D\zeta_1(x) : D\zeta_2(x) \, \mathrm{d}x \tag{6.5}$$

for variations $\zeta_1, \zeta_2 \in H^1(\Omega, \mathbb{R}^n)$. With this metric, applying A^{-1} is equivalent to one implicit heat equation time step with step size $\frac{\sigma^2}{2}$ on each scalar component.

Because of the regularizing properties of the inverse of the representation of g_σ combined with the time discretization, this approach is closely related to iterative Tikhonov regularization, which is known to lead to smooth paths from the initial deformations towards the set of minimizers, cf. [113, 51, 50]. In case the inverse of the metric A^{-1} has a smoothing effect, the gradient flow is sometimes also referred as *regularized gradient descent*. Note that the choice of the metric does not alter the energy landscape itself in any way, but solely the descent path towards the set of minimizers. With an explicit time discretization (i. e. forward Euler) of $\partial_t x = -A^{-1}E'[x]$ for a step size τ, we end up with the following explicit gradient flow scheme

$$x^{k+1} = x^k - \tau A^{-1} E'[x]$$

that is used as basis for most minimization algorithms used in this work.

6.3 Step size control

The following simple example shows that the step size in an explicit gradient flow scheme cannot be chosen freely: Consider $E : \mathbb{R} \to \mathbb{R}, x \mapsto x^2$. The corresponding explicit gradient flow scheme for an initial value x_0 is

$$x_{k+1} = x_k - \tau \operatorname{grad} E(x_k) = x_k - 2\tau x_k.$$

Choosing $\tau = 1$, one obtains $x_{k+1} = -x_k$, leading to

$$x_k = (-1)^k x_0.$$

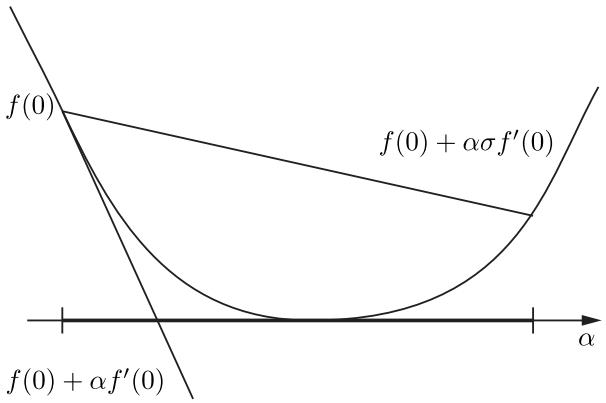


Figure 6.1: Obviously the slope ratio is not necessarily bigger or equal to one, but there exists a $\tau > 0$ such that the ratio exceeds an arbitrary but fixed $\sigma \in (0, 1)$.

Therefore, for any initial value $x_0 \neq 0$, the sequence (x_k) does not not converge to the minimizer $x = 0$, it does not even converge at all. The objective function E cannot be held responsible for this failure, as it fulfills all important properties one can hope for: It is smooth, even analytic, and strictly convex. Hence it is necessary to carefully choose the step size.

Given an objective functional $E : X \to \mathbb{R}$, an estimate of the minimizer $x \in X$ and a descent direction $d \in X$ towards the minimizer, consider

$$f : \mathbb{R} \to \mathbb{R}, t \mapsto E[x + td].$$

Finding the optimal step size for the current gradient flow step is equivalent to minimizing the one-dimensional function f. We chose a proven approach to take care this one-dimensional optimization problem, the so-called *Armijo rule* [7, 25]. The idea is to find $\tau > 0$ such that slope ratio of the secant through $(0, f(0))$ and $(\tau, f(\tau))$ to the tangent of the graph of f in $(0, f(0))$ fulfills

$$\frac{\text{secant slope}}{\text{tangent slope}} \geq \sigma,$$

cf. Figure 6.1. The ratio condition is called Successive Reduction Rule (SRR) and ensures that a specified percentage of the decay predicted by the tangent slope is realized by the selected step size. Here, $\sigma \in (0, 1)$ denotes a parameter that selects the desired percentage. The tangent slope is given by

$$f'(0) = \langle E'[x], d \rangle$$

and does not vanish as long as 0 is no minimizer of f. The secant slope is

$$\frac{f(\tau) - f(0)}{\tau} = \frac{E[x + \tau d] - E[x]}{\tau}.$$

Thus the ratio is fulfilled if

$$\sigma \leq \frac{f(\tau) - f(0)}{\tau f'(0)} = \frac{E[x + \tau d] - E[x]}{\tau \langle E'[x], d \rangle}. \tag{6.6}$$

