



UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS AND
IMAGE COMPUTING

Symmetrische Registrierung medizinischer Bilder

Symmetric Registration of Medical Images

Bachelorarbeit

im Rahmen des Studiengangs
Computational Life Science
der Universität zu Lübeck

vorgelegt von
Annemarie Hoff

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

mit Unterstützung von
Dr. Lars Ruthotto
Institute of Mathematics and Image Computing

Lübeck, den 01. Februar 2013



IM FOCUS DAS LEBEN

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt und durch meine Unterschrift, dass ich diese Bachelorarbeit selbstständig verfasst und noch nicht anderweitig zu Prüfungszwecken vorgelegt habe. Für die Anfertigung der Bachelorarbeit wurden von mir nur die angegebenen Quellen und Hilfsmittel verwendet.

Lübeck, den 01.02.2013

Danksagung

Ich danke Prof. Dr. rer. nat. Jan Modersitzki für die freundliche Aufnahme und die gute Betreuung im Institute of Mathematics and Image Computing der Universität zu Lübeck. Ganz besonderer Dank gilt Dr. Lars Ruthotto, der mich geduldige betreute und mir viele hilfreiche Hinweise gab.

Daneben gilt der Dank meinen Freunden und meiner Familie, die mich tatkräftig seelisch und moralisch unterstützen. Allen voran meinem Mann, meiner Mutter und Christian Garske, die zusätzlich viele Stunden in das Korrekturlesen dieser Arbeit investierten und aufzeigten, wo noch Erklärungsbedarf bestand.

Zusammenfassung

Ein Prozess, der zunehmend an Bedeutung gewinnt, ist die Bildverarbeitung. Vor allem in der medizinischen Diagnostik werden mehr und mehr Bilddaten ausgewertet. Zum Beispiel für Bildfusion oder auch zur Kontrolle des Behandlungserfolgs ist es dabei wichtig, Informationen aus der Kombination mehrerer Bilder zu gewinnen. Grundlegend dafür ist der Prozess der Bildregistrierung, bei dem eine Transformation gesucht wird, die das sogenannte Template so gut wie möglich in die Referenz, die unverändert bleibt, überführt.

Dabei wird in der Regel davon ausgegangen, dass die Korrespondenzen zwischen den Punkten aus Template und Referenz durch die Transformation eindeutig definiert sind. Dies ist meistens jedoch nicht der Fall, da die ermittelten Transformationen nicht bijektiv sind. Deshalb ist auch eine Invertierung des Transformationsfeldes nicht immer möglich. Eine Option, dieses Problem zu umgehen, stellen symmetrische Registrierungsansätze dar. Dabei wird nicht ein Bild fest als Referenz bestimmt, sondern es wird gleichzeitig für zwei zu vergleichenden Bilder jeweils eine Transformation gesucht, die das Bild möglichst gut an das jeweils andere Bild annähert.

In dieser Arbeit werden zunächst Grundlagen der Bildregistrierung nach FAIR erläutert. Dabei werden ein kontinuierliches Bildmodell, verschiedene Transformationen und Interpolationen sowie Distanzmaße und Regularisierer genauer beleuchtet. Im Anschluss wird ein Modell der symmetrischen Registrierung nach G.E. Christensen und H.J. Johnson vorgestellt. Dieses Modell wurde im Zuge dieser Arbeit in einen nichtparametrischen Ansatz überführt und mit Hilfe der FAIR-Software in Matlab implementiert.

Getestet wurde der Algorithmus mit einem anschaulichen Registrierungsproblem, in dem Bilder zweier Hände verglichen werden. Es wurde mit der Summe der quadratischen Differenzen als Distanzmaß und elastischer Regularisierung gearbeitet. Als Vergleichsbasis wurde auf den gleichen Daten eine Registrierung in nur eine Richtung vorgenommen. Wie erwartet ergaben sich mit dem symmetrischen Algorithmus niedrigere Werte in der Abweichung der Transformationen voneinander, allerdings ist der Rechenaufwand und da mit auch die Rechenzeit deutlich höher und die Distanz der Bilder wird durch die Transformation nicht so gut minimiert wie bei der Vergleichsrechnung, bei der nicht auf inverse Konsistenz geachtet wurde.

Inhaltsverzeichnis

1	Einleitung	2
1.1	Motivation	2
1.2	Problem der inversen Konsistenz	4
1.3	Ziel und Gliederung der Arbeit	5
2	Mathematische Grundlagen der Bildregistrierung	6
2.1	Interpolation	7
2.2	Transformationen	8
2.3	Distanzmaße	10
2.4	Regularisierer	11
2.5	Bildregistrierung	12
3	Symmetrische Bildregistrierung	15
3.1	Kontinuierliche Modellierung des Problems	15
3.2	Invertierung von Deformationsfeldern	17
3.3	Vor- und Nachteile des Verfahrens	19
4	Implementierung des Algorithmus	21
4.1	Affin-lineare Vorregistrierung	22
4.2	Umsetzung in FAIR	23
4.3	Ergebnis der Vorregistrierung	24
4.4	Separate Registrierung in beide Richtungen	25
4.5	Symmetrische Registrierung	28
4.6	Vergleich	29
5	Fazit	31

Abbildungsverzeichnis

1.1	Registrierungsbeispiel: Histologische Serienschritte	3
1.2	Beispiel für Mehrdeutigkeit: Quadrat	4
2.1	Lineare und Spline-Interpolation	7
2.2	Registrierungsbeispiel: Hände	11
3.1	Funktionsweise des Invertieralgorithmus	18
4.1	Nodales Gitter	22
4.2	Ergebnisse der affin-linearen Registrierung.	25
4.3	Ergebnis der Vorwärtstransformation	26
4.4	Ergebnis der Rückwärtstransformation.	27
4.5	Ergebnisse der symmetrischen Registrierung	29

1 Einleitung

Durch die zunehmende Digitalisierung und Technisierung unserer Umwelt sind wir mehr und mehr von Bildern umgeben. Manche dienen dabei nur der Unterhaltung, andere enthalten jedoch wichtige Informationen. Mit dieser Entwicklung steigt auch die Bedeutung der Bildverarbeitung. Zum Beispiel möchte man häufig Bilder miteinander vergleichen, um sie aneinanderzufügen oder besser auswerten zu können.

Von großer Bedeutung hierfür ist die Bildregistrierung. Im Rahmen dieser Arbeit wird speziell die symmetrische Registrierung intensiver beleuchtet. Ein entsprechender Algorithmus wurde mit Hilfe der FAIR Software [14] in Matlab implementiert.

In diesem Kapitel wird nun zunächst allgemein die Notwendigkeit der Bildregistrierung sowie die Gründe für die Nutzung eines symmetrischen Verfahrens eingegangen. Es folgt ein Ausblick auf den weiteren Aufbau der Arbeit.

1.1 Motivation

In vielen Lebensbereichen sind digitale Medien von sehr großer Bedeutung. Besonders in der medizinischen Diagnostik gibt es große Entwicklungen im Bereich der bildgebenden Verfahren, die es uns ermöglichen, einen Einblick in den menschlichen Körper zu erhalten. Dabei werden Informationen meist nicht nur aus einem einzelnen Bild gewonnen, sondern aus der Kombination mehrerer Bilder. So werden zum Beispiel bei der Bildfusion Aufnahmen unterschiedlicher Modalitäten miteinander kombiniert um ein komplexeres Bild des Körpers eines Patienten zu gewinnen. Dadurch können die jeweiligen Vorteile der einzelnen Modalitäten effektiver genutzt werden. Es kann aber auch ein Vergleich zweier Bilder viele Informationen liefern, zum Beispiel um den Erfolg einer Behandlung zu kontrollieren oder das Wachstum einer Tumors zu überwachen.

Grundlegendes Werkzeug für diese Verfahren, in denen Informationen aus der

Kombination mehrerer Bilder gewonnen werden, ist die Bildregistrierung. Dabei wird eine Transformation gesucht, die voneinander verschiedene Bilder so gut wie möglich zur Übereinstimmung bringt. Die Bilder können dafür zu verschiedenen Zeitpunkten, aus verschiedenen Perspektiven oder mit Hilfe unterschiedlicher Geräte entstanden sein. Dieser Prozess ermöglicht einen geometrischen Vergleich zweier Bilder, weil dadurch beispielsweise Verzerrungen aus den Aufnahmen entfernt werden können. Dabei wird meist ein Bild als Referenz festgelegt und bleibt unverändert. Das heißt, die Transformation wird so gesucht, dass ein anderes Bild an die Referenz angepasst wird (vergleiche [17]).

Ein weiteres Anwendungsbeispiel der Bildregistrierung zeigt Abbildung 1.1. Dabei soll aus histologischen Serienschritten eines Hirns ein dreidimensionales, digitales Modell erstellt werden. Ein Hirn wird also in dünne Scheiben geschnitten, von denen jeweils Aufnahmen gemacht werden. Aus diesen zweidimensionalen Daten soll dann ein dreidimensionales Modell gewonnen werden. Dazu müssen die einzelnen Schichten so gut wie möglich zur Deckung gebracht werden. Durch die Schnitte kommt es zu Verformungen, die mit Hilfe einer geeigneten Transformation ausgeglichen werden müssen.

