



UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS
AND IMAGE COMPUTING

Regularisierung von Deformationsfeldern mit Totaler Variation

Regularization of Deformations with Total Variation

Bachelorarbeit

im Rahmen des Studiengangs
Mathematik in Medizin und Lebenswissenschaften
der Universität zu Lübeck

vorgelegt von

Manuela Kastner

ausgegeben und betreut von

Prof. Dr. Jan Modersitzki
Institute of Mathematics and Image Computing

mit Unterstützung von

Dr. Martin Benning
Department of Applied Mathematics and Theoretical Physics
Dipl.Math. Constantin Sandmann und Thomas Polzin, M.Sc.
Institute of Mathematics and Image Computing

Lübeck, den 16. September 2015

Danksagung

An dieser Stelle möchte ich mich bei den Menschen bedanken, die diese Arbeit ermöglicht haben. Ein großes Dankeschön geht an meine Betreuer Martin, Constantin und Thomas, die mir stets mit Rat und Tat zur Seite standen und immer ein offenes Ohr für meine Fragen hatten. Danke an Herrn Modersitzki für die Ausgabe des Themas und die hilfreichen Anmerkungen. An meine Korrekturleserinnen Sandra und meine Schwester Cristina ein herzliches Danke! Zuletzt noch ein Danke an die MIC-Mitarbeiter für die freundliche Atmosphäre am Institut und an meine Kommilitonen im MIC-Pool, die stets den Tag erheitert haben.

Zusammenfassung

In der Bildregistrierung, die unter anderem in der Medizin Anwendung findet, verwendet man für unterschiedliche Deformationstypen unterschiedliche Regularisierer. Diese Bachelorarbeit behandelt eine Methode zur Regularisierung von Deformationsfeldern mit Hilfe der Totalen Variation und liefert den dazu benötigten mathematischen Hintergrund, die Grundlagen der Bildregistrierung und die numerischen Lösungsverfahren. Zusätzlich stellt sie die entsprechende Implementierung und die entstandenen Ergebnisse bereit. Die Totale Variation erlaubt Unstetigkeiten im Deformationsfeld und ist somit für die sogenannte *sliding motion* geeignet, das beispielsweise zwischen Lungen und Rippen vorkommt. Zur numerischen Umsetzung wurde zunächst ein Primal-Dualer Hybrid-Gradient-Algorithmus implementiert, der auf das Rudin-Osher-Fatemi-Modell angewendet wird und in der Bildentrauschung Anwendung findet. Der Algorithmus wurde so angepasst, dass er sich auch auf Deformationsfelder anwenden lässt und eignet sich somit für Deformationsmodelle mit Unstetigkeiten. Die Implementierung liefert nicht nur für akademische Beispiele überzeugende Ergebnisse, sondern auch für medizinische Bilder.

Abstract

In (medical) image registration different regularizers are used for different deformation models. This bachelor thesis discusses a method to regularize transformations with total variation and provides the required mathematical background, the concepts of image registration and the numerical solutions. Furthermore, it provides the respective implementation and the produced results. The total variation permits discontinuities in the deformation and thus is suitable for the so called *sliding motion* that occurs between ribs and lungs, for example. For the numerical realization we first implemented a primal-dual hybrid-gradient-algorithm that is applied to the Rudin-Osher-Fatemi model and, therefore, applicable in image denoising. We then adjusted the algorithm in such a way that it is suitable for transformations. Hence it can be used for deformations with discontinuities. Not only does the implementation provide convincing results for academic examples but also for medical images.

Eidesstattliche Erklärung

Ich versichere an Eides statt, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Lübeck,

Inhaltsverzeichnis

1	Einleitung	1
2	Mathematische Grundlagen	3
2.1	Konvexität	3
2.2	Optimierung	3
2.3	Dualität	6
3	Die Totale Variation in der Bildverarbeitung	9
3.1	Die Totale Variation	9
3.2	Das Rudin-Osher-Fatemi-Modell	12
4	Grundlagen der Bildregistrierung	13
4.1	Definition eines Bildes	13
4.2	Bildregistrierungsansätze	13
5	Numerische Lösungsverfahren	16
5.1	Die Primal-Duale Hybrid-Gradient-Methode	16
5.2	Das Rudin-Osher-Fatemi-Modell mit der Primal-Dualen Hybrid-Gradient- Methode als Entrauschungsmethode für Bilder	18
5.3	Das iterativ regularisierte Gauß-Newton-Verfahren	20
5.4	Die Totale Variation als Regularisierer in der Bildregistrierung	21
5.5	Gitter und Interpolation	23
5.6	Der Multilevel-Ansatz	23
6	Numerische Ergebnisse	25
6.1	Abbruchkriterium	25
6.2	Ergebnisse der Primal-Dualen Hybrid-Gradient- Methode	25
6.3	Ergebnisse der Bildregistrierung	25
6.4	Vergleich zu elastischem Regularisierer	29
6.5	Anwendung auf medizinische Bilder	31
6.6	Fazit	33
	Literaturverzeichnis	34

1 Einleitung

Die Bildregistrierung ist eine der schwierigsten Aufgaben in der Bildverarbeitung [14]. Die Problemstellung klingt leicht: Sind zwei Bilder gegeben, die zu unterschiedlichen Zeitpunkten oder auf unterschiedliche Weise aufgenommen wurden, so finde eine plausible Transformation, die das erste Bild so transformiert, dass es dem zweiten Bild ähnelt [14, 15]. Viele Anwendungsgebiete sind auf die Lösung dieses Problems angewiesen, wobei eines der wichtigsten die Medizin ist. Man ist an Krankheitsverläufen, Tumorenbildung und vielem mehr interessiert, das sich mit modernen medizinischen Geräten abbilden lässt. Dies reicht von Computertomographie- bis zu Ultraschallaufnahmen. Auch andere Bereiche, wie zum Beispiel die Kunst, Biologie, Chemie, Kriminologie und Genetik bedienen sich der Bildregistrierung [14].

Eine der Schwierigkeiten bei der Lösung des Problems ist die Wahl eines passenden Regularisierers. Dieser stellt sicher, dass eine gefundene Transformation *plausibel* ist [15]. Dies bedeutet, dass er nicht jede beliebige Transformation zulässt [15]. Das ist vor allem dann wichtig, wenn man jedes Pixel beliebig und unabhängig von seinen Nachbarn transformiert. Ein möglicher Regularisierer ist die Totale Variation. Sie wurde erstmals 1992 von Rudin, Osher und Fatemi [21] zum Entrauschen von Bildern benutzt. Die Totale Variation wird seitdem als Regularisierer in vielerlei Anwendungen genutzt; so auch in der Ausarbeitung von Chambolle und Pock [7], die einen äußerst effizienten Algorithmus liefert, der sich auf viele Modelle der Bildoptimierung anwenden lässt. Da die Totale Variation nicht differenzierbar ist, besteht ein großes Interesse daran, Algorithmen zu entwickeln, die Minimierungsprobleme, die auf Totaler Variation basieren, lösen.

In den letzten Jahren wurden sehr viele Methoden und Ideen entwickelt, welche die Totale Variation als Regularisierer nutzen. Einen nicht ganz vollständigen Überblick über die Totale Variation und ihre Anwendung kann man in [4] finden.

Eine neuere Methode sieht es vor, die isotrope und die anisotrope Totale Variation in Kombination zu nutzen wie es in [12] beschrieben wird, statt nur eine der beiden zu nutzen. Dabei wird die gewichtete Differenz der beiden Totalen Variationen gebildet. Diese Methode soll die klassische Verwendung der Totalen Variation, bei der nur eine der beiden Variationen verwendet wird, verbessern.

Das Ziel dieser Arbeit ist es, die Totale Variation als Regularisierer für die Bildregistrierung zu nutzen und zu implementieren. Diese Art von Regularisierer soll für Bilder funktionieren, bei denen ein Objekt an ein anderes Objekt entlang gleitet, das unverändert bleiben soll. Dabei wird zunächst ein Algorithmus betrachtet, der mit Hilfe des Rudin-Osher-Fatemi-Modells Bilder entzaubert. Schließlich werden wir den Algorithmus so erweitern, dass wir ihn auf die Bildregistrierung anwenden können. Dazu benötigen wir zunächst einige Grundlagen, die in Kapitel 2 beschrieben werden. Darin befassen wir uns zunächst mit dem Begriff der Konvexität und mit den

Grundlagen der Optimierung, um mit dem Begriff des Minimierungsproblems vertraut zu werden. Wir befassen uns dann mit der Dualität, die es uns ermöglicht, ein Sattelpunktproblem aufzustellen. In Kapitel 3 erfahren wir, was die Totale Variation ist und weshalb sie in der Bildverarbeitung Anwendung findet. Im Anschluss dazu kommen wir auch auf das Rudin-Osher-Fatemi-Modell zu sprechen. Weiter geht es in Kapitel 4 mit den Grundlagen der Bildregistrierung. Darin erfahren wir genauer, was ein Regularisierer macht und wann eine Transformation plausibel ist. In Kapitel 5 befassen wir uns mit den numerischen Lösungsverfahren. Hier wird auch die Primal-Duale Hybrid-Gradient-Methode von Chambolle und Pock vorgestellt. Diese ist neben den schon genannten Vorteilen auch noch schnell und einfach zu implementieren. Weitere Themen des Kapitels sind das Gauß-Newton-Verfahren und wie die Totale Variation numerisch in die Bildregistrierung eingebaut wird. Zu guter Letzt sehen wir uns in Kapitel 6 die Ergebnisse der Implementierung an und bewerten diese mit ein paar abschließenden Worten im Fazit.

2 Mathematische Grundlagen

Wir werden nun einige notwendige Grundlagen betrachten, die wir für das Verständnis dieser Arbeit benötigen. Dabei befassen wir uns zunächst mit dem Begriff der Konvexität, bevor wir zu den Grundlagen der Optimierung und der damit verbundenen Dualität übergehen.

2.1 Konvexität

Konvexität ist ein wichtiger Bestandteil der Optimierung. Man ist an konvexen Funktionen interessiert, da diese globale Minima besitzen. Zwar ist unsere Zielfunktion in Kapitel 5 weder konvex noch linear doch wir können sie linearisieren, sodass jeder linearisierte Schritt konvex wird. Wir schauen uns deshalb vorab die Definition der Konvexität an.

Definition 1. Eine Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ heißt konvex, wenn für alle $x_1, x_2 \in \mathbb{R}^n$ und für alle λ mit $\lambda \in [0, 1]$ die Ungleichung

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

gilt [9]. Ist die linke Seite für $\lambda \in (0, 1)$ sogar echt kleiner als die rechte Seite, so ist f strikt konvex. f ist entsprechend konkav, wenn $-f$ konvex ist.

Die Definition sagt aus, dass der Graph von f im Intervall $[x_1, x_2]$ unter der Sekante, die durch die Punkte $(x_1, f(x_1))$ und $(x_2, f(x_2))$ verläuft, liegen muss wie es in Abbildung 1 zu sehen ist.

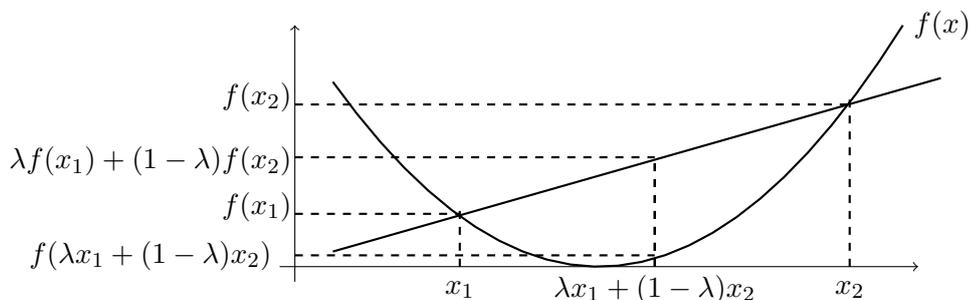


Abbildung 1: Die Funktion f ist konvex, da ihr Graph für alle x_1, x_2 unter der Sekante durch x_1 und x_2 liegt.

2.2 Optimierung

Die Optimierung behandelt Methoden zur Bestimmung des Minimums einer Funktion. Da viele Methoden der Bildverarbeitung, insbesondere auch der Bildregis-

trierung, auf der Lösung solcher Minimierungsprobleme beruhen und wir an solchen Lösungen interessiert sind, setzen wir uns nun mit den Grundlagen der Optimierung auseinander.

