UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS AND
IMAGE COMPUTING

Nichtparametrische Registrierung medizinischer Bilddaten mittels Schatten-q-Norm und nichtglatter Optimierung

Non-parametric registration of medical image data using Schatten-q-Norm and non-smooth optimization

Masterarbeit

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

vorgelegt von
Kai Brehmer

ausgegeben und betreut von
Prof. Dr. Jan Modersitzki
Institute of Mathematics and Image Computing

mit Unterstützung von
Prof. Dr. Jan Lellmann
Institute of Mathematics and Image Computing

Lübeck, den 02. Dezember 2016

Erklärung

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

Lübeck, 02. Dezember 2016

Kai Brehmer

Kurzfassung

Diese Arbeit stellt einen neuartigen, nichtparametrischen Bildregistrierungsansatz vor, welcher mittels nichtglatter Optimierungsmethoden minimiert werden soll. Hierzu werden sowohl eine Einführung in die Bildregistrierung als auch Grundlagen der mathematischen Optimierung geliefert. Die angestellten Untersuchungen prüfen das Modell auf Tauglichkeit als Registrierungsansatz im Rahmen des angepassten Optimierungsverfahrens. Ergänzend wird eine Einführung in die Segmentierung geliefert, welche Grundlage für die Entwicklung sogenannter Parameterkarten ist. Diese sollen in späteren Arbeiten mithilfe des vorgestellten Registrierungsmodells einen Mehrwert an Informationen für die medizinische Bildsegmentierung liefern. Hier werden erste Eindrücke und Ideen dazu gesammelt.

Abstract

This thesis proposes a novel, non-parametric image registration approach, which will be optimized using non-smooth optimization. To serve this purpose an introduction to image registration as well as basics of mathematical optimization are provided. Within the framework of non-smooth optimization the novel model is checked for usability as a registration approach. Furthermore an introduction to image segmentation is presented. This aims at laying the foundation for the developement of so-called parameter maps. Future works will present further investigation on parameter maps and an additional benefit for medical image segmentation by registrating these. This thesis gathers some initial ideas and impressions on this topic.

Inhaltsverzeichnis

1 Einleitung

Das Thema dieser Arbeit ist in erster Linie die Registrierung medizinischer Bilddaten. Die Bildregistrierung versucht zwei oder mehrere Bilder der gleichen Szene durch Deformation einander ähnlich zu machen. Dieser Prozess kann auf unterschiedliche Weisen erfolgen. Die verschiedenen Arten lassen sich grob in intensitätsbasierte und merkmalsbasierte Verfahren kategorisieren. Merkmalsbasierte Verfahren verwenden prominente Punkte, sogenannte Landmarken, die übereinander gelegt werden sollen, wie z.B. in [Pol12] betrachtet. Intensitätsbasierte Verfahren hingegen versuchen, mithilfe der gegebenen Intensitätswerte ein Distanzmaß zu minimieren und die Daten so einander anzugleichen. Da dieses Problem schlechtgestellt ist, benötigt man einen Regularisierer, damit plausible Deformationen entstehen können.
Diese Arbeit beschreibt einen neuartigen Registrierungsansatz, der es ermöglichen soll, verschiedene, aus den Intensitätsdaten gewonnene Parameterkarten gegeneinander zu registrieren. Diese Parameterkarten stellen verschiedene Repräsentationen der Daten dar und deren Registrierung kann eine Verbesserung der nutzbaren Informationen herbeiführen. Es geht dabei nicht nur um die Registrierung an sich, sondern auch um einen passenden Optimierungsansatz, welcher die Besonderheiten des Registrierungsansatzes händeln können muss. Dabei steht dieser Ansatz direkt an der Schnittstelle zur Bildsegmentierung, da eine Erhöhung der nutzbaren Informationen eine Erleichterung der Segmentierung bedeuten kann.

1.1 Motivation

Die Bildregistrierung ist ein anspruchsvolles Problem der digitalen Bildverarbeitung [FM03]. Viele verschiedene Bereiche, wie z.B. Kunst, Astronomie, Astrophysik, Biologie oder Chemie [FM08], finden darin ein Handwerkszeug. Vor allem aber auch die Medizin profitiert von der Bildregistrierung. So werden neben computergestützten Verfahren zur Diagnose auch Operationsplanung, Krankheitsverlaufskontrolle, Bewegungskorrekturen und verschiedene weitere Planungen in der Radiologie und zahlreichen weiteren Bereichen

möglich [FM08]. Aber nicht nur Planungen, sondern auch Fusionen der Daten verschiedener Modalitäten sind möglich. Durch die stetige Weiterentwicklung der bildgebenden Verfahren, wie Computertomographie (CT) oder Magnetresonanztomographie (MRT), ist auch eine stetige Verbesserung der Qualität der Registrierungsverfahren möglich. Dabei ist der Begriff „Verbesserung“ von der Anwendung abhängig. Verschiedene Anwendungen erfordern verschiedene Aspekte, die mehr oder weniger gut beurteilt werden können sollten. So ist z.B. für eine Operationsplanung die Genauigkeit der Überlagerung der Bilder von besonders hoher Priorität. Zum einen sind durch die verschiedenen Anwendungsfelder viele verschiedene Verfahren entstanden, zum anderen händelt jeder Ansatz die verschiedenen Störungen der Daten anders. Zu den Störungen zählen sowohl vermeidbare als auch unvermeidbare Artefakte, wie Bewegungen, die durch den Patienten verursacht werden, bzw. Organbewegungen, wie die des Herzens. Außerdem spielt das Rauschen eine wichtige Rolle. Die verschiedenen bekannten Modalitäten erstellen Daten unterschiedlicher Qualität und weisen aufgrund verschiedener Faktoren, wie z.B. der Bauart, verschiedene Rauschmuster auf. Vermeidbare Artefakte, wie solche, die durch die Atmungsbewegung der Lunge entstehen, werden z.B. durch Atemkommandos bzw. -triggerung versucht zu minimieren. Dafür erhält der Patient Anweisungen von dem/der Medizinisch-Technischen Radiologieassistenten/in, wann die Luft angehalten werden muss bzw. die Atmung wird automatisch gemessen. Eine optimale Reduktion von Unschärfen durch Bewegung kann dadurch allerdings nicht gewährleistet werden. Unter anderem setzt die Bildregistrierung hier an und versucht z.B. in zeitabhängigen Daten die Organbewegung zu erfassen und einzufrieren. Ein solcher Prozess kann einerseits der Verbesserung der Qualität der Daten und andererseits der Vorbereitung einer Segmentierung dienen. Je klarer verschiedene Regionen voneinander unterschieden werden können, desto einfacher ist eine Zerlegung der Daten in diese Regionen. In der Strahlentherapie z.B. wird zur Bestrahlungsplanung die Aufgabe der Segmentierung teilweise noch von Hand übernommen [Sau10]. Dort liefern Algorithmen meist nur einen Vorschlag einer möglichen Segmentierung, die der Anwender bestätigen oder verbessern kann. Mithilfe der Registrierung und dem Einfluss auf die Segmentierung gilt es, die bekannten Verfahren weiter zu verbessern und neue Verfahren zu entwickeln, um die Qualität für die Anwender und diejenigen, die daraus einen Nutzen ziehen, stetig auf einem hohen Niveau zu halten.

1.2 Problemstellung und Lösungsansatz

Nicht die Registrierung selbst ist die eigentliche Problemstellung, sondern der daraus resultierende Nutzen, die Daten verbessern und vorbereiten zu können, um weitere Verarbeitungsschritte zu vereinfachen. Dabei stellt sich die Frage, wie die Daten registriert werden sollen und wie daraus eine Verbesserung hervorgeht. In dieser Arbeit wird ein Registrierungsansatz behandelt, der Kanten in den Daten aneinander ausrichtet. Diese Idee ist sehr verwandt zum *Normalized Gradient Field* Ansatz von Haber und Modersitzki [HM05, HM06]. Zur Auswertung wird die Schatten-q-Norm herangezogen, um eine Nichtlinearität sowie Nichtglattheit zu erhalten. Ein solcher nichtglatter Ansatz hat sich unter anderem bei der Entrauschung von Farbbildern bewährt [MSMC15a], da die Optimierung ableitungsfrei abläuft und dadurch keine Glättungen vornimmt. Der in dieser Arbeit vorgestellte Ansatz soll dazu dienen, mehrere Parameterkanäle miteinander registrieren zu können und daraus eine Verbesserung der Daten zu erhalten. Damit ist es denkbar, eine Segmentierung vorzubereiten oder die Ergebnisse bekannter Segmentierungsalgorithmen verbessern zu können. Der Registrierungs- sowie der Optimierungsansatz wurde in MATLAB implementiert und mithilfe von MRT-Perfusionsdaten getestet. Die in dieser Arbeit verwendeten Daten wurden freundlicherweise von Jarle Rørvik und Kollegen vom Haukeland University Hospital in Bergen, Norwegen und Rashindra Manniesing, DIAG-Gruppe des Radboud University Medical Centers, Nijmegen, Niederlande zur Verfügung gestellt.

1.3 Gliederung der Arbeit

Die Arbeit ist in drei Abschnitte eingeteilt. Begonnen wird in Abschnitt I mit den Grundlagen zur Bildregistrierung und -segmentierung. Dazu zählen unter anderem Deformationsansätze, Distanzmaße, Regularisierer, Gitter und Interpolation für die Bildregistrierung in Kapitel 2 als auch verschiedene Ansätze für Grauwert- und Farbbildzerlegung für die Segmentierung in Kapitel 3. Dazu wird auch eine kurze Einführung in die Farbraumtheorie gegeben.
Weiter geht es in Abschnitt II mit der Aufbereitung des Zusammenhangs von Parameterkarten und (Farb-)Kanälen in Kapitel 4. In diesem Kapitel werden die grundlegenden Werkzeuge der Singulärwertzerlegung und der Schatten-q-Norm vorgestellt. Anschließend erhält der Leser in Kapitel 5 eine Einführung in die mathematische Optimierung. Die kurze Einführung geht rasch in das umfangreiche Gebiet der konvexen Optimierung über

und soll einige wichtige Konzepte näher bringen. Dort geht es sowohl um die Konvexität als auch um primal-duale Probleme und deren Lösungsansätze. Dabei werden erste Lösungsalgorithmen vorgestellt.
Kapitel 6 bildet den Hauptteil dieser Arbeit. Dort wird der neuartige Ansatz vorgestellt und erläutert.
In Abschnitt III sollen Ergebnisse verschiedener Experimente präsentiert und besprochen werden. Dieser Abschnitt beginnt mit Kapitel 7, in welchem ein eindimensionales Problem, mit welchem das Prinzip der Optimierung des Algorithmus' und dessen Konvergenzverhalten illustriert werden. Anschließend wird eine zweidimensionale Registrierung grauwertbasierter Daten durchgeführt. Dort finden sich auch kurze Erläuterungen des MATLAB-Quellcodes.
Abschließend folgt in Kapitel 8 eine Diskussion über die Ergebnisse mit Vor- und Nachteilen gegenüber anderer Verfahren. Dabei soll ein Ausblick auf weitere Experimente und Untersuchungen dieses Ansatzes gegeben werden.

Teil I

Grundlagen

2 Einführung in die Bildregistrierung

In diesem Kapitel soll eine kurze Einführung in das Thema der medizinischen Bildregistrierung erfolgen. Dabei werden grundlegende Prinzipien und Ansätze, auf denen die Registrierung aufbaut, erläutert. Mithilfe der Ansätze werden einige zugehörige Verfahren präsentiert und es wird beschrieben, wie diese mit anderen Bereichen der Bildverarbeitung korrespondieren. Basierend darauf soll der in dieser Arbeit neu eingeführte Ansatz entwickelt und implementiert werden. Die folgenden Ideen und Konzepte stammen, falls nicht anders erwähnt, aus den Standardwerken [MV98, ZF03, Mod04, Mod09] sowie einigen weiteren im Literaturverzeichnis aufgeführten Quellen.

2.1 Grundlagen der Registrierung

Die Bildregistrierung ist ohne Frage ein wichtiger Prozess in der modernen Bildverarbeitung. Zum Verständnis der Idee der Bildregistrierung gilt es einige mathematische Grundlagen zu betrachten. Dazu bedarf es nicht nur Methoden zur Bestimmung der Ähnlichkeit von Bilddaten, sondern auch Kenntnis über verschiedene Deformationsmöglichkeiten und deren mathematische Formulierung.
Gegeben sind für ein klassisches Registrierungsproblem zwei d-dimensionale Bilder $\mathcal{R}, \mathcal{T} : \mathbb{R}^d \to \mathbb{R}$. $\mathcal{T}$ wird Templatebild und $\mathcal{R}$ Referenzbild genannt. Für diese gilt es eine „sinnvolle“ Deformation $y : \mathbb{R}^d \to \mathbb{R}^d$ zu finden, so dass $\mathcal{R}$ und $\mathcal{T} \circ y$ ähnlich sind [Mod09]. Mathematisch formuliert sich das Registrierungsproblem als Minimierung [Mod04]:

$$\mathcal{J}[\mathcal{R}, \mathcal{T}; y] \xrightarrow{y} \min .$$

Wie sich eine „sinnvolle“ Deformation gestaltet, wann $\mathcal{R}$ und $\mathcal{T}$ ähnlich sind und wie genau das zu minimierende Zielfunktional $\mathcal{J}$ aussieht, wird im Folgenden geklärt.
Normalerweise repräsentieren sich die d-dimensionalen Bilder als kontinuierliche Grauwertbilder und sind folgendermaßen definiert [Mod04]:

Definition 2.1 (d-dimensionales Bild) *Die Abbildung* $I : \Omega \to \mathbb{R}, \quad \Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$ *mit*

1. *I hat einen kompakten Träger,*
2. $0 \leq I(x) \leq +\infty \quad \forall x \in \mathbb{R}^d$,
3. $\int_{\mathbb{R}^d} I(x)\,\mathrm{d}x < +\infty$.

heißt d-dimensionales Bild.

Das Referenzbild $\mathcal{R}$ und das Templatebild $\mathcal{T}$ sind entsprechend definiert.
Im Allgemeinen lautet die mathematische Formulierung eines Registrierungsproblems [Mod09]:

$$\min_y \mathcal{D}(\mathcal{R}, \mathcal{T} \circ y), \quad \mathcal{D} \text{ Distanzmaß.} \tag{2.1}$$

Mit dieser Formulierung allein ist die Existenz einer Lösung keineswegs gesichert. Hierzu werden in diesem Kapitel sogenannte regularisierende Funktionale vorgestellt, mit welchen eine Existenz gewährleistet werden kann [FM08]. Die regularisierenden Funktionale ändern allerdings nichts an der, in den meisten Fällen, nicht gegebenen Eindeutigkeit der Registrierungsprobleme.

2.2 Deformationen

Grundsätzlich gibt es zwei verschiedene Transformationsansätze, um $\mathcal{T} \circ y$ zu erhalten. Dabei unterscheidet man in welche „Richtung“ transformiert wird [Mod04]:

1. Lagrange-Ansatz (Vorwärtstransformation)
   Beim Lagrange-Ansatz wird jedem Bildpunkt x im zu transformierenden Bild eine neue Koordinate $y(x)$ zugeordnet.

2. Euler-Ansatz (Rückwärtstransformation)
   Beim Euler-Ansatz wird jedem Bildpunkt x im Bildgebiet Ω der Wert von $\mathcal{T}$ an der Stelle $y(x)$ zugeordnet.

Meist wird die Rückwärtstransformation verwendet, da ein festes Gitter vorliegt, auf dem interpoliert wird. Konkret werden die entsprechenden Gitterkoordinaten mithilfe der inversen Transformationsfunktion berechnet. Dadurch ist der Vergleich mit dem Referenzbild $\mathcal{R}$ später einfacher. Die Vorwärtstransformation ist zwar intuitiver, hat allerdings den Nachteil, dass auch das Gitter transformiert, also für jeden Bildpunkt ein

neues Koordinatenpaar berechnet wird. Dadurch ist ein Vergleich mit $\mathcal{R}$ unter Umständen nur schwer oder gar nicht möglich und es können mehrere Koordinaten zusammenfallen. Numerische Werkzeuge zur Umsetzung, wie Gitter und Interpolationsmethoden, werden am Ende des Kapitels näher beschrieben.

2.2.1 Parametrische Registrierung

Die Transformationen, die durch $y : \mathbb{R}^d \to \mathbb{R}^n$ dargestellt werden können, sind lineare, globale Deformationen. Die einfachsten elementaren Transformationen sind affinlineare Transformationen. Man kann sie mithilfe simpler Matrix-Vektorgleichungen der Form $y = Ax + b$ darstellen. Sei im Folgenden der zweidimensionale Fall betrachtet, wobei $A \in \mathbb{R}^{2\times 2}$ die Transformationsmatrix darstelle und $x = (x^1, x^2)^\mathsf{T} \in \mathbb{R}^2$ den Vektor der Koordinaten der Bildpunkte. $b \in \mathbb{R}^2$ sei der Translationsvektor. Man kann dieses Modell parallel auf alle Bildpunkte anwenden, indem man in $x \in \mathbb{R}^{2\times n}$ alle $n \in \mathbb{N}$ Koordinaten speichert. Analog muss b erweitert werden. Dies ist auch für höhere Dimensionen möglich.

- Translation:
  Die sogenannte Translation ist eine Parallelverschiebung. Diese erhält man durch Addition eines Translationsvektors. Die Daten werden global verschoben. Man setzt

  $$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

  So lässt sich die Translation aus $y = Ax + b$ ermitteln. In b sind die Informationen über die Verschiebung gespeichert.

- Skalierung:
  Mithilfe der Skalierung können die verschiedenen Raumrichtungen gestreckt oder gestaucht werden. Im zweidimensionalen Fall erreicht man dies durch Setzen von $y(x) = Ax + b$ mit

  $$A = \begin{pmatrix} \sigma_{x^1} & 0 \\ 0 & \sigma_{x^2} \end{pmatrix}.$$

  Dabei sind $\sigma_{x^1}, \sigma_{x^2} \neq 0$. In diesem Fall wird $b = (0, 0)^\mathsf{T}$ gesetzt.

Es existieren noch einige weitere Transformationen wie z.B. Rotationen oder Scherungen, welche auch miteinander kombiniert werden können, auf die in dieser Arbeit nicht weiter eingegangen wird. Stellt man an die Transformationsmatrix die Bedingung $A^\mathsf{T}A = AA^\mathsf{T} = 1$, nennt man die Transformation *rigide* bzw. starr. Diese Eigenschaft bedeutet, dass

die Deformation der Daten längen- sowie winkelerhaltend ist und damit letztlich noch Translationen, Rotationen und Skalierungen möglich sind. Zur Rekonstruktion von $\mathcal{T}$ aus $\mathcal{R}$ – um später $\mathcal{T}$ und $\mathcal{R}$ vergleichen zu können – wählt man einen dieser Ansätze zur Transformation von $\mathcal{R}$. Da alle Bildpunkte von einer gleichartigen Veränderung betroffen sind, nennt man diese Art der Transformation global. Tatsächlich kann man diese Ansätze als *parametrisiert* zusammenfassen, da die Deformation y als parametrisiertes Funktional betrachtet wird. Das Problem der parametrischen Registrierung ist die meist unzureichende Annäherung tatsächlicher Deformationen realer Probleme. Man kann die parametrische Registrierung dennoch zur Vorregistrierung verwenden [Mod04]. Das bedeutet, man bringt mit diesen vereinfachten Ansätzen die Daten einander näher, um die komplexere nichtparametrische Registrierung effizienter durchführen zu können.

2.2.2 Nichtparametrische Registrierung

Praktisch relevanter sind die *nichtparametrischen* Ansätze der Registrierung. Die Deformation y kann durch ein beliebiges Funktional dargestellt werden, ist also nicht mehr vorgegeben. Das Registrierungsproblem [MV98]

$$\min_y \mathcal{D}(\mathcal{R}, \mathcal{T} \circ y), \quad \mathcal{D} \text{ Distanzmaß} \tag{2.1}$$

hat dadurch zwei wesentliche Probleme:

1. y ist in der Regel nicht eindeutig,

2. y kann unerwünschte Eigenschaften haben.

Unerwünschte Eigenschaften sind in diesem Kontext unter anderem Unstetigkeiten oder Singularitäten im Sinne der Invertierbarkeit. Nach der Definition 2.2 über die *Korrektgestelltheit* eines Problems von Jacques Hadamard [Had02], ist die Registrierung ein schlechtgestelltes Problem, da die Lösung nicht eindeutig ist und damit mindestens die zweite Bedingung verletzt ist.

Definition 2.2 (Korrektgestelltheit) *Ein mathematisches Problem heißt* korrektgestellt *(vgl. [Had02]), wenn*

1. *eine Lösung existiert,*
2. *die Lösung eindeutig ist,*
3. *die Lösung stetig von der Eingabe abhängt.*

Nicht korrektgestellte Probleme heißen schlechtgestellt.

Das Zielfunktional $\mathcal{J}(y)$ setzt sich im nichtparametrischen Ansatz aus zwei Teilen zusammen. Einerseits wird weiterhin das Distanzmaß $\mathcal{D}(y)$ benötigt, um die Ähnlichkeit von $\mathcal{T}$ und $\mathcal{R}$ messen zu können. Andererseits muss sichergestellt werden, dass die unbekannte Deformation y bestimmte Glattheitseigenschaften erfüllt [Mod04]. Die Glattheitseigenschaften werden durch ein Funktional, einen sogenannten Regularisierer $\mathcal{S}$, an die Deformation y gestellt, welcher gewichtet durch einen Faktor $\alpha \in \mathbb{R}_+$ in das Zielfunktional $\mathcal{J}(y)$ eingeht. Damit ist das Zielfunktional eine Linearkombination aus Distanzmaß $\mathcal{D}(y)$ und gewichtetem Regularisierer $\alpha\mathcal{S}(y)$ [ZF03]

$$\mathcal{J}(y) = \mathcal{D}(y) + \alpha\mathcal{S}(y). \tag{2.2}$$

Die Idee der Optimierung solcher und anderer Funktionale wird in Kapitel 5 erläutert.

Zu klären bleibt noch, wie sich Ähnlichkeit bestimmen lässt und wie die angesprochenen Regularisierer aussehen. Dies wird im Folgenden geklärt.

2.3 Distanzmaße

Üblicherweise versucht man mithilfe der Distanzmaße eine möglichst große Ähnlichkeit der Bilder zu erreichen. Das bedeutet im Umkehrschluss, das Distanzmaß soll möglichst klein werden. Der Begriff der *Ähnlichkeit* wird durch die verschiedenen Distanzmaße in unterschiedlicher Art und Weise interpretiert. Anforderungen an ein Distanzmaß kann man in folgender Weise beschreiben [MV98]:

1. klein, wenn $\mathcal{R}$ und $\mathcal{T} \circ y$ ähnlich,
2. idealerweise ist $\mathcal{D}(y)$ differenzierbar,
3. $\mathcal{D}(\mathcal{R}, \mathcal{T} \circ y) = \mathcal{D}(\mathcal{T} \circ y, \mathcal{R})$,

4. $\mathcal{D}(A, A) = \min_B \mathcal{D}(B, A)$.

Im Allgemeinen wird zwischen mono- oder multimodalen Bilddaten unterschieden. Der Begriff Modalität bezieht sich auf die Entstehung der Daten. Es wird beschrieben, ob die Daten durch ein oder mehrere Bildgebungsverfahren entstanden sind. Folgend wird von mono- oder multimodalen Maßen die Rede sein. Im Zusammenhang mit den Distanzmaßen soll der Begriff daran erinnern, für welche Art von Daten die Maße geeignet sind und welche Invarianzen sie vereinen. Monomodale Maße weisen dabei keinerlei Invarianzen gegenüber bestimmter Änderungen wie Intensitäts- oder Kontraständerungen auf. Multimodale Maße hingegen haben unter bestimmten Voraussetzungen Invarianzen, so dass sie auch für den Vergleich von Bildern verschiedener Modalitäten verwendet werden können.
Es folgen Beispiele bekannter Distanzmaße.

2.3.1 Summe der quadrierten Differenzen

Die Summe der quadrierten Differenzen – auch *sum of squared distances* (SSD) – basiert auf einem punktweisen Vergleich der Intensitätswerte der verglichenen Bilder. Es ist ein monomodales Distanzmaß. Definiert ist das Maß durch [MV98]

$$\mathrm{SSD}(\mathcal{R}, \mathcal{T} \circ y) = \mathcal{D}^{\mathrm{ssd}}(\mathcal{R}, \mathcal{T} \circ y) = \frac{1}{2} \int_\Omega (\mathcal{T}(y(x)) - \mathcal{R}(x))^2 \, \mathrm{d}x. \tag{2.3}$$

Die Ähnlichkeit basiert in diesem Fall darauf, dass die Differenz der verglichenen Bilder im Optimum verschwindet. Ein Problem dieses Distanzmaßes ist die Rauschanfälligkeit, insbesondere bei Ausreißern. Allerdings kann die Einfachheit des Maßes z.B. bei der Berechnungsgeschwindigkeit von Vorteil sein.

2.3.2 Normalisierte Kreuzkorrelation

Das Problem der SSD ist der direkte Vergleich der Grauwerte. Die Kreuzkorrelation schwächt diesen Zusammenhang so ab, dass die Grauwerte nicht direkt miteinander, sondern unter Einfluss bestimmter Skalierungsfaktoren verglichen werden. Dieser Unterschied macht dieses Maß gegenüber der SSD für multimodale Anwendungen robuster [Mod09]. Die normalisierte Kreuzkorrelation (*normalized cross-correlation*, kurz NCC) ist definiert als [Mod09]

$$\mathrm{NCC}(\mathcal{R}, \mathcal{T} \circ y) = \mathcal{D}^{\mathrm{ncc}}(\mathcal{R}, \mathcal{T} \circ y) = \frac{\langle \mathcal{T}, \mathcal{R} \rangle}{\|\mathcal{T}\| \|\mathcal{R}\|}. \tag{2.4}$$

Multipliziert man den Integranden $(\mathcal{T} - \mathcal{R})^2$ der SSD aus und betrachtet nur den entstehenden Mischterm der zweiten binomischen Formel zur Integration, so erhält man die Kreuzkorrelation.

2.3.3 Normalisiertes Gradienten-Feld

Das normalisierte Gradienten-Feld – auch *normalized gradient field* (NGF) – wurde erstmals von Haber und Modersitzki [HM05, HM06] vorgestellt und ist ein multimodales Maß. Es zielt darauf ab, nicht die Intensitäten der Bilder, sondern deren Ortsableitungen zu vergleichen. Mithilfe des NFG-Maßes versucht man zu untersuchen, welche Tupel von Bildpunkten linear abhängig sind. Das bedeutet, man versucht Kanten gleicher Richtung zu finden. Eine Normierung der Kanten wird vorgenommen, damit nur die Ausrichtung und nicht die Höhe der Kanten einfließt. Die geforderte Abhängigkeit kann mithilfe eines Kreuz- oder Skalarprodukts untersucht werden. Bekanntermaßen ist das Skalarprodukt minimal für orthogonal stehende Kanten und maximal für linear abhängige Kanten. Die Formulierung mithilfe des Skalarprodukts lautet [HM05]:

$$\mathrm{NGF}(\mathcal{R}, \mathcal{T} \circ y) = \mathcal{D}^{\mathrm{ngf}}(\mathcal{R}, \mathcal{T} \circ y) = \int_\Omega 1 - \left(\frac{\nabla \mathcal{T}(y(x))^\mathsf{T} \nabla \mathcal{R}(x)}{\|\nabla \mathcal{T}(y(x))\| \|\nabla \mathcal{R}(x)\|} \right)^2 \mathrm{d}x. \tag{2.5}$$

Es gibt auch eine Formulierung mittels Kreuzprodukt. Damit erhält man allerdings zweierlei Probleme. Einerseits kann es passieren, dass aufgrund der Differenzen, die gebildet werden, Auslöschungen auftreten. Andererseits werden zusätzliche, unerwünschte lokale Minima der Zielfunktion erzeugt, da das Maß minimal für einen Gradienten von Null würde. Es könnte während der Optimierung passieren, dass das Bild aufgrund dessen konstant würde, da damit alle Gradienten Null entsprächen. Diese Probleme können mithilfe des Skalarprodukts nicht auftreten, da keine Differenzen gebildet werden und das Maß für konstante Stellen maximal wird. Der Vorteil dieses Maßes ist die Invarianz gegenüber differenzierbaren Kontraständerungen, welche monomodale Maße nicht aufweisen. Angenommen es seien zwei Bilder A, B gegeben und eine beliebige, stetig differenzierbare Funktion $g \in C^1(\mathbb{R}, \mathbb{R})$. Für diesen Fall erhält man $B(x) = g(A(x))$. Setzt man dies in das Maß ein, sieht man schnell, dass mithilfe der Kettenregel daraus folgt $\mathrm{NGF}(A, B) = \mathrm{NGF}(A, A)$.

