

Morphologische multimodale Bildanpassung

Diplomarbeit
von
Dominik Löchel

Angefertigt am Mathematischen Institut
der Heinrich Heine Universität Düsseldorf,
Lehrstuhl für Angewandte Mathematik

19. März 2007

Inhaltsverzeichnis

1 Einführung	1
1.1 Definitionen	2
1.2 Aufgabenstellung	2
1.3 Das Minimierungsproblem	3
2 Zwei Abstandsmaße zur morphologischen multimodalen Bildanpassung	5
2.1 Ein physikalisch motiviertes Abstandsmaß für multimodale Bildvergleiche	5
2.1.1 Herleitung der RWA zum Abstandsmaß im zweidimensionalen Fall	6
2.1.2 Herleitung der RWA zum Abstandsmaß im dreidimensionalen Fall	8
2.2 Ein alternatives Abstandsmaß	12
3 Das Regularisierungsfunktional	17
4 Die vollständige Randwertaufgabe	19
5 Die Unterprogramme	21
5.1 Die Fouriertransformation	21
5.1.1 Die Cosinustransformation	21
5.1.2 Die Sinustransformation	21
5.1.3 Trigonometrische Transformation im mehrdimensionalen	22
5.2 Glätten mit der Wärmeleitungsgleichung	22
5.3 Interpolation des verformten Bildes	24
5.4 Die Suchrichtung	25
5.4.1 Die lineare Transformation	25
5.4.2 Die nicht-lineare Transformation	26
5.4.3 Lösen der Randwertaufgabe	26
5.4.4 Liniensuche	29
5.5 Veränderung der Auflösung	35
5.5.1 Halbierung der Auflösung	35
5.5.2 Hochinterpolation der Verformung	37
5.6 Einbinden von Kontrollpunkten	37
6 Der Algorithmus	41
6.1 Die Iteration über die Auflösungsstufen	41
6.2 Die Iterationen über die Glättungsstufen	42
6.3 Die innere Iteration der Bildanpassung	45
6.4 Der Speicherbedarf	46
6.5 Veranschaulichung des Registrierungsablaufes an einem einfachen zweidimensionalen Beispiel	47
6.6 Veranschaulichung des Registrierungsablaufes an einem zweiten Beispiel	53
7 Ergebnisse	57
7.1 Die Wirkung der Regularisierung	57
7.1.1 Elastische Verformung	62
7.1.2 Der Einfluss der Randbedingung	64
7.2 Regularisierung in der Liniensuche	65
7.3 Vergleich der beiden Maße	67
7.4 Glättung und abweichende Farben	67
7.5 Die Verwendung einer Gewichtungsmaske	72
7.6 Der Einsatz des CG-Verfahrens	73
7.7 Die Auswirkung von Bildstörungen	73
7.8 Der Einsatz von Kontrollpunkten	76
8 Schlussfolgerung	79
A Die Randwertaufgabe zum Navier-Lamé Operator	81
B CD mit Beispielbildern und Programmcode	83
Erklärung der Originalität	84
Literaturverzeichnis	85

Abbildungsverzeichnis	86
Algorithmenverzeichnis	89
Stichwortverzeichnis	90

1 Einführung

Die Bildanpassung ist ein Prozess bei dem eine Transformationsabbildung zwischen zwei Bildern berechnet wird, so dass die transformierten Bilder einander ähnlicher sind. In der Literatur sind auch die Begriffe Registrierung, Matching, Warping, Morphing und Ausrichtung zu finden. Bei diesen Verfahren wird im allgemeinen das eine Bild so transformiert, dass es zum anderen passt. Das zu verformende Bild wird im weiteren als Templatebild (T) und das andere als Referenzbild (R) bezeichnet.

Die Transformationen können in lineare und nicht-lineare unterteilt werden. Die linearen werden in der Literatur wiederum in rigide und nicht-rigide Transformationen gegliedert. Erstere erhalten das Volumen des Bildes und sind somit nur Verschiebungen und Drehungen. Die nicht-rigiden linearen Transformationen sind die Stauchung und die Scherung. In dieser Arbeit ist vor allem die nicht-lineare Transformation von Interesse, obwohl auch eine lineare Transformation zur globalen Ausrichtung von Größe und Lage des Bildes verwendet werden wird.

Die Bildanpassung ist zur Zeit ein sehr aktives Forschungsgebiet mit Anwendung in der medizinischen Bildauswertung, bei der Objektverfolgung, bei Erkennungsdiensten und bei der Auswertung von Luft- bzw. Satellitenbildern. Im Bereich der heutigen Hirnforschung ist die Bildanpassung ein gern benutztes Hilfsmittel, das es erlaubt, verschiedene Daten eines einzelnen Lebewesens oder mehrerer Lebewesen untereinander zu vergleichen. Bei ersterem werden Aufnahmen verschiedener Modalitäten wie z.B. Magnetresonanzbilder (MRI), Computertomographie (CT), Positron-Emissionstomographie (PET), Single-Photon-Emission-Computed-Tomography (SPECT) oder Gewebeschnitte verglichen. Jede dieser Aufnahmemodalitäten kann bestimmte Materialien gut wiedergeben, andere sind weniger gut zu erkennen oder sogar unsichtbar. Anhand derjenigen Strukturen die in beiden Modalitäten sichtbar sind, sollen die Bilder passend übereinandergelegt werden, so dass der Informationsgehalt aus beiden Aufnahmen gemeinsam genutzt werden kann. Bei zweidimensionalen Bildern ist dieser Vorgang noch halbwegs von Hand machbar. Dagegen ist die manuelle Ausrichtung von dreidimensionalen Bildern kaum möglich, da zu einem Zeitpunkt immer nur einzelne Ebenen des 3D-Bildes betrachtet werden können. Ein weiterer Anwendungsfall sind Bildvergleiche zwischen Bildern, die zu unterschiedlichen Zeiten aufgenommen wurden. Hier soll die Bildanpassung dazu dienen, die Unterschiede in den Bildern besser zu erkennen. Dabei ist zu fordern, dass die Transformation in gewissem Sinne „glatt“ ist, damit die wesentliche Form angepasst wird, aber Unterschiede in den Details erhalten bleiben. Ein Einsatzbereich in der Medizin ist die Beobachtung von Tumorentwicklungen. Bei den Bildvergleichen zwischen verschiedenen Individuen handelt es sich meist um den Vergleich eines individuellen Gehirns zu einem Referenzgehirn bzw. Atlssystem.