Man nennt die zu minimierende Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ in diesem Fall Zielfunktion. f soll dabei eigentlich, schwach unterhalbstetig und level-beschränkt sein, damit ein Minimum existiert. Wir schauen uns deshalb die Bedeutung dieser Begriffe an.

Definition 2 ([19, s.24]). Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$, dann gilt

$$f \text{ ist eigentlich} :\Leftrightarrow \text{dom} f \neq \emptyset \text{ und } f(x) > -\infty \forall x \in \mathbb{R}^n$$

mit $\text{dom} f := \{x \in \mathbb{R}^n \mid f(x) < +\infty\}$, der Definitionsmenge von f .

Nun betrachten wir die Definition der schwachen Unterhalbstetigkeit.

Definition 3 (Schwache Unterhalbstetigkeit [19, s.51]).

Für $f : \mathbb{R}^n \rightarrow \mathbb{R}$ definieren wir

$$\liminf_{x \rightarrow x_0} f(x) := \lim_{k \rightarrow 0} \inf_{\|x - x_0\|_2 \leq k} f(x).$$

f ist schwach unterhalbstetig in $x_0 \in \mathbb{R}^n$ genau dann, wenn

$$f(x_0) \leq \liminf_{x \rightarrow x_0} f(x).$$

f ist schwach unterhalbstetig auf ganz \mathbb{R}^n genau dann, wenn f in jedem $x_0 \in \mathbb{R}^n$ schwach unterhalbstetig ist.

Jetzt folgt noch die Definition der Level-Beschränktheit.

Definition 4. [20, Kapitel 1] Eine Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ heißt level-beschränkt, wenn die Sublevelmenge

$$\text{lev}_{\leq \alpha} := \{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}$$

für alle $\alpha \in \mathbb{R}$ beschränkt ist.

Mit den letzten drei Definitionen kann sichergestellt werden, dass Minima existieren. Der folgende Satz liefert deren Existenz.

Satz 5. [20, Kapitel 1] Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$ schwach unterhalbstetig, level-beschränkt und eigentlich. Dann ist

$$\inf_x f(x) \in (-\infty, +\infty)$$

und die Menge $\arg \min f$ ist nicht leer.

Für den Beweis des Satzes und weitere Informationen zur konvexen Optimierung verweisen wir auf [20] und [19].

Wir haben nun alle nötigen Zutaten, um uns auf die Minimierung von f zu konzentrieren. Ist ein Minimum von f gefunden, so nennt man die Stelle $x^* \in \mathbb{R}^n$ an der f das Minimum annimmt, einen Minimierer. Man kann zwischen zwei Arten

der Optimierung unterscheiden. Es gibt zum einen die unrestringierte Optimierung, die keine Nebenbedingung an die Minimierung stellt. Dann schreibt man kurz

$$\min_{x \in \mathbb{R}^n} f(x). \quad (2.1)$$

Zum anderen existiert die restringierte Optimierung, das heißt der Minimierer muss noch zusätzliche Nebenbedingung erfüllen. Diese Nebenbedingungen werden in die Funktionen $g_i : \mathbb{R}^m \rightarrow \mathbb{R}$, $h_j : \mathbb{R}^l \rightarrow \mathbb{R}$ eingebunden. Erfüllt ein x alle Nebenbedingungen, so nennt man x eine zulässige Lösung. Das Minimierungsproblem lautet nun

$$\min_{x \in \mathbb{R}^n} f(x) \text{ so dass (kurz : s.d.) } \begin{cases} g_i(x) \leq 0 \\ h_j(x) = 0. \end{cases} \quad (2.2)$$

mit $i = 1, \dots, m$ und $j = 1, \dots, l$. Man nennt (2.1) und (2.2) auch mathematische Programme. Sie sind konvex, wenn f und g_i für alle $i = 1, \dots, m$ konvex sind. Wir betrachten ab jetzt nur noch unrestringierte Probleme für konvexe Funktionen f . Man kann noch zusätzlich zwischen globalen und lokalen Minimierern unterscheiden.

Definition 6 (Globale und lokale Minimierer [15]).

x^* ist ein globaler Minimierer von (2.1), wenn für alle $x \in \mathbb{R}^n$ gilt, dass $f(x^*) \leq f(x)$.
 x^* ist ein lokaler Minimierer von (2.1), wenn es eine hinreichend kleine Umgebung U^* um x^* gibt, sodass für alle $x \in U^*$ gilt, dass $f(x^*) \leq f(x)$.

Ist f konvex, so ist jedes lokale Minimum von f auch ein globales Minimum. Um einen Minimierer zu finden, wird häufig die erste und, wenn f nicht konvex ist, auch die zweite Ableitung von f benötigt. Da unsere Zielfunktion, die wir in Kapitel 5 aufstellen, nicht differenzierbar ist, müssen wir auf eine andere Weise an die Lösung des Problems herangehen. Hierfür beschreiben wir in Kapitel 5 das Gauß-Newton-Verfahren und die Armijo-Liniensuche und führen nachfolgend das Subdifferential ein.

Das Subdifferential stellt die erste Ableitung dar und wir erhalten analog zur Optimalitätsbedingung

$$f'(x) \stackrel{!}{=} 0$$

bezüglich der Ableitung die Optimalitätsbedingung bezüglich des Subdifferentials in Satz 10.

Das Subdifferential verallgemeinert den Begriff der Ableitung für konvexe und nicht-differenzierbare Funktionen. Die Betragsfunktion $f(x) = |x|$, $x \in \mathbb{R}$ ist beispielsweise an der Stelle $x = 0$ nicht differenzierbar, man kann jedoch das Subdifferential der Funktion bestimmen [24].

Definition 7. Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$ gegeben. Das Subdifferential $\partial f(x_0)$ an der Stelle x_0 besteht aus allen $p \in \mathbb{R}$ für die

$$f(x) \geq f(x_0) + \langle p, x - x_0 \rangle$$

gilt [24]. Das bedeutet, dass der Graph von f für alle x über den Graphen der Gerade durch $(x_0, f(x_0))$ mit Steigung p liegt. Die Elemente von p heißen Subgradienten von $f(x)$ an der Stelle x_0 .

Das Subdifferential $\partial f(x)$ ist für $f(x) = |x|$ also

$$\partial f(x) = \begin{cases} \{1\}, & x > 0 \\ [-1, 1], & x = 0 \\ \{-1\}, & x < 0, \end{cases}$$

denn für $x_0 = 0$ gilt

$$f(x) \geq 0 + \langle p, x - 0 \rangle \Leftrightarrow |x| \geq \langle p, x \rangle$$

genau dann, wenn $-1 \leq p \leq 1$ gilt. Selbiges kann man für $x_0 > 0$ und $x_0 < 0$ berechnen.

2.3 Dualität

Eine Klasse von Optimierungsproblemen, die wir genauer betrachten werden, sind primal-duale Sattelpunktprobleme. Wir beschäftigen uns mit diesen, da es nicht immer möglich ist (2.1) direkt zu bestimmen. Man minimiert dann über die primale Variable $x \in \mathbb{R}^n$ und maximiert über eine duale Variable $y \in \mathbb{R}^k$. Dieses Programm wird numerisch iterativ gelöst, indem in jedem Schritt das duale Programm aufgerufen wird, um die primale Lösung zu verbessern [10]. Diese Art von Programm wird in [7] präsentiert, worauf wir in Kapitel 5 näher eingehen werden. Wir benötigen dafür die sogenannte Fenchel-Dualität und gehen aufgrund dessen zunächst auf die Definition der Fenchel-Konjugierten ein.

Definition 8. Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Die Funktion $f^* : \mathbb{R}^n \rightarrow \mathbb{R}$ mit

$$f^*(y) = \sup_{x \in \text{dom} f} (y^\top x - f(x))$$

ist die konvexe Fenchel-Konjugierte der Funktion f [2].

f^* ist stets konvex, unabhängig davon, ob f konvex ist oder nicht. Falls f konvex ist, gilt zudem $f^{**} = f$. Die Fenchel-Konjugierte wird genutzt, um das Duale eines primalen Minimierungsproblems aufzustellen. Aus der Definition erhält man zusätzlich die Fenchelungleichung

$$f(x) + f^*(y) \geq x^\top y.$$

Beispiel 9. Wir wollen nun die Fenchel-Konjugierte von $f(x) = e^x$ mit $f : \mathbb{R} \rightarrow \mathbb{R}$ betrachten. Es gilt

$$f^*(y) = \sup_{x \in \text{dom} f} (yx - f(x)) = \sup_{x \in \text{dom} f} (yx - e^x) = \begin{cases} y \ln(y) - y & y > 0 \\ 0 & y = 0 \\ +\infty, & y < 0 \end{cases}$$

mit $f^* : \mathbb{R} \rightarrow \mathbb{R}$. Wir sehen die Funktion und ihre Fenchelkonjugierte in Abbildung 2.

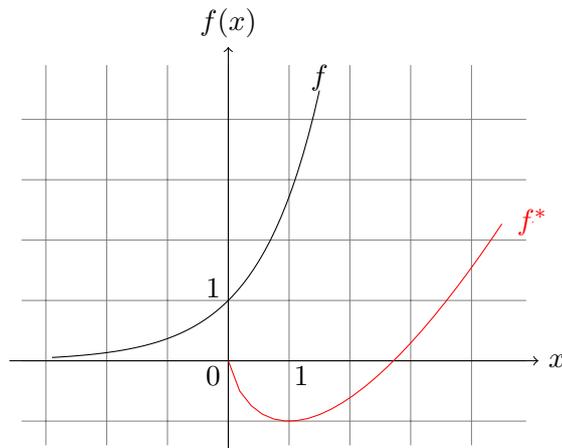


Abbildung 2: Die Exponentialfunktion f und ihre Fenchelkonjugierte f^* .

Wir wenden im Folgenden die Fenchel-Konjugierte auf konvexe Minimierungsprobleme an und schreiben (2.1) zu

$$\min_{x \in X} G(x) + F(Kx) \quad (\text{MP})$$

um. $G : X \rightarrow [0, +\infty]$ und $F : Y \rightarrow [0, +\infty]$ sind eigentliche, konvexe und schwach unterhalbstetige Funktionen auf den reellen Banachräumen X und Y . $K : X \rightarrow Y$ ist ein linearer Operator.

Wir erhalten ein Sattelpunktproblem, indem wir die Fenchel-Konjugierte von $F(Kx)$ bilden, also hat man

$$F^*(y) = \sup_x \langle Kx, y \rangle - F(Kx)$$

mit $\langle \cdot, \cdot \rangle$ als Standardskalarprodukt. Man beachte, dass $(Kx)^\top y = \langle Kx, y \rangle$ gilt. Durch die Fenchelungleichung erhalten wird

$$F(Kx) \geq \langle Kx, y \rangle - F^*(y)$$

und somit

$$F(Kx) \geq \sup_y \langle Kx, y \rangle - F^*(y).$$

Wenn wir nun annehmen, dass das Supremum angenommen wird, erhält man das Sattelpunktproblem

$$\min_{x \in X} \max_{y \in Y} J(x, y) \quad (\text{SPP})$$

mit $J(x, y) = \langle Kx, y \rangle + G(x) - F^*(y)$.

Wir betrachten im Folgenden die allgemeine Optimalitätsbedingung für eine eigentliche und konvexe Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Satz 10. [3, s.272] Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$ eigentlich und konvex. Dann gilt

$$x \in \arg \min f \Leftrightarrow 0 \in \partial f(x).$$

Beweis.

$$\begin{aligned} & 0 \in \partial f(x) \\ \Leftrightarrow & f(x) + \langle 0, y - x \rangle \leq f(y) \quad \forall y \in \mathbb{R}^n \\ \Leftrightarrow & f(x) \leq f(y) \quad \forall y \in \mathbb{R}^n \\ \Leftrightarrow & x \in \arg \min f. \end{aligned}$$

Ein Sattelpunkt $(\hat{x}, \hat{y}) \in X \times Y$ löst (SPP) genau dann, wenn die Optimalitätsbedingungen

$$\begin{aligned} K\hat{x} & \in \partial F^*(\hat{y}) \text{ und} \\ -(K^*\hat{y}) & \in \partial G(\hat{x}) \end{aligned} \tag{2.3}$$

gelten, denn es gilt laut Satz 10

$$(0, 0) \in \partial J(x, y).$$

Den Beweis hierzu findet man in [3] in Korollar 6.70. Mit den Optimalitätsbedingungen (2.3) erhält man die Lösung des Sattelpunktproblem (SPP), die durch den Resolventenoperator

$$x = (I + \tau \partial G)^{-1}(y) = \arg \min_x \left\{ \frac{\|x - y\|^2}{2\tau} + G(x) \right\}, \tag{2.4}$$

in diesem Fall bezüglich G , gegeben ist. Diesen werden wir für die numerische Umsetzung in Kapitel 5 noch nutzen.