2.4 Regularisierung

Regularisierer repräsentieren die Plausibilität der Transformation y [Mod09]. Durch die Addition des regularisierenden Funktionals $\mathcal{S}$ geht zusätzliche Information in das Problem ein, welche mithilfe des bereits erwähnten Faktors $\alpha \in \mathbb{R}_+$ gesteuert werden kann. Man versucht mit $\alpha\mathcal{S}$ die Korrektgestelltheit des Problems [Had02], wie sie in Definition 2.2 zu finden ist, zu beeinflussen. Das heißt, die Plausibilität einer Transformation spiegelt sich darin wieder, wie schlechtgestellt das Problem mit eben dieser Transformation ist. Üblicherweise soll die Regularisierung dazu führen, dass die Lösung nicht nur eindeutig wird, sondern das Zielfunktional im besten Fall auch konvex [Mod09]. Mithilfe bestimmter Optimierungsverfahren kann man die Konvexität zumindest teilweise vernachlässigen, dazu mehr in Kapitel 5.

2.4.1 L_2-Regularisierer

Die meisten Regularisierer, die in der Bildregistrierung verwendet werden, sind üblicherweise Varianten von Ableitungen in L_2-Normen [Mod09]. Diese Ableitungen beziehen sich meist auf die Verrückung

$$u = y - \mathrm{Id},$$

das bedeutet zumeist $y(x) = x + u(x)$. Durch diese Verrückung ist es möglich, Anteile der Deformation nicht zu regularisieren. Die hier gewählte Identität Id bewirkt, dass nur lokale Änderungen der Deformation regularisiert werden. Die allgemeine Darstellung der Verrückung ist $u = y - y^{\mathrm{ref}}$ [Mod09]. Damit gilt hier $y^{\mathrm{ref}} = \mathrm{Id}$.
Gewöhnliche L_2-Regularisierer stellen sich demnach dar als [Mod09]

$$\mathcal{S}(u) = \frac{\alpha}{2} \int_\Omega \|\nabla u\|_2^2 \,\mathrm{d}x, \quad \alpha > 0. \tag{2.6}$$

2.4.2 Elastische Regularisierung

Die elastische Regularisierung ist physikalisch motiviert und der Elastizitätstheorie zuzuordnen. Erstmals vorgestellt von Broit [Bro81], bildet sie eine der Grundlagen der klassischen Materialtheorie. Die Idee dieser Art der Regularisierung ist die Modellierung eines elastischen Materials, durch das die Bilddaten deformiert werden. Mithilfe der sogenannten *Lamé-Konstanten* $\lambda, \mu \geq 0$ ist dieser Regularisierer durch ein Potenzial

gegeben als [Bro81]

$$\mathcal{S}^{\text{elas}}(y) = \frac{1}{2}\int_\Omega \sum_{j=1}^{d} \mu\|\nabla u_j\|^2 + (\lambda+\mu)(\nabla\cdot u)^2\,\mathrm{d}x \tag{2.7}$$

Dabei ist $d \in \{1,\dots,4\}$ die räumliche Dimension bzw. Anzahl der Raumrichtungen der Daten, wie in Definition 2.1 festgelegt. Mit der Divergenz div und der Frobenius-Norm $\|\cdot\|_{\mathrm{F}}$ kann man eine andere Schreibweise mit

$$\mathcal{S}^{\text{elas}} = \frac{1}{2}\int_\Omega \sum_{j=1}^{d} \frac{\mu}{2}\|\nabla u_j + (\nabla u_j)^{\mathsf{T}}\|_{\mathrm{F}}^2 + \lambda(\operatorname{div} u)^2\,\mathrm{d}x$$

finden. Diese Schreibweise teilt die Lamé-Koeffizienten den Summanden klar zu.

2.4.3 Diffusive Regularisierung

Der diffusive Regularisierer – erstmals erwähnt im Zusammenhang mit optischem Fluss [HS81] – erhält seinen Namen von der diskretisierten Euler-Lagrange-Gleichung [Wei98] und ist eine generalisierte Diffusionsgleichung, vergleichbar mit der Wärmeleitungsgleichung. Für die Bildregistrierung beschrieben und angewandt in [FM01, FM02b] ist der Regularisierer gegeben als

$$\mathcal{S}^{\text{diff}} = \frac{1}{2}\sum_{j=1}^{d}\int_\Omega \|\nabla u_j\|^2\,\mathrm{d}x. \tag{2.8}$$

2.4.4 Krümmungsbasierte Regularisierung

Dieser Regularisierer – auch *curvature* Regularisierer genannt – versucht, mittels Laplace-Operator Δ die Krümmung der Komponenten zu beeinflussen. Üblicherweise kann man die zweiten Ableitungen einer Funktion als Krümmung an der abgeleiteten Stelle interpretieren. Mit

$$\mathcal{S}^{\text{curv}}(y) = \frac{1}{2}\int_\Omega \sum_{j=1}^{d} \|\Delta u_j\|^2\,\mathrm{d}x \tag{2.9}$$

erhält man einen Regularisierer, der die zweiten Ableitungen behandelt. Vorgestellt wurde dieser Ansatz erstmals von Fischer und Modersitzki in [FM02a, FM03] . Aufgrund der Regularisierung der Krümmung sind die Transformationen, die man durch die Optimierung mit diesem Ansatz erhält, glatter als in den anderen vorgestellten Ansätzen.

2.5 Numerische Methoden

2.5.1 Gitter

In der Theorie kann man die Daten – wie bisher in dieser Arbeit – kontinuierlich betrachten. In der Praxis jedoch werden die Daten durch Abtastung diskretisiert und man kann im weiteren Verlauf der Verarbeitung mittels Interpolation auch an nicht gegebenen Stellen auswerten. So wird das kontinuierliche Modell approximiert. Diese Approximation ist vor allem für die spätere Implementierung in dieser Arbeit wichtig.
Die gegebene Auswertung liegt auf einem bestimmten Raster von Punkten vor, welches üblicherweise als Gitter bezeichnet wird und je nach Definition unterschiedlicher Art sein kann. Das betrachtete Gebiet Ω gestaltet sich in diesem Fall als $\Omega = (0, \omega_1) \times \cdots \times (0, \omega_d)$ mit $\omega_j \in \mathbb{R}_+$. Das Standardgitter, mit $\bar{m} = \Pi_{i=1}^{d} m_i$ und $m = (m_1, \ldots, m_d)$ Punkten, ist das sogenannte *zellzentrierte Gitter*, welches wie folgt aufgebaut ist [Mod09]:

Definition 2.3 (Zellzentriertes Gitter) *Seien die Zellbreiten der Raumrichtungen k als*

$$h_k = \frac{\omega_{2k} - \omega_{2k-1}}{m_k}, \quad k \in \{1, \ldots, d\}$$

in einem Vektor $h = (h_1, \ldots, h_d)$ zusammengefasst und die Zellmittelpunkte der j-ten Zelle stellen sich dar als

$$x_j^k \coloneqq \omega_{2k-1} + (j_k - 0.5)h_k, \quad j_k \in \{1, \ldots, m_k\},\ x^k \coloneqq (x_1^k, \ldots, x_{\bar{m}}^k)^{\mathsf{T}}.$$

Dann ist

$$x^{\mathcal{CC}} = x^{\mathcal{CC}}(\Omega, m) \coloneqq (x_j^k)_{\substack{j=1,\ldots,\bar{m} \\ k=1,\ldots,d}} = \begin{pmatrix} x^1 \\ x^2 \\ \vdots \\ x^d \end{pmatrix} \in \mathbb{R}^{d\bar{m}} \tag{2.10}$$

das zellzentrierte Gitter über dem Gebiet Ω mit $\bar{m}$ Diskretisierungspunkten.

Definition 2.3 folgt einer lexikographischen Ordnung $j \in \{1, \ldots, \bar{m}\}$. Das bedeutet, die Komponenten werden der Reihenfolge nach von Raumrichtung $x^1 \in \mathbb{R}^{\bar{m}}$ bis hin zu $x^k \in \mathbb{R}^{\bar{m}}$ gespeichert. Für die einzelnen Raumrichtungen werden die j_k sequentiell abgearbeitet. Zuerst läuft j_1 von 1 bis m_1, während alle weiteren $j_k = 1$ sind. Nach dem ersten Durchlauf erhöht sich j_2 auf $j_2 = 2$ und alle weiteren j bleiben 1. Dieser Vorgang läuft so lange, bis $j_2 = m_2$ gilt. Dann erhöht sich das nächste j_k, wofür der Vorgang analog läuft. Dies setzt sich fort, bis alle Zellen nummeriert sind. Dieser Ablauf ist in

Abbildung 2.1 für den zweidimensionalen Fall illustriert.

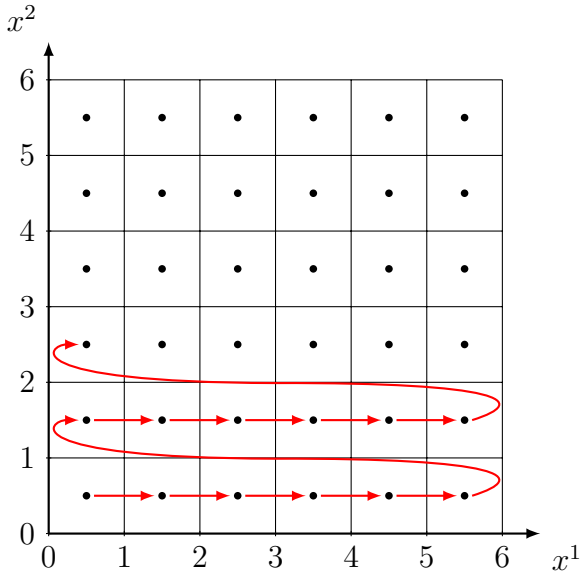


Abbildung 2.1: Die Pfeile illustrieren, wie sich das gezeigte, zweidimensionale, zellzentrierte Gitter lexikographisch aufbaut.

Neben dem zellzentrierten Gitter gibt es unter anderem nodale und auch gestaffelte Gitter, die nicht den Mittelpunkt der betrachteten Zellen interpolieren, sondern auf den Eckpunkten bzw. Kanten positioniert sind. Die Positionierung der Punkte für die verschiedenen Gittertypen ist in Abbildung 2.2 dargestellt.

2.5.2 Interpolation

Da das Referenzbild $\mathcal{R}$ nur auf den Punkten des vorgegebenen Gitters gegeben ist, benötigt man eine Möglichkeit, die Werte des durch y transformierten Gitters des Templatebildes $\mathcal{T}$ zu bestimmen. Dazu ist eine sogenannte Interpolationsfunktion $I : \mathbb{R}^d \to \mathbb{R}$ mit $I(x_j) = \mathcal{T}(x_j) \quad \forall\, j \in \{1, \ldots, \bar{m}\}$ nötig. Dabei soll $I(y_j), j \in \{1, \ldots, \bar{m}\}$ möglichst einfach berechenbar und $I \in C^1(\Omega)$ sein [Mod04].
Die folgend vorgestellten Interpolationsfunktionen sollen einen Überblick über verfügbare Methoden geben und sind deshalb nur eindimensional angegeben. Mithilfe des *Kroneckerprodukts* können diese Methoden jederzeit auf höhere Dimensionen erweitert werden [Mod04].

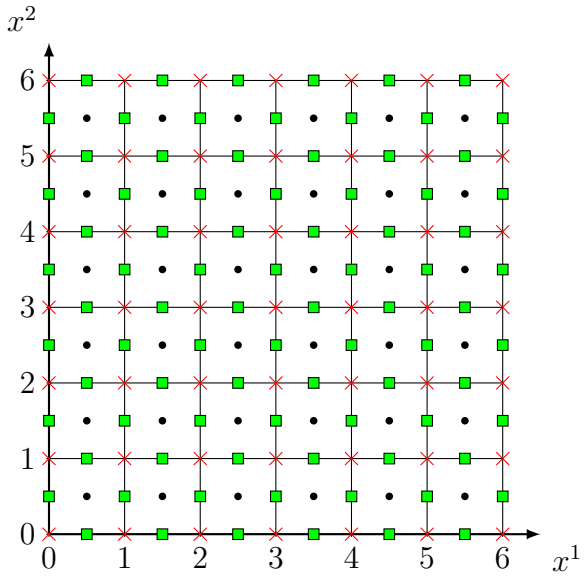


Abbildung 2.2: Zu sehen sind die verschiedenen Gittertypen. Die schwarzen Punkte stellen ein zellzentriertes Gitter dar. Die roten Kreuze auf den Ecken stellen das nodale Gitter dar und die grünen Quadrate sind eine Repräsentation eines gestaffelten Gitters.

Definition 2.4 (Kroneckerprodukt) *Seien $A \in \mathbb{R}^{m\times n}$ und $B \in \mathbb{R}^{p\times q}$ zwei Matrizen. Dann heißt*

$$A \otimes B \coloneqq \begin{pmatrix} a_{1,1}B & \dots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \dots & a_{m,n}B \end{pmatrix} \in \mathbb{R}^{mp\times nq} \tag{2.11}$$

das Kroneckerprodukt von A und B.

Nächste-Nachbarn-Interpolation

Die einfachste Art der Interpolation ist es, den nächsten Nachbarn zu wählen. Dieser Ansatz wird dargestellt als [ZF03]

$$I^{\mathrm{nn}}(y) = \begin{cases} \mathcal{T}(x_j), & |y - x_j| < |y - x_s| \,\forall s \in \{1, \dots, \bar{m}\} \\ 0, & y \notin \Omega. \end{cases} \tag{2.12}$$

Diese Interpolationsmethode ist zwar schnell und einfach berechenbar, jedoch an den Zellrändern nicht differenzierbar und im Inneren der Zellen gilt stets $I'(y) = 0$.

Lineare Interpolation

Die lineare Interpolation behebt das Problem der Nächste-Nachbarn-Interpolation und ist auch an den Zellrändern differenzierbar. Die Idee der linearen Interpolation ist die Darstellung der Stellen zwischen den Gitterpunkten mittels Konvexkombination. Mithilfe einer Transformation der Punkte [Mod09]

$$\Phi : \mathbb{R} \to \mathbb{R},\ y \mapsto \Phi(y) =: \hat{y} = \frac{y - \omega_1}{h_1} + 0.5$$

lässt sich die Darstellung der Interpolation vereinfachen. Dabei wird das Gebiet Ω abgebildet auf $\hat{\Omega} = (0.5, m_1 + 0.5)$, sowie die Gitterpunkte $\Phi(x_j) = j\, \forall j \in 1, \ldots, m_1$. So lässt sich jede zu interpolierende Zwischenstelle $\hat{y} \in \mathbb{R}$ in einen ganzzahligen Anteil $p \in \mathbb{Z}$ und einen Rest $r \in [0, 1)$ zerlegen. Aus r ergeben sich die gesuchten Gewichtungen der Konvexkombination als [Mod09]

$$r = \hat{y} - p \quad \text{mit } p = \lfloor \hat{y} \rfloor \coloneqq \max\{j \in \mathbb{Z} \,|\, j \leq \hat{y}\}.$$

Das bedeutet, dass die Gewichtung der Konvexkombination dem Abstand der Zwischenstelle $\hat{y}$ zu den nächsten Gitterpunkten, also r und $1 - r$ entspricht. Damit folgt:

$$I^{\text{lin}}(\hat{y}) = \begin{cases} \mathcal{T}(x_p)(1 - r) + \mathcal{T}(x_{p+1})r, & \hat{y} \in (0, m_1 + 1) \\ 0, & \text{sonst.} \end{cases} \tag{2.13}$$

Um Randartefakte zu verhindern, führt man üblicherweise noch die beiden Datenpaare $(0, 0)$ und $(m_1 + 1, 0)$ ein [Mod09].

Kubische Spline-Interpolation

Die Interpolation mit den bisher vorgestellten Verfahren ist zur Interpolation von Bildern zu ungenau. Die kubische Spline-Interpolation bietet einen – im Sinne der anfangs genannten Bedingungen – besseren Ansatz. Der Ansatz ist dabei die Minimierung der Krümmung unter den genannten Bedingungen als Nebenbedingungen [Mod09]. Das resultierende Minimierungsproblem lautet [Mod09]:

$$\mathcal{S}(I) = \int_\Omega (I''(x))^2 \, \mathrm{d}x \stackrel{!}{=} \min \text{ u.d.N. } I(x_j) = \mathcal{T}(x_j). \tag{2.14}$$

Lösen kann man dieses Problem durch Linearkombination kubischer Splines $b_j(y)$, $j \in \{1, \ldots, m_1\}$. Dabei sind die b_j Translationen der Basisfunktion b_0, auch „Mutterspline“

genannt [Mod09]. Weiterhin findet die Transformation Φ aus dem Abschnitt über lineare Interpolation Anwendung. Vernachlässigt ist im Folgenden – äquivalent zur Schreibweise in [Pol12] – die Notation des Zirkumflex über y. Mit $b_j(y) \coloneqq b_0(y-j)$ und

$$b_0(y) \coloneqq \begin{cases} (y+2)^3, & \text{für } -2 \leq y < -1, \\ -y^3 - 2(y+1)^3 + 6(y+1), & \text{für } -1 \leq y < 0, \\ y^3 + 2(y-1)^3 - 6(y-1), & \text{für } 0 \leq y < 1, \\ (2-y)^3, & \text{für } 1 \leq y < 2, \\ 0, & \text{sonst} \end{cases} \tag{2.15}$$

erhält man

$$I^{\text{spline}}(y) \coloneqq \sum_{j=1}^{m_1} c_j b_j(y), \quad c_j \in \mathbb{R},\, j = 1, \ldots, m_1. \tag{2.16}$$

Der entscheidende Unterschied zu den anderen vorgestellten Interpolationsarten ist der glatte Übergang der abschnittsweisen Splines, so dass eine insgesamt glatte Interpolationsfunktion entsteht. Vollständigkeitshalber sei noch angegeben, wie man die c_j bestimmt. Hierzu ist ein lineares Gleichungssystem folgender Gestalt zu lösen

$$B_{m_1} c = T$$

mit

$$B_{m_1} = \begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix} \in \mathbb{R}^{(m_1 \times m_1)} \text{ und } c = (c_1, \ldots, c_{m_1})^{\mathsf{T}}.$$

Hier ist T das diskretisierte Templatebild. Für eine genauere Herleitung sei auf geeignete Literatur – wie beispielsweise [TBU00] – verwiesen.
Schlussendlich erhält man durch die Tatsache, dass $b_0(y) = 0\ \forall y \notin (-2, 2)$ gilt, für jeden Punkt $y = p + r$ (wie in der linearen Interpolation definiert) [Mod09]

$$I^{\text{spline}}(y) = c_{p-1} b(r+1) + c_p b(r) + c_{p+1} b(r-1) + c_{p+2} b(r-2).$$

Aufgrund des kompakten Trägers der Basisfunktionen sind pro Basisfunktion nur vier Koeffizienten verschieden von Null.

Wie bereits erwähnt, werden diese Interpolationsansätze mittels Kroneckerprodukt auf die benötigte Dimension erweitert.

2.5.3 Diskrete Operatoren

Damit die in den bisher genannten Konzepten verwendeten Ableitungsoperatoren ebenfalls diskretisiert und implementiert werden können, benötigt man die diskreten Operatoren. Mithilfe finiter Differenzen lassen sich diskrete Begriffe für Ableitungen bzw. Gradienten sowie den Laplace-Operator einführen. Für $d = 2$ erhält man als Differenzenmatrix für eine erste Ableitung über zentrale Differenzen die folgende Darstellung, ohne Berücksichtigung bestimmter Randwerte [Mod09]:

$$D_j \coloneqq \frac{1}{2h_j} \begin{pmatrix} -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \dots & 0 & -1 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{m_j \times m_j}. \tag{2.17}$$

Mithilfe des Kronecker-Produkts aus Definition 2.4 erhält man die Differenzen in den verschiedenen Raumrichtungen durch:

$$\begin{aligned} \partial_1 &\coloneqq I_{m_2} \otimes D_1 \\ \partial_2 &\coloneqq D_2 \otimes I_{m_1}. \end{aligned} \tag{2.18}$$

Dabei ist I_{m_j} die jeweils passende Einheitsmatrix. Dadurch ist der diskrete Pendant des Gradientenoperators ∇, z.B. für das diskrete Templatebild, gegeben durch

$$\operatorname{grad} T \coloneqq \begin{pmatrix} \partial_1 T \\ \partial_2 T \end{pmatrix}. \tag{2.19}$$

Den Laplace-Operator Δ kann man mit

$$L_{1\mathrm{D}} = \begin{pmatrix} 1 & -2 & 1 \end{pmatrix} \tag{2.20}$$

für zwei Dimensionen erweitern auf

$$L_{2\mathrm{D}} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \tag{2.21}$$

Auch diesen kann man mithilfe des Kronecker-Produkts analog auf die benötigte Größe bringen.

3 Einführung in die Segmentierung

Das folgende Kapitel soll eine kurze Einführung in das umfangreiche Gebiet der Segmentierung liefern. Es wird sowohl in die Segmentierung von grauwertbasierten Daten als auch in die Grundlagen von Farbräumen und Segmentierungsideen für Farbbilder eingeführt. Diese Einführung legt einen Grundstein für die Idee der Vorbereitung einer Segmentierung mehrkanaliger Daten. Geeignete Literatur ist zur weiteren Vertiefung ausdrücklich empfohlen, da die folgende Ideensammlung nicht mehr als ein Verständnis dafür entwickeln soll, was in diesem Bereich für Möglichkeiten zur Verfügung stehen. Die Einführungen und Ideen dienen vorrangig einer Vorbereitung weiterführender Arbeiten. Es soll ein grober Überblick über das Themengebiet der Segmentierung gegeben werden, damit die Idee des später folgenden Registrierungsansatzes einen leichteren Anschluss an die Segmentierung finden und diese vorbereiten kann. Alle vorgestellten Ideen und Konzepte stützen sich, sofern nicht anders erwähnt, vor allem auf [FM81, PP93] und einige weitere im Verlaufe des Kapitels genannte Quellen.

3.1 Grauwertbasierte Segmentierung

Segmentierung ist das Zerlegen von Daten in verschiedene Regionen. Diese Regionen haben verschiedene Eigenschaften und können mehr oder weniger sinnvoll zur weiteren Verarbeitung der Daten sein. Die Segmentierung ist ein erster analytischer Schritt in der Bildverarbeitung und wird unter anderem im medizinischen Kontext auf grauwertbasierten Daten durchgeführt. Meist ist man daran interessiert, prominente Bildbereiche, wie z.B. Gefäßbäume, von umliegenden Strukturen abzugrenzen, um verschiedenste weitere Verarbeitungsschritte durchführen zu können. Im Bereich der grauwertbasierten Segmentierung ist eine grobe Einteilung der Modelle in drei Klassen möglich [FM81]:

1. Schwellenwertverfahren und Clustering
2. Kantendetektion
3. Regionsextraktion

Für alle drei Klassen sind in 3.1 am Ende des Kapitels vergleichbare Ergebnisse dargestellt. Um eine konkretere Betrachtung des Themas durchführen zu können, muss die Zerlegung der Daten in Regionen mathematisch definiert werden. Eine solche Definition liefern z.B. [PP93] oder [FM81] in folgendem Sinne:

Definition 3.1 (Segmentierung) *Sei I eine Menge von Bildpunkten und H ein Menge bestimmter Homogenitätskriterien. Dann ist die* Segmentierung *von I eine Zerlegung in $n \in \mathbb{N}$ zusammenhängende Untermengen S_i, so dass gilt*

$$\bigcup_{i=1}^{n} S_i = I \text{ mit } S_i \bigcap S_j = \emptyset \text{ für } i \neq j. \tag{3.1}$$

Dabei sind die Homogenitätskriterien H erfüllt für alle S_i aber für keine Vereinigung $S_i \bigcup S_j$ von benachbarten S_i, S_j.

3.1.1 Schwellenwertverfahren

Das Schwellenwertverfahren ist eine der klassischen Methoden der Segmentierung. Man legt einen oder mehrere Werte fest und setzt alle Pixel, die diesen unterschreiten, auf einen festgelegten Wert, während alle anderen Pixel einen anderen erhalten, um eine klare Trennung der Pixel vornehmen zu können. Oftmals werden diese Werte mit 0 und 1 belegt, was zu einer typischen Binarisierung der Daten führt. Mithilfe solcher Binarisierungen ist es möglich, Masken zu erstellen, die später auch anderweitig verwendet werden können. Eine solche Binarisierung könnte man lokal in 2D wie folgt definieren [PP93]:

Definition 3.2 (Lokale Binarisierung) *Sei $I(x, y)$ der Grauwert an der Stelle (x, y) und s ein festgelegter Schwellenwert. Dann ist die Binarisierung von I bezüglich des Schwellenwerts s definiert als*

$$I(x,y) = \begin{cases} 0, & I(x,y) < s \\ 1, & I(x,y) \geq s \end{cases} \tag{3.2}$$

Das Schwellenwertverfahren ist oftmals sinnvoll, wenn klare Strukturen voneinander getrennt werden sollen. Ein einfaches Beispiel dafür ist in Abbildung 3.1b zu sehen. Anwendbar ist es außerdem, wenn schon Vorarbeit geleistet worden ist, wie z.B. das Aufhellen bestimmter Bereiche, damit diese optisch klarer trennbar sind. Sind die Grauwerte allerdings eng miteinander verflochten, ist es schwierig hinreichend gute Ergebnisse mit einem so einfachen Verfahren zu erzielen. Es gibt nicht nur einfache Schwellenwertverfahren, sondern auch hysterese-basierte Verfahren. Solche Verfahren klassifizieren nicht

einfach nur aufgrund des Schwellenwerts, sondern ziehen auch noch andere Kriterien wie Nachbarschaften mit ein, um zusammenhängende Regionen finden zu können. Dass diese Art von Segmentierung gut funktionieren kann, wurde eindrucksvoll für Angiogramme in [CA05] gezeigt.

3.1.2 Clustering als Erweiterung des Schwellenwertverfahrens

In [FM81] wird gezeigt, dass man das Clustering als mehrdimensionale Erweiterung der Schwellenwertverfahren verstehen kann. Für das Clustering wird nicht mehr nur ein Kriterium herangezogen, sondern nach mehreren Eigenschaften klassifiziert. Diese Kriterien können nicht nur Grauwerte sein, sondern nahezu alle denkbaren Eigenschaften, die ein Datensatz mit sich bringen kann. Der Vorteil dabei ist, dass man eventuell auch unklare Strukturen besser klassifizieren kann. Unklare Strukturen könnten z.B. solche Strukturen sein, die fließende Grauwertübergänge an den gesuchten vermeintlichen Grenzen aufweisen. Bekannte Algorithmen dieser Gruppe sind unter anderem das k-Means-Verfahren oder das Fuzzy-C-Means-Verfahren [FM81]. Der Vorteil der Clustering-Algorithmen ist die oftmals einfache Berechnung, welche zu schnellen Ergebnissen führt, dabei aber trotzdem eine Vielfalt an Kriterien berücksichtigen kann. Für den k-Means-Algorithmus werden $k \in \mathbb{N}$ Clusterzentren zufällig initialisiert und die Datenpunkte werden aufgrund eines Abstandsmaßes den Clusterzentren zugeordnet. Mithilfe dieser Zuordnung werden die Clusterzentren während der Optimierung so lange verschoben, bis ein hinreichend gutes Ergebnis erreicht ist. Ein Beispiel ist in Abbildung 3.1c zu sehen.