Das Ziel der Bildanpassung besteht darin zwei Bilder ähnlicher zu machen. Das verwendete Modell besteht aus der Minimierung eines Unähnlichkeits- bzw. Abstandsfunktional. Dazu kann entweder eine von den Bildpunkten selbst abhängige Funktion benutzt werden oder es wird der Abstand von Kontrollpunkten verwendet, die vorab ermittelt werden. Ersteres hat den Vorteil, dass alle im Bild vorhandenen Informationen genutzt werden. Hingegen wird bei den Kontrollpunkten nur Information von wenigen Punkten verwendet.

Die Verwendung der Kontrollpunkte verlangt meist ein manuelles setzen der Punkte. Die Anpassung anhand einiger Kontrollpunkte ist sehr robust, vor allem dann, wenn eines der Bilder viele Artefakte enthält. Andererseits können trotz exaktem Passen der Kontrollpunkte die Bilder im großen und ganzen voneinander stark abweichen, da keine Information zwischen den Kontrollpunkten verwendet worden ist. Zwischen den Kontrollpunkten wird die Transformation in irgendeiner Weise einen glatten Übergang herstellen. Sind die Kontrollpunkte nicht sehr genau ausgewählt so kann das Ergebnis extrem ungenau ausfallen. Bereits bei dreidimensionalen Bildern ist es sehr schwierig die Kontrollpunkte von Hand auszuwählen.

Bei der Bildanpassung von monomodalen Daten, also Bildern die mit dem gleichen bildgebenden Verfahren aufgenommen worden sind, kann als Abstand die Summe über die quadratischen Farbdifferenzen der korrespondierenden Punkte verwendet werden. Dagegen sind die absoluten Farbwerte bei der multimodalen Bildanpassung im allgemeinen echt verschieden. In diesem Fall muss die Morphologie, also die Gestalt der Objekte im Bild verwendet werden. Jedes im Bild zu erkennende Objekt zeichnet sich dadurch aus, dass die Bildfarbe an der Objektgrenze einen deutlichen Farbunterschied (Gradient) aufweist. Die Höhenlinien der Gradienten entsprechen den Objektumrissen sehr gut. In dieser Arbeit wird daher ein Verfahren hergeleitet, dass die Höhenlinien eines Templatebildes T auf die Höhenlinien des Referenzbildes R ausrichtet.

Die Bildanpassung gehört zum Bereich der „schlecht gestellten Probleme“ („ill-posed“). Ein Problem ist nach Hadamard schlecht gestellt, falls eine der Bedingungen

1. zu jeder Eingabegröße existiert eine Lösung
2. die Lösung ist eindeutig
3. die Lösung hängt stetig von der Eingabe ab

nicht erfüllt ist. Schlecht gestellte Probleme werden in der Literatur auch als „inverse Probleme“ bezeichnet, da in

der Physik normalerweise zu bekannten Kräften die an ein Objekt angreifen, die resultierende Bahnkurve gesucht ist (gutartiges Problem). Wird zu einer Bahnkurve die Kraft gesucht, so ist das die inverse Fragestellung. Hier wird bereits klar, dass sich die Kräfte i.a. nicht eindeutig finden lassen und dass sie unstetig von leichten Modifikationen an der Bahnkurve abhängen können. Bei der Bildregistrierung ist i.a. keine der drei Bedingungen erfüllt. (1.) bedeutet, dass zu zwei Bildern R, T eine Transformationsabbildung ϕ existiert, so dass $R = T^\phi$ gilt. (2.) besagt: falls es zwei Transformationsabbildungen ϕ, ψ gibt, so dass $T^\phi = R = T^\psi$ erfüllt ist, so gilt $\phi \equiv \psi$. (3.) sagt, dass sich die Transformation bei einer kleinen Änderung an den Bildern auch nur geringfügig ändert.

Die Forderung $R = T^\phi$ wird wegen der multimodalen Daten und der inversen Problemstellung durch die Minimierung eines Abstandsmaßes ersetzt. Um (3.) in den Griff zu bekommen wird an die Transformationsabbildung die Zusatzforderung gestellt, dass die Verzerrung stetig und nicht zu groß sein darf. Im ersten Teil der Arbeit wird die Minimierungsaufgabe definiert. Es werden anschließend Abstands- und Verzerrungsmaße vorgestellt. Zu dem Minimierungsproblem wird eine Suchrichtung hergeleitet, so dass es nach dem Verfahren des steilsten Abstieges gelöst werden kann. Dabei tritt ein hochgradig nicht-lineares Randwertproblem auf, das iterativ gelöst werden muss. In einem ersten Schritt werden dazu die benötigten Unterprogramme einzeln besprochen. Anschließend wird der Algorithmus des Abstiegsverfahrens vorgestellt und diskutiert. Im letzten Kapitel wird das Bildanpassungsprogramm anhand von Ergebnissen aus der Praxis untersucht.

1.1 Definitionen

Im folgenden werden ein paar Dinge definiert, die in späteren Kapiteln immer vorausgesetzt sind, ohne dass sie dort erwähnt werden.

