



UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS AND
IMAGE COMPUTING

Bachelorarbeit

Nicht-lineare Registrierung mit einem Multilevel- und Multiskalen-Ansatz

Anna Katharina Trull

Arbeit betreut von
Prof. Dr. Bernd Fischer und Dipl.-Inf. Björn Beuthien
Institute of Mathematics and Image Computing

Abgabe: 06.09.2011



IM FOCUS DAS LEBEN

Eidesstattliche Erklärung

Ich versichere, die Bachelorarbeit selbständig und lediglich unter Benutzung der angegebenen Quellen und Hilfsmittel verfasst zu haben.

Weiterhin erkläre ich, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Ort, Datum

Anna Katharina Trull

Danksagung

An dieser Stelle möchte ich mich bei den Menschen bedanken, die mich bei dieser Arbeit unterstützt haben.

Zum Einen bei Prof. Dr. rer. nat. Bernd Fischer dafür, dass ich die Arbeit am Institute of Mathematics and Image Computing schreiben durfte.

Bei Dipl. Inf. Björn Beuthien, für die umfangreiche und sehr gute Betreuung und dafür, dass er jede Woche meine Fragen beantwortet hat.

Selbstverständlich bedanke ich mich auch bei Claudia Paulick, Stefan Riedel und Björn Beuthien, die meine Arbeit Korrektur gelesen haben.

Last-but-not-least möchte ich mich bei meinen Eltern bedanken, die immer an mich glauben und mich während meines Studium zu jeder Zeit moralisch und tatkräftig unterstützt haben.

Inhaltsverzeichnis

Eidesstattliche Erklärung	2
Danksagung	3
Notation	6
1 Einleitung	8
2 Medizinische Grundlagen	9
2.1 Multiple Sklerose	9
2.2 Ursachen	9
2.3 Symptome	9
2.4 Diagnostische Möglichkeiten	9
3 Grundlagen der Bildregistrierung	11
3.1 Gitter	12
3.1.1 Cellcentered Gitter	12
3.1.2 Nodal-Gitter	13
3.1.3 Staggered-Gitter	14
3.1.4 Gitterwechsel	15
3.2 Rigide Deformation	16
4 Spline-Interpolation	17
4.1 Lineare Splines	18
4.2 Kubische Splines	19
5 Ähnlichkeit und Distanzmaß	21
5.1 Bild	21
5.2 Summe der quadrierten Differenzen (SSD)	22
6 Optimierung	24
6.1 Gauß-Newton-Verfahren	24
6.2 Abbruchkriterien	25
6.3 Schrittweitenbestimmung	26
7 Multiresolutions- und Multiskalenverfahren	28
7.1 Multiresolutionsverfahren	28
7.2 Restriktion und Prolongation	30
7.3 Multiskalen-Ansatz	32
8 Diffusive Regularisierung	34
8.1 Kontinuierliche Formulierung	35
8.2 Diskrete Formulierung	35

9 Ergebnisse	39
9.1 Synthetische Daten	39
9.1.1 Registrierung von zwei Kreisen unterschiedlicher Größe .	39
9.1.2 Registrierung von einem kleinen Kreis auf den Buchsta- ben C	42
9.2 MRT-Daten	44
10 Zusammenfassung und Ausblick	47
Abbildungsverzeichnis	48
Literatur	49

Notation

d	Dimension	$d \in \mathbb{N}$
B	Bild	$B : \mathbb{R}^d \rightarrow \mathbb{R}$
Ω	Gebiet	$\Omega \subset \mathbb{R}^d$
T	Templatebild	$\Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$
R	Referenzbild	$\Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$
m	Anzahl an Diskretisierungspunkten	$m \in \mathbb{R}^d$
ω	Maße des Gitters	$\omega \in \mathbb{R}^d$
h	Zelllänge	$h \in \mathbb{R}^+$
x^{cc}	Cellcentered-Gitter	$x^{cc} \in \mathbb{R}^m$
x^n	Nodal-Gitter	$x^n \in \mathbb{R}^{m+1}$
x^s	Staggered-Gitter	$(x_1^s, x_2^s) \in \mathbb{R}^2$
W	Gitterwechselmatrix: Nodal- auf Cellcentered-Gitter	$W \in \mathbb{R}^{m,m+1}$
P_h^H	Prolongationsoperator	$P_h^H \in \mathbb{R}^{2m+1,m+1}$
P_H^h	Restriktionsoperator	$P_H^h \in \mathbb{R}^{m+1,2m+1}$
\mathbf{y}	Deformation	$\mathbf{y} : \mathbb{R}^d \rightarrow \mathbb{R}^d$
Q	Rotationsmatrix	$Q \in \mathbb{R}^{d,d}$
β	Drehwinkel	$\beta \in [0, 2\pi)$
γ	Translation	$\gamma \in \mathbb{R}^d$
D	Distanzmaß	$\mathbb{R}^n \rightarrow \mathbb{R}$
S	Regularisierer	$\mathbb{R}^n \rightarrow \mathbb{R}$
α	Regularisierungsparameter	$\alpha \in \mathbb{R}$
\odot	Hadamard-Produkt	$C = A \odot B$ mit $A, B, C \in \mathbb{R}^{m,n}$, $c_{ij} = a_{ij}b_{ij}$, $i = 1, \dots, m$, $j = 1, \dots, n$

\otimes	Hadamard-Division	$C = A \oslash B$ mit $A, B, C \in \mathbb{R}^{m,n}$, $c_{ij} = a_{ij}/b_{ij}$, $b_{ij} \neq 0, i = 1, \dots, m,$ $j = 1, \dots, n$
\otimes	Kronecker-Produkt	$C = A \otimes B$ mit $A \in \mathbb{R}^{m_1, n_1}$, $B \in \mathbb{R}^{m_2, n_2}$ und $C \in \mathbb{R}^{m_1 m_2, n_1 n_2}$ $C = \begin{pmatrix} a_{11}B & \cdots & a_{1n_1}B \\ \vdots & \ddots & \vdots \\ a_{m_1 1}B & \cdots & a_{m_1 n_1}B \end{pmatrix}$
f	Funktion	$\mathbb{R}^n \rightarrow \mathbb{R}$
∇f	Gradient	$\mathbb{R}^n \rightarrow \mathbb{R}^n$
$\nabla^2 f$	Hessematrix	$\mathbb{R}^n \rightarrow \mathbb{R}^{n,n}$
g	mehrdimensionale Funktion	$\mathbb{R}^l \rightarrow \mathbb{R}^l$
J	Jakobimatrix	$J = (\nabla g_1, \dots, \nabla g_n),$ $J : \mathbb{R}^d \rightarrow \mathbb{R}^{n,l}$
$\text{diag}(x)$	Diagonalmatrix	$\begin{pmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_m \end{pmatrix} \in \mathbb{R}^{m,m}$
E_m	Einheitsmatrix	$\mathbb{R}^{m,m}$

1 Einleitung

Multiple Sklerose (MS) ist eine chronisch-entzündliche Autoimmunerkrankung des Nervensystems. Weltweit sind ca. 2,5 Millionen Menschen von dieser Krankheit betroffen. Typischerweise leiden die Patienten an unterschiedlichen Symptomen, die fälschlicherweise auf andere Krankheiten schließen lassen. Aus diesem Grund besteht die Diagnostik aus verschiedenen Bereichen. Ein Abschnitt, auf den wir in dieser Arbeit eingehen möchte, ist die Verwendung eines bildgebenden Diagnoseverfahrens, der Magnet-Resonanz-Tomographie (MRT). Bei dieser Untersuchung können Entzündungsherde gut erkannt werden, da Gewebeveränderungen im Gehirn anatomisch getreu abgebildet werden. Somit kann eine MS auch diagnostiziert werden, bevor Symptome auftreten.

Um die zu verschiedenen Zeitpunkten aufgenommenen MRT-Aufnahmen miteinander vergleichen zu können, wird eine Registrierung der Daten vorgenommen. Das Registrierungsproblem beschreibt den Vorgang, korrespondierende Punkte in zwei Bildern durch geometrische Deformationen aufeinander abbilden. Im Allgemeinen wird diese Aufgabe als Optimierungsproblem gestellt, bei dem eine Zielfunktion minimiert wird.

Das Kapitel 2 geht auf die medizinischen Grundlagen von Multiple Sklerose ein, um einen Einblick in die Diagnostik und die Verwendung des bildgebenden Verfahrens zu geben. Anschließend wird in Kapitel 3 die Problemstellung der rigiden Registrierung formuliert und grundlegende Begriffe, z.B. Gitter, eingeführt. Weitere notwendige Verfahren, wie die Interpolation, das Distanzmaß und das Optimierungsverfahren werden in den Kapiteln 4, 5 und 6 näher beschrieben. Das Multiresolutions- und das Multiskalenverfahren werden schließlich in Kapitel 7 erläutert. In dem 8. Kapitel wird auf die Problemstellung der nicht-linearen Registrierung eingegangen. Die Ergebnisse werden in Kapitel 9 vorgestellt.

2 Medizinische Grundlagen

2.1 Multiple Sklerose

Bei der Multiple Sklerose (lat. Encephalomyelitis disseminata) handelt es sich um eine chronisch-entzündliche Autoimmunerkrankung des Nervensystems, die mit einem Verlust der Axone und Neurone einhergeht. Aufgrund des schubartigen Verlaufs und der klinischen Vielfalt an Symptomen kommt diese Erkrankung in vielen Formen vor. Oft wird MS zwischen dem 20. und 40. Lebensjahr diagnostiziert, wobei Frauen doppelt so oft betroffen sind, wie Männer. Die Erkrankungshäufigkeit nimmt außerdem mit der Entfernung zum Äquator zu [Deu10].

2.2 Ursachen

Bei MS-Patienten treten im Gehirn, genauer gesagt in der weißen Substanz, multiple entzündliche Entmarkungsherde auf, wodurch die Nerven Informationen nicht mehr vollständig weitergeben können. Dieser Prozess wird auch als Demyelinisierung bezeichnet. Die Entmarkungsherde können zum Einen abklingen und ihre normalen Funktionen wieder aufnehmen, zum Anderen kann es aber auch zu Vernarbungen (Sklerosen) des geschädigten Gewebes kommen.

2.3 Symptome

Da das Beschwerdebild sehr unterschiedlich sein kann über die Symptomatik keine allgemeingültige Aussage getroffen werden. Eines der Frühsymptome ist z.B. eine Störungen der Augenoptik, was sich durch Doppelbilder oder auch dunkle Flecken im Sichtfeld bemerkbar macht. Es kann aber auch zu Koordinationsproblemen kommen, die sich durch Gleichgewichtsstörungen oder auch Sprachstörungen bemerkbar machen.

2.4 Diagnostische Möglichkeiten

Die McDonald Kriterien bilden die Grundlage der Diagnose. Sie setzen sich aus einer Kombination von klinischen Untersuchungen, Labortests und bildgebenden Diagnoseverfahren, z.B. MRT-Aufnahmen, zusammen. Der Arzt kann mit diesen auf eine sichere MS, eine mögliche MS und keine MS schließen.

Im Näheren soll nun auf die Magnet-Resonanz-Tomographie eingegangen werden.

Magnet-Resonanz-Tomographie (MRT) Die Magnet-Resonanz-Tomographie wird auch als Kernspintomographie bezeichnet. Es handelt sich dabei um ein Schnittbildverfahren, d.h. der menschliche Körper wird in einzelne Ebenen unterteilt. Bei dieser Untersuchungsmethode werden die Wasserstoffprotonen in Schwingung gebracht bzw. drehen sich um die eigene Achse. Dieses Verhalten wird als Kernspin bezeichnet und erzeugt ein kleines Magnetfeld. Wenn man die Kerne nun einem, von außen kommenden, Magnetfeld aussetzt, richten sich die Kerne uniform, wie eine Kompassnadel, aus. Folgt darauf eine zeitlich begrenzte Folge von Radiowellen, nehmen die Kerne Energie auf und richten sich entgegengesetzt zum Magnetfeld aus. Die Energie wird nach dem Signal wieder abgegeben (MRT-Signal) und die Kerne gehen in die Ausgangsposition zurück. Bei einer T2-gewichteten MRT-Aufnahme wird die weiße Substanz als leicht dunkel dargestellt, die graue Substanz als leicht hell und Läsionen als hell [MM06]. Der Vorteil von MRT-Aufnahmen liegt darin, dass weiches Gewebe in einer hohen Qualität abgebildet wird. Zudem kann jede Schnittebene dargestellt werden, ohne den Patienten umlagern zu müssen [Bon11].