3 Die Totale Variation in der Bildverarbeitung

In diesem Abschnitt beschäftigen wir uns mit der Totalen Variation, die in vielen Gebieten der Bildverarbeitung Anwendung findet. Ein Hauptgebiet davon ist die Bildentrauschung. Rauschen ist im Allgemeinen ein zufälliger Störfaktor, den man entfernen möchte, während das Bild möglichst viele Informationen beibehalten soll. Die Totale Variation dient dabei als Regularisierer. Anwendung in der Bildentrauschung fand sie erstmals 1992 als Rudin, Osher und Fatemi das nach ihnen benannte ROF-Modell einführten. Da wir diesen Regularisierer für die Bildregistrierung einsetzen, beschäftigen wir uns nachfolgend mit der Funktionsweise der Totalen Variation und dem ROF-Modell.

3.1 Die Totale Variation

Bevor wir zur allgemeineren Definition der Totalen Variation kommen, betrachten wir die folgende Formulierung, die einfacher und zugänglicher ist. Die Totale Variation ist, für eine C^1 Funktion $u : \Omega \rightarrow \mathbb{R}$ mit $\Omega \subset \mathbb{R}^N$, $N \geq 1$ und $u \in L^1(\Omega)$, das Integral des Gradienten einer Funktion [5], also

$$\text{TV}(u) := \int_{\Omega} |\nabla u(x)| dx. \tag{3.1}$$

Für Bilder gilt $N = 2$. Diese Formulierung hat die Eigenschaft, dass die Totale Variation in u konvex, aber für unstetige Funktionen ungeeignet ist. Wir werden später die allgemeinere Definition sehen, die auch für unstetige Funktionen geeignet ist. Im Folgenden Beispiel bestimmen wir die Totale Variation einer Funktion mittels Formel (3.1).

Beispiel 11. Sei $f : \Omega \rightarrow \mathbb{R}$ mit $\Omega = [0, 2\pi]$ und $x \in \Omega$, durch

$$f(x) = \sin(x)$$

gegeben. Die totale Variation von f ist

$$\begin{aligned} \text{TV}(f) &= \int_0^{2\pi} |\sin'(x)| dx \\ &= \int_0^{\frac{\pi}{2}} |\cos(x)| dx + \int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} |\cos(x)| dx + \int_{\frac{3\pi}{2}}^{2\pi} |\cos(x)| dx \\ &= \left| \sin\left(\frac{\pi}{2}\right) - \sin(0) \right| + \left| \sin\left(\frac{3\pi}{2}\right) - \sin\left(\frac{\pi}{2}\right) \right| + \left| \sin(2\pi) - \sin\left(\frac{3\pi}{2}\right) \right| \\ &= 4. \end{aligned}$$

Wir können nun in Abbildung 3 sehen, dass die Funktionswerte von f auf dem gegebenen Intervall tatsächlich um vier variieren.

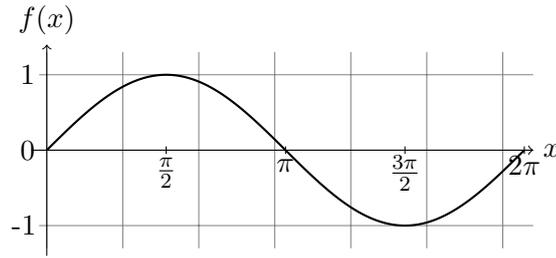


Abbildung 3: Sinusfunktion auf dem Intervall $[0, 2\pi]$

Da Bilder in der Regel Unstetigkeiten aufweisen, ist man an einer Formulierung interessiert, mit welcher man die Totale Variation von unstetigen Funktionen bestimmen kann. Man greift deshalb auf die duale Formulierung der Totalen Variation zurück. Hierfür benötigen wir zunächst die Definition der Divergenz.

Definition 12. Sei eine Funktion $f(x) = (f_1(x), \dots, f_n(x)) \in C^1(\mathbb{R}^n)$ gegeben. Dann nennt man

$$\operatorname{div} f(x) := \sum_{i=1}^n \frac{\partial f_i}{\partial x_i} \text{ mit } x \in \mathbb{R}^n$$

die Divergenz von f [22].

Die Divergenz ist der adjungierte Operator des Gradienten und für $n = 1$ die gewöhnliche Ableitung im Eindimensionalen.

Definition 13. Die duale Formulierung [5] der Totalen Variation ist durch

$$\operatorname{TV}(u) := \sup_{\xi} \left\{ \int_{\Omega} u(x) \operatorname{div} \xi(x) dx : \xi \in C_0^1(\Omega; \mathbb{R}^2), |\xi(x)| \leq 1 \forall x \in \Omega \right\} \quad (3.2)$$

gegeben, wobei C_0^1 bedeutet, dass ξ ein Mal stetig-differenzierbar und am Rand 0 ist.

Diese Formulierung der Totalen Variation ist ebenfalls konvex. Wir arbeiten im weiteren Verlauf der Arbeit mit dieser Definition. Für ein Mal differenzierbare Funktionen u stimmt diese Formulierung mit der ursprünglichen Definition überein, denn mit der Definition des Betrags einer Funktion u

$$|u| = \sup_{\|p\|_{\infty} \leq 1} up$$

und mit

$$\|u(x)\|_1 = \int_{\Omega} |u(x)| dx$$

erhält man mittels partieller Integration

$$\begin{aligned} \|\nabla u(x)\|_1 &= \sup_{\|\xi\|_\infty \leq 1} \int_{\Omega} (\nabla u(x)) \xi(x) dx \\ &= \underbrace{[u(x)\xi(x)]_{\partial\Omega}}_{=0} - \int_{\Omega} u(x)(\operatorname{div}(\xi)) dx \\ &= \int_{\Omega} u(x)(\operatorname{div}(\xi)) dx, \end{aligned}$$

da das Vorzeichen keine Rolle spielt. Wir schauen uns nun zur Veranschaulichung ein Beispiel an, bei dem die Totale Variation einer unstetigen Funktion mittels (3.2) berechnet wird.

Beispiel 14. Sei $f : \Omega \rightarrow \mathbb{R}$ mit $\Omega = [0, 1]$, durch

$$f(x) = \begin{cases} -1 & \text{für } x \in [0, \frac{1}{2}] \\ 1 & \text{für } x \in (\frac{1}{2}, 1] \end{cases}$$

gegeben. Die totale Variation ist nun

$$\begin{aligned} \operatorname{TV}(u) &= \sup_{|\xi| \leq 1} \left\{ \int_0^{\frac{1}{2}} -1 \cdot \xi'(x) dx + \int_{\frac{1}{2}}^1 1 \cdot \xi'(x) dx \right\} \\ &= \sup_{|\xi| \leq 1} \left\{ -\xi\left(\frac{1}{2}\right) + \xi(0) + \xi(1) - \xi\left(\frac{1}{2}\right) \right\} \\ &= \sup_{|\xi| \leq 1} -2\xi\left(\frac{1}{2}\right), \text{ da } \xi \text{ am Rand verschwindet} \\ &= 2, \end{aligned}$$

weil man leicht eine Funktion $\xi \in C_0^1(\Omega; \mathbb{R}^2)$ mit $\xi\left(\frac{1}{2}\right) = -1$ konstruieren kann.

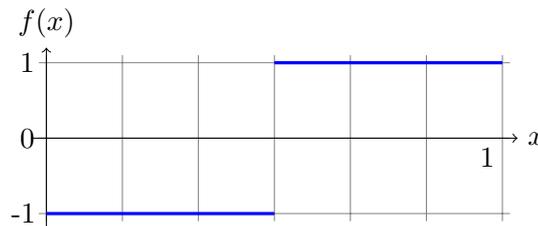


Abbildung 4: Beispielfunktion mit Variation zwei.

Die Totale Variation einer Funktion misst also wie stark sich die Funktionswerte auf einem beschränkten Gebiet Ω ändern. Wenn man die Totale Variation eines Bildes minimiert, so verringert sich das Rauschen des Bildes, um es gegebenenfalls weiter zu bearbeiten. Man hat als Bild eine Funktion $g(x, y)$ gegeben, die den Pixelwert an

x und y liefert. Dieses Bild setzt sich aus $u(x, y)$, dem entrauschten Bild und dem Rauschen $n(x, y)$ zusammen, sodass man

$$g(x, y) = u(x, y) + n(x, y).$$

erhält.

3.2 Das Rudin-Osher-Fatemi-Modell

Wir werden in Kapitel 5 das ROF-Modell mit Hilfe des Algorithmus von Chambolle und Pock implementieren, weshalb wir nun eine kurze Einführung in das Modell von Rudin, Osher und Fatemi einbringen, die erstmals die L^1 -Norm zum Entrauschen von Bildern nutzten. Man kann zwar alternativ die L^2 -Norm des Gradienten benutzen, diese erlaubt jedoch keine Unstetigkeiten in der Lösung. Für die Modellierung (3.1) hat das ROF-Modell mit

$$|\nabla u| = \sqrt{(\nabla u)_1^2 + (\nabla u)_2^2}$$

die Form

$$\min_u \lambda \underbrace{\int_{\Omega} |\nabla u(x)| dx}_{=TV(u)} + \frac{1}{2} \int_{\Omega} |u(x) - g(x)|^2 dx \quad (\text{ROF})$$

mit dem mit Rauschen behafteten Eingabebild $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ und der Lösung $u : \mathbb{R}^2 \rightarrow \mathbb{R}$. Rudin, Osher und Fatemi präsentieren die isotrope Version

$$TV(u) = \int_{\Omega} \sqrt{|u_x|^2 + |u_y|^2} \, dx dy$$

der Totalen Variation, die nicht differenzierbar ist und die wir in Kapitel 5 als Regularisierer verwenden werden. Die entsprechend anisotrope Version ist gegeben durch

$$TV(u) = \int_{\Omega} |u_x| + |u_y| \, dx dy.$$

Das ROF-Modell eignet sich deshalb so gut zur Bildentrauschung, da die Totale Variation Kanten in einem Bild erhält. Damit lassen sich unterschiedliche Gegenstände im Bild optisch deutlich voneinander trennen. Der damit einhergehende Nachteil ist, dass das entrauschte Bild comicartig wirkt. Das heißt, dass fast keine Farbverläufe mehr zu sehen sind.

4 Grundlagen der Bildregistrierung

4.1 Definition eines Bildes

Bilder können auf verschiedene Arten aufgefasst werden: Laut Chambolle [5] sind Bilder $g = (g_{i,j})_{1 \leq i,j \leq N}$ diskrete und beschränkte 2D-Signale, wobei $g_{i,j} \in [0, 1]$ oder $g_{i,j} \in \{0, \dots, 255\}$ gilt. Aubert und Kronprost [1] sagen, dass digitale Bilder durch Abtasten und Quantifizierung der wirklichen Welt erhalten werden. Dieser Vorgang hänge von der Art der Aufnahme ab. Ihre Idee dabei ist, ein Gitter auf das Bild zu legen und jedem Element im Gitter eine Nummer zuzuweisen. Diese Elemente nennt man Pixel und die Nummer kann beispielsweise ein Farbwert oder ein Grauwert sein.

Zur Charakterisierung von Bildern kann man sich deren Auflösung bedienen. Sie bezeichnet die Größe des kleinsten darstellbaren Objekts im Bild. Eine weitere besondere Eigenschaft von Bildern sind Kanten. Diese trennen in einem Bild Objekte voneinander und sind dadurch auszumachen, dass sich zwei benachbarte Pixel im Farbwert bzw. Helligkeitswert stark unterscheiden. Dieser Unterschied macht sich mathematisch in einem großen Bildgradienten bemerkbar.

Wie man sieht, gibt es unzählige Möglichkeiten Bilder zu definieren. Wir verwenden in dieser Arbeit 2D-Grauwert-Bilder, die auf einem Gebiet $\Omega \subset \mathbb{R}^2$ definiert sind und auf $\{0, \dots, 255\}$ abbilden. Die Grauwerte werden in quadratischen oder rechteckigen Pixeln in einer Matrix gespeichert.