Ein digitales Bild kann mathematisch als Gitter Ω mit Gitterkonstante h betrachtet werden. Jedem Gitterpunkt $\vec{x} \in \Omega$ wird ein Grauwert $R(\vec{x}) \in [0, 1] \subset \mathbb{R}$ zugeordnet. Dabei bedeutet Null schwarz und Eins weiß. Die übrigen Grauwerte sind linear dazwischen angeordnet. Sei n die Dimension des Bildes, d.h. $n = 2$ bei Fotos und $n = 3$ bei dreidimensionalen Bildern. n_i sei die Anzahl der Gitterpunkte entlang der i -ten Raumrichtung für $i = 1, \dots, n$. Praktisch ist dann $\Omega \subset \mathbb{Z}^n$. Bei der theoretischen Besprechung ist es sinnvoller $\Omega \subset \mathbb{R}^n$ zu verallgemeinern. Praktisch lassen sich die Zwischenwerte mit Interpolation gewinnen. Mit $\partial\Omega$ wird der Rand von Ω bezeichnet. Das sind je nach Anwendungsfall entweder die Randpixel des Bildes, oder aber die erste Pixelreihe, die nicht mehr zum Bild gehört. Im kontinuierlichen ist Ω im ersten Fall *abgeschlossen* und im zweiten *offen*. Alles was sich außerhalb von Ω befindet wird als 0 (schwarz) definiert.

Im folgenden wird der kontinuierliche Fall aufgeschrieben. Die Integrale über Ω entsprechen dann bei der diskreten Rechnung Summen über alle Gitterpunkte. Ableitungen werden mit endlichen Differenzen zweiter Ordnung berechnet, d.h.

$$\begin{aligned} \frac{\partial}{\partial x} f(x) &= \frac{f(x+h) - f(x-h)}{2 \cdot h} &= \frac{1}{2 \cdot h} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} f(x) \\ \frac{\partial^2}{\partial x^2} f(x) &= \frac{f(x+h) - 2 \cdot f(x) + f(x-h)}{h^2} &= \frac{1}{h^2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} f(x) \end{aligned}$$

und in Randpunkten werden einseitige Differenzen genommen. Mehrdimensionale Ableitungen lassen sich durch sukzessives Anwenden entlang jeder Raumrichtung bilden.

Mit $H_2(\Omega)$ wird der Sobolevraum aller Funktionen, die auf Ω quadrat-integrabel sind, und deren zweite Ableitung(en) unter dem Integral existieren (schwach differenzierbar bis Ordnung 2) bezeichnet. Weiter sei

$$H_2^D(\Omega) := \{h \in H_2(\Omega) \mid h = 0 \text{ auf } \partial\Omega\}$$

und

$$H_2^N(\Omega) := \left\{ h \in H_2(\Omega) \mid \frac{\partial}{\partial \vec{n}} h = 0 \text{ auf } \partial\Omega \right\}.$$

1.2 Aufgabenstellung

Gegeben sind zwei Graustufenbilder R und T mit gleicher Auflösung. Diese Forderung stellt keine Einschränkung dar, da das kleinere Bild ggf. durch schwarzen Rand vergrößert werden kann. Beide Bilder stellen das gleiche reale Objekt dar, können aber auf verschiedene Weise aufgenommen worden sein, zum Beispiel Fotoapparat, Magnetresonanz, Röntgentomographie und anderes mehr. Diese Bilder werden selbstverständlich nicht gleich aussehen, aber einige Strukturen werden sich wiederfinden lassen.

Nun besteht die Aufgabe darin, das „Template“-Bild T so zu verzerren, dass die Strukturen deckungsgleich zum „Referenz“-Bild R liegen. Zunächst wird ein Maß gebraucht, welches angibt, wie „nicht-deckungsgleich“ die beiden

Bilder R und T sind. Mit dem punktweise definierten Maß

$$d(R, T; \vec{x}) = d(R(\vec{x}), T(\vec{x}))$$

kann der Abstand in jedem Pixel bzw. Voxel \vec{x} gemessen werden. Der Gesamtabstand der Bilder wird dann durch

$$\mathcal{D}_{R,T} = \int_{\Omega} d(R(\vec{x}), T(\vec{x}))^2 d\vec{x} = \int_{\Omega} d(R, T; \vec{x})^2 d\vec{x}$$

aufsummiert.

Sei n die Dimension der Bilder. Dann lässt sich mit dem Verschiebungsfeld

$$\vec{u} \in \mathbb{R}^{n \times \Omega}, \quad \vec{u}(\vec{x}) \in \mathbb{R}^n \quad \forall \vec{x} \in \Omega$$

eine Koordinatentransformation

$$\phi(\vec{u}) : \Omega \rightarrow \mathbb{R}^n, \quad \phi(\vec{u})(\vec{x}) = \vec{x} - \vec{u}(\vec{x}) \quad \forall \vec{x} \in \Omega$$

beschreiben. Das verzerrte Templatebild ist dann $T \circ \phi(\vec{u})$ wobei

$$T \circ \phi(\vec{u})(\vec{x}) = T(\phi(\vec{u})(\vec{x})) = T(\vec{x} - \vec{u}(\vec{x})) \quad \forall \vec{x} \in \Omega.$$

Es kann passieren, dass es $\vec{x} \in \Omega$ gibt, so dass $\phi(\vec{u})(\vec{x}) \notin \Omega$. In diesem Fall muss das Bild T über Ω hinaus fortgesetzt werden. Die einfachste Möglichkeit besteht darin, allen Bildpunkten außerhalb von Ω einen einheitlichen Default-Farbwert, z.B. schwarz zu zuweisen. Der Default-Farbwert sollte der gleiche sein wie bei der Größenanpassung der beiden Bilder.