Bildregistrierung bei Multiple Sklerose Im Folgenden werden wir verschiedene Techniken kennenlernen um Bilder zu vergleichen. Die Daten wurden dabei zu verschiedenen Zeitpunkten aufgenommen. Mit diesen Verfahren haben wir die Möglichkeit Veränderungen im Gewebe zu erkennen und können aufgrund der Grauwertintensität entscheiden ob es sich um einen Rückgang oder nicht. Wir werden uns im weiteren Verlauf MRT-Bilder von MS-Patienten anschauen und unsere Herangehensweise daran testen.

3 Grundlagen der Bildregistrierung

In der Registrierung wird nach einer *Deformation* y gesucht, die das Templatbild \mathcal{T} an das Referenzbild \mathcal{R} angleicht, sodass sich beide Bilder möglichst *ähnlich* sind [Pap08].

Mathematisch formuliert bedeutet dies:

$$\min_y \mathcal{D}[y] = \min_y \mathcal{D}[\mathcal{T}, \mathcal{R}; y], \quad (1)$$

wobei \mathcal{D} eine Distanz beschreibt.

Die rigide Deformation, d.h. eine Rotation und eine Translation, bildet in dieser Arbeit die erste mögliche Deformation. Im weiteren Verlauf werden wir diese weiterführen zur nicht-linearen Registrierung. Des Weiteren wird der *Discretize-then-Optimize-Ansatz* verwendet, der die kontinuierliche Fragestellung erst diskretisiert und anschließend optimiert. Diese Herangehensweise hat den Vorteil, dass das Registrierungsproblem mit Hilfe von Newton-artigen Optimierungsverfahren gelöst werden kann. In dieser Arbeit wird hierbei das Gauß-Newton-Verfahren angewendet, auf das wir in dem Kapitel 6 näher eingehen werden.

Die Gleichung (1) wirft verschiedene Fragestellungen auf, z.B. soll erläutert werden, was eine Deformation ist, was für eine Distanz gewählt wird oder auch wie sich die Ähnlichkeit zweier Bilder ausdrücken lässt. Diese und auch andere Problematiken sollen im Folgenden angesprochen werden.

Zunächst wollen wir erklären was Gitter sind.

3.1 Gitter

Gitter teilen ein Gebiet Ω in Zellen gleicher Größe ein. Wir unterscheiden zwischen dem *Cellcentered*-, *Nodal*- und *Staggered-Gitter* [Pap08].

Das Cellcentered-Gitter wird im Allgemeinen für die Diskretisierung von Bildern (vgl. Kapitel 5) verwendet. Die verbleibenden Gitter werden zur Beschreibung von Deformationen genutzt. Die unterschiedlichen Gittertypen werden wir in den folgenden Abschnitten genauer beschrieben.

3.1.1 Cellcentered Gitter

Im *eindimensionalen Fall* ($d = 1$) wird das Intervall $\Omega = (0, \omega)$ in $m \in \mathbb{N}$ Zellen unterteilt. Die einzelnen Zellen haben die Länge $h = \frac{\omega}{m}$. Für die einfachere Umsetzung werden die Koordinaten der Zellmittelpunkte in Vektoren $x^{cc} \in \mathbb{R}^m$ gespeichert, d.h.

$$(x^{cc})_k = (k - \frac{1}{2}) \cdot h, k = 1, \dots, m.$$

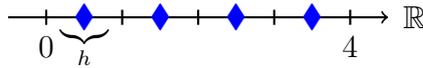


Abbildung 1: Cellcentered-Gitter für den eindimensionalen Fall mit $\omega = 4$; die blauen Rauten geben die Mittelpunkte an.

Bei einem *zweidimensionalen Cellcentered-Gitter* ($d = 2$) ist das Gebiet $\Omega \subset \mathbb{R}^2$ durch $(0, \omega_1) \times (0, \omega_2)$ gegeben, wobei $\omega = (\omega_1, \omega_2)$ ist. Die Anzahl der Diskretisierungspunkte in x_1 - und x_2 -Richtung sei durch $m = (m_1, m_2) \in \mathbb{N}^2$ gegeben. Unter Verwendung des eindimensionalen Falls und $h_k = \frac{\omega_k}{m_k}$, $k = 1, 2$ erhalten wir den Mittelpunkt für eine einzelne Zelle (i, j) mittels

$$x_{i,j}^{cc} = ((i - \frac{1}{2}) \cdot h_1, (j - \frac{1}{2}) \cdot h_2) \text{ mit } i = 1, \dots, m_1, j = 1, \dots, m_2.$$

Seien $x_i^{cc} \in \mathbb{R}^{m_i}$ für $i = 1, 2$, dann folgt mit Hilfe des Vektors $e_{m_i} = (1, \dots, 1) \in \mathbb{R}^{m_i}$

$$x^{cc} = \begin{pmatrix} e_{m_2} \otimes x_1^{cc} \\ x_2^{cc} \otimes e_{m_1} \end{pmatrix} \in \mathbb{R}^{2m_1m_2}. \quad (2)$$

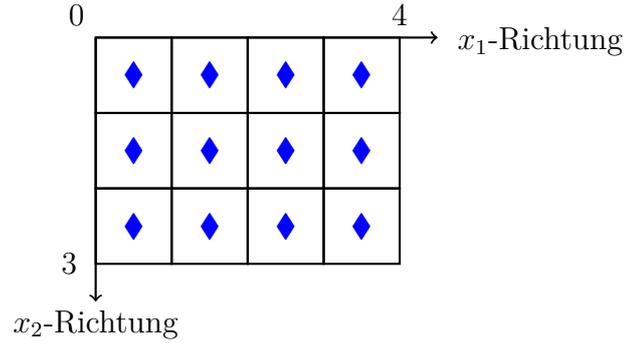


Abbildung 2: Cellcentered-Gitter für den zweidimensionalen Fall mit $\omega_1 = 4, \omega_2 = 3$; die blauen Rauten geben die Mittelpunkte an.

3.1.2 Nodal-Gitter

Ebenso, wie beim Cellcentered-, wird bei dem Nodal-Gitter im *eindimensionalen Fall* das Intervall $\Omega = (0, \omega)$ in m Zellen unterteilt, die die Länge $h = \frac{\omega}{m}$ haben. Jedoch werden beim Nodal-Gitter die Randpunkte der Zellen und nicht die Mittelpunkte beschrieben. Für eine einfachere Umsetzung werden auch diese Koordinaten in Vektoren gespeichert, d.h.

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

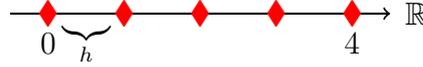


Abbildung 3: Nodal-Gitter für den eindimensionalen Fall mit $\omega = 4$; die roten Rauten geben die Zellrandpunkte an.

Im *zweidimensionalen Fall* lässt sich das Nodal-Gitter mittels einer Kombination von zwei eindimensionalen Gittern bilden. Seien $x_i^n \in \mathbb{R}^{m_i+1}$ für $i = 1, 2$, dann folgt mit $e_{m_i+1} = (1, \dots, 1) \in \mathbb{R}^{m_i+1}$

$$x^n = \begin{pmatrix} e_{m_2+1} \otimes x_1^n \\ x_2^n \otimes e_{m_1+1} \end{pmatrix} \in \mathbb{R}^{2(m_1+1)(m_2+1)}. \quad (3)$$

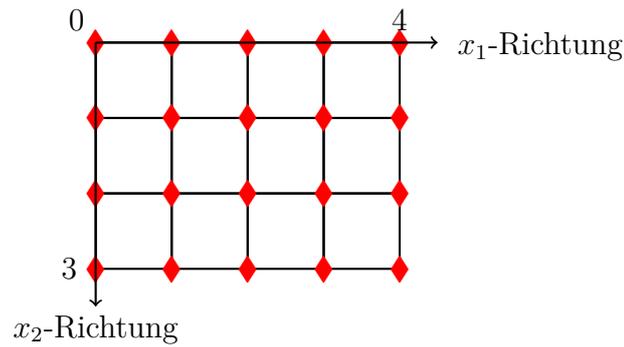


Abbildung 4: Nodal-Gitter für den zweidimensionalen Fall mit $\omega_1 = 4, \omega_2 = 3$; die roten Rauten geben die Randpunkte an.

3.1.3 Staggered-Gitter

Eine Kombination aus einem Nodal- und Cellcentered-Gitter bildet das Staggered-Gitter. Sei die Dimension $d = 2$ und die Komponenten $x_1 = x^n \in \mathbb{R}^{m_1+1}$ und $x_2 = x^{cc} \in \mathbb{R}^{m_2}$, so lassen sich die Gitterpunkte $(x_1^s, x_2^s) \in \mathbb{R}^2$ wie folgt für $i = 1, \dots, m_1 + 1$ und $j = 1, \dots, m_2$ konstruieren,

$$x_1^s = x_i^n + \left(i - \frac{1}{2}\right)h_1$$

$$x_2^s = x_j^{cc} + \left(j - \frac{1}{2}\right)h_2.$$

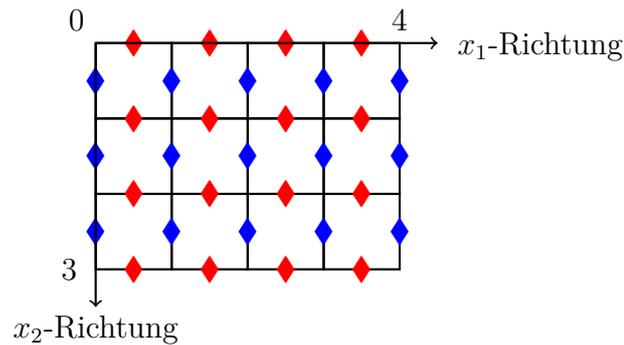


Abbildung 5: Staggered-Gitter für den zweidimensionalen Fall mit $\omega_1 = 4, \omega_2 = 3$; die roten Rauten beschreiben das verschobene Nodal-Gitter und die blauen Rauten das verschobene Cellcentered-Gitter.

3.1.4 Gitterwechsel

Ein Gitterwechsel ist von Nöten, wenn die Daten nicht auf dem selben Gitter vorliegen und sie dennoch verglichen werden sollen [Pap08].

