



Aus dem Institut für Mathematik
der Universität zu Lübeck
Direktor: Prof. Dr. Jürgen Prestin

Simultane Homogenisierung und Registrierung von Bildern histologischer Serienschnitte

Inauguraldissertation

zur

Erlangung der Doktorwürde
der Universität zu Lübeck

- Aus der Technisch-Naturwissenschaftlichen Fakultät -

Vorgelegt von

Stefan Wirtz

aus Köln

Lübeck, den 7. Oktober 2009

Zusammenfassung

In dieser Arbeit beschäftigen wir uns mit der Registrierung, Homogenisierung und Rekonstruktion histologischer Bilddaten. Ausgangspunkt und damit Motivation dieser Arbeit sind Bilder von histologischen Serienschichten von Gehirnen von Maus, Ratte und Mensch. Diese histologischen Serienschichten, etwa 6000 Einzelschichten für ein menschliches Gehirn und ca 500-600 Einzelschichten für eine Maus oder Ratte, sind durch den Verarbeitungsprozess derart deformiert, dass ein erneutes Zusammenfügen der Schichten nicht zu einer dreidimensionalen Rekonstruktion des Gehirns führt. Neben den Deformationen sind Inhomogenitäten innerhalb der Bilder, die durch den manuellen Färbeprozess entstanden sind, dafür verantwortlich, dass benachbarte Schichten deutliche Intensitätsunterschiede bei gleichen Strukturen aufweisen und damit nicht mehr ohne weiteres zueinander passen.

Der Schwerpunkt dieser Arbeit liegt auf der Bildregistrierung. Aufgrund der Anwendung der Registrierung in der Histologie zur Rekonstruktion von Serienschichten kommt darüber hinaus die Notwendigkeit der Homogenisierung auf. Da sich die Homogenisierung und Registrierung gegenseitig bedingen ist die Idee der simultanen Registrierung und Homogenisierung geboren, bei der die Bilder nicht zuerst homogenisiert und anschließend registriert werden, sondern beide Schritte auf einmal gemacht werden. Formuliert wird diese Beziehung durch ein Minimierungsproblem, das bezüglich einer Deformation (Registrierung) und eines Korrekturterms (Homogenisierung) optimiert wird. Die zugehörigen Euler-Lagrange-Gleichungen stellen eine notwendige Bedingung für ein Minimum dar und führen zu einem System nicht-linearer partieller Differentialgleichungen. Für dieses System werden Lösungsmöglichkeiten vorgestellt und diskutiert.

Neben der Lösung des Systems nicht-linearer partieller Differentialgleichungen ist eine der größten praktischen Herausforderungen dieser Arbeit die Bewältigung der enormen Datenmengen, die bei Bildern von histologischen Serienschichten entstehen. Daher werden neben effizienten Registrierungsmethoden auch sinnvolle Strategien für Speicherbedarfsreduktion und verbessertes Datenhandling kurz vorgestellt. Eine Option zur automatischen Wahl von Materialkonstanten ermöglicht eine vollautomatische elastische Registrierung, die keine weitere Benutzerinteraktion notwendig macht. Einige wenige so genannte Stellschrauben geben dennoch dem Benutzer die Möglichkeit, lenkenden Einfluss auf die Registrierung auszuüben, um so individuelle Schwerpunkte und Korrekturen setzen zu können.

Danksagung

Zu Beginn möchte ich mich bei einigen Menschen ganz besonders bedanken, die auf ihre eigene Art und Weise zum Gelingen dieser Arbeit beigetragen haben.

Bei Bernd Fischer der schon an mich glaubte, als ich noch Student war, mich an die Hand nahm und in die Welt der Registrierung einführte. Er lehrte mich Vorträge spannend zu halten und die Menschen dabei zu erreichen. Er lehrte mich Mensch zu sein und zu bleiben, auch wenn man damit manchmal gegen den Strom schwimmt. Er lehrte mich, dass die Familie das wichtigste auf der Welt ist und dass mit einer liebevollen Familie alles möglich ist. Danke, Bernd, dass Du mich nicht aufgegeben hast.

Bei Oliver Schmitt, dem ich dieses Thema und alle nur denkbare medizinisch-fachliche Unterstützung verdanke. Der mir immer wieder bewiesen hat, dass Dinge erreichbar sind, die man sich anfangs kaum vorstellen konnte. Mit dem ich intensive Stunden des Austauschs verbringen durfte und der mir trotz aller Fachlichkeit stets als Mensch begegnete.

Bei Jan Modersitzki, der mir auf seine spezielle, aber sehr liebenswerte Art die Registrierung von einer ganz anderen Seite gezeigt hat und damit meinen Horizont stark erweitert hat. Der mich mit einem Lächeln auf den Lippen an meine Grenzen gebracht und trotzdem die Hand zum Schutze hielt.

Bei Stefan Heldmann und Nils Papenberg für viele fruchtbare Gespräche, den gemeinsamen Weg in einer tollen Arbeitsgruppe und für die langjährige Freundschaft.

Bei Tobias Böhler für seine Freundschaft, viele sehr schöne Stunden der gemeinsamen Arbeit, die zahlreichen Diskussionen, die erfrischenden Ideen, seine Menschlichkeit. Bei Richard Rascher-Friesenhausen für seine bestechend klaren Analysen und Hinterfragungen und damit einher kommenden wichtigen Hinweise und für den gelegentlich notwendigen Tritt, um auf dem Weg zu bleiben. Bei Dörte van Straaten fürs akribische Lesen und Korrigieren der Literaturstellen.

Bei allen Kolleginnen und Kollegen am Fraunhofer Institut MEVIS in Bremen für die täglich neue und begeisterte Zusammenarbeit, den vielseitigen Möglichkeiten, die mir dort geboten werden und den Herausforderungen, denen ich mich immer wieder neu stellen darf, besonders bei der Institutsleitung, die immer Verständnis

für mich zeigten und mir den nötigen Freiraum für meine Arbeit gaben. Bei allen Mitarbeitern des Instituts für Mathematik in Lübeck, die mich ein gutes Stück meines frühen wissenschaftlichen Wegs begleitet haben.

Bei meinen Eltern Doris und Matthias, ohne die ich jetzt nicht hier wäre und die zusammen mit meinen Großeltern mich in vielerlei Hinsicht unterstützten und immer wieder antrieben, diese Arbeit zu Ende zu bringen.

Bei meinen Kindern Noah, Melina-Maja und Dominik, die mich oft entbehren mussten und dennoch mir uneingeschränkt ihre Liebe schenkten.

Und aus tiefster Seele und von ganzem Herzen meiner geliebten Frau Tanja, die alles ertragen und mich immer unterstützt hat.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Histologische Serienschnitte - eine Einführung	2
1.1.1	Maushirn-Projekt	4
1.1.2	Herstellungsprozess	4
1.2	Schwierigkeiten bei der Bearbeitung von Bildserien	9
1.2.1	Verlust der ursprünglichen Geometrie und Topologie	9
1.2.2	Artefakte durch die Bildherstellung	10
1.2.3	Inhomogene Färbung	11
1.2.4	Extreme Datengröße	12
2	Registrierungsverfahren für Bilder von Serienschnitten	13
2.1	Bild und Deformationsfeld	14
2.2	Affin-lineare Registrierung	15
2.3	Elastische Registrierung	18
3	Homogenisierung von Bildern	23
3.1	Inhomogenitäten in Serienschnitten	23
3.2	Erweiterte HUM-Methode	25
3.3	Total Variation	27
3.3.1	TV-Entrauschen bzw. TV-Glättung nach Rudin et al.	28
3.3.2	Abwandlung des Verfahrens	31
3.3.3	Anwendung auf Bilder der Serienschnitte	33
3.3.4	Hinweise zur Implementierung	34
3.3.5	Regularisierung der TV-Norm	35
4	Simultane Homogenisierung und Registrierung	37
4.1	Die Methode: Aufstellen des Minimierungsproblems	37
4.1.1	Distanzmaß	37
4.1.2	Registrierungsproblem	41
4.2	Herleitung der Euler-Lagrange-Gleichungen	44
4.3	Linearisierung der PDG	48
4.4	Erweiterung der Methode für Serienschnitte	50
5	Diskretisierung	57
5.1	Diskrete Bilder	57
5.2	Lineare Interpolation	60
5.3	Differenzenquotienten	62

5.4	Diskrete Euler-Lagrange-Gleichungen	66
6	Implementierung	69
6.1	Lösungsverfahren	69
6.1.1	Direktes Lösungsverfahren zur elastischen Registrierung . . .	69
6.1.2	Approximatives Lösungsverfahren mit FIR-Filtern	77
6.1.3	Lösungsverfahren zur TV-Regularisierung	81
6.2	Strategien für einen robusteren Algorithmus	82
6.2.1	Multiresolutionsansatz	82
6.2.2	Abbruchkriterien	84
6.2.3	Schrittweitensteuerung	85
6.2.4	Automatische Wahl der Lamé-Konstanten	85
7	Ergebnisse und Diskussionen anhand von Beispielen	87
7.1	Akademische Beispiele	87
7.2	Reale Beispiele 2D	95
7.3	Serienschnittregistrierung	98
8	Projekte	105
8.1	Scanning Probe Microscopy, Bayreuth/Chemnitz	105
8.2	SSD-Distanzmaß höherer Ordnung	107
8.3	FLIRT - Universität zu Lübeck	109
8.3.1	Der generelle Entwurf	109
8.3.2	FLIRT-Integration in MeVisLab	112
A	Mathematische Begriffe und Hilfsmittel	117
A.1	Gaußscher Integralsatz und Greensche Formel	117
A.2	Einführung in die Variationsrechnung	120
B	Ausführliche Rechnungen & Ergänzungen	125
B.1	Gâteaux-Ableitung von $\mathcal{D}^{\text{cSSD}}$	125
B.2	Gâteaux-Ableitung von $\mathcal{S}^{\text{elast}}$	125
B.3	Gâteaux-Ableitung von \mathcal{S}^{TV}	129
B.4	Iterationsvorschrift zum TV Entrauschen mit Standardabweichung .	129
B.5	Herleitung der Euler-Lagrange-Gleichungen zur TV Bildglättung . .	131
B.6	Herleitung der Bestimmung von λ	132
C	Matlab Implementierungen	135
C.1	Kochrezept zur Homogenisierung	135
C.2	FIR-Filter	136
C.3	Registrierung & Homogenisierung	137
	Literaturverzeichnis	145

1 Einleitung

In dieser Arbeit beschäftigen wir uns mit der Registrierung, Homogenisierung und Rekonstruktion histologischer Bilddaten. Der Schwerpunkt dieser Arbeit liegt dabei auf der Bildregistrierung. Aufgrund der Anwendung der Registrierung in der Histologie zur Rekonstruktion von Serienschnitten kam die Notwendigkeit der Homogenisierung auf. Da sich die Homogenisierung und Registrierung gegenseitig bedingen ist die Idee der simultanen Registrierung und Homogenisierung geboren.

Eine der größten Herausforderungen dieser Arbeit ist die Bewältigung der enormen Datenmengen, die bei Bildern von histologischen Serienschnitten entstehen. Daher müssen neben effizienten Registrierungsmethoden auch sinnvolle Strategien für Speicherverwaltung, Festplattenmanagement und Datenhandling gefunden werden. Des Weiteren soll eine möglichst automatische Wahl von Parametern eine vollautomatische Registrierung ermöglichen, die keine weitere Benutzerinteraktion notwendig macht. Einige wenige so genannte Stellschrauben sollen dennoch dem Benutzer die Möglichkeit geben, lenkenden Einfluss auf die Registrierung auszuüben, um so individuelle Schwerpunkte und Korrekturen setzen zu können.

Die Arbeit ist in insgesamt acht Kapitel aufgeteilt und durch drei Anhänge ergänzt. Nach der Darlegung der Motivation sprechen wir in Kapitel 2 zwei Möglichkeiten der Registrierung von Bildern und damit zur Rekonstruktion von Bildserien an. Im darauf folgenden Kapitel 3 sprechen wir über die Möglichkeiten der Homogenisierung. Anschließend stellen wir die grundsätzliche Methode der simultanen Homogenisierung und Registrierung (Kapitel 4) vor. In diesem Kapitel werden alle notwendigen Begriffe definiert und eine Lösungsstrategie für die simultane Homogenisierung und Registrierung entwickelt. Um den gesamten Entwicklungsprozess bestmöglich verfolgen zu können, haben wir auf ausführliche Rechnungen und Einführungen in mathematische Grundlagen (Variationsrechnung) in diesem Kapitel verzichtet und in den Anhang gesetzt. Bevor die Ergebnisse der Algorithmen in Kapitel 7 vorgestellt werden können, beschäftigen wir uns in den Kapiteln 5 und 6 mit der Diskretisierung und der Implementierung der mathematischen Methode. Im letzten Kapitel 8 stellen wir Projekte vor, in denen die in dieser Arbeit verwendete Methoden getestet oder angewendet wurden, bzw. Methoden entwickelt wurden, die nicht in dieser Arbeit ihren Platz fanden. In Anhang A werden mathematische Grundlagen wie Variationsrechnung und Gaußscher Integralsatz vorgestellt. Alle ausführlichen Rechnungen und Ergänzungen zu Kapitel 4 finden sich in Anhang B. Schließlich werden Matlab-Implementierungen beispielhaft in Anhang C aufgeführt.

In den nächsten Abschnitten wollen wir den medizinischen Hintergrund dieser

Arbeit beleuchten, der immer wieder die Motivation gebracht hat, den oftmals technischen Schwierigkeiten entgegen zu treten und diese Arbeit voran zu bringen. Mit der Rekonstruktion histologischer Serienschritte hat die Arbeit an der Registrierung von Bildserien begonnen. Die ersten Forschungen und Entwicklungen im Rahmen dieser Arbeit beschränkten sich ausschließlich mit diesem Bereich. Erst in späteren Phasen der Arbeit, besonders während der praktischen Entwicklung (Implementierung), wurde der Fokus auf die generelle Sicht der Registrierung von Bildserien gelenkt, um dadurch auch eine Übertragung auf andere Teilgebiete zu ermöglichen. Daher wird hier in dieser Arbeit im wesentlichen die spezielle Sicht auf die Rekonstruktion von Bildern von histologischen Serienschritten gelegt, in Kapitel 8 aber zusätzlich kurz ein weiteres Anwendungsgebiet der Registrierung von Serienschritten erläutert.

1.1 Histologische Serienschritte - eine Einführung

Moderne bildgebende Verfahren wie die *Magnet Resonanz Tomographie* (MRT) oder die *Computer Tomographie* (CT) können für die medizinische Diagnostik hervorragende Bilder in diagnostisch relevanter Aufnahmequalität liefern. Für Untersuchungen auf neuronaler Ebene sind sie aber wegen der zu geringen Ortsauflösung (ca. $1\text{ mm} \times 1\text{ mm}$ pro Pixel, bei modernsten Scannern bis zu $0.5\text{ mm} \times 0.5\text{ mm}$) nicht einsetzbar. Nur spezielle Kleintier-MR-Tomographen erlauben räumliche Auflösungen im Mikrometerbereich (MR-Mikroskopie, μMRT) [JCGH02], wobei zusätzliche messtechnische Schwierigkeiten die Bilder ungewünscht verzerren.

In den Neurowissenschaften sind vor allem Strukturen in Neuronengröße, also im μm -Bereich essentiell. Neben einzelnen Neuronen sind die Verbindungen dieser untereinander von Bedeutung und können mit optischen Verfahren wie z.B. Mikroskopie dargestellt werden. Dadurch lassen sich Aussagen über die Zellverteilung, die Zellgröße und Konnektivität in den Bildern erkennen. Wegen des sehr geringen Kontrastes von Neuronen und umliegendem Gewebe ist grundsätzlich eine Färbung [SE97], z.B. nach Gallyas [GHB93], notwendig. Neuronen und andere Zellen des Nervensystems lassen sich nach einer geeigneten Färbung leicht mit der Licht- bzw. Videomikroskopie darstellen, wo Auflösungen mit klassischer Lichtmikroskopie (LM) von ca. $333\mu\text{m}$ möglich sind [Rie88, Gök88].

In der medizinischen Histologie ist es schon lange Zeit üblich, interessante Objekte bzw. Spezies in dünne Schichten zu zerschneiden, um so ein spezielles Einfärben und anschließendes Durchleuchten mittels Lichtmikroskopie zu ermöglichen. Auf diese Weise können auch dreidimensionale (3D) Strukturen dargestellt, oder auch die Verteilung biologischer Objekte wie Zellen, Fasern oder Proteinen weiter untersucht werden. Die durch diesen Prozess entstandenen Schichten können direkt von *Serienschritten* gewonnen werden, oder durch optische Tomographie innerhalb eines histologischen Schnittes. Da es sich um Bilder eindeutiger, konsekutiver Ordnung handelt, lassen sich komplexe dreidimensionale Strukturen von Schnitt zu Schnitt verfolgen. Durch den Schneideprozess verliert man allerdings die dreidi-

mensionale Information, bzw. die ursprüngliche Geometrie und Topologie, wodurch beispielsweise eine Verfolgung einer Nervenbahn oder eines Gefäßes über mehrere Schichten hinweg sehr aufwendig, teilweise sogar unmöglich ist. Daher ist man bereits sehr früh an einer 3D-Rekonstruktion der *Bilder* von diesen Schichten interessiert gewesen, um so die Schichtkorrespondenzen zu vereinfachen. 1883 veröffentlichte Born [Bor83] die erste 3D-Rekonstruktion, in der er Serienschritte und eine Modellierung über eine Wachsplattentechnik verwendete. Die erste Computer-Rekonstruktion stammt aus dem Jahr 1965 von Glaser und Glaser [GG65]. Sie benutzten lichtmikroskopische Bilder, die von Serienschritten hergestellt wurden. Seit dieser Zeit wurden immer wieder neue Veröffentlichungen zur Rekonstruktion von Serienschritten herausgebracht [BLJ⁺90, GBJ⁺91, SF84, Sch96, SZ98, ORS⁺01]. Die verwendeten Techniken wurden dabei teilweise ausschliesslich für einen einzigen speziellen Datensatz entwickelt und optimiert.

Serienschritte zu erzeugen, zweckentsprechend einzufärben und unter dem Mikroskop zu untersuchen, ist speziell wegen der hohen örtlichen Auflösung nur durch aufwendige und weniger flexible Techniken wie der Elektronenmikroskopie oder Laser-Scanning-Mikroskopie zu überbieten. Allerdings birgt die Lichtmikroskopie auch einige herstellungstechnische Schwierigkeiten. Wegen des Schneidens, Färbens und Aufziehens der Schnitte auf Objektträger ist eine perfekte Überlagerung aller Schnittbilder nicht mehr möglich. Durch das Schneiden entstehen nichtlineare Deformationen. Färbungen sind nicht homogen durchführbar, sowohl innerhalb einzelner Schichten, als auch von Schicht zu Schicht. Beim Aufbringen auf den Objektträger können Risse und/oder Falten entstehen. Diese nicht auszuschliessenden Störungen müssen, sofern möglich, manuell oder digital korrigiert werden. Die Korrektur der Deformation wird als *Registrierung* bezeichnet, die Korrektur der Färbung als *Homogenisierung*.

In erster Linie steht in dieser Arbeit die Registrierung im Vordergrund. Die ersten in der Literatur erwähnten Rekonstruktionen arbeiteten mit manueller Registrierung [WL75, Die76, MLS79, PG82, JC83, SM83, ACC91, BS94], halbautomatisch oder für eine ganz spezielle Serie. Eine Ausrichtung der einzelnen Schichten von Hand erfordert eine sehr hohe manuelle Präzision und vor allem viel Zeit, gerade bei vielen Schichten. Daher ist eine automatische Registrierung essentiell, um reproduzierbar, unabhängig von der speziellen Serie und auch im größeren Umfang Rekonstruieren zu können. Die einsetzbaren Verfahren unterscheiden sich in zwei wesentliche Klassen: affin-lineare und rigide Methoden (z.B. die Hauptachsentransformation, principle axis transformation (PAT)) [BT80, HH88, ABKC90, ZYG93, RTL⁺93, BT94, TB93, TAQ⁺94, SZ97, SDZ97] und freie, nicht-lineare bzw. nicht-rigide Methoden (z.B. elastische oder fluidale Registrierung) [Bro81, HS81, BK89, Chr94, Vio95, VW95, BNG96, Thi98, MOSL99, SE99, SMO99, FM02, Wir00, Wir03]. Eine Übersicht über verschiedene Registrierungsverfahren in unterschiedlichen Bereichen finden sich beispielsweise in [Bro92, vdEPV93, TB93, MV98, LA99, HBHH01, Mod04].

1.1.1 Maushirn-Projekt

Die grundlegende Motivation dieser Arbeit kommt aus dem *Maushirn-Projekt* [SWM⁺04] von Schmitt [Sch01]. Ziel dieses Projekts ist es, wie auch im *Human Neuroscanning Project* [SEM05, SMO99], höchst aufgelöste digitale Atlanten mit funktionell, strukturell und symbolisch relevanten Variablen zu erzeugen. Dazu ist es notwendig, Bilder in einer Auflösung zu erzeugen, in welcher strukturell relevante, bzw. zu analysierende Objekte unterscheidbar sind. Diese Informationen lassen sich auf zellulärer Ebene analysieren und schliesslich in 3D-Modellen rekonstruieren. Wichtige Informationen sind neben der Zelldichte, der Verteilung der Zellen und der durchschnittlichen Zellgröße auch die Benennung funktioneller und zusammenhängender Einheiten.

Ist solch ein Atlas einmal eingerichtet, können z.B. mit stereotaktischen Positionierungen Vergleiche mit Erkrankungen und genetischen Veränderungen durchgeführt und die kodierten Informationen auf neue Datensätze übertragen werden. Das benötigte Bildmaterial für solche Atlanten kann ausschliesslich durch Serienschritte gewonnen werden, da es keine alternativen Verfahren gibt, die solche Bildqualitäten und spezifischen Informationen liefern.

1.1.2 Herstellungsprozess

In diesem Abschnitt soll der Herstellungsprozess, der für das Maushirn-Projekt benutzt wurde, ausführlich dargestellt werden, um so nicht nur den Ablauf, sondern auch die auftretenden Probleme in den Bildern besser verstehen zu können.



Abbildung 1.1: Präparation des Maushirns (*links*), Maushirn in Paraffin eingebettet (*rechts*).

Für die Datensätze, die innerhalb dieser Arbeit dargestellt werden, wurde eine spezielle Labor-Zucht-Maus, vom Typ C57BL/6J, verwendet. Die Maus war männlich und 60 Tage alt. Prinzipiell lässt sich die Herstellung des Bildmaterials in zwei grundlegend verschiedene Teile zerlegen. Zuerst einmal dem biologischen bzw. histologischen Teil, in dem im wesentlichen handwerkliche Dinge geschehen, Paraffineinbettung und Mikrotomie (Anfertigen von $> 5\mu\text{m}$ dicken Schnitten), und

dem computerunterstützten bzw. digitalen Teil, in dem die Bilder digitalisiert und weiterverarbeitet werden. In Abbildung 1.7 sind die einzelnen Schritte schematisch dargestellt. An den biologischen Teil schliesst sich dann der computergestützte Teil an und ist damit der Anfang der *digitalen Rekonstruktion*.

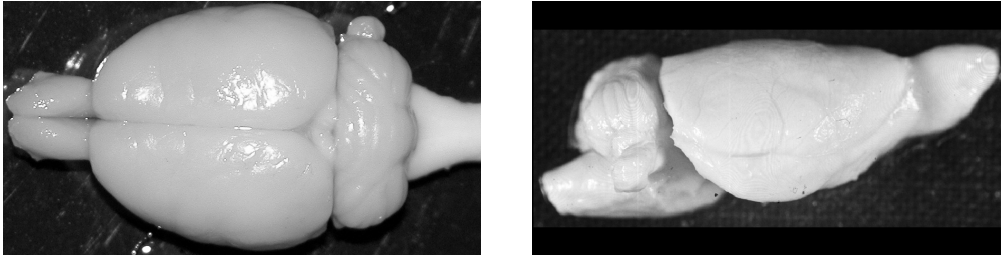


Abbildung 1.2: Immersionsfixiertes Hirn, sagittal (*links*), longitudinal (*rechts*).

Die Herstellung der histologischen Serienschritte ist ein zeitaufwändiger Prozess. Ein Hirn einer Maus kann bei intensiver Arbeit in ca. 14 Tagen wie folgt verarbeitet werden. Die noch lebende Maus wird zunächst anästhesiert. Transkardial wird das Blut mit Hilfe einer Kochsalzlösung (NaCl) ausgeschwemmt, während dessen das Tier stirbt. Nach vollständiger Ausschwemmung des Blutes wird eine 6% PBS Formalin Perfusion, ebenfalls transkardial, durchgeführt, um so den Körper der Maus vollständig zu konservieren. Anschliessend wird das Hirn der Maus herauspräpariert (s. Abb. 1.1, *links*) und über 24 Stunden immersionsfixiert (s. Abb. 1.2). Schliesslich wird das Hirn in Paraffin eingebettet (s. Abb. 1.1, *rechts*). Während des gesamten Vorgangs schrumpft das Hirn um ca. 40%, wodurch eine wesentliche Geometrieänderung hervorgerufen wird. Durch das Paraffin ist das Hirn einerseits so stark fixiert, dass es durch Lagerung nicht mehr die äußere Form verändert, andererseits kann es bequem zerschnitten werden.

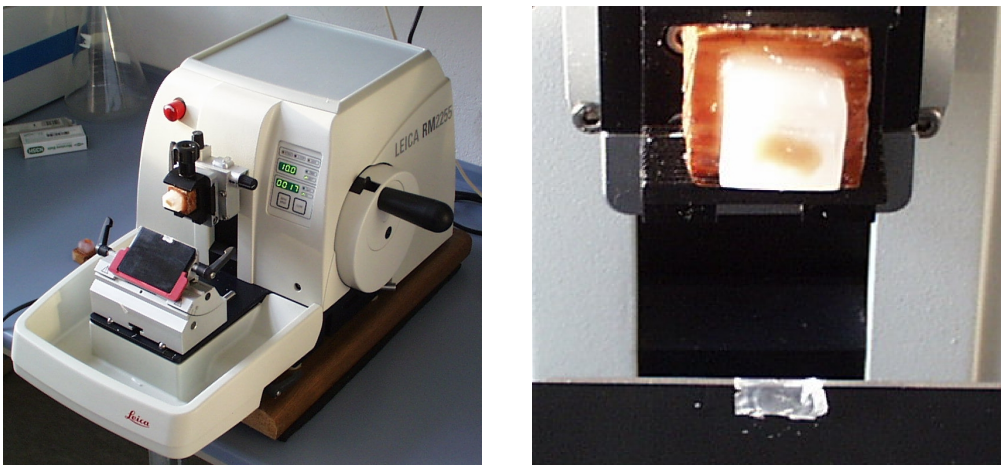


Abbildung 1.3: Rotationsmikrotom (*links*), Schnitteinheit mit Paraffinblock oben und abgehobelter Schicht unten (*rechts*).

Im nächsten Abschnitt des histologischen Teils werden mit Hilfe eines Rotationsmikrotoms (s. Abb. 1.3, *links*) aus dem Paraffinblock coronale bzw. frontale, $20\ \mu\text{m}$ dünne Schnitte erzeugt (s. Abb. 1.3, *rechts*). Wie man in der Abbildung erkennt, entstehen dadurch starke Stauchungen der einzelnen Schnitte, die in einem 54°C warmen Wasserbad im wesentlichen rückgängig gemacht bzw. geglättet werden. Die gestreckten Schichten werden dann mit einem Objektträger aus dem Wasserbad genommen und auf diesem entparaffiniert [Rom89]. Nach einer Methode von Gallyas [GHB93] werden die Schichten anschliessend eingefärbt, um so den Kontrast in den interessanten Bereichen zu erhöhen, bzw. die Strukturen besser sichtbar zu machen. Schliesslich, als Abschluss des biologischen Teils, werden die

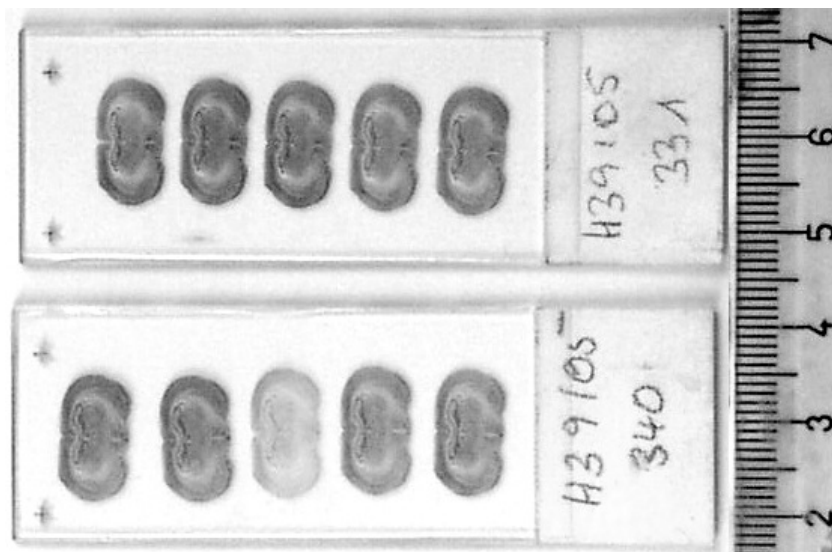


Abbildung 1.4: Fertiger Schnitt auf Objektträger, gefärbt und eingedeckt.

Schichten auf dem Objektträger eingedeckt (s. Abb. 1.4) und damit vor äusseren Einflüssen geschützt. Nun spricht man von *histologischen Serienschnitten*.

Bevor die Serienschnitte zu einem Volumen rekonstruiert werden können, müssen sie digitalisiert werden. In unserem Fall ist dies mit einem Scanner der Firma Heidelberg (Nexscan F4100) geschehen. Der Scanner arbeitet mit einer maximalen Auflösung von 5040 PPCM (Pixel pro Zentimeter), was einer isotropen lateralen Auflösung eines Schnittbildes von $5.04\ \mu\text{m}$ pro Pixel bedeutet. Ein Voxel hat damit eine Auflösung von $5.04\ \mu\text{m} \times 5.04\ \mu\text{m} \times 20\ \mu\text{m}$. Die eingescannten Bilder werden anschliessend manuell durch einen Experten fragmentkorrigiert (s. Abb. 1.5), d.h. Teile der eigentlich zusammenhängenden Schicht, die durch den Herstellungsprozess auseinander gerissen wurden, werden manuell (nahe) an ihre korrekte Position verschoben, was mit der Funktionalität von Puzzle-Lösern (jigsaw assembling) [GMB02, CFF98, BBB05] vergleichbar ist.

Nach dieser zeitaufwändigen Prozedur wird durch eine Segmentierung nach Otsu [Ots79] Objekt und Hintergrund voneinander getrennt (s. Abb. 1.6). Dadurch kann gewährleistet werden, dass im Bild Pixel, die nicht zum Objekt gehören, einen

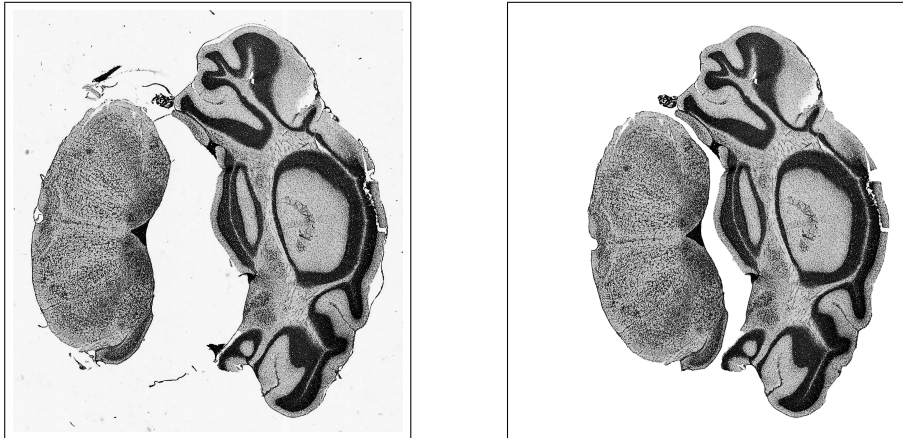


Abbildung 1.5: Originalbild Nr.423 direkt nach dem Scannen (*links*), fragmentkorrigiertes und segmentiertes Bild Nr.423 (*rechts*).

einheitlichen Grauwert besitzen und Fremdkörper wie Haare und Luftblasen die Registrierung nicht negativ beeinflussen.

Für einige Algorithmen, in unserem Fall ist dies die Bildregistrierung, ist es notwendig, dass die Bildgrößen innerhalb der Serie identisch sind, so dass im nächsten Schritt die Größe der Bilder angepasst wird. Nach dieser Bildgrößenanpassung kann die Registrierung und Homogenisierung der Bilder durchgeführt werden. Begonnen wird mit einer linearen Vorregistrierung, beispielsweise eine Hauptachsentransformation, oder einer vollen affin-linearen Registrierung. Diese Vorregistrierung ist notwendig, damit die teilweise großen globalen Transformationen (im wesentlichen Translation und Rotation) durch eine algorithmisch einfache und besonders schnelle Registrierung eingefangen werden. Die feinen lokalen Deformationen werden anschließend durch eine elastische Registrierung kompensiert. Optional kann ein Homogenisierungsschritt eingefügt werden, oder wie hier, mit der elastischen Registrierung simultan laufen.

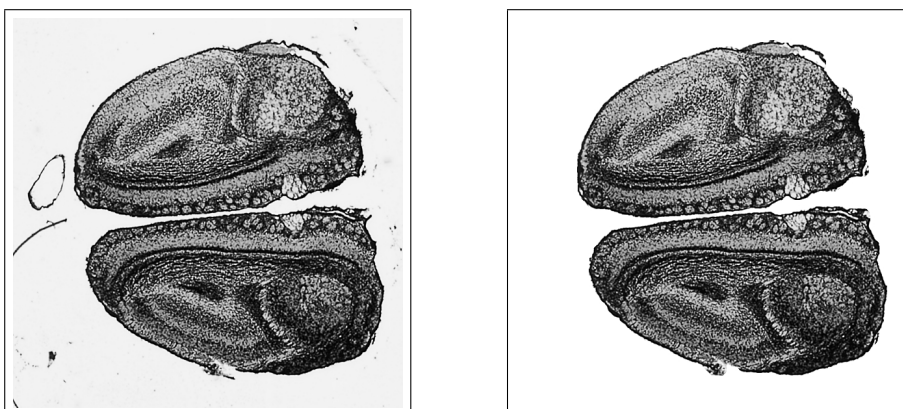


Abbildung 1.6: Originalbild Nr.83 direkt nach dem Scannen (*links*), segmentiertes Bild Nr.83 (*rechts*).

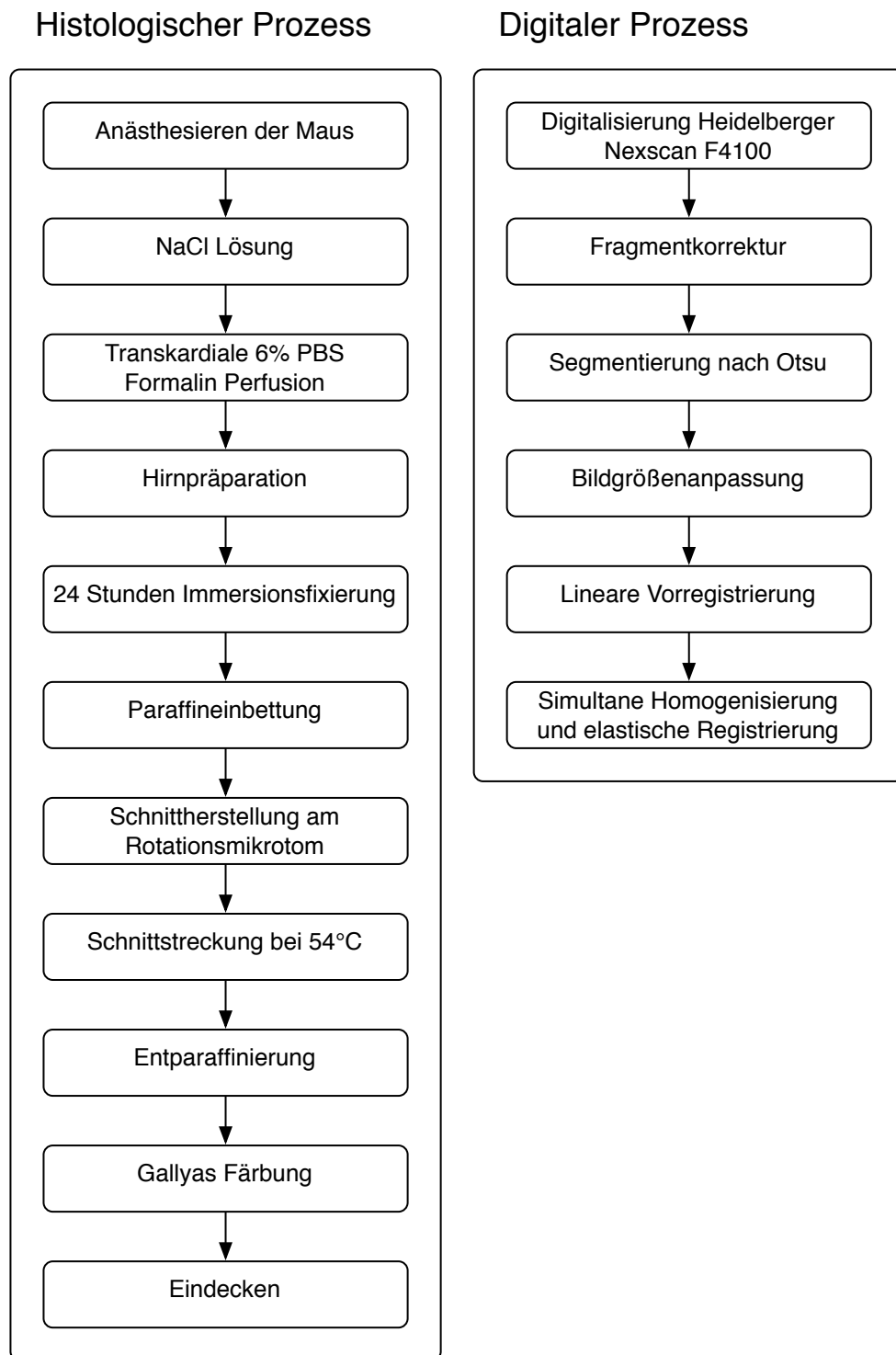


Abbildung 1.7: Prozessablaufplan zur Herstellung der histologischen Serienschritte im Maushirn-Projekt.

1.2 Schwierigkeiten bei der Bearbeitung von Bildserien

Im vorherigen Abschnitt wurden histologische Serienschritte als Motivation der Notwendigkeit der Registrierung von Bildserien vorgestellt. In diesem Abschnitt wollen wir die generellen Schwierigkeiten bei der Registrierung und Bearbeitung von Bildserien konzentriert herausarbeiten und weiterführende Schritte erläutern.

1.2.1 Verlust der ursprünglichen Geometrie und Topologie

Während des gesamten histologischen Prozesses wird die ursprüngliche Geometrie und Topologie des Hirns sehr stark verändert. Im Vergleich zum Hirn der lebenden Maus schrumpft dieses durch Ausschwemmung, Fixierung und Paraffinieren um ca. 40%. Durch das Schneiden in dünne Schichten geht zusätzlich der räumliche Zusammenhang verloren und es kommt durch mittige Platzierung der Schnitte auf dem Objektträger zum so genannten Bananeneffekt (siehe Abb. 1.8), d.h. Krümmungen und ähnliches, die orthogonal zur Schneiderichtung verlaufen, können nicht mehr rekonstruiert werden. Damit ist es möglich, dass Objekte un-

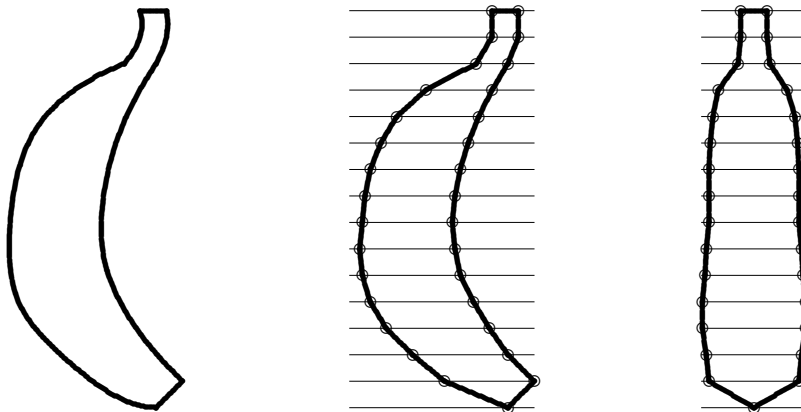


Abbildung 1.8: Bananeneffekt; nach Zerschneiden der Banane (*links*) in dünne Scheiben (schematisch, *mitte*) kann durch Registrierung die ursprüngliche Geometrie nicht mehr hergestellt werden (*rechts*).

terschiedlicher Geometrie die gleiche Rekonstruktion besitzen. Um die ursprüngliche Geometrie wieder herzustellen, muss zusätzlich zur Rekonstruktion eine 3D-Registrierung mit einer Geometrie erhaltenden Modalität (z.B. MRT) durchgeführt werden. Zusätzlich zur Geometrieänderung kann sich die Topologie durch Risse, Gewebsablösungen, Falten u.ä. verändern. Der Verlust der ursprünglichen Topologie kann mit automatisierten Verfahren nicht generell wieder hergestellt werden. Durch manuelle Vorkorrekturen, wie replazierung abgelöster Gewebeteile, partieller Rotationen einzelner Objekte einer Schicht, u.a. kann auch die Topologie sinnvoll rekonstruiert werden.

1.2.2 Artefakte durch die Bildherstellung

Die einzelnen Schritte des Herstellungsprozesses erzeugen nicht nur, wie oben bereits erwähnt, einen Verlust der Geometrieinformation, sondern bewirken auch verschiedenste Deformationen. Die Schrumpfung des Hirns ist ein Prozess, der durch eine Skalierung approximativ eingefangen, bzw. rückgängig gemacht werden kann. Das Zerschneiden in Schichten erzeugt allerdings Verzerrungen, die nicht mehr affin-linear zu korrigieren sind. Mögliches Einreißen des Gewebes bei der manuel-

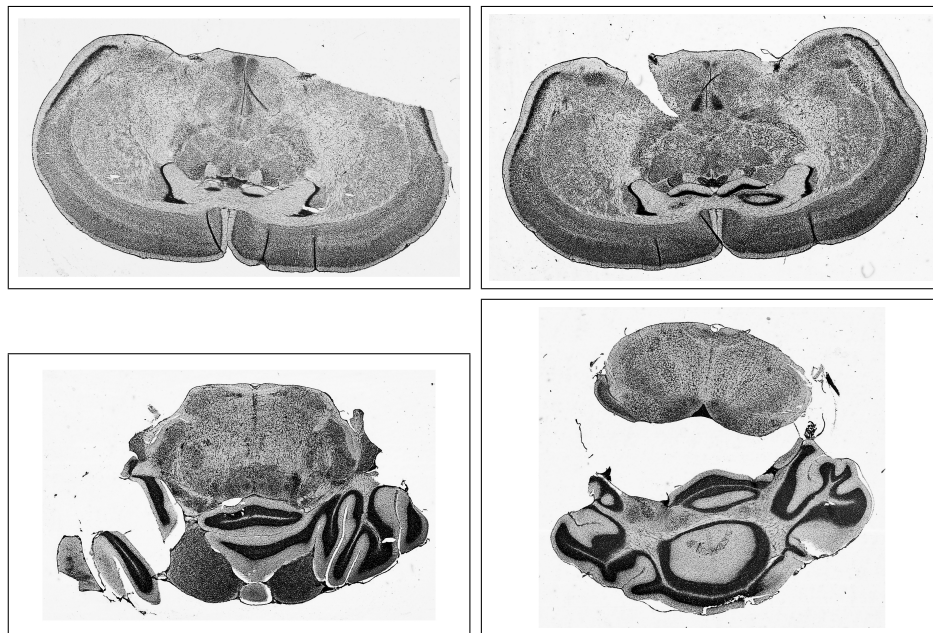


Abbildung 1.9: Durch Einreißen des Gewebes entstehen zusätzliche nichtlineare Deformationen und unnatürliche Kanten (rechte Spalte). Gewebsverlust und Ablösungen erzeugen Korrespondenzlücken (linke Spalte).

len Verarbeitung (siehe Abb. 1.9) erzeugt lokal starke, nichtlineare Deformationen, die von der Registrierung nicht wieder automatisch korrigiert werden können. Daher soll die Registrierung von diesen Effekten möglichst nicht beeinflusst werden. Für eine gute dreidimensionale Rekonstruktion ist es daher notwendig, nicht nur affin-linear, sondern auch frei bzw. elastisch zu registrieren, sondern auch bei besonderen Begebenheiten manuell zu korrigieren, um damit dem anschließenden automatischen Algorithmus zu helfen. Durch Einrisse des Gewebes, die während der Herstellung der Schnitte entstehen (siehe Abb. 1.9), entstehen künstliche Kanten in den Bildern und stören damit die Korrespondenz zwischen den einzelnen Schichten. Reißen einzelne Teile des Objekts sogar ab, wird ein ursprünglich zusammenhängender Teil in zwei voneinander getrennte Bereiche aufgeteilt, die nicht notwendigerweise vom Algorithmus als zusammengehörig erkannt werden können. Im extremen Fall reißt das Gewebe nicht nur ein, sondern ab und geht verloren. Die verlorenen Teile des Objekts können damit nicht mehr zu einer guten Registrierung beitragen.

Eine weitere Schwierigkeit bei histologischen Bildern ist die Entstehung von Falten bzw. Überlappungen des Objekts während der Herstellung. Durch Falten kommt es zu Überlagerungen von unterschiedlichen Objektteilen in den Bildern. Es entstehen dadurch neue Strukturen und eine besondere Art von Gewebeverlust (das Gewebe ist nicht verloren, sondern nur durch Überlagerung von zwei Teilen für die Registrierung unbrauchbar geworden). Überlagerungen können durch eine Registrierung nicht „entfaltet“, also die Deformation nicht mehr zurück gerechnet werden, da nicht mehr eindeutig differenzierbar ist, welche Bildinformationen durch Kombination entstanden sind und welche echte Strukturen sind. Neben evtl. neuartigen Strukturen ist physikalisch gesehen die Schichtdicke wesentlich größer, wodurch zusätzliche Kanten entstehen und das Färbeverhalten anders ist, als in der direkten Umgebung. Bereiche mit Falten bleiben in den Nachbarschichten ohne Korrespondenz, was zu ungewünschtem Schrumpfen oder Ausdehnen führen kann.

Diese herstellungsbedingten Artefakte sind weder zu vermeiden, noch zu korrigieren. Damit es in Bereichen von Gewebsverlust, Ablösungen, Falten und Überlappungen nicht zu unnatürlichen Deformationen kommt, muss sich ein Algorithmus zur Registrierung von histologischen Serienschnitten diesbezüglich robust verhalten, d.h. Artefakte nicht durch größere, unnatürliche Deformationen von angrenzenden Gewebeteilen ausgleichen, sondern diese Bereiche möglichst unverändert lassen.

1.2.3 Inhomogene Färbung

Der Färbeprozess trägt zwar einerseits zur Kontrastanreicherung und damit zur Verbesserung der Bildqualität bei, allerdings ist nicht gewährleistet, dass die Färbung homogen abläuft. Die Menge an Färbemittel, die Einwirkzeit und die gründlichen Auswaschung des Färbemittels haben einen entscheidenden Einfluss auf die Bildqualität. So kommt es zu Unterschieden der Färbeintensität von Schicht zu Schicht, da jede Schicht einzeln behandelt wird. Darüber hinaus kommt es durch

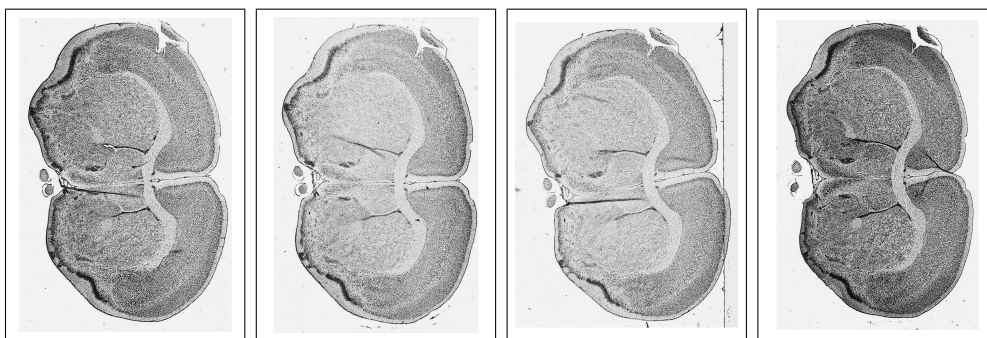


Abbildung 1.10: Unterschiedliche Färbemittelmengen und Einwirkzeiten erzeugen unterschiedlich starke Färbungen bei aufeinander folgenden Schichten.

unterschiedliche Diffusionsprozesse zu teilweise starken Unterschieden der Intensitätsverteilung innerhalb einer Schicht. Um direkte Vergleiche und damit auch eine

Registrierung gut durchführen zu können, muss eine möglichst gute Homogenisierung stattfinden. Dabei ist einerseits die Homogenisierung der einzelnen Schicht, als auch die Homogenisierung der Schichten untereinander zu beachten. Da sich Homogenisierung und intensitätsbasierte Registrierung gegenseitig bedingen, ist eine simultane Homogenisierung und Registrierung sinnvoll.

1.2.4 Extreme Datengröße

Bisher wurden ausschließlich Artefakte der digitalisierten Bilder betrachtet, die eine Rekonstruktion erschweren. Die größte Schwierigkeit, die selbst bei perfekt aufbereiteten Daten immer bestehen bleibt, ist die enorme Datenmenge, die durch die Digitalisierung der histologischen Serienschritte entsteht. Bei einer Auflösung von $5.04\mu\text{m} \times 5.04\mu\text{m} \times 20\mu\text{m}$ entstehen etwa 600 Bilder mit einer Auflösung von bis zu 2650×1392 Pixel. Damit beinhaltet der gesamte Datensatz ca. 2 213 280 000 Pixel. Bei einer Kodierung mit *8bit* bzw. *1byte* in Graustufen ergibt dies ca. *2.1GByte* an Bilddaten. Diese Datenmenge muss dabei einerseits auf der Festplatte Platz finden und andererseits effizient von dort geladen und im Registrierungsalgorithmus verarbeitet werden. Selbst mit modernen, gut ausgerüsteten PCs lässt sich diese Aufgabe nicht so einfach realisieren. Daher ist aus Datenverarbeitungssicht interessanter Bereich dieser Arbeit die Bewältigung dieser Datenmenge.

Führt man sich die oben genannten Schwierigkeiten bei der Verarbeitung von Bildserien vor Augen, wird sofort ersichtlich, dass für die robuste Verarbeitung von Bildserien von histologischen Serienschritten nicht notwendigerweise der beste Algorithmus aus den klassischen Registrierungsbereichen der beste ist, sondern ein Algorithmus, der am besten mit oben genannten Schwierigkeiten umgehen kann. Vor allem muss der Algorithmus effizient im Umgang mit den Daten sein, damit die großen Datenmengen überhaupt verarbeitet werden können. In den nachfolgenden Kapiteln, insbesondere in Kapitel 2 wird daher ein Algorithmus vorgestellt, der die oben genannten Schwierigkeiten, vor allem 'Bananeneffekt', 'Inhomogenitäten' und 'extreme Datengröße' meistert.

2 Registrierungsverfahren für Bilder von Serienschnitten

In diesem Kapitel möchten wir zuerst einmal die Begriffe *Bildregistrierung* und *Registrierung* klären, um dann zwei konkrete Varianten der Registrierung vorzustellen, die sich für den Einsatz an Serienschnitten besonders qualifiziert haben. Damit eindeutig festgelegt ist, was wir unter einem Bild zur Registrierung verstehen, werden die Begriffe *Bild*, *Deformationsfeld* und *deformiertes Bild* zuerst definiert.

Unter *Bildregistrierung*, oder auch *Registrierung* von Bildern verstehen wir den Prozess der algorithmischen Anpassung von Bildern durch eine vorgegebene Art von *Transformation*. Die verschiedenen Methoden zur Registrierung unterscheiden sich weitestgehend durch die Art der Transformation, die benutzt wird, die Anzahl der *Freiheitsgrade* der Transformation, die Art und Weise, wie Bilder miteinander verglichen werden und wie ein optimales Ergebnis berechnet werden kann. In der Literatur werden die Registrierungsalgorithmen in verschiedene Klassen unterteilt und wir verweisen daher auf die einschlägigen Arbeiten von Brown [Bro92] und Maintz & Viergever [MV98]. Tiefergehende Ausführungen und Varianten der Registrierung werden in Arbeiten von Bro-Nielsen, Christensen, Modersitzki oder Woods et al. [BN96, Bro92, Chr94, Mod04, WGH⁺98, WGW⁺98] gegeben. In dieser Arbeit hier möchten wir daher nur die speziellen Registrierungsmethoden besprechen, die für den Einsatz an Bildern von Serienschnitten besonders geeignet sind.

In Kapitel 1 haben wir gesehen, dass während der Herstellung der Serienschnitte Deformationen entstehen, die heraus gerechnet werden müssen, um eine dreidimensionale Rekonstruktion der Bilddaten zu ermöglichen. In der Praxis werden hierzu im wesentlichen *globale* Registrierungsmethoden verwendet, die Transformationen nutzen, die gleichförmig auf das ganze Bild und damit global, angewendet werden. Am verbreitetsten ist die *rigide* Registrierung als spezielle Variante der *affinen* bzw. *linearen* Registrierung, bei der als Transformation ausschließlich Drehungen und Verschiebungen genutzt werden. Der Hauptvorteil von linearen Methoden ist, dass die Anzahl der Freiheitsgrade ausschließlich von der Dimension der Bilder, nicht aber von deren Größe abhängt. Als zweite Variante der Registrierung, die wir im weiteren vorstellen werden, ist die *elastische* Registrierung zu nennen, die im Gegensatz zur linearen Registrierung nicht globale, sondern *lokale* Transformationen nutzt. Elastische Transformationen sind physikalisch motiviert und entsprechen damit tendenziell eher den natürlichen Deformationsprozessen von elastischem Gewebe als es lineare Transformationen tun.

2.1 Bild und Deformationsfeld

Bevor wir erklären, was wir unter der Registrierung von Bildern verstehen, wollen wir zunächst einmal genau festlegen, was in diesem Sinne ein Bild ist. Darüber hinaus führen wir den Begriff des Deformationsfeldes ein, um schließlich zu deformierten Bildern zu gelangen.

Mathematisch gesehen ist ein Bild eine Abbildung bzw. Funktion, die einer Menge von Punkten im Raum einen Intensitätswert (Farbwert, Grauwert, Strahlungswert) zuordnet. Abbildungen können bestimmte Eigenschaften wie Differenzierbarkeit und Integrierbarkeit besitzen, wodurch mathematische Methoden zur Analyse oder Manipulation von Bildern herangezogen werden.

Definition 2.1.1 (Bild)

Seien $d \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$ eine offene, beschränkte, einfach zusammenhängende Menge mit einem, bis auf endlich viele Punkte, glatten Rand Γ und $\mathbb{G} \subset \mathbb{N}$ ein abgeschlossenes Intervall auf \mathbb{N} . Ist weiterhin die Abbildung

$$B : \Omega \rightarrow \mathbb{G}$$

$$\mathbf{x} \mapsto B(\mathbf{x})$$

zweimal stetig differenzierbar und quadratisch integrierbar über Ω , $B \in C^2(\Omega)$, dann heisst B *Bild* mit Dimension d und Grau- bzw. Intensitätswertbereich \mathbb{G} .

Bei dieser Definition fällt auf, dass aus mathematischer Sicht Bilder beliebiger Dimension sein können. Klassisch gesehen werden die Dimensionen $d = 2$ und $d = 3$ als Bilder bzw. Volumen gesehen. Für die Anschauung ist dies von Vorteil. Die in dieser Arbeit vorgestellten Methoden sind allerdings auf beliebige Dimensionen anwend- bzw. erweiterbar. Für unsere Anwendung ist es vollkommen ausreichend, dass ausschließlich auf ganzzahlige Werte (Grauwerte) abgebildet wird. Bei anderen Anwendungen kann dies nicht der Fall sein, wodurch beispielsweise auf einen Vektor (z.B. RGB) oder Tensor (hier können beispielsweise Richtungsinformationen abgelegt werden) abgebildet wird. Die hier vorgestellte Methode lässt sich für viele dieser Anwendungen erweitern, was hier aber nicht weiter besprochen wird.

Im Zusammenhang der Bildregistrierung werden Bilder gerne als *Referenzbild* R und als *Templatebild* T bezeichnet. Alternativ sind die Begriffe *Fixed Image* für das Referenzbild und *Moving Image* für das Templatebild gebräuchlich. Das Referenzbild gilt dann dem Namen entsprechend als unveränderbar, während das Templatebild durch eine Transformation verändert werden kann. Im Falle der elastischen Registrierung wird eine Transformation durch ein *Deformationsfeld* repräsentiert.

Definition 2.1.2 (Deformationsfeld, Deformation)

Sei $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$ eine zweimal stetig differenzierbare Abbildung

$$\mathbf{u} = (u_1, \dots, u_d)^\top, \quad u_i : \Omega \rightarrow \mathbb{R} \quad \text{für } i = 1, \dots, d$$

$$\mathbf{x} = (x_1, \dots, x_d)^\top \mapsto (u_1(\mathbf{x}), \dots, u_d(\mathbf{x}))^\top$$

heißt *Deformationsfeld* oder kurz *Deformation*.

Mit Hilfe eines Deformationsfeldes kann ein Bild, bzw. ein Gebiet transformiert werden.

Definition 2.1.3 (Deformiertes Bild, deformiertes Gebiet)

Sei $T : \Omega \rightarrow \mathbb{R}$ ein Bild. Wendet man ein Deformationsfeld \mathbf{u} auf ein Bild T an,

$$T(\mathbf{x}) \mapsto T_{\mathbf{u}}(\mathbf{x}) := T(\mathbf{x} - \mathbf{u}(\mathbf{x})),$$

so spricht man von einem *deformierten Bild*. Wendet man das Deformationsfeld auf alle Elemente eines Gebiets Ω an,

$$\Omega \mapsto \mathbf{u}(\Omega) = \Omega_{\mathbf{u}} := \{\mathbf{u}(\mathbf{x}); \mathbf{x} \in \Omega\},$$

so spricht man von einem *deformierten Gebiet* $\Omega_{\mathbf{u}}$.

In Abb. 2.1 ist exemplarisch die Anwendung einer Deformation auf ein Gebiet

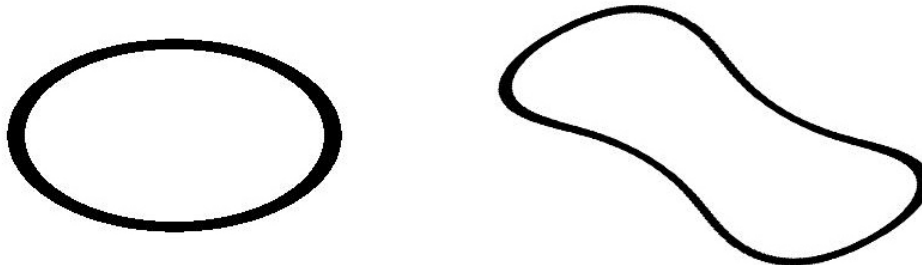


Abbildung 2.1: links: Gebiet Ω , rechts: deformiertes Gebiet $\Omega_{\mathbf{u}}$.

dargestellt. Links befindet sich das ursprüngliche, rechts das deformierte Gebiet. Dargestellt ist Γ , der Rand von Ω .

2.2 Affin-lineare Registrierung

Wir beginnen mit einer grundlegenden Art an Transformation und damit mit einer Registrierungsmethode, die besonders gut aufzufassen und nachzuvollziehen ist. Die *affin-lineare*, kurz lineare Registrierung zeichnet sich durch eine geringe Anzahl

an Parametern bzw. Freiheitsgraden für die Transformation aus. Sie ist unabhängig von der tatsächlichen Bildgröße, sondern hängt ausschließlich von der Dimension der Bilder ab.

Zu den linearen Transformationen zählen Translation, Rotation, Skalierung, Scherung und alle beliebigen Kombinationen davon. Beschränkt sich eine Transformation auf Translation und Rotation, so spricht man von einer *rigiden* Transformation, ansonsten von einer affinen Transformation.

Lineare Transformationen wirken stets *global*. Das bedeutet, dass ein und dieselbe Transformation auf alle Bildpunkte angewendet wird.

Bei zweidimensionalen Bildern lässt sich die Transformation durch sechs Parameter (zwei zur Translation, einer zur Rotation, zwei zur Skalierung und einer zur Scherung), bei dreidimensionalen Bildern durch zwölf Parameter (jeweils drei Parameter für Translation, Rotation, Skalierung und Scherung) darstellen.