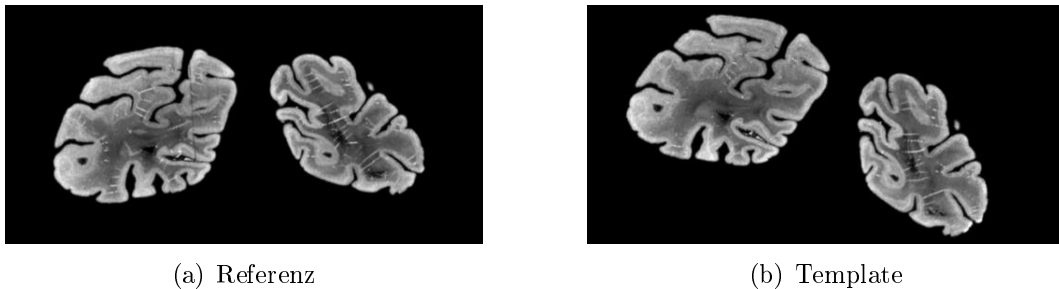


Abbildung 1.1: Histologische Serienschritte aus dem Human Neuro Scanning Project; zur Verfügung gestellt von O. Schmitt, Institut für Anatomie, Universität Rostock [16].

Praktisch gestaltet sich die Suche nach einer solchen Transformation schwierig, da es meist mehrere mögliche Lösungen gibt. Abbildung 1.2 zeigt ein solches Beispiel, bei dem schon auf den ersten Blick mehrere Lösungen gefunden werden können. Dieses Problem ist lösbar durch Rotation um den Bildmittelpunkt, Spiegelung am Bildmittelpunkt, Verschiebung oder auch komplexere, nichtlineare Transformationen.

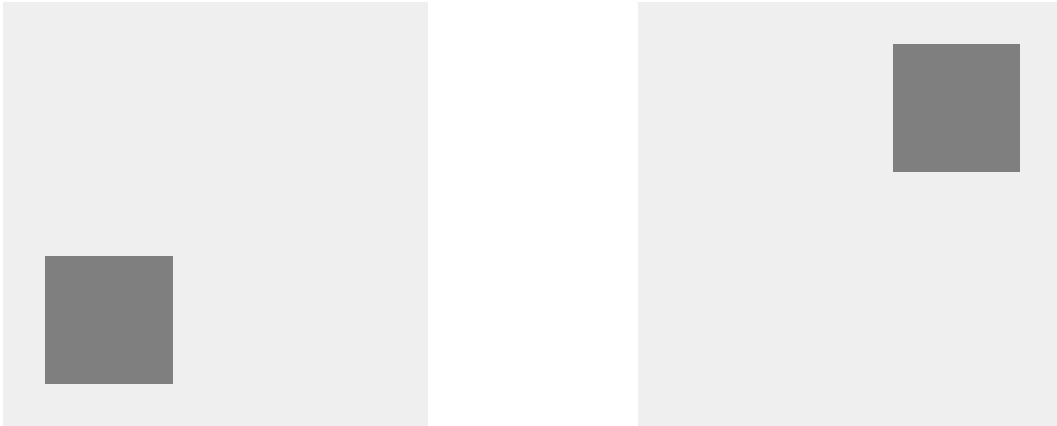


Abbildung 1.2: Beispiel für ein Registrierungsproblem mit mehreren, leicht ersichtlichen Lösungen.

Hier gilt es nun abzuwägen, welche Transformation am geeignetsten ist. Dies entscheidet sich meist nur in Abhängigkeit der jeweiligen Anwendung und kann nicht pauschal formuliert werden.

Es gibt zahlreiche verschiedene Methoden, Registrierungsprobleme zu lösen, die sich je nach Anwendungsgebiet unterschiedlich entwickelt haben (vergleiche [3], [8], [10], [11], [17]). Nicht jedes dieser Verfahren ist für jeden Einsatzbereich geeignet, deshalb gibt es kein allgemein bestes Verfahren zur Lösung solcher Probleme. Außerdem muss wie bei den meisten computerbasierten Berechnungen eine gute Balance zwischen Rechenaufwand und -zeit und der Genauigkeit der Berechnung gefunden werden.

1.2 Problem der inversen Konsistenz

In vielen Fällen wird vorausgesetzt, dass durch die ermittelte Transformation eindeutige Zugehörigkeiten zwischen den Punkten der verglichenen Bilder geschaffen werden. Dies ist jedoch bei nichtlinearen Transformationen häufig nicht der Fall, da diese meist nicht bijektiv sind. Aus der nicht gesicherten

Bijektivität, ergibt sich außerdem das Problem der Invertierbarkeit der Transformationen. Ist Invertierbarkeit gegeben, so kann das Transformationsfeld approximativ invertiert werden. Dies kann zum Beispiel mittels eines Fixpunktansatzes erfolgen [5], erfordert jedoch eine nachträgliche Berechnung

Das Ergebnis der Registrierung hängt stark davon ab, welches Bild als Referenz gewählt wurde [15]. Deshalb kann man nicht davon ausgehen, dass die separate Registrierung in beide Richtungen zueinander inverse Abbildungen liefert [7].

Einen alternativen Ansatz, um eine bessere Korrespondenz zwischen den Punkten der Bilder zu erhalten, bietet die symmetrische Bildregistrierung. Hierbei wird keine eindeutige Referenz festgelegt, sondern es wird gleichzeitig für beide Bilder eine Transformation gesucht, die das jeweils andere Bild möglichst gut approximiert unter der Nebenbedingung, dass diese beiden Transformationen invers zueinander sind. Somit lässt sich leichter bestimmen, welche Punkte des einen Bildes mit denen des anderen Bildes korrespondieren. Allerdings sind hierbei die Berechnungen aufwändiger als bei herkömmlichen Verfahren, weshalb die symmetrische Registrierung nicht immer sinnvoll ist.

1.3 Ziel und Gliederung der Arbeit

Im Rahmen dieser Bachelorarbeit wird ein Algorithmus zur symmetrischen Bildregistrierung in Matlab implementiert. Dazu werden zunächst die Grundlagen der Bildregistrierung nach Modersitzki [14] sowie ein symmetrischer Registrierungsansatz nach G.E. Christensen und H.J. Johnson vorgestellt.

Kapitel 2 erläutert die grundlegenden Begriffe und Methoden der Bildregistrierung nach Modersitzki [14]. Dabei wird auf Interpolationsmodelle, Distanzmaße und Regularisierer genauer eingegangen. In Kapitel 3 folgt die kontinuierliche Betrachtung der symmetrischen Registrierung nach Christensen und Johnson [6], wobei die Zielfunktion des betrachteten Optimierungsproblems modelliert wird. Anschließend folgt in Kapitel 4 die Umsetzung der theoretischen Darstellungen der vorherigen Kapitel mit Hilfe der FAIR-Software in Matlab. Dabei werden Details der Implementierung beleuchtet und Tests des Verfahrens sowie ein Vergleich mit einem Verfahren, das lediglich in eine Richtung registriert, durchgeführt. Abschließend fasst Kapitel 5 die Ergebnisse zusammen und gibt einen Ausblick auf weitere Möglichkeiten des Algorithmus.

2 Mathematische Grundlagen der Bildregistrierung

Um die symmetrische Registrierung genauer erläutern zu können, müssen zunächst einige Grundlagen der Bildregistrierung vorgestellt werden. Im Rahmen dieser Arbeit wird dafür hauptsächlich auf die Darstellungen des Buches "FAIR: Flexible Algorithms for Image Registration" von Jan Modersitzki [14] zurückgegriffen. Sofern nicht anders erwähnt, beruhen die Darstellungen dieses Kapitels auf FAIR.

Um detaillierter auf die mathematischen Registrierungsgrundlagen eingehen zu können, muss zunächst der Begriff des Bildes definiert werden. In dieser Arbeit wird lediglich der zweidimensionale Fall betrachtet.

Definition (Bild).

Sei $\Omega \subset \mathbb{R}^2$ eine zusammenhängende offene Menge der Form $\Omega = (\omega_1, \omega_2) \times (\omega_3, \omega_4)$.

Dann ist ein *Bild* eine stetig differenzierbare Abbildung $\mathcal{I} : \Omega \rightarrow \mathbb{R}$ mit kompaktem Träger, die jedem Bildpunkt aus Ω einen Grauwert zuordnet.

Hier ist also zu unterscheiden zwischen den diskreten Daten, die mit Hilfe von Geräten erzeugt werden, und dem kontinuierlichen Begriff des Bildes. Für die kontinuierliche Modellierung müssen deshalb die diskreten Bilddaten interpoliert werden.

Die Nutzung eines kontinuierlichen Bildmodells ist eine Besonderheit in FAIR. Dies soll nicht nur die auch im Realen kontinuierlichen Verhältnisse besser darstellen, sondern hat auch praktische Bedeutung. Die meisten Transformationen führen nämlich zu derartigen Verschiebungen der Bildpunkte, dass diese nicht mehr mit den gegebenen diskreten Datenpunkten übereinstimmen. Durch die vorherige Interpolation ist das kontinuierliche Modell jedoch an allen Punkten aus Ω eindeutig definiert.

2.1 Interpolation

Um das kontinuierliche Modell umzusetzen, benötigt man Interpolationsverfahren. Dabei sollen aus den gegebenen Datenpunkten die Werte derjenigen Punkte, die zwischen diesen Gitterpunkten liegen, approximiert werden. Dafür kommen verschiedene Verfahren in Frage, von denen in diesem Kapitel ausgewählte vorgestellt werden sollen.

Zum Beispiel gibt es die Möglichkeiten, jeden Grauwert durch den Wert des jeweils nächstliegenden gegebenen Datenpunktes darzustellen. Das so entstandene Bild ist allerdings nicht stetig und dementsprechend auch nicht differenzierbar, was für die weiteren Betrachtungen nötig ist.

Eine bessere Alternative bezüglich der Glattheitseigenschaften des Bildes stellt die lineare Interpolation dar. Dabei werden benachbarte Datenpunkte mit einer Strecke verbunden und somit unbekannte Werte der Funktion, die zwischen den Datenpunkten liegen, mit Hilfe dieser Gerade approximiert (siehe Abbildung 2.1(a)). Dieses Interpolationsverfahren hat den Vorteil einer einfachen Berechnung und es entsteht ein stetiges Bild. Allerdings bleibt auch hier das Problem der Differenzierbarkeit an den Gitterpunkten.