Im *eindimensionalen Fall* wird eine Mittelwertbildung verwendet, um den Wechsel von einem Nodal-Gitter auf ein Cellcentered-Gitter vorzunehmen. Somit ergibt sich ein Gitterwechseloperator W_{cc}^n der Form

$$W_{cc}^n(m) = \frac{1}{2} \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ & & & & & 1 & 1 \end{pmatrix} \in \mathbb{R}^{m,m+1} \quad (4)$$

und

$$x^{cc} = W_{cc}^n(m) \cdot x^n.$$

Im *zweidimensionalen Fall* lässt sich der eindimensionale Gitterwechseloperator mit Hilfe des Kronecker-Produktes in den zweidimensionalen Fall überführen. Sei zur Vereinfachung $W_i = W_{cc}^n(m_i) \in \mathbb{R}^{m_i, m_i+1}$ mit $i = 1, \dots, d$ und $E_d \in \mathbb{R}^{d,d}$ die Einheitsmatrix, somit ergibt sich der Gitterwechseloperator für den zweidimensionalen Fall

$$W_{cc}^n(m_1, m_2) = (E_2 \otimes (W_2 \otimes W_1))$$

und der Gitterwechsel wird beschrieben durch

$$x^{cc} = W_{cc}^n(m_1, m_2) \cdot x^n.$$

Um den Gitterwechsel von einem Staggered- zu einem Cellcentered-Gitter durchzuführen, verwenden wir das Kronecker-Produkt der Einheitsmatrix $E_{m_i} \in \mathbb{R}^{m_i, m_i}$ und der Matrix W_{m_i+1} . In den folgenden Gleichungen repräsentieren die Indizes die jeweilige Richtung bzw. die Größe der Matrix. Somit sind die Gitterwechseloperatoren beschrieben durch

$$W_{x_1} = E_{m_2} \otimes W_{m_1+1} \quad (5)$$

$$W_{x_2} = W_{m_2+1} \otimes E_{m_1} \quad (6)$$

mit $W_1 \in \mathbb{R}^{m_1 m_2, (m_1+1)m_2}$ und $W_2 \in \mathbb{R}^{m_1 m_2, m_1(m_2+1)}$. Der Gitterwechsel lässt sich nun mittels

$$x^{cc} = (E_2 \otimes (W_{x_2} \otimes W_{x_1})) \cdot x^s$$

darstellen.

3.2 Rigide Deformation

Eine rigide Deformation beschreibt eine geometrische Veränderung, die sich für alle Punkte aus dem Gebiet gleich verhält und die Strukturen des Objektes erhält. Sie setzt sich aus einer Rotation und eine Translation zusammen. Diese Art an Deformation werden wir aus den oben genannten Gründen als Vorregistrierung verwenden. Es gibt noch weitere Deformation, jedoch möchten wir nur auf den rigiden Fall näher eingehen, da dies den Rahmen der Arbeit überschreiten würde.

Im *zweidimensionalen Fall* könne wir die Rotation und die Translation zu

$$y = Q(\beta) \cdot x + \gamma, \quad Q \in \mathbb{R}^{2,2}, \quad \gamma, x \in \mathbb{R}^2$$

zusammenfassen. Dabei stellt Q die Rotationsmatrix unter dem Winkel $\beta \in [0, 2\pi)$ dar, d.h.

$$Q = \begin{pmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \end{pmatrix}$$

und γ beschreibt die zwei Translationsparameter.

4 Spline-Interpolation

Bei der Interpolation handelt es sich um ein Verfahren, das für diskrete Daten eine kontinuierliche Funktion sucht, die diese Daten beschreibt. Diese Methode findet ihre Anwendung in der Bildregistrierung, weil Bilder im Allgemeinen nicht als kontinuierliche Funktion, sondern in diskreter Form vorliegen. Damit wir diese Bilder an allen Stellen auswerten können, werden wir sie durch die Wahl einer geeigneten Interpolationsmethode kontinuierlich darstellen. Der Vorteil von dieser Vorgehensweise besteht darin, dass wir die Bilder analytisch ableiten können. Die Ableitung werden wir im weiteren Verlauf für das Gauß-Newton-Optimierungsverfahren benötigen. Ein weiterer Vorzug von der Interpolation ist, dass wir die Bilder in beliebiger Auflösung darstellen können ohne Bildinformationen zu verlieren [Zim08]. Im Fokus, in dieser Arbeit, steht die Interpolation unter Verwendung von linearen und kubischen Splines [Pap08],[Wer92].

Der ursprüngliche Name der Splinefunktionen kommt aus dem Schiffsbau und beschreibt die Spline-Interpolation dritten Grades. Die Namensgebung leitet sich von elastischen Latten (engl. Splines) ab, die an einzelnen Punkten fixiert und somit gebogen wurden.

Das Problem lässt sich folgendermaßen mathematisch beschreiben:

Gegeben ist eine Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$, die an den äquidistant verteilten Stützstellen x_j mit $j = 1, \dots, m$ ausgewertet ist. Gesucht wird eine Funktion $I : \mathbb{R} \rightarrow \mathbb{R}$, die sich als Linearkombination von m Basisfunktionen darstellen lässt [Pap08], d.h.

$$I(x) = \sum_{j=1}^m \alpha_j \cdot b_j(x) \quad (7)$$

$$\text{unter der Nebenbedingung } I(x_j) = f(x_j) \quad (8)$$

$$b_j : \mathbb{R} \rightarrow \mathbb{R}, j = 1, \dots, m. \quad (9)$$

Die Gewichte der einzelnen Basisfunktionen b_j werden durch $\alpha_j \in \mathbb{R}$ beschrieben.

4.1 Lineare Splines

Bei der linearen Interpolation werden die Stützstellen durch stückweise lineare Funktionen verbunden. Hierbei werden die Basisfunktion b_j durch

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

beschrieben.

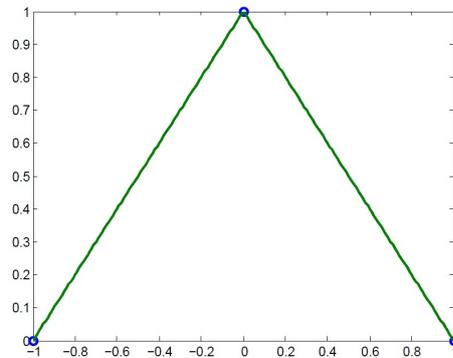


Abbildung 6: Darstellung der Basisfunktion $b_0(x)$

Die Gewichte $\alpha = (\alpha_1, \dots, \alpha_m)$ lassen sich unter Verwendung der Interpolationsbedingung (8) leicht berechnen. Es folgt, dass

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix} \cdot \alpha = \mathbf{f} \text{ für } x \in \mathbb{R}^m \text{ und } \mathbf{f} = (f(x_1), \dots, f(x_m))^T,$$

wodurch sich die Gewichte α direkt ablesen lassen.

4.2 Kubische Splines

Im Gegensatz zu den linearen Splines setzen sich die kubischen Splines aus stückweisen Funktionen vom Grad 3 zusammen [Pap08], [Wer92].

Bei der kubischen Spline-Interpolation wird eine Funktion f verwendet, die zweimal stetig differenzierbar ist und an den äquidistant verteilten Stellen x ausgewertet wird. Um eine möglichst kleine Krümmung der Funktion zu erhalten, wird ein Minimierungsproblem

$$\min_I S^{Spline}(I) \quad (10)$$

$$\text{unter den Nebenbedingungen } I(x_j) = f(x_j), j = 1, \dots, m \quad (11)$$

$$\text{wobei } S^{Spline}(I) = \int_{\Omega} ((I'')^2) dx \quad (12)$$

$$\text{und } I(x) = \sum_{j=1}^m \alpha_j \cdot b_j(x) \quad (13)$$

gelöst. Somit besitzt S^{Spline} eine minimale zweite Ableitung.

Aus [Wer92] ist bekannt, dass mit Hilfe von Randbedingungen das Minimierungsproblem eindeutig lösbar ist. In dieser Arbeit werden wir die natürlichen Randbedingungen, $I''(a) = I''(b) = 0$, verwenden.

Die gesuchte Funktion wird durch (10),(12) und den Randbedingung beschrieben wird, sodass wir die Gewichte α_j durch die Basisfunktion

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

und dessen Ableitung

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

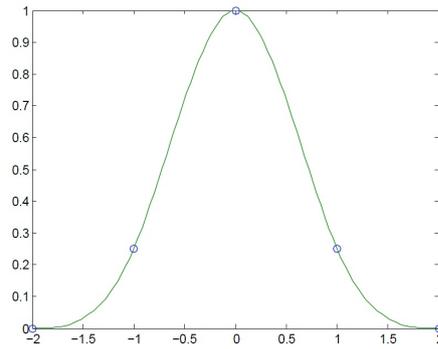


Abbildung 7: Darstellung der Basisfunktion b_0

bestimmen können. Die erste Ableitung wird benötigt, da wir später ableitungsbasierte Optimierungsverfahren verwenden werden, um das Minimierungsproblem zu lösen.

Die Gewichte $\alpha = (\alpha_1, \dots, \alpha_m)$ lassen sich unter Verwendung der Interpolationsbedingung (12) berechnen. Es folgt, dass

$$\underbrace{\begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix}}_M \alpha = \mathbf{f} \text{ für } x \in \mathbb{R}^m \text{ und } \mathbf{f} = (f(x_1), \dots, f(x_m))^T.$$

Die Matrix M ist symmetrisch und diagonaldominant, sodass wir das Inverse von M bilden können. Im Anschluss lassen sich die die Gewichte α mit $M^{-1}\mathbf{f} = \alpha$ eindeutig bestimmen [Wer92].

5 Ähnlichkeit und Distanzmaß

Zwei Bilder sind ähnlich, wenn sie an den gleichen Positionen korrespondierende Informationen enthalten. Diese Eigenschaft lässt sich mittels Distanzmaße \mathcal{D} beschreiben, dabei wird durch eine kleine Distanz, eine große Ähnlichkeit der Bilder ausgedrückt. Im Folgenden werden wir das Maß der Summe der quadrierten Differenzen verwenden. Es kann aber auch durch andere Maße ersetzt werden [Pap08].

Zunächst werden wir erklären, was ein *Bild* ist.

5.1 Bild

Ein Bild \mathcal{B} ist eine stetige differenzierbare Abbildung, die jedem Punkt in einem Gebiet einen Grauwert zuweist [Der11], d.h.

$$\mathcal{B} \in \mathcal{C}^1(\mathbb{R}^d, \mathbb{R}), d \in \mathbb{N}.$$

Im Folgenden werden wir nur noch Bilder der Dimension $d = 2$ betrachten.

Während der Registrierung werden wir das Bild \mathcal{B} unter Verwendung von Gittern, an den Stellen x_j mit $j = 1, \dots, m_1 m_2$ auswerten und somit diskretisieren, d.h. in Vektorschreibweise

$$\mathcal{B}(x_j)_{j=1}^{m_1 m_2} =: B$$

Um das Registrierungsproblem lösen zu können suchen wir eine Deformation y , die das Templatebild \mathcal{T} an das Referenzbild \mathcal{R} angleicht, sodass sich beide Bilder möglichst *ähnlich* sind [Pap08].

Im Folgenden bezeichnen wir das Templatebild \mathcal{T} , welches unter Verwendung von einer Deformation $\mathbf{y} = (y_1(\mathbf{x}), \dots, y_m(\mathbf{x}))$ transformiert wird als deformiertes Templatebild.

$$\mathcal{T}(y_j)_{j=1}^m =: T\mathbf{y}$$

5.2 Summe der quadrierten Differenzen (SSD)

Beim SSD-Maß handelt es sich um ein sehr intuitives Maß der Bildregistrierung. Es wird verwendet, um Bilder gleicher Art, z.B zwei CT-Bilder, aufeinander zu registrieren.

Es werden bei dieser Methode alle Grauwerte des Templatebildes punktweise mit denen vom Referenzbild verglichen, d.h.

$$\mathcal{D}[y] = \frac{1}{2} \int_{\Omega} (\mathcal{T}(y(x)) - \mathcal{R}(x))^2 dx. \quad (14)$$

Sind zwei Bilder nahezu gleich, nähert sich die Differenz Null an. Da wir den *Discretize-then-Optimize-Ansatz* verwenden, wird (14) im Folgenden diskretisiert.

Im *eindimensionalen Fall* sei $\Omega = (0, \omega)$ und die Anzahl an Diskretisierungspunkten durch m gegeben. Zudem sei $x^{cc} = (x_1, \dots, x_m)$ ein Cellcentered-Gitter und $r : \mathbb{R} \rightarrow \mathbb{R}$ [Pap08]. Mit der Rechteckregel [Wer92] folgt

$$\int_{\Omega} r(x) dx \approx h \cdot \sum_{j=1}^m r(x_j),$$

dabei sei $h = \frac{\omega}{m}$ Abstand zwischen den äquidistant verteilten Gitterpunkten. Da das SSD-Maß über eine quadrierte Funktion bestimmt wird, ergibt sich folgende Gleichung:

$$\int_{\Omega} r^2(x) dx \approx h \cdot \sum_{j=1}^m r^2(x_j) = h \cdot \|r(\mathbf{x})\|^2$$

mit $r(\mathbf{x}) = (r(x_1), \dots, r(x_m))^T$.