Um die Wirkungsweise zu verdeutlichen, betrachten wir Abbildung 2.2, in der die elementare Transformationen der Reihe nach exemplarisch dargestellt sind. Um die Veränderung der Transformation möglichst gut darzustellen, haben wir

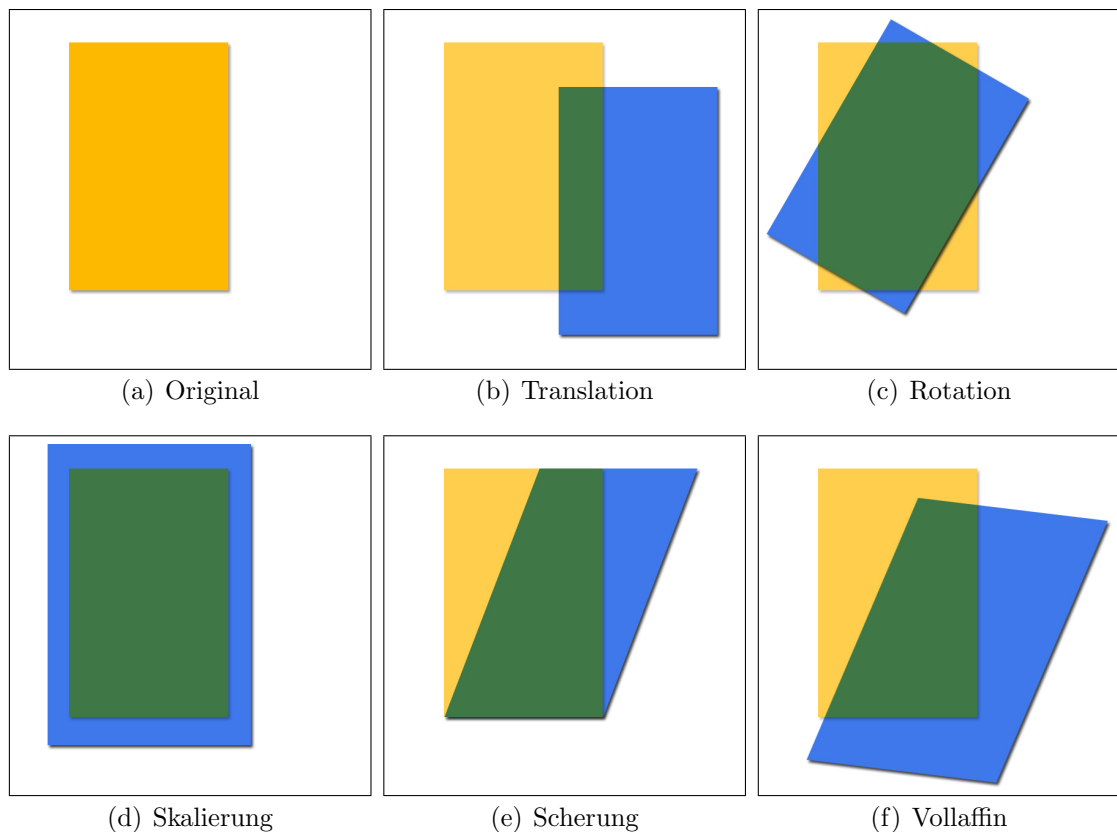


Abbildung 2.2: Affine Transformation: (a) Original, (b) Translation in x_1 und x_2 , (c) Rotation um 30° , (d) Skalierung in x_1 und x_2 , (e) Scherung, (f) Kombination aus Translation, Rotation, Skalierung und Scherung.

das Originalbild farblich gelb gekennzeichnet und mit dem Transformationsergeb-

nis in blau überlagert. So sehen wir in der oberen Zeile einmal das Originalbild (a) mit einem Rechteck als Objekt. Daneben befinden sich angewandte Transformationen mit reinem Translations- (b) und Rotationsanteil (c). Die Rotation um 30° wurde um den Mittelpunkt des Rechtecks durchgeführt. In der zweiten Zeile sehen wir eine Skalierung (d), eine Scherung (e) und schließlich eine Kombination aller Grundtransformationen.

Da es sich bei der affinen Transformation um eine lineare Abbildung handelt, kann diese durch eine Matrix-Vektor-Operation $A\mathbf{x} + \mathbf{t}$ ausgedrückt werden mit einer Transformationsmatrix $A \in \mathbb{R}^{d \times d}$, d ist die Dimension, und einem Translationsvektor $\mathbf{t} \in \mathbb{R}^d$. Bei der linearen Transformation eines ganzen Bildes werden alle Punkte $\mathbf{x} := (x_1, \dots, x_d) \in \mathbb{R}^d$ des Bildes auf die gleiche Weise transformiert. Handelt es sich bei der Transformationsmatrix A um eine reine Rotationsmatrix, so wird die Transformation als rigide bezeichnet. Generell stellt die Transformationsmatrix das Produkt aus Rotations- R , Skalierungs- S und Scherungsmatrix Q dar,

$$A = R \cdot Q \cdot S.$$

Im dreidimensionalen Fall ist die Rotationsmatrix das Produkt aus den drei Rotationsmatrizen für jede Achse, d.h.

$$\begin{aligned} R &= R_{x_3} \cdot R_{x_1} \cdot R_{x_2} \\ &= \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \end{aligned}$$

mit den Rotationswinkeln α , β und γ . Die Reihenfolge der Matrizen ist hierbei genau zu beachten. Die Reihenfolge x_3 , x_1 , x_2 wird in der Literatur als Euler-Darstellung bezeichnet. Genauer findet sich in [Fol95]. Der Vollständigkeit halber wollen wir nun auch noch die Scherungsmatrix, die Skalierungsmatrix und den Translationsvektor vorstellen:

$$Q = \begin{pmatrix} 1 & q_{x_1} & q_{x_1} \\ q_{x_2} & 1 & q_{x_2} \\ q_{x_3} & q_{x_3} & 1 \end{pmatrix}, \quad S = \begin{pmatrix} s_{x_1} & 0 & 0 \\ 0 & s_{x_2} & 0 \\ 0 & 0 & s_{x_3} \end{pmatrix} \quad \text{und} \quad \mathbf{t} = \begin{pmatrix} t_{x_1} \\ t_{x_2} \\ t_{x_3} \end{pmatrix}.$$

Die Subindizes geben hier die Richtung an, für die die Komponente gilt, z.B. t_{x_1} für Translation in x_1 -Richtung.

Die dreidimensionale Registrierung kann zur Registrierung auf eine geometriehaltenden Modalität genutzt werden, um dadurch Veränderungen in Geometrie und Topologie rückgängig zu machen. Für die Registrierung der Bilder der einzelnen Schichten der Serie reicht aus algorithmischer Sicht eine zweidimensionale Registrierung. Die Deformationen innerhalb der Serie liegen ausschließlich innerhalb jeder Schicht und nicht schichtübergreifend. Wendet man sukzessive eine zweidimensionale Registrierung von benachbarten Schichten an, so ergibt sich eine dreidimensionale Rekonstruktion.

Im zweidimensionalen Fall sehen die Matrizen wie folgt aus:

$$R = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}, Q = \begin{pmatrix} 1 & q_{x_1} \\ q_{x_2} & 1 \end{pmatrix}, S = \begin{pmatrix} s_{x_1} & 0 \\ 0 & s_{x_2} \end{pmatrix}, \mathbf{t} = \begin{pmatrix} t_{x_1} \\ t_{x_2} \end{pmatrix}.$$

Unter einer affin-linearen Registrierung versteht man die Optimierung der o.g. Parameter bezüglich eines Ähnlichkeitsmaßes bzw. einer (Halb-)Norm. Im Zusammenhang der Registrierung von Bildern von Serienschritten ist die lineare Registrierung (meist sogar nur rigide) immer noch state-of-the-art, da die gesamte Registrierung auch bei sehr großen Einzelbildgrößen effizient und relativ schnell geht. Bei Untersuchungen, die sich auf Strukturen beziehen, die in etwa die Größe von wenigen Pixeln haben, ist die affin-lineare Registrierung allerdings nicht geeignet. In solchen Fällen dient sie aber als gute Vorverarbeitung, um die meist großen globalen Deformationen, die durch den Herstellungsprozess, vor allem der Digitalisierung, entstanden sind, einzufangen.

2.3 Elastische Registrierung

Lineare Transformationen wirken gleichförmig auf das ganze Bild. Bei der Herstellung der Serienschritte wird allerdings jede Schicht einzeln geschnitten und manuell weiterverarbeitet. Dadurch entstehen für jede einzelne Schicht individuelle Deformationen, die nicht global korrigiert werden können (vergl. dazu Kapitel 1 und Abb. 2.3). Bei einer problemorientierten Registrierung muss dieser Faktor berücksichtigt werden. Körpergewebe, wie in unserem Fall ein Gehirn, unterliegt den physikalischen Gesetzen der Elastizität. Auch durch den Prozess des Paraffinierens verändert sich daran nichts. Im Gegenteil, durch das Paraffinieren wird das elastische Verhalten für das ganze Objekt gleich, während vorher durch unterschiedliche Zellverteilungen und Zelldichten die elastischen Eigenschaften unterschiedlich waren. Daher liegt es nahe, die Elastizität zu modellieren. Bei einer elastischen Registrierung dürfen nur Deformationen entstehen, die aus physikalischer Sicht dem Elastizitätsprinzip gehorchen, um dadurch die Deformation zu erlangen, die dem zugrunde liegenden Material am gerechtesten wird und den bei der Herstellung entstandenen Deformationen am besten entspricht. Die über die elastischen Deformationen hinaus entstandenen Deformationen durch Risse oder Falten, sind, wie in Kapitel 1 beschrieben, per se nicht rekonstruier- bzw. korrigierbar und werden daher nicht explizit modelliert.

Physikalisch wird das Verhalten elastischen Materials unter Einwirkung einer Kraft von aussen durch die Navier-Lamé-Gleichungen

$$\mu \Delta \mathbf{u} + (\lambda + \mu) \nabla \operatorname{div} \mathbf{u} = \mathbf{f} \quad (2.1)$$

beschrieben. Auf der linken Seite von Gleichung (2.1) steht die innere Kraft, die sich aus der Deformation \mathbf{u} , den Materialkonstanten μ und λ , den Differentialoperatoren Δ für die zweite Ableitung und ∇ für die erste Ableitung und der Divergenz

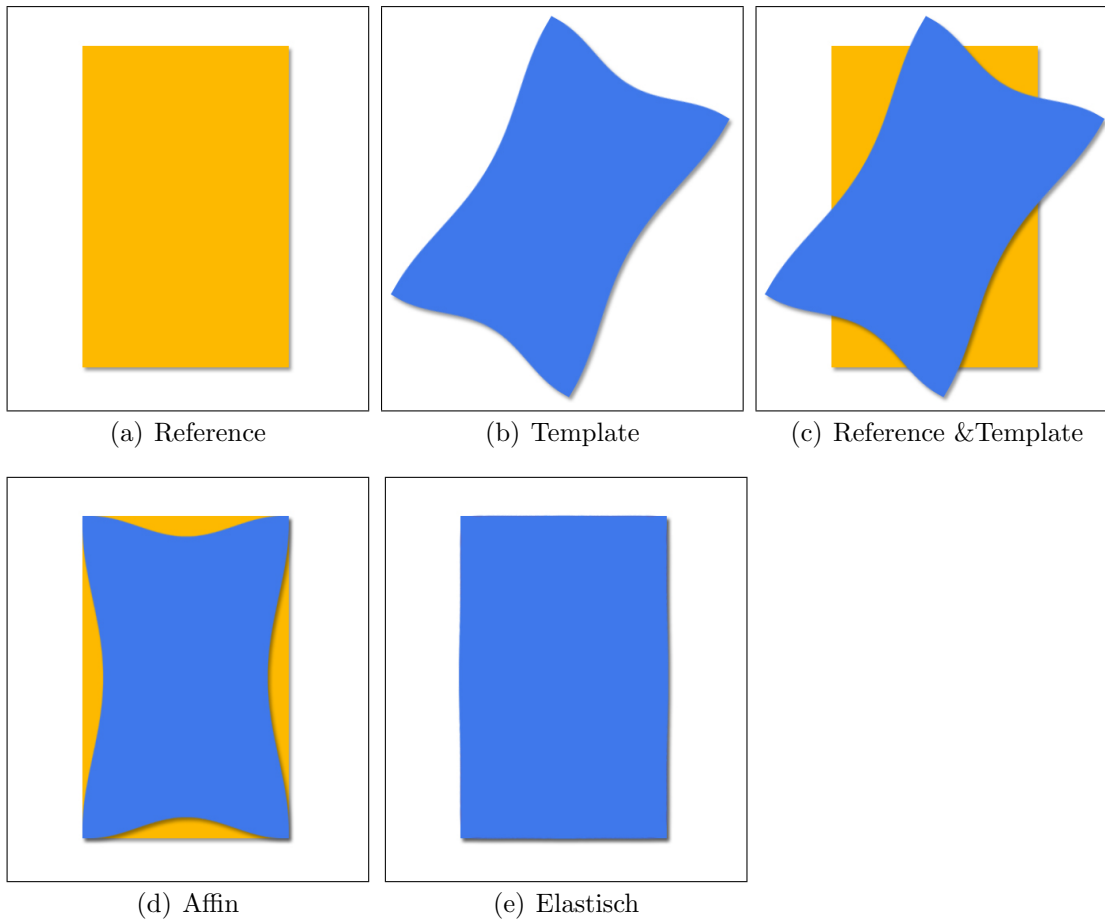


Abbildung 2.3: Schematische Darstellung der individuellen Deformationen durch den Schneideprozess bei histologischen Serienschritten. (a) ursprüngliche Geometrie, (b) individuelle Deformationen nach Schneiden und Scannen, (c) direkter Vergleich. Eine affine Transformation (d) trifft bis auf kleine Unterschiede die ursprüngliche Geometrie. Eine elastische Deformation (e) kann ein perfektes Ergebnis erzielen.

div zusammen setzt. Auf der rechten Seite der Gleichung steht die von außen wirkende bzw. eingewirkte Kraft \mathbf{f} , die auch äußere Kraft genannt wird. Durch die elastischen Eigenschaften des Materials entstehen innere Kräfte, die den äußeren Kräften entgegen wirken. Kommt es zum Gleichgewicht, ist die Deformation erreicht, die durch die Kraft von außen bewirkt wurde. Lässt diese nach, so geht auch die Wirkung (Deformation) wieder zurück.

Die individuellen Materialeigenschaften werden durch die Lamé-Konstanten μ (Schermodul) und λ (Kompressionsmodul) eindeutig beschrieben und geben darüber an, ob das Material eher flexibel wie Gummi ($\mu \approx 3 \cdot 10^{-4}$, $\lambda \approx 3 \cdot 10^{-3}$) oder starr wie Stahl ($\mu = 80$, $\lambda = 78$) ist. Der Quotient

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu}, \quad E > 0$$

wird Elastizitätsmodul genannt (engl. **Young's modulus**) und gibt an, wie viel Widerstand ein Material dem äußeren Druck entgegen bringt. Je kleiner dieser Wert ist, desto flexibler ist das Material (Gummi 10^{-2} , Stahl 200). Ein weiterer Quotient der **Lamé-Konstanten**,

$$\nu = \frac{\lambda}{2(\lambda + \mu)}, \quad 0 \leq \nu \leq 0.5,$$

ist die **Poisson-Zahl**, die das Verhältnis von Längsdehnung zur Querkontraktion angibt (Gummi 0.5, Stahl 0.28). Da Gummi eine **Poisson-Zahl** von 0.5 besitzt, bedeutet dies, dass die Querkontraktion so stark ist, dass keine Volumenänderung entsteht. Diese Eigenschaft lässt leider nicht durch einfache Angabe der Konstanten in die Bildverarbeitung übertragen, sondern muss durch zusätzliche Bedingungen gewährleistet werden.

Wie oben bereits erwähnt, ist das elastische Verhalten einer einzelnen Schicht durch die Einbettung in Paraffin überall gleich. Die elastischen Eigenschaften des Paraffins überwiegen die des ursprünglichen Gewebes. Daher ist eine globale Wahl der Materialkonstanten möglich. In anderen Fällen kann es sinnvoll sein, die Materialkonstanten lokal zu variieren, um so unterschiedliches elastisches Verhalten in das Verfahren mit einzubringen. Selbst bei globaler Wahl der Materialkonstanten ist es nicht leicht, diese passend für die Registrierung zu wählen, so dass der Registrierungsverlauf erwartungskonform ist. Eine direkte Übertragung der physikalischen Materialkonstanten führt nicht zum erwarteten Ergebnis (werden die Materialkonstanten beispielsweise auf Stahl gesetzt, so muss das Registrierungsergebnis nicht notwendigerweise ein Deformationsfeld liefern, das einem stählerne Objekt entspricht). Dies liegt an der individuellen Repräsentierung eines Bildes durch Intensitätswerte. Je nach Bildaufnahmetechnik (Videomikroskopie, Rasterelektronenmikroskopie, Röntgenstrahlung, Computertomographie, Magnetresonanztomographie, Ultraschall,...) variieren die Intensitätswerte sehr stark. Selbst innerhalb einer Aufnahmetechnik kann durch unterschiedlich intensive Färbungen und Belichtungszeiten die Repräsentierung so stark variieren, dass es keine Standardlösung hierfür gibt.

Die Wahl der Materialkonstanten hat eine direkte Auswirkungen auf die Deformationen die entstehen, sobald Kräfte von außen einwirken. Im Gegensatz zur inneren Kraft ist die äußere Kraft a priori nicht bekannt und muss daher geschätzt werden. Diese Schätzung wird auf Basis der für die Registrierung zugrunde liegenden Norm, bzw. dem Ähnlichkeitsmaß vollzogen und ist damit abhängig von der konkreten Repräsentierung durch Intensitätswerte. Dies hat zur Folge, dass die Materialkonstanten für jedes neue Datenpaar angepasst werden müssen, um das gewünschte elastische Verhalten zu erreichen. Um aus praktischen Gründen nicht bei jeder Registrierung die Materialkonstanten anpassen zu müssen, kann durch verschiedene Normierungen ein optimaler Satz an Materialkonstanten und Normierungsfaktoren für jede Modalität gefunden werden. Die Normierung kann einerseits bei der Grauwertverteilung, andererseits bei der Berechnung der Kraft stattfinden. Durch diese Normierung lassen sich dann ohne Veränderung der Konstanten er-

wartungskonforme Registrierungen auf Bildern gleichen Typs durchführen.

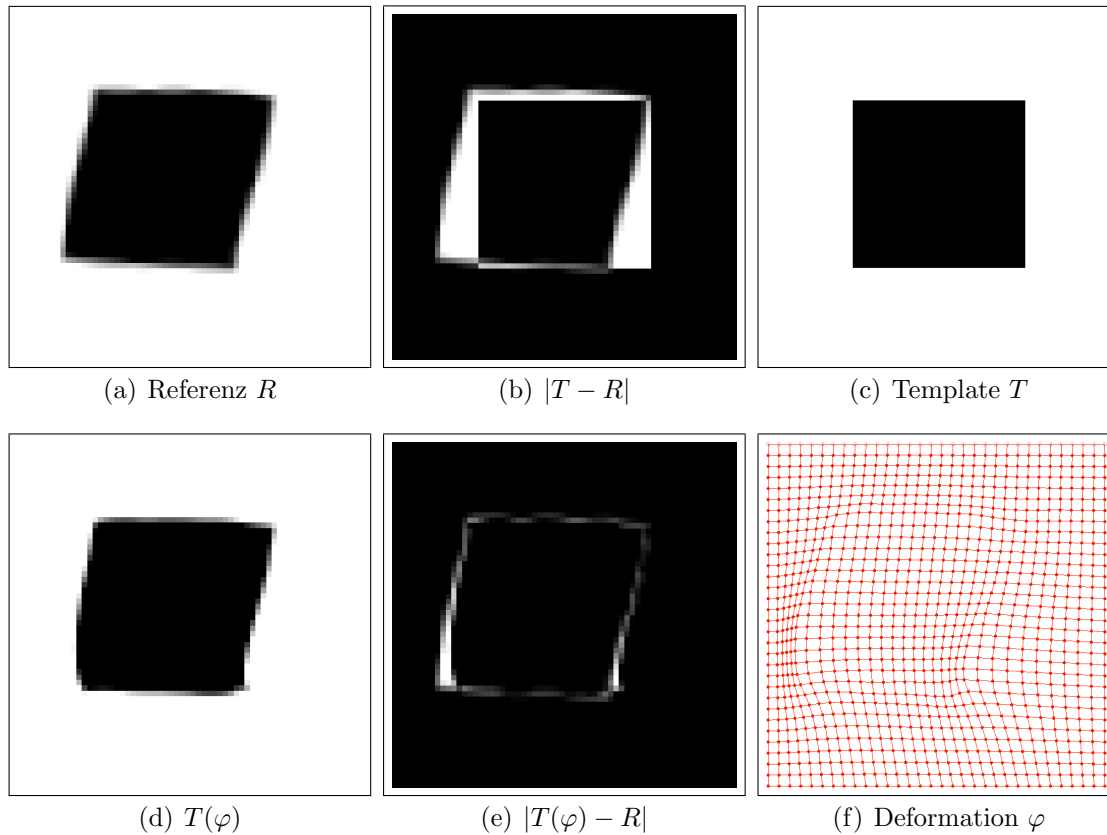


Abbildung 2.4: Elastische Registrierung einer globalen Transformation aus Rotation und Translation und einer lokalen Deformation (Sinusschwingung) ohne lineare Vorregistrierung. Das Deformationsfeld in der Gitterdarstellung zeigt, dass große Kräfte gewirkt haben. Die Registrierung endete, da innere und äußere Kräfte sich aufgehoben haben. Die restliche Deformation kann damit nicht mehr erreicht werden. Durch Anwendung einer linearen Vorregistrierung kann ein perfektes Ergebnis erzielt werden, da nur sehr geringe lokale Deformationen enthalten sind.

Der scheinbare Vorteil der elastischen gegenüber der linearen Registrierung, lokale Veränderungen durchführen zu können, lässt über einen großen Nachteil schnell hinweg sehen. Große und globale Deformationen (beispielsweise Translationen über große Teile des Bildes hinweg) werden nur sehr schlecht und mit großem algorithmischen Aufwand eingefangen. Um große Deformationen zu erzeugen, müssen sehr große Kräfte von außen wirken. Die Anwendung großer äußerer Kräfte erzeugt aufgrund des elastischen Potentials große innere Kräfte. Ein Gleichgewicht der Kräfte kann daher schon vor Erreichen der erforderlichen Gesamtdeformation eintreten, wodurch das Ergebnis nicht zufrieden stellend wird (siehe Abb. 2.4). Des weiteren können durch die Anwendung von sehr großen Kräften Windungen u.ä. entstehen, die nicht zur gewünschten Gesamtdeformation gehören.

Zur besseren Vorstellung hilft es, sich das zu registrierende Bild auf einer Gum-

mihaut aufgetragen und in einen Rahmen eingepasst vorzustellen. Zur Registrierung müssten dann z.B. mit Hilfe von Nadeln, einzelne Bereiche gezogen werden, bis die gewünschte Deformation erreicht ist. Nun wird sehr schnell klar, wie groß der Aufwand ist, eine einfache Translation auszuführen, da sich mit zunehmender Zugkraft der Widerstand des Gummis erhöht. Würde man aber die Gummihaut vom Rahmen lösen, geeignet verschieben und anschließend neu fest spannen, dann wären keine äußeren Kräfte notwendig, um die Position zu halten und es entstehen auch keinerlei Verzerrungen bzw. Windungen des Bildes. Aus mathematischer Sicht bedeutet dies, dass affin-lineare Transformationen bei der elastischen Registrierung nicht im Kern liegen und damit haben sie einen starken Einfluss auf die inneren Spannungen. Aufgrund großer innerer Spannungen bei affinen Transformationen ist es besonders wichtig, dass vor einer elastischen Registrierung eine lineare Vorregistrierung stattfindet, damit alle globalen Transformationsanteile verschwunden sind, bevor die elastische Registrierung startet.

Die genaue Umsetzung, alle Voraussetzungen und algorithmischen Details zur elastischen Registrierung finden sich in den Kapiteln 4, 5 und 6.

3 Homogenisierung von Bildern

In Kapitel 1 wurden verschiedene Artefakte in Bildern vorgestellt, die die Registrierung von Bildern erschweren. In diesem Kapitel wollen wir nun die Homogenisierung, die Eliminierung der Inhomogenitäten, genauer unter die Lupe nehmen. Zuerst wird ein kurzer Überblick über mögliche Inhomogenitäten bei Bildern gegeben. Nachfolgend wird eine Methode zur Homogenisierung vorgestellt, die in der medizinischen Histologie sehr verbreitet ist. Schließlich wird ein Verfahren namens *Total Variation* vorgestellt, mit dem iterativ, unter Erhaltung von Kanten, Bilder homogenisiert werden können.

3.1 Inhomogenitäten in Serienschnitten

Inhomogenitäten in der Grauwertrepräsentierung von Objekten innerhalb eines Bildes treten in den verschiedensten Bereichen der Bildgebung auf. Die Inhomogenitäten lassen sich auf unterschiedliche Ausleuchtung, Belichtung, Bestrahlung, Färbung und ähnliches zurückführen. Inhomogenitäten entstehen in erster Linie durch technische oder systembedingte Gegebenheiten der bildgebenden Geräte. In der Computertomographie (CT) nimmt die Intensitäten von der Bildmitte nach außen hin radial ab. Dies liegt an der radialen Signalmessung, bei der die häufigsten Überschneidungen von gemessenen Signalen im Bildzentrum liegen. Wegen dieser Abschwächung werden identische Objekte im CT je nach Ort im Bild (näher oder weiter vom Zentrum des Bildes entfernt) mit unterschiedlichen Grauwerten dargestellt. Ähnliche Phänomene gibt es bei jeder bildgebenden Modalität.

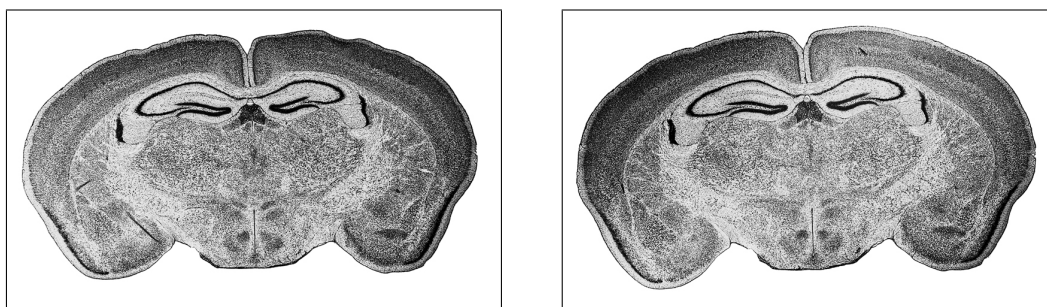


Abbildung 3.1: Unterschiedliche Färbungen innerhalb eines Schnittes und relativ zum Nachbarn. *links:* Schnitt 229, starke Färbung rechts oben, schwache Färbung links unten. *rechts:* Schnitt 230, starke Färbung links oben und rechts unten, schwache Färbung in der Mitte.

Im Gegensatz dazu entstehen bei histologischen Schnitten die Inhomogenitäten

nicht durch ein bildgebendes Gerät, sondern durch manuelles Einfärben. Hierbei unterliegt die Färbung einer menschlich bedingten Schwankung, bei der nicht nur unterschiedlich stark (Einwirkdauer und Menge des Färbemittels), sondern innerhalb einer Schicht, örtlich ungleichmässig gefärbt wird (vgl. dazu Abb. 3.1 und 3.2). Diese Schwankungen lassen sich selbst mit höchst präziser Arbeit nicht ganz

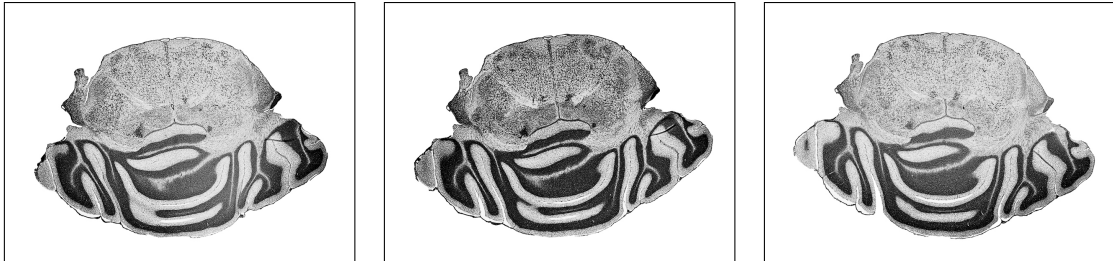


Abbildung 3.2: Deutliche Unterschiede der Intensität der Färbung bei aufeinander folgenden Schnitten. v.l.n.r. Schnitt 385, 386, 387.

verhindern und sind daher typisch für histologische Schnitte.

Bei intensitätsbasierten Verfahren zur Registrierung ist der Grauwert (bzw. Intensitätswert) die einzige Information, die bei der Verarbeitung der Bilder berücksichtigt wird. Das heisst, dass ein Objekt nur dann als ein Objekt zählt (und nicht als zwei verschiedene), wenn die Darstellung durch Grauwerte gleichwertig ist. Wenn beispielsweise ein Bildabschnitt den Intensitätswert 128 hat und ein anderer Bildabschnitt des gleichen Objekts aufgrund der unterschiedlichen Färbung/Beleuchtung den Intensitätswert 203, so werden diese Bereiche algorithmisch zwei unterschiedlichen Objekten zugeordnet und damit allein durch den Grauwert nicht in Korrespondenz gebracht. Betrachten wir nun Abb. 3.1 etwas genauer, so erkennt man im linken Bild, dass die Grosshirnrinde im Bildteil unten links einen sehr hohen Intensitätswert (hell) und im Bildteil oben rechts einen sehr niedrigen Intensitätswert (dunkel) besitzt. Diese unterschiedliche Repräsentierung des gleichen anatomischen Objektteils macht damit einen direkten (automatischen) Vergleich oder eine Objekterkennung bzw. Segmentierung ohne weitere Vorverarbeitung unmöglich.

Um für Vergleiche, Segmentierungen, Registrierungen und/oder Rekonstruktionen optimale Voraussetzungen zu schaffen, muss eine so genannte *Homogenisierung* der Bilddaten erfolgen. Eine Homogenisierung ist ein Prozess, bei dem inhomogene Bereiche von ihrer Intensitätsrepräsentierung derart aneinander angepasst werden, dass eine homogene Verteilung der Grauwerte erreicht wird und damit anatomisch korrespondierende Bereiche miteinander in Bezug zu setzen sind. Zur Homogenisierung werden in der Literatur verschiedene Techniken beschrieben [DMD⁺04, MB03]. Allen Verfahren ist gemein, dass die Färbung, Beleuchtung, Belichtung, Bestrahlung oder das Rauschen in irgendeiner Form geschätzt wird (z.B. statistisch oder morphologisch), um so die Inhomogenitäten zu korrigieren. Hierbei versuchen statistische Methoden Verteilungen und Erwartungswerte zu berechnen und Plausibilitätstafeln aufzustellen. Die Schätzung der Verteilungen

kann dabei durch statistische Modellannahmen, Histogrammanalysen oder spezielle morphologische Operationen bzw. morphologische Filter umgesetzt werden. Eine sehr rudimentäre Intensitätsanpassung ist die lineare Filterung. Hierbei wird der vermeintliche Grauwert durch einfache lokale Nachbarschaftsverhältnisse ermittelt. Diese Verfahren arbeiten lokal in einem sehr kleinen Gebiet und werden daher nicht weiter betrachtet.

In den nachfolgenden Abschnitten werden wir zwei grundlegend unterschiedliche Verfahren vorstellen, die speziell im Bereich der Homogenisierung von histologischen Bildern Verwendung finden. Zuerst werden wir in Abschnitt 3.2 das so genannte *Kochrezept* vorstellen, das in der digitalen histologischen Praxis sehr erfolgreich eingesetzt wird. Das Kochrezept basiert auf der HUM-Methode [ACL87, BMR98] und eignet sich hervorragend zur Homogenisierung einzelner Bilder, indem die Färbeintensitäten durch morphologische Operationen angeglichen werden. Schliesslich beschreiben wir in Abschnitt 3.3 die Homogenisierung durch Bildglättung mittels *Total-Variation-Norm* (TV-Norm). Hier diffundieren durch scharfe Kanten getrennte Bereiche zu homogenen Flächen mit mittlerem Grauwert, wobei die Kanten scharf erhalten bleiben. Ist die Varianz der Inhomogenität bekannt, so kann mit einer Variante des Verfahrens das Bild nahezu ideal homogenisiert werden.

Neben den beiden hier vorgestellten Methoden gibt es in der Literatur noch zahlreiche weitere Verfahren, die für unterschiedlichste Aufgabenstellungen entworfen und optimiert sind. In dieser Arbeit beschränken wir die Auswahl auf diese zwei Verfahren, da sie sich im Problemzusammenhang als besonders geeignet erwiesen haben. Im ersten Fall, des Kochrezepts, handelt es sich um eine Methode, die im histologischen Alltag ein Standard ist und täglich eingesetzt wird. Die zweite Variante ist eine Methode, die für die gleichzeitige Homogenisierung und Registrierung am geeignetsten erscheint, da es das Bild entrauscht und homogenisiert, aber die Kanten, die für die Registrierung wichtig sind, ideal erhält.

3.2 Erweiterte HUM-Methode

In 1987 wurde von Axel et al. [ACL87] eine Methode vorgestellt, die sich HUM (homomorphic-unsharp-masking) nennt. Diese Methode wurde für die Intensitätswertkorrektur in MR-Bildern entwickelt und dort erfolgreich eingesetzt. Eine Optimierung der Methode bezüglich Performanz stellten Brinkmann et al. [BMR98] vor. Darüber hinaus gaben sie einen Überblick über die verschiedenen Einsatzgebiete der HUM-Methode. Neben der von Brinkmann et al. vorgestellten Einsatzgebiete hielt die Methode Einzug in die digitale Histologie. Die ursprüngliche Methode wurde durch morphologische Operationen ergänzt und verfeinert [APG06]. Diese Variante der HUM-Methode hat sich als so genanntes *Kochrezept* in der digitalen Histologie durchgesetzt. Im Gegensatz zu Histogramm-basierten Verfahren lässt sich eine Homogenisierung der extrem großen Datenmengen schnell und effizient durchführen, ohne dass Vorwissen über die Inhomogenität vorliegen muss.

Nachfolgendes Schema erläutert die erweiterte Methode zur Homogenisierung von histologischen Schnittbildern in einzelnen Schritten. Die verwendeten Parameter haben sich in der histologischen Praxis im Team um Prof. O. Schmitt, Universität Rostock, bewährt und werden daher nicht weiter kommentiert:

1. Überführung der Daten aus dem redundanten RGB-System in Grauwerte durch Isolierung eines einzelnen Kanals (z.B. Rotkanal).
2. Gegebenenfalls Invertierung, sodass interessante Strukturen den höchsten Intensitätswert haben und der Hintergrund den Wert 0 besitzt.
3. Einfache Segmentierung des Objekts (Trennung von Objekt und Hintergrund) mittels Threshold zur Generierung einer Objekt-Maske.
4. *Filling*, d.h. Hintergrundpixel, die vom Objekt eingeschlossen sind werden zur Binärmaske hinzugenommen.
5. *Erosion* mit rundem Kern mit Durchmesser 2.
6. *Opening* mit rundem Kern mit Durchmesser 2.
7. Maskierung des Grauwert-konvertierten Bildes mit der Binärmaske.
8. Tiefpassfilterung des maskierten Bildes mit Filter der Größe 29, vier Iterationen.
9. Maskierung des gefilterten Bildes mit Binärmaske.
10. Invertierung des maskierten, gefilterten Bildes.
11. Multiplikation oder Subtraktion mit offset 255 des invertierten Bildes mit Bild aus Schritt 7.
12. Normalisierung auf adäquate Grauwertskala.

Der Ablauf ist in Anhang C.1 als Matlab-code unter Verwendung der Image-Processing-Toolbox angegeben.

Die Methode ist leicht zu implementieren und durch die einfachen Operationen mit kleinen Filterkernen insgesamt sehr schnell. Wie in Abb. 3.3 zu sehen ist, wird ein Bild einer gefärbten histologischen Schicht unabhängig vom Ort und der Intensität der Inhomogenität gleichmäßig homogenisiert. Wie sich allerdings gut erkennen lässt, geht der lokale Kontrast verloren und anhand der Intensitäten lassen sich die unterschiedlichen Gewebetypen nicht mehr differenzieren. Darüber hinaus lässt sich die Methode ausschließlich auf einzelne Bilder anwenden, wodurch unter Umständen benachbarte Schichten der Bildserie wiederum unterschiedliche Grauwertrepräsentierungen des gleichen Gewebes, bzw. der gleichen Struktur bekommen können. Ein weiterer Nachteil ist, dass die Methode nicht skalierbar ist, d.h. die Homogenisierung mit der HUM-Methode kann nur vollständig oder gar nicht ausgeführt werden, nicht aber in kleinen fortschreitenden Schritten, die eine

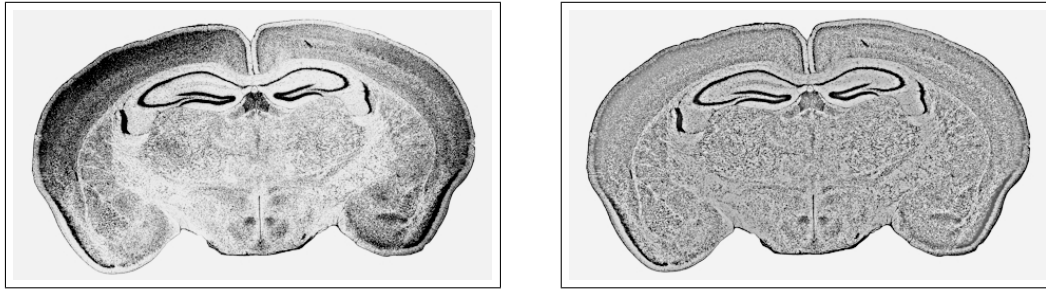


Abbildung 3.3: Homogenisierung eines Bildes mit der HUM-Methode. *links:* ohne Homogenisierung, starke Farbeunterschiede sichtbar, hoher lokaler Kontrast; *rechts:* mit HUM-Homogenisierung, vollständig homogen, allerdings unter Einbußen des lokalen Kontrasts.

Übertragung von Zwischenergebnissen in anderer Schichten ermöglicht. Wendet man diese Methode nach einer Registrierung an, so kann durch eine dreidimensionale Erweiterung der komplette Datensatz derart homogenisiert werden, dass eine optisch harmonische Gesamtdarstellung möglich wird.

3.3 Total Variation

Betrachten wir den Färbeprozess und damit die Inhomogenitäten von histologischen Bildern genauer, so erkennen wir, dass die Zellen und Zellverbände die Farbstoffe annehmen und damit den Kontrast bilden. Zellen sind in charakteristischen Strukturen und gewebespezifisch angeordnet. In den Bildern von histologischen Schichten werden einzelne Zellen bis hin zu Zellverbänden im wesentlichen nur durch einen einzelnen Pixel oder einem Teil davon repräsentiert. Damit lässt sich die Verteilung von Zellen wie die Verteilung eines bestimmten Rauschens interpretieren. Die Färbintensität bei histologischen Bildern ist damit äquivalent zur Intensität von Rauschen bei klassischen verrauschten digitalen Bildern. Bei zusammenhängenden Strukturen bzw. Gewebetypen sind die Zellen gleichermaßen angeordnet, so dass die Inhomogenität der Färbung als charakteristisches Rauschen für diese Struktur interpretiert werden kann. Durch die genannte Übertragung von Inhomogenität der Färbung zu Rauschen und Rauschverteilung lassen sich Methoden und Verfahren, die sich als hervorragend für das *Entrauschen*, die Entfernung von Rauschen, bzw. *Glätten von Bildern* herausgestellt haben, auf die Homogenisierung übertragen. Aufgrund der unterschiedlichen Zelldichten in den unterschiedlichen Gewebe- und Strukturtypen muss ein Verfahren verwendet werden, dass lokal individuell entrauscht und nicht global arbeitet. Um den scharfen Grenzen zwischen den einzelnen Gewebetypen gerecht zu werden, darf ein Verfahren die Kanten nicht verwischen. Dadurch fallen klassischen Filterverfahren aus.

Bereits Anfang der neunziger Jahre wurden sehr aufwendige Verfahren entwickelt, um das Rauschen in Bildern zu eliminieren. Vor allem zur Verbesserung von Satellitenfotos wurden diese Verfahren eingesetzt, um die Bildqualität derart

zu verbessern, dass Details und Kontraste wieder zu erkennen sind, die vorher durch das Rauschen unterdrückt wurden. Mit den bis dahin bekannten Verfahren wurde zwar das Rauschen unterdrückt, doch verlor sich dadurch auch der lokale Kontrast und damit die Kanten. Damit waren die unterschiedlichen Objekte auf den Bildern nicht mehr eindeutig zu erkennen. Verwendete man allerdings Kanten erhaltende Filter, so reduzierte sich das Rauschen nicht genug, um die Objekte zu erkennen.

3.3.1 TV-Entrauschen bzw. TV-Glättung nach Rudin et al.

Rudin, Osher und Fatemi [ROF92] stellten 1992 ein Verfahren zur nichtlinearen Entrauschung auf Basis der totalen Variation vor. Ausgehend von der im Vorfeld bekannten Standardabweichung des additiven Rauschens eliminiert das Verfahren iterativ das Rauschen, wobei Kanten scharf erhalten bleiben. Wir stellen diesen Algorithmus besonders in den Vordergrund, da dieser als Minimierung unter Nebenbedingungen formuliert wurde und damit zu zeitabhängigen nichtlinearen partiellen Differentialgleichungen führt, wie es bei der Registrierung ebenfalls der Fall ist. Die Grundannahme beim Verfahren von Rudin et al. ist, dass ein gemessenes bzw. verrauschtes Bild $B_0(\mathbf{x})$ (zur genauen Definition eines Bildes, siehe Definition 2.1.1 auf Seite 14) mit $\mathbf{x} \in \Omega$ aus einem reinen, rauschfreien Bild $B(\mathbf{x})$ und einem additiven Rauschanteil $r(\mathbf{x})$ besteht,

$$B_0(\mathbf{x}) = B(\mathbf{x}) + r(\mathbf{x}).$$

Die Annahme an das Rauschen ist, dass es den Mittelwert 0 und die Varianz σ^2 hat. Die Annahmen über Mittelwert und Varianz gehen in das Minimierungsproblem als Nebenbedingungen ein. Osher und Rudin haben in [OR90] bereits bei den Untersuchungen zu Shock Filtern gezeigt, dass Aufgrund der Oszillationen weder eine L_2 -, noch eine L_1 -Minimierung sachgemäß ist. Statt dessen schlugen sie die Minimierung der L_1 -Norm des Gradienten vor und zeigten, dass dies der geeignete Raum ist.

Definition 3.3.1 (TV-Norm)

Die L_1 -Norm des Gradienten eines Bildes $B(\mathbf{x})$,

$$\int_{\Omega} |\nabla B(\mathbf{x})|_2 \, d\mathbf{x},$$

wird als *TV-Norm* von $B(\mathbf{x})$ bezeichnet.

Das Minimierungsproblem zur nichtlinearen Entrauschung von Bildern nach Rudin et al. lässt sich wie folgt formulieren:

Problem 3.3.2 (Nichtlineares Entrauschen/Glätten)

Minimiere

$$\int_{\Omega} |\nabla B(\mathbf{x})|_2 \, d\mathbf{x} \stackrel{2D}{=} \int_{\Omega} \sqrt{B_{x_1}(x_1, x_2)^2 + B_{x_2}(x_1, x_2)^2} \, dx_1 dx_2, \quad (3.1)$$

bezüglich $B(\mathbf{x})$ mit den Nebenbedingungen

$$\int_{\Omega} B(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} B_0(\mathbf{x}) \, d\mathbf{x} \quad (3.2)$$

und

$$\int_{\Omega} \frac{1}{2} (B(\mathbf{x}) - B_0(\mathbf{x}))^2 \, d\mathbf{x} = \sigma^2, \quad \text{mit gegebenem } \sigma > 0. \quad (3.3)$$

In der Nebenbedingung (3.2) steckt, dass der Mittelwert des Rauschens Null ist, während in Nebenbedingung (3.3) die a priori bekannte Standardabweichung des Rauschens mit einfließt. Nach Rudin et al. konvergiert ein Minimierungsprozess zu obigem Minimierungsproblem gegen eine rauschfreie Version von $B(\mathbf{x})$.

In den nächsten Schritten werden wir eine Möglichkeit vorstellen, die harten Nebenbedingungen in das Minimierungsproblem einfließen zu lassen, um anschließend eine Lösungsvorschrift zu formulieren.

Als notwendige Voraussetzung für ein Minimum, muss die erste Ableitung Null werden. Damit die Nebenbedingungen stets berücksichtigt werden, bauen wir sie fest in unsere Formulierung ein. Nehmen wir dazu nun an, dass B stetig differenzierbar ist. Da es sich bei Problem 3.3.2 um ein klassisches Optimierungsproblem unter Nebenbedingungen handelt, verwenden wir den Lagrange-Ansatz [NW06] mit den zugehörigen Lagrange-Parametern $\lambda_1, \lambda_2 \in \mathbb{R}^{>0}$,

$$\begin{aligned} F(\mathbf{x}, \lambda_1, \lambda_2) &= \int_{\Omega} |\nabla B(\mathbf{x})|_2 \, d\mathbf{x} \\ &+ \lambda_1 \left(\int_{\Omega} B(\mathbf{x}) \, d\mathbf{x} - \int_{\Omega} B_0(\mathbf{x}) \, d\mathbf{x} \right) \\ &+ \lambda_2 \left(\int_{\Omega} \frac{1}{2} (B(\mathbf{x}) - B_0(\mathbf{x}))^2 \, d\mathbf{x} - \sigma^2 \right). \end{aligned}$$

Bildet man nun die Ableitungen der Lagrangefunktion F nach \mathbf{x} , λ_1 und λ_2 und setzt diese gleich Null, so bildet das entstehende Gleichungssystem eine notwendige Bedingung für ein Minimum von Problem 3.3.2 (ausführliche Herleitung im

Anhang B.5:

$$0 = \frac{\nabla^\top(\nabla B(\mathbf{x}))}{|\nabla B(\mathbf{x})|_2} - \lambda_1 - \lambda_2(B(\mathbf{x}) - B_0(\mathbf{x})) \quad (3.4)$$

$$= \frac{\partial}{\partial x_1} \left(\frac{B_{x_1}(\mathbf{x})}{\sqrt{B_{x_1}(\mathbf{x})^2 + B_{x_2}(\mathbf{x})^2}} \right) + \frac{\partial}{\partial x_2} \left(\frac{B_{x_2}(\mathbf{x})}{\sqrt{B_{x_1}(\mathbf{x})^2 + B_{x_2}(\mathbf{x})^2}} \right) - \lambda_1 - \lambda_2(B(\mathbf{x}) - B_0(\mathbf{x})) \quad (3.5)$$

innerhalb Ω , mit

$$\frac{\partial B}{\partial \mathbf{n}}(\mathbf{x}) = 0 \quad \text{auf dem Rand } \Gamma \text{ von } \Omega. \quad (3.6)$$

Hierbei ist $\mathbf{n} = (n_1, \dots, n_d) : \Gamma \rightarrow \mathbb{R}^d$ das äussere Einheits-Normalenfeld. Bedingung (3.6) wird **Neumann-Randbedingung** genannt.

Rudin et al. verwendeten zur Lösung der Differentialgleichung (3.4) eine Gradientenabstiegsmethode und die Zeit als Evolutionsparameter, um so zu einer numerischen Lösung zu gelangen. Mit $B_t(\mathbf{x})$ bezeichnen wir die zeitliche Ableitung von $B(\mathbf{x})$, die im Falle der Konvergenz zu Null wird. In (3.4) steht der Parameter λ_1 für die erste Nebenbedingung (3.2). Startet man mit $B(\mathbf{x}, 0) = B_0(\mathbf{x})$, so ist und bleibt die erste Nebenbedingung bereits erfüllt, wodurch der Parameter λ_1 nicht weiter betrachtet werden muss. Da wir damit nur noch einen Parameter in unserer Gleichung haben, bezeichnen wir diesen im folgenden mit λ . Setzen wir einmal voraus, dass λ bekannt wäre, so lässt sich Gleichung (3.4) mit den Nebenbedingungen wie folgt zu einem Zeitschrittverfahren umformulieren:

$$B_t(\mathbf{x}, t) = \frac{\partial}{\partial x_1} \left(\frac{B_{x_1}(\mathbf{x}, t)}{\sqrt{B_{x_1}(\mathbf{x}, t)^2 + B_{x_2}(\mathbf{x}, t)^2}} \right) + \frac{\partial}{\partial x_2} \left(\frac{B_{x_2}(\mathbf{x}, t)}{\sqrt{B_{x_1}(\mathbf{x}, t)^2 + B_{x_2}(\mathbf{x}, t)^2}} \right) - \lambda(B(\mathbf{x}, t) - B_0(\mathbf{x})), \quad \text{für } t > 0, x_1, x_2 \in \Omega, \quad (3.7)$$

$$B(\mathbf{x}, 0) := B_0(\mathbf{x}) \quad \text{und} \quad \frac{\partial B}{\partial \mathbf{n}} = 0 \quad \text{auf } \Gamma.$$

Da λ selbst wiederum eine Unbekannte ist, müssen wir überlegen, wie λ berechnet werden kann. Betrachten wir dazu Gleichung (3.7) genauer. Gehen wir davon aus, dass Konvergenz eintritt, so verändert sich B nicht mehr und damit wird die zeitliche Ableitung B_t zu Null. Multipliziert man dann die rechte Seite mit der Differenz $(B - B_0)$, integriert partiell und nutzt Gleichung (3.3) aus, so kann nach λ aufgelöst werden (Details zur Rechnung siehe Anhang B.6; der Übersicht halber

haben wir das Argument (\mathbf{x}, t) von B weggelassen) und es folgt:

$$\begin{aligned} \lambda &= -\frac{1}{2\sigma^2} \int_{\Omega} |\nabla B|_2 - \frac{\langle \nabla B_0, \nabla B \rangle_2}{|\nabla B|_2} \, d\mathbf{x} \\ &\stackrel{2D}{=} -\frac{1}{2\sigma^2} \int_{\Omega} \left[\sqrt{B_{x_1}^2 + B_{x_2}^2} - \left(\frac{(B_0)_{x_1} B_{x_1}}{\sqrt{B_{x_1}^2 + B_{x_2}^2}} + \frac{(B_0)_{x_2} B_{x_2}}{\sqrt{B_{x_1}^2 + B_{x_2}^2}} \right) \right] dx_1 dx_2 \end{aligned} \quad (3.8)$$

Die Schwierigkeit nun ist, dass λ erst ausgerechnet werden kann, wenn B konvergiert ist. Andererseits fließt λ in die Berechnung des neuen B ein. Numerisch lässt sich dies durch eine *zweistufige Iterationsvorschrift* lösen, d.h. zuerst wird Gleichung (3.7) für einen neuen Zeitpunkt ausgerechnet und anschließend Gleichung (3.8). Diese Berechnungen wechseln sich bis zur Konvergenz von B und damit auch von λ ab.

Im Falle des Entrauschens von Bildern ist dieses Vorgehen angebracht. Bei den histologischen Serienschritten besteht allerdings die Schwierigkeit, dass es keine eindeutige Standardabweichung für das ganze Bild gibt. Durch unterschiedliche Gewebetypen und den damit verbundenen unterschiedlichen Zellverteilungen kann unabhängig von der individuellen Färbung keine einheitliche Standardabweichung angegeben werden. Daher lösen wir uns im nachfolgenden Abschnitt von der Annahme, dass die Standardabweichung bekannt ist, nehmen dafür eine geringere Entrauschleistung in Kauf, gewinnen damit aber den numerischen Vorteil durch den Verzicht auf ein zweistufiges Iterationsverfahren.

3.3.2 Abwandlung des Verfahrens

Bei der Konstruktion des Verfahrens wurde vorausgesetzt, dass die Varianz des Rauschens explizit bekannt ist. Sollte die Varianz nicht bekannt sein, so kann eine Abwandlung des Verfahrens in vielen Fällen trotzdem zu zufrieden stellenden Ergebnissen führen. Dazu wird die zweite Nebenbedingung (3.3) fallen gelassen, bzw. $\lambda_2 = \lambda \equiv 0$ angenommen. Praktisch bedeutet dies, dass in jeder Iteration nicht wie bisher optimal zur gegebenen Varianz des Rauschens das Bild geglättet wird, sondern stets auf die gleiche Weise. Die Kanten im Bild bleiben zwar nach wie vor erhalten, doch kommt es zu einem stärkeren Zusammenschluss einzelner Gebiete und damit zu einem Effekt, der als Weichzeichner bekannt ist, d.h. der lokale Kontrast weicht im Laufe der Iterationen immer weiter auf und es entstehen zunehmend scharf umgrenzte Gebiete mit gleichem Grauwert. Durch den fortschreitenden Glättungseffekt nennen wir das Verfahren nun *TV Bildglättung*:

Algorithmus 3.3.3 (Iterationsvorschrift zur TV Bildglättung)

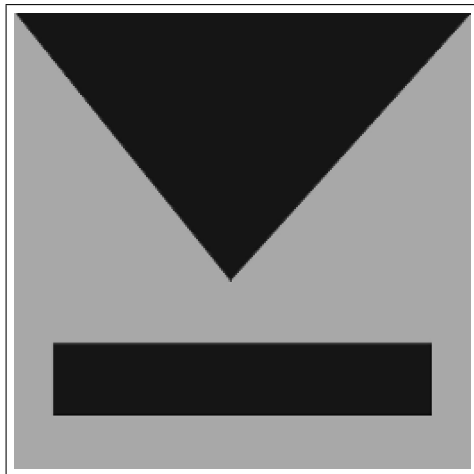
Zu einem gegebenen verrauschten Bild B_0 mit unbekannter Varianz des Rauschens, einem Zeitschritt von $\delta t \in \mathbb{R}^{>0}$, dem Iterationszähler $k \in \mathbb{N}$, dem daraus resultierenden Iterationsschritt $t_k := k \cdot \delta t$ und der maximalen Iterationszahl k_{max} kann durch folgende Iterationsvorschrift eine rauschreduzierte, bzw. geglättete Version B berechnet werden:

1. Initialisiere, so dass die Nebenbedingung zum Mittelwert von B erfüllt ist,
 $B^{(0)} := B_0, \quad k := 0, \quad 0 < \varepsilon \ll 1.$
2. Für alle $k = 0, \dots, k_{max} - 1$ berechne $B^{(k+1)} = B^{(k)} + \delta t B_t^{(k)}$, bzw. im zweidimensionalen Fall wie in Gleichung (3.7),

$$B^{(k+1)} = B^{(k)} + \delta t \left[\frac{\partial}{\partial x_1} \left(\frac{B_{x_1}^{(k)}}{\sqrt{(B_{x_1}^{(k)})^2 + (B_{x_2}^{(k)})^2 + \varepsilon^2}} \right) + \frac{\partial}{\partial x_2} \left(\frac{B_{x_2}^{(k)}}{\sqrt{(B_{x_1}^{(k)})^2 + (B_{x_2}^{(k)})^2 + \varepsilon^2}} \right) \right].$$

Um den Unterschied einmal genauer zu betrachten nehmen wir ein akademisches Beispiel. In Abbildung 3.4 sehen wir oben links ein rauschfreies Bild mit zwei Strukturen mit harter Kante. Auf dieses Bild haben wir ein additives Rauschen mit Standardabweichung $\sigma = 20$ gegeben (Abb. 3.4, oben rechts). Verwenden wir $\lambda \equiv 0$, so bleibt im Bild ein Rauschen zurück, welches allerdings eine deutlich geringere Standardabweichung besitzt. Das Bild ist gegenüber der verrauschten Variante deutlich glatter geworden, doch es sind immer noch Störungen erkennbar. Im Gegensatz dazu ist im Bild unten rechts eine nahezu perfekte Rekonstruktion zu sehen. Das Rauschen ist vollständig verschwunden, nur noch kleinste Bereiche zeigen leichte Schwankungen um den tatsächlichen Grauwert.

Das Verfahren zur TV-Bildglättung bietet die Möglichkeit, je nach Voraussetzung auf zwei unterschiedliche Weisen zu funktionieren. Ist die Varianz des Rauschens bekannt, so kann das Rauschen optimal entfernt werden. Ist die Varianz nicht bekannt, so kann das Verfahren durch eine kantenerhaltene Glättung das Rauschen im Bild stark reduzieren. Wählt man im Verfahren $\lambda \equiv 0$, so fließt keine Annahme an die Varianz ein und das Verfahren muss durch eine feste Anzahl an Iterationsschritten beschränkt werden. Ist die Varianz σ^2 des Rauschens bekannt, so wird λ iterativ angepasst und das Verfahren konvergiert automatisch nach vollständiger Entrauschung.



(a) Ideales, rauschfreies Bild

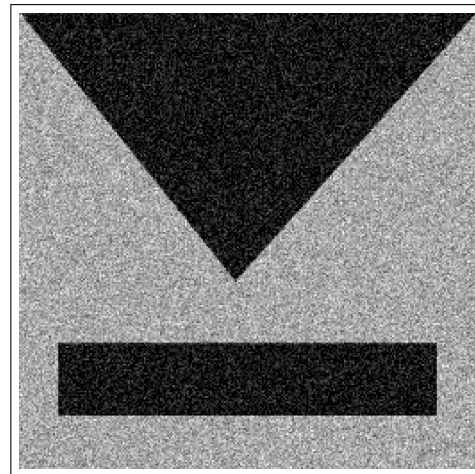
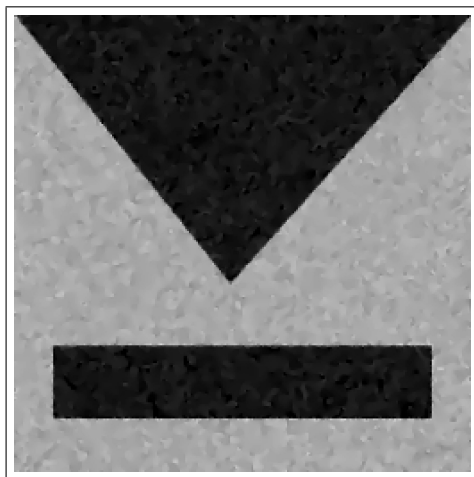
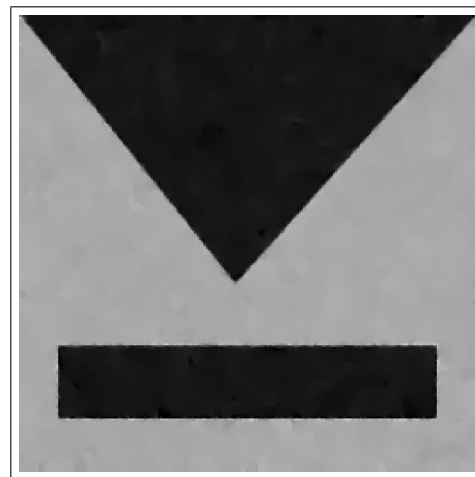
(b) Bild mit additivem Rauschen, $\sigma = 20$ (c) Entraushtes Bild, $\lambda \equiv 0$ (d) Entraushtes Bild, $\lambda > 0$

Abbildung 3.4: Verdeutlichung des TV-Entrauschens/Glättens mit ein- und zweistufiger Iteration ($k_{max} = 80$, $\sigma = 20$).

3.3.3 Anwendung auf Bilder der Serienschritte

Im Falle der Serienschrittregistrierung ist, wie oben bereits erwähnt, die Varianz des Rauschens nicht bekannt. Daher beschränken wir uns auf die iterative Bildglättung nach Algorithmus 3.3.3. In Abbildung 3.5 sind verschiedene Zeitpunkte der TV-Bildglättung dargestellt. Es ist deutlich zu erkennen, dass die ursprünglichen Kanten scharf erhalten bleiben, das Rauschen aber sukzessive verschwindet, indem homogene Bereiche entstehen, die den mittleren Grauwert der Region annehmen. In Abbildung 3.6 lässt sich dieser Effekt noch etwas genauer erkennen. Mit zunehmender Iterationszahl verschmelzen kleinere Bereiche zu größeren und es entstehen homogene, scharf abgetrennte Bereiche. Die Kanten bleiben über die Iterationen hin optimal erhalten und entwickeln sich zu glatten Konturen.

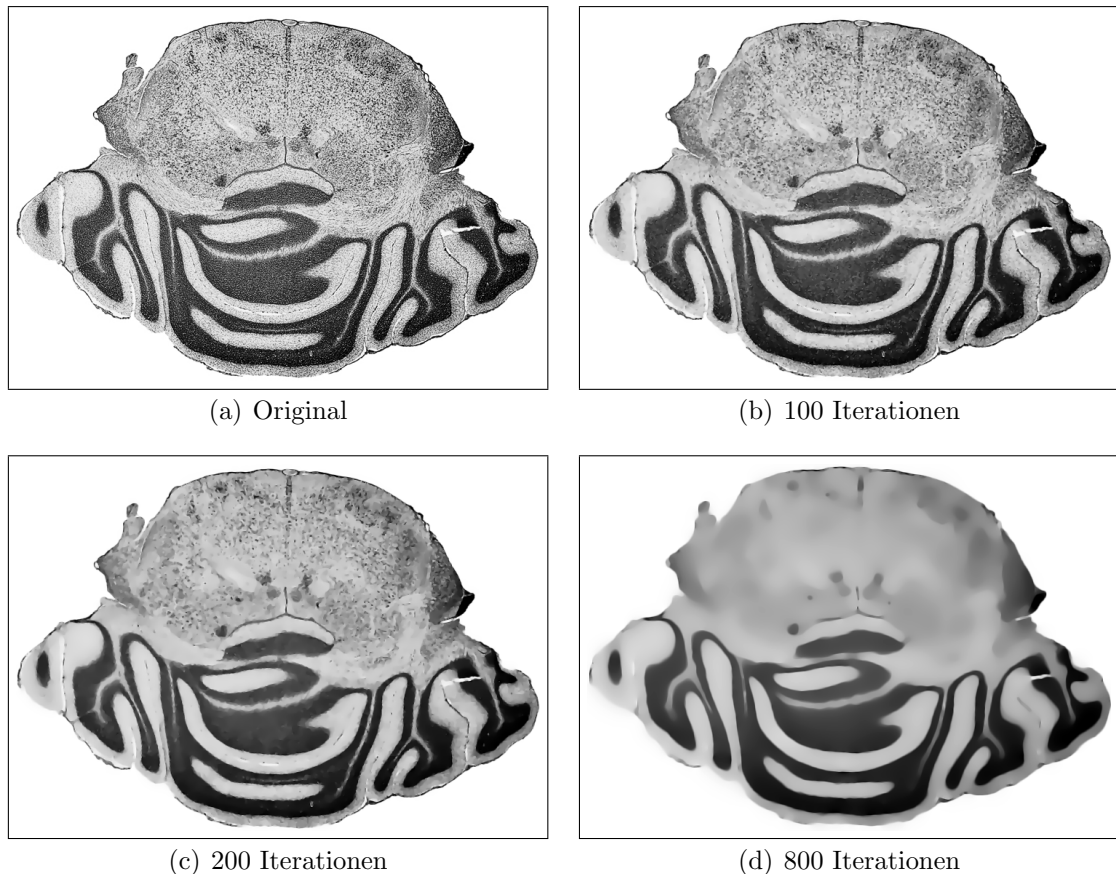


Abbildung 3.5: TV-Bildglättung, $\varepsilon = 0.5$: mit zunehmender Iterationszahl nimmt das Rauschen ab und es entstehen homogene Bereiche mit dem Mittelwert als Grauwert. Die Kanten bleiben scharf erhalten und verwischen nicht.