If f is sufficiently smooth, according to Taylor's theorem it holds that

$$f(t) = f(0) + tf'(0) + O(t^2)$$

and we get

$$\frac{f(t) - f(0)}{tf'(0)} = \frac{f(0) + tf'(0) + O(t^2) - f(0)}{tf'(0)} = 1 + O(t).$$

Therefore, for each $\sigma < 1$, there exists a $t_0 > 0$, such that for all $0 < \tau < t_0$ it holds that

$$\frac{f(\tau) - f(0)}{\tau f'(0)} \geq \sigma.$$

At least for smooth f, this ensures the existence of τ with the ratio property we are looking for. For $\beta \in (0, 1)$, the so-called *Armijo rule with widening* determines the smallest $m \in \mathbb{Z}$, such that

$$\frac{f(\beta^m) - f(0)}{\beta^m f'(0)} \geq \sigma.$$

In other words, β^m is the biggest possible step size (in the set $\{\beta^n | n \in \mathbb{Z}\}$) that fulfills (6.6). The following algorithm is used to determine τ:

Algorithm 6.1: Armijo rule with widening

```
% Initialize τ from the last gradient flow step;
τ = τ_old (or τ = 1);
% Find the largest τ fulfilling SSR;
if SSR succeeds then
    repeat
        τ ← τ/β;
    until SSR fails ;
    τ ← βτ;
else
    repeat
        τ ← βτ;
    until SSR succeeds ;
end
```

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor, Martin Rumpf, for all his help, support and guidance. I am particularly thankful for the opportunity to be part of his group for several years, which was an extremely pleasant and instructive experience.

Moreover, I am indebted to (in the order of the chapters) Jingfeng Han and Joachim Hornegger (Institute of Pattern Recognition, University of Erlangen-Nuremberg), and Marc Kotowski and Carlo Schaller (University Hospital Geneva), and Jasmin Scorzin (Department of Neurosurgery, University Hospital Bonn), and Andreas Rätz and Axel Voigt (Institute of Scientific Computing, Dresden University of Technology), and Leah Bar and Guillermo Sapiro (Department of Electrical and Computer Engineering, University of Minnesota) for the fruitful cooperation on joint projects within this thesis.

My thanks also go to all my colleagues at the Institute for Numerical Simulation, University of Bonn. In particular, I want to thank my office mate Ole Schwen for our numerous discussions on mathematical, implementational and typographical topics. I furthermore thank my colleagues Benedict Geihe, Ole Schwen, Martina Teusner and Benedikt Wirth for proofreading my thesis and their valuable comments. Additionally, I am grateful to my colleague Martin Lenz for his assistance, particularly in Linux related questions.

In addition, I acknowledge Rüdiger Bock (Institute of Pattern Recognition, University of Erlangen-Nuremberg) for providing the retinal images, and Geoffrey H. Campbell (Lawrence Livermore National Lab), Nick Schryvers (Antwerp University) and David M. Tricker (Cambridge University) for providing the transmission electron microscopy images.

Special thanks go to the German Research Foundation for their financial support, in particular via the Priority Program 1114 *Mathematical Methods for Time Series Analysis and Digital Image Processing* and the Collaborative Research Center 611 *Singular Phenomena and Scaling in Mathematical Models*.

Last but not least, I am deeply grateful to my family and my girlfriend for all their support over the years.

Benjamin Berkels

Bibliography

[1] The USC-SIPI image database. `http://sipi.usc.edu/database/`. University of Southern California, Signal and Image Processing Institute.

[2] F. Alter, V. Caselles, and A. Chambolle. A characterization of convex calibrable sets in $\mathbb{R}^N$. *Math. Ann.*, 332(2):329–366, 2005.