3.1.3 Kantendetektion

Die Kantendetektion basiert als Segmentierungsverfahren auf Grauwertsprüngen. Es geht vor allem darum, große, abrupte Änderungen der Grauwerte benachbarter Pixel zu finden. Solche Unstetigkeitsstellen beschreiben den größten Teil der Struktur eines Bildes. Je nach Verfahren und Autor treten verschiedene Definitionen von Kanten in der Literatur auf. Richtet man sich nach [FM81] und [PP93], so kann man die Kantendetektion allgemein in zwei Klassen unterteilen. Einerseits gibt es die parallelen und andererseits die sequentiellen Methoden. Wenn man von Kantendetektion im Allgemeinen spricht, meint man oftmals die parallelen Methoden. Diese suchen Kantenelemente und können anschließend diese Elemente zu Kanten verbinden. Die sequentiellen Methoden hingegen prüfen nicht jeden Pixel für sich, sondern machen das Ergebnis, ob ein Kantenelement vorliegt, abhängig von benachbarten Pixeln.

Einer der bekanntesten Operatoren in der parallelen Anwendung ist der Gradientenoperator, wie er schon in [KWT88] für die Minimierung einer Kantenenergie von aktiven Konturen Anwendung gefunden hat:

$$-\left|\nabla I(x,y)\right|^2 \tag{3.3}$$

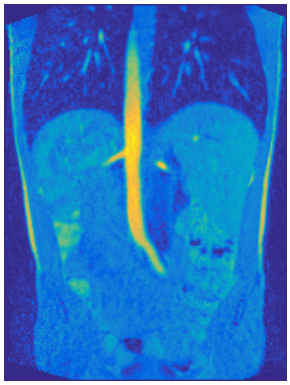
Es gibt unzählige verschiedene Versionen dieses Operators, die alle verschiedene Fensterungen haben, in denen die Grauwertsprünge betrachtet werden. Das Problem dieser Operatoren ist, dass nur sehr kleine Regionen betrachtet, an den Kanten also hohe Kontraste erwartet werden. Das macht diese Operatoren sehr rauschanfällig. Ein Vorteil solcher Verfahren ist die Existenz semiautomatischer Methoden, welche im Ablauf beeinflusst werden können. Es gibt auch andere Ansätze der Kantendetektion, z.B. mit Richtungsoperatoren. Dafür wird unter anderem versucht, Kanten als zusammenhängende Bereiche gleicher Richtung zu identifizieren. Ein Ergebnis einer möglichen Segmentierung mithilfe einer solchen aktiven Kontur ist in Abbildung 3.1d abgebildet.

3.1.4 Regionsextraktion

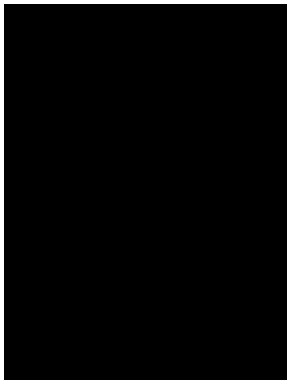
Die Regionsextraktion (*region extracting*) als solches kann nach [FM81] in zwei wesentliche Kategorien eingeteilt werden. Einerseits das Regionsverschmelzen (*region merging*) und andererseits das Regionsteilen (*region dividing*). Eigentlich wird in [FM81] angegeben, dass die Regionsextraktion in drei Kategorien eingeteilt werden kann, wobei die dritte Kategorie aber lediglich eine Kombination der beiden genannten ist. Ein prominenter Ansatz ist es, Regionen anwachsen zu lassen, was man unter dem Stichwort *region growing* z.B. in [Zuc76] finden kann. Hierzu ist ein Beispiel in Abbildung 3.1e abgebildet. Die Idee, wie in [Zuc76] sehr genau dargestellt, ist das Zusammenfügen von benachbarten Datenpunkten zu größeren, zusammenhängenden Regionen. Initialisiert werden diese Verfahren durch sogenannte Saatpunkte, die im besten Fall in verschiedenen Regionen liegen. Nachteil solcher Verfahren ist ein „Verwachsen“ der Regionen durch unklare Regionsränder, was dazu führen kann, dass das gesamte Bild lediglich eine einzige Region ist. Mehr zu diesen Verfahren in geeigneter Literatur.

3.1.5 Atlas-basierte Segmentierung

Die atlas-basierte Segmentierung passt nicht in eine der drei in [FM81] vorgestellten Kategorien. Dabei geht es nicht um die Segmentierung durch Auffinden von Kanten oder Unterscheidung von Grauwerten. Die atlas-basierte Segmentierung verwendet einen

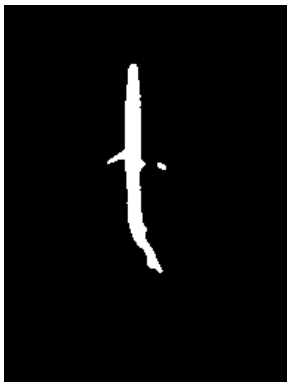


(a) Ein zweidimensionaler Schnitt einer Perfusions-MR-Aufnahme zu einem festen Zeitpunkt.



(b) Mögliches Ergebnis eines Schwellenwertverfahrens.

(c) Mögliches Ergebnis eines Clusterings.



(d) Mögliches Ergebnis einer Segmentierung mit aktiven Konturen.

(e) Mögliches Ergebnis einer Segmentierung mit wachsenden Regionen.

Abbildung 3.1: Vergleich der entstehenden Masken nach Segmentierung einer Aorta aus einem Perfusion-MR-Datensatz mithilfe der bisher genannten Segmentierungsverfahren. Für die Abbildungen 3.1b, 3.1c und 3.1d wurden selbst implementierte sowie MATLAB-Methoden verwendet. Für das Regionswachstum in 3.1e wurde der Algorithmus *Region Growing* von Dirk-Jan Kroon verwendet [Kro08]. Die MR-Daten in 3.1a wurden freundlicherweise zur Verfügung gestellt von Jarle Rørvik vom Haukeland University Hospital in Bergen, Norwegen.

sogenannten Atlas, der bereits segmentierte Regionen enthält. Dieser Atlas ist eine Art Templatedatensatz. Stelle man sich dazu ein menschliches Gehirn vor. Ein hierzu passender Atlas gibt die Segmentierung eines „durchschnittlichen“ menschlichen Gehirns vor, in dem die verschiedenen bekannten Regionen des Hirns markiert sind. Zur Segmentierung eines speziellen Datensatzes versucht man diesen Atlas möglichst passgenau über diesen Datensatz zu legen. Dazu sind möglichwerweise verschiedene, nichtlineare Transformationen notwendig. Abschließend erhält man eine Abbildung der Markierungen des Atlasdatensatzes auf den speziellen Datensatz. Im besten Fall sind nach einer solchen Segmentierung die gewünschten Regionen klar abgegrenzt.
Vorteil dieses Verfahrens sind die bereits markierten Regionen. Theoretisch benötigt man kein weiteres Wissen über die Strukturen als der Atlas schon liefert. Damit dieser Vorteil tatsächlich nutzbar ist, muss die Struktur des zu segmentierenden Datensatzes dem Atlas zumindest so ähnlich sein, dass man diese durch Deformation des Atlasdatensatzes näherungsweise erhalten kann. Dies ist zugleich der wohl größte Nachteil dieses Verfahrens. Angenommen, ein mit diesem Verfahren zu segmentierender Datensatz eines Hirns enthält – z.B. aufgrund einer Krankheit – eine bestimmte Hirnregion gar nicht, so kann der Atlas entweder gar nicht mit dem Datensatz überlagert werden oder muss so überlagert werden, dass diese Region bewusst ohne Überlagerung bleibt. Die atlas-basierte Segmentierung liefert eine direkte Schnittstelle von mathematischer Bildsegmentierung und mathematischer Bildregistrierung, denn die gesuchte Deformation wird mittels Registrierung bestimmt.

3.2 Farbräume und Farbraumsegmentierung

Das folgende Kapitel beschäftigt sich mit Farbräumen und zugehörigen Transformationen. Diese Zusammenhänge sollen beispielhaft aufgezeigt werden, um einen späteren Übergang auf abstraktere Kanäle bzw. Räume zu verstehen.
Gegeben sei zunächst ein dreidimensionaler Datensatz. Dieser Datensatz beschreibe im Sinne der folgenden Definition – welche sich mit Definition 2.1 überschneidet – ein nD-Bild mit k-Kanälen:

Definition 3.3 (Mehrkanaliges nD-Bild) *Ein* mehrkanaliges nD-Bild *sei eine Abbildung*

$$I : \mathbb{R}^n \to G \subseteq \mathbb{R}^k. \tag{3.4}$$

Dabei beschreibt $k \geq 1$ die Anzahl der Kanäle. Weiterhin sei $n \in \mathbb{N}$ und G berücksichtige eventuelle Einschränkungen. Außerdem seien die für Definition 2.1 genannten Eigenschaften erfüllt.

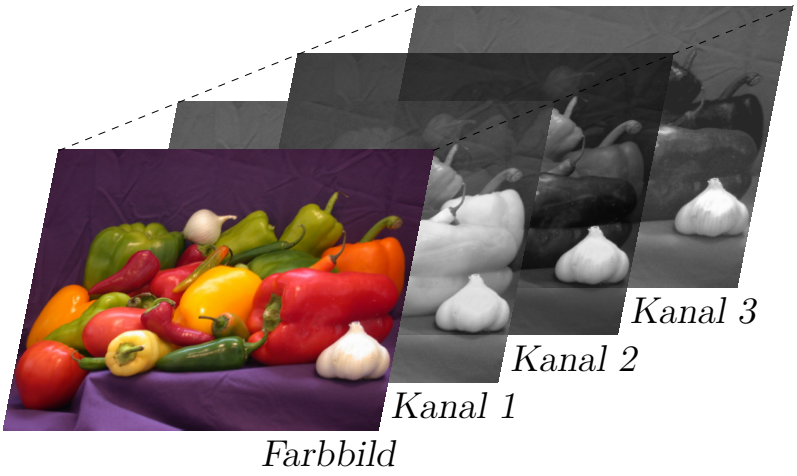


Abbildung 3.2: Beispiel eines mehrkanaligen Bilddatensatzes. Man sieht, dass sich die Farben des eigentlichen Bildes aufgrund der unterschiedlichen Informationen der Kanäle zusammensetzen. Die Abbildung zeigt die RGB-Kanäle des Bildes in der genannten Reihenfolge von vorne nach hinten. Das gezeigte Bild ist ein MATLAB-Standard namens „peppers“.

3.2.1 Der RGB-Farbraum und das HSI-System

Einer der üblich verwendeten Farbräume ist der RGB-Farbraum. Abbildung 3.2 zeigt ein Beispiel hierfür. Wie der Name vermuten lässt, ist dieser Raum dreidimensional, hat also

drei Kanäle. R für Rot, G für Grün und B steht für Blau. Liegen Daten dieses Farbraums vor, kann man einige Anwendungen direkt auf diesen Daten laufen lassen. Oftmals hat man durch die verschiedenen Kanäle, welche man als einzelne Grauwertbilder betrachten kann, einige Informationen mehr, als ein einzelnes Grauwertbild der gleichen Szene liefert. Dies kann man vor allem auch im Bereich der Segmentierung zum Vorteil nutzen und z.B. bewährte Algorithmen für Grauwertbilder auf den einzelnen Kanälen anwenden. Das Besondere an diesem Raum ist seine Linearität, weshalb er oft – wie in Abbildung 3.3 dargestellt – als Farbwürfel veranschaulicht wird.

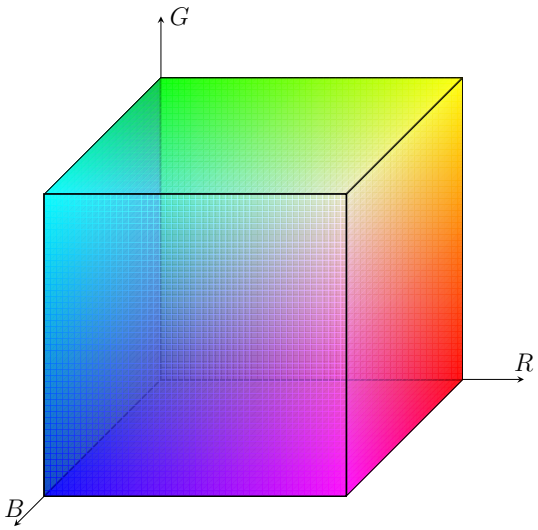


Abbildung 3.3: Visualisierung des RGB-Farbraums als Würfel

Dabei beschreibt jedes Koordinatentupel eine eindeutige Farbkombination, weshalb man mithilfe dieser drei Farbkomponenten jede Farbe darstellen kann [CJSW01]. Zur Visualisierung ist dieser Raum durchaus geeignet. Zur Segmentierung aber kann die starke Korrellation der einzelnen Komponenten untereinander zum Nachteil werden. Verändert sich z.B. die Helligkeit einer Szene, so ändern sich automatisch alle Komponenten. Aufgrund dessen versucht man zur Segmentierung oftmals in andere Farbräume zu wechseln, welche man mittels (nicht)linearer Transformationen des RGB-Farbraums erhalten kann.

Normalisierung des RGB-Raums

Da die Komponenten des RGB-Farbraums sehr voneinander abhängig sind, spielen Beleuchtungsänderungen für alle Komponenten gleichermaßen eine Rolle. Um die Beleuchtungsabhängigkeit in den einzelnen Komponenten vermindern zu können, ist es

manchmal schon ausreichend, die Farbkomponenten zu normalisieren, um den Einfluss der Lichtintensität auf das gesamte Farbspektrum zu verteilen [CJSW01]. Hier gibt es etwa Ansätze wie nRGB oder YT_1T_2, welche in [Nev77] vorgestellt werden.
Die Normalisierung findet für das nRGB-Modell durch Dividieren des Gesamtwertes statt. Also für $r = \frac{R}{R+G+B}$ und analog für die anderen Komponenten, so dass $r + g + b = 1$ gilt. Das Modell aus [Nev77] ist etwas anders normalisiert und stellt durch die Normalisierung eine Konvexkombination her:

$$\begin{aligned} Y &= c_1R + c_2G + c_3B \\ T_1 &= \frac{R}{R+G+B} \\ T_2 &= \frac{G}{R+G+B} \end{aligned} \tag{3.5}$$

Dabei sind c_1, c_2, c_3 beliebige Konstanten für die $c_1 + c_2 + c_3 = 1$ gilt. So kann man in der Komponente Y die Beleuchtung steuern.

Das HSI-System

Häufig verwendet wird auch das HSI-System. Dabei steht das Akronym HSI für Hue, Saturation und Intensity. Man überführt den RGB-Raum also in einen Raum, der nicht mehr nur Farbmischungen angibt, sondern einzelne Komponenten wie Helligkeit und Sättigung. Mithilfe dieser Parameter schlagen sich z.B. Helligkeitsänderungen in nur einer Komponente nieder, wohingegen alle anderen Komponenten unberührt bleiben.
Man kann sich vorstellen, dass die Raumdiagonale aus dem Ursprung in die vordere, obere Ecke des RGB-Würfels die Helligkeit wiedergibt. Diese Diagonale läuft von Schwarz im Ursprung nach Weiß in der äußersten Ecke des Würfels. Aus unter anderem solchen Zusammenhängen ergeben sich Formeln [CJSW01] für die Umrechnung in andere Systeme, wie das HSI-System. So berechnet sich die Intensität des HSI-Systems aus dem RGB-Farbraum als

$$I = \frac{R+G+B}{3}. \tag{3.6}$$

Diese Komponente wird oftmals für die Grauwertalgorithmen verwendet, da sich in dieser Komponente nur die Helligkeitswerte widerspiegeln. Für Datensätze mit schwankenden Helligkeitsverhältnissen bietet es sich an, den Hue-Kanal zu betrachten, welcher sich aus den RGB-Werten wie folgt berechnet:

$$H = \arctan\left(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right) \tag{3.7}$$

Man kann anhand der Formel erkennen, dass dieser Kanal einen Nachteil birgt. Sollte $R = G = B$ erfüllt sein, entsteht eine Singularität, die nicht beseitigt werden kann.
Für die Sättigung rechnet man

$$S = 1 - \frac{\min(R,G,B)}{I}. \tag{3.8}$$

Mit den Gleichungen (3.6), (3.7) und (3.8) kann man sich das HSI-System nun als Zylinder, wie in Abbildung 3.4 dargestellt, vorstellen. Die Rotationsachse, die durch die Mitte des Zylinders verläuft, beschreibt die Intensität. Die Entfernung von der Achse zum Mantel des Zylinders ist die Sättigung. Aus Formel (3.7) ersieht man, dass es sich für H um Winkel handelt. Der Mantel des Zylinders spiegelt das Spektrum der Farben in voller Intensität wieder. Zur Farbwahl dreht man den Zylinder und um die Helligkeit und Intensität zu regeln, bewegt man sich auf den senkrecht zueinander stehenden Achsen wie in einem kartesischen Koordinatensystem. Stellt man sich vor, man klebt ein A4-Blatt, auf dem das Farbspektrum abgebildet ist, an den kurzen Enden zusammen, hat man an der Klebestelle einen Sprung. So kann man sich auch die Singularität des H-Kanals vorstellen. Sie tritt nur unter einem bestimmten Winkel auf, verläuft aber praktisch über die gesamte Länge der Achsen der beiden anderen Kanäle.

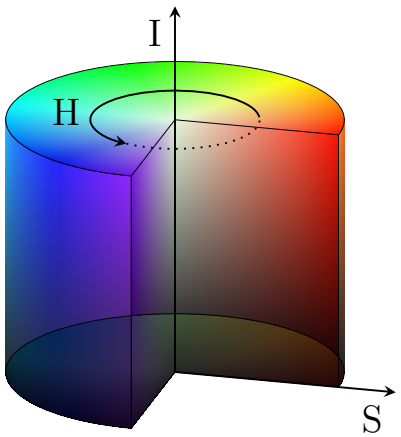


Abbildung 3.4: Visualisierung des HSI-Systems als Zylinder

Es gibt noch deutlich mehr Farbräume. An dieser Stelle reicht es allerdings, die Idee der Transformation von RGB in das HSI-System zu verstehen.
Nachdem nun klar ist, in welchen Dimensionen mit Farbdaten gearbeitet wird, kann man

dazu übergehen zu betrachten, wie man an das Segmentierungsproblem solcher Daten herangeht.

3.2.2 Segmentierung von Farbdaten

Die Segmentierung von Farbdaten ist in der Literatur längst nicht so ausgeprägt behandelt wie die Segmentierung von grauwertbasierten Daten. Oft zu finden sind Ansätze, die sich der grauwertbasierten Methoden bedienen und auf den verschiedenen Kanälen laufen gelassen werden. Denkt man allerdings daran, dass die Farbdaten neben der Intensität auch noch weitere Informationen liefern [CJSW01], die durchaus zur Segmentierung verwendet werden können, kann mit solchen Ansätzen nicht das volle Potential ausgeschöpft werden. Mittlerweile ist es nicht mehr denkbar, einige Anwendungen in der Computer Vision oder Pattern Recognition ohne Farbinformationen laufen zu lassen. Manchmal liefern mehrere Kanäle die entscheidenden Informationen, die benötigt werden, um eine saubere Trennung der Regionen durchführen zu können, welche mit reinen Grauwerten nicht funktionieren würde. Folgend sollen Ideen der Farbsegmentierung knapp erläutert werden.

Kantendetektion auf Farbdaten

Kanten sind – wie bereits in Kapitel 3 erwähnt – Unstetigkeiten in den Grauwerten. Für Farbdaten ist dies nicht ganz so simpel zu definieren. Für die angesprochenen Räume verteilt sich die Farbinformation über drei Kanäle und ist damit so verstreut, dass man besser nicht jeden Kanal einzeln nach Unstetigkeiten absucht. Stattdessen sollten Kanten auf Farbdaten vielmehr als Unstetigkeiten in einem mehrdimensionalen Raum angesehen werden [CJSW01]. Im Falle des RGB oder HSI-Systems liegen solche Unstetigkeiten z.B. im $\mathbb{R}^3$.
Die Entscheidung, welche Operatoren man zur Kantendetektion verwendet oder entwickelt, hängt von der Definition einer Kante auf den mehrdimensionalen Daten ab. Der Autor von [Nev77] schlägt drei verschiedene Möglichkeiten vor, wie man die gesuchten Unstetigkeiten definieren kann. Der erste Vorschlag handelt von einer metrischen Distanz auf dem verwendeten Farbraummodell. Welches Distanzmaß tatsächlich verwendet wird, ist anwendungsabhängig. Der Autor weist darauf hin, dass man bei der Verwendung einer solchen Definition allerdings tatsächlich ein niedriger dimensionales Problem betrachtet, welches wahrscheinlich keine deutlich anderen Ergebnisse liefern wird als eine Kantendetektion auf Grauwerten. Dies liegt an der Verminderung der Dimensionen durch die Verrechnung der Kanäle miteinander aufgrund des Distanzmaßes. Der zweite Vorschlag ist die Berechnung der Kanten auf jedem einzelnen Kanal mithilfe bekannter Methoden für grauwertbasierte Daten. Nach der Einzelberechnung müssen die Einzelergebnisse

miteinander verrechnet werden. Welche Methoden hierzu verwendet werden können, lässt der Autor von [Nev77] offen. Vorstellbar ist, dass man durch diesen Ansatz, wie auch schon im ersten Vorschlag, keine deutlich anderen Ergebnisse erzielen wird. Das ist davon abhängig, wie die einzelnen Ergebnisse verrechnet werden. Gerade in dieser Prozedur kann es allerdings zu diversen Schwierigkeiten kommen, die hier nicht weiter erläutert werden sollen. Der dritte Ansatz behandelt auch Kanten auf den einzelnen Kanälen. Dabei können die Kanten weitestgehend unabhängig voneinander sein. Die einzige Einschränkung in der Unabhängigkeit ist die Orientierung. Zur Detektion der Kantenorientierung kann man verschiedenste Operatoren verwenden. In [Nev77, CJSW01] wird z.B. der sogenannte *Hückel-Operator* aus [Hue71] für diesen Zweck eingesetzt. Dieser Operator betrachtet eine bestimmte Nachbarschaft von Bildpunkten und sucht nach Kantenpunkten, die vorgegebenen Kriterien entsprechen. Dabei werden die Kriterien und die möglichen Kantenpunkte durch Ausgleichsmethoden verglichen [SH81]. Da sich der Operator in der eigentlichen Definition aus [Hue71] auf Grauwertdaten beschränkt, werden in [Nev77, CJSW01] Erweiterungsmöglichkeiten vorgeschlagen. Für weitere Informationen zu diesem Verfahren sei auf die genannten Artikel sowie [Nev82] verwiesen.

Regionsbasierte Ansätze

Bei den regionsbasierten Ansätzen geht es zumeist – wie in Abschnitt 3.1.4 bereits erwähnt – darum, Bildpunkte in verschiedene, möglichst homogene Regionen einzuteilen. In [TB97] wird eine Mischung aus Regionswachstum (region growing) und Regionsverschmelzung (region merging) vorgestellt. Homogenitätskriterien werden über euklidische Distanzen auf den Farbkanälen definiert, wobei konkret drei Fälle entstehen, die sozusagen die Vergleichsrichtungen zur Homogenitätsfeststellung angeben. Es werden sowohl benachbarte Pixel und lokale Vergleiche von kleinen Regionen durchgeführt als auch globale Vergleiche der betrachteten Regionen. Ein Problem bei diesen Vergleichen ist, dass der Algorithmus z.B. an Schatten scheitert. Schatten spielen in der medizinischen Bildgebung eher eine kleinere Rolle, man könnte sich aber vorstellen, dass Überlagerungen oder schattenartige Artefakte Probleme bereiten könnten.

Teil II

Methoden

4 Methoden zur Zerlegung und Untersuchung von Parameterkarten

Dieses Kapitel soll aufzeigen, wie die Ideen der Segmentierung von mehrkanaligen Daten auf den in dieser Arbeit verfolgten Ansatz adaptiert werden können. Die Segmentierung mehrkanaliger Daten arbeitet wie zuvor beschrieben mit verschiedenen Kanälen, die unterschiedliche Informationen enthalten und sich teilweise auch untereinander verrechnen lassen. Die Zusammenhänge der Kanäle – wie z.B. im RGB-Modell – sind hinreichend untersucht, um eindeutige Zusammenhänge erklären zu können. Im Folgenden soll untersucht werden, mit welchen Methoden verschiedene Parameterkarten auf Zusammenhänge geprüft und welche Parameter eingesetzt werden können, um verschiedene Repräsentationen der Bilddaten zu erzeugen. Bezüglich der Parameterkarten wird sich das Kapitel auf Ideen und erste Eindrücke beschränken, da die weitere Untersuchung in folgenden Arbeiten fortgesetzt werden soll.
Die Ideen dieses Kapitels stützen sich vorwiegend auf die Artikel [Gan08, MSMC15a, Wat11] und [KF09].

4.1 Dekomposition mittels Singulärwertzerlegung

Die Singulärwertzerlegung – auch *singular value decomposition* (SVD) – kann die Frage klären, wie die Karten untereinander zusammenhängen und dient später zur Auswertung des Datenterms der Registrierung.
Die SVD einer komplexen Matrix liefert ein Produkt, dessen Gestalt in folgendem Satz beschrieben wird [Gan08]:

Satz 4.1 (Singulärwertzerlegung) *Sei $M \in \mathbb{C}^{m\times n}$ eine Matrix mit $m > n$ und $m, n \in \mathbb{N}$. Dann existieren Matrizen $U \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ sowie eine Diagonalmatrix $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{C}^{m\times n}$, so dass*

$$M = U\Sigma V^{\mathsf{H}} \tag{4.1}$$

gilt. Weiter gilt $\sigma_1, \ldots, \sigma_r > 0$ und somit $\operatorname{rang}(M) = r \in \mathbb{N}$.

Üblicherweise werden die σ_i, $i = 1, \ldots, r$, $r \in \mathbb{N}$ als Singulärwerte der Matrix $M \in \mathbb{C}^{m\times n}$ bezeichnet.

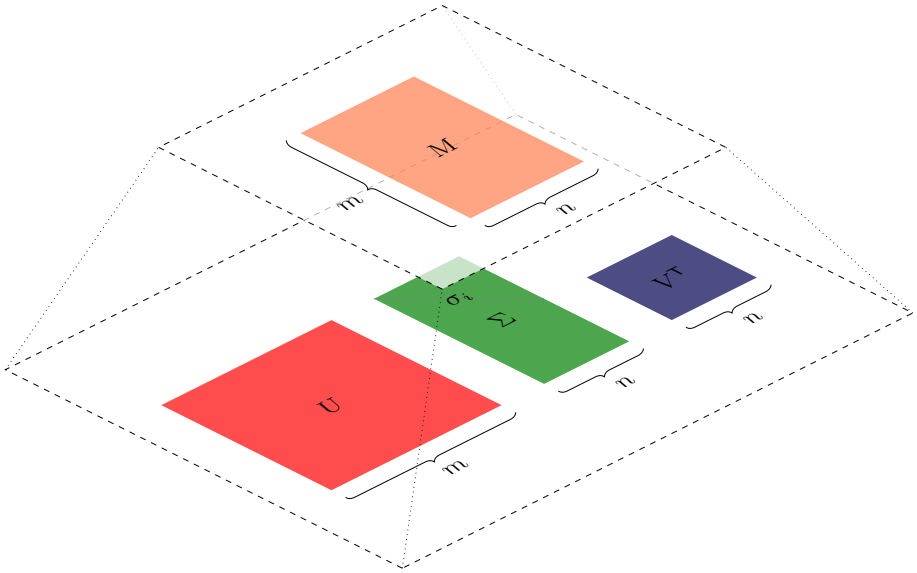


Abbildung 4.1: Visualisierung der Proportionen einer reellen Singulärwertzerlegung. Die obere Ebene zeigt die Matrix M, welche in die drei Matrizen $U\Sigma V^{\mathsf{T}}$ zerlegt werden kann. Diese sind in der darunter liegenden Ebene dargestellt. In der Darstellung von Σ ist die diagonale Anordnung der Einträge $\sigma_i, i \in \mathbb{N}$ aufgezeichnet. Alle anderen Einträge sind 0.

Folgend wird in einem konstruktiven Beweis die Singulärwertzerlegung über Normen für den reellen Fall hergeleitet [Gan08].