Wenn wir die Vorgehensweise auf den *zweidimensionalen Fall* übertragen ergibt sich, mit $r(\mathbf{y}) = T\mathbf{y} - R$, folgendes Schema

$$\begin{aligned} \mathcal{D}[y] &= \frac{1}{2} \cdot \int_{\Omega} (\mathcal{T}(y(x)) - \mathcal{R}(x))^2 dx \\ &\approx \frac{h_1 \cdot h_2}{2} \cdot \sum_{j=1}^{m_1 \cdot m_2} (T(y(x_j^{cc})) - R(x_j^{cc}))^2 \\ &= \frac{h_1 \cdot h_2}{2} \cdot \|T\mathbf{y} - R\|^2 \\ &= \frac{h_1 \cdot h_2}{2} \|r(\mathbf{y})\|^2 \\ &=: D[\mathbf{y}] \end{aligned}$$

mit $x_j \in x^{cc}$, wobei $T, R, Ty \in \mathbb{R}^{m_1 m_2}$ und $h_k = \frac{\omega_k}{m_k}$, $k = 1, 2$, entspricht.

Die Ableitungen der Distanz und dem dazugehörigen Maß sind mittels der Kettenregel gegeben durch

$$\begin{aligned}\nabla D[\mathbf{y}] &= h_1 \cdot h_2 \cdot r(\mathbf{y})^T \cdot \nabla r(\mathbf{y}) \\ \nabla r(\mathbf{y}) &= \nabla T\mathbf{y},\end{aligned}$$

mit $\mathbf{y} = (y_1(\mathbf{x}), \dots, y_m(\mathbf{x}))$.

6 Optimierung

Das Registrierungsproblem verwendet einen Optimierungsansatz, um die Bilder anzugleichen. Im Zuge des Discretize-then-Optimize-Ansatzes haben wir unser Problem erst diskretisiert und können nun optimieren. Im folgenden Abschnitt möchten wir das Gauß-Newton-Verfahren mit den Abbruchkriterien von Gill, Murry und Wright und der Armijo-Schrittlängenbestimmung näher beschreiben. Hierbei folgen wir [NW00].

6.1 Gauß-Newton-Verfahren

Das Gauß-Newton-Verfahren ist eine Modifikation des Newton-Verfahrens mit Schrittlängenbestimmung und wird zu den Quasi-Newton-Verfahren gezählt. Dabei sei die Funktion $f \in \mathbb{C}^2(\mathbb{R}^n, \mathbb{R})$ und demnach zweimal differenzierbar. Anwendung findet dieses Verfahren bei Least-Squares Problemen. Daraus folgt:

$$f(\mathbf{y}) = \frac{1}{2} \cdot \|r(\mathbf{y})\|^2 \quad (15)$$

$$\nabla f(\mathbf{y}) = \sum_{j=1}^m r_j(\mathbf{y}) \cdot \nabla r_j(\mathbf{y}) = J(\mathbf{y})^T \cdot r(\mathbf{y}) \quad (16)$$

$$\nabla^2 f(\mathbf{y}) = \sum_{j=1}^m \nabla r_j(\mathbf{y}) \cdot \nabla r_j(\mathbf{y})^T + \sum_{j=1}^m r_j(\mathbf{y}) \cdot \nabla^2 r_j(\mathbf{y}) \quad (17)$$

$$\nabla^2 f(\mathbf{y}) \approx \sum_{j=1}^m \nabla r_j(\mathbf{y}) \cdot \nabla r_j(\mathbf{y})^T = J(\mathbf{y})^T \cdot J(\mathbf{y}). \quad (18)$$

Des Weiteren sei $\mathbf{y} \in \mathbb{R}^n$, die Residuumsfunktion $r : \mathbb{R}^n \rightarrow \mathbb{R}$ und Jakobi-Matrix $\nabla r = J(\mathbf{y}) \in \mathbb{R}^{n,q}$.

Bei dem Gauß-Newton-Verfahren wird die zweite Ableitung (17) durch (18), d.h durch Terme erster Ordnung, approximiert.

Die Iterationvorschrift für das Newtonverfahren besagt:

$$\nabla^2 f(\mathbf{y}^{(k)}) \cdot p^{(k)} = - \nabla f(\mathbf{y}^{(k)}). \quad (19)$$

Durch Einsetzen von (16) und (18) in (19) mit $J^{(k)} := J(\mathbf{y}^{(k)})$, ergibt sich die Iterationvorschrift für das Gauß-Newton-Verfahren

$$J^{(k)T} \cdot J^{(k)} \cdot p^{(k)} = -J^{(k)T} \cdot r(\mathbf{y}^{(k)}) \Leftrightarrow p^{(k)} = -(J^{(k)T} \cdot J^{(k)})^{-1} \cdot J^{(k)T} \cdot r(\mathbf{y}^{(k)}).$$

Dabei kann der Schritt $(J^{(k)T} \cdot J^{(k)})^{-1}$ nur durchgeführt werden, wenn es sich bei dem Produkt der Jakobi-Matrizen um eine reguläre Matrix handelt. Des Weiteren ist zu erwähnen, dass die Funktion ein linear konvergentes Verhalten aufweist [NW00]. Der Algorithmus 1 beschreibt die Vorgehensweise dieses Verfahrens.

Algorithmus 1 Gauß-Newton Algorithmus

```

k=0;
while Abbruchkriterien sind nicht erfüllt do
  Löse  $p^{(k)} = -(J^{(k)T} \cdot J^{(k)})^{-1} \cdot J^{(k)T} \cdot r(\mathbf{y}^{(k)})$ 
  Bestimme Schrittweite  $\alpha$ 
  setze  $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \alpha p^{(k)}$ 
  k = k+1
end while

```

Wenn $p^{(k)} \cdot \nabla f^{(k)} < 0$ kann unter Verwendung der Taylorentwicklung gezeigt werden, dass $p^{(k)}$ eine Abstiegsrichtung beschreibt. Es handelt sich genau dann um eine Abstiegsrichtung, wenn es sich bei Approximation der zweiten Ableitung um eine symmetrisch-positiv-definite Matrix handelt (vgl.[NW00]). Außerdem wird die Schrittweite mit α bezeichnet und ein Minimum von (15) mit Hilfe des Gauß-Newton-Verfahrens bestimmt.

6.2 Abbruchkriterien

Im folgenden Schritt werden Toleranzen eingeführt um sicherzustellen, dass das Verfahren abbricht, wenn das Minimum erreicht wurde. Der Abbruch soll stattfinden, falls sich die Funktionswerte kaum noch verändern, die Norm des Gradienten klein genug sind und die Iterierten sich nicht mehr stark unterscheiden oder die Maschinengenauigkeit unterschritten bzw. die maximale Anzahl an Iterationen überschritten wird.

Mathematisch beschrieben wurden diese Abbruchkriterien von Gill, Murray und Wright [PG81] und lauten folgendermaßen:

1. $\|f(y^{(k-1)}) - f(y^{(k)})\| < \tau \cdot (1 + \|f(y^{(k)})\|)$
2. $\|y^{(k-1)} - y^{(k)}\| < \sqrt{\tau} \cdot (1 + \|y^{(k)}\|)$
3. $\|\nabla f(y^{(k)})\| < \sqrt[3]{\tau} \cdot (1 + \|f(y^{(k)})\|)$
4. $\|\nabla f(y^{(k)})\| < \epsilon$
5. $k > k_{max}$

Dabei beschreibt $\tau \in \mathbb{R}$ den Toleranzwert, $\epsilon \in \mathbb{R}$ die Maschinengenauigkeit und $k_{max} \in \mathbb{N}$ die maximale Anzahl an Iterationsschritten. Der Algorithmus wird abgebrochen, wenn folgendes Kriterium erfüllt ist:

$$((1) \wedge (2) \wedge (3)) \vee (4) \vee (5).$$

6.3 Schrittweitenbestimmung

Um die Schrittweite $\alpha \in \mathbb{R}$ zu bestimmen, werden wir in der vorliegenden Arbeit die Armijo-Schrittweitenbestimmung verwendet (vgl. [NW00]).

Ausgehend von $\alpha = 1$, wird die Armijo-Bedingung,

$$\varphi(\alpha) < \varphi(0) + \epsilon \alpha \varphi'(0),$$

wobei $\epsilon \in (0, 1)$ und $\varphi(\alpha) = f(x + \alpha p^{(k)})$ sei, geprüft. Ist diese erfüllt, wird das α als Schrittweite verwendet. Sollte dies nicht der Fall sein, halbiert sich die Schrittweite, bis die Bedingung erfüllt ist. Weitere Informationen zu der Schrittweitenbestimmung gibt es in [NW00]. Der Algorithmus 2 beschreibt den Ablauf der Schrittweitenbestimmung.

Algorithmus 2 Armijo-Liniensuche

```

 $\alpha = 1$ 
while  $\varphi(\alpha) \geq \varphi(0) + \epsilon \alpha \varphi'(0)$  do
     $\alpha \leftarrow \frac{\alpha}{2}$ 
end while

```

Mit Hilfe des Gauß-Newton-Optimierungsverfahrens lässt sich ein Minimum der Funktion finden. Jedoch ist bis jetzt noch nicht sichergestellt, dass es sich bei dem gefundenen Wert um einen globalen Minimierer handelt. Um lokale Minima ausschließen zu können, ist es notwendig einen Startwert in der Nähe des globalen Minimums zu wählen. Im nächsten Kapitel wollen wir auf Verfahren eingehen die lokale Minima ausschließen.

7 Multiresolutions- und Multiskalenverfahren

Bei der Verwendung von Optimierungsverfahren, ist die Wahl eines geeigneten Startwerts eine wichtige Komponente, um lokale Minima zu vermeiden. Im Folgenden wollen wir auf zwei Methoden näher eingehen, die sich mit dieser Problematik beschäftigen. Zum Einen handelt es sich um das diskrete Multiresolutionsverfahren und zum Anderen um den kontinuierlichen Multiskalen-Ansatz [Mod09],[Pap08].

7.1 Multiresolutionsverfahren

Ein Multiresolutionsverfahren realisiert ein Verfahren der Bildregistrierung, bei dem zunächst die Bilder unter einer groben Auflösung registriert werden. In diesem Kontext beschreibt eine grobe Auflösung eine geringe Anzahl an Diskretisierungspunkten m . Da im Folgenden die Auflösung halbiert bzw. verdoppelt wird, sei zur Vereinfachung $m = 2^k$.

Die Lösung dieses Teilproblems dient anschließend als Startwert für die neue Registrierung unter einer höheren Auflösung. Diese Vorgehensweise wird solange durchgeführt, bis die gewünschte Auflösung erreicht ist. Da die Daten an die Auflösungsstufen durch Restriktion und Prolongation angepasst werden, erhalten wir in den groben Auflösungen glatte Zielfunktionen.

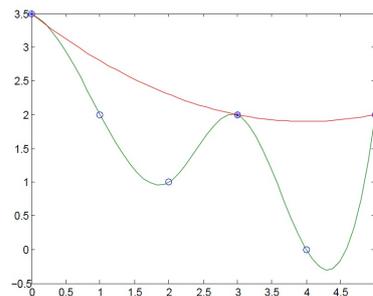


Abbildung 8: Darstellung einer Funktion, die an sechs Stellen (grün) und an drei Stellen (rot) ausgewertet wurde

Diese Methode hat den Vorteil, dass wir für den nächsten Schritt einen Startwert erhalten, der näher an der gesuchten Lösung liegt. Zudem können lokale Minima durch eine grobe Auflösung vermieden werden. Aus der Glattheit lässt sich folgern, dass die Funktion im Idealfall wenige bis keine lokalen Minima

aufweist. Aus dieser Eigenschaft folgt, dass unter Verwendung dieses Verfahrens die Registrierung schneller und effizienter wird.