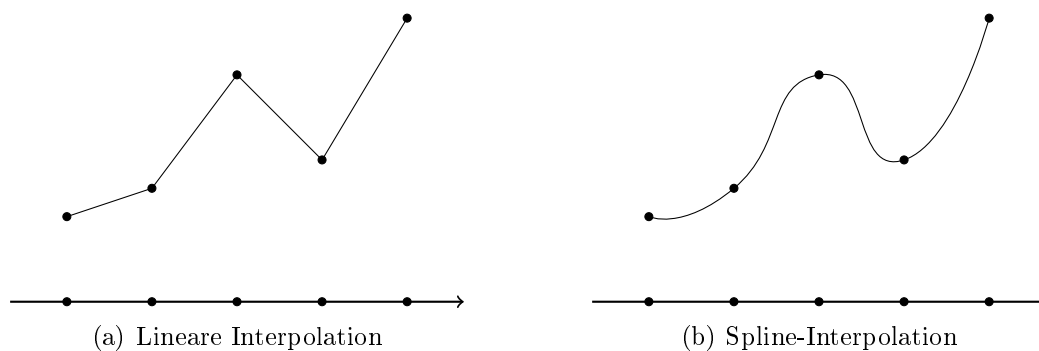


Abbildung 2.1: Grafische Darstellung zu Interpolationsverfahren: (a) Lineare Interpolation (b) Spline-Interpolation.

Um dieses Problem zu umgehen, bietet sich die Spline-Interpolation an (sie-

he Abbildung 2.1(b)) . Hierzu werden die Daten stückweise interpoliert und die Approximationsfunktionen werden als Summe von bestimmten Basisfunktionen dargestellt. Zwar steigt damit der Berechnungsaufwand, aber es lässt sich beliebige Glattheit der Funktion erreichen.

2.2 Transformationen

Bei der Suche nach einer geeigneten Registrierungstransformation muss man sich bewusst sein, welche verschiedenen Transformationen es gibt (siehe zum Beispiel [12]).

Die wohl einfachsten Transformationen sind rigide Transformationen, also solche, die nur aus Translation und Rotation bestehen. Die allgemeine Form einer rigiden Abbildung lautet

$$y = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \cdot x + t \quad \text{mit } \theta \in \mathbb{R} \text{ und } x, t \in \mathbb{R}^2.$$

Rigide Transformationen sind sowohl längenerhaltend als auch winkeltreu, sie verschieben also alle Punkte von Ω gleichermaßen.

Komplexer sind affin-lineare Transformationen, die neben Translation und Rotation auch Scherung und Skalierung vornehmen können. Hier sieht die allgemeine Form wie folgt aus:

$$y = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \cdot \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \cdot \begin{pmatrix} 1 & sh_x \\ sh_y & 1 \end{pmatrix} \cdot x + t$$

mit $\theta, S_x, S_y, sh_x, sh_y \in \mathbb{R}$ und $x, t \in \mathbb{R}^2$.

Affin-lineare Transformationen sind im Allgemeinen weder längen- noch winkeltreu. Aber sie sind geraden-, parallelen- und verhältnistreu.

Es gibt Möglichkeiten, affin-lineare Abbildungen als einfache Matrixmultiplikation zu schreiben. Dazu kann man zum Beispiel auf homogene Koordinaten zurückgreifen (siehe [13] oder [9]). Alternativ ergibt sich durch umschreiben

der allgemeinen Form affin-linearer Transformationen:

$$\begin{aligned}
 y &= A \cdot x + t \text{ mit } A = \begin{pmatrix} w_1 & w_2 \\ w_4 & w_5 \end{pmatrix} \text{ und } t = \begin{pmatrix} w_3 \\ w_6 \end{pmatrix} \\
 \implies \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} &= \begin{pmatrix} w_1 x_1 + w_2 x_2 + w_3 \\ w_4 x_1 + w_5 x_2 + w_6 \end{pmatrix} \\
 &= \begin{pmatrix} x_1 & x_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & x_2 & 1 \end{pmatrix} \cdot w \text{ mit } w = (w_1, w_2, w_3, w_4, w_5, w_6)^T
 \end{aligned}$$

und damit

$$y = Q(x) \cdot w \text{ für } Q(x) = \begin{pmatrix} x_1 & x_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & x_2 & 1 \end{pmatrix}.$$

Im weiteren Verlauf der Registrierung ist es nötig, mit der Ableitung der Transformation zu arbeiten. Diese ist bei affin-linearen Abbildungen leicht zu berechnen, da y hier eine Funktion der Parameter w_i ist. Es gilt:

$$dy(w) = Q(x).$$

Affin-linearen Funktionen ist gemein, dass sie die Grundmenge des Bildes Ω verändern. Skalierungen mit S_x oder S_y größer als eins, vergrößern den Definitionsbereich des Bildes zum Beispiel in x - beziehungsweise y -Richtung.

Praktisch besteht vor allem im medizinischen Bereich das Problem, dass affin-lineare Transformationen nicht ausreichen, um auftretende Deformationen in den Bilddaten auszugleichen [13]. Zwar verhalten sich Knochen im Allgemeinen rigide, für Gewebe gilt das allerdings nicht. Und auch bei Registrierungsproblemen, die gleiche Körperteile verschiedener Patienten vergleichen, kann man wegen der anatomischen Differenzen nicht davon ausgehen, das Problem mit affin-linearen Transformationen lösen zu können. Dafür werden nichtlineare Transformationen benötigt, die deutlich komplexer sind und viel mehr Freiheitsgrade haben. Hierbei müssen nämlich nicht alle Bildpunkte der gleichen Veränderung unterliegen, sondern die Transformation kann in jedem Bildpunkt unterschiedlich sein.

Wünschenswerte Eigenschaften einer Transformation, die ein Registrierungsproblem löst, sind wie bereits erwähnt Invertierbarkeit und Differenzierbarkeit.

keit. Deshalb werden häufig Diffeomorphismen betrachtet, da diese gut die gewünschten Anforderungen an eine Transformation erfüllen. Es folgt die Definition eines Diffeomorphismus nach [2].

Definition (Diffeomorphismus).

Eine Abbildung $f : U \rightarrow U$ mit $U \subset \mathbb{R}^n$ und offen heißt *Diffeomorphismus*, falls gilt

- f ist invertierbar
- f ist stetig differenzierbar
- f^{-1} ist stetig differenzierbar

Voraussetzung für Invertierbarkeit einer Funktion ist dabei, dass die Funktion bijektiv ist.

2.3 Distanzmaße

Für die Bildregistrierung werden stets zwei oder mehr Vergleichsbilder benötigt. Eines davon heißt Referenz und wird mit \mathcal{R} bezeichnet. Die anderen Bilder heißen Objektbilder oder Template und werden mit \mathcal{T} bezeichnet.

Nun stellt sich die Frage der Vergleichbarkeit zweier Bilder. Für den Vergleich zweier Bilder, die ähnliche Strukturen zeigen, muss als Grundlage ein Abstand der Bilder zueinander definiert werden. Dieses Distanzfunktional kann entweder als Maß für die Ähnlichkeit oder als Maß für den Unterschied der Bilder definiert werden. Im Verlauf der Registrierung soll dies so groß bzw. so klein wie möglich werden. Im Folgenden werden lediglich Minimierungsprobleme betrachtet. Es wird also eine Transformation gesucht, die den Abstand des transformierten Templates zur Referenz so gut wie möglich verkleinert. Deshalb kann Bildregistrierung als ein Optimierungsproblem des Abstandes über die Transformation betrachtet werden.

Es gibt zahlreiche verschiedene Möglichkeiten für die Wahl eines Distanzmaßes (siehe beispielsweise [14]). Ein anschauliches und recht einfaches Beispiel dafür ist die Summe der quadratischen Differenzen (Sum of Squared Differences,

SSD). Dafür gilt:

$$\mathcal{D}^{SSD}[\mathcal{T}, \mathcal{R}] = \frac{1}{2} \int_{\Omega} (\mathcal{T}(x) - \mathcal{R}(x))^2 dx. \quad (2.1)$$

Abbildung 2.2 zeigt ein Beispiel zur Berechnung und grafischen Darstellung des punktwweisen quadratischen Abstandes zweier Bilder.

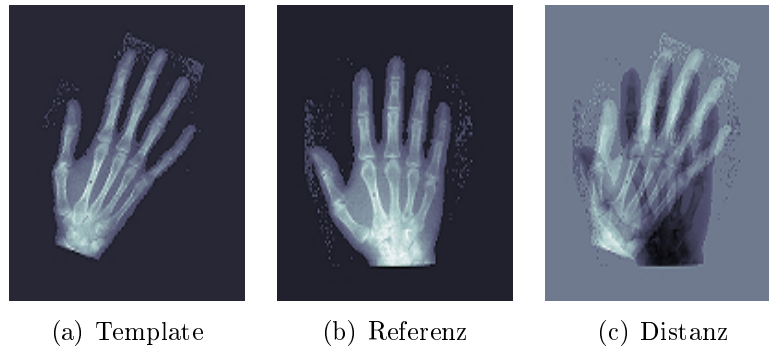


Abbildung 2.2: Röntgenaufnahmen zweier Hände, die sich in Lage und Form unterscheiden: (a) Template, (b) Referenz, (c) Distanz der Bilder nach quadratischer Abstandsberechnung.

2.4 Regularisierer

Bei der Betrachtung mathematischer Modelle ist eine wichtige Frage, ob ein Problem korrekt gestellt ist. Dabei definiert sich ein korrekt gestelltes Problem darüber, dass eine Lösung existiert, diese eindeutig ist und stetig von den Daten abhängt. Ist eine dieser Eigenschaften nicht erfüllt, spricht man von einem schlecht gestellten Problem.