4.2 Bildregistrierungsansätze

Wir befassen uns nachfolgend mit den Bildregistrierungsansätzen wie in [14] oder [15] beschrieben. Wir gehen von zwei Bildern aus, wobei ein Bild als Referenz und das zweite als Template dient. Diese zwei Bilder können beispielsweise Röntgen- oder Magnetresonanztomographie-Aufnahmen sein. Werden die Bilder zu unterschiedlichen Zeiten aufgenommen, so sind sie, wenn man sie überlagert, häufig nicht mehr kongruent. Dies kann vielerlei Gründe haben. Zum einen können die Unterschiede durch Bewegung zustande kommen, wenn sich beispielsweise die Lunge beim Ein- und Ausatmen ausdehnt. Zum anderen kann der zeitliche Rahmen auch größer sein und man möchte feststellen, ob sich innerhalb einer bestimmten Zeitspanne gewisse Strukturen in Organen oder generell im Körper verändert haben.

Das Ziel der Bildregistrierung ist es, ein *plausibles* Deformationsfeld $\varphi_{\text{ref}} : \Omega \rightarrow \mathbb{R}^2$, auch Transformation genannt, zu finden, sodass sich das transformierte Templatebild $T(\varphi) : \Omega \rightarrow \mathbb{R}$ und das Referenzbild $R : \Omega \rightarrow \mathbb{R}$ möglichst ähneln. Das Referenz- und Templatebild können wie in Abbildung 5 aussehen.

Um ein solches Deformationsfeld zu finden, benötigen wir noch ein Distanzmaß D , das die Ähnlichkeit zwischen R und T misst. Ferner benötigt man einen Regularisierer R , der die plausible Transformation gewährleistet, weil die Transformation

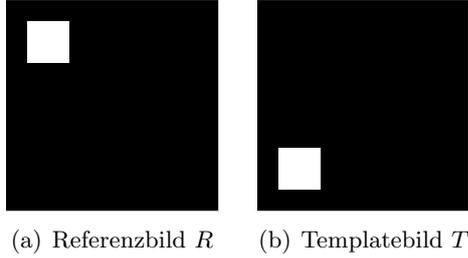


Abbildung 5: Beispiel für R und T

möglicherweise nicht mehr eindeutig ist. Plausibel bedeutet, dass die Transformation so aussieht wie man es erwarten würde. Das bedeutet beispielweise, dass die Anordnung von Gegenständen in einem Bild nicht vertauscht wird [15, s.111], also zum Beispiel die Finger an einer Hand. Es bedeutet aber auch, dass man bei unterschiedlichen Organen unterschiedliche Transformationen erwartet. Nimmt man als Beispiel die Leber, so erwartet man, dass sich die Leber elastisch verformt, also benötigt man einen Regularisierer, der elastische Transformationen zulässt. Bei einer Lunge würde man erwarten, dass im Bereich der Rippen nur sehr kleine Unterschiede in R und T zu sehen sind, die Lungen selbst jedoch, aufgrund des Ein- und Ausatmens, an den Rippen entlang transformiert werden. Hier benötigt man einen Regularisierer, der die Kante zwischen Rippe und Lunge aufrecht erhält.

Da die Bildregistrierung ein schlecht gestelltes Problem ist, ist die Regularisierung unvermeidbar. Ein gut gestelltes Problem muss die folgenden drei Kriterien nach Hadamard erfüllen, die in [13, s.8] zu finden sind.

- Das Problem hat eine Lösung.
- Die Lösung ist eindeutig.
- Die Lösung ist kontinuierlich von den Eingangsdaten abhängig.

Wird eines der drei Kriterien nicht erfüllt, so ist das Problem schlecht gestellt. Wir sehen in Abbildung 5, dass es kein eindeutiges Deformationsfeld gibt, um T in R zu überführen, denn es ist sowohl eine Drehung im Uhrzeigersinn um 270° als auch eine entgegen des Uhrzeigersinns um 90° möglich.

Wir werden nun das oben beschriebene Problem mathematisch modellieren. Eine Formulierung des Problems ist durch

$$\min_{\varphi} J(\varphi) \text{ mit} \tag{4.1}$$

$$J(\varphi) = \frac{\lambda}{2} D(T(\varphi), R) + S(\varphi - \varphi_{\text{ref}})$$

gegeben, mit dem transformierten Templatebild $T(\varphi)$, dem Referenzbild R , dem ursprünglichen Deformationsfeld $\varphi_{\text{ref}} : \Omega \rightarrow \mathbb{R}^2$ und D und T als Distanz- bzw. Regularisierungsterm. $\lambda \in \mathbb{R}$ bestimmt wie viel Einfluss Regularisierer und Distanzmaß jeweils auf das Minimum haben. Der Regularisierer sollte idealerweise ein gut

gestelltes Problem erzeugen. Die Vorstellung ist praktisch jedoch, aufgrund der fehlenden Konvexität des Datenterms, kaum umsetzbar. Es können im Allgemeinen unterschiedliche Regularisierer verwendet werden, sodass man unterschiedliche Minimierer in (4.1) erhält und es ist derjenige mit den plausibelsten Ergebnissen zu wählen.

Als Distanzmaß verwenden wir den Abstand der kleinsten Quadrate, kurz SSD [14], welche die Form

$$D(T(\varphi), R) = \frac{\lambda}{2} \int_{\Omega} (T(\varphi) - R(\varphi_{\text{ref}}))^2 d\varphi \quad (4.2)$$

hat. Die Verwendung dieses Distanzmaßes ist dann sinnvoll, wenn Template und Referenz aus der gleichen Quelle entstanden sind, das bedeutet sie sind beide Röntgenaufnahmen oder Computertomographie-Aufnahmen [15]. Unsere verwendeten Bilder erfüllen diese Voraussetzung, deshalb können wir dieses Distanzmaß verwenden. T und R sind dann ähnlich, wenn die Kanten der jeweiligen Bilder übereinander liegen [15, s.90].

Man kann zwischen zwei Arten der Bildregistrierung unterscheiden, nämlich zwischen der parametrischen und der nicht-parametrischen Bildregistrierung. Bei der parametrischen Registrierung werden nur affin-lineare Transformationen, wie Translationen, Drehungen, Scherungen und Skalierungen der Form $\varphi = A\varphi_{\text{ref}} + b$ zugelassen. In diesem Fall ist ein Regularisierer nicht zwingend notwendig, da keine beliebigen Transformationen zugelassen werden. Häufig reicht die parametrische Registrierung bei komplexeren Transformationen nicht mehr aus, um ein geeignetes Deformationsfeld zu finden. Man verwendet dafür die nicht-parametrische Registrierung, die wir für unser Modell einsetzen werden. Diese betrachtet jedes Pixel einzeln und verschiebt dieses an die passende Stelle. Jedes Pixel ist somit unabhängig von seinem Nachbarpixel, folglich wird hierbei die Transformation nicht parametrisiert. Somit wird auch klar, weshalb hier die Regularisierung wichtig ist: Besonders in diesem Fall können fälschlicherweise ähnliche Stellen im Templatebild vertauscht werden wie beispielsweise Zeige- und Ringfinger, wenn man als Template und Referenz eine Hand gegeben hat.

Als Regularisierer für die Bildregistrierung möchten wir die Totale Variation einsetzen. Dabei ist zu beachten, dass wir nicht mehr ein Bild $u : \mathbb{R}^2 \rightarrow \mathbb{R}$, sondern das Deformationsfeld $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ regularisieren. Folglich lautet die isotrope Totale Variation nun

$$\text{TV}(v) = \int_{\Omega} \sqrt{(v_x^1)^2 + (v_y^1)^2 + (v_x^2)^2 + (v_y^2)^2} dx = \| |J_v|_{\text{Fro}} \|_1$$

mit der diskreten Version

$$\|\nabla v\|_1 = \sum_{i,j} \sqrt{((\nabla v_1)_{i,j}^1)^2 + ((\nabla v_1)_{i,j}^2)^2 + ((\nabla v_2)_{i,j}^1)^2 + ((\nabla v_2)_{i,j}^2)^2},$$

wobei $|\cdot|_{\text{Fro}}$ die Frobeniusnorm und J_v die Jacobimatrix von v bezeichnet.

5 Numerische Lösungsverfahren

Wir möchten im Folgenden (SPP) numerisch lösen, um die Transformation zu erhalten, die T in R überführt. Wir verwenden dafür das Gauß-Newton-Verfahren, für das Schrittweite und Suchrichtung zu bestimmen sind. Um die Suchrichtung zu erhalten, verwenden wir die modifizierte Primal-Duale Hybrid-Gradient-Methode von Chambolle und Pock. Für die Schrittweite führen wir eine Armijo-Liniensuche durch. Wir gehen also nachfolgend auf die Primal-Duale Hybrid-Gradient-Methode ein und sehen als Beispiel wie damit das ROF-Modell zu lösen ist, um Bilder zu entrauschen. Wir erläutern dann die Armijo-Liniensuche und das Gauß-Newton-Verfahren und modifizieren schließlich die Primal-Duale Hybrid-Gradient-Methode so, dass sie für die Bildregistrierung anwendbar wird. Im letzten Unterkapitel beschäftigen wir uns noch mit den für die Bildregistrierung notwendigen Themen Gitter und Interpolation und mit dem Multilevel-Ansatz.

5.1 Die Primal-Duale Hybrid-Gradient-Methode

Wir werden im Folgenden den vorgeschlagenen Algorithmus von Chambolle und Pock [7] betrachten. Ziel des Algorithmus ist es, Bilder zu rekonstruieren. Er ist sehr flexibel, da er auf viele unterschiedliche Modelle anwendbar ist. Zudem liefert er in sehr kurzer Zeit gute Ergebnisse. Wir leiten zunächst den Algorithmus in allgemeiner Form her und wenden ihn exemplarisch zum Lösen des ROF-Modells an, bevor wir ihn zum Lösen des Registrierungsproblems anpassen. Gegeben sei das Sattelpunktproblem (SPP)

$$\min_{x \in X} \max_{y \in Y} \langle Kx, y \rangle + G(x) - F^*(y), \quad (5.1)$$

wobei $G : X \rightarrow [0, +\infty)$ und $F^* : Y \rightarrow [0, +\infty)$ konvexe, schwach unterhalbstetige und eigentliche Funktionen und $K : X \rightarrow Y$ ein linearer Operator mit induzierter Norm

$$\|K\| = \max\{\|Kx\| : x \in X, \|x\| \leq 1\}$$

ist. X, Y sind zwei endlich dimensionale Vektorräume, die mit einem Skalarprodukt $\langle \cdot, \cdot \rangle$ und einer Norm $\|\cdot\| = \langle \cdot, \cdot \rangle^{\frac{1}{2}}$ ausgestattet sind. F^* bezeichnet die in Kapitel 2 definierte Fenchelkonjugierte.

Wir geben nachfolgend eine Motivation, Algorithmus 1 zu erhalten. Dazu erinnern wir uns an die Optimalitätsbedingung

$$K\hat{x} \in \partial F^*(\hat{y}). \quad (5.2)$$

Gleichung (5.2) lässt sich umschreiben zu $K\hat{x} - \hat{p} = 0$ für $\hat{p} \in \partial F^*(\hat{y})$ mit $\hat{p} \in Y$.

Davon ausgehend erhält man

$$\begin{aligned}
K\hat{x} - \hat{p} &= 0 \\
\Leftrightarrow \sigma K\hat{x} - \sigma\hat{p} &= 0 \\
\Leftrightarrow \hat{y} + \sigma K\hat{x} - \sigma\hat{p} - \hat{y} &= 0 \\
\Leftrightarrow \hat{y} - (\hat{y} + \sigma K\hat{x}) + \sigma\hat{p} &= 0.
\end{aligned}$$

Mit $p^{n+1} \in \partial F^*(y^{n+1})$ erhält man die Approximation

$$y^{n+1} - (y^n + \sigma Kx^n) + \sigma p^{n+1} = 0.$$

Man rechnet leicht aus, dass y^{n+1}

$$\begin{aligned}
y^{n+1} &= \arg \min_y \left\{ \frac{1}{2} \|y - (y^n + \sigma Kx^n)\|_2^2 + \sigma F^*(y) \right\} \\
&= (I + \sigma \partial F^*)^{-1}(y^n + \sigma Kx^n)
\end{aligned}$$

erfüllt. Selbiges kann man auch für die Optimalitätsbedingung $-K^*\hat{y} \in \partial G(\hat{x})$ herleiten.

Den daraus resultierenden Algorithmus bezeichnet man Primal-Duale Hybrid-Gradient-Methode.