[3] C. V. Alvino and A. J. Yezzi. Fast Mumford-Shah segmentation using image scale space bases. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6498, 2007.

[4] L. Ambrosio and V. M. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. *Comm. Pure Appl. Math.*, 43:999–1036, 1990.

[5] Luigi Ambrosio, Nicola Fusco, and Diego Pallara. *Functions of bounded variation and free discontinuity problems*. Oxford Mathematical Monographs. Oxford University Press, New York, 2000.

[6] I. Arganda-Carreras, C. O. S. Sorzano, R. Marabini, J. M. Carazo, C. Ortiz-De-Solorzano, and J. Kybic. Consistent and elastic registration of histological sections using vector-spline regularization. In *Workshop of the 9th European Conference on Computer Vision (CVAMIA-06)*, volume 4241 of *LNCS*, pages 85–95, 2006.

[7] Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

[8] Mois I. Aroyo, Ulrich Müller, and Hans Wondratschek. *International Tables for Crystallography*, volume A1, chapter Historical introduction, pages 2–5. Springer, 2006.

[9] J. Ashburner, J. Andersson, and K. J. Friston. High-dimensional nonlinear image registration using symmetric priors. *NeuroImage*, 9:619–628, 1999.

[10] Jean-François Aujol, Gilles Aubert, and Laure Blanc-Féraud. Wavelet-based level set evolution for classification of textured images. *IEEE Transactions on Image Processing*, 12(12):1634–1641, 2003.

[11] Jean-François Aujol and Antonin Chambolle. Dual norms and image decomposition models. *International Journal of Computer Vision*, 63(1):85–104, 2005.

[12] Jean-François Aujol and Tony F. Chan. Combining geometrical and textured information to perform image classification. *Journal of Visual Communication and Image Representation*, 17(5):1004–1023, 2006.

[13] George Bachmann and Lawrence Narici. *Functional Analysis*. Academic Press, 1966.

[14] R. Backofen, A. Rätz, and A. Voigt. Nucleation and growth by a phase field crystal (PFC) model. *Philosophical Magazine Letters*, 87(11):813–820, 2007.

[15] J. M. Ball. Convexity conditions and existence theorems in nonlinear elasiticity. *Archive of Rational Mechanics and Analysis*, 63:337–403, 1977.

[16] J.M. Ball. Global invertibility of Sobolev functions and the interpenetration of matter. *Proceedings of the royal Society of Edinburgh*, 88A:315–328, 1981.

[17] L. Bar, N. Sochen, and N. Kiryati. Semi-blind image restoration via Mumford-Shah regularization. *IEEE Transactions on Image Processing*, 15(2):483–493, 2006.

[18] L. Bar, N. Sochen, and N. Kiryati. Restoration of images with piecewise space-variant blur. In *Proceedings of the First International Conference on Scale Space Methods and Variational Methods in Computer Vision*, pages 533–544. Springer, 2007.

[19] Leah Bar, Benjamin Berkels, Martin Rumpf, and Guillermo Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *Eleventh IEEE International Conference on Computer Vision (ICCV 2007)*, 2007.

[20] Benjamin Berkels. An unconstrained multiphase thresholding approach for image segmentation. In *Proceedings of the Second International Conference on Scale Space Methods and Variational Methods in Computer Vision (SSVM 2009)*, volume 5567 of *Lecture Notes in Computer Science*, pages 26–37. Springer, 2009.

[21] Benjamin Berkels, Martin Burger, Marc Droske, Oliver Nemitz, and Martin Rumpf. Cartoon extraction based on anisotropic image classification. In *Vision, Modeling, and Visualization Proceedings*, pages 293–300, 2006.

[22] Benjamin Berkels, Andreas Rätz, Martin Rumpf, and Axel Voigt. Identification of grain boundary contours at atomic scale. In *Proceedings of the First International Conference on Scale Space Methods and Variational Methods in Computer Vision (SSVM 2007)*, volume 4485 of *Lecture Notes in Computer Science*, pages 765–776. Springer, 2007.