Die Abb. 8 veranschaulicht, die Glättung der Zielfunktion und die Vorgehensweise des Multiresolutionsverfahrens. Die grüne Funktion wurde an sechs Stellen ausgewertet und hat zwei lokale Minima. Wenn wir die Auswertung an halb so vielen Stellen durchführen, erhalten wir eine glatter Funktion, die nur ein Minimum ausweist. Das globale Minimum, der roten Funktion, kann nun verwendet werden um das globale Minimum der grünen Funktion zu berechnen.

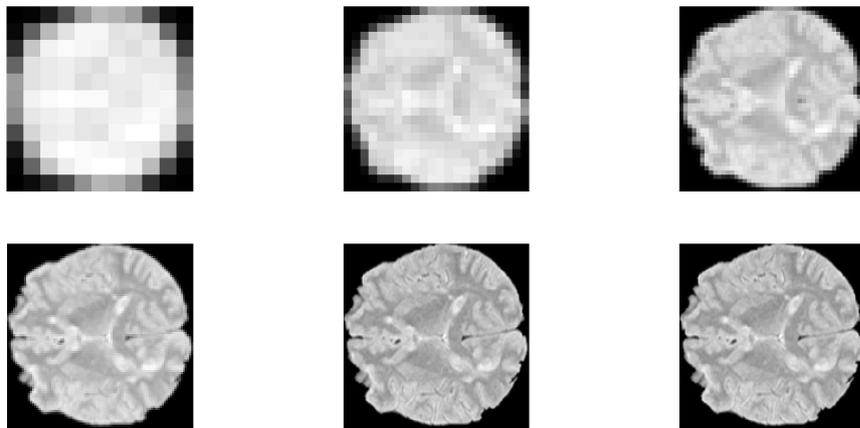


Abbildung 9: Multiresolutionsverfahren. Von oben links nach unten rechts: Ansteigendes $m = (2^3, 2^3), (2^4, 2^4), \dots, (2^8, 2^8)$

7.2 Restriktion und Prolongation

Für das oben genannte Verfahren werden Operatoren benötigt, die einen Gitterwechsel von einer groben auf eine feinere Auflösung und umgekehrt realisieren. Zunächst werden wir diese für den eindimensionalen Fall betrachtet, um sie anschließend auf den zweidimensionalen Fall zu übertragen [Pap08], [Zim08].

Im weiteren Verlauf sei $P_h^H : \Omega_h \rightarrow \Omega_{\frac{h}{2}}$ der Operator, der die Transformation von einem groben auf ein feineres Gitter realisiert. Dieser Vorgang wird auch als Prolongation bezeichnet. Unter Restriktion verstehen wir den Wechsel von einem feinen auf ein gröberes Gitter. Der dazugehörige Operator sei $P_H^h : \Omega_h \rightarrow \Omega_{2h}$.

Im *eindimensionalen Fall* sei das Intervall $\Omega = (0, \omega)$ wieder in $m = 2^k$ Zellen der Länge $h = \frac{\omega}{m}$ unterteilt. Die Daten seien zudem auf einem Nodal-Gitter ($x^n \in \mathbb{R}^{m+1}$) gegeben. Um Informationsverlust zu vermeiden soll während des Wechsels die Masse der kontinuierlichen Funktion $y : \Omega \rightarrow \mathbb{R}$ erhalten bleiben, welche durch $\int_{\Omega} y(x) dx$ beschrieben wird. Das Integral kann mit Hilfe der Trapezformel [Wer92] wie folgt diskretisiert werden:

$$\begin{aligned} \int_{\Omega} y(x) dx &\approx \frac{h}{2} (y_1 + 2 \cdot \sum_{i=2}^{\omega-1} y_i + y_{m+1}) \\ &= h \left(\frac{1}{2}, e^T, \frac{1}{2} \right) \mathbf{y} \end{aligned}$$

mit $\mathbf{y} = (y_1(x), \dots, y_{m+1}(x)) \in \mathbb{R}^{m+1}$.

Es gibt viele verschiedene Art und Weisen eine Restriktion und eine Prolongation durchzuführen. Wenn wir die Massenerhaltung fordern, muss für die Operatoren die Gleichung

$$2h \left(\frac{1}{2}, e^T, \frac{1}{2} \right) P_H^h(m) \mathbf{y} = h \left(\frac{1}{2}, e^T, \frac{1}{2} \right) \mathbf{y} = \frac{h}{2} \left(\frac{1}{2}, e^T, \frac{1}{2} \right) P_h^H(m) \mathbf{y}$$

erfüllt sein. Auf diese Weise entspricht $P_H^h(m) \mathbf{y} \in \mathbb{R}^{2m+1}$ halb so vielen Diskretisierungspunkten von \mathbf{y} . Somit ergibt sich die Gitterweite von $2h$. Doppelt so viele Werte von \mathbf{y} weist $P_h^H(m) \mathbf{y} \in \mathbb{R}^{\frac{m}{2}+1}$ auf und die Weite des Gitters ist daher $\frac{h}{2}$.

Im *zweidimensionalen Fall* wird mittels des Kronecker-Produkts der Gitterwechsel beschrieben. Ebenso wird das Intervall $\Omega = (0, \omega_1) \times (0, \omega_2)$ in $m = (2^{k_1}, 2^{k_2})$ Zellen unterteilt.

Die Prolongation wird nun dargestellt durch

$$P_h^H(m_1, m_2) = (P_h^H(m_2) \otimes E(2m_1 + 1))(E(m_2 + 1) \otimes P_h^H(m_1)),$$

wobei das neue Gitter anschließend $m = (2m_1 + 1, 2m_2 + 1)$ Diskretisierungspunkte beinhaltet und die Restriktion lautet

$$P_h^h(m_1, m_2) = (P_h^h(m_2) \otimes E(\frac{m_1}{2} + 1))(E(m_2 + 1) \otimes P_h^h(m_1))$$

mit der Auflösung $m = (\frac{m_1}{2} + 1, \frac{m_2}{2} + 1)$.

7.3 Multiskalen-Ansatz

Ein weiteres Verfahren um lokale Minima zu vermeiden ist der Multiskalen-Ansatz. Er unterscheidet sich vom Multiresolutionsverfahren in der Art, dass bei diesem Ansatz keine Veränderung der diskreten Bildauflösung vorgenommen, sondern vielmehr die Bildinformation verringert wird [Pap08].

Skalenräume beschreiben eine Folge von Bildern, bei denen der Informationsgehalt des Bilder kontinuierlich bzgl. eines Parameters reduziert wurde. Analog zum Multiresolutionsverfahren verwendet der Multiskalenansatz zuerst eine grobe Skala, d.h. einen hohen Wert für Skalenraumparameter. Die Lösung dieses Teilproblems dient im Anschluss als Startwert für die neue Registrierung unter einer feineren Skala. Diese Vorgehensweise wird solange wiederholt bis die gewünschte Skala erreicht ist. Der Vorteil dieser Methode ist, dass sie auflösungsunabhängig ist und deshalb keine Prolongation bzw. Restriktion benötigt wird.

Die Abb. 11 zeigt eine Glättung mit einer Funktion mittels eines Gauß-Filters. Es wurde die Standardabweichung $\sigma = 2$ verwendet.

Gaußscher Skalenraum: Bei dem Gaußschen Skalenraum handelt es sich um eine Menge an Bildern, die durch Faltung eines Bildes mit dem Gaußfilter, unter Verwendung verschiedener Skalenraumparameter σ^2 , entstanden sind. Die Abb. 12 veranschaulicht einen Gaußschen Skalenraum mit steigendem σ . Hierbei haben wir die Gleichung

$$g(T, N_{\sigma^2}) = T * N_{\sigma^2}$$

für $\sigma = 2, 4, 8, 16, 32, 64$ gelöst, wobei die zweidimensionale Gauß'sche Normalverteilung die Form

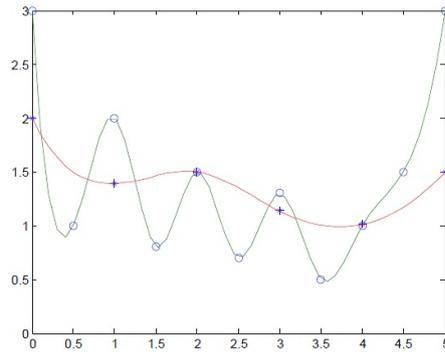


Abbildung 11: Darstellung einer Funktion, die an sechs Stellen (grün) und durch einen Gauß-Filter mit der Standardabweichung $\sigma = 2$ geglättet wurde (rot).

$$N_{\sigma^2}(x_1, x_2) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_1^2 + x_2^2)}{2\sigma^2}\right)$$

hat. Es ist deutlich zu erkennen, wie die Gehirnstrukturen mit steigendem σ über das Bild verschmiert werden. Durch diese Glättung kann Rauschen unterdrückt werden. Für den Umgang mit den Randpunkten gibt es unterschiedliche Herangehensweisen. Da MRT-Bilder im Allgemeinen einen schwarzen Rand haben wurden die Randpunkte auf Null gesetzt.

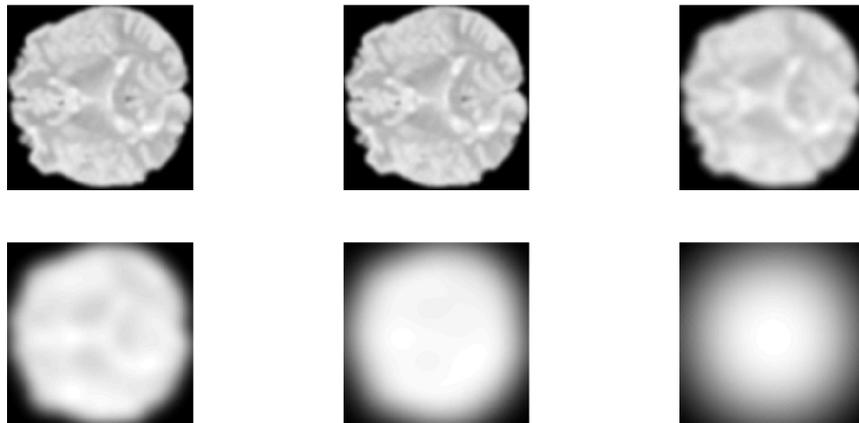


Abbildung 12: Multiskalen-Ansatz mit einer Darstellung des Gaußschen Skalenraums. Von oben links nach unten rechts: Ansteigende Varianz, $\sigma = 2, 4, 8, 16, 32, 64$.

8 Diffusive Regularisierung

Bei der nichtlinearen Bildregistrierung wird, im Gegensatz zum rigiden Ansatz, eine Deformation verwendet, die lokal unterschiedlich ist. Im Folgenden werden wir beschreiben, warum es sich bei der Bildregistrierung um ein schlecht gestelltes Problem handelt.

Nach Hadamard [Had02] ist ein Problem gut gestellt, wenn

1. eine Lösung existiert (Existenz),
2. genau eine Lösung existiert (Eindeutigkeit),
3. die Lösung stetig von den Eingangsdaten abhängt (Stabilität).

Ist mindestens eine Bedingung nicht erfüllt, handelt es sich um ein schlecht gestelltes Problem.

Bei der Bildregistrierung ist die Eindeutigkeit der Lösung nicht sichergestellt, da z.B. in einem schwarz-weiß Bild jeder schwarze Pixel beliebig auf einen anderen schwarzen Pixel registriert werden kann.

Um dieses schlecht gestellte Problem in ein gut gestelltes zu konvertieren, verwenden wir einen Regularisierer. Dieser wird mit Hilfe von zusätzlichen Bedingungen, z.B. Ableitungen, die Glattheit der Funktion verändern. Dieses Merkmal wird auch Glattheitsanforderung genannt, weshalb der Regularisierer auch als Glätter (engl. Smoother) S bezeichnet wird. Die Gewichtung dieses Regularisierers erfolgt über den Parameter $\alpha \in \mathbb{R}$, der auch Regularisierungsparameter genannt wird.