Algorithmus 1 (Primal-Duale Hybrid-Gradient-Methode). [5]

- Wähle $\tau, \sigma > 0, \theta \in [0, 1], (x^0, y^0) \in X \times Y$ und setze $\bar{x}^0 = x^0$.
- Für $n \geq 0$ Iterationen aktualisiere x^n, y^n und \bar{x}^n wie folgt:

$$\begin{cases}
y^{n+1} = (I + \sigma \partial F^*)^{-1}(y^n + \sigma K\bar{x}^n) \\
x^{n+1} = (I + \tau \partial G)^{-1}(x^n - \tau K^* y^{n+1}) \\
\bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n)
\end{cases}$$

Dabei ist $x \in X$ das Bild, das wir nach $n \geq 0$ Durchläufen erhalten möchten und $y \in Y$ ist die duale Variable. $\tau \in \mathbb{R}$ und $\sigma \in \mathbb{R}$ sind Schrittweiten. $K : X \rightarrow Y$ und $K^* : Y \rightarrow X$ sind der lineare und adjungierte Operator. Man kann zeigen, dass der Algorithmus für $\theta = 1$ konvergiert. Der Beweis ist in [7] zu finden.

Für das ROF-Modell müssen wir noch betrachten, wie G, F, K und K^* aufgebaut sind. G soll unser Distanzterm sein, der den Abstand zwischen den diskreten Bildern $g \in \mathbb{R}^2$ und $u \in \mathbb{R}^2$ misst. Dabei ist g das verrauschte Bild und u das zu entrauschende Bild. Der Regularisierungsterm ist durch F gegeben. K und K^* sind der Gradient und die Divergenz. Da wir diskrete Bilder betrachten, müssen wir den Gradienten und die Divergenz diskretisieren. Dazu benutzen wir finite Differenzen. Zur Veranschaulichung bilden wir zunächst die Ableitung eines Vektors.

Definition 15. Sei $u \in \mathbb{R}^n$ ein Vektor. Der Gradient von u ist

$$\frac{1}{h} \underbrace{\begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -1 & 1 \\ 0 & \dots & 0 & 0 & 0 \end{pmatrix}}_{:=D} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{pmatrix} = \frac{1}{h} \begin{pmatrix} u_2 - u_1 \\ u_3 - u_2 \\ \vdots \\ u_n - u_{n-1} \\ 0 \end{pmatrix} \quad (5.3)$$

mit $D \in \mathbb{R}^{n \times n}$ und $h \in \mathbb{R}$.

Möchte man nun die Ableitung einer Matrix $u \in \mathbb{R}^{n \times m}$ bilden, so erhält man

$$\nabla u = \begin{pmatrix} \nabla u_x \\ \nabla u_y \end{pmatrix} = \begin{pmatrix} D_1 u \\ u D_2 \end{pmatrix},$$

wobei D_1 wie D in Gleichung (5.3) aufgebaut ist und D_2 die Gestalt

$$D_2 = \begin{pmatrix} -1 & 0 & 0 & \dots & 0 \\ 1 & -1 & 0 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \dots & \ddots & \ddots & -1 & 0 \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix}$$

hat. Gilt $n = m$, so hat man $D_2 = D_1^\top$. Für die Divergenz von $p \in Y$ mit $p \in \mathbb{R}^{n \times m \times 2}$ rechnet man [6]

$$(\operatorname{div} p)_{ij} = \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{wenn } 1 < i < N, \\ p_{i,j}^1 & \text{wenn } i = 1, \\ -p_{i-1,j}^1 & \text{wenn } i = N, \end{cases} + \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{wenn } 1 < j < N, \\ p_{i,j}^2 & \text{wenn } j = 1, \\ -p_{i,j-1}^2 & \text{wenn } j = N. \end{cases}$$

5.2 Das Rudin-Osher-Fatemi-Modell mit der Primal-Dualen Hybrid-Gradient-Methode als Entrauschungsmethode für Bilder

Wir wenden nun Algorithmus 1 an, indem wir ihn auf das ROF-Modell übertragen. Dabei gehen wir wie in [7] vor. Wir haben als Eingabe das verrauschte Bild $g \in \mathbb{R}^2$ und wollen das entrauschte Bild $u \in \mathbb{R}^2$ erhalten. G hat die Gestalt $G(u) = \frac{\lambda}{2} \|u - g\|_2^2$ und misst die Ähnlichkeit zwischen dem Eingabebild und der Rekonstruktion. F hat die Form $F(\nabla u) = \|\nabla u\|_1$, wobei

$$\|\nabla u\|_1 = \sum_{i,j} \sqrt{((\nabla u)_{i,j}^1)^2 + ((\nabla u)_{i,j}^2)^2}$$

die diskrete Version der isotropen Totalen Variation ist [7]. Unser diskretes Minimierungsproblem ist also durch

$$\min_{u \in X} \|\nabla u\|_1 + \frac{\lambda}{2} \|u - g\|_2^2$$

gegeben. Das Ziel ist somit, den Abstand zwischen Eingabe- und Ausgabebild und die Totale Variation zu minimieren. Der Parameter $\lambda \in \mathbb{R}$ regelt den Einfluss der Totalen Variation und des Distanzmaßes auf die Lösung des Minimierungsproblems. Ist λ klein, so hat F einen großen Einfluss auf die Lösung. Das heißt, dass die Kanten des Bildes deutlich hervorgehoben werden und das Bild somit comicartig aussieht. Ist λ hingegen groß, so wird u sehr ähnlich zu g sein. Für die numerische Lösung formulieren wir (5.2) wie in Kapitel 2 um und erhalten

$$\min_{u \in X} \max_{p \in Y} -\langle u, \operatorname{div} p \rangle_X + \frac{\lambda}{2} \|u - g\|_2^2 - \delta_P(p) \quad (5.4)$$

mit $\langle u, v \rangle_X = \sum_{i,j} u_{i,j} v_{i,j}$. Wir führen die konvexe Menge $P = \{p \in Y : \|p\|_\infty \leq 1\}$ ein, wobei $p : Y \rightarrow X$ mit $p = (p^1, p^2)$ die duale Variable ist und

$$\|p\|_\infty = \max_{i,j} |p_{i,j}| \quad \text{mit} \quad |p_{i,j}| = \sqrt{(p_{i,j}^1)^2 + (p_{i,j}^2)^2}$$

gilt. δ_P bezeichnet eine Indikatorfunktion mit

$$\delta_P(p) = \begin{cases} 0 & \text{wenn } p \in P \\ +\infty & \text{wenn } p \notin P. \end{cases}$$

Wenn wir nun (5.4) und (SPP) miteinander vergleichen, so sieht man, dass $G(u) = \frac{\lambda}{2} \|u - g\|_2^2$ und $F^*(p) = \delta_P(p)$ ist. F^* ist also die Indikatorfunktion einer konvexen Menge und der Resolventenoperator (2.4) bezüglich F^* reduziert sich zu

$$p = (I + \sigma \partial F^*)^{-1}(\tilde{p}) \iff p_{i,j} = \frac{\tilde{p}_{i,j}}{\max(1, |\tilde{p}_{i,j}|)} \quad (5.5)$$

mit $\tilde{p} = y^n + \sigma K \bar{x}^n$.

Die Lösung des Resolventenoperators (2.4) lässt sich durch einfaches Differenzieren und Auflösen bestimmen. Man erhält für $\tilde{u} = u^n - \tau K^* p^{n+1}$

$$\begin{aligned} & \frac{d}{du} \frac{\|u - \tilde{u}\|_2^2}{2\tau} + \frac{\lambda}{2} \|u - g\|_2^2 \stackrel{!}{=} 0 \\ \Leftrightarrow & \frac{(u - \tilde{u})}{\tau} + \lambda(u - g) = 0 \\ \Leftrightarrow & (1 + \tau\lambda)u = \tilde{u} + \tau\lambda g \\ \Leftrightarrow & u = \frac{\tilde{u} + \tau\lambda g}{1 + \tau\lambda} \end{aligned}$$

und somit

$$u = (I + \tau \partial G)^{-1}(\tilde{u}) \iff u_{i,j} = \frac{\tilde{u}_{i,j} + \tau \lambda g_{i,j}}{1 + \tau \lambda}.$$

Wir stellen nachfolgend eine Möglichkeit vor, Minimierungsprobleme für zweimal differenzierbare Funktionen f zu lösen. Diese Methode werden wir später zur Lösung von (5.8) verwenden. Da f nicht notwendig konvex sein muss, wird uns das Verfahren ein lokales Minimum und bestenfalls ein globales Minimum liefern. Dabei nähert man sich dem Minimum schrittweise. Man erreicht dies, indem man eine Suchrichtung s_k bestimmt, die angibt, in welche Richtung man als nächstes geht. Dazu bestimmt man noch eine Schrittweite α_k , die besagt wie weit man in diese Richtung geht. Man geht also wie folgt vor [15]:

Algorithmus 2.

Setze $k = 0$ und bestimme einen Startwert x_0 . Solange die Stoppkriterien nicht erfüllt sind

- bestimme s_k an $f(x_k)$,
- bestimme α_k und
- setze $x_{k+1} = x_k + \alpha_k s_k$ und $k = k + 1$.

Ein Abstieg erfolgt dann, wenn $f(x_{k+1}) \leq f(x_k)$. Zur Bestimmung der Suchrichtung bedienen wir uns des Gauß-Newton Verfahrens. Für die Schrittweitenbestimmung verwenden wir die Armijo-Liniensuche. Dazu sei $\psi_1, \psi_2 : \mathbb{R} \rightarrow \mathbb{R}$ mit $\psi_1(\alpha_k) = f(x + \alpha_k s_k)$ und $\psi_2(\alpha_k) = f(x_k) + c\alpha_k f_k^\top s_k$ mit $c \in (0, 1/2)$, das sicherstellt, dass ein Abstieg erfolgt und meist sehr klein ist.

Algorithmus 3 (Armijo-Liniensuche).

- Setze $\alpha = 1$.
- Solange $\psi_1(\alpha) > \psi_2(\alpha)$, halbiere die Schrittweite $\alpha \leftarrow \alpha/2$.

Da unsere Zielfunktion nicht differenzierbar ist, müssen wir auf die naive Bedingung $f(x_{k+1}) \leq f(x_k)$ zurückgreifen.

5.3 Das iterativ regularisierte Gauß-Newton-Verfahren

Wir werden in Unterkapitel 5.4 die für die Bildregistrierung benötigte Zielfunktion mit der Totalen Variation als Regularisierer aufstellen. Da die Zielfunktion nichtlinear ist, verwenden wir zu deren Lösung das iterativ regularisierte Gauß-Newton-Verfahren. Zum Verständnis gehen wir zunächst auf das Gauß-Newton-Verfahren ein, welches [18] entnommen ist. Dabei hat man eine nichtlineare Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ und $E : \mathbb{R}^n \rightarrow \mathbb{R}$ mit $E(x) = \|f(x)\|_2^2$ gegeben. Wir minimieren E , also

$$\min_x E(x),$$

indem wir $f(x)$ zunächst linearisieren. Dafür nehmen wir eine Taylorapproximation vor, sodass

$$f(x) = f(x_0) + J_f(x_0)(x - x_0),$$

wobei $J_f(x_0)$ die Jacobi-Matrix von f an der Stelle x_0 , und x_0 ein Startvektor in der Nähe des Minimums ist. Das Minimierungsproblem lautet nun

$$\min_x \|f(x_0) + J_f(x_0)(x - x_0)\|_2^2. \quad (5.6)$$

Führen wir dies iterativ durch und fügen eine Liniensuche hinzu, so erhalten wird das Gauß-Newton-Verfahren zur Näherungslösung von (5.6).

Algorithmus 4 (Gauß-Newton-Verfahren).

Initialisiere x_0 . Für $k \geq 0$

- Bestimme s^k des Minimierungsproblems $\min_{s^k} \|f(x_k) + J_f(x_k)s^k\|_2^2$.
- Aktualisiere x_k , indem $x_{k+1} = x_k + \alpha_k s_k$.

s_k ist die im Kapitel 2 eingeführte Suchrichtung und α_k die Schrittweite.

Bei dem Iterativ regularisierten Gauß-Newton-Verfahren [8] hat man

$$\min_x J_k(x) = \|f(x_k) + J_f(x_k)(x - x_k) - g\|_2^2 + \lambda_k \|S(x - x_{\text{ref}})\|_2^2 \quad (5.7)$$

mit g als Referenzbild zu lösen. (5.7) sieht unserer linearisierten Zielfunktion in Unterkapitel 5.4 ähnlich mit dem Unterschied, dass die 2-Norm statt der TV-Norm verwendet wird und λ nicht konstant bleibt. Dies ist jedoch kein Problem, da wir (5.7) nicht differenzieren müssen, um die Schrittweite s_k zu erhalten, sondern Algorithmus 1 anwenden.

