



Aus dem Institut für Mathematik
der Universität zu Lübeck

Direktor:
Prof. Dr. Jürgen Prestin

Ein genereller Registrierungsansatz
mit Anwendung in der
navigierten Leberchirurgie

Inauguraldissertation
zur
Erlangung der Doktorwürde
der Universität zu Lübeck
- Aus der Technisch-Naturwissenschaftlichen Fakultät -

Vorgelegt von
Dipl.-Inf. Nils Papenberg
aus Peine

Lübeck, den 20. August 2008

1. Berichterstatter: Prof. Dr. B. Fischer
2. Berichterstatter: Prof. Dr. H. Handels
3. Berichterstatter: Prof. Dr. J. Hornegger

Tag der mündlichen Prüfung 16. Dezember 2008

Zum Druck genehmigt,
Lübeck, 16. Dezember 2008

gez. Prof. Dr. J. Prestin
- Dekan der Technisch-Naturwissenschaftlichen Fakultät -

Inhaltsverzeichnis

1	Einleitung	1
1.1	Das medizinische Problem	1
1.2	Der FUSION-Ansatz	4
1.3	Bisherige Arbeiten	7
1.4	Aufbau der Arbeit und eigene Beiträge	8
I	Allgemeiner Registrierungsansatz	11
2	Das Registrierungsproblem	13
2.1	Kontinuierliche Formulierung	13
2.2	Genereller Ansatz	15
3	Optimierung	19
3.1	Schrittweitenbestimmung	21
3.2	Suchrichtung	22
3.3	Abbruchkriterien	24
3.4	Bedingte Optimierung	25
4	Gitter	31
4.1	Eindimensionale Gitter	32
4.2	Zweidimensionale Gitter	33
4.3	Dreidimensionale Gitter	36
5	Interpolation	41
5.1	Notation	41
5.2	Interpolation im Eindimensionalen	44

5.3	Interpolation im Dreidimensionalen	53
5.4	Diskrete Bilder und Ableitungen	56
5.5	Testdaten	58
5.6	Bildbereiche	59
6	Distanzmaße	63
6.1	Summe der quadrierten Differenzen	64
6.2	Mutual Information	65
6.3	Normalisiertes Gradienten-Feld	67
7	Parametrische Bildregistrierung	71
7.1	Affin-lineare Deformation	71
7.2	Rigide Deformation	72
7.3	Verfahren	75
8	Nicht parametrische Registrierung	77
8.1	Kontinuierliche Formulierung von Regularisierern	78
8.2	Ableitungsoperatoren im Eindimensionalen	79
8.3	Diskrete Regularisierung	81
8.4	Verfahren	84
8.5	Multigrid-Verfahren	86
9	Multiresolutionsverfahren	89
9.1	Motivation	89
9.2	Gitterwechsel im Eindimensionalen	90
9.3	Gitterwechsel für Bilddaten	94
9.4	Gitterwechsel für Deformationen	95
9.5	Verfahren	97
II	Spezielle Registrierungstechniken	101
10	Skalenraumansatz	103
10.1	Skalenräume	103
10.2	Multiskalenverfahren	105
10.3	Kombination von Multiskalen- und Multiresolutionsverfahren . .	107

11 Fokussierung	115
11.1 Neuer Ansatz	116
11.2 Ergebnisse	119
11.3 Eigenschaften	121
12 Landmarkenregistrierung	129
12.1 Motivation	129
12.2 Definition und Notation	130
12.3 Lineare Landmarkenregistrierung	132
12.4 Landmarkenregistrierung mit Thin-Plate-Splines	133
12.5 Landmarken-Vorregistrierung	134
13 Kombinierte Verfahren	137
13.1 CoLD 2003	137
13.2 Diskretisierung der Landmarkenbedingungen	139
13.3 Soft-CoLD	143
13.4 CoLD mit SQP	146
13.5 CoLD mit SVD	150
III Anwendung und Ergebnisse	159
14 Anwendung	161
14.1 Bildgebung	161
14.2 Ergebnisse	164
14.3 FUSION	173
15 Ausblick	177
A Anhang	181
A.1 Singulärwert-Zerlegung	181
A.2 Matrix-Vektor-Multiplikation	182
A.3 Übersicht zu Trackingmethoden	183
Literaturverzeichnis	185

Danksagung

An dieser Stelle möchte ich mich bei mehreren Menschen bedanken, die zum Entstehen der Arbeit beigetragen haben:

Bei Bernd Fischer für seinen fachlichen Rat, seine Führung und seine Erfahrung, die in diese Arbeit eingegangen sind, für seine stets offene Tür und für seine besondere Art, eine Forschungsgruppe zu leiten und eine zugleich produktive und familiäre Atmosphäre zu schaffen, in der zu forschen eine große Freude ist.

Bei Jan Modersitzki, von dem ich lernte, Bilder zu registrieren und dessen Art, präzise zu denken, Erreichtes zu hinterfragen und neue Wege zu betreten, mich lehrte wissenschaftlich zu arbeiten.

Bei Stefan Heldmann für viele Stunden der gemeinsamen Arbeit, besonders aber für seinen motivierenden Bildregistrierungsenthusiasmus.

Bei Thomas Lange für die fruchtbare projektübergreifende Zusammenarbeit und die Bereitstellung der klinischen Datensätze.

Bei allen Mitgliedern der SAFIR-Forschungsgruppe, mit denen ich in den letzten Jahren arbeiten durfte: bei Silke, Janine, Sven, Tia, Konstantin, Sven, Hanno und Stefan - und bei allen Mitarbeitern des Institutes für Mathematik.

Bei Stefan Schlichting und Volker Martens für die Zusammenarbeit im FUSION-Projekt und bei Markus Kleemann und Prof. Dr. Bruch für die hilfreichen Einblicke in die Chirurgie.

Bei Hilke, Janine und Konstantin für das Korrekturlesen der Arbeit.

Bei meinen Eltern, die auch in schwereren Zeiten an mich geglaubt und mich unterstützt haben.

Und - zu allermeist - bei meiner Frau Hilke.

Kapitel 1

Einleitung

In der vorliegenden Arbeit befassen wir uns mit der Registrierung medizinischer Bilddaten, im Speziellen mit Daten, die in der navigierten Leberchirurgie genutzt werden. Sie hat zwei Schwerpunkte: Erstens beschreiben wir einen vollständig neuen Ansatz zur Lösung des Registrierungsproblems. Zweitens spezialisieren wir diesen allgemeinen Ansatz dann auf die Anforderungen aus dem Bereich der Leberchirurgie.

Die Arbeit entstand im Rahmen des FUSION-Projekts (vergleiche Kapitel 14), das sich mit der Erforschung und Verbesserung der Leberchirurgie und der Navigation befasst.

Wir beschreiben zunächst die medizinische Problemstellung und wie sich aus ihr die Notwendigkeit für Registrierungsverfahren ergibt. Anschließend erläutern wir den Aufbau der Arbeit.

1.1 Das medizinische Problem

Neben dem wohl häufigsten Eingriff am Leber-Gallenwegssystem, der Cholezystektomie, der Entfernung der Gallenblase, ist die Resektion von Tumoren aus der Leber einer der Haupteingriffe. Allein in Deutschland erkranken jedes Jahr 20.000 Personen am primären hepatzellulären Karzinom (HCC) [49]. Neben diesem primären Tumor bilden sich in der Leber aufgrund ihrer Funktion als Blutfilterorgan zwischen Darm und systematischem Blutkreislauf bei vielen Krebserkrankungen Lebermetastasen. Rund 90% aller Colonkarzinome streuen in die Leber [6], wobei Darmkrebs bei Männern die dritt- und bei Frauen die zweithäufigste Krebserkrankung in Deutschland darstellt. Bei etwa 60% aller Krebsfälle treten im Krankheitsverlauf Lebermetastasen auf.

Die erfolgreichste Therapie dieser Lebermetastasen stellt deren chirurgische Resektion dar. Drei Faktoren beeinflussen dabei den Therapieerfolg:

- die Resektion des Tumors mit einem ihm umgebenen tumorfreien Sicherheitsrand. Dessen Notwendigkeit wird in [4, 26] belegt.
- die Maximierung des Restvolumens der Leber
- die Minimierung des Patiententraumas.

Die zweite Forderung steht in Konkurrenz zur ersten. Ein Gleichgewicht der Form *so viel Gewebe wie nötig aber so wenig wie möglich* ist anzustreben.

Die Forderungen lassen sich aus chirurgischen Studien über den Erfolg von Leberoperationen ablesen, vergleiche [1, 76, 77, 82].

Die auftretenden Schwierigkeiten dieses Eingriffes sind durch den Aufbau der Leber gegeben. Diesen stellen wir nun vor, er ist [18] entnommen.

1.1.1 Das Organ

Die Leber liegt innerhalb der Bauchhöhle (intraperitoneal) direkt unterhalb des Zwerchfells. Sie wiegt bei einem erwachsenen Menschen durchschnittlich 1500g. Sie stellt damit das größte parenchymatöse Organ des Menschen dar.

Die Leber ist durch das *Ligamentum coronarium hepatis* mit dem Zwerchfell und der hinteren Bauchwand verbunden. Der *Sulcus sinister* ist Ansatz für das *Ligamentum falciforme hepatis* und das *Ligamentum teres hepatis*. Weiterhin trennt der Sulcus sinister die Leber in einen rechten und linken *Leberlappen*. Diese Ansatzpunkte bilden zusammen mit den Eintritts- und Austrittspunkten der (weiter unten) beschriebenen Gefäßsysteme die einzigen Oberflächenmerkmale der Leber.

Die Leber wird von vier unterschiedlichen Gefäßsystemen durchzogen:

- Leberarterien. Hauptarterie ist die *Arteria hepatica propria*. Über dieses System wird die Leber mit Sauerstoff versorgt.
- Pfortadern. Eintrittsvene ist die *Vena portae*. Über dieses System gelangt das mit Nährstoffen angereicherte Blut aus dem Darm in die Leber.
- Lebervenen. Austrittsvene ist die *Vena cava inferior*. Über dieses System wird das Blut zurück in den großen Kreislauf geführt.
- Gallengänge. Die Gallenblase ist die einzig gut sichtbare Struktur in der Leber.

Allen vier Systemen ist gemein, dass sie jeweils nur einen Eintritts- oder Austrittspunkt in die Leber haben. Sie bilden innerhalb der Leber jeweils eine Baumstruktur. In ihrem grobem Aufbau folgen sie einem anatomischen System, sie sind jedoch in ihrem Feinaufbau individuell verschieden.

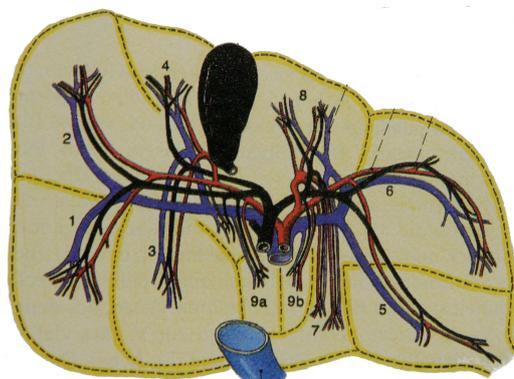


Abbildung 1.1: Gliederung der Leber in Segmente; *schwarz*: Gallengänge; *rot*: Leberarterien; *blau*: Pfortadern, die zusammen das Leber-Trias bilden und die Gliederung der Leber definieren; Grafik entnommen aus [18]

Das Volumen der Leber wird in neun *Segmente nach Couinaud* gegliedert. Diese Gliederung folgt der sogenannten *Trias* aus Leberarterien, Pfortadern und Gallengängen. Diese verlaufen parallel und verzweigen sich in 9 Unterbereiche, vergleiche Abbildung 1.1. Die Segmentgrenzen finden sich in keiner anatomischen Struktur auf der Leberoberfläche wieder.

Die Leber zeichnet sich durch ihre Regenerationsfähigkeit aus. Es können bei Erhaltung des Versorgungssystems bis zu 80% einer gesunden Leber reseziert werden [1]. Der verbleibende Organrest übernimmt die Funktion des Organs und kann sich innerhalb eines Jahres auf bis zu 75% des ursprünglichen Volumens regenerieren. Diese Fähigkeit stellt eine Grundlage für die Tumorsektion dar. Voraussetzung ist, dass die versorgenden Systeme dieses Abschnittes erhalten bleiben [42].

1.1.2 Operationsarten

Zur Resektion von Lebertumoren werden zur Zeit drei verschiedene Operationsarten genutzt.

- Bei der **offenen Leberchirurgie** wird die Bauchdecke durch einen großen Schnitt geöffnet. Diese Art des Zuganges erlaubt die Verwendung großer Werkzeuge und ermöglicht dem Chirurgen eine haptische Kontrolle des Organs und damit das Ertasten der Lage von Tumoren.
- Die **laparoskopische Leberchirurgie** nutzt drei bis vier kleine Zugänge durch die Bauchdecke mittels sogenannter Trokare zur Operation. Neben den Operationsinstrumenten werden auch Licht und Kamera zur Orientierung benötigt. Der nötige Platz zum Arbeiten im Bauchraum wird

durch Aufblasen der Bauchhöhle mittels CO₂-Gas geschaffen. Bei dieser Operationsart steht dem Chirurgen nicht die Möglichkeit der Ertastung zur Verfügung. Die Vorteile dieser Operationsmethode gegenüber der offenen Chirurgie und ein detaillierter Vergleich finden sich in [13, 14, 78].

- Die **Radiofrequenzablation** RFA ist keine Operationsmethode im klassischen Sinne. Unter der Kontrolle von Ultraschall und CT wird dem Patienten durch die Haut eine Nadel in den Tumor geführt, über deren Spitze dann das Gewebe mittels elektromagnetischer Wellen zerstört wird. Diese Eingriffsart kann nur für kleine Metastasen eingesetzt werden [48].

1.2 Der FUSION-Ansatz

In den beschriebenen Eigenschaften der Leber liegen die Schwierigkeiten ihrer Operabilität begründet:

Das Gewebe wird während des chirurgischen Eingriffes mobilisiert, d.h. ein Teil der Bänder, die das Organ in der Bauchhöhle befestigen, werden durchtrennt, um an die zu operierende Struktur zu gelangen und sie darzustellen. Hierbei und während der weiteren Operationsschritte wird die Leber deformiert, da keine festen Strukturen, wie etwa Knochen, sie in ihrer Form halten.

Durch die wenigen Oberflächenmerkmale und den nicht gegebenen Zusammenhang zwischen diesen und der Leberstruktur in der Tiefe fallen Orientierungsmöglichkeiten für den Operateur weg.

Bei der laparoskopischen Operation, die zur Minderung des Patiententraumas - dem dritten Faktor einer erfolgreichen Therapie - eingesetzt werden kann, fehlt dem Arzt zusätzlich auch noch die taktile Wahrnehmung, mit der bei offenem Zugang die Tumorposition ertastet werden kann.

Durch die Baumstruktur der versorgenden Gefäße ergeben sich weitere Schwierigkeiten. Wird während der Operation ein größeres Gefäß zertrennt, geschieht dies irreversibel. Damit verlieren die Bereiche der Leber, die durch dieses Gefäß versorgt werden, ihre Funktionalität und sterben ab. Der Operateur muss daher vor dem Durchtrennen eines Gefäßes entscheiden, ob das Gefäß einen Bereich versorgt, der sowieso im Verlauf der Tumorsektion entfernt werden muss oder ob das Gefäß andere Bereiche versorgt und er mit einer Trennung das funktionale Restvolumen der Leber verringert.

Diese Fragestellung wird umso kritischer, wenn die zu entfernenden Tumore nahe der großen Gefäße der Trias liegen und eine Resektion mit Sicherheitsabstand bedeuten würde, die Gefäße zu durchtrennen.

Um diesen Schwierigkeiten zu begegnen und um dem Arzt eine Orientierung im Operationsgebiet zu geben, werden medizinische Bildgebungsverfahren ein-

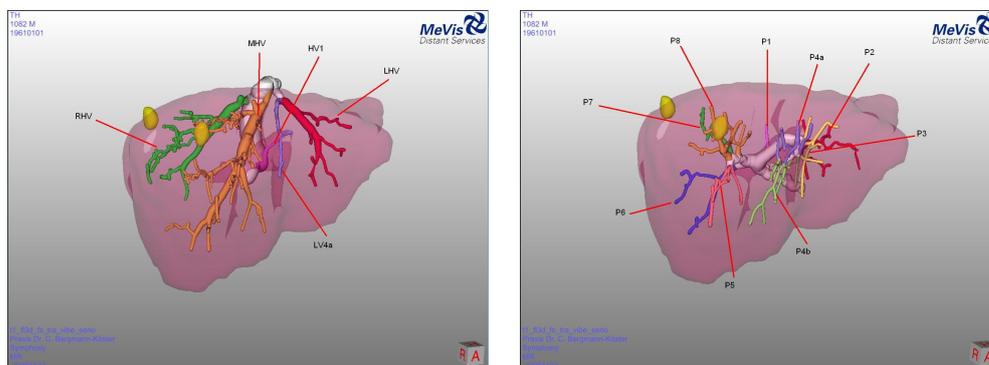


Abbildung 1.2: Darstellung der Planungsdaten einer menschlichen Leber; *links*: Lebervenen; *rechts*: Portalvenen; die Farben korrespondieren zu den einzelnen Lebersegmenten

gesetzt. Dies erfolgt in zwei Schritten:

Vor der Operation wird vom Patienten eine Thorax-Abdomen CT-Aufnahme erstellt. Die Computertomographie erlaubt dabei eine feine Auflösung innerhalb des Zielgebietes der späteren Operation. Durch die Gabe von Kontrastmitteln gelingt es, große Teile der Gefäßsysteme deutlich darzustellen.

Aus diesen Daten werden die Lage der Gefäßstrukturen der Leber und die der Tumore segmentiert. Dies erlaubt ein individuelles Bestimmen der Lebersegmente [75]. Zum anderen kann die Segmentierung genutzt werden, um eine genaue Operationsplanung zu entwickeln. Es ist möglich, Schnittführungen zu bestimmen, die bei geeigneter Lage der Tumore die größeren Gefäßstrukturen umgehen. Ziel ist eine segmentgerechte Resektion der Tumore [43].

In Abbildung 1.2 ist ein mögliches Ergebnis einer solchen Planung visualisiert. Sie ist von der Firma *MeVis Distance Service GmbH, Bremen* erstellt worden. Sie wird dem Arzt als Menge von Einzelbildern und Filmen, die den räumlichen Eindruck vermitteln sollen, präsentiert. Die unterschiedlichen Färbungen der Gefäße korrespondieren mit ihrer Segmentzugehörigkeit. Die zu resezierenden Tumore sind als gelbe Volumina dargestellt.

Das zweite bildgebende Verfahren kommt während der Operation zum Einsatz. Es dient der Orientierung und Navigation der Instrumente durch den Arzt beim Eingriff. Hierzu wird Ultraschall verwendet. Dies Verfahren liefert sehr schnell Bilddaten und ist nicht invasiv. Anders als intra-operatives CT oder MR ist seine Anwendung in jedem Operationssaal ohne bauliche Eingriffe möglich. Zwar wird die Geschwindigkeit des Ultraschall auf Kosten der Bildqualität erkaufte, doch wird während der Operation nur eine Orientierung benötigt. Durch die Hilfe des Ultraschalls ist es möglich, nicht invasiv in das Zielgebiet zu schauen und es zu visualisieren.

Ziel ist es nun, vor der Operation den Eingriff genau zu planen und diese

Daten während der Operation zusammen mit dem Ultraschall zur Navigation zu nutzen. Ein Teilziel ist hierbei, die Ultraschallbilder, die die aktuelle Lage der Leber darstellen, mit den Informationen der Planung zu überlagern.

Eine einfache Überlagerung der Planungsdaten mit den Ultraschalldaten ist nicht möglich, da die Leber während der Operation deformiert und mobilisiert wird.

Bisher muss sich der Chirurg an den Ultraschalldaten orientieren und dann gedanklich das dargestellte Volumen in den Planungsdaten finden. Er muss dazu einen Abgleich der Deformation machen und die dann in der Planung vorhandenen Informationen, wie die Schnittführung, auf das Ultraschallvolumen zurück übertragen. Diese kann er dann für seine Entscheidungen berücksichtigen.

Eine Vereinfachung und Verbesserung dieser Arbeitsschritte stellt die rechnergestützte Fusion dieser Daten dar. Dazu wird die Deformation der aktuellen Leber bestimmt und zurück gerechnet. Danach befinden sich anatomisch korrespondierende Strukturen in den Planungsdaten und auch in den Navigationsdaten an gleicher Position in den jeweiligen Bilddaten. Dieser Vorgang wird Bildregistrierung genannt.

Nach diesem Schritt ist es möglich, die Informationen der Planungsdaten auf die Navigationsdaten zu übertragen und dort zu nutzen, das heißt alle Informationen, wie Schnittführungen, Gefäßnamen, Tumorpositionen, der Planungsdaten in die Navigationsdaten einzuzeichnen und damit navigiert zu operieren.

Zur navigierten Chirurgie muss also an zentraler Stelle ein Registrierungsproblem, nämlich das Anpassen der Planungsdaten an die räumliche Struktur der Navigationsdaten, gelöst werden.

Dieses Problem ist bisher noch nicht zufriedenstellend gelöst worden (vergleiche den folgenden Abschnitt 1.3).

Wir werden uns in dieser Arbeit mit der Registrierung von prä-operativen Planungsdaten und intra-operativen Navigationsdaten befassen.

Dies werden wir in zwei großen Abschnitten durchführen. Wir werden zunächst einen allgemeinen Registrierungsansatz vorstellen, der auf dem Discretize-then-Optimize-Ansatz beruht (vergleiche Kapitel 2.2). Ziel dieses Abschnittes ist dessen vollständige und detaillierte Ausführung. Im zweiten Abschnitt werden wir diesen noch allgemeinen Ansatz durch die Präsentation von Verfahren und Techniken auf die Anforderungen der Registrierung von Navigations- und Planungsdaten spezialisieren.

1.3 Bisherige Arbeiten

Im Bereich der Registrierung medizinischer Daten gibt es eine Vielzahl von Veröffentlichungen, die vollständig zu nennen nicht mehr möglich ist. Wir verweisen nur auf wenige Arbeiten, die dafür in der Literatur herausragend oft zitiert werden oder in der Nähe der medizinischen Anwendung stehen.

Die vorliegende Arbeit präsentiert einen geschlossenen Ansatz zur Bildregistrierung, ähnlich dem Vorgehen in [54]. In diesem Buch liegt der Schwerpunkt eher bei der Präsentation theoretischer Ergebnisse. Es wird ein Optimize-then-Discretize-Ansatz (vergleiche Kapitel 2.2.1) verwendet. Im Gegensatz dazu nutzen wir einen Discretize-then-Optimize-Ansatz, der für die Registrierung erstmalig kurz in [29] beschrieben wird. Auch in [33] wird dieser Ansatz in einem Kapitel genutzt.

Weitere Arbeiten, die einen Überblick über die Registrierung geben und wegen ihres sukzessiven Erscheinens die Entwicklung dieses Themenbereiches innerhalb der letzten fünfzehn Jahre beleuchten, sind [10, 51, 35, 64].

Die Methode, zunächst aus tomographischen Daten eine Eingriffsplanung durchzuführen und diese auf intra-operative Ultraschalldaten zu registrieren, wird in verschiedenen Bereichen der Medizin eingesetzt:

In der Unfallchirurgie und Orthopädie [2, 88]. Die Registrierungsverfahren basieren auf einer Segmentierung und Registrierung der Knochenoberflächen. Aufgrund deren Starrheit sind nur rigide (vergleiche Kapitel 7.2) Verfahren beschrieben.

In der Neurologie und Neurochirurgie zur Kontrolle des Brainshifts bei Gehirnoperationen [66].

In der interventionellen Kardiologie zur Gewebeablation mittels Herzkatheter, wobei während der Intervention mittels transkutanem Ultraschall die Lage des Katheters mit den Planungsdaten des Herzens abgeglichen wird [80, 39]. Auch hier sind bisher nur rigide Verfahren vorgestellt worden.

Im Bereich der Leber sind, obwohl deren Deformation während der Operation nicht-rigide ist, zunächst rigide Registrierungsverfahren vorgestellt worden. Diese basieren entweder auf den Grauwerten der Daten [67, 80] oder auf Bildeigenschaften (Features) [65, 63], wobei aufgrund ihrer leichten Identifizierung in CT und US Lebergefäße als Features genutzt werden. Erweiterungen derartiger Ansätze auf nicht-rigide Algorithmen finden sich in [45, 44, 66]. Um nicht intra-operativ aus den Ultraschalldaten diese Features extrahieren zu müssen, ist in [3] ein hybrider Ansatz vorgestellt worden, der intra-operativ im Ultraschall gemessene Grauwerte auf prä-operativ bestimmte Features registriert.

In [46] beschreiben die Autoren ein Distanzmaß, das erlaubt, prä-operative Gefäßbaumstrukturen nicht-parametrisch auf Ultraschalldaten zu registrieren. Ein weiteres Verfahren, die unterschiedlichen Modalitäten der verwendeten

Bildgebung abzugleichen, wird in [86] vorgeschlagen, das aus den prä-operativen Daten ein Ultraschallbild simuliert, das dann mit Registrierungsmethoden für mono-modale Bilddaten angepasst werden kann (mono-modal heißen Bilddaten, wenn in zwei Bildgebungen gleiche Strukturen auf gleiche Grauwerte abgebildet werden).

Offensichtlich ist, dass zur Registrierung von Navigations- und Planungsdaten nicht-rigide Methoden genutzt werden müssen, um die physikalische Deformation der Leber wiederzugeben. Wir stellen in dieser Arbeit dazu erstmalig ausführlich die Verwendung von variationellen Methoden vor.

1.4 Aufbau der Arbeit und eigene Beiträge

Wir beginnen mit der Präsentation des allgemeinen Registrierungsansatzes in Teil I - Allgemeiner Registrierungsansatz - der Arbeit.

In dessen erstem Kapitel 2 modellieren wir das Registrierungsproblem als kontinuierliches Minimierungsproblem einer speziellen Zielfunktion. Wir diskutieren dafür zwei unterschiedliche Ansätze, die zur Minimierung der Zielfunktion in der Literatur vorgestellt werden. Wir werden im weiteren den Discretize-then-Optimize-Ansatz verfolgen, in dem zunächst die Zielfunktion diskretisiert und dann minimiert wird. Dieser Ansatz setzt sich aus mehreren Bausteinen zusammen, die in den folgenden Kapiteln vorgestellt werden.

Zunächst befassen wir uns in Kapitel 3 mit Methoden der numerischen Optimierung, die wir zur Minimierung der Zielfunktion einsetzen werden. Diese Methoden stellen den Kern später zu beschreibener Registrierungsalgorithmen dar.

Zur Diskretisierung der Zielfunktion und der gesuchten Deformation werden wir Gitter benötigen; deren Aufbau und Numerik ist Inhalt von Kapitel 4.

Notwendige Eigenschaft von Bilddaten bei der Registrierung ist, dass sie an beliebigen Stellen auszuwerten sind; hierzu stellen wir bekannte Methoden und eine neue - die linear-glatte Interpolation - in Kapitel 5 vor.

Im Anschluss daran (Kapitel 6) präsentieren wir drei Distanzmaße, die in den späteren Anwendungen genutzt werden. Kapitel 7 befasst sich mit parametrischen Bildregistrierungsmethoden, die wir zur Vorregistrierung von Bilddaten verwenden werden. Darauf beschreiben wir nicht-parametrische Methoden (Kapitel 8). Diese sind, genau wie die zuvor beschriebenen Distanzmaße, zwar in der Literatur bekannt, ihre Diskretisierung und Numerik wird hier aber einheitlich in einen allgemeinen Ansatz eingebettet.

Dieser Abschnitt endet mit der Beschreibung von Multiresolutionsverfahren in Kapitel 9. Zusätzlich klären wir, wie die durch Diskretisierung auftretenden linearen Gleichungssysteme effizient und schnell gelöst werden.

Damit beschreiben wir in dieser Arbeit erstmalig vollständig die Registrierung mittels des Discretize-then-Optimize-Ansatzes. Wir präsentieren einen einheitlichen Rahmen beginnend bei Optimierungsmethoden, über Interpolations- und Diskretisierungstechniken sowie Distanzmaßen und Regularisierern, bis hin zu numerischen Lösungstechniken für die auftretenden linearen Gleichungssysteme.

Der folgende Abschnitt II befasst sich mit der Präsentation von Techniken und Verfahren, die den allgemeinen Ansatz in Richtung der späteren Anwendung spezialisieren.

In Kapitel 10 nutzen wir erstmals für die Registrierung Multiskalenverfahren, um der speziellen Röhrenstruktur der Gefäße gerecht zu werden.

Danach beschreiben wir eine neue Multiresolutionsstrategie in Kapitel 11, die unter der Voraussetzung, dass dem Anwender nur an der Registrierung eines Bildausschnittes gelegen ist, gegenüber herkömmlichen Ansätzen eine deutliche Beschleunigung ermöglicht und es somit erstmalig erlaubt, nicht-lineare Verfahren in zeitkritischen Anwendungen einzusetzen.

In den letzten beiden Kapiteln dieses Abschnittes befassen wir uns damit, wie wir Zusatzinformationen in Form von Landmarken in den Registrierungsprozess einbinden können. Dazu beschreiben wir in Kapitel 12 die neue Problemstellung und nur auf Landmarken basierende Verfahren. Im darauffolgenden Kapitel 13 präsentieren wir schließlich neue Methoden, die den allgemeinen Ansatz mit den Landmarken-Verfahren kombinieren.

Im letzten Abschnitt III zeigen wir die Anwendung der vorgestellten Methoden auf reale Daten aus der Leberchirurgie und präsentieren Ergebnisse. Wir stellen das FUSION-Projekt vor, in das diese Arbeit eingebettet ist und erläutern dann zwei Anwendungsfälle, in der die Registrierung von Navigations- und Planungsdaten genutzt wird.

Mit einem Ausblick auf weiterführende Forschungsaufgaben in der Registrierung von CT- und Ultraschallaufnahmen von Weichgewebe endet diese Arbeit.

Teil I

Allgemeiner Registrierungsansatz

Kapitel 2

Das Registrierungsproblem

In diesem Kapitel werden wir das Registrierungsproblem beschreiben und es als Minimierungsproblem formulieren. Zu dessen Lösung werden wir zwei unterschiedliche Wege vorstellen und diskutieren.

2.1 Kontinuierliche Formulierung

Beginnen wir damit, dass wir das Problem der Bildregistrierung knapp formulieren:

Gegeben sind zwei Bilder, ein Referenzbild \mathcal{R} und ein Templatebild \mathcal{T} .

Gesucht ist eine *plausible* Transformation y , die, angewendet auf \mathcal{T} , dieses *ähnlich* zu \mathcal{R} *transformiert* und eventuellen *Nebenbedingungen* genügt.

Um diesen Satz mit Inhalt zu füllen, müssen wir beantworten, wie seine einzelnen Bausteine zu verstehen sind:

1. Was ist ein Bild und wie wird es transformiert?
2. Wie wird die Transformation gesucht?
3. Was bedeutet Plausibilität?
4. Was bedeutet Ähnlichkeit?
5. Wie gehen Nebenbedingungen ein?

Die Antworten auf diese Fragen werden bedingt durch zusätzliche Anforderungen, die auf die Praktikabilität und Flexibilität der späteren Algorithmen abzielen.

Die Beantwortung dieser Fragen wird sich über die nächsten Kapitel erstrecken. An dieser Stelle werden wir durch knappe Skizzierungen dieser Antworten einen Überblick über die kommenden Kapitel geben.

Zu Frage 1: Als Bilder betrachten wir in dieser Arbeit kontinuierliche Funktionen vom \mathbb{R}^d in die reellen Zahlen. Ihre Kontinuität werden wir durch Interpolation der gegebenen Bilddaten erzielen (vergleiche Kapitel 5).

Zu Frage 2: Die gesuchte Deformation (oder Transformation) y werden wir durch Lösung eines kontinuierlichen Minimierungsproblems bestimmen. Hierzu dienen Methoden der numerischen Optimierung (vergleiche Kapitel 3).

Zu Frage 3: Unter Ähnlichkeit zweier Bilder verstehen wir, dass die Positionen in beiden Bildern, die korrespondierende Informationen enthalten, räumlich nahe beieinander liegen. Zur Bestimmung der Ähnlichkeit werden wir Distanzmaße (oder Ähnlichkeitsmaße) \mathcal{D} formulieren. Darunter verstehen wir Abbildungen, die zwei gegebenen Bildern eine Zahl zuordnen (vergleiche Kapitel 6). Da die Distanz der Bilder von der das Templatebild deformierenden Transformation abhängt, gilt $\mathcal{D} = \mathcal{D}(y)$.

Zu Frage 4: Der Begriff der Plausibilität ist noch schwerer zu fassen als der der Ähnlichkeit. Wir definieren negativ. Eine Transformation ist nicht plausibel, wenn sie unserer natürlichen Vorstellung der zugrunde liegenden Deformation zuwiderläuft. Beispielsweise ist eine Transformation, die bei der Registrierung zweier Gesichtsbilder das linke mit dem rechten Auge vertauscht, nicht plausibel. Wir werden die Plausibilität einer Transformation durch einen Regularisierer $\mathcal{S} = \mathcal{S}(y)$ messen (Kapitel 8) oder die Transformation explizit als Linearkombination gegebener Basisfunktionen beschreiben (Kapitel 7).

Zu Frage 5: Nebenbedingungen sind zusätzliche, die Ähnlichkeit der Bilder und die Plausibilität der Transformation erweiternde Forderungen an die Deformation. Ein Beispiel, das wir in Kapitel 13 erläutern, ist die Erfüllung von Landmarkenbedingungen. Hier wird gefordert, dass die Transformation vorher in beiden Bildern bestimmte Punktpaare aufeinander abbildet, die Ähnlichkeits- und Plausibilitätseigenschaften aber auch erfüllt. Nebenbedingungen fließen als zusätzliche Maße (soft constraints) $\mathcal{C}^{\text{soft}}$ oder als explizit formulierte Bedingungen (hard constraints) $\mathcal{C}^{\text{hard}}$ in das Minimierungsproblem ein.

Wir erhalten damit

$$\begin{aligned} \mathcal{J}(y) &= \mathcal{D}(y) + \alpha \mathcal{S}(y) + \beta \mathcal{C}^{\text{soft}}(y) \rightarrow \min \\ \text{u.d.N. } \mathcal{C}^{\text{hard}}(y) &= 0, \end{aligned} \tag{2.1}$$

(u.d.N. heißt *unter den Nebenbedingungen*). Eine solche Aufteilung des Problems in Maße für Ähnlichkeit, Plausibilität und Nebenbedingungen bietet den Vorteil, dass wir einen allgemeinen Rahmen für die Registrierung erhalten. Durch Kombination verschiedener Maße lassen sie Spezialisierungen des allgemeinen Problems für beliebige Anwendungen ableiten. Wir erhalten einen

flexiblen Rahmen, der Grundlage zur Lösung verschiedenster Registrierungsprobleme ist. Dieser Rahmen ist in [25] zum ersten Mal beschrieben worden und geht ein in [62].

Zu klären bleibt, warum wir, obwohl immer diskrete Bilddaten als Eingabe für die Registrierung vorliegen, das Registrierungsproblem kontinuierlich formulieren. Eine diskrete Formulierung basierend auf den originären Eingabedaten scheint naiv gesehen das Problem besser zu beschreiben.

Wir geben folgende Antworten:

Die physikalischen Auflösungsmöglichkeiten medizinischer Bildgebungsgeräte werden seit Jahren feiner, die Ortsauflösung fällt. Stärke der kontinuierlichen Formulierung ist, dass sie die Problemstellung unabhängig von der gegebenen Voxelauflösung wiedergibt und unabhängig von der Entwicklung ist.

Mit dem Argument in Zusammenhang steht die Möglichkeit, aus der kontinuierlichen Formulierung direkt eine Multiresolutionsstrategie für Bildregistrierungsalgorithmen abzuleiten (Kapitel 9). Wir sind in der Lage, aus der kontinuierlichen Formulierung durch Wahl immer feiner werdener Diskretisierungen eine auf das kontinuierliche Problem zustrebende Folge von diskreten Problemen zu formulieren und zu lösen.

Als dritten Vorteil können wir anführen, dass aus der kontinuierlichen Theorie allgemeine Aussagen über die Problemstellung, wie Existenz und Eindeutigkeit von Lösungen, möglich sind.

2.2 Genereller Ansatz

Wir haben in (2.1) ein kontinuierliches Optimierungsproblem formuliert, dessen Minimierer wir als Lösung des Registrierungsproblems ansehen. Diesen Minimierer gilt es zu bestimmen. Hierzu sind in der Literatur [54] zwei generelle Ansätze unterschiedlichen Charakters vorgestellt worden. Wir bezeichnen sie mit ihren englischen Namen. Sie heißen *Optimize-then-Discretize* und *Discretize-then-Optimize*. Wir werden in dieser Arbeit den Letzteren verfolgen und dies durch Beschreibung und Diskussion beider Ansätze begründen.

2.2.1 Optimize-then-Discretize-Ansatz

Unter diesem Namen ist in der Literatur das folgende Vorgehen bekannt: Für das kontinuierliche Problem werden durch Methoden der Variationsrechnung (Gateaux-Ableitung, Greensche Formel, Gaußscher Integralsatz) ebenfalls kontinuierliche Bedingungen erster Ordnung für ein Minimum von (2.1) hergeleitet. Diese lassen sich als ein System partieller Differentialgleichungen darstellen, den sogenannten *Euler-Lagrange-Gleichungen*, die punktweise eine Bedingung

an den gesuchten Minimierer stellen. Erster Ordnung sind die Bedingungen, da sie auf den ersten Ableitungen des Funktionals \mathcal{J} basieren und damit eine notwendige (aber nicht hinreichende) Bedingung an den Minimierer stellen.

Ist - wie im Fall der später zu spezialisierenden Distanzmaße - das Funktional \mathcal{J} nicht linear in y , so ist auch das System partieller Differentialgleichungen nicht linear in der gesuchten Lösung.

Das System wird mittels finiter Elemente oder finiter Differenzen diskretisiert, die Nicht-Linearität durch Iterationstechniken wie Fixpunktiteration oder Zeitschrittverfahren aufgelöst. Eine sehr detaillierte Beschreibung dieses Ansatzes findet sich in [54], die für eine Vielzahl an Maßen diesen Weg geht.

Betrachten wir die Vor- und Nachteile dieses Ansatzes:

Vorteile

- Die Lösungstechniken aus dem Bereich der partiellen Differentialgleichungen können für die Registrierung genutzt werden.
- Es existieren Techniken zur schnellen Lösung der auftretenden linearen Gleichungssysteme.
- Durch geschickte Formulierungen lassen sich die auftretenden Nicht-Linearitäten des Systems in der rechten Seite elegant „verstecken“.
- Die Lösung wird in kleinen, kontrollierbaren Schritten bestimmt.

Nachteile

- Das hinter dem System partieller Differentialgleichungen liegende Optimierungsproblem wird nicht berücksichtigt.
- Die Nicht-Linearitäten gehen nur in die rechte Seite des Systems ein.
- Durch kurze Schritte bei der Bestimmung der Lösung kann kein schnelles Verfahren erreicht werden.

2.2.2 Discretize-then-Optimize-Ansatz

Dieser Ansatz vertauscht die Schritte. Hier wird zunächst das kontinuierliche Problem diskretisiert, so dass ein diskretes, endliches Minimierungsproblem hergeleitet wird. Dieses Problem wird mit Methoden der numerischen Optimierung mittels Newton-artiger Methoden gelöst. Da in dieser Arbeit dieser Ansatz gewählt wird, beinhalten die nachfolgenden Kapitel eine detaillierte

Schilderung der Diskretisierung und der sich daran anschließenden Optimierung.

Der Ansatz wurde im Bereich der Bildregistrierung bei [29] skizziert. Vor- und Nachteile sind:

Vorteile

- Zur Lösung des Minimierungsproblems werden Newton-artige Verfahren verwendet, die im Falle der Konvergenz quadratisch konvergieren.
- Abbruchkriterien lassen sich durch die Theorie der numerischen Optimierung sauber herleiten.
- Die Nicht-Linearität des Problems geht sowohl in die rechte Seite als auch in die Systemmatrix ein.
- Das der Registrierung zugrunde liegende Minimierungsproblem wird direkt gelöst.
- Durch die Theorie der restringierten Optimierung ist geklärt, wie Nebenbedingungen in die Minimierung einfließen.
- Die Verwendung von Multiresolutionsansätzen lässt sich leicht über die Wahl einer feineren (bzw. gröberen) Diskretisierung des Minimierungsproblems erklären und verwirklichen.

Nachteile

- Damit die auftretenden linearen Gleichungssysteme effizient gelöst werden können, sind Bedingungen an die Diskretisierung zu stellen (vergleiche Kapitel 8.5).

Wir wählen - wie oben gesagt - diesen Weg.

Fazit

Wir haben das Registrierungsproblem als (bedingtes) Minimierungsproblem formuliert. Die Zielfunktion ist dabei aus Bausteinen zusammengesetzt. Zur Lösung des Problems haben wir zwei Wege beschrieben und uns nach Diskussion deren Vor- und Nachteile für den Discretize-then-Optimize-Ansatz entschieden. Durch die im Kapitel gestellten fünf Fragen ist der Inhalt der folgenden Kapitel vorgegeben. Wir beginnen mit der Beschreibung von Optimierungsmethoden, die wir zur Registrierung nutzen werden.

Kapitel 3

Optimierung

Wir haben das Registrierungsproblem als ein Minimierungsproblem formuliert, dessen Minimierer die gesuchte Transformation ist. Daher werden wir in diesem Kapitel - zunächst unabhängig von der Registrierungsthematik - Begrifflichkeiten und Methoden der numerischen Optimierung vorstellen. Die numerische Optimierung verstehen wir als Werkzeug nicht als Forschungsgegenstand dieser Arbeit. Unser Schwerpunkt wird nicht auf theoretischen Aussagen liegen sondern so gewählt sein, dass wir am Ende des Kapitels die nötigen Methoden der Optimierung nutzen können.

Wir folgen in dieser Arbeit dem Buch *Numerical Optimization* von Nocedal und Wright [56], die in sehr ausführlicher Form einen weiten und detaillierten Blick in das Feld der numerischen Optimierung geben. Diesem kann der geneigte Leser weiterführende Aussagen und Methoden entnehmen.

Das Minimierungsproblem ohne Nebenbedingungen hat die allgemeine Form: Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $n \geq 1$ eine glatte Funktion. Gesucht ist ein $x_\star \in \mathbb{R}^n$ mit

$$x_\star = \operatorname{argmin} f(x) \quad (3.1)$$

oder in einer anderen Schreibweise

$$f(x) \xrightarrow{x} \min. \quad (3.2)$$

Die verwendete Notation ist in Tabelle 3.1 aufgelistet. Iterations- und andere Indizes werden in diesem Kapitel immer unten rechts an die entsprechenden Variablen geschrieben. In späteren Kapiteln ist dies nicht mehr so, um eine klare Darstellung zu erhalten. Welcher Index die Iteration angibt, wird aber aus dem Kontext klar werden.

Beim Lösen des Problems ist zu berücksichtigen, dass wir nur die Funktionswerte und eventuelle Ableitungen der Funktion an einer Reihe einzelner Punkte x_0, x_1, \dots kennen. Des Weiteren ist die (numerische) Auswertung der Funktion im allgemeinen zeitintensiv und soll, um numerisch schnelle Algorithmen zu erhalten, nicht unnötig erfolgen.

Tabelle 3.1: Notation, die im Kapitel über Optimierung genutzt wird; wir nutzen den Nabla-Operator, um stehende und liegende Ableitung zu unterscheiden

Funktion	f	$:\mathbb{R}^n \rightarrow \mathbb{R}$	
Ableitung	df	$:\mathbb{R}^n \rightarrow \mathbb{R}^n$	$df(x) = (\partial_1 f(x), \dots, \partial_n f(x))$
	∇f	$:\mathbb{R}^n \rightarrow \mathbb{R}^n$	$\nabla f = df^\top$
zweite Ableitung	$d_2 f$	$:\mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$	$(d_2 f(x))_{ij} = \partial_i \partial_j f(x)$
Hesse-Matrix	$\nabla^2 f$	$:\mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$	$\nabla^2 f = d_2 f$
mehrdim. Funktion	g	$:\mathbb{R}^n \rightarrow \mathbb{R}^m$	
Jacobi-Matrix	dg	$:\mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$	$dg = (\nabla g_1, \dots, \nabla g_m)$

In der Literatur werden globale und lokale Minimierer unterschieden. Es gilt $x_\star \in \mathbb{R}^n$ ist *globaler Minimierer* der Funktion f , falls $f(x_\star) \leq f(x)$ für alle x aus dem Urbildbereich von f .

Ein Punkt $x_\star \in \mathbb{R}^n$ ist *lokaler Minimierer*, falls es eine Umgebung \mathcal{N} um x_\star gibt, mit $f(x_\star) \leq f(x)$ für alle $x \in \mathcal{N}$.

Da wir, wie oben erwähnt, die Funktion f nur an ausgewählten Punkten kennen, können wir nicht davon ausgehen, ein globales Minimum zu finden. Die hier vorzustellenden Algorithmen sind in der Lage, lokale Minima zu bestimmen.

Allen hier vorgestellten Optimierungsalgorithmen ist gemein, dass sie ausgehend von einem von außen vorgegebenen Startwert $x_0 \in \mathbb{R}^n$ eine Iterationsfolge $x_0, x_1, x_2, \dots, x_k \in \mathbb{R}^n$ generieren, derart, dass die Funktionswerte monoton fallen $f(x_k) \geq f(x_{k+1})$. Zur Bestimmung des Vektors x_{k+1} werden die Funktionswerte und Ableitungen der Iteration k und eventuell (aller) vorherigen Iterationen genutzt. Die Iteration terminiert nach später festzulegenden Abbruchkriterien.

Für den Iterationsschritt $k \mapsto k + 1$ gibt es zwei Techniken, Line-Search- und Trust-Region-Verfahren. Wir nutzen in dieser Arbeit nur Line-Search-Verfahren. Die Klasse dieser Verfahren ist dadurch ausgezeichnet, dass zu jeder Iteration k eine Suchrichtung $p_k \in \mathbb{R}^n$ und eine Schrittweite $\alpha \in \mathbb{R}$ bestimmt wird, so dass

$$x_{k+1} = x_k + \alpha p_k$$

gilt. Die Verfahren dieser Klasse unterscheiden sich in der Wahl der Suchrichtung und der Schrittweitenbestimmung.

3.1 Schrittweitenbestimmung

Wir klären zunächst, wie wir in dieser Arbeit die Schrittweite α bestimmen. Ziel der Schrittweitenbestimmung ist es, bei gegebener Suchrichtung das nun in α eindimensionale Minimierungsproblem $\alpha^* = \operatorname{argmin} f(x_k + \alpha p_k)$ effizient zu lösen. Auf eine exakte Lösung wird aus Kostengründen verzichtet, es werden aber Bedingungen an die neue Iterierte gestellt, die die Schrittweite α erfüllen muss.

Wir nutzen die aus der Literatur bekannte Wolfe-Bedingung, die eine hinreichende Bedingung an die Schrittweite stellt. Sie ist durch

$$f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f(x_k)^\top p_k \quad (3.3)$$

und $c \in (0, 1)$ gegeben. Zu interpretieren ist diese Bedingung an α , dass die Verminderung der Funktionswerte proportional zur Schrittweite α und zur Richtungsableitung $\nabla f(x_k)^\top p_k$ ist.

In der Literatur sind weitere Bedingungen bekannt. Wir belassen es für diese Arbeit aber bei der Wolfe-Bedingung und klären nun, wie bei gegebenen Suchrichtung p_k die Schrittweite so gewählt werden kann, dass die Bedingung (3.3) eingehalten wird.

Voraussetzung an p_k ist, dass es eine Abstiegsrichtung ist, das heißt

$$\nabla f(x_k)^\top p_k < 0,$$

denn damit und

$$f(x_k + \epsilon p_k) = f(x_k) + \epsilon \nabla f(x_k)^\top p_k + \mathcal{O}(\epsilon^2)$$

gilt

$$f(x_k + \epsilon p_k) \leq f(x_k)$$

für hinreichend kleines $\epsilon > 0$.

Wir nutzen dazu nun einen einfachen Algorithmus, vgl. [56]. Dieser heißt Armijo-Liniensuche. Er ist in Algorithmus 1 dargestellt. Wir stellen ihn in der später genutzten Version vor. Zur Bestimmung der Schrittweite in jeder Iteration sei $\alpha^{LS} \in \mathbb{R}$, $p_k \in \mathbb{R}^n$ und $\mu \in (0, 1)$ gegeben. Es wird getestet, ob α^{LS} die Armijo-Bedingung erfüllt. Tritt dies ein, so wird α^{LS} als Schrittweite für diesen Iterationsschritt gewählt. Im anderen Fall wird der Wert von α^{LS} halbiert und der Ablauf wiederholt. Aus praktischer Sicht ist empfohlen, diese Iteration auf wenige Durchläufe zu beschränken, da beispielsweise nach zehn Durchläufen der Unterschied zwischen x_k und x_{k+1} minimal wird.

Zu bemerken ist, dass dieser Algorithmus nur den Wert $f(x_k + \alpha p_k)$ in jeder Iteration neu bestimmen muss. Die Größen $f(x_k)$ und $\nabla f(x_k)^\top p_k$ sind vorab bekannt und während der Iterationen konstant.

Aufgrund der Wahl von p_k als Abstiegsrichtung konvergiert der Algorithmus.

Algorithmus 1 Linesearch-Verfahren; LS

Eingabe: f, x_k, p_k **Parameter:** $\alpha^{LS}, \mu \in (0, 1)$ **Ausgabe:** x_{k+1} 1: **while** $f(x_k + \alpha^{LS} p_k) \geq f(x_k) + \mu \alpha^{LS} \nabla f(x_k)^\top p_k$ 2: $\alpha^{LS} \leftarrow \alpha^{LS} / 2$ 3: **end**

3.2 Suchrichtung

Beantworten wir nun die Frage, wie die Suchrichtung p_k in jedem Schritt des Iterationsverfahrens $x_{k+1} = x_k + \alpha p_k$ bestimmt werden kann. Wir stellen dazu drei Verfahren vor.

3.2.1 Gradientenabstiegsverfahren

Aus der Taylorreihenentwicklung $f(x_k + \alpha p_k) = f(x_k) + \alpha \nabla f(x_k)^\top p_k + \mathcal{O}(\alpha^2)$ ergibt sich die Suchrichtung $p_k = -\nabla f(x_k)$. Entlang der Gradientenrichtung am Punkt x_k variiert die Funktion am stärksten, daher wird das Verfahren auch Verfahren des steilsten Abstieges genannt. Vorteil des Verfahrens ist, dass sich die Suchrichtung direkt aus der ersten Ableitung der Funktion f herleiten lässt. Ableitungen höherer Ordnung müssen nicht bestimmt werden. Nachteil des Verfahrens ist, dass es im Allgemeinen langsam konvergiert.

3.2.2 Newtonverfahren zur Optimierung

Für eine zweimal differenzierbare Funktion bestimmt das Newtonverfahren die Suchrichtung mit Hilfe der zweiten Ableitung. Der Funktionswert $f(x_k + p_k)$ wird mit Hilfe der Taylorreihenentwicklung durch ein quadratisches Modell approximiert, $f(x_k + p_k) \approx f(x_k) + \nabla f(x_k)^\top p_k + p_k^\top \nabla^2 f(x_k) p_k =: m_k(x_k; p_k)$. Da in einer Umgebung eines Minimierers bei einer zweifach differenzierbaren Funktion die Hessematrix $\nabla^2 f$ positiv semidefinit ist, können wir annehmen, dass eine Suchrichtung durch die Minimierung von m_k gegeben ist. Durch Nullsetzen der Ableitung von m_k erhalten wir

$$p_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k). \quad (3.4)$$

Die Newtonrichtung ist eine Abstiegsrichtung unter der Annahme, dass $\nabla^2 f$ positiv definit ist. Es gilt dann

$$\nabla f(x_k)^\top p_k = -p_k^\top \nabla^2 f(x_k) p_k \approx -c \|p_k\|, \quad c > 0.$$

Der Vorteil des Verfahrens liegt in seiner Konvergenzeigenschaft. Das Verfahren ist lokal quadratisch konvergent. Nachteil des Verfahrens ist die Nutzung der Inversen der Hessematrix von f , das lineare System (3.4) muss gelöst werden.

3.2.3 Gauß-Newton Verfahren

Das Gauß-Newton-Verfahren stellt eine Modifizierung des Newton-Verfahrens für spezielle Zielfunktionen dar, bei dem die Inverse der Hessematrix numerisch einfach approximiert wird. Das Verfahren ist in [56] für Funktionen der Klasse $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f(x) = \|r(x)\|^2$, $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $n, m \in \mathbb{N}$ beschrieben. Wir stellen in dieser Arbeit eine allgemeinere Formulierung vor, die später auf die Zielfunktionen aus dem Kapitel 6 über Distanzmaße passt. Sei im folgenden

$$f : \mathbb{R}^n \rightarrow \mathbb{R}, \quad x \mapsto f(x) = \varphi(r(x))$$

und $\varphi : \mathbb{R}^m \rightarrow \mathbb{R}, \quad r : \mathbb{R}^n \rightarrow \mathbb{R}^m.$

Die Funktion r nennen wir Residuumsfunktion. Setzen wir φ als $\varphi(x) = \|r(x)\|^2$ erhalten wir die Formulierung aus dem Buch [56]. Grundlegende Idee des Ansatzes ist es, vor der Minimierung eine Linearisierung der Funktion r innerhalb von φ vorzunehmen. Mit $r = r_0 + drq$, $q \in \mathbb{R}^n$ erhalten wir ein neues Minimierungsproblem in q

$$q = \operatorname{argmin} \varphi(q) = \operatorname{argmin} \varphi(r_0 + drq)$$

Nutzen wir nun das Newtonverfahren zur Minimierung, erhalten wir die Suchrichtung

$$p_k = - (dr_k^\top d_2 \varphi_k dr_k)^{-1} d\varphi_k dr_k$$

Das Verfahren kann auch interpretiert werden als Newtonverfahren zur Minimierung von f als Funktion in x , wobei die Hessematrix approximiert wird durch Terme erster Ordnung,

$$\nabla^2 f = dr_k^\top d_2 \varphi_k dr_k + \text{Terme höherer Ordnung.}$$

Das Verfahren profitiert von der einfachen Form der jeweiligen Ableitungen von φ . Im Falle der beispielhaft vorgestellten Norm gilt $d_2 \varphi = 2$. In diesem Fall wird die Matrix $(dr^\top d_2 \varphi dr)$ nur durch die erste Ableitung der Residuumsfunktion gebildet. Hier zeigt sich der große Vorteil des Verfahrens, da billig eine Approximation der Inversen der Hessematrix $\nabla^2 f$ bestimmt werden kann.

Für den Fall, dass die äußere Funktion φ eine quadrierte Norm darstellt, kann das Verfahren wie folgt interpretiert werden: Die Residuumsfunktion r wird innerhalb der Norm um die jeweilige Iterierte x_k linearisiert, $r \rightarrow r(x_k) + dr(x_k)p_k$. Die Suchrichtung ist Lösung des Minimierungsproblems

$$p_k = \operatorname{argmin} \|r(x_k) + dr(x_k)p_k\|^2$$

und kann mit Hilfe der Normalengleichungen bestimmt werden,

$$p_k = - (dr(x_k)^\top dr(x_k))^{-1} dr(x_k)^\top r(x_k).$$

Zu diesem Fall finden sich weitere Aussagen in [56], die die Stärke des Verfahrens herausstellen, hier aber nicht weiter diskutiert werden sollen.

Wir haben nun geklärt, wie die Suchrichtung und die Schrittweite für Liniensuchverfahren bestimmt werden. Für eine vollständige Beschreibung fehlen jetzt noch Aussagen, nach welchen Kriterien wir die Iteration abbrechen.

3.3 Abbruchkriterien

Klären wir nun die Frage, unter welchen Bedingungen der Algorithmus terminiert. Diese Bedingungen sollen gewährleisten, dass die Iteration abbricht, wenn ein Minimierer gefunden worden ist, zusätzlich aber auch verhindern, dass die Iteration beliebig lange läuft, obwohl kein Minimierer gefunden werden kann.

Wir nutzen in dieser Arbeit die Kriterien, die in [27] von Gill, Murray und Wright vorgeschlagen worden sind. Sie lauten

$$(K1) \quad \|f(x_{k-1}) - f(x_k)\| < \tau(1 + \|f(x_k)\|)$$

$$(K2) \quad \|x_{k-1} - x_k\| < \sqrt{\tau}(1 + \|x_k\|)$$

$$(K3) \quad \|\nabla f(x_k)\| < \sqrt[3]{\tau}(1 + \|f(x_k)\|)$$

$$(K4) \quad \|\nabla f(x_k)\| < \epsilon$$

$$(K5) \quad k > k_{\max} .$$

Es ist $\tau \in \mathbb{R}^+$ ein vom Benutzer festgelegter Toleranzwert, $\epsilon \in \mathbb{R}^+$ die Maschinengenauigkeit und $k_{\max} \in \mathbb{N}$ eine obere Grenze für die Anzahl der Iterationen des Optimierungsalgorithmus.

Das erste Kriterium (K1) überwacht den Abstieg der Funktionswerte. Ist die Differenz zwischen zwei aufeinanderfolgenden Funktionswerten klein, wird angenommen, dass der Optimierungsalgorithmus einen Minimierer gefunden hat und weitere Iterationsschritte keine signifikante Änderung mehr ergeben werden. Aus

$$f(x_{k+1}) - f(x_k) = df(x_k)(x_{k+1} - x_k) + \mathcal{O}(\|x_{k+1} - x_k\|^2)$$

ergibt sich die Formulierung der rechten Seite des Kriteriums, da sonst für $f(x^*) \approx 0$ die Bedingung

$$\|f(x_k) - f(x_{k+1})\| \leq \tau \|f(x_k)\|$$

nicht erfüllt werden kann.

Im Kriterium (K2) wird die Änderung der Iterationswerte $\|x_{k-1} - x_k\|$ überwacht. Ist diese klein, so wird unter der Annahme, weitere Iterationen würden

keine weitere Änderung einbringen, die Iteration gestoppt. Die verwendete Formulierung folgt dem zuvor beschriebenen Kriterium.

Im Kriterium (K3) wird die Länge des Gradienten betrachtet und damit auf die notwendige Bedingung erster Ordnung für einen lokalen Minimierer Bezug genommen.

Die Kriterien (K4) und (K5) dienen als „Notbremsen“ zum Stoppen des Optimierungsalgorithmus. Sie greifen, wenn die Norm des Gradienten unterhalb der Maschinengenauigkeit liegt (K4) oder die vorgegebene Maximalanzahl von Iterationen überschritten ist (K5).

Ein Abbruch der Iteration erfolgt, wenn die Verknüpfung der Kriterien

$$((K1) \wedge (K2) \wedge (K3)) \vee (K4) \vee (K5)$$

erfüllt ist.

Damit sind die drei Fragen beantwortet und wir haben für das nicht bedingte (nicht restringierte) Optimierungsproblem $x_\star = \operatorname{argmin} f(x)$ verschiedene Werkzeuge (Gradientenabstiegsverfahren, Newtonverfahren, Gauß-Newton-Verfahren jeweils mit Armijo-Liniensuche und Abbruchkriterien nach Gill, Murray und Wright) kennengelernt.

Abschließend fassen wir das vollständige Gauß-Newton-Verfahren in Algorithmus 2 zusammen.

3.4 Bedingte Optimierung

Wir erweitern das Problem um Nebenbedingungen, mit denen wir Eigenschaften an den gesuchten Minimierer stellen können. Wir formulieren sie als Nullstellen einer Funktion $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$. (Der Funktionsname c leitet sich aus dem

Algorithmus 2 Gauß-Newton-Verfahren; GN

Eingabe: f, x_0

Ausgabe: x_\star

1: $k \leftarrow 0$

2: **while** !stopp

3: $p_k \leftarrow - (dr_k^\top d_2 \varphi_k dr_k)^{-1} d\varphi_k dr_k$

4: $\alpha^{LS} \leftarrow \text{LS}(f, x_k, p_k)$

5: $x_{k+1} \leftarrow x_k + \alpha^{LS} p_k$

6: $k \leftarrow k + 1$

7: **end**

8: $x_\star \leftarrow x_k$

englischen Begriff *constraints* = Bedingungen ab.) Das Problem (3.1), (3.2) ist nun durch

$$f(x) \rightarrow \min \quad \text{u.d.N.} \quad c(x) = 0 \quad (3.5)$$

gegeben. Neben der Forderung nach Gleichheitsbedingungen $c(x) = 0$ ist es auch möglich, Ungleichheitsbedingungen $c(x) > 0$ zu formulieren. Wir beschränken uns hier auf die Theorie für Gleichheitsbedingungen.

Analog zu einem Minimierer in der nicht bedingten Optimierung existieren notwendige Bedingungen erster Ordnung für das bedingte Problem (3.5), die *Karush-Kuhn-Tucker-Bedingungen* (*KKT-Bedingungen*) heißen. Für ein bedingtes Optimierungsproblem mit Gleichheitsnebenbedingungen lauten sie

$$\begin{aligned} \nabla_x L(x_*, \lambda_*) &= 0 \\ c_j(x_*) &= 0, \quad j = 1, \dots, m \\ \lambda_j^* c_j(x_*) &= 0 \quad j = 1, \dots, m. \end{aligned}$$

Hierbei ist

$$\begin{aligned} L(x, \lambda) &= f(x) + \sum_{i=1}^m \lambda_i c_i(x) \\ &= f(x) + \lambda^\top c(x), \end{aligned}$$

die sogenannte Lagrange-Funktion mit den Lagrange-Parametern $\lambda \in \mathbb{R}^m$.

3.4.1 Sequential Quadratic Programming

Als eine Methode der bedingten Optimierung stellen wir das SQP-Verfahren (*sequential quadratic programming*) vor. Es basiert darauf, die aktuelle Iterierte $x_k \in \mathbb{R}^n$ zu verbessern, in dem an dieser Stelle das Problem (3.5) durch ein quadratisches Problem angenähert wird. Dessen Lösung wird dann als neue Iterierte x_{k+1} genutzt. Die Annäherung durch ein quadratisches Problem kann gesehen werden als die Anwendung des Newtonverfahrens zur Nullstellenbestimmung [57, 74], um eine Lösung der ersten KKT-Bedingung zu finden.

Sei $dc_k \in \mathbb{R}^{n \times m}$ die Jacobi-Matrix der Bedingungs-Funktion c an der Stelle der Iterierten x_k ,

$$dc_k = (\nabla c_1(x_k), \dots, \nabla c_m(x_k)).$$

Die Bedingung $\nabla_x L(x_*, \lambda_*) = 0$ lässt sich nun schreiben als

$$\begin{pmatrix} \nabla f(x_*) + dc_*^\top \lambda \\ c(x_*) \end{pmatrix} = 0 \quad (3.6)$$

mit den $n + m$ Unbekannten x_*, λ_* .