Beweis. Sei $\|M\| = \max\limits_{v\neq 0} \frac{Mv}{v} = \max\limits_{\|v\|=1} \|Mv\|$ die Matrixnorm von M. Dann existiert ein Vektor v mit $\|v\| = 1$, so dass $w = Mv$ und damit $\|w\| = \|M\| =: \sigma$.
Normiert man w zu $x = \frac{w}{\|w\|}$, erhält man $Mv = \sigma x$ für $\|v\| = \|x\| = 1$.
Erstellt man nun die Matrizen $V = [v, v_1]$ und $U = [x, u_1]$ so, dass v_1 und u_1 orthogonal auf v bzw. x stehen, erhält man

$$M_1 = U^{\mathsf{T}}MV = \begin{bmatrix} x^{\mathsf{T}}Mv & x^{\mathsf{T}}Mv_1 \\ u_1{}^{\mathsf{T}}Mv & u_1{}^{\mathsf{T}}Mv_1 \end{bmatrix} = \begin{bmatrix} \sigma & y^{\mathsf{T}} \\ 0 & N \end{bmatrix}.$$

Hier gehen die orthogonalen Konstruktionen von U und V ein.
Nun gilt es zu zeigen, dass $y^{\mathsf{T}} = x^{\mathsf{T}} M v_1 = 0$ ist. Man betrachte hierzu

$$M_1 \begin{pmatrix} \sigma \\ y \end{pmatrix} = \begin{pmatrix} \sigma^2 + \|y\|^2 \\ Ny \end{pmatrix}.$$

Nun ist

$$\left\| M_1 \begin{pmatrix} \sigma \\ y \end{pmatrix} \right\| = (\sigma^2 + \|y\|^2)^2 + \|Ny\|^2.$$

Mit der Abschätzung

$$(\sigma^2 + \|y\|^2)^2 + \|Ny\|^2 \geq (\sigma^2 + \|y\|^2)^2$$

und dem Wissen, dass $\|M_1\| = \|U^{\mathsf{T}} M V\| = \|M\| = \sigma$, folgt

$$\sigma^2 = \|M_1\|^2 = \max_{\|v\| \neq 0} \|M_1 v\|^2 \geq \frac{\left\| M_1 \begin{pmatrix} \sigma \\ y \end{pmatrix} \right\|^2}{\left\| \begin{pmatrix} \sigma \\ y \end{pmatrix} \right\|^2} \geq \frac{(\sigma^2 + \|y\|^2)^2}{\sigma^2 + \|y\|^2}$$

und damit

$$\sigma^2 \geq \sigma^2 + \|y\|^2.$$

Hieraus folgt $\|y\| = 0$. So erhält man

$$M_1 = U^{\mathsf{T}} M V = \begin{bmatrix} \sigma & 0 \\ 0 & N \end{bmatrix}.$$

Abschließend geht man analog für N vor und erhält am Ende aus M_1 die gesuchte Diagonalmatrix $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$. □

Mithilfe des Beweises kann man erkennen, dass die Singulärwerte σ einer Matrix M die Quadratwurzeln der Eigenwerte λ der Matrix $M^{\mathsf{H}} M$ sind. Vor allem wird auch die besondere Eigenschaft klar, dass man für jede Matrix mindestens eine Zerlegung finden kann.
Wie kann nun die SVD beim Verständnis des Zusammenhangs der Parameterkarten helfen?
Angenommen, man zerlegt eine Matrix $P \in \mathbb{R}^{d \times p}$, die verschiedene Parameterkarten

enthält, zu $P = U\Sigma V^{\mathsf{H}}$. Es liegen die verschiedenen Karten in der Matrix als Spaltenvektoren vor. Hier sind es p Parameterkarten. Somit ist $V^{\mathsf{H}} \in \mathbb{R}^{p\times p}$ und gibt die Mischungsverhältnisse der Parameterkarten untereinander wieder. Eine Veranschaulichung der Proportionen ist in Abbildung 4.1 auf Seite 40 zu sehen.
Aufgrund der Zerlegung mittels SVD kann die Matrix, welche die Parameterkarten enthält, genau untersucht werden. So werden unter anderem Zusammenhänge zwischen den Spalten klar. Da die Parameterkarten als Spaltenvektoren in die Matrix eingetragen sind, ist dies eine hilfreiche Methode der Untersuchung. Zerlegt man auf gleiche Art und Weise z.B. ein HSI-Bild, so sollten die gezeigten Farbzusammenhänge zwischen den Kanälen erkennbar sein. Man kann dies unter anderem einmal in MATLAB ausprobieren und wird feststellen, dass man die lineare Abhängigkeit von H, S und I mithilfe der SVD erkennen kann, selbst wenn man diese vorher nicht gekannt hätte.

4.2 Schatten-q-Normen

An dieser Stelle werden die sogenannten *Schatten-q-Normen* eingeführt. Diese dienen später in der Optimierung als Werkzeug zur Auswertung des Datenterms. Die Schatten-q-Normen basieren auf der in Satz 4.1 vorgestellten Singulärwertzerlegung, die zur Untersuchung der Kanalzusammenhänge dienen soll. Mithilfe der folgenden Norm-Definition 4.1 [MSMC15a, Wat11] kann anschließend ein Interpretationsspielraum geschaffen werden, der eine Art *Ausrichtung* der Kanäle denkbar macht. Eine solche Ausrichtung ist im Sinne einer Registrierung als auch im Sinne einer (Farb-)Rauschreduzierung – wie in [MSMC15a] – denkbar. Die Schatten-q-Normen lassen sich wie folgt definieren [Sch13]:

Definition 4.1 (Schatten-q-Norm) *Sei $A \in L(X,Y)$ ein Operator und $q \in \mathbb{R}_+$. Dann heißt*

$$\|A\|_{S,q} \coloneqq \left(\sum_i \sigma_i^q\right)^{\frac{1}{q}} \tag{4.2}$$

bzw.

$$\|A\|_{S,q} = \left[\mathrm{Tr}\left(\left(A^{\mathsf{T}}A\right)^{\frac{q}{2}}\right)\right]^{\frac{1}{q}} \tag{4.3}$$

die Schatten-q-Norm *von A. Dabei ist σ_i der i-te Singulärwert von A bzw.* Tr *der Spuroperator.*

Hinweis: Wählt man $q \in (0,1)$, so erhält man eine Quasinorm.
Die Schatten-q-Normen erfüllen einige nützliche Eigenschaften wie z.B.

- Monotonie: $\|A\|_{\mathrm{S},q} \geq \|A\|_{\mathrm{S},p}$ für $1 \leq q \leq p$, $A \in L(X,Y)$

- Isometrische Invarianz: $\|A\|_{\mathrm{S},q} = \|UAV^{\mathsf{T}}\|_{\mathrm{S},q}$ für $q \in [1,\infty]$, $A \in L(X,Y)$
- Submultiplikativität: $\|AB\|_{\mathrm{S},q} \leq \|A\|_{\mathrm{S},q}\|B\|_{\mathrm{S},q}$ für $q \in [1,\infty]$, $A, B \in L(X,Y)$ mit der Voraussetzung, dass AB existiert

Weiterhin kann man sehen, dass man für die Fälle $q = 2$ und $q = \infty$ bekannte Normen erhält. Für $q = 2$ erhält man die Frobenius-Norm und für $q = \infty$ erhält man die durch die Vektor-2-Norm induzierte Spektralnorm. Auch die Fälle $q = 0$ und $q = 1$ liefern nützliche Ergebnisse. Setzt man $q = 0$, erhält man einen Operator, der den Rang der Matrix wiedergibt und für $q = 1$ erhält man die sogenannte *nuclear norm*, auch Spurnorm oder Ky-Fan-Norm genannt. In diesem Fall werden die Singulärwerte aufsummiert.
In [MSMC15a] wird unter anderem eine Schatten-q-Norm für $q < 1$ zur Regularisierung im Rahmen einer Farbrauschreduzierung verwendet. Die Idee dort basiert auf einer Rangminimierung der Jacobimatrix der gegebenen Bilddaten mithilfe der Schatten-1-Norm. Die Autoren interpretieren die konvexe Relaxierung, erzeugt durch die Berechnung der Schatten-1-Norm (nuclear norm) der Jacobimatrix, als Rangminimierung. Setzt man hingegen $q = 0$, bekommt man – wie bereits erwähnt – tatsächlich den Rang der Matrix. Ausgehend von dieser Idee erkennen die Autoren, dass verschiedene Konfigurationen bezüglich des Rangs der Jacobimatrix zur Regularisierung in deren Falle sinnvoll sein können. Nimmt man beispielsweise an, der Rang der Jacobimatrix sei 1, dann bedeutet das die lineare Abhängigkeit des n-Tupels der in der Jacobimatrix eingetragenen Vektoren. Ist $n \in \mathbb{N}$ die Anzahl der Farbkanäle eines Farbbildes, so sind die Kanäle paarweise linear abhängig. Das bedeutet auch, dass die Normalenvektoren der Kanäle (anti-)parallel sein müssen und man – bildlich gesprochen – eine klare Grenze zwischen den Farben der einzelnen Kanäle zieht und keine unerwünschten Mischungen, wie sie durch Rauschen auftreten können, mehr vorliegen. Es wäre allerdings auch denkbar, dass ein solcher konvexer Rangminimierungsansatz der Jacobimatrix zu einem Rang von 0 führt. Auch diese Möglichkeit wird in [MSMC15a] angeführt und auf die Interpretation verwiesen, dass in diesem Falle alle Ableitungen Null seien, da dieser Fall nur für die Nullmatrix eintreten kann. Da man dort die Jacobimatrix betrachtet, bedeutet das nichts anderes, als dass die Kanäle untereinander punktweise keinerlei Änderungen mehr aufweisen. Warum die Autoren $q < 1$ wählen, wird darin begründet, dass die Verteilungsfunktion der Gradienten der Daten mithilfe nichtkonvexer Funktionen besser als mit konvexen Funktionen approximiert werden kann. Unter anderem wird das Paper [KF09] in [MSMC15a] referenziert, welches genau dieses Problem behandelt. Dort wird gezeigt, dass eine Dekonvolution von Bilddaten mithilfe nichtkonvexer Gradienten-Verteilungsfunktionen bessere Ergebnisse

liefert und sogar schneller berechnet werden kann. Den Autoren von [KF09] nach liegt dies daran, dass die nichtkonvexen Ansätze die Ausläufer der empirisch beobachteten Verteilungsfunktion der Gradienten besser erfasst.

4.3 Parameterkarten als Repräsentanten prominenter Bildmerkmale

Die sogenannten Parameterkarten sind Datensätze charakteristischer Parameter des Bilddatensatzes. Man kann z.B. für einen vierdimensionalen Perfusionsdatensatz aufgrund der Zeitabhängigkeit mehrere verschiedene dreidimensionale Karten erstellen. Die zeitabhängigen Intensitätskurven jedes einzelnen Bildpunktes können dazu herangezogen werden. So lassen sich diese Kurven z.B. auf Charakteristiken untersuchen, die mitunter durch spezielle Gewebearten erzeugt werden. Beispielsweise weisen die weiße oder graue Substanz des menschlichen Gehirns unter Kontrastmitteleinfluss verschiedene charakteristische Kurvenverläufe auf. Auch Gefäße lassen sich durch unterschiedliche Kurven kennzeichnen. Erzeugt man mithilfe verschiedener Parameter Karten dieser genannten Charakteristiken der Intensitätskurven, erhält man für jeden Datensatz individuelle Repräsentationen. Jede Karte zeigt dabei mehr oder weniger verschiedene, für diesen Parameter charakteristische Punkte des Datensatzes. Denkbar zur Kartenerstellung sind z.B. die Maxima der Intensitätskurven, die Fläche unter der Kurve oder die verstrichene Zeit bis zum Intensitätsmaximum. Dabei sollte man beachten, dass die Parameter möglichst robust gewählt werden. Das Maximum einer solchen Intensitätskurve kann durchaus eine extreme, durch Rauschen verursachte Spitze sein, welche die Karte verfälschen würde. Eine Idee zur robusten Ermittlung eines solchen Parameters wäre z.B. die Approximation der Intensitätskurve mit einer Gammakurve, also die Bestimmung der Parameter einer Gammaverteilung mit statistischen Methoden. Mehr zu diesem Thema sei in geeigneter Literatur wie z.B. [GD60] nachgeschlagen. Illustriert ist eine solche Gammakurve im Vergleich der eigentlichen Konzentrationsverlaufskurve in Abbildung 4.2. Konzentrationskarten entstehen durch anwendungsabhängige Umrechnungsmethoden aus den intensitätsbasierten Datensätzen. Auch hierzu sei auf weiterführende Literatur wie z.B. [Sou10, SB13] verwiesen.
Diese Gammakurven haben den Vorteil, dass einzelne Ausreißer nicht das eigentliche Maximum verfälschen, allerdings meist auch nur eine Tendenz des Maximums geben. Mithilfe eines solchen robusteren Modells können auch weitere Parameter deutlich robuster bestimmt werden.

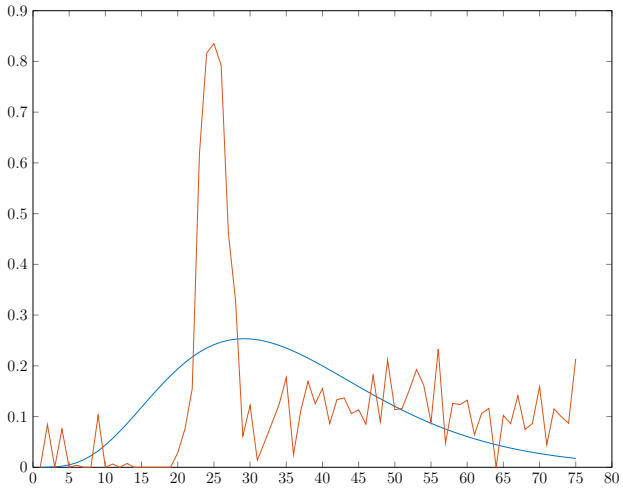
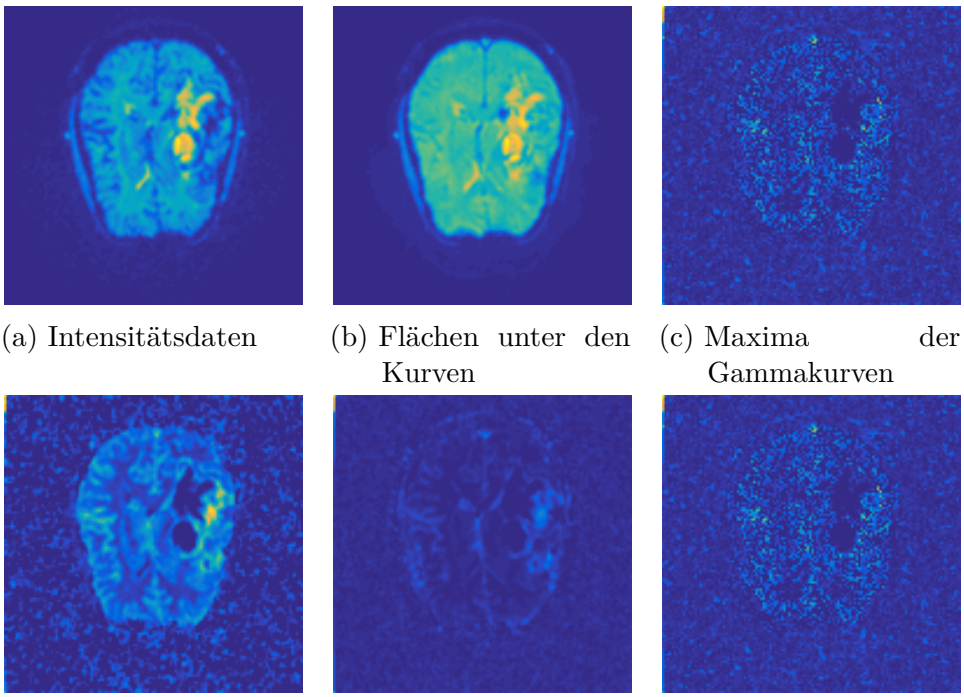


Abbildung 4.2: Eine Gammakurve im Vergleich mit der Konzentrationsverlaufskurve eines bestimmten Voxels eines Hirn-Perfusions-MR-Datensatzes. Die Gammakurve ist blau dargestellt.

In Abbildung 4.3 sind für einen Hirndatensatz, – freundlicherweise zur Verfügung gestellt von Rashindra Manniesing, DIAG-Gruppe des Radboud University Medical Centers, Nijmegen, Niederlande – Parameterkarten für die Intensitätsdaten und für die Konzentrationsdaten dargestellt.
Auffällig ist, wie ähnlich sich die Parameterkarten der Maxima der Gammakurven sind. Hier entsteht eine erste Vermutung, dass Parameter existieren, mit welchen man Karten erzeugen kann, die unabhängig von der Art der Datenrepräsentation sind. Für genau solche Untersuchungen könnten sowohl die SVD als auch die Schatten-q-Normen helfen. Die in Abbildung 4.3 gezeigten Karten sind unabhängig voneinander für intensitätsbasierte sowie für konzentrationsbasierte Daten entstanden.



(a) Intensitätsdaten (b) Flächen unter den Kurven (c) Maxima der Gammakurven

(d) Konzentrationskarte (e) Flächen unter den Kurven (f) Maxima der Gammakurven

Abbildung 4.3: Vergleich einiger Parameterkarten eines 2D-Ausschnittes aus einem 4D-Perfusionsdatensatzes eines menschlichen Gehirns. Dargestellt sind die Daten mit der MATLAB-eigenen Farbkarte „parula“. In (a) - (c) sind intensitätsbasierte Karten zu sehen, wohingegen (d) - (f) konzentrationsbasierte Karten zeigen. Der Datensatz wurde freundlicherweise zur Verfügung gestellt von Rashindra Manniesing, DIAG-Gruppe des Radboud University Medical Centers, Nijmegen, Niederlande.

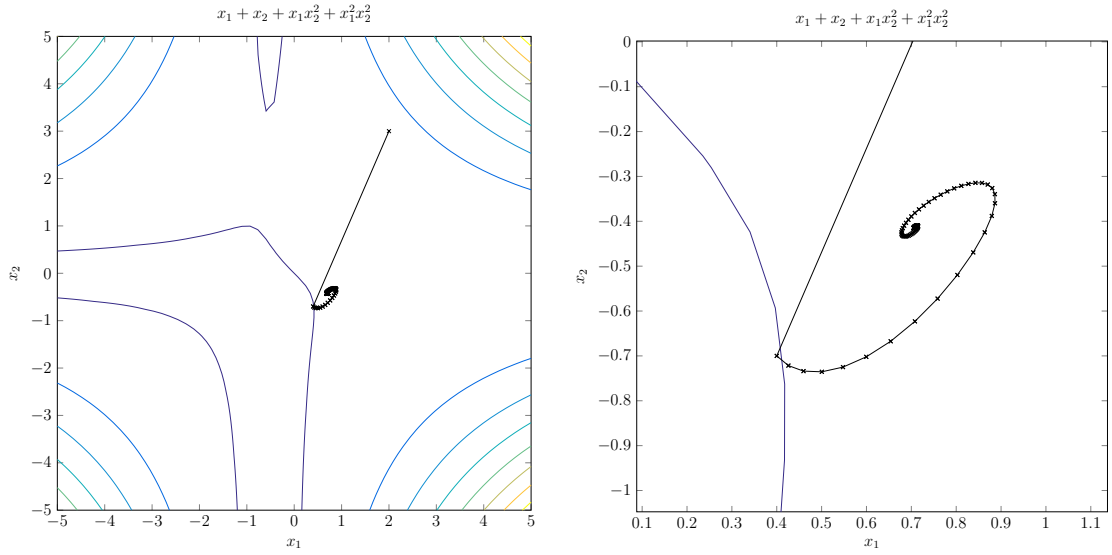
5 Optimierung

Dieses Kapitel soll erläutern, wie das in der vorliegenden Arbeit behandelte Problem optimiert wird. Dazu werden Grundlagen der mathematischen Optimierung wiederholt und Ideen sowie Interpretationen besprochen, um verstehen zu können, wie sich die spezielleren Verfahren aufbauen, welche später Anwendung finden sollen. Einige in den vorherigen Kapiteln besprochenen Ideen der Minimierung werden in diesem Kapitel aufbereitet. Sofern nicht anders erwähnt, beruhen alle Ideen und Ansätze in diesem Kapitel auf den genannten Quellen und Standardwerken wie [Him72, Dan98, Lue73, Ber99, Roc70].

5.1 Grundlagen der Optimierung

Nicht nur in der Wissenschaft ist es oft das Ziel, das Bestmögliche hinreichend anzunähern. Dafür gilt es, Eigenschaften und Bedingungen eines Problems möglichst klar zu formulieren. Nicht zuletzt ist eine Herausforderung der Optimierung die Formulierung einer „besten“ Lösung. Dies liegt meist daran, dass Eigenschaften und Bedingungen der betrachteten Probleme nicht klar formuliert sind [BZ13]. Die Mathematik liefert zwar die Werkzeuge, um eine Aufgabe in die nötige Abstraktheit zu setzen. Der kreative Prozess, um zu einer solchen Formulierung zu gelangen, bietet allerdings unzählig viele Möglichkeiten, diese Werkzeuge einzusetzen. Einerseits fordert man eine möglichst präzise Formulierung, andererseits muss die Komplexität zu der geforderten Lösungsgenauigkeit passen. Oftmals steigt die Komplexität, je genauer die Aufgabe formuliert wird. Das ist der Grund dafür, dass man versucht, wenige, wichtige Eigenschaften zu erfüllen, um die Problematik so genau wie möglich zu beschreiben. Gleichzeitig soll aber eine Einfachheit erhalten bleiben, sodass bekannte Methoden eingesetzt werden können. Steht eine hinreichend genaue Formulierung als sog. Zielfunktion fest, muss man mögliche Lösungen unterscheiden. Die allgemeine Lösungsmenge eines Minimierungsproblems kann man angeben als $\arg\min_{x \in D} f(x) = \{x \in D : f(x) \leq f(y) \, \forall y \in D\}$ [AHK+15]. Diese Formulierung liefert zwar eine notwendige Bedingung an die Lösungen, dadurch

enthalten diese aber noch keine Informationen darüber, wie die Menge D – in der die Lösungen enthalten sind – konkret aussieht. Es gibt unter Umständen auch Lösungen, die nicht in der Menge D liegen. Man unterscheidet dahingehend zwischen zulässigen und unzulässigen Lösungen. Nicht jedes Problem hat eine optimale Lösung. Für solche Fälle müssen Kriterien gefunden werden, die eine mögliche, optimale Lösung möglichst genau an die Formulierung der Problematik annähern. In der Praxis ist die eigentliche Existenz einer optimalen Lösung nicht die größte Schwierigkeit [AHK+15]. Oft reicht eine Näherungslösung. Bei sehr komplexen Zielfunktionen passiert es schnell, dass es viele lokale Optima gibt, die es sehr schwer gestalten, das tatsächliche, globale Optimum zu finden.



(a) Die Annäherung an das Minimum ist sichtbar. Vom Startpunkt aus ist ein großer Schritt möglich gewesen.

(b) Die Schritte passen sich in Richtung des Minimums immer weiter an, um ein möglichst genaues Ergebnis zu erzielen.

Abbildung 5.1: Ein Beispiel eines Gradientenabstiegs, dargestellt in Höhenlinienplots.

Ein Beispiel für ein sog. *unrestringiertes Optimierungsverfahren* – also ein Verfahren ohne Nebenbedingungen – ist der Gradientenabstieg der Form [Dan98]

$$b = a - \tau \nabla f(a). \tag{5.1}$$

Dabei ist die Abbildung $f : \mathbb{R}^n \to \mathbb{R}$ differenzierbar und $f(b) < f(a)$. Die Punkte a, b stammen aus der Definitionsmenge D_f von f. Die Idee liegt darin, einen möglichst schnellen Abstieg in kleinen Schritten zu finden. Angenommen, man steht auf einem Gipfel eines Berges und das Wetter schlägt so um, dass die Sicht sehr stark beeinträchtigt

ist. In einem solchen Fall gilt es, einen möglichst schnellen Abstieg zu finden. Dazu tastet man sich in kleinen Schritten jedes Mal neu in eine absteigende Richtung vor und erreicht am Ende des Abstieges das Tal (vgl. zu dieser Anekdote [AHK+15]). Ein Beispiel ist in Abbildung 5.1 zu sehen. Dies bedeutet, übertragen auf die mathematische Formulierung in (5.1), dass man einen Ausgangspunkt a, eine Schrittweite $\tau \in \mathbb{R}$ und eine Abstiegsrichtung $-\nabla f(a)$ benötigt. Damit der Gradient $\nabla f(a)$ gebildet werden kann, muss die Funktion an der Stelle a glatt, also differenzierbar, sein. Bildet man den Gradienten, so erhält man den steilsten Anstieg in a und mit umgekehrtem Vorzeichen auch den steilsten Abstieg.
Die restringierte Optimierung behandelt Probleme mit Nebenbedingungen. Diese Nebenbedingungen beschreiben eine Menge, auf der optimiert werden soll. Der Lösungsraum wird dadurch unter Umständen eingeschränkt. Ein möglicher Lösungsweg ist die Umformulierung der Zielfunktion. Man versucht, die Nebenbedingungen direkt mit einzubeziehen. Einen solchen Ansatz beschreiben die sog. *Lagrange-Multiplikatoren*. Die neue Zielfunktion wird aus der eigentlichen Zielfunktion und den Nebenbedingungen in einer Linearkombination zusammengefasst. Dabei sind die Faktoren der Nebenbedingungen die sogenannten Lagrange-Multiplikatoren oder auch *duale Variablen*. Betrachtet man der Einfachheit halber das lineare Optimierungsproblem [Dan98]

$$\begin{aligned} \min\ & c^\mathsf{T} x \\ \text{unter } & Ax \leq b \\ & x \geq 0 \end{aligned}$$

mit $A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m, c \in \mathbb{R}^n$, erhält man als Lagrangefunktion $L(x, \lambda) = c^\mathsf{T} x + \lambda^\mathsf{T}(Ax - b)$. Hier ist $\lambda \in \mathbb{R}$ der Lagrange-Multiplikator. Der alternative Begriff der dualen Variablen kommt daher, dass man das Problem als Maximierungsaufgabe über λ umschreiben kann, welches man das *duale Problem* nennt. Für den vorliegenden Fall kann man zur sogenannten Standardform für lineare Optimierungsprogramme über die *Lagrange-Dualität* gelangen [Dan98].
Diese Theorien lassen sich für simple Aufgaben gut anwenden. Für komplexere Problematiken besteht oft die Schwierigkeit, dass man nicht nur differenzierbare Funktionen mit oder ohne Nebenbedingungen optimieren muss, sondern auch die Differenzierbarkeit nicht notwendigerweise als gegeben anzunehmen ist.

5.2 Konvexe Optimierung

Das genannte Gradientenverfahren in (5.1) ist eine frühe Entwicklung der konvexen Optimierung, dazu zählen unter anderem auch lineare Programme. Oft sind es allerdings nicht lineare Probleme, die die Realität hinreichend approximieren. Liegt Konvexität vor, findet man meist auf effizientem Wege eine brauchbare Lösung. Umstände können teilweise die Randbereiche bereiten. Insbesondere bei den nichtlinearen, konvexen Problemen muss das Optimum nicht mehr notwendigerweise im Randbereich liegen. Weiterhin ist die konvexe Optimierung insofern schwieriger zu behandeln, als dass auch Funktionen auftreten können, die konvex aber nichtglatt sind. In dem angeführten Kontext nichtglatter Funktionen müssen allerdings einige Begriffe eingeführt werden, die beispielsweise eine Art Ableitungsbegriff bzw. Gradienten für nichtglatte Funktionen beschreiben. Zunächst soll der Begriff der Konvexität wiederholt werden [Ber99]:

5.2.1 Konvexität

Definition 5.1 *Eine Teilmenge $C \subset \mathbb{R}$ ist genau dann konvex, wenn*

$$(1-\lambda)x + \lambda y \in C \tag{5.2}$$

für alle $x, y \in C$ und $\lambda \in (0,1)$.

Äquivalent definiert man die Konvexität einer Funktion f als

$$f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y) \quad \forall x, y \in \mathbb{R}^n,\ \lambda \in (0,1). \tag{5.3}$$

Ist eine solche Abbildung f differenzierbar auf einer konvexen Menge C, erhält man einige nützliche Eigenschaften, wie unter anderem:

- f ist strikt konvex,
- $f(x) + \langle x - y, \nabla f(x) \rangle \leq f(y) \quad \forall x, y \in C.$

Mithilfe der Konvexität lässt sich ein Minimum sofort als lokal oder global klassifizieren, denn jedes lokale Minimum einer konvexen Funktion ist automatisch ein globales Minimum. Ist die betrachtete Funktion strikt konvex, so ist der globale Minimierer eindeutig.