5.4 Die Totale Variation als Regularisierer in der Bildregistrierung

Für die Bildregistrierung haben wir nun das Minimierungsproblem

$$\min_{\varphi} f(\varphi) \text{ mit} \quad (5.8)$$

$$f(\varphi) = G(\varphi) + F(K(\varphi - \varphi_{\text{ref}})) = \frac{\lambda}{2} \|T(\varphi) - R(\varphi_{\text{ref}})\|_2^2 + TV(\varphi - \varphi_{\text{ref}})$$

gegeben. Hierbei ist φ die Transformation und $T(\varphi)$ das transformierte Templatebild. $F(K(\varphi - \varphi_{\text{ref}}))$ ist die Totale Variation der Transformation. G ist im Allgemeinen nicht konvex, deshalb verwenden wir zur Lösung von (5.8) das iterativ regularisierte Gauß-Newton-Verfahren und gehen dabei wie folgt vor: Wir linearisieren $T(\varphi)$ durch eine Taylor-Approximation, sodass wir die konvexe approximierte Zielfunktion

$$f_L(\varphi) = \frac{\lambda}{2} \|T(\varphi_0) + \nabla T(\varphi_0)(\varphi - \varphi_0) - R(\varphi_{\text{ref}})\|_2^2 + TV(\varphi - \varphi_{\text{ref}}) \quad (5.9)$$

erhalten, die wir über φ minimieren. Anschließend führen wir nacheinander die folgenden Schritte durch:

Solange das Stoppkriterium nicht erfüllt ist,

-
- minimiere $f_L(\varphi)$ mittels Algorithmus 1, um die Abstiegsrichtung $s_k = (\varphi - \varphi_0)$ zu erhalten,
 - führe eine Armijo-Liniensuche entlang s_k mit $f(\varphi)$, um die optimale Schrittweite zu bestimmen (man erhält $s_N = (\frac{1}{2})^k s_k$) und
 - aktualisiere $\varphi_0 \leftarrow \varphi_0 + s_N$.

Wir führen für jeden Linearisierungsschritt einen kompletten Zyklus des modifizierten Algorithmus 1 aus. Für die Modifikation müssen wir die Resolventenoperatoren anpassen. Wir erhalten für $G(x) = \frac{\lambda}{2} \|Ax - b\|_2^2$ mit

$$A = \nabla T(\varphi_0) \text{ und } b = -T(\varphi_0) + \nabla T(\varphi_0)\varphi_0 + R(\varphi_{\text{ref}})$$

die folgende Lösung

$$\begin{aligned} & \frac{d}{dx} \left(\frac{\|x - \tilde{y}\|^2}{2\tau} + \frac{\lambda}{2} \|Ax - b\|^2 \right) \stackrel{!}{=} 0 \\ \Leftrightarrow & \frac{(x - \tilde{y})}{\tau} + \lambda A^\top (Ax - b) = 0 \\ \Leftrightarrow & (I + \tau \lambda A^\top A)x = \tau \lambda A^\top b + \tilde{y} \\ \Leftrightarrow & x = (I + \tau \lambda A^\top A)^{-1} (\tau \lambda A^\top b + \tilde{y}). \end{aligned}$$

Das Gleichungssystem der letzten Zeile wollen wir mittels der Woodbury-Marix-Identität [17] lösen, da diese effizient ist. Die Woodbury-Gleichung hat die Gestalt

$$(B + UCV)^{-1} = B^{-1} - B^{-1}U(C^{-1} + VB^{-1}U)^{-1}VB^{-1} \quad (5.10)$$

mit $B \in \mathbb{R}^{n \times n}$, $U \in \mathbb{R}^{n \times m}$, $V \in \mathbb{R}^{m \times n}$ und $C \in \mathbb{R}^{m \times m}$. Wenn wir nun B und C als Einheitsmatrizen I_1 und I_2 auffassen, $U = A^\top$ und $C = A$, so erhält man für $(I + \tau \lambda A^\top A)^{-1}$ mit $A \in \mathbb{R}^{n \times m}$

$$\begin{aligned} (I + \tau \lambda A^\top A)^{-1} &= (I_1 + \tau \lambda A^\top I_2 A)^{-1} \\ &\stackrel{(5.10)}{=} I_1^{-1} - I_1^{-1} \tau \lambda A^\top (I_2^{-1} + A I_1^{-1} \tau \lambda A^\top)^{-1} A I_1^{-1} \\ &= I_1 - \tau \lambda A^\top (I_2 + A \tau \lambda A^\top)^{-1} A. \end{aligned}$$

Nun ist $AA^\top \in \mathbb{R}^{m \times m}$ eine Diagonalmatrix, das heißt wir müssen keine Invertierung vornehmen, um die Inverse zu lösen, sondern $D = (I_2 + \tau \lambda AA^\top)$ und $D^{-1} = 1/D$ und erhalten somit

$$(I + \tau \lambda A^\top A)^{-1} = I_1 - \tau \lambda A^\top D^{-1} A.$$

Für den Resolventenoperator (5.5) müssen wir lediglich beachten, dass sich die Dimension von \tilde{p} verdoppelt, also $\tilde{p} \in \mathbb{R}^{n \times m \times 4}$.

Nachfolgend wollen wir betrachten wie man mit einem Bild arbeitet, das durch die Transformation $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ beschrieben wird.

5.5 Gitter und Interpolation

Der nachstehende Abschnitt ist [14] entnommen. In der Bildregistrierung sind Gitter und Interpolation ein wichtiger Aspekt. Man benötigt Gitter für den sogenannten *Diskretisiere-dann-optimiere-Ansatz*, bei dem die Modellierung (4.1) zunächst diskretisiert und dann optimiert wird. Somit wird das Registrierungsproblem numerisch bestimmt, was die Implementierung in Matlab möglich macht. Üblicherweise hat das Gitter, auf dem unser Bild gegeben ist, so viele Stützstellen wie das Bild Pixel enthält. Ein mögliches Gitter ist das sogenannte zellzentrierte Gitter. Die Gitterpunkte befinden sich dabei in der Mitte eines Pixels. Durch Interpolation der Pixel erhalten wir ein kontinuierliches Bild. Dies ist wichtig, da wir das Templatebild nun auch an anderen Stellen als den Gitterpunkten auswerten können, was für das transformierte Templatebild in den meisten Fällen auch notwendig ist.

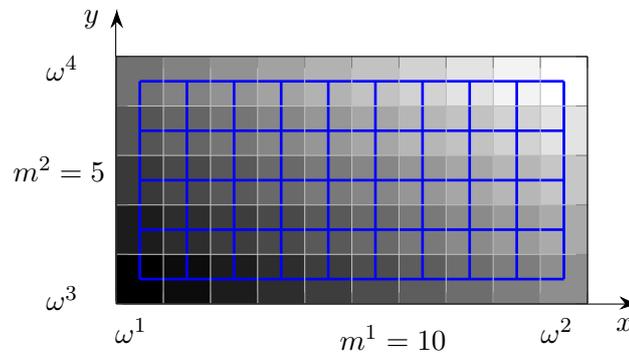


Abbildung 6: zellzentriertes Gitter

Wir sehen in Abbildung 6 ein 2D-Bild bestehend aus Grauwerten, das auf $\Omega = (\omega^1, \omega^2) \times (\omega^3, \omega^4) \subset \mathbb{R}^2$ mit $\omega^1, \omega^2, \omega^3, \omega^4 \in \mathbb{R}$ gegeben ist, wobei häufig $\omega^1, \omega^3 = 0$ gilt. In blauer Farbe ist das zellzentrierte Gitter zusehen.

Zum Interpolieren der Daten wird die lineare Interpolation genutzt, da man diese einfach und effizient berechnen kann. Der Grundgedanke der linearen Interpolation im 1D-Fall besteht darin, dass man zwei aufeinander folgende Punkte x_j, x_{j+1} durch eine Gerade verbindet. Man erhält dadurch eine stückweise lineare Funktion, die an den Stützstellen nicht differenzierbar ist. Wir verweisen für eine tiefergehende Einführung auf [16]. Für das Gitter und die Interpolation bedienen wir uns der Funktionen der FAIR-Toolbox [14].

5.6 Der Multilevel-Ansatz

Der Multilevel-Ansatz [14] wurde eingeführt, um zu verhindern, dass man bei der Optimierung in lokale Minima hängen bleibt. Die Idee ist dabei, das Templatebild

zunächst auf einer groben Auflösung zu transformieren, sodass man schon hinreichend nahe an die globale Lösung kommt, bevor auf einer feineren Auflösung die vorherige Transformation übernommen wird und nur noch kleine Transformationen notwendig sind. Ein zusätzlicher Vorteil ist die Effizienz mit der die Transformation berechnet wird, da auf der groben Auflösung die Transformation schnell berechnet ist und auf der feineren Auflösung keine großen Transformationen mehr stattfinden. Für die Multilevel Umsetzung verwenden wir ebenfalls die FAIR-Toolbox.

6 Numerische Ergebnisse

Die Abbildungen dieses Kapitels wurden in Matlab 2014b erzeugt mittels unserer Implementierungen und den Funktionen der FAIR Toolbox [14].

6.1 Abbruchkriterium

Für haben in unserer Implementierung zwei Schleifen, für die Abbruchkriterien zu wählen sind. Wir haben zwei unterschiedliche Möglichkeiten für die Abbruchbedingungen. In der inneren Schleife befindet sich der modifizierte Algorithmus 1, für den man als Abbruchkriterium

$$\frac{\|x^{n+1} - x^n\|}{x^{n+1}} < \varepsilon \text{ und } \frac{\|y^{n+1} - y^n\|}{y^{n+1}} < \varepsilon \quad (6.1)$$

mit hinreichend kleinem ε wählen kann. Die äußere Linearisierungs-Schleife brechen wir ab, wenn für die Zielfunktion (5.8)

$$f_k - f_{k+1} < \beta$$

gilt. Man bricht somit ab, wenn die Zielfunktion nur noch wenig minimiert wird. Aus Effizienzgründen kann man eine feste Anzahl an Schritten vorgeben, was bei (6.1) sinnvoll ist, da der Algorithmus so schneller durchläuft und ebenso gute Ergebnisse liefert. Eine Anzahl von 500 ist genügend.

6.2 Ergebnisse der Primal-Dualen Hybrid-Gradient-Methode

Wir sehen in Abbildung 7 den Einfluss der Totalen Variation auf die Bilder, wenn man sie mittels Algorithmus 1 mit dem ROF-Modell entrauscht. In (b) wurde ein Rauschen auf (a) gelegt und in (c) und (d) sind die entrauschten Ergebnisse zu sehen. Dabei wurde in (d) stärker regularisiert als in (c), weshalb der eben genannte comicartige Effekt auftritt. Chambolle und Pock schlagen $\tau = 0.01$ und $\sigma = 1/(\tau L^2)$ mit $L^2 = \|\nabla\|^2 = \|\text{div}\|^2 \leq 8/h^2$ vor, wobei bei uns $h = 1$ sein soll. Wir wählen $\varepsilon < 0.01$.

6.3 Ergebnisse der Bildregistrierung

Wir sehen nachfolgend Ergebnisse der Implementierung mit unterschiedlichen Parametern. Für das Referenz- und Templatebild betrachten wir (zunächst) zwei Rechtecke wie sie in Abbildung 8 zu sehen sind. Wir erwarten, dass das untere Rechteck des Templatebildes nach rechts verschoben wird, das obere Rechteck nicht transformiert wird und dass keine vertikale Transformation stattfindet. Als Startwerte für x^0 und y^0 wählen wir stets die Identität, also ein zellzentriertes Gitter bzw.



(a) Originalbild



(b) Bild mit Rauschen



(c) $\lambda = 0.03$



(d) $\lambda = 0.01$

Abbildung 7: Bildentrauschung mittels Algorithmus 1. Die unteren zwei Abbildungen sind die entrauschten Ergebnisse. Dabei ist $\sigma = 1/(8\tau)$ mit $\tau = 0.01$.

zwei zellzentrierte Gitter, da y^0 doppelt so viele Einträge wie x^0 hat. Für τ und σ gilt stets $\tau = 1/\lambda$ und $\sigma = 1/(8\tau)$. Die zusätzliche Regularisierung mithilfe des Multilevel-Ansatzes ist notwendig, da es sonst nicht möglich ist das Referenzbild zu rekonstruieren. Dies liegt daran, dass keine Transformation φ erzeugt werden kann, die T ausreichend transformiert, um R zu erhalten. In den nachfolgenden Bildern sehen wir stets den letzten Level, also die ursprüngliche Auflösung der Bilder.