Um eine Nullstelle von (3.6) zu finden, nutzen wir nun das Newtonverfahren zur Nullstellenbestimmung. Die dazugehörige Jacobi-Matrix ist durch

$$\begin{pmatrix} d_2L(x, \lambda) & dc(x)^\top \\ dc(x) & 0 \end{pmatrix}$$

gegeben. Dabei ist $d_2L(x, \lambda)$ die Hesse-Matrix bezüglich x der Lagrange-Funktion.

Das Verfahren bestimmt nun Suchrichtungen $p_k \in \mathbb{R}^n, q_k \in \mathbb{R}^m$ für x_k und λ_k mit

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} + \begin{pmatrix} p_k \\ q_k \end{pmatrix}.$$

Diese Suchrichtungen sind Lösung des *KKT-Systems*

$$\begin{pmatrix} d_2L_k & dc_k^\top \\ dc_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = \begin{pmatrix} -\nabla f_k - dc_k^\top \lambda_k \\ -c_k \end{pmatrix}. \quad (3.7)$$

Die Matrix auf der linken Seite wird *KKT-Matrix* genannt. Die Iteration ist wohl definiert, wenn die KKT-Matrix nicht singulär ist.

Eine Forderung zur Erfüllbarkeit dieser Bedingung ist, dass die Jacobi-Matrix der Bedingungen dc_k vollen Rank hat. Dies bedeutet, dass alle Gradienten $\nabla c_i, i = 1, \dots, m$ linear unabhängig sein müssen. Wir werden später die Auswirkung dieser Forderung auf die Registrierung diskutieren.

Ein Basisalgorithmus für das SQP-Verfahren ist leicht herzuleiten. Er basiert auf einer Iteration, in der die gesuchten Größen p_k, q_k durch die Lösung des Systems (3.7) bestimmt werden. Als Abbruchkriterien verwenden wir die KKT-Bedingungen.

Die KKT-Matrix ist aufgrund ihrer Blockstruktur mit dem Null-Block indefinit. Dies wirkt sich auf die numerische Lösung des Systems aus; wir werden dies in Kapitel 13.4.1 diskutieren, wenn das Verfahren zur Registrierung eingesetzt wird.

Analog zu den Methoden der nicht bedingten Optimierung sind auch bei den bedingten Methoden Kontrollen eingeführt worden, die testen, ob die bestimmten Suchrichtungen p_k, q_k sinnvoll im Sinne des bedingten Optimierungsproblems sind.

3.4.2 Merit-Funktion

Die Merit-Funktion $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$ soll sicherstellen, dass die SQP-Methode konvergiert, indem die Suchrichtung auf ihre Plausibilität geprüft wird. Die Merit-Funktion wird so formuliert, dass sie die Aufgabe der Zielfunktion f bei der nicht bedingten Optimierung übernimmt. Jeder Schritt im SQP-Verfahren soll dann eine Verminderung der Funktionswerte der Merit-Funktion liefern,

$$\psi(x_k + p_k) < \psi(x_k).$$

Verschiedene Möglichkeiten zur Definition einer Merit-Funktion sind in der Literatur beschrieben: Wir wählen die l_1 -Merit-Funktion. Sie ist durch

$$\psi(x, \mu) = f(x) + \frac{1}{\mu} \|c(x)\|_1$$

gegeben. Ihr Name leitet sich aus der Verwendung der l_1 -Norm ab. Die Wahl des Parameters μ erfolgt in jeder Iteration. Lemma 18.2 in [56] besagt, dass die Wahl

$$\mu = \|\lambda_{k+1}\|_\infty + \bar{\delta}$$

optimal ist. Hierbei ist wiederum $\lambda_{k+1} = \lambda_k + q_k$.

Nach dessen Bestimmung erfolgt eine Iteration ähnlich wie in Abschnitt 3.1 über die Schrittweitenbestimmung im nicht-restringierten Fall, eben nur für die Merit-Funktion. Daraus resultiert ein Gewichtungssparameter α^{LS} und damit eine neue Iterierte $x_{k+1} = x_k + \alpha^{LS} p_k$.

Damit erhalten wir abschließend als Gesamtalgorithmus zur bedingten Optimierung das in Algorithmus 3 beschriebene Verfahren.

Algorithmus 3 SQP-Verfahren; SQP

Eingabe: f, c, x_0, λ_0

Ausgabe: x_*, λ_*

```

1:  $k \leftarrow 0$ 
2: while !stopp
3:   Löse  $\begin{pmatrix} d_2 L_k & dc_k^\top \\ dc_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = \begin{pmatrix} -\nabla f_k - dc_k^\top \lambda_k \\ -c_k \end{pmatrix}$ 
4:    $\mu = \|\lambda_{k+1}\|_\infty + \bar{\delta}$ 
5:    $\alpha^{LS} \leftarrow 1$ 
6:   while  $\psi(x_k + \alpha^{LS} p_k, \mu) \geq \psi(x_k, \mu)$ 
7:      $\alpha^{LS} \leftarrow \alpha^{LS} / 2$ 
8:   end
9:    $x_{k+1} \leftarrow x_k + \alpha^{LS} p_k$ ;  $\lambda_{k+1} \leftarrow \lambda_k + q_k$ 
10:   $k \leftarrow k + 1$ 
11: end
12:  $x_* \leftarrow x_k$ ;  $\lambda_* \leftarrow \lambda_k$ 

```

Fazit

Mit dem Gauß-Newton- und dem SQP-Verfahren haben wir zwei Methoden der nicht bedingten und bedingten Optimierung beschrieben. Sie sind mit Linesearch-Verfahren und Merit-Funktion vollständig beschrieben. Die Verfahren sind allgemein gehalten, eine Spezialisierung auf die Registrierungsproblematik hat bisher nicht stattgefunden. Dies wird durch die Konkretisierung der

Zielfunktion in den Kapiteln 6 und 8 geschehen. Die Verfahren sind so detailliert beschrieben, dass zur Herleitung von Registrierungsalgorithmen nur die Zielfunktion samt ihrer Ableitungen oder Approximation der Ableitungen sowie ein Startwert angegeben werden müssen. Dies wird die Formulierung der Registrierungsverfahren erleichtern.

Die Optimierungsverfahren sind anwendbar auf Funktionen f der Form $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Um das kontinuierliche Problem als diskretes Optimierungsproblem schreiben zu können, müssen alle beteiligten Größen diskretisiert werden. Ein wichtiges Werkzeug hierfür stellen die im nächsten Kapitel definierten Gitter dar.

Kapitel 4

Gitter

Ein wesentliches Mittel, um mit diskreten Daten operieren zu können sind Kenntnisse über regelmäßige Gitter. Ein Gitter ist eine Menge von Punkten, regelmäßig ist dies, wenn seine Punkte im Raum einem Muster folgen. Regelmäßige Gitter benötigen wir sowohl zur Beschreibung von Interpolationsmethoden als auch bei der Diskretisierung des Registrierungsproblems. Daher stellen wir sie diesem Kapitel voran.

Ziel des Kapitels ist es, zu erläutern, wie ein gegebenes Gebiet $\Omega \subset \mathbb{R}^d$ der Gestalt $\Omega = (0, \omega_1) \times \dots \times (0, \omega_d)$ durch $m = (m_1, \dots, m_d) \in \mathbb{N}^d$ Diskretisierungspunkte unterteilt werden kann und wie wir derartige Strukturen nutzen können. Wir stellen drei verschiedene Arten von Gittern vor, die wir im weiteren Verlauf der Arbeit für unterschiedliche Diskretisierungen von Funktionen und deren Ableitungen benötigen.

Vorweg noch eine Anmerkung zur Namensgebung: Die Attribute *cellcentered*, *nodal* und *staggered* für die Beschreibung von Gittern sind in der englischsprachigen Literatur üblich. Wir verwenden sie hier, da deutsche Äquivalente sehr künstlich wirken.

Zunächst wiederholen wir Begrifflichkeiten der linearen Algebra, die für dieses und die folgenden Kapitel nützlich sind. Sie ermöglichen eine klare Notation höherdimensionaler ($d \geq 2$) Gitter abgeleitet aus der eindimensionalen Notation.

Definition 4.0.1 Seien $A \in \mathbb{R}^{m_1 \times m_2}$ und $B \in \mathbb{R}^{n_1 \times n_2}$ zwei Matrizen, dann heißt

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1m_2}B \\ \vdots & \ddots & \vdots \\ a_{m_11}B & \cdots & a_{m_1m_2}B \end{pmatrix} \in \mathbb{R}^{m_1n_1 \times m_2n_2}$$

Kronecker-Produkt von A und B .

Definition 4.0.2 Seien $A, B \in \mathbb{R}^{m_1 \times m_2}$ zwei Matrizen, dann heißt

$$A \odot B = \begin{pmatrix} a_{11}b_{11} & \cdots & a_{1m_2}b_{1m_2} \\ \vdots & \ddots & \vdots \\ a_{m_11}b_{m_11} & \cdots & a_{m_1m_2}b_{m_1m_2} \end{pmatrix} \in \mathbb{R}^{m_1 \times m_2}$$

Hadamard-Produkt von A und B und mit $b_{ij} \neq 0$

$$A \oslash B = \begin{pmatrix} a_{11}/b_{11} & \cdots & a_{1m_2}/b_{1m_2} \\ \vdots & \ddots & \vdots \\ a_{m_11}/b_{m_11} & \cdots & a_{m_1m_2}/b_{m_1m_2} \end{pmatrix} \in \mathbb{R}^{m_1 \times m_2}$$

Hadamard-Division von A und B .

Definition 4.0.3

1. Seien $A \in \mathbb{R}^{m_1 \times m_2}$ eine Matrix, dann bezeichnet $A_{k,:}$ die k .te Zeile und $A_{:,l}$ die l .te Spalte von A .
2. Sei $m \in \mathbb{N}$, dann bezeichnet $e_m \in \mathbb{R}^m$ einen Spaltenvektor, dessen Einträge Einsen sind.
3. Sei $A \in \mathbb{R}^{m_1 \times m_2}$, und $1 \leq k_1 \leq k_2 \leq m_1$, $1 \leq l_1 \leq l_2 \leq m_2$, dann bezeichnet $A_{k_1:k_2, l_1:l_2} \in \mathbb{R}^{k_2-k_1+1 \times l_2-l_1+1}$ die Matrix, die aus den k_1 bis k_2 Zeilen und den l_1 bis l_2 Spalten von A besteht, das heißt $A_{k_1:k_2, l_1:l_2} = (a_{kl})_{k=k_1, \dots, k_2; l=l_1, \dots, l_2}$

4.1 Eindimensionale Gitter

Das Intervall $\Omega = (0, \omega)$ wird in gleichgroße Zellen der Größe $h = \omega/m$ zerteilt. Die Mittelpunkte dieser Zellen bilden ein *Cellcentered-Gitter*, die Randpunkte der Zellen ein *Nodal-Gitter*, vergleiche Abbildung 4.1. Die Koordinaten der jeweiligen Punkte speichern wir in Vektoren. Für das Cellcentered-Gitter erhalten wir

$$\mathbf{x}^{cc} \in \mathbb{R}^m \quad (\mathbf{x}^{cc})_k = \left(k - \frac{1}{2}\right) h \quad k = 1, \dots, m$$

und für das Nodal-Gitter

$$\mathbf{x}^n \in \mathbb{R}^{m+1} \quad (\mathbf{x}^n)_k = kh \quad k = 0, \dots, m.$$



Abbildung 4.1: Eindimensionale Gitter; cellcentered (rote Kreuze) und nodal (grüne Kreise)

Später wird es von Interesse sein, Daten, die auf einem der beiden Gitter gegeben sind, auf die Koordinaten des anderen Gitters umzurechnen. Hierzu nutzen wir Gitterwechseloperatoren. Durch

$$P_{cc}^n = \frac{1}{2} \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \end{pmatrix} \in \mathbb{R}^{m \times m+1}$$

erhalten wir einen Wechsel von einem Nodal- zu einem Cellcentered-Gitter und durch

$$P_n^{cc} = \frac{1}{2} \begin{pmatrix} 3 & -1 & & & \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ & & & -1 & 3 \end{pmatrix} \in \mathbb{R}^{m+1 \times m}$$

einen Wechsel von einem Cellcentered- zu einem Nodal-Gitter. Es gilt für ein festes Ω und m

$$\mathbf{x}^n = P_n^{cc} \mathbf{x}^{cc}, \quad \mathbf{x}^{cc} = P_{cc}^n \mathbf{x}^n.$$

4.2 Zweidimensionale Gitter

Gegeben sei das Gebiet $\Omega = (0, \omega_1) \times (0, \omega_2) \subset \mathbb{R}^2$, $\omega = (\omega_1, \omega_2)$ und $m = (m_1, m_2) \in \mathbb{N}^2$ die Anzahl der Diskretisierungspunkte in x_1 - und x_2 -Richtung. Dann ist $h = \omega \oslash m \in \mathbb{R}^2$ die Zellgröße. Die Zelle (k, l) hat also den Mittelpunkt $((k - \frac{1}{2})h_1, (l - \frac{1}{2})h_2)$. Wir sammeln alle diese Zellmittelpunkte in einer Matrix, indem wir das Gebiet speziell durchlaufen. Hierbei beginnen wir mit der Zelle $(1, 1)$, erhöhen zunächst den Index der x_1 -Richtung (bis zu $(m_1, 1)$) und dann den Index der x_2 -Richtung (also folgend $(1, 2), \dots, (m_1, 2), (1, 3), \dots, (m_1, m_2)$). Wir erhalten also

$$\bar{\mathbf{x}}^{cc} \in \mathbb{R}^{m_1 m_2 \times 2}, \quad (\bar{\mathbf{x}}^{cc})_{k+(m_1-1)l,:} = \left(\left(k - \frac{1}{2} \right) h_1, \left(l - \frac{1}{2} \right) h_2 \right).$$

Hieraus ist abzuleiten, dass der Nullpunkt des von uns gewählten Koordinatensystems sich unten links befindet.

Die $k + (m_1 - 1)$. Zeile von $\bar{\mathbf{x}}^{cc}$ enthält also die Koordinaten des Zellmittelpunktes der $k + (m_1 - 1)$. Zelle.

Um für die später folgende Optimierung eine Zielgröße des Format $n \times 1$ zu erhalten, formatieren wir $\bar{\mathbf{x}}^{cc}$ durch spaltenweises Untereinanderschreiben um zu \mathbf{x}^{cc} ,

$$\mathbf{x}^{cc} = \begin{pmatrix} \bar{\mathbf{x}}_{:,1}^{cc} \\ \bar{\mathbf{x}}_{:,2}^{cc} \end{pmatrix}.$$

Dieses Vorgehen illustrieren wir zur besseren Anschauung in einem Beispiel:

Beispiel 4.2.1 Seien $\omega = (4, 3)$ die Gebietsgrenzen und $m = (4, 3)$ die Anzahl der Diskretisierungspunkte. Die 12 Zellen, die wir durch diese Diskretisierung erhalten sind in Abbildung 4.2, links mit den Koordinaten ihrer Zellmittelpunkte dargestellt. In der gleichen Abbildung rechts, ist die Durchlaufrichtung über die Zellen durch Angabe ihrer Position dargestellt. Wir erhalten somit

$$\bar{\mathbf{x}}^{cc} = \begin{pmatrix} (0.5, 0.5), & (1.5, 0.5), & (2.5, 0.5), & (3.5, 0.5), & \dots \\ (0.5, 1.5), & (1.5, 1.5), & (2.5, 1.5), & (3.5, 1.5), & \dots \\ (0.5, 2.5), & (1.5, 2.5), & (2.5, 2.5), & (3.5, 2.5) \end{pmatrix}^T \in \mathbb{R}^{12 \times 2},$$

$$\mathbf{x}^{cc} = \begin{pmatrix} 0.5, & 1.5, & 2.5, & 3.5, & 0.5, & 1.5, & 2.5, & 3.5, & \dots \\ 0.5, & 1.5, & 2.5, & 3.5, & 0.5, & 0.5, & 0.5, & 0.5, & \dots \\ 1.5, & 1.5, & 1.5, & 1.5, & 2.5, & 2.5, & 2.5, & 2.5 \end{pmatrix}^T \in \mathbb{R}^{24}.$$

Zu beobachten ist, dass wir das 2D-Cellcentered-Gitter durch die Kombination zweier 1D-Cellcentered-Gitter erhalten können. Dafür sei $\mathbf{x}_1^{cc} \in \mathbb{R}^{m_1}$ das Gitter in x_1 -Richtung und $\mathbf{x}_2^{cc} \in \mathbb{R}^{m_2}$ das Gitter in x_2 -Richtung. Das 2D-Cellcentered-Gitter erhalten wir durch

$$\mathbf{x}^{cc} = \begin{pmatrix} e_{m_2} \otimes \mathbf{x}_1^{cc} \\ \mathbf{x}_2^{cc} \otimes e_{m_1} \end{pmatrix} \in \mathbb{R}^{2m_1m_2}.$$

Als Koordinatenvektor erhalten wir

$$\bar{\mathbf{x}}^{cc} = (\mathbf{x}_{1:n}^{cc}, \mathbf{x}_{n+(1:n)}^{cc}), \quad n = m_1m_2.$$

Ein solches Gitter ist in Abbildung 4.3 durch rote Kreuze visualisiert. Analog zum 1D-Fall können wir ein 2D-Nodal-Gitter erzeugen. Dieses können wir durch die Kombination zweier 1D-Nodal-Gitter erhalten. Dazu seien $\mathbf{x}_1^n \in \mathbb{R}^{m_1+1}$ und $\mathbf{x}_2^n \in \mathbb{R}^{m_2+1}$ die jeweiligen Gitter, dann ist

$$\mathbf{x}^n = \begin{pmatrix} e_{m_2+1} \otimes \mathbf{x}_1^n \\ \mathbf{x}_2^n \otimes e_{m_1+1} \end{pmatrix} \in \mathbb{R}^{2(m_1+1)(m_2+1)}$$

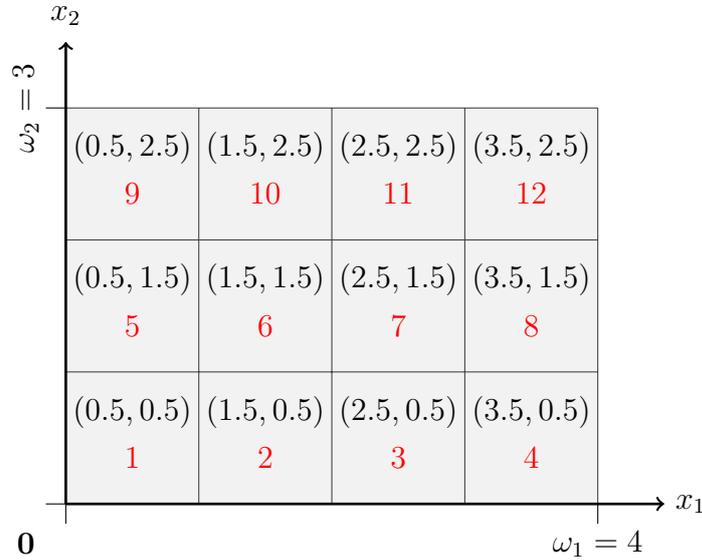


Abbildung 4.2: Koordinaten und Durchlauf für ein 2D-Cell-Centered-Gitter; $\omega = (4, 3)$, $m = (4, 3)$; In jede Zelle sind die Koordinaten ihres Mittelpunktes eingetragen; rot: Nummerierung der Zellen dem im Text beschriebenen Durchlaufsmuster folgend

das 2D-Nodal-Gitter. Zusätzlich zum 1D-Fall besteht im 2D-Fall die Möglichkeit, ein 1D-Nodal-Gitter mit einem 1D-Cellcentered-Gitter zu einem sogenannten Staggered-Gitter zu kombinieren. Dazu seien im ersten Fall $\mathbf{x}_1^n \in \mathbb{R}^{m_1+1}$ und $\mathbf{x}_2^{cc} \in \mathbb{R}^{m_2}$, dann ist

$$\mathbf{x}^{stg1} = \begin{pmatrix} e_{m_2} & \otimes & \mathbf{x}_1^n \\ \mathbf{x}_2^{cc} & \otimes & e_{m_1+1} \end{pmatrix} \in \mathbb{R}^{2(m_1+1)m_2}.$$

Analog erhalten wir für

$$\mathbf{x}^{stg2} = \begin{pmatrix} e_{m_2+1} & \otimes & \mathbf{x}_1^{cc} \\ \mathbf{x}_2^n & \otimes & e_{m_1} \end{pmatrix} \in \mathbb{R}^{2m_1(m_2+1)}.$$

Später benötigen wir für effiziente Registrierungsalgorithmen eine Mischung aus beiden Staggered-Gittern: Die x_1 -Komponenten des \mathbf{x}^{stg1} und die x_2 -Komponenten des \mathbf{x}^{stg2} . Dieses Gitter bezeichnen wir ab jetzt als Staggered-Gitter

$$\mathbf{x}^{stg} = \begin{pmatrix} e_{m_2} & \otimes & \mathbf{x}_1^n \\ \mathbf{x}_2^n & \otimes & e_{m_1} \end{pmatrix} \in \mathbb{R}^{(m_1+1)m_2+m_1(m_2+1)}.$$

Die Gitter sind ebenfalls in Abbildung 4.3 dargestellt.

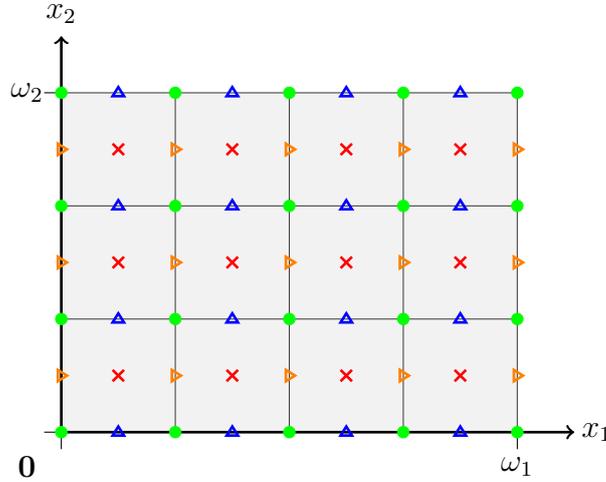


Abbildung 4.3: Zusammenfassung 2D-Gitter; Gebiet Ω hellgrau hinterlegt; Cellcentered-Gitter (rote Kreuze); Nodal-Gitter (grüne Kreise); Staggered-Gitter in x_1 (gelbe Dreiecke); Staggered-Gitter in x_2 (blaue Dreiecke)

4.3 Dreidimensionale Gitter

Im 2D-Fall haben wir geklärt, wie die 1D-Operatoren auf höhere Dimensionen angepasst werden können. Dies geschah durch die Schreibweise als Kronecker-Produkt mit den Hilfsvektoren e_* . Weiter haben wir gesehen, dass es in zwei Dimensionen nicht nur Nodal- und Cellcentered-Gitter gibt, sondern dass durch die Kombination beider Gitter-Typen auch qualitativ andere Gitter, Staggered-Gitter, entstehen. Diese Gitter wurden als Vektoren gespeichert, wobei wir eine spezielle Durchlaufreihenfolge durch die Punkte des Gitters ausgewählt haben. Diese Konzepte übertragen wir nun auf drei Dimensionen. Seien wie im vorherigen Fall $\mathbb{R}^3 \supset \Omega = (0, \omega_1) \times (0, \omega_2) \times (0, \omega_3)$, $\omega = (\omega_1, \omega_2, \omega_3)$ und $m \in \mathbb{N}^3$, $h = \omega \otimes m$. Seien $\mathbf{x}_1^{cc} \in \mathbb{R}^{m_1}$, $\mathbf{x}_2^{cc} \in \mathbb{R}^{m_2}$ und $\mathbf{x}_3^{cc} \in \mathbb{R}^{m_3}$ eindimensionale Cellcentered-Gitter und $\mathbf{x}_1^n \in \mathbb{R}^{m_1+1}$, $\mathbf{x}_2^n \in \mathbb{R}^{m_2+1}$ und $\mathbf{x}_3^n \in \mathbb{R}^{m_3+1}$ eindimensionale Nodal-Gitter zu den jeweiligen Diskretisierungsgrößen m_1, m_2, m_3 . Dann erhalten wir für den 3D-Fall ein Cellcentered-Gitter

$$\mathbf{x}^{cc} = \begin{pmatrix} e_{m_3} \otimes e_{m_2} \otimes \mathbf{x}_1^{cc} \\ e_{m_3} \otimes \mathbf{x}_2^{cc} \otimes e_{m_1} \\ \mathbf{x}_3^{cc} \otimes e_{m_2+1} \otimes e_{m_1} \end{pmatrix}$$

und genauso ein Nodal-Gitter

$$\mathbf{x}^n = \begin{pmatrix} e_{m_3+1} \otimes e_{m_2+1} \otimes \mathbf{x}_1^n \\ e_{m_3+1} \otimes \mathbf{x}_2^n \otimes e_{m_1+1} \\ \mathbf{x}_3^n \otimes e_{m_2+1} \otimes e_{m_1+1} \end{pmatrix}.$$

Im 2D-Fall erhielten wir durch diese Kombination Vektoren, bei denen zuerst die x_1 -Komponente, dann die x_2 -Komponente aufgeführt wurde. Jede Kom-

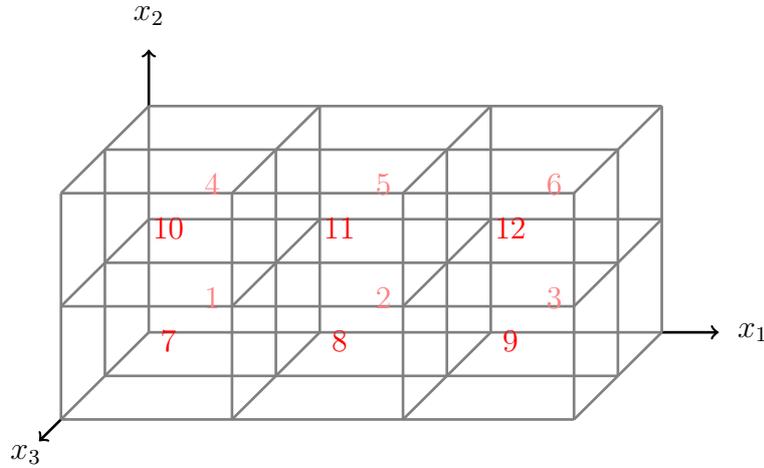


Abbildung 4.4: Durchlaufreichtung in drei Raumdimensionen

ponente folgte der Reihenfolge, wie in Beispiel 4.2.1 beschrieben. Diese Reihenfolge erwirkt einen Durchlauf der Ordnung: erst Zeilenweise, dann Spaltenweise, zuletzt Schichtenweise, wobei Schichten, die Struktur in Richtung der x_3 -Raumkomponente meint.

Durch die Kombination eines Nodal-Gitters mit einem Cellcentered-Gitter ergaben sich im 2D-Fall zwei Staggered-Gitter. Im dreidimensionalen Fall können wir nun ein Cellcentered-Gitter mit zwei Nodal-Gittern kombinieren und erhalten je eines von drei Staggered-2-Gittern oder wir kombinieren zwei Cellcentered-Gitter mit einem Nodal-Gitter und erhalten je eines von drei Staggered-1-Gittern. Später benötigen wir nur die letztgenannten, führen aber der Vollständigkeit halber alle sechs Gitterarten hier auf:

$$\mathbf{x}^{stg1} = \begin{pmatrix} e_{m_3} \otimes e_{m_2} \otimes \mathbf{x}_1^n \\ e_{m_3} \otimes \mathbf{x}_2^{cc} \otimes e_{m_1+1} \\ \mathbf{x}_3^{cc} \otimes e_{m_2} \otimes e_{m_1+1} \end{pmatrix},$$

$$\mathbf{x}^{stg2} = \begin{pmatrix} e_{m_3} \otimes e_{m_2+1} \otimes \mathbf{x}_1^{cc} \\ e_{m_3} \otimes \mathbf{x}_2^n \otimes e_{m_1} \\ \mathbf{x}_3^{cc} \otimes e_{m_2+1} \otimes e_{m_1} \end{pmatrix},$$

$$\mathbf{x}^{stg3} = \begin{pmatrix} e_{m_3+1} \otimes e_{m_2} \otimes \mathbf{x}_1^{cc} \\ e_{m_3+1} \otimes \mathbf{x}_2^{cc} \otimes e_{m_1} \\ \mathbf{x}_3^n \otimes e_{m_2} \otimes e_{m_1} \end{pmatrix},$$

$$\mathbf{x}^{Stg1} = \begin{pmatrix} e_{m_3+1} \otimes e_{m_2+1} \otimes \mathbf{x}_1^{cc} \\ e_{m_3+1} \otimes \mathbf{x}_2^n \otimes e_{m_1} \\ \mathbf{x}_3^n \otimes e_{m_2+1} \otimes e_{m_1} \end{pmatrix},$$

$$\mathbf{x}^{Stg_2} = \begin{pmatrix} e_{m_3+1} \otimes e_{m_2} \otimes \mathbf{x}_1^n \\ e_{m_3+1} \otimes \mathbf{x}_2^{cc} \otimes e_{m_1+1} \\ \mathbf{x}_3^n \otimes e_{m_2} \otimes e_{m_1+1} \end{pmatrix},$$

$$\mathbf{x}^{Stg_3} = \begin{pmatrix} e_{m_3} \otimes e_{m_2+1} \otimes \mathbf{x}_1^n \\ e_{m_3} \otimes \mathbf{x}_2^n \otimes e_{m_1+1} \\ \mathbf{x}_3^{cc} \otimes e_{m_2+1} \otimes e_{m_1+1} \end{pmatrix}.$$

Interpretieren wir diese acht 3D-Gitter, die in Abbildung 4.5 für ein kleines $m = (2, 2, 2)$ dargestellt sind. Wir erhalten in 3D quaderförmige Zellen; das Nodal-Gitter lebt auf den Eckpunkten dieser Zellen, das Cellcentered-Gitter auf deren Mittelpunkten. Die Punkte der jeweiligen \mathbf{x}^{Stg_j} -Gitter liegen auf den Mittelpunkten der Zellkanten, die senkrecht zur x_j -Raumrichtung stehen. Hingegen leben die \mathbf{x}^{stg_j} auf den Mittelpunkten der Zellflächen, die senkrecht zur Raumrichtung x_j stehen. Analog zum zweidimensionalen Fall benötigen wir später nur ein zusammengesetztes Staggered-Gitter, das als j .te Komponente die j . Komponente von \mathbf{x}^{stg_j} erhält. Dieses Gitter bezeichnen wir im folgenden als Staggered-Gitter:

$$\mathbf{x}^{stg} = \begin{pmatrix} e_{m_3} \otimes e_{m_2} \otimes \mathbf{x}_1^n \\ e_{m_3} \otimes \mathbf{x}_2^n \otimes e_{m_1} \\ \mathbf{x}_3^n \otimes e_{m_2} \otimes e_{m_1} \end{pmatrix}.$$

Wie schon zu Beginn des Kapitels beschrieben, werden wir die spezielle Wahl solcher Gitter erst in den Registrierungskapiteln dieser Arbeit motivieren; ein Vorgriff würde nur Unordnung im Geiste schaffen.

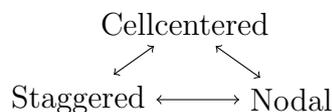
Für den 2D-Fall sind wir es schuldig geblieben, die Gitter-Wechsel-Operatoren vorzustellen. Wir holen dies nun als Spezialfall der Operatoren für drei Raumdimensionen nach. Durch die Kronecker-Struktur aller vorgestellten Gitter wird es möglich, die für den 1D-Fall präsentierten Operatoren P_{cc}^n und P_n^{cc} als Grundsteine für die hochdimensionalen Operatoren zu nutzen. Um unsere Schreibweise zu vereinfachen, setzen wir

$$P_j = P_{cc}^n(m_j) \in \mathbb{R}^{m_j \times m_j + 1}$$

als eindimensionalen Gitter-Wechsel-Operator eines Nodal-Gitters auf ein Cellcentered-Gitter der Größe m_j und analog

$$Q_j = P_n^{cc}(m_j) \in \mathbb{R}^{m_j + 1 \times m_j}$$

als 1D Gitter-Wechsel-Operator von Cellcentered-Gitter nach Nodal-Gitter. Für die uns später wichtigen Wechsel



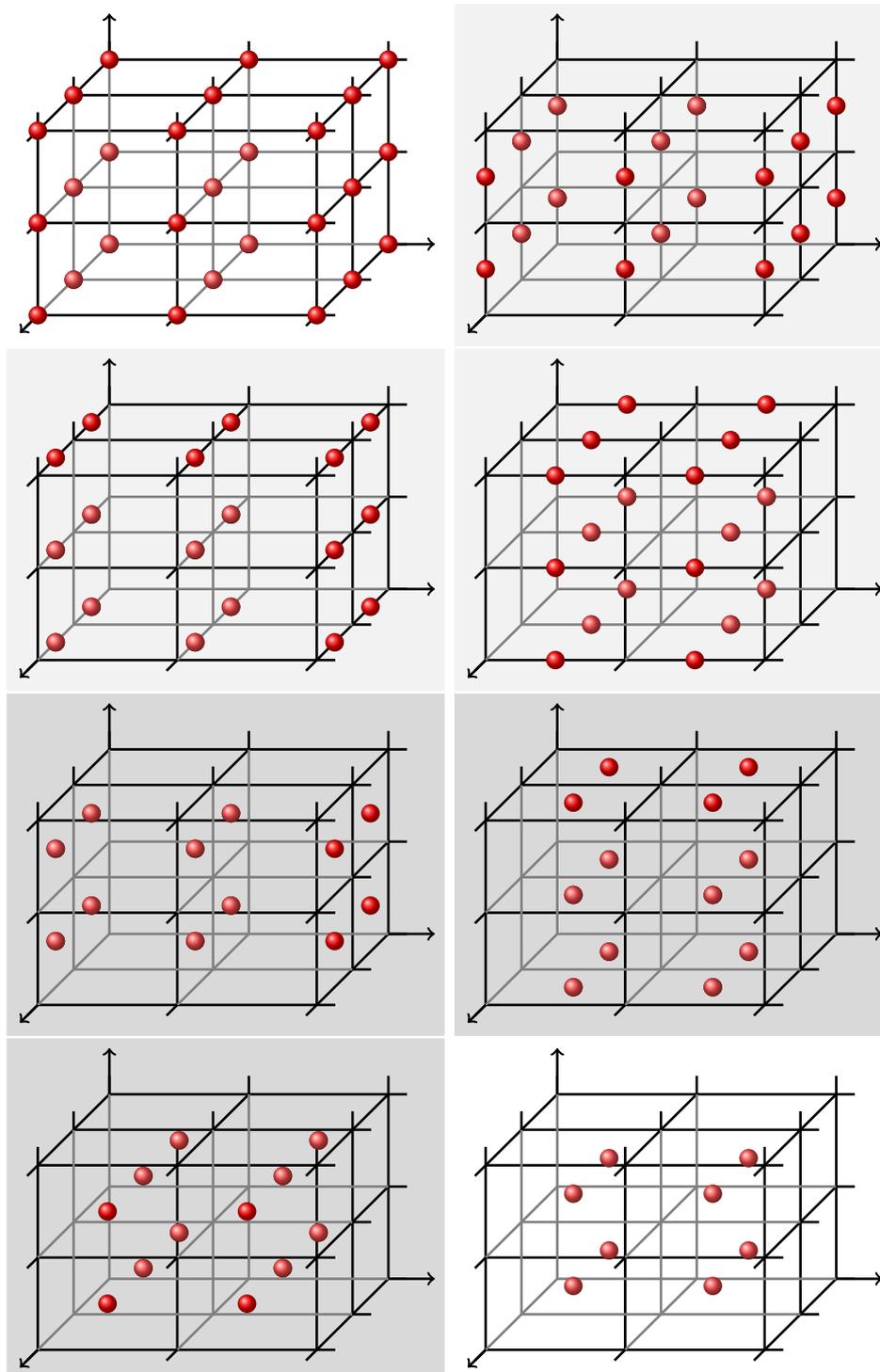


Abbildung 4.5: 3D-Gitter; von oben nach unten, von links nach rechts: Nodal-Gitter; Staggered-2 je in x_1, x_2, x_3 (hellgrau hinterlegt); Staggered-1 je in x_1, x_2, x_3 (dunkelgrau hinterlegt) und Cell-Centered-Gitter

erhalten wir die sechs folgenden Operatoren:

$$\begin{aligned}
P_{cc}^n &= (I_3 \otimes (P_3 \otimes P_2 \otimes P_1)) & P_n^{cc} &= (I_3 \otimes (Q_3 \otimes Q_2 \otimes Q_1)) \\
P_{stg}^{cc} &= \begin{pmatrix} I_{m_3} \otimes I_{m_2} \otimes Q_1 & & \\ & I_{m_3} \otimes Q_2 \otimes I_{m_1} & \\ & & Q_3 \otimes I_{m_2} \otimes I_{m_1} \end{pmatrix} \\
P_{cc}^{stg} &= \begin{pmatrix} I_{m_3} \otimes I_{m_2} \otimes P_1 & & \\ & I_{m_3} \otimes P_2 \otimes I_{m_1} & \\ & & P_3 \otimes I_{m_2} \otimes I_{m_1} \end{pmatrix} \\
P_{nod}^{stg} &= \begin{pmatrix} Q_3 \otimes Q_2 \otimes I_{m_1+1} & & \\ & Q_3 \otimes I_{m_2+1} \otimes Q_1 & \\ & & I_{m_3+1} \otimes Q_2 \otimes Q_1 \end{pmatrix} \\
P_{stg}^{nod} &= \begin{pmatrix} P_3 \otimes P_2 \otimes I_{m_1+1} & & \\ & P_3 \otimes I_{m_2+1} \otimes P_1 & \\ & & I_{m_3+1} \otimes P_2 \otimes P_1 \end{pmatrix}
\end{aligned}$$

und $n = m_1 m_2 m_3$, $\bar{n} = (m_1 + 1)(m_2 + 1)(m_3 + 1)$.

Um später in knapper Notation mit diesen Gittern arbeiten zu können, definieren wir die Abkürzungen

$$\mathbf{x}^{cc}(\Omega, m), \quad \mathbf{x}^n(\Omega, m), \quad \mathbf{x}^{stg}(\Omega, m)$$

für jeweilige Cellcentered, Nodal- und Staggered-Gitter auf einem Gebiet Ω und der Anzahl von Diskretisierungspunkten m . Eine weitere Kurzschreibweise ist das Zellvolumen

$$\bar{h} = h_1 h_2 h_3,$$

das wir für spätere Quadraturformeln benötigen.

Fazit

Wir haben in diesem Kapitel drei verschiedene Typen von Gitterstrukturen ausführlich kennengelernt. Diese erfüllen im Weiteren unterschiedliche Aufgaben. Zum einen interpretieren wir gegebene Bilddaten als Daten über einem regelmäßigen Gitter, so dass diese Gitter Grundlage der im nächsten Kapitel beschriebenen Interpolationsmethoden sind. Zum anderen dienen Gitter zur Diskretisierung von Deformationen und damit auch zur Bestimmung diskreter Bilder, vergleiche Kapitel 5.4.

Wir haben drei Gitterstrukturen und die dazugehörigen Gitterwechseloperatoren zuerst für den eindimensionalen Raum beschrieben und die Ergebnisse dann auf zwei und drei Dimensionen erweitert. Dieses Prinzip werden wir weiterhin verwenden, um Grundsätzliches zunächst klar beschreiben zu können.

Kapitel 5

Interpolation

Wir werden eine kontinuierliche Formulierung des Bildregistrierungsproblems herleiten. Ein wesentlicher Schritt darauf zu ist die Nutzung kontinuierlicher Bilder. In diesem Abschnitt werden wir zeigen, wie wir aus gegebenen diskreten Bilddaten eine kontinuierliche Funktion modellieren. Hierzu verwenden wir das Konzept der Interpolation, um die gegebenen Daten durch eine kontinuierliche Funktion zu beschreiben. Um der Forderung nach schnellen Algorithmen zu genügen, benötigen wir später differenzierbare Maße, woraus folgt, dass wir ein differenzierbares Bildmodell nutzen müssen. Zusammen mit dem allgemeinen Wunsch nach schnellen Algorithmen ergeben sich drei Forderungen an die Interpolante:

- sie soll die Daten gut beschreiben,
- sie soll schnell berechenbar sein,
- sie soll differenzierbar sein.

Wir wollen Verfahren einsetzen, die davon ausgehen, dass die Zielfunktion differenzierbar ist. Diese konvergieren im Allgemeinen schneller als Verfahren, die nicht Ableitungen der Zielfunktion verwenden.

Nach diesen Kriterien werden wir die vorzustellenden Methoden bewerten.

Der folgende Abschnitt zerfällt in drei Teile: Eine allgemeine Einführung zu den auftretenden Größen und die hierzu verwendete Notation, zweitens die Beschreibung dreier Interpolationsmethoden im Eindimensionalen und drittens die Übertragung dieser Methoden auf zwei-, drei- oder höherdimensionale Daten.

5.1 Notation

Beginnen wir mit der Beschreibung der zu interpolierenden Daten.

Die Daten sind über einem Gebiet $\Omega \subset \mathbb{R}^d$ gegeben. Hierbei ist $d \in \mathbb{N}$ die Bilddimension. Mögliche Werte für d sind $d = 2, 3, 4$. Das Gebiet Ω wird auf den Nullpunkt normiert. Wir erhalten

$$\Omega = (0, \omega_1) \times \dots \times (0, \omega_d)$$

mit $\omega_j \in \mathbb{R}^+$.

Die Daten sind an diskreten Punkten eines regelmäßigen Cellcentered-Gitteres \mathbf{x}^D gemessen. Hierbei steht das hochgestellte D für *Daten*, um eine Verwechslung mit den später einzuführenden Gittern zu vermeiden. Das Datengitter hat

$$m^D = (m_1^D, \dots, m_d^D) \in \mathbb{N}^d$$

Gitterpunkte. Die Gesamtzahl der Punkte beträgt also

$$n^D = m_1^D \cdot \dots \cdot m_d^D \in \mathbb{N}.$$

Damit ergibt sich eine Datenvoxelgröße von

$$h^D = (h_1^D, \dots, h_d^D) \quad \text{mit} \quad h_j^D = \omega_j \oslash m_j^D, \quad j = 1, \dots, d$$

und für das Datengitter

$$\mathbf{x}^D = \mathbf{x}^{cc}(\Omega, m^D).$$

An jedem einzelnen Punkt des Gitters ist ein Datum gegeben. Diese werden in dem Vektor

$$t^D \in \mathbb{R}^{n^D}$$

zusammengefasst. Mit dem im vorherigen Kapitel festgelegten Umsortieren eines Cellcentered-Gitters von einem Vektor des \mathbb{R}^{3n} hin zu einem Element des $\mathbb{R}^{n \times 3}$ ergeben sie die Positions-Werte-Paare

$$(\bar{\mathbf{x}}_j^D, t_j^D) \in \mathbb{R}^4, \quad j = 1, \dots, n.$$

Die eingeführten Größen werden für den zweidimensionalen Fall in Abbildung 5.1 illustriert und sind in der folgenden Tabelle zusammengefasst:

Datendimension	d		$\in \mathbb{N}$
Voxelanzahl	m^D	$= (m_1^D, \dots, m_d^D)$	$\in \mathbb{R}^d$
Voxelzahl	n^D	$= m_1 \cdot \dots \cdot m_d$	
Voxelgröße	h^D	$= (h_1^D, \dots, h_d^D)$	$\in \mathbb{R}^d$
Gebiet	Ω	$= (0, \omega_1) \times \dots \times (0, \omega_d)$	$\subset \mathbb{R}^d$
Daten-Gitter	\mathbf{x}^D	$= \mathbf{x}^{cc}(\Omega, m^D)$	$\in \mathbb{R}^{d \cdot n^D}$
gegebene Daten	t^D		$\in \mathbb{R}^{n^D}$
Positions-Werte-Paare	$(\bar{\mathbf{x}}_j^D, t_j^D)$	$j = 1, \dots, n^D$	

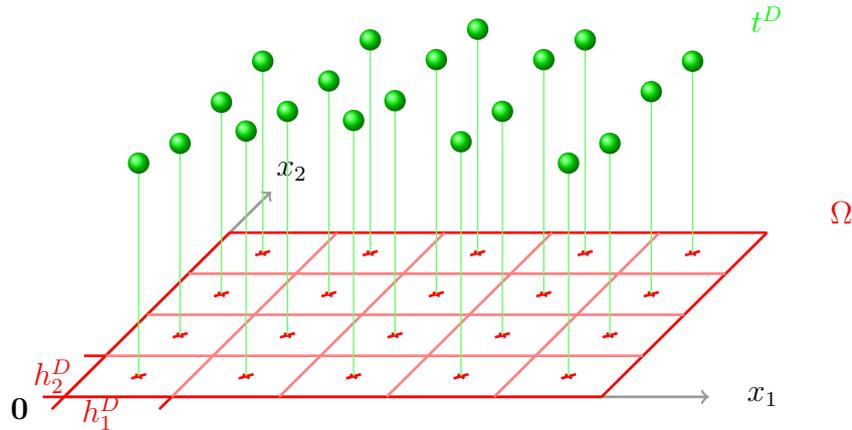


Abbildung 5.1: Illustration der Nomenklatur für die Interpolation für den zwei-dimensionalen Fall; Es ist $d = 2$, $m^D = (5, 4)$ und $n^D = 20$

Gesucht ist nun eine Funktion $I : \mathbb{R}^d \rightarrow \mathbb{R}$ mit

$$I(\bar{\mathbf{x}}_j^D) = t_j^D, \quad j = 1, \dots, n^D$$

oder, wenn wir die Interpolationsbedingung zu einer Approximationsbedingung aufweichen,

$$I(\bar{\mathbf{x}}_j^D) \approx t_j^D, \quad j = 1, \dots, n^D.$$

Mit der vorgestellten Notation gelingt die Behandlung aller in dieser Arbeit zu behandelnden Datensätze, für die Beschreibung von Interpolationsmethoden ist sie jedoch zu detailliert, weswegen wir zur reinen Niederschrift der Interpolation von ihr in diesem Kapitel absehen. Zur Beschreibung genügt das Wissen über die Anzahl der Datenpunkte m in den jeweiligen Raumdimensionen. Wir setzen daher zur Beschreibung der Interpolation die Voxelgröße $h = (1, \dots, 1)$ und transformieren das Gebiet Ω linear auf $(\frac{1}{2}, m_1 + \frac{1}{2}) \times \dots \times (\frac{1}{2}, m_d + \frac{1}{2})$, womit wir für die Position der gegebenen Datenpunkte Tupel natürlicher Zahlen erhalten.

Im Anwendungsbereich der medizinischen Bildverarbeitung sind dreidimensionale Datensätze der Standard, daher werden wir auch Interpolationsmethoden für diese Dimensionalität entwickeln. Zur Beschreibung und Untersuchung dieser Methoden bieten sich aber dreidimensionale Daten nicht an, da ihre Visualisierung sehr komplex ist und es nur mit großem Aufwand gelingt, das Wesentliche der Interpolationsmethoden herauszuarbeiten. Wir nutzen daher aus, dass die hier darzustellenden Verfahren auf eindimensionalen Verfahren fußen und befassen uns zuerst ausführlich mit diesen. Wir beschreiben sie und analysieren ihre Vor- und Nachteile. Daran anschließend stellen wir eine allgemeine Methodik vor, wie aus Verfahren für eindimensionale Daten Verfahren für höherdimensionale Daten abgeleitet werden können, wobei die Eigenschaften des jeweiligen Algorithmus aber erhalten bleiben. Wir nutzen diese Methodik dann zur Beschreibung der Verfahren für dreidimensionale Daten.

Beginnen wir nun mit der Vorstellung der eindimensionalen Verfahren:

5.2 Interpolation im Eindimensionalen

Alle Ansätze haben gemeinsam, dass sich die gesuchte Funktion $I : \mathbb{R} \rightarrow \mathbb{R}$ als Linearkombination von Funktionen schreiben lässt:

$$b_j : \mathbb{R} \rightarrow \mathbb{R}, \quad j = 1, \dots, N \quad (5.1)$$

$$I = \sum_{j=1}^N \alpha_j b_j \quad (5.2)$$

Bis auf Randfunktionen wird weiterhin für ein $x \in \mathbb{R}^3$ und $j = 1, \dots, N$ gelten

$$b_j(x) = b_0(x - j)$$

mit einer Basisfunktion b_0 .

Als weitere gemeinsame Eigenschaften gilt es festzuhalten, dass in allen vorzustellenden Fällen die Basisfunktionen positiv und gerade sind und sie kompakten Träger haben. Dies wirkt sich aus numerischer Sicht positiv auf das Bestimmen der Summe in (5.2) aus, da diese für ein gegebenes $x \in \mathbb{R}$ auf wenige Summanden zusammenfällt. Wir werden nun für vier verschiedene Interpolationsmethoden die Basisfunktion b_0 und die Wahl der Gewichte α bestimmen und ihre jeweiligen Eigenschaften diskutieren.

5.2.1 Lineare Interpolation

Die Methode der linearen Interpolation ist in der Literatur [57, 74] als Standard beschrieben. Wir fassen uns kurz und beschreiben ein Verfahren, das wir als Referenzverfahren und als Entwicklungsstufe für das folgende Verfahren gebrauchen. Die Basisfunktion für diese Methode ist gegeben durch

$$b_0(x) = \begin{cases} 1 + x, & \text{falls } x \in (-1, 0] \\ 1 - x, & \text{falls } x \in (0, 1) \\ 0, & \text{sonst;} \end{cases} \quad (5.3)$$

ihr Graph ist in Abbildung 5.2 oben links dargestellt. Diese Funktion hat als Träger das Intervall $(-1, 1)$, das heißt für ein beliebiges $x \in \mathbb{R}$ fällt die Summe (5.2), die Linearkombination der Interpolante aus den verschobenen Basisfunktionen, auf maximal zwei Summanden $b_k(x), b_{k+1}(x)$, $k = \lfloor x \rfloor$ zusammen. Das Intervall $[0.5, m + 0.5]$ wird von $N = m$ Basisfunktionen überdeckt, somit gilt $I = \sum_{j=1}^m \alpha_j b_j$ und mit $b_j(\mathbf{x}_k^D) = \delta_{jk}$ und der Interpolationsbedingung $I(\mathbf{x}_j^D) = t_j^D$, $j, k = 1, \dots, m$ folgt $\alpha = t^D$. Somit lassen sich die Gewichte schnell

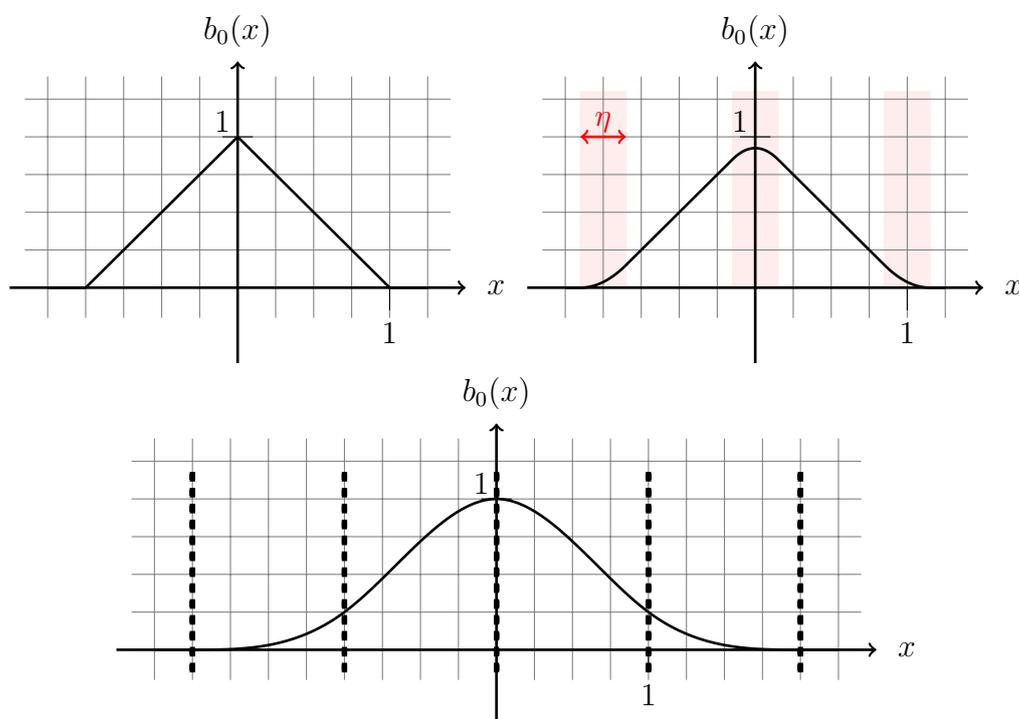


Abbildung 5.2: Darstellung der Basisfunktionen b_0 zu den jeweils vorgestellten Interpolationsarten; *oben links*: Basisfunktion für lineare Interpolation; *oben rechts*: Basisfunktion für linear glatte Interpolation, die quadratisch approximierten Bereiche sind rot hinterlegt, $\eta = 0.15$; *unten*: Basisfunktion für kubi-sche Spline-Interpolation, die einzelnen Abschnitte sind durch Linien getrennt

bestimmen und die Basisfunktionen und die Interpolante schnell auswerten. Der Graph der Interpolante lässt sich als Verbindungslinien der Datenpunkte interpretieren. Die Interpolante erfüllt ein Minimum-Maximum-Kriterium, ihre Werte leben im Intervall $[\min t^D, \max t^D]$. Aus $\alpha = t^D$ (und der graphischen Interpretation) folgt, dass eine Änderung eines einzelnen Datums nur lokale Auswirkungen auf die Interpolante hat. Nachteil des Verfahrens ist die Nicht-Differenzierbarkeit der Basisfunktion b_0 an den Punkten $x \in \{-1, 0, 1\}$ und damit die Nicht-Differenzierbarkeit der Interpolante. Dies steht im Widerspruch zu den von uns geforderten Eigenschaften, die bedingen, dass wir später schnell konvergierende Lösungsverfahren erhalten.

5.2.2 Linear glatte Interpolation

Die Methode der linear glatten Interpolation ist neu. Sie ist am Institut für Mathematik der Universität zu Lübeck entwickelt worden und bisher nur in der Masterarbeit von Jens Heyder [34] beschrieben worden. Grundlegende Idee

des Verfahrens ist es, den Nachteil der Nicht-Differenzierbarkeit der linearen Interpolation aufzuheben, dabei aber die positiven Eigenschaften der linearen Interpolation (lokaler Einfluss der Daten, kein Ringing (siehe Interpolation mit Splines 5.2.3), schnelle Auswertung) zu erhalten. Erzielt wird dies durch die Änderung der Basisfunktion in einer η -Umgebung um die nicht differenzierbaren Punkte $x \in \{-1, 0, 1\}$ durch ein Polynom zweiter Ordnung, das stetig differenzierbar eingepasst wird. Betrachten wir detailliert die jeweiligen Punkte:

Für $x = 0$ suchen wir ein Polynom $q_0 \in \Pi_2$ mit

$$\begin{aligned} q_0(-\eta) &= 1 + \eta, & q_0(\eta) &= 1 - \eta, \\ q'_0(-\eta) &= 1, & q'_0(\eta) &= -1. \end{aligned}$$

Die erste Zeile spiegelt die Forderung nach einer stetigen Fortsetzung der linearen Teilabschnitte wieder, die zweite Zeile die Forderung nach stetiger Differenzierbarkeit. Aus Symmetriegründen fällt eine Bedingung weg und wir bekommen das eindeutig bestimmte Polynom q_0 mit $q_0(x) = (1 - \eta) - (x - \eta) - \frac{1}{2\eta}(x - \eta)^2$.

Für $x = 1$ erhalten wir die Bedingungen

$$\begin{aligned} q_1(1 - \eta) &= \eta, & q_1(1 + \eta) &= 0, \\ q'_1(1 - \eta) &= -1, & q'_1(1 + \eta) &= 0 \end{aligned}$$

und wir erhalten q_1 mit $q_1(x) = \frac{1}{4\eta}(x - 1 - \eta)^2$. Für den verbleibenden Punkt $x = -1$ erhalten wir aus Symmetriegründen das Polynom q_{-1} mit $q_{-1}(x) = \frac{1}{4\eta}(x + 1 + \eta)^2$.

Somit erhalten wir als Basisfunktion b_0 für die linear glatte Interpolation

$$b_0(x) = \begin{cases} \frac{1}{4\eta}(x + 1 + \eta)^2, & \text{falls } x \in (-(1 + \eta), -(1 - \eta)) \\ 1 + x, & \text{falls } x \in [-(1 - \eta), -\eta] \\ (1 - \eta) - (x - \eta) - \frac{1}{2\eta}(x - \eta)^2, & \text{falls } x \in (-\eta, +\eta) \\ 1 - x, & \text{falls } x \in [\eta, (1 - \eta)] \\ \frac{1}{4\eta}(x - 1 - \eta)^2, & \text{falls } x \in (1 - \eta, 1 + \eta) \\ 0, & \text{sonst.} \end{cases} \quad (5.4)$$

Diese Funktion ist stetig differenzierbar und ihre Ableitung ist

$$b'_0(x) = \begin{cases} \frac{1}{2\eta}(x + 1 + \eta), & \text{falls } x \in (-(1 + \eta), -(1 - \eta)) \\ 1, & \text{falls } x \in [-(1 - \eta), -\eta] \\ -1 - \frac{1}{\eta}(x - \eta), & \text{falls } x \in (-\eta, +\eta) \\ -1, & \text{falls } x \in [\eta, (1 - \eta)] \\ \frac{1}{2\eta}(x - 1 - \eta), & \text{falls } x \in (1 - \eta, 1 + \eta) \\ 0, & \text{sonst.} \end{cases} \quad (5.5)$$

Der Graph der Basisfunktion ist in Abbildung 5.2, oben rechts dargestellt. Die Basisfunktion hat Träger $(-1 - \eta, 1 + \eta)$, das Intervall $(0.5, m + 0.5)$ wird von $N = m$ Basisfunktionen überdeckt, für ein gegebenes $x \in (0.5, m + 0.5)$ gilt, dass zwei oder drei Basisfunktionen von Null verschieden sind, abhängig davon, ob $x \in \bigcup_{j=1}^m \{t | \mathbf{x}_j^D - \eta < t < \mathbf{x}_j^D + \eta\}$, also in der η -Umgebung um einen Datenpunkt liegt oder nicht.

Damit $I = \sum_{j=1}^n \alpha_j b_j$ die Interpolationsbedingungen erfüllt, bestimmen wir die Gewichte α durch

$$\begin{aligned} I(\bar{\mathbf{x}}_j^D) &= t_j^D, \text{ für alle } j \in \{1, \dots, m\} \\ \Rightarrow \alpha_{j-1} b_{j-1}(\bar{\mathbf{x}}_j^D) + \alpha_j b_j(\bar{\mathbf{x}}_j^D) + \alpha_{j+1} b_{j+1}(\bar{\mathbf{x}}_j^D) &= t_j^D, \text{ für alle } j \in \{1, \dots, m\} \\ \Rightarrow \underbrace{\begin{pmatrix} 1 - \frac{\eta}{2} & \frac{\eta}{4} & & & & & \\ \frac{\eta}{4} & 1 - \frac{\eta}{2} & \frac{\eta}{4} & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \frac{\eta}{4} & 1 - \frac{\eta}{2} & \frac{\eta}{4} & \\ & & & & \frac{\eta}{4} & 1 - \frac{\eta}{2} & \\ & & & & & & \frac{\eta}{4} & 1 - \frac{\eta}{2} \end{pmatrix}}_{=:P} \alpha &= t^D \end{aligned}$$

Die Matrix M_{LS} (LS = linear smooth) ist diagonaldominant und symmetrisch positiv definit, daher invertierbar und wir erhalten $\alpha = M_{LS}^{-1} t^D$. Da M_{LS} Tridiagonalgestalt hat, sind die Gewichte mit Hilfe des Thomas-Algorithmus [79] effizient zu bestimmen. Ein solches Vorgehen weist Nachteile auf. Die so bestimmte Interpolante enthält Überschwinger, vergleiche Abbildung 5.3 und erfüllt somit auch nicht das Minimum-Maximum-Kriterium. Diese Überschwinger sind unter den Namen Gibbs-Phänomen bekannt [16]. Gilt für die Daten t^D die Bedingung $t^D \geq 0$ (elementweise), muss dies die Interpolante nicht mehr tun. Eine Beschreibung von konstanten Daten ist somit nicht gegeben. Als weiterer Nachteil ist anzuführen, dass die inverse Matrix M_{LS}^{-1} voll besetzt ist und dadurch der (nur) lokale Einfluss der Daten t^D auf die Interpolante verloren geht. Ändert sich ein Datum, so ändern sich alle Gewichte.

Diese Argumentation wird unterstützt durch den Extremfall $\eta = 0.5$. Hier hat die Basisfunktion b_0 keine linearen Anteile und stellt die Basis für quadratische Splines dar, die alle aufgezählten nachteiligen Eigenschaften auch besitzen. Der Parameter η kann also als Regler interpretiert werden, der die Länge der linearen Abschnitte zwischen quadratischen Splines steuert.

Diese Nachteile sind zu umgehen, wählen wir die Gewichte so wie im linearen Fall beschrieben, $\alpha = t^D$. Dies ist motiviert durch die Sicht, dass das vorliegende Verfahren nur eine geringe Modifikation der linearen Interpolation ist. Eine solche Wahl der Gewichte führt zum Auflösen der Interpolationseigenschaft. Anschaulich wird dies im Extremfall $t^D = 1$. Die Funktion I schreibt sich nun als $I = 1 \cdot b_1$ und $I(1) = 1 - \eta/2 \neq 1$, (vergleiche Abbildung 5.2)

Die so berechnete Funktion interpoliert Daten, wenn diese konstant in einem

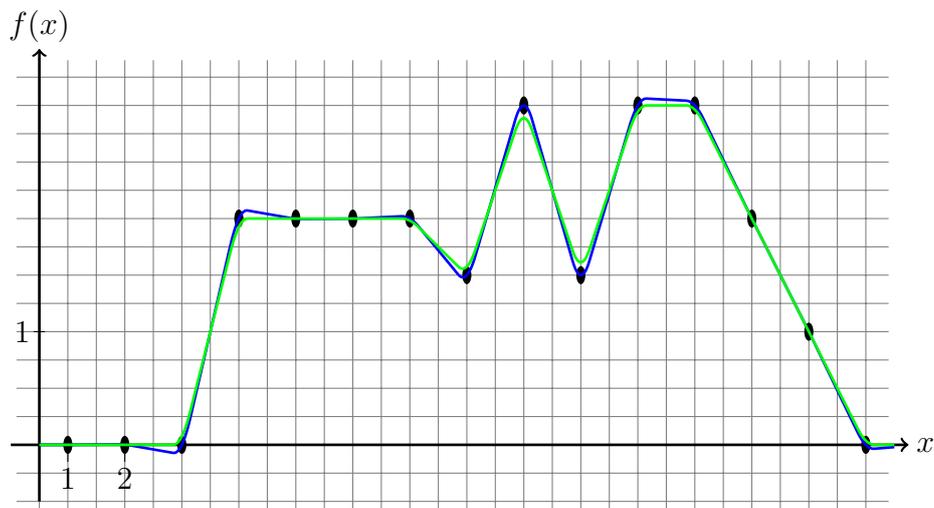


Abbildung 5.3: Vergleich der Methoden basierend auf linear glatter Interpolation; *schwarze Kreise*: Interpolationsdaten (X^D, t^D) ; *grüne Linie*: Graph der Interpolante mittels linear glatter Approximation $(\alpha = t^D)$; *blaue Linie*: Graph der Interpolante mittels linear glatter Interpolation $(\alpha = M^{-1}t^D)$

Bereich sind.