5.2.2 Subdifferential

Für glatte Funktionen findet man einen eindeutigen Gradienten an jedem Punkt. Nichtglatte, konvexe Funktionen hingegen haben eine Menge von Subgradienten. So kann auch an nicht differenzierbaren Stellen eine Art Differential gebildet werden. Zwischen der Konvexität einer Funktion und der Bildung des Subdifferentials besteht ein Zusammenhang. Ist die Funktion konvex, kann man an jeder Stelle das Subdifferential bestimmen. Andersherum ist eine Funktion nicht konvex, wenn man eine Stelle finden kann, an der das Subdifferential die leere Menge ist. Einen Beweis hierzu findet man z.B. in [BZ13]. Das Subdifferential ist definiert als [Roc70]

Definition 5.2 *Sei $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ und $x \in \mathbb{R}^n$. Dann ist*

$$\partial f(x) \coloneqq \{v \in \mathbb{R}^n \mid f(x) + \langle v, y - x\rangle \leq f(y) \quad \forall y \in \mathbb{R}^n\} \tag{5.4}$$

die Menge der Subgradienten von f an der Stelle x.

Mithilfe der Definition 5.2 kann man ein Optimalitätskriterium für konvexe Funktionen formulieren: *x ist genau dann ein globaler Minimierer, wenn* $0 \in \partial f(x)$. Außerdem gilt für Funktionen, die an der Stelle x differenzierbar sind, dass $\partial f(x) = \{\nabla f(x)\}$.
Im weiteren Verlauf ist es sinnvoll, die Regeln der Invertierbarkeit des Subdifferentials zu kennen, um verschiedene Umformungen bezüglich primal-dualer Schreibweisen verstehen zu können. Zu Hilfe genommen sei dafür die *Legendre-Fenchel Transformation*, die wie folgt definiert ist [Roc70]:

Definition 5.3 *Sei $f : \mathbb{R}^n \to \bar{\mathbb{R}}$. Dann heißt*

$$\begin{aligned} f^* &: \mathbb{R}^n \to \bar{\mathbb{R}}, \\ f^*(v) &\coloneqq \sup_{x \in \mathbb{R}^n} \{\langle v, x\rangle - f(x)\} \end{aligned} \tag{5.5}$$

die Konjugierte von f. Die Abbildung $f \mapsto f^*$ heißt Legendre-Fenchel Transformation.

Mithilfe dieser Definition kann man folgenden, wichtigen Satz definieren, der zur Inversionsregel für Subdifferentiale führt [Roc70]:

Satz 5.1 *Sei $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ eine eigentliche, unterhalbstetige, konvexe Abbildung. Dann gilt*

$$f^{**} = f \tag{5.6}$$

Dadurch kann die Inversion des Subdifferentials vereinfacht werden:

Satz 5.2 *Sei $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ eine eigentliche, unterhalbstetige, konvexe Abbildung. Dann ist*

$$\partial f^* = (\partial f)^{-1} \tag{5.7}$$

Genauere Ausführungen und Beweise zu diesen Theoremen können in geeigneter Literatur zu konvexer Analysis – wie z.B. [Roc70] – gefunden werden.
Satz 5.2 impliziert, dass

$$v \in \partial f(x) \Leftrightarrow f(x) + f^*(v) = \langle v, x\rangle \Leftrightarrow x \in \partial f^*(v).$$

Betrachtet man dazu die Definition 5.2 der Menge der Subgradienten, so erhält man die Formulierungen [Roc70]

$$\begin{aligned} \partial f(x) &= \arg\max_v \{\langle v, x\rangle - f^*(v)\}, \\ \partial f^*(v) &= \arg\max_x \{\langle v, x\rangle - f(x)\}. \end{aligned} \tag{5.8}$$

5.2.3 Dualität und Lagrange-Multiplikatoren

Wie bereits im Vorwort dieses Kapitels für lineare Programme beispielhaft aufgezeigt, ist es manchmal sinnvoll, eine sogenannte Lagrangefunktion zu formulieren. Sobald man nicht mehr nur lineare Programme betrachtet, wird die Theorie zu Lagrangefunktionen allerdings etwas umfangreicher. An dieser Stelle soll anhand eines beispielhaften Problems gezeigt werden, wie man mithilfe der Dualität und einer Sattelpunktformulierung nach Lagrange ein solches, nichtlineares Optimierungsproblem formulieren kann.
Gesucht sei das Infimum einer separierbaren Abbildung der Form [CP11, Lel13]

$$\inf_{u\in\mathbb{R}^n} f(u) = g(u) + h(Au), \tag{5.9}$$

wobei g und h die erforderlichen Bedingungen – also unter anderem Konvexität – erfüllen. Mit der vorausgegangenen Theorie kann man hierzu auch ein duales Problem formulieren, welches lautet [CP11]:

$$\sup_{v\in\mathbb{R}^m} -h^*(v) - g^*(-A^\mathsf{T} v). \tag{5.10}$$

An dieser Stelle fließt in die konkave, duale Zielfunktion die duale Variable v ein. Das duale Problem allein ist allerdings nicht unbedingt einfacher zu lösen als das ursprüngliche primale Problem in (5.9). Zur Lösung der Aufgabe hilft es, die Lagrangefunktion zu betrachten, welche eine Sattelpunktformulierung liefern wird, da sowohl ein Minimierungs- als auch ein Maximierungsproblem gegeben ist. Wie die zugehörige Lagrangefunktion

aussieht, erfährt man, wenn man die Überführung des primalen in das duale Problem aufschlüsselt [Lel13]:

$$\begin{aligned}
&\inf_{u\in\mathbb{R}^n} g(u) + h(Au) && (5.9)\\
\overset{\text{Satz 5.2}}{=}\ &\inf_{u\in\mathbb{R}^n} g(u) + h^{**}(Au)\\
\overset{\text{Def. 5.3}}{=}\ &\inf_{u\in\mathbb{R}^n} g(u) + \sup_{v\in\mathbb{R}^m}\{\langle Au, v\rangle - h^*(v)\}\\
=\ &\inf_{u\in\mathbb{R}^n}\sup_{v\in\mathbb{R}^m}\{g(u) + v^\mathsf{T}Au - h^*(v)\} && (5.11)\\
=\ &\sup_{v\in\mathbb{R}^m}\inf_{u\in\mathbb{R}^n}\{g(u) + v^\mathsf{T}Au - h^*(v)\}\\
=\ &\sup_{v\in\mathbb{R}^m} - \sup_{u\in\mathbb{R}^n}\{h^*(v) + \langle -A^\mathsf{T}v, u\rangle - g(u)\}\\
\overset{\text{Satz 5.2}}{=}\ &\sup_{v\in\mathbb{R}^m} -h^*(v) - g^*(-A^\mathsf{T}v) && (5.10)
\end{aligned}$$

An der Umformung kann man sehen, wie man vom primalen (5.9) zum dualen Problem (5.10) kommt. Dabei erhält man in (5.11) die Sattelpunktformulierung, in der man sieht, dass man in Richtung u minimieren und in Richtung v maximieren muss.
Da Sattelpunktformulierungen nach Lagrange hinreichend untersucht sind, kann man direkt Optimalitätskriterien angeben, welche lauten [Lel13]:

$$u' \in \arg\min_{u\in\mathbb{R}^n} L(u, v') = g(u) + \langle u, A^\mathsf{T}v'\rangle - h^*(v') \tag{5.12}$$

$$v' \in \arg\max_{v\in\mathbb{R}^m} L(u', v) = g(u') + \langle Au', v\rangle - h^*(v). \tag{5.13}$$

Hier ist L eine Lagrangefunktion der Form $L(x, y) = \inf_u\{f(x, u) - \langle y, u\rangle\}$ [Roc70]. Sind diese beiden Kriterien erfüllt, ist das Tupel (u', v') eine Primal-Duale-Lösung. Damit weiß man für jeden primalen Punkt u, sowie auch für jeden dualen Punkt v, dass gilt [Lel13]:

$$f_d(v) \leq f_d(v') = L(u', v') = f(u') \leq f(u). \tag{5.14}$$

Dabei steht f_d für die duale Funktion [Lel13]. Hieran sieht man, dass – wie bereits erwähnt – die Aufgabe aus zwei Optimierungsproblemen besteht, da der Optimierer an einem Sattelpunkt liegt.
Eine Lösung dafür kann über viele verschiedene Wege gefunden werden. An dieser Stelle

soll die Konvexität ausgenutzt werden. Zur Bereitstellung einer sequentiellen Lösung muss das Problem in kleine, einfachere Probleme zerlegt werden. Dabei helfen *Rückwärtsschritte* bzw. das sogenannte *proximal mapping*, wie in [Roc70, CP11, MSMC15a] erwähnt.

Definition 5.4 *Sei f eine geeignete, unterhalbstetige Abbildung mit $f : X \to \mathbb{R} \bigcup \{\infty\}$. Dann ist das* proximal mapping *gegeben durch:*

$$\text{prox}_{\tau,f}(y) \coloneqq \arg\min_x \frac{1}{2}\|x - y\|^2 + \tau f(x). \tag{5.15}$$

Das proximal mapping ist genau dann wohldefiniert, wenn $f(x)$ konvex ist, denn die Norm $\|x - y\|^2$ ist strikt konvex.
Später wird außerdem die Moreau-Identität hilfreich sein [Roc70]:

$$\text{prox}_{\sigma,f^*}(x) = x - \sigma\,\text{prox}_{\sigma^{-1},f}(\sigma^{-1}x). \tag{5.16}$$

Das zugrunde liegende Optimierungsproblem lautet:

$$\text{Finde } x \in \mathbb{R}^n \text{ so, dass } 0 \in \partial f(x).$$

Manchmal ist Definition 5.4 in der Literatur auch als Rückwärtsschritt zu finden. Den Sinn dieser Benennung erkennt man mithilfe dieses Optimierungsproblems und der Darstellung [Roc70]

$$x^{k+1} \in x^k - \tau_k \partial f(x^{k+1}). \tag{5.17}$$

Man sieht hier, dass im Gegensatz zu einem Vorwärtsschritt nicht das x^k für die Berechnung der nächsten Iterierten verwendet wird, sondern die nächste Iterierte x^{k+1} selbst. Dabei ergibt sich aus der Darstellung (5.17) und Definition 5.4 durch Umformulierung der Zusammenhang, dass die neue Iterierte in der Minimierung zusammen mit der gegebenen Funktion nah an der alten Iterierten gehalten wird. Die erwähnte Umformulierung sieht wie folgt aus [Lel13]:

$$\begin{aligned} & y \in \arg\min \left\{ \frac{1}{2}\|y - x\|_2^2 + \tau f(y) \right\} \\ \Leftrightarrow\ & 0 \in y - x + \tau \partial f(y) \\ \Leftrightarrow\ & x \in (I + \tau \partial f)(y) \\ \Leftrightarrow\ & y \in (I + \tau \partial f)^{-1}(x). \end{aligned} \tag{5.18}$$

Der Zusammenhang zwischen (5.17) und (5.18) wird klar, wenn man überlegt, woraus die Darstellung in (5.18) folgt [Lel13]. Die üblichen Vorwärtsschritte des Gradientenabstiegs $x_t = -\nabla f(x)$ werden diskretisiert als $x^{k+1} = x^k - \tau_k \nabla f(x^k)$, mit Schrittweite τ_k. Sobald f allerdings nichtglatt ist, ist diese Diskretisierung nicht mehr möglich. Unter bestimmten Voraussetzungen, die in den genannten Standardwerken ausführlicher beschrieben sind, ist es möglich, die zu diskretisierende, partielle Differentialgleichung umzuschreiben in $\frac{\mathrm{d}}{\mathrm{d}t}x(t) \in -\partial f(x(t))$. Die Diskretisierung kann nun in einem Vorwärts- oder einem Rückwärtsschritt realisiert werden und führt so auf (5.18). Dabei wäre $(I - \tau_k \partial f)x^k$ der resultierende Vorwärtsschritt.
Zurück zum eigentlichen Problem: Mit den Gleichungen (5.12) und (5.13) war ein Sattelpunktproblem gegeben. Die Idee mithilfe des proximal mapping aus Definition 5.4 ist, dass man eine Richtung fixiert und in die jeweils andere Richtung Rückwärtsschritte durchführt. Fixiert man erst die u-Richtung in (5.13) und danach die v-Richtung in (5.12), so erhält man mit abschließendem Update folgenden Algorithmus [CP11]:

Algorithmus 5.1 :

$$v^{k+1} = \mathrm{prox}_{\sigma h^*}(v^k + \sigma A \bar{u}^k)$$
$$u^{k+1} = \mathrm{prox}_{\tau g}(u^k - \tau A^\mathsf{T} v^{k+1})$$
$$\bar{u}^{k+1} = u^{k+1} + \theta(u^{k+1} - u^k)$$

Man kann für diesen speziellen Fall zeigen, dass die Konvergenz unter den Bedingungen $\theta = 1$ und $\sigma\tau < \frac{1}{\|A\|^2}$ sichergestellt ist [CP11].

5.3 Nichtkonvexe Optimierung

Für Algorithmus 5.1 ist ein besonderes Detail zu beachten. Meist ist es nur sehr schwer möglich, die Duale einer Funktion explizit zu bestimmen, wie es für den ersten Updateschritt gefordert ist. Damit tatsächlich Algorithmus 5.1 erhalten wird, muss $h(Au)$ – aus dem eigentlichen Problem (5.9) – konvex sein. Ist dies nicht der Fall, kann man versuchen, eine andere Formulierung zu finden, mithilfe derer man die Duale eliminiert. In [MSMC15a, MSMC15b] hat man sich intensiv damit auseinander gesetzt. Dort wird angenommen, dass $h(Au)$ quasikonvex ist, was bedeutet, dass h^{-1} – falls die Inverse existiert – konvex ist [DHK16]. Für die *Quasikonvexität* muss die Bedingung [Him72]

$$\lambda x + (1-\lambda)y \leq \sup\{h(x), h(y)\}, \quad \forall x, y \in \mathbb{R}^n, \lambda \in (0,1) \tag{5.19}$$

erfüllt sein. Diese Bedingung ist schwächer als die Konvexitätsbedingung in Definition 5.1.

Sei das folgende Problem – wie in [MSMC15a, MSMC15b] zu finden – definiert als:

$$\min_{u,g} G(u) + F(g) \text{ mit } g = Ku. \tag{5.20}$$

Durch die Bedingung, dass F nichtkonvex sein kann, ist dieses Problem eine Verallgemeinerung des eigentlichen Problems (5.9). Alle anderen Eigenschaften sind wie für (5.9).
Die Autoren von [MSMC15a, MSMC15b] geben an, dass der Algorithmus unter den genannten Bedingungen wie folgt aussehen kann [MSMC15a]:

Algorithmus 5.2 :

$$\begin{aligned} g^{n+1} &\in \text{prox}_{\sigma^{-1},F}(K\bar{u}^n + \sigma^{-1}q^n) \\ q^{n+1} &= q^n + \sigma(K\bar{u}^n - g^{n+1}) \\ u^{n+1} &= \text{prox}_{\tau,G}(u^n - \tau K^\mathsf{T} q^{n+1}) \\ \bar{u}^{n+1} &= u^{n+1} + \theta(u^{n+1} - u^n) \end{aligned}$$

Warum zusätzlich eine Variable g eingeführt wird und wie sich daraus der proxunabhängige Schritt für q ergibt, zeigt eine Herleitung in [MSMC15b]. Dort wird aufgeführt, wie sich die folgenden Gleichungen aus einer Matrix-Vektor-Multiplikation ableiten. Da hier die Idee der Lagrange'schen Sattelpunkte zuvor in der Herleitung zu Algorithmus 5.1 erläutert wurde, wird die Herleitung über die Matrix-Vektor-Multiplikation ausgelassen. Dafür sei auf [MSMC15b] verwiesen.
Es wird in [MSMC15b] angegeben, dass Algorithmus 5.2 ohne Vereinfachungen wie folgt aussieht:

$$\begin{aligned} u^{n+1} &= (I + \tau\partial G)^{-1}(u^n - \tau K^\mathsf{T} q^n) \\ q^{n+1} &= (I + \sigma(\partial F)^{-1})(q^n + \sigma K(2u^{n+1} - u^n)) \end{aligned} \tag{5.21}$$

Annahme dabei ist, dass $\theta = 1$ entspricht. Bei genauer Betrachtung und Vergleich mit Algorithmus 5.2 fallen mindestens drei Dinge auf. Das Update für u wurde in einen eigenen Schritt ausgelagert und mit θ in der Schrittweite verallgemeinert. Weiterhin fällt in der aus der Umformung (5.18) bekannten Formulierung für q^{n+1} auf, dass dort $(\partial F)^{-1}$ auftritt. Außerdem ist in den Gleichungen (5.21) kein g aufzufinden.

Nutzt man die in [MSMC15b] angegebene Identität

$$(\partial F)^{-1} = (\sigma I - \sigma(I + \sigma^{-1}\partial F)^{-1})^{-1} - \sigma^{-1} I \tag{5.22}$$

aus und setzt diese in die Gleichung für q^{n+1} ein, erhält man nach einigen Umformungen [MSMC15b]

$$\begin{aligned} q^{n+1} &= (I + \sigma(\partial F)^{-1})(q^n + \sigma K(2u^{n+1} - u^n)) \\ &\overset{(5.22)}{=} (I + (\sigma I - \sigma(I + \sigma^{-1}\partial F)^{-1})^{-1} - \sigma^{-1} I)^{-1}(q^n + \sigma K(2u^{n+1} - u^n)) \\ &= \dots \\ &= q^n - q^n(I + \sigma^{-1}\partial F)^{-1} + \sigma K(2u^{n+1} - u^n) - \sigma(I + \sigma^{-1}\partial F)^{-1} K(2u^{n+1} - u^n) \\ &= q^n + \sigma(K \underbrace{(2u^{n+1} - u^n)}_{\overset{\theta=1}{=} \bar{u}^{n+1}} - \underbrace{(I + \sigma^{-1}\partial F)^{-1}(K(2u^{n+1} - u^n) + \sigma^{-1} q^n)}_{:= g^{n+1}}). \end{aligned} \tag{5.23}$$

Man erhält hieraus die Gleichungen [MSMC15b]

$$\begin{aligned} g^{n+1} &= (I + \sigma^{-1}\partial F)^{-1}(K(u^{n+1} + \theta(u^{n+1} - u^n)) + \sigma^{-1} q^n) \\ &= \operatorname{prox}_{\sigma^{-1}, F}(K\bar{u}^{n+1} + \sigma^{-1} q^n) \end{aligned}$$

und

$$\bar{u}^{n+1} = u^{n+1} + \theta(u^{n+1} - u^n).$$

Mithilfe der Umformungen in (5.23) ist nun ersichtlich, wie aus den relativ länglichen beiden Termen in (5.21) vier kürzere Terme abzuleiten sind und damit der Algorithmus 5.2 entsteht.
Für konvexes F erhält man die Identität $(\partial F)^{-1} = \partial F^*$ aus Satz 5.2. Damit vereinfacht sich der Algorithmus zum sogenannten *Primal Dualen Hybrid Gradienten* [CP11].
Ohne Kenntnis der Umformungen (5.23) ist nicht klar zu erkennen, inwiefern die Minimierung mithilfe der Lagrange'schen Sattelpunktformulierung stattfindet. Mit Algorithmus 5.2 jedoch wird klar, dass auch der soeben vorgestellte Ansatz dem für Algorithmus 5.1 erklärten Schema folgt. Man setzt die primalen Variablen fest und optimiert in Richtung der dualen Variablen und danach andersherum.

Die zugehörige Sattelpunktformulierung geben die Autoren in [MSMC15a] an als

$$\max_{q} \min_{u,g} G(u) + F(g) + \langle q, Ku - g \rangle. \tag{5.24}$$

Konvergenzbeweise und weitere Informationen zu den Einflüssen der Schrittweiten auf die Konvergenz sind auch in [MSMC15a, MSMC15b] zu finden.

6 Ein nichtparametrischer, nichtglatter Registrierungsansatz

Dieses Kapitel bildet das Hauptaugenmerk dieser Arbeit. Mithilfe der vorausgegangenen Grundlagen und Einführungen soll in diesem Abschnitt ein Registrierungsansatz vorgestellt werden, der mittels nichtkonvexer Optimierungsmethoden nichtglatte Zielfunktionen für die Bildregistrierung verarbeiten kann. Vor allem die Herleitung des Algorithmus 5.2 aus Kapitel 5 spielt eine wichtige Rolle, da der folgende Ansatz mithilfe eines solchen Algorithmus' optimiert werden soll.

6.1 Das Registrierungsmodell

Der folgende Registrierungsansatz folgt dem der nichtparametrischen Registrierung, wie sie in Gleichung (2.1) dargestellt wurde. Die Zielfunktion soll demnach wie in (2.2) als

$$\mathcal{J}(y) = \mathcal{D}(y) + \alpha\mathcal{S}(y)$$

darstellbar sein. Der Regularisierer kann, je nach Bedarf, einer der in Abschnitt 2.4 vorgestellten Regularisierer sein.
Der neue Ansatz soll Referenz- und Templatebild lokal mithilfe der Schatten-q-Norm aus Satz 4.1 auf Ähnlichkeit untersuchen. Nicht $\mathcal{R}$ und $\mathcal{T}$ sollen als solches verglichen werden, sondern die Gradienten der Bilddaten. Schreibt man diese Gradienten $\nabla\mathcal{T}$ und $\nabla\mathcal{R}$ in ein gemeinsames Datenarray und setzt diese in die Schatten-q-Norm $\|\cdot\|_{\mathrm{S},q}$, vergleicht man punktweise den Versatz durch die Deformation y und die Ausrichtung der einzelnen Gradienten. Das Distanzmaß sieht hierfür wie folgt aus:

$$\mathcal{D}^{\mathrm{S},q}(\mathcal{R}, \mathcal{T} \circ y) \coloneqq \int_\Omega \|(\nabla\mathcal{T}(y(x))|\nabla\mathcal{R}(x))\|_{\mathrm{S},q}\,\mathrm{d}x. \tag{6.1}$$

Je nachdem, wie man q wählt, ist dieses Maß nichtkonvex. Wählt man beispielsweise $q = 0$, erhält man aus der Schatten-Norm eine Quasi-Norm, die auf den Rang der enthaltenen

Matrix abbildet. Wie auch in Abschnitt 4.2 schon beschrieben und von den Autoren in [MSMC15b] verwendet, kann man $q \in (0, 1)$ wählen, um die dort angesprochene Relaxierung zu erhalten. Diese Relaxierung sollte – in Anlehnung an die sog. *nuclear norm* – den Rang der enthaltenen Matrix approximieren. Damit soll in diesem Fall eine Ausrichtung der Kanten der verglichenen Bilder erzeugt werden, ähnlich dem NGF-Distanzmaß aus Gleichung (2.5).
Im Folgenden muss sich darum gekümmert werden, wie das Maß im Zusammenhang mit einem Regularisierer als Zielfunktion optimiert werden kann. Der nächste Abschnitt wird eine Umformulierung präsentieren, die im Kontext der Lagrange'schen Sattelpunktform das Optimierungsproblem durch verschiedene Substitutionen in die gewünschte Form bringt.

6.2 Entwurf des Optimierungsverfahrens

Das Zielfunktional $\mathcal{J}(y)$ soll sich aus dem in Gleichung (6.1) vorgestellten Distanzmaß $\mathcal{D}^{\mathrm{S},q}(\mathcal{R}, \mathcal{T} \circ y)$ und beispielhaft einem *curvature* Regularisierer $\mathcal{S}^{\mathrm{curv}}(y)$ – wie in Gleichung (2.9) zu finden – zusammensetzen. Entsprechend gilt (vereinfacht für die Deformation y und nicht $u = y - y^{\mathrm{ref}}$ (vgl. Abschnitt 2.4)):

$$\mathcal{J}(y) = \int_\Omega \|(\nabla \mathcal{T}(y(x)) | \nabla \mathcal{R}(x))\|_{\mathrm{S},q} \,\mathrm{d}x + \frac{\alpha}{2} \int_\Omega \sum_{j=1}^{d} \|\Delta y_j(x)\|_2^2 \,\mathrm{d}x. \tag{6.2}$$

Diesen Ansatz mit Algorithmus 5.2 zu optimieren ist nicht ohne Weiteres möglich. Zunächst muss das Modell insofern separiert werden, als dass die Summanden G und F definiert sind und zwar so, dass das Modell [MSMC15a]

$$\min_{u,g} G(u) + F(g) \text{ mit } g = Ku$$

aus Gleichung (5.20) aufgegriffen werden kann. Für dieses Modell liegt in F ein linearer Operator K und in G liegt u ohne weitere Operationen vor. Das Registrierungsproblem (6.2) lässt sich in eine solche Form wie folgt umsetzen:

$$\inf_y G(Ly) + F(\bar{K}(y)). \tag{6.3}$$

Für diese Umformulierung wird F zum Datenterm und G zur Regularisierung. Zu beachten ist, dass in diesem Modell ein linearer Operator in G auftritt und ein nichtlinearer

Operator in F. Der Operator L kann den Laplace-Operator der Regularisierung abbilden und der nichtlineare Operator $\bar{K}$ die Matrix der Gradienten von $\mathcal{T}$ und $\mathcal{R}$. Damit die Optimierung wie in Algorithmus 5.2 funktioniert, wird der nichtlineare Operator $\bar{K}$ mittels Taylor-Entwicklung linearisiert und im Laufe der Optimierung in jedem Schritt angepasst. Durch die Linearisierung erhält man folgende Form:

$$\begin{aligned} \bar{K}(y) &\approx \bar{K}(y^n) + \nabla\bar{K}(y^n)(y - y^n) \\ &= \underbrace{(\bar{K}(y^n) - \nabla\bar{K}(y^n)y^n)}_{=:\, b^n} + K^n y \\ &\text{mit } K^n := \nabla\bar{K}(y^n). \end{aligned} \tag{6.4}$$

Mit dieser Linearisierung ändert sich das Problem zu

$$\inf_y G(Ly) + F(b^n + K^n y). \tag{6.5}$$

Mit einer Umformulierung über die Lagrange'sche Sattelpunktformulierung kann man das Problem in die gesuchte Form bringen:

$$\begin{aligned} &\inf_y G(Ly) + F(b^n + K^n y) \\ \Leftrightarrow &\inf_{y,u} \sup_w G(u) + \langle Ly - u, w\rangle + \tilde{F}(K^n y) \\ \Leftrightarrow &\inf_{y,u} \sup_{v,w} G(u) + \underbrace{\langle Ly - u, w\rangle + \langle K^n y, v\rangle}_{\left\langle \underbrace{\begin{pmatrix} K^n & 0 \\ L & -I \end{pmatrix}}_{\boldsymbol{K}^n} \underbrace{\begin{pmatrix} y \\ u \end{pmatrix}}_{x}, \underbrace{\begin{pmatrix} v \\ w \end{pmatrix}}_{z} \right\rangle} - \tilde{F}^*(v) \\ \Leftrightarrow &\inf_x G(x) + \tilde{F}^{**}(\boldsymbol{K}^n x) \\ \Leftrightarrow &\inf_x G(x) + \underbrace{\tilde{F}((K^n, 0)x) + \delta_{\{0\}}((L, -I)x)}_{F(\boldsymbol{K}^n x)} \end{aligned} \tag{6.6}$$

Mit dieser Darstellung lässt sich das Verfahren, ähnlich wie in [MSMC15b] angegeben, durchführen. Bei der Durchführung sollte darauf geachtet werden, dass die Terme $G(x)$ und $\tilde{F}(K^n y)$ korrekt ausgewertet werden. Aufgrund der Rechenregeln des Rückwärtsschrittoperators prox vereinfacht sich die Berechnung auf die oben dargestellte. Da für den Rückwärtsschritt auf G nur $G(u)$ ausgewertet werden muss, ist das explizite Lösen des linearen Gleichungssystems (LGS) für Ly nicht notwendig und fällt für die Ausführung des Algorithmus' nicht ins Gewicht. Das LGS wird implizit durch die Iterationen gelöst.