Wie bereits im einleitenden Beispiel in Abbildung 1.2 gezeigt, existiert im Allgemeinen keine eindeutige Lösung für Registrierungsprobleme. Demnach ist Bildregistrierung im Allgemeinen ein schlecht gestelltes Problem.

Nun gilt es, ein Maß für die Qualität einer möglichen Lösung zu finden. Dazu wird ein Regularisierer verwendet.

Bei medizinischen Objekten kann man davon ausgehen, dass das in den Bildern dargestellte Gewebe elastischen Gesetzen folgt. Bei Kräfteinwirkungen

oder schlicht Lageveränderungen kann es also zu Verformungen kommen. Dabei verändern sich die relativen Abstände des Gewebes und damit der Bildpunkte zueinander, es kommt jedoch nicht zu einem Positionstausch der Punkte und damit einer Faltung innerhalb der Transformation. Diese Eigenschaften soll ein elastischer Regularisierer gewähren, der von Broit eingeführt wurde [1]. Dadurch sollen physikalische Deformationen nachgebildet werden und Nachbarschaftsbeziehungen erhalten bleiben [3]. Die allgemeine Form des elastischen Regularisierers lautet:

$$\mathcal{S}[y] = 1/2 \cdot \int_{\Omega} |\mathcal{B}[y]|^2 \text{ mit } \mathcal{B} = \begin{bmatrix} \sqrt{\mu} \cdot \nabla & 0 \\ 0 & \sqrt{\mu} \cdot \nabla \\ \sqrt{\lambda + \mu} \cdot \partial_1 & \sqrt{\lambda + \mu} \cdot \partial_2 \end{bmatrix} \quad (2.2)$$

Dabei bezeichnet $\nabla \cdot y = \partial_1 y_1 + \partial_2 y_2$ und λ und μ sind die sogenannten Lamé-Konstanten, also Materialkonstanten, die das Verhalten des jeweiligen Gewebes beschreiben.

Eine alternative Wahl wäre ein hyperelastischer Regularisierer. Im Gegensatz zur elastischen Regularisierung wird dabei gesichert, dass die Transformation diffeomorph ist [4].

Meist werden die Regularisierer nicht nur auf y angewendet sondern viel mehr auf die Distanz $y - y^{\text{ref}}$. Dadurch kann eine Ähnlichkeit der ermittelten Transformation y zu einer vorher gewählten Funktion y^{ref} erzeugt werden. Dadurch werden Funktionen, die weit von y^{ref} entfernt sind, mit einem großen Term für den Regularisierer bestraft. Als y^{ref} kann man zum Beispiel die Identitätsfunktion wählen.

Der Regularisierer wird zusätzlich mit einem Parameter α gewichtet. Demnach müsste genauer $\alpha \mathcal{S}[y - y^{\text{ref}}]$ statt $\mathcal{S}[y]$ geschrieben werden, aus Vereinfachungsgründen wird jedoch weiterhin $\mathcal{S}[y]$ verwendet.

2.5 Bildregistrierung

Mit diesen vorgestellten Grundlagen lässt sich nun der eigentliche Registrierungsprozess genauer erläutern.

Dabei wird eine Transformation y gesucht, die das Template \mathcal{T} so genau wie möglich in das Referenzbild \mathcal{R} überführt. Dies wird als Optimierungsproblem einer Zielfunktion dargestellt. Typischerweise werden dabei das Referenzbild

und auch die Intensitäten des Templates nicht verändert. Es werden lediglich Verschiebungen der Template-Bildpunkte vorgenommen.

Die Zielfunktion zur Berechnung der Transformation ergibt sich wie folgt:

$$\mathcal{J}[y] = \mathcal{D}[\mathcal{T}[y], \mathcal{R}] + \mathcal{S}[y] \xrightarrow{y} \min \quad (2.3)$$

wobei $\mathcal{T}[y] = \mathcal{T}(y(x))$ mit $x \in \Omega$ das transformierte Template beschreibt, \mathcal{R} das Referenzbild, \mathcal{D} das Distanzmaß und $\mathcal{S}[y]$ den Regularisierer.

FAIR arbeitet mit einem Discretize-then-Optimize-Ansatz, der sich auf numerische Optimierung stützt. Die interpolierten Bilddaten werden also erneut diskretisiert. Dabei wird allerdings nicht nur ein Diskretisierungsgrad gewählt, sondern es wird ein Multilevel-Ansatz verfolgt. Das bedeutet die Daten werden zunächst auf einem groben Gitter ausgewertet und so ein erstes approximatives y bestimmt. Im Anschluss wird das Gitter Schritt für Schritt verfeinert und die Näherung wird verbessert. So hat man für die Auswertung der Zielfunktion auf der angestrebten Gitterfeinheit einen guten Anfangswert für die Optimierung. Für die numerische Optimierung wird in FAIR standardmäßig ein Gauß-Newton-Algorithmus gewählt. Dieser kann als modifiziertes Newton-Verfahren betrachtet werden, mit dem ein nichtlineares Ausgleichsproblem gelöst werden soll. Die Optimierung nach dem Newton-Verfahren würde der folgenden Iteration folgen:

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k).$$

∇^2 bezeichnet dabei die Hesse-Matrix und ∇ den Gradienten der Zielfunktion f . Diese hat die Form

$$f(x) = \sum_{j=1}^m r_j^2(x).$$

r_j ist dabei wiederum die Residuumsfunktion, also der Abstand der Messdaten zur gesuchten Approximation dieser Messdaten.

Der Gauß-Newton-Algorithmus beruht nun darauf, dass

$$\nabla^2 f = J^T J \quad \text{mit } J \text{ ist Jacobimatrix von } f$$

approximiert wird. Damit gilt nun

$$J_k^T J_k p_k = -J_k^T r_k \text{ mit } \nabla f_k = J_k^T r_k.$$

Statt der Hesse-Matrix muss dementsprechend lediglich die Jacobi-Matrix berechnet werden, was den Rechenaufwand des Optimierungsverfahren erheblich senkt.

Dieses Kapitel stellt alle Grundlagen für die Bildregistrierung zur Verfügung, die im späteren Verlauf der Arbeit benötigt werden. Es wurden bereits das Distanzmaß, sowie der Regularisierer und Interpolationsverfahren vorgestellt, die für die Implementierung des symmetrischen Verfahrens genutzt werden.

3 Symmetrische Bildregistrierung

Wie bereits erwähnt, ist eine Erwartung an die Transformation, die Ergebnis der Bildregistrierung sein soll, dass sie eindeutige Korrespondenzen der Punkte in beiden Bildern liefert. Das ist jedoch häufig nicht der Fall, da diese Abbildungen nicht unbedingt bijektiv sind. Außerdem liefert die Registrierung unter Vertauschung von Template und Referenz nicht unbedingt eine Inverse der Transformation und die direkte Berechnung einer Inversen ist nicht immer möglich. Dieses Verfahren würde außerdem einen nachträglichen Berechnungsschritt der inversen Transformation erfordern.

Eine Idee, diese Probleme zu umgehen, ist die symmetrische Bildregistrierung. G.E. Christensen und H.J. Johnson liefern in ihrem Paper “Consistent Image Registration” aus dem Jahr 2001 [6] einen möglichen Ansatz dafür, der in diesem Kapitel genauer vorgestellt werden soll. Sofern nicht anders gekennzeichnet, stammen die folgenden Betrachtungen aus diesem Paper.

3.1 Kontinuierliche Modellierung des Problems

Bevor die Bildregistrierung durchgeführt wird, werden in der Regel zunächst Template und Referenz bestimmt. Diese Zuordnung kann allerdings auch umgetauscht werden. So ergeben sich zwei mögliche Wege der Registrierung, die bei der symmetrischen Bildregistrierung beide gleichzeitig betrachtet werden. Das heißt, hier wird nicht mehr nur eine Transformation y gesucht, die den Abstand zwischen $\mathcal{T}[y]$ und \mathcal{R} minimiert, sondern gleichzeitig soll eine Transformation z gefunden werden, die den Abstand zwischen $\mathcal{R}[z]$ und \mathcal{T} minimiert, unter der Nebenbedingung, dass $z = y^{-1}$ gilt. Die Zielfunktion 2.3 erweitert sich also auf

$$\mathcal{J}[y, z] = \mathcal{D}[\mathcal{T}[y], \mathcal{R}] + \mathcal{D}[\mathcal{R}[z], \mathcal{T}] + \mathcal{S}[y] + \mathcal{S}[z] \text{ subject to } z = y^{-1}.$$

Damit ergeben sich weitere Variablen als

$$\begin{aligned}
u(x) &:= y(x) - x \\
v(x) &:= z(x) - x \\
\tilde{u}(x) &:= y^{-1}(x) - x \\
\tilde{v}(x) &:= z^{-1}(x) - x
\end{aligned} \tag{3.1}$$

Christensen und Johnson beschränken ihre Betrachtungen lediglich auf selbst-abbildende Diffeomorphismen, das heißt, sie gehen davon aus, dass y und z invertierbar sind. Die Deformationsfelder \tilde{u} und \tilde{v} sind im Allgemeinen nicht invertierbar, deshalb erfolgt ihre Berechnung über den Umweg der Invertierung von y und z . Genauer zur iterativen Berechnung der Inversen findet sich in Abschnitt 3.2.

Es werden also nur stetig differenzierbare Transformationen y und z betrachtet, die invertierbar sind und deren Umkehrabbildungen ebenfalls stetig differenzierbar sind. Zusätzlich soll gelten $y : \Omega \rightarrow \Omega$ und $z : \Omega \rightarrow \Omega$.