In Abbildung 5.3 haben wir die Ergebnisse beider Ansätze für die linear glatte Interpolation dargestellt. Deutlich ist das Überschwingverhalten der linear glatten Interpolation (blau) zu erkennen. Der grüne Graph ist durch linear glatte Approximation bestimmt. Im Bereich der Oszillation ist deutlich die Approximationseigenschaft zu erkennen. Im Bereich des Datenplateaus und im Bereich des konstanten Abfalls interpoliert die Funktion wie gezeigt.

Die zu interpolierenden Daten sind mit Bedacht gewählt. Sie enthalten für Bilddaten typische Strukturen: Einen Sprung, konstante Daten (Plateau), Oszillation und gleichmäßige Änderungen.

5.2.3 Interpolation mit kubischen Splines

Die Methodik der Spline-Interpolation ist in der Literatur ausführlich beschrieben [16, 73, 74], worauf wir uns auch hier kürzer fassen und nur Grundlegendes erwähnen, das später benötigt wird, und es in den Kontext der bisherigen Methoden stellen. Idee der Spline-Interpolation ist, dass die interpolierende Funktion I ein Maß minimiert, das die Glattheit der Funktion misst. Für den Fall der kubischen Splines wird die Interpolante bestimmt durch das Lösen des

Minimierungsproblems

$$\begin{aligned} \mathcal{S}^{spline}(I) &\rightarrow \min \quad \text{u.d.N.} \quad I(\mathbf{x}_j^D) = t_j^D, \quad j = 1, \dots, m \\ \text{und} \quad \mathcal{S}^{spline}(I) &= \int_{\Omega} (I''(x))^2 dx. \end{aligned} \quad (5.6)$$

Gesucht ist also eine Funktion I , die die gegebenen Daten interpoliert und die minimale zweite Ableitung besitzt. Das Maß \mathcal{S}^{spline} stellt eine Approximation an die Krümmungsenergie von I dar. Aus der Literatur ist weiterhin bekannt, dass dies Problem zuzüglich der (notwendigen) Hinzunahme von Randbedingungen, über die gleich zu schreiben ist, an I eindeutig lösbar ist. Die gesuchte Funktion lässt sich schreiben als $I = \sum \alpha_j b_j$ (womit sie in das vorgegebene Schema passt) mit der Basisfunktion

$$b_0(x) = \begin{cases} (2+x)^3, & \text{falls } x \in (-2, -1] \\ -x^3 - 2(x+1)^3 + 6(x+1), & \text{falls } x \in (-1, 0] \\ x^3 + 2(x-1)^3 - 6(x-1), & \text{falls } x \in (0, 1] \\ (2-x)^3, & \text{falls } x \in (1, 2) \\ 0, & \text{sonst} \end{cases} \quad (5.7)$$

und ihrer Ableitung

$$b'_0(x) = \begin{cases} 3(2+x)^2, & \text{falls } x \in (-2, -1] \\ -(9x+12)x, & \text{falls } x \in (-1, 0] \\ (9x-12)x, & \text{falls } x \in (0, 1] \\ -3(2-x)^2, & \text{falls } x \in (1, 2) \\ 0, & \text{sonst.} \end{cases} \quad (5.8)$$

Die Basisfunktion setzt sich also aus stückweise kubischen Polynomen zusammen, die zweimal stetig differenzierbar aneinanderstoßen, sie ist in Abbildung 5.2, unten dargestellt. Die Funktion hat als Träger das Intervall $(-2, 2)$, das heißt für ein beliebiges $x \in \mathbb{R}$ fällt die Summe auf vier von Null verschiedene Summanden zusammen. Das Intervall $(0.5, m+0.5)$ wird von $N = m+4$ Basisfunktionen überdeckt. Zur eindeutigen Bestimmung der Gewichte α werden also neben den m Interpolationsbedingungen noch weitere Bedingungen nötig (vergleiche [16]). Wir nutzen in dieser Arbeit die natürlichen Randbedingungen des Problems und erhalten zur Bestimmung der Gewichte

$$\underbrace{\begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 \end{pmatrix}}_{=:P} \alpha = t^D. \quad (5.9)$$

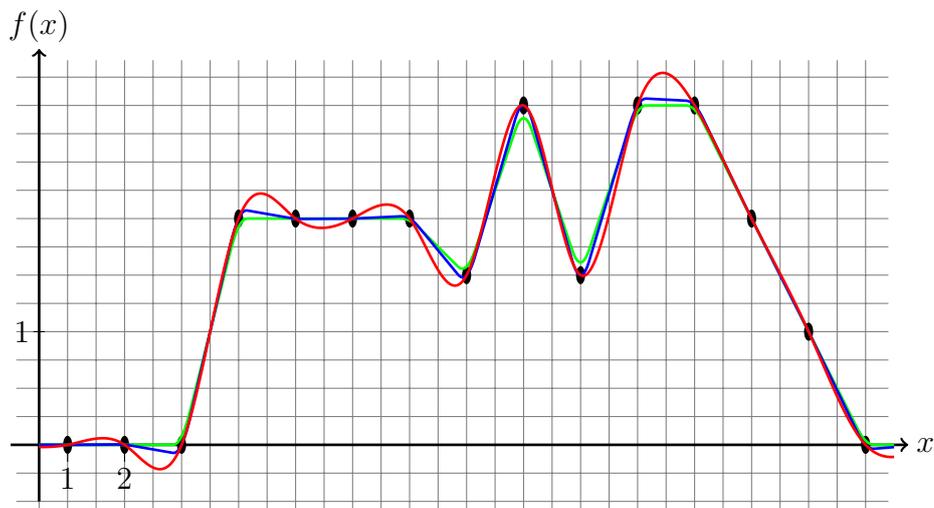


Abbildung 5.4: Vergleich der differenzierbaren Interpolationsmethoden; *schwarze Kreise*: Interpolationsdaten (X^D, t^D) ; *grüne Linie*: Graph der Interpolante mittels linear glatter Approximation; *blaue Linie*: Graph der Interpolante mittels linear glatter Interpolation; *rote Linie*: Graph der Interpolante mittels Spline-Interpolation

Die erwähnten Randbedingungen gehen in die erste und letzte Zeile der Matrix P ein. Die Matrix P ist symmetrisch und diagonaldominant.

Die Eigenschaften dieser Methode sind in [16, 73] ausführlich beschrieben. Wir fassen für uns zusammen: Die Gewichte der Interpolante sind schnell mittels des Thomas-Algorithmus [79] zu bestimmen, die Summe schnell auswertbar. Die Interpolante ist glatt (im Sinne des Glattheitsmaßes \mathcal{S}^{spline}), sogar glatter als die bisher beschriebenen Verfahren. Als Nachteile werten wir wie im vorangegangenen Fall das Ringing-Verhalten, das Nichteinhalten des Minimum-Maximum-Kriteriums und den globalen Einfluss eines Datums t_j^D auf die Interpolante. Um die Eigenschaften zu visualisieren, illustrieren wir das Interpolationsergebnis dieser Methode für die schon bekannten 1D-Daten in Abbildung 5.2.3. Deutlich sind Überschwinger zu erkennen.

5.2.4 Interpolation mit regularisierten Splines

Als letztes Verfahren beschreiben wir eine Erweiterung der Methode der kubischen Splines. Wir ersetzen hierzu die Forderung nach genauer Interpolation der Daten durch ein globales Ähnlichkeitsmaß [16], das den Abstand der Funktion I zu den Daten misst. Die Idee hinter diesem Vorgehen ist, global glatte Funktionen zu bestimmen, die die Daten nur noch approximieren und glatter sind als Funktionen, die durch reine Spline-Interpolation bestimmt werden. Im

Unterschied zur Methode der linear glatten Funktion haben hierbei die einzelnen Daten unterschiedlichen Einfluss auf die gesuchte Funktion I . Bei der linear glatten Interpolation werden die Daten so approximiert, dass die Funktionswerte $I(x_j^D)$ in einer η -Umgebung um die Werte t^D liegen.

Wir betrachten nun das Maß

$$\mathcal{D}(\alpha) = \|P\alpha - t^D\|^2. \quad (5.10)$$

Dies misst, wie gut gegebene Koeffizienten α für die oben beschriebenen Basisfunktionen die Daten (\mathbf{x}^D, t^D) treffen. Wir haben im letzten Abschnitt beschrieben, dass dieses Maß eine eindeutige Lösung hat, $\alpha = P^{-1}t^D$.

Das Verfahren der regularisierten Splines setzt auf oben beschriebenen Basisfunktionen auf. Die Wahl der Gewichte ändert sich. Wir bestimmen die Gewichte als Lösung eines Minimierungsproblems. Hierzu formulieren wir das Glattheitsmaß \mathcal{S}^{spline} (vergl. (5.6)) als Funktion in α :

$$\begin{aligned} \mathcal{S}^{spline}(I) &= \int_{\Omega} (I''(x))^2 dx = \int_{\Omega} \left(\sum_{j=1}^m \alpha_j b_j''(x) \right)^2 dx \\ &= \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j \int_{\Omega} b_i'' b_j''(x) dx = \alpha^\top M \alpha = \|\alpha\|_M \end{aligned} \quad (5.11)$$

mit der symmetrischen Toeplitz-Matrix M , auf deren Diagonalen der Vektor $(6, 0, -54, 96, -54, 0, 6)$ steht, das heißt

$$M = \begin{pmatrix} \ddots & \ddots \\ & 6 & 0 & -54 & 96 & -54 & 0 & 6 \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix},$$

denn

$$b_0''(x) = \begin{cases} 6(2+x), & \text{falls } x \in (-2, -1] \\ -(18x+12), & \text{falls } x \in (-1, 0] \\ (18x-12), & \text{falls } x \in (0, 1] \\ 6(2-x)^2, & \text{falls } x \in (1, 2) \\ 0, & \text{sonst} \end{cases} \quad (5.12)$$

und somit

$$\begin{aligned} \int_{\Omega} b_0'' b_0''(x) dx &= 96 & \int_{\Omega} b_0'' b_1''(x) dx &= -54 \\ \int_{\Omega} b_0'' b_2''(x) dx &= 0 & \int_{\Omega} b_0'' b_3''(x) dx &= 6. \end{aligned}$$

Die Matrix M enthält die Momente der Basisfunktionen, die oben aufgeführten Integrale über die zweiten Ableitungen. Daher wird das Verfahren auch ein Momentenbasiertes genannt.

Wir suchen nun Gewichte α , die die Funktion

$$\begin{aligned} \mathcal{D}(\alpha) &+ \theta \mathcal{S}^{spline}(\alpha) \rightarrow \min \\ \Rightarrow \|P\alpha - t^D\|^2 &+ \theta \alpha^\top M \alpha \rightarrow \min \end{aligned} \quad (5.13)$$

minimieren. Wir interpretieren die Funktion (5.13) als gewichtetes Mittel zwischen Approximation der Daten und Glattheit der Funktion I . Die Gewichtung wird über den Parameter θ gesteuert. Für $\theta = 0$ erhalten wir das originäre Spline-Problem zurück. Wählen wir θ groß, fällt der Einfluss des Datenmaßes \mathcal{D} gegenüber dem Glattheitsmaß und wir suchen eine Funktion, die global glatt ist, die Daten aber nur noch approximiert. Die Lösung des Minimierungsproblems (5.13), die Gewichte α , erhalten wir durch die dazugehörige Normalengleichung

$$(P^\top P + \theta M) \alpha = P^\top t^D. \quad (5.14)$$

Für die Bestimmung der Lösung eines Minimierungsproblems mittels Normalengleichungen vergleiche [28].

Die Struktur der Gleichung zeigt, wieso wir von einem regularisierten Verfahren sprechen. Die Matrix M wirkt als Regularisierer, gewichtet mit dem Parameter θ auf das originäre Spline-Problem.

Veranschaulicht wird dies am Beispiel der Daten in Abbildung 5.5. Für kleine Werte von θ , $\theta \in \{0, 0.1\}$ sind lokale Eigenschaften der Daten, die Oszillation Mitte rechts und das Plateau Mitte links, sichtbar. Für grobe Skalen, $\theta \in \{10, 1000\}$ sind diese Eigenschaften, die die Daten vorgeben, nicht mehr durch die Funktion I repräsentiert. Dennoch bleiben Eigenschaften, wie das Ansteigen der Werte zur Intervallmitte in I sichtbar.

Wir fassen zum Ende dieses Abschnittes die algorithmischen Gemeinsamkeiten der vorgestellten Methoden zusammen. Aussagen, wann welche Interpolationsmethode in der Registrierung von Nutzen ist, werden wir am Ende des Kapitels tätigen.

Wir haben verschiedene Methoden vorgestellt und sind in der Lage, diese Verfahren in einem Rahmen zu präsentieren. Jedes Verfahren lässt sich als gewichtete Summe einer jeweils verschobenen Basisfunktion schreiben. Die dazugehörigen Gewichte werden mittels eines linearen Gleichungssystems mit der Matrix P bestimmt. Die Verfahren unterscheiden sich in der Wahl der Basisfunktionen und - damit zusammenhängend - in der Wahl der Matrix P .

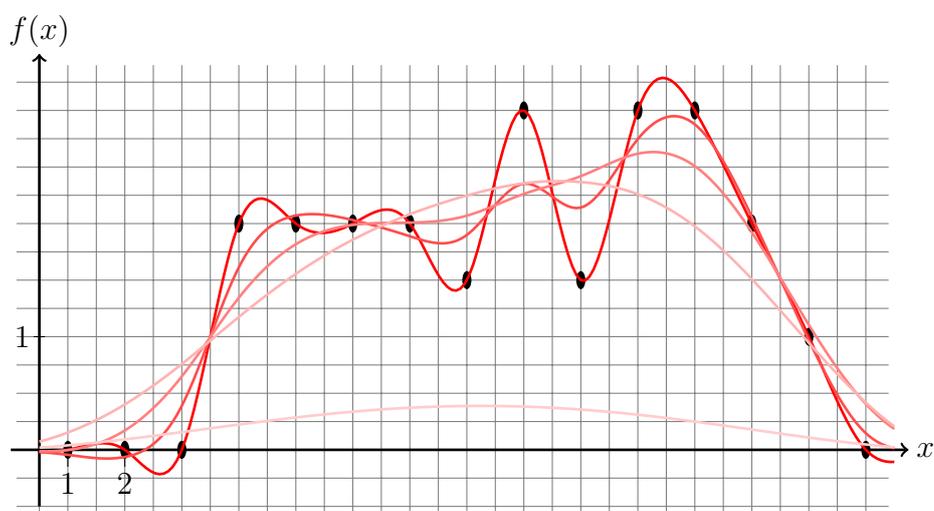


Abbildung 5.5: Darstellung des Einflusses des Regularisierungsparameters θ bei der Interpolation mittels regularisierter Splines; *schwarze Kreise*: Interpolationsdaten (\mathbf{x}^D, t^D) ; rote Linien: Von intensiv nach blass steigt der Wert für θ ; $\theta \in \{0, 0.1, 1, 10, 1000\}$

5.3 Interpolation im Dreidimensionalen

Befassen wir uns nun damit, aus den ausführlich beschriebenen 1D-Interpolationsverfahren Methoden für höher dimensionale Daten zu entwickeln. Zur Vereinfachung der Notation legen wir uns auf Verfahren für dreidimensionale Daten fest, da diese zum einen in späteren Kapiteln die Hauptlast der anfallenden Datensätze darstellen und zum anderen die Techniken zur Dimensionsanpassung auf zwei, vier oder noch höher dimensionale Daten anwendbar sind.

Wir betrachten Positions-Werte-Paare $(\bar{\mathbf{x}}_j^D, t_j^D) \in \mathbb{R}^4$, $j = 1, \dots, n^D$, vergleiche Abschnitt 5.1. Nach der dort beschriebenen Koordinatentransformation gilt für ein $j \in \{1, \dots, n^D\}$, $\bar{\mathbf{x}}_j^D = (i, k, l) \in \mathbb{N}^3$. Die Positionen der gegebenen Daten sind Tupel natürlicher Zahlen.

Wir nutzen einen Tensoransatz zur Formulierung der Interpolationsmethoden. Mit jedem gegebenen Datum wiederum verknüpfen wir eine Basisfunktion. Zur Position $\bar{\mathbf{x}}_j^D = (i, k, l) \in \mathbb{N}^3$ die Funktion

$$b_{ijk} : \mathbb{R}^3 \rightarrow \mathbb{R}.$$

Wir behalten die Darstellung der Interpolante als gewichtete Summe verschobener Basisfunktionen bei.

Zwei Fragen werden wir in den nächsten Abschnitten beantworten:

1. Wie wählen wir die dreidimensionale Basisfunktion?
Und damit direkt verbunden: Wie leiten wir aus den eindimensionalen Funktionen Basen für drei Dimensionen her?

2. Wie werden die Gewichte $\alpha \in \mathbb{R}^{n^D}$ bestimmt?

Wir werden die Fragen am Beispiel der linearen Interpolation beantworten. Dies stellt keine Einschränkung dar, vereinfacht aber die Notation.

Bei der Verwendung eines Tensoransatzes wird die mehrdimensionale Basisfunktion so gewählt, dass sie sich in der Richtung der Koordinatenachsen so verhält, wie die eindimensionalen Funktionen. Mit $x = (x_1, x_2, x_3) \in \mathbb{R}^3$ gilt also, dass sich die Basisfunktion an der Position (i, k, l) schreiben lässt als

$$b_{ijk}(x) = b_i(x_1) \cdot b_k(x_2) \cdot b_l(x_3).$$

Hierbei sind $b_i, b_k, b_l : \mathbb{R} \rightarrow \mathbb{R}$ die eindimensionalen Basisfunktionen. Mit der Verschiebe-Eigenschaft der eindimensionalen Funktion gilt dann

$$b_{ijk}(x) = b_0(x_1 - i) \cdot b_0(x_2 - k) \cdot b_0(x_3 - l).$$

Die dreidimensionale Basisfunktion separiert also in den einzelnen Raumdimensionen,

$$I = \sum_{l=1}^{m_3} \sum_{k=1}^{m_2} \sum_{i=1}^{m_1} \alpha_{ijk} b_{ijk}. \quad (5.15)$$

Zur vollständigen Beschreibung einer 3D-Interpolationsmethode fehlt die Bestimmung der Gewichte α . Hier nutzen wir aus, dass die Basisfunktionen b_{ijk} als Produkt von 1D-Basisfunktionen geschrieben werden, deren Gewichtsbestimmung wir kennen. Um eine kompakte algebraische Schreibweise zu erhalten, sortieren wir die Summe $\sum_{k=1}^{N_3} \sum_{j=1}^{N_2} \sum_{i=1}^{N_1}$ um. Jede Basisfunktion (und jedes Gewicht) korrespondiert mit einer Raumposition auf einem Cellcentered-Gitter und für solche Gitter haben wir in Kapitel 4 eine Durchlaufreihenfolge festgelegt. Nutzen wir diese jetzt aus, erhalten wir einen Vektor $\alpha \in \mathbb{R}^{n^D}$ und einen Vektor mit Basisfunktionen $b = (b_j)_{j=1}^{n^D}$. Durch die Kronecker-Struktur des Gitters \mathbf{x}^D und die gewählte Sortierreihenfolge erhalten wir eine einfache Gleichung zur Bestimmung der Gewichte. Hierzu seien $P_{m_1}, P_{m_2}, P_{m_3}$ die Matrizen P vom gewählten Interpolationsverfahren. Die Indizes korrespondieren zu den Dimensionen der Matrizen. (Die vollständige Notation wäre $P_{m_j^D}$, im Subskript lassen wir den Superskript zur Übersicht weg.) Wir erhalten

$$(P_{m_3} \otimes I_{m_2} \otimes I_{m_1}) (I_{m_3} \otimes P_{m_2} \otimes I_{m_1}) (I_{m_3} \otimes I_{m_2} \otimes P_{m_1}) \alpha = t^D. \quad (5.16)$$

Zu interpretieren ist diese Gleichung (5.16) wie folgt: Wir bestimmen zuerst die Gewichte für die Interpolation der Daten entlang der x_1 -Raumrichtung (die Zeilen des Datenvolumens), dies für jede Spalte und jede Schicht. Diese Gewichte werden dann sukzessiv in die Raumrichtungen x_2, x_3 interpoliert.

Auf die Besonderheiten im Falle der Interpolation mit regularisierten Splines gehen wir gleich ein.

Durch die Wahl einer der 1D-Interpolationsmethoden können wir nun eine Methode zur Interpolation von dreidimensionalen Daten ableiten.

Dazu wollen wir noch drei Anmerkungen und ein Beispiel geben:

Übertragen der 1D-Eigenschaften

Durch den gewählten Ansatz (5.15), (5.16) übertragen sich Eigenschaften der eindimensionalen Interpolante auf höhere Dimensionen. Zwischen zwei benachbarten Datenpunkten ist die höherdimensionale Interpolante gleich der Eindimensionalen.

Linear glatte Interpolation

Im 1D-Fall wird in einer Umgebung um die Datenpunkte quadratisch approximiert. Durch den in (5.15) gewählten Ansatz erhalten wir im höherdimensionalen Fall eine Gitterstruktur, die Aussagen über die jeweilige Interpolationsart gibt. Für den zweidimensionalen Fall ist dies in Abbildung 5.6 illustriert. Die orange gefärbten Flächen geben die Bereiche an, in denen in x_1 -Richtung quadratisch approximiert wird, die grünen Bereiche analog für die x_2 -Richtung. An den Kreuzungspunkten wird in beide Richtungen quadratisch approximiert.

Im dreidimensionalen Fall ergeben sich je Raumrichtung Flächen, in die in eine Richtung quadratisch approximiert wird, diese schneiden sich in Geraden, in denen in je zwei Raumrichtungen quadratisch approximiert wird, und die wiederum in Kuben, in denen in drei Raumrichtungen quadratisch approximiert wird.

Ein Wort zur **Implementierung** dieser Methode für drei Dimensionen:

Der in (5.15) und (5.16) vorgestellte Weg ist zur Bestimmung von Funktionswerten $I(\mathbf{x})$ numerisch zu aufwendig. Im Eindimensionalen haben wir gezeigt, dass für ein beliebiges $x \in \Omega$ drei Basisfunktionen in $\sum_j \alpha_j b_j$ von Null verschieden sind. Im dreidimensionalen Fall werden 27 Basisfunktionsauswertungen je Bestimmung eines Funktionswertes benötigt.

Wir empfehlen eine Unterscheidung in acht Fälle zur Bestimmung von Funktionswerten. Fall eins, die Urbilder liegen im Gebiet, in dem linear interpoliert wird, Fälle zwei bis vier, in einer Richtung wird quadratisch approximiert, Fälle fünf bis sieben, in zwei Richtungen wird quadratisch approximiert, Fall acht, in alle drei Richtungen. In MATLAB kann dies effizient mit Hilfe der `find`-Funktion umgesetzt werden.

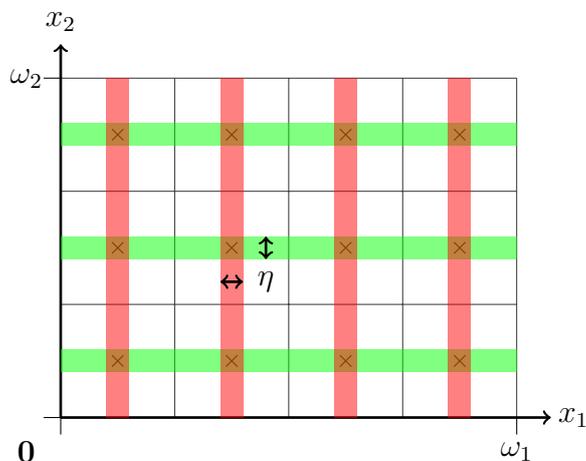


Abbildung 5.6: Aufteilung des Gebietes Ω für die linear-glatte Interpolation in 2D, *rot*: Bereiche, in denen in x_1 -Richtung approximiert / interpoliert wird; *grün*: Bereiche, in denen in x_2 -Richtung approximiert / interpoliert wird

Methode der regularisierten Splines

Für die Methode der regularisierten Splines gestaltet sich das Bestimmen der Gewichte α anders als in (5.16) beschrieben, da das 1D-System (5.14) von den anderen 1D-Systemen abweicht. Die grundlegende Idee der sukzessiven Anwendung der 1D-Systeme ist jedoch die gleiche. Sie liest sich in Matrix-Vektor-Notation:

$$\begin{aligned} (I_{m_3} \otimes I_{m_2} \otimes P_{m_1}^\top P_{m_1} + \theta M_1) \bar{\alpha} &= (I_{m_3} \otimes I_{m_2} \otimes P_{m_1}) t^D \\ (I_{m_3} \otimes P_{m_2}^\top P_{m_2} + \theta M_2 \otimes I_{m_1}) \bar{\alpha} &= (I_{m_3} \otimes P_{m_2} \otimes I_{m_1}) \bar{\alpha} \\ (P_{m_3}^\top P_{m_3} + \theta M_3 \otimes I_{m_2} \otimes I_{m_1}) \alpha &= (P_{m_3} \otimes I_{m_2} \otimes I_{m_1}) \bar{\alpha}. \end{aligned}$$

Bevor wir den Inhalt des Kapitels zusammenfassen, wenden wir uns zwei Themen zu, die wir für den folgenden Teil der Arbeit benötigen und die wir mit dem bisher Vorgestellten beschreiben können.

5.4 Diskrete Bilder und Ableitungen

Wir präsentieren als erstes, wie wir aus kontinuierlichen Bilddaten mit Hilfe der in 4 vorgestellten Gitter diskrete Bilder mit beliebiger, vom Benutzer festgelegter Auflösung bestimmen, wie eine diskrete Transformation bestimmt ist und wie wir Bilder an deren Positionen auswerten. Wir erläutern die im weiteren verwendete Notation und zeigen, wie sich die für die Optimierung wichtige Ableitung eines Bildes nach einer gegebenen Transformation bestimmen lässt.

Wir diskretisieren das Gebiet Ω_R durch ein Cellcentered-Gitter $\mathbf{x} = \mathbf{x}^{cc}(\Omega_R, m)$ mit $m = (m_1, m_2, m_3) \in \mathbb{N}^3$ Diskretisierungspunkten und $n = m_1 m_2 m_3$ Zellen. Damit ist $\mathbf{x} \in \mathbb{R}^{3n}$. Zu beachten ist, dass die gewählte Diskretisierung m unabhängig von der Anzahl der Voxel m^D ist, die den Datensatz beschreiben. Durch Interpolation erhalten wir eine beliebig auszuwertende Funktion \mathcal{R} und wir setzen

$$R = \mathcal{R}(\mathbf{x}) = \mathcal{R}(\bar{\mathbf{x}}) \in \mathbb{R}^n.$$

Der Vektor R enthält also elementweise den Wert der Interpolante \mathcal{R} an den Positionen aus $\bar{\mathbf{x}}$.

Das kontinuierliche Bild wird im Registrierungsprozess an den Stellen $\mathbf{x} = \mathbf{x}^{cc}(\Omega_R, m)$ ausgewertet, das R zugrunde liegende Gitter ist Argument der Transformation y . Das diskrete Deformationsfeld ist durch die Auswertung einer kontinuierlichen Deformation an Positionen eines gegebenen Gitters gegeben. Es ist

$$\bar{\mathbf{y}} = y(\bar{\mathbf{x}}) \in \mathbb{R}^{n \times 3} \quad \text{und} \quad \mathbf{y} \in \mathbb{R}^{3n}.$$

Damit können wir schreiben

$$\mathcal{T}(y(\bar{\mathbf{x}})) =: \mathcal{T}_{\bar{\mathbf{y}}} \in \mathbb{R}^n.$$

Als Zielgröße für die Optimierung suchen wir eine Deformation, die das deformierte Templatebild $\mathcal{T}_{\bar{\mathbf{y}}}$ ähnlich zu R transformiert. Daher ist von Interesse, wie die Ableitung des deformierten Bildes nach der Transformation \mathbf{y} gegeben ist. Es ist

$$\mathcal{T}_{\bar{\mathbf{y}}} = \mathcal{T}(\bar{\mathbf{y}}) = \begin{pmatrix} \mathcal{T}(\mathbf{y}_1, \mathbf{y}_{1+n}, \mathbf{y}_{1+2n}) \\ \mathcal{T}(\mathbf{y}_2, \mathbf{y}_{2+n}, \mathbf{y}_{2+2n}) \\ \vdots \\ \mathcal{T}(\mathbf{y}_n, \mathbf{y}_{2n}, \mathbf{y}_{3n}) \end{pmatrix}$$

und damit gilt für die Differentiation nach $\mathbf{y}_{1:n}$

$$d_{\mathbf{y}_{1:n}} \mathcal{T}_{\bar{\mathbf{y}}} = \begin{pmatrix} \partial_{x_1} \mathcal{T}(\mathbf{y}_1, \mathbf{y}_{1+n}, \mathbf{y}_{1+2n}) & & & & \\ & \partial_{x_1} \mathcal{T}(\mathbf{y}_2, \mathbf{y}_{2+n}, \mathbf{y}_{2+2n}) & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \partial_{x_1} \mathcal{T}(\mathbf{y}_n, \mathbf{y}_{2n}, \mathbf{y}_{3n}) \end{pmatrix},$$

für die Differentiation nach $\mathbf{y}_{n+1:2n}$

$$d_{\mathbf{y}_{n+1:2n}} \mathcal{T}_{\bar{\mathbf{y}}} = \text{diag}(\partial_{x_2} \mathcal{T}(\bar{\mathbf{y}}))$$

und für die Differentiation nach $\mathbf{y}_{2n+1:3n}$

$$d_{\mathbf{y}_{2n+1:3n}} \mathcal{T}_{\bar{\mathbf{y}}} = \text{diag}(\partial_{x_3} \mathcal{T}(\bar{\mathbf{y}})).$$

Hierbei ist für einen gegebenen Vektor $p \in \mathbb{R}^n$

$$\text{diag}(p) = \begin{pmatrix} p_1 & & \\ & \ddots & \\ & & p_n \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

Zusammenfassend gilt

$$d\mathbf{T}_y = (\text{diag}(\partial_{x_1} \mathcal{T}(\bar{\mathbf{y}})), \text{diag}(\partial_{x_2} \mathcal{T}(\bar{\mathbf{y}})), \text{diag}(\partial_{x_3} \mathcal{T}(\bar{\mathbf{y}}))) \in \mathbb{R}^{n \times 3n}.$$

Bevor wir zum zweiten Punkt kommen, werden wir Testdaten präsentieren, die wir im weiteren Verlauf zur einfachen Illustration nutzen können.

5.5 Testdaten

Wir nutzen zweidimensionale Daten, um die Darstellung innerhalb der Arbeit nicht zu verkomplizieren. Dabei sind die Testdaten so gewählt, dass sie sich Idealfällen von Ultraschall und CT annähern und ähnlichen Bildinhalt (Gefäßstruktur) haben.

Wir geben die Daten vor, indem wir Gefäßbäumen mittels Gefäßmittellinien und -radien beschreiben. In der Abbildung 5.7, oben links, sind diese Linien für die Testdatensätze gezeigt. Rote Punkte geben die Anfangs- und Endpunkte an.

In der Abbildung, oben rechts, sind die Binärmasken gezeigt, die die durch Linien und Radien erzeugten Gefäßbäume wiedergeben. Hierbei sind die Ränder des Gefäßbaumes rot, die des Tumors violett. Es ist die Position eines Tumors vorgegeben, dessen Lage auch in den Bildern wiedergegeben ist. Zur Erzeugung von Grauwertbildern wird für die drei unterschiedlichen Bereiche (Gefäße, Tumore, Hintergrund) jeweils ein mittlerer Grauwert und eine Standardabweichung der Grauwerte vorgegeben. Diese sind in der folgenden Tabelle wiedergegeben:

	Gefäße mG + STD	Tumore mG + STD	Hintergrund mG + STD
Ultraschall	200 ± 15	100 ± 15	50 ± 50
Computertomographie	20 ± 15	100 ± 15	200 ± 50

Die Werte sind so gewählt, dass die entstehenden Bilder in den Grauwerten Ähnlichkeit zu der Bildgebung der jeweiligen Modalität besitzen.

Der Satz der Kontrollpunkte für Gefäßbaum und Tumor ist nicht-linear transformiert worden. Aus dem ursprünglichen Datensatz entsteht ein Pseudo-CT-Datensatz; aus den deformierten Daten erzeugen wir einen Pseudo-Ultraschall-Datensatz. Sie sind in der Abbildung 5.7, unten dargestellt.

Der gewählte Weg einer parametrischen Repräsentation der Testdaten hat mehrere Vorteile: Wir können die Bildauflösung frei wählen. (Die verwendeten Datensätze haben die Auflösung 256×256) und wir können sowohl monomodale als auch multimodale Bildpaare nutzen. Für die Testdaten ist $\Omega = (15, 10)$, mit $m = (256, 256)$ ergibt sich eine Auflösung von $h \approx (0.05859, 0.03906)$.

5.6 Bildbereiche

Der Ansatz erlaubt die Verwendung unterschiedlich großer Bildbereiche Ω_T, Ω_R , wie sie in der medizinischen Praxis dann vorkommen, wenn Bilder unterschiedlicher Aufnahmegeräte registriert werden sollen. Möglich ist dies, da wir von den diskreten Bilddaten weg- und hin zu einer kontinuierlichen Funktion \mathcal{T} gehen.

Das Vorgehen ist das Folgende: Der Datensatz, der einen Ausschnitt des anderen zeigt, wird als Referenz verwendet, womit die Informationen des Referenzbildes eine Untermenge des Templatebildes bilden. Dies ist in der Praxis der Ultraschall-CT-Registrierung ein typisches Problem. Das CT-Bild enthält Informationen über das komplette Operationsgebiet samt benachbarter Strukturen. Im Ultraschallbild findet sich nur ein Ausschnitt dieser Informationen, nämlich die, die im vom Operateur mittels Ultraschall dargestellten Bereich liegen, wieder. Ziel der Registrierung soll es hier sein, die zu den Informationen im Ultraschallbild korrespondierenden Punkte im CT-Bild zu finden und zuzuordnen. Hierbei ist das geschaltete Gebiet Ω_R kleiner als das Übersichtsgebiet Ω_T . Die Wahl des kleineren Gebietes als Referenzgebiet ist plausibel, da zu diesem die korrespondierenden Strukturen gefunden werden sollen und das Templatebild somit eine Obermenge dieser Informationen enthält. Zu beachten ist hierbei, dass nicht das Templatebild vollständig deformiert wird, sondern im Bildbereich Ω_T die Positionen zu den Punkten \mathbf{x} aus Ω_R korrespondierenden Punkte gefunden werden.

Dies soll das folgende Beispiel illustrieren:

Wir nutzen die im vorherigen Absatz beschriebenen Testdaten. In Abbildung 5.8, rechts, ist als Bildhintergrund der Templatedatensatz gezeigt, für diesen gilt, $\omega = (15, 10)$ und die gegebene Auflösung beträgt $h = (0.0586, 0.0391)$. Als Referenz nutzen wir einen nichtlinear deformierten Ausschnitt aus dem eigentlichen Referenzdatensatz. Dieser Datensatz zeigt ein Gebiet der Größe $\omega = (2.5, 2.5)$. Die gegebene Auflösung beträgt $h = (0.0195, 0.0195)$. Die Datensätze unterscheiden sich also in ihrer Auflösung und in der Größe des repräsentierten (physikalischen) Gebietes. In die gleiche Abbildung ist in den Templatedatensatz eine Deformation \mathbf{y} eingezeichnet, die ein Registrierungsergebnis zwischen den beiden Bildern darstellt.

Offensichtlich ist, dass nur ein Bereich des Templatebildes in die Korrespondenz

einfließt. Wir erhalten keine Deformation des vollständigen Templatebildes. Dies wäre aufgrund der Informationen in R auch nicht möglich, da dieses Bild nur Informationen über einen Teilbereich von T enthält.

Fazit

Mit den beschriebenen Interpolationsmethoden haben wir die Möglichkeit geschaffen, für diskrete Daten eine kontinuierliche Funktion zu bestimmen und damit ein kontinuierliches Registrierungsproblem zu beschreiben. Des Weiteren können wir Bilder nun an jeder beliebigen Stelle ihres Urbildes auswerten.

Dazu haben wir vier Verfahren beschrieben, von denen die Methode der linear-glatten Interpolation hier erstmalig ausführlich vorgestellt wurde.

Wir werden die Interpolationsmethoden in der Arbeit wie folgt verwenden: Die lineare Interpolation nutzen wir, wenn wir schnelle Interpolation benötigen oder die Datenmenge so groß ist, dass auf herkömmlichen Desktop-Rechnern (bsp. iMac 2.8 Ghz Intel Core Duo mit 4GB RAM) die Interpolation mittels Splines zuviel Rechenzeit und Speicherplatz beanspruchen würde. Ist die Differenzierbarkeit der Interpolante gefordert, wechseln wir zu linear-glatter Interpolation. Die regularisierte Spline-Interpolation werden wir für die zweidimensionalen Testdaten nutzen und mit ihr in Kapitel 10 ein Multiskalenverfahren herleiten.

Mit der Beschreibung der Testdaten haben wir einen Kompromiss zwischen einfacher Darstellbarkeit von Ergebnissen und Komplexität der Daten geschlossen.

Mit den bisher erarbeiteten Punkten - diskrete Gitter und Interpolation - sind wir in der Lage, im nächsten Kapitel Distanzmaße zum Messen von Ähnlichkeit zwischen Bildern zu beschreiben.

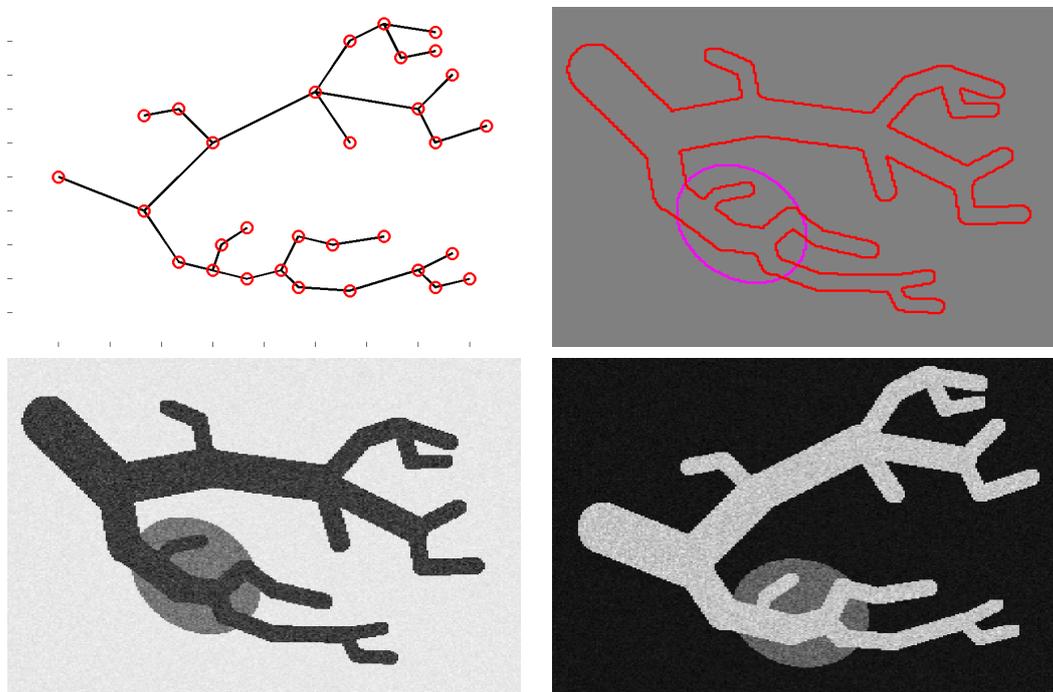


Abbildung 5.7: Zweidimensionale Testdaten



Abbildung 5.8: Beispiel für unterschiedliche Bildbereiche; *links*: Referenzbild R mit $\Omega_R = (2.5, 2.5)$; *rechts*: Templatebild T mit $\Omega_T = (15, 10)$ und dem Registrierungsergebnis \mathbf{y} (*rot*)

Kapitel 6

Distanzmaße

Wir wenden uns nun dem nächsten Baustein der Registrierung zu: der Ähnlichkeit zwischen zwei Bildern (oder andersherum gedacht: dem Unterschied zwischen zwei Bildern). Grundlegende (und nicht neue) Idee ist es, ein Maß zu formulieren, das die Distanz zweier gegebener Bilder misst. Konvention soll in dieser Arbeit sein, dass eine große Ähnlichkeit (oder eine geringe Distanz) durch kleine Werte des Maßes ausgedrückt wird.

Wir stellen im Weiteren drei Distanzmaße vor. Unter einem Distanzmaß verstehen wir eine differenzierbare Funktion in der gesuchten Transformation, die so beschaffen ist, dass sie, wenn Referenz und deformierbares Template sich ähnlich sind, klein, wenn sie sich unähnlich sind, groß ist. Bei Gleichheit der Bilder soll sie minimal sein. Die Unterschiede zwischen den Maßen entstehen durch unterschiedliche Interpretation des Begriffes *Ähnlichkeit*.

Ziel ist die numerische Minimierung des Maßes mit den in Kapitel 3 vorgestellten Methoden. Da die Distanz zwischen Referenz und deformiertem Template bestimmt wird, tritt in der ersten Ableitung des Maßes aufgrund der Kettenregel der Term $d_1 T_y$, in der zweiten Ableitung der Term $d_2 T_y$ auf. Um die Nutzung zweiter Bildableitungen zu vermeiden, nutzen wir Gauß-Newton-Verfahren (Kapitel 3.2.3), die die zweite Ableitung approximieren.

Wir verwenden weiterhin die Notation der vorherigen Kapitel; damit erhalten wir als allgemeine Formulierung des diskreten Distanzmaßes:

$$\begin{aligned} \mathcal{D} : \mathbb{R}^{3n} &\rightarrow \mathbb{R} \\ \mathbf{y} &\mapsto \mathcal{D}(\mathbf{y}) = \varphi(r(\mathbf{y})) \end{aligned}$$

mit den Funktionen

$$\begin{aligned} \varphi : \mathbb{R}^{\tilde{n}} &\rightarrow \mathbb{R} \\ \text{und } r : \mathbb{R}^{3n} &\rightarrow \mathbb{R}^{\tilde{n}}. \end{aligned}$$

Daraus ergibt sich mit der Kettenregel für die Ableitungen

$$\begin{aligned} d\mathcal{D}(\mathbf{y}) &= d\varphi(r(\mathbf{y})) \cdot dr(\mathbf{y}) \\ \text{und } d_2\mathcal{D}(\mathbf{y}) &\approx dr(\mathbf{y})^\top \cdot d_2\varphi(r(\mathbf{y})) \cdot dr(\mathbf{y}). \end{aligned}$$

Damit müssen wir für jedes Maß nur die Funktionen φ , $d\varphi$, $d_2\varphi$, r und dr beschreiben.

Da alle Maße in der Literatur bereits beschrieben sind, fassen wir uns in ihrer Motivation und Interpretation kurz. Wir stellen vielmehr einen allgemeinen Rahmen vor, in den sie sich fassen lassen und in dem sie modular austauschbar sind.

Wir beginnen mit dem wohl ältesten Distanzmaß der Bildregistrierung:

6.1 Summe der quadrierten Differenzen

6.1.1 Definition

Das Maß der Summe der quadrierten Differenzen (SSD) basiert auf dem punktweisen Vergleich der Grauwerte von (deformiertem) Templatebild und Referenzbild. In seiner linearisierten Form ist es bereits in den Arbeiten von Horn und Schunk [36] und Lukas und Kanade [50] im Jahr 1981 verwendet worden. Beide Arbeiten gelten als historische Grundlagen der Bewegungserkennung, deren Gebiet mit der Bildregistrierung eng verwandt ist. Das Maß ist durch

$$\mathcal{D}^{ssd}[\mathcal{T}, \mathcal{R}, y] = \frac{1}{2} \int_{\Omega} (\mathcal{T}(y(x)) - \mathcal{R}(x))^2 dx \quad (6.1)$$

gegeben. Es basiert auf der Annahme, dass zwei Bilder ähnlich sind, wenn ihr Differenzbild verschwindet.

6.1.2 Diskretisierung

Zur Diskretisierung des Maßes nutzen wir die Mittelpunktsregel [57]. Zur Anschauung verwenden wir den 1D-Fall. Sei $\Omega = (0, \omega)$ gegeben und m die Anzahl der Diskretisierungspunkte. Dann ist $\mathbf{x} = \mathbf{x}^{cc}(\omega, m)$ ein Cellcentered-Gitter mit der Zellgröße $h = \omega/m$ und wir können das Integral $\int_{\Omega} f(x) dx$ approximieren durch

$$\int_{\Omega} f(x) dx \approx h \sum_{j=1}^m f(\mathbf{x}_j) = Q_h(f).$$

Für den Fall, dass wir ein Integral über eine quadrierte Funktion bestimmen, können wir schreiben $\int_{\Omega} (f(x))^2 dx \approx h \sum_{j=1}^m f(\mathbf{x}_j)^2 = h \|\mathbf{f}\|^2$, wobei $\mathbf{f} \in \mathbb{R}^m$

der Vektor $(f(x_j))_{j=1}^m$ der Funktionswerte von f an den Positionen des Gitters ist. Mit den im vorherigen Kapitel definierten Größen T_y, R erhalten wir

$$\frac{1}{2} \int_{\Omega} (\mathcal{T}(y(x)) - \mathcal{R}(x))^2 dx \approx \frac{\bar{h}}{2} \|r(\mathbf{y})\|^2 =: \varphi(r(\mathbf{y}))$$

mit $\varphi : \mathbb{R}^{3n} \rightarrow \mathbb{R}$ und

$$\begin{aligned} r : \mathbb{R}^{3n} &\rightarrow \mathbb{R}^{3n} \\ \mathbf{y} &\mapsto r(\mathbf{y}) = T_y - R. \end{aligned}$$

Die Funktion φ bildet ein diskretes Gitter auf den Wert des diskretisierten Maßes ab, die Funktion r bildet einen Punkt des Gitters auf den lokalen Beitrag zum Maß ab.

Die Ableitungen sind durch

$$d\varphi(r(\mathbf{y})) = \bar{h} \cdot r(\mathbf{y})^\top \cdot dr(\mathbf{y}), \quad d_2\varphi(r(\mathbf{y})) = dr(\mathbf{y}) \bar{h} dr(\mathbf{y}) \quad \text{und} \quad dr(\mathbf{y}) = dT_y.$$

gegeben.

Das Maß der quadrierten Differenzen ist Standardmaß der Bildregistrierung. Es ist schnell auszuwerten und robust gegenüber Rauschen. Da es auf dem Vergleich der Bildgrauwerte basiert, ist es nicht für die Registrierung zweier Bilder geeignet, bei denen korrespondierende Strukturen durch unterschiedliche Grauwerte repräsentiert werden. Dies ist bei der Registrierung von Ultraschall und CT-Daten aber gegeben, so dass dieses Maß hierfür nicht genutzt werden kann.

6.2 Mutual Information

6.2.1 Definition

Als zweites Distanzmaß stellen wir das Maß der *Mutual Information* (MI) vor. Dieses Maß wurde von P. A. Viola in [83, 84] präsentiert. Es stellt mittlerweile das Standardmaß für die Registrierung von multimodalen Bildern dar. Für das Maß der Mutual Information gibt es in der Literatur verschiedene Definitionen, für eine Übersicht vergleiche [33]. Wir geben hier die Definition unter der Verwendung der Bild-Entropien an. Sei $I : \Omega \rightarrow G \subset \mathbb{R}^k$, $k = 1, 2$ eine kontinuierliche Funktion, die nach P_I verteilt ist und deren dazugehörige kontinuierliche Dichte ρ_I ist, dann ist durch

$$H(I) = \int_G \rho_I(g) \log(\rho_I(g)) dg$$

mit der Konvention $\log(0) := 0$ die Entropie des Bildes I gegeben.

Die Mutual Information der gegebenen Bilder \mathcal{R}, \mathcal{T} wird durch

$$\text{MI}(\mathcal{R}, \mathcal{T}) = H(\mathcal{R}) + H(\mathcal{T}) - H((\mathcal{R}, \mathcal{T})) \quad (6.2)$$

beschrieben. Die Mutual Information kann als Maß für die stochastische Abhängigkeit der gegebenen Bilder interpretiert werden. Es ist Null, wenn die Bilder stochastisch unabhängig sind, nach oben i. A. aber nicht beschränkt. Um in Kontext der zu minimierenden Zielfunktion zu bleiben, wird das Distanzmaß durch

$$\mathcal{D}^{MI}(\mathcal{R}, \mathcal{T}, y) = -\text{MI}(\mathcal{R}, \mathcal{T}(y)) \quad (6.3)$$

definiert.

6.2.2 Diskretisierung

Wir wählen einen Ansatz, der in den Rahmen der in dieser Arbeit verwendeten Optimierungsstrategie passt, also eine Implementierung von Mutual Information mittels Gauß-Newton-Verfahren.

Das Distanzmaß hat als äußere Funktion φ keine Norm (wie das SSD-Maß und das folgende NGF-Maß), sondern die Funktion

$$\varphi(\mathbf{y}) = \rho_R^\top \cdot \log(\rho_R + \text{tol}) + \rho_T^\top \cdot \log(\rho_T + \text{tol}) - \rho^\top \cdot \log(\rho + \text{tol}).$$

Hierbei ist die diskrete Verbunddichteschätzfunktion ρ in der Rolle des Residuums r und $\text{tol} \in \mathbb{R}^+$ ein Toleranzwert.

Es ist $\rho \in \mathbb{R}^{n_R \times n_T}$, wobei n_R, n_T die Anzahl der verwendeten Bins (= die Anzahl der Unterteilungen der Grauwertbereiche) für die jeweiligen Bilder angeben. Wie die diskrete Dichteschätzung realisiert wird, diskutieren wir unten. Mit den Hilfsgrößen

$$S_R = e_{n_2}^\top \otimes I_{n_1} \quad \text{und} \quad S_T = I_{n_2} \otimes e_{n_1}^\top$$

lassen sich die Dichten der einzelnen Bilder schreiben als

$$\rho_R = S_R \rho = \sum_{i=1}^{n_T} \bar{\rho}_{i,:} \quad \text{und} \quad \rho_T = S_T \rho = \sum_{j=1}^{n_R} \bar{\rho}_{:,j},$$

womit sich für die Ableitungen ergibt:

$$\begin{aligned} d\varphi(\mathbf{y}) &= \left((\log(\rho_R + \text{tol}) + \rho_R \oslash (\rho_R + \text{tol}))^\top \cdot S_R \right. \\ &\quad \left. + (\log(\rho_T + \text{tol}) + \rho_T \oslash (\rho_T + \text{tol}))^\top \cdot S_T \right. \\ &\quad \left. - (\log(\rho + \text{tol}) + \rho \oslash (\rho + \text{tol}))^\top \right) \cdot d\rho \\ d^2\varphi(\mathbf{y}) &= S_R^\top \cdot \text{diag}((\rho_R + 2 \cdot \text{tol}) \oslash ((\rho_R + \text{tol}) \odot (\rho_R + \text{tol}))) \cdot S_R \\ &\quad + S_T^\top \cdot \text{diag}((\rho_T + 2 \cdot \text{tol}) \oslash ((\rho_T + \text{tol}) \odot (\rho_T + \text{tol}))) \cdot S_T \\ &\quad - \text{diag}((\rho + 2 \cdot \text{tol}) \oslash ((\rho + \text{tol}) \odot (\rho + \text{tol}))). \end{aligned}$$

Mutual Information besitzt mehrere Eigenschaften, die ihre Anwendung im Rahmen der Registrierung erschweren.

- Das kontinuierliche Maß ist nicht nach unten beschränkt. Erst durch die Verwendung von Dichteschätzern entsteht eine untere Schranke, vergleiche [33].
- Zur Schätzung der Verbundsdichte ρ existiert eine Vielzahl von Möglichkeiten (Histogrammschätzer, Kerndichteschätzer), so dass von *einem* Maß zu sprechen nicht möglich ist.
- Die oben vorgestellte Approximation der zweiten Ableitung des Maßes ist dicht besetzt, so dass der Speicheraufwand bei der Nutzung eines Gauß-Newton-Verfahrens beliebig groß ist, und effiziente Verfahren schwer erreicht werden können.

Wir stellen mit dem folgenden Maß eine Alternative zur Mutual Information vor, die sowohl für multimodale Bilddaten geeignet ist als auch die numerischen Probleme umgeht und nicht auf einer stochastischen Interpretation der Daten basiert.

6.3 Normalisiertes Gradienten-Feld

6.3.1 Definition

Wir beschreiben nun das Maß, das von E. Haber und J. Modersitzki [30, 31] vorgestellt worden ist, das *normalized gradient field* (NGF). Dieses Maß zielt nicht auf die Grauwerte der Bilder, sondern auf deren Ortsableitungen.

Dazu ähnliche Maße werden auch in der Bewegungserkennung genutzt. In den Arbeiten [11, 59] stellen die Autoren ein auf ersten und zweiten Bildableitungen basierendes Maß zur Erkennung von 2D-Bewegung in Bildsequenzen vor.

Idee des Maßes ist es, Punkte als korrespondierend anzusehen, wenn an diesen Stellen in Referenz- und Templatebild Bildkanten in gleicher Richtung verlaufen. Als Indikator für Kanten wird der Bildgradient genutzt. Ziel ist, ein Maß zu konstruieren, das misst, ob die Bildgradienten in einem Punktepaar linear abhängig sind. Da die Höhe der Kante nicht einfließen soll, werden die Bildgradienten normiert.

Zur Bestimmung dieser Abhängigkeit können das Kreuz- und das Skalarprodukt herangezogen werden. Die Norm des Kreuzproduktes misst (in zwei oder drei Dimensionen) die Fläche des von den Gradienten aufgespannten Parallelogramms. Es ist maximal ($= 1$), wenn die Gradienten senkrecht zueinander stehen, und minimal ($= 0$), wenn sie linear abhängig sind. Eine erste Formulierung des Maßes lautet

$$\mathcal{D}^{ngf}[\mathcal{T}, \mathcal{R}, y] = \frac{1}{2} \int_{\Omega} \left\| \frac{\nabla \mathcal{T}(y(x))}{\|\nabla \mathcal{T}(y(x))\|} \times \frac{\nabla \mathcal{R}(x)}{\|\nabla \mathcal{R}(x)\|} \right\|^2 dx.$$

Diese Formulierung hat jedoch zwei Nachteile. Der erste ist numerischer Art. Bei der Bestimmung des Kreuzproduktes werden Differenzen gebildet. Hier kann es zu Auslöschungsphänomenen kommen. Der zweite Nachteil ist, dass durch die Formulierung des Maßes lokale Minima der Zielfunktion erzeugt werden. Denn gilt $T_{\mathbf{y}} = \text{const}$ und somit $\nabla T_{\mathbf{y}} = 0$, so folgt $\mathcal{D}^{ngf} = 0$. Während der Optimierung kann also das Verfahren versuchen, das Templatebild so zu deformieren, dass es konstant wird. Durch die Annahme bei der Interpolation, die das Bild außerhalb von Ω mit Null fortsetzt, ist ein Minimum des Maßes gegeben, wenn das Gitter \mathbf{y} aus dem Gebiet Ω herausgeschoben wird.

Daher nutzen wir das Skalarprodukt. Sind die normierten Bildgradienten linear abhängig, so gilt, ihr Skalarprodukt ist Eins. Bilden sie einen von Null verschiedenen Winkel, so liegt ihr Skalarprodukt zwischen Null und Eins, mit dem Minimum Null, falls sie senkrecht zueinander stehen. Zur Einbettung in den globalen Minimierungsprozess der Registrierung dient dann folgende Formulierung:

$$\mathcal{D}^{ngf}[\mathcal{T}, \mathcal{R}, y] = \int_{\Omega} 1 - \left(\frac{\langle \nabla \mathcal{T}(y(x)), \nabla \mathcal{R}(x) \rangle}{\|\nabla \mathcal{T}(y(x))\| \|\nabla \mathcal{R}(x)\|} \right)^2 dx.$$

Und mit $\bar{\omega} = \omega_1 \cdot \omega_2 \cdot \omega_3$ ergibt sich

$$\mathcal{D}^{ngf}[\mathcal{T}, \mathcal{R}, y] = \bar{\omega} - \int_{\Omega} \left(\frac{\langle \nabla \mathcal{T}(y(x)), \nabla \mathcal{R}(x) \rangle}{\|\nabla \mathcal{T}(y(x))\| \|\nabla \mathcal{R}(x)\|} \right)^2 dx.$$

Auslöschungsprobleme treten hierbei nicht auf, da keine Subtraktion genutzt wird. Lokale Minima der oben beschriebenen Form können nicht auftreten. Mit $T_{\mathbf{y}} = \text{const}$ und $\nabla T_{\mathbf{y}} = 0$ folgt, dass das Maß maximal wird.

6.3.2 Diskretisierung

Wir leiten nun die für dieses Maß notwendigen Funktionen φ, r samt ihren Ableitungen her. Es ist wieder

$$\begin{aligned} \varphi(r(\mathbf{y})) &= \bar{\omega} - \frac{\bar{h}}{2} \|r(\mathbf{y})\|^2, \\ d\varphi(\mathbf{y}) &= -\bar{h} \cdot r(\mathbf{y})^\top \cdot dr(\mathbf{y}), \\ d^2\varphi(\mathbf{y}) &= -dr(\mathbf{y})^\top \cdot \bar{h} \cdot dr(\mathbf{y}). \end{aligned}$$

Von Interesse ist jetzt nur noch die Funktion r und ihre Ableitung. Zur diskreten Bestimmung der Gradienten $\nabla \mathcal{T}$ und $\nabla \mathcal{R}$ nutzen wir einen Ansatz mittels

finiter Differenzen, dazu sei

$$D_{m_j} = \frac{1}{2h_j} \begin{pmatrix} -1 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{m_j \times m_j}$$

die die 1D-Ableitung approximierende Matrix und

$$\begin{aligned} D_1 &= I_{m_3} \otimes I_{m_2} \otimes D_{m_1} \\ D_2 &= I_{m_3} \otimes D_{m_2} \otimes I_{m_1} \\ D_3 &= D_{m_3} \otimes I_{m_2} \otimes I_{m_1} \end{aligned}$$

die Ableitungsmatrizen für die Bilddaten. Es gilt $D_j R \in \mathbb{R}^n$, $n = m_1 m_2 m_3$, approximiert die Ableitung in die j -te Raumrichtung vom Bild \mathcal{R} für jeden Punkt $x \in \mathbf{x}$. Hiermit können wir approximieren:

1. Das Skalarprodukt zwischen den Gradienten. Mit

$$C_i = D_i \mathcal{T}_y \odot D_i R$$

erhalten wir als Zähler der Residuumsfunktion

$$r_1(\mathbf{y}) = C_1 + C_2 + C_3.$$

2. Die Länge der Gradientenvektoren $\|\nabla \mathcal{T}(y)\|$ und $\|\mathcal{R}\|$ durch die Größen

$$\begin{aligned} l\mathcal{T}_y &= \sqrt{D_1 \mathcal{T}_y \odot D_1 \mathcal{T}_y + D_2 \mathcal{T}_y \odot D_2 \mathcal{T}_y + D_3 \mathcal{T}_y \odot D_3 \mathcal{T}_y + \epsilon^2} \in \mathbb{R}^n, \\ lR &= \sqrt{D_1 R \odot D_1 R + D_2 R \odot D_2 R + D_3 R \odot D_3 R + \epsilon^2} \in \mathbb{R}^n. \end{aligned}$$

Die Vektoren enthalten für jeden Diskretisierungspunkt eine Approximation an die Länge der jeweiligen Gradienten. Die Größe ϵ sorgt für Stabilität in konstanten Bildbereichen, sie ist kein variabler Parameter. Wir setzen $\epsilon = 0.001$. Für den Nenner erhalten wir somit die Funktion

$$r_2(\mathbf{y}) = (l\mathcal{T}_y \odot lR).$$

Damit ergibt sich für die Residuumsfunktion

$$r(\mathbf{y}) = r_1 \oslash r_2.$$

Zur Bestimmung der Ableitung dr betrachten wir Zähler und Nenner nacheinander. Es gilt

$$dr_1 = \text{diag}(D_1 R) D_1 + \text{diag}(D_2 R) D_2 + \text{diag}(D_3 R) D_3 \quad (6.4)$$

und

$$dr_2 = - \text{diag}(1 \otimes (lR \odot lT_y^3)) \cdot (\text{diag}(D_1 T_y) \cdot D_1 + \text{diag}(D_2 T_y) \cdot D_2 + \text{diag}(D_3 T_y) \cdot D_3)$$

und mit Hilfe der Kettenregel erhalten wir

$$dr(y) = (\text{diag}(r_2) \cdot dr_1 + \text{diag}(r_1) \cdot dr_2) \cdot dT_y.$$

Dadurch, dass wir nicht auf die analytischen Ableitungen von \mathcal{T} und \mathcal{R} zurückgegriffen haben (die wir durch die Interpolation erhalten), sondern durch die Matrizen D_1, D_2 und D_3 explizit den numerischen Ableitungsoperator vorgeben, machen wir uns unabhängig von der Wahl der Interpolationsmethode. Dies wirkt sich auf die Darstellung der Ableitung dr aus. Hier spielen die Ableitungsoperatoren D_1, D_2, D_3 explizit herein (vergl. (6.4)). Unter Verwendung der analytischen Ableitung gelingt es nicht, einen Ausdruck für die Ableitung anzugeben, der unabhängig vom gewählten Interpolationsverfahren ist.

Fazit

Wir haben drei verschiedene, in der Literatur bekannte Distanzmaße so vorgestellt, dass sie in den Discretize-then-Optimize-Ansatz passen und innerhalb des Ansatzes austauschbar sind. Wir haben sie so formuliert, dass ihre für die Minimierung bezüglich der Transformation, die auf das Templatebild wirkt, relevanten Größen für das Gauß-Newton-Verfahren gegeben sind.

Für das Standardmaß der multimodalen Bildregistrierung, Mutual Information, haben wir mit dem NGF-Maß eine leicht zu implementierende Alternative vorgestellt; deren Einbettung in den Discretize-then-Optimize-Ansatz ist erstmalig ausführlich beschrieben.

Damit haben wir den Baustein der Ähnlichkeit zwischen zwei Bildern erläutert und können nun im nächsten Kapitel den ersten Registrierungsalgorithmus herleiten.