3.3.4 Hinweise zur Implementierung

Für die Implementierung des Verfahrens haben wir eine Diskretisierung der Differentialoperatoren mit Finiten Differenzen gewählt. Dadurch können wir die Randbedingungen $\partial B = 0$ auf Γ implizit umsetzen, indem die Randpunkte wiederholt werden und damit der Ableitungswert Null garantiert ist. Um den Unterschied plastisch zeigen zu können, haben wir zusätzlich das zweistufige Iterationsverfahren implementiert, um am akademischen Beispiel die Wirksamkeit zu demonstrieren. Hierbei haben wir das Integral zur Bestimmung von λ durch eine gemittelte Summe approximiert. Um eine Beschleunigung des gesamten Verfahrens bei zweistufiger Iteration zu erzielen, haben wir die Berechnung von λ verzögert. D.h., dass in jedem Iterationsschritt ein neues B berechnet wird, allerdings λ nur alle 10 Iterationen berechnet wird. Die Änderung von λ ist über die Iterationen relativ gering, so dass die einzelnen, kleinen Schritte zu einem größeren Schritt zusammen gefasst werden. In unseren Beispielen haben wir dadurch ca. 30% der Gesamtrechenzeit eingespart. Optisch lassen sich die Unterschiede nicht festmachen, so dass wir von

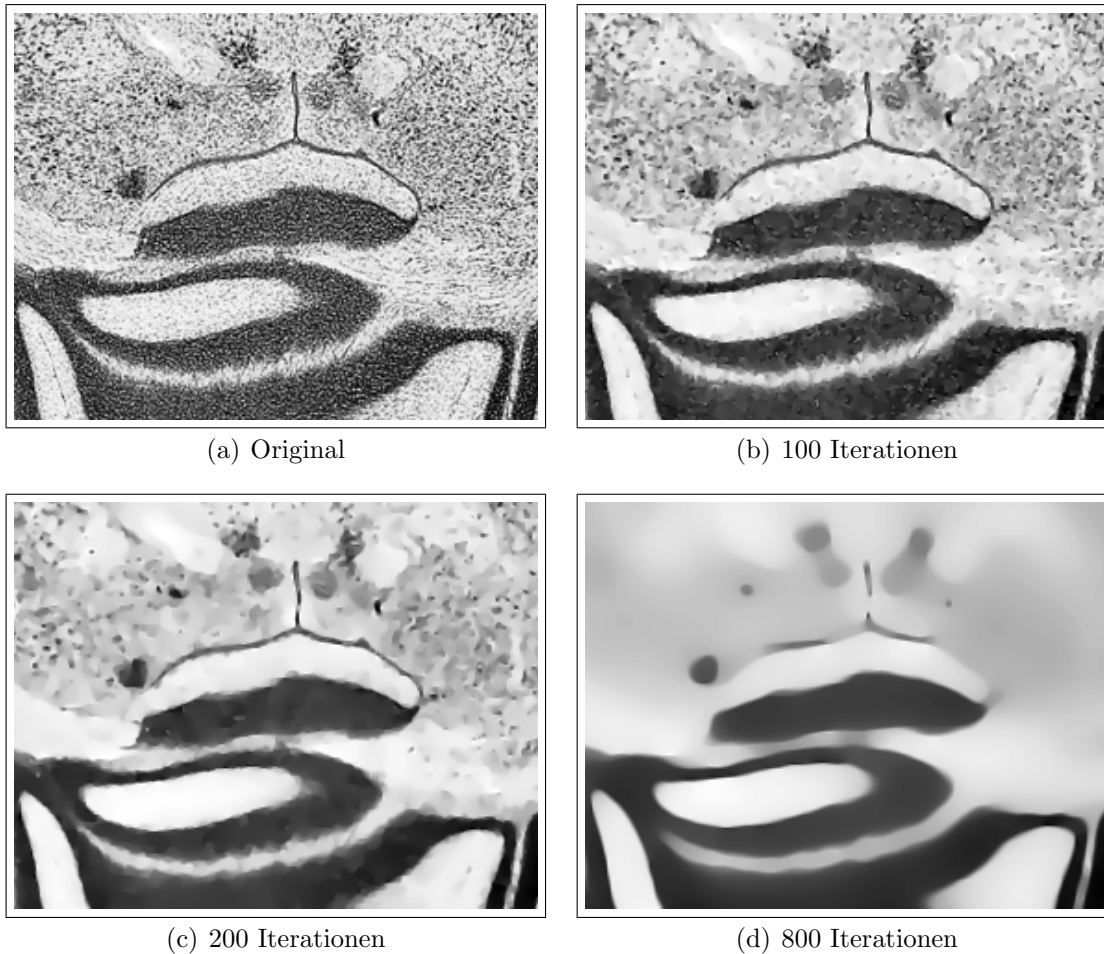


Abbildung 3.6: TV-Bildglättung im Detail, $\varepsilon = 0.5$. Einzelne Zellen verschmelzen zu homogenen Bereichen. Feine Strukturen verschwinden mit zunehmenden Iterationen. Scharfe Kanten bleiben erhalten und bilden sich zu sehr glatten Konturen aus.

einem gleichwertigen Ersatz der ursprünglich definierten Iterationsvorschrift sprechen.

Für alle gezeigten Beispiele haben wir für die Regularisierung $\varepsilon = 0.5$ verwendet. Wird ε zu klein gewählt, werden für das Entrauschen/Glätten zu viele Iterationen benötigt. Wird im Gegensatz dazu ε zu groß gewählt, so bleiben die Kanten nicht mehr scharf erhalten und verwischen ähnlich wie bei einem einfachen Glättungsfilter (z.B. Gaußfilter).

3.3.5 Regularisierung der TV-Norm

Betrachten wir nun die TV-Norm aus Definition 3.3.1 etwas genauer, so stellen wir fest, dass der Integrator in 0 nicht differenzierbar ist. Wenn wir Differenzierbarkeit gewährleisten können, so ist im Allgemeinen eine schnellere Konvergenz zu erwarten. Neben der schnelleren Konvergenz möchten wir zusätzlich den Einfluss

von Ausreißern dämpfen. Geman und Reynolds schlugen dazu in [GR92] Regularisierungsfunktionen vor.

Definition 3.3.4 (Regularisierte TV-Norm)

Sei $\Psi_\varepsilon : \mathbb{R} \rightarrow \mathbb{R}$ eine stetig differenzierbare Regularisierungsfunktion. Dann wird die regularisierte L_1 -Norm des Gradienten eines Bildes $B(\mathbf{x})$,

$$\int_{\Omega} \Psi_\varepsilon(|\nabla B(\mathbf{x})|_2^2) \, d\mathbf{x}, \quad (3.9)$$

als *regularisierte TV-Norm* von $B(\mathbf{x})$ bezeichnet.

In der Literatur sind verschiedene Formen der Regularisierung diskutiert worden (siehe auch Definition 4.1.2 auf Seite 39). Für die Registrierung der Serienschritte haben wir uns für die von Acar und Vogel in [AV94] vorgestellte Regularisierungsfunktion

$$\Psi_\varepsilon(s^2) = \sqrt{s^2 + \varepsilon^2}, \quad \text{mit } 0 < \varepsilon \ll 1 \quad (3.10)$$

entschieden. Diese Regularisierung hebt die Werte dicht bei 0 auf ein höheres Niveau, drosselt aber die großen Werte, so dass ein harmonischerer Verlauf gewährleistet ist. Durch die Addition von ε wird sichergestellt, dass der Wert Null nie erreicht wird, so dass selbst nach einer Differenzierung das Argument wohldefiniert bleibt. Die regularisierte TV-Norm wird in Kapitel 4 weiter verwendet und diskutiert.

Wir haben nun zwei verschiedene Verfahren zur Homogenisierung kennen gelernt. Beide Verfahren haben ihre besonderen Eigenheiten. Während die erste Variante, die HUM-Methode, hervorragend geeignet ist, nach der Registrierung als Vorbereitung für eine Visualisierung zu fungieren, stellt die TV-Methode eine gute Möglichkeit dar, die Schichten unter Erhaltung des lokalen Kontrastverhältnisses zu homogenisieren, um damit die Registrierung zu verbessern. Aus diesem Grund verwenden wir die TV-Methode in Kapitel 4 als Regularisierungsfunktion. Dabei verwenden wir die Homogenisierung nicht mehr in einem expliziten Schritt, sondern simultan mit der Registrierung, um so die gegenseitigen Bedingungen bestmöglich einzufangen.

4 Mathematische Beschreibung: Simultane Homogenisierung und Registrierung

In diesem Kapitel wollen wir das Problem der Homogenisierung und Registrierung von Bildserien mathematisch korrekt beschreiben. In Abschnitt 4.1 werden wir die Methode zur Simultanen Homogenisierung und Registrierung als Minimierungsproblem vorstellen. Im darauf folgenden Abschnitt 4.2 stellen wir die zugehörigen Euler-Lagrange-Gleichungen auf, die ein System nicht-linearer partieller Differentialgleichungen darstellen. Die Linearisierung des Systems werden wir im Abschnitt 4.3 zeigen. Im letzten Abschnitt 4.4 des Kapitels erläutern wir die Erweiterung der Methode zur Übertragung auf die Serienschnitte. Die nachfolgend beschriebene Methode wurde in 2006 [MW06a] veröffentlicht.

4.1 Die Methode: Aufstellen des Minimierungsproblems

Dazu ist es notwendig einige Begriffe zu definieren, um so eindeutig deren Sinn festzulegen. In Abschnitt 2.1 haben wir bereits die Begriffe *Bild*, *Deformationsfeld* und *deformiertes Bild* geklärt. In den nachfolgenden Abschnitten werden wir das in dieser Arbeit verwendete Distanzmaß vorstellen, einen *Korrekturterm* einführen, der die Homogenisierung während der Registrierung übernehmen wird, anschließend das *Registrierungsproblem* formulieren und eine notwendige Bedingung zur Lösung dieses Problems herleiten. Abschließen werden wir dieses Kapitel mit der Erweiterung der vorgestellten Methode auf Serienschnitte. Bis dahin werden wir, der besseren Übersicht halber, die Begriffe für die Registrierung von zwei Bildern klären.

4.1.1 Distanzmaß

Im Rahmen dieser Arbeit wollen wir uns aufgrund der Hauptanwendung, der Registrierung von Bildern histologischer Serienschnitte, auf intensitätsbasierte Distanzmaße beschränken. Die Serienschnitte sind gleicher Modalität, sehr hoch aufgelöst und weisen lokal einen sehr hohen Kontrast auf, wodurch ein schnell berechenbares Distanzmaß zu bevorzugen ist. Die nachfolgend vorgestellte Methode ist allerdings

allgemein gültig und kann daher auch mit anderen Distanzmaßen (z.B. Mutual Information, Cross-Correlation, etc.) genutzt werden.

Um Bilder miteinander vergleichen zu können muss ein Maß definiert werden, das den Unterschied der Bilder durch eine Kennzahl ausdrückt. Wie die euklidische Distanz von einzelnen Punkten ein Maß für den Unterschied der Position der Punkte ist, so kann der Unterschied zweier Bilder durch ein Distanzmaß ausgedrückt werden. Eine effizient zu berechnendes Maß ist die Summe der quadrierten Differenzen SSD .

Definition 4.1.1 (Sum of Squared Differences, SSD)

Seien R und T zwei Bilder auf Ω . Die *Summe der quadrierten Differenzen* (engl: *sum of squared differences*), kurz SSD , von R und T ,

$$\mathcal{D}^{\text{SSD}}[R, T] := \frac{1}{2} \int_{\Omega} \left(T(\mathbf{x}) - R(\mathbf{x}) \right)^2 d\mathbf{x},$$

ist eine Halbnorm und gibt den Unterschied zwischen R und T wieder.

Je kleiner die Distanz zwischen zwei Bildern ist, desto ähnlicher sind die Bilder. Bei identischen Bildern ist die Distanz Null $\mathcal{D}^{\text{SSD}}[R, R] = 0$. Soll die Distanz zwischen einem Bild T und einem Bild R nach Anwendung einer Deformation \mathbf{u} auf T bestimmt werden, so kann dies mit dem SSD-Maß geschehen, indem ein bereits deformiertes Bild $T_{\mathbf{u}}$ als neues Templatebild eingesetzt wird,

$$\mathcal{D}^{\text{SSD}}[R, T; \mathbf{u}] := \mathcal{D}^{\text{SSD}}[R, T_{\mathbf{u}}].$$

In vielen Anwendungsfällen reicht dieses Distanzmaß aus, um den Anforderungen zu genügen. Ein besonderer Vorteil dieses Maßes ist die einfache und effizient durchführbare Berechnung. Allerdings bereiten Rauschen, unterschiedliche Beleuchtung, ungleiche Färbung oder Kontrastmittel verschiedene Probleme (siehe auch Kapitel 3), da SSD gerade durch die Quadratur zu empfindlich reagiert und dadurch Ausreißer unverhältnismässig stark ins Gewicht fallen. Sind beispielsweise zwei Bilder geometrisch identisch, unterscheiden sich aber in ihrer Grauwertrepräsentierung durch z.B. unterschiedliche Beleuchtung oder Färbung, so tritt eine von Null verschiedene Distanz zwischen den eigentlich gleichen Bildern auf. Um den Einfluss von Ausreißern zu mildern, lassen sich verschiedene Techniken einsetzen. Hier möchten wir eine vielfach einsetzbare Methode vorstellen. Durch die Einführung einer Penalizer-Funktion wird eine Regularisierung der Abbildung des Distanzmaßes durchgeführt, so dass der Einfluss der Ausreißer nicht mehr so stark ins Gewicht fällt und damit das Maß robuster wird.

Definition 4.1.2 (Robustes SSD)

Sei \mathcal{D}^{SSD} wie oben, Ψ_ε eine monoton wachsende Abbildung und $\varepsilon \ll 1$. Dann bezeichnet

$$\mathcal{D}^{\text{SSD}}[R, T; \Psi_\varepsilon] := \frac{1}{2} \int_{\Omega} \Psi_\varepsilon \left(\left(T(\mathbf{x}) - R(\mathbf{x}) \right)^2 \right) d\mathbf{x}.$$

das *robuste SSD-Distanzmaß*.

Verwenden wir beispielsweise eine Penalizer-Funktion der Form

$$\Psi_\varepsilon(s^2) := \sqrt{s^2 + \varepsilon^2},$$

so ist der quadratische Einfluss des SSD-Maßes aufgehoben. Neben dieser speziellen Wahl sind auch andere Penalizer-Funktionen möglich. Weitere Beispiele aus der Literatur sind

Charbonnier	$\Psi_\varepsilon(s^2) = \frac{1}{\sqrt{1 + s^2/\varepsilon^2}}$	[CBFAB94],
Perona-Malik I	$\Psi_\varepsilon(s^2) = \frac{1}{1 + s^2/\varepsilon^2}$	[PM88],
Perona-Malik II	$\Psi_\varepsilon(s^2) = \exp\left(-\frac{s^2}{2\varepsilon^2}\right)$	[PM88] und
Weickert	$\Psi_\varepsilon(s^2) = \begin{cases} 1, & (s^2 = 0) \\ 1 - \exp\left(\frac{-3.31488}{s^8/\varepsilon^8}\right), & (s^2 > 0) \end{cases}$	[Wei98].

Die Einführung einer Penalizer-Funktion mildert zwar schon den Einfluss von Ausreißern, doch Inhomogenitäten durch z.B. Färbung bleiben weiterhin unbeeinflusst. In Abschnitt 8.2 wird auf diese Problematik weiter eingegangen und ein Distanzmaß höherer Ordnung vorgestellt. Die praktischen Nachteile, wie vor allem zu hoher Rechenaufwand und Speicherbedarf, machen allerdings den Einsatz in der Registrierung von Serienschritten derzeit nicht möglich.

Sollen des weiteren die Inhomogenitäten sogar ausgeglichen werden, so muss zusätzlich zur reinen Deformation der einzelnen Pixel eine Veränderung der Grauwerte stattfinden. Dazu führen wir einen *Korrekturterm* ein, der durch eine Skalierung der Grauwerte Inhomogenitäten ausgleichen kann, die nicht durch Geometrieänderungen durchgeführt werden können.

Definition 4.1.3 (Korrekturterm)

Eine zweimal stetig differenzierbare Abbildung auf Ω

$$\begin{aligned} c : \Omega &\rightarrow \mathbb{R}^+ \\ \mathbf{x} &\mapsto c(\mathbf{x}) \end{aligned}$$

die eine Skalierung von Intensitätswerten von Bildern realisiert, heisst *Korrekturterm*.

Mit Hilfe der Grauwertkorrektur lässt sich nun ein auf SSD basierendes erweitertes Distanzmaß festlegen, in das neben zwei Bildern die Deformation \mathbf{u} und der Korrekturterm c eingehen.

Definition 4.1.4 (Erweitertes Distanzmaß)

Seien R und T Bilder auf Ω , $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$ ein Deformationsfeld und $c : \Omega \rightarrow \mathbb{R}^+$ ein Korrekturterm für die Grauwerte in T . Dann wird

$$\begin{aligned} \mathcal{D}^{\text{cSSD}}[R, T; \mathbf{u}, c] &:= \mathcal{D}^{\text{SSD}}[R, cT_{\mathbf{u}}] \\ &= \frac{1}{2} \int_{\Omega} \left(c(\mathbf{x})T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}) \right)^2 d\mathbf{x} \end{aligned}$$

erweitertes SSD-Distanzmaß genannt.

Durch das erweiterte Distanzmaß wird bei der Berechnung des Unterschieds zwischen zwei Bildern neben den Intensitätswerten der Bilder zusätzlich der Einfluss von Deformationsfeld und Korrekturterm berücksichtigt. Damit kann zu fest gegebenen Bildern die Distanz einerseits durch Deformationsfelder, andererseits durch Korrekturterme reduziert oder vergrößert werden. Die Gefahr ist nun, dass durch ungünstige Wahl von Deformationsfeldern oder Korrekturtermen die Distanz der Bilder vom Wert her reduziert wird, aber praktisch keine Veränderung stattfindet, die einen Fortschritt darstellt. Betrachtet man hier folgende spezielle Wahl für c ,

$$c(\mathbf{x}) := \frac{R(\mathbf{x})}{T(\mathbf{x} - \mathbf{u}(\mathbf{x}))},$$

so wird die Distanz zu Null, sofern $T_{\mathbf{u}}$ von Null verschieden ist. Dies ist sicherlich nicht sehr sinnvoll, so dass über eine geeignete Regularisierung von c nachgedacht werden muss, um damit möglichst sinnvolle Korrekturterme zu ermöglichen.

Serienschnitte sehr gut einfangen, da neben einer Unempfindlichkeit gegenüber inhomogener Färbung gleichzeitig eine Homogenisierung stattfindet. Durch die simultane Optimierung bezüglich des Deformationsfeldes u und des Korrekturterms c werden nicht nur geometrische Veränderungen, sondern auch Veränderungen der Grauwerte zur Minimierung der Unterschiede berücksichtigt und damit die Wechselwirkungen von Homogenisierung und Registrierung berücksichtigt.

Im weiteren Vorgehen werden wir uns auf die Betrachtung des erweiterten Distanzmaßes $\mathcal{D}^{\text{cSSD}}$ beschränken, da es dem Problem, der Registrierung von Serienschritten, am besten entspricht.

4.1.2 Registrierungsproblem

In den letzten Abschnitten haben wir gesehen, dass Unterschiede von Bildern mit Hilfe eines Distanzmaßes gemessen werden. Wir verwenden hier das erweiterte SSD-Maß, in das neben den Bildern das Deformationsfeld und der Korrekturterm eingehen. Das Registrierungsproblem besteht nun darin, eine optimale Deformation u und einen optimalen Korrekturterm c zu finden, so dass das erweiterte SSD-Maß minimal wird.

Definition 4.1.5 (Elementares Registrierungsproblem)

Seien $R, T : \Omega \rightarrow \mathbb{G}$ Bilder, $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$ ein Deformationsfeld und $c : \Omega \rightarrow \mathbb{R}^+$ ein Korrekturterm. Dann wird mit

$$\mathcal{D}^{\text{cSSD}}[R, T; \mathbf{u}, c] = \frac{1}{2} \int_{\Omega} (cT_{\mathbf{u}} - R)^2 \, d\mathbf{x} \xrightarrow{\mathbf{u}, c} \min \quad (4.1)$$

das *elementare Registrierungsproblem* bzgl. \mathbf{u} und c bezeichnet.

Dieses Minimierungsproblem ist allerdings, wie wir u.a. im vorigen Abschnitt gesehen haben, nicht eindeutig lösbar und besitzt daher mehrere gleichwertige, im allgemeinen sogar unendlich viele Lösungen. Schauen wir dazu auf Abbildung 4.1. In der linken Abbildung sehen wir einen Kreis als Referenz- und Templatebild. Eine einfache Translation liefert ein optisch perfektes Ergebnis. Wird das Bild zusätzlich zur Translation um einen beliebigen Winkel gedreht, so entsteht erneut ein optisch perfektes, da identisches Ergebnis, trotz unterschiedlicher Transformation. In der rechten Abbildung reichen Rotationen um 45° , 135° , 225° oder 270° , um optisch identische und ebenfalls perfekte Ergebnisse zu liefern.

Hier geht es nicht darum, irgendeine Lösung zu finden, sondern eine Lösung, die aus Sicht der Serienschritte möglichst sinnvoll ist. Durch unseren variationellen Ansatz erweitern wir das Minimierungsproblem aus (4.1) additiv um so genannte *weiche Nebenbedingungen* (engl. *soft constraints*), die dafür sorgen, dass bestimmte Lösungen bevorzugt werden. In Kapitel 2 im Abschnitt 2.3 haben wir gesehen, dass sich die Deformation elastisch verhalten soll. Wir modellieren daher elastisches Verhalten, indem wir das lineare elastische Potential des Deformationsfeldes u betrachten.

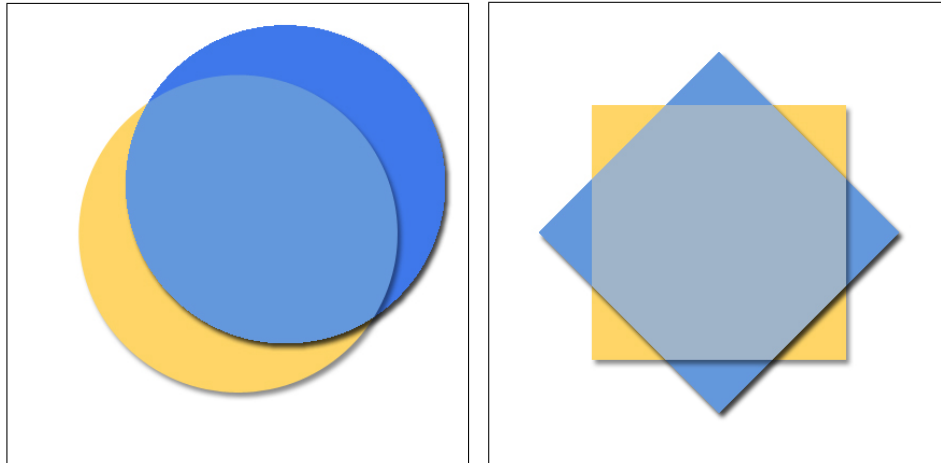


Abbildung 4.1: Zwei einfache Beispiele zur Veranschaulichung der uneindeutigen Lösbarkeit des Minimierungsproblems. *links:* Eine einfache Translation ist eine Lösung, doch jede zusätzliche Rotation um einen beliebigen Winkel erzeugt eine neue, optisch identische Lösung. *rechts:* Eine Rotation um 45° , 135° , 225° oder 270° erzeugt optisch ein identisches Ergebnis.

Definition 4.1.6 (Lineares elastisches Potential)

Sei $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$ ein Deformationsfeld, dann bezeichnet

$$\mathcal{S}^{\text{elast}}[\mathbf{u}] = \int_{\Omega} \frac{\mu}{4} \sum_{j,k=1}^d (\partial_{x_j} u_k + \partial_{x_k} u_j)^2 + \frac{\lambda}{2} (\text{div } \mathbf{u})^2 \, d\mathbf{x} \quad (4.2)$$

das *lineare elastische Potential* der Deformation \mathbf{u} .

$\mathcal{S}^{\text{elast}}[\mathbf{u}]$ drückt aus, wie groß die *innere Spannung* bei gegebener Deformation \mathbf{u} ist. Über die Konstanten μ und λ können die Materialeigenschaften bestimmt werden, wodurch diese Konstanten auch *Materialkonstanten* genannt werden (siehe dazu Abschnitt 2.3). Durch additive Hinzunahme dieses Terms zum elementaren Registrierungsproblem (siehe Def. 4.1.5) und gemeinsamer Minimierung ergibt sich die variationelle Formulierung der *elastischen Registrierung*, wie sie in [Bro81, Mod04, Wir03] verwendet wird.

Definition 4.1.7 (Elastisches Registrierungsproblem)

Seien $R, T : \Omega \rightarrow \mathbb{G}$ Bilder, $u : \Omega \rightarrow \mathbb{R}^d$ ein Deformationsfeld, $c : \Omega \rightarrow \mathbb{R}^+$ ein Korrekturterm und $\alpha \in \mathbb{R}^+$. Dann wird mit

$$\mathcal{D}^{\text{cSSD}}[R, T; \mathbf{u}, c] + \alpha \mathcal{S}^{\text{elast}}[\mathbf{u}] \xrightarrow{\mathbf{u}, c} \min$$

das *Elastische Registrierungsproblem* bzgl. \mathbf{u} und c bezeichnet.

Der Skalierungsfaktor α steuert hierbei den Einfluss des elastischen Potentials. Darüber lässt sich regeln, ob die Deformation eher die Bilder maximal ähnlich machen soll (kleines α), oder ob die Deformation möglichst geringe innere Spannungen des Materials erzeugen soll (großes α).

Bei der Suche nach einem geeigneten Korrekturterm soll ebenfalls nicht irgendeine Lösung gefunden werden, sondern eine, bei der die Färbeartefakte in den Bildern reduziert werden, während aber Kanten möglichst gut erhalten bleiben sollen. Wie wir in Abschnitt 3.3 gesehen haben, erfüllt die TV-Regularisierung genau diese Forderung, Färbeartefakte zu reduzieren und Kanten zu erhalten. Daher verwenden wir für die Regularisierung des Korrekturterms die *regularisierte Total-Variation-Norm* (TV-Norm, TV-Regularisierung) [CGM99, ROF92, VO96], die, angewendet auf (nahezu) homogene Bereiche, diffusiv glättet, aber Kanten dabei erhält.

Als Regularisierer für den Korrekturterm verwenden wir die regularisierte TV-Norm, die auf Seite 36 in Definition 3.3.4 eingeführt wurde. Wir bezeichnen sie fortwährend mit

$$\begin{aligned} \mathcal{S}^{\text{TV}}[c] &= \int_{\Omega} \Psi_{\varepsilon}(|\nabla c|_2^2) \, d\mathbf{x}, \quad \text{mit } \Psi_{\varepsilon}(s^2) = \sqrt{s^2 + \varepsilon^2}, \quad (4.3) \\ &= \int_{\Omega} \sqrt{|\nabla c|_2^2 + \varepsilon^2} \, d\mathbf{x}, \quad 0 < \varepsilon \ll 1. \end{aligned}$$

Die Regularisierung der TV-Norm erfolgt durch die in Definition 4.1.2 eingeführte Penalizer-Funktion Ψ_{ε} . Wie bereits in der Definition von Penalizer-Funktionen erwähnt, können hier neben der fest gewählten Abbildung Ψ_{ε} auch andere Abbildungen Ψ verwendet werden, die monoton wachsend und stetig differenzierbar sind. Wählt man zum Beispiel $\Psi(s^2) = s^2$, so erhält man die TV-Norm, die in [ROF92] ausführlich diskutiert wurde.

Die Regularisierung des Korrekturterms wird ebenfalls additiv zum Minimierungsproblem hinzugefügt und gemeinschaftlich minimiert. Die drei Teile Distanzmaß, Regularisierer des Deformationsfeldes und Regularisierer des Korrekturterms fassen wir zu einem Funktional zusammen, um später das Registrierungsproblem für Serienschritte kompakter formulieren zu können. Die einzelnen Summanden zur Regularisierung werden jeweils mit einem Faktor α, β versehen, der den jeweiligen Einfluss des Regularisierers zum gesamten Funktional regelt. Damit kann beispielsweise gesteuert werden, ob das Bild stärker durch Deformation (großes β)

oder Umfärben (kleines β) verändert werden soll, bzw. ob mehr Wert auf glatte Deformationsfelder (grosses α) oder reine Ähnlichkeit (kleines α) gelegt werden soll.

Schließlich fassen wir alle Teile zusammen und formulieren das endgültige Minimierungsproblem.

Definition 4.1.8 (Registrierungsproblem)

Seien $R, T : \Omega \rightarrow \mathbb{G}$, $R, T \in C^2(\Omega)$ Bilder, $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$, $\mathbf{u} \in C^2(\Omega)$ ein Deformationsfeld und $c : \Omega \rightarrow \mathbb{R}^+$, $c \in C^2(\Omega)$ ein Korrekturterm und $\alpha, \beta \in \mathbb{R}^+$. Die Minimierung des Funktionals \mathcal{J} bzgl. \mathbf{u} und c ,

$$\mathcal{J}[R, T; \mathbf{u}, c] := \mathcal{D}^{\text{cSSD}}[R, T; \mathbf{u}, c] + \alpha \mathcal{S}^{\text{elast}}[\mathbf{u}] + \beta \mathcal{S}^{\text{TV}}[c] \xrightarrow{\mathbf{u}, c} \min,$$

bezeichnen wir als *Registrierungsproblem zur Simultanen Homogenisierung und Registrierung*.

Um dieses Registrierungsproblem zu lösen, berechnen wir stationäre Punkte des Funktionals \mathcal{J} (vgl. Anhang A). Dabei verwenden wir die dort eingeführten Hilfsmittel. Stationäre Punkte können durch Nullsetzen der Gâteaux-Ableitung von \mathcal{J} berechnet werden,

$$d\mathcal{J}[R, T; \mathbf{u}, c] = \begin{pmatrix} \frac{d}{d\mathbf{u}} \mathcal{J}[R, T; \mathbf{u}, c] \\ \frac{d}{dc} \mathcal{J}[R, T; \mathbf{u}, c] \end{pmatrix} = \mathbf{0}. \quad (4.4)$$

Da Differentialoperatoren lineare Operatoren sind, kann die Summe des Funktionals auseinander gezogen und der Differentialoperator auf jeden einzelnen Summanden $\mathcal{D}^{\text{cSSD}}$, $\mathcal{S}^{\text{elast}}$ und \mathcal{S}^{TV} einzeln angewendet werden,

$$\begin{pmatrix} \frac{d}{d\mathbf{u}} \mathcal{J} \\ \frac{d}{dc} \mathcal{J} \end{pmatrix} = \begin{pmatrix} \frac{d}{d\mathbf{u}} \mathcal{D}^{\text{cSSD}} + \alpha \frac{d}{d\mathbf{u}} \mathcal{S}^{\text{elast}} + \beta \frac{d}{d\mathbf{u}} \mathcal{S}^{\text{TV}} \\ \frac{d}{dc} \mathcal{D}^{\text{cSSD}} + \alpha \frac{d}{dc} \mathcal{S}^{\text{elast}} + \beta \frac{d}{dc} \mathcal{S}^{\text{TV}} \end{pmatrix} = \mathbf{0}.$$

Demnach können wir die Gâteaux-Ableitung eines jeden Summanden einzeln ausrechnen, um so die Ableitung des gesamten Funktionals, und damit die Euler-Lagrange-Gleichungen, zu bekommen.

4.2 Herleitung der Euler-Lagrange-Gleichungen

Wie wir im vorigen Abschnitt gesehen haben, lässt sich das Registrierungsproblem als Minimierungsproblem darstellen. Die Ableitung des Funktionals kann dann in drei Teile aufgeteilt werden, die zur analytischen Bearbeitung unabhängig voneinander betrachtet werden können. Im Anhang B stellen wir die Gâteaux-Ableitung des Distanzmaßes und den Regularisierer für \mathbf{u} und c ausführlich vor. Da aufgrund der Länge die Übersicht leicht verloren geht, präsentieren wir nachfolgend

die entscheidenden Teilergebnisse mit deren Hilfe wir die Ableitung von \mathcal{J} zusammensetzen können. Damit können wir schließlich die Euler-Lagrange-Gleichungen des erweiterten Registrierungsproblems formulieren.

Aus der Variationsrechnung (siehe Anhang A.2) wissen wir, dass sich zu einer Funktion

$$f(\mathbf{u}) = \int_{\Omega} F(x, y, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_y) \, dx dy, \quad \mathbf{u} \in V = \{\mathbf{v} \in C^2(\Omega)\},$$

mit einer zweimal stetig differenzierbaren Abbildung $F(x, y, r, s, t)$ eine notwendige Bedingung für ein Minimum aus einer Nullstelle der Gâteaux-Ableitung herleiten lässt. Verwenden wir nun Dirichlet-Randbedingungen, so liefert

$$\begin{aligned} \frac{\partial F}{\partial \mathbf{u}} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial \mathbf{u}_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial \mathbf{u}_y} \right) &= 0 \quad \text{in } \Omega \\ \text{und } \mathbf{u} &= \alpha \quad \text{auf } \Gamma \end{aligned}$$

eine notwendige Bedingung für ein Minimum von $f(\mathbf{u})$.

In diesem Sinne seien nun

$$f_1(\mathbf{u}) = \frac{1}{2} \int_{\Omega} (c(\mathbf{x})T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}))^2 \, d\mathbf{x}, \quad (\text{mit bel., aber festem } c),$$

$$f_2(c) = \frac{1}{2} \int_{\Omega} (c(\mathbf{x})T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}))^2 \, d\mathbf{x}, \quad (\text{mit bel., aber festem } \mathbf{u}),$$

$$f_3(\mathbf{u}) = \int_{\Omega} \frac{\mu}{4} \sum_{j,k=1}^d (\partial_{x_j} u_k \partial_{x_k} u_j)^2 + \frac{\lambda}{2} (\operatorname{div} \mathbf{u})^2 \, d\mathbf{x}, \quad \text{und}$$

$$f_4(c) = \int_{\Omega} \Psi_{\varepsilon} (|\nabla c|^2) \, d\mathbf{x}$$

Funktionen, wie sie oben beschrieben wurden. Dann folgen als Nullstellen der Gâteaux-Ableitungen, bei Dirichlet-Randbedingungen

$$f_1 : \quad (c(\mathbf{x})T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}))c(\mathbf{x})\nabla T(\mathbf{x} - \mathbf{u}(\mathbf{x})) = 0,$$

$$f_2 : \quad (c(\mathbf{x})T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}))T(\mathbf{x} - \mathbf{u}(\mathbf{x})) = 0,$$

$$f_3 : \quad \mu \Delta \mathbf{u} + (\mu + \lambda) \nabla \operatorname{div} \mathbf{u} = 0, \quad \text{und}$$

$$f_4 : \quad \nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}} = 0.$$

Fassen wir nun alles zusammen,

$$\begin{aligned}
 0 &= \begin{pmatrix} \frac{d\mathbf{u}}{d\mathbf{u}}\mathcal{J}[R, T; \mathbf{u}, c] \\ \frac{dc}{dc}\mathcal{J}[R, T; \mathbf{u}, c] \end{pmatrix} \\
 &= \begin{pmatrix} \frac{d}{d\mathbf{u}}\mathcal{D}^{\text{cSSD}}[R, T; \mathbf{u}, c] + \alpha \frac{d}{d\mathbf{u}}\mathcal{S}^{\text{elast}}[\mathbf{u}] + \beta \frac{d}{d\mathbf{u}}\mathcal{S}^{\text{TV}}[c] \\ \frac{d}{dc}\mathcal{D}^{\text{cSSD}}[R, T; \mathbf{u}, c] + \alpha \frac{d}{dc}\mathcal{S}^{\text{elast}}[\mathbf{u}] + \beta \frac{d}{dc}\mathcal{S}^{\text{TV}}[c] \end{pmatrix} \\
 &= \begin{pmatrix} (cT_{\mathbf{u}} - R)c\nabla T_{\mathbf{u}} + \alpha(\mu\Delta\mathbf{u} + (\lambda + \mu)\nabla\text{div}\mathbf{u}) + \beta \cdot 0 \\ (cT_{\mathbf{u}} - R)T_{\mathbf{u}} + \alpha \cdot 0 + \beta \nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}} \end{pmatrix},
 \end{aligned}$$

so ergeben sich die Euler-Lagrange-Gleichungen des erweiterten Registrierungsproblems

$$\begin{aligned}
 \alpha(\mu\Delta\mathbf{u} + (\lambda + \mu)\nabla\text{div}\mathbf{u}) &= -(cT_{\mathbf{u}} - R)c\nabla T_{\mathbf{u}} \\
 \text{und} \quad cT_{\mathbf{u}}^2 + \beta \nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}} &= RT_{\mathbf{u}}. \tag{4.5}
 \end{aligned}$$

mit $\mathbf{u}(\Gamma) = 0$ und $c(\Gamma) = 1$.

Bei diesen Euler-Lagrange-Gleichungen handelt es sich um ein System nichtlinearer partieller Differentialgleichungen. Um eine bessere Übersicht zu bekommen, möchten wir einige neue Begriffe definieren.

Definition 4.2.1 (Elastizitätsoperator)

Sei $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$ ein Deformationsfeld, $\alpha \in \mathbb{R}^+$ und $\mu, \lambda \in \mathbb{R}^+$ Materialkonstanten. Dann bezeichnen wir \mathcal{A}_α , mit

$$\mathcal{A}_\alpha \mathbf{u} := \alpha(\mu\Delta\mathbf{u} + (\lambda + \mu)\nabla\text{div}\mathbf{u}) \tag{4.6}$$

als *Elastizitätsoperator*.

Der Elastizitätsoperator gibt die innere Spannung bei vorgegebenem Material und der Deformation an.

Ebenso wie für die Wirkung der Deformation ein Operator eingeführt wurde, wollen wir nun einen Operator für die Wirkung des Korrekturterms einführen.

Definition 4.2.2 (TV-Operator)

Sei $T : \Omega \rightarrow \mathbb{G}$ ein Bild, $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$ ein Deformationsfeld, $c : \Omega \rightarrow \mathbb{R}^+$ ein Korrekturterm, $\beta \in \mathbb{R}^+$ und $\varepsilon \ll 1 \in \mathbb{R}^+$, dann bezeichnen wir \mathcal{B}_β , mit

$$\mathcal{B}_\beta[T_{\mathbf{u}}](c) := cT_{\mathbf{u}}^2 + \beta \nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}} \quad (4.7)$$

als *TV-Operator*.

Die Anwendung des TV-Operators führt eine diffusive Glättung des Bildes unter Kantenerhaltung durch.

In der Physik wird die rechte Seite einer Differentialgleichung meist *Kraft* genannt. So wollen auch wir die rechte Seite der Differentialgleichungen Kraft nennen. Dabei gibt es eine Kraft, die die Registrierung über die Deformation antreibt und eine Kraft, die inhomogene Bereiche der Bilder aufzeigt.

Definition 4.2.3 (Kraft)

Seien $R, T : \Omega \rightarrow \mathbb{G}$ Bilder, $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$ ein Deformationsfeld und $c : \Omega \rightarrow \mathbb{R}^+$ ein Korrekturterm. Dann führen wir als rechte Seite der Differentialgleichungen die Kräfte f und g ein, mit

$$f[R, T; \mathbf{u}, c] := -(cT_{\mathbf{u}} - R)c\nabla T_{\mathbf{u}} \quad (4.8)$$

und

$$g[R, T; \mathbf{u}] := RT_{\mathbf{u}}. \quad (4.9)$$

Mit diesen neuen Begriffen, lassen sich die Euler-Lagrange-Gleichungen (4.5) wesentlich übersichtlicher aufschreiben und in eine Form bringen, die an lineare Gleichungssysteme erinnert.

Satz 4.2.4 (Euler-Lagrange-Gleichungen Registrierungsproblem)

Seien $R, T \in C^2(\Omega) : \Omega \rightarrow \mathbb{G}$ Bilder, $\mathbf{u} \in C^2(\Omega) : \Omega \rightarrow \mathbb{R}^d$ ein Deformationsfeld und $c \in C^2(\Omega) : \Omega \rightarrow \mathbb{R}^+$ ein Korrekturterm. Ein stationärer Punkt des Registrierungsproblems aus Definition 4.1.8 ist Lösung der Euler-Lagrange-Gleichungen

$$\begin{aligned} \mathcal{A}_\alpha \mathbf{u} &= f[R, T; \mathbf{u}, c] \\ \text{und } \mathcal{B}_\beta[T_{\mathbf{u}}](c) &= g[R, T; \mathbf{u}] \end{aligned} \quad (4.10)$$

mit Dirichlet-Randbedingungen $\mathbf{u}(\Gamma) = 0$ und $c(\Gamma) = 1$.

Hierbei ist die äußere Form gleichzeitig Programm. Die Lösung des Systems (4.10)

lässt sich im allgemeinen nicht analytisch bestimmen. Daher möchten wir eine numerische Lösung finden. Im nächsten Schritt werden wir aus dem System nichtlinearer partieller Differentialgleichungen ein System von linearen partiellen Differentialgleichungen machen und dafür eine bestimmte Form der Linearisierung vorstellen.

Bisher haben wir Dirichlet-Randbedingungen verwendet, da sie nicht nur sehr gut nachvollziehbar sind, sondern auch die Variationsrechnung vereinfacht haben. Von der Problemstellung her sind sie ebenfalls gut geeignet. Im nachfolgenden Teil der Arbeit möchten wir allerdings gerne auf periodische Randbedingungen wechseln, da dadurch die Nutzung eines sehr effizienten Löser für die elastische partielle Differentialgleichung möglich wird. Damit lösen wir zwar nicht mehr das ursprünglich formulierte Registrierungsproblem, doch die damit ausgerechneten Lösungen sind sehr dicht dran, da wir bei den histologischen Serienschritten spezielle Voraussetzungen haben. Das zu registrierende Objekt hat grundsätzlich viel Rand um sich herum. Darüber hinaus wird affin vorregistriert, so dass nur noch lokale Deformationen gesucht werden, die das Objekt selbst kaum verlassen werden. Damit entsteht am Rand des Bildes automatisch $\mathbf{u} = 0$ bzw. $c = 1$, was wiederum Dirichlet-Randbedingungen sind. Würden Dirichlet oder Neumann Randbedingungen verwendet werden, so würde dies im weiteren Verlauf auf andere Gleichungssysteme führen, die zur Lösung anders behandelt werden müssen. Dazu können beispielsweise Mehrgitterverfahren herangezogen werden.

Durch die Wahl von periodischen Randbedingungen und der Diskretisierung durch Finite Differenzen (siehe Kapitel 5) wird aus dem Differentialgleichungssystem ein Differenzensystem und damit ein lineares Gleichungssystem, welches sich dann numerisch lösen lässt.

4.3 Linearisierung der nichtlinearen partiellen Differentialgleichungen durch Fixpunktiteration

Wie wir im vorigen Abschnitt gesehen haben, berechnen wir einen stationären Punkt des Funktionals. Fixpunktverfahren konvergieren, wenn sie konvergieren, gegen stationäre Punkte. Daher ist es nahe liegend für die Linearisierung des Verfahrens eine Fixpunktiteration zu verwenden.

Bei der Fixpunktiteration wird ein Iterationszähler, ähnlich einer künstlichen Zeit eingeführt, um die gesuchte Variable in alte Iteration (schon bekannt) und neue Iteration (noch nicht bekannt) aufzuteilen. Betrachten wir nun wieder System (4.10),

$$\mathcal{A}_\alpha \mathbf{u} = f[R, T; \mathbf{u}, c]$$

und

$$\mathcal{B}_\beta[T_\mathbf{u}](c) = g[R, T; \mathbf{u}],$$

lassen aber der Übersicht halber die Formulierung der Randbedingungen hier weg. An notwendiger Stelle werden wir sie wiederum aufnehmen.

Prinzipiell gibt es nun zwei Möglichkeiten der Linearisierung dieses Systems. Im ersten Fall betrachtet man das gesamte System und linearisiert über beide Gleichungen,

$$\begin{aligned}\mathcal{A}_\alpha \mathbf{u}^{(k+1)} &= f[R, T; \mathbf{u}^{(k)}, c^{(k)}], \\ \mathcal{B}_\beta [T_{\mathbf{u}^{(k)}}] (c^{(k+1)}) &= g[R, T; \mathbf{u}^{(k)}],\end{aligned}\tag{4.11}$$

mit $k \in \mathbb{N}$, $\mathbf{u}^{(0)} \equiv \mathbf{0}$ und $c^{(0)} \equiv \mathbf{1}$. Dadurch erhält man ein partielles Differentialgleichungssystem, das in jedem Iterationsschritt gelöst werden muss. Die Lösung der zweiten Differentialgleichung ist in dieser Form allerdings sehr schwierig, da im Quotienten c sowohl im Zähler, als auch im Nenner auftaucht,

$$\mathcal{B}_\beta [T_{\mathbf{u}^{(k)}}] (c^{(k+1)}) = c^{(k+1)} T_{\mathbf{u}^{(k)}}^2 + \beta \nabla \cdot \frac{\nabla c^{(k+1)}}{\sqrt{|\nabla c^{(k+1)}|_2^2 + \varepsilon^2}}$$

und damit in $c^{(k+1)}$ nicht-linear ist. Durch eine kleine Abwandlung lässt sich \mathcal{B} auch in $c^{(k)}$ linearisieren (der Leser achte besonders auf den Iterationsindex)

$$\mathcal{B}_\beta [T_{\mathbf{u}^{(k)}}] (c^{(k+1)}) = c^{(k)} T_{\mathbf{u}^{(k)}}^2 + \beta \nabla \cdot \frac{\nabla c^{(k+1)}}{\sqrt{|\nabla c^{(k)}|_2^2 + \varepsilon^2}}.$$

Damit vereinfacht sich das System $\mathcal{B}_\beta [T_{\mathbf{u}^{(k)}}] (c^{(k+1)}) = g[R, T; \mathbf{u}^{(k)}]$ zu

$$\beta \nabla \cdot \nabla c^{(k+1)} = \beta \Delta c^{(k+1)} = \sqrt{|\nabla c^{(k)}|_2^2 + \varepsilon^2} (RT_{\mathbf{u}^{(k)}} - c^{(k)} T_{\mathbf{u}^{(k)}}^2).$$

Der Vorteil bei dieser Linearisierung ist, dass die Kopplung von \mathbf{u} und c implizit stattfindet und damit in einem Berechnungsschritt beide Abbildungen aktualisiert werden. Der Nachteil hierbei ist allerdings, dass die Lösung des Systems sehr aufwändig ist und der Speicherbedarf zu groß werden kann.

Die genannten Schwierigkeiten legen einen kleinen Trick aus der Numerik nahe, die Lösung der Systeme zu entkoppeln und nur die Wirkung zu übertragen, was uns zur zweiten Variante der Linearisierung führt. In diesem Fall werden beide Gleichungen des Systems getrennt voneinander betrachtet und gelöst, d.h. im ersten Schritt wird

$$\mathcal{A}_\alpha \mathbf{u}^{(k+1)} = f[R, T; \mathbf{u}^{(k)}, c^{(k)}]$$

und im nächsten Schritt, da $\mathbf{u}^{(k+1)}$ bereits bekannt ist, wird

$$\mathcal{B}_\beta [T_{\mathbf{u}^{(k+1)}}] (c^{(k+1)}) = g[R, T; \mathbf{u}^{(k+1)}]$$

gelöst. Der Hauptvorteil dieser Trennung besteht in der Tatsache, dass für beide Teilsysteme jeweils effiziente Löser verwendet werden können, die beispielsweise die spezielle Struktur der Teilblöcke ausnutzen, ohne, dass sie vom anderen System wissen müssen. Eine Kopplung der Gleichungen zu einem Gesamtsystem

wird über die Iterationen durch die Einbindung von c in der ersten und \mathbf{u} in der zweiten Gleichung gewährleistet. Der Nachteil dieser Variante ist, dass die Bindung von \mathbf{u} und c nicht so stark ist wie in der ersten Variante. Aufgrund der extremen Datengröße von Bildern histologischer Serienschritte und der Möglichkeit des Einsatzes von effizienten Lösern für die Teilsysteme, wählen wir die zweite Variante der Linearisierung.

4.4 Erweiterung der Methode für Serienschritte

In den letzten Abschnitten haben wir das Registrierungsproblem für die Registrierung von zwei Bildern eingeführt und erläutert, um so ein direktes und allgemeines Verständnis zu ermöglichen. Bei den histologischen Serienschritten handelt es sich in der Regel aber nicht um zwei Bilder, sondern um eine ganze Serie von Bildern. In diesem Abschnitt möchten wir daher eine Erweiterung der bisher vorgestellten Methode auf die Registrierung von Bildern von Serienschritten vorstellen. Bevor wir nun zwei verschiedene Strategien vorstellen, klären wir den Begriff der Serie von Bildern.

Definition 4.4.1 (Serie von Bildern, Bildserie)

Sei $M \in \mathbb{N}$ die Anzahl der Bilder der Serie, $\Omega = \Omega_1 \cup \dots \cup \Omega_M \subset \mathbb{R}^d$, $d \in \mathbb{N}$ das Gebiet auf denen Bilder definiert sind und $\mathbb{G} \subset \mathbb{R}$ die Menge der Grau- bzw. Intensitätswerte der Bilder. Sind des weiteren R^1, \dots, R^M mit $R^\nu : \Omega_\nu \rightarrow \mathbb{G}$, $\nu = 1, \dots, M$ Bilder auf Ω in einer festen Reihenfolge, so nennt man

$$\mathbf{R} := (R^1, \dots, R^M) \quad \text{bzw.} \quad \mathbf{R}(\mathbf{x}) := (R^1(\mathbf{x}), \dots, R^M(\mathbf{x})),$$

mit $\mathbf{x} \in \Omega$, eine *Serie von Bildern* oder auch *Bildserie*.

Zu einer Serie von Bildern gehören entsprechend eine Serie von Deformationsfeldern $(\mathbf{u}^1(\mathbf{x}), \dots, \mathbf{u}^M(\mathbf{x}))$, mit $\mathbf{u}^\nu : \Omega \rightarrow \mathbb{R}^d$, für alle $\nu = 1, \dots, M$ und eine Serie von Korrekturtermen $(c^1(\mathbf{x}), \dots, c^M(\mathbf{x}))$, mit $c^\nu : \Omega \rightarrow \mathbb{R}^+$, für alle $\nu = 1, \dots, M$. Hierbei sind die \mathbf{u}^ν Deformationsfelder wie in Def. 2.1.2 und die c^ν Korrekturterme wie in Def. 4.1.3.

Beginnen wollen wir mit einer sehr nahliegenden Strategie, bei der die Bilder der Reihe nach, paarweise aufeinander folgend, beginnend beim ersten Bild, zweidimensional registriert werden (siehe dazu Algorithmus 4.1).

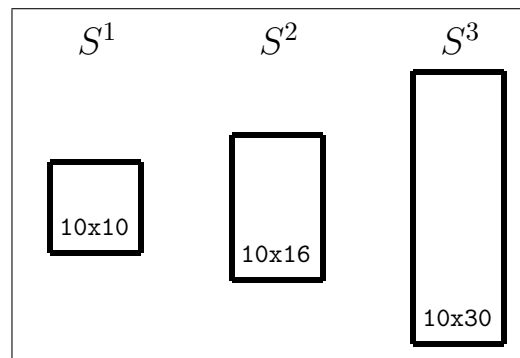
Dieser Algorithmus ist leicht zu implementieren und verwendet nur eine zweidimensionale Registrierung. Für das erste Bild der Serie wird weder ein Deformationsfeld, noch ein Korrekturterm ausgerechnet. Dadurch zählt es als Grundreferenz für alle nachfolgenden Bilder der Serie. Der wesentliche Nachteil dieses Ansatzes ist, dass die Nachbarschaftsverhältnisse der einzelnen Bilder nicht mit eingehen, d.h. die Deformation eines Bildes hat keinen Einfluss auf die Nachbarbilder. Be-

Algorithmus 4.1 Einfache Strategie zur Registrierung von Bildserien

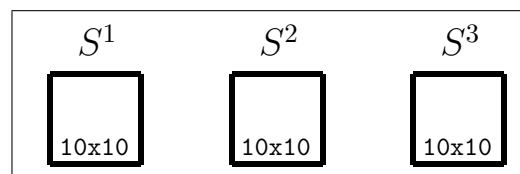
$\mathbf{R} = (R^1, \dots, R^M)$ Bildserie aus M vielen Bildern
 $\mathbf{u}^\nu \in C^2(\Omega_\nu) : \Omega_\nu \rightarrow \mathbb{R}^d, \nu = 1, \dots, M$ Deformationsfelder
 $c^\nu \in C^2(\Omega_\nu) : \Omega_\nu \rightarrow \mathbb{R}, \nu = 1, \dots, M$ Korrekturterme

- 1: $\mathbf{u}^1 \leftarrow \mathbf{0}$;
- 2: $c^1 \leftarrow \mathbf{1}$
- 3: **for** $\nu = 2$ to M **do**
- 4: $R \leftarrow S^{\nu-1}$;
- 5: $T \leftarrow S^\nu$;
- 6: $\mathbf{u}^\nu, c^\nu \leftarrow \text{registration2D}(R, T)$;
- 7: **end for**

trachten wir folgende Bildserie:



Die Bildserie besteht aus drei Bildern S^1 bis S^3 . Die Rechtecke repräsentieren schematisch die Außenkontur der Objekte auf den Bildern. Innerhalb der Rechtecke ist ihre Ausbreitung in Einheiten abgebildet. Eine perfekte Registrierung der Serie nach Algorithmus 4.1 würde folgendes Ergebnis liefern:



Die Objekte auf den Bildern hätten nach der Registrierung alle die gleiche Kontur. Für Bild S^2 und S^3 bedeutet dies eine extreme Schrumpfung. Insgesamt würde bei einer perfekten Registrierung nach Algorithmus 4.1 ein Zylinder entstehen, bei dem alle Bilder gleich aussehen.¹ Praktisch gesehen würden damit alle individuellen Unterschiede innerhalb der Bilder verloren gehen. Bei der Registrierung von

¹Zur Veranschaulichung des ganzen, hier ein kleiner Vergleich. Würde man einen Laib Brot zerschneiden und nach obigem Algorithmus registrieren, entstünde von der äußeren Form her ein Kastenbrot.

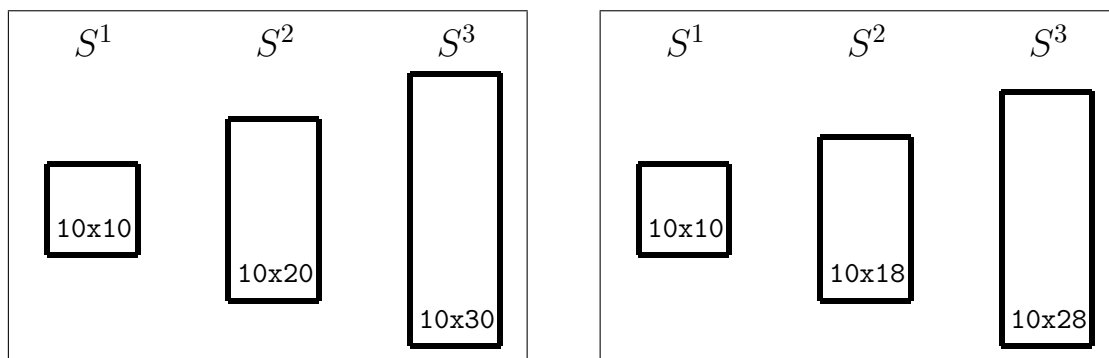
Bildserien, bei denen die einzelnen Bilder deutliche strukturelle Unterschiede aufweisen ist diese Methode daher ungeeignet. In manchen Anwendungsfällen kann dieser Effekt sicherlich zu guten Ergebnissen führen. Hier wird der Algorithmus allerdings nicht weiter betrachtet.

Algorithmus 4.2 Kopplungsstrategie zur Registrierung von Bildserien

$\mathbf{R} = (R^1, \dots, R^M)$ Bildserie aus M vielen Bildern
 $\mathbf{u}^\nu \in C^2(\Omega_\nu) : \Omega_\nu \rightarrow \mathbb{R}^d, \nu = 1, \dots, M$ Deformationsfelder
 $c^\nu \in C^2(\Omega_\nu) : \Omega_\nu \rightarrow \mathbb{R}, \nu = 1, \dots, M$ Korrekturterme

- 1: $\mathbf{u}^1, \mathbf{u}^M \leftarrow \mathbf{0}$;
 - 2: $c^1, c^M \leftarrow \mathbf{1}$;
 - 3: **for** $\nu = 2$ to $M-1$ **do**
 - 4: $R \leftarrow \frac{1}{2}(S^{\nu-1} + S^{\nu+1})$;
 - 5: $T \leftarrow S^\nu$;
 - 6: $\mathbf{u}^\nu, c^\nu \leftarrow \text{registration2D}(R, T)$;
 - 7: **end for**
-

Durch eine Kopplung der Bilder wie in Algorithmus 4.2 kann die Entstehung eines Zylinders durch die Registrierung verhindert und die Auswirkung der Deformation eines Bildes auf die Nachbarbilder übertragen werden. In dieser Variante wird das erste und letzte Bild der Serie unverändert gelassen, wodurch die Bilder dazwischen derart registriert werden, dass sie zu beiden Seiten gut passen und nicht nur zu einer, wie bei Algorithmus 4.1. Bei der Registrierung mit Kopplung der Bilder wird ein Bild nicht ausschliesslich relativ zum vorangegangenen Bild registriert, sondern relativ zu den direkten Nachbarbildern. Damit hat eine Deformation in einem Bild automatisch Auswirkungen auf die Deformation der Nachbarbilder.



Im direkten Vergleich mit der Ausgangssituation sehen wir in der linken Darstellung, dass das erste und dritte Bild gleich geblieben, während sich das zweite Bild so vergrößert hat, dass ihre Länge der mittleren Länge der Nachbarbilder entspricht. Eine Variante von Algorithmus 4.2 ist es, nur ein Bild der Serie unverändert zu lassen. In obiger Darstellung rechts bleibt S^1 unverändert. Dadurch

dass S^3 nun nicht mehr unveränderbar ist, schrumpft es und damit wird S^2 auch weiter zusammenschrumpfen, als im ersten Fall.

Für die Rekonstruktion histologischer Serienschritte ist Algorithmus 4.2 die geeignetere Variante. Nachdem die Begriffe Bild, Deformationsfeld und Korrekturterm schon im Zusammenhang der Bildserien klar geworden sind, möchten wir nun die Begriffe *erweitertes Distanzmaß*, *lineares elastische Potential*, *regularisierte TV-Norm* und *Registrierungsproblem* auf die Bildserien übertragen. Im Gegensatz zur bisherigen Sicht auf die Registrierung werden bei der Registrierung von Bildserien nicht zwei Serien aufeinander registriert, sondern die Bilder der Serie untereinander. Das Distanzmaß misst damit nicht die Distanz zu einem anderen Datensatz, sondern die Distanz der Bilder innerhalb der Serie. Aus analogischen Gründen wurde hier der gleiche Begriff gewählt.

Definition 4.4.2 (SSD-Distanz einer Serie von Bildern)

Sei $\mathbf{R} := (R^1, \dots, R^M)$ eine Bildserie auf Ω . Dann wird mit

$$\begin{aligned} \mathcal{D}^{\text{SSD}}[\mathbf{R}] &:= \sum_{\nu=2}^M \mathcal{D}^{\text{SSD}}[R^\nu, R^{\nu-1}] \\ &= \frac{1}{2} \sum_{\nu=2}^M \int_{\Omega} (R^\nu(\mathbf{x}) - R^{\nu-1}(\mathbf{x}))^2 \, d\mathbf{x} \end{aligned}$$

die *SSD-Distanz einer Serie von Bildern* bezeichnet.

Dementsprechend ist die Erweiterung auf das *erweiterte Distanzmaß* zu realisieren. Im Gegensatz zur klassischen Registrierung, bei dem ein Bild deformiert und grauwertkorrigiert wird, werden diese Korrekturen bei der Registrierung von Bildserien auf alle Bilder der Serie angewendet.

Definition 4.4.3 (Erweitertes Distanzmaß für Bildserien)

Sei $\mathbf{R} := (R^1, \dots, R^M)$ eine Bildserie, $\mathbf{U} := (\mathbf{u}^1, \dots, \mathbf{u}^M)$ eine Serie von Deformationsfeldern und $\mathbf{c} := (c^1, \dots, c^M)$ eine Serie von Korrekturtermen. Dann wird

$$\begin{aligned} \mathcal{D}^{\text{cSSD}}[\mathbf{R}; \mathbf{U}, \mathbf{c}] &:= \sum_{\nu=2}^M \mathcal{D}^{\text{SSD}}[c^\nu R_{\mathbf{u}^\nu}^\nu, c^{\nu-1} R_{\mathbf{u}^{\nu-1}}^{\nu-1}] = \\ &\frac{1}{2} \sum_{\nu=2}^M \int_{\Omega} \left(c^\nu(\mathbf{x}) R^\nu(\mathbf{x} - \mathbf{u}^\nu(\mathbf{x})) - c^{\nu-1}(\mathbf{x}) R^{\nu-1}(\mathbf{x} - \mathbf{u}^{\nu-1}(\mathbf{x})) \right)^2 \, d\mathbf{x} \end{aligned}$$

erweitertes Distanzmaß für Bildserien mit Grauwertkorrektur genannt.

Im Integranden werden zwei Bilder in Bezug zueinander gesetzt. Durch die Sum-

me werden alle Bilder der Serie miteinander gekoppelt, so dass Veränderungen durch Deformationsfelder oder Korrekturterme mit jeder Iteration Auswirkungen auf weitere Bilder der Serie haben.

Wie wir in Definition 4.2.3 eingeführt haben, bezeichnen wir die Ableitung des Distanzmaßes als treibende *äußere Kraft* der Registrierung. Durch die Erweiterung des Distanzmaßes auf Bildserien sind damit die äußeren Kräfte eindeutig beschrieben. Kommen wir nun zu den inneren Kräften der Registrierung. Sowohl die Deformationen, als auch die Korrekturterme müssen regularisiert werden, wodurch die entgegen wirkenden *inneren Kräfte* beschrieben werden. Wie wir es bei den Distanzmaßen in Def. 4.4.3 gemacht haben, durch eine Summation den Zusammenhang unter den Bildern herstellen, werden wir es ebenso bei den Deformationsfeldern und Korrekturtermen machen.