Als Distanzmaß wird das in dieser Arbeit bereits erläuterte Prinzip der Summe der quadratischen Differenzen (siehe Gleichung (2.1)) verwendet. Damit ergibt sich das Gesamtdistanzmaß der symmetrischen Registrierung als

$$\mathcal{D}[\mathcal{T}[y], \mathcal{R}] + \mathcal{D}[\mathcal{R}[z], \mathcal{T}] = \int_{\Omega} |\mathcal{T}(y(x)) - \mathcal{R}(x)|^2 dx + \int_{\Omega} |\mathcal{R}(z(x)) - \mathcal{T}(x)|^2 dx.$$

Für die Verwendung dieses Distanzmaßes ist es nötig, dass die Intensitäten der beiden Bilder auf den gleichen Bereich skaliert sind.

Das Distanzmaß ist zwar symmetrisch aufgebaut und betrachtet die Registrierung in beide Richtungen gleichzeitig, dadurch wird jedoch noch nicht sichergestellt, dass $y = z^{-1}$ gilt. Dafür muss noch eine Nebenbedingung hinzugefügt werden. Diese wird hier als Penalty zur Zielfunktion hinzugefügt, der die inverse Konsistenz (Inverse Consistency Constraint, ICC) der Transformation sichern soll.

$$\begin{aligned}
\mathcal{P}^{ICC} &= \int_{\Omega} \|y(x) - z^{-1}(x)\|^2 dx + \int_{\Omega} \|z(x) - y^{-1}(x)\|^2 dx \\
&\stackrel{(3.1)}{=} \int_{\Omega} \|u(x) - \tilde{v}(x)\|^2 dx + \int_{\Omega} \|v(x) - \tilde{u}(x)\|^2 dx
\end{aligned} \tag{3.2}$$

Zusätzlich wird noch ein Regularisierer hinzugefügt. Dafür verwenden Christensen und Johnson [6] den ebenfalls bereits erläuterten elastischen Regularisierer (siehe Gleichung (2.2)).

Somit ergibt sich die Gesamtzielfunktion als

$$\begin{aligned}\mathcal{J}[y, z] &= \mathcal{D}[\mathcal{T}[y], \mathcal{R}] + \mathcal{D}[\mathcal{R}[z], \mathcal{T}] + \mathcal{P}^{ICC} + \mathcal{S}[y] + \mathcal{S}[z] \\ &= \int_{\Omega} |\mathcal{T}(y(x)) - \mathcal{R}(x)|^2 dx + \int_{\Omega} |\mathcal{R}(z(x)) - \mathcal{T}(x)|^2 dx \\ &\quad + \int_{\Omega} \|u(x) - \tilde{v}(x)\|^2 dx + \int_{\Omega} \|v(x) - \tilde{u}(x)\|^2 dx + \mathcal{S}[y] + \mathcal{S}[z]\end{aligned}$$

Für die Optimierung wird im hier vorgestellten Paper ein parametrischer Ansatz mit Hilfe von Fourier-Reihen verwendet. Dies wird in dieser Arbeit allerdings nicht getan, da ein nichtparametrischer Ansatz mehr Freiheitsgrade mit sich bringt.

3.2 Invertierung von Deformationsfeldern

Für den verwendeten Algorithmus ist es nötig, die Inversen von y und z zu berechnen. Dafür stellen Christensen und Johnson einen Invertierungsalgorithmus vor, für den vorausgesetzt wird, dass die Determinante der Jacobi-Matrix der jeweiligen Transformation positiv sind. Damit ist sichergestellt, dass es sich um einen Diffeomorphismus handelt und die jeweilige Transformation invertierbar ist. Die Inverse wird dann iterativ nach Algorithmus 1 berechnet, welcher ebenfalls im Verlauf dieser Arbeit implementiert wurde.

Der Algorithmus nähert für jeden Punkt x einen möglichen Urbildpunkt x_i

mit $y(x_i) = x$ an und invertiert so das Transformationsfeld punktweise. Die genaue Funktionsweise wird in Abbildung 3.1 dargestellt.

```

foreach  $x \in \Omega$  do
   $\delta := [1, 1]^T$ ;  $x_i := x$ ;  $i = 0$ 
  while  $\|\delta\| > threshold$  do
     $\delta = x - y(x_i)$ ;
     $x_i = x_i + \delta/2$ ;
     $i = i + 1$ ;
    if  $i > max\_iteration$  then
      | Algorithmus konvergiert nicht
    end
     $y^{-1}(x) = x_i$ ;
  end
end

```

Algorithmus 1: Algorithmus zur Invertierung eines Deformationsfeldes

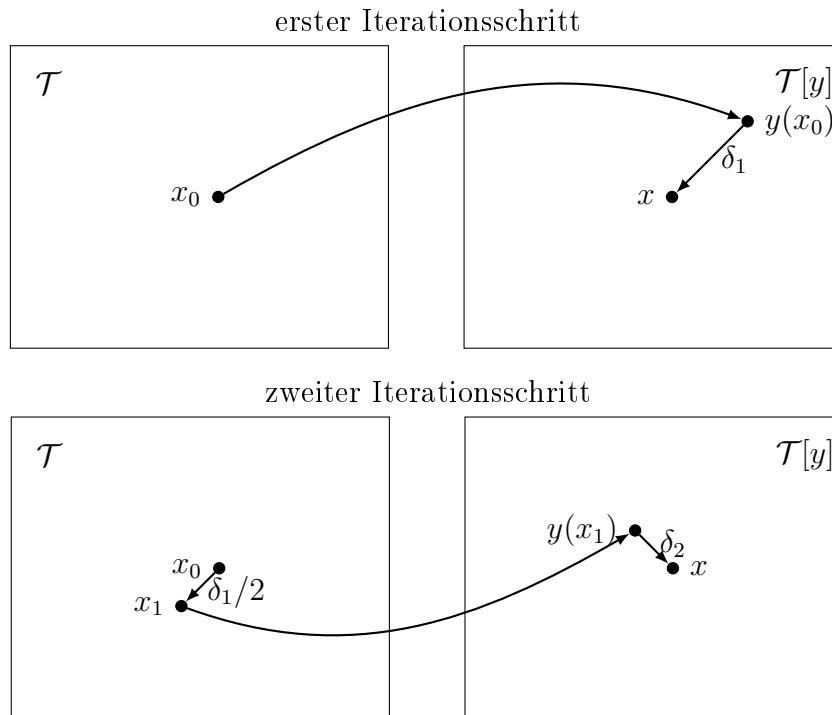


Abbildung 3.1: Funktionsweise des Invertierungsalgorithmus

Damit lassen sich also y^{-1} und z^{-1} näherungsweise berechnen und daraus er-

geben sich auch $\tilde{u}(x) = y^{-1}(x) - x$ und $\tilde{v}(x) = z^{-1}(x) - x$.

Genauere Betrachtung erfordert die Berechnung von $y(x_i)$, da x_i normalerweise kein Gitterpunkt ist. Es muss also interpoliert werden. Probleme entstehen bei diesem Algorithmus, wenn im Verlauf der Iteration x_i den Bildbereich Ω verlässt. Dann wäre ein entsprechendes Extrapolationsverfahren erforderlich. Extrapoliert man jedoch mit Null außerhalb des Bildbereichs, so ist die Konvergenz des Algorithmus nicht gesichert. Christensen und Johnson umgehen dieses Problem durch die Einschränkung, dass mögliche Transformationen Selbstabbildungen sein sollen und somit die Transformation an keinem Punkt den Bildbereich verlässt. Dies ist jedoch eine sehr starke Einschränkung, die nur wenige Transformationen erfüllen. Vor allem einfache affin-lineare Transformationen verändern den Bildbereich stark und dementsprechend gibt es viele Punkte, die durch die Transformation Ω verlassen. Für alle diese Fälle würde es keine Konvergenz des Algorithmus geben. Für nichtlineare Transformationen, die in etwa den Bildbereich gleich lassen, konvergiert der Algorithmus jedoch sehr gut. Deshalb wird dieses Problem in dieser Arbeit umgangen, indem eine affin-lineare Vorregistrierung durchgeführt wird. Diese ist leicht zu invertieren und bedarf keines iterativen Algorithmus. Im Anschluss kann dann noch eine nicht-lineare Registrierung durchgeführt werden, die mit diesem Algorithmus gut zu invertieren ist.

3.3 Vor- und Nachteile des Verfahrens

Ziel des vorgestellten Algorithmus ist, ein konsistentes Bildregistrierungsverfahren zu entwickeln, d.h. ein Verfahren, in dem eindeutige Korrespondenzen zwischen den einzelnen Punkten in Referenz und Template geschaffen werden. Dies ist jedoch nicht für alle Bildregistrierungsverfahren in der Praxis sinnvoll, da die Berechnung hierfür deutlich aufwändiger ist. So gibt es beispielsweise keine biologische Grundlage, Hirnscans verschiedener Patienten symmetrisch zu registrieren [7].

Außerdem erwarten Christensen und Johnson zum Beispiel bei Landmark-Algorithmen kaum Vorteile durch ihr Verfahren, da bei diesen Algorithmen bereits eindeutige Korrespondenzen zwischen ausgewählten Punkten festgelegt werden. Für Ansätze, die als Grundlage die Minimierung eines Distanzmaßes haben, erwarten sie jedoch eine Verbesserung der Ergebnisse, da in diesen Ansätzen häufig keine klaren Korrespondenzen entstehen.

Christensen und Johnson stellten in ihren Tests fest, dass der elastische Regularisierer allein nicht garantiert, dass y und z invers zueinander sind. Größte Konsistenz konnte durch Verwendung beider Nebenbedingungen, also Regularisierung und Penalty, erreicht werden.