Kapitel 7

Parametrische Bildregistrierung

Die diesem Kapitel zu Grunde liegende Idee besteht darin, die gesuchte, ein gegebenes Distanzmaß minimierende Deformation \mathbf{y} als Linearkombination vorgegebener Basisfunktionen zu beschreiben. Dies vereinfacht die Komplexität der Registrierung durch die Verkleinerung der Dimension des Suchraumes drastisch.

Als motivierendes Beispiel betrachten wir ein gegebenes Gitter mit $n = 512^3$ Diskretisierungspunkten, suchen also ein $\mathbf{y} \in \mathbb{R}^{3n}$ mit $3n = 402653184$. Suchen wir diese Deformation aber innerhalb der affin-linearen Funktionen verkleinert sich der Suchraum auf den \mathbb{R}^{12} . Typische Wahl solcher Suchräume ist (wie oben erwähnt) die affin-linearen Funktionen, rigide Abbildungen, aber auch über B-Spline parametrisierte Abbildungen [72]. Der durch die Reduktion des Suchraumes entstehende Nachteil liegt darin, dass die so bestimmten Deformationen häufig die realen Deformationen nur unzureichend modellieren und wiedergeben.

Wir verwenden daher parametrische Bildregistrierungsalgorithmen nur zur Vorregistrierung der Daten, deren genaue Einbettung in einen Registrierungsrahmen in Kapitel 9.5 beschrieben wird. Inhalt dieses Kapitels ist die Beschreibung eines generellen Rahmens für die parametrische Registrierung und dessen Spezialisierung auf die Fälle der rigiden und affin-linearen Abbildungen. Im zweiten Abschnitt werden wir dann aus den bisher beschriebenen Bausteinen einen ersten Registrierungsalgorithmus herleiten.

7.1 Affin-lineare Deformation

Beginnen wir mit der Beschreibung von affin-linearen Deformationen. Die Transformation ist Linearkombination von Rotation (mit dem Ursprung als Drehzentrum), Translation, Scherung und Skalierung. Wir nutzen zugleich die diskrete Beschreibungsmöglichkeit. Sei dazu $\Omega = (0, \omega_1) \times \dots \times (0, \omega_3)$ das Bildgebiet,

$m = (m_1, m_2, m_3)$, die Anzahl der Diskretisierungspunkte, $n = m_1 m_2 m_3$, und $\mathbf{x} = \mathbf{x}^{cc}(\Omega, m)$ ein Cellcentered-Gitter. Wir stellen nun die diskrete Transformation dar als

$$\mathbf{y} = Q\boldsymbol{\gamma} \quad \text{und} \quad Q \in \mathbb{R}^{3n \times p}, \boldsymbol{\gamma} \in \mathbb{R}^p. \quad (7.1)$$

Der Vektor $\boldsymbol{\gamma}$ enthält die zu bestimmenden Linearfaktoren, die Spalten der Matrix Q die Basisfunktionen ausgewertet an den Positionen $\bar{\mathbf{x}}$ des gegebenen Gitters. Es gilt für jeden Punkt des Gitters

$$\bar{\mathbf{y}}_{j,:} = \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 \\ \gamma_5 & \gamma_6 & \gamma_7 \\ \gamma_9 & \gamma_{10} & \gamma_{11} \end{pmatrix} \bar{\mathbf{x}}_{j,:} + \begin{pmatrix} \gamma_4 \\ \gamma_8 \\ \gamma_{12} \end{pmatrix}, \quad j = 1, \dots, n \quad (7.2)$$

und somit

$$\begin{aligned} \hat{\mathbf{y}}_{j,1} &= \gamma_1 \bar{\mathbf{x}}_{j,1} + \gamma_2 \bar{\mathbf{x}}_{j,2} + \gamma_3 \bar{\mathbf{x}}_{j,3} + \gamma_4 \\ \hat{\mathbf{y}}_{j,2} &= \gamma_5 \bar{\mathbf{x}}_{j,1} + \gamma_6 \bar{\mathbf{x}}_{j,2} + \gamma_7 \bar{\mathbf{x}}_{j,3} + \gamma_8 \quad j = 1, \dots, n \\ \hat{\mathbf{y}}_{j,3} &= \gamma_9 \bar{\mathbf{x}}_{j,1} + \gamma_{10} \bar{\mathbf{x}}_{j,2} + \gamma_{11} \bar{\mathbf{x}}_{j,3} + \gamma_{12}. \end{aligned}$$

Die zwölf Parameter korrespondieren zu den Freiheitsgraden der Basis: drei Rotationswinkel, drei Translationen, drei Scherungen und drei Skalierungen (je in eine Koordinatenachsenrichtung). Sortieren wir die Gleichung nach der Gitterdurchlaufreihenfolge aus Kapitel 4 um, so erhalten wir mit

$$Q = I_d \otimes (\bar{\mathbf{x}}, e_n) = \begin{pmatrix} \bar{\mathbf{x}} & e_n & & \\ & \bar{\mathbf{x}} & e_n & \\ & & \bar{\mathbf{x}} & e_n \end{pmatrix} \quad (7.3)$$

die in Gleichung (7.1) dargestellte Form für den affin-linearen Fall. Die Matrix Q beinhaltet also die linearen und konstanten Polynome ausgewertet an den Positionen aus $\bar{\mathbf{x}}$. Durch Kenntnis des Gebietes Ω und der Diskretisierungspunkte m sowie der zwölf beschriebenen Parameter $\boldsymbol{\gamma}$ ist \mathbf{y} eindeutig bestimmt.

Zur Nutzung ableitungsbasierter Minimierungsverfahren benötigen wir im nächsten Schritt die Ableitung der Funktion $\mathbf{y} : \mathbb{R}^{12} \rightarrow \mathbb{R}^n$ nach den Parametern $\boldsymbol{\gamma}$. Dies ist für den affin-linearen Fall äußerst einfach :

$$\begin{aligned} d_{\boldsymbol{\gamma}} \mathbf{y} : \mathbb{R}^{12} &\rightarrow \mathbb{R}^{n \times 12} \\ \boldsymbol{\gamma} &\mapsto d_{\boldsymbol{\gamma}} \mathbf{y} = Q. \end{aligned}$$

7.2 Rigide Deformation

Wir vereinfachen das Modell auf rigide Abbildungen. Die Transformation setzt sich aus Rotationen und Translationen zusammen. Diese passen nicht in den von den affin-linearen Abbildungen vorgegebenen Rahmen, da für Rotationen

die Parameter nicht-linear in die Transformation \mathbf{y} eingehen. Zum besseren Verständnis präsentieren wir hier zunächst den 2D Fall.

Analog zu Gleichung (7.2) lässt sich für jeden Punkt des Gitters \mathbf{x} die rigide Transformation \mathbf{y} schreiben als

$$\bar{\mathbf{y}}_j := \begin{pmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{pmatrix} \bar{\mathbf{x}}_j + \begin{pmatrix} \gamma_2 \\ \gamma_3 \end{pmatrix}, \quad j = 1, \dots, n,$$

wobei $c_1 = \cos \gamma_1, s_1 = \sin \gamma_1$. Auffällig ist, dass der Parameter γ_1 nicht-linear in $\hat{\mathbf{y}}_j$ eingeht. Sortieren wir diese Gleichung um, erhalten wir

$$\mathbf{y} = \underbrace{\begin{pmatrix} \bar{\mathbf{x}} & e_n \\ & \bar{\mathbf{x}} & e_n \end{pmatrix}}_{=:Q} \begin{pmatrix} c_1 \\ -s_1 \\ \gamma_2 \\ s_1 \\ c_1 \\ \gamma_3 \end{pmatrix}.$$

Eine Matrix der gleichen Gestalt wie in (7.3) kommt hier vor, die Gleichung weicht nur in der Darstellung des Parametervektors ab, in die der Rotationsparameter γ_1 viermal eingeht. Betrachten wir \mathbf{y} als Funktion in $\gamma: \mathbf{y}: \mathbb{R}^3 \rightarrow \mathbb{R}^n$ ergibt sich für die Ableitung

$$d_\gamma \mathbf{y}: \mathbb{R}^3 \rightarrow \mathbb{R}^{n \times 3} \quad (7.4)$$

$$\gamma \mapsto d_\gamma \mathbf{y} = Q \begin{pmatrix} -s_1 & & \\ -c_1 & & \\ 0 & 1 & \\ c_1 & & \\ -s_1 & & \\ 0 & & 1 \end{pmatrix}. \quad (7.5)$$

Eine gleiche Struktur erwarten wir nun auch für den Fall rigider Deformationen in drei Raumdimensionen. Wir erhalten in diesem Fall drei mögliche Drehachsen mit den dazugehörigen Rotationsmatrizen

$$R_1 = \begin{pmatrix} 1 & & \\ & c_1 & -s_1 \\ & s_1 & c_1 \end{pmatrix}, R_2 = \begin{pmatrix} c_2 & & -s_2 \\ & 1 & \\ s_2 & & c_2 \end{pmatrix}, R_3 = \begin{pmatrix} c_3 & -s_3 & \\ s_3 & c_3 & \\ & & 1 \end{pmatrix},$$

und den abkürzenden Schreibweisen $c_j = \cos \gamma_j, s_j = \sin \gamma_j, j = 1, 2, 3$. Das Produkt dieser Matrizen ist durch

$$R_1 R_2 R_3 = \begin{pmatrix} c_2 c_3 & -c_2 s_3 & -s_2 \\ -s_1 s_2 c_3 + c_1 s_3 & s_1 s_2 s_3 + c_1 c_3 & -s_1 c_2 \\ s_1 s_3 + c_1 s_2 c_3 & s_1 c_3 - c_1 s_2 s_3 & c_1 c_2 \end{pmatrix}$$

3. Wir kennen für die vorgeschlagenen parametrischen Deformationen ihre Zuordnungsvorschrift und Ableitung bezüglich des Parametervektors γ . Dieses Modell ist auf andere parametrische Deformationen übertragbar. Ein Beispiel hierfür ist die Parametrisierung der Deformation durch Splineskoeffizienten [72].

7.3 Verfahren

Durch die Vorarbeiten der vorausgegangenen Kapitel haben wir uns nun sämtliche Bausteine erschlossen, die wir zur Formulierung eines ersten Registrierungsverfahrens benötigen. Gesucht ist eine diskrete Transformation \mathbf{y} , die über eine Klasse von Deformationen und einen Parametervektor γ bestimmt ist und die für gegebene Bilder \mathcal{R}, \mathcal{T} ein gegebenes Distanzmaß $\mathcal{D} = \mathcal{D}(\mathcal{R}, \mathcal{T}, \mathbf{y})$ minimiert. Wir wissen nun:

1. Der Bildbereich Ω_R des Referenzbildes ist durch Vorgabe der Anzahl von Diskretisierungspunkten m zu diskretisieren. Als Resultat erhalten wir ein Gitter $\mathbf{x} = \mathbf{x}^{cc}(\Omega_R, m)$. Gesucht ist eine darauf aufbauende Transformation $y = \mathbf{y}(\mathbf{x})$.
2. Die Plausibilität der Transformation gewährleisten wir durch Wahl einer Parametrisierung der Transformation.
3. Die so erzielte Transformation ging in Kapitel 6 in die Diskretisierung der dort vorgestellten Distanzmaße ein. Für die jeweiligen Maße haben wir die für die Optimierung relevanten Größen hergeleitet. Im Fall einer Parametrisierung von \mathbf{y} ist nun zu beachten, dass nicht die gesuchte Transformation selbst, sondern der sie parametrisierende Vektor γ Zielvariable der Optimierung ist, $\mathbf{y} = y(\gamma)$. Die Kettenregel beantwortet die Frage, wie wir mit diesem Faktum umzugehen haben.
4. In Kapitel 3 haben wir mit der Gauß-Newton-Methode ein Verfahren vorgestellt, das nun zu formulierende Problem lösen kann. Die Zielfunktion ist so beschrieben worden, dass sie in den Rahmen der Optimierungsmethode passt. Die Iterationsvorschrift des Gauß-Newton-Verfahrens liest sich nun

$$(d_\gamma \mathbf{y}^\top d_y r^\top \bar{h} d_y r d_\gamma \mathbf{y}) \delta \gamma = -\bar{h} d_y r d_\gamma \mathbf{y} \quad (7.6)$$

Das Verfahren ist knapp in Algorithmus 4 zusammengefasst.

In Abbildung 7.1 haben wir ein Ergebnis visualisiert, das wir mit der parametrischen Registrierung für die Testdaten bestimmt haben. Wir haben als Distanz das NGF-Maß genutzt und ein affin-lineares Modell verwendet. Spline-Interpolation mit $\theta = 15$ ist verwendet worden. Startwert für γ ist die Identität.

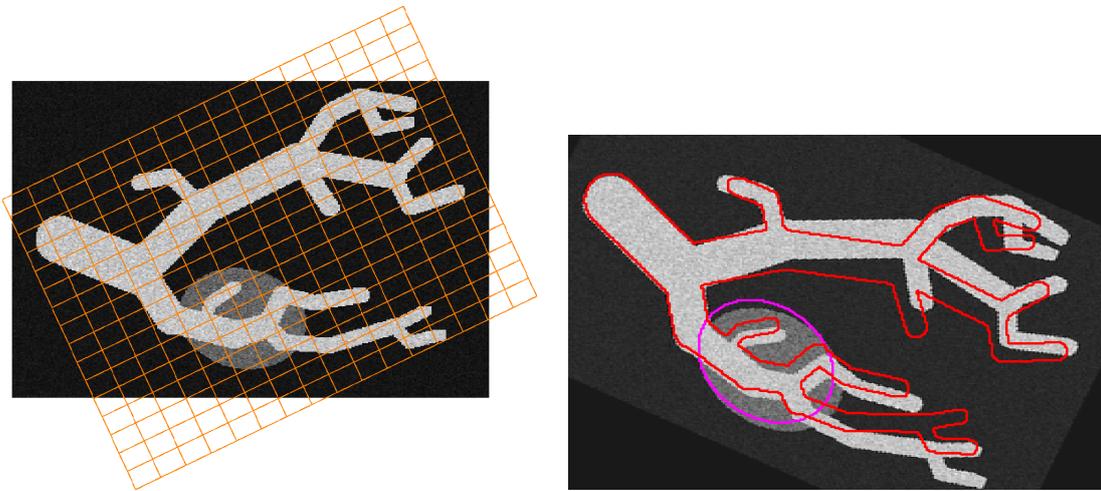


Abbildung 7.1: Ergebnisse der parametrischen Registrierung; *links*: Template-Daten mit berechnetem Registrierungsergebnis; *rechts*: deformiertes Template mit Rändern des Referenzbildes

Als Ergebnis haben wir $\gamma^{opt} = (0.8436, -0.4215, 3.8827, 0.4086, 0.9192, -2.9317)$ erhalten. Zu erkennen ist, dass das Verfahren, obwohl nur ein eingeschränktes Deformationsmodell genutzt wurde, für die Hauptäste subjektiv passable Ergebnisse liefert.

Fazit

Durch die Vorgabe einer Klasse von Deformationen haben wir den Suchraum der Optimierung beschränkt und damit ein schnelles Registrierungsverfahren beschrieben. Die vorgestellten Klassen von Deformationen sind klein (rigide, affin-linear) aber auflösungsunabhängig. Wir werden diese Verfahren zur Vorregistrierung nutzen, um globale Deformationsanteile vor der nicht-linearen Registrierung bestimmen zu können, die wir im nächsten Kapitel beschreiben.

Algorithmus 4 Parametrische Registrierung; PIR

Eingabe: $\mathcal{T}, \mathcal{R}, \gamma^0$

Ausgabe: γ^{opt}

1: $\gamma^{opt} \leftarrow \text{GN}(\mathcal{D}(\mathcal{T}, \mathcal{R}, \gamma), \gamma^0)$

Kapitel 8

Nicht parametrische Registrierung

In diesem Abschnitt befassen wir uns mit freien, nicht parametrischen Transformationen in der Registrierung. Wir geben nicht mehr eine Klasse von Deformationen vor, sondern wählen als Transformation eine beliebige, (zweimal) stetig differenzierbare Funktion $y : \Omega \rightarrow \mathbb{R}^3$. Wir führen sogenannte Regularisierer ein, Maße, die Glattheitseigenschaften der Transformation messen. Dies wird in der Literatur durch verschiedene Argumentationen als notwendig beschrieben, vergleiche [54]. Wir beschränken uns hier darauf, dass durch Hinzunahme des Regularisierers der Suchraum für die Transformation eingeschränkt wird und es möglich ist, Glattheitseigenschaften an die gesuchte Transformation zu stellen.

Wir präsentieren in diesem Kapitel drei verschiedene Regularisierer, die in der Literatur bekannt sind und zeigen ihre Einbettung in den Discretize-then-Optimize-Ansatz.

Wir formulieren die Regularisierer zunächst als Funktional \mathcal{S} , das einer gegebenen Transformation y eine Größe zuordnet, die deren Glattheit bezüglich des Maßes entspricht. Als Zielfunktion der Registrierung nutzen wir

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

wobei \mathcal{D} ein beliebiges Distanzmaß ist, vergleiche Kapitel 6. Der Faktor $\alpha \in \mathbb{R}^+$ gewichtet den Einfluss des Regularisierers auf die Zielfunktion.

Da die Regularisierer die Glattheit der Zielfunktion modellieren, gibt es verschiedene weiterführende Ansätze, sie zu wählen. Beispielsweise ist es durch geeignete Wahl möglich, nur stückweise glatte Deformationen zu bestimmen, vergleiche [40, 71].

Wir beginnen mit der kontinuierlichen Formulierung der Regularisierer und zeigen dann detailliert, wie eine Diskretisierung dieser Maße erfolgen kann.

8.1 Kontinuierliche Formulierung von Regularisierern

Elastische Registrierung

In der Literatur existieren sich widersprechende Verwendungen des Begriffes „elastische Registrierung“. Wir bezeichnen damit nicht-parametrische Registrierung, die den folgenden Regularisierer nutzt. Dieser wurde 1981 von Broit in [9] vorgestellt. Motivation für diesen Ansatz war, die Regularität der gesuchten Transformation physikalisch zu messen. Grundlegendes Modell ist, dass die gegebenen Bilddaten auf einem elastischen Material aufgetragen sind. Dies wird deformiert und die Bilddaten geben das Bild vor und nach der Deformation wieder. Modelliert wird dies durch das Messen des linearen elastischen Potentials der zugrunde liegenden Funktion

$$\mathcal{S}^{\text{elas}}[y] = \frac{1}{2} \int_{\Omega} \sum_{j=1}^3 \mu \|\nabla(y_j - y_j^{\text{Kern}})\|^2 + (\mu + \lambda) \text{div}^2(y - y^{\text{Kern}}) dx, \quad (8.1)$$

wobei für $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ die Divergenz durch $\text{div} = \partial_1 f_1 + \partial_2 f_2 + \partial_3 f_3$ gegeben ist. Weiterhin sind $\lambda, \mu \geq 0$ Materialkonstanten, durch die unterschiedliche physikalische Materialien modelliert werden können.

Die Funktion $y^{\text{Kern}} : \Omega \rightarrow \mathbb{R}^3$ erlaubt die explizite Vorgabe von Anteilen der Deformation, die im Kern des Regularisierers liegen sollen, d. h. $\mathcal{S}(y^{\text{Kern}}) = 0$. Hiermit ist es möglich, Anteile der Deformation nicht zu regularisieren. Eine geeignete Wahl ist, $y^{\text{Kern}} = x$ zu wählen, so dass nur lokale Änderungen der Transformation regularisiert werden.

Diffusive Registrierung

Der diffusive Regularisierer wurde 1981 von Horn und Schunk in einer Arbeit [36] über optischen Fluss zum ersten Mal im Bereich der Bewegungserkennung vorgestellt. Für die Registrierung wurde er in der Arbeit [19, 21] von Fischer und Modersitzki 2001 beschrieben. Idee des Regularisierers ist es, Oszillationen der Transformation y zu unterdrücken. Diese werden im Regularisierer

$$\mathcal{S}^{\text{diff}}[y] = \frac{1}{2} \int_{\Omega} \sum_{j=1}^d \|\nabla(y_j - y_j^{\text{kern}})\|^2 dx \quad (8.2)$$

durch erste Ableitungen der Transformation gemessen. Die Namensgebung „diffusiv“ ist aus der Euler-Lagrange-Gleichung abgeleitet, die bei der Verwendung dieses Regularisierers im Optimize-then-Discretize-Ansatz auftritt. Diese stellt eine generalisierte Form einer Diffusionsgleichung dar. Zu Interpretieren ist dies als eine komponentenweise Glättung des Deformationsfeldes nach Art der Wärmeleitung [85].

Curvature basierte Registrierung

Das durch den Regularisierer

$$\mathcal{S}^{\text{curv}}[y] = \frac{1}{2} \int_{\Omega} \sum_{j=1}^3 \|\Delta(y_j - y_j^{\text{kern}})\|^2 dx \quad (8.3)$$

beschriebene Verfahren heißt Curvature-basiertes Verfahren. Es wurde zuerst von Fischer und Modersitzki in [24, 20] vorgestellt. Für eine zweimal stetig differenzierbare Funktion $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ ist durch $\Delta f = \partial_{11}f + \partial_{22}f + \partial_{33}f$ der Laplace-Operator definiert. Jeder in der obigen Summe auftretende Ausdruck Δy_j kann als Approximation an die Krümmung der jeweiligen Komponentenfunktion interpretiert werden, woher sich der Name für das Verfahren ableitet.

Charakteristisch für diesen Ansatz ist, dass lineare Transformationen im Kern des Regularisierers liegen, d.h. $\mathcal{S}^{\text{curv}}[y] = 0$, für $y : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, mit $y(x) = Ax + b$, $A \in \mathbb{R}^{3 \times 3}$, $b \in \mathbb{R}^3$. Globale, lineare Anteile (Translation, Rotation, Skalierung und Scherung) von Deformationen werden damit nicht bestraft. Durch die Regularisierung der zweiten Ableitungen sind die Transformationen in diesem Ansatz glatter als in den anderen Ansätzen.

Allen Regularisierern ist gleich, dass sie aus gewichteten Ableitungstermen bestehen. Um dem Discretize-then-Optimize-Ansatz zu folgen, geben wir eine Diskretisierung dieser Ableitungsterme an.

8.2 Ableitungsoperatoren im Eindimensionalen

Wir motivieren die Wahl von diskreten Ableitungsoperationen zunächst im Eindimensionalen, da hier, genau wie bei der Beschreibung der Gitter und der Interpolation (Kapitel 4, 5), Wesentliches leicht zugänglich ist.

Sei $\Omega = (0, \omega_1) \in \mathbb{R}$ und f eine zweimal stetig differenzierbare Funktion, $f : \Omega \rightarrow \mathbb{R}$. Sei weiterhin $m \in \mathbb{N}$ die vorgegebene Anzahl von Diskretisierungspunkten und seien $\mathbf{x}^n = \mathbf{x}^n(\Omega, m)$ und $\mathbf{x}^c = \mathbf{x}^{cc}(\Omega, m)$ die zu Ω, m gehörigen Nodal- und Cellcentered-Gitter. Dann approximiert $D \cdot f(\mathbf{x}^n)$ die erste Ableitung von f an den Positionen \mathbf{x}^c mit der Ordnung h^2 , $h = \omega/m$, das heißt $Df(\mathbf{x}^n) = f'(\mathbf{x}^c) + \mathcal{O}(h^2)$, mit

$$D = \frac{1}{h} \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{m \times m+1}. \quad (8.4)$$

Die Matrix $D(h, m)$ stellt also eine Approximation der ersten Ableitung mittels kurzer Differenzen dar. Die Approximationsordnung ergibt sich aus dem Satz

Davon abgesehen, dass wir für die Randwerte \mathbf{x}_1^c , \mathbf{x}_8^c keinen Wert der Ableitung bestimmt haben, erkennt der so definierte Ableitungsoperator nicht, dass die zugrundeliegende Funktion oszilliert, da er konstante Ableitungswerte zurückliefert.

Halten wir fest: durch Wahl eines Nodal-Gitters zur Auswertung der gegebenen Funktion f können wir die Ableitung mittels kurzer, symmetrischer Differenzen, die Oszillation in den Werten $f(x^n)$ erkennen, und ohne exklusive Behandlung der Randpunkte, auf einem Cellcentered-Gitter bestimmen. Dies nutzen wir jetzt im dreidimensionalen Raum aus.

8.3 Diskrete Regularisierung

In diesem Abschnitt werden wir für die drei vorgestellten Regularisierer eine auf den in Kapitel 4 beschriebenen Gittern basierende Diskretisierung anbieten.

Elastisch

Dieser Regularisierer enthält nicht nur Gradiententerme sondern auch einen Divergenzausdruck. Bei der Diskretisierung ist zu beachten, diesem Term ein Gitter zuzuordnen, auf dem er bestimmt wird. Auf diesem Gitter müssen dann auch alle partiellen Ableitungen der Form $\partial_i f_i$, $i = 1, 2, 3$ leben. Dabei ist eine naive Portierung des für den eindimensionalen Fall Gelernten nicht möglich. Denn falls die Funktion $f : \Omega \rightarrow \mathbb{R}^3$, $\Omega = (0, \omega_1) \times (0, \omega_2) \times (0, \omega_3)$ - wie im vorherigen Abschnitt - auf einem Nodal-Gitter gegeben ist, leben die Ableitungen - so bestimmt wie im 1D Fall - auf Staggered-Gittern und zwar die j -te Ableitung von y_k , $k = 1, 2, 3$ auf dem j -ten Staggered-Gitter, vergleiche Kapitel 4.

Wir zeigen nun, dass, falls f auf einem Staggered-Gitter gegeben ist, dieses Problem gelöst wird. Sei dazu $\mathbf{x}^{stg} = \mathbf{x}^{stg}(\Omega, m)$. Als Abkürzungen nutzen wir

$$\begin{aligned} n &= m_1 m_2 m_3 & n_1 &= (m_1 + 1) m_2 m_3, \\ n_2 &= m_1 (m_2 + 1) m_3, & n_3 &= m_1 m_2 (m_3 + 1). \end{aligned}$$

Der Vektor $f(\mathbf{x}^{stg}) \in \mathbb{R}^{n_1+n_2+n_3}$ ist zu lesen als

$$f(\mathbf{x}^{stg}) = (f_1(\mathbf{x}^{stg_1}), f_2(\mathbf{x}^{stg_2}), f_3(\mathbf{x}^{stg_3}))^\top. \quad (8.5)$$

Er enthält als Einträge also nacheinander die jeweiligen Komponentenfunktionen von f ausgewertet an Staggered-Gittern für die jeweiligen Dimensionen.

Mit der Hilfsmatrix aus (8.4) definieren wir

$$\begin{aligned}
D_{11} &= I_{m_3} \otimes I_{m_2} \otimes D(h_1, m_1) \\
D_{12} &= I_{m_3} \otimes D(h_2, m_2 - 1) \otimes I_{m_1+1} \\
D_{13} &= D(h_3, m_3 - 1) \otimes I_{m_2} \otimes I_{m_1+1} \\
\\
D_{21} &= I_{m_3} \otimes I_{m_2+1} \otimes D(h_1, m_1 - 1) \\
D_{22} &= I_{m_3} \otimes D(h_2, m_2) \otimes I_{m_1} \\
D_{23} &= D(h_3, m_3 - 1) \otimes I_{m_2+1} \otimes I_{m_1} \\
\\
D_{31} &= I_{m_3+1} \otimes I_{m_2} \otimes D(h_1, m_1 - 1) \\
D_{32} &= I_{m_3+1} \otimes D(h_2, m_2 - 1) \otimes I_{m_1} \\
D_{33} &= D(h_3, m_3) \otimes I_{m_2} \otimes I_{m_1}.
\end{aligned}$$

Durch die Kronecker-Struktur erhalten wir Matrizen D_{ij} , die die erste Ableitung der i -ten Komponentenfunktion f_i in die Raumrichtung x_j approximieren. Die Kronecker-Produkt-Struktur bewirkt, dass die jeweilige 1D-Matrix auf die richtigen Einträge trifft. Durch das Hinzunehmen von Nullblöcken erhalten wir so die geforderten Ableitungsoperatoren, es gilt

$$\begin{aligned}
(D_{11}, \mathbf{0}_{n,n_2}, \mathbf{0}_{n,n_3}) f(\mathbf{x}^{stg}) &\approx \partial_1 f_1(\mathbf{x}^c), \\
(\mathbf{0}_{n,n_1}, D_{22}, \mathbf{0}_{n,n_3}) f(\mathbf{x}^{stg}) &\approx \partial_2 f_2(\mathbf{x}^c), \\
(\mathbf{0}_{n,n_1}, \mathbf{0}_{n,n_2}, D_{33}) f(\mathbf{x}^{stg}) &\approx \partial_3 f_3(\mathbf{x}^c),
\end{aligned}$$

wobei $\mathbf{0}_{m,n} \in \mathbb{R}^{m,n}$ eine Null-Matrix darstellt.. Da alle partiellen Ableitungen auf dem Cellcentered-Gitter leben, können wir den für die elastische Registrierung nötigen Divergenzausdruck durch

$$(D_{11}, D_{22}, D_{33}) f(\mathbf{x}^{stg}) \approx \partial_1 f_1(\mathbf{x}^c) + \partial_2 f_2(\mathbf{x}^c) + \partial_3 f_3(\mathbf{x}^c) = \text{div} f(\mathbf{x}^c)$$

approximieren.

Den elastischen Regularisierer approximieren wir in zwei Schritten. Mit

$$B^{\text{elas}} = \begin{pmatrix} D_{11} \\ D_{12} \\ D_{13} \\ \\ D_{21} \\ D_{22} \\ D_{23} \\ \\ \\ D_{31} \\ D_{32} \\ D_{33} \\ D_{11} & D_{22} & D_{33} \end{pmatrix} \quad (8.6)$$

gilt

$$B^{\text{elas}} f(\mathbf{x}^{stg}) \approx (\partial_1 f_1, \partial_2 f_1, \dots, \partial_3 f_3, \text{div} f)^\top \quad (8.7)$$

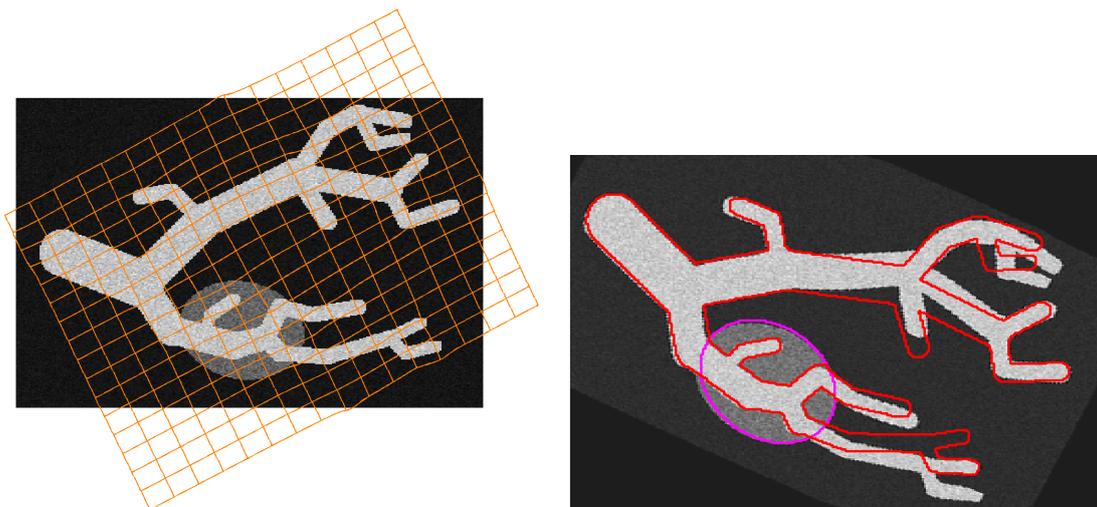
Algorithmus 5 Nicht-parametrische Registrierung; NPIR**Eingabe:** $\mathcal{R}, \mathcal{T}, \mathbf{y}^{\text{kern}}, \mathbf{y}^0, \alpha$ **Ausgabe:** \mathbf{y}_{opt} 1: $\mathbf{y}^{\text{opt}} \leftarrow \text{GN}(\mathcal{D}(\mathcal{R}, \mathcal{T}, \mathbf{y}) + \alpha \mathcal{S}(\mathbf{y} - \mathbf{y}^{\text{kern}}), \mathbf{y}^0)$ 

Abbildung 8.1: Ergebnisse der nicht-parametrischen Registrierung; *links*: Template-Daten mit berechnetem Registrierungsergebnis; *rechts*: deformiertes Template mit Rändern des Referenzbildes

Eingabe ist ein Startgitter \mathbf{y}^0 . Des Weiteren beschreibt \mathbf{y}^{kern} die Anteile der gesuchten Lösung, die nicht regularisiert werden sollen. Für beide gilt, dass eine naive Wahl die Identität ist, $\mathbf{y}^0 = \mathbf{y}^{\text{kern}} = \mathbf{x}^{\text{stg}}(\Omega_R, m)$. Das Verfahren minimiert dann die aus Distanzmaß und Regularisierer zusammengesetzte Funktion mittels Gauß-Newton-Verfahren (GN), wobei die zweite Ableitung des Regularisierers nicht approximiert wird.

Das Verfahren ist an den Beispieldaten illustriert. Als Zielfunktion verwenden wir das NGF-Maß mit dem elastischen Regularisierer. Als Interpolationsmethode nutzen wir Splines mit $\theta = 15$ und als Regularisierungsparameter wird $\alpha = 0.01$ verwendet. Startwert für die nicht-parametrische Registrierung ist das Ergebnis der affin-linearen Registrierung aus dem vorherigen Kapitel. Die Zielfunktion ist auf 87.4% minimiert worden. Die Ergebnisse sind in Abbildung 8.1 visualisiert. Zu erkennen ist, dass die berechnete Lösung in einem lokalen Minimum liegt, da einzelne Äste zwar auf andere Äste abgebildet werden, diese aber nicht korrespondieren.

8.5 Multigrid-Verfahren

Wir wenden uns abschließend der Frage zu, wie wir das im Gauß-Newton-Verfahren auftretende Gleichungssystem

$$(dr^\top d_2\varphi dr + \alpha B^\top B) \delta \mathbf{y} = - (d\mathcal{D}(\mathbf{y}) + \alpha B^\top B \mathbf{y}) \quad (8.14)$$

effizient lösen. Hierzu nutzen wir Multigrid-Techniken. Die detaillierte Beschreibung dieser Techniken ist nicht Gegenstand dieser Arbeit, für weiterführende Informationen vergleiche [81, 8].

Wir wiederholen die Idee des Verfahrens: Die Lösung des Gleichungssystems $Ax = b$ wird zunächst mittels weniger Schritte eines Relaxationsverfahrens approximiert. Das so bestimmte Residuum $r = Ax^k - b$ wird mittels Restriktion auf ein gröber diskretisiertes Gitter gebracht. Hier wird die Residuums-gleichung $Ae = r$ gelöst. Deren Lösung e wird auf das feinere Gitter zurück prolongiert und dient weiteren (wenigen) Schritten des Relaxationsverfahrens als Startwert. Die Lösung der Residuums-gleichung auf dem gröberen Gitter erfolgt durch rekursive Anwendung dieses Schemas, vergleiche [81, 8]. Im Falle der Konvergenz erhalten wir ein Verfahren, das die Lösung des Gleichungssystems in der Zeitkomplexität von $\mathcal{O}(n)$ bestimmt, wobei $x \in \mathbb{R}^n$.

Wir konkretisieren im Folgenden, mit welcher Wahl von Unterfunktionen und Parametern wir das Multigridverfahren nutzen:

- Als Relaxations-Verfahren verwenden wir ein gedämpftes Jacobi-Verfahren mit Dämpfungsfaktor $\nu = \frac{2}{3}$.
- Als Restriktions- und Prolongations-Operatoren für die Fehler und Residuen nutzen wir die in Abschnitt 9.4 beschriebenen Operatoren.
- Wir behandeln die Summanden der Matrix $(dr^\top d_2\varphi dr + \alpha B^\top B)$ unterschiedlich. Der erste Summand $dr^\top d_2\varphi dr$ wird nur durch eine Diagonalmatrix approximiert, die gleich seiner Hauptdiagonale ist. Dies erlaubt schnelle Gitterwechsel, für die wiederum die in Abschnitt 9.4 beschriebenen Operatoren genutzt werden. Des Weiteren ist sie auf dem groben Gitter leicht zu invertieren.
- Die Matrix $B^\top B$ wird gemäß Abschnitt 8.3 dieses Kapitels für jedes Gitter neu aufgebaut.
- Wir nutzen als Multigrid-Schema den V-Zyklus.
- Auf dem größten Gitter ($A \in \mathbb{R}^{n \times n}$) wird das System explizit gelöst.
- Durch die Verwendung von Staggerd-Gittern ist die h-Elliptizität des Systems gegeben, vergleiche [29]. Das Verfahren konvergiert.

Fazit

Ähnlich dem Vorgehen im Kapitel über Distanzmaße haben wir drei in der Literatur bekannte Regularisierer vorgestellt und in den Kontext des Discretize-then-Optimize-Ansatz eingebettet. Auch diese lassen sich modular austauschen, so dass wir mit dem bisherigen Ansatz und je drei Distanzmaßen und drei Regularisierern den Rahmen für neun verschiedene Verfahren beschrieben haben.

Durch die Verwendung eines Regularisierers wird die Plausibilität der zu bestimmenden Deformation gewährleistet.

Durch die Minimierung einer Zielfunktion aus Distanzmaß und Regularisierer mittels Gauß-Newton-Verfahren haben wir einen nicht-parametrischen Registrierungsalgorithmus hergeleitet; für die effiziente Bestimmung der Suchrichtung haben wir kurz ein Multigridverfahren skizziert.

Wir haben gesehen, dass das Verfahren abhängig vom Startwert ist, zu dessen Bestimmung wir bisher das Ergebnis einer parametrischen Registrierung nutzen. Im folgenden Kapitel stellen wir ein Multiresolutionsverfahren vor, das diese Aufgabe übernehmen wird.

Kapitel 9

Multiresolutionsverfahren

In diesem Kapitel werden wir die Frage nach der geeigneten Wahl eines Startwertes für die Registrierung und nach der Vermeidung möglicher lokaler Minima im Optimierungsprozess beantworten. Dieses ist ein Multiresolutionsverfahren.

Beim Multiresolutionsverfahren wird zunächst eine grob diskretisierte Deformation durch einen Registrierungsalgorithmus bestimmt. Grob diskretisiert meint, dass die Anzahl an Diskretisierungspunkten m gering ist und damit die Gitterweite h groß. Die so gewonnene Deformation wird mittels Gitterwechseloperatoren auf ein feineres Gitter (fein = größere Anzahl an Diskretisierungspunkten und damit kleinere Gitterweite) transformiert und dient nun als Startwert für den gleichen Registrierungsalgorithmus. Dieses wiederholt sich, bis die gewünschte Auflösung erreicht ist. Herauszustellendes Merkmal ist, dass die Auflösung der Transformation über die verschiedenen Iterationen zunimmt.

9.1 Motivation

Anderes als in der bisherigen Notation bezeichnen wir hier mit Gitterwechsel nicht den Wechsel von Staggered-Gittern zu Cellcentered-Gittern o. ä., sondern die Änderung der Gittergröße h .

Durch die Verwendung eines solchen Ansatzes wollen wir verschiedene Ziele erreichen, die miteinander verwoben sind.

- Erstes Ziel ist die Bestimmung eines guten Startwertes für den Registrierungsalgorithmus. Durch die Nutzung der optimalen Lösung einer groben Auflösung als Startwert für die nächst feinere hoffen wir einen guten Wert zu erreichen. Gut heißt, dass dieser nahe der gesuchten Lösung liegt. Das der Registrierung zugrunde liegende Optimierungsverfahren konvergiert dann schnell.

- Zweites Ziel ist die Vermeidung lokaler Minima der Zielfunktion als Lösung. Durch das Anpassen der Daten an die jeweilige Auflösungsstufe (vergleiche Abschnitt 9.3) erhalten wir bei groben Auflösungen eine glatte Zielfunktion, die einfach zu optimieren ist und im Idealfall keine lokalen Minima hat.
- Drittes Ziel ist die Formulierung eines effizienten und schnellen Verfahrens. Glatte Zielfunktionen sind schnell zu optimieren. Durch Auflösungswechsel werden zusätzliche Bilddetails in die Registrierung einbezogen, die jedoch nur zu lokalen Anpassungen des Deformationsfeldes führen; somit liegt die Lösung auf der feinen Auflösung nahe der Lösung auf der groben Auflösung. Idealerweise wird auf der feinsten Auflösung nur ein Schritt im Registrierungsverfahren gerechnet.

Wir gehen in zwei Schritten vor. Zunächst erklären wir, wie im Eindimensionalen ein Wechsel der Gittergröße für Cellcentered- und Nodal-Gitter gelöst werden kann und wie wir die dazugehörigen Operatoren definieren. Im darauffolgenden Schritt nutzen wir diese Operatoren, um zwei Verfahren abzuleiten: Erstens die Verminderung der Auflösung von Bilddaten und zweitens die Verfeinerung der Auflösung von Transformationen.

9.2 Gitterwechsel im Eindimensionalen

Wir betrachten zunächst Daten auf Cellcentered-Gittern. Gegeben ist $\Omega = (0, \omega_1)$ und ein Gitter mit $m = 2k$ Diskretisierungspunkten und Gitterweite $h = \omega_1/m$, $\mathbf{x} = \mathbf{x}^{cc}(\Omega, m)$. Des Weiteren haben wir an den Positionen \mathbf{x}_j die Werte einer Funktion $f : \Omega \rightarrow \mathbb{R}$ gegeben, $\mathbf{f} = f(\mathbf{x}) \in \mathbb{R}^m$. Wir definieren zwei Operatoren, einen für den Wechsel auf ein Gitter mit doppelt so feiner Auflösung P_h^H und einen für den Wechsel auf ein Gitter mit halb so feiner Auflösung P_H^h . Die Bezeichnung der Operatoren ist aus Gründen der Lesbarkeit einfach gehalten. Ihre Dimension geht aus dem Kontext hervor.

Für die Wahl eines Multiresolutionsverfahrens gibt es eine Vielzahl von Möglichkeiten (vergleiche unter anderem [32, 38]). Wir nutzen hier einfache Operatoren, vergleiche [81]. Um noch weiter zu vereinfachen, nutzen wir Zweierpotenzen für die Anzahl m der Gitterpunkte, das heißt $m = 2^k$ (und im späteren Dreidimensionalen $m = (2^{k_1}, 2^{k_2}, 2^{k_3})$).

Wir stellen zwei Forderungen an die Operatoren.

1. Die Operatoren sollen die Funktion möglichst gut auf das gröbere / feinere Gitter abbilden. Dazu fordern wir, dass die Masse $\int_{\Omega} f(x) dx$ der Funktion f bei einem Gitterwechsel erhalten bleibt. Das Integral der

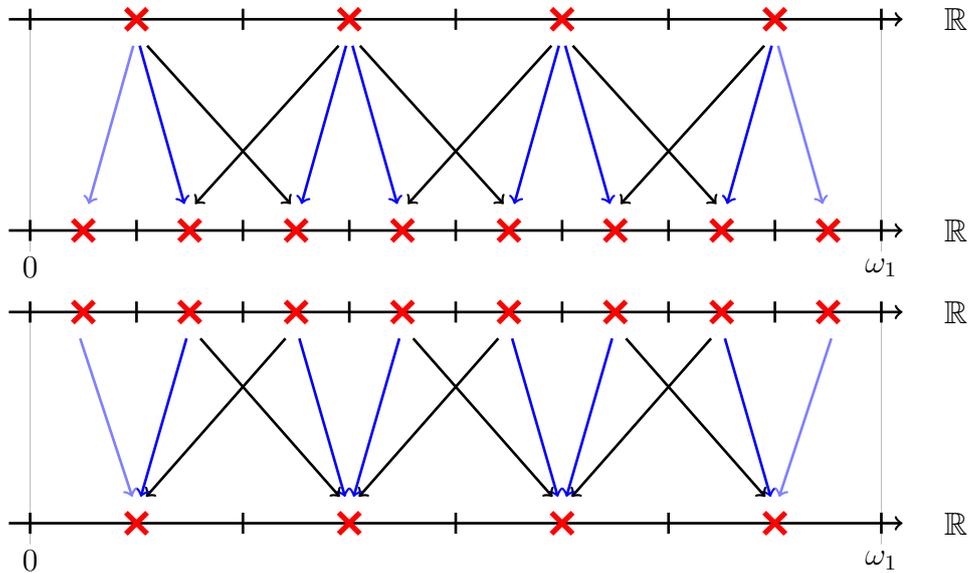


Abbildung 9.1: Illustration des Gitterwechsels 1D für Cellcentered-Gitter. *Oben:* Wechsel von grobem auf feines Gitter. Pfeile zeigen die Verteilung der Zellmasse an, deren Farbe codiert die Gewichtung. Blau = $3/4$, Schwarz = $1/4$, hellblaue Pfeile am Rand = 1 (Randbedingungen). *Unten:* Wechsel von feinem auf grobes Gitter. Gewichte: Blau = $3/8$, schwarz = $1/8$, hellblau = $1/2$

Durch die Randbedingungen haben die Operatoren nicht die Eigenschaft, Gitter der Auflösung h auf Gitter der Auflösungen $2h$ oder $h/2$ abzubilden,

$$P_H^h \mathbf{x}^{cc}(\Omega, m) \neq \mathbf{x}^{cc}(\Omega, 2m) \quad \text{und} \quad P_h^H \mathbf{x}^{cc}(\Omega, m) \neq \mathbf{x}^{cc}(\Omega, m/2).$$

Dies illustrieren wir an einem Beispiel:

Sei $\Omega = 8$ und $m = 8$, dann ist $\mathbf{x} = \mathbf{x}^{cc}(\Omega, m) = (0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5)^\top$. Somit gilt

$$P_H^h \mathbf{x} = (1.125, 3, 5, 6.875)^\top \neq (1, 3, 5, 7)^\top = \mathbf{x}^{cc}(\Omega, 4)$$

und

$$P_h^H \mathbf{x} = (0.5, 0.75, \dots, 7.25, 7.5)^\top \neq (0.25, 0.75, \dots, 7.25, 7.75)^\top = \mathbf{x}^{cc}(\Omega, 16).$$

Bei der Transformation von Gittern von einer groben auf eine feine Auflösung wirkt sich diese Eigenschaft nachteilig aus. In Abschnitt 9.4 werden wir uns damit befassen.

Schematisch ist in Abbildung 9.1 die Wirkungsweise beider Operatoren illustriert. Hierbei geben die Pfeile an, wohin die „Masse“ einer Zelle transportiert wird. Die Farbe der Pfeile codiert die Gewichtung der Anteile.

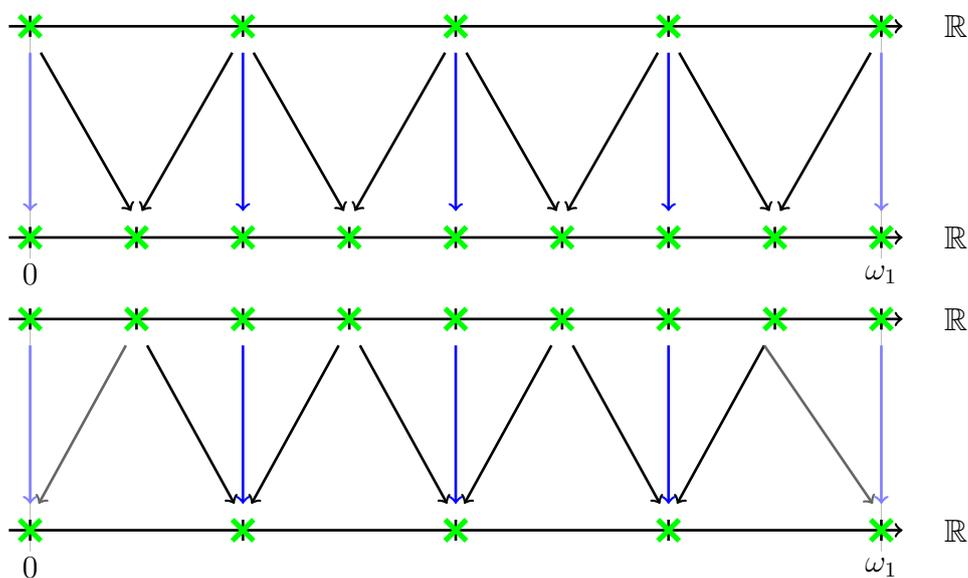


Abbildung 9.2: Illustration des Gitterwechsels 1D für Nodal-Gitter. *Oben:* Wechsel von grobem auf feines Gitter. Pfeile zeigen die Verteilung der Zellmasse an. Deren Farbe codiert die Gewichtung. Blau = 1, Schwarz = 1/2, hellblaue Pfeile am Rand = 1. *Unten:* Wechsel von feinem auf grobes Gitter. Gewichte: Blau = 1/2, schwarz = 1/4, hellblau = 2/3, hellgrau = 1/3

9.3 Gitterwechsel für Bilddaten

Wenden wir uns nun der Aufgabenstellung zu, mit Hilfe der definierten Operatoren gegebene Bilddaten auf Gitter unterschiedlicher Auflösung abzubilden. Durch die Beschränkung auf Zweierpotenzen bei der Definition der Operatoren im Eindimensionalen sind wir auf Bilddaten in diesen Größen angewiesen. Das vorgeschlagene Vorgehen ist folgendes. Sei $m = (m_1, m_2, m_3) = (2^{k_1}, 2^{k_2}, 2^{k_3})$ die Anzahl an Diskretisierungspunkten der gesuchten Deformation in der feinsten Auflösung. Daraus ergibt sich die Anzahl an Auflösungsstufen $k_{max} = \min\{k_1, k_2, k_3\}$. Dann werden die Bilddaten auf einem Cellcentered-Gitter dieser Auflösung ausgewertet und auf Auflösungen der Größe $m = m/2, m/4, \dots, m/2^{k_{max}}$ restringiert. Dazu nutzen wir die Matrix

$$\mathbf{P} = (P_H^h \otimes I_{m_2/2} \otimes I_{m_1/2}) (I_{m_3} \otimes P_H^h \otimes I_{m_1/2}) (I_{m_3} \otimes I_{m_2} \otimes P_H^h).$$

Bilddaten \mathbf{f}^k der Auflösung $m = 2^k$ erhalten wir durch mehrfache Anwendung dieser Matrix auf die Bilddaten feinsten Auflösung, wobei in jedem Schritt die Dimensionen der Matrix angepasst werden müssen. Die so gewonnenen Bilddaten fließen nun mittels Interpolation in das Bildmodell für diese Auflösungsstufe ein. Dabei bezeichnen wir mit f^k die zu den Bilddaten \mathbf{f}^k gehörige kontinuierliche Funktion. Wir illustrieren dies in Abbildung 9.3. Dort ist für die Testdaten die entstehende Folge von Bilddaten gezeigt. Beginnend oben links fällt die

Auflösung der Daten von links nach rechts, von oben nach unten.

Durch die Eigenschaft des Operators P_H^h , Informationen global zu verteilen, bleiben, anders als bei der direkten Auswertung der feinen Bilddaten auf einem groben Gitter, Bildinformationen erhalten. Beispielfhaft gezeigt ist dies in Abbildung 9.4. Dort ist links der Testdatensatz gezeigt, mit Hilfe der Restriktionsoperatoren auf eine Auflösung von $m = (8, 8)$ gebracht. Rechts daneben ist der gleiche Datensatz ohne vorherige Restriktion auf einem Gitter der Größe $m = (8, 8)$ ausgewertet. Deutlich ist der Unterschied zwischen den Bildern zu erkennen.

9.4 Gitterwechsel für Deformationen

Jetzt klären wir die Lösung der Aufgabe, Deformationen, die auf einem Gitter der Auflösung $2h$ gegeben sind, auf ein Gitter der Zellgröße h zu transformieren. Wir werden uns auf den Fall beschränken, dass die Deformation auf einem Staggered-Gitter gegeben ist.

Im eindimensionalen Fall haben wir gelernt, dass Cellcentered-Gitter durch die definierten Operatoren nicht auf ein Cellcentered-Gitter halber Auflösung abgebildet werden. Dies liegt an den Randbedingungen des Prolongationsoperators. Wir vermeiden diesen Fehler nun durch einen anderen Ansatz. Jede Transformation lässt sich als Summe der Identität und eines Inkrementes schreiben, $y(x) = x + u(x)$. Damit kann die Prolongation der Transformation als Summe der Prolongation der Identität und der Prolongation des Inkrementes bestimmt werden. Die Prolongation der Identität muss aber nicht bestimmt werden, da sie durch ein Gitter der passenden Auflösung gegeben ist.

Es ist also für ein gegebenes m und $H = \omega \circ m$ die Transformation \mathbf{y}^H gegeben. Dann ist

$$\mathbf{y}^h = u^h + \mathbf{x}^{stg}(\Omega, 2m)$$

und $h = \omega \circ (2m)$ und u^h mit den Abkürzungen

$$n_1 = (m_1 + 1)m_2m_3, \quad n_2 = m_1(m_2 + 1)m_3, \quad n_3 = m_1m_2(m_3 + 1)$$

und

$$\mathbf{u}_1^H = u_{1:n_1}^H, \quad \mathbf{u}_2^H = u_{n_1+1:n_2}^H, \quad \mathbf{u}_3^H = u_{n_1+n_2+1:n_1+n_2+n_3}^H$$

durch

$$\begin{aligned} \mathbf{u}_1^h &= (P_h^H(cc) \otimes I_{2m_2} \otimes I_{2m_1+1}) (I_{m_3} \otimes P_h^H(cc) \otimes I_{2m_1+1}) \\ &\quad (I_{m_3} \otimes I_{m_2} \otimes P_h^H(n)) \mathbf{u}_1^H \\ \mathbf{u}_2^h &= (P_h^H(cc) \otimes I_{2m_2+1} \otimes I_{2m_1}) (I_{m_3} \otimes P_h^H(n) \otimes I_{2m_1}) \\ &\quad (I_{m_3} \otimes I_{m_2+1} \otimes P_h^H(cc)) \mathbf{u}_2^H \end{aligned}$$

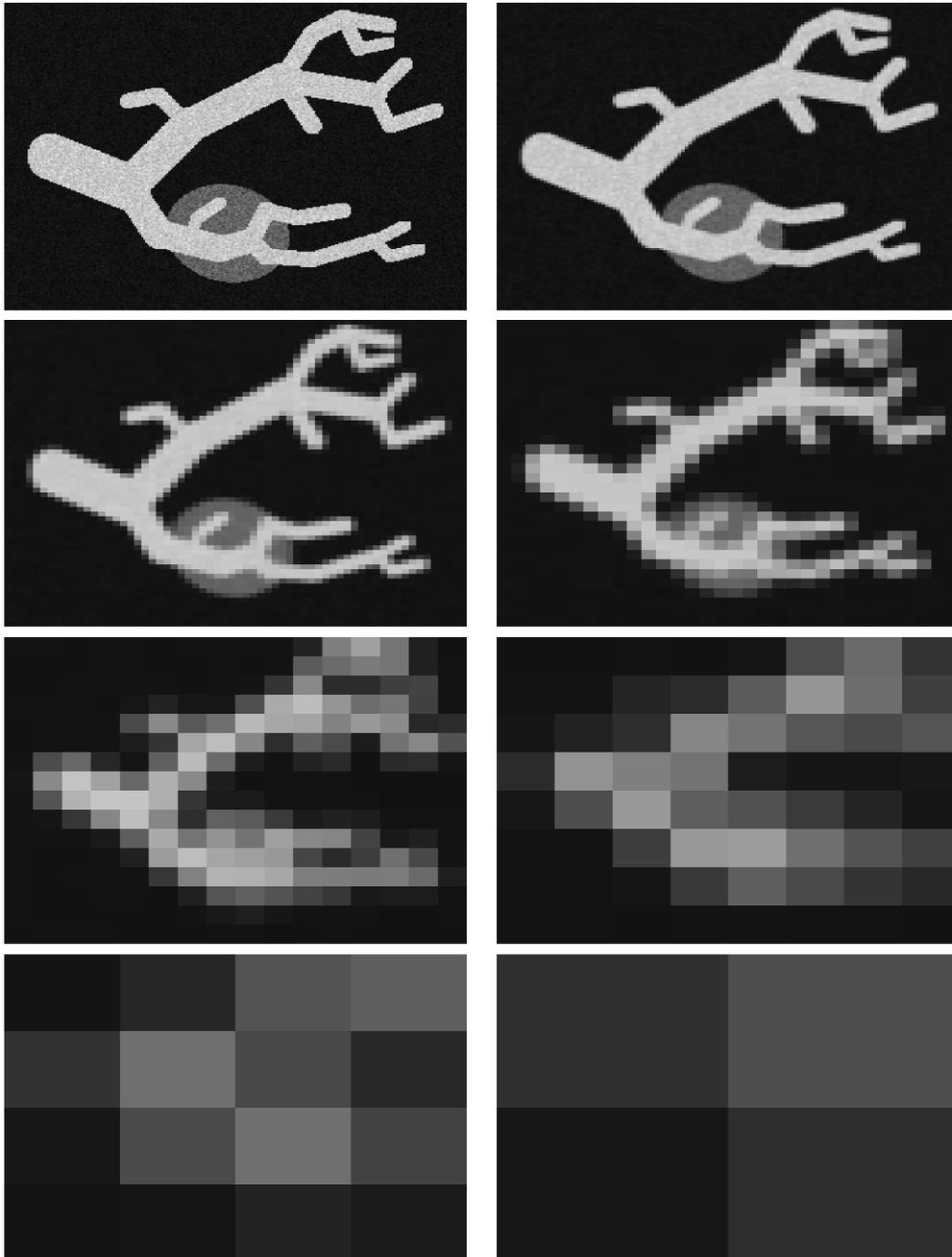


Abbildung 9.3: Multiresolution in 2D; *oben links*: Testdaten; *von links nach rechts, von oben nach unten*: Restriktion der Daten auf die Auflösungen $m = (128, 128), (64, 64), \dots, (2, 2)$

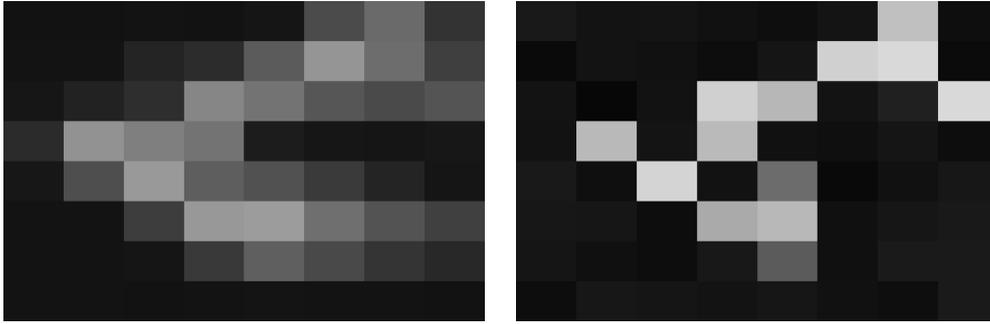


Abbildung 9.4: Gegenüberstellung der Auflösungsverminderung; *links*: Restriktion; *rechts*: Unterabtasten; im linken Bild ist die Struktur der hoch aufgelösten Daten erhalten

$$\mathbf{u}_3^h = (P_h^H(n) \otimes I_{2m_2} \otimes I_{2m_1}) (I_{m_3+1} \otimes P_h^H(cc) \otimes I_{2m_1}) \\ (I_{m_3+1} \otimes I_{m_2} \otimes P_h^H(cc)) \mathbf{u}_3^H$$

gegeben.

Hierbei ist $P_h^H(cc)$ ein 1D-Gitterwechseloperator für Cellcentered-Gitter und analog $P_h^H(n)$ für Nodal-Gitter.

Mit den beiden Wechselverfahren haben wir alle Bausteine zusammen, um ein Multiresolutionsverfahren zu definieren.

9.5 Verfahren

Wir erweitern den in Kapitel 8.4 beschriebenen Algorithmus zur nicht-parametrischen Registrierung:

Als obere und untere Auflösungsgrenzen werden k_{min} , k_{max} bestimmt, zwischen denen der Multiresolutionsansatz ablaufen soll, d.h. $m_{min} = 2^{k_{min}}$, $m_{max} = 2^{k_{max}}$. Zu jeder Auflösungsstufe werden die Bilddaten mittels des in Abschnitt 9.3 vorgestellten Verfahren angepasst. Beginnend auf der größten Auflösung wird als Vorregistrierung eine parametrische Registrierung (PIR) durchgeführt, vergleiche Kapitel 7, deren Lösung durch γ^{opt} auflösungsunabhängig gegeben ist und dazu verwendet wird, den Kern des Regularisiers auf jeder Auflösungsstufe zu bestimmen, indem ein Gitter der aktuellen Auflösung mittels γ^{opt} deformiert wird. Auf dem größten Level dient die Lösung der parametrischen Registrierung als Startwert für das in 8.4 beschriebene nicht-parametrische Registrierungsverfahren. Die Lösung \mathbf{y}^{opt} wird im folgenden Schritt auf die nächst feinere Auflösung transformiert mittels des in Abschnitt 9.4 beschriebenen Gitterwechsels und dient auf dieser Stufe als Startwert. Die Iteration bricht ab, wenn in der maximalen Auflösung eine Deformation bestimmt worden ist. Das Verfahren ist in Pseudocode in Algorithmus 6 dargestellt.

Algorithmus 6 Multiresolutionsansatz für nicht-parametrische Registrierung;
MLIR

Eingabe: \mathcal{T}, \mathcal{R}

Parameter: k_{min}, k_{max}

Ausgabe: $\gamma^{opt}, \mathbf{y}^{opt}$

```

1: for  $k = k_{min} : k_{max}$ 
2:    $m \leftarrow (2^k, \dots, 2^k); \mathbf{x}^h \leftarrow \mathbf{x}^{stg}(\Omega, m)$ 
3:    $\mathcal{T}^k, \mathcal{R}^k \leftarrow \text{Res}(\mathcal{T}, \mathcal{R}, k);$  vgl. (9.13)
4:   if  $k == k_{min}$  then
5:      $\gamma^{opt} \leftarrow \text{PIR}(\mathcal{T}^k, \mathcal{R}^k, \gamma^0, m)$ 
6:      $\mathbf{y}^0 \leftarrow y(\mathbf{x}^h, \gamma^{opt})$ 
7:   else
8:      $\mathbf{y}^0 \leftarrow \text{Prol}(\mathbf{y}^{opt});$  vgl. (9.14)
9:   end
10:   $\mathbf{y}^{\text{Kern}} \leftarrow y(\mathbf{x}^h, \gamma^{opt})$ 
11:   $\mathbf{y}^{opt} \leftarrow \text{NPIR}(\mathcal{T}^k, \mathcal{R}^k, \mathbf{y}^0, \mathbf{y}^{\text{Kern}})$ 
12: end

```

In Abbildung 9.5 ist die Wirkungsweise des Multiresolutionsansatzes illustriert. Gezeigt sind Zwischen- und Endergebnisse des Verfahrens. Für die Auflösungsstufen $m = (16, 16)$, $m = (64, 64)$ und $m = (256, 256)$ sind die resultierenden Deformationsfelder dargestellt. Zu erkennen ist, dass mit zunehmender Auflösung nur lokale und kleine Änderungen in der Transformation auftreten. Globale Anteile können auf grober Auflösung und mit wenig Diskretisierungspunkten in geringer Rechenzeit vorbestimmt werden.