Zur Regularisierung der Deformationen \mathbf{U} betrachten wir wieder das lineare elastische Potential der Deformationsfelder (siehe Def. 4.1.6) und koppeln diese durch eine Summation untereinander.

Definition 4.4.4 (Lineares elastisches Potential für Serien)

Sei $\mathbf{U} := (\mathbf{u}^1, \dots, \mathbf{u}^M)$ eine Serie von Deformationen. Dann bezeichnet

$$\begin{aligned} \mathcal{S}^{\text{elast}}[\mathbf{U}] &= \sum_{\nu=1}^M \mathcal{S}^{\text{elast}}[\mathbf{u}^\nu] \\ &= \sum_{\nu=1}^M \int_{\Omega} \frac{\mu}{4} \sum_{j,k=1}^2 (\partial_{x_j} u_k^\nu + \partial_{x_k} u_j^\nu)^2 + \frac{\lambda}{2} (\text{div } \mathbf{u}^\nu)^2 \, d\mathbf{x} \end{aligned}$$

das *lineare elastische Potential der Serie von Deformationsfeldern*.

Im Gegensatz zum Distanzmaß ist hier pro Summand nur ein einzelnes Deformationsfeld notiert, wodurch die individuelle Deformation des einzelnen Bildes gewährleistet bleibt. Die Gesamtwirkung wird allerdings über die Summation eingefangen.

Die Regularisierung der Korrekturterme \mathbf{c} erfolgt wie in Def. 3.3.4 durch die regularisierte TV-Norm mit der Kopplung durch die Summation.

Definition 4.4.5 (Regularisierte TV-Norm für Serien)

Sei $\mathbf{c} := (c^1, \dots, c^M)$ eine Serie von Korrekturtermen und $\Psi_\varepsilon = \sqrt{s^2 + \varepsilon^2}$ eine Penalizer-Funktion. Dann definiert sich die *regularisierte TV-Norm* für eine Serie von Korrekturtermen durch

$$\begin{aligned} \mathcal{S}^{\text{TV}}[\mathbf{c}] &= \sum_{\nu=1}^M \mathcal{S}^{\text{TV}}[c^\nu] \\ &= \sum_{\nu=1}^M \int_{\Omega} \Psi_\varepsilon (|\nabla c^\nu|_2^2) \, d\mathbf{x} \\ &= \sum_{\nu=1}^M \int_{\Omega} \sqrt{|\nabla c^\nu|_2^2 + \varepsilon^2} \, d\mathbf{x}, \quad 0 < \varepsilon \ll 1. \end{aligned}$$

Für Ψ_ε können auch andere Regularisierungs- und Penalizerfunktionen (wie z.B. auf Seite 39 benutzt werden).

Definition 4.4.6 (Funktional HomReg für Serien)

Sei $\mathbf{R} := (R^1, \dots, R^M)$ eine Bildserie, $\mathbf{U} := (u^1, \dots, u^M)$ eine Serie von Deformationsfeldern, $\mathbf{c} := (c^1, \dots, c^M)$ eine Serie von Korrekturtermen und $\alpha, \beta \in \mathbb{R}^+$. Dann bezeichnet

$$\mathcal{J}[\mathbf{R}; \mathbf{U}, \mathbf{c}] := \mathcal{D}^{\text{cSSD}}[\mathbf{R}; \mathbf{U}, \mathbf{c}] + \alpha \mathcal{S}^{\text{elast}}[\mathbf{U}] + \beta \mathcal{S}^{\text{TV}}[\mathbf{c}], \quad (4.12)$$

ein *Funktional* zur simultanen Homogenisierung und Registrierung von Bildserien, bestehend aus Distanzmaß und Regularisierern.

Ein Minimum von \mathcal{J} erfüllt notwendigerweise die Euler-Lagrange-Gleichungen,

$$d\mathcal{J}[R; \mathbf{u}, \mathbf{c}] = \begin{pmatrix} \frac{d}{d\mathbf{u}} \mathcal{J}[\mathbf{R}; \mathbf{U}, \mathbf{c}] \\ \frac{d}{d\mathbf{c}} \mathcal{J}[\mathbf{R}; \mathbf{U}, \mathbf{c}] \end{pmatrix} = \mathbf{0}.$$

Die Herleitung erfolgt analog zu den Herleitungen in Abschnitt 4.2. Der Unterschied formaler Natur ist, dass bei Bildserien eine Summe vorangestellt ist. Aufgrund der Linearität können wir diese einfach mitziehen und wir erhalten unmittelbar die Euler-Lagrange-Gleichungen:

$$\begin{aligned}
 0 = & - \sum_{\nu=2}^{M-1} \left(c^{\nu+1} R^{\nu+1}(x - u^{\nu+1}) - 2c^{\nu} R^{\nu}(x - u^{\nu}) + c^{\nu-1} R^{\nu-1}(x - u^{\nu-1}) \right) R^{\nu}(x - u^{\nu}) \\
 & + \left(c^2 R^2(x - u^2) - c^1 R^1(x - u^1) \right) R^1(x - u^1) \\
 & + \left(c^{M-1} R^{M-1}(x - u^{M-1}) - c^M R^M(x - u^M) \right) R^M(x - u^M),
 \end{aligned}$$

für das Distanzmaß,

$$0 = \sum_{\nu=1}^M \mu \Delta \mathbf{u}^{\nu} + (\mu + \lambda) \nabla \operatorname{div} \mathbf{u}^{\nu},$$

für die elastische Regularisierung der Deformationsfelder bzw. zu

$$\sum_{\nu=1}^M \nabla \cdot \frac{\nabla c^{\nu}}{\sqrt{|\nabla c^{\nu}|_2^2 + \varepsilon^2}} = 0$$

für die TV-Regularisierung der Korrekturterme.

Betrachten wir die **Euler-Lagrange-Gleichung** für die Registrierung von Serien nun etwas genauer, so stellen wir fest, dass sich die Registrierung einer Bildserie auf die zweidimensionale Registrierung zurück führen lässt. Die Kopplung der einzelnen Schichten findet im wesentlichen über das Distanzmaß statt, bei dem drei benachbarte Schichten in die Berechnung mit einfließen. Darauf werden wir in Kapitel 6 noch etwas genauer eingehen.

5 Diskretisierung

Im vorigen Kapitel haben wir gesehen, dass wir durch Lösen eines Systems nichtlinearer partieller Differentialgleichungen das erweiterte Registrierungsproblem lösen können. In Abschnitt 4.3 haben wir eine Möglichkeit vorgestellt bekommen, das System zu linearisieren. In diesem Kapitel wollen wir nun das Problem diskretisieren und eine numerische Lösung erarbeiten. Dazu werden wir zunächst klären, was diskrete Bilder sind. Dadurch wird sehr schnell die Frage aufgeworfen, wie evtl. benötigte Bildinformationen aus Punkten, die nicht zum diskreten Bild gehören, gewonnen werden können. Damit muss die Frage der Interpolation geklärt werden. Schliesslich wollen wir eine Möglichkeit der diskreten Approximation der Differentialoperatoren durch Finite Differenzen vorstellen, um dann die diskrete linearisierte Euler-Lagrange-Gleichung des Registrierungsproblem aufzustellen. Um einen direkteren Bezug zum kontinuierlichen Teil und eine bessere Übersicht zu behalten, wollen wir die Bezeichner für diskrete und kontinuierliche Begriffe gleich halten.

5.1 Diskrete Bilder

In Abschnitt 4.1.2 wurde ein Bild als eine Abbildung definiert, die zweimal stetig differenzierbar und quadratisch integrierbar über einem Gebiet Ω ist. Dabei ist sowohl das Bild, als auch das Urbild der Abbildung kontinuierlich. Sprechen wir nun von diskreten Bildern, wie sie jeder z.B. aus Digitalkameras kennt, so sind diese Eigenschaften nicht mehr gegeben. Statt eines kontinuierlichen Gebiets Ω sprechen wir nunmehr von einem *diskretem Gebiet* bzw. einem *diskretem Gitter* $\hat{\Omega}$. Nachfolgende Definition soll die diskrete Fassung eindeutig festlegen.

Definition 5.1.1 (Diskretes Gebiet, Gitter, Gitterweite)

Seien $d \in \mathbb{N}$ die Dimension und $n_1, \dots, n_d \in \mathbb{N}$ die Anzahl der Punkte in Richtung x_1, \dots, x_d . Weiterhin wird mit $[0, 1]_n \subset \mathbb{Q}$, $n \in \mathbb{N}^{>1}$, eine diskrete Menge zwischen 0 und 1 mit n äquidistant verteilten Punkten bezeichnet. Daraus ergibt sich ein *Punktabstand* von $h := \frac{1}{n-1}$. Dann wird

$$\hat{\Omega} \subset [0, 1]_{n_1} \times \dots \times [0, 1]_{n_d}$$

diskretes Gebiet oder *Gitter* mit *Gitterweiten* $h_1 = \frac{1}{n_1-1}, \dots, h_d = \frac{1}{n_d-1}$ genannt.

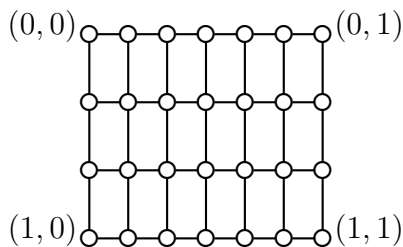


Abbildung 5.1: Diskretes Gitter $\hat{\Omega}$ mit $d = 2$, $n_1 = 4$, $n_2 = 7$ und $h_1 = \frac{1}{3}$ und $h_2 = \frac{1}{6}$.

Zur Veranschaulichung der Definition siehe Abb. 5.1. Dort ist ein Gitter $\hat{\Omega} = [0, 1]_4 \times [0, 1]_7$ mit 4×7 Gitterpunkten mit Abstand h_1, h_2 zu sehen. Ist der Abstand der Gitterpunkte in jeder Dimension gleich, so handelt es sich um ein *isotropes Gitter*.

Mit einem Gitter wie in Definition 5.1.1 sind die Achsen, bzw. das Koordinatensystem eindeutig festgelegt. Der Ursprung liegt damit oben links im Gitter und

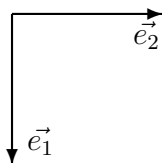


Abbildung 5.2: Koordinatensystem mit Matrixorientierung. e_1 bzw. e_2 sind die Einheitsvektoren in Richtung x_1 , bzw. x_2

die erste Dimension geht nach unten, die zweite Dimension nach rechts weg (vgl. Abb. 5.2). Diese Orientierung nennt sich *Matrixorientierung*, da es sich an die Schreibweise für Matrizen anlehnt.

Ein diskretes Gebiet bzw. Gitter kann als diskrete Punktmenge mit fester Ordnung angesehen werden. Damit lässt sich die Definition eines diskreten Bildes als Abbildung einer diskreten Punktmenge in einen Grauwertbereich auffassen, in dem jedem Gitterpunkt ein Grauwert zugeordnet wird.

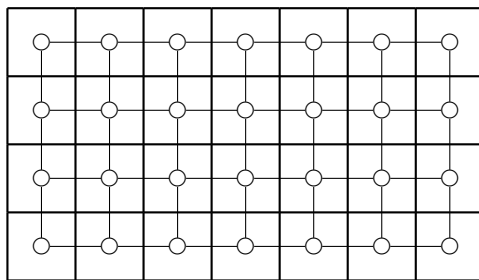


Abbildung 5.3: Diskretes Bild mit darunter liegendem Gitter mit $d = 2$, $n_1 = 4$, $n_2 = 7$ und $h_1 = h_2$. Jedem Gitterpunkt wird ein Grauwert zugeordnet, der bei der Darstellung als Bild die Fläche $h_1 \times h_2$ zugeordnet bekommt. Damit liegen die *Pixel* dicht nebeneinander, während die Gitterpunkte den Abstand h zueinander haben.

Definition 5.1.2 (Diskretes Bild, Pixel, Voxel)

Ein *diskretes Bild* B ist die Abbildung eines diskreten Gitters $\widehat{\Omega}$ in eine diskrete Menge von Grau- bzw. Intensitätswerten $\widehat{\mathbb{G}} \subset \mathbb{N}$,

$$B : \widehat{\Omega} \rightarrow \widehat{\mathbb{G}}$$

$$\mathbf{x} \mapsto B(\mathbf{x}).$$

Ein Bildpunkt eines zweidimensionalen Bildes heisst *Pixel* (*engl.*: picture element) und eines dreidimensionalen Bildes *Voxel* (*engl.*: volume element).

Abbildung 5.4 zeigt schematisch die Zuordnung von Gitterpunkt und Pixel. Während der Gitterpunkt als Punkt im Raum keine Fläche einnimmt, wird dem Pixel eine Fläche der Größe $h_1 \times h_2$ zugeordnet mit dem Gitterpunkt als Mittelpunkt. Damit liegen die Pixel dicht aneinander und lassen sich so als uns bekanntes

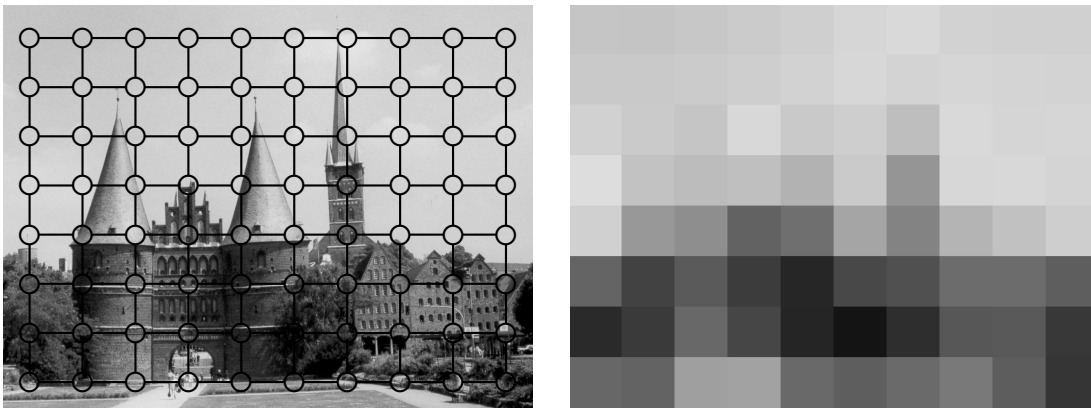


Abbildung 5.4: *links:* Diskretisierung eines zweidimensionalen Bildes durch auflegen eines Gitters mit $n_1 = 8$ und $n_2 = 10$ Gitterpunkten. *rechts:* Ergebnis

Bild (ohne Löcher) darstellen. In Abbildung 5.4 und 5.5 wird an einer Fotografie des Holstentors in Lübeck der Effekt verschieden grosser bzw. feiner Gitter auf die optische Qualität des diskreten Bildes dargestellt. Je feiner das Gitter gewählt wird, desto mehr Detailinformation steckt in den Bildern. Die Digitalisierung kann dabei auf zwei verschiedene Weisen vollzogen werden. Im ersten Fall wird dem Pixel der Wert zugeordnet, an dem der Gitterpunkt sitzt. In der Praxis erfolgt die Digitalisierung in der Form, dass ein Bild an den Gitterpunkten nicht den exakten Grauwert der Position zugeordnet bekommt, sondern den Mittelwert aus der Fläche, die ein Pixel am Gitterpunkt einnimmt. Dieser Pixeleffekt ist sehr gut durch die Blockbildung im diskreten Bild zu erkennen.

Neben der Wahl des Gitters wird bei diskreten Bildern die Anzahl der Grauwerte, die zur Darstellung verwendet werden, festgelegt. Die so genannte *Tiefe* des



Abbildung 5.5: Diskrete Bilder mit unterschiedlich feinen Gittern. Die Auflösungen betragen *v.l.n.r.*: (8×10) , (38×50) , (76×100) und (761×1000) . Je feiner das Gitter ist, desto mehr Detailinformation ist in den Bildern enthalten.

Bildes (Anzahl der Bits, die zur Darstellung der Zahl verwendet werden) hängt in den meisten Fällen vom Sensor ab, der das Signal misst. So hat eine CT-Aufnahme meist die Tiefe 10 und damit 1024 Graustufen, während bei MR-Aufnahmen die Tiefe 12 verwendet wird und damit 4096 Graustufen vorhanden sind. Die An-



Abbildung 5.6: Diskrete Bilder mit unterschiedlich grossem Grauwertbereich (*v.l.n.r.*: 256, 4, 2 Grauwerte).

zahl der Graustufen hat Auswirkungen auf den Kontrast und die Details im Bild. Während beispielsweise wie in Abb. 5.6, im linken Bild mit 256 Graustufen noch Mauersteine zu erkennen sind, ist im mittleren Bild nur noch zwischen Mauer im Schatten und Mauer in der Sonne unterscheidbar.

Zusammenfassend lässt sich sagen, dass bei einem diskreten Bild die Auswahl von Pixelzahl und Bildtiefe darüber entscheidet, ob die Details, die man erkennen möchte, darstellbar sind oder nicht.

5.2 Lineare Interpolation

Wie wir in Definition 5.1.2 gesehen haben, ist das Bild nur an den diskreten Gitterpunkten definiert. Wird durch die Anwendung einer Deformation oder Abbildung auf das Bild ein Punkt erreicht, der nicht auf dem Gitter liegt, so muss der Wert aus den vorhandenen Punkten in der Nachbarschaft interpoliert werden. In unserem Fall verwenden wir hierfür die lineare Interpolation, bei der der fehlende Wert durch die Verbindung der benachbarten Punkte mit einer Geraden approximiert wird. Am einfachsten sieht man das Verhalten der Interpolation im eindimensionalen Fall. In Abb. 5.7 ist eine Sigmoidfunktion in rot dargestellt. Durch die

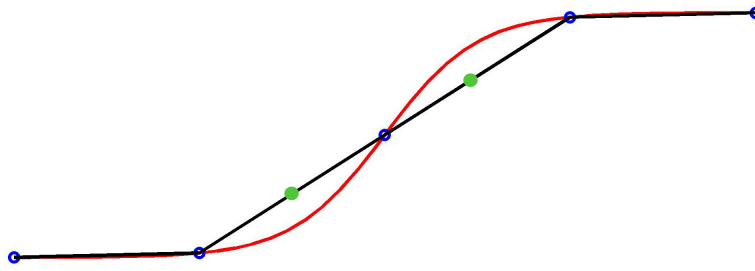


Abbildung 5.7: Veranschaulichung der linearen Interpolation im eindimensionalen Fall. Die kontinuierliche Abbildung (rot) wird nur durch vier diskrete Punkte (blau) dargestellt. Durch lineare Interpolation (schwarze Linie) wird die ursprüngliche Abbildung nur sehr schlecht wiedergegeben, indem an den neuen Positionen (grüne Punkte) Werte zugeordnet werden, die von der ursprünglichen Abbildung stark abweichen.

Diskretisierung ohne Mittelwertbildung reduziert sich die Funktion auf die blauen Punkte. Wird nun ein Wert angefordert, der nicht mit einem der blauen Punkte übereinstimmt, so muss interpoliert werden. Dabei wird zwischen den benachbarten Punkten eine Gerade gezogen und an der entsprechenden Stelle des neuen Punktes ausgewertet. Wie im Falle der Sigmoidfunktion kann der *interpolierte Wert* stark vom tatsächlichen Wert abweichen.

Im zweidimensionalen Fall kann ebenfalls linear interpoliert werden. Der Wert eines Bildpunktes, der nicht auf dem Gitter liegt, wird durch lineare Interpolation der Werte der vier Nachbarpunkte berechnet. Wird nunmehr der Wert am Punkt p gesucht, so müssen im ersten Schritt die vier benachbarten Punkte von p ermittelt werden (vgl. Abb. 5.8).

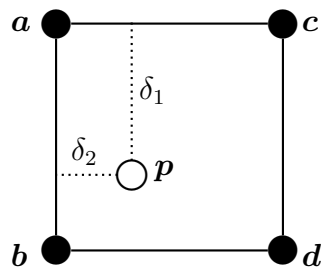


Abbildung 5.8: Lineare Interpolation im zweidimensionalen Fall. Der interpolierte Wert für p berechnet sich aus den Werten der umgebenden Punkte a , b , c und d . Die Abstände δ_1 und δ_2 zum Referenzpunkt a dienen als Gewichtungsfaktoren.

Definition 5.2.1 ((Bi-)Lineare Interpolation, interpolierter Wert)

Seien n_1, n_2 und h_1, h_2 wie in Definition 5.1.1, \vec{e}_1, \vec{e}_2 die Einheitsvektoren in Richtung x_1, x_2 und \mathbf{p} ein Punkt im inneren des Gitters $\widehat{\Omega}$. Dann bestimmen sich $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \widehat{\Omega}$ und $\delta_1 \in [0, h_1), \delta_2 \in [0, h_2)$ aus Abb. 5.8 nach folgendem Schema:

$$\begin{aligned} \mathbf{a} &= (a_1, a_2)^\top := \frac{\lfloor m\mathbf{p} \rfloor}{m} = \left(\frac{\lfloor m_1 p_1 \rfloor}{m_1}, \frac{\lfloor m_2 p_2 \rfloor}{m_2} \right)^\top & \mathbf{b} &:= \mathbf{a} + h_1 \vec{e}_1 \\ \mathbf{c} &:= \mathbf{a} + h_2 \vec{e}_2 & \mathbf{d} &:= \mathbf{a} + h_1 \vec{e}_1 + h_2 \vec{e}_2 \\ \delta_1 &:= p_1 - a_1 & \delta_2 &:= p_2 - a_2 \end{aligned}$$

Dann berechnet sich der *interpolierte Wert* im Punkt \mathbf{p} durch *bilineare Interpolation* mit

$$B(\mathbf{p}) := \frac{1}{h_1 h_2} \left[\begin{aligned} & B(\mathbf{a})(h_1 - \delta_1)(h_2 - \delta_2) + B(\mathbf{b})\delta_1(h_2 - \delta_2) \\ & + B(\mathbf{c})(h_1 - \delta_1)\delta_2 + B(\mathbf{d})\delta_1\delta_2 \end{aligned} \right].$$

Anmerkung: Stimmt der Punkt \mathbf{p} mit einem der benachbarten Gitterpunkte $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ überein, liegt somit selbst auf dem Gitter, so verändert sich der Wert durch die Interpolation nicht.

5.3 Differenzenquotienten

Wir möchten die in Kapitel 4 hergeleitete partielle Differentialgleichung diskretisieren und überführen diese dazu in eine Differenzengleichung. Die kontinuierlichen Operatoren werden hierbei mittels Differenzensternen approximiert. Wir leiten nun Approximationen für die Operatoren ∇ , Δ und ∇div her.

Beginnen wollen wir mit der Approximation des ∇ -Operators. Sei dazu für beliebig, aber fest gewähltes $x_2 \in [0, 1]$, $v(x_1) = \mathbf{u}(x_1, x_2)$. Des weiteren sei h_1 der Abstand zweier Punkte in x_1 -Richtung. Dann liefert die Taylorentwicklung,

$$v(x_1 + h_1) = v(x_1) + h_1 \frac{\partial v}{\partial x_1}(x_1) + \mathcal{O}(h_1^2), \quad (5.1)$$

bzw.

$$v(x_1 - h_1) = v(x_1) - h_1 \frac{\partial v}{\partial x_1}(x_1) + \mathcal{O}(h_1^2). \quad (5.2)$$

Subtrahieren wir nun die zweite von der ersten Gleichung und stellen nach der ersten Ableitung um, so erhalten wir,

$$\frac{\partial v}{\partial x_1}(x_1) = \frac{v(x_1 + h_1) - v(x_1 - h_1)}{2h_1} + \mathcal{O}(h_1^2). \quad (5.3)$$

Hier fällt auf, dass der Punkt, in dem die Ableitung bestimmt wird, nicht mehr enthalten ist. Auf analoge Weise gilt dies für die zweite Komponente. Sei nunmehr für beliebig, aber fest gewähltes $x_1 \in [0, 1]$, $w(x_2) = \mathbf{u}(x_1, x_2)$. Dann folgt aus der Taylorentwicklung,

$$\frac{\partial w}{\partial x_2}(x_2) = \frac{w(x_2 + h_2) - w(x_2 - h_2)}{2h_2} + \mathcal{O}(h_2^2). \quad (5.4)$$

Die Zusammenfassung von Gleichung (5.3) und (5.4) wird unter Vernachlässigung des Restglieds als *zentraler Differenzenquotient erster Ordnung* bezeichnet,

$$\nabla \mathbf{u}(x_1, x_2) \approx \begin{pmatrix} \frac{\mathbf{u}(x_1+h_1, x_2) - \mathbf{u}(x_1-h_1, x_2)}{2h_1} \\ \frac{\mathbf{u}(x_1, x_2+h_2) - \mathbf{u}(x_1, x_2-h_2)}{2h_2} \end{pmatrix}. \quad (5.5)$$

Der zentrale Differenzenquotient kann nur an den Punkten des Gitters ausgerechnet werden, an denen die Nachbarpunkte $\mathbf{x} \pm \mathbf{h}$ Teil des Gitters sind. Damit fallen die Randpunkte des Gitters weg. Um den ∇ -Operator aber auch dort auswerten zu können, definieren wir noch den *vorwärts bzw. rückwärts gerichteten Differenzenquotienten*. Diese folgen direkt aus der Taylorentwicklung, so liefert Gleichung (5.1) den vorwärts gerichteten Operator und Gleichung (5.2) den rückwärts gerichteten Operator,

$$\nabla \mathbf{u}(x_1, x_2) \approx \begin{pmatrix} \frac{\mathbf{u}(x_1+h_1, x_2) - \mathbf{u}(x_1, x_2)}{h_1} \\ \frac{\mathbf{u}(x_1, x_2+h_2) - \mathbf{u}(x_1, x_2)}{h_2} \end{pmatrix}, \text{ bzw. } \nabla \mathbf{u}(x_1, x_2) \approx \begin{pmatrix} \frac{\mathbf{u}(x_1, x_2) - \mathbf{u}(x_1-h_1, x_2)}{h_1} \\ \frac{\mathbf{u}(x_1, x_2) - \mathbf{u}(x_1, x_2-h_2)}{h_2} \end{pmatrix}.$$

Für die Approximation der ersten Ableitung hat die Taylorentwicklung erster Ordnung ausgereicht. Um nun für die zweite Ableitung eine Approximation mit annehmbarer Güte zu erhalten, müssen wir die Taylorentwicklung bis zur Ordnung drei weiter führen, damit der diskrete Operator die Fehlerordnung $\mathcal{O}(h^2)$ erreichen kann.

Sei nun für ein beliebig, aber fest gewähltes $x_2 \in [0, 1]$, $v(x_1) = \mathbf{u}(x_1, x_2)$. Die Taylorentwicklung liefert dann,

$$v(x_1 + h_1) = v(x_1) + h_1 \frac{\partial v}{\partial x_1}(x_1) + \frac{h_1^2}{2} \frac{\partial^2 v}{\partial x_1^2}(x_1) + \frac{h_1^3}{6} \frac{\partial^3 v}{\partial x_1^3}(x_1) + \mathcal{O}(h_1^4),$$

bzw.

$$v(x_1 - h_1) = v(x_1) - h_1 \frac{\partial v}{\partial x_1}(x_1) + \frac{h_1^2}{2} \frac{\partial^2 v}{\partial x_1^2}(x_1) - \frac{h_1^3}{6} \frac{\partial^3 v}{\partial x_1^3}(x_1) + \mathcal{O}(h_1^4).$$

Addieren wir beide Formeln und stellen nach der zweiten Ableitung um, so ergibt sich

$$\frac{\partial^2 v}{\partial x_1^2}(x_1) = \frac{v(x_1 + h_1) - 2v(x_1) + v(x_1 - h_1))}{h_1^2} + \mathcal{O}(h_1^2). \quad (5.6)$$

Verfährt man analog mit beliebig, aber fest gewähltem $x_1 \in [0, 1]$ und $w(x_2) = u(x_1, x_2)$, so ergibt

$$\frac{\partial^2 w}{\partial x_2^2}(x_2) = \frac{w(x_2 + h_2) - 2w(x_2) + w(x_2 - h_2))}{h_2^2} + \mathcal{O}(h_2^2). \quad (5.7)$$

Fassen wir nun Gleichung (5.6) und (5.7) zusammen und resubstituieren, so erhalten wir als Approximation an den Δ -Operator

$$\begin{aligned} \Delta \mathbf{u}(x_1, x_2) \approx & \frac{\mathbf{u}(x_1 + h_1, x_2) - 2\mathbf{u}(x_1, x_2) + \mathbf{u}(x_1 - h_1, x_2))}{h_1^2} \\ & + \frac{\mathbf{u}(x_1, x_2 + h_2) - 2\mathbf{u}(x_1, x_2) + \mathbf{u}(x_1, x_2 - h_2))}{h_2^2} \end{aligned} \quad (5.8)$$

Dieser Operator wird als *zentraler Differenzenquotient zweiter Ordnung* bezeichnet.

Nun wollen wir den div -Operator durch einen Differenzenquotienten approximieren, um dann $\nabla \text{div } \mathbf{u}$ diskretisieren zu können. Die Divergenz ist definiert als die Summe der ersten partiellen Ableitungen,

$$\text{div} = \frac{\partial}{\partial x_1} + \frac{\partial}{\partial x_2}.$$

Nach den Gleichung (5.3) und (5.4) ergibt sich sofort

$$\text{div } \mathbf{u}(x_1, x_2) \approx \frac{\mathbf{u}(x_1 + h_1, x_2) - \mathbf{u}(x_1 - h_1, x_2))}{2h_1} + \frac{\mathbf{u}(x_1, x_2 + h_2) - \mathbf{u}(x_1, x_2 - h_2))}{2h_2} \quad (5.9)$$

die Diskretisierung von $\text{div } \mathbf{u}$.

Die Diskretisierung von $\nabla \text{div } \mathbf{u}$ erfolgt nunmehr durch Anwendung der obigen Techniken. Die Kombination der Operatoren liefert

$$\nabla \text{div } \mathbf{u} = \begin{pmatrix} \frac{\partial^2}{\partial x_1^2} u_1 + \frac{\partial^2}{\partial x_1 \partial x_2} u_2 \\ \frac{\partial^2}{\partial x_1 \partial x_2} u_1 + \frac{\partial^2}{\partial x_2^2} u_2 \end{pmatrix}.$$

Die Ableitungen zweiter Ordnung sind bereits in den Gleichungen (5.6) und (5.7) angegeben, somit fehlen nur noch die gemischten Ableitungen. Da aus Symmetriegründen die Rechnungen identisch sind, wollen wir hier die Anwendung der Ableitung in x_2 auf $\frac{\partial}{\partial x_1} u_1$ vorführen. Gegeben ist

$$\frac{\partial}{\partial x_1} u_1 \approx \frac{u_1(x_1 + h_1, x_2) - u_1(x_1 - h_1, x_2))}{2h_1}.$$

Bildet man erneut einen Differenzenquotienten für die erste Ableitung in x_2 , so ergibt sich

$$\frac{\partial^2}{\partial x_1 \partial x_2} u_1 \approx \frac{u_1(x_1 + h_1, x_2 + h_2) - u_1(x_1 - h_1, x_2 + h_2)}{2h_1 2h_2} - \frac{u_1(x_1 + h_1, x_2 - h_2) - u_1(x_1 - h_1, x_2 - h_2)}{2h_1 2h_2}. \quad (5.10)$$

Analoges gilt für $\frac{\partial^2}{\partial x_1 \partial x_2} u_2$.

Zum Schluss wollen wir die Approximation des TV-Operators $\nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}}$ durch Differenzenquotienten zeigen. Die Approximation des ∇ -Operators findet sich bereits in (5.5). Fehlt nunmehr nur noch die Approximation von $|\nabla c|_2^2$. Die Anwendung der Normdefinition und des ∇ -Operators liefert

$$|\nabla c|_2^2 = (\nabla c)^\top (\nabla c) = \left(\frac{\partial}{\partial x_1} c \right)^2 + \left(\frac{\partial}{\partial x_2} c \right)^2.$$

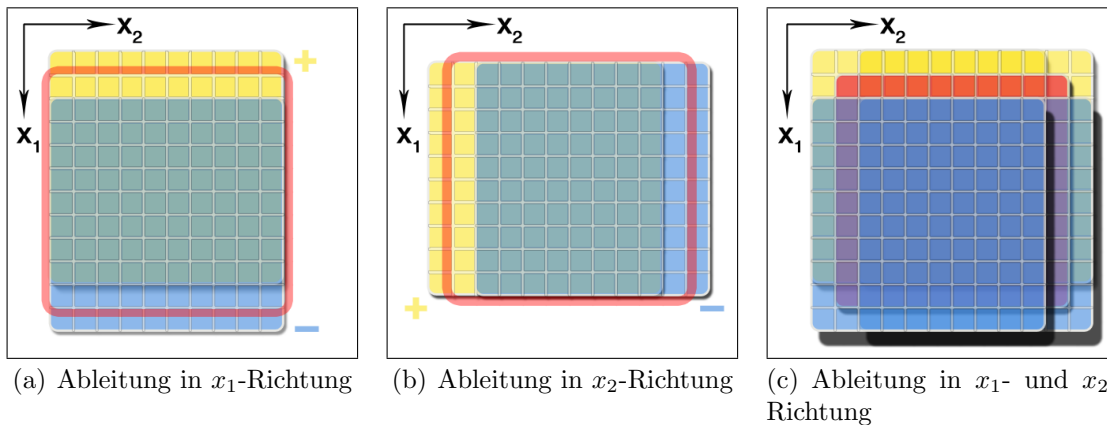


Abbildung 5.9: Veranschaulichung der Überlagerung der Matrixeinträge, die zur Berechnung der Ableitungen heran gezogen werden.

Zur Veranschaulichung zeigen wir nun noch in Abbildung 5.9 die Anordnung der Matrizen für die Ableitungen in x_1 - und x_2 -Richtung und der gemischten Ableitung $x_1 x_2$ bildlich dargestellt. Der rote Rahmen zeigt den tatsächlichen Bildbereich. In Blau und Gelb sind die Verschiebungen der Bildmatrix angegeben. Fehlende Bildreihen oder Spalten werden durch kopieren der benachbarten Reihe bzw. Spalte ersetzt. Wie bereits in den Bildern angedeutet, werden die blauen Teile mit negativen, die gelben mit positiven Vorzeichen gewichtet und gehen so in die Berechnung ein.

5.4 Diskrete Euler-Lagrange-Gleichungen

Nachdem wir im vorigen Kapitel die Operatoren aus der Euler-Lagrange-Gleichung diskretisiert haben, können wir aus den Differentialgleichung Differenzgleichungen machen. Zur Wiederholung sind hier nochmals die kontinuierlichen Euler-Lagrange-Gleichungen aus (4.5) dargestellt,

$$\alpha(\mu\Delta\mathbf{u} + (\lambda + \mu)\nabla\operatorname{div}\mathbf{u}) = -(cT_{\mathbf{u}} - R)c\nabla T_{\mathbf{u}}$$

$$\text{und} \quad cT_{\mathbf{u}}^2 + \beta \nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}} = RT_{\mathbf{u}}.$$

Diese Differentialgleichungen werden wir in den nächsten Schritten derart umformen, dass die diskreten Operatoren aus dem vorigen Abschnitt direkt einzusetzen sind. Beginnen wollen wir mit der ersten Gleichung.

$$\alpha(\mu\Delta\mathbf{u} + (\lambda + \mu)\nabla\operatorname{div}\mathbf{u}) = -(cT_{\mathbf{u}} - R)c\nabla T_{\mathbf{u}}$$

$$\implies \alpha \left(\mu \left(\frac{\partial^2 \mathbf{u}}{\partial x_1^2} + \frac{\partial^2 \mathbf{u}}{\partial x_2^2} \right) + (\lambda + \mu) \nabla \left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right) \right) = -(cT_{\mathbf{u}} - R)c \begin{pmatrix} \frac{\partial}{\partial x_1} T_{\mathbf{u}} \\ \frac{\partial}{\partial x_2} T_{\mathbf{u}} \end{pmatrix}$$

$$\implies \alpha \left(\mu \begin{pmatrix} \frac{\partial^2 u_1}{\partial x_1^2} + \frac{\partial^2 u_1}{\partial x_2^2} \\ \frac{\partial^2 u_2}{\partial x_1^2} + \frac{\partial^2 u_2}{\partial x_2^2} \end{pmatrix} + (\lambda + \mu) \begin{pmatrix} \frac{\partial^2 u_1}{\partial x_1^2} + \frac{\partial^2 u_2}{\partial x_1 \partial x_2} \\ \frac{\partial^2 u_1}{\partial x_1 \partial x_2} + \frac{\partial^2 u_2}{\partial x_2^2} \end{pmatrix} \right) = -(cT_{\mathbf{u}} - R)c \begin{pmatrix} \frac{\partial}{\partial x_1} T_{\mathbf{u}} \\ \frac{\partial}{\partial x_2} T_{\mathbf{u}} \end{pmatrix}$$

$$\implies \alpha \begin{pmatrix} \left((\lambda + 2\mu) \frac{\partial^2}{\partial x_1^2} + \mu \frac{\partial^2}{\partial x_2^2} \right) u_1 + (\lambda + \mu) \frac{\partial^2}{\partial x_1 \partial x_2} u_2 \\ (\lambda + \mu) \frac{\partial^2}{\partial x_1 \partial x_2} u_1 + \left(\mu \frac{\partial^2}{\partial x_1^2} + (\lambda + 2\mu) \frac{\partial^2}{\partial x_2^2} \right) u_2 \end{pmatrix} = - \begin{pmatrix} (cT_{\mathbf{u}} - R)c \frac{\partial}{\partial x_1} T_{\mathbf{u}} \\ (cT_{\mathbf{u}} - R)c \frac{\partial}{\partial x_2} T_{\mathbf{u}} \end{pmatrix} \quad (5.11)$$

Um eine bessere Übersicht zu erhalten, verwenden wir für den rechten Teil der Gleichung nach Definition 4.2.3 den Begriff Kraft und die Abkürzung

$$\mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} := - \begin{pmatrix} (cT_{\mathbf{u}} - R)c \frac{\partial}{\partial x_1} T_{\mathbf{u}} \\ (cT_{\mathbf{u}} - R)c \frac{\partial}{\partial x_2} T_{\mathbf{u}} \end{pmatrix}.$$

Verwenden wir auch noch die abkürzende Schreibweise mit dem in Definition 4.2.1 eingeführten Elastizitätsoperator \mathcal{A}_α , so können wir Gleichung (5.11) in eine Matrix-Vektor-orientierte Schreibweise überführen, indem wir die Operatoren zusammenfassen, die auf jeweils eine Komponente von \mathbf{u} angewendet werden,

$$\mathcal{A}_\alpha \mathbf{u} = \mathbf{f} \quad \implies \quad \alpha \begin{pmatrix} \mathcal{A}^{11} & \mathcal{A}^{12} \\ \mathcal{A}^{21} & \mathcal{A}^{22} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}.$$

Die neuen Operatoren

$$\mathcal{A}^{11} = (\lambda + 2\mu) \frac{\partial^2}{\partial x_1^2} + \mu \frac{\partial^2}{\partial x_2^2}$$

$$\mathcal{A}^{12} = \mathcal{A}^{21} = (\lambda + \mu) \frac{\partial^2}{\partial x_1 \partial x_2}$$

$$\mathcal{A}^{22} = \mu \frac{\partial^2}{\partial x_1^2} + (\lambda + 2\mu) \frac{\partial^2}{\partial x_2^2}$$

können nach der Anwendung auf \mathbf{u} mit Hilfe der Gleichungen (5.5), (5.8) und (5.10) diskretisiert werden. Betrachten wir nunmehr die Teilausdrücke $\mathcal{A}^{11}u_1$, $\mathcal{A}^{12}u_2$, $\mathcal{A}^{21}u_1$ und $\mathcal{A}^{22}u_2$ mit ihren zugehörigen Diskretisierungen,

$$\begin{aligned} \mathcal{A}^{11}u_1 &\approx (\lambda + 2\mu) \frac{u_1(x_1 + h_1, x_2) - 2u_1(x_1, x_2) + u_1(x_1 - h_1, x_2)}{h_1^2} \\ &\quad + \mu \frac{u_1(x_1, x_2 + h_2) - 2u_1(x_1, x_2) + u_1(x_1, x_2 - h_2)}{h_2^2}, \end{aligned}$$

$$\begin{aligned} \mathcal{A}^{12}u_2 &\approx (\lambda + \mu) \left(\frac{u_2(x_1 + h_1, x_2 + h_2) - u_2(x_1 - h_1, x_2 + h_2)}{2h_1 2h_2} \right. \\ &\quad \left. - \frac{u_2(x_1 + h_1, x_2 - h_2) - u_2(x_1 - h_1, x_2 - h_2)}{2h_1 2h_2} \right), \end{aligned}$$

$$\begin{aligned} \mathcal{A}^{21}u_1 &\approx (\lambda + \mu) \left(\frac{u_1(x_1 + h_1, x_2 + h_2) - u_1(x_1 - h_1, x_2 + h_2)}{2h_1 2h_2} \right. \\ &\quad \left. - \frac{u_1(x_1 + h_1, x_2 - h_2) - u_1(x_1 - h_1, x_2 - h_2)}{2h_1 2h_2} \right), \end{aligned}$$

$$\begin{aligned} \mathcal{A}^{22}u_2 &\approx \mu \frac{u_2(x_1 + h_1, x_2) - 2u_2(x_1, x_2) + u_2(x_1 - h_1, x_2)}{h_1^2} \\ &\quad + (\lambda + 2\mu) \frac{u_2(x_1, x_2 + h_2) - 2u_2(x_1, x_2) + u_2(x_1, x_2 - h_2)}{h_2^2}, \end{aligned}$$

so handelt es sich bei

$$\alpha \begin{pmatrix} \mathcal{A}^{11}u_1 + \mathcal{A}^{12}u_2 \\ \mathcal{A}^{21}u_1 + \mathcal{A}^{22}u_2 \end{pmatrix} = \begin{pmatrix} (cT_{\mathbf{u}}(x_1, x_2) - R(x_1, x_2)) c \frac{T_{\mathbf{u}}(x_1+h_1, x_2) - T_{\mathbf{u}}(x_1-h_1, x_2)}{2h_1} \\ (cT_{\mathbf{u}}(x_1, x_2) - R(x_1, x_2)) c \frac{T_{\mathbf{u}}(x_1, x_2+h_2) - T_{\mathbf{u}}(x_1, x_2-h_2)}{2h_2} \end{pmatrix}.$$

um Differenzgleichungen.

Anmerkung: Die Operatoren \mathcal{A}^{12} und \mathcal{A}^{21} sind identisch, da die Reihenfolge der Differentiation egal ist. Die Operatoren \mathcal{A}^{11} und \mathcal{A}^{22} sind bis auf den Austausch von x_1 und x_2 ebenfalls identisch, wodurch sich diese als symmetrisch ansehen lassen.

Betrachten wir nun den zweiten Teil der Euler-Lagrange-Gleichungen aus (4.5),

$$cT_{\mathbf{u}}^2 + \beta \nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}} = RT_{\mathbf{u}}.$$

Die Teile $cT_{\mathbf{u}}^2$ und $RT_{\mathbf{u}}$ sind diskret sofort klar. Bleibt der mittlere Teil,

$$c_t := \nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}}.$$

Für den zweidimensionalen Fall folgt dann:

$$\begin{aligned} c_t &= \frac{\partial}{\partial x_1} \left(\frac{c_{x_1}}{\sqrt{c_{x_1}^2 + c_{x_2}^2}} \right) + \frac{\partial}{\partial x_2} \left(\frac{c_{x_2}}{\sqrt{c_{x_1}^2 + c_{x_2}^2}} \right) \\ &= \frac{\partial}{\partial x_1} \left(c_{x_1} (c_{x_1}^2 + c_{x_2}^2)^{-\frac{1}{2}} \right) + \frac{\partial}{\partial x_2} \left(c_{x_2} (c_{x_1}^2 + c_{x_2}^2)^{-\frac{1}{2}} \right) \\ &= \frac{c_{x_1 x_1} (c_{x_1}^2 + c_{x_2}^2) - (c_{x_1}^2 c_{x_1 x_1} + c_{x_1} c_{x_2} c_{x_1 x_2})}{(c_{x_1}^2 + c_{x_2}^2)^{\frac{3}{2}}} \\ &\quad + \frac{c_{x_2 x_2} (c_{x_1}^2 + c_{x_2}^2) - (c_{x_2}^2 c_{x_2 x_2} + c_{x_1} c_{x_2} c_{x_1 x_2})}{(c_{x_1}^2 + c_{x_2}^2)^{\frac{3}{2}}} \\ &= \frac{c_{x_1 x_1} c_{x_2}^2 - c_{x_1} c_{x_2} c_{x_1 x_2}}{(c_{x_1}^2 + c_{x_2}^2)^{\frac{3}{2}}} + \frac{c_{x_2 x_2} c_{x_1}^2 - c_{x_1} c_{x_2} c_{x_1 x_2}}{(c_{x_1}^2 + c_{x_2}^2)^{\frac{3}{2}}} \\ &= \frac{c_{x_1}^2 c_{x_2 x_2} - 2c_{x_1} c_{x_2} c_{x_1 x_2} + c_{x_2}^2 c_{x_1 x_1}}{(c_{x_1}^2 + c_{x_2}^2)^{\frac{3}{2}}}. \end{aligned}$$

Verwenden wir nun wiederum die in Abschnitt 5.3 eingeführten Approximationen an die Differentialoperatoren, so ergibt sich unmittelbar

$$\begin{aligned} c_{x_1} &\approx \frac{c(x_1 + h_1, x_2) - c(x_1 - h_1, x_2)}{2h_1}, \\ c_{x_2} &\approx \frac{c(x_1, x_2 + h_2) - c(x_1, x_2 - h_2)}{2h_2}, \\ c_{x_1 x_2} &\approx \frac{c(x_1 + h_1, x_2 + h_2) - c(x_1 - h_1, x_2 + h_2)}{2h_1 2h_2} \\ &\quad - \frac{c(x_1 + h_1, x_2 - h_2) - c(x_1 - h_1, x_2 - h_2)}{2h_1 2h_2}, \\ c_{x_1 x_1} &\approx \frac{c(x_1 + h_1, x_2) - 2c(x_1, x_2) + c(x_1 - h_1, x_2)}{h_1^2}, \\ c_{x_2 x_2} &\approx \frac{c(x_1, x_2 + h_2) - 2c(x_1, x_2) + c(x_1, x_2 - h_2)}{h_2^2}. \end{aligned}$$

Daraus ergibt sich dann die Differenzengleichung

$$cT_{\mathbf{u}}^2 + \beta c_t = RT_{\mathbf{u}}.$$

6 Implementierung

In diesem Kapitel werden wir Lösungsverfahren für die elastische Registrierung und die TV-Regularisierung vorstellen. Beginnen wollen wir mit einem besonders effizienten direkten Lösungsverfahren für die elastische Registrierung. Als Alternative dazu stellen wir anschließend eine approximative Lösungsstrategie vor, die besonders leicht zu implementieren ist. Im Anschluss daran stellen wir eine einfache Lösungsstrategie für die TV-Regularisierung vor. Zur Verdeutlichung des gesamten Ablaufs der Registrierung geben wir den Gesamtalgorithmus in Pseudo-Matlab-Code an. Zum Abschluss dieses Kapitels stellen wir Strategien vor, die den Algorithmus insgesamt effizienter und robuster gestaltet haben. Dazu gehören intelligente Abbruchkriterien, die Einführung eines Multiresolutionsansatzes und ein einfaches Verfahren zur automatischen Bestimmung der Lamé-Konstanten für die elastische Registrierung.

6.1 Lösungsverfahren

In Abschnitt 4.4 haben wir die Erweiterung der Methode auf Bildserien vorgestellt bekommen. Dort wurde bereits zu Beginn eine intelligente Kopplung der einzelnen Bilder der Serie über das Distanzmaß diskutiert. Am Ende des Abschnitts wurden die Euler-Lagrange-Gleichungen für Bildserien vorgestellt. Dort konnte man die anfangs beschriebene Kopplung sofort entdecken. Der entscheidende Schluss dieses Ansatzes ist, dass für die Registrierung von Bildserien kein eigenständiges Lösungsverfahren implementiert werden muss, sondern dass sich die Lösung aus einer Reihe von Lösungen zweidimensionaler Registrierungen zusammensetzt. Diese Eigenschaft machen wir uns im weiteren zu nutze und stellen die Lösungsverfahren für die zweidimensionale Registrierung vor. Der übergeordnete Zusammenbau der erhaltenen Teillösungen wird dann in Abschnitt C.3 durch den Pseudo-Matlab-Code dargestellt.

6.1.1 Direktes Lösungsverfahren zur elastischen Registrierung

In Abschnitt 5.4 haben wir die diskreten Euler-Lagrange-Gleichungen als Differenzgleichung vorgestellt bekommen. Wir haben dort

$$\mathcal{A}_\alpha \mathbf{u} = \mathbf{f} \quad \Longrightarrow \quad \alpha \begin{pmatrix} \mathcal{A}^{11} & \mathcal{A}^{12} \\ \mathcal{A}^{21} & \mathcal{A}^{22} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

gesehen, mit

$$\begin{aligned}\mathcal{A}^{11} &= (\lambda + 2\mu) \frac{\partial^2}{\partial x_1^2} + \mu \frac{\partial^2}{\partial x_2^2}, \\ \mathcal{A}^{12} &= \mathcal{A}^{21} = (\lambda + \mu) \frac{\partial^2}{\partial x_1 \partial x_2}, \\ \text{und } \mathcal{A}^{22} &= \mu \frac{\partial^2}{\partial x_1^2} + (\lambda + 2\mu) \frac{\partial^2}{\partial x_2^2}.\end{aligned}$$

Betrachten wir die dort vorgestellte Approximation der Differentialoperatoren durch Finite Differenzen genauer,

$$\begin{aligned}\mathcal{A}^{11}u_1 &\approx (\lambda + 2\mu) \frac{u_1(x_1 + h_1, x_2) - 2u_1(x_1, x_2) + u_1(x_1 - h_1, x_2)}{h_1^2} \\ &\quad + \mu \frac{u_1(x_1, x_2 + h_2) - 2u_1(x_1, x_2) + u_1(x_1, x_2 - h_2)}{h_2^2}, \\ \mathcal{A}^{12}u_2 &\approx (\lambda + \mu) \left(\frac{u_2(x_1 + h_1, x_2 + h_2) - u_2(x_1 - h_1, x_2 + h_2)}{2h_1 2h_2} \right. \\ &\quad \left. - \frac{u_2(x_1 + h_1, x_2 - h_2) - u_2(x_1 - h_1, x_2 - h_2)}{2h_1 2h_2} \right), \\ \mathcal{A}^{21}u_1 &\approx (\lambda + \mu) \left(\frac{u_1(x_1 + h_1, x_2 + h_2) - u_1(x_1 - h_1, x_2 + h_2)}{2h_1 2h_2} \right. \\ &\quad \left. - \frac{u_1(x_1 + h_1, x_2 - h_2) - u_1(x_1 - h_1, x_2 - h_2)}{2h_1 2h_2} \right), \\ \mathcal{A}^{22}u_2 &\approx \mu \frac{u_2(x_1 + h_1, x_2) - 2u_2(x_1, x_2) + u_2(x_1 - h_1, x_2)}{h_1^2} \\ &\quad + (\lambda + 2\mu) \frac{u_2(x_1, x_2 + h_2) - 2u_2(x_1, x_2) + u_2(x_1, x_2 - h_2)}{h_2^2},\end{aligned}$$

so erkennt man, dass die Finite-Differenzen-Approximation über eine Filterung (diskrete Faltung) mit Filterkernen der Größe (3×3) durchgeführt werden kann. Die notwendigen Filterkerne lauten S^{ij} , $i, j = 1, 2$, mit

$$\begin{aligned}S^{11} &:= \begin{pmatrix} 0 & (\lambda + 2\mu) & 0 \\ \mu & -2(\lambda + 3\mu) & \mu \\ 0 & (\lambda + 2\mu) & 0 \end{pmatrix} =: (S^{22})^\top, \\ S^{12} &:= \frac{\lambda + \mu}{4} \begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix} =: S^{21}.\end{aligned}\tag{6.1}$$

Daraus ergibt sich die wiederum kompakte Schreibweise

$$\mathcal{A}_\alpha \mathbf{u} \approx \alpha \begin{pmatrix} S^{11} * u_1 + S^{12} * u_2 \\ S^{21} * u_1 + S^{22} * u_2 \end{pmatrix}. \quad (6.2)$$

Prinzipiell hätten wir an dieser Stelle schon ein Verfahren, mit dem wir die Lösung ausrechnen könnten. Die Schwierigkeit hierbei ist allerdings, dass die notwendigen Faltungen so zeitintensiv bei größeren Bildern wären, dass dies nicht mehr praktikabel ist. Daher bedienen wir uns eines weiteren Kniffs aus der Numerik und machen aus der Darstellung über die Faltung eine Matrix-Vektor-Operation.

Definition 6.1.1 (vec-Operation)

Sei X eine zweidimensionale Matrix der Größe $(n_1 \times n_2)$. Dann bezeichne $\vec{X} := \text{vec}(X)$ die Umordnung der Matrix X zu einem Vektor \vec{X} in folgender Weise

$$\text{vec}(X) := (\mathbf{x}_{(j)})_{j=1}^{n_1 n_2} \in \mathbb{R}^{n_1 n_2}$$

mit $\mathbf{x}_{(k_1+(k_2-1) \cdot n_1)} := X(k_1, k_2)$, $k_1 \in \{1, \dots, n_1\}$, $k_2 \in \{1, \dots, n_2\}$ (spaltenweises Durchlaufen der Matrix).

Aus (6.2) folgt dann nach Anwendung der vec-Operation:

$$\begin{aligned} \mathcal{A}_\alpha \mathbf{u} &\approx \begin{pmatrix} A^{11} \vec{u}_1 + A^{12} \vec{u}_2 \\ A^{21} \vec{u}_1 + A^{22} \vec{u}_2 \end{pmatrix} \\ &= \begin{pmatrix} A^{11} & A^{12} \\ A^{21} & A^{22} \end{pmatrix} \cdot \begin{pmatrix} \vec{u}_1 \\ \vec{u}_2 \end{pmatrix} = A \vec{\mathbf{u}}. \end{aligned}$$

Die Matrix $A \in \mathbb{R}^{(2n_1 n_2) \times (2n_1 n_2)}$ besteht aus vier Blöcken $A^{pq} \in \mathbb{R}^{(n_1 n_2) \times (n_1 n_2)}$, $p, q = 1, 2$, wobei n_1 und n_2 die Anzahl der Punkte in x_1 - und x_2 -Richtung bezeichnen.

Die so generierte Matrix A hat eine spezielle Struktur (siehe Abb. 6.1), ist dünn besetzt (wenig von Null verschiedene Einträge) und blocksymmetrisch mit symmetrischen Blöcken, d.h. die Blöcke sind symmetrisch angeordnet und jeder Block selbst ist symmetrisch. Wählen wir nun periodische, statt wie anfangs motiviert Dirichlet, Randbedingungen, so haben die Blöcke neben der Symmetrie eine weitere Eigenschaft, die wir im weiteren Verlauf ausnutzen wollen. Jeder einzelne Block ist damit zirkulant [Dav79] und jeder Block ist ebenfalls zirkulant angeordnet. Nachfolgend werden wir uns die Struktur genauer anschauen und faktorisieren. Zuvor definieren wir noch eine Operation, die uns die Faktorisierung erleichtert.

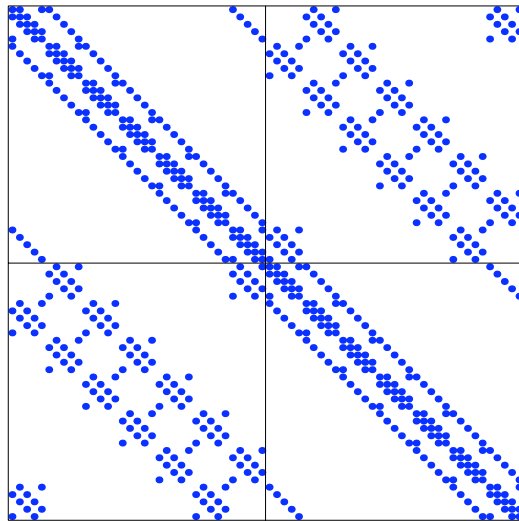


Abbildung 6.1: spy-plot der Matrix $A \in \mathbb{R}^{(2n_1n_2) \times (2n_1n_2)} = \mathbb{R}^{70 \times 70}$ mit $n_1 = 5$, $n_2 = 7$ und periodischen Randbedingungen (Anzahl der Einträge ungleich Null: 630 von 4900). Die von Null verschiedenen Einträge der Matrix sind mit einem Punkt gekennzeichnet. Zur besseren optischen Differenzierung sind die jeweils $((n_1n_2) \times (n_1n_2))$ -großen Blöcke durch Linien voneinander getrennt.

Definition 6.1.2 (Kronecker-Produkt)

Seien $A \in \mathbb{C}^{m \times n}$ und $B \in \mathbb{C}^{p \times q}$. Dann heißt

$$A \otimes B := \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ \vdots & \ddots & \ddots & \vdots \\ a_{m1}B & \dots & \dots & a_{mn}B \end{pmatrix} \in \mathbb{C}^{mp \times nq}$$

Kronecker-Produkt von A und B .

Die Matrix $A \in \mathbb{R}^{(2n_1n_2) \times (2n_1n_2)}$ ist eine 2-mal-2 Blockmatrix

$$A = \begin{pmatrix} A^{11} & A^{12} \\ A^{21} & A^{22} \end{pmatrix} \in \mathbb{R}^{(2n_1n_2) \times (2n_1n_2)} \tag{6.3}$$

mit blockzirkulanten Matrizen A^{pq} , $p, q = 1, 2$,

$$\begin{aligned} A^{pq} &= C_{n_2}^{-1} \otimes A_3^{pq} + C_{n_2}^0 \otimes A_2^{pq} + C_{n_2}^1 \otimes A_1^{pq} \\ &= \begin{pmatrix} A_2^{pq} & A_1^{pq} & A_3^{pq} \\ A_3^{pq} & \ddots & \ddots \\ A_1^{pq} & \ddots & \ddots & A_1^{pq} \\ A_3^{pq} & A_3^{pq} & A_2^{pq} \end{pmatrix} \in \mathbb{R}^{(n_1n_2) \times (n_1n_2)} \end{aligned} \tag{6.4}$$

mit der elementaren zirkulanten Matrix C_{n_2} und zirkulanten Blöcken A_s^{pq} , $s = 1, 2, 3$,

$$\begin{aligned} A_s^{pq} &= S_{3s}^{pq} \cdot C_{n_1}^{-1} + S_{2s}^{pq} \cdot C_{n_1}^0 + S_{1s}^{pq} \cdot C_{n_1}^1 \quad (6.5) \\ &= \begin{pmatrix} S_{2s}^{pq} & S_{1s}^{pq} & S_{3s}^{pq} \\ S_{3s}^{pq} & \ddots & \ddots \\ & \ddots & \ddots & S_{1s}^{pq} \\ S_{1s}^{pq} & & S_{3s}^{pq} & S_{2s}^{pq} \end{pmatrix} \in \mathbb{R}^{n_1 \times n_1} \end{aligned}$$

und den Filterkernen S^{pq} aus (6.1).

Satz 6.1.3 (Diagonalisierung von zirkulanten Matrizen)

Sei $n \in \mathbb{N}$ und $F_n := \frac{1}{\sqrt{n}} \left(\omega_n^{(j-1)(k-1)} \right)_{j,k=1,\dots,n}$, mit $\omega_n := \exp(-2\pi i/n)$ eine Fourier-Matrix der Dimension n . Dann gilt, dass jede beliebige zirkulante Matrix $Z_n = \sum_{j=1}^n z_j(C_n)^{j-1}$ diagonalisierbar ist durch eine Fourier-Matrix F_n :

$$F_n^H Z_n F_n = \sum_{j=1}^n z_j (\mathcal{W}_n)^{j-1}.$$

Der Beweis dieses Satzes findet sich in [Dav79]. Mit Hilfe von Fourier-Matrizen können die zirkulanten Teilmatrizen diagonalisiert werden.

Satz 6.1.4 (Faktorisierung einer strukturierten Matrix)

Sei $n_1, n_2 \in \mathbb{N}$, $p, q = 1, 2$, $s = 1, 2, 3$.

1. Die Matrizen $A_s^{pq} \in \mathbb{R}^{n_1 \times n_2}$ (siehe (6.5)) werden durch F_{n_1} diagonalisiert:

$$\begin{aligned} D_s^{pq} &:= F_{n_1}^H A_s^{pq} F_{n_1} \\ &= \text{diag}(D_{s;j}^{pq}, j = 1, \dots, n_1) \\ \text{mit } D_{s;j}^{pq} &= S_{2s}^{pq} + S_{3s}^{pq} \bar{\omega}_{n_1}^{j-1} + S_{1s}^{pq} \omega_{n_1}^{j-1}. \end{aligned}$$

2. Die Matrizen $A^{pq} \in \mathbb{R}^{(n_1 n_2) \times (n_1 n_2)}$ (siehe (6.4)) werden durch $F_{n_2} \otimes F_{n_1}$ diagonalisiert:

$$\begin{aligned} D^{pq} &:= (F_{n_2} \otimes F_{n_1})^H A^{pq} (F_{n_2} \otimes F_{n_1}) \\ &= \text{diag} \left(D_{k;j}^{pq}, \begin{array}{l} j = 1, \dots, n_1 \\ k = 1, \dots, n_2 \end{array} \right) \\ \text{mit } D_{k;j}^{pq} &= D_{2;j}^{pq} + \bar{\omega}_{n_2}^{k-1} D_{3;j}^{pq} + \omega_{n_2}^{k-1} D_{1;j}^{pq}. \end{aligned}$$

3. Die Matrix $A \in \mathbb{R}^{(2n_1 n_2) \times (2n_1 n_2)}$ (siehe (6.3)) wird durch $F = I_2 \otimes F_{n_2} \otimes F_{n_1}$ blockdiagonalisiert:

$$F^H A F = \begin{pmatrix} D^{11} & D^{12} \\ D^{21} & D^{22} \end{pmatrix}.$$

Der Beweis dieses Satzes findet sich in [FM99].

Der wesentliche Vorteil dieser Faktorisierung besteht darin, dass nunmehr nicht mehr die ganze Matrix gespeichert werden muss, sondern ausschließlich die Hauptdiagonalen der Diagonalblöcke. Dadurch ist der Speicheraufwand nur noch linear und nicht, wie ursprünglich, quadratisch in der Anzahl der Pixel der Ausgangsdaten.