4 Implementierung des Algorithmus

Ziel dieses Kapitels ist es, die Implementierung eines symmetrischen Registrierungsverfahrens mit Hilfe von FAIR vorzustellen. Im Gegensatz zu dem von G.E. Christensen und H.J. Johnson entwickelten Verfahren, wird bei dieser Implementierung ein nichtparametrischer Ansatz verfolgt, der unendlich viele Freiheitsgrade hat und damit bessere Ergebnisse verspricht. Die Implementierung erfolgt in Matlab.

Für die numerische Optimierung ist zunächst eine Diskretisierung erforderlich. Dazu wird ein Gitter verwendet, also eine Teilmenge von regelmäßig angeordneten Punkten der Menge Ω . Dies erfordert zunächst die Unterteilung der Menge in Zellen. Es wird hier mit einem nodalen Gitter gearbeitet.

Zur Vereinfachung sei ein Eckpunkt der Menge Ω in den Nullpunkt verschoben.

Definition (Nodales Gitter).

Sei $\Omega = [0, \omega_1] \times [0, \omega_2]$ die betrachtete Grundmenge und $m = (m_1, m_2)$ die Anzahl der Diskretisierungspunkte auf der Menge Ω .

Dann ergibt sich die Zellgröße wie folgt:

$$h = (h_1, h_2) = (\omega_1/m_1, \omega_2/m_2).$$

Dann bezeichnet man die Menge der Punkte $x = (a_1 h_1, a_2 h_2)$ mit $a_1, a_2 \in \mathbb{N}$ und $0 \leq a_1 \leq m_1, 0 \leq a_2 \leq m_2$ als *nodales Gitter*.

In einem nodalen Gitter werden also die Eckpunkte der Zellen betrachtet (siehe Abbildung 4.1).

Bei der Betrachtung von Gittern ist zu beachten, dass bei dem hier verwendeten Euler-Ansatz die Transformationen auf das Gitter angewendet werden und

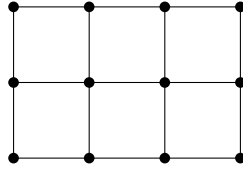


Abbildung 4.1: Nodales Gitter

sich das Bild genau entgegengesetzt verändert. Wird zum Beispiel das Gitter nach links verschoben, so wird das Bild nach rechts verschoben. Rotation des Gitters im Uhrzeigersinn führt zu Rotation des Bildes gegen den Uhrzeigersinn.

Mit dieser Diskretisierung ergibt sich das Distanzmaß als

$$\mathcal{D}[\mathcal{T}[y], \mathcal{R}] + \mathcal{D}[\mathcal{R}[z], \mathcal{T}] = \frac{\omega_1 \omega_2}{m_1 m_2} \cdot \sum_{x \in \Omega} (|T(y(x)) - R(x)|^2 + |R(z(x)) - T(x)|^2).$$

Analog gilt für die inverse Konsistenz:

$$\mathcal{P}^{ICC} = \frac{\omega_1 \omega_2}{m_1 m_2} \cdot \sum_{x \in \Omega} (\|y(x) - z^{-1}(x)\|^2 + \|z(x) - y^{-1}(x)\|^2).$$

4.1 Affin-lineare Vorregistrierung

Da es beim Invertierungsalgorithmus Probleme mit der Extrapolation der Bilder und dementsprechend bei der Invertierung von Transformationen, die den Bildbereich Ω verlassen, gibt, wurde zunächst eine affin-lineare Vorregistrierung vorgenommen. Diese ist leicht zu invertieren.

$$\begin{aligned} y &= A \cdot x + t \\ x &= A^{-1} \cdot (y - t) \\ x &= A^{-1} \cdot x - A^{-1} \cdot t \end{aligned}$$

Die Umkehrung dieser Vorregistrierung ist also ebenfalls eine affin-lineare Abbildung. Die Inverse von A lässt sich im Zweidimensionalen leicht berechnen

durch

$$A^{-1} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}^{-1} = \frac{1}{\det(A)} \begin{pmatrix} a_4 & -a_2 \\ -a_3 & a_1 \end{pmatrix}$$

Die Gesamttransformation y erhält man dann durch Nacheinanderausführung der affinen Transformation und der symmetrisch berechneten Transformation. Analog entsteht die inverse Transformation z durch Nacheinanderausführen der Transformation, die aus der symmetrischen Registrierung resultiert, und der Inversen der affinen Transformation.

4.2 Umsetzung in FAIR

Nach der affin-linearen Vorregistrierung bleibt noch die Berechnung des nicht-linearen Transformationsanteils mit Hilfe des symmetrischen Registrierungsansatzes.

Für die Realisierung mit FAIR wird die symmetrische Zielfunktion aufgeteilt und abwechselnd über y bzw. z optimiert (siehe Algorithmus 2). Dafür wird stets das Inverse der jeweils anderen Funktion benötigt. Die beiden Zielfunktionen ergeben sich wie folgt:

$$\begin{aligned} \mathcal{J}_1[y] &= \frac{\omega_1 \omega_2}{m_1 m_2} \cdot \sum_{x \in \Omega} (|T(y(x)) - R(x)|^2 + \|y(x) - z^{-1}(x)\|^2) + \mathcal{S}[y] \\ \mathcal{J}_2[z] &= \frac{\omega_1 \omega_2}{m_1 m_2} \cdot \sum_{x \in \Omega} (|R(z(x)) - T(x)|^2 + \|z(x) - y^{-1}(x)\|^2) + \mathcal{S}[z] \end{aligned}$$

Dabei muss nach jedem Optimierungsschritt die Inverse wieder neu bestimmt werden.

Die Implementierung der Zielfunktion erfolgt durch Abwandlung der in der FAIR-Toolbox enthaltenen NPIRobjFctn. Dabei muss ein zusätzlicher Term für die Berechnung der inversen Konsistenz eingefügt werden. Es wird also im Gegensatz zu dem im vorherigen Kapitel vorgestellten Paper von Christensen und Johnson ein nichtparametrischer Ansatz verwendet.

Aufgrund der positiven Glattheitseigenschaften wird die Spline-Interpolation verwendet und wie auch im Paper von Christensen und Johnson wird ein elas-

tischer Regularisierer verwendet.

```
xc=nodalGrid; yc=xc; zc=xc; yinv=xc; zinv = xc;
for level=minLevel:maxLevel do
  for iter=1:maxIter do
    yc=GaussNewton(yc,zinv)
    invert yc
    zc=GaussNewton(zc,yinv)
    invert zc
  end
end
```

Algorithmus 2: Symmetrische Registrierung

Der Gauss-Newton-Algorithmus wird dabei für jeweils einen Schritt ausgeführt.

4.3 Ergebnis der Vorregistrierung

Für den Test des Algorithmus wurden die Multileveldaten der in Abbildung 2.2 vorgestellten Röntgenaufnahmen zweier Hände verwendet.

Der ursprüngliche Abstand der Daten auf dem feinsten Level berechnet mit Verwendung der Summe der quadratischen Differenzen nach Gleichung (2.1) beträgt $7,9 \times 10^5$.

Auf diese Daten wurde in beiden Verfahren zunächst eine parametrische Optimierung für affin-lineare Transformationen durchgeführt. Dies wurde mit Hilfe der FAIR-Funktion MLPiR umgesetzt. Abbildung 4.2 zeigt dafür das Template überlagert mit dem ermittelten Transformationsgitter, die Referenz, die ursprüngliche Distanz der Bilder zueinander und die Distanz nach der Vorregistrierung.

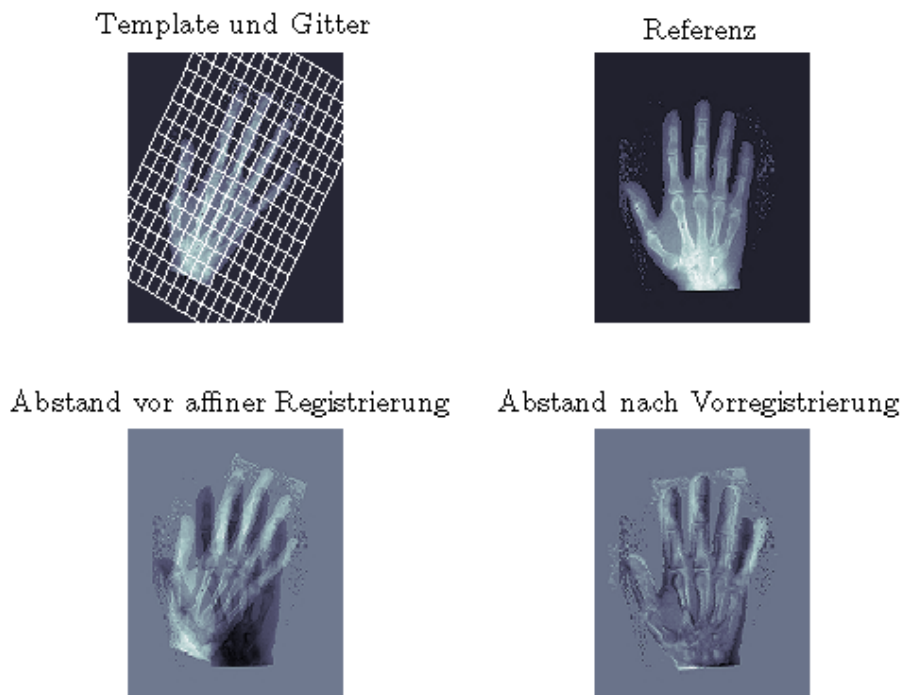


Abbildung 4.2: Ergebnisse der affin-linearen Registrierung.

Nach diesem Verfahren verringert sich die Distanz der Bilder deutlich auf $1,4 \times 10^5$.

Auf diese vorregistrierten Daten wurden nun verschiedene Verfahren angewendet.