Fazit

Abschließend mit dem Multiresolutionsverfahren haben wir in den letzten Kapiteln alle in Kapitel 2 aufgeworfenen Fragen beantwortet und ein nicht-lineares, allgemeines Registrierungsverfahren hergeleitet, das aufgrund seiner modularen Struktur für viele in der Praxis auftretende Probleme geeignet ist. Wir haben alle Aspekte dieses Verfahrens - Optimierung, Interpolation, Diskretisierung, Maße für Ähnlichkeit und Plausibilität, Multiresolutionsstrategie und Gleichungssystemlöser - ausführlich diskutiert.

Mit diesem Verfahren beenden wir den Teil über allgemeine Registrierungsverfahren und wenden uns jetzt Methoden zu, die diesen Ansatz spezialisieren.

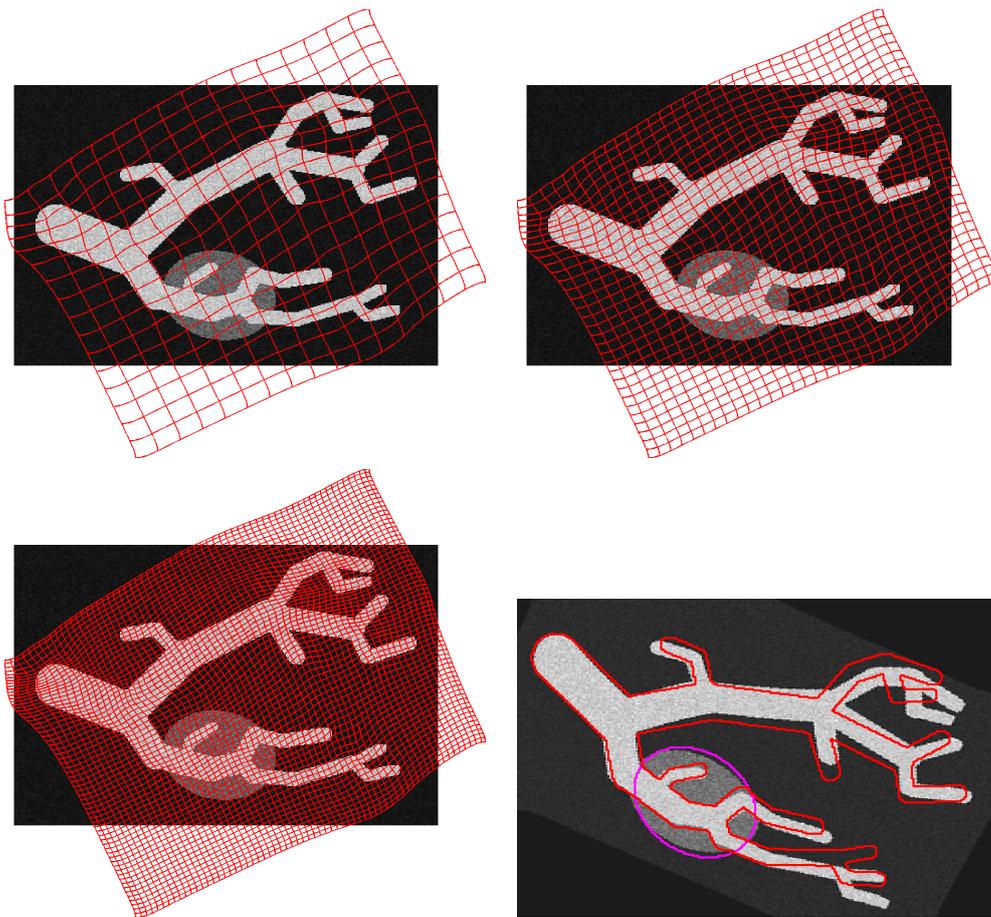


Abbildung 9.5: Illustration Multiresolution-Verfahren; Zwischen- und Endergebnisse der Methode; Auflösung der Deformationsfelder $m = (16, 16)$, $m = (64, 64)$ und $m = (256, 256)$; Parameter $\alpha = 0.3$, $\theta = 5$

Teil II

**Spezielle
Registrierungstechniken**

Kapitel 10

Skalenraumansatz

In diesem ersten Kapitel über spezielle Registrierungsmethoden befassen wir uns mit einem Registrierungsansatz, der auf eine spezielle Eigenschaft der Bilddaten abzielt. Bei beiden Bildern, Planungs- und Navigationsdaten, sind hauptsächlich Gefäßsysteme dargestellt, vergleiche Kapitel 5.5 und 14.1. Diese Systeme bilden röhrenartige Strukturen, so dass nur an wenigen Positionen in den Bildern registrierungsrelevante Informationen gegeben sind. Dies stellt sich aus zwei Gründen als nachteilig heraus:

- Die röhrenartigen Strukturen lassen sich nicht auf grob diskretisierte Gitter abbilden, vergleiche Abbildung 9.3, gerade Röhren kleinen Radius gehen verloren. Die Verwendung dieser Daten in Multiresolutionsansätzen ist damit verbesserungswürdig.
- Die Iteration im Gauß-Newton-Verfahren 3.2.3 wird durch den Gradienten der Zielfunktion getrieben, insbesondere vom Gradienten des Distanzmaßes. Aufgrund der Kettenregel ist ein Faktor dieses Gradienten immer durch $d\mathcal{T}_y$ gegeben, beispielsweise im SSD-Maß, $d\mathcal{D}(\mathbf{y}) = (\mathcal{T}_y - R)d\mathcal{T}_y$. Damit ist für röhrenartige Strukturen, die einen signifikanten Gradienten nur an den Röhrengrenzen haben, dieser Bereich im Verhältnis zum Gebiet Ω sehr klein. Auch dies ist verbesserungswürdig.

Dazu führen wir im nächsten Abschnitt Skalenräume und darauf basierend Multiskalenverfahren zur Registrierung ein.

10.1 Skalenräume

Die Nutzung von Skalenräumen in der Bildverarbeitung ist nicht neu. Sie findet sich hauptsächlich im Bereich der Bildglättung wieder [85]. Wir werden sie erstmals im Bereich der Registrierung röhrenartiger Strukturen nutzen.

Skalenräume basieren auf einer Folge von Bildern, die, beginnend bei dem zu registrierenden Originalbild, immer weniger Bildinformationen beinhalten. Es werden von Bild zu Bild mehr Informationen ausgeblendet. Dieses Ausblenden wird über einen Parameter gesteuert. Die Folge von Bildern nennen wir *Skalenraum*, der Parameter heißt *Skalenraumparameter*.

Ein Skalenraum ist auflösungsunabhängig, kann aber, wie wir später zeigen, mit einem Multiresolutionsverfahren kombiniert werden. Im Vergleich zu Multiresolutionsverfahren erlaubt ein Multiskalenansatz das Ausblenden von Informationen.

Eine exakte Definition eines Skalenraumes findet sich in der Literatur nicht, vergleiche [85].

Im Folgenden werden wir zwei Beispiele für Skalenräume geben, zeigen wie sich diese für die Registrierung nutzen lassen und dann ein kombiniertes Multiresolution-Multiskalen-Verfahren vorstellen.

Gaußscher Skalenraum

Der Gaußsche Skalenraum ist der wohl bekannteste Skalenraum in der Literatur [85]. Gegeben sein ein Bild $I : \Omega \rightarrow \mathbb{R}$, dann ist der Skalenraum durch die Faltung des Bildes mit einer Gaußschen Glockenkurve unterschiedlicher Standardabweichung gegeben, die als Skalenraumparameter dient. Da die Gaußsche Glockenkurve separiert, betrachten wir zur Erläuterung der Umsetzung des Skalenraumes den eindimensionalen Fall. Hier ist die Kurve durch

$$g_\sigma : \mathbb{R} \rightarrow \mathbb{R} \quad (10.1)$$

$$x \mapsto g_\sigma(x) = \frac{1}{2\sqrt{\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (10.2)$$

mit der Standardabweichung $\sigma \in \mathbb{R}^+$ gegeben. Die diskrete Form erhalten wir durch Abtasten der Funktion auf einem gegebenen um Null zentrierten endlichen Gitter und anschließender Normierung.

Der so entstehende Skalenraum ist in Abbildung 10.1 in der linken Spalte für ausgewählte Skalenraumparameter $\sigma \in \{0, 0.05, 0.25, 0.5, 1\}$ dargestellt. Es ist zu erkennen, wie die Gefäßstrukturen über das Bild verschmiert werden, was die im letzten Abschnitt beschriebenen Nachteile für die Ableitung des Distanzmaßes aufheben kann.

B-Spline-Skalenraum

Wie in Kapitel 5 über Interpolation bereits beschrieben, kann mittels regularisierter B-Spline-Interpolation ein Skalenraum beschrieben werden. Hierbei dient der Regularisierungsparameter θ als Skalenraumparameter. Auch dieser

Skalenraum ist in Abbildung 10.1 dargestellt, für $\theta \in \{0, 1, 10, 100, 1000\}$. Zu erkennen ist ein dem Gaußschen Skalenraum ähnliches Verschmieren der Bildinformationen.

In der Literatur existiert eine Vielzahl von weiteren Skalenräumen [85]. Wir beschränken uns auf die zwei vorgestellten Vertreter. Im Folgenden nutzen wir den B-Spline-Skalenraum zur Beschreibung eines Multiskalenverfahrens, da dieser nicht über zusätzliche Faltungsoperatoren zu erzielen, sondern durch die Wahl des Interpolators gegeben ist.

10.2 Multiskalenverfahren

Das Vorgehen in einem Multiskalenverfahren ist analog zu dem in einem Multi-resolutionsansatzes. Beginnend mit der Registrierung der Daten auf einer groben Skala wird die so bestimmte Lösung als Startwert auf der nächst feineren Skala verwendet. Diese Iteration wiederholt sich, bis die Registrierung der Originaldaten erreicht ist. Da ein Skalenraum unabhängig von der Diskretisierung m ist, ist diese über die Skalen konstant.

Der algorithmische Ablauf ist wie folgt:

1. Auswahl einer bestimmten Anzahl von Skalen über die Angabe der Skalenraumparameter.
2. Vorregistrierung auf der größten Skala.
Diese dient auf der gleichen Skala als Startwert für die nicht-lineare Registrierung und als Kern für die Regularisierung auf allen Skalen.
3. Das Ergebnis auf der größten Skala wird als Startwert auf der nächst feineren Skala verwendet. Dabei ist anders als im Multi-resolutionsverfahren kein Zwischenschritt (Prolongation) nötig, da die Diskretisierungsauflösung sich von Skala zu Skala nicht ändert.
4. Iteration des letzten Schrittes, bis ein Ergebnis auf der feinsten Skala vorliegt.

Für die Testdaten sind Zwischenergebnisse nach jeder Skala und das Endergebnis in Abbildung 10.2 visualisiert. Der Regularisierungs-Parameter wurde mit $\alpha = 800$ gewählt, der Skalenraum wurde von $\theta \in \{1000, 100, 10, 0\}$ aufgespannt, Auflösung ist $m = (256, 256)$. Dargestellt sind von links nach rechts, von oben nach unten, die Ergebnisse jeder Skala. Es wurden zur besseren Darstellung der Ergebnisse aber mit den jeweils bestimmten Deformationen die Originaldaten deformiert.

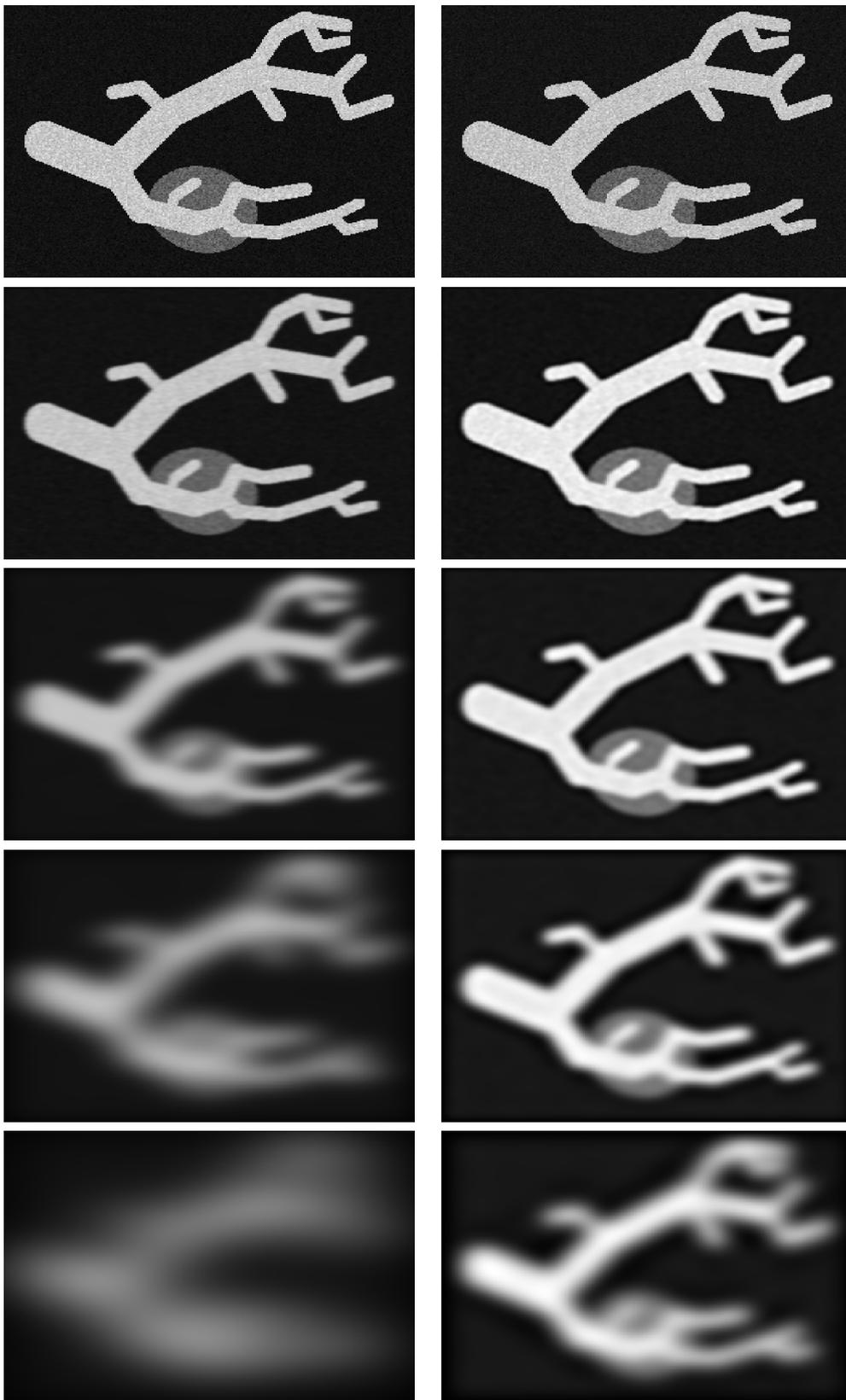


Abbildung 10.1: Beispiel Skalenräume; *links*: Gauß-Skalenraum mit Skalenparameter $\sigma = 0, 0.05, 0.25, 0.5, 1$; *rechts*: B-Spline-Skalenraum mit Skalenparameter $\theta = 0, 1, 10, 100, 1000$

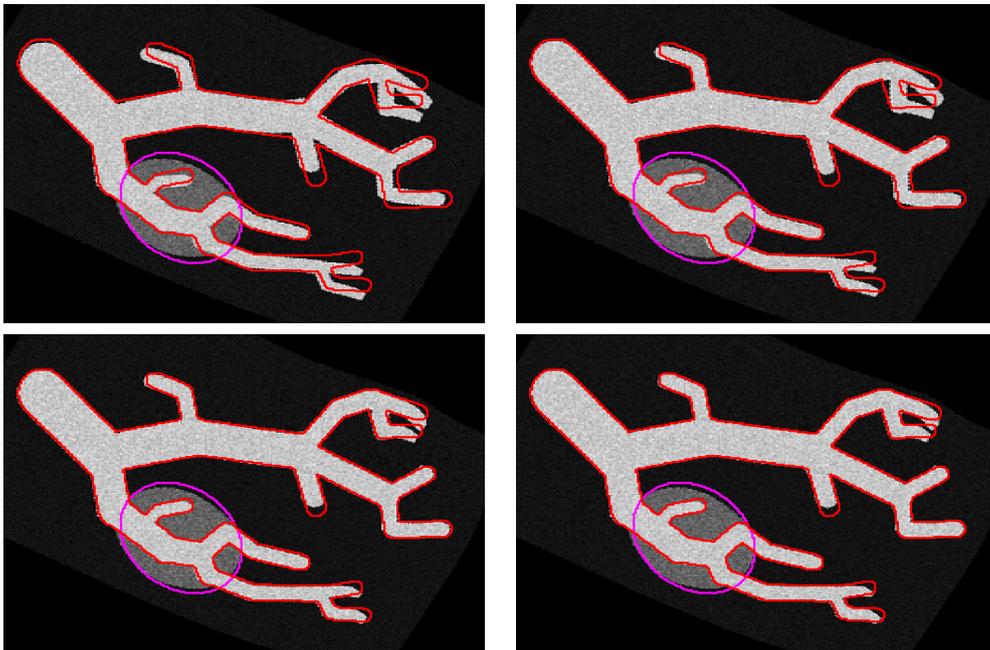


Abbildung 10.2: Ergebnis B-Spline-Skalenraum; ohne Multiresolution; $\alpha = 800$, $\theta = 1000, 100, 10, 0$

Der Nachteil einer derartigen Implementierung liegt in ihrer langen Laufzeit. Da keine Auflösungsverminderung der Daten stattfindet, werden auf jeder Skala Gleichungssysteme in der Größenordnung $n = 3m_1m_2m_3$ gelöst.

Um den Geschwindigkeitsnachteil auszugleichen, kombinieren wir den neu eingeführten Multiskalenansatz mit dem schon bekannten Multiresolutionsansatz.

10.3 Kombination von Multiskalen- und Multiresolutionsverfahren

Für die Kombination beider Verfahren gibt es prinzipiell zwei Möglichkeiten. Zum einen können zuerst die Daten für einen Multiresolutionsansatz mittels der in 9.3 beschriebenen Restriktion bestimmt werden. Für jede Auflösung der Daten kann dann ein Skalenraum erstellt werden.

Dies ist exemplarisch in Abbildung 10.3 dargestellt. Die Auflösungsreduktion verläuft von oben nach unten, der Skalenraum wird von links nach rechts aufgebaut. Wir erhalten eine zweidimensionale Matrix von Bilddaten.

Ein solcher Aufbau der Matrix ignoriert aber die Argumente für die Verwendung eines Multiskalenansatzes, die ja gerade durch Ausblenden oder Ver-

schmieren von Informationen eine gute Darstellung der Gefäßstrukturen auf groben Auflösungen erlauben soll. Wie in der Abbildung in den Bildern (3,2), (3,3) und (3,4) zu sehen ist, wird die gesamte Gefäßstruktur ohne erkennbare Informationen repräsentiert.

Die zweite Möglichkeit der Kombination von Multiskalen und Multiresolution umgeht dies, wenn zunächst für die feinste Auflösung der Daten ein Skalenraum aufgebaut wird und für jedes Element dieses Skalenraumes dann eine Auflösungsverminderung mittels Restriktion bestimmt wird.

Dies ist in Abbildung 10.4 gezeigt. Der Aufbau ist gleich der vorherigen Abbildung. Die Unterschiede zwischen beiden Möglichkeiten zeigen sich gerade in den zuvor angesprochenen Bildern (3,2), (3,3) und (3,4), die nun eine visuell erkennbare Ähnlichkeit zu den Ausgangsdaten besitzen.

Diese Argumentation spricht für die Verwendung der zweiten Möglichkeit, die wir im Folgenden nutzen werden.

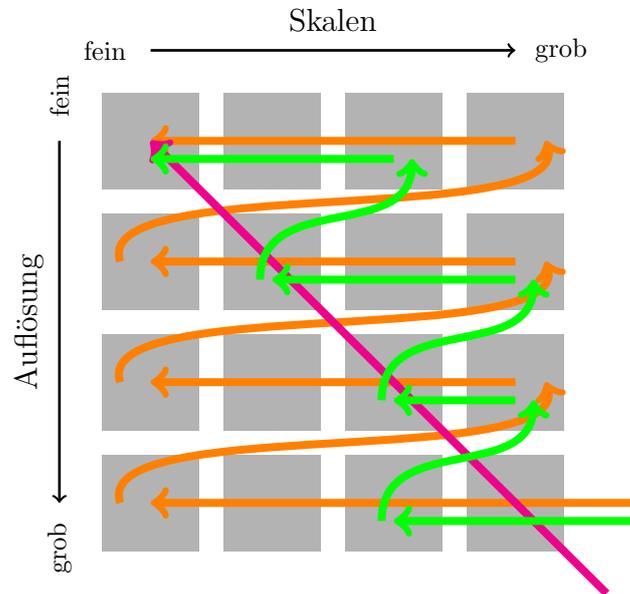
Durchlauf der Matrix

Nachdem wir den Aufbau der Matrix für den kombinierten Ansatz geschrieben haben, klären wir nun die Frage, wie diese Matrix durchlaufen wird, um ein kombiniertes Verfahren zu erreichen. Dabei sind vier Aspekte zu beachten:

1. Ziel ist die obere linke Ecke der Matrix.
2. Auf jeder Auflösungsstufe soll die Transformation in der Auflösung der Bilder liegen, wie wir im Kapitel über Multiresolutionsverfahren argumentiert haben. Die Auflösung von Transformation und Bilddaten ist also gekoppelt.
3. Aus numerischer Sicht sind Daten mit hoher Auflösung so spät wie möglich zu nutzen, da dies zu großen linearen Gleichungssystemen führt.
4. Beim Wechsel von einer hohen Auflösung der Transformation und damit der Bilddaten zu einer niedrigen Auflösung gehen Informationen verloren.

Daraus leitet sich ab, dass es keinen Schritt von Bilddaten höherer Auflösung zu solchen niedriger Auflösung geben soll.

Die verbleibenden Durchlaufmöglichkeiten sind anhand dreier Vertreter skizziert:



- Alle Elemente der Matrix werden durchlaufen. Es werden somit für jede Auflösung alle Skalen durchlaufen.
- Auf einer Auflösung werden nur bestimmte Skalen durchlaufen.
- Es wird pro Auflösung eine hiervon abhängige Skala genutzt. Dafür schlagen wir vor, die Matrix auf ihrer Hauptdiagonale zu durchlaufen. Dies wird durch die Tatsache, dass mit gröber werdender Skala weniger Informationen in den Bilddaten enthalten sind, die dann wiederum durch weniger Punkte repräsentiert werden, motiviert.

Verfahren

Zur Beschreibung eines Algorithmus nutzen wir die zuletzt beschriebene Durchlaufmöglichkeit. Dazu erweitern wir das in Kapitel 9.5 beschriebene Multiresolutionsverfahren. Die Erweiterung ist in Algorithmus 7 knapp skizziert. Zusätzlich ist ein Vektor von Skalenraumparametern θ gegeben, dessen Länge identisch mit der Anzahl der Multiresolutionsstufen ist. In Zeile 3 werden die Bilddaten in die jeweilige Skala transformiert, und auf diesen Daten dann die neue Stufe des Multiresolutionsverfahrens bestimmt. Der Ablauf mit parametrischer und nicht-parametrischer Registrierung bleibt erhalten.

Das Verfahren ist wiederum an den Testdaten erprobt worden. Dazu wurde ein B-Spline-Skalenraum verwendet. Als Regularisierungsparameter wurde $\alpha = 1000$ gewählt. Der Skalenraum ist durch $\theta = 10, 5, 1, 0$ aufgebaut worden. Die dazugehörigen Auflösungsstufen sind $m = (16, 16), (32, 32), (64, 64)$ und $(128, 128)$. Die Ergebnisse der Registrierung sind in Abbildung 10.5 präsentiert.

Algorithmus 7 Kombiniertes Multiskalen- und Multiresolutionsansatz für nicht-parametrische Registrierung; MSMLIR

Eingabe: \mathcal{T}, \mathcal{R}

Parameter: $k_{min}, k_{max}, \theta \in \mathbb{R}^{k_{max}-k_{min}+1}$

Ausgabe: \mathbf{y}_{opt}

```

1: for  $k = k_{min} : k_{max}$ 
2:    $m \leftarrow (2^k, \dots, 2^k)$ ;  $\mathbf{x}^h \leftarrow \mathbf{x}^{stg}(\Omega, m)$ 
3:    $\mathcal{T}_{\theta_k}, \mathcal{R}_{\theta_k} \leftarrow \text{Skala}(\mathcal{T}, \mathcal{R}, \theta_k)$ 
4:    $\mathcal{T}^k, \mathcal{R}^k \leftarrow \text{Res}(\mathcal{T}_{\theta_k}, \mathcal{R}_{\theta_k}, k)$ 
5:   if  $k == k_{min}$  then
6:      $\gamma^{opt} \leftarrow \text{PIR}(\mathcal{T}^k, \mathcal{R}^k, \gamma^0, m)$ 
7:      $\mathbf{y}^0 \leftarrow y(\mathbf{x}^h, \gamma^{opt})$ 
8:   else
9:      $\mathbf{y}^0 \leftarrow \text{Prol}(\mathbf{y}^{opt})$ 
10:  end
11:   $\mathbf{y}^{Kern} \leftarrow y(\mathbf{x}^h, \gamma^{opt})$ 
12:   $\mathbf{y}^{opt} \leftarrow \text{NPIR}(\mathcal{T}^k, \mathcal{R}^k, \mathbf{y}^0, \mathbf{y}^{Kern})$ 
13: end

```

Dargestellt ist jeweils das Ergebnis der Registrierung auf den jeweiligen Stufen des Verfahrens. Um eine bessere Visualisierung zu erzielen, sind die einzelnen Deformationen auf die volle Auflösung prolongiert worden. Das Endergebnis (unten rechts) stellt subjektiv das bisher beste Ergebnis dar.

Fazit

Zur Aufarbeitung der Daten haben wir - erstmalig im Bereich der Registrierung von Leberdaten - Skalenräume eingeführt und mit diesen den bestehenden Ansatz um Multiskalenverfahren erweitert. Eine reine Anwendung von Multiskalenverfahren erlaubt zwar das Verschmieren der Bildinformationen, bringt aber keinen Zeitgewinn. Erst die Kombination von Multiresolutions- und Multiskalenverfahren erzielt subjektiv gute Ergebnisse bezüglich Rechenzeit und Qualität der Registrierung.

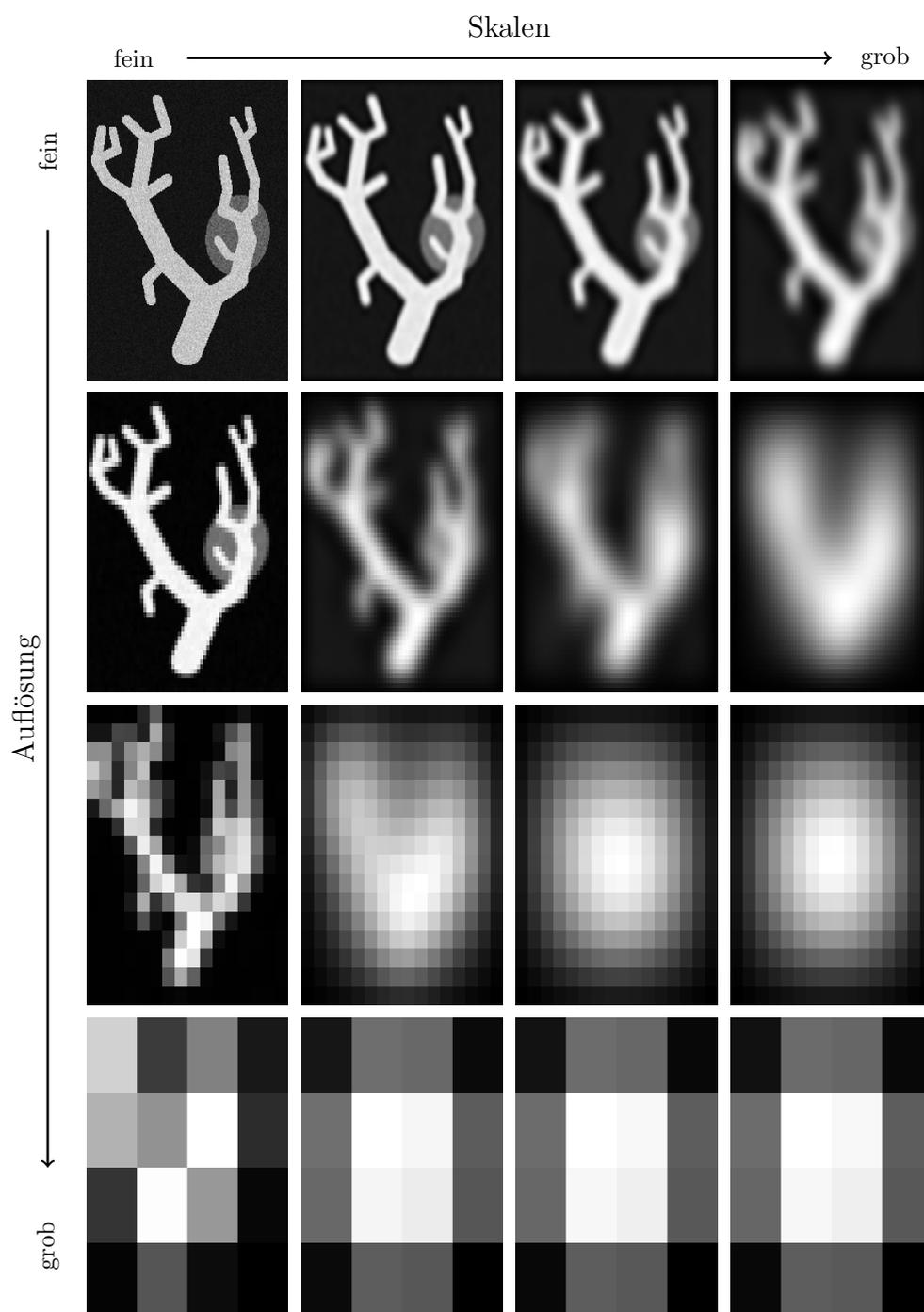


Abbildung 10.3: Kombination von Multiresolutions- und Multiskalenansatz; Multiresolution auf den Originaldaten, dann pro Stufe ein Skalenraum; Zur besseren Darstellung sind die einzelnen Bilder um 90° gedreht; *von oben nach unten*: Änderung der Auflösung $m = (256, 256), (64, 64), (16, 16), (4, 4)$; *von links nach rechts*: B-Spline-Skalenraum auf der jeweiligen Auflösung mit Parameter $\theta = 0, 10, 100, 1000$

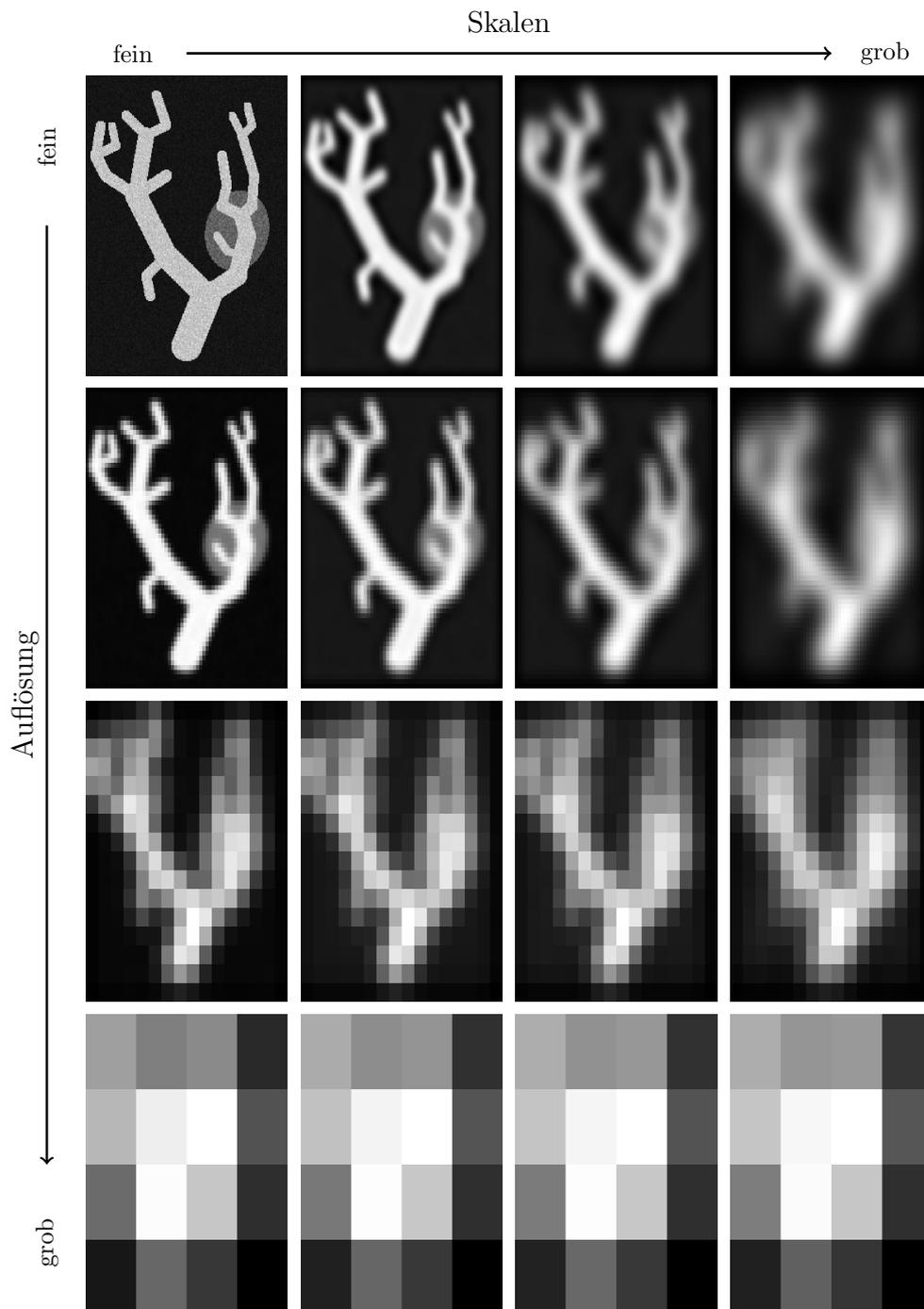


Abbildung 10.4: Neue Kombination von Multiresolutions- und Multiskalenansatz; Skalenraum auf den Originaldaten, dann Multiresolution auf den Stufen des Skalenraumes; Zur besseren Darstellung sind die einzelnen Bilder um 90° gedreht; *von oben nach unten*: Änderung der Auflösung $m = (256, 256), (64, 64), (16, 16), (4, 4)$; *von links nach rechts*: B-Spline-Skalenraum auf der jeweiligen Auflösung mit Parameter $\theta = 0, 10, 100, 1000$

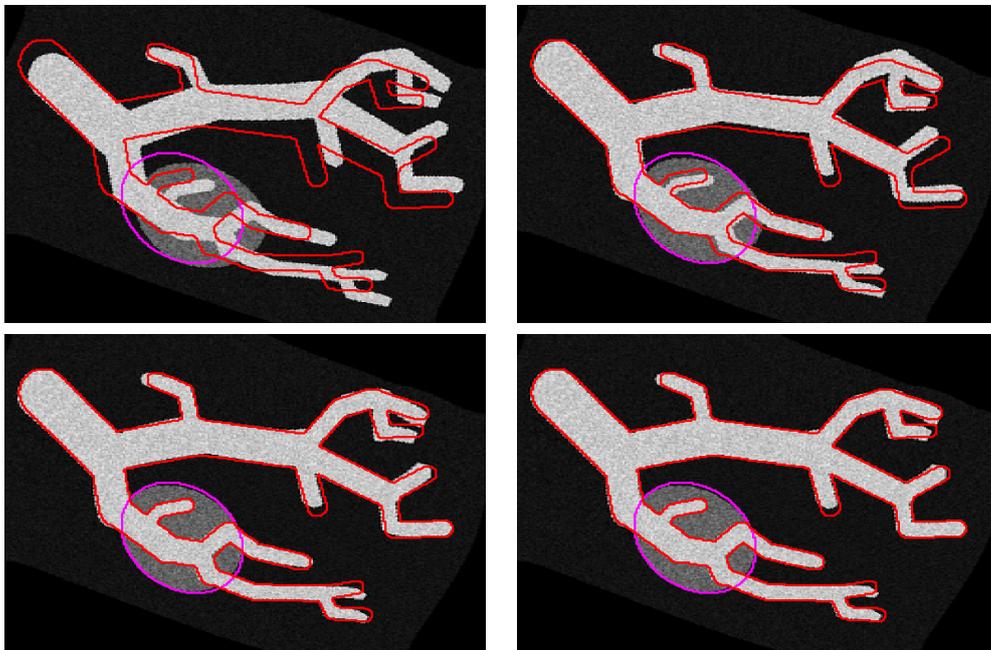


Abbildung 10.5: Ergebnis B-Spline-Skalenraum; mit Multiresolution; $\alpha = 1000$, $\theta = 10, 5, 1, 0$, $m = (16, 16), (32, 32), (64, 64), (128, 128)$

Kapitel 11

Fokussierung

Wir werden nun eine neue Multiresolutionsstrategie vorstellen und untersuchen. Diese Strategie führt zu einem großen Zeitgewinn bei der Registrierung, wenn der Anwender nur an ausgewählten Bereichen der Daten interessiert ist. Am Beispiel der Leberchirurgie verdeutlicht sich dies:

Während der Operation ist für den Chirurgen eine genaue Navigation in der Umgebung seiner Instrumente von Interesse. Für diesen Bereich ist eine genaue Registrierung der Planungsdaten mit den Navigationsdaten nötig. Entfernt liegende Gebiete, wie etwa die andere Seite des Organs oder benachbarte Organe sind nicht von aktuellem Interesse. Das folgende Verfahren berücksichtigt dies.

Bisher stellt sich die Registrierungsproblematik wie folgt dar:

Gegeben sind zwei Datensätze, die als kontinuierliche Funktionen interpretiert werden, gesucht ist eine Transformation, die angewendet auf den einen Datensatz diesen ähnlich zum Anderen deformiert. Für diese Transformation können wir eine Auflösung vorgeben, die unabhängig von der Auflösung gewählt werden kann, in der die Daten gegeben sind. Dies ist dann nützlich, wenn die Registrierung zeitkritisch ist, das heißt, das Ergebnis schnell bestimmt werden soll. Wird die Auflösung der Transformation gröber als die der Bilddaten gewählt, so verringert sich die Zahl der Unbekannten und die auftretenden linearen Gleichungssysteme sind schneller zu lösen.

Eine weitere Möglichkeit, die Registrierung zu beschleunigen, ist die Anwendung von Multiresolutionsansätzen. Diese verfeinern die Auflösung der Deformation bis zu einer Größe, die für die Anwendung nötig ist.

Ziehen wir nun in Betracht, dass die für den Anwender relevanten Gebiete klein sind im Verhältnis zum Gebiet Ω , auf dem die Daten gegeben sind, so müssen wir festhalten, dass zu hohen Zeitkosten die feinaufgelöste Registrierung für die restlichen Gebiete mitbestimmt wird. An dieser Stelle wird das neue Verfahren ansetzen.

11.1 Neuer Ansatz

Wir beschreiben zunächst den Aufbau und Ablauf des Ansatzes und diskutieren dann seine Vor- und Nachteile. Das Verfahren ist erstmalig in [61] präsentiert worden.

Sei

$$\Upsilon_0 = (v_1^1, v_2^1) \times (v_1^2, v_2^2) \times (v_1^3, v_2^3) \subset \Omega$$

das vom Anwender ausgewählte Zielgebiet innerhalb des Bildbereiches und sei durch $h^0 \in \mathbb{R}^3$ die Auflösung in diesem Gebiet vorgegeben. Sei $m = (m_1, m_2, m_3)$ mit $m_j = \lceil h_j / (v_2^j - v_1^j) \rceil$ und seien $\mathbf{x}^{cc}(\Upsilon_0, m)$, $\mathbf{x}^{stg}(\Upsilon_0, m)$ und $\mathbf{x}^n(\Upsilon_0, m)$ die in Kapitel 4 beschriebenen Gitterdefinitionen für das Gebiet Υ_0 (die Techniken zur Gittererzeugung sind identisch, sämtliche Gitterwechseloperatoren können ohne Änderung übernommen werden).

Wir stellen den Algorithmus unter mehreren Annahmen vor, die die Notation vereinfachen und das Verständnis erleichtern. Diese Annahmen sind nicht zwingend und lassen sich aufweichen:

- Es wird *ein* Zielgebiet festgelegt.
- Das Zielgebiet ist quaderförmig mit entlang der Koordinatenachsen laufenden Kanten.
- Das Zielgebiet wird gleichmäßig vergrößert.
- Die Anzahl der Diskretisierungspunkte ist auf jeder Stufe fest.

Ausgehend vom Zielgebiet Υ_0 leiten wir nun sukzessiv Gebiete $\Upsilon_k, k = 1, 2, \dots, N_\Upsilon$ her mit $\Upsilon_0 \subset \Upsilon_1 \subset \dots \subset \Upsilon_{N_\Upsilon}$, die konzentrisch um das Gebiet Υ_0 liegen. Wir setzen dazu

$$p^\Upsilon = \left(\frac{v_1^1 + v_2^1}{2}, \frac{v_1^2 + v_2^2}{2}, \frac{v_1^3 + v_2^3}{2} \right)$$

und

$$l^\Upsilon = (v_2^1 - v_1^1, v_2^2 - v_1^2, v_2^3 - v_1^3)$$

und damit

$$\begin{aligned} \Upsilon_k &= \left(p_1^\Upsilon - c^k \frac{l_1^\Upsilon}{2}, p_1^\Upsilon + c^k \frac{l_2^\Upsilon}{2} \right) \\ &\times \left(p_2^\Upsilon - c^k \frac{l_2^\Upsilon}{2}, p_2^\Upsilon + c^k \frac{l_2^\Upsilon}{2} \right) \times \left(p_3^\Upsilon - c^k \frac{l_3^\Upsilon}{2}, p_3^\Upsilon + c^k \frac{l_3^\Upsilon}{2} \right). \end{aligned} \quad (11.1)$$

Hierbei ist $c \in \mathbb{R}^{>1}$ der Faktor, der die Gebietsgrößen reguliert. Setzen wir beispielsweise $c = \sqrt{3}$ verdoppelt sich das Volumen des Gebietes von Folgeglied zu Folgeglied. Der Gitteraufbau ist - zur Veranschaulichung für den eindimensionalen Fall - in Abbildung 11.1 und 11.2 dargestellt. In 11.1 wird gezeigt,

wie sich der Schritt von Gebiet Υ_k zu Υ_{k+1} vollzieht, hierbei ist $c = 1.5$ und in Abbildung 11.2 ist die Folge $\Upsilon_k, k = 0, 1, 2, 3$ für ein gegebenes Gebiet Υ_0 dargestellt.

Die Folge bricht ab, wenn eine Gebietsgrenze außerhalb des Bildbereiches Ω liegt.

Die Folge $(\Upsilon_0, \dots, \Upsilon_{N_\Upsilon})$ von Gebieten wird rückwärts $k = N_\Upsilon, \dots, 1$, für einen modifizierten Multiresolutionsansatz genutzt. Für jedes Gebiet Υ_k erhalten wir durch

$$\mathbf{x}_k = \mathbf{x}^{cc}(\Upsilon_k, m) \quad \text{oder} \quad \mathbf{x}_k = \mathbf{x}^{stg}(\Upsilon_k, m)$$

eine Diskretisierung, die Grundlage der Registrierung ist. Wie bei dem herkömmlichen Registrierungsansatz dient die Lösung \mathbf{y}_k der Registrierung auf dem Gebiet Υ_k als Startwert für die Registrierung auf dem Gebiet Υ_{k-1} . Zu klären ist nun, wie die Deformation \mathbf{y}_k auf das Gebiet Υ_{k-1} transformiert wird.

Gegeben sind die Werte $\bar{\mathbf{y}}_k$ an den Positionen $\bar{\mathbf{x}}_k$. Gesucht sind die Werte $\bar{\mathbf{y}}_{k-1}$ an den Positionen $\bar{\mathbf{x}}_{k-1}$. Dies ist die Formulierung eines Interpolationsproblems, für das wir (in abgewandelter Form) in Kapitel 5 Lösungswege beschrieben haben.

Wir illustrieren dies an einem zweidimensionalen Beispiel. In Abbildung 11.3 sind die Gebiete $\Upsilon_k, \Upsilon_{k+1}$ eingezeichnet. Abbildung 11.3, Mitte zeigt die zu den Gebieten gehörenden Diskretisierungen. Zu erkennen ist, dass bei gleicher Anzahl m von Diskretisierungspunkten durch die Verminderung der Gebietsgröße die Diskretisierungsauflösung fällt. In der Abbildung rechts ist die auf dem Gebiet Υ_{k+1} bestimmte Deformation dargestellt. Innerhalb sehen wir den durch Interpolation neu bestimmten Startwert auf dem Gebiet Υ_k .

Die Kombination dieses neuen Multiresolutionsverfahrens mit den bisher beschriebenen Registrierungsbausteinen kann auf verschiedene Art geschehen. Wir schlagen folgenden Algorithmus vor:

- Die Anzahl m der Diskretisierungspunkte wird so gewählt, dass die benötigte Auflösung des Zielgebietes Υ_0 erreicht ist.
- Die vollständigen Daten werden affin-linear vorregistriert.

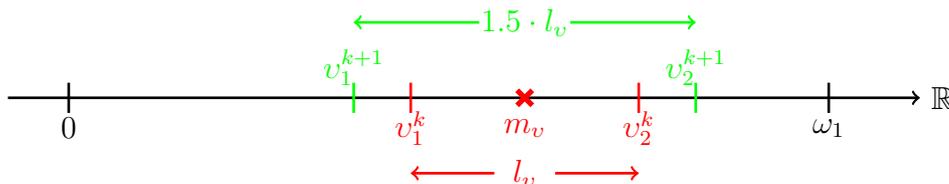


Abbildung 11.1: Vergrößerung des Zielgebietes in 1D

- Beginnend auf dem größten Gebiet Υ_{N_Υ} wird sukzessiv auf jedem Gebiet $\Upsilon_k, k = N_\Upsilon, \dots, 0$ eine nicht-parametrische Registrierung durchgeführt.
- Als Startwert dient auf jedem Gitter die mittels Interpolation an das aktuelle Gitter angepasste Lösung der Registrierung des vorherigen Gebietes.
- Als Startwert auf dem größten Gebiet dient das Ergebnis einer affinen Registrierung (Kapitel 7.3).
- Als Kern des Regularisierers nutzen wir ebenso das Ergebnis einer affinen Registrierung.

Dieser Algorithmus ist unter Algorithmus 8 in einer Kurzschreibweise zusammengefasst.

Algorithmus 8 Fokussierungsschema; FOKUS

Eingabe: $\mathcal{T}, \mathcal{R}, \Upsilon_0, m$

Ausgabe: $\mathbf{y}_\Upsilon^{opt}$

- 1: $\gamma^{opt} \leftarrow \text{PIR}(\mathcal{T}, \mathcal{R}, \gamma^0, m)$
 - 2: **for** $k = N_\Upsilon : -1 : 0$
 - 3: $\Upsilon_k \leftarrow (11.1); \mathbf{x}^k \leftarrow \mathbf{x}^{stg}(\Upsilon_k, m)$
 - 4: **if** $k == N_\Upsilon$ **then**
 - 5: $\mathbf{y}_k^0 \leftarrow y(\gamma^{opt}, \mathbf{x}_k)$
 - 6: **else**
 - 7: $\mathbf{y}_k^0 \leftarrow \text{Interp}(\mathbf{y}_{k+1}^{opt}, \mathbf{x}^k)$
 - 8: **end**
 - 9: $\mathbf{y}_k^{\text{Kern}} \leftarrow y(\gamma^{opt}, \mathbf{x}_k)$
 - 10: $\mathbf{y}_k^{opt} \leftarrow \text{NPIR}(\mathcal{T}, \mathcal{R}, \mathbf{y}_k^0, \mathbf{y}_k^{\text{Kern}})$
 - 11: **end**
-

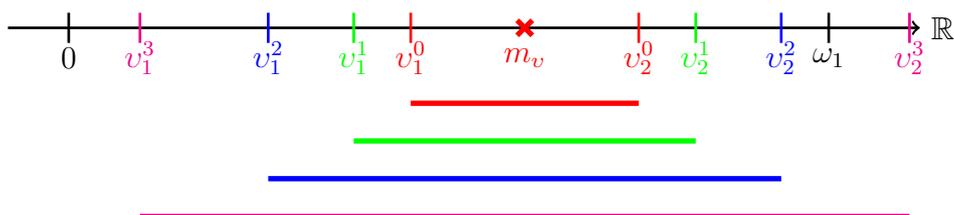


Abbildung 11.2: Verschachtelte Zielgebiete in 1D

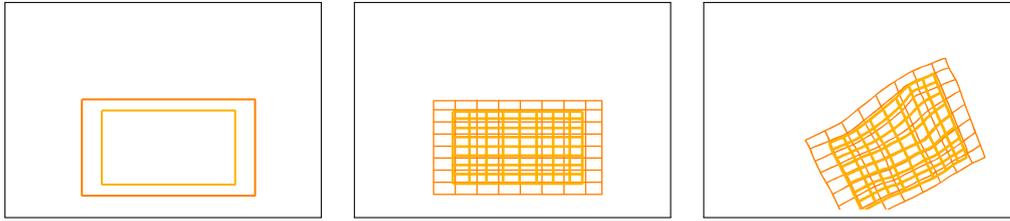


Abbildung 11.3: Darstellung Gitterwechsel 2D; *links*: Position der Gebiete; *Mitte*: Initialisierung der Gebiete durch Cellcentered-Gitter; *rechts*: orange - Lösung auf dem großen Gebiet; gelb - Startwert auf dem kleinen Gebiet

11.2 Ergebnisse

Wir demonstrieren den Ablauf des Algorithmus an den zweidimensionalen Testdaten. In Abbildung 11.4 ist oben links das vom Anwender festgelegte Zielgebiet in die Referenzdaten eingezeichnet. Daneben sind die sukzessiv wachsenden Zielgebiete $\Upsilon_k, k = 1, \dots, N_\Upsilon$ visualisiert. Rechts ist der Inhalt des Zielgebietes vergrößert dargestellt. Dieser dient später der Bewertung des Registrierungsresultates. In der zweiten Zeile ist links das Ergebnis der parametrischen Registrierung visualisiert, das als Startwert auf dem größten Gebiet dient. Dies ist durch den Algorithmus in Abschnitt 7.3 bestimmt worden. Im Bild der Zeilenmitte ist die Lage dieses Gitters in Relation zu den Templatedaten gestellt. Rechts ist das Ergebnis $\mathbf{y}_{N_\Upsilon}^{opt}$ auf diesem Gebiet dargestellt, das als Startwert auf dem nächsten Gebiet dient. In der dritten Zeile ist links gezeigt, wie das neue Startgitter mittels Interpolation aus der bekannten Lösung bestimmt wird. Die anderen Bilder dieser Zeile sind analog zur vorherigen. In der nächsten Zeile ist dies für das Zielgebiet dargestellt. Um Platz zu sparen sind die Zwischenergebnisse für einige Gebiete weggelassen worden. In der letzten Zeile ist links das Templatebild ausgewertet an den Positionen des Registrierungsresultates \mathbf{y}_0^{opt} auf dem Zielgebiet. Eine deutliche Übereinstimmung mit dem Zielgebiet des Referenzbildes ist zu erkennen. In der Mitte ist dieses Bild in das Gebiet Ω_R des Referenzbildes zusammen mit den Kantenvergleichsdaten visualisiert. Auch hier zeigt sich die gute Qualität der Ergebnisse. Das Bild unten rechts interpretieren wir später.

Für zwei weitere Zielbereiche sind die gleichen Zwischen- und Endergebnisse in den Abbildungen 11.5 und 11.6 dargestellt. In Abbildung 11.5 sind die Ergebnisse von ähnlicher Qualität wie im ersten Fall. Im dritten Fall ist sogar zu erkennen, dass der Algorithmus in den Gebieten, in denen die herkömmliche Registrierung Schwierigkeiten hat, das plausible Ergebnis zu finden, eine andere Lösung für diesen Ausschnitt findet, vergleiche Kapitel 9.5.

In Abbildung 11.7 sind die Registrierungsresultate denen des herkömmlichen Multiresolutionsverfahrens aus Kapitel 9 gegenüber gestellt. Das neue Verfahren liefert ähnliche Ergebnisse wie das herkömmliche Multiresolutions-Ver-

Tabelle 11.1: Rechenzeiten der Fokussierungsstrategie für reale Daten

Volumen	Auflösung	abs. Zeit	rel. Zeit
Ganzes Bild	(32, 32, 32)	191.2s	100%
Zielgebiet außen	(24, 24, 24)	11.38s	5.9%
	(16, 16, 16)	7.3s	3.8%
Zielgebiet innen	(30, 30, 20)	29.54s	15.5%
	(16, 16, 16)	10.39s	5.4%
	(8, 8, 8)	1.98s	1.1%

fahren. Für den dritten Testfall (dritte Spalte) sind diese sogar subjektiv besser.

Die Stärke der neuen Strategie zeigt sich bei dreidimensionalen Daten. Um dies an dieser Stelle der Arbeit zu erläutern, greifen wir auf Experimente vor, die erst in Kapitel 14.2.5 beschrieben werden. Wir nutzen einen realen klinischen Datensatz. Ein Anschnitt des Datensatzes zusammen mit zwei Zielgebieten ist in Abbildung 11.8, links dargestellt. Die Zielgebiete sind quaderförmig. Mit welchen Algorithmen und auf welchem Testrechner die Ergebnisse erzielt worden, ist in Kapitel 14.2.4 und 14.2.5 beschrieben. Wir interessieren uns für die benötigte Rechenzeit.

In der Tabelle 11.1, sind für drei Gebiete und verschiedene Auflösungen die Zeiten dargestellt, absolut und in Relation zur Zeit für die Registrierung des gesamten Volumens. Die erste Zeile dient als Referenzgröße. Die Zielgebiete sind so diskretisiert, dass die Auflösung der Deformation innerhalb feiner ist als in der Referenzgröße. Die erzielten Zeiten für die Zielgebiete unterbieten drastisch die Laufzeit des herkömmlichen Verfahrens, sind aber dennoch von subjektiv guter Qualität (vergleiche Kapitel 14.2.5).

Das bisher beschriebene Vorgehen nutzt die Informationen der vorherigen Lösung nur in den Bereichen des neuen Zielgebietes. Wählt man die Randbedingungen des Regularisierers \mathcal{S} als Dirichlet-Null-Randbedingungen, so wird das Deformationsfeld auf dem Rand festgehalten. Damit greifen die einzelnen Deformationen ineinander, wobei sie auf den Rändern der Gebiete gleiche Werte haben. Damit kann eine immer feiner werdende globale Deformation bestimmt werden, die dann zur Berechnung eines annähernd vollständigen Template-Bildes genutzt werden kann, bei dem die Zellen entsprechend den unterschiedlichen Gebieten auch unterschiedliche Auflösung besitzen. In den einzelnen Abbildungen ist unten rechts das so bestimmte Bild dargestellt.

11.3 Eigenschaften

Die Vorteile des Verfahrens sind:

- Für dreidimensionale Daten haben wir gezeigt, dass das Verfahren sehr schnell die Registrierung innerhalb des Zielgebietes berechnen kann. Die gemessenen Zeiten lassen darauf schließen, dass damit nicht-parametrische Verfahren unter Echtzeitanforderungen genutzt werden können, wenn die Voraussetzung - die Wahl eines Zielgebietes - erfüllt ist.
- Das Verfahren benutzt nur Bausteine der Registrierung, die wir schon vorgestellt haben.
- Durch die Fokussierungs- oder Zoomeigenschaft des Verfahrens ist es möglich, globale Bildinformationen in das Registrierungsergebnis einfließen zu lassen.
- Das Zielgebiet muss nur im Referenzbild vorgegeben werden. Das Verfahren findet den dazu korrespondierenden Bereich im Templatebild automatisch.
- Die Anzahl an Diskretisierungspunkten kann so gewählt werden, dass eine beliebig feine Auflösung der Transformation im Zielgebiet erreicht wird.

Nachteile des Verfahrens sind:

- Der Geschwindigkeitszuwachs hängt vom Größenverhältnis des Zielgebietes zum Gebiet Ω ab. Ist das Zielgebiet relativ sehr klein, so ist, wie beschrieben, der Geschwindigkeitszuwachs enorm. Ist es aber nicht möglich, ein kleines Zielgebiet zu wählen, fällt der Zuwachs gering aus.
- Für das Verfahren existiert keine theoretische Fundierung. Seine Nützlichkeit ist zur Zeit rein experimentell. In den drei obigen Beispielen ist dies gezeigt worden, auch für dreidimensionale Daten ist dies erfolgt. Eine garantierte Übereinstimmung mit den Ergebnissen herkömmlicher Methoden ist aber nicht zu gewährleisten, wobei aber auch deren Lösung nicht als „wahre“ Registrierung angesehen werden kann.

Fazit

Das neue Verfahren ist, um eine klare Definition zu geben, unter vier Annahmen beschrieben worden, die jedoch aufzuweichen sind: Das Zielgebiet muss kein zu den Koordinatenachsen paralleler Quader sein, vergleiche Abbildung 11.9, oben links. Es kann durch einen geschlossenen Polygonzug begrenzt sein, Abbildung

11.9, unten rechts. Für den ersten Fall ist eine Vergrößerungsstrategie trivial; für den zweiten Fall kann in einem ersten Schritt der Polygonzug mit einem ihn umfassenden Quader minimalen Volumens eingefasst werden, womit wir das Problem auf den ersten Fall zurückgeführt haben. Wir können mehrere Zielgebiete wählen, die dann sukzessiv verarbeitet werden müssen. Der Vergrößerungsfaktor kann von Stufe zu Stufe variabel sein.

Wir haben damit eine neue Multiresolutionsstrategie präsentiert, die unter der Voraussetzung einer vom Benutzer zuvor erfolgten Definition eines Zielgebietes einen großen Zeitvorteil erzielt, dabei aber die Qualität der Ergebnisse beibehält. Die Tauglichkeit des Verfahrens werden wir in Kapitel 14 an klinischen Daten demonstrieren.