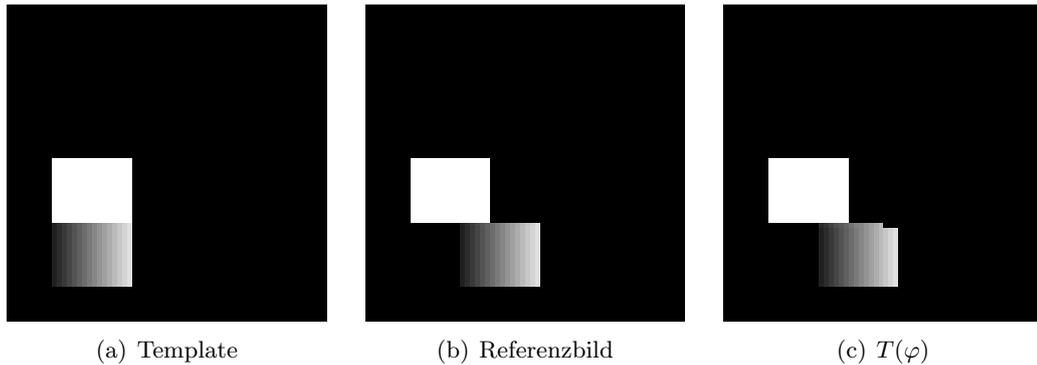


Abbildung 8: Ergebnis der Registrierung mit $\lambda = 2.1, \beta = 10^{-2}$, vier Leveln und einer festen Anzahl an Schritten für Algorithmus 1. $T(\varphi)$ ist das transformierte Templatebild.

Wir sehen in Abbildung 8, dass offensichtlich nur das untere Rechteck des Templatebildes verschoben wurde, wie es zu erwarten war. Man erkennt außerdem, dass im transformierten Templatebild ein kleiner Teil nicht transformiert wurde und somit fehlt. Dies ist bei allen Registrierungen der Fall unabhängig von der Parameterwahl. Wir gehen darauf in Unterkapitel 6.6 ein. Das dazugehörige Deformationsfeld sehen wir in Abbildung 9. Man erkennt, dass die Transformation in die entgegengesetzte Richtung zeigt. Dies ist auf den Euler-Ansatz zurückzuführen, bei dem man φ fixiert und sich anschaut von wo man da hingekommen ist (φ_{ref}) [11]. Eine ausführlichere Erklärung findet man in [11]. Ferner sieht man, dass am Rand des Bildes auch Transformationen stattfinden, obwohl dies nicht der Fall sein sollte. Wir können dies durch eine Modifikation von λ beheben und werden die Ergebnisse dazu noch sehen.

Wir sehen in Abbildung 10 einen Intensitätsplot der effektiven Transformation $\varphi - \varphi_{\text{ref}}$. In der linken Abbildung wird die erwartete Verschiebung um zehn Pixel nach rechts dargestellt. In der rechten Abbildung sieht man auch eine minimale Transformation in vertikaler Richtung. In beiden Abbildungen ist die Transformation am Rand zu erkennen.

Wir werden nun eine kleine Modifikation vornehmen, um den Einfluss des Distanz- und Regularisierungsterms besser regeln zu können. Es sei $f(\varphi) = (1 - c)D(T(\varphi) - R(\varphi_{\text{ref}})) + cS(\varphi - \varphi_{\text{ref}})$ mit $c = \lambda/(\lambda + 1)$, $c \in \mathbb{R}$. Das Ergebnis in Abbildung 11 unterscheidet sich nicht wesentlich von dem in Abbildung 8. Lediglich in der oberen rechten Ecke im unteren Rechteck scheint ein größeres Stück zu fehlen. Wenn wir uns das zugehörige Deformationsfeld in Abbildung 12 ansehen, so stellen wir fest, dass der Parameter c den gewünschten Effekt hat und der Rand somit nicht mehr trans-

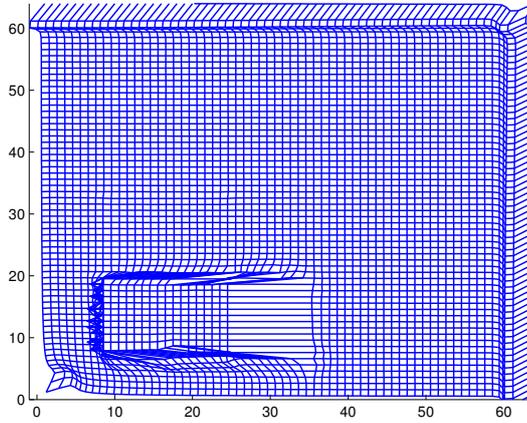


Abbildung 9: Deformationsfeld der Registrierung in Abbildung 8.

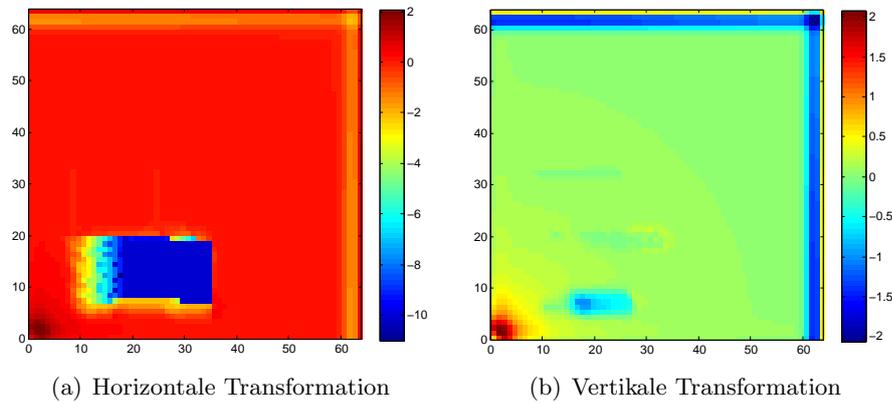


Abbildung 10: Ergebnisse der Registrierung für Template und Referenz aus Abbildung 8. Wir sehen das aufgeteilte Deformationsfeld als Intensitätsabbildung mit Regler λ .

formiert wird. Es scheint jedoch, dass in vertikaler Richtung mehr transformiert wird als zuvor. Der Intensitätsplot [13](#) bestätigt unseren Verdacht. Zwar sieht man hier, dass in horizontaler Richtung wie gewünscht transformiert wurde, in horizontaler Richtung ist die Transformation jedoch stärker als zuvor.

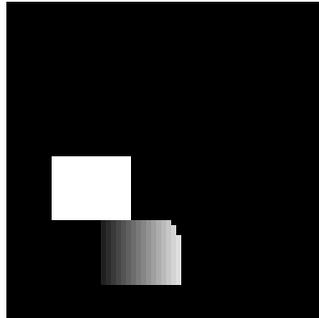


Abbildung 11: Das transformierte Templatebild der Registrierung mit $\lambda = 3$, $\beta = 10^{-1}$ und $\epsilon = 0.0001$. Template und Referenz sind in [Abbildung 8](#) zu sehen.

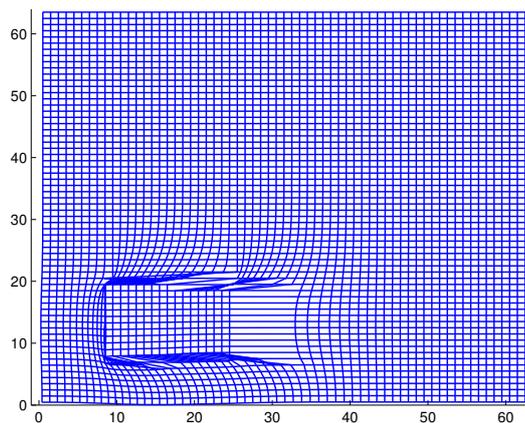


Abbildung 12: Deformationsfeld der Registrierung in [Abbildung 11](#).

6.4 Vergleich zu elastischem Regularisierer

Wir erkennen in der Abbildung wieso andere Regularisierer, zum Beispiel ein elastischer, für sliding motion ungeeignet ist. Es ist nicht möglich, das untere Rechteck als Ganzes zu verschieben und der Regularisierer versucht die beiden Rechtecke zu vermischen. Folglich ist das Ergebnis nicht zufriedenstellend.

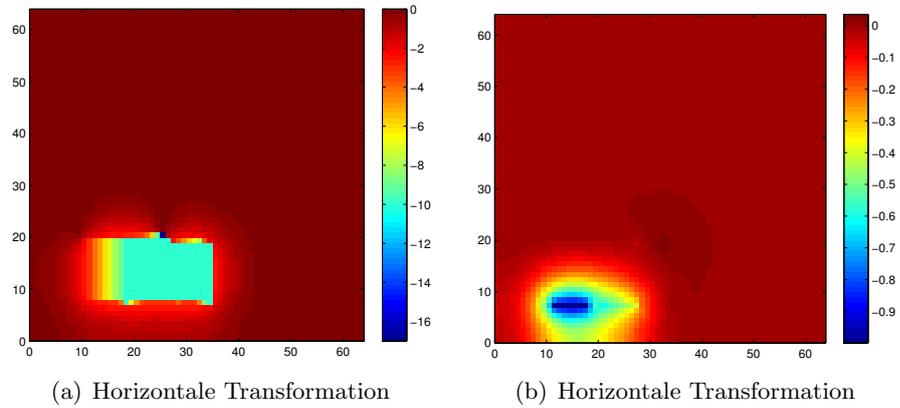


Abbildung 13: Ergebnisse der Registrierung für Template und Referenz aus Abbildung 8. Wir sehen das aufgeteilte Deformationsfeld als Intensitätsabbildung mit Regler λ .

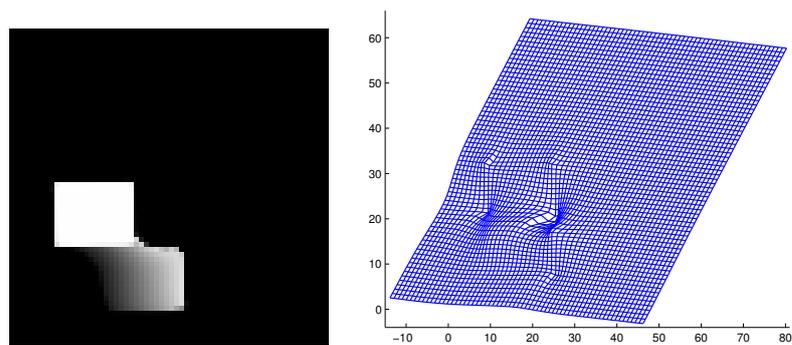


Abbildung 14: Transformiertes Templatebild und Gitter der Registrierung mit einem elastischen Regularisierer. Ergebnis des letzten von vier Leveln. Diese zwei Bilder wurden ausschließlich mit der FAIR-Toolbox erzeugt. Template und Referenz sind in Abbildung 8 zu finden.

6.5 Anwendung auf medizinische Bilder

Wir möchten im Folgenden eine Registrierung durchführen, die näher an der Realität liegt. Dafür betrachten wir Bilder einer menschlichen Lunge. Die Bilder stammen aus dem DIR-LAB Datensatz und stehen auf <http://dir-lab.com> frei zur Verfügung. Die Motivation der Autoren, die Bilder frei zur Verfügung zu stellen, ist dass jeder seine Implementierung an den Daten testen kann und die Ergebnisse den Autoren zukommen lässt, die anhand von Landmarken überprüfen, wie gut die Registrierung ist.

Wir verwenden die Methode, in der das c zum Einsatz kommt, da hierbei das Deformationsfeld stabiler zu sein scheint.