Den Abstand des verzerrten Bildes $T \circ \phi(\vec{u})$ zum Referenzbild R beschreibt

$$\mathcal{D}_{R,T}(\phi(\vec{u})) = \int_{\Omega} d(R(\vec{x}), T(\phi(\vec{u})(\vec{x})))^2 d\vec{x} = \int_{\Omega} d(R, T \circ \phi; \vec{x})^2 d\vec{x}.$$

Im weiteren Teil wird das Argument \vec{u} in ϕ nicht immer explizit hingeschrieben.

Es entsteht das folgende Minimierungsproblem:

Finde eine Koordinatentransformation ϕ^{opt} , so dass $\mathcal{D}(\phi^{\text{opt}})$ minimal ist.

Nun wird ϕ^{opt} im allgemeinen die Bildpunkte kreuz und quer vertauschen. Das ist nicht die gesuchte Lösung. Benachbarte Punkte in T müssen in $T(\phi)$ weiterhin benachbart sein. Diese Forderung wird im folgenden unter dem Begriff „Topologieerhaltung“ verstanden. Anschaulich kann das Bild als dehnbare Gummihaut betrachtet werden, die sich verzerren lässt, indem man an einigen Stellen Kräfte greifen lässt. Um dies mathematisch in den Griff zu bekommen, wird zu dem Abstandsmaß ein Regularisierungsterm addiert, der zu extreme Verzerrungen bestraft. Diese Beschreibung wird im nächsten Abschnitt präziser formuliert.

1.3 Das Minimierungsproblem

Die Aufgabe lautet: Finde eine Koordinatentransformation

$$\phi(\vec{u}) : \Omega \rightarrow \mathbb{R}^n, \quad \phi(\vec{u})(\vec{x}) = \vec{x} - \vec{u}(\vec{x}), \quad \vec{x} \in \Omega, \quad (\text{Koordinatentransformation})$$

so dass der Abstand zwischen R und $T \circ \phi$ minimal wird, mit der Nebenbedingung, dass die Topologie erhalten bleibt.

Der Abstand der Bilder R und T lässt sich in jedem einzelnen Punkt $\vec{x} \in \Omega$ formal durch ein Abstandsmaß

$$d(R, T; \cdot) : \Omega \rightarrow \mathbb{R} \quad (\text{Abstandsmaß})$$

beschreiben. Der Gesamtabstand der Bilder ist die Summe der quadratischen Abstände in jedem einzelnen Punkt

$$\mathcal{D}_{R,T}(\phi) = \int_{\Omega} (d(R, T \circ \phi; \vec{x}))^2 d\vec{x}. \quad (\text{Abstand zweier Bilder})$$

Das Minimierungsproblem lässt sich jetzt formulieren als

$$\text{Finde } \vec{u} \in \mathbb{R}^{n \times \Omega}, \text{ so dass } \mathcal{D}_{R,T}(\phi(\vec{u})) \text{ minimal wird und die Topologie erhalten bleibt.} \quad (\text{M1})$$

Die Topologieerhaltung ist also eine Regularisierungsaufgabe, d.h. der Grad der Stetigkeit des Verschiebungsfeldes \vec{u} muss gemessen werden und als Strafterm in das Minimierungsproblem eingehen. Die Topologie bleibt erhalten, wenn $\frac{1}{\varepsilon} \cdot |\vec{u}(\vec{x}) - \vec{u}(\vec{x} + \varepsilon \cdot \vec{h})|$ hinreichend klein ist für jede beliebige Richtung $\vec{h} \in \mathbb{R}^{n \times \Omega}$ mit $\|\vec{h}\|_2 = 1$ und jedes kleine $\varepsilon > 0$. Der Grad der Stetigkeit lässt sich also mit der Größe der Ableitung messen.

Ein Punktweises Maß für die Stetigkeit des Verschiebungsfeldes \vec{u} im Punkt $\vec{x} \in \Omega$ ist z.B. die erste Ableitung

$$s(\vec{u})(\vec{x}) = \left(\sum_{i=1}^n \|\nabla u_i(\vec{x})\|_2^2 \right)^{\frac{1}{2}} = \left(\sum_{i=1}^n \sum_{j=1}^n \left(\frac{\partial}{\partial x_j} u_i(\vec{x}) \right)^2 \right)^{\frac{1}{2}}.$$

Das globale Maß für die Stetigkeit von \vec{u} erfolgt wieder durch Aufsummieren über alle Punkte

$$S(\vec{u}) = \int_{\Omega} (s(\vec{u})(\vec{x}))^2 d\vec{x}. \quad (\text{Regularisierungsterm})$$

Das Minimierungsproblem lautet schließlich

$$\text{minimiere}_{\vec{u} \in \mathbb{R}^{n \times \Omega}} \mathcal{D}_{R,T}(\phi(\vec{u})) + \alpha \cdot S(\vec{u}) \quad (\text{Minimierungsproblem})$$

wobei $\alpha > 0$ ein Gewichtungsfaktor für den Strafterm ist. Es wird also einerseits gefordert, dass das transformierte Templatebild $T \circ \phi(\vec{u})$ möglichst deckungsgleich zu R ist und andererseits, dass die Verzerrung \vec{u} nicht zu stark sein soll. Die optimale Wahl von $\alpha > 0$ wird sich später als Inverse der Schrittweite herausstellen. Bei dem Minimierungsproblem handelt es sich um ein zu minimierendes Funktional

$$I_{R,T}(\vec{u}) = \mathcal{D}_{R,T}(\phi(\vec{u})) + \alpha \cdot S(\vec{u}) \quad (\text{zu minimierendes Funktional I})$$