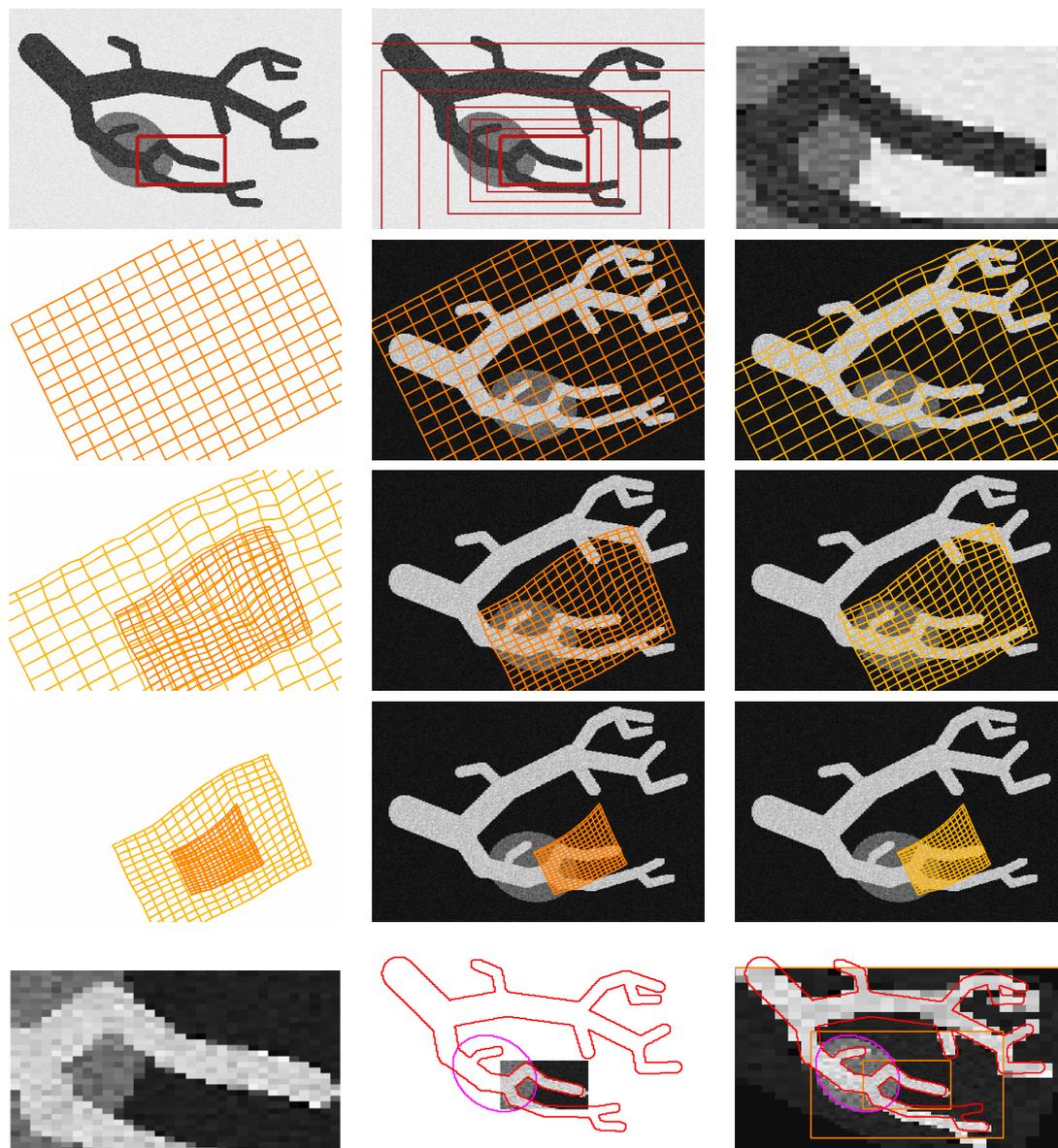


Abbildung 11.4: Erstes 2D-Beispiel ; Parameter $\alpha = 0.1$; NGF; 7 ineinander greifende Gebiete; nur die Gebiete 7, 4 und 1 sind gezeigt; Als Skalenraum wurde ein B-Spline-Raum verwendet mit dem Skalenraumparameter $\theta = (3.1, 8.25, 21.54, 56.23, 146.79, 383.11, 1000)$

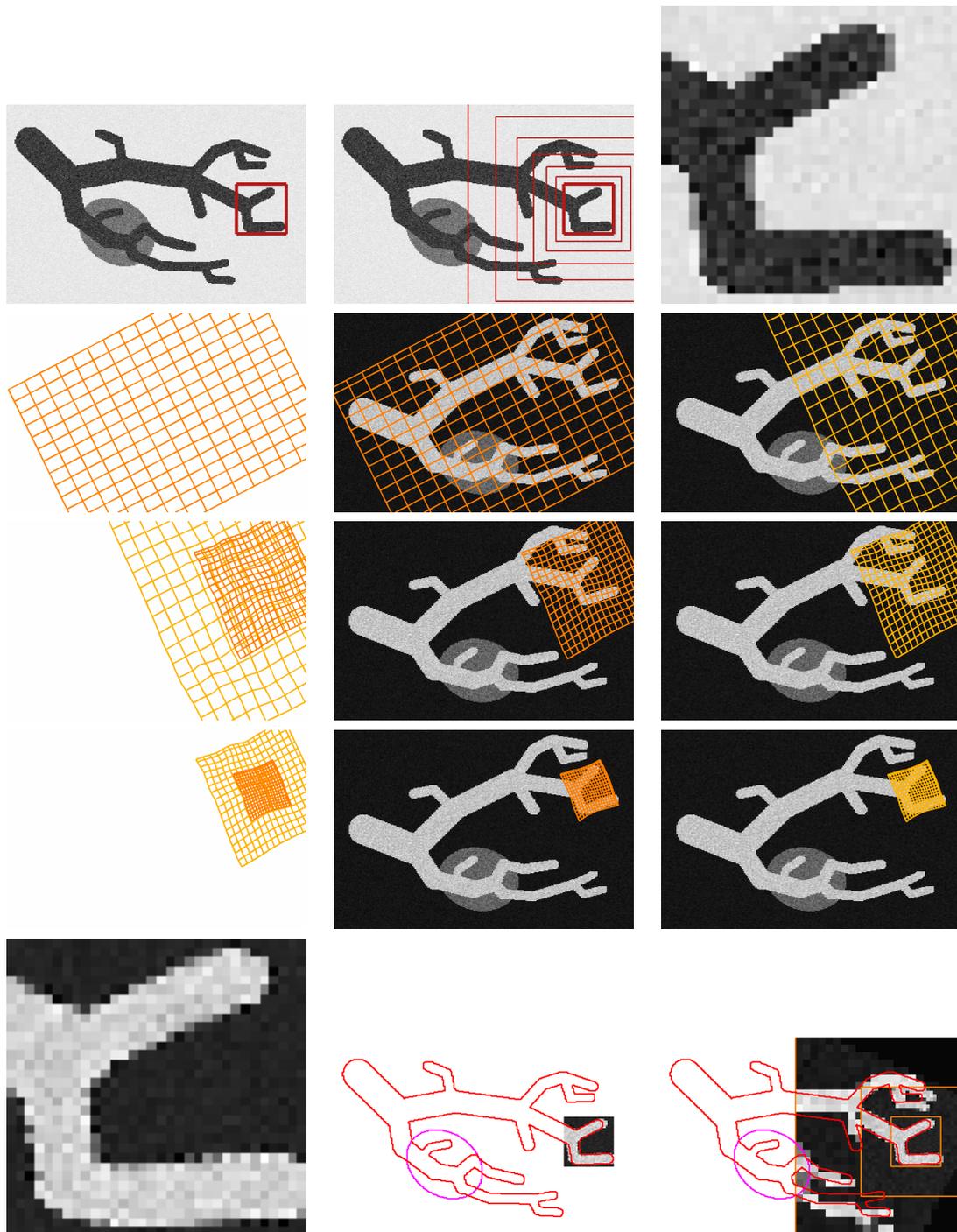


Abbildung 11.5: Zweites Beispiel (Parameter identisch zu erstem Beispiel)

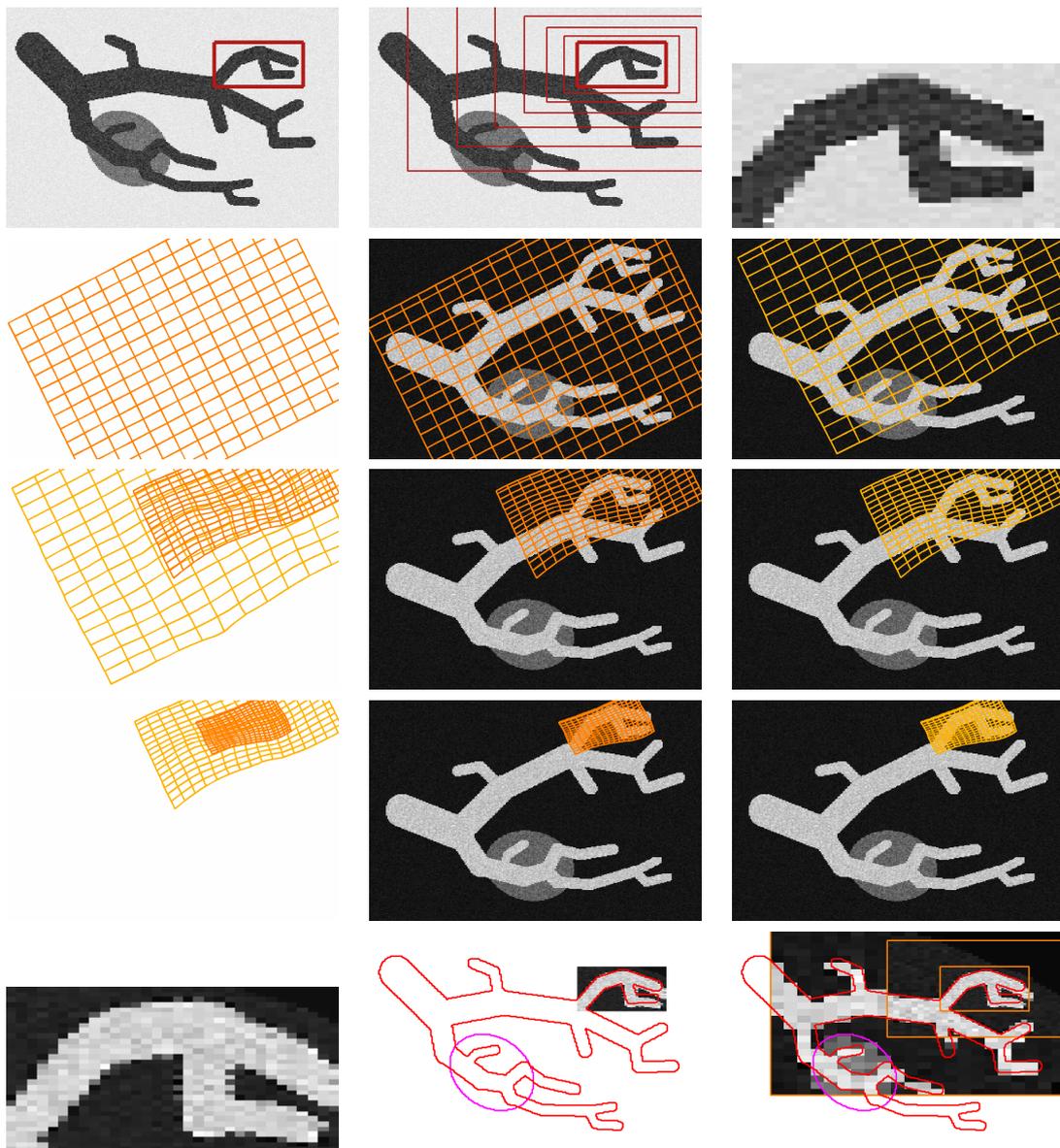


Abbildung 11.6: Drittes Beispiel (Parameter identisch zu erstem Beispiel)

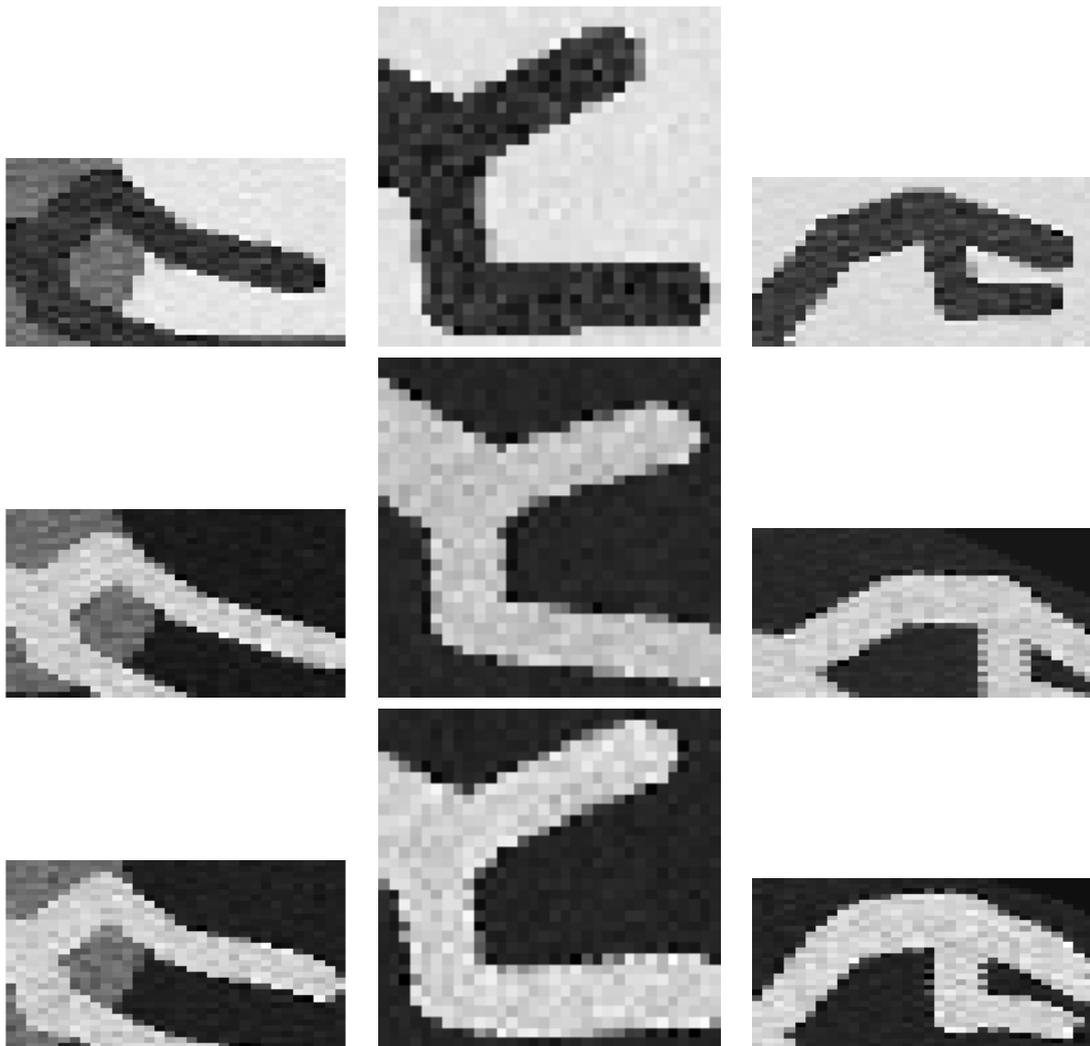


Abbildung 11.7: Vergleich der deformierten Templatebilder für drei Zielgebiete; *oben*: Ausschnitt im Referenzbild; *Mitte*: Ausschnitt im deformierten Template mittels herkömmlicher Verfahren; *unten*: deformiertes Template mit neuer Fokussierungsstrategie

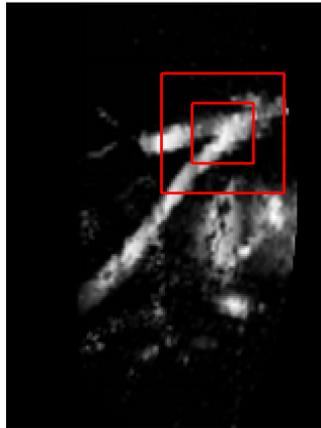


Abbildung 11.8: Fokussierung für reale Daten; Referenz mit zwei quaderförmigen Zielgebieten; nur ein Abschnitt ist visualisiert

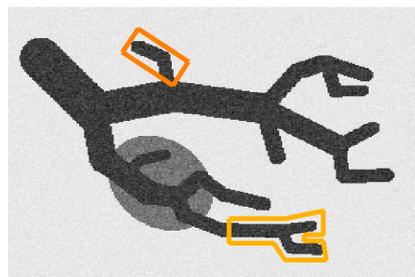


Abbildung 11.9: Beispiel: Möglichkeiten für die Zielgebietswahl; 1. nicht nur ein Zielgebiet; 2. gedrehte Quader (dunkles Orange); 3. unregelmäßige Gebiete (helles Orange)

Kapitel 12

Landmarkenregistrierung

12.1 Motivation

Bisher haben wir Registrierungsalgorithmen beschrieben, die auf den Bildinformationen basieren. In diesem Abschnitt erarbeiten wir eine Erweiterung der bisherigen Ansätze, die auf der zusätzlichen Nutzung von Landmarken basiert. Mit Landmarken bezeichnen wir Paare korrespondierender Punkte, bestehend aus je einem Punkt des Referenz- und einem Punkt des Templatebildes, die die gesuchte Transformation aufeinander abbilden sollen.

In medizinischen Anwendungen werden anatomisch auffällige Strukturen, die in beiden Bildern deutlich sichtbar sind, genutzt, um Positionen für Landmarken festzulegen. Im Fall der Leberregistrierung bieten sich hierfür Gefäßverzweigungspunkte an, da Gefäßbäume in den Planungsdaten und in den Navigationsdaten zu erkennen sind. Trotzdem kann nicht von einer exakten Korrespondenz der Landmarken ausgegangen werden, da beim Platzieren Fehler gemacht werden, so dass sich die Forderung nach exakter Abbildung der Landmarken aufeinander dazu als widersprüchlich erweist. Um dies zu beheben, sind verschiedene Wege vorgeschlagen worden.

Zum einen sind Verfahren entwickelt worden, die Landmarken automatisch zu platzieren [69]. Dies umfasst sowohl Verfahren, die den Anwender bei der Platzierung der Landmarken unterstützen, bsp. durch schnellen auf die nächstliegende Kante, als auch Verfahren, die voll automatisch Punktepaare finden [69]. Auf derartige Techniken gehen wir in dieser Arbeit nicht ein. Zum anderen wurden Methoden entworfen, die die Forderung nach Exaktheit der Abbildung aufweichen, hin zu einer Minimierung eines globalen Landmarken-Fehlers.

Wie die Landmarken in der klinischen Anwendung der Methoden während einer Operation gesetzt werden, diskutieren wir später in Kapitel 14. Wir gehen jetzt davon aus, dass die Landmarken gegeben sind und dass ihre Anzahl klein ist, als Zugeständnis an das intra-operative Setzen.

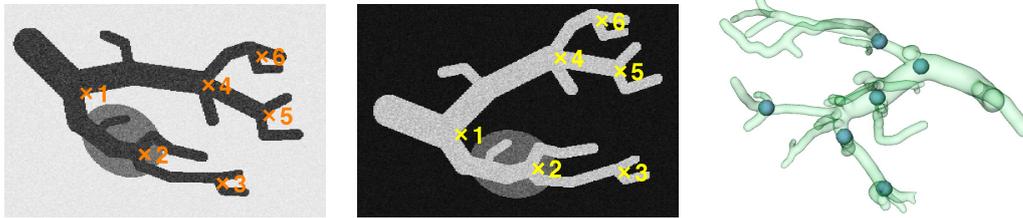


Abbildung 12.1: Position der Landmarken in den Testdaten; *rechts*: Beispiel für Landmarken in klinischen Planungsdaten

In Abbildung 12.1 sind die Positionen von Landmarkenpaaren in den Testdaten gezeigt. Diese wurden an Verzweigungspunkten der Gefäße gesetzt, was typisch für die spätere Anwendung ist. Die Zahlen drücken die jeweiligen Korrespondenzen aus. In der gleichen Abbildung rechts sind durch Kugeln Landmarkenpositionen in einem realen Datensatz gesetzt, um zu zeigen, dass ein derartiges Vorgehen unter OP-Bedingungen möglich ist. Die dargestellten Gefäße sind dabei aus dem Ultraschalldatensatz mittels Schwellwertsegmentierung herausgehoben worden.

Wir beginnen dieses Kapitel mit der Definition der Landmarkenbedingungen und der Angabe der später zu nutzenden Notation. Danach stellen wir Verfahren vor, die allein aus den gegebenen Landmarken eine Deformation bestimmen. Diese Verfahren werden wir zur Vorregistrierung nutzen, um für die nicht-linearen Methoden einen geeigneten Startwert zu bestimmen, der auf dem Wissen eines Experten beruht (, der die Landmarken gesetzt hat). Nachteil dieses Ansatzes, der beide Verfahren hintereinander nutzt, ist, dass nach dem zweiten Schritt die Landmarkenpositionen nicht mehr aufeinander abgebildet werden müssen. Nicht-lineare Verfahren haben kein Wissen um deren Positionen. Wir präsentieren im folgenden Kapitel 13 drei Methoden, die diesen Nachteil aufheben, in dem die Landmarkenbedingungen auf unterschiedliche Art in die Zielfunktion der Registrierung \mathcal{J} einbezogen werden.

12.2 Definition und Notation

Wir befassen uns zunächst mit der Notation der Landmarken. Sei $N_{LM} \in \mathbb{N}$ die Anzahl der gegebenen Landmarkenpaare in Referenz und Template. Der Term

$$\bar{r} \in \mathbb{R}^{N_{LM} \times 3}$$

enthält dann in der j . Zeile die Koordinaten der j . Landmarke im Referenzbild, analog enthält der Term

$$\bar{t} \in \mathbb{R}^{N_{LM} \times 3}$$

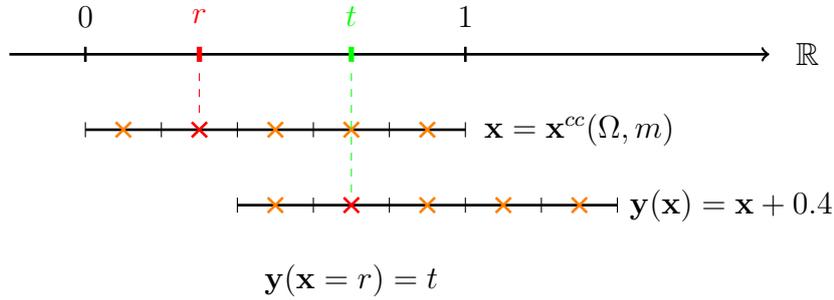


Abbildung 12.2: Illustration der Landmarkenbedingung $y(r_1) = t_1$; Es ist $r_1 = \frac{3}{10}$, $t_1 = \frac{7}{10}$, $\Omega = (0, 1)$, $m = 5$, $\mathbf{x} = \mathbf{x}^{cc}(\Omega, m)$; zur Veranschaulichung sind die Landmarkenpositionen so gewählt, dass sie mit Gitterpunkten zusammenfallen, d.h. $r_1 = \mathbf{x}_2$; für die Transformation \mathbf{y} gilt $\mathbf{y}(\mathbf{x}_2) = t_1$

die Landmarkenkoordinaten des Templatebildes. Die Landmarkenbedingung, korrespondierende Punkte aufeinander abzubilden, ist nun durch

$$y(\bar{r}_{j,:}) = \bar{t}_{j,:}, \quad j = 1, \dots, N_{\text{LM}}$$

gegeben, (und nicht etwa durch den umgedrehten Fall $y(\bar{t}_{j,:}) = \bar{r}_{j,:}$, wie es sich falsch aus der Ähnlichkeitsbedingung $\mathcal{T}(y) \approx \mathcal{R}$ ergeben könnte). Die Argumentation hierfür ist wie folgt: Die gesuchte Transformation y bildet Punkte aus dem Urbildbereich des Bildes \mathcal{R} in den Urbildbereich des Bildes \mathcal{T} ab. Die gegebenen Landmarken sind Elemente der jeweiligen Urbildbereiche, also muss $y(\bar{r}_{j,:}) = \bar{t}_{j,:}$ gelten, vergleiche Kapitel 5.6. Das folgende Beispiel soll dies noch einmal illustrieren.

Sei $\Omega = (0, 1)$ und $r_1 = \frac{3}{10}$ und $t = \frac{7}{10}$. Die gesuchte Transformation ist mit diesen Informationen also eine Translation. Die Transformation soll das Template demnach nach links verschieben, was durch eine entgegengesetzte Translation des Urbildraumes von \mathcal{R} geleistet wird, der also nach rechts verschoben wird. Damit ist $y(x) = x + \frac{2}{5}$ und $y(\frac{3}{10}) = \frac{3}{10} + \frac{2}{5} = \frac{7}{10}$, also $y(r_1) = t_1$. In Abbildung 12.2 ist dies grafisch illustriert.

Wir definieren die Vektoren $r, t \in \mathbb{R}^{3N_{\text{LM}}}$ analog zur Notation für Gitter, in dem wir die Größen \bar{r} bzw. \bar{t} spaltenweise durchlaufen. Zunächst werden alle x_1 -Koordinaten abgespeichert, dann alle x_2 - und x_3 -Koordinaten.

Für die Testdaten aus Kapitel 5.5 sind die Landmarken durch

$$\bar{r} = \begin{pmatrix} 3.8191 & 6.5495 & 10.1786 & 9.5219 & 12.3560 & 12.0104 \\ 5.8122 & 2.9435 & 1.5956 & 6.1924 & 4.7753 & 7.5403 \end{pmatrix}^{\top}$$

und

$$\bar{t} = \begin{pmatrix} 4.8560 & 8.4505 & 12.4597 & 9.4873 & 12.2869 & 11.4228 \\ 3.8767 & 2.2869 & 2.1141 & 7.4712 & 6.8491 & 9.1993 \end{pmatrix}^{\top}$$

gegeben. Sie wurden per Hand und entsprechend der Herangehensweise des Chirurgen auf Ränder von Gefäßverzweigungen gesetzt. Ihre Lage ist in Abbildung 12.1, links und Mitte dargestellt.

Es wird sich zeigen, dass diese sechs Landmarken-Korrespondenzen für die gegebenen zweidimensionalen Testdaten mehr als ausreichend sind, um eine geeignete Transformation zu bestimmen. Um aber zu zeigen, wie die Methoden mit weniger Landmarken-Informationen auskommen, werden wir zum Vergleich die Verfahren auch mit drei Landmarken-Paaren testen. Damit die Ergebnisse vergleichbar bleiben, nehmen wir hierzu immer die Landmarken-Paare 1, 3 und 6.

12.3 Lineare Landmarkenregistrierung

Das erste Modell wird die gesuchte Transformation als rigide oder affin-lineare Transformation darstellen, wie aus Kapitel 7 über parametrische Bildregistrierung bekannt. Die bestimmte Transformation erfüllt die Landmarkenbedingung nur noch in einem globalen Mittel. Diese Form der Registrierung ist in [69] und [54] ausführlich beschrieben.

Beginnen wir mit dem affin-linearen Modell. Gesucht ist also ein Vektor $\gamma \in \mathbb{R}^{12}$, der die affin-lineare Transformation parametrisiert, vergl 7.1. Wir fordern, dass der gemittelte Abstand zwischen den einzelnen Landmarkenpaaren minimal wird:

$$\|Q_{\text{LM}}\gamma - t\|^2 \rightarrow \min \quad \text{mit} \quad Q_{\text{LM}} = \begin{pmatrix} \bar{r} & e_{N_{\text{LM}}} & & & & \\ & & \bar{r} & e_{N_{\text{LM}}} & & \\ & & & & \bar{r} & e_{N_{\text{LM}}} \end{pmatrix}.$$

Den optimalen Vektor γ erhalten wir durch Lösen der Normalgleichungen

$$Q_{\text{LM}}^{\text{T}} Q_{\text{LM}} \gamma = Q_{\text{LM}}^{\text{T}} t.$$

Das System $Q_{\text{LM}}^{\text{T}} Q_{\text{LM}}$ ist 12×12 groß und damit schnell zu lösen. Für den Fall, dass vier Landmarkenpaare gegeben sind, die je Bild nicht in einer Ebene liegen, stimmt die Lösung von (12.3) mit der Lösung von $Q_{\text{LM}}\gamma = r$ überein. Die so bestimmte Deformation erfüllt die Landmarkenbedingung exakt.

Im rigiden Fall können wir analog zum Vorgehen der rigiden Registrierung den Vektor q (vergleiche Kapitel 7.2) nutzen. Analog zum affin-linearen Fall ist dann auch für dieses Deformationsmodell ein lineares Ausgleichsproblem zu lösen. Bei zwei gegebenen Landmarken, die nicht identisch sind, erfüllt die bestimmte Deformation die exakte Bedingung $y(\bar{r}_{j,:}) = \bar{t}_{j,:}$.

In Abbildung 12.3 sind die Ergebnisse der affin-linearen Landmarken-Registrierung dargestellt. In der ersten Zeile für sechs in der zweiten für drei gegebene

Landmarken. Im letzten Fall können die Landmarken genau aufeinander abgebildet werden, da die drei Landmarken sechs Bedingungen an die Transformation stellen. Es ist offensichtlich, dass die Gefäßstrukturen nach der Registrierung nicht korrespondieren, sie gehen aber auch nur indirekt (über die Positionen der Landmarken) in das Verfahren ein. Für den Fall der sechs Landmarken ist das Ergebnis global besser, da mehr Landmarken genutzt werden.

12.4 Landmarkenregistrierung mit Thin-Plate-Splines

Eine Möglichkeit, die exakte Landmarkenregistrierung zu erreichen, besteht in der Verwendung von Polynomen höherer Ordnung. Diese sind jedoch nicht injektiv, so dass die so bestimmte Transformation unbefriedigend ist. Daher befassen wir uns in einem zweiten Modell mit sogenannten Thin-Plate-Splines (TPS). Wir beschränken uns hier auf die Angabe des Algorithmus und verweisen für die Herleitung auf [54, 69, 7]. Bei der Registrierung mittels TPS minimiert die Transformation zusätzlich zum Erfüllen der Landmarkenbedingungen ein Glattheitsmaß. Für den dreidimensionalen Fall ist dies durch

$$\mathcal{S}^{\text{TPS}}(y) = \int_{\Omega} (\partial_{11}y)^2 + (\partial_{22}y)^2 + (\partial_{33}y)^2 + 2((\partial_{12}y)^2 + (\partial_{13}y)^2 + (\partial_{23}y)^2) dx$$

gegeben. Es wird also die Lösung des restringierten Optimierungsproblems

$$\mathcal{S}^{\text{LM}}(y) \rightarrow \min \quad \text{u.d.N.} \quad y(\bar{r}_j) = \bar{t}_j, j = 1, \dots, N_{\text{LM}}$$

gesucht. In [54] ist gezeigt, dass dieses Problem eine analytische Lösung besitzt. Die gesuchte Transformation lässt sich als Summe von sogenannten radialen Basisfunktionen $\rho_j : \mathbb{R}^3 \rightarrow \mathbb{R}$, $j = 1, \dots, N_{\text{LM}}$ und einem polynomiellen Anteil darstellen. Für den dreidimensionalen Fall gilt für ein $x \in \mathbb{R}^3$

$$\rho_j(x) = \|\bar{r}_{j,:} - x\|.$$

Die gesuchte Transformation $y^{\text{TPS}} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ ist durch

$$y_j(x) = c_{1,j}^p + c_{2,j}^p x_1 + c_{3,j}^p x_2 + c_{4,j}^p x_3 + \sum_{k=1}^{N_{\text{LM}}} c_{k,l}^p \rho(\bar{r}_{j,:} - x)$$

gegeben. Die Koeffizienten $c^p \in \mathbb{R}^{4 \times 3}$ und $c^\rho \in \mathbb{R}^{N_{\text{LM}} \times 3}$ bestimmen sich durch

$$Q^{\text{TPS}} \begin{pmatrix} c^\rho \\ c^p \end{pmatrix} = \bar{r}.$$

Hierbei ist

$$Q^{\text{TPS}} = \begin{pmatrix} A & B \\ B^\top & 0 \end{pmatrix} \in \mathbb{R}^{N_{\text{LM}}+4 \times 3}$$

und

$$A = \begin{pmatrix} 0 & \rho(\|\bar{r}_{1,:} - \bar{r}_{2,:}\|) & \cdots & \rho(\|\bar{r}_{1,:} - \bar{r}_{N_{LM},:}\|) \\ \rho(\|\bar{r}_{2,:} - \bar{r}_{1,:}\|) & 0 & \cdots & \rho(\|\bar{r}_{2,:} - \bar{r}_{N_{LM},:}\|) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(\|\bar{r}_{N_{LM},:} - \bar{r}_{1,:}\|) & \rho(\|\bar{r}_{N_{LM},:} - \bar{r}_{2,:}\|) & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{N_{LM} \times N_{LM}}$$

und

$$B = (1 \quad \bar{r}) \in \mathbb{R}^{N_{LM} \times 4}.$$

Für ein gegebenes Gitter $\mathbf{x} = \mathbf{x}^{cc}(\Omega, m)$, $n = m_1 m_2 m_3$ ergibt sich die diskrete Transformation $\mathbf{y}^{\text{TPS}} \in \mathbb{R}^{3n}$ also durch

$$\bar{\mathbf{y}}_{:,j} = \mathbf{y}(\bar{\mathbf{x}}) = c_{1,j}^p e_n + c^p 2, j\bar{\mathbf{x}}_{:,1} + c^p 3, j\bar{\mathbf{x}}_{:,2} + c^p 4, j\bar{\mathbf{x}}_{:,3} + \sum_{k=1}^{N_{LM}} c_{k,j}^p \rho(\bar{r}_{j,:} - \bar{\mathbf{x}}).$$

Für den zweidimensionalen Fall ändert sich die radiale Basisfunktion [54] zu

$$\begin{aligned} \rho_j & : \quad \mathbb{R}^2 \rightarrow \mathbb{R} \\ x & \mapsto \rho_j(x) := \|\bar{r}_{j,:} - x\|^2 \log(\|\bar{r}_{j,:} - x\|). \end{aligned}$$

Die gleichen Datensätze wie für die affin-lineare Landmarkenregistrierung haben wir auch zur Demonstration dieses Verfahrens genutzt. Die Ergebnisse sind in Abbildung 12.3 in den Zeilen drei und vier dargestellt. Zu erkennen ist, dass in beiden Fällen die Landmarken direkt aufeinander abgebildet werden. Für die Registrierung mit sechs gegebenen Landmarken ist zu sehen, dass das finale Resultat so gut wie ausreicht; nur feinere Strukturen erfordern geringe Anpassungen. Das dargestellte Deformationsfeld ist nicht-linear.

Für drei gegebene Landmarken ist das Ergebnis gleich der affin-linearen Registrierung, da diese für drei Landmarken eine exakte Abbildung liefert.

Zur Registrierung von Landmarken sind weitere Verfahren entwickelt, die der gesuchten Deformation physikalische Eigenschaften [15, 41] hinzufügen. Des Weiteren gibt es Ansätze, die Unsicherheit beim Setzen von Landmarken mit in den Registrierungsprozess einfließen zu lassen [70, 89]. Wir verfolgen diesen Weg nicht weiter, formulieren aber im folgenden Kapitel einen Ansatz, der Landmarken mit dem variationellen Ansatz kombiniert, so dass mit Hilfe von Regularisierern physikalische Eigenschaften in die Landmarkenregistrierung einfließen.

12.5 Landmarken-Vorregistrierung

Wir setzen nun die Landmarkenregistrierung als Vorregistrierung für die nicht-linearen Verfahren ein. Dafür beschreiben wir zwei unterschiedliche Wege:

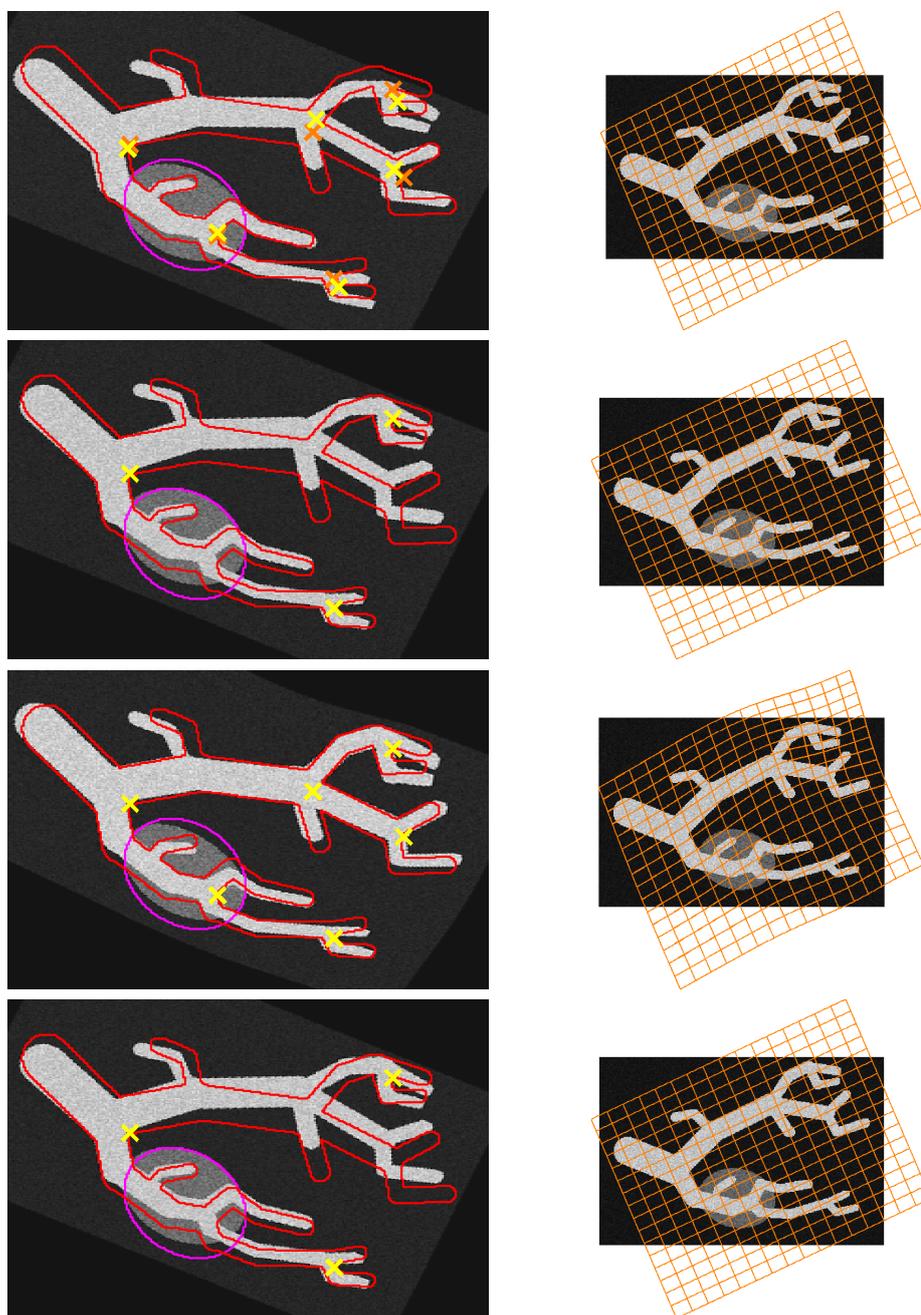


Abbildung 12.3: Ergebnisse der Landmarkenregistrierung; *Zeile eins*: Affin lineare Registrierung mit sechs Landmarken; *Zeile zwei*: Affin lineare Registrierung mit drei Landmarken; *Zeile drei*: TPS-Registrierung mit sechs Landmarken; *Zeile vier*: TPS-Registrierung mit drei Landmarken; *jeweils von links nach rechts*: deformiertes Template mit Lage der transformierten Landmarken; Differenzbild nach der Registrierung; Template mit Transformation

Erstens, das Ergebnis der Landmarkenregistrierung wird als Startwert auf dem größten Level innerhalb eines Multiresolutionsverfahrens genutzt, wobei sie die parametrische Registrierung ersetzen kann. Zweitens, das Ergebnis der Landmarkenregistrierung wird als Startwert für eine nicht-parametrische Registrierung (ohne Multiresolution) genutzt. Dies ist gerechtfertigt, wenn das landmarkenbasierte Ergebnis schon von ausreichender Qualität ist.

Wir illustrieren dies an zwei Beispielen. Als Erstes nutzen wir die Thin-Plate-Spline-Registrierung für sechs gegebene Landmarken. Wie oben gezeigt, liefert diese Vorregistrierung einen guten Startwert für die nicht-lineare Registrierung. Wir verwenden daher kein Multiresolutionsverfahren und zeigen in Abbildung 12.4, links das Ergebnis der Registrierung. Zu erkennen ist, dass die finale Transformation die Landmarken nicht mehr aufeinander abbildet.

Im zweiten Beispiel setzen wir die Landmarkenregistrierung mit drei Landmarken ein, um für den Multiresolutionsansatz einen Startwert zu erhalten. In Abbildung 12.4, rechts ist das Endergebnis dargestellt.

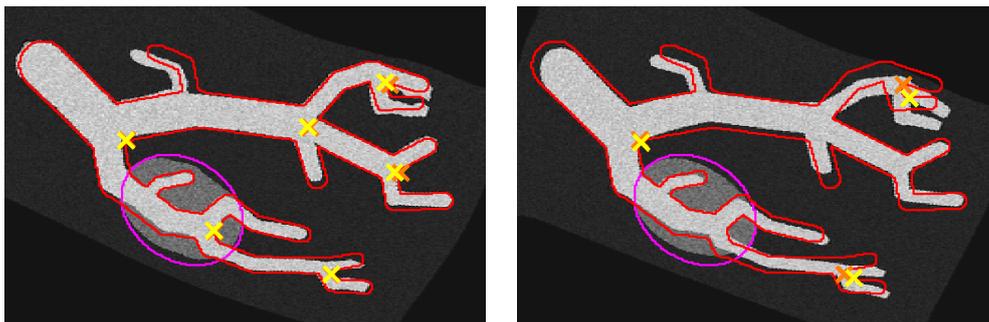


Abbildung 12.4: Landmarkenregistrierung als Startwert für nicht-lineare Registrierung; Ergebnisse *links*: Sechs Landmarken, TPS-Registrierung und nicht-lineare Registrierung ohne Multiresolution; *rechts*: Drei Landmarken, TPS-Registrierung auf größtem Level, Multiresolutionsansatz

Fazit

Wir haben in diesem Kapitel Zusatzinformationen in Form von Paaren korrespondierender Landmarken genutzt, um ein neues, Bilddaten-unabhängiges Verfahren zur Registrierung zu beschreiben. Neben der reinen Landmarkenregistrierung haben wir diese zur Vorregistrierung für die nicht-linearen Methoden verwendet, wobei die Abbildungseigenschaft, Landmarken aufeinander abzubilden, während der Iteration verloren gehen kann. Im folgenden Kapitel beschreiben wir Methoden, die diesen Nachteil aufheben.

Kapitel 13

Kombinierte Verfahren

Im vorherigen Kapitel über Landmarkenregistrierung haben wir gezeigt, dass es eine erste Möglichkeit ist, Vorwissen in Form von Landmarken in die Registrierung zu integrieren, indem zuerst eine Landmarkenregistrierung durchgeführt wird und das so gewonnene Deformationsfeld als Startwert (innerhalb eines Multiresolutionsansatzes) für die nicht-parametrische Registrierung genutzt wird.

Nachteil dieses Ansatzes ist es, dass die Eigenschaft der Transformation, zu Beginn des Verfahrens die Landmarken exakt aufeinander abzubilden, durch den nachgeschalteten nicht parametrischen Ansatz verloren gehen kann, da dieser Ansatz keinerlei Informationen über die Landmarkenbedingungen besitzt.

In diesem Kapitel werden wir drei Verfahren vorstellen, die diesen Nachteil aufheben. Grundlegende Idee ist ein kombinierter Ansatz, der sowohl Bildinformationen als auch Landmarkeninformationen nutzt, um eine Deformation zu bestimmen. Der Name der vorzustellenden Verfahren - CoLD - leitet sich ab von **C**ombined **L**andmarks and **D**istance measure. Der vorgestellte Ansatz fußt auf den Arbeiten [23, 22, 55] von Fischer und Modersitzki.

13.1 CoLD 2003

Die Idee, Landmarkeninformationen in nicht-lineare Registrierungsverfahren zu integrieren und damit ein Verfahren zu beschreiben, das eine exakte Abbildung der Landmarken aufeinander kombiniert mit der gleichzeitigen Minimierung einer aus Distanzmaß und Regularisierer bestehenden Zielfunktion, ist 2003 von Fischer und Modersitzki (vergl. Literatur letzter Absatz) vorgestellt worden. Es nutzt den in Kapitel 2.2.1 beschriebenen Optimize-then-Discretize-Ansatz zur Minimierung eines Funktionals unter Nebenbedingungen. Es ist

$$\mathcal{J}(u) := \mathcal{D}(u) + \alpha \mathcal{S}(u) \rightarrow \min \quad \text{u.d.N.} \quad \mathcal{C}_j(u) = 0, \quad j = 1, \dots, N_{\text{LM}}, \quad (13.1)$$

wobei die gesuchte Transformation y aufgeteilt wird in Identität und Verrückungsfeld, $y(x) = x + u(x)$. Die Maße \mathcal{D} , \mathcal{S} sind gewählt wie in Kapitel 6 und 8 über Distanzmaße und Regularisierer beschrieben. Die Nebenbedingungen $\mathcal{C}_j(u)$ fordern die exakte Abbildung korrespondierender Landmarken.

Der Ansatz basiert auf zwei Schritten:

Zunächst werden Lösungen der N_{LM} Probleme

$$(\mathcal{S}(u_j) \rightarrow \min \quad \text{u.d.N.} \quad \mathcal{C}_j(u_j) = 0), \quad j = 1, \dots, N_{\text{LM}}, \quad (13.2)$$

bestimmt, die glatt im Sinne des Regularisierers sind und je eine Landmarkenbedingung erfüllen. Diese Transformationen werden Fundamentallösungen genannt.

Im zweiten Schritt werden iterativ die zu

$$\mathcal{J}(u) = \mathcal{D}(u) + \alpha\mathcal{S}(u) \rightarrow \min \quad (13.3)$$

gehörigen Euler-Lagrange-Gleichungen durch Diskretisierung und Fix-Punkt- oder Zeitschritt-Verfahren gelöst. Hierbei wird nach jedem Iterationsschritt die aktuelle Iterierte mit einer Linearkombination der Fundamentallösungen aus dem ersten Schritt überlagert, wobei die Linearfaktoren so gewählt werden, dass die so entstehende neue Transformation die Landmarkenbedingungen erfüllt. Diese Gewichte werden durch das Lösen eines linearen Gleichungssystems gefunden, dessen Größe proportional zu den Landmarken ist. Die so in jedem Iterationsschritt bestimmte Verrückungsfunktion erfüllt die notwendigen Bedingungen, die aus der (kontinuierlichen) Lagrange-Funktion

$$\mathcal{L}(u, \lambda) = \mathcal{D}(u) + \alpha\mathcal{S}(u) + \lambda^\top \mathcal{C}(u) \quad (13.4)$$

abgeleitet werden, vergleiche Kapitel 3.

Das Verfahren besitzt mehrere Einschränkungen; so müssen Landmarken auf Gitterknoten liegen, was Multiresolutionsverfahren ausschließt. Des Weiteren erfordert die Bestimmung der Fundamentallösung einen Glattheitsterm mit zweiten Ableitungs-Operatoren, so dass zum Beispiel elastische Regularisierer ausgeschlossen sind.

Wir werden im folgenden eine Weiterentwicklung des Ansatzes vorstellen. Dieser basiert auf der Nutzung des Discretize-then-Optimize-Ansatz. Dazu werden wir zunächst in Abschnitt 13.2 die Landmarkenbedingungen diskretisieren und damit die Grundlage schaffen für die Nutzung von Multiresolutionsverfahren. Das diskrete Problem erlaubt die Verwendung schneller Verfahren der restringierten Optimierung wie SQP, so dass effektive Algorithmen zur Verfügung stehen werden. Durch die Diskretisierung der Landmarkenbedingung und das Vermeiden der Fundamentallösungen können andere Regularisierer eingesetzt werden.

13.2 Diskretisierung der Landmarkenbedingungen

Bevor wir den neuen Ansatz beschreiben, lösen wir zunächst ein grundsätzliches Problem: Die Diskretisierung der Landmarken. Befassen wir uns nun damit, wie wir die Bedingung

$$y(\bar{r}_j) = \bar{t}_j, \quad j = 1, \dots, N_{\text{LM}} \quad (13.5)$$

diskret abbilden können. Die Landmarkenbedingungen stellen - wie bereits in Kapitel 12 beschrieben - eine Interpolationsbedingung an die Transformation y dar. Dies werden wir ausnutzen. Wir betrachten zunächst den eindimensionalen Fall mit einer gegebenen Landmarke, den wir anschließend auf mehrere Landmarken und dann auf höhere Dimensionen erweitern.

Es ist $\Omega = (0, \omega_1)$, $m \in \mathbb{N}$, $h = \omega/m$, $\mathbf{x} = \mathbf{x}^{cc}(\Omega, m)$ und $r \in \Omega$ die Position der Landmarke. Sei nun

$$p = \max\{j \mid \mathbf{x}_j \leq r\}, \quad (13.6)$$

dann gilt für $d \in \mathbb{R}^m$ mit

$$d_j = \begin{cases} \frac{1}{h} - (r - \mathbf{x}_p), & \text{falls } j = p \\ r - \mathbf{x}_p, & \text{falls } j = p + 1 \\ 0, & \text{sonst} \end{cases} \quad j = 1, \dots, m \quad (13.7)$$

$$d^\top \mathbf{x} = r \quad (13.8)$$

und für eine Transformation $\mathbf{y} \in \mathbb{R}^m$

$$d^\top \mathbf{y} \approx y(r). \quad (13.9)$$

Die Approximation folgt der Genauigkeit der linearen Interpolation. Analog zu obiger Definition des Vektors d für Cellcentered-Gitter lässt sich die Definition für Nodal-Gitter schreiben.

Wir haben so die diskrete (eindimensionale) Approximation für die Landmarken-Bedingung hergeleitet. Mit $t \in \Omega$ als Position der zu r korrespondierenden Landmarke ergibt sich die diskrete Form zu $y(r) = t$ durch

$$d^\top \mathbf{y} - t = 0. \quad (13.10)$$

Der Vektor d kann als diskrete Punktauswertungsfunktion (engl. point evaluation function, PEF) interpretiert werden. Im Kontinuierlichen kann die Landmarkenbedingung (13.5) als $\delta_r \star y = y(r)$ geschrieben werden, deren diskrete Form die Gleichung (13.9) ist. Analog zur kontinuierlichen Theorie, $\int \delta_r \, dx = 1$ gilt auch hier $h \sum_j d_j = 1$.

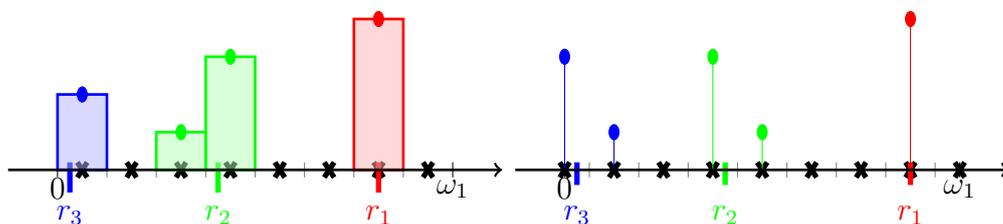


Abbildung 13.1: Approximation der PEF auf eindimensionalem Gitter; die Farbe gibt die Lage der Landmarke bezüglich der Gitterknoten an; *rot*: auf Gitterknoten; *grün*: zwischen Gitterknoten; *blau*: am Rand; *links*: Cell-Centered Gitter; *rechts*: Nodal-Gitter

Die Interpretation des Vektors d als diskrete PEF ist in Abbildung 13.1 illustriert. Für drei verschiedene Positionen der Landmarke ist jeweils der zugehörige Vektor d dargestellt. Die Positionen geben dabei die drei Möglichkeiten wieder, in der die Position r zu den Gitterknoten liegen kann: genau auf einem Gitterknoten, zwischen zwei Gitterknoten und, für den Fall des Cellcentered-Gitters, außerhalb der Gitterknoten. Zu erkennen ist, dass für Cellcentered-Gitter Randartefakte auftreten können (durch das Nichtvorhandensein eines zweiten Gitterknotens), die für Nodal-Gitter nicht auftreten.

Dies könnte sich umgehen lassen, indem Landmarken nicht auf Randzellen eines Cellcentered-Gitters platziert werden, doch wird dies in einem Multiresolutionsansatz scheitern, da bei Gitterauflösungen von $m = (2, 2, 2)$ oder $m = (4, 4, 4)$ fast nur Randzellen gegeben sind.

Die gewählte Diskretisierung der Landmarken ist unabhängig von der gewählten Auflösung m . Diese Eigenschaft erlaubt es, die Landmarken-Bedingung später in einem Multiresolutionsverfahren einsetzen zu können. Für ein Cellcentered-Gitter ist in Abbildung 13.2 die Veränderung der diskreten PEF bei Verfeinerung der Auflösung dargestellt. Die Änderung der Höhe der auftretenden Spitzen ist durch die Forderung $h \sum_j d_j = 1$ bedingt.

Für mehrere Landmarken $r \in \mathbb{R}^{N_{\text{LM}}}$ erhalten wir N_{LM} Bedingungen. Sei dazu $d_{r_j} \in \mathbb{R}^m$ die diskrete PEF zur j . Landmarke. Mit

$$D = \begin{pmatrix} d_{r_1}^\top \\ \vdots \\ d_{r_{N_{\text{LM}}}}^\top \end{pmatrix} \quad (13.11)$$

erhalten wir für die Landmarkenbedingung die diskrete Form

$$D\mathbf{y} - t = 0, \quad (13.12)$$

wobei $t \in \mathbb{R}^{N_{\text{LM}}}$.

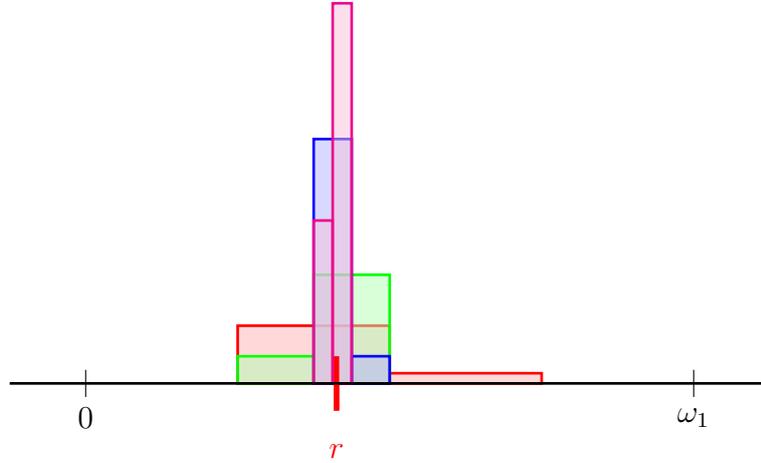


Abbildung 13.2: Verhalten diskreter PEF bei Änderung der Gitter-Auflösung; *rot*: $m = 4$; *grün*: $m = 8$; *blau*: $m = 16$; *magenta*: $m = 32$; die Masse der PEF (schraffierte Fläche) bleibt beim Gitterwechsel konstant ($= 1$)

Die Übertragung auf den dreidimensionalen Fall verläuft analog zu den bisherigen Übertragungen von ein auf drei Dimensionen. Wir geben sie nur für den Cellcentered-Fall an. Seien durch $r, t \in \mathbb{R}^{3N_{LM}}$ die Landmarken gegeben mit der Bedingung $y(\bar{r}_j) = \bar{t}_j$. Seien $m = (m_1, m_2, m_3)$, $n = m_1 m_2 m_3$ und $\mathbf{x} = \mathbf{x}^{cc}(\Omega, m)$ ein Cellcentered-Gitter, auf dem \mathbf{y} gegeben ist. Durch $d_{\bar{r}_j,1} \in \mathbb{R}^{m_j}$, $j = 1, 2, 3$ seien die diskreten eindimensionalen PEF in die jeweiligen Raumrichtungen gegeben. Wir definieren dann die diskrete PEF für die j . Landmarke auf dem dreidimensionalen Gitter durch

$$(d_{\bar{r}_j})_{i_1+m_1(i_2-1)+m_1m_2(i_3-1)} := (d_{\bar{r}_j,1})_{i_1}(d_{\bar{r}_j,2})_{i_2}(d_{\bar{r}_j,3})_{i_3} \in \mathbb{R}^n. \quad (13.13)$$

Diese Gleichung ist als Erweiterung des zweidimensionalen Falles zu verstehen, in dem gilt

$$(d_{\bar{r}_j})_{i_1+m_1(i_2-1)} = (d_{\bar{r}_j,1})_{i_1}(d_{\bar{r}_j,2})_{i_2} = (d_{\bar{r}_j,1})(d_{\bar{r}_j,2})^\top. \quad (13.14)$$

Da für ein Landmarkenpaar drei Interpolationsbedingungen gelten, können wir nun

$$\mathbf{y}(\bar{r}_j) = \begin{pmatrix} y_1(\bar{r}_j) \\ y_2(\bar{r}_j) \\ y_3(\bar{r}_j) \end{pmatrix} = \begin{pmatrix} \bar{t}_{j,1} \\ \bar{t}_{j,2} \\ \bar{t}_{j,3} \end{pmatrix} = \bar{t}_j, \quad j = 1, \dots, N_{LM}, \quad (13.15)$$

durch

$$(I_3 \otimes d_{\bar{r}_j})\mathbf{y} = \begin{pmatrix} d_{\bar{r}_j} & & \\ & d_{\bar{r}_j} & \\ & & d_{\bar{r}_j} \end{pmatrix} \mathbf{y} = (\bar{t}_{j,1}, \bar{t}_{j,2}, \bar{t}_{j,3})^\top = \bar{t}_j^\top \quad (13.16)$$

approximieren.

Bei N_{LM} gegebenen Landmarken erhalten wir unter Berücksichtigen der Sortierung der Vektoren r, t mit

$$\bar{D} = \begin{pmatrix} d_{\bar{r}_1} \\ \vdots \\ d_{\bar{r}_{N_{\text{LM}}}} \end{pmatrix} \quad (13.17)$$

die diskrete Form der Landmarkenbedingung im dreidimensionalen

$$D\mathbf{y} = (I_3 \otimes \bar{D})\mathbf{y} = \begin{pmatrix} \bar{D} & & \\ & \bar{D} & \\ & & \bar{D} \end{pmatrix} \mathbf{y} = t. \quad (13.18)$$

Damit haben wir eine Diskretisierung für den Fall mehrerer gegebener Landmarken für drei Raumdimensionen hergeleitet.

Wir behandeln abschließend die Möglichkeit, eine gegebene Transformation auf die Landmarken zu projizieren. Diese Technik ist nützlich in den kommenden Verfahren und ist nicht aufwendig, da die Bedingungen $D\mathbf{y} - t = 0$ linear in \mathbf{y} sind. Die Projektion \mathbf{y}^P einer beliebigen Transformation \mathbf{y} ist durch

$$\mathbf{y}^P = \mathbf{y} - D^\top (DD^\top)^{-\dagger} (D\mathbf{y} - t) \quad (13.19)$$

gegeben. Sie leitet sich aus der Lösung des quadratischen Minimierungsproblems

$$\frac{1}{2} \|\mathbf{y}^P - \mathbf{y}\|^2 \xrightarrow{\mathbf{y}^P} \min \quad \text{u.d.N.} \quad D\mathbf{y}^P - t = 0 \quad (13.20)$$

her. Wir bestimmen diese Lösung, in dem wir die zugehörige Lagrange-Funktion

$$L(\mathbf{y}^P, \lambda) = \frac{1}{2} \|\mathbf{y}^P - \mathbf{y}\|^2 + \lambda^\top (D\mathbf{y}^P - t), \quad (13.21)$$

mit den Lagrange-Multiplikatoren $\lambda \in \mathbb{R}^{3N_{\text{LM}}}$ bezüglich ihrer Argumente \mathbf{y} und λ ableiten und diese Ableitungen gleich Null setzen. Es ist

$$d_{\mathbf{y}^P} L(\mathbf{y}^P, \lambda) = (\mathbf{y}^P - \mathbf{y}) + D^\top \lambda, \quad (13.22)$$

$$d_\lambda L(\mathbf{y}^P, \lambda) = D\mathbf{y}^P - t. \quad (13.23)$$

Durch Nullsetzen von (13.22) erhalten wir zunächst

$$\mathbf{y}^P = -D^\top \lambda + \mathbf{y}. \quad (13.24)$$

Dieses Resultat in (13.23) eingesetzt ergibt

$$D(-D^\top \lambda + \mathbf{y}) - t = 0 \quad (13.25)$$

$$\Leftrightarrow \lambda = (DD^\top)^{\dagger} (D\mathbf{y} - t). \quad (13.26)$$

Hierbei ist $(DD^\top)^\dagger$ die Pseudo-Inverse von (DD^\top) . Sie wird mit Hilfe der Singulärwertzerlegung (SVD) bestimmt, vergleiche Anhang A.1. Diese Werte für λ liefern wiederum in (13.24) die in (13.19) beschriebene Projektionsvorschrift

$$\mathbf{y}^P = \mathbf{y} - D^\top (DD^\top)^\dagger (D\mathbf{y} - t).$$

Die Matrix $(DD^\top)^{-\dagger}$ kann einmalig für ein gegebenes Gitter bestimmt werden, da sie unabhängig von der Transformation ist. Diese Matrix ist in der Größenordnung der Landmarken und nicht in der des Gitters, so dass ihre Bestimmung billig ist.

Nach diesen technischen Vorarbeiten widmen wir uns nun den eigentlichen Registrierungsalgorithmen.

13.3 Soft-CoLD

Wir beginnen mit der Präsentation der Algorithmen, die die Registrierung basierend auf den reinen Bildinformationen mit den Landmarkenbedingungen kombinieren. Als ersten Ansatz nutzen wir die Erweiterung der Zielfunktion durch einen zusätzlichen Strafterm, der die gesuchte Transformation an die Landmarken treibt. Sei dazu $\mathbf{y} \in \mathbb{R}^n$, $n = 3m_1m_2m_3$ und $D \in \mathbb{R}^{3N_{LM} \times n}$ wie oben beschrieben. Wir binden die Landmarken im sogenannten „soft-constraints“ ein, vergleiche Kapitel 2.1, woraus sich auch der Name des Verfahrens ableitet. Es approximiert

$$\mathcal{P}(\mathbf{y}) = \|D\mathbf{y} - t\|^2$$

den quadrierten euklidischen Landmarkenfehler

$$\sum_{j=1}^{N_{LM}} \|y(\bar{r}_j) - \bar{t}_j\|^2,$$

(der Bezeichner \mathcal{P} leitet sich vom englischen *penalty* ab). Die Approximation liegt in der Fehlerordnung der linearen Interpolation. Das Maß ist für jede Transformation größergleich Null. Für eine Transformation, die die Landmarkenbedingung erfüllt, gilt somit $\mathcal{P}(\mathbf{y}) = 0$. Wir weichen in diesem Ansatz die Landmarkenbedingung auf, in dem wir (nur) fordern, dass die gesuchte Transformation das Maß \mathcal{P} minimiert. Wir erweitern die Zielfunktion \mathcal{J} aus Kapitel 8.4 um diesen Strafterm und erhalten

$$\mathcal{J}(\mathbf{y}) = \mathcal{D}(\mathbf{y}) + \alpha\mathcal{S}(\mathbf{y}) + \beta\mathcal{P}(\mathbf{y}). \quad (13.27)$$

Hierbei gewichtet $\beta \in \mathbb{R}^+$ den Einfluss des Strafterms auf das Gesamtfunktional. Vorteil des Ansatzes ist seine leichte Einbettung in den bestehenden Optimierungsalgorithmus, das Gauß-Newton-Verfahren. Hierfür werden die ersten

und zweiten Ableitungen des Strafterms \mathcal{P} nach der gesuchten Transformation benötigt. Es ist

$$\begin{aligned} \mathcal{P} : \mathbb{R}^n &\rightarrow \mathbb{R}, & d\mathcal{P} : \mathbb{R}^n &\rightarrow \mathbb{R}^n & d_2\mathcal{P} : \mathbb{R}^n &\rightarrow \mathbb{R}^{n \times n} \\ d\mathcal{P}(\mathbf{y}) &= D^\top(D\mathbf{y} - t) & d_2\mathcal{P}(\mathbf{y}) &= D^\top D. \end{aligned}$$

13.3.1 Eigenschaften des Verfahrens

Wir diskutieren Eigenschaften des Ansatzes. Wie für die folgenden Verfahren auch erläutern wir die Einbettung des Ansatzes in ein Multiresolutionsverfahren. Dazu geben wir die Wahl der dafür notwendigen Deformationen an.

- Die von uns genutzte Wahl der Transformationen \mathbf{y}^{Ref} , \mathbf{y}^{Kern} und \mathbf{y}^0 innerhalb eines Multiresolutionsansatzes ist in der folgenden Tabelle gegeben.

jedes Level	$\mathbf{y}^{\text{Ref}} = \mathbf{x}^{cc}$ $\mathbf{y}^{\text{Kern}} = \mathbf{y}^{TPS}$
Min. Level	$\mathbf{y}^0 = \mathbf{y}^{TPS}$
\sim Min. Level	$\mathbf{y}^0 = P(\mathbf{y}^{opt} - \mathbf{x}^{cc}(\Omega, 2m)) + \mathbf{x}^{cc}(\Omega, m)$

Als Kern der Regularisierung setzen wir die Thin-Plate-Spline Lösung ein. Dies hat zur Folge, dass in jedem Level nur die Anteile der Transformation vom Regularisierer bestraft werden, die von den Landmarken wegweisen. Als Startwert wählen wir auf dem größten Level die TPS-Lösung, auf jedem anderen Level (\sim Min. Level) nutzen wir mittels Prolongation die zuvor auf dem gröberen Level bestimmte Lösung.

13.3.2 Ergebnisse

Zur Illustration des Algorithmus nutzen wir wieder die Testdaten, die wir im Abschnitt über die Bildgebung vorgestellt haben. Wie wir in den letzten Kapitel über die Registrierung mittels Grauwerten gesehen haben, bestimmen die vorgestellten nicht-parametrischen Verfahren eine plausible Transformation. Hier werden wir nun demonstrieren, wie sich der Einfluss der Landmarken auf die Registrierung zeigt.

In der ersten Zeile in Abbildung 13.3 ist links das schon bekannte Referenzbild mit den gegebenen sechs Landmarkenpositionen dargestellt. In der gleichen Zeile sind rechts daneben die Ergebnisse der Landmarkenregistrierung mittels Thin-Plate-Splines für sechs und drei Landmarken gezeigt, die uns als Vergleich für die Ergebnisse des Soft-CoLD-Verfahrens dienen. Da die Landmarkenbasierte Registrierung für sechs gegebene Landmarken schon ein sehr gutes Ergebnis liefert, sind kaum Unterschiede in der Lösung des Soft-CoLD-Verfahrens

zu erkennen, das in der zweiten Zeile der Abbildung dargestellt ist. Allein der vom oberen Hauptast abgehende kleine Ast in Form eines um 180° gedrehten L_s wird besser an die Lage im Referenzbild angepasst. Als Parameter wurden $\alpha = 0.1$ und $\beta = 0.01$ gewählt. Im linken Bild der Zeile ist zu erkennen, dass der Landmarkenfehler sehr gering ist, die korrespondierenden Landmarken liegen eng beieinander. Der Wert für β ist durch das Verhältnis von Distanzmaß, Glätter und Strafterm motiviert.