Nun kann die aus (1) bekannte Minimierungsproblem durch einen Term S ergänzt werden. Wir erhalten:

$$\min_y \mathcal{D}[y] + \alpha \mathcal{S}(y), \quad (22)$$

wobei \mathcal{D} ein Distanzmaß beschreibt.

8.1 Kontinuierliche Formulierung

Nicht nur die Wahl des Distanz-Maßes, sondern auch die Wahl des Regularisierers ist abhängig von dem Anwendungsproblem. In dieser Arbeit möchten wir einen einfachen Regularisierer verwenden, der eine Glattheit der Deformation y fordert, indem er große Ableitungen bestraft. Der sogenannte diffusive Regularisierer S^{diff} fand 2001 durch Fischer und Modersitzki [FM01], [FM02] Anwendung in der Bildregistrierung. Er lässt sich wie folgt formulieren,

$$S^{diff}(y) = \frac{1}{2} \cdot \int_{\Omega} \sum_{j=1}^d \|\nabla y_j\|^2 dx. \quad (23)$$

8.2 Diskrete Formulierung

Genauso wie bei der Interpolation, führen wir die Diskretisierung als erstes im *eindimensionalen Fall* durch [Pap08].

Sei die Funktion $y : \Omega \rightarrow \mathbb{R}$ und $y \in C^2$. Des Weiteren sei das Intervall $\Omega = (0, \omega) \in \mathbb{R}$ in m Zellen mit der Länge $h = \frac{\omega}{m}$ unterteilt. Die Daten seien auf einem Nodal-Gitter x^n gegeben und sollen auf einem Cellcentered-Gitter x^{cc} ausgewertet werden, was bedeutet, dass $x_i^n = x_i^{cc} - h/2$ und $x_{i+1}^n = x_i^{cc} + h/2$ entsprechen. Die erste Ableitung der Funktion an einer Stelle $i = 1, \dots, m$ erhalten wir unter Zuhilfenahme des Satzes von Taylor, d.h.

$$y(x_i^{cc} + h/2) \approx y(x_i^{cc}) + \frac{h}{2} y'(x_i^{cc}) \quad (24)$$

$$y(x_i^{cc} - h/2) \approx y(x_i^{cc}) - \frac{h}{2} y'(x_i^{cc}). \quad (25)$$

Mittels Subtraktion von (24) und (25) ergibt sich die Approximation

$$y'(x_i^{cc}) \approx \frac{1}{h} (-y(x_i^{cc} - h/2) + y(x_i^{cc} + h/2)). \quad (26)$$

Nun wollen wir (26) in Matrixschreibweise darstellen. Für diese Beschreibung sei $A_{m+1} \cdot y(x^n)$ die Approximation der erste Ableitung der Funktion y . Unter Verwendung von

$$A_{m+1} = \frac{1}{h} \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{m,m+1} \quad (27)$$

folgt

$$A_{m+1} \cdot y(x^n) \approx y'(x^c).$$

Für den *zweidimensionalen Fall* resultiert eine Ableitungsmatrix (27) in x_1 - und x_2 - Richtung auf dem Gebiet $\Omega = (0, \omega_1) \times (0, \omega_2)$, die durch die Operatoren

$$A_{x_1} = \frac{1}{h_1} (E_{m_2+1} \otimes A_{m_1+1}) \quad (28)$$

$$A_{x_2} = \frac{1}{h_2} (A_{m_2+1} \otimes E_{m_1+1}) \quad (29)$$

ausgedrückt werden, und somit wird die Ableitung auf einem Staggered-Gitter beschrieben [Der11].

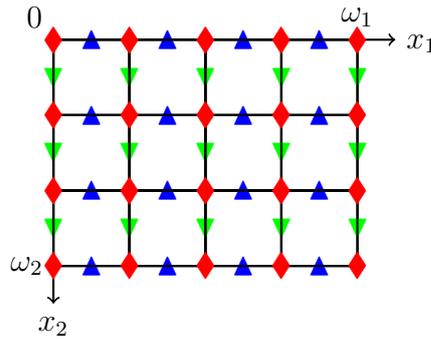


Abbildung 13: Veranschaulichung der Diskretisierung im Fall $d = 2$.

Hierbei ist $\omega_1 = 4, \omega_2 = 3$. Die roten Rauten beschreiben das ursprüngliche Nodal-Gitter. Die blauen Dreiecke stellen die Ableitung in x_1 - und die grünen Dreiecke die Ableitung in x_2 - Richtung dar.

Um die Diskretisierung weiter durchführen zu können, werden nun die Ableitungsmatrizen (28) und (29) verwendet. Außerdem werden die Terme quadriert, um dann unter Verwendung der Wechseloperatoren (5) und (6) die Zellmittelpunkte zu beschreiben. Zudem sei $e = (1, \dots, 1)^T \in \mathbb{R}^{m_1 m_2}$ und $\text{diag}(x)$ stellt eine Diagonalmatrix

$$\text{diag}(x) := \begin{pmatrix} x_1 & & \\ & \ddots & \\ & & x_m \end{pmatrix} \in \mathbb{R}^{m,m}$$

dar. Somit ergibt sich mittels der Rechteckregel und $h = h_1 \cdot h_2$

$$\begin{aligned} \mathcal{S}^{diff}(y) &= \frac{1}{2} \cdot \int_{\Omega} (\|\nabla y_1\|^2 + \|\nabla y_2\|^2) dx \\ &\approx \frac{h}{2} \cdot (e^T W_{x_2} \text{diag}(A_{x_1} y_1) A_{x_1} y_1 + e^T W_{x_1} \text{diag}(A_{x_2} y_1) A_{x_2} y_1 \\ &\quad + e^T W_{x_2} \text{diag}(A_{x_1} y_2) A_{x_1} y_2 + e^T W_{x_1} \text{diag}(A_{x_2} y_2) A_{x_2} y_2). \end{aligned}$$

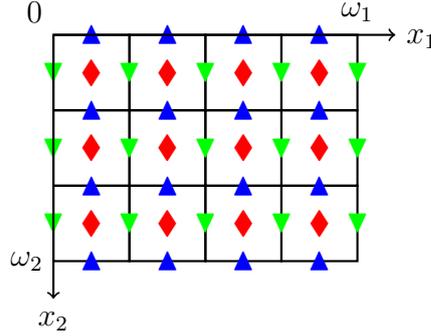


Abbildung 14: Veranschaulichung der Diskretisierung im Fall $d = 2$.

Hierbei ist $\omega_1 = 4, \omega_2 = 3$. Die blauen Dreiecke stellen die Ableitung in x_1 - und die grünen Dreiecke die Ableitung in x_2 - Richtung dar. Die roten Rauten beschreiben das entstandene Cellcentered-Gitter.

Unter Verwendung der Eigenschaft, dass $(e^T W_{x_1})^T = W_{x_1}^T e$ gilt, folgt

$$\begin{aligned} \mathcal{S}^{diff}(y) &\approx \frac{h}{2} \cdot (y_1^T A_{x_1}^T \text{diag}(W_{x_2}^T e) A_{x_1} y_1 + y_1^T A_{x_2}^T \text{diag}(W_{x_1}^T e) A_{x_2} y_1 \\ &\quad + y_2^T A_{x_1}^T \text{diag}(W_{x_2}^T e) A_{x_1} y_2 + y_2^T A_{x_2}^T \text{diag}(W_{x_1}^T e) A_{x_2} y_2) \\ &= \frac{h}{2} \cdot [y_1^T (A_{x_1}^T \text{diag}(W_{x_2}^T e) A_{x_1} + A_{x_2}^T \text{diag}(W_{x_1}^T e) A_{x_2}) y_1 \\ &\quad + y_2^T (A_{x_1}^T \text{diag}(W_{x_2}^T e) A_{x_1} + A_{x_2}^T \text{diag}(W_{x_1}^T e) A_{x_2}) y_2] \end{aligned}$$

und mit $C = A_{x_1}^T \text{diag}(W_{x_2}^T e) A_{x_1} + A_{x_2}^T \text{diag}(W_{x_1}^T e) A_{x_2}$ lässt sich die Gleichung nun wie folgt schreiben als

$$\mathcal{S}^{diff}(y) \approx \frac{h}{2} \cdot (y_1^T \cdot C \cdot y_1 + y_2^T \cdot C \cdot y_2) = \frac{h}{2} (\|C \cdot y_1\|^2 + \|C \cdot y_2\|^2).$$

Unter Verwendung von $y = (y_1, y_2)^T$ und $\tilde{C} = \text{diag}(C)$, ergibt sich schließlich

$$\mathcal{S}^{diff}(y) \approx \frac{h}{2} \cdot (y^T \cdot \tilde{C} \cdot y) = \mathbf{S}^{diff}(y) \quad (30)$$

als Diskretisierung.

Die Ableitung des Regularisierers lautet somit,

$$\begin{aligned}\nabla \mathbf{S}^{diff}(y) &= h \cdot \tilde{\mathbf{C}}^{1/2} \cdot \tilde{\mathbf{C}}^{1/2} \cdot y \\ &= h \cdot \tilde{\mathbf{C}} \cdot y\end{aligned}$$

Der Algorithmus 3 fasst alle Bestandteile der nicht-linearen Registrierung noch einmal zusammen.

Algorithmus 3 Nicht-lineare Registrierung

Eingabe: R, T, Ω, m, α

Ausgabe: y^{opt}

Rigide Vorregistrierung der Daten:

$$y^0 = [0, 0, 0];$$

for Für alle Level bzw. Skalen **do**

$$y_{rigide}^{opt} \leftarrow \text{GN}(\mathcal{D}(T, R; y), y^0);$$

$$y^0 = y_{rigide}^{opt};$$

end for

Nicht-lineare Registrierung der Daten:

$$y^0 = y_{rigide}^{opt};$$

for Für alle Level bzw. Skalen **do**

$$y^{opt} \leftarrow \text{GN}(\mathcal{D}(T, R; y) + \alpha S(y - y_{rigide}^{opt}), y^0);$$

$$y^0 = y^{opt};$$

if Registrierung mit Multilevel-Ansatz und das Level ist nicht die maximale Auflösung **then**

 Prolongation

end if

end for

9 Ergebnisse

Im folgenden Kapitel werden die Ergebnisse vorgestellt, die durch Anwendung der nicht-linearen Registrierung entstanden sind. Dabei wurde bei jeder Auswertung entweder der Multilevel- und/oder der Multiskalen-Ansatz verwendet und alle Daten rigide vorregistriert.

Für die Registrierung wurde zudem das SSD- Distanzmaß und der diffusive Regularisierer verwendet. Des Weiteren kam zur Optimierung der Gauß-Newton Algorithmus mit der Armijo-Schrittlängenbestimmung zur Anwendung. Darüber hinaus wurde die maximale Anzahl an Iterationsschritten auf $k_{max} = 100$ beschränkt. Die Maschinengenauigkeit beträgt $\epsilon = 10^{-9}$ und der Toleranzwert haben wir auf $\tau = 10^{-2}$ gesetzt.

Der Schwerpunkt dieser Arbeit liegt auf der Erarbeitung der nicht-linearen Registrierung und den Testläufen auf synthetischen und realen Daten. Dabei wurden z.B. der Regularisierer, das Distanzmaß und die Gitterwechseloperatoren eigenverantwortlich in Matlab implementiert. Andere Methoden, unter anderem die einzelnen Gitter, wurden der FAIR-Toolbox von Prof. Jan Mordersitzki entnommen.

9.1 Synthetische Daten

Um die nichtlineare Registrierung zu testen, wurden zunächst synthetische Daten auf $\Omega = (0, 127) \times (0, 127)$ erzeugt. Je nach Registrierungsproblem schwankt der Regularisierungsparameter α zwischen 1 und 10^4 .