Dieses lässt sich mit Standardmethoden aus der Optimierung wie z.B. das *Verfahren des steilsten Abstiegs* oder die *Simplexmethode von Nelder und Mead* (vgl. [9]) aufgrund der großen Anzahl von

$$A = n \cdot \prod_{i=1}^n n_i \quad (\text{Anzahl der Variablen})$$

Variablen in \vec{u} nicht effizient lösen. Bei einem recht kleinen dreidimensionalen Bild der Auflösung 256^3 Voxel sind das bereits $A = 3 \cdot 256^3 > 50$ Millionen Variablen. Das *Verfahren des steilsten Abstiegs* benötigt in jedem Schritt den Gradienten. Um diesen auszurechnen sind beim Differenzenverfahren erster Ordnung $A + 1$ viele Auswertungen von $I_{R,T}$ notwendig und beim Differenzenverfahren zweiter Ordnung $2 \cdot A$ viele. Die *Simplexmethode von Nelder und Mead* kommt zwar ohne Gradienten aus, dafür müssen $A + 1$ viele affin-linear unabhängige Punkte \vec{u}_i gewählt werden, zu denen der Funktionswert $I_{R,T}(\vec{u}_i)$ berechnet und verglichen wird. Zum einen muss dabei zumindest im ersten Schritt das Abstandsfunktional $A + 1$ mal ausgewertet werden und zum anderen müssen $(A + 1) \cdot A$ viele Variablen gespeichert werden. Bei obigem Beispiel sind das mehr als $2 \cdot 10^{15}$ (zwei Billionen) Stück. Es wird sich später zeigen, dass die Auswertung des Abstandsfunktionals $\mathcal{D}_{R,T}$ der zeitaufwändigste Teil des ganzen Algorithmus ist.

Im folgenden wird die Richtungsableitung von I nach \vec{u} für konkrete Abstandsfunktionale \mathcal{D} und Regularisierungsfunktionale S analytisch hergeleitet. Bei der numerischen Berechnung der Abstiegsrichtung braucht das Abstandsfunktional I dann – im Gegensatz zum Verfahren des *steilsten Abstiegs* oder der *Simplexmethode von Nelder und Mead* – kein einziges Mal ausgewertet zu werden. Es wird sich zeigen, dass lokale Minima von I Lösungen einer hochgradig nicht-linearen Randwertaufgabe sind.

2 Zwei Abstandsmaße zur morphologischen multimodalen Bildanpassung

2.1 Ein physikalisch motiviertes Abstandsmaß für multimodale Bildvergleiche

In diesem Abschnitt wird ein physikalisch motiviertes Abstandsmaß $d(R, T)$ für multimodale Bilder vorgestellt. Bei monomodalen Bildern kann die Differenz der Farbwerte $d(R, T; \vec{x}) = R(\vec{x}) - T(\vec{x})$ als Abstandsmaß verwendet werden. Im multimodalen Fall sind die Farbwerte zu gleichen Massen-Stellen des Objektes im allgemeinen echt verschieden. Zwar ließen sich die Farben durch Kenntnis der Aufnahmegerate gegebenenfalls umrechnen, das hier vorgestellte Abstandsmaß wird dagegen nicht auf solche Informationen angewiesen sein.

Gegeben sei ein Objekt O mit Massenverteilung $m(\vec{x})$. O wird mit zwei verschiedenen Bildgebenden Verfahren χ und ζ aufgenommen. Die Grauwertverteilung in den Ausgaben dieser Verfahren seien

$$\begin{aligned} R(\vec{x}) &:= \chi(m(\vec{x})) \\ T(\vec{x}) &:= \zeta(m(\vec{x})), \end{aligned}$$

wobei von den Funktionen χ und ζ nur bekannt ist, dass sie differenzierbar von $m(\vec{x})$ abhängen. Nun gilt

$$\begin{aligned} \nabla R(\vec{x}) &= \nabla(\chi(m(\vec{x}))) = \chi'(m(\vec{x})) \cdot \nabla m(\vec{x}) \\ \nabla T(\vec{x}) &= \nabla(\zeta(m(\vec{x}))) = \zeta'(m(\vec{x})) \cdot \nabla m(\vec{x}), \end{aligned}$$

d.h. falls R und T richtig ausgerichtet sind, so unterscheiden sich ihre Gradienten nur in der Länge. Also sind $\nabla R(\vec{x})$ und $\nabla(T \circ \phi)(\vec{x})$ für das gesuchte ϕ linear abhängig. Sind zwei Vektoren linear abhängig, so ist ihr Senkrechtanteil Null. Der Betrag des Senkrechtanteils ist daher ein sinnvolles Abstandsmaß für multimodale Bilder.

Der Senkrechtanteil zweier Vektoren lässt sich im zweidimensionalen als

$$d(R, T \circ \phi; \vec{x}) := \nabla(T \circ \phi)^t(\vec{x}) \cdot \nabla R^\perp(\vec{x}) := \frac{\partial}{\partial x_1} R(\vec{x}) \cdot \frac{\partial}{\partial x_2} (T \circ \phi)(\vec{x}) - \frac{\partial}{\partial x_2} R(\vec{x}) \cdot \frac{\partial}{\partial x_1} (T \circ \phi)(\vec{x})$$

und im dreidimensionalen mit dem Vektor-Kreuzprodukt

$$d(R, T \circ \phi; \vec{x}) := \sqrt{\left(\nabla(T \circ \phi)(\vec{x}) \times \nabla R(\vec{x}) \right)^t \cdot \left(\nabla(T \circ \phi)(\vec{x}) \times \nabla R(\vec{x}) \right)} = \|\nabla(T \circ \phi)(\vec{x}) \times \nabla R(\vec{x})\|_2$$

schreiben. So wie es hier steht, gilt im zweidimensionalen $d(R, T, \vec{x}) \in \mathbb{R}$ und im dreidimensionalen $d(R, T, \vec{x}) \in \mathbb{R}_+$. Es wird sich in der folgenden Rechnung allerdings zeigen, dass der dreidimensionale Fall aus einer Summe von zweidimensionalen Unterraumproblemen besteht. Dort ist das Abstandsmaß dann wieder vorzeichenbehaftet.