[23] Benjamin Berkels, Andreas Rätz, Martin Rumpf, and Axel Voigt. Extracting grain boundaries and macroscopic deformations from images on atomic scale. *Journal of Scientific Computing*, 35(1):1–23, 2008.

[24] Marc Berthod, Zoltan Kato, Shan Yu, and Josiane B. Zerubia. Bayesian image classification using Markov random fields. *Image and Vision Computing*, 14(4):285–295, May 1996.

[25] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.

[26] Charles Bouman and Michael Shapiro. Multiscale random field model for Bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, March 1994.

[27] Diedrich Braess. *Finite Elemente*. Springer, Berlin / Heidelberg / New York, 1997.

[28] A. Braides. Γ-*convergence for beginners*. Oxford University Press, Oxford, 2002.

[29] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[30] X. Bresson, S. Esedoḡlu, P. Vandergheynst, J. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, June 2007.

[31] M. Bro-Nielsen and C. Gramkow. Fast fluid registration of medical images. In K. H. Höhne and R. Kikinis, editors, *Visualization in Biomedical Computing: 4th International Conference, VBC*, volume 1131 of *LNCS*, pages 267–276, 1996.

[32] Chaim Broit. *Optimal Registration of Deformed Images*. PhD thesis, University of Pensylvania, 1981.

[33] Thomas Brox, Christoph Bregler, and Jitendra Malik. Large displacement optical flow. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[34] D. Bürkle, T. Preußer, and M. Rumpf. Transport and anisotropic diffusion in time-dependent flow visualization. In *Proceedings Visualization 2001*, 2001.

[35] Ali Can, Charles V. Stewart, Badrinath Roysam, and Howard L. Tanenbaum. A feature-based, robust, hierarchical algorithm for registering pairs of images of the curved human retina. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):347–364, 2002.

[36] Ali Can, Charles V. Stewart, Badrinath Roysam, and Howard L. Tanenbaum. A feature-based technique for joint, linear estimation of high-order image-to-mosaic transformations: mosaicing the curved human retina. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):412–419, 2002.

[37] V. Caselles, L. Garrido, and L. Igual. A contrast invariant approach to motion estimationa contrast invariant approach to motion estimation. In *Scale Space and PDE Methods in Computer Vision. 5th International Conference, Scale-Space 2005, Hofgeismar, Germany, April 7-9, 2005*, pages 242–253, 2005.

[38] Vicent Caselles, Francine Catté, Tomeu Coll, and François Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66:1–31, 1993.

[39] Vicent Caselles, Antonin Chambolle, and Matteo Novaga. The discontinuity set of solutions of the TV denoising problem and some extensions. *Multiscale Modeling & Simulation*, 6(3):879–894, 2007.

[40] Antonin Chambolle. Inverse problems in image processing and image segmentation: some mathematical and numerical aspects. *ICTP Lecture Notes Series*, 2, 2000.

[41] Antonin Chambolle. An algorithm for mean curvature motion. *Interfaces and free Boundaries*, 6:195–218, 2004.

[42] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, November 2004.

[43] Antonin Chambolle. Total variation minimization and a class of binary MRF models. In A. Rangarajan et al., editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757 of *LNCS*, pages 136–152. Springer, 2005.

[44] Antonin Chambolle and Jérôme Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84(3):288–307, 2009.

[45] T. F. Chan, B. Y. Sandberg, and L. A. Vese. Active contours without edges for vector–valued images. *Journal of Visual Communication and Image Representation*, 11:130–141, 2000.

[46] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[47] Tony F. Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7(3):370–375, March 1998.

[48] G. E. Christensen and H. J. Johnson. Consistent image registration. *IEEE Transactions on Medical Imaging*, 20(7):568–582, 2001.

[49] G. E. Christensen, S. C. Joshi, and M. I. Miller. Volumetric transformations of brain anatomy. *IEEE Trans. Medical Imaging*, 16, no. 6:864–877, 1997.

[50] U. Clarenz, M. Droske, and M. Rumpf. Towards fast non–rigid registration. In *Inverse Problems, Image Analysis and Medical Imaging, AMS Special Session Interaction of Inverse Problems and Image Analysis*, volume 313, pages 67–84. AMS, 2002.