9.1.1 Registrierung von zwei Kreisen unterschiedlicher Größe

Der erste Test, der durchgeführt wurde, ist ein klassisches Registrierungsproblem. Es befasste sich mit der Registrierung von einem kleinen auf einen großen Kreis und dem dazugehörigen Deformationsfeld. Um den Algorithmus zu testen, wurden die Bilder unter Verwendung des Multilevel- und Multiskalen-Ansatzes registriert. Des Weiteren wurde der gleiche Gewichtungsparemeter α für beide Ansätze verwendet. Dieser Parameter ist bei der nicht-linearen Registrierung frei wählbar und wurde auf $\alpha = 100$ gesetzt. Wie sich die Wahl des Gewichtungsparemters auf die Registrierung auswirkt, wollen wir in Abschnitt 9.1.2 erläutern. Zunächst möchten wir auf die Ergebnisse unter Verwendung des Multilevel-Ansatzes eingehen, um sie danach mit den Ergebnissen des Multiskalen-Ansatzes vergleichen zu können.

Der Multilevel-Ansatz verwendet für die Registrierung Auflösungen mit $m = (2^4, 2^4), \dots, (2^7, 2^7)$ Diskretisierungspunkte. Die Abb. 15 veranschaulicht die Ergebnisse. Wie sehen ein deformiertes Gitter, dass in der Mitte zusammengezogen wurde. Das Verhalten ist dadurch begründet, dass wenn wir das Gitter auf den Template anwenden, sich dieses ausdehnt und sich somit der Kreis vergrößert. Da das Gitter keine Faltungen aufweist, können wir hier von einem glatten Gitter sprechen und davon ausgehen, dass der Algorithmus plausible Ergebnisse liefert.

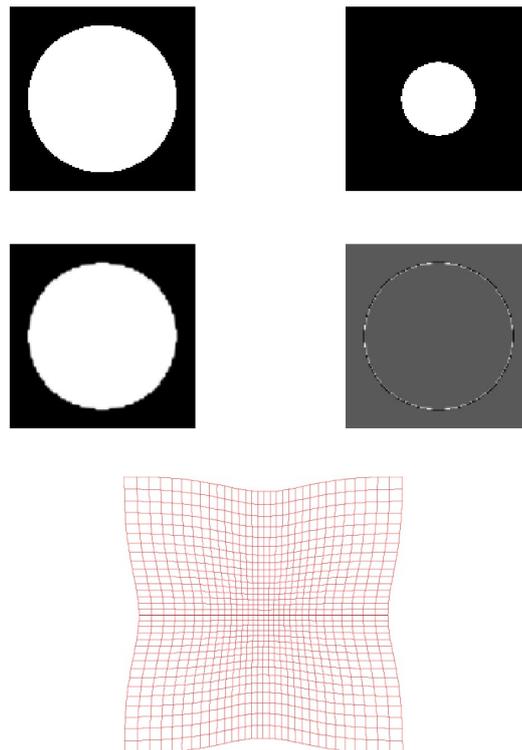


Abbildung 15: Nichtlineare Registrierung von einem kleinen auf einen großen Kreis mit Multilevel-Ansatz . Von oben links nach unten rechts: Referenzbild, Template, deformiertes Template, Differenzbild von Referenz- und deformierten Templatebild, deformiertes Gitter

Bei dem Multiskalen-Ansatz kommen unterschiedliche Werte für die Standardabweichung σ der Normalverteilung zum Einsatz. Wir haben hierfür die Werte $\sigma = 4, 2, 1$ gewählt. Die Abb. 16 stellt die Ergebnisse dieser Registrierung dar, die wir nun mit dem Multilevel-Ansatz vergleichen möchten, um die Vor- und Nachteile der Ansätze zu verstehen.

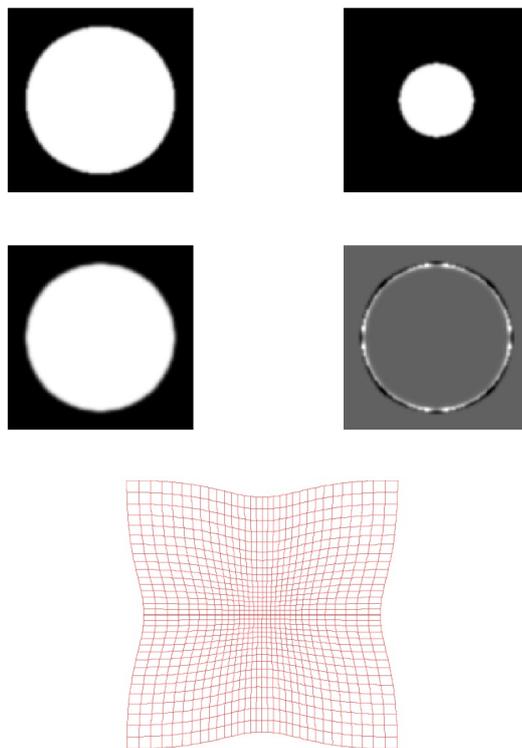


Abbildung 16: Nichtlineare Registrierung von einem kleinen auf einen großen Kreis mit Multiskalen-Ansatz. *Von oben links nach unten rechts:* Referenzbild, Template, deformiertes Template, Differenzbild von Referenz- und deformierten Templatebild, deformiertes Gitter

Obwohl die deformierten Gitter ähnlich aussehen, ist der Rand des Kreises beim Multiskalen-Ansatz (vgl. Abb. 16) deutlicher zu erkennen als bei dem Multilevel-Ansatz (vgl. Abb. 15). Der Multiskalen-Ansatz verschmiert die Bildinformation über das Bild, was zu einer Glättung der Kanten führt. Das Ergebnis lässt darauf schließen, dass die Ränder während der Registrierung nicht vollkommen wieder hergestellt werden können. Bei dem Multilevel-Ansatz hingegen wird die Information durch Restriktion und Prolongation von einem auf ein anderes Level gebracht und auf Grund der Massenerhaltung bleiben die Ränder besser erhalten.

Eine weitere Möglichkeit diese beiden Ansätze miteinander zu vergleichen ist über die Anzahl an Iterationen. Der Multilevel-Ansatz benötigt bei der Auflösungsstufe $m = (2^4, 2^4)$ und $m = (2^6, 2^6)$ fünf, bei $m = (2^5, 2^5)$ acht und bei $m = (2^7, 2^7)$ zehn Iterationsschritte um ein Optimum zu erreichen. Wir können erkennen, dass entgegen der intuitiven Annahme, dass die Anzahl an Iterationsschritten mit steigendem m abnimmt, für dieses Beispiel nicht gilt. Dieses Verhalten weist hingegen der Multiskalen-Ansatz auf. Die Anzahl an Iterationsschritten beträgt bei diesem Ansatz für die Standardabweichung $\sigma = 4$ acht, für $\sigma = 2$ vier und für $\sigma = 1$ drei.

Neben diesen Unterschieden finden wir auch Abweichung in der Laufzeit. Die Rechenzeit, bei einem Computer mit einem Dual Core Prozessor mit 1,3 GHz und 4 GB RAM, betrug für den Multilevel-Ansatz 0.83 min und für den Multiskalen-Ansatz 1.04 min. Die längere Laufzeit lässt sich auf die fehlende Auflösungsreduktion bei dem Multiskalen-Ansatz zurückführen. Auf Grund der oben genannten Vorteile des Multilevel-Ansatzes werden, wir für die den nächsten Abschnitt nur diesen Ansatz verwenden.

9.1.2 Registrierung von einem kleinen Kreis auf den Buchstaben C

Dieser Test befasste sich mit der Registrierung von einem großen Kreis auf den Buchstaben C und dem dazugehörigen Deformationsfeld. Zudem wollen wir uns den Einfluss des Gewichtungsparameters α auf die Registrierung anschauen.

In Abb.17 ist zu entnehmen, dass obwohl der Kreis nun gut auf das C registriert wurde, das Deformationsfeld unerwünschte Faltungen aufweist. Diese Eigenschaft ist sichtbar, wenn die Gewichtung zu stark auf dem Distanzmaß liegt und der Regularisierer keinen großen Einfluss hat.

Im Folgenden soll gezeigt werden, wie sich die Deformationsgitter und das deformierte Templatebild verändern, wenn verschiedene Werte für den Gewichtungsparameter α verwendet werden (vgl. Abb. 18). Je kleiner der Parameter gewählt wird, desto mehr entspricht das Resultat dem Buchstaben. Nimmt α einen relativ großen Wert an, verändert sich das deformierte Template nicht so stark bzgl. des Templatebildes. Dieses Verhalten lässt sich auch anhand des deformierten Gitters erkennen. Je kleiner α gewählt wird, desto größer ist die unerschwünschte Faltung.

Wir können sagen, dass die Wahl des Gewichtungsparameters ein Abwiegen zwischen einer exakten Registrierung und einer Glattheit des deformierten Gitters ist. Dies führt zu der Schlussfolgerung, dass diese Wahl je nach Registrierungsproblem unterschiedlich ist und kein allgemeingültiger Wert angegeben werden kann.

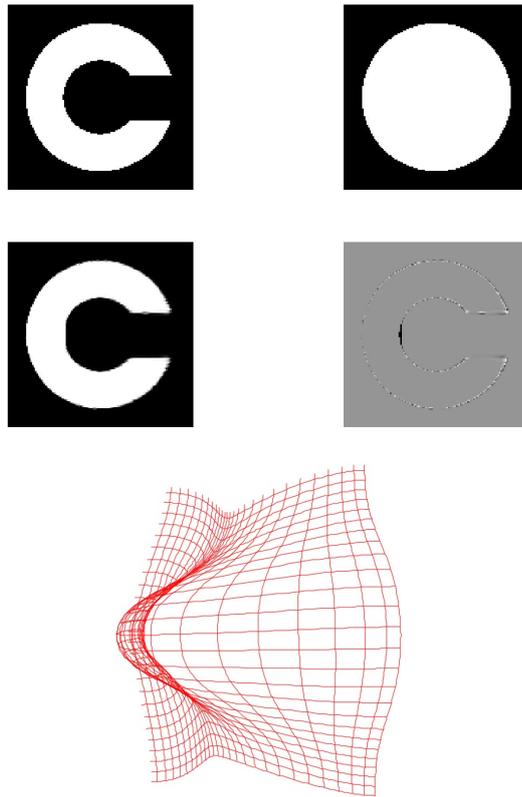


Abbildung 17: Nichtlineare Registrierung von einem Kreis auf den Buchstaben C. Von oben links nach unten rechts: Referenzbild, Template, deformiertes Template, Differenzbild vom Referenz- und deformierten Templatebild, Deformationsgitter. Als Gewichtungsparemeter wurde $\alpha = 1$ gewählt.