An Kanten ist der Gradient ungleich Null. An parallel verlaufenden Kanten sind die Gradienten ∇R und ∇T entweder genau parallel oder genau antiparallel, also ist der Senkrechtanteil hier Null. An sich kreuzenden Kanten ist der Senkrechtanteil der beiden Gradienten ungleich Null und ebenso das Abstandsmaß. Es hängt außerdem stetig sowohl von der Größe der sich kreuzenden Kanten $\|\nabla R(\vec{x})\|_2$ und $\|\nabla(T \circ \phi)(\vec{x})\|_2$, als auch vom Winkel $\angle(\nabla R(\vec{x}), \nabla(T \circ \phi)(\vec{x}))$ ab. Zusammenfassend lässt sich feststellen, dass dieses Maß misst, wie schlecht die Höhenlinien der beiden Bilder auf einander ausgerichtet sind.

Dieses Maß hat die folgenden Eigenschaften:

1. $d(R, R) = 0$, d.h. jedes Bild hat zu sich selbst den Abstand Null.
2. $d(R, T) = d(T, R)$, d.h. es ist symmetrisch in den Argumenten.
3. $d(R, C) = 0$ für jedes Bild C mit $\nabla C = 0$, also konstanter Helligkeit.
4. $d(R, T + C) = d(R, T)$ für jedes Bild C mit $\nabla C = 0$. d ist unabhängig von der absoluten Helligkeit.
5. $d(R, \mu \cdot T) = \mu \cdot d(R, T) \forall \mu \in \mathbb{R}$.
6. $d(R, \mu \cdot R) = 0 \forall \mu \in \mathbb{R}$. Ein Bild, dass sich nur in der Helligkeit unterscheidet, hat Abstand Null.

Die Eigenschaft (3.) ist unter Umständen ein Problem. Angenommen das Objekt O hat einen Bereich in dem die Massenverteilung konstant ist. Liegt T so über R , dass eine Kante von T über dem Bereich konstanter Massenverteilung in R liegt, so misst d den Abstand Null. Es besteht die Chance, dass das nicht-passen an anderen Stellen dafür sorgt, dass die Kante diesen Bereich verlässt und doch noch an die zugehörige Stelle verschoben wird.

Diese Eigenschaft ist andererseits von Vorteil, falls die beiden bildgebenden Verfahren nicht jede Änderung der Massenverteilung abbilden. Z.B. bildet die Computer-Röntgen-Tomographie die Knochen sehr gut ab, die Weichteile dafür sehr schlecht. Die Weichteile sind im Magnetresonanzbild dagegen gut zu erkennen. In diesem Fall wird das Templatebild anhand der Lage der Knochen ausgerichtet.

2.1.1 Herleitung der RWA zum Abstandsmaß im zweidimensionalen Fall

In diesem Abschnitt wird zu dem oben vorgestellten Maß

$$d(R, T \circ \phi; \vec{x}) = \nabla(T \circ \phi)^t(\vec{x}) \cdot \nabla R^\perp(\vec{x}) = \frac{\partial}{\partial x_1} R(\vec{x}) \cdot \frac{\partial}{\partial x_2} (T \circ \phi)(\vec{x}) - \frac{\partial}{\partial x_2} R(\vec{x}) \cdot \frac{\partial}{\partial x_1} (T \circ \phi)(\vec{x})$$

die Abstiegsrichtung hergeleitet. Diese wird später als rechte Seite in der Randwertaufgabe eingesetzt. Zuvor wird das Vektor-Kreuzprodukt für den \mathbb{R}^2 definiert.

Definition 1. Vektor-Kreuzprodukt im \mathbb{R}^2

$$\vec{a} \times \vec{b} := a_2 \cdot b_1 - a_1 \cdot b_2$$

Wie sich später zeigt, sind mit dieser Definition die Ergebnisse im \mathbb{R}^3 abwärtskompatibel zum \mathbb{R}^2 , denn nun lässt sich d als

$$d(R, T \circ \phi; \vec{x}) = \nabla(T \circ \phi)^t(\vec{x}) \cdot \nabla R^\perp(\vec{x}) = \nabla T(\phi(\vec{x})) \times \nabla R(\vec{x})$$

schreiben.

Das lokale Abstandsmaß $d(R, T; \vec{x})$ im \mathbb{R}^2

Sei

$$\phi(\vec{u}) : \Omega \rightarrow \mathbb{R}^2, \quad \phi(\vec{u})(\vec{x}) := \begin{pmatrix} \phi^1(u_1)(\vec{x}) \\ \phi^2(u_2)(\vec{x}) \end{pmatrix} = \begin{pmatrix} x_1 - u_1(\vec{x}) \\ x_2 - u_2(\vec{x}) \end{pmatrix}$$

die Koordinatentransformation des Templatebildes. Im folgenden wird der Übersichtlichkeit halber nur $\phi(\vec{x})$ statt $\phi(\vec{u})(\vec{x})$ geschrieben und $(\nabla T)(\phi(\vec{x})) = \nabla T^\phi(\vec{x})$ abgekürzt.

$$\begin{aligned} d(R, T, \phi; \vec{x}) &= \nabla T(\phi(\vec{x})) \times \nabla R(\vec{x}) \\ &= \frac{\partial}{\partial x_1} R(\vec{x}) \cdot \frac{\partial}{\partial x_2} T(\phi(\vec{x})) - \frac{\partial}{\partial x_2} R(\vec{x}) \cdot \frac{\partial}{\partial x_1} T(\phi(\vec{x})) \\ &= \frac{\partial}{\partial x_1} R(\vec{x}) \cdot (\nabla T)^t(\phi(\vec{x})) \cdot \frac{\partial}{\partial x_2} \phi(\vec{x}) - \frac{\partial}{\partial x_2} R(\vec{x}) \cdot (\nabla T)^t(\phi(\vec{x})) \cdot \frac{\partial}{\partial x_1} \phi(\vec{x}) \\ &= (\nabla T)^t(\phi(\vec{x})) \cdot D\phi(\vec{x}) \cdot \left(-\frac{\partial}{\partial x_2} R(\vec{x}), \frac{\partial}{\partial x_1} R(\vec{x}) \right)^t \\ &= (\nabla T^\phi)^t(\vec{x}) \cdot D\phi(\vec{x}) \cdot \nabla R^\perp(\vec{x}) \end{aligned}$$

$$\text{oder kurz: } d(R, T, \phi) = (\nabla T^\phi)^t \cdot D\phi \cdot \nabla R^\perp$$