Für eine bessere Konvergenz kann man, wie in Algorithmus 2.1 aus [Val14] gezeigt, den Rückwärtsschritt auf F statt mit $K^n = \nabla\bar{K}(y^n)$, mit $\bar{K}(y)$ selbst ausführen. Für den Rückwärtsschritt auf G entfällt b^n, da b^n nur von den konstanten y^n abhängt, die nicht minimiert werden sollen. Konkret ist die Sattelpunktformulierung des Problems gegeben durch

$$\inf_x \sup_z G(x) + \left\langle \bar{K}(y), z \right\rangle - F^*(z)$$

und nach dem Einsetzen der Linearisierung von $\bar{K}(y)$ erhält man

$$\inf_x \sup_z G(x) + \left\langle \bar{K}(y^n) - \nabla\bar{K}(y^n)y^n + \nabla\bar{K}(y^n)y, z \right\rangle - F^*(z).$$

Da der Rückwärtsschritt auf G den Vektor x und damit auch y minimiert, ist der Teil $b^n = \bar{K}(y^n) - \nabla\bar{K}(y^n)y^n$ konstant und das Skalarprodukt kann umgeschrieben werden zu

$$\left\langle y, \left(\nabla\bar{K}(y^n)\right)^\mathsf{T} z^n \right\rangle = \left\langle y, (K^n)^\mathsf{T} z^n \right\rangle.$$

Letztlich führt dieses Modell auf folgenden Algorithmus:

Algorithmus 6.1 :

$$\begin{aligned}
g^{n+1} &\in \operatorname{prox}_{\sigma^{-1},F}(\bar{K}(\bar{x}^n) + \sigma^{-1}z^n) \\
z^{n+1} &= z^n + \sigma(\bar{K}(\bar{x}^n) - g^{n+1}) \\
x^{n+1} &= \operatorname{prox}_{\tau,G}(x^n - \tau\nabla\bar{K}(\bar{x}^n)^\mathsf{T} z^{n+1}) \\
\bar{x}^{n+1} &= x^{n+1} + \theta(x^{n+1} - x^n)
\end{aligned}$$

Dieser Algorithmus nutzt die Konvergenzeingenschaften von Algorithmus 5.1 aus und kann eins zu eins in MATLAB umgesetzt werden. Wie die Operatoren $\bar{K}$ und $\nabla\bar{K}$ implementiert werden können, wird in der numerischen Umsetzung im nächsten Kapitel gezeigt. Der Algorithmus soll die Bedingungen $\sigma\tau < \frac{1}{\|\nabla\bar{K}\|^2}$ und $\theta = 1$ erfüllen. Im nächsten Kapitel wird außerdem an einem praktischen Beispiel gezeigt, wie die Schrittweite θ etwas variabler gehalten werden kann, um die Konvergenz zu verbessern. Auf konkrete Konvergenzuntersuchungen und -beweise wird verzichtet.

Teil III

Ergebnisse

7 Experimente

Dieser Abschnitt behandelt verschiedene Experimente, die zeigen sollen, unter welchen Bedingungen der in Abschnitt 6 gezeigte Ansatz funktioniert. Zuerst soll ein skalares Problem zeigen, wie gut sich der Algorithmus globalen Minima auf verschieden schwierigen, nichtglatten Funktionen annähert. Mit Kenntnis dieser Informationen soll das Problem auf zweidimensionale Daten erweitert werden, wie es analog auch für höherdimensionale Daten passieren kann. Abschließend erfolgt eine Betrachtung über die Anpassung des Ansatzes zur Verwendung in der Bildsegmentierung. Alle im Folgenden gezeigten Ergebnisse sind in MATLAB erzeugt und können auch in verschiedenen anderen Programmiersprachen erzeugt werden.

7.1 Evaluation mithilfe eines akademischen Problems

Dieser Unterabschnitt behandelt ein skalares Problem, das ähnliche Eigenschaften zu dem in Abschnitt 6 vorgestellten Ansatz aufweist. Damit soll untersucht werden, inwieweit der Algorithmus in lokalen Minima abbricht.
Für dieses Vorhaben soll im Teil F des Modells [MSMC15a]

$$\min_{u,g} G(u) + F(g) \text{ mit } Ku = g \tag{5.20}$$

bzw.

$$\inf_{y} G(Ly) + F(\bar{K}(y)) \tag{6.3}$$

die Eigenschaft der doppelten Nichtglattheit gegeben sein. Der regularisierende Teil G hingegen kann beliebig glatt sein.
Eine Funktion in diesem Sinne, die möglichst viele Stellen aufweist, an denen sie nicht konvex ist, kann mithilfe einer trigonometrischen Funktion, wie z.B. der Sinus-Funktion, modelliert werden. Die Nichtglattheit wird durch den Absolutbetrag und die Quadrat-

wurzel dargestellt. Letztendlich erhält man

$$F(\bar{K}(x)) = \| \sin(x) \|_{\mathrm{S},q}. \tag{7.1}$$

In Abbildung 7.1 ist die Periodizität durch den trigonometrischen Anteil sowie die nichtglatten Abschnitte der resultierenden Funktion zu sehen. Die Minimalstellen sind nichtglatt und treten zahlreich auf. Mit herkömmlichen Verfahren wie z.B. einem Newtonverfahren wird man auf dieser Funktion nicht bzw. nur Abschnittsweise optimieren können.

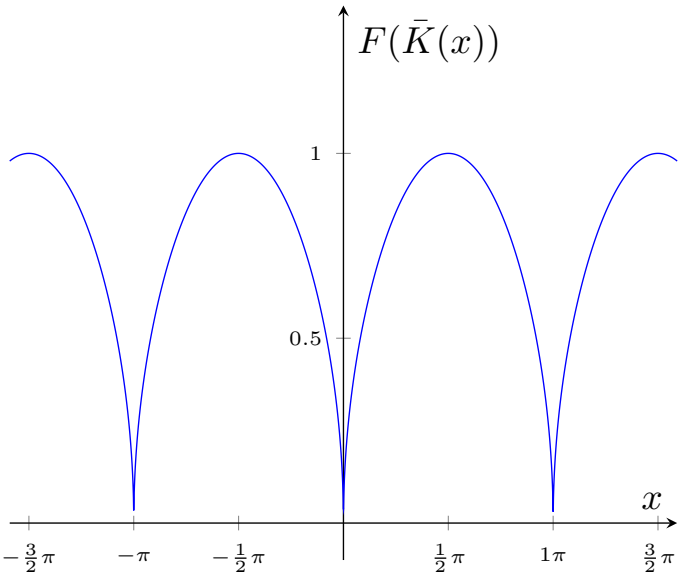


Abbildung 7.1: Die resultierende Funktion ist periodisch und für x an Vielfachen von $x = \pi$ nichtglatt. Im Plot dargestellt ist der Term $\bar{K}(x) = \sqrt{|\sin(x)|}$.

Betrachtet man die aus F resultierende Funktion als solches, liegt erschwerend kein eindeutiges globales Minimum vor. Dies kann der regularisierende Teil G als streng konvexe Funktion regeln, wie in Abbildung 7.2 zu sehen. Dazu sei G für den skalaren Test gewählt als

$$G(x) = \frac{\lambda}{2}\|x - c\|_2^2, \quad \lambda, c \in \mathbb{R}. \tag{7.2}$$

Die Summe der Funktionen (7.1) und (7.2) ergibt das gesuchte skalare Modell und lässt ausreichende Freiheiten zu Testzwecken durch die Variationsmöglichkeiten in den Parametern λ und c.
Der Ansatz lautet:

$$\min_x \frac{\lambda}{2}(x - c)^2 + \sqrt{|\sin(x)|}. \tag{7.3}$$

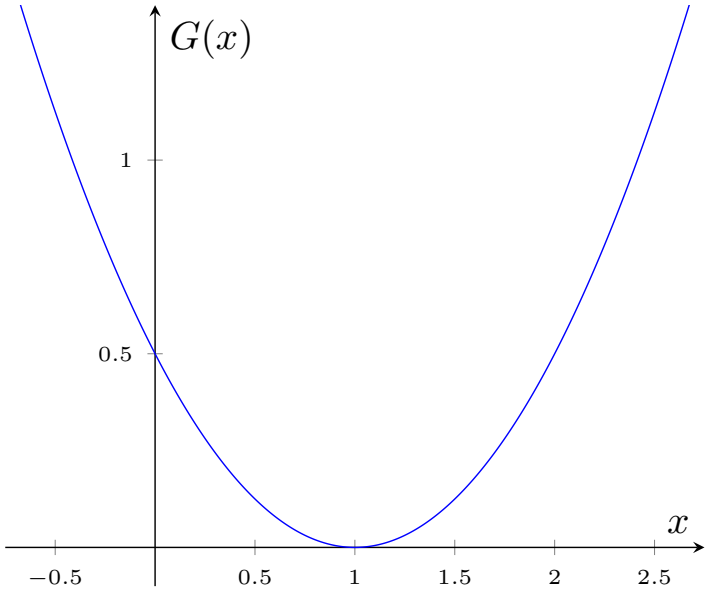


Abbildung 7.2: Die hier dargestellte Funktion ist ein Spezialfall für $\lambda, c = 1$ und ist eine verschobene Normalparabel. Dargestellt ist dieser Plot durch $G(x) = \frac{1}{2}(x-1)^2$.

In Abbildung 7.3 ist dieser Ansatz als Summe der in Abbildungen 7.1 und 7.2 gezeigten Funktionen dargestellt, wobei $\lambda = 0.1$ gesetzt ist, um die lokalen Minima zuzuspitzen. In Abbildung 7.4 sind zwei Variationen der Parameter zu sehen, um zu zeigen, wie diese auf die Gesamtfunktion einwirken. Die lokalen Minima in Abbildung 7.4b, in denen ein Algorithmus durchaus abbrechen könnte, sind zahlreich vertreten. Durch die Wahl des c ist der quadratische Anteil aus G sehr weit geöffnet, wodurch die Suche des globalen Minimums deutlich erschwert wird, da der globale negative Anstieg deutlich geringer ausfällt als in den anderen illustrierten Beispielen.

7.1.1 Numerische Umsetzung

Die Umsetzung des skalaren Problems in MATLAB basiert vor allem auf den in [MSMC15a] gezeigten Ansätzen zur Lösung des dort verwendeten skalaren Ansatzes. Der Ansatz, wie er in Algorithmus 5.2 vorgestellt wurde, kann im Prinzip direkt umgesetzt werden. Acht zu geben ist dabei auf die Auswertung der Rückwärtsschritte prox. Zum Einsatz kommt außerdem das in [SC14, MSMC15a] vorgestellete Schrittweiten-Kriterium.

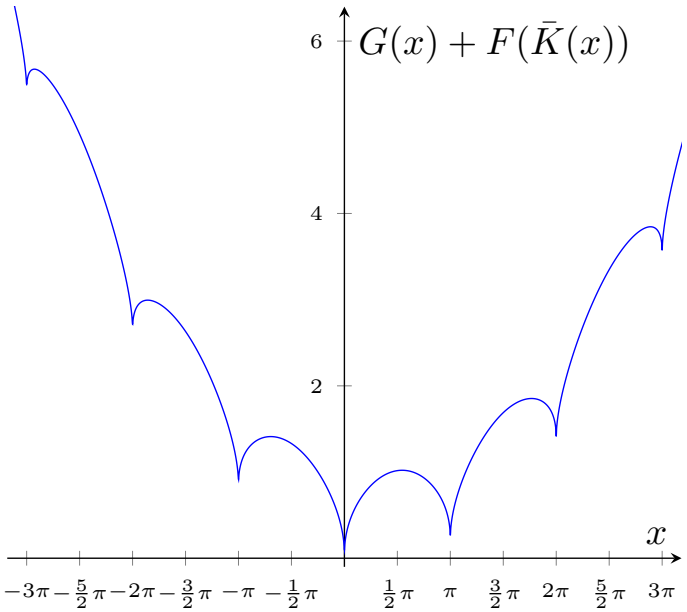
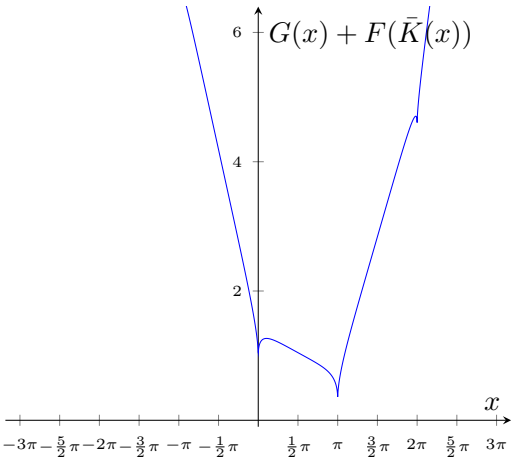
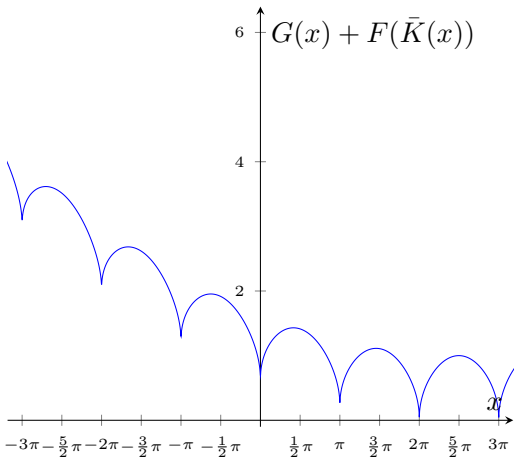


Abbildung 7.3: Durch die Wahl von $\lambda = 0.1$ sieht man eine deutliche Zuspitzung der lokalen Minima. Je größer λ gewählt wird, desto weicher sind die Minima. Zu erkennen ist, dass die Summe durch die Konvexität aus G global einen konvexen Eindruck macht, durch die Nichtglattheiten in F allerdings viele lokale Minima aufweist.



(a) Die Zielfunktion mit den Parametern $\lambda = 0.5$ und $c = 2$ weist deutlich kleinere lokale Minima auf und wirkt insgesamt deutlich konvexer auf die Zielfunktionen in den Abbildungen 7.4b und 7.3.



(b) Die Zielfunktion mit den Parametern $\lambda = 0.02$ und $c = 8$ ist deutlich nichtkonvex und weist starke, lokale Minima auf. Außerdem ist die Öffnung des quadratischen Anteils sehr groß.

Abbildung 7.4: Vergleich der Wirkung der Wahl der freien Parameter λ und c auf die Zielfunktion des aufgezeigten skalaren Testproblems.

7.1.2 Adaptive Schrittweite

Die in [SC14, MSMC15a, Val14] vorgestellte, adaptive Schrittweite zielt darauf ab, die Schrittweite in der dualen Variable unendlich zu vergrößern, für die primale Variable allerdings nicht über 1 hinaus zu wachsen.

$$\begin{aligned} \theta_n &= (1 + 2\gamma\tau_n)^{-\frac{1}{2}} \\ \sigma_{n+1} &= \frac{\sigma_n}{\theta_n} \\ \tau_{n+1} &= \tau_n\theta_n \\ &\text{mit } \tau_0\sigma_0\|K\|^2 < 1 \end{aligned} \tag{7.4}$$

Durch die Gleichungen (7.4) erhält man direkt ein Updateschema, welches man ohne weiteres Zutun umsetzen kann. Durch die Bedingung

$$\sigma_n \to +\infty$$

kann man zeigen, dass für streng konvexes G und differenzierbares F in Algorithmus 5.2 die duale Variable q eliminiert werden kann. Durch den Grenzwert in der dualen Schrittweite σ_n führt das Verfahren in diesem Fall auf das sogenannte *forward-backward splitting* [MSMC15a]. Mehr zum forward-backward splitting sei in geeigneter Literatur nachgeschlagen. Das auftretende γ ist eine heuristische Größe, welche bereits in [MSMC15a] als $\gamma = \lambda$ gewählt wurde. Dort hatte sich diese Wahl als praktikabel herausgestellt, so wie es auch für das vorliegende Problem der Fall ist. Normalerweise wird dieser Parameter aufgrund einer Konvexitätskonstante für konvexe G gewählt [MSMC15a].

7.1.3 Auswertung der Rückwärtsschritte

Die Auswertung der sogenannten prox-Schritte muss für den vorliegenden Fall für die Schatten-q-Norm betrachtet werden, für den Fall, dass $0 < q < 1$ gesetzt ist. Ein Ansatz dazu ist in [MSMC15a] niedergeschrieben.
Aufgrund der Separabilität der prox-Operatoren kann die Auswertung punktweise durchgeführt werden [MSMC15a]. Punktweise bedeutet punktweise für die Schatten-q-Norm $\|\cdot\|_{\mathrm{S},q}^q$ in

$$\mathrm{prox}_{\tau,\|\cdot\|_{\mathrm{S},q}^q}(g_0) = \arg\min_g \|g\|_{\mathrm{S},q}^q + \frac{1}{2\tau}\|g - g_0\|_2^2. \tag{7.5}$$

In [MSMC15a] wird zur Zerlegung eine Singulärwertzerlegung des Arguments g_0 durch-

geführt und in den Operator eingesetzt. Aufgrund der unitären Invarianz der Schatten- wie auch der euklidischen Norm, kann argumentiert werden, dass sich das Problem

$$\arg\min_g \|g\|_{\mathrm{S},q}^q + \frac{1}{2\tau}\|g - U\Sigma_{g_0}V\|_2^2 \tag{7.6}$$

reduziert zu

$$\arg\min_g \|\Sigma\|_{\mathrm{S},q}^q + \frac{1}{2\tau}\|\Sigma - \Sigma_{g_0}\|_2^2. \tag{7.7}$$

Die unitäre Invarianz kann man mithilfe des Satzes 4.1 und dem zugehörigen Beweis nachvollziehen. Ist (7.7) gelöst, kann die Lösung durch

$$\hat{g} = U\hat{\Sigma}V^{\mathsf{T}} \tag{7.8}$$

rekonstruiert werden [MSMC15a].
Die eigentliche Schwierigkeit der Auswertung dieses Operators ist das Minimierungsproblem (7.7). Die Autoren von [MSMC15a] haben zur Lösung des prox-Operators für die euklidische Norm ein explizites Vorgehen angegeben. Dieses kann man durch die gegebene Separabilität auch auf (7.7) anwenden. Dieses Vorgehen basiert auf der Lösung eines skalaren Problems mithilfe eines Newton-Verfahrens zur Minimierung. Man kann zeigen, dass die Lösung für den prox-Operator über der euklidischen Norm ein Vielfaches des Eingabearguments ist:

$$\tilde{g} = tg_0 \quad \text{für } t \in \mathbb{R}_+. \tag{7.9}$$

Der Beweis dazu ist im Anhang von [BTP13] zu finden.
Da Σ in (7.7) eine Diagonalmatrix ist und eine punktweise Auswertung stattfindet, reduziert sich das Problem auf eine skalare Auswertung. Es spielt deshalb keine Rolle, ob die Schatten-q- oder die euklidische Norm verwendet wird. Dies liegt an der Tatsache, dass die Singulärwertzerlegung eines Skalars eben diesen Skalar im Absolutbetrag liefert. In [MSMC15a] wurde die euklidische Norm zur Auswertung verwendet, um das skalare Problem zu lösen, da dies in jenem Fall die gleiche Lösung liefert.
Man kann

$$\arg\min_g \|g\|_2^q + \frac{1}{2\tau}\|g - \Sigma_{g_0}\|_2^2$$

vereinfachen zu

$$\arg\min_{t>0} \underbrace{\alpha t^q + \frac{1}{2}(t-1)^2}_{=:\, f(t)}, \tag{7.10}$$

indem man $g = tg_0$ einsetzt. Durch Äquivalenzumformungen und Einführung von $\alpha = \tau \|g_0\|_2^{q-2} \geq 0$ erhält man die vorgestellte Form [MSMC15a].
Für die Minimierung von $f(t)$ lassen sich für spezielle q geschlossene Lösungsterme angeben [MSMC15a]. Für den allgemeinen Fall allerdings ist ein Newton-Verfahren notwendig, welches jedoch in wenigen Schritten konvergiert, da die Ableitung f' auf $[\hat{t}, 1]$ konvex und monoton steigend ist [MSMC15a]. Dabei ist $\hat{t}$ der Minimierer von $f(t)$.
In [MSMC15a] wird der Newton-Schritt mithilfe der ersten und zweiten Ableitungen f' bzw. f'' durchgeführt und erhält so folgende Gestalt:

$$t_{k+1} = t_k - \frac{f'(t_k)}{f''(t_k)}. \tag{7.11}$$

Beachtet werden muss, dass α einen bestimmten Wert nicht überschreiten darf, da sonst nicht der gesuchte Minimierer, sondern die 0 als Lösung für t gefunden wird. Ist α zu groß, ist der Randwert in $f(t)$ für $t = 0$ kleiner als das gesuchte Minimum. Die Autoren von [MSMC15a] haben dies in der Bedingung

$$\alpha > \frac{1}{2-q} \left(2 \frac{1-q}{2-q} \right)^{1-q} \tag{7.12}$$

aufgeschrieben.
Mit diesem Wissen ist es möglich einen geschlossenen Algorithmus anzugeben, der dieses skalare Problem lösen kann. Siehe dazu in [MSMC15a] bzw. Algorithmus 7.1.

7.1.4 Ergebnisvergleich

Im Folgenden sollen die in (7.3) verwendeten Parameter λ und c sowie der Startwert für x variiert werden, um verschiedene Ergebnisse zu erzeugen und das Verhalten des skalaren Algorithmus genauer untersuchen zu können.
Zur Erzeugung der Ergebnisse ist die Anzahl an Iterationen festgesetzt auf 5000. Die Newton-Schritte werden wie in Algorithmus 7.1 [MSMC15a] ausgeführt. Die dort angegebenen Startwerte entsprechen den tatsächlich verwendeten. Die Konvergenzplots zeigen nur jede 100. Iteration, um eine Tendenz des Konvergenzverhaltens aufzuzeigen. Ein genaueres Verhaltensmuster ist in Abbildung 7.5 dargestellt.
Durchgeführt wurden sechs Variationen für c sowie zwölf Variationen für λ mit jeweils zwei verschiedenen Startwerten x^{start}. Einzusehen sind die Ergebnisse in Tabelle 7.1. Diese Ergebnisse zeigen, dass für $\lambda = 0.1$ fast nie das globale Minimum erreicht wird,

Algorithmus 7.1 : Newton-Algorithmus zur Lösung des Rückwärtsschritts (7.5)

Eingabe : g_0, $\tau > 0$, $q \in (0, 1)$, Maschinengenauigkeit ϵ
Ergebnis : Minimierer $\hat{g}$ von (7.5)
wenn $\|g_0\|_2 \leq 0$ **dann**
  $\hat{g} = 0$;
sonst
  $\alpha = \tau \|g_0\|_2^{q-2}$;
  wenn α (7.12) *erfüllt* **dann**
    $\hat{t} = 0$;
  sonst
    $t_0 = 0$;
    $k = 1$;
    solange $f'(t_k)/f''(t_k) \geq \epsilon$ **tue**
      $t_{k+1} = t_k - f'(t_k)/f''(t_k)$;
      $k = k + 1$;
    Ende
    $\hat{t} = t_k$;
  Ende
  $\hat{g} = \hat{t} g_0$;
Ende

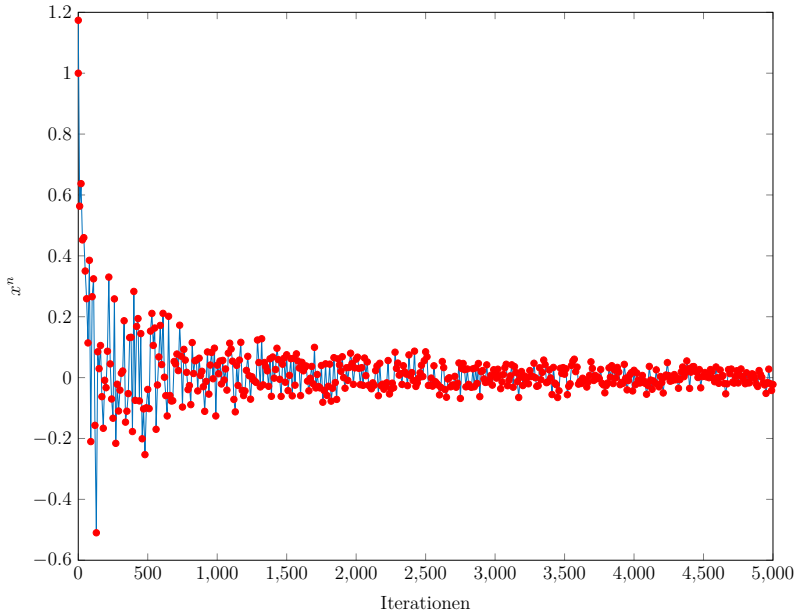


Abbildung 7.5: Dargestellt ist das Konvergenzverhalten für jede zehnte – von 5000 – Iteration mit $\lambda = 0.1$ und $c = 2$. Der tatsächliche Minimierer liegt bei $x = 0$.

es sei denn, x^{start} liegt hinreichend nah am globalen Minimierer, sodass kein lokales Minimum dazwischen liegt. Man sieht, dass der Algorithmus ansonsten jedes Mal in einem lokalen Minimum vorher abbricht. Ist $x^{\text{start}} = -15$ gesetzt, werden tatsächlich viele lokale Minima übergangen, doch das letzte lokale Minimum vor dem globalen Minimum scheint „zu tief“ zu sein. Auch aus der anderen Richtung von $x^{\text{start}} = 5$ bleibt der Algorithmus in dem letzten lokalen Minimum stehen. Wählt man λ jedoch etwas größer, wie z.B. mit $\lambda = 0.5$ – wie auch in Tabelle 7.1 zu sehen – ist die Zielfunktion deutlich glatter und der Algorithmus erreicht jedes Mal das globale Minimum. Abzulesen ist aus Tabelle 7.1 außerdem, dass relativ kleine Fehler des gefundenen Minimierers eine große Wirkung auf den Wert des zugehörigen Minimums haben. Dies liegt an den Exponenten der Zielfunktion. Dazu zählen sowohl der gebrochen rationale Exponent durch die Wurzel sowie auch das Quadrat der euklidischen Norm. Einige Ergebnisse sind in Abbildung 7.6 und ein Vergleich für verschiedene λ in Abbildung 7.7 dargestellt.
Die Ergebnisse hinterlassen insgesamt den Eindruck, dass der Algorithmus relativ robust gegenüber lokalen Minima ist. Diese Relativierung bezieht sich vor allem darauf, dass der Algorithmus für den Fall $\lambda = 0.1$ kurz vor dem globalen Minimum abgebrochen hat. Die gefundenen lokalen Minima weichen nicht allzu stark von dem tatsächlichen globalen Minimum ab. Wie sich dies in der Praxis auswirkt, muss getestet werden. Zu erwähnen ist, dass der in Tabelle 7.1 angegebene tatsächliche Minimierer $\tilde{x}^*$ über die letzten 100 Iterationen gemittelt ist. In den Konvergenzplots kann man sehen, dass der Algorithmus nur noch leicht oszilliert. Diese Oszillationen hören auch bei deutlich größerer Iterationszahl nicht auf. Dadurch liegt die Vermutung nahe, dass der Algorithmus lediglich einem ergodischen Konvergenzverhalten folgt. Die Oszillation ist beispielsweise gut in Abbildung 7.5 zu sehen. Die Amplitude wird immer geringer, bleibt ab circa 4000 Iterationen allerdings auf einem ähnlichen Niveau. Nähere Untersuchungen zu einem solchen Konvergenzverhalten bezüglich primal-dualer Algorithmen sind z.B. in [CP15] zu finden.

c	λ	x^{start}	tatsächlicher Minimierer x^*	tatsächliches Minimum $f(x^*)$	gefundener Minimierer $\tilde{x}^*$	gefundenes Minimum $f(\tilde{x}^*)$
-1	0.1	-15	0	0.05	-6.2786	1.4611
		5			3.1384	0.9127
	0.5	-15	0	0.25	$5.5{\cdot}10^{-5}$	0.2575
		5			0.0003	0.2672
0	0.1	-15	0	0	-3.138	0.5524
		5			3.1399	0.5339
	0.5	-15	0	0	$8.3{\cdot}10^{-5}$	0.0091
		5			-0.0003	0.0181
1	0.1	-15	0	0.05	-3.1422	0.8815
		5			3.1435	0.2739
	0.5	-15	0	0.25	0.0010	0.2813
		5			0.0006	0.2748
2	0.1	-15	π	0.0652	-3.1411	1.3448
		5			3.1419	0.0838
	0.5	-15	π	0.3258	3.141	0.3504
		5			3.1415	0.3330
3	0.1	-15	π	0.0010	0.0035	0.5078
		5			3.144	0.0499
	0.5	-15	π	0.0050	3.1407	0.0356
		5			3.1421	0.0273
4	0.1	-15	π	0.0368	0.0050	0.8689
		5			6.2833	0.2709
	0.5	-15	π	0.1842	3.143	0.221
		5			3.1421	0.2067

Tabelle 7.1: Übersicht der Ergebnisse verschiedener Durchläufe des skalaren Problems (7.3).