Von besonderem Interesse sind die Ergebnisse der dritten Zeile, wo nur drei Landmarken genutzt werden. Das Ergebnis der Thin-Plate-Spline Registrierung zeigt deutliche Abweichungen von einer optimalen Registrierung. Hier zeigt die Anwendung des Soft-CoLD-Verfahrens eine deutliche Verbesserung des Registrierungsergebnisses. Fast alle Strukturen werden optimal abgebildet. Deutlich ist zu erkennen, dass der obere Ast auf die korrespondierende Struktur transformiert wird. Die Landmarken bleiben aufeinander abgebildet. Auch für diesen Ansatz wurde $\alpha = 0.1$, $\beta = 0.01$ gewählt.

Zum Demonstration des Einflusses des Gewichtungsparameters β auf das Ergebnis wurde das Experiment für $\beta = 0.001$ und $\beta = 0.1$ wiederholt. Die Ergebnisse sind in der untersten Zeile der Abbildung dargestellt. Für den ersten Fall ist deutlich die Abweichung der Landmarke oben rechts zu erkennen. Die Registrierung ist in einem lokalen Minimum gefangen, da (nicht korrespondierende) Kanten von Template und Referenz aufeinander liegen (obere rechte Bildecke).

Die Ergebnisse erlauben die Annahme, dass mit der Landmarkenregistrierung als Vorregistrierung und dem Soft-CoLD-Verfahren eine Methode gefunden wurde, die sich für die Registrierung von Ultraschall- und CT-Bildern gut eignet. Bei geeigneter Wahl von β wird die Landmarkenbedingung annähernd erreicht. Der Landmarkenfehler ist gering. Dies ist gerade dann positiv zu bewerten, wenn die Position der Landmarken in beiden Bildern nicht exakt gegeben ist. Wir kommen auf diesen Punkt später zurück.

Nachteil der Methode ist das Einbringen eines zusätzlichen Parameters β , dessen geeignete Wahl (derzeit) nur heuristisch möglich ist, da der Einfluss des Strafterms gegenüber dem Wert des Bilddaten-abhängigen Distanzmaßes gewichtet werden muss. Ein nicht passender Wert führt, wie demonstriert, zu nicht befriedigenden Registrierungsergebnissen. Die zwei folgenden Verfahren kommen ohne diesen Parameter aus.

13.4 CoLD mit SQP

Nach dem im vorherigen Verfahren die Landmarkenbedingung nur approximativ erfüllt wird, präsentieren wir nun ein Verfahren, das ein Deformationsfeld berechnet, das das vorgegebene Funktional $\mathcal{J} = \mathcal{D} + \alpha\mathcal{S}$ minimiert und das zusätzlich die Landmarkenbedingung erfüllt.

Anders als im CoLD-Verfahren von 2003 nutzen wir nun Methoden der diskreten Optimierung. Durch die Vorarbeiten in den Kapiteln über Optimierung und nicht-parametrische Registrierung (Kapitel 3 und 8) formulieren wir dieses Problem gleich im Diskreten ohne Angabe eines kontinuierlichen Äquivalents.

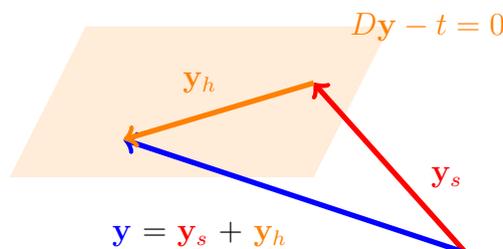
Wir lösen das diskrete Problem

$$\mathcal{J}(\mathbf{y}) = \mathcal{D}(\mathbf{y}) + \alpha\mathcal{S}(\mathbf{y}) \rightarrow \min \quad \text{u.d.N.} \quad D\mathbf{y} - t = 0 \quad (13.28)$$

Wir verwenden innerhalb dieses Ansatz eine Zerlegung der gesuchten Transformation. Es ist

$$\mathbf{y} = \mathbf{y}_s + \mathbf{y}_h,$$

wobei \mathbf{y}_s eine spezielle und \mathbf{y}_h eine homogene Lösung bezüglich der Nebenbedingungen darstellt. Die Begriffe sind analog zu denen der Lösung eines linearen Gleichungssystems aus der linearen Algebra gewählt. Die folgende Skizze illustriert die Begriffe.



Der spezielle Anteil \mathbf{y}_s erfüllt die Landmarkenbedingung

$$D\mathbf{y}_s = t.$$

Dieser Anteil ist nicht eindeutig. Eine Möglichkeit der Wahl einer solchen Funktion ist die Transformation, die durch Lösen des Landmarkenproblems mit Hilfe von Thin-Plate-Splines (vergleiche Kapitel 12.4) entsteht, $\mathbf{y}_s = \mathbf{y}_{TPS}$.

Der homogene Anteil liegt im Kern der Matrix D . Die Vorteile dieser Zerlegung diskutieren wir im Abschnitt über die Eigenschaften des Verfahrens.

Wir nehmen nun die spezielle Lösung als gegeben an und bestimmen die homogene Lösung durch die Minimierung von

$$\mathcal{J}(\mathbf{y}_h) = \mathcal{D}(\mathbf{y}_s + \mathbf{y}_h) + \alpha\mathcal{S}(\mathbf{y}_h) \rightarrow \min \quad \text{u.d.N.} \quad D\mathbf{y}_h = 0. \quad (13.29)$$

Im Kapitel über Optimierung haben wir zur Lösung des bedingten Minimierungsproblems das SQP-Verfahren vorgestellt 3.4.1, welches wir nun zur Lösung von (13.29) nutzen werden.

Zu klären ist, wie welche Größen des SQP-Problems mit den nun gegebenen Funktionen korrespondieren. Diese sind durch

$$\begin{aligned} f(\mathbf{y}_h) &= \mathcal{J}(\mathbf{y}_h) = \mathcal{D}(\mathbf{y}_s + \mathbf{y}_h) + \alpha \mathcal{S}(\mathbf{y}_h) \\ c(\mathbf{y}_h) &= D\mathbf{y}_h \end{aligned}$$

gegeben. Daraus ergibt sich die Lagrange-Funktion L als

$$L(\mathbf{y}, \lambda) = \mathcal{D}(\mathbf{y}_s + \mathbf{y}_h) + \alpha \mathcal{S}(\mathbf{y}_h) + \lambda^\top (D\mathbf{y}_h)$$

mit $\lambda \in \mathbb{R}^{3N_{LM}}$. Während der Iteration werden die Größen

$$\nabla f(\mathbf{y}_h)^\top = d\mathcal{D}(\mathbf{y}_s + \mathbf{y}_h) + \alpha d\mathcal{S}(\mathbf{y}_h)$$

(vergleiche Kapitel 8.4) sowie

$$dc(\mathbf{y}_h) = D$$

benötigt. Das resultierende KKT-System ist durch

$$\begin{pmatrix} d_2L & D^\top \\ D & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{y} \\ \delta\lambda \end{pmatrix} = \begin{pmatrix} -dJ + D^\top\lambda \\ -D\mathbf{y}_h \end{pmatrix}$$

gegeben. Den oberen linken Block der KKT-Matrix, die Hesse-Matrix bezüglich \mathbf{y} der Lagrange-Funktion nähern wir durch die schon im Gauß-Newton-Verfahren verwendeten Approximation

$$d_2L(\mathbf{y}_h) \approx dr^\top d_2\varphi dr + \alpha B^\top B$$

an. Da die Nebenbedingungen in \mathbf{y}_h linear sind, leisten sie keinen Beitrag zur Hesse-Matrix.

Insgesamt erhalten wir ein Verfahren, in dem iterativ die Lösung von (13.28) gefunden wird, indem in jedem Schritt durch das System (13.4) eine Suchrichtung und mit Hilfe der in 3.4.2 beschriebenen Merrit-Funktion eine Schrittweite bestimmt wird.

13.4.1 Eigenschaften des Verfahrens

Wir werden wieder Eigenschaften des Verfahrens diskutieren.

- Um die Lagrange-Multiplikatoren λ initial nicht wählen zu müssen, formen wir das KKT-System um. Zur Wiederholung, es ist $\delta\mathbf{y} = \mathbf{y}^{k+1} - \mathbf{y}^k$, $\delta\lambda = \lambda^{k+1} - \lambda^k$. Das System ist damit äquivalent zu

$$\begin{pmatrix} d_2L & D^\top \\ D & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{y} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} -dJ \\ -D\mathbf{y}_h \end{pmatrix}.$$

Dieses System ist unabhängig von λ^k , womit die Vorgabe von λ^0 entfällt. Mittels dieses Systems bestimmen wir $\delta\mathbf{y}$ und im Anschluss über

$$\|dJ + \lambda^{k+1}D^\top\|^2 \rightarrow \min$$

die Lagrange-Multiplikatoren mittels zugehöriger Normalengleichungen

$$(DD^\top)\lambda^{k+1} = -D^\top dJ,$$

zu deren Lösung die Pseudo-Inverse $(DD^\top)^\dagger$ benötigt wird (vergleiche Abschnitt 13.2, unten).

- Eine mögliche Wahl der Transformationen \mathbf{y}^{Ref} , \mathbf{y}^{Kern} und \mathbf{y}^0 ist in der folgenden Tabelle gegeben.

	Ansatz $\mathbf{y} = \mathbf{y}_s + \mathbf{y}_h$
jedes Level	$\mathbf{y}_s = \mathbf{y}^{TPS}$ $\mathbf{y}^{\text{Ref}} = \mathbf{x}^{cc}$
Min. Level	$\mathbf{y}_h^0 = 0$
\sim Min. Level	$\mathbf{y}_h^0 = P(\mathbf{y}^{opt})$

Auf jedem Level im Multiresolutionsansatz wählen wir als Referenzgröße für die Abbruchkriterien die Identität, $\mathbf{y}^{\text{Ref}} = \mathbf{x}^{cc}$. Den speziellen Anteil der Lösung bestimmen wir mittels Landmarkenregistrierung durch Thin-Plate-Splines. Auf dem größten Level (Min Level) wählen wir als Startwert $\mathbf{y}^0 = \mathbf{y}^{TPS}$, woraus $\mathbf{y}_h^0 = 0$ folgt. Auf jedem Level erhalten wir durch den Algorithmus eine Lösung $\mathbf{y}^{opt} = \mathbf{y}_s + \mathbf{y}_h^{opt}$, deren homogenen Anteil wir als Startwert \mathbf{y}_h^0 auf dem nächst feineren Gitter nutzen. Hierzu nutzen wir die in Abschnitt 9.4 vorgestellte Prolongation. Der Kern der Regularisierung ist nicht speziell aufgeführt, da wir nur über die homogenen Anteile \mathbf{y}_h regularisieren, die ja schon um die spezielle Lösung bereinigt sind. Wir wählen als Kern also die Null.

- Das KKT-System innerhalb des SQP-Verfahrens ist indefinit bedingt durch den Null-Block unten rechts. Daher kann es nicht effizient mit Methoden wie dem MATLAB-internen Gleichungssystemlöser gelöst werden. Auch die in Kapitel 8.5 vorgestellten Multigrid-Verfahren versagen. Das System kann aber mit Methoden, die speziell zur Lösung indefiniter Systeme entwickelt wurden, gelöst werden. Die in der Literatur bekannteste Methode hierfür ist das MinRes-Verfahren [58].
- Dieses Verfahren bildet die Landmarken exakt aufeinander ab, was bei fehlerhaft platzierten Landmarken zu unbefriedigenden Deformationen führen kann.

Eine Abhilfe läge in der Aufgabe der Gleichheitsforderung in den Nebenbedingungen. Wir formulieren stattdessen eine Nebenbedingung der Form

$$c(\mathbf{y}) = \|D\mathbf{y} - \bar{t}\|^2 \leq \text{tol}_{LM}, \quad (13.30)$$

die garantiert, dass der Landmarkenfehler kleinergleich einer gegebenen Toleranz tol_{LM} ist. Das SQP-Verfahren kann, wie in Kapitel 3 angedeutet, restringierte Optimierungsprobleme mit Ungleichsbedingungen lösen.

Der Unterschied zum zuvor beschriebenen Verfahren liegt darin, dass nun algorithmisch zugesichert wird, dass der Landmarkenfehler kleiner einer Vorgabe ist, hingegen beim Soft-CoLD-Ansatz „nur“ die gewichtete Summe aus Distanzmaß, Regularisierer und Strafterm minimiert wird, vergleiche Kapitel 15.

13.4.2 Ergebnisse

Mit den folgenden Ergebnissen des Verfahrens für die bekannten Testdaten wollen wir zwei grundlegende Eigenschaften des Algorithmus illustrieren: die Einhaltung der Landmarkenbedingung und die Anwendbarkeit auf die gegebenen Daten.

Wiederum nutzen wir als Eingabe einmal sechs und einmal drei gegebene Landmarken. In Abbildung 13.4 ist links das Ergebnis des CoLD-Verfahrens gezeigt. Die Landmarken sind exakt aufeinander abgebildet. Außerdem liegt in beiden Fällen ein subjektiv gutes Registrierungsergebnis vor.

Obwohl die Thin-Plate-Spline-basierte Registrierung (rechts in der Abbildung) für sechs Landmarken schon eine gute Transformation liefert, verbessert das CoLD-Verfahren diese noch. Dies ist besonders an den feinen Ast-Strukturen und an der Passform des Tumors zu erkennen. Für den Fall dreier gegebener Landmarken ist die Verbesserung der reinen Landmarkenregistrierung noch leichter zu beobachten. Auch die große Struktur des oberen Astes wird auf die korrespondierende Position abgebildet. Die bestimmten Transformationsfelder sind glatt im Sinne des Regularisierers. Beide Ergebnisse wurden mit $\alpha = 0.1$ in einem Multiresolutionsansatz bestimmt.

13.4.3 Forderungen an die Landmarken-Bedingung

Die Landmarkenbedingungen sind formuliert als Nebenbedingungen $c(\mathbf{y}) = D\mathbf{y} - t$. In Abschnitt 3.4 des Kapitels über Optimierungsmethoden haben wir festgestellt, dass es eine Voraussetzung für die Anwendung des SQP-Verfahrens ist, dass die Jacobi-Matrix dc vollen Rank hat. In unserem Fall gilt $dc = D$.

Wir untersuchen, unter welchen Bedingungen die Annahme erfüllt ist, dass D vollen Rank hat.

Die Matrix D hat Diagonal-Block-Struktur, $D = I_3 \otimes \bar{D}$, das heißt, wir müssen lediglich die Matrix \bar{D} auf ihren Rank untersuchen. Es gilt $\text{rank} D = 3 \text{rank} \bar{D}$.

Tritt eine Landmarke in \bar{r} doppelt auf, das heißt für $i \neq j$ gilt $\bar{r}_i = \bar{r}_j$, so hat \bar{D} zwei linear abhängige Zeilen und damit nicht vollen Rank.

Mit n sei die Anzahl der Gitterpunkte des Gitters y bezeichnet, das heißt für Cellcentered-Gitter $n = 3\bar{m}$, für Nodal-Gitter $n = 3(m_1 + 1)(m_2 + 1)(m_3 + 1)$. Falls gilt

$$3N_{\text{LM}} > n,$$

also mehr Landmarkenbedingungen als Freiheitsgrade des Gitters vorliegen, kann \bar{D} nicht vollen Rank haben.

Von praktischem Interesse ist diese Eigenschaft bei der Nutzung von Multiresolutionsansätzen. In der Praxis liegen Landmarken in der Größenordnung von 10 vor. Somit werden 30 Bedingungen an das Deformationsfeld gestellt. In Multiresolutionsansätzen gilt es nun anzupassen, dass die verwendeten Gitter so viele Freiheitsgrade besitzen. Bei einer Anzahl von Gitterzellen von $m = (4, 4, 4)$ hat ein Gitter 64 Freiheitsgrade, es können also 21 Landmarkenbedingungen gestellt werden.

Ein Verfahren, das diese Bedingung an die Auflösung des Gitters nicht hat, stellen wir nun vor.

13.5 CoLD mit SVD

Anders als der vorherige Ansatz, der auf der Formulierung eines restringierten Optimierungsproblems basierte, kommt dieser Ansatz mit Methoden der nicht bedingten Optimierung aus, löst aber das bedingte Optimierungsproblem.

Wir benötigen damit nicht die zuletzt diskutierten Bedingungen zwischen Landmarken und Gitterpunkten. Das Verfahren nutzt Methoden der nicht restringierten Optimierung; ein indefinites System, wie es das KKT-System im letzten Verfahren war, wird nicht auftreten. Es wird sich sogar zeigen, dass die neue Methode mit wenigen Veränderungen des Standardansatzes der Registrierung auskommt.

Dies ist deswegen möglich, weil die Landmarkenbedingungen linear im gesuchten Deformationsfeld \mathbf{y} sind, $D\mathbf{y} - t = 0$. Wir wählen als Ansatz für die gesuchte Transformation wieder die Zerlegung $\mathbf{y} = \mathbf{y}_s + \mathbf{y}_h$.

Der homogene Anteil lässt sich als Linearkombination der Elemente einer Basis des Nullraumes von D schreiben. Sei Z eine Matrix, deren Spalten diese Basiselemente sind, so haben wir

$$\mathbf{y}_h = Z\mathbf{w} \tag{13.31}$$

mit $Z \in \mathbb{R}^{3n \times r}$ und $w \in \mathbb{R}^r$. Hierbei ist r die Dimension des Nullraumes von D und w der Koeffizientenvektor.

Mit der oben beschriebenen Wahl führt dieser Ansatz zu einer Deformation, die nur noch durch den Koeffizientenvektor w parametrisiert wird,

$$\mathbf{y} = \mathbf{y}_s + \mathbf{y}_h = \mathbf{y}_s + Zw.$$

Jedes so bestimmte \mathbf{y} erfüllt die Landmarkenbedingung. Wir sind nun in der Lage, den Parametervektor w so zu bestimmen, dass er optimal wird bezüglich der weiteren Forderung der Registrierung, nämlich Minimierung eines Distanzmaßes und eines Regularisierers. Dies erzielen wir durch den schon bekannten Algorithmus für nicht bedingte Registrierung, den wir nun bezüglich des Vektors w verwenden werden. Es ist

$$\mathcal{J}(w) := \mathcal{J}(\mathbf{y}(w)) = \mathcal{D}(\mathbf{y}_s + Zw) + \alpha \mathcal{S}(w).$$

Für den Regularisierer nutzen wir nun die Freiheit in der Wahl des Kerns, indem wir $\mathbf{y}^{\text{kern}} = \mathbf{y}_s$ setzen. Das bedeutet, wir werden die Anteile aus der Landmarkenvorregistrierung nicht bestrafen. Dies führt zu

$$\mathcal{S}(w) = \|B(\mathbf{y}_s + Zw) - \mathbf{y}^{\text{kern}}\|^2 = \|BZw\|^2.$$

Der nun erzielte Vorteil ist, dass wir aus Sicht der Modellierung mit

$$\mathcal{J}(w) = \mathcal{D}(w) + \alpha \|BZw\|^2 \xrightarrow{w} \min$$

ein nicht bedingtes Optimierungsproblem formuliert haben, dessen Lösung die Landmarkenbedingungen per Konstruktion erfüllt. Hinzu kommt, dass in jedem Schritt eines iterativen Optimierungsverfahrens zur Lösung dieses Problems jede Iterierte \mathbf{y}_k auch die Landmarkenbedingungen erfüllt.

Zentraler Bestandteil dieses Ansatzes ist die Bestimmung einer Basis des Nullraumes von D . Dies behandeln wir nun. Nach Abschnitt 13.2 ist die Matrix D durch $D = I_3 \otimes \bar{D}$ gegeben. Dies erlaubt, die Landmarkenbedingungen $D\mathbf{y} - t = 0$ umzuschreiben in drei Gleichungssysteme

$$\bar{D}\bar{\mathbf{y}} = \bar{t},$$

wobei $\bar{\mathbf{y}} \in \mathbb{R}^{n \times 3}$ und $\bar{t} \in \mathbb{R}^{N_{\text{LM}} \times 3}$ die in Kapitel 4 und 12.2 umsortierten Größen sind. Wir haben also für jede Komponente von \mathbf{y} ein Gleichungssystem mit gleicher Matrix \bar{D} zu lösen. Wir betrachten also im Folgenden nur diese Matrix \bar{D} . Wir nutzen nun einen Satz aus der linearen Algebra [28], der einen Zusammenhang zwischen dem Nullraum einer gegebenen Matrix und ihrer Singulärwertzerlegung herstellt, der wie folgt lautet: Sei mit $A = U\Sigma V^\top$

mit

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \sigma_r & \\ & & & & \mathbf{0} \\ & \mathbf{0} & & & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

eine Singulärwertzerlegung von A gegeben, dann hat der Nullraum von A die Dimension r und eine Basis ist durch die letzten $n - r$ Spalten der Matrix V gegeben,

$$Z = V(:, r + 1 : n).$$

Dies wenden wir nun auf die Matrix \bar{D} an. Sei mit $\bar{D} = U\Sigma V^\top$ ihre Singulärwertzerlegung gegeben. Liegen keine Abhängigkeiten der Form vor, wie sie in Abschnitt 13.4.3 beschrieben wurden, gilt $r = \text{rank} \bar{D} = n$. Damit hat Σ die Form

$$\Sigma = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{pmatrix} \in \mathbb{R}^{N_{LM} \times n}.$$

Eine Basis des Nullraumes von \bar{D} ist dann durch die letzten $n - r$ Spalten von V gegeben,

$$Z = V(:, r + 1 : n).$$

13.5.1 Eigenschaften des Verfahrens

- Auch hier diskutieren wir zunächst die Wahl der Transformationen im Multiresolutionsverfahren.

Ansatz $\mathbf{y} = \mathbf{y}_s + \mathbf{y}_h$	
jedes Level	$\mathbf{y}^{\text{Ref}} = \mathbf{x}^{cc}$ $\mathbf{y}^{\text{Kern}} = \mathbf{y}^{TPS}$ $w^0 = 0$ Bestimme \bar{D} Bestimme Basis Z
Min. Level	$\mathbf{y}_s = \mathbf{y}^{TPS}$
\sim Min Level	$\mathbf{y}'_s = P(\mathbf{y}^{opt} - \mathbf{x}^{cc}(\Omega, m/2)) + \mathbf{x}^{cc}(\Omega, m)$ $\mathbf{y}_s = \mathbf{y}'_s - D^\top (DD^\top)^\dagger (D\mathbf{y}'_s - t)$

Diese ist fast gleich zu der im letzten Verfahren. Einzige Änderung ist die Wahl der speziellen und der homogenen Lösung auf den unterschiedlichen Levels, da die homogene Lösung in Form ihrer Basiskoeffizienten w

gegeben ist. Da w keine Gitterfunktion ist, können wir bisher vorgestellte Prolongationsoperatoren nicht nutzen.

Eine Möglichkeit ist es nun, als spezielle Lösung wiederum die TPS-Lösung auf jedem Level zu wählen. Die homogene Lösung wird zunächst auf jedem Level bestimmt, $\mathbf{y}_h^{opt} = Dw^{opt}$, dann prolongiert, auf die Landmarken-Bedingungen projiziert und abschließend werden die neuen Baskoeffizienten w^0 durch Lösen des linearen Gleichungssystems $Dw^0 = y'_h$ bestimmt.

Wir wählen einen anderen Weg, in dem wir auf jedem Level den Startwert für die homogene Lösung konstant Null wählen und die auf den größeren Leveln bestimmten Anteile der homogenen Lösung mit in die aktuelle spezielle Lösung einbringen. Dazu sei $y^{opt} = \mathbf{y}_s + Dw^{opt}$ auf dem größeren Level. Diese wird mittels der Prolongations-Operatoren auf das feinere Level transportiert und dann auf die Landmarken-Bedingungen projiziert. Damit erfüllt sie die Bedingungen und wird als spezielle Lösung für das aktuelle Gitter genutzt.

- Für die Auswertung des Distanzmaßes und die Bestimmung der damit zusammenhängenden Residuumsfunktionen r, dr ist das vollständige Deformationsfeld $\mathbf{y} = \mathbf{y}_s + Zw$ zu bestimmen, damit ändert sich das Distanzmaß zu $\mathcal{D}(w) = \mathcal{D}(\mathbf{y}_s + Zw)$ und es gilt für die im Gauß-Newton-Verfahren verwendeten Ableitungen

$$d_w \mathcal{D}(w) = Z^\top d\mathcal{D} \quad \text{und} \quad d_2 w \mathcal{D}(w) = Z^\top d_2 \mathcal{D} Z,$$

wobei $d\mathcal{D}$ und $d_2 \mathcal{D}$ die in Kapitel 6 beschriebenen Terme sind. Der Regularisierer ändert sich analog zum Distanzmaß. Es ist

$$\mathcal{S}(w) = \|BZw\|^2 \quad \text{und} \quad d_w \mathcal{S}(w) = Z^\top B^\top BZw, \quad d_w^2 \mathcal{S}(w) = Z^\top B^\top BZ.$$

Damit ändert sich das Gleichungssystem zur Bestimmung der Suchrichtung zu

$$(Z^\top d_2 \mathcal{D} Z + \alpha Z^\top B^\top BZ) \delta w = - (Z^\top d\mathcal{D} + \alpha Z^\top B^\top BZw).$$

- Das in Abschnitt 8.5 beschriebene Multigrid-Verfahren löst dieses System aufgrund seiner Einfachheit nicht. Daher verwenden wir in der aktuellen MATLAB-Implementierung ein CG-Verfahren (`pcg.m`).
- Die Verfahren kann speicherplatz- und zeiteffizient implementiert werden, wenn die besondere Struktur der Matrix \bar{D} ausgenutzt wird.

Die Matrix \bar{D} beinhaltet (für den dreidimensionalen Fall) in jeder Zeile maximal acht von Null verschiedene Elemente, die Interpolationsgewichte für die trilineare Interpolation. Im Verhältnis zur Anzahl der Spalten der

Matrix ist dies gering. Durch ein Umsortieren der Spalten lässt sich nun effizienter eine Basis des Nullraumes von \bar{D} bestimmen.

Dazu sei $P \in \mathbb{R}^{n \times n}$ eine Matrix, die aus der Einheitsmatrix im \mathbb{R}^n durch Umsortieren der Zeilen entsteht, das heißt jede Zeile von P und jede Spalte von P enthält genau eine Eins. Wir nennen P *Permutationsmatrix*, da eine Multiplikation mit P auf einen Vektor $x \in \mathbb{R}^n$ eine Umsortierung der Elemente von x durchführt, wobei für $a_{ij} = 1$ das j .te Element von x an die i .te Stelle gesetzt wird. Da P orthogonal ist, ist mit P^\top ihre Inverse gegeben.

Wir erläutern dies kurz an einem Beispiel: Sei $x = (1, 2, 3, 4, 5)^\top \in \mathbb{R}^5$ und

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

Dann ist $Px = (4, 1, 3, 5, 2)^\top$.

Wir wählen nun eine Permutationsmatrix P so, dass die Matrix $\bar{D}P^\top$ die Gestalt

$$\bar{D}P^\top = (\tilde{D} \quad \mathbf{0} \quad)$$

erhält. Sei $\tilde{D} \in \mathbb{R}^{N_{LM} \times \tilde{n}}$. Die Größe der Matrix \tilde{D} ist proportional zu der Anzahl der Landmarken und damit klein im Verhältnis zur Matrix \bar{D} . Daher ist die Bestimmung ihrer Singulärwertzerlegung billig.

Bestimmen wir eine Basis \tilde{Z} des Nullraumes von \tilde{D} mittels Singulärwertzerlegung, so erhalten wir durch

$$Z = \begin{pmatrix} \tilde{Z} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \quad (13.32)$$

eine Basis für den Nullraum von $\bar{D}P^\top$. Ist der Rank von \tilde{D} durch \tilde{r} gegeben, so ist $\tilde{Z} \in \mathbb{R}^{\tilde{n} \times \tilde{n} - \tilde{r}}$ und $Z \in \mathbb{R}^{n \times n - \tilde{r}}$. Die Einheitsmatrix in (13.32) hat also $n - \tilde{n}$ Spalten,

$$\bar{D}P^\top P\mathbf{y} = \bar{D}P^\top PZw = 0.$$

13.5.2 Ergebnisse

Die prinzipielle Eignung dieser Methode zeigen wir ebenfalls an den zweidimensionalen Testdaten. Wir beschränken uns hier auf den Fall dreier gegebener Landmarken, da für diesen Fall durch die „schlechtere“ Thin-Plate-Spline-Registrierung mehr Anpassung zu leisten ist. Das Ergebnis der Registrierung

ist in Abbildung 13.5 dargestellt. Es ist deutlich zu erkennen, dass die Umsetzung die theoretischen Eigenschaften beibehält. Die Landmarken werden aufeinander abgebildet, die Transformation ist glatt im Sinne des gewählten Curvature-Regularisierers. Der Landmarkenfehler beträgt $1.12e - 15$.

In Abbildung 13.6 sind die Ergebnisse der CoLD-Verfahren mittels SQP und SVD gegeneinander gestellt. Hierbei ist als Gitter der jeweils homogene Anteil dargestellt, während wir die Thin-Plate-Spline-Lösung als speziellen Anteil nehmen, $y = y_{sqp,svd}^{opt} - y_{TPS}$. Die Ergebnisse unterscheiden sich. Dies rührt daher, dass die speziellen Anteile in beiden Verfahren unterschiedlich gebildet werden und somit der jeweils zu bestimmende homogene Anteil anders zu interpretieren ist.

Fazit

Durch die CoLD-Verfahren haben wir eine Möglichkeit aufgezeigt, Landmarkeninformationen in den nicht-linearen Registrierungsprozess einzubetten. Durch die Verwendung des Discretize-then-Optimize-Ansatzes und der Methoden der restringierten Optimierung haben wir eine klare Numerik dafür erarbeitet.

Wir haben drei unterschiedliche Verfahren beschrieben. Die Soft-CoLD-Methode minimiert dabei den Landmarkenfehler, eine exakte Abbildung der Landmarken kann nicht gewährleistet werden. Dies ist vorteilhaft, wenn die Positionen der Landmarken selbst mit Fehlern versehen sind, so dass eine exakte Abbildung physikalisch nicht gewollt ist. Andererseits muss der Gewichtungsparameter β datenabhängig vom Benutzer bestimmt werden. Die CoLD-Verfahren leisten die Zusicherung der Landmarkenbedingung ohne einen zusätzlichen Parameter. Das CoLD mit SVD - Verfahren kommt dabei ohne die Lösung eines indefiniten KKT-Systems aus. Daher werden wir es als Testverfahren zusammen mit dem Soft-CoLD-Verfahren im nächsten Kapitel nutzen.

Damit beenden wir den zweiten Teil über spezielle Registrierungsverfahren, in dem wir Multiskalenverfahren zur Aufarbeitung der Röhrenstruktur der Bildaten erarbeitet haben, gefolgt von einer Fokussierungsstrategie, die es erlaubt, in kurzer Zeit die Registrierung für einen kleinen Bildausschnitt zu bestimmen, und die CoLD-Verfahren, die Zusatzwissen mit in die Registrierung einbetten. Im nächsten Kapitel werden wir zeigen, dass das Zusammenspiel dieser Ansätze es ermöglicht, CT- und Ultraschalldaten mit nicht-parametrischen Verfahren zu registrieren.

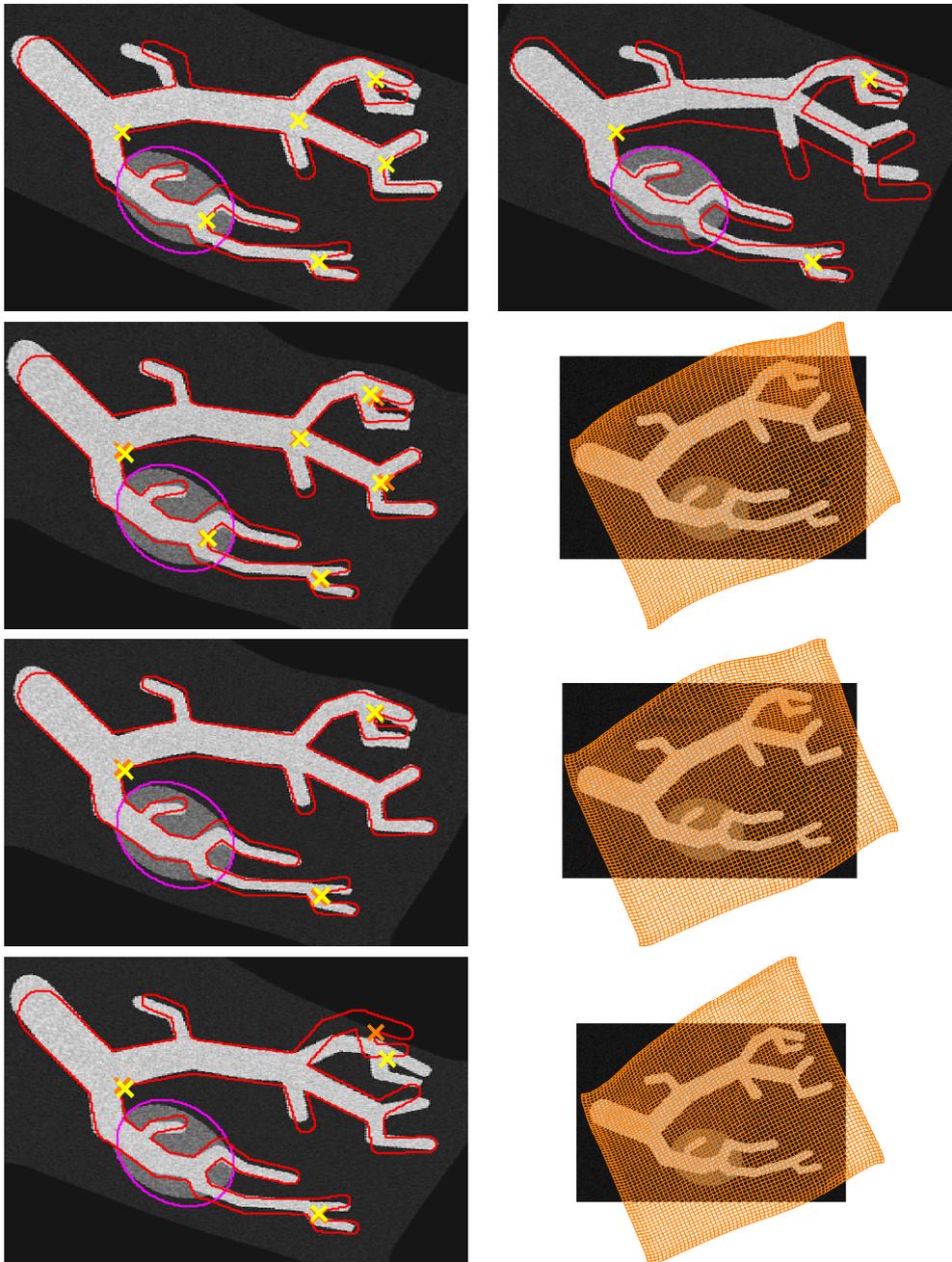


Abbildung 13.3: Ergebnisse SOFT-CoLD; *Erste Zeile:* Thin-Plate-Spline-Registrierung zum Vergleich der Ergebnisse *links:* sechs Landmarken; *rechts:* drei Landmarken; *zweite Zeile:* Ergebnis für $\alpha = 0.1$, $\beta = 0.01$, $\theta = 15$ und sechs Landmarken; *dritte Zeile:* Ergebnis für gleiche Parameter und drei Landmarken; *vierte Zeile:* Ergebnis für $\alpha = 0.1$, $\beta = 0.001$, $\theta = 15$ und drei Landmarken

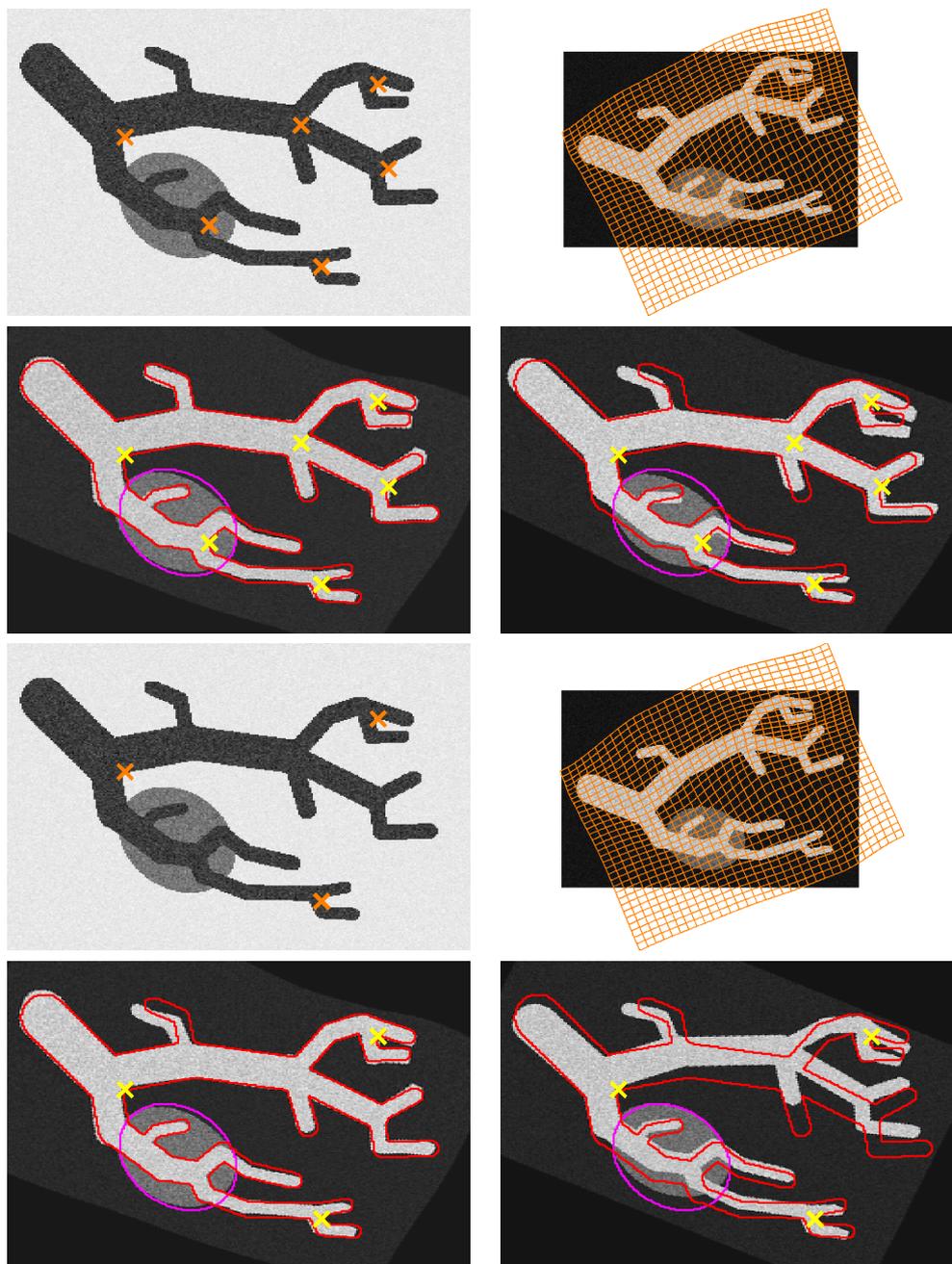


Abbildung 13.4: Ergebnisse CoLD mit SVP; Parameter $\alpha = 0.1$, $\theta = 15$; *obere Hälfte*: für sechs Landmarken; *untere Hälfte* drei Landmarken; jeweils *oben links*: Referenz mit Landmarken; *oben rechts*: Template mit deformiertem Gitter; *unten links*: deformiertes Template mit Landmarken und Rändern des Referenzbildes; *unten rechts*: Landmarkenregistrierung mittels TPS zum Vergleich

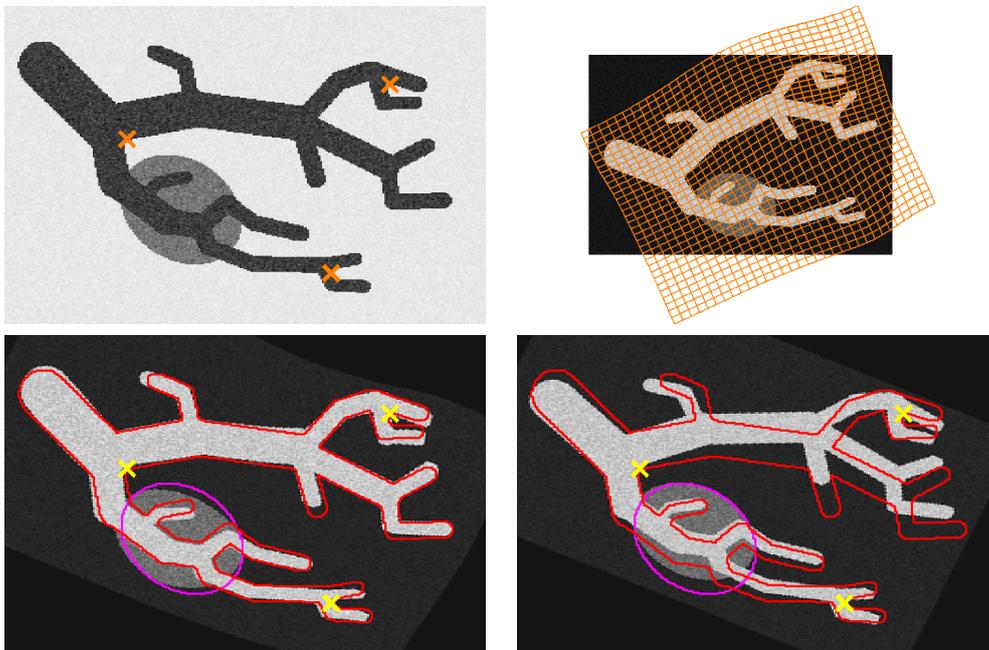


Abbildung 13.5: Ergebnisse CoLD mit SVD; Parameter $\alpha = 0.3$, $\theta = 10$, drei Landmarken; Aufbau wie in Abbildung 13.4

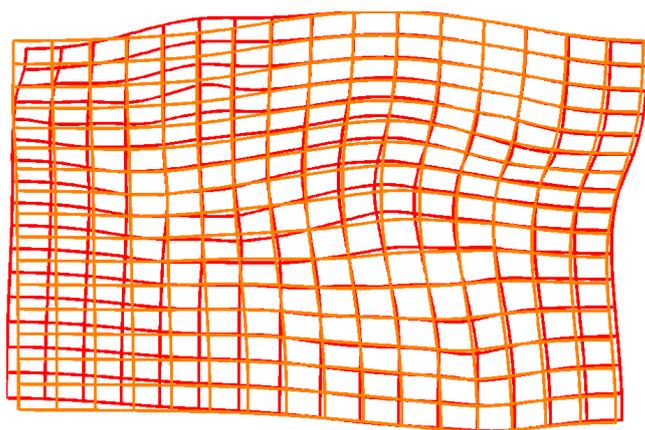


Abbildung 13.6: Vergleich zwischen beiden CoLD-Implementierungen; *rot*: homogener Anteil $\mathbf{y}_h^{\text{SQP}}$ des Verfahrens mit SQP; *orange*: homogener Anteil $\mathbf{y}_h^{\text{SVD}}$ des Verfahrens mit SVD

Teil III

Anwendung und Ergebnisse

Kapitel 14

Anwendung

In diesem Kapitel befassen wir uns schließlich mit der klinischen Anwendung der vorgestellten Methoden. Dazu beschreiben wir zunächst, wie die zu registrierenden Daten in der Klinik aufgenommen und vorverarbeitet werden. Danach zeigen wir an klinischen Daten die prinzipielle Anwendbarkeit der Registrierungs-Algorithmen (proof of concept). Wir beschreiben abschließend das FUSION-Projekt, in dem die Algorithmen eingesetzt werden sollen und diskutieren nächste Schritte hin zu einer klinischen Anwendung.

14.1 Bildgebung

Wir erläutern in diesem Abschnitt die in der Anwendung eingesetzten bildgebenden Verfahren. Für die navigierte Leberchirurgie werden prä-operativ gewonnene aufgearbeitete Computertomographie-Daten zur Operationsplanung und intra-operativer Ultraschall zur Navigation eingesetzt. Diese betrachten wir nun detaillierter ohne aber auf die physikalischen Hintergründe einzugehen, vergleiche [87].

14.1.1 Planungsdaten

Mehrere Tage vor der geplanten Operation werden CT-Daten des Operationsgebietes aufgenommen. Diese erlauben eine feine Auflösung des Gebietes. Durch Kontrastmittelgabe wird versucht, die Gefäßsysteme der Leber deutlich darzustellen. Dabei entstehen hintereinander mehrere Bilder mit unterschiedlich stark eingeströmten Kontrastmittel. Der Ablauf ist dabei wie folgt:

Zu Beginn wird mittels eines Testbolus [37] das patientenindividuelle Einströmverhalten aufgezeichnet, wobei die Zeitspanne zwischen venöser Gabe und Anflutung im Herzen gemessen wird. Aus dieser Zeitspanne werden Zeitpunkte bestimmt, bei der nach Gabe des Kontrastmittels dieses in den Leberarterien,

im Pfortadersystem und in die Lebervenen einströmt. Zu diesen Zeitpunkten wird jeweils eine Aufnahme gemacht, die ein Gefäßsystem der Leber deutlich darstellt:

1. arterielle Phase,
2. portalvenöse Phase,
3. spätvenöse (oder auch lebervenenöse) Phase

In Abbildung 14.1 sind aus einem Patientendatensatz je eine Schicht der portalvenösen (links) und der spätvenösen (rechts) Phase dargestellt; deutlich sind Unterschiede zwischen den dargestellten (angeschnittenen) Gefäßsystemen sichtbar.

Die Daten werden jeweils in ausgeatmetem Zustand aufgenommen, dennoch ist von einer Patientenbewegung zwischen den Aufnahmen auszugehen, so dass eine einfache Überlagerung der Daten, um einen Datensatz mit drei sichtbaren Gefäßsystemen zu erhalten, nicht möglich ist. Dafür ist wiederum ein Registrierungsproblem zu lösen, das aber nicht Gegenstand dieser Arbeit ist.

Nach diesem Registrierungsschritt werden aus den CT-Daten die Planungsdaten bestimmt: Hierzu werden die Gefäße segmentiert und mittels Skelettierungsverfahren [12] in Mittelpunktslinien und Radien parametrisiert. Jedes System wird als Folge zusammenhängender und sich verzweigender Röhren dargestellt. Anhand dieser Daten wird ein patientenindividuelles Lebermodell [75] erstellt, das die individuelle Lage der Lebersegmente approximiert. Anhand dieses Modelles werden dann eine Operationsplanung und klinisch relevante Parameter, wie das funktionale Restvolumen, bestimmt. Eine solche Planung ist im Einleitungskapitel in Abbildung 1.2 dargestellt.

14.1.2 Navigationsdaten

Intra-operativ wird aufgrund der Zeit- und Platzrestriktionen während der Operation Ultraschall zur Navigation genutzt. Zu physikalischen Grundlagen des Ultraschall, vergleiche [87].

Bei der Ultraschall-Bildgebung gibt es zwei typische Modalitäten:

1. **B-mode Daten:** Die Grauwerte der aufgenommenen Bilder korrespondieren zur Eigenschaft des Gewebes, Schall zu reflektieren. An Gewebegrenzen treten deutliche Gradienten auf.
2. **Power-Doppler-Daten:** Durch die Ausnutzung des Doppler-Effektes [17] ist es möglich, die Richtung und Geschwindigkeit bewegter Flüssigkeiten zu messen und zu visualisieren.

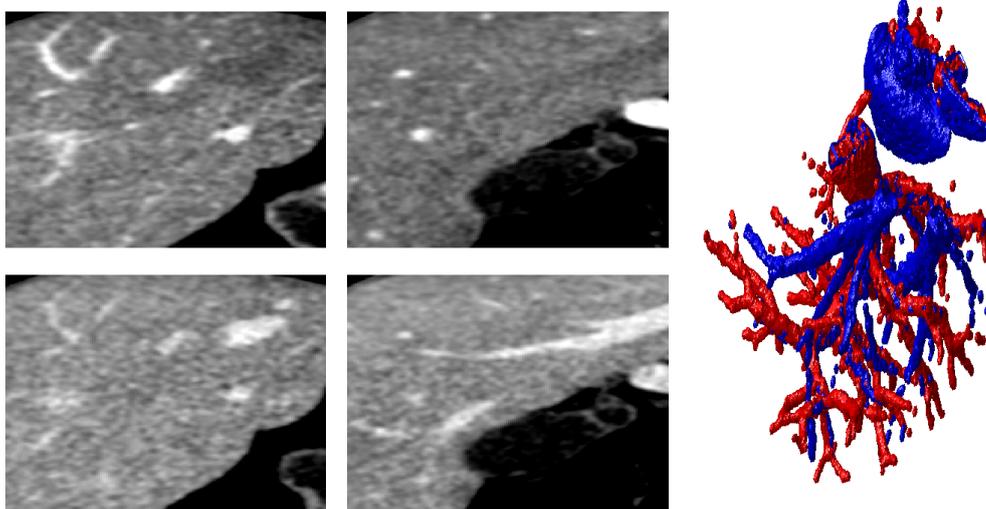


Abbildung 14.1: Ausschnitt aus prä-operativen Leber-CT-Aufnahmen; *oben*: Anschnitte aus portalvenöser Phase *unten*: gleiche Anschnitte aus spätvenöser Phase; *rechts*: 3D-Visualisierung der Gefäße *blau*: Lebervenen; *rot*: Portalvenen

Ultraschall-Daten sind anfällig für Rauschen und sind häufig mit Artefakten versehen. Diese entstehen, wenn Objekte den ausgesendeten Schall komplett reflektieren (Schallschatten) oder nicht direkt zurücksenden (Schallstreuung).

In Abbildung 14.2 sind links intra-operativ gemessene B-mode Daten gezeigt, auf denen die Gefäße zu sehen sind, in gleicher Abbildung rechts sind Power-Doppler-Daten gezeigt, wobei durch Farbe Richtung und Geschwindigkeit kodiert sind. Im linken Bild sind deutlich Schallschatten zu erkennen.

Es sind mehrere Techniken bekannt, um mit Ultraschall ein dreidimensionales Volumen aufzunehmen.

- Die Verwendung eines dreidimensionalen Kopfes, der aus aneinandergereihten Köpfen für normale (zweidimensionale) Ultraschall-Bildgebung besteht.
- Die Verwendung eines herkömmlichen Kopfes, der mittels eines Motors über das aufzunehmende Volumen geschwenkt wird.
- Die Verwendung eines herkömmlichen Kopfes, der per Hand über das Volumen bewegt wird.

Allen Aufnahmemöglichkeiten ist gemein, dass der Ultraschallkopf getrackt wird (vergl. Anhang A.3), um die Position des aufgenommenen Volumen im Raum bestimmen zu können, um dazu die ebenfalls zu navigierenden Instrumente in Relation zu setzen.

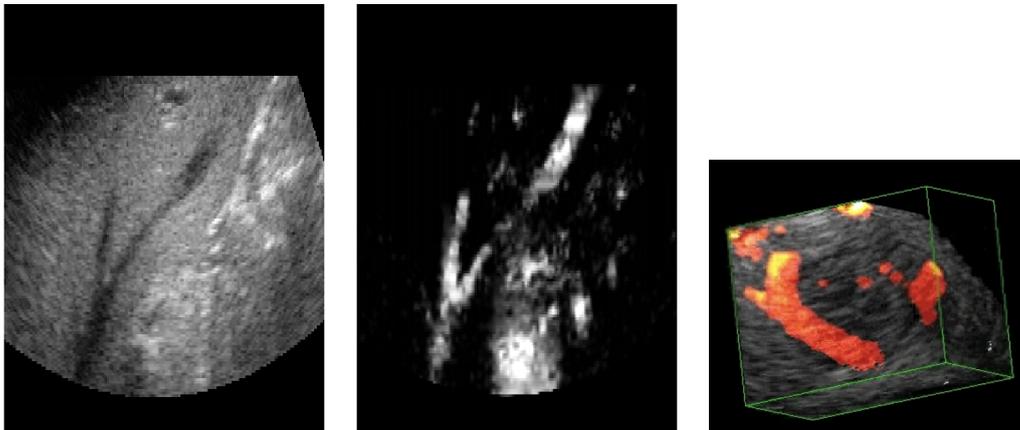


Abbildung 14.2: Ultraschall-Bildgebung; *links*: B-Mode Daten; *Mitte*: Power-Doppler Daten; beide Datensätze zeigen den gleichen Organausschnitt; *rechts*: Power-Doppler-Ultraschall-Volumen; in *grün*: Bounding-Box

14.2 Ergebnisse

14.2.1 Datenlage

Wir nutzen Daten zum Testen der Verfahren, die aus einer Kooperation mit Thomas Lange und Prof. Dr. Peter M. Schlag von der Klinik für Chirurgie und chirurgische Onkologie, Charité Universitätsmedizin Berlin stammen. In dieser Arbeitsgruppe werden ähnliche Aufgaben bearbeitet, wie innerhalb des FUSION-Projekts, vergleiche [5]. Sie ist nicht Mitglied im Konsortium.

Die Aufarbeitung der prä-operativen Daten erfolgt innerhalb der Arbeitsgruppe. Dabei werden auch hier aus den Daten die Gefäßbäume, die Tumore und das Lebergewebe segmentiert, die Gefäßbäume skelletiert und daraus eine Planung für die Operation entworfen. Prä-operative Landmarken werden halb-automatisch mittels der Skelettierung gesetzt.

Intra-operativ wird ein dreidimensionales Ultraschallvolumen aufgenommen. Es wird ein Volusion 730 der Firma GE Healthcare eingesetzt. Die Aufnahme eines Volumens dauert ca. 15sec, Power-Doppler- und B-mode-Daten werden gleichzeitig aufgenommen. Die aufgenommenen Daten liegen auf einem Torusausschnitt. Das Gerät errechnet die Darstellung des Volumens in kartesischen Koordinaten. Aufgrund der Ultraschall-Kopfgröße kann das Verfahren nur in der offenen Leberchirurgie eingesetzt werden. Die Landmarken in den Planungsdaten werden per Hand gesetzt.

Die Größe der Daten ist $m = (214, 156, 192)$, wobei die CT-Daten auf die Größe der Ultraschall-Daten zugeschnitten (gecrop) worden sind. Die Auflösung beträgt nach dem Zuschneiden sowohl für die Ultraschall-Daten als auch für die CT-Daten $h = (0.0500, 0.0511, 0.0526)mm$.

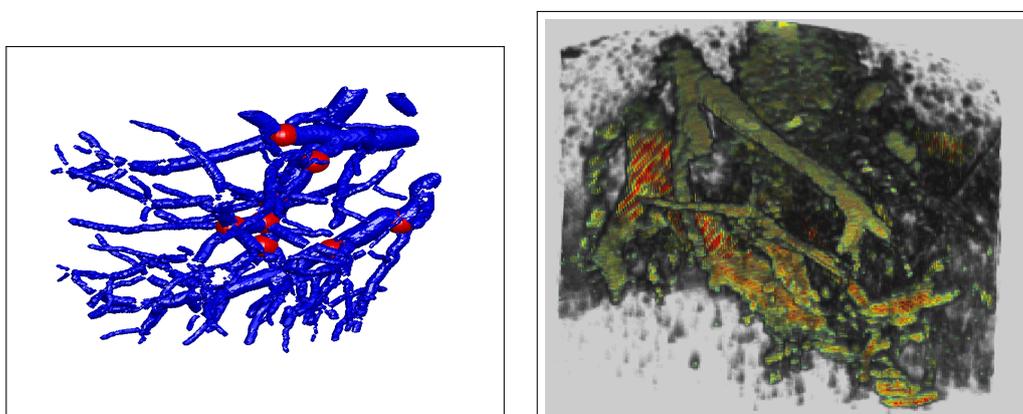


Abbildung 14.3: Illustration der klinischen Daten; *links*: Gefäßbaum CT-Daten; *rot*: Position der Landmarken; *rechts*: Ultraschall-Power-Doppler Volumen

Von den verwendeten Daten sind exemplarisch in Abbildung 14.3 ein CT- und ein Ultraschall-Volumen visualisiert. Hierbei beschränken wir uns darauf, die Gefäßbäume zu zeigen, die schon mittels rigider Landmarkenregistrierung vorbearbeitet sind, um die Orientierung der Daten zu einander zu verbessern. In der Abbildung sind acht Paare von korrespondierenden Landmarken eingezeichnet, die in der späteren Anwendung genutzt werden.

14.2.2 Generelles Vefahren

Wir zeigen die Ergebnisse verschiedener Registrierungsalgorithmen:

Als Testrechner dient ein Apple iMac 2.8 GHz Intel Core 2 Duo mit 4 GB RAM. Die Programme sind in MATLAB implementiert.

Wir nutzen die landmarken-basierte Registrierung als Vorregistrierung, um einen guten Startwert für die Daten zu erhalten.

Damit die Landmarkeninformation nicht verloren geht, nutzen wir kombinierte Verfahren.

Als Eingabe für die Registrierung dienen Power-Doppler-Ultraschalldaten als Referenz und die CT-Daten als Template.

Als Distanzmaß nutzen wir das NGF-Maß; als Regularisierer nutzen wir den elastischen Regularisierer.

Wir visualisieren die Ergebnisse, indem wir einzelne Schichten aus dem Referenzbild abbilden und Kanten des (deformierten) Templatebildes darüber legen. Dies entspricht den Anforderungen für die spätere medizinische Anwendung im Operationssaal, wo Gefäße im Ultraschallbild mit Informationen des Templatebildes überlagert werden sollen.

Die Kanten im Templatebild bestimmen wir durch einen einfachen Schwellwert (das heißt es werden Kanten des Binärbildes $T_y \geq c$ betrachtet).

14.2.3 Soft-CoLD

Als erstes Verfahren verwenden wir das Soft-CoLD-Verfahren (Kapitel 13.3), bei dem die Landmarkenbedingung durch einen zusätzlichen Strafterm berücksichtigt wird. Als Gewichtungssparameter wählen wir $\beta = 0.1$, der Regularisierungsparameter ist $\alpha = 0.1$.

Die Deformation wurde in einem Multiresolutionsverfahren mit $m = (16, 16, 16)$ und $m = (32, 32, 32)$ bestimmt. Die so gewonnene Transformation ist mittels Interpolation auf die Auflösung der Daten gebracht worden.

Die Daten wurden in einem kombinierten Multiresolution-Multiskalen-Ansatz für die Registrierung eingebettet. Es wurde ein Gaußscher Skalenraum genutzt.

In Abbildung 14.4 sind die Ergebnisse des Algorithmus für den realen Datensatz gezeigt. In der linken Spalte sind drei ausgewählte Schichten des Referenzbildes visualisiert. Die Lage der Schichten im Datensatz ist in gleicher Abbildung in der obersten Zeile illustriert. Als rote Kanten sind die angeschnittenen Gefäßkanten des Templatebildes vor der Registrierung darübergerlegt. In der mittleren Spalte sind die gleichen Schichten überlagert mit den Anschnitten der Kanten nach einer Thin-Plate-Spline basierten Landmarkenregistrierung (Kapitel 12.4). Eine deutliche Verbesserung der Überlagerung von Gefäßen ist gerade in den zentralen Strukturen in der Bildmitte zu erkennen. In der rechten Spalte sind wiederum die Schichten mit dem Ergebnis der Soft-CoLD-Registrierung überlagert (gelb). Eine deutliche Verbesserung gegenüber dem Thin-Plate-Spline-Verfahren ist zu erkennen, da die Gefäßränder des Templates gut zu denen der Referenz passen.

Die Abstände der Landmarken vor und nach der Registrierung sowie der gemittelte Landmarkenfehler, der durch das Verfahren minimiert werden soll, sind in Tabelle 14.1 dargestellt. Der mittlere Fehler wird minimiert, obwohl einzelne Landmarkenabstände wachsen (Landmarke 8).

14.2.4 CoLD mit SVD

In einem zweiten Versuch haben wir die den CoLD-Algorithmus getestet, wobei wir die auf der SVD-Zerlegung basierende Variante gewählt haben, um das Auftreten indefiniter KKT-Systeme zu umgehen.