[51] U. Clarenz, S. Henn, and K. Rumpf, M. Witsch. Relations between optimization and gradient flow methods with applications to image registration. In *Proceedings of the 18th GAMM Seminar Leipzig on Multigrid and Related Methods for Optimisation Problems*, pages 11–30, 2002.

[52] D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005.

[53] Daniel Cremers and Christoph Schnörr. Statistical shape knowledge in variational motion segmentation. *Image and Vision Computing*, 21(1):77–86, January 2003.

[54] B. Dacorogna. *Direct methods in the calculus of variations*. Springer-Verlag, New York, 1989.

[55] Ingrid Daubechies, Michel Defrise, and Christine de Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, August 2004.

[56] E. De Giorgi, M. Carriero, and A. Leaci. Existence theorem for a minimum problem with free discontinuity set. *Arch. Rat. Mech. and Anal.*, 108:195–218, 1989.

[57] G. Doretto, D. Cremers, P. Favaro, and S. Soatto. Dynamic texture segmentation. In B. Triggs and A. Zisserman, editors, *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1236–1242, Nice, Oct. 2003.

[58] M. Droske and W. Ring. A Mumford-Shah level-set approach for geometric image registration. *SIAM Journal on Applied Mathematics*, 66(6):2127–2148, 2006.

[59] M. Droske and M. Rumpf. A variational approach to non-rigid morphological registration. *SIAM Journal on Applied Mathematics*, 64(2):668–687, 2004.

[60] M. Droske and M. Rumpf. Multi scale joint segmentation and registration of image morphology. *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 29(12):2181–2194, 2007.

[61] Marc Droske, Wolfgang Ring, and Martin Rumpf. Mumford-Shah based registration: a comparison of a level set and a phase field approach. *Computing and Visualization in Science*, 12:101–114, 2009.

[62] Ivar Ekeland and Roger Téman. *Convex analysis and variational problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[63] N. El-Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby. Graph cut optimization for the Mumford-Shah model. In *Visualization, Imaging, and Image Processing - 2007*, 2007.

[64] K. R. Elder and M. Grant. Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals. *Physical Review E*, 70(5):051605–1–051605–18, November 2004.

[65] Selim Esedoḡlu and Yen-Hsi Richard Tsai. Threshold dynamics for the piecewise constant Mumford-Shah functional. *Journal of Computational Physics*, 211(1):367–384, January 2006.

[66] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 1998.

[67] Lawrence C. Evans and Ronald F. Gariepy. *Measure theory and fine properties of functions*. Studies in Advanced Mathematics. CRC Press, Boca Raton, FL, 1992.

[68] Paolo Favaro, Martin Burger, and Stefano Soatto. Scene and motion reconstruction from defocused and motion-blurred images via anisotropic diffusion. In *Proceedings of the 8th European Conference on Computer Vision (ECCV 2004)*, volume 3021 of *Lecture Notes in Computer Science*, pages 257–269. Springer, 2004.

[69] Paolo Favaro and Stefano Soatto. A variational approach to scene reconstruction and image segmentation from motion-blur cues. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, volume 1, pages 631–637, 2004.

[70] J. Michael Fried. Multichannel image segmentation using adaptive finite elements. *Computing and Visualization in Science*, 12(3):125–135, March 2009.

[71] M. E. Gurtin. *An Introduction to Continuum Mechanics*. Academic Press, 1981.

[72] Jingfeng Han. *One-to-one Edge Based Registration and Segmentation Based Validations in Hybrid Imaging*. PhD thesis, Friedrich-Alexander-Universität Erlangen, 2009.

[73] Jingfeng Han, Benjamin Berkels, Marc Droske, Joachim Hornegger, Martin Rumpf, Carlo Schaller, Jasmin Scorzin, and Horst Urbach. Mumford-Shah model for one-to-one edge matching. *IEEE Transactions on Image Processing*, 16(11):2720–2732, 2007.

[74] Jingfeng Han, Benjamin Berkels, Martin Rumpf, Joachim Hornegger, Marc Droske, Michael Fried, Jasmin Scorzin, and Carlo Schaller. A variational framework for joint image registration, denoising and edge detection. In *Bildverarbeitung für die Medizin 2006*, pages 246–250. Springer, March 2006.