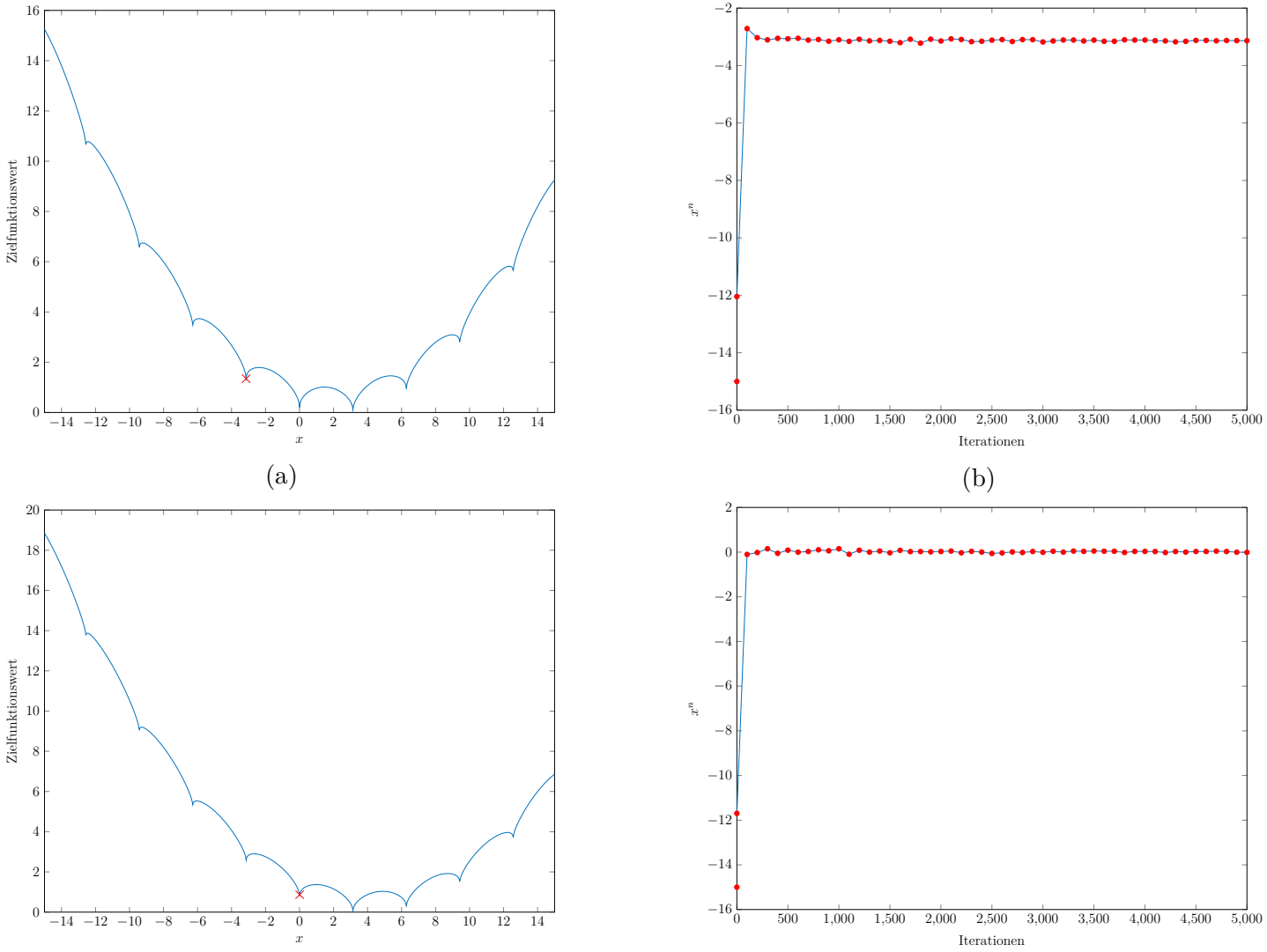


Abbildung 7.6: Das skalare Problem für $x^{\text{start}} = -15$ und $\lambda = 0.1$, oben für $c = 2$, unten für $c = 4$. Jeweils links die Zielfunktion mit gefundenem Minimierer und rechts der zugehörige Konvergenzplot.

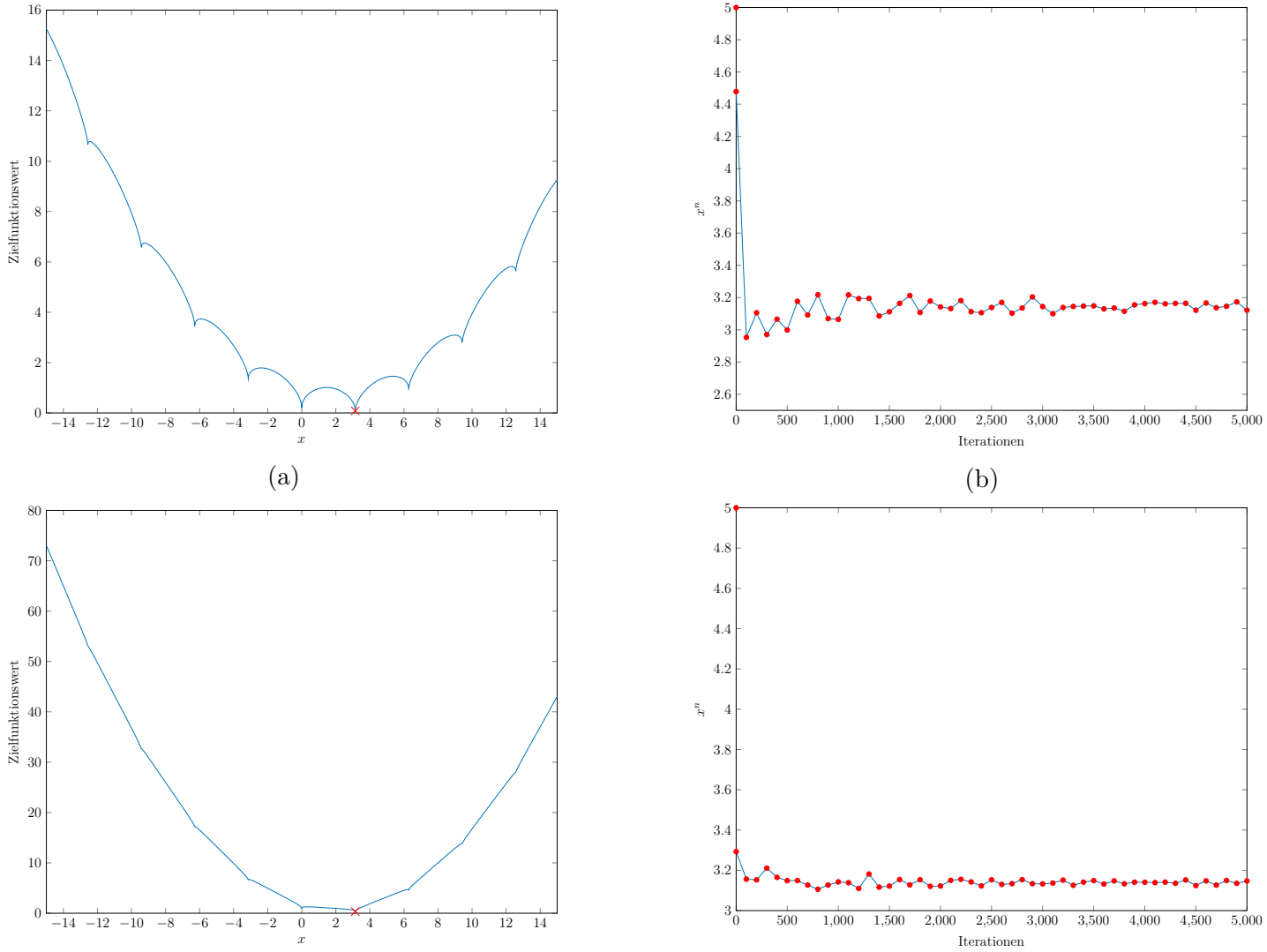


Abbildung 7.7: Das skalare Problem für $x^{\text{start}} = 5$ und $c = 2$, oben für $\lambda = 0.1$, unten für $\lambda = 0.5$. Jeweils links die Zielfunktion mit gefundenem Minimierer und rechts der zugehörige Konvergenzplot.

7.2 Registrierung zweidimensionaler Grauwertdaten

Die Durchführung des Ansatzes auf zweidimensionalen Grauwertdaten soll zeigen, ob der Vergleich der Gradienten $\nabla\mathcal{T}$ und $\nabla\mathcal{R}$ mithilfe der Schatten-q-Norm im Datenterm das gewünschte Ergebnis einer Registrierung liefert. Dazu konnte das Grundgerüst der MATLAB-Implementierung des skalaren Problems aus Kapitel 7.1 übernommen werden. Hinzu gekommen sind Funktionen zur Berechnung der benötigten Operatoren und zur Auswertung der Rückwärtsschritte sowie der Zielfunktion. Anwendung gefunden haben dazu auch Methoden aus der FAIR-Toolbox [Mod09] von Jan Modersitzki. Unter anderem wird zur Interpolation der Werte, die im Verlauf des Algorithmus nicht auf dem Gitter liegen, die Splineinterpolation aus dieser Toolbox verwendet. Das Prinzip des Codes der Interpolation entspricht genau dem in Kapitel 2.5.2 beschriebenen. Ausführlicheres zur Toolbox und der Theorie der enthaltenen Funktionen ist in [Mod09] zu finden.

7.2.1 Evaluation des Modells

Zunächst soll das Modell evaluiert werden. Dazu werden zwei Rechtecke gegeneinander verschoben. Dabei hat die Fläche des Rechtecks konstant den Intensitätswert 1 und die restliche Umgebung den Wert 0. Die Auswertung erfolgt mittels eines zellzentrierten Gitters, welches schrittweise verschoben wird. Dabei wird die Schatten-q-Norm des Operators

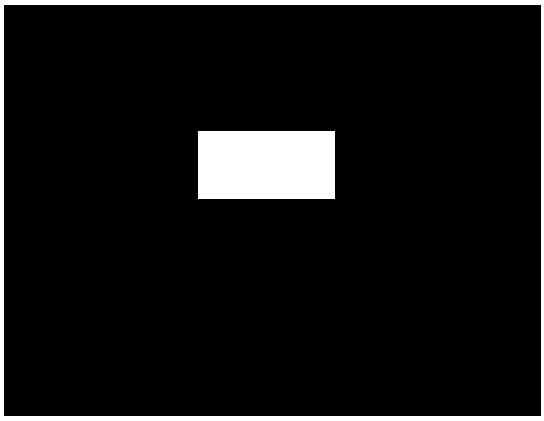
$$\bar{K} = (\nabla\mathcal{T}(y(x))|\nabla\mathcal{R}(x))$$

ausgewertet, so wie in Gleichung (6.1) bereits angegeben.
Es sollte sich ergeben, dass das Distanzmaß am geringsten ist, wenn sich die Rechtecke vollständig überdecken. Je nachdem, wie man den Exponenten der Schatten-q-Norm wählt, sollte man die Konvexität der resultierenden Funktion beeinflussen können. In Abbildung 7.8 sind einige Ergebnisse dargestellt. Man kann erkennen, dass auch auf den zweidimensionalen Daten der Parameter q die gleiche Wirkung hat wie in dem aufgezeigten akademischen Problem. Je kleiner man q wählt, desto spitzer werden die Minima der Funktion. Eine Besonderheit, die später für die Registrierung beachtet werden sollte, ist in den Darstellungen am rechten Rand der Plots zu beobachten. Von links nach rechts zeigen die Graphen die Funktionswerte der ausgewerteten Schatten-q-Norm von dem oben genannten $\bar{K}$. Vor allem in Abbildung 7.8e ist besonders gut zu erkennen, dass die Funktion zwei kleinere Spitzen in gleichem Abstand um eine deutlich tiefere aufweist. Diese stellt die hundertprozentige Überlagerung der Rechtecke dar. Die beiden kleineren

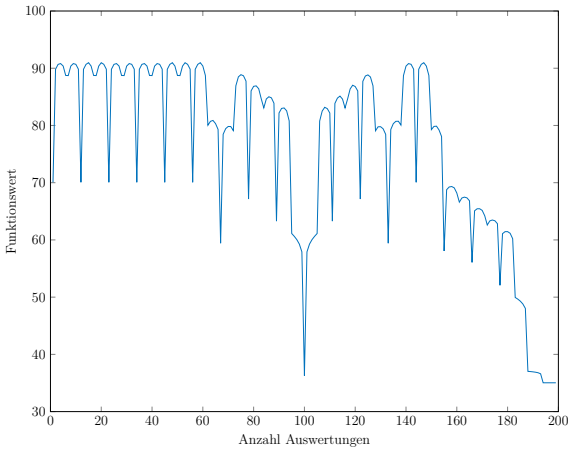
Spitzen stellen das Zusammentreffen der Kanten dar. Für die Registrierung sucht man die mittlere, tiefere der drei Spitzen. Die erwähnte Besonderheit ist das starke Abfallen der Funktionswerte am rechten Rand. Dieser Abfall ist der Austritt des bewegten Rechtecks aus dem betrachteten Bereich. Der niedrigste Wert wird erreicht, wenn das Rechteck komplett aus dem Bereich verschwunden ist. Dies liegt an der Auswertung der Norm. Ist das zweite Rechteck komplett verschwunden, reduziert sich die Auswertung letztlich auf die Norm des festen Rechtecks. Die Werte für das eine Rechteck allein sind niedriger, da die Kanten ohne Überlagerung zweier Rechtecke gleicher Intensität nur halb so hoch sind als mit einer Überlagerung.
Die Erwartung liegt aufgrund der Evaluation des Modells bei einer erfolgreichen Registrierung. Man kann in Abbildung 7.8 sehen, dass man ein deutliches Minimum für die korrekte Übereinstimmung der zu registrierenden Objekte finden kann. Solange das zu registrierende Objekt des Templatebildes im betrachteten Bereich bleibt, ist dieses Minimum sogar eindeutig. Wie schon das akademische, eindimensionale Problem gezeigt hat, ist das verwendete Optimierungsverfahren durchaus in der Lage, diesem Minimum hinreichend nahe zu kommen. Hinzu kommt, dass die in dieser Evaluation gezeigten Minima so extrem sind, dass es recht wahrscheinlich ist, diese zu finden.

7.2.2 Registrierungsversuche und numerische Umsetzung

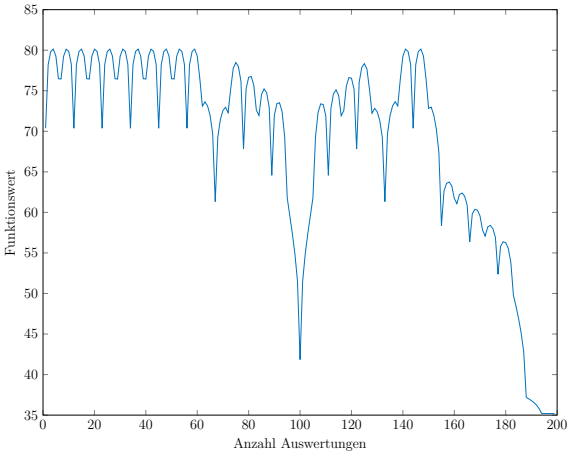
Zur numerischen Bestimmung der verwendeten Operatoren $\bar{K}(y)$ und $\nabla\bar{K}(y)$ in MATLAB war es einerseits notwendig, das Templatebild auf das Gitter anzupassen, welches sich in jeder Iteration ändert. Dazu kann eine Interpolation verwendet werden. Andererseits müssen die Gradienten der Bilder sowie der Gradient des Operators $\bar{K}(y)$ nach y bestimmt werden. Aufgrund der Änderungen in jedem Schritt muss der Operator entsprechend „nachgeführt“ werden. In die Funktion, welche den Operator berechnet, ist neben dem Gitter auch die Deformation eingegangen. Mithilfe der Deformation kann das veränderte Templatebild über eine Interpolation berechnet werden. Anschließend muss der Gradient des diskreten Templatebildes in jedem Schritt erneut bestimmt werden. Dafür kann man Differenzenmatrizen, welche finite, zentrale Differenzen ermitteln, verwenden, mit denen man durch Matrix-Vektor-Multiplikation einen diskreten Gradienten erzeugen kann. Dazu vektorisiert man das Bild, indem man die Spalten der Bilddatenmatrix untereinander setzt. So erhält man einen Spaltenvektor für das Bild und kann diesen von rechts an die Differenzenmatrix multiplizieren. Zu beachten ist dabei die Richtung der Differenzen. Es werden außerdem bestimmte Randbedingungen in die Matrizen eingefügt, die in den folgenden Gleichungen nicht dargestellt sind. In der Implementierung werden jeweils



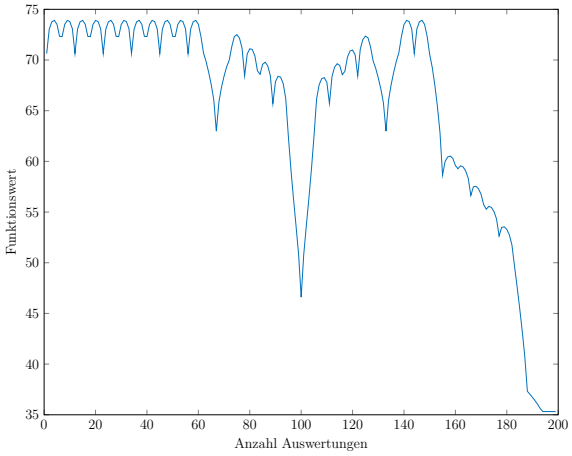
(a) Das gegen sich selbst von unten nach oben verschobene Rechteck.



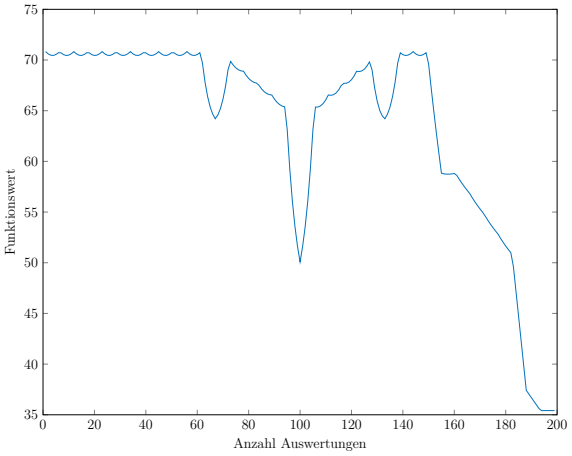
(b) Funktionswerte der Norm für $q = 0.1$.



(c) Funktionswerte der Norm für $q = 0.5$.



(d) Funktionswerte der Norm für $q = 0.8$.



(e) Funktionswerte der Norm für $q = 1$.

Abbildung 7.8: Vergleich der Entwicklung der Funktionswerte für gegeneinander verschobene Rechtecke.

die erste und letzte Zeile der Matrix auf Null gesetzt. Diese *Neumann-Randbedingungen* [Mod04] legen fest, dass die zweite Ableitung verschwindet.
Zur Erinnerung sei die Gestalt der Differenzenmatrizen aus Gleichung (2.17) nochmals angegeben [Mod09]:

$$D_j = \frac{1}{2h_j} \begin{pmatrix} -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \dots & 0 & -1 & 0 & 1 \end{pmatrix}.$$

Dafür war $j \in \{1, \dots, d\}$, wobei $d \in \mathbb{N}$ die Anzahl der Raumrichtungen beschrieb. Die Umsetzung der Berechnung des Gradienten ist durch eine Matrix-Vektor-Multiplikation möglich, indem man die diskreten Gradienten grad untereinander in eine größere Matrix schreibt und diese mit dem vektorisierten Bild multipliziert. Man erhält z.B. für das diskrete Templatebild T das Produkt

$$\operatorname{grad} T = \begin{pmatrix} \partial_1 T \\ \partial_2 T \end{pmatrix}.$$

Analog kann man den diskreten Gradienten des Referenzbildes berechnen, welche ein einziges Mal vor Ablauf der Iterationen bestimmt werden muss, da sich diese Ableitung durch den Algorithmus nicht mehr ändert.
Weiterhin muss der diskrete Laplaceoperator bestimmt werden, da dieser zur Umsetzung von Gleichung (6.6) benötigt wird. Diese Matrix lässt sich ebenfalls aus dem eindimensionalen Fall über Kroneckerprodukte aufbauen. Auch hier nochmals zur Erinnerung. Der eindimensionale Fall war gegeben durch

$$L_{1\mathrm{D}} = \begin{pmatrix} 1 & -2 & 1 \end{pmatrix}.$$

Auf zwei Dimensionen erweitert erhält man somit einen sogenannten Fünf-Punkt-Stern [Mod04]

$$L_{2\mathrm{D}} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

In die benötigte Dimension gebracht, muss auch dieser Operator nur ein einziges Mal berechnet werden. Auch hier gehen die Neumann-Randbedingungen ein.

Damit ergibt sich die Auswertung von $\bar{K}(y)$ gemäß Umformung (6.6) als Vektor

$$\bar{K} = \begin{pmatrix} \operatorname{grad} T \\ \operatorname{grad} R \\ Ly - u \end{pmatrix}. \tag{7.13}$$

Wie sich mithilfe dieser diskreten Operatoren die Berechnung von $\bar{K}(y)$ in MATLAB ausführen lässt, ist in Quellcode 7.1 illustriert. Diese Funktion arbeitet auf die beschriebene Weise und gibt $\bar{K}(y)$ als Vektor zurück. Die Eingabe m enthält die Größe des Templatebildes und *omega* die Informationen über das Bildgebiet. Die Funktion *split_x* teilt den Vektor x in y und u auf. Der Aufruf *linearInter* startet die lineare Interpolation der FAIR-Toolbox [Mod09].

```
function result = K_bar_full(x,T,dR,GRAD,L,omega,m,grid)
    [y,u] = split_x(x,m);
    T = linearInter(T,omega,grid+y);
    dT = GRAD * T(:);

    resv = [dT(:); dR(:)];
    resw = L*y - u;

    result = [resv; resw];
end
```

Quellcode 7.1: Code zur Berechnung des Operators $\bar{K}(y)$.

Für den Algorithmus wird außerdem die Ableitung $\mathrm{d}\,\bar{K}(y)$ nach y benötigt. Da nur das diskrete Templatebild T von der Deformation y abhängt, entfällt die Ableitung des diskreten Gradienten von R, wodurch sich die folgende Blockmatrix ergibt, an welche später $z = \begin{pmatrix} v \\ w \end{pmatrix}$ multipliziert werden kann:

$$\mathrm{d}\,\bar{K} = \left(\begin{array}{c|c} \partial_{1,1}T & \partial_{1,2}T \\ \hline \partial_{2,1}T & \partial_{2,2}T \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline L & -I \end{array}\right). \tag{7.14}$$

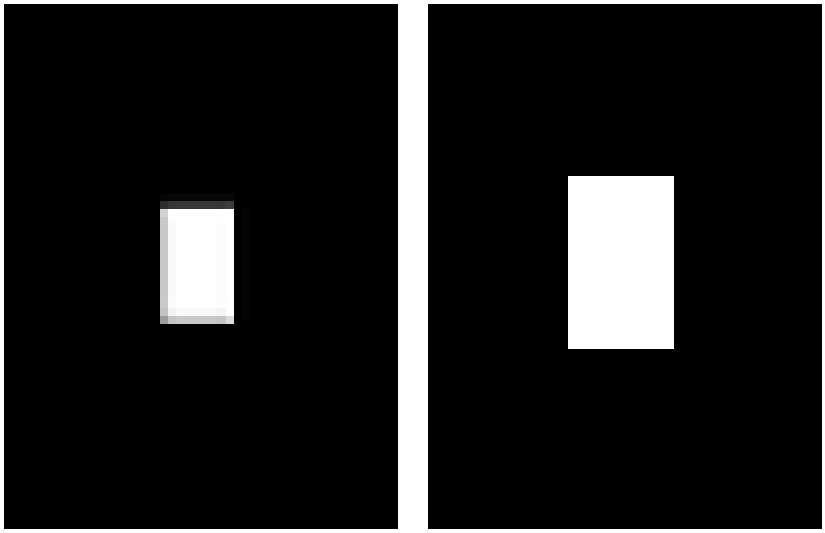
Hier sind $\partial_{i,j}T$ die diskreten, zweiten Ableitungen des diskreten Templatebildes T als Diagonalmatrizen, welche erhalten werden durch Bestimmung des Gradienten der aus der Interpolation erhaltenen Ableitung nach der Deformation. Man könnte in der Gleichung zur Verdeutlichung auch $\mathrm{d}_y\,\bar{K}$ schreiben.
Es ist zu unterscheiden, dass für den Vergleich der Gradienten von $\mathcal{T}$ und $\mathcal{R}$ lediglich die diskreten Bildgradienten benötigt werden, wohingegen die Blockmatrix aus (7.14) die Ableitung nach der Deformation enthält. Dadurch entfällt der Teil für das Referenzbild und es entstehen die Nullblöcke. Die Berechnung des Gradienten nach der Deformation erfolgt numerisch über eine bikubische Splineinterpolation. Aus der Interpolation wird der Gradient nach der Deformation entnommen, weswegen hierfür andere Symbole verwendet werden. Dieser kann anschließend mithilfe der Differenzenmatrix noch einmal abgeleitet werden. Abschließend fasst man alle Blöcke in einer Matrix zusammen. Diese Prozedur ist in Quellcode 7.2 dargestellt.

```
function result = dK_bar_full(x,T_coeff,GRAD,L,omega,m,gr)
    [y,~] = split_x(x,m);
    nPix = prod(m);
    [~,dT] = splineInter(T_coeff,omega,gr+y);
    dT1 = diag(dT(:,1:nPix));
    dT2 = diag(dT(:,nPix+1:2*nPix));

    d2T13 = GRAD * dT1;
    d2T24 = GRAD * dT2;

    K = sparse(6*nPix,4*nPix);
```

```
12      K(1:nPix,1:nPix) = sparseDiag(d2T13(1:nPix));
13      K(nPix+1:2*nPix,1:nPix) = sparseDiag(d2T13(nPix+1:2*nPix));
14      K(1:nPix,nPix+1:2*nPix) = sparseDiag(d2T24(1:nPix));
15      K(nPix+1:2*nPix,nPix+1:2*nPix) = sparseDiag(d2T24(nPix+1:2*
            nPix));
16
17      K(4*nPix+(1:2*nPix),1:2*nPix) = L;
18      K(4*nPix+(1:2*nPix),2*nPix+(1:2*nPix)) = -speye(2*nPix,2*
            nPix);
19
20      result = K;
21  end
```

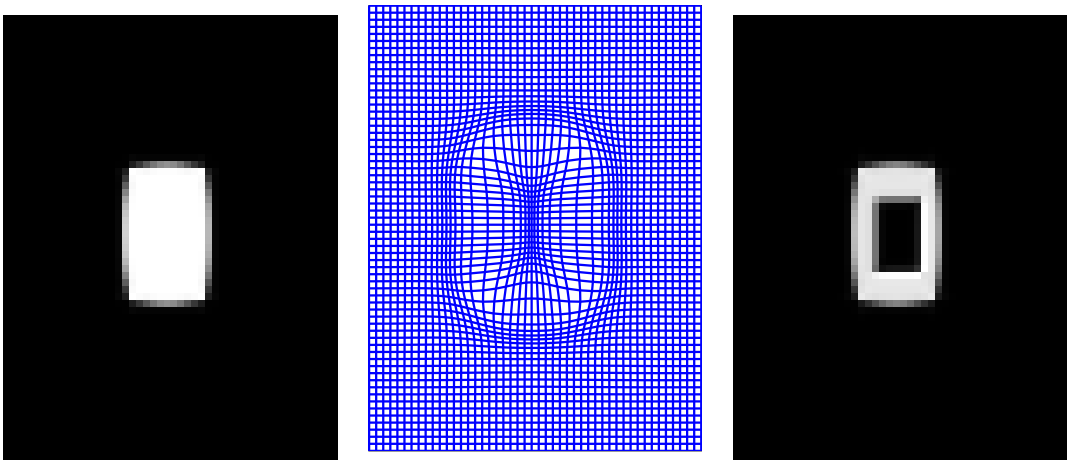
Quellcode 7.2: Code zur Berechnung des Operators $\mathrm{d}\,\bar{K}(y)$.