4.4 Separate Registrierung in beide Richtungen

Die erhaltenen Daten nach der affin-linearen Transformation mussten zunächst wieder in Multileveldaten umgewandelt werden. Um zu sichern, dass die ermittelten Transformationen elastisch sind, muss am Ende der Berechnung mit Hilfe der Funktion `detJacobian` der Bereich der Volumenänderung bestimmt werden. Negative Werte bedeuten dabei, dass im Gitter Punkte aneinander vorbei bewegt wurden und die Position getauscht haben. Dies entspricht nicht der Definition einer elastischen Transformation und muss deshalb verhindert

werden. Dafür wird der Regularisierer höher gewichtet, also α höher gewählt. Tests ergaben, dass für α ein Wert von etwa 520 hinreichend ist, um diese Eigenschaften zu gewährleisten.

Außerdem wurde der Interpolationsparameter θ auf 10^{-2} gesetzt, damit die Daten einen angemessenen Schärfegrad behalten.

Die Registrierung von Template zu Referenz ergab als Minimalwert für die Distanz etwa $1,93 \times 10^4$. Das entspricht etwa 13,79% der Distanz nach der Vorregistrierung.

Abbildung 4.3 zeigt die Referenz, das Template vor der Registrierung mit ermitteltem Transformationsgitter sowie das transformierte Template. Außerdem ist der Abstand der Bilder nach der Registrierung dargestellt.

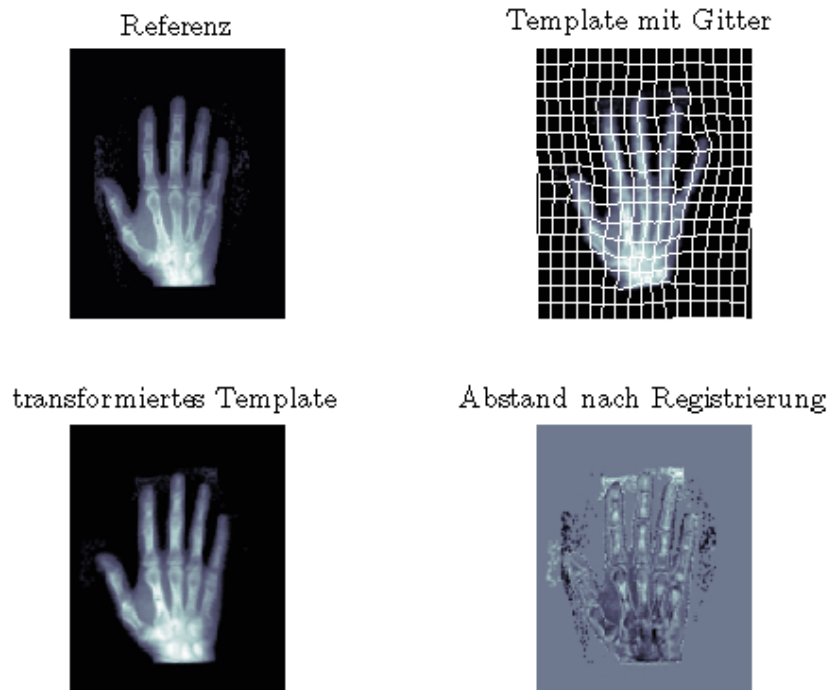


Abbildung 4.3: Ergebnis der Vorwärtstransformation

Die umgekehrte Registrierung, in der Referenz und Template vertauscht wurden, erreichte Minimalwerte der Distanz von $1,88 \times 10^4$. Das entspricht etwa 13,43% der Distanz nach der Vorregistrierung.

Abbildung 4.4 zeigt die Ergebnisse dieses Verfahrens analog zu Abbildung 4.3. Dabei ist in beiden Abbildungen zu sehen, dass die transformierten Bilder jeweils dem unveränderten Bild optisch sehr nahe kommen. Auch die Darstellung der Distanzen ist relativ gleichmäßig geworden, was bedeutet, dass an fast allen Punkten große Ähnlichkeit zwischen den jeweils verglichenen Bildern besteht.

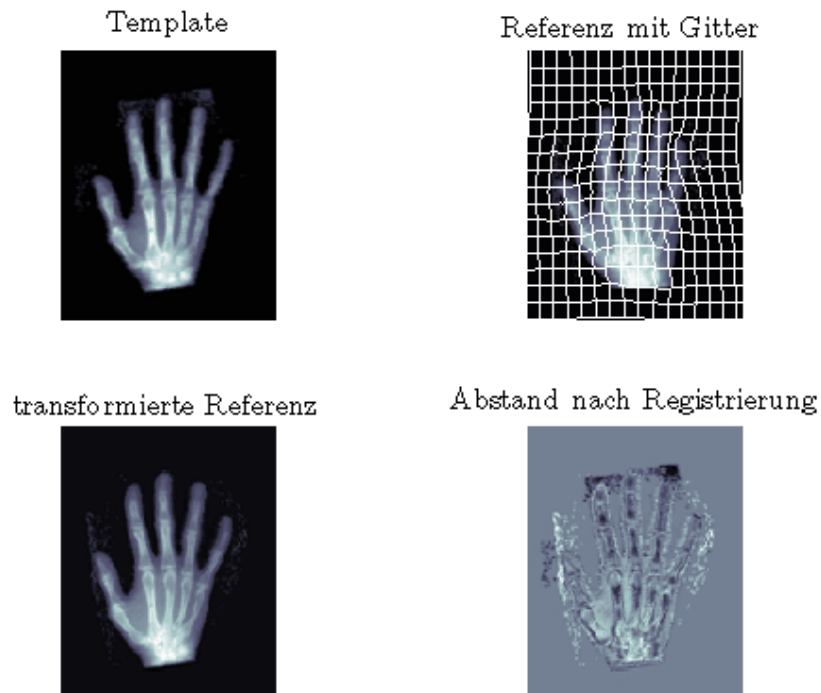


Abbildung 4.4: Ergebnis der Rückwärtstransformation.

Berechnet man im Anschluss aus den Ergebnissen für y und z mit Hilfe von Algorithmus 1 aus Abschnitt 3.2 die Inversen y^{-1} und z^{-1} , kann man so die inverse Konsistenz dieses Verfahrens überprüfen.

Die Differenz zwischen y und z^{-1} weist Werte bis zu $7,21 \times 10^{-3}$ auf, umgekehrt erreicht die Differenz zwischen z und y^{-1} Werte bis zu $7,91 \times 10^{-3}$.

4.5 Symmetrische Registrierung

Für die Umsetzung dieses Verfahrens musste zunächst die Zielfunktion formuliert und implementiert werden. Dazu wurde die Funktion `NPIRobjFctn` aus der FAIR-Toolbox um einen Term, der die inverse Konsistenz sichern soll, erweitert. Mit Hilfe dieser Funktion konnte dann das Verfahren nach Algorithmus 2 umgesetzt werden.

Dafür wurde beginnend mit dem größten Diskretisierungsgrad schrittweise über die Transformationen y bzw. z optimiert, wobei jeweils ein Schritt des Gauss-Newton-Verfahrens durchgeführt wurde. Nach jeder Berechnung einer Transformation musste die Inverse dieser mit Algorithmus 1 berechnet werden, da die aktuelle Inverse von y zur Berechnung von z benötigt wurde und umgekehrt. Dieser Vorgang wurde auf den Multileveldaten durchgeführt, das heißt zunächst wurde auf einem groben Gitter gearbeitet und das Ergebnis dieser Berechnungen wurde auf das nächstfeinere Gitter übertragen, um bessere Ergebnisse bei geringerem Rechenaufwand zu erzielen.

Auch hier wurde der Regularisierer mit einem Parameter α von etwa 500 gewichtet.

Schon mit einer sehr geringen Gewichtung des Penalty-Terms zeigt sich hier, dass die inverse Konsistenz deutlich besser ist. Die minimale Distanz von $\mathcal{T}[y]$ zu \mathcal{R} beträgt dabei $1,95 \times 10^4$ und der Abstand von $\mathcal{R}[z]$ zu \mathcal{T} liegt bei $1,79 \times 10^4$. Der Abstand von y zu z^{-1} beträgt dabei im Maximum $2,6 \times 10^{-3}$, der Abstand von z zu y^{-1} beträgt $1,4 \times 10^{-3}$. Die Distanzwerte der Bilder liegen also in einem ähnlichen Bereich wie in den separaten Vergleichsverfahren, doch die inverse Konsistenz wurde bereits auf etwa ein Drittel reduziert.

Eine höhere Gewichtung des Penalty-Terms führt zu einer deutlich verbesserten inversen Konsistenz, aber erhöht die Distanzwerte. Abbildung 4.5 zeigt die Ergebnisse des Tests mit höherer Penalty. Abgebildet werden jeweils die Ausgangsbilder überlagert mit dem ermittelten Transformationsgitter, die transformierten Bilder, die Distanz der Bilder nach der Registrierung und

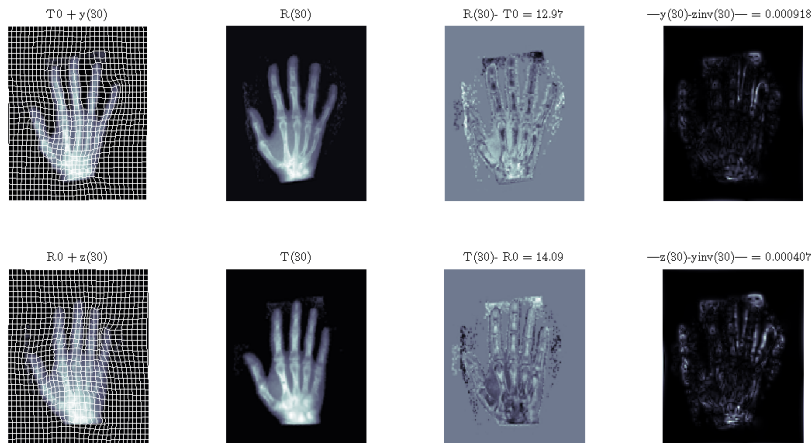


Abbildung 4.5: Ergebnisse der symmetrischen Registrierung

Die minimale Distanz von $\mathcal{T}[y]$ zu \mathcal{R} beträgt dabei $2,07 \times 10^4$ also rund 14,79% und der Abstand von $\mathcal{R}[z]$ zu \mathcal{T} liegt bei $1,90 \times 10^4$ also etwa 13,57% der Distanz nach der Vorregistrierung. Diese Werte sind also auch nur wenig größer geworden. Die maximale Abweichung zwischen Transformation und inverser der umgekehrten Transformation beträgt jedoch nur noch $9,22 \times 10^{-4}$, ist also um eine Zehnerpotenz kleiner geworden.