Durch die Approximation der Navier-Lamé-Gleichungen mit Finiten Differenzen und der Fixpunktiteration, ist aus dem System semilinearer partieller Differentialgleichungen ein lineares Gleichungssystem $A\vec{u} = \vec{f}$ geworden. Eine Lösungsmöglichkeit besteht nun durch Invertieren der Matrix A . Die Lösung des Gleichungssystem ergibt sich dann durch $\vec{u} = A^{-1}\vec{f}$. Matrix-Invertierungen sind im allgemeinen sehr teuer, so dass sich diese Variante aus praktischen Gründen eigentlich nicht ergibt. Die vorausgegangenen Strukturierung der Matrix A und die Wirkung der Inversen von $F^H A F$ schenkt uns allerdings eine neue Möglichkeit. Durch die Fouriertransformation ist eine Matrix mit Diagonalblöcken entstanden,

d.h. in jedem Block liegen die von Null verschiedenen Einträge ausschließlich auf der Hauptdiagonalen. Diagonalmatrizen lassen sich sehr einfach invertieren. Insofern die Diagonalmatrizen nicht singulär sind, entsteht die inverse Matrix einer Diagonalmatrix durch invertieren eines jeden Diagonalelements. Durch die spezielle Wahl der Filter S^{pq} , $p, q = 1, 2$ ist die Matrix A im allgemeinen singulär und damit auch die Diagonalblöcke von A . Wir berechnen daher statt der einfachen Inversen D^{-1} die Moore-Penrose Pseudoinverse D^\dagger . Die Pseudoinverse wird über eine Singulärwertzerlegung (SVD) [GvL89] berechnet.

Mit Hilfe der Fourier-Methode lässt sich die Lösung u des Gleichungssystems wie folgt berechnen:

$$A\mathbf{u} = \mathbf{f}$$

$$\iff F^H A F F^H \mathbf{u} = F^H \mathbf{f} \quad (6.6)$$

$$\implies \mathbf{u} = F D^\dagger F^H \mathbf{f}. \quad (6.7)$$

Die Faktorisierung der Matrix A nach Satz 6.1.4 und der Weg über die Pseudoinverse führt zur Pseudoinversen von A .

Satz 6.1.5 (Pseudoinverse von A)

Für $A \in \mathbb{R}^{(2n_1 n_2) \times (2n_1 n_2)}$ wie in Gleichung (6.3) und Filtern $S^{pq} \in \mathbb{R}^{3 \times 3}$ wie in Gleichung (6.1) gilt

$$A^\dagger = F D^\dagger F^H$$

mit

$$D^\dagger := \begin{pmatrix} D^{11\dagger} & D^{12\dagger} \\ D^{21\dagger} & D^{22\dagger} \end{pmatrix}.$$

Die Diagonalmatrizen $D^{pq\dagger} = \text{diag}(D_{kj}^{pq\dagger})$ sind definiert durch

$$\begin{pmatrix} D_{kj}^{11\dagger} & D_{kj}^{12\dagger} \\ D_{kj}^{21\dagger} & D_{kj}^{22\dagger} \end{pmatrix} := \begin{pmatrix} D_{kj}^{11} & D_{kj}^{12} \\ D_{kj}^{21} & D_{kj}^{22} \end{pmatrix}^\dagger, \quad \begin{matrix} j = 1, \dots, n_1, \\ k = 1, \dots, n_2. \end{matrix}$$

Der Beweis des Satzes findet sich in Buch von Modersitzki [Mod04].

Der Vollständigkeit halber führen wir hier noch kurz die konkreten Einträge der

Diagonalelemente auf,

$$\begin{aligned} D_{kj}^{11} &= D_{2;j}^{11} + \bar{\omega}_{n_2}^{k-1} D_{3;j}^{11} + \omega_{n_2}^{k-1} D_{1;j}^{11} \\ &= S_{22}^{11} + S_{12}^{11} 2 \cos(2\pi(j-1)/n_1) + S_{21}^{11} 2 \cos(2\pi(k-1)/n_2), \end{aligned}$$

$$\begin{aligned} D_{kj}^{12} &= D_{kj}^{21} = D_{2;j}^{12} + \bar{\omega}_{n_2}^{k-1} D_{3;j}^{12} + \omega_{n_2}^{k-1} D_{1;j}^{12} \\ &= -4 \cdot S_{11}^{12} \cdot \sin\left(\frac{2\pi(j-1)}{n_1}\right) \cdot \sin\left(\frac{2\pi(k-1)}{n_2}\right), \end{aligned}$$

$$\begin{aligned} D_{kj}^{22} &= D_{2;j}^{22} + \bar{\omega}_{n_2}^{k-1} D_{3;j}^{22} + \omega_{n_2}^{k-1} D_{1;j}^{22} \\ &= S_{22}^{11} + S_{21}^{11} 2 \cos(2\pi(j-1)/n_1) + S_{12}^{11} 2 \cos(2\pi(k-1)/n_2). \end{aligned}$$

Eine ausführliche Herleitung findet sich in [Wir03].

Die angeführte Faktorisierung der Matrix A über die Fourier-Matrizen kann praktisch mit der schnellen Fourier-Transformation (FFT) und der schnellen inversen Fourier-Transformation (iFFT) durchgeführt werden. In Algorithmus 6.1 fassen wir

Algorithmus 6.1 Lösung der diskreten Navier-Lamé-Gleichungen

```

1: for  $d = 1, 2$  do
2:   berechne  $F^d := \text{fft2}(f_d(X))$ ;
3: end for
4: for  $j = 1, \dots, n_1$  do
5:   for  $k = 1, \dots, n_2$  do
6:     berechne  $\begin{pmatrix} U_{kj}^1 \\ U_{kj}^2 \end{pmatrix} := \begin{pmatrix} D_{kj}^{11} & D_{kj}^{12} \\ D_{kj}^{21} & D_{kj}^{22} \end{pmatrix}^\dagger \begin{pmatrix} F_{kj}^1 \\ F_{kj}^2 \end{pmatrix}$ ;
7:   end for;
8: end for;
9: for  $d = 1, 2$  do
10:  berechne  $U^d := \text{ifft2}(U^d)$ ;
11: end for;
```

den gesamten Lösungsschritt algorithmisch zusammen. Für einen Schritt der zur Linearisierung verwendeten Fixpunktiteration kann die Lösung direkt berechnet werden.

Komplexität des Algorithmus

Die Gesamtkomplexität des Verfahrens lässt sich am besten an Algorithmus 6.1 ersehen. Die Fourier-Transformation und ihre Inverse lässt sich durch die FFT (im zweidimensionalen durch die Funktion `fft2`) bzw. iFFT (im zweidimensionalen durch die Funktion `ifft2`) als $\mathcal{O}(N \log N)$ -Algorithmus realisieren (N ist die Anzahl der Pixel in einem Bild $B \in \mathbb{R}^{n_1 \times n_2}$, $N = n_1 n_2$). Die Kraft \mathbf{f} kann linear in N berechnet werden. Desweiteren werden N viele Matrix-Vektor-Operationen der

Größe ($d \times d$), also wieder linearer Aufwand, durchgeführt. Der Berechnungsaufwand für die Bestimmung der Pseudoinversen für Matrizen der Größe ($d \times d$) ist konstant. Damit ergibt sich eine Gesamtkomplexität bei einer festen Anzahl an Iterationen von $\mathcal{O}(N \log N)$.

6.1.2 Approximatives Lösungsverfahren mit FIR-Filtern

Eine alternative Möglichkeit zur Lösung der partiellen Differentialgleichung aus Gleichung (B.14) besteht über einen Ansatz, den Bro-Nielsen et al. [BN96, BNG96] vorgeschlagen haben. Sie verwenden FIR (finite-impulse-respond) Filter. Dieser Filter besteht aus den Eigenwerten und Eigenvektoren des Elastizitätsoperators (siehe Definition 4.2.1).

Für konstante Kraft und Zeit ist die PDE linear. Daher definiert man einen linearen Elastizitätsoperator analog zu (4.6) durch:

$$\mathcal{A}\mathbf{u} := (\mu + \lambda)\nabla(\operatorname{div} \mathbf{u}) + \mu\Delta\mathbf{u}. \quad (6.8)$$

Nach Christensen [Chr94] hat dieser Operator unter den Randbedingungen (sliding boundary conditions)

$$\begin{aligned} u_1(0, x_2) = u_1(1, x_2) = u_2(x_1, 0) = u_2(x_1, 1) &= 0 && \text{(Dirichlet)} \\ \frac{\partial u_1}{\partial x_2} \Big|_{(x_1, 0)} = \frac{\partial u_1}{\partial x_2} \Big|_{(x_1, 1)} = \frac{\partial u_2}{\partial x_1} \Big|_{(0, x_2)} = \frac{\partial u_2}{\partial x_1} \Big|_{(1, x_2)} &= 0 && \text{(Neumann)} \end{aligned} \quad (6.9)$$

auf dem Gebiet $\Omega = [0, 1] \times [0, 1]$ die Eigenwerte

$$\begin{aligned} \kappa_{1,i,j} &= -\pi^2(2\mu + \lambda)(i^2 + j^2) \\ \kappa_{2,i,j} &= -\pi^2\mu(i^2 + j^2) \end{aligned} \quad (6.10)$$

mit den zugehörigen Eigenvektoren

$$\begin{aligned} \varphi_{1ij}(\mathbf{x}) &= \alpha(i, j) \begin{bmatrix} i \sin(i\pi x_1) \cos(j\pi x_2) \\ j \cos(i\pi x_1) \sin(j\pi x_2) \end{bmatrix} \\ \varphi_{2ij}(\mathbf{x}) &= \alpha(i, j) \begin{bmatrix} -j \sin(i\pi x_1) \cos(j\pi x_2) \\ i \cos(i\pi x_1) \sin(j\pi x_2) \end{bmatrix} \end{aligned} \quad (6.11)$$

und Normierungsfaktoren

$$\alpha(i, j) = \sqrt{\frac{4}{\Lambda_{ij}(i^2 + j^2)}}$$

mit

$$\Lambda_{ij} = \begin{cases} 1 & \text{wenn keiner der } i, j \text{ Null ist} \\ 2 & \text{wenn einer der } i, j \text{ Null ist} \end{cases}.$$

Die Normierungsfaktoren bringen die Eigenvektoren auf die Länge 1.

Damit gilt die Eigenwertgleichung:

$$\mathcal{A}\varphi_{kij}(\mathbf{x}) = \kappa\varphi_{kij}(\mathbf{x}), \quad k = 1, 2 \quad i, j \geq 0 \quad \mathbf{x} \in \Omega. \quad (6.12)$$

Berechnung der Impulsantwort von \mathcal{A}

Bezüglich der durch die Eigenwertzerlegung erhaltenen Basis ist die Impulsantwort des linearen Operators \mathcal{L} gesucht. Die Impulsantwort in x_1 -Richtung einer Impulskraft $\mathbf{f}_z = \begin{pmatrix} \delta(\mathbf{x} - \mathbf{z}) \\ 0 \end{pmatrix}$, $\mathbf{z} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$ in der Mitte \mathbf{z} von Ω ist bestimmt als ein Verrückungsfeld.

Mit Hilfe der orthonormalen Basis aus Eigenfunktionen läßt sich die Deformation \mathbf{u} schreiben als:

$$\mathbf{u}(\mathbf{x}) = \sum_{i,j=0}^{\infty} \sum_{k=0}^2 a_{kij} \varphi_{kij}(\mathbf{x}) \quad (6.13)$$

mit Entwicklungskoeffizienten a_{kij} . Mit (6.8), (6.12) und (6.13) ergibt sich:

$$\begin{aligned} \mathcal{A}\mathbf{u}(\mathbf{x}) &= -\mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})) && \text{nach (6.8)} \\ \mathcal{A} \sum_{i,j=0}^{\infty} \sum_{k=0}^2 a_{kij} \varphi_{kij}(\mathbf{x}) &= -\mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})) && \text{nach (6.13)} \\ \sum_{i,j=0}^{\infty} \sum_{k=0}^2 a_{kij} \kappa_{kij} \varphi_{kij}(\mathbf{x}) &= -\mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})) && \text{nach (6.12)}. \end{aligned} \quad (6.14)$$

Gesucht sind nunmehr die Entwicklungskoeffizienten a_{kij} . Bildet man das innere Produkt $\langle a, b \rangle_2 = \int_{\Omega} a^T b \, d\mathbf{x}$ von (6.14) mit der Impulskraft \mathbf{f}_z und $\varphi_{lmn}(\mathbf{x})$ mit $l = 1, 2$ und $m, n \geq 0$ beliebig, aber fest, so ergibt sich für die x_1 -Richtung:

$$\begin{aligned} \left\langle \sum_{i,j=0}^{\infty} \sum_{k=0}^2 a_{kij} \kappa_{kij} \varphi_{kij}(\mathbf{x}), \varphi_{lmn}(\mathbf{x}) \right\rangle_2 &= \langle -\mathbf{f}_z, \varphi_{lmn}(\mathbf{x}) \rangle_2 \\ \implies a_{lmn} \kappa_{lmn} &= \langle -\mathbf{f}_z, \varphi_{lmn}(\mathbf{x}) \rangle_2. \end{aligned}$$

Der Schritt von der ersten zur zweiten Zeile ist leicht nachzuvollziehen mit der Tatsache, dass $\langle \varphi_{kij}, \varphi_{lmn} \rangle_2 = 0$ für $\{kij\} \neq \{lmn\}$. Löst man nun nach a_{lmn} auf, so folgt

$$a_{lmn} = -\frac{1}{\kappa_{lmn}} \langle \mathbf{f}_z, \varphi_{lmn}(\mathbf{x}) \rangle_2 = -\frac{1}{\kappa_{lmn}} \varphi_{lmn}^{x_1}(\mathbf{z}), \quad (6.15)$$

wobei $\varphi_{lmn}^{x_1}$ die x_1 -Komponente von φ_{lmn} ist.

Nun können wir die Zerlegung von \mathbf{u} komplett aufschreiben (vgl. (6.13)). Für

eine Impulskraft $\mathbf{f}_z = (\delta(\mathbf{x} - \mathbf{z}), 0)^\top$ in x_1 -Richtung bei $\mathbf{z} = (0.5, 0.5)^\top$ ergibt sich:

$$\begin{aligned}
 \mathbf{u}^{x_1}(\mathbf{x}) &= \sum_{i,j=0}^{\infty} \sum_{k=0}^2 a_{kij} \varphi_{kij}(\mathbf{x}) \\
 &= - \sum_{i,j=0}^{\infty} \sum_{k=0}^2 \frac{1}{\kappa_{kij}} \varphi_{kij}^{x_1}(\mathbf{x}) \varphi_{kij}(\mathbf{x}) \\
 &= \frac{4}{\pi^2 \mu (2\mu + \lambda)} \sum_{i,j=0}^{\infty} \frac{\sin i \frac{\pi}{2} \cos j \frac{\pi}{2}}{(i^2 + j^2)^2 \Lambda_{ij}} \begin{bmatrix} (\mu i^2 + (2\mu + \lambda) j^2) \sin i \pi x_1 \cos j \pi x_2 \\ -(\mu + \lambda) i j \cos i \pi x_1 \sin j \pi x_2 \end{bmatrix}.
 \end{aligned} \tag{6.16}$$

Entsprechend für die Impulskraft in x_2 -Richtung:

$$\mathbf{u}^{x_2}(\mathbf{x}) = \frac{4}{\pi^2 \mu (2\mu + \lambda)} \sum_{i,j=0}^{\infty} \frac{\cos i \frac{\pi}{2} \sin j \frac{\pi}{2}}{(i^2 + j^2)^2 \Lambda_{ij}} \begin{bmatrix} -(\mu + \lambda) i j \sin i \pi x_1 \cos j \pi x_2 \\ ((2\mu + \lambda) i^2 + \mu j^2) \cos i \pi x_1 \sin j \pi x_2 \end{bmatrix}. \tag{6.17}$$

Diskretisierung der Impulsantwort

Allgemein gesprochen ist die im vorigen Abschnitt hergeleitete Impulsantwort bereits der benötigte Filter. Im kontinuierlichen Fall erzeugt allerdings eine auf einen einzigen Punkt wirkende Kraft (Impulskraft) eine unendlich große Verrückung dieses speziellen Punktes. Im diskreten Fall wird daher der Filter nur auf ein diskretes Gitter projiziert und eine Tiefpaßfilterung unter Nyquist Bedingungen, d.h. Frequenzen oberhalb der Nyquistrate werden abgeschnitten (Beschränkung der Summe auf einen endlichen Wert, durchgeführt, um Alias-Effekte zu vermeiden). Dadurch wird die Kraft gleichzeitig über einen kleinen Bereich geglättet. Der Begriff der Frequenz ist in diesem Zusammenhang dadurch gegeben, dass Detailinformationen in einem Bild im Spektrum durch hohe Frequenzen repräsentiert sind. Eine weiter führende Erläuterung findet sich in [Jäh97].

Die Zerlegung der Impulsantwort mit Hilfe der Eigenvektoren ist frequenzbestimmt, d.h. große i und j gehören zu großen Frequenzen und kleine zu kleinen Frequenzen. Dadurch wird eine ideale Tiefpaßfilterung erreicht, wenn in der Reihenentwicklung nur bis zu einem bestimmten Glied gegangen und der Rest abgeschnitten wird.

Nachfolgend habe der Filter nun die Größe $D \times D$, mit D ungerade, über $\Omega = [0, 1] \times [0, 1]$. Das Abtastintervall ist dann $d = \frac{1}{D-1}$. Nach Shannons Abtasttheorem ergibt sich dann eine Grenzfrequenz f aus der Ungleichung $d \leq \frac{1}{2f}$ bzw. $f \leq \frac{1}{2d} = \frac{D-1}{2}$. Die Grenzfrequenzen lauten dann mit den aus Gleichung (6.16) und (6.17) Summationsvariablen

$$F_i = \frac{i}{2} \quad F_j = \frac{j}{2}$$

und damit ergeben sich die „Abschneidepunkte“ $i = j = D - 1$. Setzt man alles zusammen ergibt sich folgender Filter:

Definition 6.1.6 (Linearer elastischer Filter)

Sei $D \times D$, D ungerade, die vorgegebene Filtergröße und $\mathbf{y} = (y_1, y_2)^T$, mit $y_k \in [-\frac{D-1}{2}, \frac{D-1}{2}] \cap \mathbb{Z}$, $k = 1, 2$ ein diskretes Gitter. Dann sind

$$\Theta^{x_1}(\mathbf{x}) = \frac{4}{\pi^2 \mu (2\mu + \lambda)} \sum_{i,j=0}^{D-1} \frac{\sin i \frac{\pi}{2} \cos j \frac{\pi}{2}}{(i^2 + j^2)^2 \Lambda_{ij}} \begin{bmatrix} (\mu i^2 + (2\mu + \lambda) j^2) \sin i \pi x_1 \cos j \pi x_2 \\ -(\mu + \lambda) i j \cos i \pi x_1 \sin j \pi x_2 \end{bmatrix}$$

und

$$\Theta^{x_2}(\mathbf{x}) = \frac{4}{\pi^2 \mu (2\mu + \lambda)} \sum_{i,j=0}^{D-1} \frac{\cos i \frac{\pi}{2} \sin j \frac{\pi}{2}}{(i^2 + j^2)^2 \Lambda_{ij}} \begin{bmatrix} -(\mu + \lambda) i j \sin i \pi x_1 \cos j \pi x_2 \\ ((2\mu + \lambda) i^2 + \mu j^2) \cos i \pi x_1 \sin j \pi x_2 \end{bmatrix}$$

mit

$$\mathbf{x} = \frac{1}{D-1} \mathbf{y} + \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix},$$

die zu \mathcal{A} gehörenden Filter in x_1 - und x_2 -Richtung.

Die Filter Θ^{x_1} und Θ^{x_2} können für fest gewähltes μ und λ vorberechnet und dann im gesamten Verfahren angewendet werden.

Eine Lösung der partiellen Differentialgleichung in \mathbf{x} kann dann durch die Faltung der Impulsantwort des Operators \mathcal{A} mit der Kraft \mathbf{f} näherungsweise wie folgt berechnet werden:

$$\begin{aligned} \mathbf{u}(\mathbf{x}) &= \mathbf{f}(\mathbf{x}) * \Theta(\mathbf{x}) \\ &= \mathbf{f}^{x_1}(\mathbf{x}) * \Theta^{x_1}(\mathbf{x}) + \mathbf{f}^{x_2}(\mathbf{x}) * \Theta^{x_2}(\mathbf{x}) \\ &= \sum_{i,j=-n}^n (f^{x_1}(\mathbf{x} - \mathbf{y}_{ij}) \cdot \Theta^{x_1}(\mathbf{y}_{ij}) + f^{x_2}(\mathbf{x} - \mathbf{y}_{ij}) \cdot \Theta^{x_2}(\mathbf{y}_{ij})) \end{aligned} \quad (6.18)$$

mit $n = \frac{D-1}{2}$ und $\mathbf{y}_{ij} = (i/n, j/n)^T$.

In Abbildung 6.2 ist das Verrückungsfeld der Filter Θ^{x_1} und Θ^{x_2} der Größe $D = 15$ zu sehen. Die Impulskraft \mathbf{f}_z wurde in $\mathbf{z} = (0.5, 0.5)^T$ angesetzt. Dort ist auch die größte Verrückung sichtbar. Es ist sehr gut erkennbar, wie Material von hinten zum Mittelpunkt herangezogen und nach vorne weg gedrückt wird. Die Filter sind sogar bis auf Rotation um 90° identisch. Eine Matlab-Implementierung hierzu findet sich in Anhang C.2.

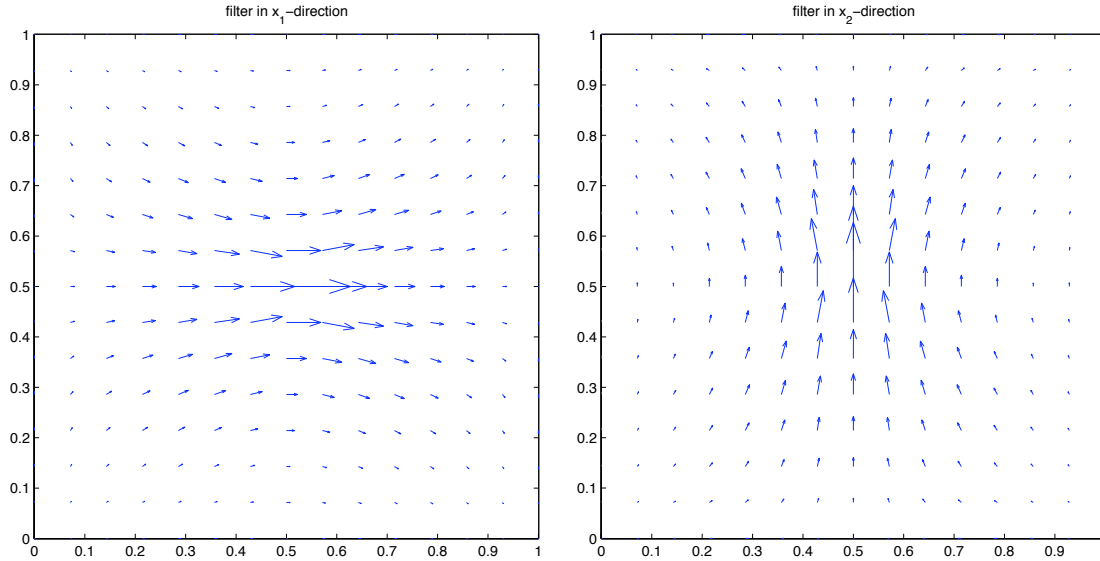


Abbildung 6.2: Darstellung eines 2D-FIR-Filters der Größe 15 über ein Vektorfeld.

6.1.3 Lösungsverfahren zur TV-Regularisierung

Zum Schluss stellen wir ein einfaches Lösungsschema für die TV-Regularisierung vor, die sich streng an die Veröffentlichung von Rudin et al. [ROF92] hält. Zu lösen ist die Differenzengleichung

$$cT_u^2 + \beta c_t = RT_u.$$

Die Randbedingungen setzen wir implizit durch entsprechende Wiederholung der Randpunkte um. Zur besseren Darstellung verwenden wir Matlab- bzw. Python-Notation. Sei nun c ein Korrekturterm mit $n_1 \times n_2$ Pixeln. Mit $c[i, j]$ wird dann der Wert eines Korrekturterms an der Bildkoordinate (i, j) , d.h. i -te Zeile und j -te Spalte im Bild, bezeichnet. $c[:, :] = c[1 : n_1, 1 : n_2]$ bezeichnet dann das ganze Bild, $c[:, 1]$ die erste Spalte und $c[1, :]$ die erste Zeile. Die Ableitungen von c berechnen sich durch

$$\begin{aligned} c_{x_1} &\doteq (c[[2 : n_1, n_1], :] - c[[1, 1 : n_1 - 1], :]) / 2, \\ c_{x_2} &\doteq (c[:, [2 : n_2, n_2]] - c[:, [1, 1 : n_2 - 1]]) / 2, \\ c_{x_1 x_1} &\doteq c[[2 : n_1, n_1], :] - 2c + c[[1, 1 : n_1 - 1], :], \\ c_{x_2 x_2} &\doteq c[:, [2 : n_2, n_2]] - 2c + c[:, [1, 1 : n_2 - 1]], \\ c_{x_1 x_2} &\doteq \left(\left(c[[2 : n_1, n_1], [2 : n_2, n_2]] + c[[1, 1 : n_1 - 1], [1, 1 : n_2 - 1]] \right) \right. \\ &\quad \left. - \left(c[[1, 1 : n_1 - 1], [2 : n_2, n_2]] + c[[2 : n_1, n_1], [1, 1 : n_2 - 1]] \right) \right) / 4. \end{aligned}$$

Eine Lösung c der Differenzengleichung berechnet sich durch Algorithmus 6.2. Eine alternative Lösungsstrategie findet sich in der PhD-Arbeit von Brox [Bro05].

Algorithmus 6.2 Lösung der TV-Regularisierung

```

1:  $c \leftarrow 0$ ;
2:  $\delta_t \leftarrow 0.2$ 
3:  $\varepsilon = 1e^{-8}$ 
4: for  $j = 1, \dots, \text{maxIter}$  do
5:    $c_{x_1} \leftarrow \left( c \left[ \begin{smallmatrix} 2 : n_1, n_1 \\ \cdot \end{smallmatrix} \right] - c \left[ \begin{smallmatrix} 1, 1 : n_1 - 1 \\ \cdot \end{smallmatrix} \right] \right) / 2$ ;
6:    $c_{x_2} \leftarrow \left( c \left[ \begin{smallmatrix} \cdot \\ 2 : n_2, n_2 \end{smallmatrix} \right] - c \left[ \begin{smallmatrix} \cdot \\ 1, 1 : n_2 - 1 \end{smallmatrix} \right] \right) / 2$ ;
7:    $c_{x_1 x_1} \leftarrow c \left[ \begin{smallmatrix} 2 : n_1, n_1 \\ \cdot \end{smallmatrix} \right] - 2c + c \left[ \begin{smallmatrix} 1, 1 : n_1 - 1 \\ \cdot \end{smallmatrix} \right]$ ;
8:    $c_{x_2 x_2} \leftarrow c \left[ \begin{smallmatrix} \cdot \\ 2 : n_2, n_2 \end{smallmatrix} \right] - 2c + c \left[ \begin{smallmatrix} \cdot \\ 1, 1 : n_2 - 1 \end{smallmatrix} \right]$ ;
9:    $c_{x_1 x_2} \leftarrow \left( \left( c \left[ \begin{smallmatrix} 2 : n_1, n_1 \\ 2 : n_2, n_2 \end{smallmatrix} \right] + c \left[ \begin{smallmatrix} 1, 1 : n_1 - 1 \\ 1, 1 : n_2 - 1 \end{smallmatrix} \right] \right) \right. \\ \left. - \left( c \left[ \begin{smallmatrix} 1, 1 : n_1 - 1 \\ 2 : n_2, n_2 \end{smallmatrix} \right] + c \left[ \begin{smallmatrix} 2 : n_1, n_1 \\ 1, 1 : n_2 - 1 \end{smallmatrix} \right] \right) \right) / 4$ ;
10:   $c_t \leftarrow \left( c_{x_1 x_1} (\varepsilon^2 + c_{x_2}^2) - 2c_{x_1} c_{x_2} c_{x_1 x_2} + c_{x_2 x_2} (\varepsilon^2 + c_{x_1}^2) \right) / \left( (\varepsilon^2 + c_{x_1}^2 + c_{x_2}^2)^{\frac{3}{2}} \right)$ ;
11:   $c \leftarrow c + \delta_t c_t$ ;
12: end for

```

6.2 Strategien für einen robusteren Algorithmus

In diesem Abschnitt wollen wir nun die Strategien vorstellen, die den Algorithmus robust gemacht haben. Zum einen gehört dazu aus Optimierungssicht eine spezielle Wahl von Abbruchbedingungen, des weiteren ein Multiresolutionsansatz und schließlich eine geeignete Schrittweitensteuerung. Zum anderen gehört dazuein Verfahren zur automatischen Berechnung geeigneter Lamé-Konstanten. Beginnen wollen wir nun mit dem Multiresolutionsansatz.

6.2.1 Multiresolutionsansatz

Bei algorithmischen Optimierungen besteht im allgemeinen die Gefahr, dass das globale Optimum nicht gefunden werden kann, weil der Optimierungsweg durch ein lokales Minimum geführt hat, das nicht mehr verlassen werden kann. Werden Deformationen gesucht, die lokal besonders groß sind, ist die Wahrscheinlichkeit sehr hoch, dass nur ein lokales Minimum durch den Algorithmus gefunden wird, das darüber hinaus noch ein nicht zufrieden stellendes Ergebnis liefert. Um die lokalen Minima im wesentlichen zu umschiffen, verwenden wir einen Multiresolutionsansatz. Bei diesem wird das Optimierungsproblem auf verschiedenen Auflösungsstufen des Bildes betrachtet. In Abbildung 6.3 sind Beispiele für verschiedene Stufen gegeben. Wie dort zu erkennen ist, sind die Bildstufen in pyramidenform aufgebaut. Das gegebene Bild, in Originalauflösung, stellt dabei die feinste und damit unterste Stufe der Pyramide dar. Wir vergeben für die Multiresolutionsstufen Nummern, wobei die unterste Stufe die Nummer 0 erhält. Eine Pyramide mit 4 Stufen wie in Abb. 6.3(a) hat dann die Stufen 0 bis 3.

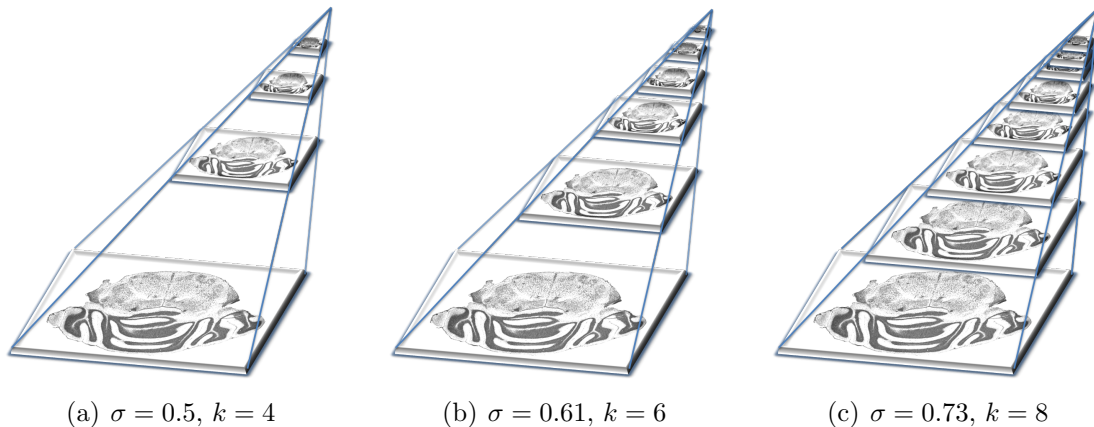


Abbildung 6.3: Bildpyramide mit verschiedenen Skalierungsfaktoren $\sigma \in (0, 1]$ und Stufen $k \in \mathbb{N}^{\geq 1}$. Faktoren $\sigma \gg 0.5$ bauen mehr und damit dichtere Stufen auf (siehe 6.3(c)), so dass der Wechsel von einer Stufe zur Nächsten weniger Detailinformationen hinzu bringt.

Definition 6.2.1 (Bildpyramide)

Sei B ein diskretes Bild (siehe Def. 5.1.2), $\sigma \in (0, 1] \subset \mathbb{R}$ ein Skalierungsfaktor und $k \in \mathbb{N}^{\geq 1}$ die Anzahl der Stufen. Dann wird mit

$$\{B_{(\sigma,0)}, \dots, B_{(\sigma,k)}\}, \quad B_{(\sigma,j)} = B([0, 1]_{\sigma^j \cdot n_1} \times [0, 1]_{\sigma^j \cdot n_2}), j = 0, \dots, k$$

eine *Bildpyramide* bezeichnet. Durch σ und k ist die Pyramide eines Bildes B eindeutig bestimmt.

Je größer die Stufe innerhalb der Pyramide ist, desto weniger Detailinformationen sind im Bild enthalten (vergleiche dazu Abb. 5.5). Dies bedeutet, dass weniger Frequenzen im Bild enthalten sind. Praktisch bedeutet dies, dass bei der Erzeugung der Bildpyramide ein Filter verwendet werden sollte, der genau so viele Frequenzen ausblendet, wie die notwendige Stufe nicht mehr halten kann. Im einfachen Fall kann ein Gaußfilter als Tiefpass verwendet werden. Werden Skalierungsfaktoren verwendet, die 0.5 und kleiner sind, so wird mit steigender Stufennummer von Stufe zu Stufe sehr viel Detailinformation verworfen.

Die Lösung des Optimierungsproblems wird auf dem Gitter mit der höchsten Nummer begonnen. Dort ist die Anzahl der Bildpunkte am geringsten und damit lässt sich sehr schnell eine Lösung des Optimierungsproblems finden. Das Ergebnis wird anschließend auf die nächst feinere Stufe propagiert. Die Startposition für die Optimierung ist hierdurch schon sehr nahe an der globalen Lösung. Mit jeder Stufe und damit mit allen zusätzlichen Details wird sich sukzessive an das globale Optimum angenähert. Je größer σ , desto kleiner ist der Detailzuwachs bei einem

Stufenwechsel und damit um so kleiner die Wahrscheinlichkeit in einem lokalen Optimum stecken zu bleiben. Gleichzeitig hat ein zu großes σ den Nachteil, dass zu viele Iterationen und Stufenwechsel vollzogen werden müssen, bis ein Ergebnis erreicht ist.

6.2.2 Abbruchkriterien

Abbruchkriterien sind entscheidend für den Verlauf iterativer Verfahren im Rahmen der Optimierung. Die Wahl der Abbruchkriterien hat einen direkten Einfluss auf die Laufzeit des Algorithmus. Bei der Verwendung von Multiresolution entscheiden die Abbruchkriterien des weiteren darüber, wann eine Stufe verlassen und auf die nächst feinere Stufe gewechselt werden muss. Sind Abbruchkriterien zu sanft formuliert (Abbruchkriterium greift frühzeitig), so terminiert ein Verfahren unter Umständen zu früh und ist damit weit von einer sinnvollen Lösung entfernt. Im Gegensatz dazu führen zu streng formulierte Abbruchkriterien (greifen sehr spät) zu einer unnötig großen Anzahl an Iterationen, in denen möglicherweise keine Fortschritte gemacht werden.

Gute Abbruchkriterien bestehen daher stets aus einer Sammlung verschiedener Kriterien zur Terminierung, die in einer geschickten Kombination wirksam werden. In der Literatur sind verschiedene Abbruchkriterien für diverse Optimierungsprobleme vorgestellt. Wir haben uns in dieser Arbeit für die Abbruchkriterien nach Gill, Murray & Wright [GMW81] entschieden. Diese Abbruchkriterien bestehen aus fünf Teilen.

Definition 6.2.2 (Abbruchkriterien nach Gill, Murray & Wright)

Sei $f : \Omega \rightarrow \mathbb{R}$ die zu beobachtende Funktion und $\mathbf{x} \in \Omega$ das dazu gehörige Argument. Weiterhin seien ε die Maschinengenauigkeit, $\tau = 10^{-r}$ ein vom Benutzer vorgegebener Toleranzwert (Anzahl der Dezimalstellen, in der die Zielfunktion und die Minimalstelle übereinstimmen sollen), k der Iterationszähler und k_{\max} die maximale Anzahl an Iterationen. Mit

$$(K1) \quad |f_{k-1} - f_k|_2 < \tau(1 + |f_k|_2)$$

$$(K2) \quad \|\mathbf{x}_{k-1} - \mathbf{x}_k\|_2 < \sqrt{\tau}(1 + \|\mathbf{x}_k\|_2)$$

$$(K3) \quad \|\nabla f_k\|_2 \leq \sqrt[3]{\tau}(1 + |f_k|_2)$$

$$(K4) \quad \|\nabla f_k\|_2 < \varepsilon$$

$$(K5) \quad k \geq k_{\max}$$

werden die Kriterien bezeichnet, die geprüft werden. Das Abbruchkriterium ist erfüllt, wenn $(K1) \wedge (K2) \wedge (K3) \vee (K4) \vee (K5)$ erfüllt ist.

Die Kriterien (K4) und (K5) sind hierbei eine Art Notbremse. Dadurch gibt es eine garantierte Terminierung nach einer fest vorgegebenen maximalen Anzahl

an Schritten ($K5$), bzw. wenn bei den Berechnungen Zahlen resultieren, die in der Größenordnung der Maschinengenauigkeit liegen ($K4$). Die übrigen Kriterien überwachen die Änderungen der Funktionswerte ($K1$), die Änderung der Argumente ($K2$) und die Steigung der Funktion. Die Kriterien sind in einer Mischform aus relativer und absoluter Verhältnisse formuliert, wodurch eine zusätzliche Skalierung bzw. Anpassung der Werte nicht notwendig ist. Im Idealfall terminiert das Verfahren, wenn die drei ersten Kriterien erfüllt sind.

6.2.3 Schrittweitensteuerung

Zur Lösung des Optimierungsproblems verwenden wir ein Gradientenabstiegsverfahren. Um in jedem Iterationsschritt ein möglichst ideale Schrittweite zu verwenden und damit zu einer möglichst raschen Konvergenz zu gelangen, verwenden wir die Armijo-Schrittweitensteuerung (siehe Algorithmus 6.3). Diese Schrittweiten-

Algorithmus 6.3 Armijo-Schrittweitenregelung

$\mathbf{x}^{(k)} \in \Omega$, \mathbf{x} im Iterationsschritt k
 $f : \Omega \rightarrow \mathbb{R}$, $f \in C^1(\Omega)$, Optimierungsfunktion
 $\mathbf{s}^{(k)} \in \Omega$, Such- bzw. Abstiegsrichtung zum Iterationsschritt k

- 1: $t \leftarrow 1$;
- 2: $\gamma \in (0, 1)$ beliebig, aber fest;
- 3: $\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$
- 4: **while** $f(\mathbf{x}^{(k)} + t\mathbf{s}^{(k)}) > f(\mathbf{x}^{(k)}) + \gamma t \nabla f(\mathbf{x}^{(k)})^\top \mathbf{s}^{(k)}$ **do**
- 5: $t \leftarrow \frac{t}{2}$
- 6: **end while**

steuerung sichert zu, dass \mathbf{s}_k stets eine echte Abstiegsrichtung ist und ermöglicht dadurch eine Konvergenz des gesamten Verfahrens.

6.2.4 Automatische Wahl der Lamé-Konstanten

Eine große Schwierigkeit bei der elastischen Registrierung ist die konkrete Wahl der Materialkonstanten μ und λ . In der Physik sind die Konstanten für jedes Material fest bestimmt. Wie in Kapitel 2.3 bereits erwähnt wurde, ist dies für die Numerik nicht gegeben. Eine weit verbreitete Lösung dieses Dilemmas ist eine fixe Wahl der Konstanten und eine dynamische Schrittweitenregelung für das Verfahren. Der Nachteil der sukzessiven Skalierung auf eine feste Schrittweite ist allerdings, dass unabhängig von den wirkenden Kräften, die gleiche Deformationsänderung erreicht wird und damit eine konvergentes Verhalten der Deformationsänderung nicht erkennbar ist.

Ideal für das Registrierungsverfahren ist es, wenn durch eine automatische Bestimmung der Materialkonstanten eine zusätzliche Skalierung der Deformationsänderungen innerhalb der Iterationen nicht notwendig ist. Schauen wir uns daher

den Elastizitätsoperator aus Definition 4.2.1 etwas genauer an:

$$\mathcal{A}_\alpha \mathbf{u} = \alpha \left(\mu \Delta \mathbf{u} + (\lambda + \mu) \nabla \operatorname{div} \mathbf{u} \right)$$

Wählen wir speziell $\lambda = 0$, so ist der Operator linear in μ

$$\tilde{\mathcal{A}}_\alpha \mathbf{u} = \mu (\Delta \mathbf{u} + \nabla \operatorname{div} \mathbf{u}).$$

Wählt man nun zu Beginn $\mu = 1$ und löst die partielle Differentialgleichung, so erhält man die direkte Wirkung des Operators als Deformationsupdate. Darauf hin kann

$$\mu = \frac{\mathit{maxStepSize}}{\max_{\mathbf{x} \in \Omega} \|\mathbf{u}(\mathbf{x})\|_2}$$

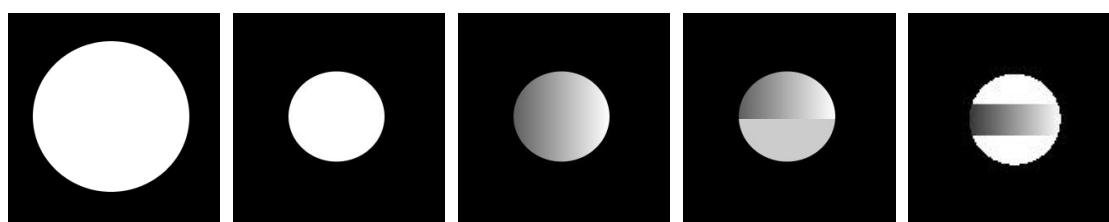
gewählt und das Registrierungsverfahren gestartet werden. Die maximale Schrittweite $\mathit{maxStepSize}$ (in Pixel/Voxel) kann durch den Benutzer festgelegt werden. Die automatische Wahl von μ hat den Vorteil, dass das Verhältnis von Kraft zu Deformation über alle Iterationen gleich bleibt. Der entscheidende Nachteil dieser Methode ist, dass $\lambda = 0$ gesetzt wurde und damit das Kompressionsmodul Null ist. Dadurch ergibt sich stets, unabhängig von der Wahl von μ , eine Poisson-Zahl von 0, was bedeutet, dass es zu einer Längskontraktion keine Querkontraktion gibt und umgekehrt.

7 Ergebnisse und Diskussionen anhand von Beispielen

In diesem Kapitel wollen wir nun Ergebnisse zeigen, die mit den zuvor vorgestellten Methoden erzeugt wurden und daran die Prinzipien, Eigenschaften, Vor- und Nachteile der einzelnen Varianten diskutieren. Um die prinzipiellen algorithmischen Eigenschaften besser studieren zu können, beginnen wir mit akademischen Beispielen. Diese haben den Vorteil, dass sich bestimmte Eigenschaften der Algorithmen überschaubar und einfach darstellen und überprüfen lassen. Akademische Beispiele sind einfach zu beschreiben und die Ergebnisse sind gut nachvollziehbar. Anschließend werden wir zweidimensionale Beispiele aus Histologie und Magnetresonanztomographie ansehen, um so die Wirkung von simultaner Homogenisierung und Registrierung an realen Beispielen zu erfahren. Im letzten Abschnitt dieses Kapitels werden wir dann Ergebnisse der Registrierung von histologischen Serienschnitten vorstellen und diskutieren.

7.1 Akademische Beispiele

In diesem Abschnitt betrachten wir akademische Beispiele, um an ihnen die Wirkung und Funktionsweise der simultanen Homogenisierung und Registrierung zu erfahren. Als einfaches Objekt haben wir einen Kreis ausgewählt. Er dient als einfache Approximation der zumeist sehr komplexen Organe und Strukturen im menschlichen Körper, wie Gehirn/Schädel, Herz, Leber, Niere, Milz, etc. Wir betrachten hier den Idealfall, also ohne zusätzliche Störungen wie Rauschen o.ä. Als Referenzbild R (siehe Abb. 7.1(a)) haben wir ein ideales, rauschfreies Bild



(a) Reference R (b) Template T_1 (c) Template T_2 (d) Template T_3 (e) Template T_4

Abbildung 7.1: Akademisches Beispiele zur Veranschaulichung der Wirkung der simultanen Homogenisierung und Registrierung.

mit homogener Färbung ausgewählt. Um die verschiedenen Aspekte beleuchten zu

können, haben wir insgesamt vier Templatebilder erzeugt. Allen Templatebildern gemein ist die Form, ein Kreis mit deutlich kleinerem Radius, als bei der Referenz. Das erste Templatebild T_1 (siehe Abb. 7.1(b)) ist aus Registrierungssicht der einfachste Fall. Es ist wie die Referenz ebenfalls homogen und besitzt den gleichen Intensitätswert wie das Referenzbild. Im Unterschied dazu sind die Intensitäten in T_2 (siehe Abb. 7.1(c)) mit einer so genannten Rampe linear verändert. Dies simuliert den Diffusionsprozess bei der histologischen Färbung, d.h. das Färbemittel ist trotz gleichem Material räumlich unterschiedlich verteilt. Das dritte Templatebild T_3 (siehe Abb. 7.1(d)) beinhaltet ebenfalls Grauwertveränderungen wie in T_2 , doch ist zusätzlich durch eine horizontale Linie in zwei Bereiche unterteilt. Der obere Teil des Kreises besitzt den gleichen Farbverlauf wie in T_2 , doch der untere Teil ist homogen, allerdings mit einem deutlich dunkleren Grauwert als das Referenzbild. Durch die zusätzliche Kante innerhalb des Objekts ist nicht eindeutig klar, ob es sich um ein Objekt mit deutlich inhomogener Färbung, oder um zwei angrenzende Objekte handelt. Darüber hinaus ist nicht klar, wenn es sich um zwei Objekte handelt, welches der zur Referenz korrespondierende Teil ist. Mit dem vierten Templatebild T_4 (siehe Abb. 7.1(e)) soll ein typischer Färbeverlauf in der Histologie simuliert werden, bei dem ein Objekt nicht vollständig, sondern nur in Teilen histologisch eingefärbt wird. Dadurch entstehen zwei neue Kanten innerhalb des Objekts, die sehr scharf begrenzt sind.

Für alle vier Templatebilder T_1, T_2, T_3, T_4 ist prinzipiell die gleiche Transformation gesucht. Wir gehen davon aus (Vorwissen), dass es sich auf Referenz und Template um *ein* korrespondierendes Objekt handelt, das sich ausschließlich durch ihre Repräsentierung unterscheidet. Die ideale Transformation wäre eine einfache Skalierung, da die Mittelpunkte der Kreise in Referenz und Template identisch sind. Wir erwarten Betrachte wir nun der Reihe nach die Effekte, die bei der Registrierung entstehen. Für jede Kombination aus Referenz und Template werden zwei Registrierungsvorgänge durchgeführt. Im ersten Durchlauf wird ausschließlich registriert, also auf eine simultane Homogenisierung verzichtet. Als zweites wird eine simultane Homogenisierung und Registrierung durchgeführt und der Unterschied zur ersten Variante diskutiert.

Für das erste Beispiel laufen beide Formen der Registrierung identisch ab. Der Korrekturterm für die Homogenisierung ergibt sich zu identisch Eins, $c \equiv 1$. In Abb. 7.2 ist das Ergebnis der Registrierung von R und T_1 gezeigt. In Bild 7.2(c) ist das deformierte Template dargestellt. Das Ergebnis ist perfekt, wodurch sich auf dem Differenzbild 7.2(e) keine Kontraste mehr abzeichnen. Das errechnete Deformationsfeld lässt sich am besten über ein deformiertes Gitterbild 7.2(f) abbilden. Hierin kann sofort erkannt werden, dass die Deformation symmetrisch und damit erwartungskonform verlaufen ist. Bei simultaner Homogenisierung und Registrierung sind deformiertes Template und das Deformationsfeld identisch, wodurch sie hier, der besseren Übersicht halber, nicht abgebildet werden.

Aus Registrierungssicht spannender und interessanter wird es nun bei den restlichen Templatebildern. Betrachten wir nun als nächstes T_2 . Im Unterschied zu T_1 ist hier das Objekt nicht mehr in gleicher Intensität wie im Referenzbild repräsen-

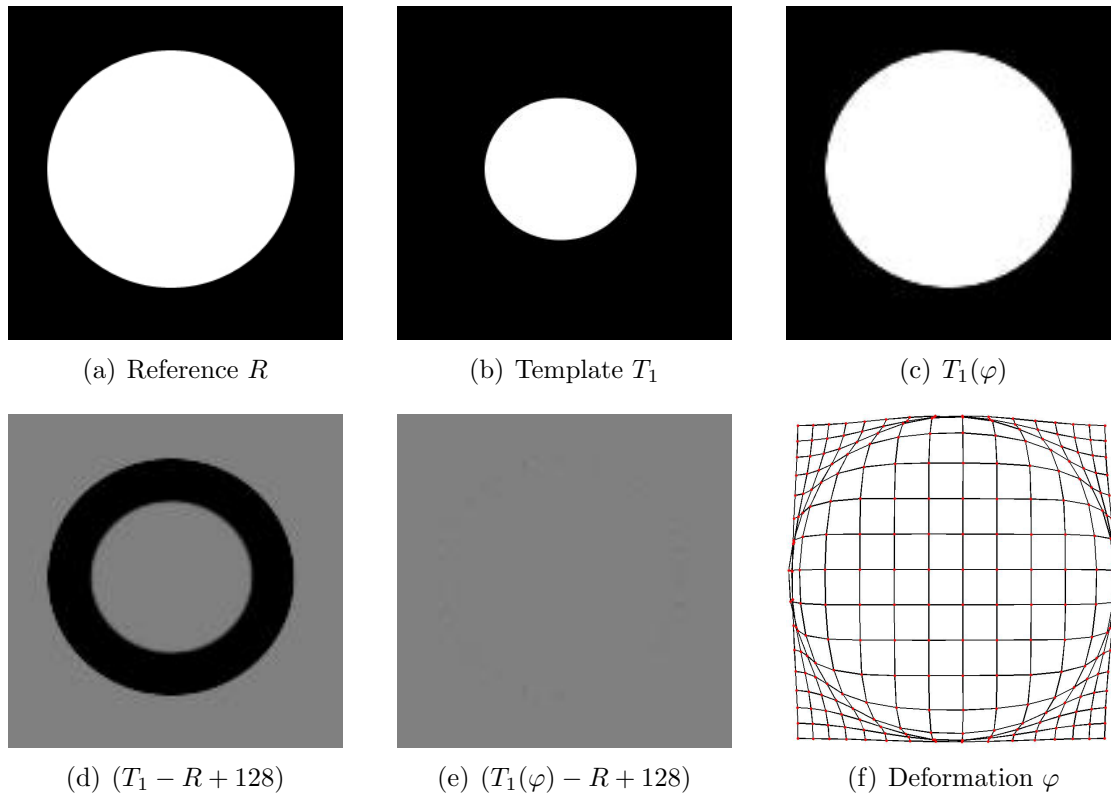


Abbildung 7.2: Bei der Registrierung von homogenen Objekten gleicher Intensität ist eine Homogenisierung nicht notwendig. Das Deformationsfeld ist optimal symmetrisch, das transformierte Templatebild deckt sich perfekt mit dem Referenzbild. In der ersten Reihe sind Referenzbild, Templatebild und deformiertes Templatebild dargestellt. In der zweiten Reihe sind Differenzbilder von Referenz- und Templatebild ohne und mit Anwendung des Deformationsfeldes, das unten rechts in Form eines deformierten Gitters dargestellt ist.

tiert. Darüber hinaus kommt hinzu, dass das Objekt nicht homogen gefärbt ist. In Abb. 7.3 sehen wir das Ergebnis der elastischen Registrierung. Der von den Intensitäten dem Referenzbild am naheliegenden rechte Teil des Templatebildes wird durch die Registrierung bevorzugt, was man im deformierten Gitter (Abb. 7.3(f)) sehr gut als Trend nach links über den Rand hinaus und der starken Asymmetrie erkennen kann. Das deformierte Template entspricht von der äußeren Form her sehr gut der Referenz (siehe Abb. 7.3(c), bzw. 7.3(e)), doch ist die Deformation nicht erwartungskonform. Verwenden wir im Gegensatz dazu die simultane Homogenisierung und Registrierung, ergibt sich eine völlig andere Situation. Der lineare Färbetrend ist im Korrekturterm sehr gut zu erkennen (siehe Abb. 7.4(f)). Die Homogenisierung läuft perfekt ab, so dass das deformierte und homogenisierte Template (Abb. 7.4(d)) perfekt der Referenz entspricht (siehe dazu das Differenzbild 7.4(c)). Ein leichter Unterschied ist im Differenzbild nur an der Außenkante erkennbar. Dies kann allerdings durch die Interpolation entstanden sein. Das De-

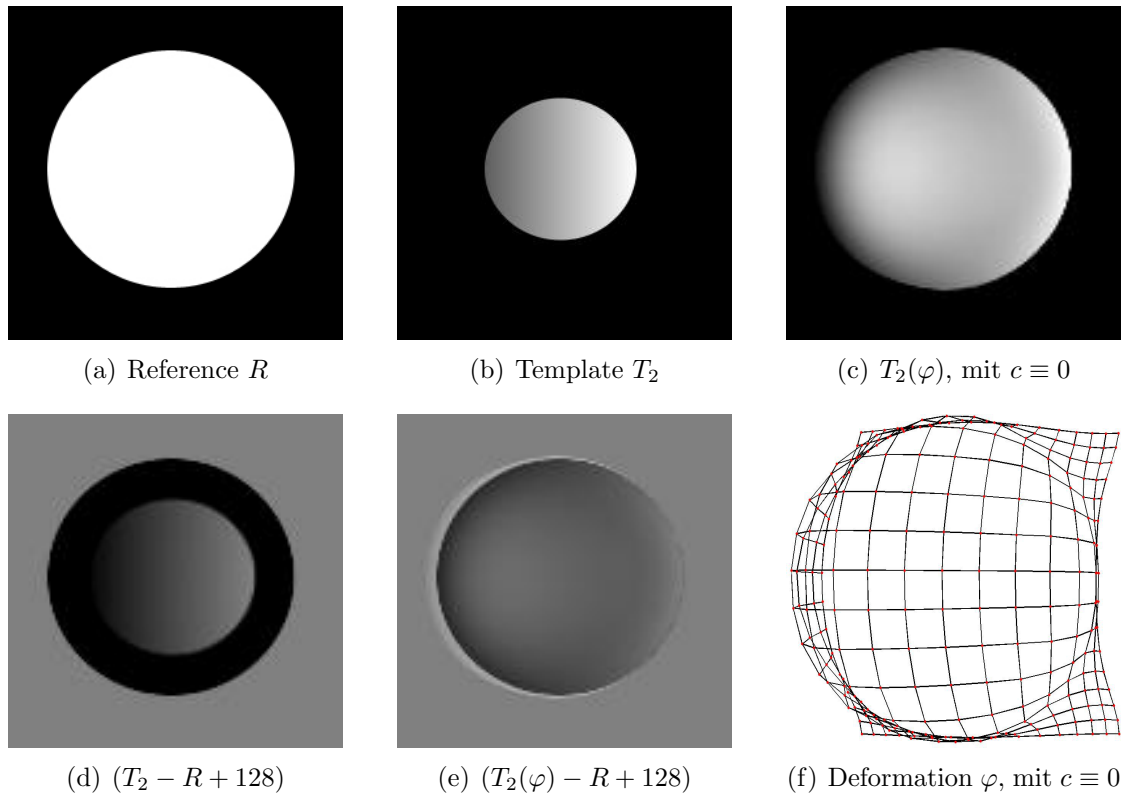


Abbildung 7.3: Registrierung von Objekten mit verschiedenen Intensitätswerten. Das Referenzbild ist homogen repräsentiert mit Intensitätswert 256. Das Templatebild hat einen linearen Färbetrend mit Intensitätswerten von 90 bis 255. Bei reiner Registrierung dominieren die hellen Bildteile und sorgen für einen Trend nach links im Deformationsfeld.

formationsfeld ist nahezu symmetrisch und entspricht fast dem Deformationsfeld aus der Registrierung von T_1 (siehe Abb. 7.2(f)).

Betrachten wir nun das dritte Beispiel (Abb. 7.5). Template T_3 besteht aus zwei verschiedenen Teilen, die durch eine scharfe Kante voneinander getrennt sind. Der untere Teil des Kreises ist homogen und vom Intensitätswert her ähnlich zum Referenzbild. Der obere Teil des Kreises hat einen linearen Färbetrend und ähnelt nur im rechten Teil des Kreisabschnitts dem Referenzobjekt. Wie oben bereits erwähnt besteht für den Registrierungsalgorithmus nun die Schwierigkeit, die Korrespondenz der Objektteile zu identifizieren. Die Gefahr ist nunmehr gegeben, dass der homogene Teil des Kreises gegenüber dem inhomogenen Teil dominiert und sich dadurch massiv ausdehnt, um im Idealfall die äußere Form des Referenzbildes anzunehmen. Durch spezielle Wahl der Materialkonstanten kann gesteuert werden, wie weich oder zäh ein Objekt transformiert werden kann. Wir wählen die Materialkonstanten wie bei den vorigen Beispielen auch. In Abb. 7.5(c) sehen wir das transformierte Templatebild. Während des Registrierungsprozesses sind zwei Trendbewegungen erkennbar. Zum einen drückt der untere, homogene Teil

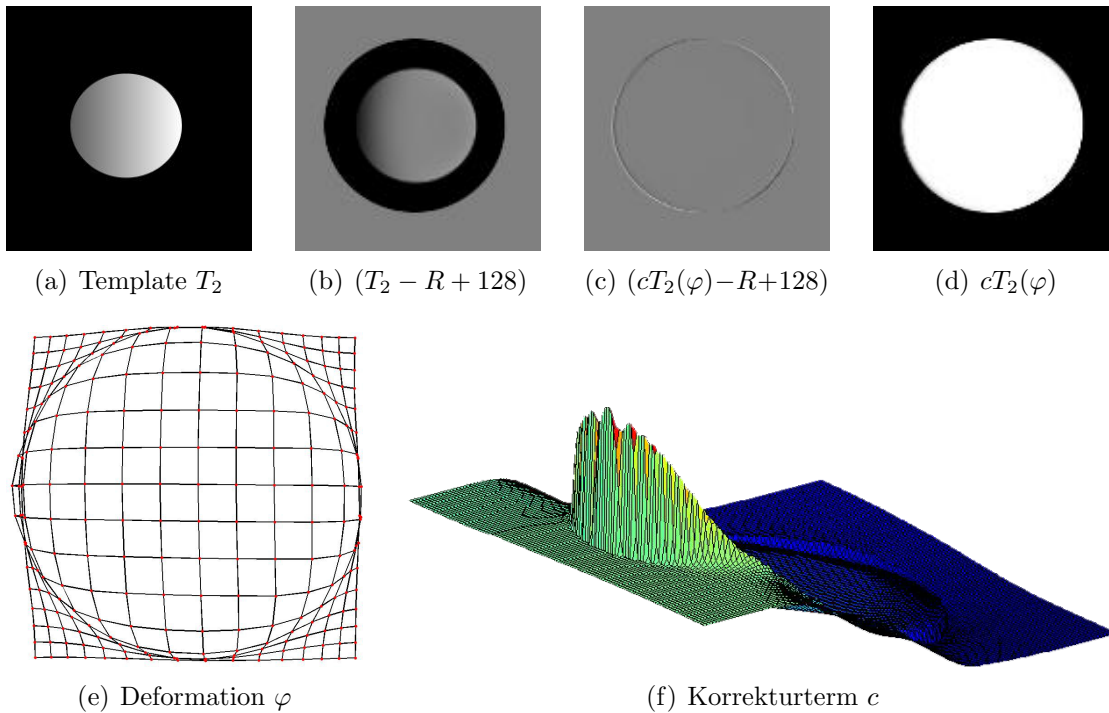


Abbildung 7.4: Simultane Homogenisierung und Registrierung ermöglicht eine erwartungskonforme Deformation. Das Deformationsfeld ist nahezu symmetrisch. Der Korrekturterm c bildet den linearen Intensitätsverlauf unter Erhaltung der Kanten sehr gut ab. Das deformierte und intensitätskorrigierte Template entspricht dem Referenzbild perfekt.

des Kreises nach oben, um die ganze Fläche des Referenzobjekts einzunehmen. Gleichzeitig und damit gegenläufig dazu, drängt der rechte Teil der oberen Hälfte des Kreises nach links und wirkt damit dem Aufwärtstrend von unten entgegen. Als Resultat entsteht ein Trend in diagonaler Richtung nach links oben, was sich im Deformationsfeld (Abb. 7.5(f)) sehr gut erkennen lässt. Durch die Registrierung decken sich schließlich die äußeren Kanten von Referenz und deformiertem Template gut, doch das Differenzbild 7.5(e) zeigt noch deutliche Unterschiede. Insgesamt kann die Transformation als nicht zufrieden stellend eingestuft werden, da eine Deformation entstanden ist, die den Erwartungen deutlich widerspricht.