Das globale Abstandsfunktional $\mathcal{D}(\phi)$ im \mathbb{R}^2

Das Abstandsmaß d ist lokal in jedem Punkt definiert. Um den Gesamtabstand der Bilder R und T in Abhängigkeit der Koordinatentransformation ϕ zu erhalten, wird der quadratische Abstand über alle Punkte aufsummiert. Dies macht das Funktional

$$\mathcal{D}_{R,T}(\phi) = \int_{\Omega} d^2(R, T, \phi; \vec{x}) \, d\vec{x} = \int_{\Omega} \left(\nabla(T \circ \phi)^t(\vec{x}) \cdot \nabla R^\perp(\vec{x}) \right)^2 \, d\vec{x}.$$

Die Richtungsableitung des Abstandsfunktionals \mathcal{D} im \mathbb{R}^2

Das Abstandsfunktional \mathcal{D} ist bezüglich ϕ zu minimieren. Dazu wird im folgenden die Abstiegsrichtung hergeleitet.

Sei ϕ hier eine beliebige Koordinatentransformation und die Richtung $h \in H_2(\Omega)^2$ beliebig gewählt. Die Richtungsableitung des Abstandsfunktionals \mathcal{D} in Richtung von h ist

$$\begin{aligned}
\mathcal{D}'_{R,T}(\phi)h &:= \frac{\partial \mathcal{D}_{R,T}}{\partial \phi}(\phi)h = \int_{\Omega} \frac{\partial d^2(R, T, \phi; \vec{x})}{\partial \phi} h \, d\vec{x} \\
&= \int_{\Omega} 2 \cdot d(R, T, \phi; \vec{x}) \cdot \lim_{\varepsilon \rightarrow 0} \frac{d(R, T, \phi + \varepsilon \cdot h; \vec{x}) - d(R, T, \phi; \vec{x})}{\varepsilon} d\vec{x} \\
&= \int_{\Omega} 2 \cdot d(R, T, \phi; \vec{x}) \cdot \lim_{\varepsilon \rightarrow 0} \frac{(\nabla T^{\phi + \varepsilon \cdot h})^t \cdot D(\phi + \varepsilon \cdot h) \cdot \nabla R^{\perp} - (\nabla T^{\phi})^t \cdot D\phi \cdot \nabla R^{\perp}}{\varepsilon} d\vec{x} \\
&= \int_{\Omega} 2 \cdot d(R, T, \phi; \vec{x}) \cdot \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \left((\nabla T^{\phi + \varepsilon \cdot h})^t \cdot (D\phi + D(\varepsilon \cdot h)) - (\nabla T^{\phi})^t \cdot D\phi \right) \cdot \nabla R^{\perp} d\vec{x} \\
&\text{Taylorentwicklung von } \nabla T^{\phi + \varepsilon \cdot h} \text{ in } \nabla T^{\phi} \\
&= \int_{\Omega} 2 \cdot d(R, T, \phi; \vec{x}) \cdot \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \left(((\nabla T^{\phi})^t + \varepsilon \cdot h^t \cdot D^2 T^{\phi} + \mathcal{O}(\varepsilon^2)) \cdot (D\phi + \varepsilon \cdot Dh) - (\nabla T^{\phi})^t \cdot D\phi \right) \cdot \nabla R^{\perp} d\vec{x} \\
&= \int_{\Omega} 2 \cdot d(R, T, \phi; \vec{x}) \cdot \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \left((\nabla T^{\phi})^t \cdot D\phi + (\nabla T^{\phi})^t \cdot \varepsilon \cdot Dh + \varepsilon \cdot h^t \cdot D^2 T^{\phi} \cdot D\phi \right. \\
&\quad \left. + \varepsilon \cdot h^t \cdot D^2 T^{\phi} \cdot \varepsilon \cdot Dh + \mathcal{O}(\varepsilon^2) - (\nabla T^{\phi})^t \cdot D\phi \right) \cdot \nabla R^{\perp} d\vec{x} \\
&= \int_{\Omega} 2 \cdot d(R, T, \phi; \vec{x}) \cdot \lim_{\varepsilon \rightarrow 0} \left((\nabla T^{\phi})^t \cdot Dh + h^t \cdot D^2 T^{\phi} \cdot D\phi + \varepsilon \cdot h^t \cdot D^2 T^{\phi} \cdot Dh + \mathcal{O}(\varepsilon) \right) \cdot \nabla R^{\perp} d\vec{x} \\
&= \int_{\Omega} 2 \cdot d(R, T, \phi; \vec{x}) \cdot \lim_{\varepsilon \rightarrow 0} \left((\nabla T^{\phi})^t \cdot Dh + h^t \cdot D^2 T^{\phi} \cdot D\phi + \mathcal{O}(\varepsilon) \right) \cdot \nabla R^{\perp} d\vec{x} \\
&= 2 \cdot \int_{\Omega} d(R, T, \phi; \vec{x}) \cdot \left((\nabla T^{\phi})^t \cdot Dh + h^t \cdot D^2 T^{\phi} \cdot D\phi \right) \cdot \nabla R^{\perp} d\vec{x}.
\end{aligned}$$