Mit dieser Umsetzung ist es möglich, eine Registrierung durchzuführen. Ein erstes, einfaches Beispiel ist die Registrierung eines grauwertbasierten Rechtecks mit konstantem Grauwert. Das Templatebild für diesen Versuch enthält ein etwas größeres Rechteck, als es im Referenzbild der Fall ist. Der Grauwert des Rechtecks ist derselbe. Das bedeutet, es liegt für diesen Fall eine parametrische, rigide Transformation vor. Dargestellt sind die Ergebnisse in Abbildung 7.10 und 7.11. Man kann erkennen, dass das transformierte Templatebild dem Referenzbild nicht sehr nahe kommt. Trotz über 20000 Iterationen ist die Zielfunktion nicht minimiert worden. Es liegt aufgrund eines solchen Ergebnis' die Vermutung eines systematischen Fehlers nahe.
Ein ähnliches Problem kann man in dem zweiten Versuch sehen. Es sollten zwei Röntgenbilder einer Hand registriert werden. Die Daten stammen aus der FAIR-Toolbox [Mod09]. Auch dort ist zu sehen, dass die Registrierung, trotz knapp 120000 Iterationen, nicht wie gewünscht funktioniert, obwohl die Zielfunktion offensichtlich minimiert wird. Die Ergebnisse dazu sind in Abbildung 7.12, 7.13 und 7.14 zu sehen. Wie für das Rechteck wird das Gitter an den Kanten des Objekts auseinander gezogen. Aufgrund der vorliegenden Ergebnisse werden an dieser Stelle keine weiteren Registrierungsergebnisse präsentiert. Weitere Diskussionen zum fehlerhaften Verlauf des Verfahrens folgen in Kapitel 8.



(a) Das etwas kleinere Referenzobjekt. (b) Das Templateobjekt.

Abbildung 7.9: Die Daten haben eine Auflösung von 64×48 Bildpunkten.



(a) Das transformierte Rechteck. (b) Das Gitter zeigt eindeutig die Deformation an den Kantenpixeln. (c) Das Differenzbild zeigt, dass die Deformation nicht zielführend ist.

Abbildung 7.10: Die Registrierung des Rechtecks hat mit knapp 21000 Iterationen nicht funktioniert. Worauf dies zurückzuführen ist, muss geklärt werden.

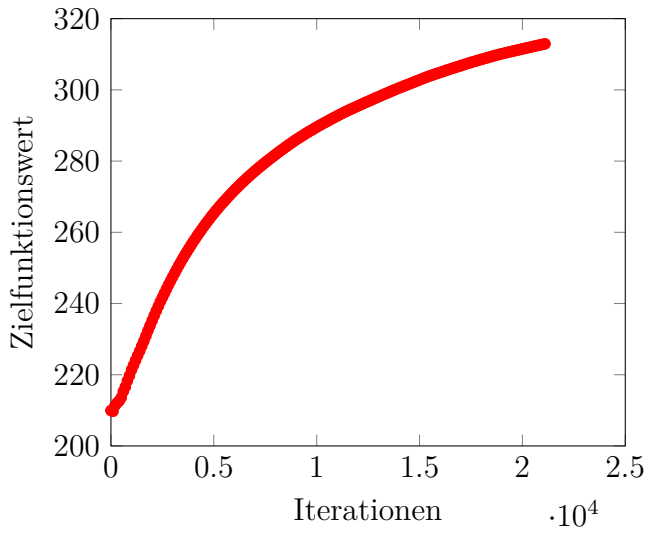
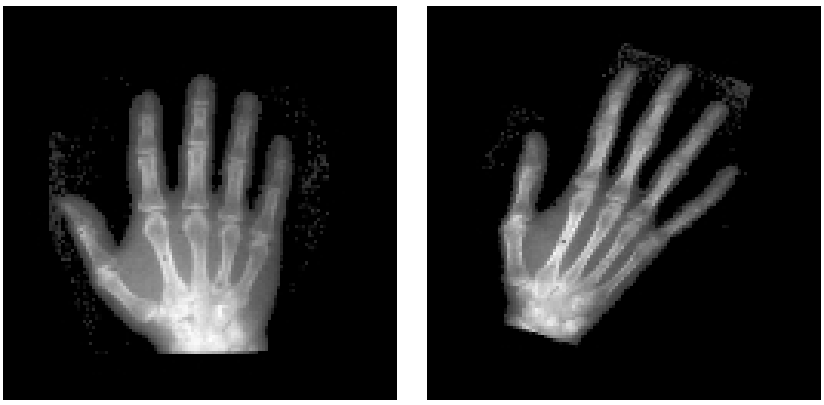


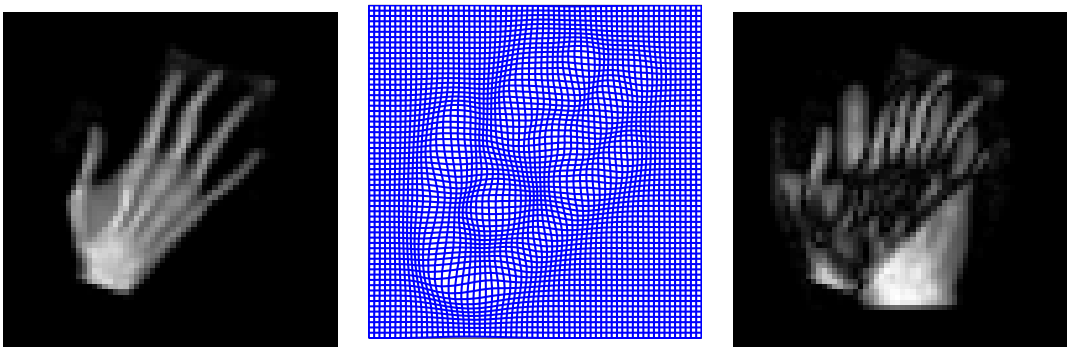
Abbildung 7.11: Die Entwicklung der Werte der Zielfunktion zeigt in diesem Fall sogar eine Maximierung der Energie und führt dadurch nicht zur gewünschten Transformation des Templatebildes.



(a) Eine Röntgenaufnahme einer Hand dorso-palmar [ZB13] als Referenzbild.

(b) Das Templatebild ist leicht verdreht und eventuell verzerrt.

Abbildung 7.12: Die Daten des zweiten Registrierungsversuchs (in Originalauflösung von 128×128 Bildpunkten) entstammen der FAIR-Toolbox [Mod09].



(a) Das deformierte Templatebild.

(b) Das zugehörige, deformierte Gitter.

(c) Das Differenzbild von Referenz- und Templatebild.

Abbildung 7.13: Zu sehen sind die Ergebnisse des Registrierungsversuchs der Röntgenaufnahme aus Abbildung 7.12. Für die Registrierung wurden die Daten auf eine Auflösung von 64×64 Bildpunkten reduziert.

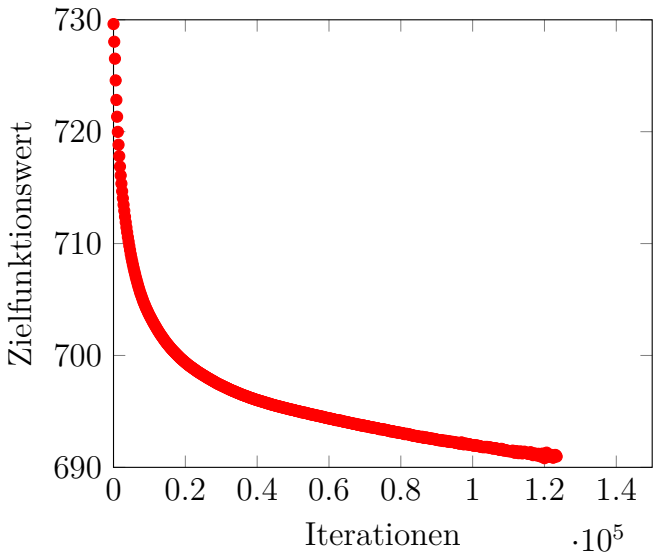


Abbildung 7.14: Die Entwicklung der Werte der Zielfunktion zeigt eine Minimierung der Energie.

8 Diskussion

In diesem Kapitel sollen die durchgeführten Versuche und die damit erzielten Ergebnisse beurteilt werden. Zum Schluss folgt ein Fazit, welche Erwartungen erfüllt worden sind und welche in Zukunft zu verbessern sein werden sowie ein Ausblick auf weitere Arbeiten zu diesem Thema.

8.1 Auswertung

8.1.1 Beurteilung des akademischen Beispiels

Die Experimente wurden mit einem akademischen, eindimensionalen Beispiel gestartet, um zu untersuchen, inwieweit der vorgestellte Optimierungsalgorithmus mit einer Funktion doppelter Nichtglattheit umgehen kann. Dazu wurde im Datenterm eine Sinus-Funktion in die Schatten-q-Norm gesetzt. Als Regularisierungsterm kam eine Verschiebung einer strikt konvexen, quadrierten Norm zum Einsatz. Die spezielle Wahl des Modells als

$$\min_x \frac{\lambda}{2}(x-c)^2 + \sqrt{|\sin(x)|}$$

bedeutete zwei Freiheitsgrade. Durch den Regularisierungsparameter λ konnte man direkten Einfluss auf die Konvexität der Zielfunktion nehmen. Je mehr Einfluss die Regularisierung auf den Zielfunktionswert hatte, desto konvexer war die Funktion schließlich. Der zweite Freiheitsgrad war die Verschiebung c. Diese hat den „Scheitelpunkt“ der Zielfunktion verschoben. Die verschiedenen Durchläufe auf der Zielfunktion haben zweierlei Urteile zugelassen. Einerseits war es möglich zu beurteilen inwieweit der Optimierungsalgorithmus lokale Minima berücksichtigt und andererseits konnte man feststellen, wie sehr das Verfahren von Oszillationen behaftet ist. Diese Oszillationen wurden in hinreichend vielen Grafiken zum Konvergenzverhalten dargestellt. Durch eine vorliegende Konvergenz trotz der Oszillationen kam die Vermutung auf, dass eine Art ergodisches Konvergenzverhalten vorliegt. Diese Vermutung kann in dieser Arbeit nicht weiter verifiziert werden, kann aber durchaus Thema weiterer Arbeiten werden. Das Verfahren hat weiterhin

gezeigt, dass es stark von der Wahl der Parameter abhängt, wie sehr lokale Minima berücksichtigt werden. Je stärker die Regularisierung gewählt wurde, desto einfacher konnten die globalen Minima in den verschiedenen Experimenten gefunden werden. Dabei hat der Startwert von x kaum eine Rolle gespielt. Selbst wenn der Minimierer eine Distanz von über 20 zum Startwert hatte, konnte dieser bei hinreichend starker Regularisierung gefunden werden. Das legt die Vermutung nahe, dass das Verfahren durchaus für ein Registrierungsvorhaben geeignet ist.

8.1.2 Beurteilung der Registrierung zweidimensionaler Grauwertdaten

Nach Abschluss der Untersuchungen in einer Dimension wurden weitere Untersuchungen in zwei Dimensionen vorgenommen. Hierzu sollte vorab das Registrierungsmodell des Gradientenvergleichs in der Schatten-q-Norm evaluiert werden. Damit sollte sichergestellt werden, dass das zu minimierende Energiefunktional möglichst ein globales Minimum für die hundertprozentige Überlagerung der registrierten Bilder aufweist. Dazu wurde ein Bild mit einem weißen Rechteck erzeugt und gegen sich selbst verschoben. Für jeden Verschiebungsschritt wurde die Energie des Modells bestimmt. Dabei stellte sich heraus, das ein starkes Minimum für die Überlagerung vorliegt. Dieses war in allen durchgeführten Experimenten deutlich von anderen lokalen Minima unterscheidbar. Es stellte sich durch die Experimente allerdings ein Problem heraus, welches durchaus eine Rolle für die Registrierung spielen kann. Sobald das Templatebild aus dem betrachteten Bereich Ω „herausgeschoben“ wurde, verringerte sich die Energie des Zielfunktionals deutlich stärker als für die gewünschte Überlagerung. Dies kann dazu führen, dass Objekte, die nah am Rand des betrachteten Bereiches liegen, nicht zur Überlagerung neigen, sondern im Laufe des Verfahrens aus dem Bereich geschoben werden.
Ob diese Vermutung stimmt, wurde in den weiteren Experimenten untersucht, indem ein einfaches Beispiel eines Rechtecks versucht wurde zu registrieren. Dabei war das Rechteck des Templatebildes etwas größer als das des Referenzbildes. Weiterhin wurde ein relativ großer Abstand zum Rand gewählt und die Rechtecke übereinander gelegt, um gewährleisten zu können, dass diese nicht zum Rand gezogen werden. Dies sollte ein erster Versuch sein, ob die Registrierung mithilfe des verwendeten Verfahrens ein solches, rigides Problem lösen kann. Dieser Versuch hat gezeigt, dass tatsächlich nur an den Kanten registriert wird. Die Kantenpixel wurden so weit in der Intensität verringert, bis diese verschwanden. Dies ist auf dem Gitter als Auseinanderziehen an den Kantenpunkten zu

beobachten gewesen. Je kleiner die Auflösung des Bildes gewählt wurde, desto näher kam man der gewünschten Registrierung. Dies lag an der Vergrößerung der Pixel. Dadurch hatte die Kante einen größeren Anteil am Objekt selbst. Damit war zwar eine Skalierung relativ gut möglich, Translationen oder Rotationen aber waren dennoch mit keinem hinreichend guten Ergebnis möglich. In diesen Fällen waren zum Schluss noch immer unveränderte Pixel der vollen Ausgangsintensität im Differenzbild zu beobachten, welches möglichst geringe Intensitätswerte aufweisen sollte, wenn die Registrierung erfolgreich gewesen wäre. Diesem Verhalten konnte man durch eine Verstärkung der Regularisierung entgegenwirken, was allerdings auch bedeutete, dass die Anzahl an benötigten Iterationen, um einen ähnlichen Fortschritt zu erzielen, deutlich vergrößert wurde. Außerdem führte dies meist dazu, dass die Energie der Zielfunktion über mehrere tausend Iterationen hinweg vergrößert statt verkleinert wurde. Aufgrund der Vergrößerung konnte der Algorithmus nur ohne Abbruchkriterien laufen. Mit typischen Abbruchkriterien wie denen von Gill, Murray und Wright [GMW81] wäre der Algorithmus jedes Mal nach wenigen Iterationen abgebrochen.

Generell stellte man während der Experimente fest, dass das Verfahren sehr viele Iterationen benötigt, um zu konvergieren. Allein das akademische, eindimensionale Problem benötigte meist zwischen 500 und 5000 Iterationen, um ein hinreichend akzeptables Ergebnis zu liefern. Die Zahl an Iterationen für das zweidimensionale Problem hatte sich entsprechend drastisch erhöht. Selbst für die Registrierung der Rechtecke mit einer relativ niedrigen Auflösung von 64×48 Bildpunkten benötigte man mehrere tausend Schritte, um überhaupt einen Fortschritt feststellen zu können. Dies mag vielleicht aber auch an dem Fehlverhalten des Verfahrens liegen.

Weiterhin reagierte das Verfahren sehr sensitiv auf verschiedene Parameterkonfigurationen bezüglich Schrittweite, Regularisierungsstärke und dem Exponenten der Schatten-q-Norm. Damit das Verfahren überhaupt stabil lief, musste man die Norm des Operators aus der Konvergenzbedingung $\sigma\tau < \frac{1}{\|\nabla K\|^2}$ festsetzen, und zwar auf sehr große Werte, die die eigentliche Norm der Operatoren um das Zehnfache überstieg. Auch hierfür sind weitere Untersuchungen anzustellen.

Warum sich die Registrierung so verhalten hat, ist noch nicht ganz klar. Aufgrund der Tests bezüglich des Modells ist es recht unwahrscheinlich, dass der Fehler dort zu suchen ist, da die Energie wie gewünscht minimiert wird. Wahrscheinlicher ist ein Fehler in der Implementierung des Verfahrens. Dort könnte einerseits die Diskretisierung der Operatoren mittels finiter Differenzen ein Problem sein, andererseits ist es auch möglich, dass kleinere, unbeabsichtigte Fehler im Code die Ursache sind.

8.2 Fazit und Ausblick

Die Untersuchungen konnten zeigen, dass sowohl das Modell als auch das Verfahren durchaus für eine Registrierung geeignet wären. Die voneinander unabhängigen Untersuchungen zum Verfahren und zum Modell konnten zeigen, dass diese funktionieren. Warum die Auswertung der zweidimensionalen Experimente fehlgeschlagen ist, muss in folgenden Arbeiten weiter untersucht werden. Wenn man davon ausgeht, dass das Modell mit dem Optimierungsverfahren wie gewünscht funktioniert, müssen in folgenden Arbeiten außerdem die Untersuchungen in höheren Bilddimensionen fortgesetzt werden. Aufgrund der festgestellten Fehler war es in dieser Arbeit nicht möglich, eine Registrierung verschiedener Parameterkanäle durchzuführen. Die Hoffnung diesbezüglich ist, dass durch die Registrierung der verschiedenen Karten ein Mehrwert an Informationen entstünde, die aus den Daten abzuleiten wären. Solche Mehrwerte sind z.B. in anderen Bereichen der Registrierung bereits möglich. Es gibt Verfahren, mithilfe derer man eine Funktionsanalyse z.B. der Lunge vornehmen kann, obwohl die Daten dies ursprünglich nicht zugelassen hätten. Ein solcher Mehrwert wäre für die Registrierung von Parameterkarten wünschenswert. Aufgrund der Vermutung solcher Verbesserungen der Informationen wurde auch der Segmentierungsteil dieser Arbeit geschrieben. Dieser hat nicht nur Ideen für die Parameterkarten geliefert, sondern soll in weiteren Arbeiten auch eine Grundlage bieten, diese Ideen ausbauen zu können. Es ist denkbar, dass die Registrierung der Parameterkarten Informationen liefert, die zu einer erheblichen Vereinfachung der Trennbarkeit verschiedener Gewebearten beitragen kann. Unter anderem zur Bestimmung verschiedener, wichtiger Parameter der Hirndurchblutung müssen verschiedene Gefäß- und Gewebearten unterschieden werden [SB13]. Hier könnte eine solche Registrierung zu einer Verbesserung der automatisch generierten Ergebnisse beitragen.
In folgenden Arbeiten sollen die Methoden zur Erstellung verschiedener Parameterkarten genauer untersucht und erläutert, als auch weitere Verfeinerungen des hier vorgestellten Registrierungsmodells angestrebt werden. Wenn diese Themen vorangetrieben werden, können erste Versuche im Bereich der Segmentierung durchgeführt werden.

Literaturverzeichnis

[AHK+15] Arens, Tilo, Frank Hettlich, Christian Karpfinger, Ulrich Kockelkorn, Klaus Lichtenegger und Hellmuth Stachel: *Mathematik*. Springer-Verlag, 2015.

[Ber99] Bertsekas, Dimitri P: *Nonlinear programming*. Athena scientific Belmont, 1999.

[Bro81] Broit, Chaim: *Optimal registration of deformed images*. 1981.

[BTP13] Bouaziz, Sofien, Andrea Tagliasacchi und Mark Pauly: *Sparse iterative closest point*. In: *Computer graphics forum*, Band 32, Seiten 113–123. Wiley Online Library, 2013.

[BZ13] Burkard, Rainer E und Uwe T Zimmermann: *Einführung in die Mathematische Optimierung*. Springer-Verlag, 2013.

[CA05] Condurache, Alexandru-paul und Til Aach: *Vessel segmentation in angiograms using hysteresis thresholding*. In: *Proceedings of the Ninth IAPR conference on Machine Vision Applications 2005, Tsukuba Science City, Japan*. Citeseer, 2005.

[CJSW01] Cheng, Heng-Da, XH Jiang, Ying Sun und Jingli Wang: *Color image segmentation: advances and prospects*. Pattern recognition, 34(12):2259–2281, 2001.

[CP11] Chambolle, Antonin und Thomas Pock: *A first-order primal-dual algorithm for convex problems with applications to imaging*. Journal of Mathematical Imaging and Vision, 40(1):120–145, 2011.

[CP15] Chambolle, Antonin und Thomas Pock: *On the ergodic convergence rates of a first-order primal–dual algorithm*. Mathematical Programming, Seiten 1–35, 2015.

[Dan98] Dantzig, George Bernard: *Linear programming and extensions*. Princeton university press, 1998.

[DHK16] Drapeau, Samuel, Andreas H Hamel und Michael Kupper: *Complete Duality for Quasiconvex and Convex Set-Valued Functions*. Set-Valued and Variational Analysis, 24(2):253–275, 2016.

[FM81] Fu, King-Sun und JK Mui: *A survey on image segmentation*. Pattern recognition, 13(1):3–16, 1981.

[FM01] Fischer, Bernd und Jan Modersitzki: *A super fast registration algorithm*. In: *Bildverarbeitung für die Medizin 2001*, Seiten 169–173. Springer, 2001.

[FM02a] Fischer, Bernd und Jan Modersitzki: *Curvature based registration with applications to MR-mammography*. In: *International Conference on Computational Science*, Seiten 202–206. Springer, 2002.

[FM02b] Fischer, Bernd und Jan Modersitzki: *Fast diffusion registration*. Contemporary Mathematics, 313:117–128, 2002.

[FM03] Fischer, Bernd und Jan Modersitzki: *Curvature based image registration*. Journal of Mathematical Imaging and Vision, 18(1):81–85, 2003.

[FM08] Fischer, Bernd und Jan Modersitzki: *Ill-posed medicine—an introduction to image registration*. Inverse Problems, 24(3):034008, 2008.

[Gan08] Gander, Walter: *The Singular Value Decomposition*. 2008.

[GD60] Greenwood, J Arthur und David Durand: *Aids for fitting the gamma distribution by maximum likelihood*. Technometrics, 2(1):55–65, 1960.

[GMW81] Gill, Philip E, Walter Murray und Margaret H Wright: *Practical optimization*. 1981.

[Had02] Hadamard, Jacques: *Sur les problèmes aux dérivées partielles et leur signification physique*. Princeton university bulletin, 13(49-52):28, 1902.

[Him72] Himmelblau, David Mautner: *Applied nonlinear programming*. McGraw-Hill Companies, 1972.

[HM05] Haber, Eldad und Jan Modersitzki: *Beyond mutual information: A simple and robust alternative.* In: *Bildverarbeitung für die Medizin 2005*, Seiten 350–354. Springer, 2005.

[HM06] Haber, Eldad und Jan Modersitzki: *Intensity gradient based registration and fusion of multi-modal images.* In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Seiten 726–733. Springer, 2006.

[HS81] Horn, Berthold KP und Brian G Schunck: *Determining optical flow.* Artificial intelligence, 17(1-3):185–203, 1981.

[Hue71] Hueckel, Manfred H: *An operator which locates edges in digitized pictures.* Journal of the ACM (JACM), 18(1):113–125, 1971.

[KF09] Krishnan, Dilip und Rob Fergus: *Fast image deconvolution using hyper-Laplacian priors.* In: *Advances in Neural Information Processing Systems*, Seiten 1033–1041, 2009.

[Kro08] Kroon, Dirk-Jan: *Region Growing.* https://de.mathworks.com/matlabcentral/fileexchange/19084-region-growing/content/regiongrowing.m, 2008. [Online; Abgerufen 10-November-2016].

[KWT88] Kass, Michael, Andrew Witkin und Demetri Terzopoulos: *Snakes: Active contour models.* International journal of computer vision, 1(4):321–331, 1988.

[Lel13] Lellmann, Jan: *Convex Optimization with Applications to Image Processing.* Vorlesungsskript, University of Cambridge, 2013.

[Lue73] Luenberger, David G: *Introduction to linear and nonlinear programming*, Band 28. Addison-Wesley Reading, MA, 1973.

[Mod04] Modersitzki, Jan: *Numerical methods for image registration.* Oxford University Press on Demand, 2004.

[Mod09] Modersitzki, Jan: *FAIR: flexible algorithms for image registration*, Band 6. SIAM, 2009.

[MSMC15a] Möllenhoff, Thomas, Evgeny Strekalovskiy, Michael Möller und Daniel Cremers: *Low rank priors for color image regularization.* In: *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Seiten 126–140. Springer, 2015.

[MSMC15b] Möllenhoff, Thomas, Evgeny Strekalovskiy, Michael Möller und Daniel Cremers: *The primal-dual hybrid gradient method for semi-convex splittings.* SIAM Journal on Imaging Sciences, 8(2):827–857, 2015.

[MV98] Maintz, JB Antoine und Max A Viergever: *A survey of medical image registration.* Medical image analysis, 2(1):1–36, 1998.

[Nev77] Nevatia, Remakant: *Color Edge Detector and Its Use in Scene Segmentation*, 1977.

[Nev82] Nevatia, Ramakant: *Machine perception.* PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, NJ 07632, 1982, 209, 1982.

[Pap08] Papenberg, Nils: *Ein genereller Registrierungsansatz mit Anwendungen in der navigierten Leberchirurgie.* Doktorarbeit, Universität zu Lübeck, 2008.

[Pol12] Polzin, Thomas: *Lungenregistrierung mittels automatisch detektierter Landmarken.* Masterarbeit, Universität zu Lübeck, 2012.

[PP93] Pal, Nikhil R und Sankar K Pal: *A review on image segmentation techniques.* Pattern recognition, 26(9):1277–1294, 1993.

[Roc70] Rockafellar, Ralph Tyrell: *Convex analysis.* Princeton university press, 1970.

[Sau10] Sauer, Rolf: *Strahlentherapie und Onkologie.* Elsevier, Urban&FischerVerlag, 2010.

[SB13] Sourbron, Steven P und David L Buckley: *Classic models for dynamic contrast-enhanced MRI.* NMR in Biomedicine, 26(8):1004–1027, 2013.

[SC14] Strekalovskiy, Evgeny und Daniel Cremers: *Real-time minimization of the piecewise smooth Mumford-Shah functional.* In: *European Conference on Computer Vision*, Seiten 127–141. Springer, 2014.

[Sch13] Schatten, Robert: *Norm ideals of completely continuous operators*, Band 27. Springer-Verlag, 2013.

[SH81] Simon, Jean Claude und Robert M. Haralick: *Digital Image Processing*. D. Reidel Publishing Company, 1981.

[Sou10] Sourbron, Steven: *Technical aspects of MR perfusion*. European journal of radiology, 76(3):304–313, 2010.

[TB97] Tremeau, Alain und Nathalie Borel: *A region growing and merging algorithm to color segmentation*. Pattern recognition, 30(7):1191–1203, 1997.

[TBU00] Thévenaz, Philippe, Thierry Blu und Michael Unser: *Image interpolation and resampling*. Handbook of medical imaging, processing and analysis, Seiten 393–420, 2000.

[Val14] Valkonen, Tuomo: *A primal–dual hybrid gradient method for nonlinear operators with applications to MRI*. Inverse Problems, 30(5):055012, 2014.

[Wat11] Watrous, John: *Theory of Quantum Information, 2.3 Norms of operators*. lecture notes, University of Waterloo, 2011.

[Wei98] Weickert, Joachim: *Anisotropic diffusion in image processing*, Band 1. Teubner Stuttgart, 1998.

[ZB13] Zimmer-Brossy, Marianne: *Lehrbuch der röntgendiagnostischen Einstelltechnik: Begründet von Marianne Zimmer-Brossy*. Springer-Verlag, 2013.

[ZF03] Zitova, Barbara und Jan Flusser: *Image registration methods: a survey*. Image and vision computing, 21(11):977–1000, 2003.

[Zuc76] Zucker, Steven W: *Region growing: Childhood and adolescence*. Computer graphics and image processing, 5(3):382–399, 1976.