Bei der simultanen Homogenisierung und Registrierung treten die vorher beobachteten Effekte nicht so stark auf. Zwar lässt sich im Registrierungsverlauf beobachten, dass ebenfalls einen diagonalen Trend nach links oben gibt, doch wirkt er sich nur sehr schwach aus. Darüber hinaus bewirkt die Homogenisierung, dass das transformierte Template perfekt zur Referenz passt. Betrachten wir das Deformationsfeld genauer, so erkennen wir, den leichten diagonalen Trend. Dem entsprechend ist die Symmetrie der Deformation im linken Teil des Bildes leicht gestört. Durch eine intensivere Gewichtung des Homogenisierungsanteils ließe sich die Transformation sicherlich noch mehr Richtung idealer Transformation bringen,

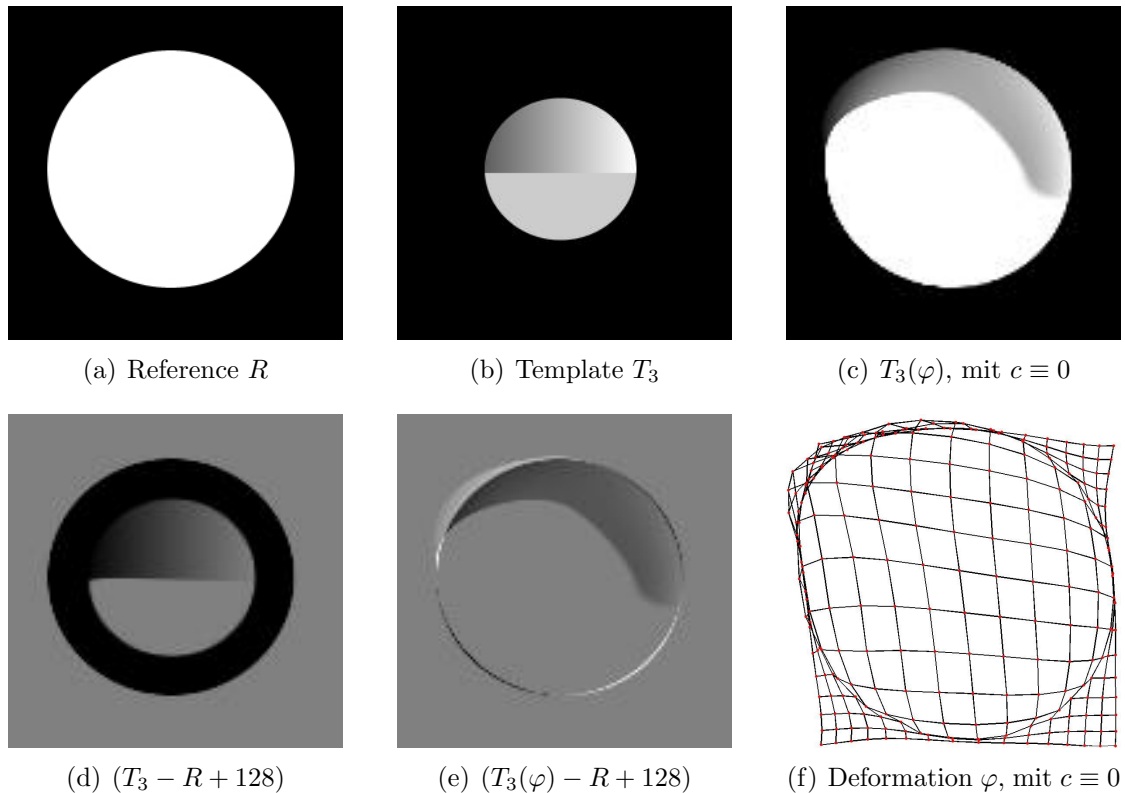


Abbildung 7.5: Registrierung von durch unterschiedliche Färbung inhomogenen Objekts auf ein homogenes Objekt sorgt für unnatürliche und unerwartete Deformationen. Der untere homogene und der Referenz ähnlichere Bereich dominiert die Transformation. Die äußere Form des transformierten Templates ähnelt zwar der äußeren Form der Referenz, doch die Deformation entspricht keineswegs einer symmetrischen Vergrößerung.

doch das setzt voraus, dass bekannt ist, dass es sich um ein und das selbe Objekt bei Referenz und Template handelt. Im allgemeinen Fall der Registrierung sind die Korrespondenzen nicht von vorne herein klar, sondern müssen sich durch die Registrierung erst ergeben. Dem entsprechend kann hier von einer guten Lösung gesprochen werden.

Als letztes akademische Beispiel wollen wir nun T_4 betrachten. Das Besondere dieses Templates ist, dass es sich von T_1 nur durch einen Streifen in horizontaler Richtung mit linearem Färbetrend unterscheidet. Aus algorithmischer Sicht ist es besonders schwer zu entscheiden, ob es sich dabei um ein neues Objekt handelt, dass gegenüber der Referenz hinzu gekommen ist und damit erhalten bleiben soll, ob es sich um das gleiche Objekt handelt, aber anders gefärbt ist, oder ob es sich um eine neues Objekt handelt, das in der Form erhalten bleiben soll und damit weder schrumpfen, noch wachsen darf. Gehen wir davon aus, dass es sich um ein Objekt handelt, so besteht für den Korrekturterm die Herausforderung, einerseits dem starken Färbetrend entgegen zu wirken, aber die Kanten dabei scharf zu erhalten, ohne dabei unstetig zu werden. In Abb. 7.7 sehen wir die Ergebnisse beider

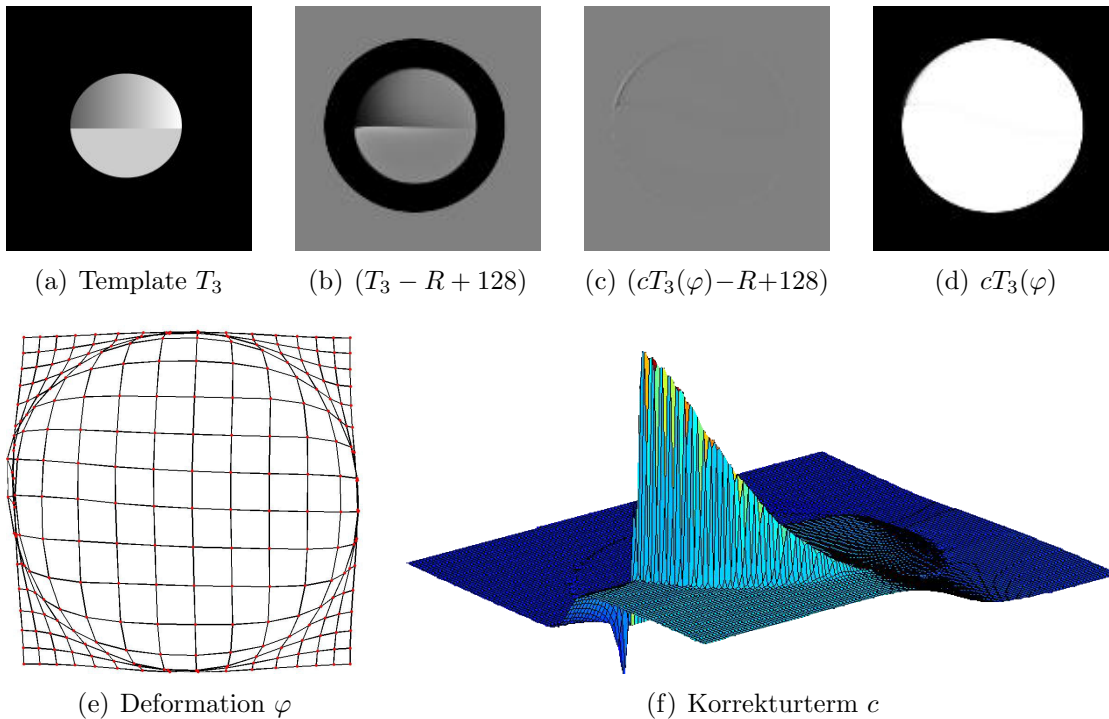


Abbildung 7.6: Simultane Homogenisierung und Registrierung liefert wiederum ein nahezu perfektes Ergebnis. Der Korrekturterm sorgt für eine ideale Intensitätsanpassung. Die Deformation ist dadurch nahezu symmetrisch.

Registrierungsdurchläufe. Bei der Registrierung ohne Intensitätskorrektur entstehen zwei sich gegenseitig beeinflussende Bewegungen. Zum einen zieht sich der Streifen in der Mitte auf der linken Seite zusammen, da die zur Referenz intensitätsähnlichen Teile die Lücke schließen möchten. Auf der anderen Seite drückt der helle Teil des Streifens in der Mitte nach links und dehnt sich damit leicht aus. Es entsteht förmlich ein Keil, der sich sowohl im Differenzbild 7.7(f), als auch im Deformationsfeld 7.7(h) sehr gut erkennen lässt. Unter Verwendung des Korrekturterms kommt es nicht zu diesem Effekt. Aufgrund der Glattheit des Korrekturterms ist ein leichtes Schrumpfen des Streifens am linken Teil zu Beginn der Registrierung erkennbar, doch bereits nach wenigen Iterationen wird der Färbetrend in der Mitte des Kreises perfekt korrigiert, was zur erneuten Glättung des Deformationsfeldes und der Weitung des Keils führt. Auf dem Bild des Korrekturterms lässt sich sehr gut der lineare Färbetrend erkennen. Da sich im Laufe der Iterationen erst die Schanzenform ausbildet, vergehen ein paar Iterationen, bis der Streifen vollständig intensitätskorrigiert ist. Auf dem Differenzbild 7.7(g) und im Deformationsfeld 7.7(i) lässt sich gut erkennen, dass das Ergebnis den Erwartungen entspricht und damit weitestgehend symmetrisch ist. Bei genauer Betrachtung des Deformationsfeldes kann minimal in der Mitte ein Trend zum Keil erkannt werden.

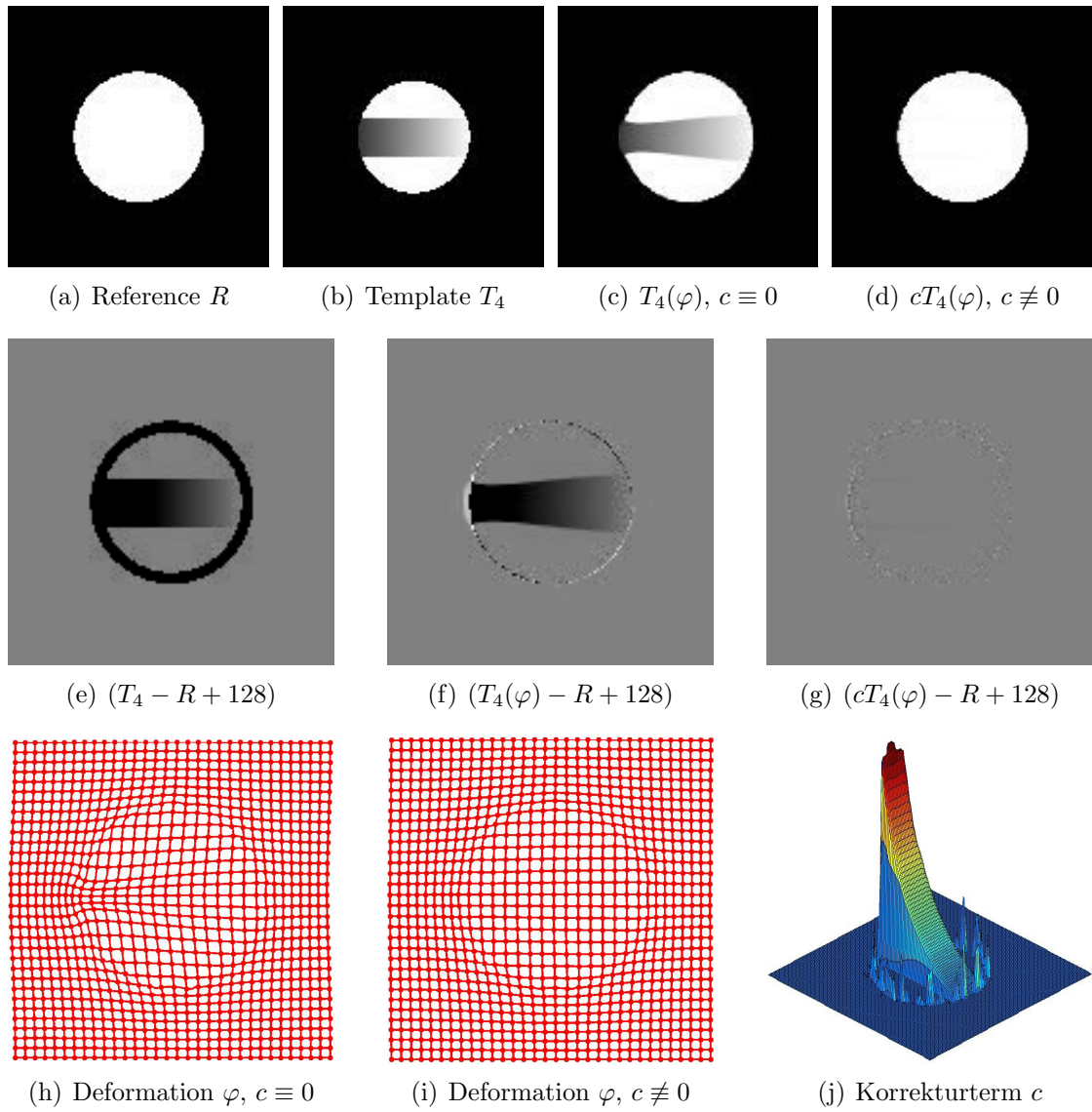


Abbildung 7.7: Registrierung ohne Intensitätskorrektur sorgt für Schrumpfen im dunkel repräsentierten Teil des Kreises. Bei der intensitätskorrigierten Version wird ein symmetrisches Deformationsfeld erzeugt, die scharfen Kanten bleiben scharf erhalten, das transformierte Bild passt perfekt zur Referenz. Beim Korrekturterm lässt sich sehr gut der lineare Farbverlauf erkennen. Die Kanten des Farbbandes sind deutlich zu erkennen. Korrekturpeaks am Rand des Kreises sind Interpolationsartefakte.

7.2 Reale Beispiele 2D

In diesem Abschnitt wollen wir nun den Blick auf realistische Beispieldaten werfen. Im vorherigen Abschnitt haben wir sehr gut sehen können, worin der Vorteil der simultanen Homogenisierung und Registrierung liegt. Bei realen Daten kommen allerdings stets Effekte hinzu, die nicht vorab berechnet werden können, so dass die Ergebnisse selten von der (optischen) Qualität wie bei den akademischen Beispielen ist. Wir stellen hier nun zwei Beispiele vor, bei denen es kein optisch perfektes Ergebnis geben wird, doch dies ist aus Sicht der Aufgabenstellung durchaus gewünscht.

Beginnen wollen wir mit einer Resonanztomographieaufnahme eines menschlichen Gehirns. Das Templatebild ist eine direkt benachbarte Schicht des Referenzbildes, auf das eine künstliche Inhomogenisierung angewendet wurde. Dadurch dass benachbarte Schichten ausgewählt wurden existieren Korrespondenzen zu allen Strukturen innerhalb eines Bildes, so dass Intensitätsunterschiede ausschließlich auf inhomogene Darstellung und nicht durch fehlende oder zusätzliche Strukturen herrührt. Durch die Einführung künstlicher Inhomogenitäten können mit einem Datensatz unterschiedliche Empfangsspulen bzw. MR-Scanner simuliert werden.

Betrachten wir nun Abbildung 7.8. Im Differenzbild 7.8(c) können wir den Unterschied zwischen Referenz- und Templatebild sehr gut erkennen. Hier kommt die Inhomogenität als horizontales Band durch den Datensatz sehr gut zur Geltung. Die wesentlichen Unterschiede aus anatomischer Sicht sind im Schädelknochen, im schädelnnahen Teil des Frontal- und Okzipitallappens und im Ventrikel. Nach der simultanen Homogenisierung und Registrierung sehen wir im Differenzbild 7.8(g) kaum noch Unterschiede. Besonders die Inhomogenitäten wurden ideal entfernt, so dass das transformierte und intensitätskorrigierte Templatebild optisch dem Referenzbild sehr gut entspricht (siehe Abb. 7.8(d)). Die verbliebenen Unterschiede lassen sich auf anatomische Unterschiede zurück führen, die durchaus erwünscht sind. Ein genauer Blick auf das Deformationsfeld in 7.8(f) zeigt, dass die Transformation sehr gut nachvollziehbar ist, da keine starken Verzerrungen erkennbar sind. Entsprechend ist der Korrekturterm c in Abb. 7.8(g) sehr ähnlich zu einer Sattelfläche, was der Simulation tatsächlich entspricht.

Als letztes Beispiel bei der zweidimensionalen Betrachtung wollen wir zwei benachbarte Schichten einer histologischen Serie (Abb. 7.9) betrachten. Die Schichten unterscheiden sich anatomisch nur sehr gering, wurden aber manuell deutlich unterschiedlich eingefärbt. Während die Referenzschicht nahezu ideal homogen gefärbt wurde, ist beim Template ein deutlicher Trend zu erkennen. Der horizontal wirkende Trend sorgt in den untergefärbten Bereichen links für einen starken Verlust des Kontrastes. Die intensive Färbung am rechten Bildteil lässt die harte Abgrenzung zwischen heller und dunkler Gehirnschicht stark verwischen, so dass es zu falschen Korrespondenzen während der Registrierung kommen kann. Wie besonders im Deformationsfeld 7.9(f) zu erkennen ist, findet in diesem Hirnareal eine intensive Transformation statt. Auf der Darstellung des Korrekturterms 7.9(g) lässt sich der lineare Färbeverlauf sehr gut erkennen.

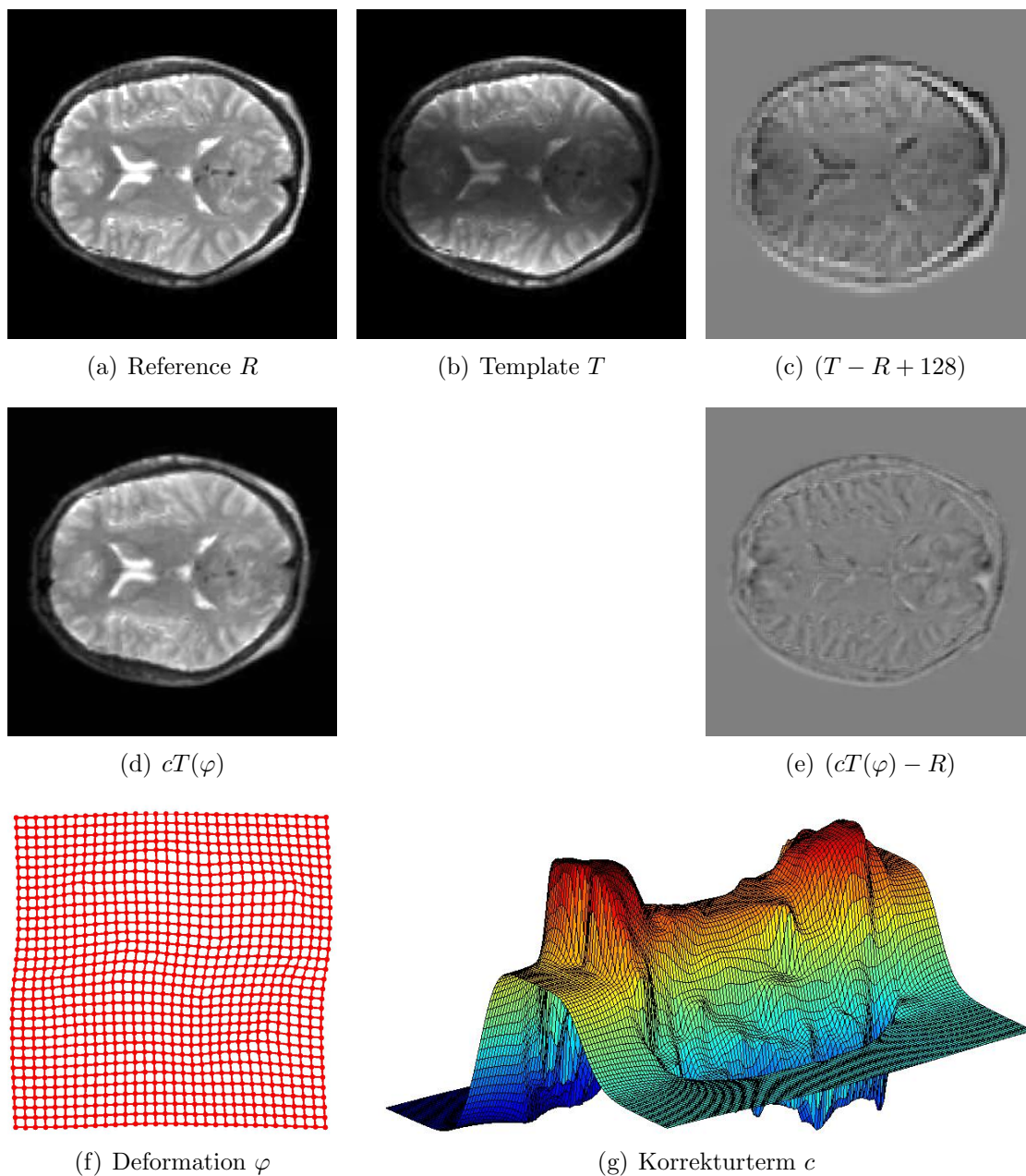


Abbildung 7.8: Je nach verwendeter Empfangsspule stellt sich eine Schicht unterschiedlich dar. Eine simulierte Abschwächung im Templatebild wird im Differenzbild gut erkennbar. Trotz Korrekturterm lässt sich keine optisch perfekte, aber eine aus Problemsicht gute Transformation finden. Bei der Darstellung des Korrekturterms lässt sich sehr gut die Inhomogenitätsverteilung als Sattelfläche erkennen.

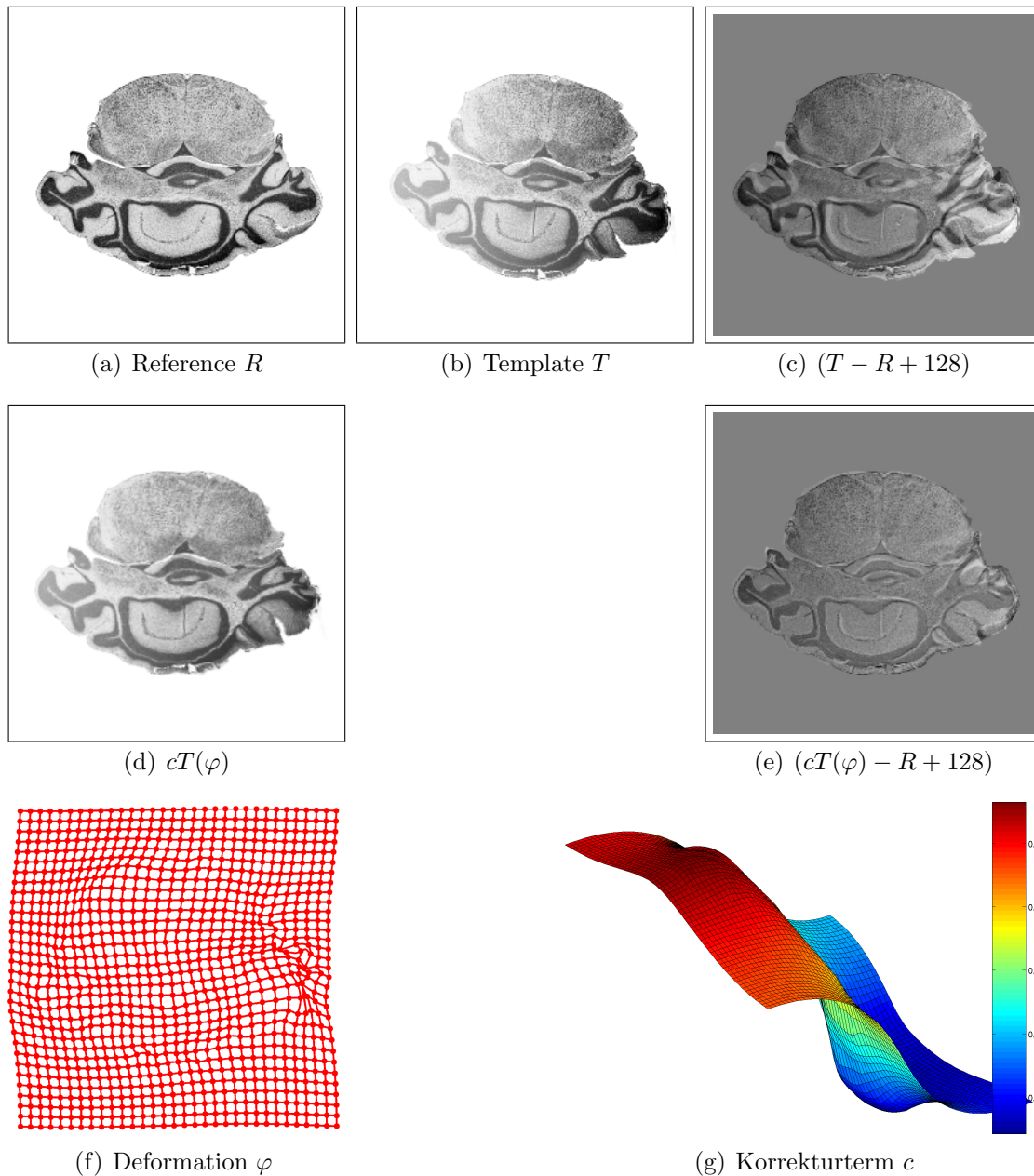


Abbildung 7.9: Typische Färbeartefakte bei histologischen Bildern. Während das Referenzbild homogen gefärbt ist, erweist sich im Templatebild ein deutlicher Färbetrend in horizontaler Richtung. Der Färbetrend spiegelt sich deutlich im Korrekturterm wider. Nach der Korrektur der Intensitäten ist der Färbetrend zwar noch erkennbar, aber nicht mehr so deutlich ausgeprägt. Im Differenzbild sind im Gegensatz zu vor der Transformation und Korrektur kaum noch Unterschiede erkennbar.

7.3 Serienschmittregistrierung

Kommen wir nun zum letzten Abschnitt in diesem Kapitel, der Registrierung von histologischen Serienschritten.

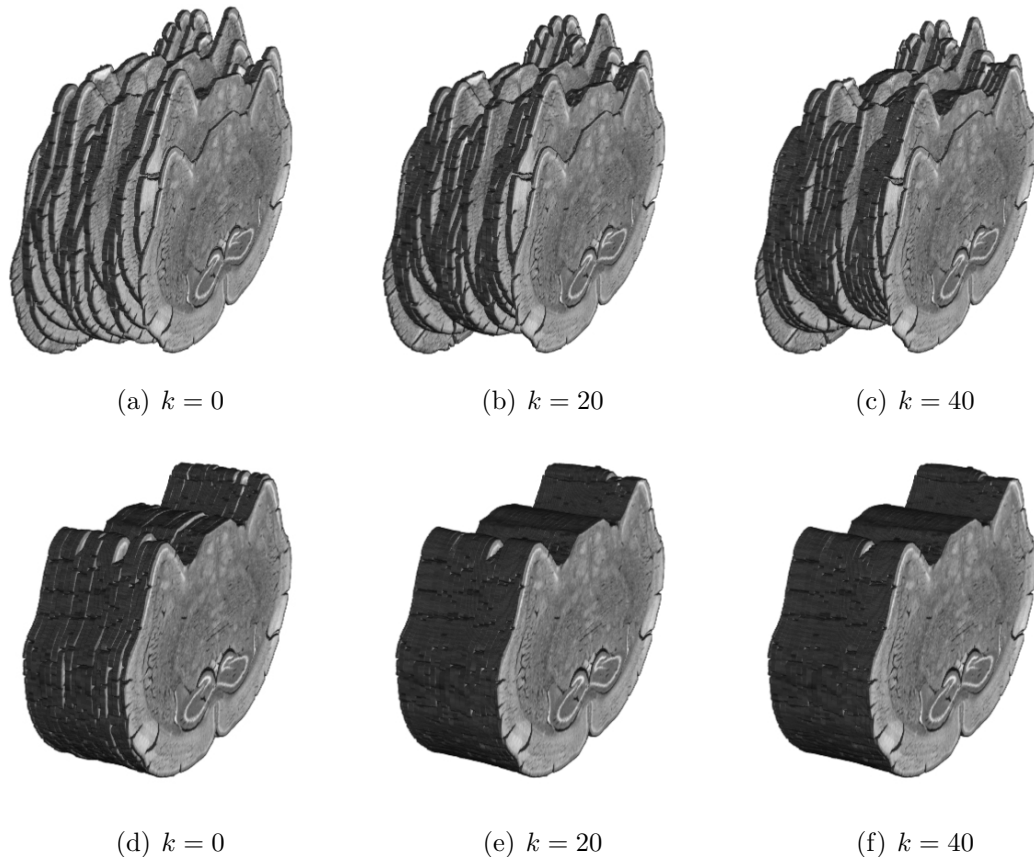


Abbildung 7.10: Darstellung des Unterschieds zwischen einer Registrierung von Serienschritten ohne lineare Vorregistrierung (oben) und mit linearer Vorregistrierung (unten). Ohne Vorregistrierung werden große Deformationen über viele Iterationsschritte mit kleinen Schritten approximiert.

In Abb. 7.10 ist ein Teilstück des Gehirns, bestehend aus 15 Schichten zu den Iterationsschritten $k = 0, 20, 40$ dargestellt. Ohne Vorregistrierung bleiben einzelne Schichten relativ lange unverändert, bis sich die benachbarten Schichten sortiert haben. Die Deformation in jeder einzelnen Schicht ist relativ groß, wodurch starke innere Spannungen entstehen, die eine Fein Anpassung von speziellen Teilbereichen unmöglich macht. Mit Vorregistrierung sind die einzelnen Schichten schon vor der linearen sehr gut angeglichen, wodurch der Algorithmus im wesentlichen nur noch lokale Feinarbeit leisten muss. Das Ergebnis nach 20 bzw. 40 Schritten ist dementsprechend gut.

Eine gute Möglichkeit, die Rekonstruktionsleistung eines Algorithmus zu bewerten ist die Darstellung über orthogonal zur Schnittebene verlaufende Schichten. In

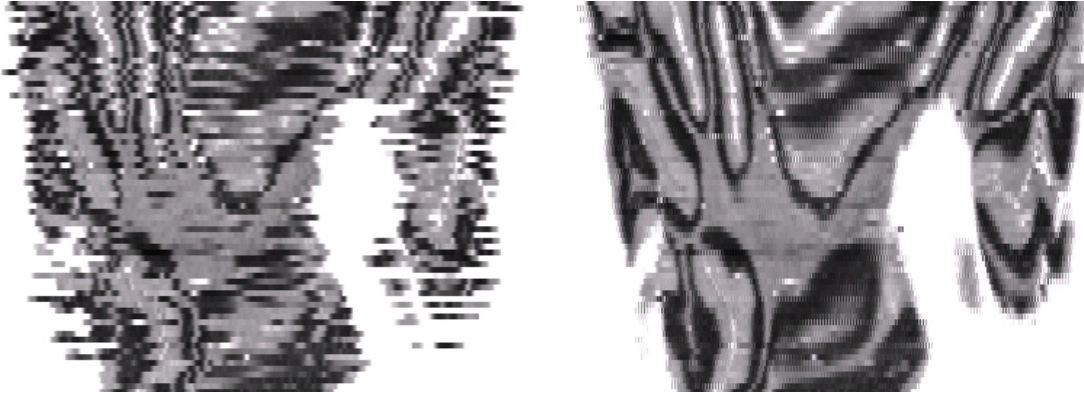


Abbildung 7.11: Orthogonal zur Schnittebene verlaufende Darstellung eines Teils des Rattenhirns. Ungeordnete Teile setzen sich durch die Registrierung zu zusammenhängenden Strukturen zusammen, die über die einzelnen Schichten hinweg in Korrespondenz stehen.

Abb. 7.11 wurde eine solche Darstellung gewählt, um den Unterschied zwischen der Schichtanordnung vor und nach der Registrierung aufzuzeigen. Ein Bild in dieser Darstellung wird dadurch aufgebaut, dass aus jedem ursprünglichen Bild die mittlere Bildzeile bzw. Bildspalte in das neue Bild übertragen wird. Gelingt es durch die Registrierung zusammenhängende Strukturen in den orthogonalen Ansichten zu erzeugen, so gilt die Registrierung als erfolgreich. Als Beispiel einer erfolgreichen Registrierung sei wiederum auf Abb. 7.11 verwiesen. Auf der linken Seite sind die internen Strukturen zwar schon zu erkennen, doch sind bei einigen Bereichen die Zusammenhänge und Korrespondenzen noch nicht eindeutig. Erst die Registrierung schiebt die teilweise willkürlich angeordneten Teile des Bildes zu sinnvollen Strukturen zusammen. Dadurch werden einige Strukturen erst eindeutig identifizierbar. Gerade diese Identifizierung ist es, auf die es ankommt.

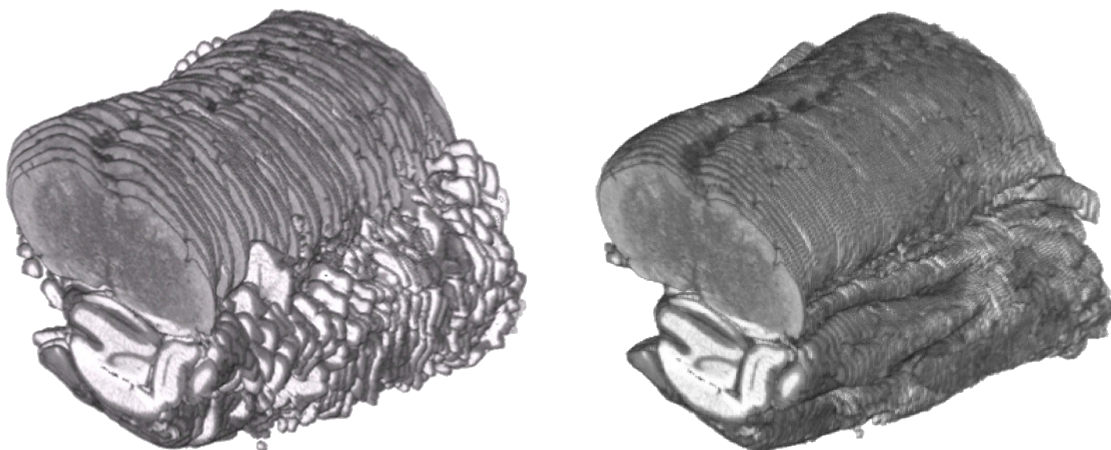


Abbildung 7.12: Dreidimensionale Darstellung des Hirnstamms, links nach der linearen Vorregistrierung, rechts nach der elastischen Registrierung.

Um eine bessere Vorstellung zu bekommen, wie wichtig die elastische Registrierung gegenüber der linearen Registrierung ist, betrachten wir Abb. 7.12. Auf der linken Seite sehen wir eine Rekonstruktion mit ausschließlich linearer Registrierung. Während der Hirnstamm im oberen Teil des Bildes schon sehr gut zusammenhängt, sind im Kleinhirnbereich noch deutliche Verschiebungen zu erkennen. Nach der elastischen Registrierung (rechts) können die einzelnen Bereiche und komplexen Windungen in diesem Teil des Gehirns sehr gut über die einzelnen Schichten verfolgt werden. Darüber hinaus lässt sich die dreidimensionale Rekonstruktion in direkten Bezug zum vollständigen Gehirn nach der Paraffinierung (siehe Abb. 1.2) ziehen.

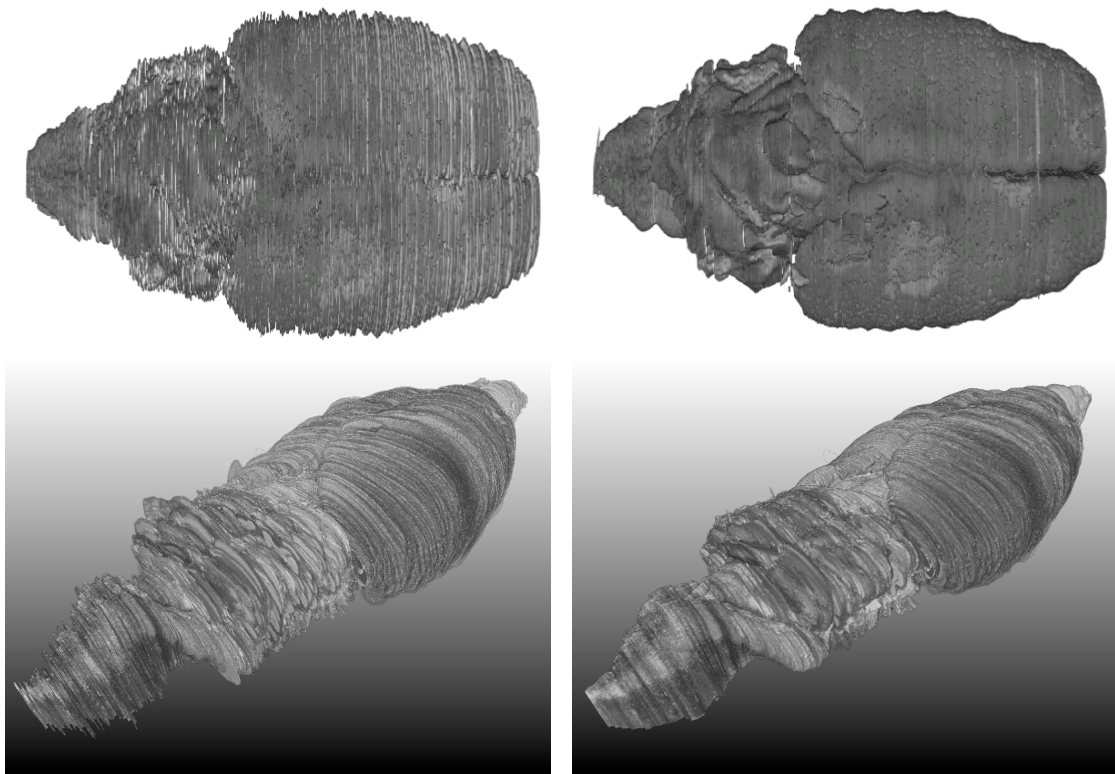


Abbildung 7.13: Dreidimensionale Darstellung eines vollständig rekonstruierten Rattenhirns (oben) und Maushirns (unten), links nach der linearen Vorregistrierung, rechts nach der elastischen Registrierung.

Das gleiche gilt für die Rekonstruktion des vollständigen Rattenhirns. Abb. 7.13 verdeutlicht sehr gut, dass erst durch die elastische Registrierung eine anatomisch korrekte Darstellung möglich ist. Wie im Bild rechts schön zu sehen ist, kann durch die Registrierung sehr gut zwischen Großhirn mit linker und rechter Hemisphäre, Kleinhirn, Hirnstamm und Riechkolben unterschieden werden. Die Übergänge waren vorher weder eindeutig, noch fest umgrenzt.

In Abb. 7.14 ist ein sehr schönes Ergebnis dargestellt. Dieser Teilbereich des Hirns einer Maus (Schichten 250-330) ist besonders interessant, da es den Hippocampus enthält. Der Hippocampus ist der einzige Teil im Gehirn, der auch nach

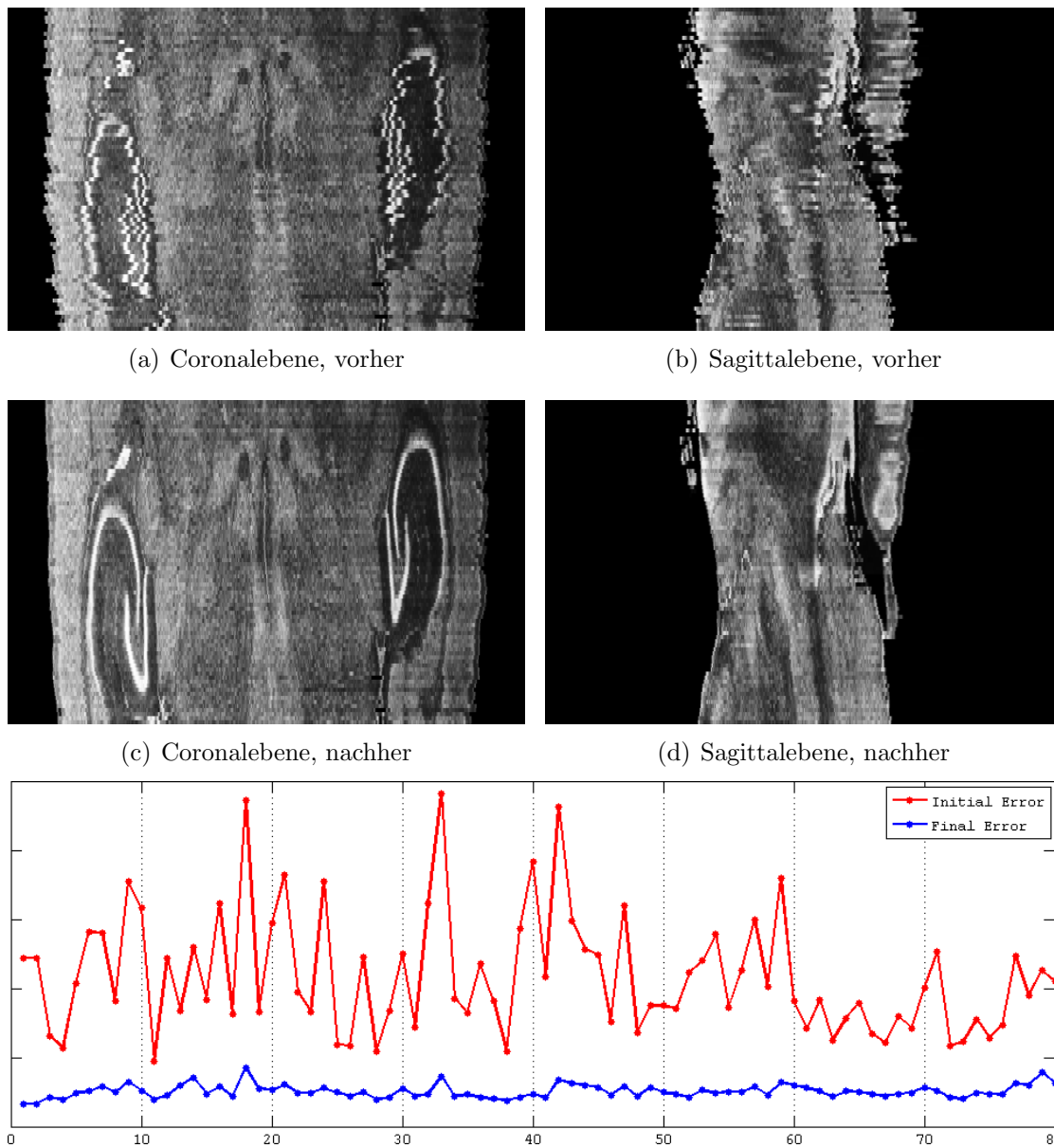


Abbildung 7.14: Orthogonale Ebenenansicht zu den Axialebenenschnitten eines Maushirns, *oben* vor und *mitte* nach der elastischen Registrierung der Schichten 250-330. Darunter, SSD-Werte vor (rot) und nach (blau) der Registrierung. Der vor der Registrierung stark schwankende SSD-Fehler der einzelnen Schichten gleicht sich im Laufe der Registrierung systematisch an. Interne Strukturen wie beispielsweise der Hippocampus wurden anatomisch korrekt rekonstruiert. Die Kurven unten stellen den SSD-Fehler zu Beginn der Registrierung (rot) und nach der Registrierung (blau) dar. Die initial starken Peaks in der Kurve sind nach der Registrierung verschwunden, was für einen sanften Übergang zwischen den Schichten steht.

der Vollentwicklung sich weiter verändern kann. Dort führen Lernprozesse unmittelbar Veränderungen auf neuronaler Ebene. In Abb. 7.14(a) und 7.14(b) ist ein Orthogonalschnitt in der Coronal- bzw. Sagittalebene dargestellt. Die Schichten sind bereits rigide vorregistriert, wodurch schon hier die prinzipiellen Strukturen erkennbar sind. Nichtsdestotrotz ist der Versatz der Schichten teilweise so stark, dass die direkten Zusammenhänge und Korrespondenzen der einzelnen Bereiche nicht erkennbar ist. Die elastische Registrierung dieses Teilbereiches dauerte ca. 25 Minuten bei insgesamt 42 Iterationsschritten und 5 Stufen des Multiresolutionsansatzes mit $\sigma = 0.5$. Das Ergebnis der Registrierung ist in den Abb. 7.14(c) und 7.14(d) zu sehen. In der Rekonstruktion sind nun alle Strukturen eindeutig zu erkennen. Das Ergebnis ist insgesamt sehr glatt und bietet damit eine ideale Voraussetzung für eine dreidimensionale Darstellung. Im unteren Teil der Abbildung ist der Einzelfehler (in SSD gemessen) für alle 80 Schichten aufgeführt. Der Anfangsfehler (rot) zeigte noch starke Sprünge und damit starke Unterschiede zwischen den aufeinander folgenden Schichten. Wie an der blauen Kurve sehr schön zu sehen ist, haben sich die Schichten derart angeglichen, dass die Kurve sehr glatt geworden ist und die starken Sprünge nahezu verschwunden sind. Die verbliebenen Peaks in der Kurve lassen sich auf Abnormalitäten innerhalb der Schnitte zurückführen. Dies können Risse, Überlappungen oder auch Verlust von Gewebe sein.

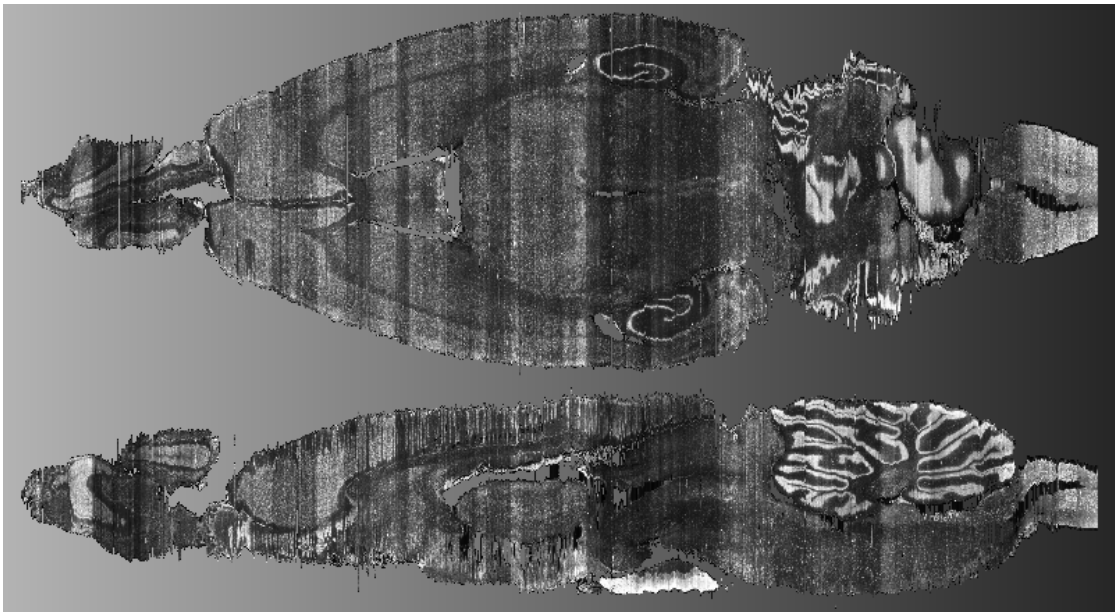


Abbildung 7.15: Orthogonaler Schnitt durch alle Schichten eines ohne simultane Homogenisierung registrierten Mäusehirns. *oben* Coronalebene, *unten* Sagittalebene.

In Abb. 7.15 sehen wir die orthogonale Schnitte durch ein registriertes Mäusehirn. Bei dieser Registrierung wurde auf die Homogenisierung verzichtet. Gerade die inneren Strukturen sind sehr schön registriert. Einbußen gibt es in den Randbereichen, wodurch die Oberfläche des Gehirns nicht so schön glatt geworden ist,

wie im vorigen Beispiel. Bei voller Auflösung von 1408×2048 Pixel pro Schicht und insgesamt 587 Schichten dauerte die Registrierung ca. 7,5 Tage.

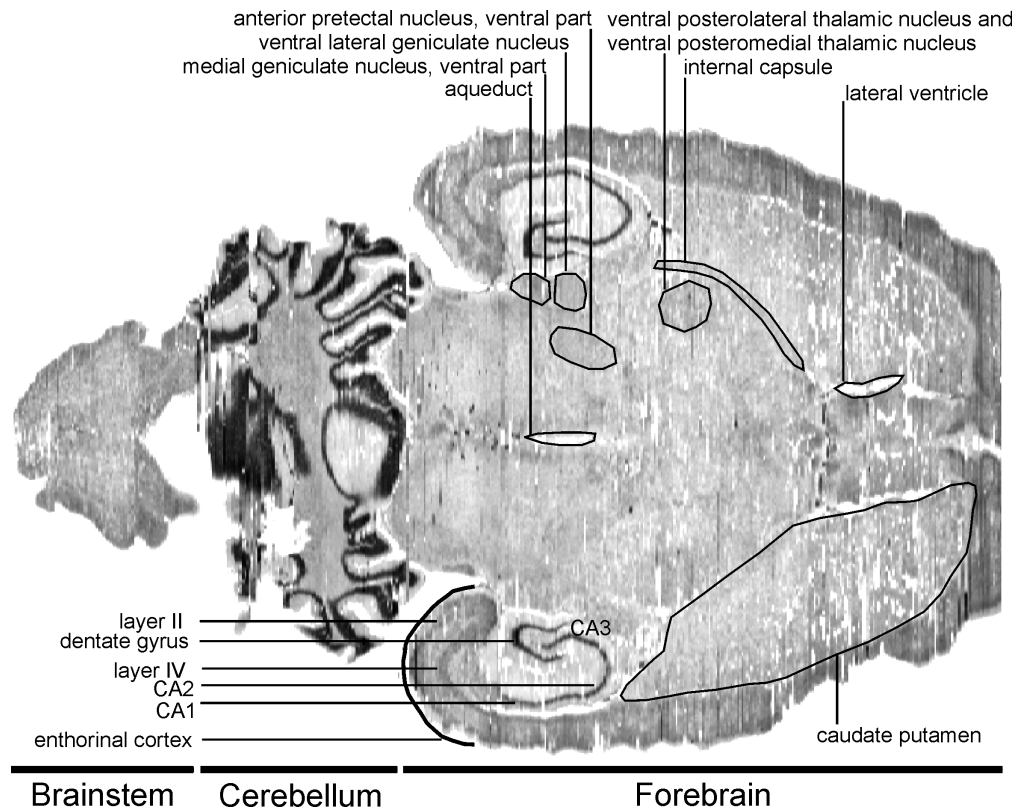


Abbildung 7.16: Orthogonaler Schnitt durch alle Schichten eines registrierten Rattenhirns. Durch die Rekonstruktion können verschiedene Bereiche eindeutig identifiziert und beschriftet werden (*Regionenzuordnung und Beschriftung durch Prof. Dr. Oliver Schmitt, siehe [WFMS04a]*).

Nahezu zum Abschluss der Arbeit möchten wir noch ein Ergebnis aus 2004 zeigen. In Abb. 7.16 handelt es sich um eine orthogonale Ansicht in der Coronalebene eines Rattenhirns. Zum damaligen Zeitpunkt war die Registrierung solcher Daten speziell aus Informatik-sicht eine immense Herausforderung, da die Rechner von damals relativ wenig Arbeitsspeicher und langsame Festplatten besaßen. Darüber hinaus war die histologische Verarbeitungskette noch nicht so ausgereift, wie sie derzeit ist, so dass sehr viele Artefakte in den Bildern vorhanden waren. Trotz dieser Schwierigkeiten konnte der vorgestellte Algorithmus eine Rekonstruktion des Rattenhirns nach einer Berechnung von nur ca. 3 Wochen erwirken, die eine Identifizierung funktionaler Areale ermöglichte.

Nachdem die Registrierung erfolgreich verlaufen und eine dreidimensionale Rekonstruktion geglückt ist, können verschiedene funktionell zusammenhängende Areale im Gehirn gekennzeichnet werden. In Abb. 7.17 sind die erkannten Areale farblich unterschiedlich gekennzeichnet und dreidimensional visualisiert. Das Bild stammt aus einer Arbeit von Prof. Oliver Schmitt aus Rostock, bei der die hier

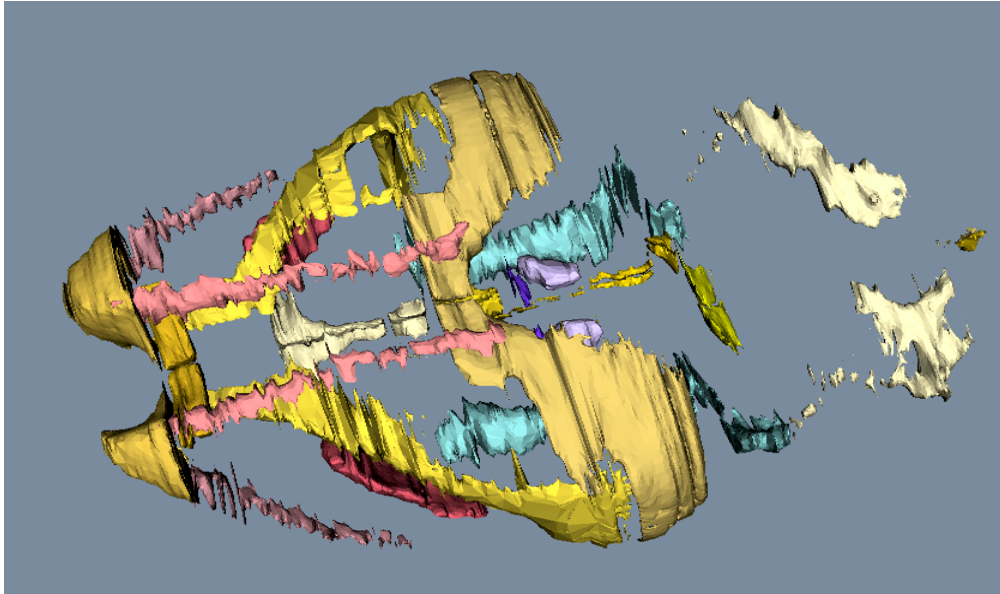


Abbildung 7.17: 3D-Visualisierung der manuell eingezeichneten Konturen der Kommissurenfasern (Corpus callosum, Commissura anterior u.a.) in registrierten Bildern (*Grafik übernommen mit freundlicher Genehmigung von Prof. Dr. Oliver Schmitt*).

vorgestellte Registrierung zum Einsatz kam. Nach Aussage von Prof. Schmitt ist die Registrierung der existenzielle Bestandteil der Arbeit. Ohne die Registrierung wäre eine genaue Lokalisation der verschiedenen Strukturen mit Analyse der Zellgrößen, Zellverteilung und Zelldichte nicht möglich. Die einzelnen Teile der Verarbeitungskette sind allerdings so stark miteinander verknüpft, dass auf allen Ebenen möglichst perfekt gearbeitet werden muss. Dazu gehört neben der Registrierung vor allem die Generierung und Aufbereitung der Schichten und Bilder davon. Je besser dieser Teil läuft, desto bessere Ergebnisse können bei der Registrierung erzielt werden. Das Team um Prof. Schmitt ist mittlerweile derart routiniert, dass im Registrierungsteil der Arbeit gänzlich auf eine affine Vorregistrierung verzichtet werden kann, was eine enorme Zeitersparnis bedeutet.

Über die hier in der Arbeit vorgestellten Ansätze hinaus bestehen weiterhin Bemühungen die Registrierung und damit die dreidimensionale Rekonstruktion, systematisch zu verbessern. Dazu zählen sowohl die Beschleunigung und Verbesserung der Algorithmen, als auch die Optimierung des manuellen Verarbeitungsablaufs.

8 Projekte

In diesem Kapitel möchte ich drei der von mir bearbeiteten Projekten vorstellen. Zum einen handelt es sich um den Einsatz des Prinzips der Serienschichtregistrierung in der physikalischen Chemie. Dort werden nicht durch ein Mikrotom, sondern durch chemische Prozesse eine Serie von Schichtbildern aufgenommen, die anschließend wieder zum dreidimensionalen Objekt rekonstruiert werden müssen. Des weiteren habe ich mit Nils Papenberg zusammen die Erweiterung des SSD-Distanzmaßes auf höhere Ordnung mit zusätzlicher Regularisierung untersucht. Für die Registrierung der Serienschicht hatte diese Entwicklung leider keine Verbesserung gebracht, so dass sie hier gesondert kurz erwähnt sei, da diese Methode in anderen Einsatzgebieten sicherlich erfolgreich sein wird. Schließlich möchte ich die gemeinschaftliche Arbeit FLIRT aus der Arbeitsgruppe SAFIR vorstellen. Dabei handelt es sich um eine Sammlung von Registrierungsalgorithmen, die in C/C++ implementiert sind. Neben der curvature- und diffusiven-Registrierung gibt es eine elastische Registrierung, die ich beigesteuert habe. Das Projekt hatte zum Ziel einen generellen und möglichst flexiblen Softwareentwurf hervor zu bringen, in dem verschiedenen Registrierungsarten und Distanzmaße flexibel integriert und ausgetauscht werden können. Die zweidimensionalen Algorithmen wurden später dann in die Software MeVisLab integriert.

8.1 Scanning Probe Microscopy – Institut für Physikalische Chemie, Universität Bayreuth/Chemnitz

Um wie in der Nanotomographie die Oberflächen von mikro- und nanostrukturierten Materialien abzubilden wird die Rasterkraftmikroskopie eingesetzt. Im Gegensatz zu durchleuchtenden Aufnahmetechniken wie CT und MR wird hier die Oberfläche des Materials mit einer Spitze zeilenweise abgetastet (siehe Abb. 8.1) und anschließend eine Schicht chemisch weggeätzt, um dann die nächste Schicht abzutasten. Eine Schicht wird durch sukzessives Abtasten paralleler Zeilen aufgebaut, bei der eine Zeile vorwärts und rückwärts abgetastet wird und dann die Spitze orthogonal zur Abtastrichtung verschoben.

Durch das ungleichmäßige wegätzen der Oberfläche und der damit verbundenen Abtastung einer Berg-/Tal-Struktur handelt es sich bei den rekonstruierten 2D-Bildern nicht um planare Bilder, sondern ähneln eher einem Kartoffelchip (siehe dazu Abb. 8.2).

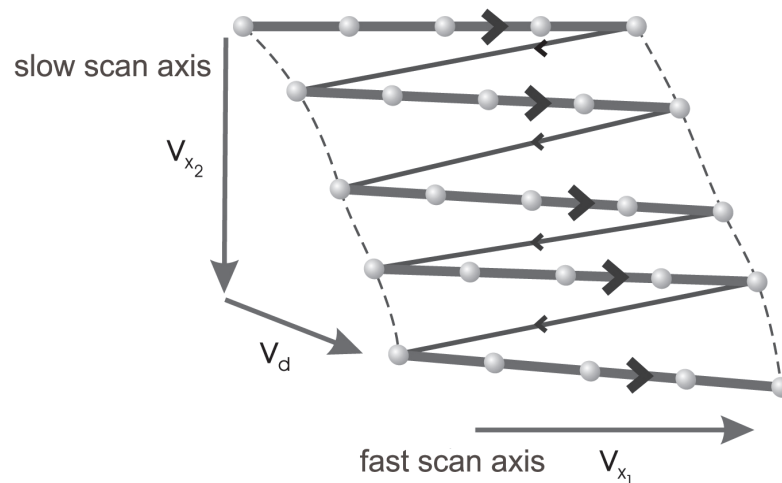


Abbildung 8.1: Schematische Darstellung der relativen Bewegung der Abtastspitze eines Rasterkraftmikroskops, wie es in der Nanotomographie eingesetzt wird. Eingezeichnet sind die Abtastpunkte und der Weg der Abtastspitze während der Aufnahme. (Bild aus [SWRM06] mit freundlicher Genehmigung der Autoren)

Bisherige Rekonstruktionsmethoden verwenden zur Korrektur entweder eine Referenzaufnahme eines bekannten Gitters oder eine lineare Registrierung. In vielen Fällen ist die Aufnahme einer Referenz aber nicht möglich. Die Korrekturen durch die Referenz sind allerdings oftmals ungenau, da durch den Probenwechsel im Gerät, der anderen Ausrichtung und Verarbeitung, die Verzerrungen andersartig sind im Verhältnis zu den Referenzaufnahmen. Zur linearen Registrierung wird bei den einzelnen Schichten von planaren Ebenen im Raum ausgegangen. In den so rekonstruierten Daten lassen sich zusammenhängende Strukturen im Nanometerbereich nicht mehr erkennen (siehe Abb. 8.3).

Eine wesentlich Verbesserung ließ sich bereits durch die Registrierung mit der curvature-Methode erzielen. Auch hier wurde wieder von der Darstellung einer Schicht als planare Ebene ausgegangen. Zusätzlich wurde die Registrierung in

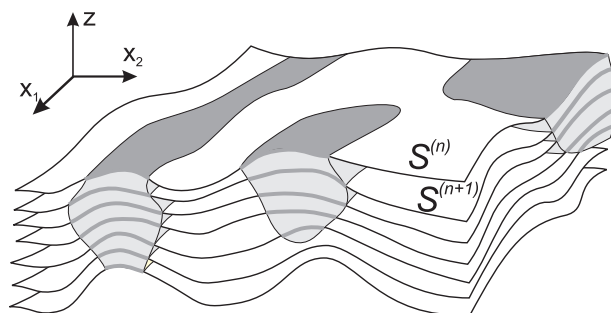


Abbildung 8.2: Prinzipielle Darstellung der Schichten nach einem nanotomographischen Scan mit einem Rasterkraftmikroskops. (Quelle: Magerle, Phys. Rev. Lett., vol. 85, 2749, in 2000, © 2000 by the American Physical Society. [Mag00])

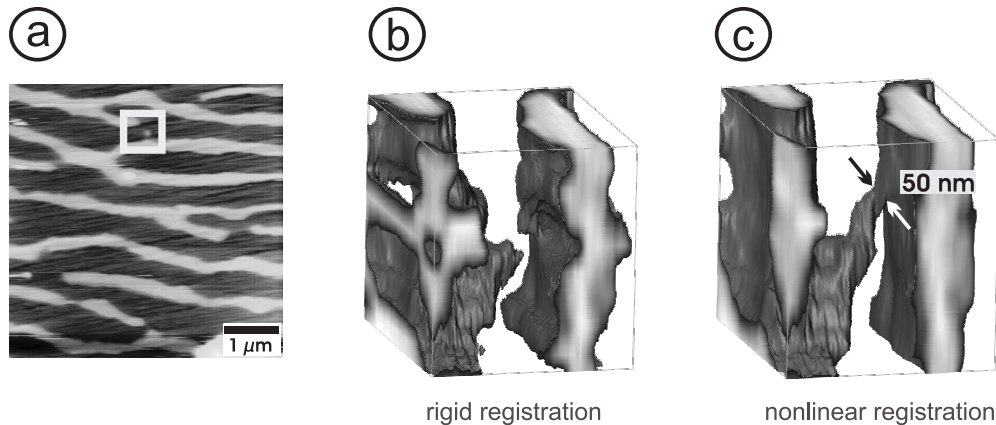


Abbildung 8.3: Durch die curvature-Serienschnittregistrierung konnten zusammenhängende Strukturen wieder rekonstruiert werden (c), was bei linearer Registrierung (b) nicht gelang. Dargestellt ist ein Topographiebild einer NiAl/Ni-Basislegierung nach chemomechanischem Polieren (a). Das helle Quadrat in (a) zeigt den Ausschnitt an, der in die 3D-Darstellung von (b) und (c) eingeht. (Bilder aus [SWRM06] mit freundlicher Genehmigung der Autoren)

den Serienschnitt-Rahmen eingefasst, was wiederum zu deutlichen Verbesserungen führte, so dass nun die zusammenhängenden Strukturen im Nanometerbereich in der 3D-Rekonstruktion erkennbar sind (siehe Abb. 8.3).