Auch hier zeigt der optische Vergleich der Ausgangsbilder und der transformierten Bilder, dass die Distanz deutlich kleiner ist. Große Differenz der inversen Konsistenz befinden sich dabei in den verrauschten Bereichen vor den Zeige- und Ringfingern. Eine Reduktion dieser Fehlerbereiche könnte dazu führen, dass die Ergebnisse noch deutlich geringer sind.

4.6 Vergleich

Verglichen wird der symmetrische Ansatz mit einem einfachen Optimierungsverfahren mittels FAIR. Dabei werden separat eine Transformation vom Template zur Referenz und eine von der Referenz zum Template berechnet und anschließend die inverse Konsistenz dieser Abbildungen überprüft und mit den Ergebnissen des symmetrischen Verfahrens verglichen. Für beide Verfahren wurde zunächst eine affin-lineare Vorregistrierung durchgeführt, um die Vergleichbarkeit der Ergebnisse zu gewährleisten.

Die Ergebnisse sind in Tabelle 4.1 dargestellt, exemplarisch wurden hier die Werte für Gewichtung des Penalty-Terms zur inversen Konsistenz mit 10^4 und 10^6 dargestellt.

Verfahren	$\mathcal{D}[\mathcal{T}[y], \mathcal{R}]$	$\mathcal{D}[\mathcal{R}[z], \mathcal{T}]$	$\ y - z^{-1}\ $	$\ z - y^{-1}\ $
Registrierung in je eine Richtung	$1,93 \times 10^4$	$1,88 \times 10^4$	$7,21 \times 10^{-3}$	$7,01 \times 10^{-3}$
symm. Registrierung mit Gewichtung 10^4	$2,06 \times 10^4$	$1,9 \times 10^4$	$9,22 \times 10^{-4}$	$4,07 \times 10^{-4}$
symm. Registrierung mit Gewichtung 10^6	$5,9 \times 10^4$	$5,82 \times 10^4$	$3,32 \times 10^{-6}$	$8,87 \times 10^{-6}$

Tabelle 4.1: Testergebnisse

Mit Tabelle 4.1 wird klar, dass man bei der Verwendung eines symmetrischen Verfahrens abwägen muss, worauf das Augenmerk liegt. Ist die inverse Konsistenz von großer Bedeutung und geht deshalb mit großer Gewichtung in die Zielfunktion ein, so geschieht dies zu Lasten der Distanz. Mit weniger starker Gewichtung können allerdings ähnlich gute Ergebnisse bezüglich der Distanz der Bilder erreicht werden, wie in einem einfachen Registrierungsverfahren. Es gilt also: je besser die inverse Konsistenz ist, desto schlechter lässt sich der Abstand der Bilder minimieren.

5 Fazit

Im Zuge dieser Arbeit wurde ein symmetrisches, nichtparametrisches Registrierungsverfahren mit Hilfe der FAIR-Toolbox implementiert. Dafür wurden zunächst einige Grundlagen der Bildregistrierung vorgestellt und anschließend ein symmetrischer Verfahrensansatz erläutert. Die Implementierung des Algorithmus erfolgte in Matlab.

Bei der symmetrischen Bildregistrierung werden zwei Transformationen y und z gesucht, wobei y auf das Template und z auf das Referenzbild angewendet wird, unter der Nebenbedingung, dass $y = z^{-1}$ gilt. Dazu wurde ein iterativer Algorithmus erstellt, der abwechselnd über eine der Transformationen optimiert und die verbesserte Transformation anschließend invertiert. Die Inverse wird dabei benötigt, um die jeweils andere Transformation zu berechnen.

Verglichen wurde dieses Verfahren mit einer einfachen Bildregistrierung in beide Richtungen. Dabei wurden also y und z separat berechnet, wobei für die Berechnung von z Template und Referenzbild vertauscht wurden. Anschließend wurde für diese beiden berechneten Transformationen überprüft, in welchem Maße sie von der Bedingung der inversen Konsistenz abweichen. Für beide Verfahren wurde zunächst eine affin-lineare Vorregistrierung durchgeführt, da sonst Probleme mit der Invertierung der Transformationsfelder auftreten können.

Getestet wurde der Algorithmus mit Röntgenaufnahmen zweier Hände. Dabei ergab sich, dass mit entsprechender Parameterisierung des Penalty-Terms für die inverse Konsistenz bei Nutzung des symmetrischen Verfahrens ähnliche Werte für das Distanzmaß der zu registrierenden Bilder erreicht werden können und zusätzlich eine bessere inverse Konsistenz erzielt werden kann. Diese war schon mit relativ kleiner Wichtung des Penalty-Terms eine Zehnerpotenz kleiner als bei dem Vergleichsalgorithmus.

Wird der Penalty-Terms allerdings stärker gewichtet, um das Hauptaugenmerk auf die inverse Konsistenz zu richten, dann wird schnell deutlich, dass dadurch der Abstand der Bilder nicht so gut optimiert wird wie zuvor.

Der Algorithmus ist also gut in der Lage eine deutlichere Zuordnung von Punkten des Templates zu Punkten der Referenz zu ermöglichen. Wie erwartet geht dies allerdings mit einer schlechteren Optimierung des Distanzmaßes einher. Hier sollte also im Einzelfall je nach Anwendung entschieden werden, welches Verfahren verwendet wird und wie die Parameter gewichtet werden.

Zukünftig könnten weitere Tests des Algorithmus durchgeführt werden. Zum Beispiel könnten dreidimensionale Daten als Testobjekte dienen oder Variationen in Interpolationsverfahren durchgeführt werden. Ebenso könnten Test mit anderen Distanzmaßen und Regularisierern weitere Erkenntnisse bringen. Dafür bietet FAIR zahlreiche weitere Möglichkeiten.

Abschließend ist zu sagen, dass der Algorithmus die Erwartungen bestätigt hat: bei einem intensitätsbasierten Registrierungsproblem kann damit eine deutlich verbesserte inverse Konsistenz erreicht werden, was aber zur Folge hat, dass die Distanz weniger gut minimiert wird. Außerdem ist die Berechnung deutlich aufwändiger und langwieriger.

Literaturverzeichnis

- [1] C. Broit. *Optimal registration of deformed images*. PhD thesis, 1981.
- [2] I. N. Bronstein, K. A. Semendjajew, and G. Musiol. *Taschenbuch der Mathematik, m. CD-ROM*. Deutsch (Harri), 2008.
- [3] L. G. Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376, Dec. 1992.
- [4] M. Burger, J. Modersitzki, and L. Ruthotto. A hyperelastic regularization energy for image registration. *SIAM Journal on Scientific Computing*, 35(1):B132–B148, 2013.
- [5] M. Chen, W. Lu, Q. Chen, K. J. Ruchala, and G. H. Olivera. A simple fixed-point approach to invert a deformation field. *Medical Physics*, 35(1):81, 2008.
- [6] G. E. Christensen and H. J. Johnson. Consistent image registration. *IEEE Trans Med Imaging*, 20(7):568–582, July 2001.
- [7] W. R. Crum, O. Camara, and D. J. Hawkes. Methods for inverting dense displacement fields: evaluation in brain image registration. In *Proceedings of the 10th international conference on Medical image computing and computer-assisted intervention - Volume Part I, MICCAI'07*, pages 900–907, Berlin, Heidelberg, 2007. Springer-Verlag.
- [8] M. Droske and M. Rumpf. A variational approach to non-rigid morphological registration. *SIAM Appl. Math.*, 64:668–687.
- [9] J. Ehrhardt. Skriptum zur vorlesung computergrafik, 2012.
- [10] B. Fischer and J. Modersitzki. Ill-posed medicine—an introduction to image registration. *Inverse Problems*, 24(3):034008+, June 2008.

- [11] J. Fitzpatrick, D. Hill, and C. Maurer. Image Registration. *Handbook of Medical Imaging, Volume 2. Medical Image Processing and Analysis*, pages 447–513, June 2000.
- [12] C. A. Glasbey and K. V. Mardia. A review of image warping methods. *Journal of Applied Statistics*, 25:155–171, 1998.
- [13] D. G. Gobbi and T. M. Peters. *Generalized 3D nonlinear transformations for medical imaging: an object-oriented implementation in VTK*, volume 27. 2003.
- [14] J. Modersitzki. *Fair: Flexible Algorithms for Image Registration (Fundamentals of Algorithms)*. Society for Industrial and Applied Mathematics, Oct. 2009.
- [15] P. Rogelj and S. Kovacic. Symmetric image registration. In *Proceedings of SPIE, Vol. 5032, Medical Imaging 2003: Image Processing*, pages 484–493, 2003.
- [16] O. Schmitt. *Die multimodale Architektonik des menschlichen Gehirns*. Habilitation, Institut für Anatomie, Universität zu Lübeck, 2001.
- [17] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, Oct. 2003.