Die Vorverarbeitung der Daten ist gleich der im vorherigen Abschnitt. Auch hier wurde die Deformation in einem Multiresolutionsverfahren mit

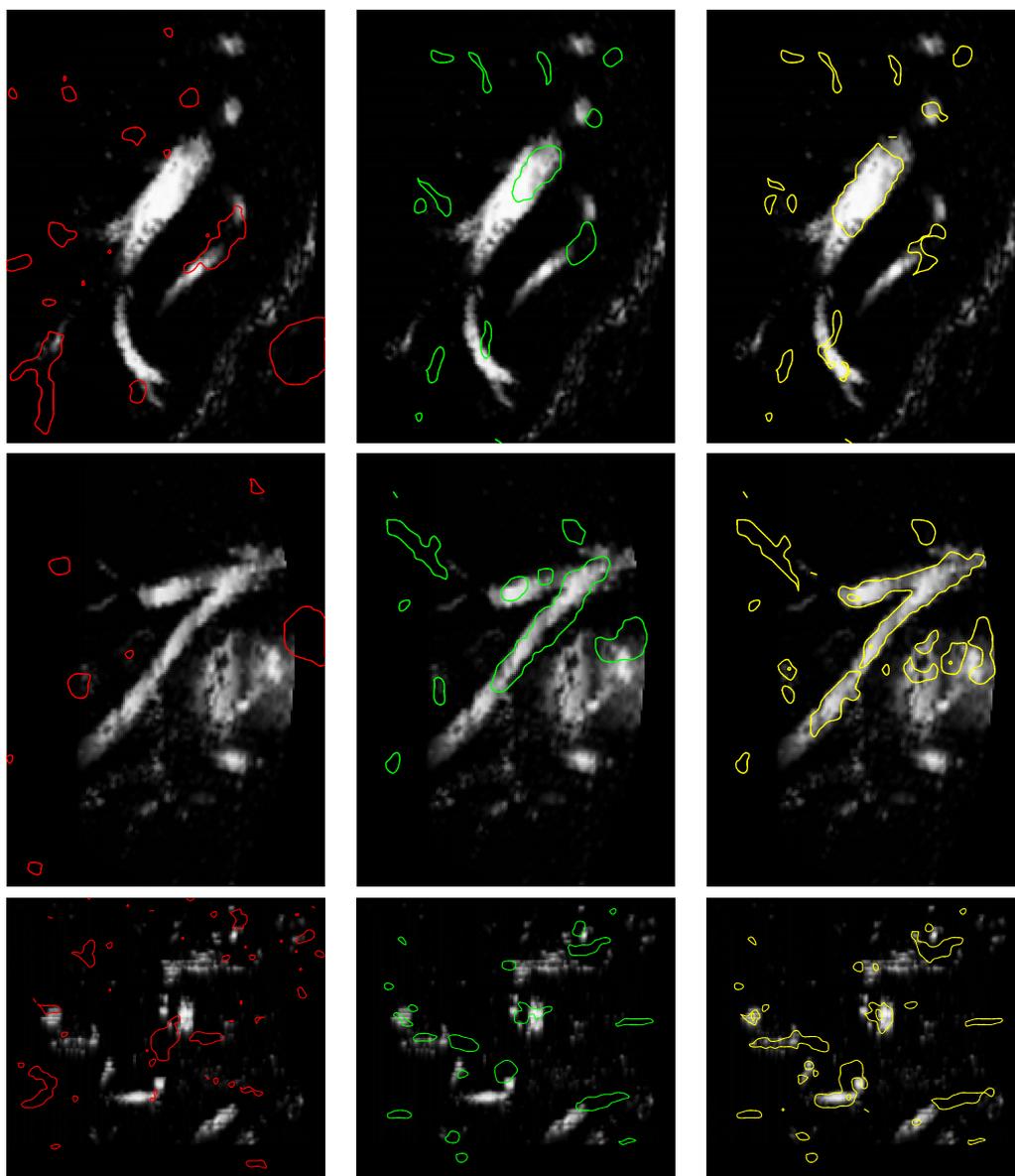


Abbildung 14.4: Ergebnis Soft-CoLD für klinische Daten; *je Zeile*: ein Schnitt durch das Ultraschall-Volumen; Überlagerung der Schnitte mit Kanten des Templatebildes *linke Spalte (rote Ränder)*: Kanten des rigide vorregistrierten CT-Volumens; *mittlere Spalte (grüne Ränder)*: Kanten des mittels TPS-Registrierung deformierten CT-Volumens; *rechte Spalte (gelbe Ränder)*: Kanten des mittels Soft-CoLD-Verfahrens deformierten CT-Volumens

Tabelle 14.1: Abstand der Landmarken vor und nach Soft-CoLD Registrierung

LM	vor	nach
1	0.8773	0.1256
2	0.8123	0.1524
3	0.2124	0.1079
4	0.2748	0.3524
5	0.2049	0.4481
6	0.2512	0.2297
7	0.5895	0.6518
8	0.2968	0.7399
Mittelwert	0.4399	0.3510

$m = (16, 16, 16)$ und $m = (32, 32, 32)$ bestimmt und dann mittels Interpolation auf die Auflösung der Daten gebracht.

Die Ergebnisse sind in Abbildung 14.5 dargestellt. Der Aufbau ist analog zu den vorherigen Ergebnissen. Zu diskutieren sind die Unterschiede zwischen TPS- und CoLD-Registrierung. Beide Verfahren bilden die Landmarken exakt aufeinander ab. Dennoch ist in der mittleren Spalte zu erkennen, dass die Landmarkenregistrierung allein nicht ausreicht, die Gefäße aufeinander abzubilden. Der CoLD-Ansatz, der zusätzlich Bildinformationen nutzt, erzielt visuell überzeugendere Ergebnisse.

Der gleiche Algorithmus ist an weiteren klinischen Datensätzen getestet worden. Die Ergebnisse sind in den Abbildungen 14.6 und 14.7 visualisiert. In der ersten sieht man, dass auch bei vielen kleinen Gefäßen der Algorithmus die Gefäße aufeinander abbildet.

14.2.5 Fokussierung

Wir demonstrieren als letztes Beispiel die Wirkungsweise der in Kapitel 11 vorgestellten Fokussierungsstrategie an klinischen Daten. Wir verwenden dazu den ersten Datensatz aus dem vorherigen Abschnitt. Als Zielgebiet wählen wir eine Gefäßverzweigung, die wir als Anschnitt auch in Abbildung 14.5, zweite Zeile gezeigt haben. Das Zielgebiet ist manuell ausgewählt worden und ist gegeben durch $\Upsilon = (-2.4616, -0.9625) \times (7.9656, 9.4983) \times (3.4489, 4.5011)$. In Abbildung 14.8 ist links ein Anschnitt des Ultraschallvolumens samt Anschnitte der Zielgebiete dargestellt, die in der Fokussierungsstrategie von außen nach innen genutzt werden.

Wir gehen wie folgt vor: Zunächst berechnen wir basierend auf den gegebenen acht Landmarken eine Thin-Plate-Spline-Registrierung. Das Ergebnis ist

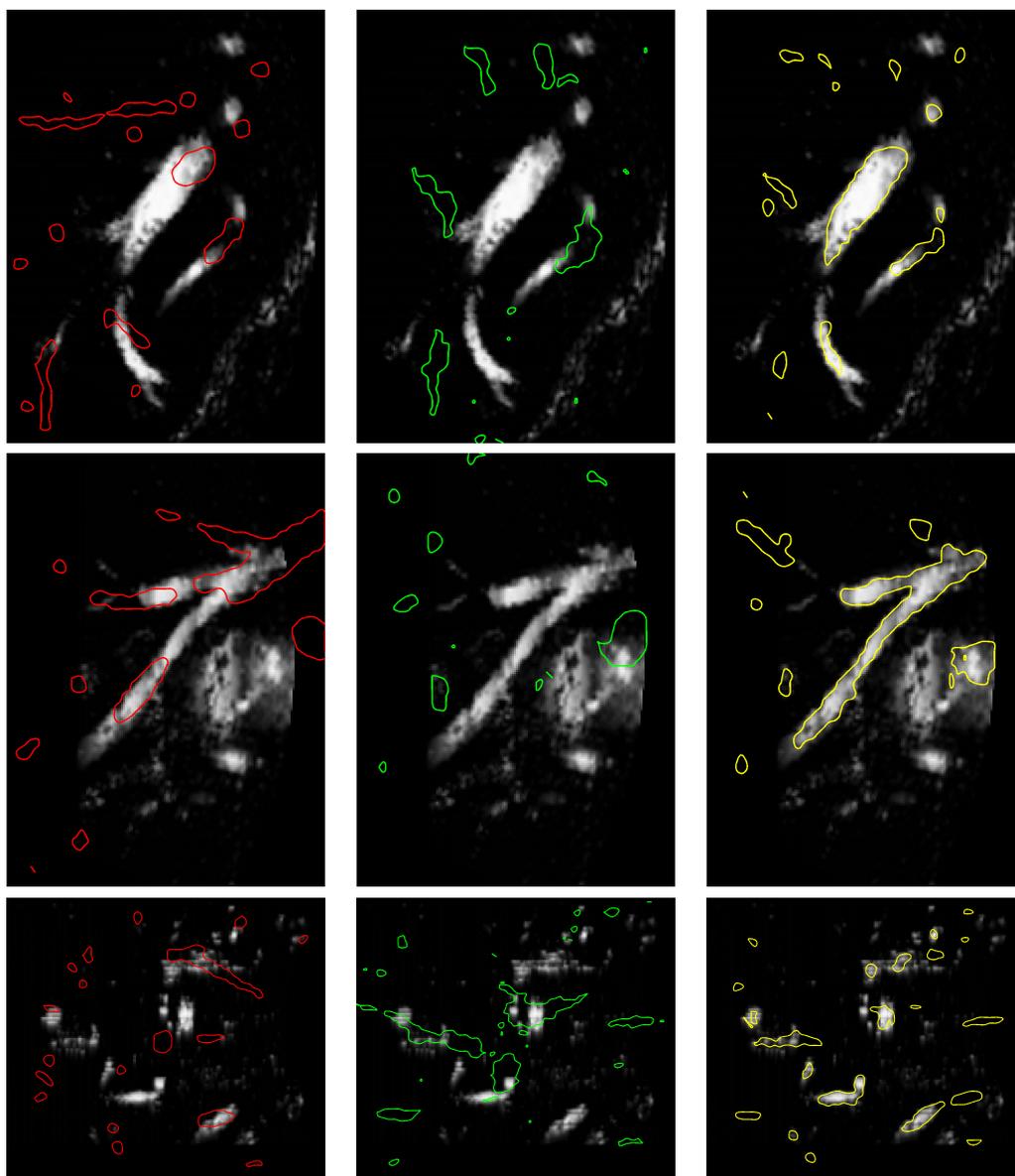


Abbildung 14.5: Ergebnis CoLD für klinische Daten; erster Datensatz; je Zeile: ein Schnitt durch das Ultraschall-Volumen; Überlagerung der Schnitte mit Kanten des Templatebildes *linke Spalte (rote Ränder)*: Kanten des rigide vorregistrierten CT-Volumens; *mittlere Spalte (grüne Ränder)*: Kanten des mittels TPS-Registrierung deformierten CT-Volumens; *rechte Spalte (gelbe Ränder)*: Kanten des mittels CoLD-Verfahrens deformierten CT-Volumens

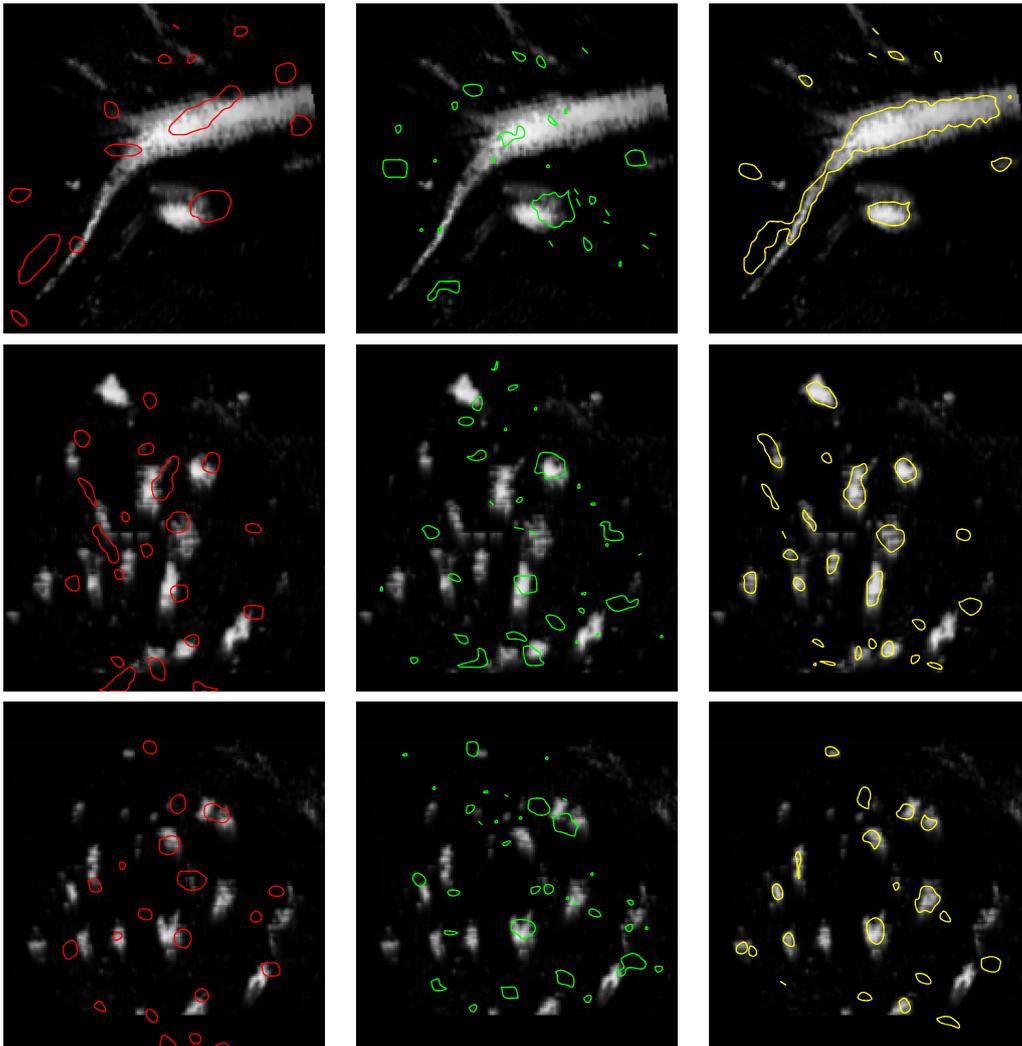


Abbildung 14.6: Ergebnis CoLD für klinische Daten; zweiter Datensatz; Aufbau der Abbildung wie in Abbildung 14.5

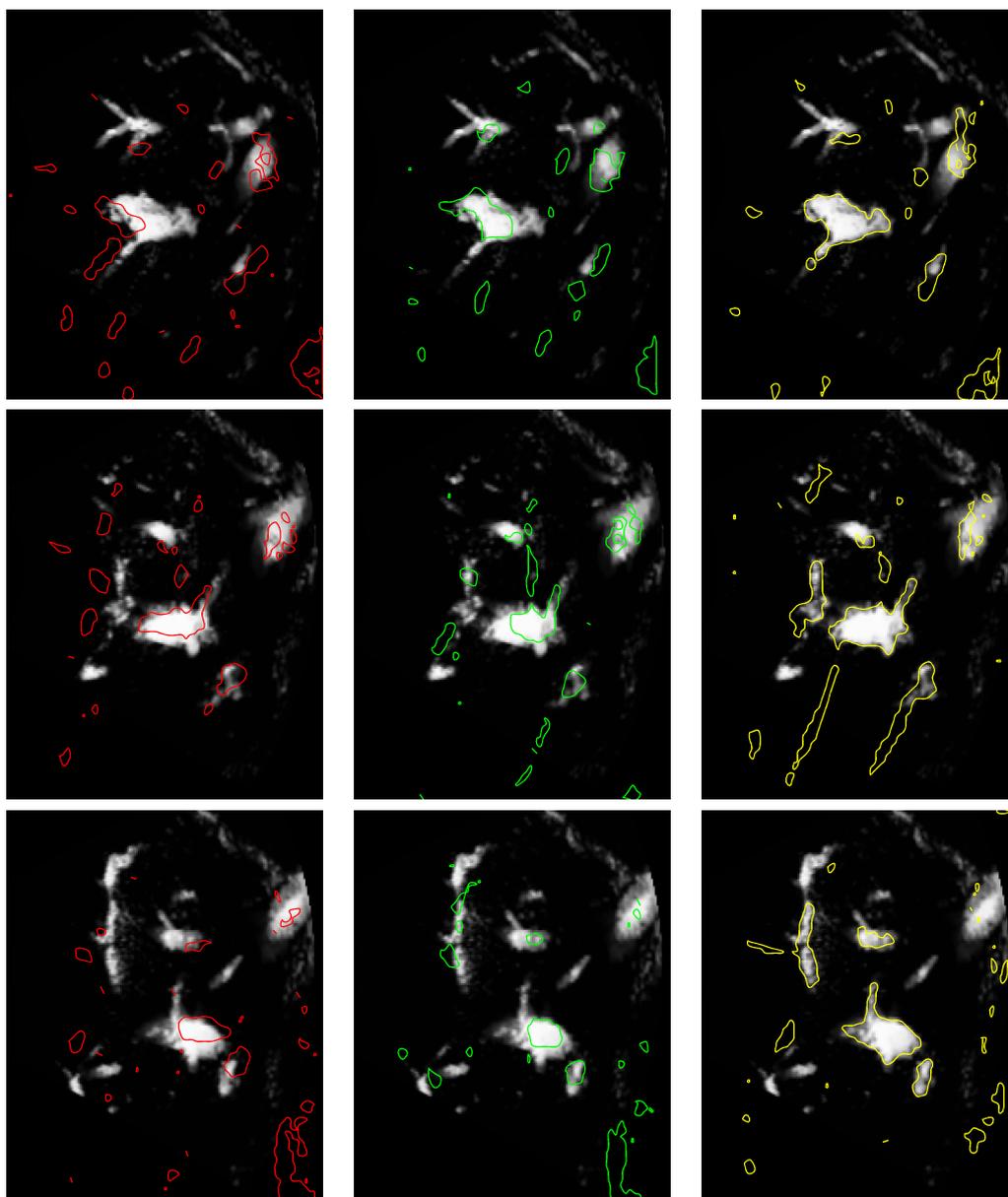


Abbildung 14.7: Ergebnis CoLD für klinische Daten; dritter Datensatz; Aufbau der Abbildung wie in Abbildung 14.5

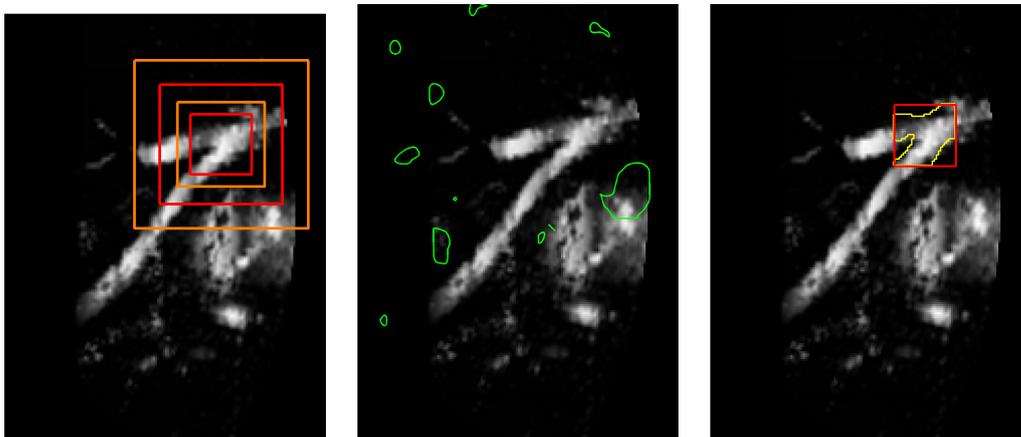


Abbildung 14.8: Ergebnis Fokussierung für ersten Datensatz; *links*: Ausgewählte Schicht des Referenzbildes mit Rändern der quaderförmigen Zielgebiete für diese Schicht; *Mitte*: Ergebnis der TPS-Registrierung (Startwert für Fokussierung); *rechts*: Ergebnis der Fokussierung für das Zielgebiet

in Abbildung 14.8, Mitte dargestellt. Dieses übertragen wir auf das äußerste Zielgebiet. Wir nutzen im weiteren das NGF-Maß und einen elastischen Regularisierer mit $\alpha = 1$. Jedes einzelne Zielgebiet wird mit $m = (16, 16, 16)$ Punkten diskretisiert. Damit ist die Auflösung der Deformation im Zielgebiet während der Registrierung weitaus höher als bei den vorherigen Verfahren, jetzt $h = (0.0937, 0.0958, 0.0658)$, bisher $h = (0.3342, 0.2491, 0.3157)$

Das Ergebnis der Fokussierungsstrategie ist in Abbildung 14.8, rechts, dargestellt. Innerhalb des Zielgebietes ist das Referenzbild mit den Kanten des deformierten Templatebildes überlagert. Zu erkennen ist, dass das Registrierungsergebnis qualitativ ähnlich dem des CoLD-Verfahrens ist vergleiche Abbildung 14.5, zweite Zeile, rechts.

Der Vorteil des Verfahrens liegt in der benötigten Rechenzeit, die wir bereits in Kapitel 11.2 vorgestellt haben.

Die Ergebnisse zeigen, dass die Fokussierung auch bei realen, klinischen Daten eingesetzt werden kann. Somit ist es erstmalig möglich, nicht-lineare Registrierungsverfahren in zeitkritischen Anwendungen einzusetzen.

14.2.6 Fazit

Mit diesen Ergebnissen haben wir gezeigt, dass variationelle Registrierungsverfahren im Bereich der navigierten Leberchirurgie einsetzbar sind und zu guten Ergebnissen führen. Es ist auch deutlich geworden, dass ein naives Vorgehen nicht zielführend ist, sondern erst die Kombination verschiedener Ansätze den Einsatz dieser Methoden ermöglicht. Dies sind:

- Eine Aufarbeitung der Eingangsdaten durch Multiresolutions- und Multi-skalenverfahren.
- Die Verwendung spezieller Distanzmaße.
- Die Nutzung von Zusatzinformationen, in diesem Fall von Landmarken.
- Die Einbettung der Landmarkenbedingungen in Form eines Strafterms (Soft-CoLD) oder durch die Formulierung von Nebenbedingungen für die Optimierung (CoLD).

Die Ergebnisse wurden zum Teil bereits publiziert. Der Soft-CoLD-Ansatz ist für ein parametrisches Bewegungsmodell auf der SPIE 2008 Medical Imaging [60] vorgestellt worden; erste Ergebnisse des CoLD-Ansatzes auf der CARS 2008 [47].

Damit liegt ein prinzipieller „proof of concept“ vor.

14.3 FUSION

In einem nächsten Schritt werden die Verfahren im FUSION-SOMIT-Projekt eingesetzt. Hierbei ist SOMIT (Schonendes Operieren mit innovativer Technik) der Name eines vom Bundesministerium für Bildung und Forschung (BMBF) getragenen Förderprogrammes und FUSION (Future Environment for gentle Liver Surgery using Image-guided Planning and Intra-operative Navigation) ein innerhalb dieses Programms gefördertes Verbundvorhaben. Es wird über einen Zeitraum von fünf Jahren gefördert und befasst sich mit der Verbesserung der Leberchirurgie. Es wird von Prof. Dr. med H.P. Bruch, Direktor der Klinik für Chirurgie am Universitätsklinikum Schleswig-Holstein, Campus Lübeck geleitet.

Das Projekt vereint mehr als 25 Partner aus Kliniken, Forschungsinstituten und der Industrie aus ganz Deutschland.

Wir fokussieren in dieser Arbeit auf die an der Lübecker Klinik für Chirurgie erforschte laparoskopische Leberchirurgie:

- Die prä-operativen Daten werden von der Firma MeVis Distance Service GmbH in Bremen aufgearbeitet. Diese erarbeiten eine Segmentierung der Gefäßsysteme und, daraus abgeleitet, eine Einteilung der Leber in ihre Segmente. Die Firma führt auch eine Risikoanalyse durch, in der ausgehend von der geplanten Resektion des Tumors das funktionale Restvolumen der Leber bestimmt wird. Durch diese Analyse kann der Therapieerfolg vor der Operation abgeschätzt werden.

- In die Planungsdaten wird die Position einer Vielzahl von Landmarken eingezeichnet. Diese Landmarken bestehen aus anatomischen Punkten, die typischerweise an Bifurkationspunkten großer Lebergefäße liegen. Beispiele sind die erste und zweite Pfortaderverzweigung und der Lebervenenstern.
- Als zentrale Schnittstelle und Steuereinheit dient die Software *LapAssistent*, die an der Klinik für Chirurgie und dem Institut für Robotik der Universität zu Lübeck entwickelt worden ist:
 - Im LapAssistenten laufen die Tracking-Informationen sämtlicher an der Operation beteiligten und getrackten Instrumente zusammen.
 - Das System sammelt sämtliche Daten der bildgebenden Verfahren während der Operation, wie die Bilder der Ultraschall-Sonde und die der laparoskopischen Kamera.
 - Innerhalb des LapAssistenten gibt es ein Programm zur Eingabe von Landmarken. Zur intra-operativen Lokalisierung einer Landmarke steuert der Chirurg diese mittels Ultraschall an. Ist die Landmarke in der Ultraschallebene sichtbar, wird das Bild eingefroren und die Position der Landmarke im Bild in Pixelkoordinaten festgelegt. Durch Angabe der korrespondierenden Landmarke in den Planungsdaten, wird ein Paar von Landmarken bestimmt. Mittels der Trackinginformationen lässt sich die Position der Landmarke im Raum errechnen, vergleiche Anhang A.3
 - Der LapAssistent visualisiert die gesammelten Daten und verbindet sie zu einer virtuellen Realität. Dies wird in Form mehrerer Ansichten verwirklicht. Die Bildschirmaufteilung ist dabei festgelegt, um dem Chirurgen eine konstante Schnittstelle zu geben. Im linken Bereich werden die Planungsdaten visualisiert. In diese können nach der Registrierung von Planungs- und Navigationsdaten die Lage der Ultraschall-Sonde und die der anderen verwendeten Instrumente eingezeichnet werden. Der Kopf der Ultraschall-Sonde kann dabei um das aktuelle Ultraschallbild in der gegenwärtigen Ebene erweitert werden. So steht dem Arzt zur Navigation eine augmented reality bereit. Ein Bildschirmfoto dieser Anwendung ist in Abbildung 14.9 gezeigt. Oben rechts wird das aktuelle Ultraschallbild dargestellt, unten rechts das aktuelle Bild der laparoskopischen Kamera. Beide Ansichten werden, wenn andere Instrumente in den Bildausschnitt gebracht werden, um diese erweitert.

Die Felder entlang der linken Bildschirmseite steuern die Interaktion mit dem Programm. Sie sind über einen Touchscreen-Monitor direkt im Operationssaal zu bedienen.

- Aus dem LapAssistenten erfolgt der Aufruf und die Steuerung der Registrierung.
- Während der Ultraschall-Aufnahme werden die Trackinginformationen genutzt, um die jeweilige Ultraschallschicht in der gemessenen Orientierung in ein Volumen einzufügen. In Abbildung 14.2, rechts ist ein so bestimmtes Volumen gezeigt. Zu erkennen ist, dass obwohl die Schichten eine nicht verschwindende Dicke haben, das Volumen Löcher enthält. Zum Auffüllen dieser Löcher sind sogenannte Compounding-Verfahren entwickelt worden.

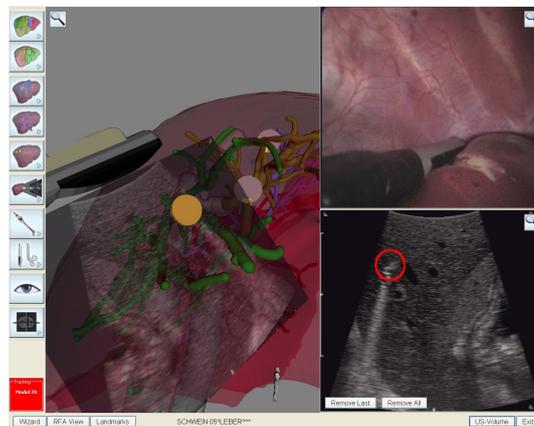


Abbildung 14.9: Screenshot der LapAssistent Software; *links*: Bedienelemente; *Mitte*: Virtuelle Realität mit eingezeichnetet Lage der getrackten Instrumente; *oben rechts*: aktuelles Bild der endoskopischen Kamera; *unten rechts*: aktuelles Ultraschallbild

Kapitel 15

Ausblick

Das Registrierungsproblem von prä-operativen CT-Daten und intra-operativen Ultraschall-Daten ist (natürlich) mit dieser Arbeit nicht abschließend gelöst. In diesem letzten Kapitel werden wir vorausblicken und weiterführende Forschungsgebiete aufzeigen, die Weiterentwicklungen dessen darstellen, was wir in der Arbeit beschrieben haben und diskutieren, welche Schritte nötig sind, um vom vorgestellten „proof of concept“ zum Einsatz in der Praxis zu gelangen.

In Kapitel 10.3 haben wir gesehen, dass kombinierte Skalen- und Multiresolutionsverfahren eine gute Möglichkeit darstellen, die Röhrenstruktur der Daten geeignet zu modellieren. Die dort verwendeten Gaußschen und B-Spline-Skalenräume bilden die Grundlage für Standardverfahren aus der Literatur und stehen in keinem direkten Bezug zu den Daten. Eine weiterführende Möglichkeit besteht nun in der Definition von „Röhrenskalenräumen“ und deren Einsatz in der Registrierung. Dieser Skalenraum basiert auf einer Skelettierung der Daten, das heißt eine Darstellung des Gefäßsystems durch Zylinder, die über die Anfangs- und Endpunkte ihrer Mittellinie sowie ihren Radius gegeben sind, vergleiche Abbildung 15.1.

Der Skalenraum wird definiert über den Zylinderradius. Zunächst werden nur Zylinder mit großem Radius für die Registrierung genutzt, in weiteren Stufen kommen dann Röhren mit kleinerem Radius hinzu. Für die zweidimensionalen Testdaten ist dies in Abbildung 15.2 dargestellt.

Motiviert wird dies Vorgehen dadurch, dass große Gefäße mit anatomisch wich-



Abbildung 15.1: Parametrisierung eines Zylinders; *grün*: Anfangspunkt; *rot*: Endpunkt; *orange*: Mittellinie; *blau*: Radius

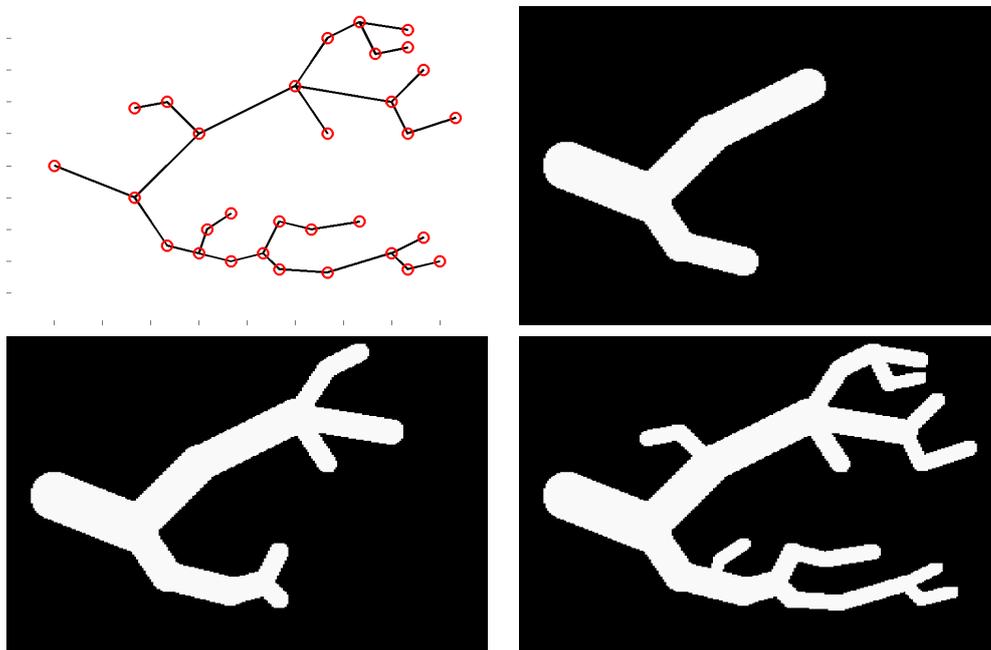


Abbildung 15.2: Skalensraum basierend auf Gefäßradien; *oben links*: Skelettierung der Gefäßdaten; *oben rechts*: Gefäßradius ≥ 0.45 ; *unten links*: Gefäßradius ≥ 0.3 ; *unten rechts*: Gefäßradius ≥ 0.2

tigen Strukturen korrespondieren, zunächst also die grobe Anatomie abgebildet wird. Als zweite Motivation dient die dadurch bewirkte Verminderung lokaler Minima der Zielfunktion, da große Röhren nur auf andere große Röhren abgebildet werden können und nicht auf solche mit kleinerem Radius.

Ein zweiter Arbeitsbereich liegt in der problemangepassten Modellierung der Landmarkenbedingung. Beim intra-operativen Setzen der Landmarken durch den Chirurgen werden unvermeidlich Fehler gemacht, so dass eine exakte Abbildung der Landmarken aufeinander, wie bei den CoLD-Verfahren, unrealistisch ist. Eine Möglichkeit, dies in ein Registrierungsverfahren einzubinden, ist die in Kapitel 13.3 vorgestellte Minimierung des Landmarkenfehlers durch einen Strafterm in der Zielfunktion. Nachteil ist die Daten abhängige Wahl des Gewichtungsparmeters β und die nicht garantierte Verkleinerung dieses Fehlers.

Eine realistische Modellierung ist durch die Formulierung eines restringierten Optimierungsproblems möglich. Hierbei misst $\mathcal{P}(\mathbf{y}) = \|D\mathbf{y} - t\|^2$ den Landmarkenfehler wie in Kapitel 13.3. Gefordert ist nun eine Minimierung der Zielfunktion $\mathcal{J}(\mathbf{y}) = \mathcal{D}(\mathbf{y}) + \alpha\mathcal{S}(\mathbf{y})$ unter der Nebenbedingung, dass $\mathcal{P}(\mathbf{y})$ kleiner als eine vorgegebene Toleranz ist:

$$\begin{aligned} \mathcal{J}(\mathbf{y}) = \mathcal{D}(\mathbf{y}) + \alpha\mathcal{S}(\mathbf{y}) &\rightarrow \min \\ \text{u.d.N. } \mathcal{P}(\mathbf{y}) &\leq \text{tol.} \end{aligned}$$

Vorteil gegenüber dem Soft-CoLD-Ansatz ist der Wegfall des Gewichtungsparameters β und der garantierte Landmarkenfehler. Gelöst wird das Problem durch Methoden der restringierten Optimierung für Ungleichheitsbedingungen (beispielsweise ein SQP-Verfahren) [56].

In einem zweiten Schritt ist es möglich, für jedes Landmarkenpaar je eine Toleranz zu definieren. Dies erlaubt dann die Verwendung von Landmarken unterschiedlicher Klasse. Es können neben den bisherigen Landmarken, die an anatomisch ausgezeichneten Punkten (Gefäßverzweigungen) platziert sind, ungenaue Landmarken der Form „Leber links“ verwendet werden. Diese lassen sich intra-operativ aufgrund ihrer Ungenauigkeit leicht ansteuern, wodurch sich das Setzen beschleunigt. Die Ungenauigkeit wird durch die Vorgabe unterschiedlicher Toleranzen modelliert.

Die numerische Umsetzung ist wie folgt: Sei N_{LM} die Anzahl der gegebenen Landmarken und D_{r_j} die diskrete PEF für die j . Landmarke (vergleiche Kapitel 13.2), dann ist mit

$$\mathcal{C}(\mathbf{y}) = \begin{pmatrix} \|D_{r_1}\mathbf{y} - \bar{t}_{1,:}^\top\|^2 \\ \vdots \\ \|D_{r_{N_{LM}}}\mathbf{y} - \bar{t}_{N_{LM},:}^\top\|^2 \end{pmatrix}$$

und mit

$$tol_{LM} = (tol_1, \dots, tol_{N_{LM}})^\top$$

das restringierte Problem

$$\begin{aligned} \mathcal{J}(\mathbf{y}) = \mathcal{D}(\mathbf{y}) + \alpha\mathcal{S}(\mathbf{y}) &\rightarrow \min \\ \text{u.d.N. } \mathcal{C}(\mathbf{y}) &\leq tol_{LM} \end{aligned}$$

mit N_{LM} Nebenbedingungen gegeben, wobei die \leq -Relation elementweise zu lesen ist.

Für grob platzierte Landmarken ist eine große, für hinreichend genau platzierte Landmarken eine kleine Toleranz zu wählen. Auch dieses Problem lässt sich mit Methoden der restringierten Optimierung (SQP) lösen.

Mit diesen beiden neuen Verfahren sind zusammen mit dem Soft-CoLD- und dem CoLD-Verfahren vier Methoden beschrieben, die Landmarken und Grauwertinformationen zur Bestimmung einer Deformation kombinieren. In Zusammenarbeit mit Thomas Lange (Charité Universitätsmedizin Berlin) ist geplant, in einer Studie die Einsatzmöglichkeiten dieser Verfahren an klinischen Daten zu vergleichen.

Neben diesen methodischen Weiterentwicklungen ist der Einsatz der Registrierungsalgorithmen im FUSION-Projekt nächstes Ziel.

In der laparoskopischen Ultraschallbildgebung werden bisher nur Schallköpfe eingesetzt, die zweidimensionale Ultraschallbilder erzeugen. Deren Lage im

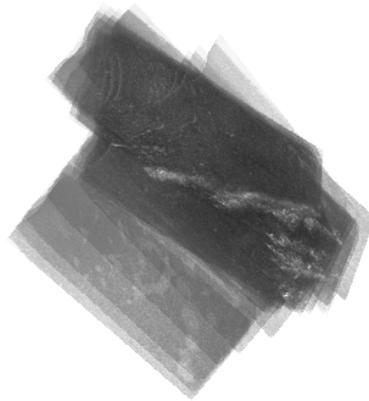


Abbildung 15.3: Ultraschall-Volumen; einzelne zweidimensionale Schichten sind in mittels Tracking-Informationen in ein Volumen eingezeichnet worden

Raum ist durch die Trackinginformationen bekannt, so dass sie in ein Volumen eingetragen werden können. Dieses weist systematisch Lücken auf, vergleiche Abbildung 15.3. Bisher werden in einem zweiten Schritt Compounding-Verfahren [68] eingesetzt, die die Bestimmung eines dichten Volumens ermöglichen. Hierbei geht die Eigenschaft des Volumens, in den Ultraschallschichten hoch aufgelöst zu sein, verloren. Eine neue Möglichkeit wäre das Erstellen eines Gitters aus den Trackinginformationen, das die Positionen der aufgenommenen Schichten beinhaltet. Nur an diesen Gitterpunkten würde die Distanz zwischen prä- und intra-operativen Daten ausgewertet, die Deformation aber für das vollständige Volumen bestimmt.

Der klinische Test der Registrierungsmethoden ist innerhalb einer klinischen Studie im FUSION-Projekt geplant. Hierzu werden die Registrierungsalgorithmen in einer Bibliothek implementiert werden und über die LapAssistent-Software angesteuert. Geplanter Beginn der Studie ist das dritte Quartal 2009.

Damit beenden wir die Arbeit.

Anhang A

Wir fassen im Anhang drei Sachverhalte zusammen, die den Inhalt voriger Kapitel ergänzen.

A.1 Singulärwert-Zerlegung

Für das in Kapitel 13.5 beschriebene Verfahren benötigen wir die Singulärwertzerlegung einer Matrix. Diese definieren wir hier, sie ist [28] entnommen.

Definition A.1.1 Sei $A \in \mathbb{R}^{m \times n}$ eine Matrix mit $\text{rank} A = r$ und seinen $\lambda_j \in \mathbb{R}$, $j = 1, \dots, r$ die Eigenwerte von $A^\top A$. Dann heißen

$$\sigma_j = \sqrt{\lambda_j}, \quad j = 1, \dots, r$$

die **Singulärwerte** von A .

Die zugehörigen Eigenvektoren $u_j \in \mathbb{R}^n$ von $A^\top A$ heißen **Rechtssingulärvektoren** von A und die zugehörigen Eigenvektoren $v_i \in \mathbb{R}^m$ von AA^\top heißen **Linkssingulärvektoren** von A . Es gilt

$$A^\top A u_j = \lambda_j u_j \quad \text{und} \quad A A^\top v_j = \lambda_j v_j \quad j = 1, \dots, r.$$

Mit den orthogonalen Matrizen $U = (u_j)_{j=1}^n \in \mathbb{R}^{n \times n}$ und $V = (v_j)_{j=1}^m \in \mathbb{R}^{m \times m}$ und der Matrix

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \ddots & & & \\ & & & \sigma_r & & \\ & & & & & \\ & & & & & 0 \\ & & & & & 0 \end{pmatrix} \in \mathbb{R}^{m \times n}$$

gilt dann

$$A = U \Sigma V^\top.$$

A.2 Matrix-Vektor-Multiplikation

Innerhalb des Multigrid-Verfahrens (Kapitel 8.5) wird bei der Relaxation die Multiplikation mit der Systemmatrix erforderlich. Da deren erster Summand durch eine Diagonalmatrix gegeben ist, ist für diesen Teil eine schnelle und effiziente Multiplikation gegeben. Für den zweiten Summanden nutzen wir in der Implementierung eine Funktion, die die Wirkung der Matrix B (resp. B^\top) auf einen gegebenen Vektor beschreibt, ohne dass die Matrix B explizit im Speicher gehalten werden muss.

Die Wirkungsweise einer solchen Funktion erklären wir exemplarisch für die Matrix B^{diff} und um uns kurz zu fassen nur für ihren ersten Block und für den Fall zweier Raumdimensionen. Sei dazu

$$B^{\text{diff}} = \begin{pmatrix} D_{11} & & & \\ D_{21} & & & \\ & & D_{12} & \\ & & & D_{22} \end{pmatrix},$$

vergleiche Kapitel 8 und sei dann

$$B' = \begin{pmatrix} D_{11} \\ D_{12} \end{pmatrix}.$$

Die Multiplikation von B' mit einem gegebenen Vektor $x \in \mathbb{R}^{(m_1+1)m_2}$, vergleiche 8.3, lässt sich dann in MATLAB schreiben als:

```
n1 = m(1)*m(2);    ns = (m(1)+1)*m(2);
n2 = (m(1)+1)*(m(2)-1);
h   = omega./m;
yStg = reshape(yStg(1:ns),m(1)+1,m(2));
```

```
By = zeros(n1+n2,1);
```

```
By(1:n1) = reshape(yStg(2:end,:)-yStg(1:end-1,:),n1,1)/h(1);
By(n1+1:n2) = reshape(yStg(:,2:end)-yStg(:,1:end-1),n2,1)/h(2);
```

Für die Multiplikation mit der Transponierten ergibt sich dann:

```
% n1, n2, ns wie oben
BTy = zeros(ns,1);
ytmp = reshape(y(1:n1),m);
BTY = reshape([-ytmp(1,:);...
              ytmp(1:end-1,:)-ytmp(2:end,:);ytmp(end,:)],ns,1)./h(1);

ytmp = reshape(y(n1+1:n2),m(1)+1,m(2)-1);
BTY = BTY + reshape([-ytmp(:,1);...
                    ytmp(:,1:end-1)-ytmp(:,2:end);ytmp(:,end)],ns,1)./h(2);
```

A.3 Übersicht zu Trackingmethoden

Die während der Operation eingesetzten Instrumente werden getrackt, d.h. ihre Position im Operationsfeld wird bestimmt.

Für das Tracking der Instrumente gibt es derzeit optische und elektromagnetische Verfahren. Bei Ersten werden die zu trackenden Instrumente mit Adaptern versehen. Deren Lage im Raum wird mittels einem stereotaktischen Kamerasystem erfasst. Wichtig hierfür ist die Sichtbarkeit der Adapter im Blickfeld der Kamera.

Beim elektromagnetischen Tracking werden Spulen als Adapter verwendet. Durch einen Generator wird ein elektromagnetisches Feld aufgebaut und die Feldveränderungen, die durch das Bewegen der Spulen in diesem entstehen, erlauben eine Lagebestimmung der Adapter.

Die Angabe der getrackten Position erfolgt relativ zu einem vom Trackingssystem vorgegebenen Koordinatensystem, dem Weltkoordinatensystem. Die Lage des Adapters wird durch Angabe eines Translationsvektors $t \in \mathbb{R}^3$ und einer Rotationsmatrix $R \in \mathbb{R}^{3 \times 3}$, $R^\top R = I_3$ angegeben. Daraus wird mit Hilfe der homogenen Matrix

$$H = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

ein neues lokales Koordinatensystem bestimmt. Es besteht zwischen den Weltkoordinaten eines Punktes im Raum x^{Welt} und seinen lokalen Koordinaten x^{lokal} der Zusammenhang

$$\begin{pmatrix} x^{\text{Welt}} \\ 1 \end{pmatrix} = H \cdot \begin{pmatrix} x^{\text{lokal}} \\ 1 \end{pmatrix},$$

vergleiche dazu Abbildung A.1.

Das Ergebnis des Trackings ist eine Folge von Matrizen H . Jedes Element dieser Folge korrespondiert mit einer Positionsbestimmung zu einem gegebenen Zeitpunkt.

Um Instrumente zu tracken, müssen diese kalibriert sein, das heißt, es muss ein Bezug zwischen getrackten Koordinaten und der Koordinate des Arbeitspunktes geschaffen werden. Das sind Punkte, die für den Anwender von Interesse sind, an denen aber kein Adapter befestigt werden kann; typische Punkte sind die Spitzen von Dissektoren, oder der Ultraschallkopf. Dieser Bezug wird in Form einer homogenen Matrix H_K (K wie Kalibrierung) instrumentenspezifisch abgebildet. Der Zusammenhang zwischen Weltkoordinatensystem des Trackings und Koordinaten im lokalen System ergibt sich aus der Hintereinanderausführung von Tracking und Kalibrierung:

$$\begin{pmatrix} x^{\text{Welt}} \\ 1 \end{pmatrix} = H \cdot H_K \cdot \begin{pmatrix} x^{\text{lokal}} \\ 1 \end{pmatrix},$$

wie in Abbildung A.1 illustriert. Einen Überblick gibt [53].

Nachteil elektromagnetischen Trackings beim laparoskopischen Operieren ist die Störung des Magnetfeldes durch das Einbringen metallischer Operationsinstrumente. Fehlerbestimmungen finden sich bei [52].

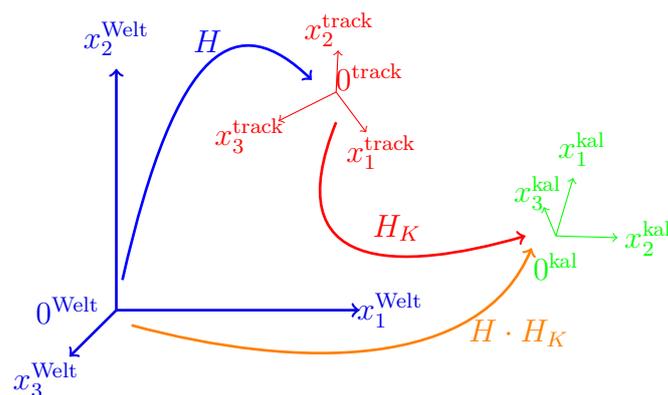


Abbildung A.1: Tracking; *blau*: Weltkoordinatensystem; *rot*: gemessenes Koordinatensystem; *grün*: kalibriertes Koordinatensystem

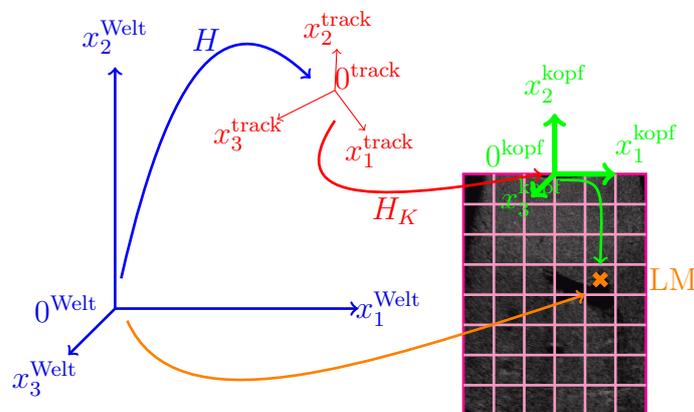


Abbildung A.2: Bestimmen der Landmarkenkoordinaten im Weltkoordinatensystem; die intra-operative Landmarke ist in Pixelkoordinaten gegeben. Mit Hilfe der Auflösung des Ultraschallbildes lassen sich ihre Koordinaten im System der Ultraschall-Sonde festlegen. Festgelegt ist, dass die Ultraschallebene in die x_1x_2 -Ebene eingebettet ist. Durch Kalibrierung und Trackinginformationen werden die Weltkoordinaten bestimmt.

Literaturverzeichnis

- [1] E.K. Abdalla, C.C. Barnett, and D. Doherty. Extended hepatectomy in patients with hepatobiliary malignancies with and without preoperative portal vein embolization. *Arch Surg*, 137, 2002.
- [2] D.V. Amin, T. Kanade, A. M. Digionia, and B. Jaramaz. Ultrasound registration of bone surface for surgical navigation. *Comput Aided Surg*, (8):1–16, 2003.
- [3] S.R. Aylward, J. Jomier, S. Weeks, and E. Bullitt. Registration and analysis of vascular images. *Int J Comput Vision*, 55(2-3):123–138, 2003.
- [4] E.A. Bakalakos, J.A. Kim, D.C. Young, and et al. Determinants of survival following hepatic resection for metastatic colorectal cancer. *World J Surg* 1998; 22:399-404. *World J Surg*, 22:399–404, 1998.
- [5] S. Beller, M. Hünnerbein, T. Lange, S. Eulenstein, B. Gebauer, and P.M. Schlag. Image-guided surgery of liver metastases by 3d ultrasound - based optoelectronic navigation. *Brit J Surg*, 94(7):866–875, 2007.
- [6] H. Bläker, W. J. Hofmann, D. Theuer, and H. F. Otto. Pathohistological findings in liver metastases. *Radiologe*, 41(1):1–7, Jan 2001.
- [7] F.L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 11(6):567–585, 1989.
- [8] W.L. Briggs, E.v. Henson, and S.F. McGeormick. *A Multigrid Tutorial*. SIAM, 2000.
- [9] C. Broit. *Optimal registration of deformed images*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1981.
- [10] L.G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

-
- [11] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, volume 3024 of *LNCS*, Berlin, 2004. Springer.
 - [12] K. Bühler, P. Felkel, and A. La Cruz. Geometric methods for vessel visualization and quantification- a survey. In G. Brunett, B. Hamman, and H. Müller, editors, *Geometric Modelling for Scientific Visualization*. Springer, 2003.
 - [13] S.E. Burpee, M. Kurian, Y. Murakame, S. Benevides, and M. Gagner. The metabolic and immune response to laparoscopic vs open liver resection. *Surg Endosc*, 16:899–904, 2002.
 - [14] I. Dagher and D. Franco. *Controversies in Laparoscopic Surgery*, chapter Commentary to Liver Surgery for Solid Tumors. Springer, 2006.
 - [15] M.H. Davis, A. Khotanzad, D.P. Flaming, and S.E. Harms. A physics-based coordinate transformation for 3-d image matching. *IEEE Trans Med Imaging*, 16(3):317–328, 1997.
 - [16] C. de Boor. *A Practical Guide to Splines*. Springer, 1978.
 - [17] C. Doppler. *Über das farbige Licht der Doppelsterne und einiger anderer Gestirne des Himmels*. Prag: Böhmische Gesellschaft der Wissenschaften, 1842.
 - [18] D. Drenckhahn and W. Zenker, editors. *Benninghoff - Anatomie*. Urban & Schwarzenberg, München, 1994.
 - [19] B. Fischer and J. Modersitzki. A super fast registration algorithm. In H Handels, A Horsch, TM Lehmann, and HPH Meinzer, editors, *Bildverarbeitung für die Medizin 2001*. Springer, 2001.
 - [20] B. Fischer and J. Modersitzki. Curvature based registration with applications to mr-mammography. In PMA Sloot, CJK Tan, JJ Dongarra, and AG Hoekstra, editors, *Computational Science - ICCS*. Springer, LNCS 2331, 2002.
 - [21] B. Fischer and J. Modersitzki. Fast diffusion registration. In M. Nasehd and O. Scherzer, editors, *Inverse Problems, Image Analysis and Medical Imaging*, volume 313 of *Contemporaray Mathematics*. AMS, 2002.
 - [22] B. Fischer and J. Modersitzki. Combination of automatic non-rigid and landmark based registration: the best of both worlds. In M Sonka and JM Fitzpatrick, editors, *Medical Imaging 2003: Image Processing*, volume 5032 of *Proceedings of the SPIE 5032*, 2003.

-
- [23] B. Fischer and J. Modersitzki. Combining landmarks and intensity driven registrations. In *PAMM Proceedings in Applied Mathematics and Mechanics*, volume 3, pages 32–35, 2003.
- [24] B. Fischer and J. Modersitzki. Curvature based image registration. *JMIV*, 18(1), 2003.
- [25] B. Fischer and J. Modersitzki. Flirt: A flexible image registration toolbox. In JC Gee, JB Maintz, and MW Vannier, editors, *2nd International Workshop on Biomedical Image Registration 2003*, volume 2717, Berlin Germany, 2003. Berlin Germany, Springer.
- [26] Y. Fong, J. Fortner, R.L. Sun, and et al. Clinical score for predicting recurrence after hepatic resection for metastatic colorectal cancer: analysis of 1001 consecutive cases. *Ann Surg*, 230:309–318, 1999.
- [27] P. E. Gill, W. Murray, and M. H. Wright. *Practical Information*. Academic Press, 1981.
- [28] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, third edition edition, 1996.
- [29] E. Haber and J. Modersitzki. A multigrid method for image registration. Technical Report TR-2004-005-A, Department of Mathematics and Computer Science, Emory University, Atlanta, 2004.
- [30] E. Haber and J. Modersitzki. Beyond mutual information: A simple and robust alternative. In H.P. Meinzer, H. Handels, A. Horsch, and T. Tolxdorff, editors, *Bildverarbeitung für die Medizin 2005*. Springer, 2005.
- [31] E. Haber and J. Modersitzki. Intensity gradient based registration and fusion of multi-modal images. *Methods of Information in Medicine*, 46(3):292–299, 2007.
- [32] H. Handels. *Medizinische Bildverarbeitung*. Teubner, 1999.
- [33] S. Heldmann. *Non-Linear Registration Based on Mutual Information*. Logos Berlin, 2007.
- [34] J. Heyder. Optimierungsansatz zur simultanen registrierung von fmrt-bildern mit volumenerhaltung. Master’s thesis, Institut für Mathematik, Universität zu Lübeck, 2006.
- [35] D. Hill, P. Batchelor, M. Holden, and D. Hawkes. Medical image registration. *Phys. Med. Biol.*, 2001.
- [36] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

- [37] S. Itoh, M. Ikeda, M. Achiwa, H. Satake, S. Iwano, and T. Ishigaki. Late-arterial and portal-venous phase imaging of the liver with a multislice ct scanner in patients without circulatory disturbances: automatic bolus tracking or empirical scan delay? *Eur Radiol*, 14:1665–1673, 2004.
- [38] B. Jähne. *Digitale Bildverarbeitung*. Springer, Berlin, 2005.
- [39] M. John, Y. Sun, S.Kadoury, W. Wein, Y. Li, J. Resnick, G. Plambeck, A. Dempsey, A. Al-Ahmad, R. Fahrig, and F. Sauer. Fusion of intracardiac ultrasound with 3d cardiac c-arm ct from animal data for electrophysiology. In *Bildverarbeitung für die Medizin 2007*, 2007.
- [40] S. Kabus. *Multiple-Material Variational Image Registration*. PhD thesis, Institut für Mathematik, Universität zu Lübeck, 2007.
- [41] J. Kohlrausch, K. Rohr, and H.S. Stiehl. A new class of elastic body splines for nonrigid registration of medical images. *Journal of Mathematical Imaging and Vision*, 23:253–280, 2005.
- [42] H. Lang. Technik der Leberresektion - Teil i. *Chirurg*, 78(8):761–774, 2007.
- [43] H. Lang, A. Radtke, M. Hindennach, T. Schroeder, N. R. Fruhauf, M. Malago, H. Bourquain, H. O. Peitgen, K. J. Oldhafer, and C. E. Broelsch. Impact of virtual tumor resection and computer-assisted risk analysis on operation planning and intraoperative strategy in major hepatic resection. *Arch. Surg*, 140(7):629–638, 2005.
- [44] T. Lange, S. Eulenstein, M. Hünerbein, H. Lamecker, and P.M. Schlag. Augmenting intraoperative 3d ultrasound with preoperative models for navigation in liver surgery. In C. Barillot, DR Haynor, and P Hellier, editors, *Medical Image Computing and Computer-Assisted Intervention*, volume 3217 of *LNCIS*. Springer, 2004.
- [45] T. Lange, S. Eulenstein, M. Hünerbein, and P.M. Schlag. Vessel-based non-rigid registration of MR/CT and 3d ultrasound for navigation in liver surgery. *Comput Aided Surg*, 8(5):228–240, 2003.
- [46] T. Lange, H. Lamecker, M. Hünerbein, S. Eulenstein, S. Beller, and P.M. Schlag. A distance measure for non-rigid registration of geometrical models to intensity data. In H Lemke and et al., editors, *CARS*, volume 2 (Supple 1) of *International Journal of Computer Assisted Radiology and Surgery*. Springer, 2007.
- [47] T. Lange, N. Papenberg, S. Heldmann, J. Modersitzk, B. Fischer, and P.M. Schlag. 3D Ultrasound–CT Alignment of the Liver using Combined Landmark–Intensity Registration. In *Internationnal Conference on Computer Assisted Radiology and Surgery 2008 (CARS '08)*, 2008, to appear.

- [48] T. Livraghi, F. Meloni, M. di Stasi, E. Rolle, L. Solbiati, C. Tinelli, and S. Rossi. Sustained complete response and complications rates after radio-frequency ablation of very early hepatocellular carcinoma in cirrhosis: Is resection still the treatment of choice? *Hepatology*, 47(1):82–89, Jan 2008.
- [49] J. M. Llovet. Updated treatment approach to hepatocellular carcinoma. *J Gastroenterol*, 40(3):225–235, Mar 2005.
- [50] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of Imaging understanding workshop*, 1981.
- [51] J. B. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 1998.
- [52] V. Martens, M. Kleemann, L. Matthäus, H.-P. Bruch, and A. Schweikard. Evaluation of em tracking systems for laparoscopic liver surgery. In *Proc of. CARS 2006*. Springer, 2006.
- [53] L. Mercier, T. Langø, F. Lindseth, and D. Collins. A review of calibration techniques for freehand 3-d ultrasound systems 2005. l. mercier, t. langø, f. lindseth, and d. collins. a review of calibration techniques for freehand 3-d ultrasound systems. *Ultrasound in Medicine Biology*, 31, 2005.
- [54] J. Modersitzki. *Numerical Methods for Image Registration*. Numerical Mathematics and Scientific Computation. Oxford University Press, 2004.
- [55] J. Modersitzki and B. Fischer. Optimal image registration with a guaranteed one-to-one point match. In T. Wittenberg et al., editor, *Bildverarbeitung für die Medizin 2003*. Springer, 2003.
- [56] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, Berlin, 2006.
- [57] G. Opfer. *Numerische Mathematik für Anfänger*. Vieweg, Braunschweig, 2002.
- [58] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analalys*, 12, 1975.
- [59] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal Computer Vision*, 67(2), April 2006.
- [60] N. Papenberg, T. Lange, J. Modersitzki, P.M. Schlag, and B. Fischer. Image registration for CT and intra-operative ultrasound data of the liver. In *SPIE Medical Imaging: Visualization, Image-guided Procedures, and Modeling*, volume 6918, 2008.

- [61] N. Papenberg, J. Modersitzki, and B. Fischer. Registrierung im Fokus. In *Bildverarbeitung für die Medizin 2008*, 2008.
- [62] 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. *Informatik Aktuell*, 2007.
- [63] G. P. Penney, J. M. Blackall, M. S. Hamady, T. Sabharwal, A. Adam, and D. J. Hawkes. Registration of freehand 3d ultrasound and magnetic resonance liver images. *Med. Image Anal.*, 8(1):81–91, 2004.
- [64] J. P. W. Pluim and J. M. Fitzpatrick. Image registration. *IEEE Trans. Med. Imag.*, 2003.
- [65] B.C. Porter, D.J. Rubens, J.G. Strang, J. Smith, S. Totterman, and K.J. Parker. Three-dimensional registration and fusion of ultrasound and mri using major vessels as fiducial markers. *IEEE Trans Med Imaging*, 20(4):354–359, 2001.
- [66] I. Reinertsen, M. Descoteaux, K. Siddiqi, and D.L. Collins. Validation of vessel-based registration for correction of brain shift. *Medical Image Analysis*, 11(4):374–388, 2007.
- [67] A. Roche, X. Pennec, G. Malandain, and N. Ayache. Rigid registration of 3-d ultrasound with mr images: A new approach combining intensity and gradient information. *IEEE Trans Med Imaging*, 20(10):1038–1049, 2001.
- [68] R. Rohling. *3D Freehand Ultrasound: Reconstruction and Spatial Compounding*. PhD thesis, University of Cambridge, Department of Engineering, 1998.
- [69] K. Rohr. *Landmark-Based Image Analysis*, volume 21 of *Computational Imaging and Vision*. Springer, 2001.
- [70] K. Rohr, H. Stiehl, R. Sprengel, T. Buzug, J. Weese, and M. Kuhn. Landmark-based elastic registration using approximating thin-plate splines. *IEEE Trans Med Imag*, 20(6):526–534, 2001.
- [71] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 1992.
- [72] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Non-rigid registration using free-form deformations: Application to breast mr images. *IEEE Transactions on Medical Imaging*, 18(8), 1999.
- [73] D. Salomon. *Computer Graphics and Geometric Modeling*. Springer, 1999.

- [74] H. Schwarz. *Numerische Mathematik*. Teubner, 1997.
- [75] D. Selle, B. Preim, A. Schenk, and H. O. Peitgen. Analysis of vasculature for liver surgical planning. *IEEE Trans Med Imaging*, 21(11):1344–1357, 2002.
- [76] K. Shirabe, M. Shimada, T. Gion, and et al. Postoperative liver failure after major hepatic resection for hepatocellular carcinoma in the modern era with special reference to remnant liver volume. *J Am Coll Surg*, 188:304–309, 1999.
- [77] M. Shoup, M. Gonen, M. D’Angelica, and et al. Volumetric analysis predicts hepatic dysfunction in patients undergoing major liver resection. *J Gastrointest Surg*, 7:325–330, 2003.
- [78] C. Simillis, V. A. Constantinides, P.P. Tekkis, A. Darzi, R. Lovegrove, L. Jiao, and A. Antoniou. Laparoscopic versus open hepatic resections for benign and malignant neoplasms—a meta-analysis. *Surgery*, 141(2):203–211, Feb 2007.
- [79] R. D. Skeel and J. B. Keiper. *Elementary Numerical Analysis*. McGraw-Hill, New-York, 1972.
- [80] P. J. Slomka, J. Mandel, D. Downey, and A. Fenster. Evaluation of voxel-based registration of 3-d power doppler ultrasound and 3-d magnetic resonance angiographic images of carotid arteries. *Ultrasound Med. Biol.*, 27(7):945–955, 2001.
- [81] U. Trottenberg, C. W. Osterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.
- [82] J.N. Vauthey, A. Chaoui, K.A. Do, and et al. Standardized measurement of the future liver remnant prior to extended liver resection: methodology and clinical associations. *Surgery*, 127:512–519, 2000.
- [83] P. A. Viola. *Alignment by Maximization of Mutual Information*. PhD thesis, Massachusetts Institute of Technology, 1995.
- [84] P. A. Viola and W. Wells. Alignment by maximization of mutual information. *International Journal Computer Vision*, 1997.
- [85] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, 1998.
- [86] W. Wein, A. Khamene, D. Clevert, O. Kutter, and N. Navab. Simulation and fully automatic multimodal registration of medical ultrasound. In *MICCAI 2007 Proceedings*, Lecture Notes in Computer Science. Springer, October 2007.

- [87] H.-F. Weiser and M. BIRTH. *Viszeralchirurgische Sonographie*. Springer, 2000.
- [88] S. Winter, B. Brendel, I. Pechlivanis, and K. Schmieder. Registrierung verschiedener Knochenstrukturen in Ultraschall- und CT-Daten anhand von prä- und intraoperativen Patientendatensätzen. In *Bildverarbeitung für die Medizin 2006*, 2006.
- [89] S. Wörz and K. Rohr. Physics-based elastic image registration using splines and including landmark localization uncertainties. In *MICCAI (2)*, pages 678–685, 2006.