8.2 SSD-Distanzmaß höherer Ordnung

Neben den Ausreißern stören besonders Inhomogenitäten durch beispielsweise unterschiedliche Beleuchtung oder Färbung die Berechnung des Deformationsfeldes. Betrachtet man allerdings nicht nur die Differenz der Bilder, sondern auch die Differenz der Gradienten erster und zweiter Ordnung der Bilder, so erhält man ein gegenüber Inhomogenitäten deutlich unempfindlicheres Maß. Der Nachteil liegt dabei in der wesentlich höheren Komplexität des Maßes. Neben den Bildern müssen auch noch die Ableitungen erster und zweiter Ordnung vorhanden sein, bzw. berechnet werden. Dies fordert zusätzliche Rechenzeit und erhöht den Speicherbedarf.

Um den Effekt anschaulicher zu gestalten, betrachten wir Abb. 8.4. In der obersten Reihe sind vier Beispielbilder dargestellt, die geometrisch identisch, aber unterschiedlich durch Grauwerte repräsentiert sind. Bild TI1 stellt das ideale Bild, frei von Artefakten, dar. In den übrigen Bildern sind Störungen auf prinzipielle Weise dargestellt, die in realen Bildern sehr häufig auftreten. TI3 stellt einen Diffusionsprozess dar, TI4 die Färbung eines bestimmten Teilbereichs¹ und TI6 eine periodische Fehldarstellung, die beispielsweise durch Übertragungsfehler, schwan-

¹Bei manchen histologischen Färbungen wird nur die Hälfte des Objekts eingefärbt und die ungefärbte Seite als Referenz zu behalten. Des weiteren kann im Färbeprozess durch z.B. Fette oder andere Barrieren das Färbemittel nicht alle Objektteile erreichen.

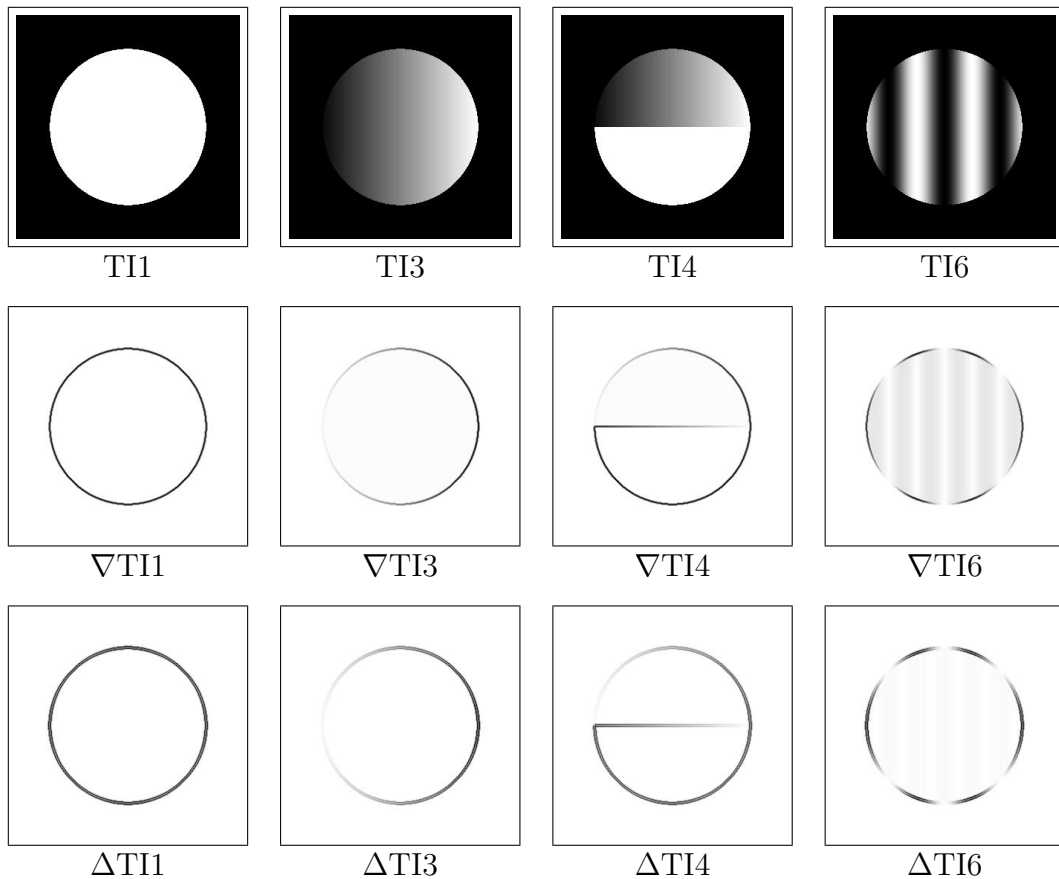


Abbildung 8.4: Geometrisch identische Bilder mit unterschiedlicher Grauwertrepräsentation. In den Ableitungen spiegeln sich diese Unterschiede kaum noch wieder.

kende Schichtdicken bei histologischen Bildern, Krümmungseffekten oder Magnetfeldinhomogenitäten hervorgerufen werden kann. Eine Registrierung dieser Bilder mit dem SSD-Distanzmaß würde unweigerlich zu Geometrieänderungen führen. Betrachtet man allerdings die Ableitungen der Bilder (erste Ableitung in der zweiten, zweite Ableitung in der dritten Zeile von Abb. 8.4), so sind diese kaum unterschiedlich. Dieses Verhalten nutzen wir aus und betrachten SSD nicht nur auf den gegebenen Bildern, sondern auch auf deren Ableitungen. Erweitern wir nunmehr das SSD-Maß durch Ableitungen erster und zweiter Ordnung.

Definition 8.2.1 (Distanzmaß höherer Ordnung)

Seien $R, T \in C^2(\Omega)$ Bilder und $\gamma_0, \gamma_1, \gamma_2 \in \mathbb{R}^{>0}$ Gewichtungsfaktoren, für die $\sum_{i=0}^2 \gamma_i = 1$ gelten soll. Dann bezeichnen wir

$$\begin{aligned} \mathcal{D}^{\text{hoSSD}}[R, T_{\mathbf{u}}] := & \frac{1}{2} \int_{\Omega} \gamma_0 \left(T_{\mathbf{u}}(\mathbf{x}) - R(\mathbf{x}) \right)^2 \\ & + \gamma_1 \left| \nabla T_{\mathbf{u}}(\mathbf{x}) - \nabla R(\mathbf{x}) \right|^2 \\ & + \gamma_2 \left(\Delta T_{\mathbf{u}}(\mathbf{x}) - \Delta R(\mathbf{x}) \right)^2 d\mathbf{x} \end{aligned}$$

als *Distanzmaß höherer Ordnung* (higher order SSD).

Die erste Ableitung eines Bildes beinhaltet nur noch die Kanten des ursprünglichen Bildes. Dadurch werden homogene Bereiche unabhängig von ihrem ursprünglichen Grauwert auf die gleiche Art repräsentiert (Grauwert 0). Die zweite Ableitung beinhaltet nur noch Krümmungsinformationen des ursprünglichen Bildes. Durch die additive Kombination werden mit $\mathcal{D}^{\text{hoSSD}}$ Intensitäts-, Kanten- und Krümmungsinformationen gleichzeitig bemessen.

Die Erweiterung auf Serienschnitte lässt sich wiederum generisch realisieren.

Definition 8.2.2 (Distanzmaß höherer Ordnung für Serien)

Sei $\mathbf{R} = (R_1, \dots, R_M)$ eine Bildserie und $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_M)$ die zu \mathbf{R} gehörende Serie von Deformationsfeldern. Dann bezeichnen wir

$$\mathcal{D}^{\text{hoSSD}}[\mathbf{R}, \mathbf{U}] := \sum_{\nu=2}^M \mathcal{D}^{\text{hoSSD}} [R^{\nu-1}(\mathbf{x} - \mathbf{u}^{\nu-1}(\mathbf{x})), R^{\nu}(\mathbf{x} - \mathbf{u}^{\nu}(\mathbf{x}))]$$

als *Distanzmaß höherer Ordnung für Bildserien*.

Ein ausführlicher Test hat allerdings ergeben, dass für die Registrierung von histologischen Serienschnitten der Vorteil dieses Maßes gegenüber des klassischen SSD aufgrund des enormen Nachteils durch erhöhte Rechenzeit und Speicherbedarf nicht ausreicht. Ergebnisse hierzu können in [WPFS05] nachgelesen werden.

8.3 Flexible Image Registration Toolbox – Institut für Mathematik, Universität zu Lübeck

8.3.1 Der generelle Entwurf

Die Arbeitsgruppe SAFIR (siehe <http://www.math.uni-luebeck.de/safir>) des Instituts für Mathematik der Universität zu Lübeck beschäftigt sich im Schwer-

punkt mit der medizinischen Bildverarbeitung aus Sicht der Numerik. In dieser Arbeitsgruppe wurde ein genereller Lösungsrahmen für das Registrierungsproblem geschaffen und algorithmisch umgesetzt. Neben der Entwicklung grundlegender und effizienter Algorithmen zur Lösung von Registrierungsproblemen ist ein C++-Software-Paket namens FLIRT entstanden. Kern der Algorithmen ist die effiziente Lösung von großen linearen Gleichungssystemen und ein modularer Aufbau.

Das Registrierungsproblem kann als ein Minimierungsproblem formuliert und variationell gelöst werden (siehe dazu auch Kapitel 4). Durch den variationellen Ansatz ist ein Kandidat, der das Funktional minimiert durch die Lösung eines nichtlinearen partiellen Differentialgleichungssystems gegeben, das durch geeignete Linearisierung und Diskretisierung, auf ein lineares Gleichungssystem zurückgeführt werden kann.

In diesem variationellen Rahmen lassen sich verschiedene Grundbausteine entdecken, die dadurch Vorlage für die Klassen in der Implementierung wurden. Die Beschreibung des Registrierungsproblems als Minimierungsproblem hat die Form

$$\mathcal{D}[R, T; \mathbf{u}] + \alpha \cdot \mathcal{S}[\mathbf{u}] \xrightarrow{\mathbf{u}} \min,$$

mit einem Distanzmaß \mathcal{D} , den Bildern R und T , einem Deformationsfeld u und einem Regularisierer \mathcal{S} . In diesem Rahmen lassen sich die Bausteine `Image` (R, T), `Displacement` (\mathbf{u}), `DistanceMeasure` (\mathcal{D}) und `Smoother` (\mathcal{S}) gut identifizieren und als eigenständige Objekte begreifen. Als eine zugrunde liegende Struktur lässt sich das `Array` benennen, wodurch dieses zur Basisklasse geworden ist. Ein `Array` ist eine Liste von Datenwerten. Wie in Abbildung 8.5 gut zu sehen ist, lassen sich aus dieser Basisklasse die Bilder und Deformationsfelder ableiten. Neben der Dimension (2D, 3D) wird bei Bildern zusätzlich noch zwischen Bild (`Image`) und transformierbarem Bild (`TransformableImage`) unterschieden. Ein Bild gilt als unveränderlich in der Geometrie, während ein transformierbares Bild durch ein Deformationsfeld in der Geometrie verändert werden kann. Ein transformierbares Bild muss daher eine Transformationsmethode besitzen, die eine Deformation auf das Bild anwendet.

Ein weiterer Baustein im Registrierungsproblem ist das Distanzmaß. Ein Distanzmaß soll zwei gegebene Bilder einen Abstand zuordnen. Damit beschränkt sich die Funktionalität einer Klasse `DistanceMeasure` auf das Auswerten (Berechnen) der Distanz (`eval`) und dem tatsächlichen Liefern des Werts (`getValue`). Die genaue Wahl des Distanzmaßes spielt in erster Linie keine Rolle. Implementiert ist bisher die Summe der quadrierten Differenzen, `SumOfSquaredDifferences` (siehe Abb. 8.6).

Der Regularisierer ist der letzte Baustein in unserer Formulierung des Registrierungsproblems. Wegen seiner glättenden Eigenschaft bzgl. des Deformationsfeldes wird er mit dem Begriff `Smoother` bedacht. Durch den Regularisierer können z.B. physikalische Eigenschaften modelliert werden, die die Art der zugelassenen Deformation beschreiben. Insgesamt sind drei Regularisierer implementiert, zu denen direkte und effiziente Lösungsschemata entwickelt wurden (`CurvatureSmoother`, `DiffusiveSmoother` und `ElasticSmoother`, siehe Abb. 8.6). Während der `Cur-`

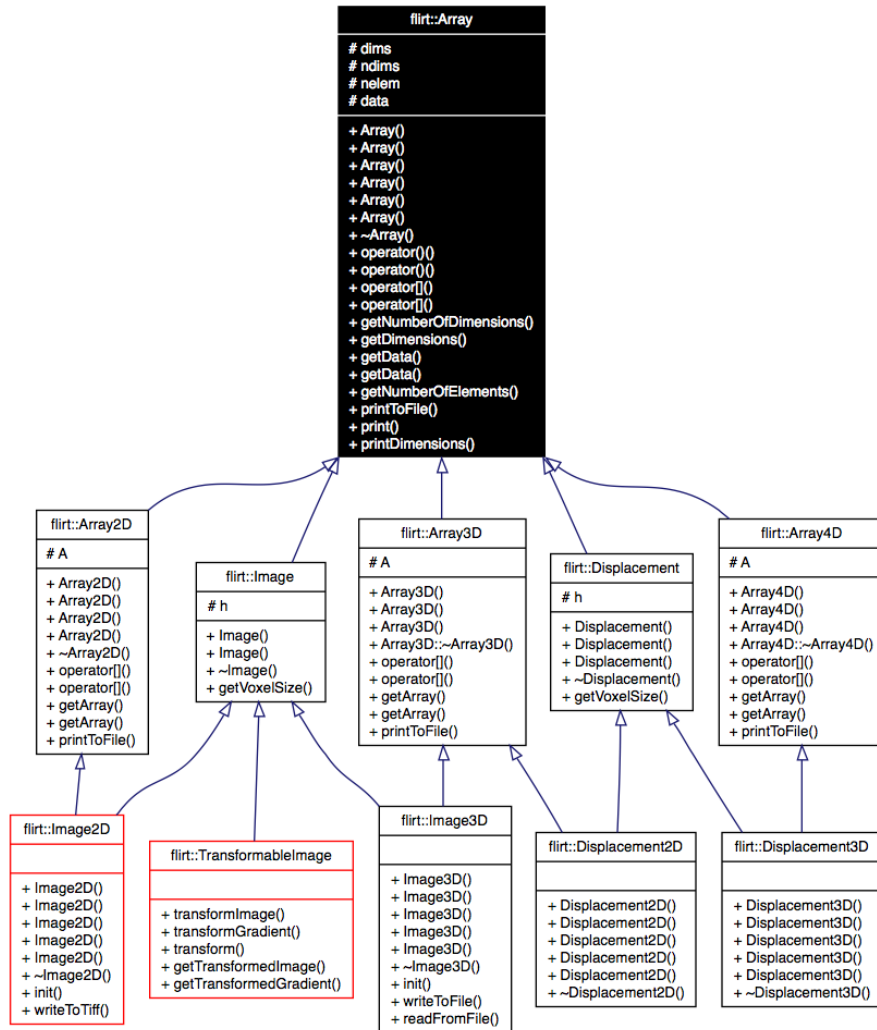


Abbildung 8.5: UML-Diagramm der Klasse Array.

atureSmoother mathematisch motiviert ist und Krümmungen begünstigt, lassen sich die beiden übrigen Regularisierer auch physikalisch deuten. Der Diffusive-Smoother begünstigt Deformationen, die sich diffus verhalten und der Elastic-Smoother begünstigt Deformationen, die als elastisch zu bezeichnen sind.

Zur Lösung des Registrierungsproblems muss, nach dem variationellen Ansatz, ein nichtlineares partielles Differentialgleichungssystem gelöst werden. Daher ist eine Solver-Klasse notwendig, in der verschiedene Methoden umgesetzt werden, die zur Lösung des Systems benötigt werden. Im wesentlichen ist die Form der Linearisierung entscheidend. Diesbezüglich sind hier zwei Methoden umgesetzt. Zum einen eine Fixpunktiteration, zum anderen ein *time marching* Verfahren. Neben der Art der Linearisierung ist die Methode zur Bestimmung der Deformationsänderung im nunmehr linearen Verfahren unterschiedlich. Daher wird der Solver durch einen Optimizer ergänzt, der abhängig von der Art der Linearisierung das Lösen des Gleichungssystems zur Bestimmung der Deformationsänderung steuert und die

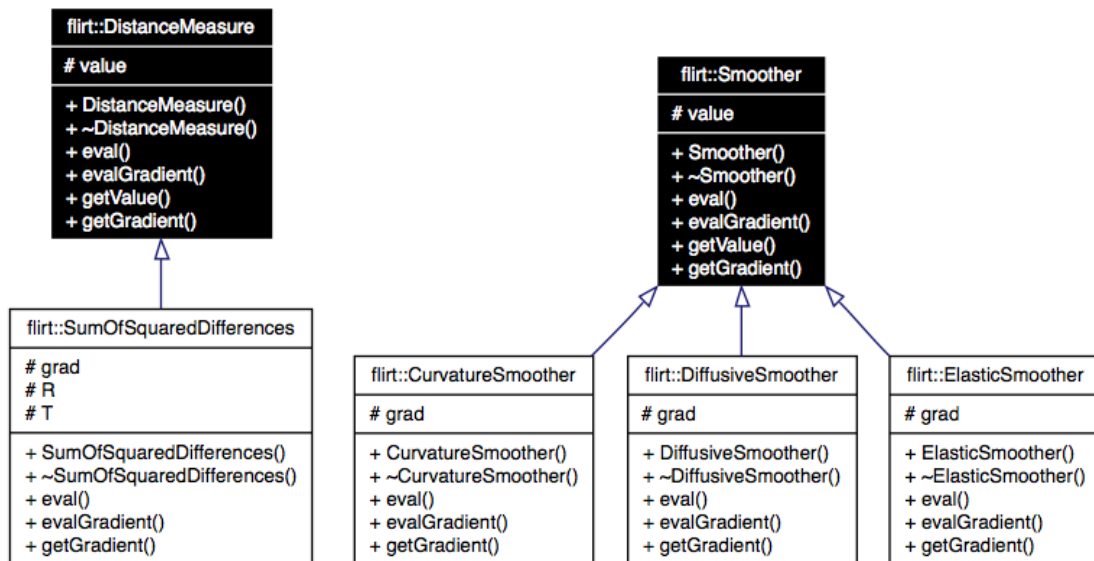


Abbildung 8.6: Distanzmass und Regularisierer.

Deformationsänderung, als Lösung des linearen Gleichungssystems, zur Verfügung stellt.

Ist das Gleichungssystem erst einmal gelöst, kann durch verschiedene Strategien wie *LineSearch* oder *TrustRegion*, die von der Oberklasse *UpdateRule* abgeleitet werden können, das Deformationsfeld aktualisiert werden.

Abgeschlossen wird der Rahmen durch die Klasse *StopRule*, in der die Abbruchkriterien implementiert sind. Mögliche Abbruchkriterien sind beispielsweise,

- Maximale Anzahl an Iterationen erreicht,
- Distanz liegt im Toleranzbereich,
- Änderung des Deformationsfeldes zu gering,
- Änderung des Fehlers, bzw. der Distanz zu gering.

Unter Verwendung der beschriebenen Objekte bzw. Klassen sieht das Hauptprogramm zur zweidimensionalen elastischen Registrierung wie im nachfolgenden Auszug aus der C++ Implementierung aus. Die Hauptschleife ist hierbei stets die gleiche, unabhängig davon, welche Art von Regularisierer, Distanzmass oder Löser verwendet wird.

8.3.2 FLIRT-Integration in MeVisLab

Das in Abschnitt 8.3.1 vorgestellte C++-Paket FLIRT wurde als so genannte stand-alone-Anwendung konzipiert und implementiert. Durch den softwaretechnisch durchdachten hierarchischen Aufbau konnte das Paket sehr schnell in die

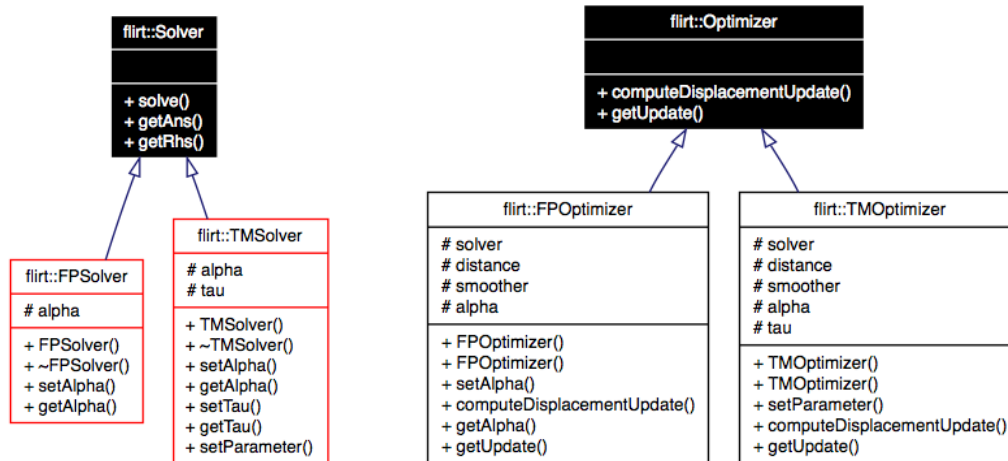


Abbildung 8.7: Gleichungssystemlöser und Optimierer.

sehr komplexe Medizinische Bildverarbeitungs- und Visualisierungssoftware MeVisLab (<http://www.mevislab.de>) integriert werden.

MeVisLab ist eine modular aufgebaute Software, die die Verarbeitung von Bildern in Form von Pipelines zusammen setzt und einzelne Bildoperationen in Form von Modulen bzw. Filtern zur Verfügung stellt. Vom Laden eines Bildes bis zur grafischen Ausgabe geht das Bild durch eine Pipeline. Durch verschiedene Module (z.B. Mittelwertfilterung, Segmentierung, Interpolationen) kann das Bild auf dem Weg modifiziert werden. In Abbildung 8.9 ist die Arbeits- und Entwicklungsumgebung MeVisLab dargestellt. Zu sehen ist ein Netzwerk, eine Konsole für Debug-Informationen und Textausgaben, ein Kommentarfenster, eine Bildvorschau und ein Parameterfenster für das Modul Flirt2D. Am Anfang der Image-Pipeline (unten im Netzwerk) steht das Modul `ImgLoad`, das für das Laden eines Datensatzes zuständig ist. Durch das Modul `Subimage` wird der Datensatz, in dem verschiedene Aufnahmezeitpunkte enthalten sind, auf ein Referenzbild (`fixedImage`) und ein Templatebild (`movingImage`), evtl. sogar auf eine einzelne Schicht eingeschränkt. Mit Hilfe des Moduls `Resample3D` kann die Größe des Datensatzes angepasst werden, so dass auch sehr große Datensätze durch ein subsampling verkleinert und damit zügig verarbeitet werden können. Die eigentliche Registrierung findet dann im Modul `Flirt2d` statt. Um nun das Ergebnis der Registrierung beurteilen zu können, wird auf die Bilder eine arithmetische (`Arithmetic`) Operation ausgeführt, die Bildung der Differenz. Es wird hier die Differenz zwischen Referenzbild und einmal dem ursprünglichen und dann registrierten Templatebild berechnet. Um zwischen diesen beiden Differenzbildern wechseln zu können wird ein `Switch`-Modul hinzugenommen. Im Bildbetrachter für zweidimensionale Bilder `View2D` kann schliesslich das Ergebnis der Pipeline betrachtet werden.

Neben der Nutzung von Flirt als Modul innerhalb von MeVisLab, kann die Möglichkeit von MeVisLab ausgenutzt werden, eigenständige, von MeVisLab losgelöste Anwendungen einfach zu generieren. In Abb. 8.10 ist ein Bildschirmfoto

```

// --- Initialisierung der Objekte ---
// Referenzbild R und Templatebild T
Image2D R(n1, n2, h1, h2);
TransformableImage2D T(n1, n2, h1, h2);

// Deformationsfeld u
Displacement2D u( R.getDimensions(), h1, h2);

// Distanzmaß (hier SSD)
SumOfSquaredDifferences distancemeasure(R,T);

// Regularisierer (hier elastisch)
ElasticSmoother smoother;

// Löser und zugehöriger Optimierer (hier Fixpunkt)
FPSolverElasticFFT2D fpsolver( u, alpha, mu, lambda);
FPOptimizer opti( distancemeasure, smoother, fpsolver, alpha);

// Aktualisierungsvorschrift und Abbruchkriterien
LineSearch ur( distancemeasure, smoother, opti, T ,u);
StopRule sr( distancemeasure, smoother, maxIter);

// Iterationszähler
int iter = 0;

// stelle aktuellen Zustand her durch Auswertung aller Objekte
T.transform(u);
distancemeasure.eval();
distancemeasure.evalGradient();
smoother.eval(u);
smoother.evalGradient(u);

// --- Hauptschleife ---
do {
    iter++;
    opti.computeDisplacementUpdate(u);
    ur.updateDisplacement(u);
    T.transform(u);
    distancemeasure.eval();
    distancemeasure.evalGradient();
    smoother.eval(u);
    smoother.evalGradient(u);
} while (!sr.stopIteration(u, iter));

```

Abbildung 8.8: Exemplarisches Hauptprogramm zur 2D Registrierung mit dem Paket FLIRT. Die Hauptschleife ist hierbei unabhängig von der Art der Regularisierung (*curvature, diffusive, elastic*) und der Linearisierungsstrategie (*fixed point, time marching*).

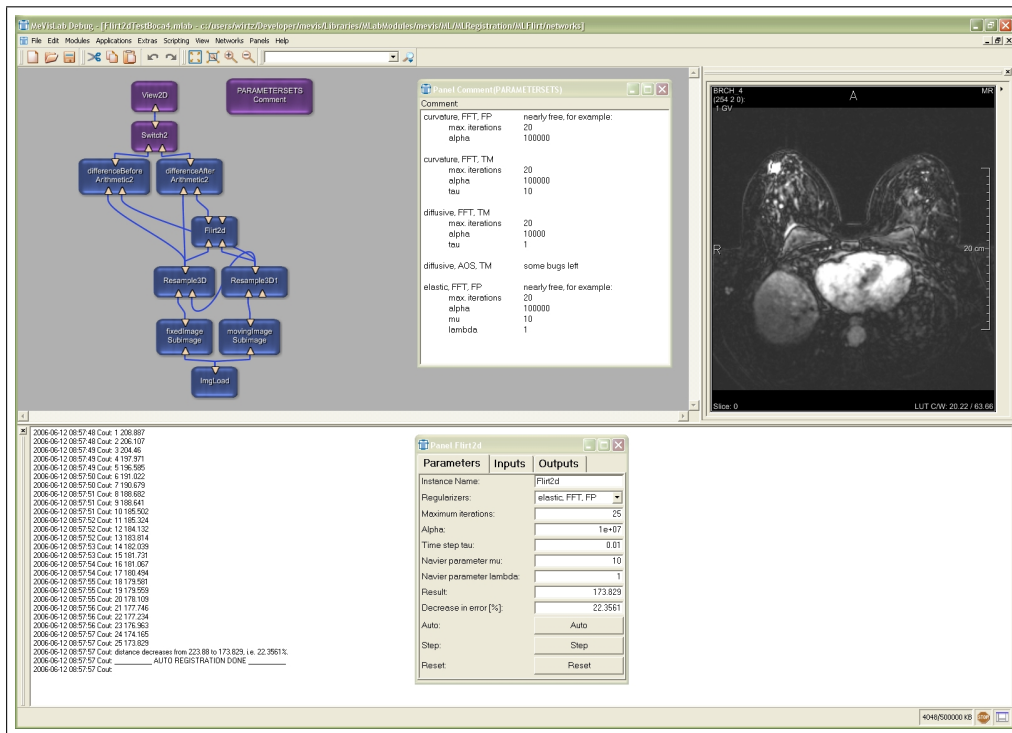


Abbildung 8.9: Netzwerkansicht und Registrierungsverlauf in MeVisLab.

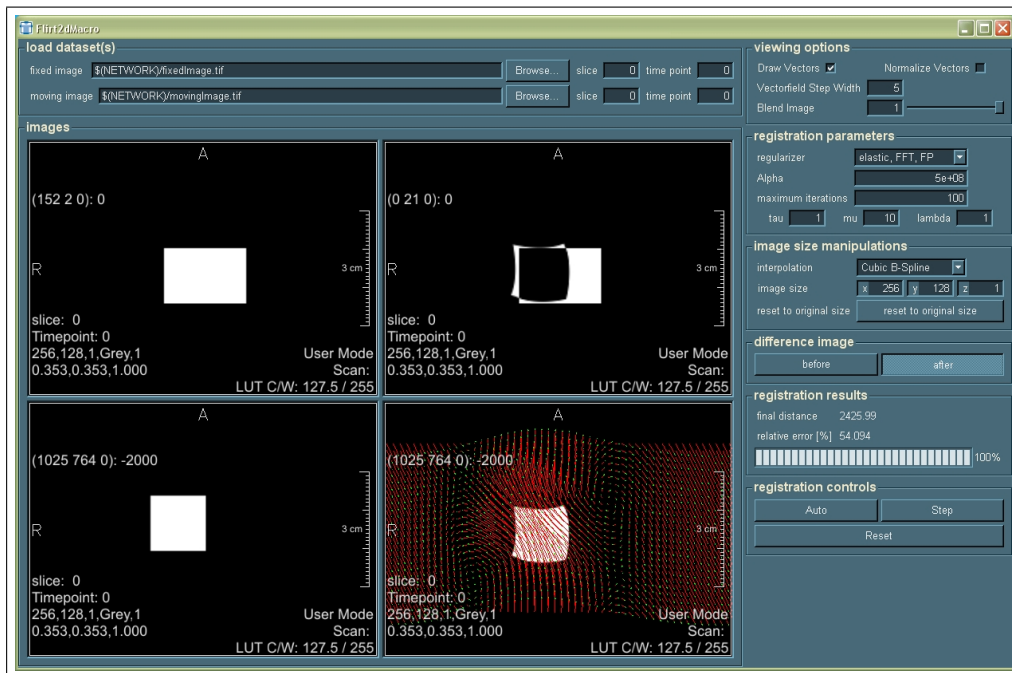


Abbildung 8.10: Flirt2D-Applikation zur schneller Evaluierung von Registrierungs-
ergebnissen.

einer solchen eigenständigen Anwendung. In den vier Bildansichten sind Referenz, Template, Differenzbild (wechselweise vor und nach der Registrierung) und das Deformationsfeld als Vektorfeld in der Überlagerung des transformierten Templates. Alternativ dazu kann ein deformiertes Gitter angezeigt werden. Über einfache graphische Benutzerelemente (GUI-Elemente) können die verschiedenen Optionen von Flirt ausgewählt werden. Die Registrierung kann auf zwei unterschiedliche Weisen benutzt werden, einmal als automatische Iteration, d.h. die Registrierung läuft so lange, bis die internen Abbruchkriterien wirken. Die zweite Variante ist die Ausführung eines einzelnen Registrierungsschritts. Die Zwischenergebnisse werden bei der Einzelschrittvariante sofort aktualisiert und angezeigt, während bei der automatischen Variante ausschließlich das Endergebnis angezeigt wird.

A Mathematische Begriffe und Hilfsmittel

In diesem Anhang legen wir für den interessierten Leser die mathematisch-technischen Grundlagen, die wir in dieser Arbeit verwendet haben. Hier führen wir in die Variationsrechnung ein und klären Begriffe wie Testfunktion, lokale Minimierer, Gâteaux-Ableitung, stationärer Punkt, Euler-Lagrange-Gleichung, Greensche Formel und formulieren den Gaußschen Integralsatz und das Variationslemma.

A.1 Gaußscher Integralsatz und Greensche Formel

Bevor wir einen kurzen Exkurs in die Variationsrechnung machen, wollen wir nun noch die (erste) Greensche-Formel einführen. Zum Beweis dieser Formel und bei der Berechnung von stationären Punkten wird der Gaußsche Integralsatz (Divergenz-satz) benötigt und daher kurz vorgestellt. Der Beweis des Gaußschen Integralsatzes würde den Rahmen hier sprengen, so dass wir darauf hier verzichten. Ausführliche Beweise finden sich z.B. in Forster, Band 3 [For84]. Die Greensche Formel wird später bei der Berechnung der Gâteaux-Ableitung verwendet.

Beginnen wollen wir mit dem Gaußschen Integralsatz. Dieser Satz heisst auch Divergenz-satz, da durch ihn ein Volumenintegral über die Divergenz eines Vektorfeldes in ein Oberflächenintegral umgewandelt werden kann.

Für die nachfolgenden Sätze und Definitionen gelte stets: Sei $\Omega \subset \mathbb{R}^d$ eine offene, beschränkte und einfach zusammenhängende Menge mit einem, bis auf endlich viele Punkte, glatten Rand Γ .

Um den Begriff Divergenz genau festzulegen, erst einmal folgende Definition.

Definition A.1.1 (Divergenz)

Sei Ω wie oben. Des weiteren sei $\mathbf{F} = (F_1, \dots, F_d) : \Omega \rightarrow \mathbb{R}^d$ ein stetig differenzierbares Vektorfeld und $\mathbf{x} = (x_1, \dots, x_d) \in \Omega$. Dann wird die Abbildung

$$\operatorname{div} \mathbf{F} = \sum_{i=1}^d \frac{\partial F_i}{\partial x_i},$$

mit $(\operatorname{div} \mathbf{F}) : \Omega \rightarrow \mathbb{R}$ als *Divergenz* von \mathbf{F} bezeichnet.

Die Divergenz ist die Summe über alle Komponenten von der schrittweisenbezogenen Vergrößerung der jeweiligen Komponente. Ist die Divergenz positiv, so handelt es sich um eine Quelle, ist sie negativ, um eine Senke.

Nachdem der Begriff der Divergenz nun klar definiert ist, können wir den Integralsatz formulieren. Dieser Satz zeigt den Zusammenhang zwischen der Divergenz eines Vektorfeldes und dem durch das Feld vorgegebene Fluss.

Satz A.1.2 (Gaußscher Integralsatz)

Sei Ω wie oben. Sei weiterhin $\mathbf{n} = (n_1, \dots, n_d) : \Gamma \rightarrow \mathbb{R}^d$ das äussere Einheits-Normalenfeld. Dann gilt für jedes stetig differenzierbare Vektorfeld $\mathbf{F} : \Omega \rightarrow \mathbb{R}^d$

$$\int_{\Omega} \operatorname{div} \mathbf{F} \, d\mathbf{x} = \int_{\Gamma} \langle \mathbf{F}, \mathbf{n} \rangle \, ds. \quad (\text{A.1})$$

Speziell für den zweidimensionalen Fall lautet dies

$$\int_{\Omega} \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} \, dx_1 dx_2 = \int_{\Gamma} n_1 F_1 + n_2 F_2 \, ds. \quad (\text{A.2})$$

Hier ohne Beweis.

Aus dem Gaußschen Integralsatz folgt als spezielle Anwendung dessen, die Green-sche Formel.

Satz A.1.3 (Greensche Formel)

Sei Ω wie oben und $\Omega \neq \emptyset$. Dann gilt für alle $f, g \in C^2(\Omega)$

$$\int_{\Omega} \langle \nabla f, \nabla g \rangle_2 + f \Delta g \, d\mathbf{x} = \int_{\Gamma} f \langle \nabla g, \mathbf{n} \rangle_2 \, ds, \quad (\text{A.3})$$

mit $\Delta f = \sum_{i=1}^n \frac{\partial^2 f}{\partial x_i^2} = \nabla \cdot \nabla f$ und dem äusseren Normalenvektor \mathbf{n} auf Γ . (Das Argument (\mathbf{x}) wurden hier für eine bessere Übersicht weggelassen).

Beweis:

Sei $F := f \nabla g$. Dann gilt unter Verwendung der Produktregel (der

Übersicht halber wird auch hier das Argument (\mathbf{x}) weggelassen):

$$\begin{aligned}
\operatorname{div} F &= \sum_{i=1}^n \frac{\partial}{\partial x_i} F \\
&= \sum_{i=1}^n \frac{\partial}{\partial x_i} (f \nabla g) \\
&= \sum_{i=1}^n \left(\left(\frac{\partial}{\partial x_i} f \right) \nabla g + f \left(\frac{\partial}{\partial x_i} \nabla g \right) \right) \\
&= \sum_{i=1}^n \left(\frac{\partial}{\partial x_i} f \right) \nabla g + \sum_{i=1}^n f \left(\frac{\partial}{\partial x_i} \nabla g \right) \\
&= \langle \nabla f, \nabla g \rangle_2 + f \Delta g.
\end{aligned}$$

Schliesslich folgt aus dem Gaußschen Integralsatz (A.1)

$$\int_{\Omega} \operatorname{div} (f \nabla g)(\mathbf{x}) \, d\mathbf{x} = \int_{\Gamma} f(\mathbf{x}) \langle \nabla g(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle \, ds(\mathbf{x}),$$

bzw. kurz

$$\int_{\Omega} \operatorname{div} (f \nabla g) \, d\mathbf{x} = \int_{\Gamma} f \langle \nabla g, \mathbf{n} \rangle \, ds.$$

■

Zum Schluss wollen wir noch eine Folgerung aus der Greenschen Formel aufzeigen.

Korollar A.1.4 (Greensche Formel für Divergenzen)

Sei Ω wie oben und $\Omega \neq \emptyset$. Dann gilt für alle $f, g \in C^2(\Omega)$,

$$\begin{aligned}
\int_{\Omega} (\operatorname{div} f)(\operatorname{div} g) \, d\mathbf{x} &= \\
&= \int_{\Gamma} (\operatorname{div} f) \langle g, \mathbf{n} \rangle_2 \, ds - \int_{\Omega} \langle \nabla \operatorname{div} f, g \rangle_2 \, d\mathbf{x}.
\end{aligned} \tag{A.4}$$

Beweis:

Sei $(\operatorname{div} F) := f$ und $G := (\nabla g)$. Damit ist $\Delta g = \operatorname{div} \nabla g = \operatorname{div} G$. Setzt man dies nun in die Greensche Formel (A.3)

$$\int_{\Omega} \langle \nabla f, \nabla g \rangle_2 + f \Delta g \, d\mathbf{x} = \int_{\Gamma} f \langle \nabla g, \mathbf{n} \rangle_2 \, ds$$

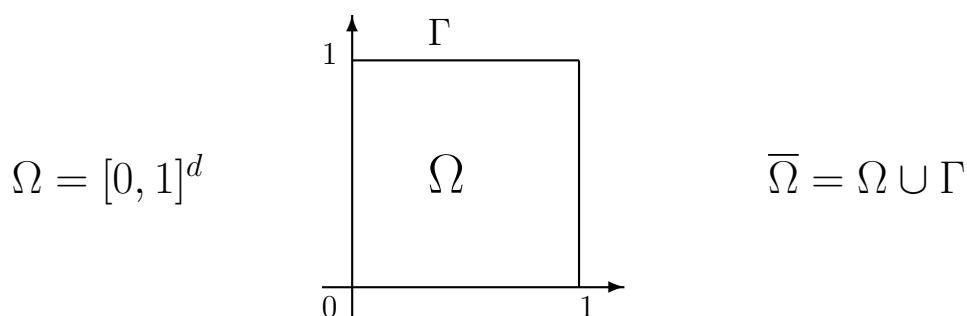
ein, so folgt

$$\int_{\Omega} \langle \operatorname{div} F, G \rangle_2 + (\operatorname{div} F)(\operatorname{div} G) \, d\mathbf{x} = \int_{\Gamma} (\operatorname{div} F) \langle G, \mathbf{n} \rangle_2 \, ds.$$

■

A.2 Einführung in die Variationsrechnung

Sei $\Omega \subset \mathbb{R}^d$ eine offene, beschränkte, einfach zusammenhängende Menge mit einem, bis auf endlich viele Punkte, glatten Rand Γ . Das Paradebeispiel einer solchen Menge ist das Einheitsquadrat bzw. der Einheitswürfel.



Weiter bezeichne $C^k(\bar{\Omega})$ die Menge aller Funktionen mit stetiger partieller Ableitung auf $\bar{\Omega}$ bis zur Ordnung k . V bezeichne eine Menge von Funktionen auf $\bar{\Omega}$, z.B.

$$\begin{aligned} V_1 = \{ & \mathbf{u} \in C^2(\bar{\Omega}); \quad \mathbf{u}(\mathbf{x}) = \alpha, \, x \in \Gamma \} \\ & \mathbf{u} : \bar{\Omega} \rightarrow \mathbb{R}^d, \quad u_i : \bar{\Omega} \rightarrow \mathbb{R} \\ & \mathbf{x} \mapsto (u_1(\mathbf{x}), \dots, u_d(\mathbf{x})). \end{aligned}$$

V heißt Menge der *zulässigen Funktionen*. V ist nicht notwendig ein Vektorraum. Hingegen ist die Menge der *Testfunktionen* bzgl. V , nämlich

$$\tilde{V} = \{ \mathbf{w}; \quad \mathbf{w} = \mathbf{u} - \mathbf{v}, \, \mathbf{u}, \mathbf{v} \in V \}$$

immer ein Vektorraum. Z.B. gilt

$$\tilde{V}_1 = \{ \mathbf{w} \in C^2(\bar{\Omega}); \quad \mathbf{w}(\mathbf{x}) = 0, \, \mathbf{x} \in \Gamma \}.$$

Der Vektorraum \tilde{V}_1 ist derart definiert, dass Dirichlet-Null Randbedingungen gelten.

Sei $\mathbf{u}^* \in V$ ein beliebiges Element, so gilt offenbar

$$V = \{ \mathbf{v}; \quad \mathbf{v} = \mathbf{u}^* + \mathbf{w}, \, \mathbf{w} \in \tilde{V} \},$$

d.h. über \tilde{V} werden „Richtungen“ definiert.

Ist auf V eine Norm $\|\cdot\|$, z.B. $\|\cdot\|_2$, definiert (V Teilmenge eines normierten Raumes), so bezeichnet für $\mathbf{u} \in V$

$$S(\mathbf{u}; \varepsilon) = \{\mathbf{v} \in V; \quad \|\mathbf{u} - \mathbf{v}\| < \varepsilon\}$$

eine Umgebung von \mathbf{u} . Mit obiger Beziehung gilt

$$S(\mathbf{u}; \varepsilon) = \left\{ \mathbf{v} \in V; \quad \mathbf{v} = \mathbf{u} + h\mathbf{w}, \mathbf{w} \in \tilde{V}, \|\mathbf{w}\| = 1, h \in [0, \varepsilon) \right\}.$$

Sei jetzt $f : V \rightarrow \mathbb{R}$ eine auf V definierte Funktion. Dann heißt $\hat{\mathbf{u}}$ *lokaler Minimierer* von f , falls gilt

$$f(\hat{\mathbf{u}}) \leq f(\mathbf{u}), \quad \text{für alle } \mathbf{u} \in V \text{ mit } \|\mathbf{u} - \hat{\mathbf{u}}\| < \varepsilon.$$

Definition A.2.1 (Gâteaux-Ableitung)

Für festes $\mathbf{u}, \mathbf{w} \in V$ definiert

$$\begin{aligned} g(h) : \mathbb{R} &\rightarrow \mathbb{R} \\ h &\mapsto f(\mathbf{u} + h\mathbf{w}) \end{aligned}$$

eine Funktion von \mathbb{R} nach \mathbb{R} . Ist g stetig differenzierbar, so heißt für $\|\mathbf{w}\| = 1$

$$g'(0) = \left. \frac{df(\mathbf{u} + h\mathbf{w})}{dh} \right|_{h=0} = \partial f(\mathbf{u}; \mathbf{w})$$

die *Richtungsableitung*, auch *Gâteaux-Ableitung* von f an der Stelle \mathbf{u} in Richtung \mathbf{w} .

Gilt

$$\partial f(\hat{\mathbf{u}}, \mathbf{w}) = 0, \quad \text{für alle } \mathbf{w} \in \tilde{V}, \|\mathbf{w}\| = 1$$

so heißt $\hat{\mathbf{u}}$ *stationärer Punkt* von f .

Satz A.2.2 (Stationärer Punkt)

Existiert $\partial f(\hat{\mathbf{u}}, \mathbf{w})$ für alle \mathbf{w} , so gilt:

$$\hat{\mathbf{u}} \text{ lokaler Minimierer von } f \quad \implies \quad \hat{\mathbf{u}} \text{ stationärer Punkt von } f.$$

Damit wir später besser argumentieren können, führen wir nun noch das Variationslemma ein.

Satz A.2.3 (Variationslemma)

Sei $\mathbf{u} \in C(\Omega)$ und für jedes $\mathbf{w} \in \mathbb{V}$ mit $\mathbf{w}(\Gamma) = \mathbf{0}$ gelte

$$\int_{\Omega} \mathbf{u}(\mathbf{x})\mathbf{w}(\mathbf{x}) \, d\mathbf{x} = 0.$$

Dann ist $\mathbf{u}(\mathbf{x}) \equiv \mathbf{0}$ auf Ω .

Beweis:

Angenommen, es gelte

$$\int_{\Omega} \mathbf{u}(\mathbf{x})\mathbf{w}(\mathbf{x}) \, d\mathbf{x} = 0$$

und es existiert ein $\hat{\mathbf{x}} \in \Omega$ mit $\mathbf{u}(\hat{\mathbf{x}}) \neq 0$, o.B.d.A. $\varepsilon := \mathbf{u}(\hat{\mathbf{x}}) > 0$.
Aufgrund der Stetigkeit von \mathbf{u} existiert ein $\delta > 0$, so dass in einer Umgebung

$$S(\hat{\mathbf{x}}, \delta) := \{\mathbf{x} \in \Omega : \|\mathbf{x} - \hat{\mathbf{x}}\| < \delta\}$$

die Abbildung \mathbf{u} weiterhin positiv ist:

$$\forall \mathbf{x} \in S(\hat{\mathbf{x}}, \delta) : \mathbf{u}(\mathbf{x}) \geq \frac{\varepsilon}{2} > 0.$$

Nun kann man eine Abbildung $\mathbf{w} \in \mathbb{V}$ konstruieren mit

$$\mathbf{w}(\mathbf{x}) \begin{cases} > 0 & : \quad \forall \mathbf{x} \in S(\hat{\mathbf{x}}, \delta), \\ = 0 & : \quad \text{sonst.} \end{cases}$$

Damit ist eindeutig erfüllt, dass \mathbf{w} auf dem Rand von Ω verschwindet, $\mathbf{w}(\Gamma) = \mathbf{0}$. Allerdings folgt daraus, dass

$$\int_{\Omega} \mathbf{u}(\mathbf{x})\mathbf{w}(\mathbf{x}) \, d\mathbf{x} = \int_{S(\hat{\mathbf{x}}, \delta)} \mathbf{u}(\mathbf{x})\mathbf{w}(\mathbf{x}) \, d\mathbf{x} \geq \frac{\varepsilon}{2} \int_{S(\hat{\mathbf{x}}, \delta)} \mathbf{w}(\mathbf{x}) \, d\mathbf{x} > 0$$

und damit Widerspruch zur ersten Annahme, d.h. es muss gelten $\mathbf{u}(\mathbf{x}) \equiv \mathbf{0}$. ■

Das Variationslemma nutzen wir später um das Integral weg zu diskutieren. Sei nun für die weiteren Betrachtungen $d = 2$ fest gewählt. Ferner sei

$$f(\mathbf{u}) = \int_{\Omega} F(x, y, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_y) \, dx dy, \quad \mathbf{u} \in V = \{\mathbf{v} \in C^2(\Omega)\}, \quad (\text{A.5})$$

wobei $F(x, y, r, s, t)$ stetige partielle Ableitungen der Ordnung ≤ 2 besitzt. Zu diesem f berechnen wir nun die Gâteaux-Ableitung. Hierbei setzen wir voraus, dass F differenzierbar ist und damit Differentiation und Integration vertauscht werden können.

$$\begin{aligned} \frac{df(\mathbf{u} + h\mathbf{w})}{dh} &= \int_{\Omega} \frac{dF(x, y, \mathbf{u} + h\mathbf{w}, \mathbf{u}_x + h\mathbf{w}_x, \mathbf{u}_y + h\mathbf{w}_y)}{dh} dx dy \\ &= \int_{\Omega} \frac{\partial F}{\partial x} \frac{dx}{dh} + \frac{\partial F}{\partial y} \frac{dy}{dh} + \frac{\partial F}{\partial \mathbf{u}} \frac{d(\mathbf{u} + h\mathbf{w})}{dh} \\ &\quad + \frac{\partial F}{\partial \mathbf{u}_x} \frac{d(\mathbf{u}_x + h\mathbf{w}_x)}{dh} + \frac{\partial F}{\partial \mathbf{u}_y} \frac{d(\mathbf{u}_y + h\mathbf{w}_y)}{dh} dx dy \\ &= \int_{\Omega} \frac{\partial F}{\partial \mathbf{u}} \mathbf{w} + \frac{\partial F}{\partial \mathbf{u}_x} \mathbf{w}_x + \frac{\partial F}{\partial \mathbf{u}_y} \mathbf{w}_y dx dy \end{aligned}$$

Bedingung für einen stationären Punkt ist das Verschwinden der Gâteaux-Ableitung:

$$\begin{aligned} 0 &= \int_{\Omega} \left(\frac{\partial F}{\partial \mathbf{u}} + \frac{\partial F}{\partial \mathbf{u}_x} + \frac{\partial F}{\partial \mathbf{u}_y} \right) dx dy \quad \text{für alle } \mathbf{w} \in \tilde{V} \\ \Rightarrow 0 &= \int_{\Omega} \left(\frac{\partial F}{\partial \mathbf{u}} + \frac{\partial F}{\partial \mathbf{u}_x} + \frac{\partial F}{\partial \mathbf{u}_y} \right. \\ &\quad \left. + \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial \mathbf{u}_x} \right) \mathbf{w} + \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial \mathbf{u}_y} \right) \mathbf{w} \right. \\ &\quad \left. - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial \mathbf{u}_x} \right) \mathbf{w} - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial \mathbf{u}_y} \right) \mathbf{w} \right) dx dy \\ \Rightarrow 0 &= \int_{\Omega} \left[\frac{\partial F}{\partial \mathbf{u}} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial \mathbf{u}_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial \mathbf{u}_y} \right) \right] \mathbf{w} dx dy \\ &\quad + \int_{\Omega} \left(\frac{\partial F}{\partial \mathbf{u}_x} \mathbf{w}_x + \frac{\partial F}{\partial \mathbf{u}_y} \mathbf{w}_y + \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial \mathbf{u}_x} \right) \mathbf{w} + \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial \mathbf{u}_y} \right) \mathbf{w} \right) dx dy \\ \Rightarrow 0 &= \int_{\Omega} \left[\frac{\partial F}{\partial \mathbf{u}} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial \mathbf{u}_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial \mathbf{u}_y} \right) \right] \mathbf{w} dx dy \\ &\quad + \int_{\Gamma} \left(n_1 \frac{\partial F}{\partial \mathbf{u}_x} \mathbf{w} + n_2 \frac{\partial F}{\partial \mathbf{u}_y} \mathbf{w} \right) ds. \tag{A.6} \end{aligned}$$

Hierbei ist $\mathbf{n} = (n_1, n_2)$ die äußere Normale. Der letzte Schritt (A.6) kommt durch die Anwendung des Gaußschen Integralsatzes zustande.

Verwenden wir nun, wie bereits oben kurz erwähnt, Dirichlet-Randbedingungen

$$\begin{aligned} V &= \{ \mathbf{v} \in C^2(\bar{\Omega}), \mathbf{v} = \alpha \text{ auf } \Gamma \} \\ \Rightarrow \tilde{V} &= \{ \mathbf{v} \in C^2(\bar{\Omega}), \mathbf{v} = 0 \text{ auf } \Gamma \}. \end{aligned}$$

Damit folgt, dass die Testfunktionen \mathbf{w} auf dem Rand Γ alle Null sind, wodurch das Randintegral in (A.6) verschwindet

$$\int_{\Gamma} \left(n_1 \frac{\partial F}{\partial \mathbf{u}_x} + n_2 \frac{\partial F}{\partial \mathbf{u}_y} \right) \mathbf{w} ds = 0, \quad \text{für alle } \mathbf{w} \in \tilde{V}.$$

Das Variationslemma (Satz A.2.3) liefert als weitere Folge die notwendige Bedingung für ein Minimum:

$$\frac{\partial F}{\partial \mathbf{u}} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial \mathbf{u}_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial \mathbf{u}_y} \right) = 0 \quad \text{in } \Omega \quad (\text{A.7})$$

und $\mathbf{u} = \alpha$ auf Γ .

B Ausführliche Rechnungen & Ergänzungen

B.1 Gâteaux-Ableitung von $\mathcal{D}^{\text{cSSD}}[R, T; \mathbf{u}, c]$

Beginnen wollen wir mit der Gâteaux-Ableitung des Distanzmaßes. Das Distanzmaß hängt sowohl vom Deformationsfeld u , als auch vom Korrekturterm c ab, sodass die Euler-Lagrange-Gleichungen zu beiden Variablen berechnet werden müssen. Da weder u von c noch c von u abhängen, können die Gâteaux-Ableitungen einzeln aufgestellt werden.

Sei nun hier nach Gleichung (A.5)

$$\begin{aligned} f_1(\mathbf{u}) &:= \mathcal{D}^{\text{cSSD}}[R, T, \mathbf{u}, c] \quad (\text{mit beliebigem, aber festem } c) \\ &= \frac{1}{2} \int_{\Omega} \left(c(\mathbf{x})T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}) \right)^2 d\mathbf{x}, \end{aligned}$$

bzw.

$$\begin{aligned} f_2(c) &:= \mathcal{D}^{\text{cSSD}}[R, T, \mathbf{u}, c] \quad (\text{mit beliebigem, aber festem } \mathbf{u}) \\ &= \frac{1}{2} \int_{\Omega} \left(c(\mathbf{x})T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}) \right)^2 d\mathbf{x}. \end{aligned}$$

Nach Gleichung A.7 ist unter Verwendung von Dirichlet-Randbedingungen

$$\left(c(\mathbf{x})T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}) \right) c(\mathbf{x}) \nabla T(\mathbf{x} - \mathbf{u}(\mathbf{x})) = 0 \quad (\text{B.1})$$

die notwendige Bedingung für ein Minimum bzgl. \mathbf{u} und

$$\left(c(\mathbf{x})T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - R(\mathbf{x}) \right) T(\mathbf{x} - \mathbf{u}(\mathbf{x})) = 0 \quad (\text{B.2})$$

die notwendige Bedingung für ein Minimum bzgl. c .

B.2 Gâteaux-Ableitung von $\mathcal{S}^{\text{elast}}[\mathbf{u}]$

Nach dem Distanzmaß betrachten wir nun den Regularisierer des Deformationsfeldes \mathbf{u} . Auch hier bestimmen wir die Gâteaux-Ableitung, um eine notwendige Bedingung für ein Minimum zu erhalten.

Sei nun nach Gleichung (A.5)

$$\begin{aligned} f(\mathbf{u}) &:= \mathcal{S}^{\text{elast}}[\mathbf{u}] \\ &= \int_{\Omega} \frac{\mu}{4} \sum_{j,k=1}^d (\partial_{x_j} u_k + \partial_{x_k} u_j)^2 + \frac{\lambda}{2} (\operatorname{div} \mathbf{u})^2 \, d\mathbf{x}. \end{aligned}$$

Die Berechnung der *Gâteaux*-Ableitung ist deutlich komplizierter und wird daher in aller Ausführlichkeit behandelt. Nach den ersten Umformungen kann die Berechnung der Ableitung in zwei Teile aufgespalten werden, die wir anschliessend getrennt voneinander weiter berechnen werden, um so eine bessere Übersicht zu behalten.

$$0 = g'(0) = \left. \frac{d}{dh} f(\mathbf{u} + h\mathbf{v}) \right|_{h=0} \quad (\text{B.3})$$

$$\begin{aligned} &= \left[\frac{d}{dh} \int_{\Omega} \frac{\mu}{4} \sum_{j,k=1}^d (\partial_{x_j} (u_k + hv_k) + \partial_{x_k} (u_j + hv_j))^2 + \frac{\lambda}{2} (\operatorname{div} (\mathbf{u} + h\mathbf{v}))^2 \, d\mathbf{x} \right]_{h=0} \\ &= \left[\frac{d}{dh} \int_{\Omega} \frac{\mu}{4} \sum_{j,k=1}^d (\partial_{x_j} (u_k + hv_k) + \partial_{x_k} (u_j + hv_j))^2 + \frac{d}{dh} \int_{\Omega} \frac{\lambda}{2} (\operatorname{div} (\mathbf{u} + h\mathbf{v}))^2 \, d\mathbf{x} \right]_{h=0} \\ &= \left[\frac{d}{dh} \int_{\Omega} \frac{\mu}{4} \sum_{j,k=1}^d (\partial_{x_j} (u_k + hv_k) + \partial_{x_k} (u_j + hv_j))^2 \, d\mathbf{x} \right]_{h=0} \end{aligned} \quad (\text{B.4})$$

$$+ \left[\frac{d}{dh} \int_{\Omega} \frac{\lambda}{2} (\operatorname{div} (\mathbf{u} + h\mathbf{v}))^2 \, d\mathbf{x} \right]_{h=0}.$$

Um die Rechnungen überschaubar weiter zu führen, bezeichnen wir den ersten Teil mit

$$\mathcal{S}_1^{\text{elast}} := \int_{\Omega} \frac{\mu}{4} \sum_{j,k=1}^d (\partial_{x_j} (u_k + hv_k) + \partial_{x_k} (u_j + hv_j))^2 \, d\mathbf{x}$$

und den zweiten Teil mit

$$\mathcal{S}_2^{\text{elast}} := \frac{\lambda}{2} \int_{\Omega} (\operatorname{div} (\mathbf{u} + h\mathbf{v}))^2 \, d\mathbf{x}.$$

Die Berechnung der *Gâteaux*-Ableitung von $\mathcal{S}^{\text{elast}}$ kann dann wie folgt umgeformt,

$$0 = g'(0) = \frac{d}{dh} \mathcal{S}_1^{\text{elast}} + \frac{d}{dh} \mathcal{S}_2^{\text{elast}}$$

und dann der Reihe nach durchgeführt werden. Beginnen wollen wir mit der Be-

rechnung der Gâteaux-Ableitung von $\mathcal{S}_1^{\text{elast}}$.

$$0 = \left[\frac{d}{dh} \mathcal{S}_1^{\text{elast}}[\mathbf{u} + h\mathbf{v}] \right]_{h=0} = \left[\frac{d}{dh} \int_{\Omega} \frac{\mu}{4} \sum_{j,k=1}^d (\partial_{x_j}(u_k + hv_k) + \partial_{x_k}(u_j + hv_j))^2 d\mathbf{x} \right]_{h=0} \quad (\text{B.5})$$

$$= \frac{\mu}{4} \int_{\Omega} \left[\frac{d}{dh} \sum_{j,k=1}^d (\partial_{x_j}(u_k + hv_k) + \partial_{x_k}(u_j + hv_j))^2 \right]_{h=0} d\mathbf{x} \quad (\text{B.6})$$

$$= \frac{\mu}{4} \int_{\Omega} \sum_{j,k=1}^d \left[\frac{d}{dh} (\partial_{x_j}(u_k + hv_k) + \partial_{x_k}(u_j + hv_j))^2 \right]_{h=0} d\mathbf{x}$$

$$= \frac{\mu}{2} \int_{\Omega} \sum_{j,k=1}^d (\partial_{x_j}u_k + \partial_{x_k}u_j) \left[\frac{d}{dh} (\partial_{x_j}(u_k + hv_k) + \partial_{x_k}(u_j + hv_j)) \right]_{h=0} d\mathbf{x}$$

$$= \frac{\mu}{2} \int_{\Omega} \sum_{j,k=1}^d (\partial_{x_j}u_k + \partial_{x_k}u_j) (\partial_{x_j}v_k + \partial_{x_k}v_j) d\mathbf{x}$$

$$= \frac{\mu}{2} \int_{\Omega} \sum_{j,k=1}^d (\partial_{x_j}u_k + \partial_{x_k}u_j) \partial_{x_j}v_k + (\partial_{x_j}u_k + \partial_{x_k}u_j) \partial_{x_k}v_j d\mathbf{x}$$

$$= \frac{\mu}{2} \int_{\Omega} \sum_{j,k=1}^d \langle \nabla u_k + \partial_{x_k}u, \nabla v_k \rangle_2 + \langle \partial_{x_j}u + \nabla u_j, \nabla v_j \rangle_2 d\mathbf{x}$$

$$= \mu \int_{\Omega} \sum_{k=1}^d \langle \nabla u_k + \partial_{x_k}u, \nabla v_k \rangle_2 d\mathbf{x} \quad (\text{B.7})$$

$$= \mu \int_{\Gamma} \sum_{k=1}^d v_k \langle \nabla u_k + \partial_{x_k}\mathbf{u}, \mathbf{n} \rangle_2 d\sigma - \mu \int_{\Omega} \sum_{k=1}^d \langle \Delta u_k + \partial_{x_k}\partial_{x_k}\mathbf{u}, v_k \rangle_2 d\mathbf{x}$$

$$= \mu \int_{\Gamma} \sum_{k=1}^d v_k \langle \nabla u_k + \partial_{x_k}\mathbf{u}, \mathbf{n} \rangle_2 d\sigma - \mu \int_{\Omega} \langle \Delta \mathbf{u} + \nabla \text{div} \mathbf{u}, \mathbf{v} \rangle_2 d\mathbf{x}. \quad (\text{B.8})$$

Bleibt nun noch die Berechnung der Gâteaux-Ableitung für den zweiten Teil von

$\mathcal{S}^{\text{elast}}$

$$\begin{aligned}
 0 &= \left[\frac{d}{dh} \mathcal{S}_2^{\text{elast}}[\mathbf{u} + h\mathbf{v}] \right]_{h=0} \\
 &= \left[\frac{d}{dh} \frac{\lambda}{2} \int_{\Omega} (\operatorname{div}(\mathbf{u} + h\mathbf{v}))^2 d\mathbf{x} \right]_{h=0} \tag{B.9}
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{\lambda}{2} \int_{\Omega} \left[\frac{d}{dh} (\operatorname{div}(\mathbf{u} + h\mathbf{v}))^2 \right]_{h=0} d\mathbf{x} \\
 &= \frac{\lambda}{2} \int_{\Omega} \left[\frac{d}{dh} (\operatorname{div} \mathbf{u} + h \operatorname{div} \mathbf{v})^2 \right]_{h=0} d\mathbf{x} \\
 &= \frac{\lambda}{2} \int_{\Omega} \left[\frac{d}{dh} (\operatorname{div} \mathbf{u})^2 + 2h(\operatorname{div} \mathbf{u})(\operatorname{div} \mathbf{v}) + h^2(\operatorname{div} \mathbf{v})^2 \right]_{h=0} d\mathbf{x} \\
 &= \frac{\lambda}{2} \int_{\Omega} \left[0 + 2(\operatorname{div} \mathbf{u})(\operatorname{div} \mathbf{v}) + 2h(\operatorname{div} \mathbf{v})^2 \right]_{h=0} d\mathbf{x} \\
 &= \lambda \int_{\Omega} (\operatorname{div} \mathbf{u})(\operatorname{div} \mathbf{v}) d\mathbf{x} \tag{B.10}
 \end{aligned}$$

$$= \lambda \int_{\Gamma} (\operatorname{div} \mathbf{u}) \langle \mathbf{v}, \mathbf{n} \rangle_2 d\sigma - \lambda \int_{\Omega} \langle \nabla \operatorname{div} \mathbf{u}, \mathbf{v} \rangle_2 d\mathbf{x}. \tag{B.11}$$

Fasst man nun die Gâteaux-Ableitungen (B.8) und (B.11) zusammen, so erhält man die Fortführung der Rechnung zur Gâteaux-Ableitung von $\mathcal{S}^{\text{elast}}$ aus (B.4),

$$\begin{aligned}
 0 &= \left[\frac{d}{dh} \mathcal{S}^{\text{elast}}[\mathbf{u} + h\mathbf{v}] \right]_{h=0} \\
 &= \mu \int_{\Gamma} \sum_{k=1}^d v_k \langle \nabla u_k + \partial_{x_k} \mathbf{u}, \mathbf{n} \rangle_2 d\sigma - \mu \int_{\Omega} \langle \Delta \mathbf{u} + \nabla \operatorname{div} \mathbf{u}, \mathbf{v} \rangle_2 d\mathbf{x} \\
 &\quad + \lambda \int_{\Gamma} (\operatorname{div} \mathbf{u}) \langle \mathbf{v}, \mathbf{n} \rangle_2 d\sigma - \lambda \int_{\Omega} \langle \nabla \operatorname{div} \mathbf{u}, \mathbf{v} \rangle_2 d\mathbf{x}. \tag{B.12}
 \end{aligned}$$

Nach Voraussetzung und Dirichlet-Randbedingungen verschwindet die Variation \mathbf{v} auf dem Rand. Damit ergeben sich die Randintegrale in (B.12) zu Null und es bleibt

$$0 = \mu \int_{\Omega} \langle \Delta \mathbf{u} + \nabla \operatorname{div} \mathbf{u}, \mathbf{v} \rangle_2 d\mathbf{x} + \lambda \int_{\Omega} \langle \nabla \operatorname{div} \mathbf{u}, \mathbf{v} \rangle_2 d\mathbf{x}. \tag{B.13}$$

Des weiteren muss Gleichung (B.13) für jede mögliche Wahl von \mathbf{v} gelten, so dass aus dem Variationslemma folgt, dass bereits die Integranden verschwinden müssen,

$$\begin{aligned}
 0 &= \mu(\Delta \mathbf{u} + \nabla \operatorname{div} \mathbf{u}) + \lambda(\nabla \operatorname{div} \mathbf{u}) \\
 &= \mu \Delta \mathbf{u} + (\mu + \lambda) \nabla \operatorname{div} \mathbf{u}. \tag{B.14}
 \end{aligned}$$

Gleichung (B.14) ist dann die zugehörige Euler-Lagrange-Gleichung des Regularisierers $\mathcal{S}^{\text{elast}}$.