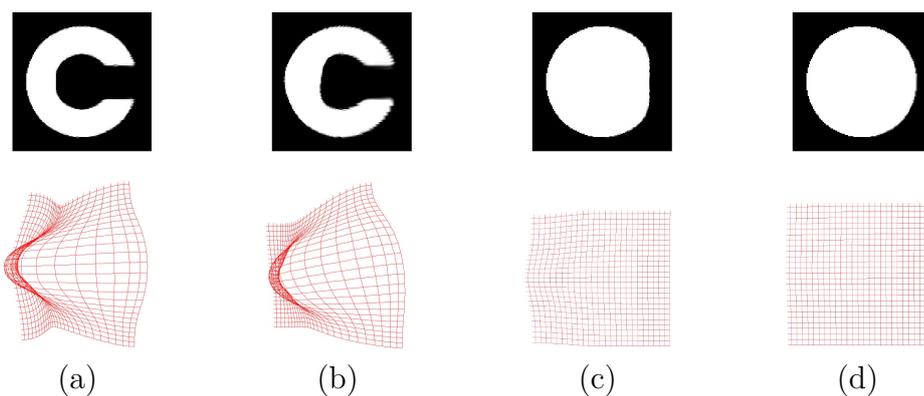


Abbildung 18: Nicht-lineare Registrierung unter Verwendung unterschiedlicher Gewichtungsparemeter α . (a) deformiertes Templatebild und das Gitter für $\alpha = 1$, (b) $\alpha = 10$, (c) $\alpha = 100$, (d) $\alpha = 1000$

9.2 MRT-Daten

Nachdem die nicht-lineare Registrierung auf die synthetischen Daten angewendet wurde, werden im Folgenden auf MRT-Daten arbeiten. Die Daten zeigen einen Transversalschnitt des Kopfes in einer T2-Gewichtung, wobei der Knochen vor der Registrierung entfernt wurde. Die beiden Originaldatensätze haben eine Größe von $m = (140, 176, 49)$ Voxeln und stammt von Dr. Charles Guttmann von dem Center für neuronale Bildgebung in Brigham und der Harvard Medical School, Boston. Sie stammen von dem gleichen Patienten und wurden im Abstand von einer Woche aufgenommen. Für die nicht-lineare Registrierung, unter Verwendung des Multilevel- und Multiskalen-Ansatzes, wurde die Schnittebene 28 verwendet und die Daten durch Interpolation auf das quadratische Format $m = (176, 176)$ gebracht. Als Regularisierungsparameter hat nach einigem probieren $\alpha = 10^8$ als gute Wahl herausgestellt, da wir dadurch ein glattes Gitter erhalten.

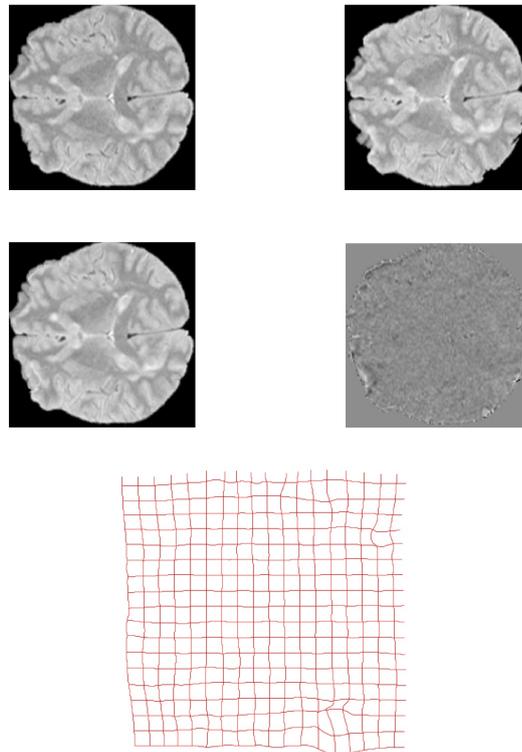


Abbildung 19: Nichtlineare Registrierung von zwei MRT-Bildern unter Verwendung des Multilevel-Ansatzes. *Von oben links nach unten rechts*: Referenzbild, Template, deformiertes Template, Differenzbild von Referenz- und deformierten Templatebild.

Analog zu den synthetischen Daten, werden wir erst auf die Registrierung unter Verwendung des Multilevel-Ansatzes (Abb. 19) eingehen, um sie danach mit den Ergebnissen des Multiskalen-Ansatzes (Abb. 20) vergleichen zu können.

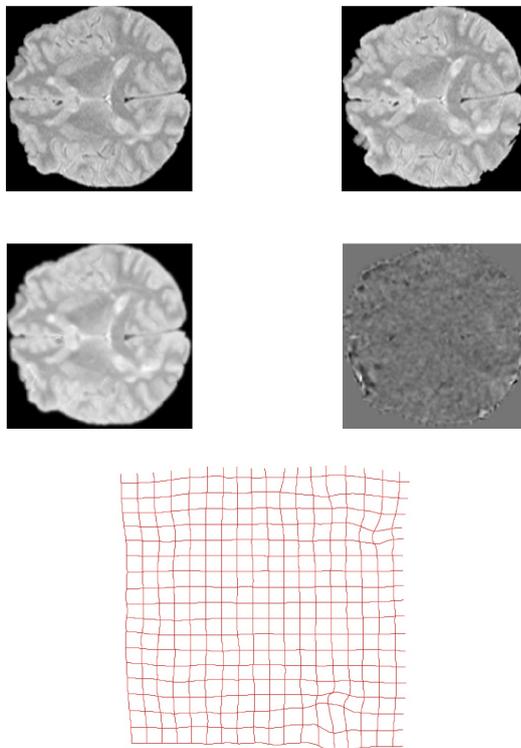


Abbildung 20: Nichtlineare Registrierung von zwei MRT-Bildern unter Verwendung des Multiskalen-Ansatzes. *Von oben links nach unten rechts:* Referenzbild, Template, deformiertes Template, Differenzbild von Referenz- und deformiertem Templatebild.

Für die verschiedenen Auflösungsstufen des Multilevel-Ansatzes wurden die Diskretisierungspunkte $m = (2^4, 2^4), \dots, (2^8, 2^8)$ verwendet und bei dem Multiskalen-Ansatz kamen, genau wie bei Registrierung der Kreise, die Werte $\sigma = 4, 2, 1$ für die Standardabweichung zum Einsatz.

Trotz ähnlicher Deformationsgitter (vgl. Abb. 19 und Abb. 20) lassen sich Unterschiede zwischen den gewählten Ansätzen feststellen.

Bei der Registrierung mit Multilevel-Ansatz fiel auf, dass die Anzahl an Iterationsschritten mit Anstieg der Auflösung nicht abnimmt. Dieses Verhalten lässt sich auch bei der Registrierung der MRT-Daten feststellen. Bei den Auflösungsstufen $m = (2^4, 2^4), \dots, (2^7, 2^7)$ wurden drei und bei $m = (2^8, 2^8)$ 37 Iterationsschritte benötigt. Diese plötzliche Erhöhung der Anzahl an Iterationsschritten kann mit der Abbildung der Gyri, d.h. den aus der Hirnmasse

hervortretende Gehirnwindungen, zu tun haben. Bei der Registrierung mit Multiskalen-Ansatz hingegen ist die Abnahme der Iterationsschritte nicht so deutlich zu erkennen wie bei der Registrierung der Kreise. Unter Verwendung der MRT-Daten werden bei der Standardabweichung $\sigma = 4$ 31, bei $\sigma = 2$ 19 und bei $\sigma = 1$ 24 Iterationsschritte benötigt.

Analog zu den synthetischen Daten, wollen wir auch die Rechenzeit vergleichen. Auf Grund der fehlenden Auflösungsreduktion ist die Laufzeit der Registrierung mit Multilevel-Ansatz länger. Die Rechenzeit bei einem Computer mit einem Dual Core Prozessor mit 1,3 GHz und 4 GB RAM betrug für den Multilevel-Ansatz 8.3 min und für den Multiskalen-Ansatz 15.72 min.

10 Zusammenfassung und Ausblick

In der vorliegenden Arbeit haben wir die nicht-lineare Registrierung mit einem Multilevel- und Multiskalen-Ansatz erarbeitet und auf MRT-Daten getestet.

Wir haben die einzelnen Bestandteile erläutert und diskretisiert. Des Weiteren können wir sagen, dass die Wahl des Gewichtungsparmeters ein Abwiegen zwischen einer exakten Registrierung und Glattheit des deformierten Gitters ist. Für die MRT-Daten liefert $\alpha = 10^8$ und für das klassische Registrierungsproblem von zwei Kreisen führt $\alpha = 10^2$ plausible Ergebnisse. Kleinere Werte bewirken unerwünschte Faltungen und größere Werte, dass sich die Transformation kaum verändert. Zudem konnten wir feststellen, dass die beiden Ansätze ähnliche Ergebnisse liefern, sich aber in der Laufzeit unterscheiden.

Für die Registrierung wurde das SSD-Maß, der diffusive Regularisierer und das Gauß-Newton Optimierungsverfahren mit Armijo-Schrittlängenbestimmung verwendet. Es blieben jedoch andere Distanzmaße Regularisierer unberücksichtigt. Zudem wurde auch die Volumenänderung der Läsionen nicht betrachtet.

Abbildungsverzeichnis

1	Cellcentered-Gitter 1D	12
2	Cellcentered-Gitter 2D	13
3	Nodal-Gitter 1D	13
4	Nodal-Gitter 2D	14
5	Staggered-Gitter 2D	14
6	Lineare Splines	18
7	Kubische Splines	20
8	Multiresolutionsverfahren Beispiel	28
9	Multiresolutionsverfahren	29
10	Prolongation und Restriktion,Nodal-Gitter	31
11	Multiresolutionsverfahren Beispiel	33
12	Multiskalen-Ansatz	33
13	Diskretisierung diffuser Regularisierer Teil 1	36
14	Diskretisierung diffuser Regularisierer Teil 2	37
15	Nicht-lineare Registrierung kleiner auf großen Kreis Teil 1	40
16	Nicht-lineare Registrierung kleiner auf großen Kreis Teil 2	41
17	Nicht-lineare Registrierung Kreis auf C	43
18	Nicht-lineare Registrierung verschiedener Parameter α	43
19	Nicht-lineare Registrierung Multilevel-Ansatz, MRT-Daten	44
20	Nicht-lineare Registrierung Multiskalen-Ansatz, MRT-Daten	45

Literatur

- [Bon11] Patienteninformation Universität Bonn, 2011. <http://imbie.meb.uni-bonn.de>, [Abruf: 20. Juli. 2011].
- [Der11] Alexander Derksen. *Bildregistrierung mit Volumenbeschränkung*. Bachelorarbeit, Universität zu Lübeck, Institute of Mathematics and Image Computing, 2011.
- [Deu10] Deutsche MS Gesellschaft, 2010. <http://www.dmsg.de>, [Abruf: 4. Jan. 2011].
- [FM01] Bernd Fischer and Jan Modersitzki. *A super fast registration algorithm*. Springer, 2001. editors: H.Handels, A.Horsch, T.M. Lehmann and H.P.H. Meinzer, *Bildverarbeitung in der Medizin 2001*.
- [FM02] Bernd Fischer and Jan Modersitzki. *Fast diffusion registration*, volume 313 of Contemporary Mathematics. AMS, 2002. editors: M. Nasehd and O. Scherzer, *Inverse Problems, Image Analyses and Medical Imaging*.
- [FM08] Bernd Fischer and Jan Modersitzki. Ill-posed medicine - an introduction to image registration. *Inverse Problems*, 24(3), 2008.
- [Had02] Jaques Hadamard. *Sur le problèmes aux dérivées partielles et leur signification physique*, chapter 13. Princeton University Bulletin, 1902.
- [Han09] Heinz Handels. *Medizinische Bildverarbeitung*. Vieweg + Teubner, 2009.
- [HB02] Martin Hanke-Bourgeois. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Teubner, 1 edition, April 2002.
- [MM06] Marco Mumenthaler and Heinrich Mattle. *Kurzlehrbuch Neurologie*. Thieme, 2006.
- [Mod09] Jan Modersitzki. *FAIR Flexible Algorithms for Image Registration*. SIAM, 2009.
- [NW00] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer, 2000.
- [Pap08] Nils Papenberg. *Ein genereller Registrierungsansatz mit Anwendung in der navigierten Leberchirurgie*. Dissertation, Universität zu Lübeck, Institut für Mathematik, 2008.
- [PG81] Margaret Wright Philip Gill, Walter Murray. *Practical Optimization*. Academic Press, 1981.

- [Vin83] Pierre J Vinken. *Handbook of Clinical Neurology*. Elsevier, 1983.
- [Wer92] Jochen Werner. *Numerische Mathematik 1*. Vieweg Studium, 1992.
- [Zim08] Veronika Zimmer. *Nicht-parametrische Registrierung von Magnetresonanz-Mammographie-Daten mit einem Multilevel-Ansatz*. Bachelorarbeit, Universität zu Lübeck, Institut für Mathematik, 2008.