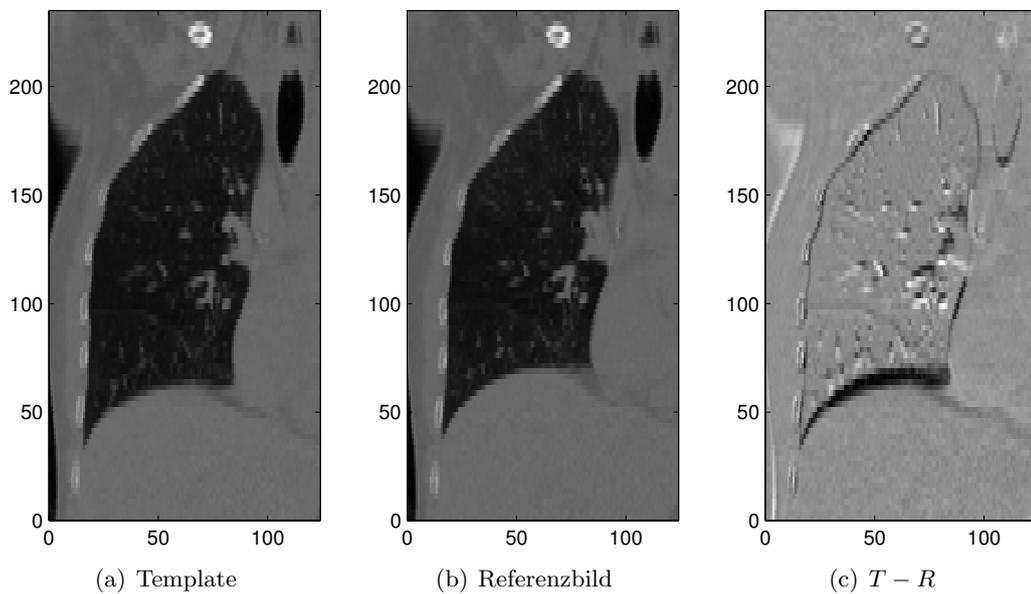


Abbildung 15: Ergebnis der Registrierung mit $\lambda = 10$, $\beta = 10^{-2}$, aber maximal 50 Iterationen, fünf Levels und mit 300 Schritten für Algorithmus 1. Die Bilder stammen aus dem DIR-LAB Datensatz 4DCT 1-10

Wir sehen in Abbildung 15 das Template- und Referenzbild und da der Unterschied optisch nicht erkennbar ist, sehen wir in der rechten Abbildung die Differenz von T und R . Darauf ist zu erkennen, dass der größte Unterschied am unteren Bereich der Lunge besteht. Das Gitter in Abbildung 16 indiziert, dass am unteren Bereich der Lunge transformiert wurde und Abbildung 17 bestätigt dies. In horizontaler Richtung wurde nur wenig transformiert, wohingegen in vertikaler Richtung eine größere Transformation stattgefunden hat, wie es bei einer Lunge beim Ein- und Ausatmen zu erwarten ist. Man erkennt außerdem, dass im Bereich der Rippen, wie gewünscht, nicht transformiert wurde. In Abbildung 16 sehen wir noch die Differenz zwischen $T(\varphi)$ und R und das zugehörige Deformationsfeld, bei dem ebenfalls im unteren Bereich der Lunge transformiert wurde.

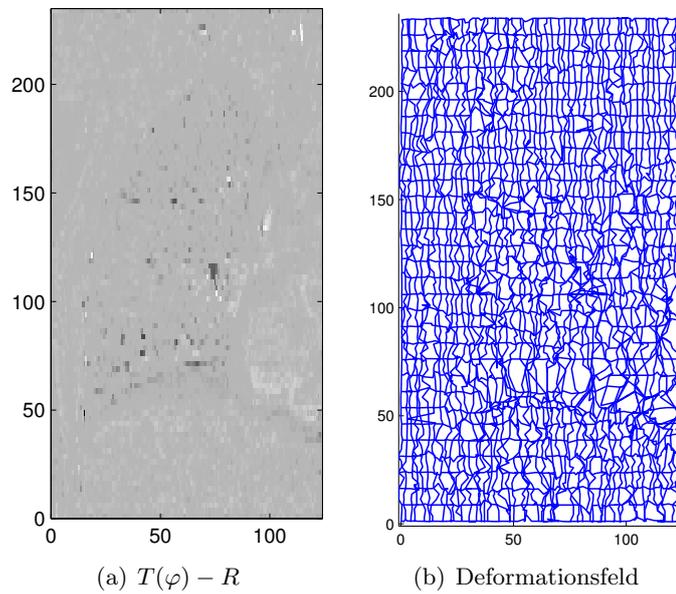


Abbildung 16: Distanz zwischen dem transformierten Templatebild $T(\varphi)$ und dem Referenzbild R aus Abbildung 15 und Deformationsfeld der Registrierung der Lunge.

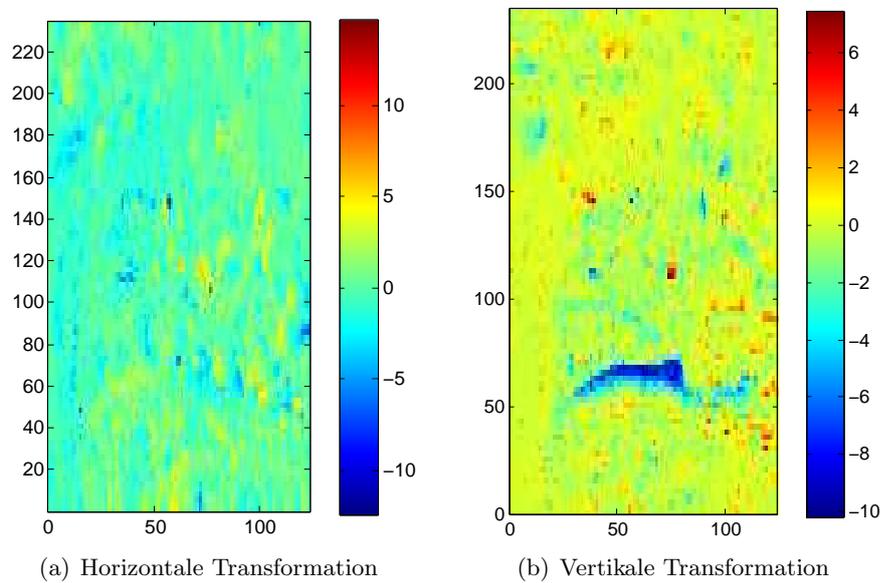


Abbildung 17: Aufgeteiltes Deformationsfeld der Lunge als Intensitätsabbildung mit Regler c .

6.6 Fazit

Wir haben in dieser Arbeit eine Methode zur Regularisierung von Deformationsfeldern implementiert. Dafür haben wir die nötigen Grundlagen vorgestellt und erklärt, wie wir das Problem numerisch umsetzen. Zusätzlich haben wir einen Regler c eingeführt, der vor dem Distanz- und Regularisierungsterm steht. Damit erscheint das Deformationsfeld insgesamt stabiler.

Es erwies sich als schwierig, die Parameter τ , σ und λ in Einklang miteinander zu bringen. Für τ und σ kann man sich zwar an den Vorschlägen von Chambolle und Pock [7] orientieren, doch es bleibt dann immer noch herauszufinden wie λ zu wählen ist und man muss gegebenenfalls noch ε und δ anpassen. Es bleibt somit noch herauszufinden wie diese Parameter am besten zu wählen sind. Regularisiert man zu stark, so kann T nicht ausreichend transformiert werden. Regularisiert man hingegen zu wenig, so ist das Deformationsfeld nicht mehr plausibel. Zudem ist noch zu klären, wieso bei den Rechtecken im transformierten Templatebild stets eine Ecke fehlt. Eine mögliche Begründung besteht darin, dass der Unterschied des Grauwertes zu den benachbarten Pixeln zu groß ist.

In Anlehnung an [12], die eine gewichtete Differenz aus iso- und anisotoper Totaler Variation vorschlagen, könnte man eine gewichtete Regularisierung mit unterschiedlichen Regularisierern implementieren. Wenn man sich die Bewegungen einer Lunge anschaut, so stellt man fest, dass diese nicht nur an den Rippen entlang gleitet, sondern sich auch elastisch verformt. Diese Art der Regularisierung ist für Objekte vorstellbar, denen man nicht eindeutig einen Regularisierer zuweisen kann. Alternativ könnte man das Huber-ROF-Modell statt das ROF-Modell verwenden, welches sowohl die Totale Variation als auch einen diffusiven Regularisierer verwendet, bei dem der comicartige Effekt nicht auftritt.

Insgesamt liefert die Implementierung gute Ergebnisse, denn sie macht das, was wir erwarten: Sie transformiert, bei einem entsprechenden Referenzbild, ein Gebiet ohne das angrenzende Gebiet zu verändern. Eine zusätzliche positive Eigenschaft ist dadurch gegeben, dass die Implementierung auch für medizinische Bilder funktioniert und nicht nur für künstlich erzeugte Beispiele.

Literaturverzeichnis

- [1] Aubert, G. und P. Kornprobst: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (Applied Mathematical Sciences)*. Springer-Verlag New York, 2006.
- [2] Boyd, S. und L. Vandenberghe: *Convex Optimization*. Cambridge University Press, New York, 2004.
- [3] Bredies, K. und D. A. Lorenz: *Mathematische Bildverarbeitung - Einführung in Grundlagen und moderne Theorie*. Vieweg+Teubner Verlag Wiesbaden, 2011.
- [4] Burger, M. und S. Osher: *A Guide to the TV Zoo*, S. 1–70. Springer International Publishing Cham, 2013.
- [5] Caselles, V., A. Chambolle, D. Cremers, M. Novaga und T. Pock: *An introduction to Total Variation for Image Analysis*. Theoretical Foundations and Numerical Methods for Sparse Recovery, De Gruyter, Radon Series Comp. Appl. Math., 9:263–340, 2010.
- [6] Chambolle, A.: *An Algorithm for Total Variation Minimization and Applications*. Journal of Mathematical Imaging and Vision, 20(1-2):89–97, 2004.
- [7] Chambolle, A. und T. Pock: *A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging*. Journal of Mathematical Imaging and Vision, 40(1):120–145, 2011.
- [8] Doicu, A., F. Schreier und M. Hess: *Iteratively regularized Gauss–Newton method for atmospheric remote sensing*. Computer Physics Communications, 148(2):214–226, 2002.
- [9] Forster, O.: *Analysis 1: Differential- und Integralrechnung einer Veränderlichen*. Vieweg+Teubner Verlag Wiesbaden, 2008.
- [10] Gritzmann, P.: *Grundlagen der Mathematischen Optimierung: Diskrete Strukturen, Komplexitätstheorie, Konvexitätstheorie, Lineare Optimierung, Simplex-Algorithmus, Dualität*. Springer Fachmedien Wiesbaden, 2013.
- [11] Haber, E., S. Heldmann und J. Modersitzki: *A Framework for Image-Based Constrained Registration with an Application to Local Rigidity*. Linear Algebra and its Applications, 431:459–470, 2009.
- [12] Lou, Y., T. Zeng, S. Osher und J. Xin: *A Weighted Difference of Anisotropic and Isotropic Total Variation Model for Image Processing*. SIAM J. Imaging Sci, 8(3):1798–1823, 2015.
- [13] Louis, A. K.: *Inverse und schlecht gestellte Probleme*. Teubner Studienbücher, Stuttgart, 1989.

-
- [14] Modersitzki, J.: *FAIR: Flexible Algorithms for Image Registration*. SIAM, Philadelphia, 2009.
- [15] Olesch, J.: *Bildregistrierung für die navigierte Chirurgie*. Aktuelle Forschung Medizintechnik. Springer Vieweg Verlag Wiesbaden, 2014.
- [16] Opfer, G.: *Numerische Mathematik für Anfänger*. Vieweg+Teubner Verlag Wiesbaden, 2008.
- [17] Press, W. H., B. P. Flannery, S. A. Teukolsky und W. T. Vetterling: *Numerical Recipes*. Cambridge University Press, 1989.
- [18] Reinhardt, R., A. Hoffmann und T. Gerlach: *Nichtlineare Optimierung: Theorie, Numerik und Experimente*. Springer Berlin Heidelberg, 2013.
- [19] Rockafellar, R. T.: *Convex analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970.
- [20] Rockafellar, R. T. und R. J. B. Wets: *Variational Analysis*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2009.
- [21] Rudin, L. I., S. Osher und E. Fatemi: *Nonlinear total variation based noise removal algorithms*. *Physica D*, 60:259–268, 1992.
- [22] Sauvigny, F.: *Analysis. Grundlagen, Differentiation, Integrationstheorie, Differentialgleichungen, Variationsmethoden*. Berlin: Springer Spektrum, 2014.
- [23] Schweizer, B.: *Partielle Differentialgleichungen: Eine anwendungsorientierte Einführung*. Springer-Verlag Berlin Heidelberg, 2013.
- [24] Zeidler, E., H. Schwarz, W. Hackbusch, B. Luderer *et al.*: *Springer-Handbuch der Mathematik III: Begründet von I.N. Bronstein und K.A. Semendjaew Weitergeführt von G. Grosche, V. Ziegler und D. Ziegler Herausgegeben von E. Zeidler*. Springer-Handbuch der Mathematik. Springer Fachmedien Wiesbaden, 2012.