[75] L. He, A. Marquina, and S. J. Osher. Blind deconvolution using TV regularization and Bregman iteration. *International Journal of Imaging Systems and Technology*, 15(1):74–83, 2005.

[76] Matthias Heiler and Christoph Schnörr. Natural image statistics for natural image segmentation. *International Journal of Computer Vision*, 63(1):5–19, 2005.

[77] Lars Hörmander. *The Analysis of Linear Partial Differential Operators I*. Springer, Berlin / New York, 2nd edition, 1990.

[78] B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.

[79] Jiaya Jia. Single image motion deblurring using transparency. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[80] H. J. Johnson and G. E. Christensen. Consistent landmark and intensity-based image registration. *IEEE Transactions on Medical Imaging*, 21(5):450–461, 2002.

[81] T. Kapur, L. Yezzi, and L. Zöllei. A variational framework for joint segmentation and registration. *IEEE CVPR - MMBIA*, pages 44–51, 2001.

[82] Hezerul Abdul Karim, Michel Bister, and Mohammad Umar Siddiqi. Multiresolution motion estimation for low-rate video frame interpolation. *Journal on Applied Signal Processing*, 11:1708–1720, 2004.

[83] Yeon-Ho Kim, Aleix M. Martinez, and Avi C. Kak. Robust motion estimation under varying illumination. *Image and Vision Computing*, 23(4):365–375, 2005.

[84] Wayne E. King, Geoffrey H. Campbell, Stephen M. Foiles, Dov Cohen, and Kenneth M. Hanson. Quantitative HREM observation of the $\Sigma 11(113)/[\bar{1}00]$ grain-boundary structure in aluminium and comparison with atomistic simulation. *Journal of Microscopy*, 190(1):131–143, 1998.

[85] Kalin Kolev, Maria Klodt, Thomas Brox, and Daniel Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, August 2009.

[86] P. Kornprobst, R. Deriche, and G. Aubert. Image sequence analysis via partial differential equations. *Journal of Mathematical Imaging and Vision*, 11:5–26, 1999.

[87] R. Krishnamurthy, J. W. Woods, and P. Moulin. Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:713–726, 1999.

[88] L. Kubecka and J. Jan. Registration of bimodal retinal images - improving modifications. In *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2004)*, volume 3, pages 1695–1698, 2004.

[89] Sridhar Lakshmanan and Haluk Derin. Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):799–813, 1989.

[90] Sang Hwa Lee, Nam Su Moon, and Choong Woong Lee. Recovery of blurred video signals using iterative image restoration combined with motion estimation. In *Proceedings of the 1997 IEEE International Conference on Image Processing (ICIP '97)*, volume 1, pages 755–758, 1997.

[91] Anat Levin. Blind motion deblurring using image statistics. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

[92] J. Lie, M. Lysaker, and X.-C. Tai. A binary level set model and some applications to Mumford-Shah image segmentation. *IEEE Transactions on Image Processing*, 15(5):1171–1181, 2006.

[93] B. S. Manjunath and Rama Chellappa. Unsupervised texture segmentation using Markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):478–482, 1991.

[94] S. Marsland, C. J. Twining, and C. J. Taylor. Groupwise non–rigid registration using polyharmonic clamped–plate splines. In R. E. Ellis and T. M. Peters, editors, *Medical Image Computing and Computer–Assisted Intervention, MICCAI*, volume 2879 of *LNCS*, pages 771–779, 2003.

[95] H. Maurer and J. Zowe. First and second-order necessary and sufficient optimality conditions for infinite-dimensional programming problems. *Mathematical Programming*, 16(1):98–110, December 1979.

[96] B. Merriman, J. K. Bence, and S. J. Osher. Diffusion generated motion by mean curvature. CAM Report 92-18, University of California Los Angeles, April 1992.

[97] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, volume 22 of *University Lecture Series*. AMS, 2001.

[98] Jan Modersitzki. *Numerical Methods for Image Registration*. Oxford University Press, 2004.