B.3 Gâteaux-Ableitung von $\mathcal{S}^{\text{TV}}[c]$

Schliesslich wollen wir die Gâteaux-Ableitung des Regularisierers des Korrekturterms $c : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$ betrachten. Wiederum stellen wir die zugehörige Euler-Lagrange-Gleichung auf. Sei nun hier nach Gleichung (A.5)

$$f(c) := \mathcal{S}^{\text{TV}}[c] = \int_{\Omega} \Psi_{\varepsilon}(|\nabla(c)|_2^2) \, d\mathbf{x}.$$

Berechnen wir nun eine Nullstelle der Gâteaux-Ableitung als notwendige Bedingung für ein Minimum.

$$0 = \left[\frac{d}{dh} \int_{\Omega} \Psi_{\varepsilon}(|\nabla(c + hv)|_2^2) \, d\mathbf{x} \right]_{h=0} \quad (\text{B.15})$$

Sei nun Ψ_{ε} wie in (4.3) gewählt. Die erste Ableitung nach dem Argument ist gegeben durch $\Psi'_{\varepsilon}(s^2) = \frac{d}{ds^2} \Psi_{\varepsilon}(s^2) = \frac{1}{2\sqrt{s^2 + \varepsilon^2}}$. Nach (A.6) gilt:

$$\begin{aligned} 0 &= \int_{\Omega} \langle 2\Psi'_{\varepsilon}(|\nabla c|_2^2) \nabla c, \nabla v \rangle_2 \, d\mathbf{x} \\ &= \int_{\Omega} \left\langle \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}}, \nabla v \right\rangle_2 \, d\mathbf{x} \end{aligned} \quad (\text{B.16})$$

$$= \int_{\Gamma} v \left\langle \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}}, \mathbf{n} \right\rangle_2 \, d\sigma - \int_{\Omega} v \nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}} \, d\mathbf{x}. \quad (\text{B.17})$$

Zeile (B.17) ist aus Zeile (B.16) durch Anwendung der Greenschen Formel (siehe (A.3)) entstanden. Wichtig ist hier die verkürzte Schreibweise des Skalarprodukts, also der Anwendung des Nabla-Operators auf einen Gradienten $\nabla \cdot \nabla g(\mathbf{x}) = \Delta g(\mathbf{x})$. Mit \mathbf{n} wird der äussere Normalenvektor auf dem Rand von Ω und mit Γ der Rand von Ω bezeichnet. Nach den Voraussetzung ist $\mathbf{u}(\Gamma) = \mathbf{0}$ und $c(\Gamma) = \mathbf{1}$ (Dirichlet-Randbedingungen). Daraus ergibt sich, dass das Randintegral in (B.17) Null ist.

Als notwendige Bedingung für ein Minimum folgt dann die Bedingung:

$$\nabla \cdot \frac{\nabla c}{\sqrt{|\nabla c|_2^2 + \varepsilon^2}} = 0. \quad (\text{B.18})$$

B.4 Iterationsvorschrift zum TV Entrauschen mit Standardabweichung

In Abschnit 3.3 haben wir die TV-Bildglättung als kantenerhaltende Glättung von Bildern kennen gelernt. Die dort vorgestellte Methode nutzt keinerlei Vorwissen über das zu entrauschende bzw. glättende Bild. Sollte allerdings die Standardabweichung des Rauschens eines Bildes bekannt sein, so kann mit Hilfe des Algorithmus B.1 ein ideales Entrauschen unter Erhaltung der im Bild vorhandenen Kanten erreicht werden.

Algorithmus B.1 Iterationsvorschrift TV Entrauschen

Sei B_0 ein Bild mit additiven Rauschen der Varianz $\sigma^2 \in \mathbb{R}^{>0}$. Weiterhin seien $\delta t \in \mathbb{R}^{>0}$ ein Zeitschritt, $k \in \mathbb{N}$ ein Iterationszähler, $t_k := k \delta t$ der resultierende Iterationsschritt, κ ein Schwellwert (z.B. $\kappa = 0.1$), N die Anzahl an Bildpunkten und $0 < \varepsilon \ll 1$.

- 1: $B^{(0)} \leftarrow B_0$; $\lambda^{(0)} \leftarrow 0$; $k \leftarrow 0$;
- 2: **while** $\left(\int_{\Omega} B^{(k)}(\mathbf{x}) - B^{(k-1)}(\mathbf{x}) \, d\mathbf{x} > \kappa \right)$ **do**
- 3: Berechne

$$B^{(k+1)} \leftarrow B^{(k)} + \delta t \left[\frac{\partial}{\partial x_1} \left(\frac{B_{x_1}^{(k)}}{\sqrt{(B_{x_1}^{(k)})^2 + (B_{x_2}^{(k)})^2 + \varepsilon^2}} \right) + \frac{\partial}{\partial x_2} \left(\frac{B_{x_2}^{(k)}}{\sqrt{(B_{x_1}^{(k)})^2 + (B_{x_2}^{(k)})^2 + \varepsilon^2}} \right) - \lambda^{(k)} (B^{(k)} - B_0) \right];$$

- 4: Berechne

$$\lambda^{(k+1)} \leftarrow \frac{1}{N 2\sigma^2} \sum_{x_1, x_2} \left[\sqrt{(B_{x_1}^{(k+1)})^2 + (B_{x_2}^{(k+1)})^2 + \varepsilon^2} - \left(\frac{(B_0)_{x_1} B_{x_1}^{(k+1)}}{\sqrt{(B_{x_1}^{(k+1)})^2 + (B_{x_2}^{(k+1)})^2 + \varepsilon^2}} + \frac{(B_0)_{x_2} B_{x_2}^{(k+1)}}{\sqrt{(B_{x_1}^{(k+1)})^2 + (B_{x_2}^{(k+1)})^2 + \varepsilon^2}} \right) \right];$$

- 5: $k \leftarrow k + 1$;
 - 6: **end while**
-

Dieses iterative Verfahren bietet den Vorteil, dass die erste Nebenbedingung (3.2) implizit erfüllt ist, indem bereits der Mittelwert des verrauschten Bildes sofort in die erste Approximation für das entrauschte Bild übernommen wird. Mit ansteigendem t , bzw. mit fortschreitender Iteration k , konvergiert das Verfahren gegen die entrauschte Version des gegebenen Bildes.

B.5 Herleitung der Euler-Lagrange-Gleichungen zur TV Bildglättung

Nach Problem 3.3.2 sind die zugehörigen Euler-Lagrange-Gleichungen des Minimierungsproblems

$$\int_{\Omega} |\nabla B(\mathbf{x})|_2 \, d\mathbf{x} + \lambda_1 \left(\int_{\Omega} B(\mathbf{x}) \, d\mathbf{x} - \int_{\Omega} B_0(\mathbf{x}) \, d\mathbf{x} \right) + \lambda_2 \left(\int_{\Omega} \frac{1}{2} (B(\mathbf{x}) - B_0(\mathbf{x}))^2 \, d\mathbf{x} - \sigma^2 \right) \xrightarrow{B} \min$$

mit $\lambda_1, \lambda_2 > 0$ gesucht. Da wir hier den Lagrange-Ansatz verwenden, muss nach den einzelnen Komponenten B , λ_1 und λ_2 der Lagrange-Funktion abgeleitet werden. Betrachten wir nun der Übersicht halber die drei Summanden getrennt voneinander.

I. Summand Beginnen wir mit der Ableitung nach B .

$$\begin{aligned} 0 &= \left[\frac{d}{dh} \int_{\Omega} |\nabla(B + hC)(\mathbf{x})|_2 \, d\mathbf{x} \right]_{h=0} \\ &= \int_{\Omega} \left[\frac{d}{dh} |\nabla(B + hC)(\mathbf{x})|_2 \right]_{h=0} \, d\mathbf{x} \\ \Rightarrow 0 &= \left[\frac{d}{dh} |\nabla(B + hC)(\mathbf{x})|_2 \right]_{h=0} \\ &= \left[\frac{d}{dh} \sqrt{\langle \nabla(B + hC)(\mathbf{x}), \nabla(B + hC)(\mathbf{x}) \rangle_2} \right]_{h=0} \\ &= \left[\frac{1}{2} \frac{2 (\nabla C(\mathbf{x}))^\top (\nabla(B + hC)(\mathbf{x}))}{\sqrt{\langle \nabla(B + hC)(\mathbf{x}), \nabla(B + hC)(\mathbf{x}) \rangle_2}} \right]_{h=0} \\ &= \frac{(\nabla C(\mathbf{x}))^\top (\nabla B(\mathbf{x}))}{|\nabla(B + hC)(\mathbf{x})|_2} \end{aligned}$$

Da dies für beliebige Testfunktionen $C(\mathbf{x})$ gelten soll, folgt die Bedingung:

$$0 = \frac{\nabla^\top (\nabla B(\mathbf{x}))}{|\nabla B(\mathbf{x})|_2} \stackrel{2D}{=} \frac{\partial}{\partial x_1} \left(\frac{B_{x_1}}{\sqrt{B_{x_1}^2 + B_{x_2}^2}} \right) + \frac{\partial}{\partial x_2} \left(\frac{B_{x_1}}{\sqrt{B_{x_1}^2 + B_{x_2}^2}} \right).$$

Die Ableitungen nach λ_1 und λ_2 verschwinden beide, da hier im ersten Summanden beide Multiplikatoren nicht enthalten sind.

II. Summand

$$0 = \lambda_1 \left(\int_{\Omega} (B(\mathbf{x}) + hC(\mathbf{x})) \, d\mathbf{x} - \int_{\Omega} B_0(\mathbf{x}) \, d\mathbf{x} \right)$$

III. Summand

$$\begin{aligned} 0 &= \left[\frac{d}{dh} \lambda_2 \left(\int_{\Omega} \frac{1}{2} (B(\mathbf{x}) + hC(\mathbf{x}) - B_0(\mathbf{x}))^2 \, d\mathbf{x} - \sigma^2 \right) \right]_{h=0} \\ &= \lambda_2 \int_{\Omega} \left[\frac{d}{dh} \frac{1}{2} (B(\mathbf{x}) + hC(\mathbf{x}) - B_0(\mathbf{x}))^2 \right]_{h=0} \, d\mathbf{x} \\ \implies 0 &= \lambda_2 \left[\frac{d}{dh} \frac{1}{2} (B(\mathbf{x}) + hC(\mathbf{x}) - B_0(\mathbf{x}))^2 \right]_{h=0} \\ &= \lambda_2 [(B(\mathbf{x}) + hC(\mathbf{x}) - B_0(\mathbf{x}))C(\mathbf{x})]_{h=0} \\ &= \lambda_2 (B(\mathbf{x}) - B_0(\mathbf{x}))C(\mathbf{x}) \end{aligned}$$

Auch hier muss dies wieder für alle Testfunktionen $C(\mathbf{x})$ erfüllt sein, daher bleibt

$$0 = \lambda_2 (B(\mathbf{x}) - B_0(\mathbf{x})).$$

Fassen wir die Ableitungen aller drei Summanden zusammen, folgen die Euler-Lagrange-Gleichungen aus Gleichung (3.4):

$$0 = \frac{\partial}{\partial x_1} \left(\frac{B_{x_1}}{\sqrt{B_{x_1}^2 + B_{x_2}^2}} \right) + \frac{\partial}{\partial x_2} \left(\frac{B_{x_2}}{\sqrt{B_{x_1}^2 + B_{x_2}^2}} \right) - \lambda_1 - \lambda_2 (B - B_0),$$

mit

$$\frac{\partial B}{\partial \mathbf{n}} = 0 \quad \text{auf } \Gamma.$$

B.6 Herleitung der Bestimmung von λ

Gegeben ist Gleichung (3.7), aus der λ bestimmt werden soll,

$$\begin{aligned} B_t &= \frac{\partial}{\partial x_1} \left(\frac{B_{x_1}}{\sqrt{B_{x_1}^2 + B_{x_2}^2}} \right) + \frac{\partial}{\partial x_2} \left(\frac{B_{x_2}}{\sqrt{B_{x_1}^2 + B_{x_2}^2}} \right) - \lambda (B - B_0) \\ &= \left\langle \nabla, \frac{\nabla B}{|\nabla B|_2} \right\rangle_2 - \lambda (B - B_0). \end{aligned}$$

Befindet sich die Iteration in einem stabilen Zustand, so ändert sich B nicht mehr und es gilt $B_t = 0$. Multiplizieren wir daher im stabilen Zustand mit $(B - B_0)$, so folgt:

$$\begin{aligned}
 0 &= (B - B_0) \left(\left\langle \nabla, \frac{\nabla B}{|\nabla B|_2} \right\rangle_2 - \lambda(B - B_0) \right) \\
 &= \left\langle \nabla(B - B_0), \frac{\nabla B}{|\nabla B|_2} \right\rangle_2 - \lambda(B - B_0)^2 \\
 &= \left\langle \nabla B, \frac{\nabla B}{|\nabla B|_2} \right\rangle_2 - \left\langle \nabla B_0, \frac{\nabla B}{|\nabla B|_2} \right\rangle_2 - \lambda(B - B_0)^2 \\
 &= -|\nabla B|_2 + \frac{\langle \nabla B_0, \nabla B \rangle_2}{|\nabla B|_2} - \lambda(B - B_0)^2 \\
 &\implies -\lambda(B - B_0)^2 = |\nabla B|_2 - \frac{\langle \nabla B_0, \nabla B \rangle_2}{|\nabla B|_2}
 \end{aligned}$$

Integrieren wir nun beide Seiten, so erkennt man, dass auf der linken Seite nach Gleichung (3.3) die Varianz erscheint:

$$\begin{aligned}
 &-\lambda 2 \frac{1}{2} \int_{\Omega} (B - B_0)^2 \, d\mathbf{x} = \int_{\Omega} |\nabla B|_2 - \frac{\langle \nabla B_0, \nabla B \rangle_2}{|\nabla B|_2} \, d\mathbf{x} \\
 \implies &-\lambda 2\sigma^2 = \int_{\Omega} |\nabla B|_2 - \frac{\langle \nabla B_0, \nabla B \rangle_2}{|\nabla B|_2} \, d\mathbf{x} \\
 \implies &\lambda = -\frac{1}{2\sigma^2} \int_{\Omega} |\nabla B|_2 - \frac{\langle \nabla B_0, \nabla B \rangle_2}{|\nabla B|_2} \, d\mathbf{x}
 \end{aligned}$$

C Matlab Implementierungen

C.1 Kochrezept zur Homogenisierung

```
1  % 1. Konvertierung in Grauwert
2  Img = rgb2gray( Img );

4  % 2. Invertierung
5  Img = imcomplement( Img );

7  % 3. Treshold segmentation
8  bw = im2bw( Img, 0.02 );

10 % 4. filling
11 bw = imfill( bw, 'holes' );

13 % 5. Erosion mit 'disk' und Durchmesser 2
14 bw = imerode( bw, strel( 'disk', 2 ) );

16 % 6. Opening mit 'disk' und Durchmesser 2
17 bw = imopen( bw, strel( 'disk', 3 ) );

19 % 7. Maskierung
20 segbin = immultiply( Img, bw );

22 % 8. 'segbin' mit low-pass der Groesse 29 in 4 Iterationen glaetten
23 filt = ones( 29 ) / ( 29^2 );
24 falt = segbin;
25 for i = 1:4, falt = filter2( falt, falt, 'same' ); end;

27 % 9. Faltungsbild erzeugen (Maske aus 6. ueber 8. legen)
28 falt = falt .* bw;

30 % 10. Faltungsbild invertieren
31 falt = 255 - falt;

33 % 11. Faltungsbild aus 10. mit 'segbin' (7.) multiplizieren und 255 addieren
34 Img = floor( falt .* double(segbin) + 255 );

36 % 12. Normalisieren
37 Img = 255 * ( Img - min( Img(:) ) ) / ( max( Img(:) ) - min( Img(:) ) );
38 Img = uint8( floor( Img ) );
```

C.2 FIR-Filter

```

1  function [teta_x,teta_y]=createfilter(N,mu,lambda,display);
2  % Aufruf: [teta_x,teta_y]=createfilter(N,mu,lambda,display);
3  %
4  % erzeuge Filter der Gr"o"se NxN
5  % mu,lambda - Materialkonstanten
6  % display - ein[1]/aus[0]

8  if nargin<4, display=0; end;

10 d=(N-1)/2;
11 [x1,x2]=meshgrid(-d:d,-d:d);
12 x1=x1/(N-1)+0.5; x2=x2/(N-1)+0.5;

14 tx1=zeros(size(x1)); tx2=tx1; ty1=tx1; ty2=tx2;
15 factor=4/(mu*(2*mu+lambda)*pi^2*(N-1)^2);

17 for i=0:N-1
18     for j=0:N-1
19         if ((i~=0) | (j~=0))
20             p=((i^2+j^2)^2*gamma(i,j));
21             temp=sin(i*pi/2)*cos(j*pi/2)/p;
22             tx1=tx1+temp*(mu*i^2+(2*mu+lambda)*j^2)*sin(i*pi*x1).*cos(j*pi*x2);
23             tx2=tx2+temp*(-(mu+lambda)*i*j*cos(i*pi*x1).*sin(j*pi*x2));
24             temp=cos(i*pi/2)*sin(j*pi/2)/p;
25             ty1=ty1+temp*(-(mu+lambda)*i*j*cos(j*pi*x2).*sin(i*pi*x1));
26             ty2=ty2+temp*(mu*j^2+(2*mu+lambda)*i^2)*sin(j*pi*x2).*cos(i*pi*x1);
27         end;
28     end;
29 end;
30 tx1=factor*tx1; tx2=factor*tx2;
31 ty1=factor*ty1; ty2=factor*ty2;

33 if display
34     figure(1), quiver(x1,x2,tx1,tx2,1.5); axis image; title('filter in x_1-direction');
35     figure(2), quiver(x1,x2,ty1,ty2,1.5); axis image; title('filter in x_2-direction');
36 end;

38 teta_x=complex(tx1,tx2);
39 teta_y=complex(ty1,ty2);

41 return;

43 %-----
44 function g=gamma(i,j);
45 if i==0
46     if j==0, g=0; % i=j=0
47     else, g=2; end; %
48 else
49     if j==0, g=2;
50     else, g=1; end; % i=j=1;
51 end;
52 return;

```

C.3 Registrierung & Homogenisierung

```

1 function [outputPath,timestamp]=elRegPyramid(varargin);
2 % call: [outputPath,timestamp]=elRegPyramid(varargin);
3 %
4 % Arguments:
5 %   pathName - Path for source image data           ['.'']
6 %   pathNameU - Path for deformation fields         ['.'']
7 %   alpha - regularizing factor for deformation field u [1]
8 %   beta - regularizing factor for scaling factor c [0]
9 %   tau - scaling parameter for stopping criteria
10 %   sigma - resampling factor for multi-resolution-approach [0.5]
11 %   kMax - maximum number of steps allowed [10]
12 %   nLevel - number of levels for multiresolution pyramide [1]
13 %   maxStepSize - maximum step size for deformation update [0.75]
14 %   cutLargeSteps - if set, cut deformation updates larger than maxStepSize [0]
15 %   method - method used in estimation of material-constant ('fehler' or 'masse') ['masse']
16 %   writeDeformedFinal - if set, the finally deformed images were written to disk [1]
17 %   writeDeformed - if set, write deformed images to disk [0]
18 %   writeStep - write deformed images every ... steps [10]
19 %   fileFormatIn - file format of input images (case sensitive) ['tif']
20 %   fileFormatOut - file format of output images (case sensitive) ['tif']
21 %   epsMachine - machine precision [1e-8]
22 %
23 % function providing elastic registration of images of a series of
24 % serial sections combined with simultaneous homogenization
25 %
26 % implemented and extended by SW in 2003/2004/2005/2007
27 % -----

29 % store start time of batch to create unique folder names and to handle data correctly
30 startTime=now;
31 timestamp=['_',datestr(startTime,30)];

33 % store name of used computer (helps analyzing results if more than one computer is used)
34 [s,hostname]=unix('hostname');

36 % check, if a valid path name is given
37 [pathName]=getopt('pathName',0,varargin{:});
38 if exist(pathName)
39     if ~strcmp(pathName(end),'/'), pathName = [pathName,'/']; end;
40 else
41     error('Given path for image data does not exist!');
42 end; % end if exist(pathName)

44 % check, if a valid deformation field path
45 [pathNameU]=getopt('pathNameU',[],varargin{:});
46 if isempty(pathNameU)
47     if exist(pathNameU)
48         if ~strcmp(pathNameU(end),'/'), pathNameU = [pathNameU,'/']; end;
49     else
50         error('Given path for deformation field does not exist!');
51     end; % end if exist(pathNameU)
52 end;

54 % parse parameters from varargin
55 [alpha]=getopt('alpha',1,varargin{:});
56 [beta]=getopt('beta',0,varargin{:});
57 [kMax]=getopt('kMax',10,varargin{:});
58 [nLevel]=getopt('nLevel',0,varargin{:});
59 [maxStepSize]=getopt('maxStepSize',0.75,varargin{:});
60 [writeDeformedFinal]=getopt('writeDeformedFinal',1,varargin{:});
61 [writeDeformed]=getopt('writeDeformed',0,varargin{:});
62 [writeStep]=getopt('writeStep',10,varargin{:});
63 [justRun]=getopt('justRun',0,varargin{:}); % skip all graphical output, just run

```

```

64 [cutLargeSteps]=getopt('cutLargeSteps',0,varargin{:});
65 [method]=getopt('method','masse',varargin{:});
66 [fileFormatIn]=getopt('fileFormatIn','tif',varargin{:});
67 [fileFormatOut]=getopt('fileFormatOut','tif',varargin{:});
68 [orthoScale]=getopt('orthoScale',1,varargin{:}); % scale orthogonal output by this factor
69 [tau] = getopt('tau',5e-1,varargin{:});
70 [sigma] = getopt('sigma',0.5,varargin{:});
71 [epsMachine] = getopt('epsMachine',1e-8,varargin{:});

73 % catch wrong parameter input for kMax
74 if kMax==0,
75     fprintf('maximale Schrittzahl ist 0, Programm beendet!\n\n');
76     return;
77 end; % if kMax==0

79 % create output
80 outputPath=[pathName,timestamp,'/'];
81 fprintf('create output folder: %s ... ',outputPath);
82 [succ,message] = mkdir(pathName,timestamp);
83 if ~succ, disp(message); error('Can't create output folder!'); end;
84 fprintf('done\n');

86 % create file "_info.txt" in output folder as log-file
87 fid=fopen([outputPath,'_info.txt'],'w');
88 if fid==-1, error('Can't create file ''_info.txt''); end;

90 % print all parameters to console and file
91 fprintf(fid,'-----\n');
92 fprintf(fid,'| Elastic Registration of images of Serial Sections |\n');
93 fprintf(fid,'-----\n');
94 fprintf(fid,'\n%s',datestr(startTime));
95 fprintf(fid,'\n\n');
96 fprintf(fid,'\tkMax           = %i\n',kMax);
97 fprintf( '\tkMax           = %i\n',kMax);
98 fprintf(fid,'\talpha          = %i\n',alpha);
99 fprintf( '\talpha          = %i\n',alpha);
100 fprintf(fid,'\tbeta           = %i\n',beta);
101 fprintf( '\tbeta           = %i\n',beta);
102 fprintf(fid,'\ttau            = %1.3f\n',tau);
103 fprintf( '\ttau            = %1.3f\n',tau);
104 fprintf(fid,'\tsigma          = %1.3f\n',sigma);
105 fprintf( '\tsigma          = %1.3f\n',sigma);
106 fprintf(fid,'\tmaxStepSize     = %f\n',maxStepSize);
107 fprintf( '\tmaxStepSize     = %f\n',maxStepSize);
108 fprintf(fid,'\tnLevel         = %i\n',nLevel);
109 fprintf( '\tnLevel         = %i\n',nLevel);
110 fprintf(fid,'\tmethod          = %s\n',method);
111 fprintf( '\tmethod          = %s\n',method);
112 fprintf(fid,'\tcutLargeSteps   = %i\n',cutLargeSteps);
113 fprintf( '\tcutLargeSteps   = %i\n',cutLargeSteps);
114 fprintf(fid,'\tepsMachine       = %1.3f\n',epsMachine);
115 fprintf( '\tepsMachine       = %1.3f\n',epsMachine);
116 fprintf(fid,'\twriteDeformedFinal = %i\n',writeDeformedFinal);
117 fprintf( '\twriteDeformedFinal = %i\n',writeDeformedFinal);
118 fprintf(fid,'\twriteDeformed    = %i\n',writeDeformed);
119 fprintf( '\twriteDeformed    = %i\n',writeDeformed);
120 fprintf(fid,'\twriteStep        = %i\n',writeStep);
121 fprintf( '\twriteStep        = %i\n',writeStep);
122 fprintf(fid,'Image Data Path      : %s\n',pathName);
123 fprintf( 'Image Data Path      : %s\n',pathName);
124 if isempty(pathNameU),
125     defFieldPathName = '- none -';
126 else
127     defFieldPathName = pathNameU;
128 end;
129 fprintf(fid,'Input Deformation Field Path: %s\n',defFieldPathName);
130 fprintf( 'Input Deformation Field Path: %s\n',defFieldPathName);

```



```

131 fprintf(fid,'Output Deformation Field Path: %s\n\n',outputPath);
132 fprintf('    Output Deformation Field Path: %s\n\n',outputPath);

134 % calculate number of slices and digits for displaying it
135 d=dir([pathName, '*. ',fileFormatIn]);
136 N=length(d);
137 if N==0, error(['Can't read any ',fileFormatIn,'-file. Please correct file-format and
    start again']); end;
138 digits = 1 + ceil(log(N+1)/log(10)); % calc, how many digits are needed for counting all
    files
139 [N1,N2] = size(imread([pathName,d(1).name])); % store original image size

141 % calculate images in different resolutions for multi-resolution scheme
142 % and store them using file names like name."fileFormatIn".level?
143 sampleSlides('level',nLevel,'pathName',pathName,'fileFormat',fileFormatIn,'sigma',sigma);

145 % calculate initial error for all image levels
146 for level=0:nLevel-1
147     %init solving system
148     calcDeformation('what','init','n1',N1*sigma^(level),'n2',N2*sigma^(level));
149     fprintf('level %i, ',level);
150     % calculate SSD, \nabla SSD, error, mass and estimated \mu
151     [DR0{level+1},fehler0{level+1},masse0{level+1},muVec{level+1}] = calcError(pathName,
        pathNameU,d,fid,level,alpha,0,maxStepSize);
152 end;

154 % create first and last column resp. row of orthogonally resampled slices
155 A = zeros(N1,orthoScale*N); B = zeros(orthoScale*N,N2);

157 % calculate image size for smallest image in image pyramid
158 n1 = N1*sigma^(nLevel-1);
159 n2 = N2*sigma^(nLevel-1);

161 % initialize misc. variables
162 k=1;
163 fehlerRatio=inf;
164 fk = inf; fk_1 = inf;
165 [maxFehler,s] = max(fehler0{1}(2:end-1));
166 s = median(s);
167 fehler = zeros(N,1);
168 fehlerOld = fehler;
169 normGradError = fehler;
170 norm_dR = zeros(N,1);
171 norm_R = zeros(N,1);

173 % initialize deformation field; if one exists, resample to lowest
174 % resolution, otherwise, create zero valued deformation field
175 if ~isempty(pathNameU)
176     initU('what','sample','uOutputPath',outputPath,'sample',-nLevel+1,'n1',N1,'n2',N2,'
        sigma',sigma);
177 else
178     initU('what','zero','uOutputPath',outputPath,'d',d,'n1',n1,'n2',n2);
179 end;
180 fprintf('start timing\n\n'); tic;

182 % start for all levels
183 for level=(nLevel-1):-1:0
184     % print number of current slice
185     fprintf('----- Level %2i -----\n',level);
186     fprintf(fid,'----- Level %2i -----\n',level);

188     % count steps for each level of pyramid
189     kLevel=1;

191     % initialize inverse matrix
192     calcDeformation('what','init','n1',n1,'n2',n2);

```



```

256     f1 = (RnuMinus1-2*Rnu+RnuPlus1).*(Rnu([2:end,end],:)-Rnu([1,1:end-1],:));
257     f2 = (RnuMinus1-2*Rnu+RnuPlus1).*(Rnu(:,[2:end,end])-Rnu(:,[1,1:end-1]));
258 end;

260 % load current (slice-depending) deformation field u from disk
261 [u1Nu,u2Nu] = getU(outputPath,d,nu,n1,n2);

263 % solve system of differential equations to get update v of deformation field u
264 [v1,v2] = calcDeformation('what','solve','f1',f1*invAlphaHat,'f2',f2*invAlphaHat,
    'mu',1.0/currMuVec(nu));

266 % calculate maximum absolute value of deformation update
267 normV = norm([v1(:);v2(:)],inf);

269 % cut deformation update v, if v>maxStepSize
270 if ((normV > maxStepSize) & (cutLargeSteps))
271     % find all matrix positions, where v is greater than maxStepSize
272     cutIndex = find( (abs(v1)>maxStepSize) | (abs(v2)>maxStepSize) );
273     % calculate maximum absolute value of v
274     maxV12 = max(abs(v1(cutIndex)),abs(v2(cutIndex)));
275     % normalize v, where at least one component is greater than maxStepSize
276     v1(cutIndex) = v1(cutIndex) ./ maxV12 .* maxStepSize;
277     v2(cutIndex) = v2(cutIndex) ./ maxV12 .* maxStepSize;
278     % calculate maximum norm again
279     normV = norm([v1(:);v2(:)],inf);
280     % clear temporal variables to save space
281     clear cutIndex maxV12;
282 else
283     [v1,v2]=calculateArmijoStep('u1',u1,'u2',u2,'v1',v1,'v2',v2,'
        maxStepsArmijo',10);
284 end;

286 % keep track of maximal and minimal deformation update
287 maxNormV = max(maxNormV,normV);
288 minNormV = min(minNormV,normV);

290 % update deformation field
291 u1Nu = u1Nu + v1;
292 u2Nu = u2Nu + v2;

294 % clear temporal variables to save RAM
295 clear v1 v2;

297 % write deformation field to disk
298 writeDeformation(u1Nu,u2Nu,outputPath,d,nu);

300 % apply current deformation without gray value correction
301 RnuOld = Rnu;
302 Rnu = getSlice('pathName',pathName,'pathNameU',outputPath,'d',d,'nu',nu,'level',
    level,'corrected',beta);
303 norm_dR(nu) = norm(RnuOld(:) - Rnu(:))/(n1*n2);
304 norm_R(nu) = norm(Rnu(:))/(n1*n2);
305 clear RnuOld;

307 % calculate current error
308 if nu>1,
309     fehler(nu) = sum( (Rnu(:) - RnuMinus1(:)).^2)/(n1*n2);
310     normGradError(nu) = sqrt(sum([reshape(((RnuMinus1-2*Rnu+RnuPlus1).*(Rnu([2:end
        ,end],:)-Rnu([1,1:end-1],:))),n1*n2,1).^2 + ...
311         reshape(((RnuMinus1-2*Rnu+RnuPlus1).*(Rnu(:,[2:end,
            end])-Rnu(:,[1,1:end-1]))),n1*n2,1).^2)))/(n1*n2
        );
312 else
313     fehler(nu) = sum( (Rnu(:) - RnuPlus1(:)).^2)/(n1*n2);
314 end;

316 % if there is gray value correction, update current correction factor

```

```

317     if beta
318         % load correction factors from disk
319         c = getC(outputPath,d,nu,n1,n2);
320         % solve linear system
321         if (nu==1)
322             c = loeseTV(RnuPlus1,Rnu,c,betaHat);
323         else
324             c = loeseTV(RnuMinus1,Rnu,c,betaHat);
325         end;
326         % write new correction factor to disk
327         writeC(c,outputPath,d,nu);
328     end;

330     % collect lines for orthogonally resampled slices
331     [il1,il2] = getImageLine(pathName,outputPath,d,nu,level,beta);
332     for sc=0:orthoScale-1
333         A(:,orthoScale*nu-sc) = il1;
334         B(orthoScale*nu-sc,:) = il2;
335     end; %for sc

337 end; % for nu

339 % write orthogonally resampled slices to disk
340 imwrite(uint8(A),[outputPath,'orthogonalView1_',num2strPad(k,digits),'.',
    fileFormatOut]);
341 imwrite(uint8(B),[outputPath,'orthogonalView2_',num2strPad(k,digits),'.',
    fileFormatOut]);

343 % calculate errorRatios and plot images, if plots are allowed
344 fehlerRatioOld=fehlerRatio;
345 f0 = fehler0[level+1];
346 if k == 1,
347     fk_1 = f0;
348 else
349     fk_1 = fk;
350 end;
351 fk = getErrorOfSingleSlice(pathName,outputPath,d,s,level,beta,sigma);
352 fehlerRatio=fk/(f0(s));

354 % store current ssd-error
355 DR(k) = sum(fehler)/N;

357 if ~justRun % if justRun is set, don't plot, just calculate
358     figure(1); colormap(gray(256));
359     subplot(1,2,1); image(A'); axis image off;
360     subplot(1,2,2); image(B); axis image off;
361     drawnow; pause(0);
362     eph = figure(2);
363     % plot curves
364     plot(1:N,fehler0[level+1],'r.-',1:N,fehler,'b.-');
365     % plot a legend
366     legend('Initial Error','Current Error');
367     % save plot as figure
368     saveas(eph,[outputPath,'error_',num2strPad(k,digits)],'png'); pause(0);
369     drawnow; pause(0);
370     figure(3);
371     plot(1:length(DR),DR,'b-',1:length(DR),DR,'r.');
```

```

381 % if writeDeformed is set, write deformed slices to disk
382 if (writeDeformed & ((k==kMax) | (mod(k,writeStep)==0) | (level>0)))
383     nummer = num2strPad(k,digits);
384     levelPad = num2strPad(1+ceil(log(nLevel+1)/log(10)));
385     mkdir(outputPath,['level',levelPad,'_',nummer,'steps']);
386     fprintf('\twrite deformed slices to folder \n\t%s\n\t',[outputPath,'level',
387         levelPad,'_',nummer,'steps/']);
387     fprintf(fid,'\twrite deformed slices to folder \n\t%s\n\t',[outputPath,'level',
388         levelPad,'_',nummer,'steps/']);
388     deformImages('dataPath',pathName,'uPath',outputPath,'outputPath',[outputPath,'
389         level',levelPad,'_',nummer,'steps/'],'sample',-level,'fileFormatIn',
390         fileFormatIn,'fileFormatOut',fileFormatOut);
389     end;

391 % update step counter (overall and per level)
392 k=k+1;
393 kLevel=kLevel+1;

395 loop = checkStoppingCriteria(k,kMax,kLevel,level,s,beta,tau,epsMachine,fehler,
396     fehlerOld,normGradError,norm_dR,norm_R,fid);
397 end; % while

398 % if on lower scale, sample deformation field and gray value correction term
399 if (level > 0),
400     % upsample u
401     initU('what','sample','uOutputPath',outputPath,'sample',1,'n1',n1,'n2',n2,'sigma',
402         sigma);
402     if beta,
403         % upsample c
404         upsampleC(outputPath,n1,n2,sigma);
405     end; % if beta
406     % adapt image sizes for next finer level
407     n1 = N1*sigma^(level-1);
408     n2 = N2*sigma^(level-1);
409     tau = sigma*tau;
410 end; % if level > 0

412 end; % for level=(nLevel-1):-1:0

414 % if writeDeformedFinal is set, write deformed slices to disk
415 if (writeDeformedFinal & ~writeDeformed)
416     mkdir(outputPath,'finalDeformation');
417     fprintf('\twrite finally deformed slices to folder \n\t%s\n\t',[outputPath,'
418         finalDeformation/']);
418     fprintf(fid,'\twrite finally deformed slices to folder \n\t%s\n\t',[outputPath,'
419         finalDeformation/']);
419     deformImages('dataPath',pathName,'uPath',outputPath,'outputPath',[outputPath,'
420         finalDeformation/'],'sample',0,'fileFormatIn',fileFormatIn,'fileFormatOut',
421         fileFormatOut);
420 end;

422 % calculate final error
423 DRend = sum(fehler)/N;

425 % print distance and error information
426 fprintf('\nInitial Distance: %f \nFinal Distance: %f \nRelative Error: %3.3f%\n\n',DR0{
427     level+1}/(N1*N2),DRend/(N1*N2),DRend/DR0{level+1}*100);
427 fprintf(fid,'\nInitial Distance: %f \nFinal Distance: %f \nRelative Error: %3.3f%\n\n',
428     DR0{level+1}/(N1*N2),DRend/(N1*N2),DRend/DR0{level+1}*100);

429 % get elapsed time and print it
430 eltime=toc;
431 fprintf('\ncomplete runtime of batch: %5.1f sec = %5.1f min = %5.3f h\n',eltime,eltime/60,
432     eltime/3600);
432 fprintf(fid,'\ncomplete runtime of batch: %5.0f sec = %5.1f min = %5.3f h\n',eltime,eltime
433     /60,eltime/3600);

```

```
433 fprintf(fid, '\tused computer: %s\n', hostname);
435 % close log-file
436 fclose(fid);
438 % if plots are allowed, plot error curves
439 if ~justRun
440     % open new figure
441     eph=figure(2);
442     % plot curves
443     plot(1:N, fehler0{level+1}, 'r.-', 1:N, fehler, 'b.-');
444     % plot a legend
445     legend('Initial Error', 'Final Error');
446     % save plot as figure
447     saveas(eph, [outputPath, 'error'], 'fig'); pause(0);
448     drawnow;
449     pause(0);
450 end;
452 % clean up and quit program
453 fprintf('clean up ...');
454 calcDeformation('what', 'clear');
455 fprintf(' done\n');
457 disp(' '); disp('... regular termination of program!');
459 return;
```

Literaturverzeichnis

- [ABKC90] N.M. Alpert, J.F. Bradshaw, D. Kennedy, and J.A. Correia, *The principal axes transformation – A method for image registration*, Journal of nuclear medicine **31** (1990), no. 10, 1717–1722.
- [ACC91] J. Aferzon, R. Chau, and D. Cowan, *A microcomputer-based system for three-dimensional reconstructions from tomographic or histologic sections*, Anal. Quant. Cytol. Histol. **13** (1991), 80–88.
- [ACL87] L. Axel, J. Costantini, and J. Listerud, *Intensity correction in surface-coil MR imaging*, Am J Roent (1987), no. 148, 418–420.
- [APG06] E. Ardizzone, R. Pirrone, and O. Gambino, *Morphological Exponential Entropy Driven-HUM*, Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE, 2006, pp. 3771–3774.
- [AV94] R. Acar and C.R. Vogel, *Analysis of bounded variation penalty methods for ill-posed problems*, Inverse Problems **10** (1994), 1217–1229.
- [BBB05] A. Biswas, P. Bhowmick, and B.B. Bhattacharya, *Reconstruction of torn documents using contour maps*, IEEE International Conference on Image Processing (ICIP), vol. 3, 2005, pp. 517–520.
- [BK89] R. Bajcsy and S. Kovačič, *Multiresolution elastic matching*, Computer Vision, Graphics and Image Processing **46** (1989), 1–21.
- [BLJ+90] C. Bron, D. Launay, M. Jourlin, H. Gautschi, T. Bächli, and J. Schüpbach, *Three dimensional electron microscopy of entire cell*, J Microsc. **157** (1990), 115–126.
- [BMR98] B.H. Brinkmann, A. Manduca, and R.A. Robb, *Optimized homomorphic unsharp masking for MR grayscale inhomogeneity correction*, IEEE Transaction on Medical Imaging **17** (1998), no. 2, 161–171.
- [BN96] M. Bro-Nielsen, *Medical image registration and surgery simulation*, Ph.D. thesis, Informatics and Mathematical Modelling, Technical University of Denmark, 1996, pp. xxviii–276.
- [BNG96] M. Bro-Nielsen and C. Gramkow, *Fast fluid registration of medical images*, Lecture Notes in Computer Science **1131** (1996), 267–276.

- [Bor83] G. Born, *Die Plattenmodellierungsmethode*, Arch. Mikr. Anat. **22** (1883), 584–599.
- [Bro81] C. Broit, *Optimal registration of deformed images*, Ph.D. thesis, Computer and Information Science, University of Pennsylvania, 1981.
- [Bro92] L.G. Brown, *A survey of image registration techniques*, ACM Computing Surveys **24** (1992), no. 4, 325–376.
- [Bro05] T. Brox, *From Pixels to Regions: Partial Differential Equations in Image Analysis*, Ph.D. thesis, Faculty of Mathematics and Computer Science, Saarland University, Germany, April 2005.
- [BS94] M. Baumann and H. Scharf, Ann. Anat. **176** (1994), 849–865.
- [BSW⁺09] J. Bieberstein, C. Schumann, A. Weihusen, T. Boehler, S. Wirtz, P. Bruners, D. Schmidt, C. Trumm, M. Niethammer, G. Haras, et al., *Fast registration of pre-and peri-interventional CT images for targeting support in radiofrequency ablation of hepatic tumors*, Proceedings of SPIE, vol. 7261, 2009, p. 72610F.
- [BT80] S.T. Barnard and W.B. Thompson, *Disparity analysis of images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-2** (1980), no. 4, 333–340.
- [BT94] P.K. Banerjee and A.W. Toga, *Image alignment by integrated rotational and translational transformation matrix*, Phy. Med. Biol. **39** (1994), 1969–1988.
- [BWP07] T. Boehler, S. Wirtz, and H.O. Peitgen, *A combined algorithm for breast MRI motion correction*, Proceedings of SPIE, vol. 6514, 2007, p. 65141R.
- [CBFAB94] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, *Two deterministic half-quadratic regularization algorithms for computed imaging*, IEEE International Conference on Image Processing (ICIP), vol. 2, 1994, pp. 168–172.
- [CFF98] M.G. Chung, M.M. Fleck, and D.A. Forsyth, *Jigsaw puzzle solver using shape and color*, 4th Intern. Conf. Signal Proc., vol. 2, 1998, pp. 877–880.
- [CGM99] T. Chan, G. Golub, and P. Mulet, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput. **20** (1999), no. 6, 1964–1977.
- [Chr94] G.E. Christensen, *Deformable shape models for anatomy*, Ph.D. thesis, Sever Institute of Technology, Washington University, 1994.

- [Dav79] P. J. Davis, *Circulant matrices*, Chelsea Publishing, New York, 1979.
- [Die76] M. Dierker, *An algorithm for the alignment of serial sections*, Computer technology on neuroscience edition, John Wiley & Sons, New York, 1976.
- [DMD⁺04] J. Dauguet, J.F. Mangin, T. Delzescaux, V. Frouin, et al., *Robust inter-slice intensity normalization using histogram scale-space analysis*, Lecture Notes in Computer Science (2004), 242–249.
- [FM99] B. Fischer and J. Modersitzki, *Fast inversion of matrices arising in image processing*, Numerical Algorithms **22** (1999), 1–11.
- [FM02] ———, *A unified approach to fast image registration and a new curvature based registration technique*, Preprint a-02-07, Institute of Mathematics, Medical University of Lübeck, 2002.
- [Fol95] J.D. Foley, *Computer Graphics: Principles and Practice in C*, Addison-Wesley Professional, 1995.
- [For84] O. Forster, *Analysis 1-3*, Vieweg, Braunschweig, 1983/84.
- [GBJ⁺91] P. Gremillet, C. Bron, M. Jourlin, T. Bachi, and J. Schüpbach, *Dedicated image analysis techniques for three-dimensional reconstruction from serial sections in electron microscopy*, Mach. Vis. Appl. **4** (1991), 289–296.
- [GG65] J. Glaser and M. Glaser, *A semi-automatic computer-microscope for the analysis of neuronal morphology*, IEEE Trans. Biomed. Eng. **12** (1965), 22–31.
- [GHB93] F. Gallyas, M. Hsu, and G. Buzsaki, *Four modified silver methods for thick sections of formaldehyde-fixed mammalian central nervous tissue: 'dark' neurons, perikarya of all neurons, microglial cells and capillaries*, Journal of Neuroscience Methods **50** (1993), no. 2, 159–164.
- [GMB02] D. Goldberg, C. Malon, and M. Bern, *A global approach to automatic solution of jigsaw puzzles*, Proceedings of the eighteenth annual symposium on Computational geometry, 2002, pp. 82–87.
- [GMW81] P.E. Gill, W. Murray, and M.H. Wright, *Practical optimization*, Academic Press, London, 1981.
- [Gök88] G. Göke, *Moderne Methoden der Lichtmikroskopie*, Kosmos-Wissenschaft, 1988.

- [GR92] D. Geman and G. Reynolds, *Constrained restoration and the recovery of discontinuities*, IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992), no. 3, 367–383.
- [GvL89] G. H. Golub and C. F. van Loan, *Matrix computations*, second ed., The Johns Hopkins University Press, Baltimore, 1989.
- [HBHH01] D.L.G. Hill, P.G. Batchelor, M. Holden, and D.J. Hawkes, *Medical image registration*, Phys. Med. Biol. **46** (2001), R1–R45.
- [HH88] L.S. Hibbard and R.A. Hawkins, *Objective image alignment for three-dimensional reconstruction of digital autoradiograms*, Journal of Neuroscience Methods **26** (1988), 55–74.
- [HS81] B.K.P. Horn and B.G. Schunck, *Determining optical flow*, Artificial Intelligence **17** (1981), 185–204.
- [Jäh97] B. Jähne, *Digitale bildverarbeitung*, vol. 4, Springer Verlag, 1997.
- [JC83] E. Johnson and J. Capowski, *A system for the three-dimensional reconstruction of biological structures*, Comp. Biomed. Res. **16** (1983), 79–87.
- [JCGH02] G.A. Johnson, G.P. Cofer, S.L. Gewalt, and L.W. Hedlund, *Morphologic phenotyping with MR microscopy: The visible mouse*, Radiology **222** (2002), no. 3, 789–793.
- [LA99] H. Lester and S.R. Arridge, *A survey of hierarchical non-linear medical image registration*, Pattern Recognition **32** (1999), 129–149.
- [Mag00] R. Magerle, *Nanotomography*, Phys. Rev. Lett. **85** (2000), no. 13, 2749–2752.
- [MB03] G. Malandain and E. Bardinet, *Intensity compensation within series of images*, Lecture Notes in Computer Science **2879** (2003), 41–49.
- [MLS79] E. Macagno, C. Levinthal, and I. Sobel, *Three-dimensional computer reconstruction of neurons and neuronal assemblies*, Annu. Rev. Biophys. Bioeng. **8** (1979), 323–351.
- [Mod04] J. Modersitzki, *Numerical methods for image registration*, Oxford University Press, 2004.
- [MOSL99] J. Modersitzki, W. Oberlöer, O. Schmitt, and G. Lustig, *Elastic matching of very large digital images on high performance clusters*, Lecture Notes in Computer Science **1593** (1999), 141–149.

- [MSW04] J. Modersitzki, O. Schmitt, and S. Wirtz, *Registration of histological serial sections*, Proceedings of ECMI 2004, 13th European conference on mathematics for industry, 2004.
- [MSW06] ———, *Registration of histological serial sectionings*, MATHEMATICS IN INDUSTRY **10** (2006), 63.
- [MV98] J.B.A. Maintz and M.A. Viergever, *A survey of medical image registration*, Medical Image Analysis **2** (1998), no. 1, 1–36.
- [MW06a] J. Modersitzki and S. Wirtz, *Combining Homogenization and Registration*, Lecture Notes in Computer Science **4057** (2006), 257–263.
- [MW06b] ———, *Mathematical models for registration and applications to medical imaging*, Mathematical Models for Registration and Applications to Medical Imaging. Mathematics in Industry (New York) (Otmar Scherzer, ed.), Springer, 2006.
- [NW06] J. Nocedal and S.J. Wright, *Numerical optimization*, second ed., Springer, New York, 2006.
- [OR90] S. Osher and L.I. Rudin, *Feature oriented image enhancement using shock filters*, SIAM J. Num. Anal. **27** (1990), no. 4, 919–940.
- [ORS⁺01] S. Ourselin, A. Roche, G. Subsol, X. Pennec, and N. Ayache, *Reconstructing a 3D structure from serial histologic sections*, Image Vis. Comp. **19** (2001), 25–31.
- [Ots79] N. Otsu, *A threshold selection method from gray-level histograms*, IEEE Trans. Sys. Man Cyber. **9** (1979), no. 1, 62–66.
- [PG82] W. Perkins and R. Green, *Three-dimensional reconstruction of biological sections*, J. Biomed. Eng. **4** (1982), 37–43.
- [PM88] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, Tech. Report UCB/CSD-88-483, EECS Department, University of California, Berkeley, 1988.
- [PSH⁺07] N. Papenberg, H. Schumacher, S. Heldmann, S. Wirtz, S. Bommersheim, K. Ens, J. Modersitzki, and B. Fischer, *A fast and flexible image registration toolbox: Design and implementation of the general approach*, Bildverarbeitung Für Die Medizin 2007: Algorithmen-Systeme-Anwendungen (2007), 106.
- [Rie88] H. Riesenberger, *Handbuch der Mikroskopie*, vol. 3, VEB Verlag Technik, Berlin, 1988.
- [ROF92] L.I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D **60** (1992), no. 1-4, 259–268.

- [Rom89] B. Romeis, *Mikroskopische Technik*, Urban und Schwarzenberg, 1989.
- [RTL⁺93] H. Rusinek, W.-H. Tsui, A. Levy, M. Noz, and M. de Leon, *Principal axes and surface fitting methods for threedimensional ultrahigh resolution image analysis*, *Biotech. Histochem.* **73** (1993), 44–51.
- [Sch96] C. Schmolke, *Tissue compartments in laminae II-V of rabbit visual cortex – three-dimensional arrangement, size and developmental changes*, *Anat. Embryol.* **193** (1996), 15–33.
- [Sch01] O. Schmitt, *Die multimodale Architektonik des menschlichen Gehirns*, Habilitation, Institute of Anatomy, Medical University of Lübeck, Germany, 2001.
- [SDZ97] T. Schormann, A. Dabringhaus, and K. Zilles, *Extension of the principle axes theory for the determination of affine transformations*, *Informatik aktuell* **19** (1997), 384–391.
- [SE97] O. Schmitt and R. Eggers, *Systematic investigations of the contrast results of histochemical stainings of neurons and glial cells in the human brain by means of image analysis*, *Micron* **28** (1997), no. 3, 197–215.
- [SE99] ———, *Flat-bed scanning as a tool for quantitative neuroimaging*, *J Microsc* **196** (1999), 337–346.
- [SEM05] O. Schmitt, R. Eggers, and J. Modersitzki, *Videomicroscopy, image processing, and analysis of whole histologic sections of the human brain.*, *Microsc Res Tech* **66** (2005), no. 4, 203–218.
- [SF84] C. Schmolke and K. Fleischhauer, *Morphological characteristics of neocortical laminae when studied in tangential semithin sections through the visual cortex of the rabbit*, *Anat. Embryol.* **169** (1984), 125–133.
- [SM83] C. Street and R. Mize, *A simple microcomputer-based three-dimensional serial section reconstruction system (MICROS)*, *J. Neurosci. Meth.* **7** (1983), 359–375.
- [SMH⁺03] O. Schmitt, J. Modersitzki, S. Heldmann, S. Wirtz, L. Hömke, W. Heide, D. Kömpf, and A. Wree, *High resolution spatial localization of the human frontal eye field by excess mass analysis and registration techniques*, 33rd Annual Meeting of the Society for Neuroscience, 2003.
- [SMH⁺04] ———, *Functional and structural location of the frontal eye field*, *The Convergence of Structure and Function*, 2004.

- [SMH⁺05] ———, *Three-dimensional cytoarchitectonic analysis of the posterior bank of the human precentral sulcus.*, *Anat Embryol (Berl)* (2005), 1–14.
- [SMH⁺07] O. Schmitt, J. Modersitzki, S. Heldmann, S. Wirtz, and B. Fischer, *Image registration of sectioned brains*, *International Journal of Computer Vision* **73** (2007), no. 1, 5–39.
- [SMO99] O. Schmitt, J. Modersitzki, and W. Oberl oer, *The human neuroscanning project*, *NeuroImage* **9** (1999), 22.
- [SWM⁺04] O. Schmitt, S. Wirtz, J. Modersitzki, B. Fischer, S. Heldmann, and A. Wree, *Microregistration and microreconstruction of a whole mouse brain*, *NeuroImage* **22** (2004), Suppl 1.
- [SWRM05] S. Scherdel, S. Wirtz, N. Rehse, and R. Magerle, *Curvature registration of scanning probe microscopy images for nanotomography*, *Proc. 10th Int. Fall Workshop – Vision Modelling and Visualisation (Berlin)* (G. Greiner, H. Hornegger, and M. Stamminger, eds.), Akademische Verlagsgesellschaft Aka GmbH, 2005, p. 87.
- [SWRM06] ———, *Non-linear registration of scanning probe microscopy images*, *Nanotechnology* **17** (2006), 881–887.
- [SZ97] T. Schormann and K. Zilles, *Limitations of the principal-axes theory*, *IEEE Transactions on Medical Imaging* **16** (1997), no. 6, 942–947.
- [SZ98] ———, *Three-dimensional linear and nonlinear transformations: An integration of light microscopical and MRI data*, *Human Brain Mapping* **6** (1998), 339–347.
- [TAQ⁺94] A. Toga, K. Ambach, B. Quinn, M. Hutchin, and J. Burton, *Post-mortem anatomy from cryosectioned whole human brain*, *J. Neurosci. Meth.* **54** (1994), 239–252.
- [TB93] A. Toga and P. Banerjee, *Registration revisited*, *J. Neurosci. Meth.* **48** (1993), 1–13.
- [Thi98] J.-P. Thirion, *Image matching as a diffusion process: an analogy with Maxwell’s demons*, *Medical Image Analysis* **2** (1998), no. 3, 243–260.
- [vdEPV93] P.A. van den Elsen, E.-J.D. Pol, and M.A. Viergever, *Medical image matching – a review with classification*, *IEEE Engineering in Medicine and Biology* **12** (1993), 26–39.
- [Vio95] P. Viola, *Alignment by maximization of mutual information*, Ph.D. thesis, Massachusetts Institute of Technology, June 1995, pp. 1–155.

- [VO96] C.R. Vogel and M.E. Oman, *Iterative methods for total variation denoising*, SIAM Journal on Scientific Computing **17** (1996), 227.
- [VW95] P. Viola and W.M. Wells III, *Alignment by maximization of mutual information*, IEEE International Conference on Computer Vision, 1995, pp. 16–23.
- [Wei98] J. Weickert, *Anisotropic diffusion in image processing*, Teubner-Verlag, Stuttgart, 1998.
- [WFMS04a] S. Wirtz, B. Fischer, J. Modersitzki, and O. Schmitt, *Superfast elastic registration of histologic images of a whole rat brain for 3d reconstruction*, vol. 5370, SPIE, 2004, pp. 328–334.
- [WFMS04b] ———, *Vollständige Rekonstruktion eines Rattenhirns aus hochaufgelösten Bildern von histologischen Serienschnitten*, Bildverarbeitung für die Medizin 2004 (T. Tolxdorff, J. Braun, H. Handels, A. Horsch, and H.-P. Meinzer, eds.), Springer, 2004, pp. 204–208.
- [WGH⁺98] R.P. Woods, S.T. Grafton, C.J. Holmes, S.R. Cherry, and J.C. Mazziotta, *Automated image registration: I. General methods and intra-subject, intramodality validation*, Journal of Computer Assisted Tomography **22** (1998), no. 1, 139–152.
- [WGW⁺98] R.P. Woods, S.T. Grafton, J.D.G. Watson, N.L. Sicotte, and J.C. Mazziotta, *Automated image registration: II. Intersubject validation of linear and nonlinear models*, Journal of Computer Assisted Tomography **22** (1998), no. 1, 153–165.
- [Wir00] S. Wirtz, *Fluid matching*, Serie B 00-07, Institut für Mathematik, Medizinische Universität zu Lübeck, November 2000.
- [Wir03] ———, *Nichtparametrische 3D Bildregistrierung - elastic matching* -, Studentenkonferenz Mathematik, Deutsche Mathematiker-Vereinigung, 2003.
- [WL75] R. Ware and V. LoPresti, *Three-dimensional reconstruction from serial sections*, Int. Rev. Cytol. **40** (1975), 325–440.
- [WPFS05] S. Wirtz, N. Papenberg, B. Fischer, and O. Schmitt, *Robust and staining-invariant elastic registration of a series of images from histologic slices*, Proceedings of SPIE 2005, Medical Imaging, 2005.
- [ZYG93] W. Zhao, T.Y. Young, and M.D. Ginsberg, *Registration and three-dimensional reconstruction of autoradiographic images by the disparity analysis method*, IEEE Transactions on Medical Imaging **12** (1993), no. 4, 782–791.

Lebenslauf

seit 01.01.2009	Fraunhofer MEVIS, Institut für Bildgestützte Medizin, <i>Bremen</i> (Wissenschaftlicher Angestellter)
15.01.2006 - 31.12.2008	MeVis Research GmbH, <i>Bremen</i> (Wissenschaftlicher Angestellter)

07.10.2007	Dissertation zum Dr. rer. nat. („ <i>magna cum laude</i> “)
01.01.2003 - 14.01.2006	Universität zu Lübeck, Institut für Mathematik (Promotionsstudent unter Förderung eines DFG-Promotionsstipendiums im Graduiertenkolleg 357 „Effiziente Algorithmen und Mehrskalenmethoden“)
12.12.2002	Diplom in Informatik, Nebenfach Medizinische Informatik (Abschlussnote: „ <i>mit Auszeichnung</i> “)
01.10.1997 - 12.12.2002	Studium der Informatik, Nebenfach Medizinische Informatik, Universität zu Lübeck
01.10.1995 - 30.09.1997	Grundstudium Mathematik, Nebenfach Informatik, Universität Köln

01.07.1995 - 30.09.1995	Tätigkeit als Werkstudent, EC Bayer Dormagen
01.07.1994 - 30.06.1995	Grundwehrdienst im 5. Stabs- und Fernmelderegiment 310, Koblenz (Hilfsausbilder in der AGA)
01.06.1994	Abitur am Norbert Gymnasium Knechtsteden
12.08.1974	geboren in Köln