[99] Jean-Michel Morel and Sergio Solimini. *Variational methods in image segmentation*. Progress in Nonlinear Differential Equations and their Applications, 14. Birkhäuser Boston Inc., Boston, MA, 1995.

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

[101] Yao Nie and Kai-Kuang Ma. Adaptive rood pattern search for fast block-matching motion estimation. *IEEE Transactions on Image Processing*, 11:1442–1449, 2002.

[102] Mila Nikolova, Selim Esedoğlu, and Tony F. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.

[103] T. Nir, R. Kimmel, and A. Bruckstein. Variational approach for joint optic-flow computation and video restoration. Technical report, Department of Computer Science, Technion - Israel Institute of Technology, Technion City, Haifa 32000, Israel, 2005.

[104] S. J. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, New York, 2002.

[105] 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.

[106] Nikos Paragios and Rachid Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97(3):259–282, 2005.

[107] A. Rav-Acha and S. Peleg. Restoration of multiple images with motion blur in differentdirections. In *Proceedings of the fifth IEEE Workshop on Applications of Computer Vision*, pages 22–28, 2000.

[108] P. Rogelj and S. Kovačič. Symmetric image registration. *Medical Image Analysis*, 10(3):484–493, 2006.

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

[110] D. Rueckert, A. F. Frangi, and J. A. Schnabel. Automatic construction of 3-d statistical deformation models of the brain using nonrigid registration. *IEEE Transactions on Medical Imaging*, 22(8):1014–1025, 2003.

[111] Berta Sandberg, Tony Chan, and Luminita Vese. A level-set and Gabor-based active contour algorithm for segmenting textured images. CAM Report 02-39, University of California Los Angeles, July 2002.

[112] Robert Schaback and Helmut Werner. *Numerische Mathematik*. Springer, 4th edition, 1992.

[113] O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision*, 12(1):43–63, 2000.

[114] D. Schryvers et al. Measuring strain fields and concentration gradients around Ni_4Ti_3 precipitates. *Materials science and engineering A: structural materials properties microstructure and processing. Special Issue*, 438:485–488, 2006.

[115] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[116] Jianhong Shen. Γ-convergence approximation to piecewise constant Mumford-Shah segmentation. In *Proceedings of the 7th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS 2005)*, volume 3708 of *Lecture Notes in Computer Science*, pages 499–506. Springer, 2005.

[117] Farsiu Sina, M. Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, October 2004.

[118] Y. Singh. Density-functional theory of freezing and properties of the ordered phase. *Physics Reports*, 207(6):351–444, 1991.

[119] C. O. S. Sorzano, P. Thévenaz, and M. Unser. Elastic registration of biological images using vector–spline regularization. *IEEE Transactions on Biomedical Engineering*, 52(4):652–663, 2005.

[120] G. Sundaramoorthi, A. Yezzi, and A. Mennucci. Sobolev active contours. *International Journal of Computer Vision.*, 73(3):345–366, 2007.

[121] Vidar Thomée. *Galerkin Finite Element Methods for Parabolic Problems*, volume 25 of *Springer Series in Computational Mathematics*. Springer, Berlin, 2nd edition, 2006.

[122] Gozde Unal, Greg Slabaugh, Anthony Yezzi, and Jason Tyan. Joint segmentation and non-rigid registration without shape priors. Technical Report SCR-04-TR-7495, Siemens Corporate Research, 2004.

[123] Markus Unger, Thomas Pock, Werner Trobin, Cremers Daniel, and Horst Bischof. TVSeg - Interactive total variation based image segmentation. In *British Machine Vision Conference (BMVC)*, Leeds, UK, September 2008.

[124] Michael Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, November 1995.

[125] Luminita Vese and Tony F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, December 2002.

[126] Luminita Vese and Stanley Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19(1-3):553–572, December 2003.

[127] P. Viola and W. M. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[128] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.

[129] J. Weickert, A. Bruhn, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[130] Jinchao Xu. *Theory of Multilevel Methods*. PhD dissertation, Cornell University, May 1989.

[131] Eberhard Zeidler. *Applied functional analysis: main principles and their applications*, volume 109 of *Applied Mathematical Sciences*. Springer, 1995.