Der erste Summand wird jetzt noch einmal partiell integriert, so dass der Term Dh verschwindet. Dazu folgende Überlegung. Dh wird hoch integriert, die anderen Multiplikatoren werden differenziert.

$$\begin{aligned}
&\int_{\Omega} d \cdot (\nabla T)^t \cdot Dh \cdot \nabla R^{\perp} \, dx dy \\
&= \int_{\Omega} d \cdot (T_x \cdot h_x^1 + T_y \cdot h_x^2) \cdot (-R_y) + d \cdot (T_x \cdot h_y^1 + T_y \cdot h_y^2) \cdot R_x \, dx dy \\
&= - \int_{\Omega} d_x \cdot (T_x \cdot h^1 + T_y \cdot h^2) \cdot (-R_y) + d_y \cdot (T_x \cdot h^1 + T_y \cdot h^2) \cdot R_x \\
&\quad + d \cdot (T_{xx} \cdot h^1 + T_{yx} \cdot h^2) \cdot (-R_y) + d \cdot (T_{xy} \cdot h^1 + T_{yy} \cdot h^2) \cdot R_x \\
&\quad + d \cdot (T_x \cdot h^1 + T_y \cdot h^2) \cdot (-R_{yx}) + d \cdot (T_x \cdot h^1 + T_y \cdot h^2) \cdot R_{xy} \, dx dy \\
&\quad + \oint_{\partial\Omega} d \cdot (T_x \cdot h^1 + T_y \cdot h^2) \cdot (-R_y) \cdot n_1 + d \cdot (T_x \cdot h^1 + T_y \cdot h^2) \cdot R_x \cdot n_2 \, ds \quad \text{mit dem Normalenvektor } \vec{n} = (n_1, n_2)^t \\
&= - \int_{\Omega} h^t \cdot \nabla T \cdot (\nabla d)^t \cdot \nabla R^{\perp} + d \cdot h^t \cdot D^2 T \cdot \nabla R^{\perp} + h^t \cdot \nabla T \cdot \underbrace{d \cdot (-R_{yx} + R_{xy})}_{=0} \, dx dy + \oint_{\partial\Omega} h^t \cdot \nabla T \cdot d \cdot (\nabla R^{\perp})^t \cdot \vec{n} \, ds \\
&= - \int_{\Omega} h^t \cdot \nabla T \cdot (\nabla d)^t \cdot \nabla R^{\perp} + d \cdot h^t \cdot D^2 T \cdot \nabla R^{\perp} \, dx dy + \oint_{\partial\Omega} h^t \cdot \nabla T \cdot d \cdot (\nabla R^{\perp})^t \cdot \vec{n} \, ds
\end{aligned}$$

Ersetzt man den linken Summanden in $\mathcal{D}'_{R,T}(\phi)h$ durch das Ergebnis der partiellen Integration, so folgt

$$\mathcal{D}'_{R,T}(\phi)h = -2 \cdot \int_{\Omega} h^t \cdot \nabla T^{\phi} \cdot (\nabla d)^t \cdot \nabla R^{\perp} \, d\vec{x} + 2 \cdot \oint_{\partial\Omega} h^t \cdot \nabla T^{\phi} \cdot d \cdot (\nabla R^{\perp})^t \cdot \vec{n} \, ds.$$

Die Randwertaufgabe zu \mathcal{D} im \mathbb{R}^2

Ist ϕ ein (lokales) Minimum von $\mathcal{D}_{R,T}$, dann gilt $\mathcal{D}'_{R,T}(\phi)h = 0$ für jede Richtung $h \in H_2(\Omega)^2$. Für das Oberflächenintegral gibt es dabei zwei Möglichkeiten. Entweder ist $h = 0$ auf $\partial\Omega$ (Dirichlet Randbedingung) oder der verbleibende Term ist Null. Die Randwertaufgabe (RWA) zum Funktional $\mathcal{D}_{R,T}$ ist

$$\begin{aligned}
\vec{f}_{\Omega}^{2D} &:= -2 \cdot \underbrace{((\nabla d)^t \cdot \nabla R^{\perp})}_{\in \mathbb{R}} \cdot \underbrace{\nabla T^{\phi}}_{\in \mathbb{R}^2} = \vec{0} \quad \text{auf } \Omega \\
\vec{f}_{\partial\Omega}^{2D} &:= 2 \cdot \underbrace{(d \cdot \vec{n}^t \cdot \nabla R^{\perp})}_{\in \mathbb{R}} \cdot \underbrace{\nabla T^{\phi}}_{\in \mathbb{R}^2} = \vec{0} \quad \text{auf } \partial\Omega.
\end{aligned} \tag{RWA zu } d \text{ im } \mathbb{R}^2 \text{ mit Neumann kompatibler Randbedingung}$$

Diese Randbedingung ist bei Glättung mit der Wärmeleitungsgleichung mit Neumann Randbedingung automatisch erfüllt. Diese Eigenschaft wird im Kapitel zur vollständigen Randwertaufgabe 4 auf Seite 19 genauer besprochen.

Wird im Raum $H_2^D(\Omega)^2$ gearbeitet, d.h. $\vec{h}, \vec{u} \in H_2^D(\Omega)^2$, so ist $h(\vec{x}) = 0 \forall \vec{x} \in \partial\Omega$. Dafür gilt aber auch für das Verschiebungsfeld $\vec{u} = \vec{0}$ auf $\partial\Omega$ und damit $\phi(\vec{x}) = \vec{x} \forall \vec{x} \in \partial\Omega$, d.h. ϕ ist die Identität auf dem Rand von Ω , also $T^\phi(\vec{x}) = T(\vec{x}) \forall \vec{x} \in \partial\Omega$. Somit passen R und T^ϕ am Rand genauso gut oder schlecht zueinander wie R und T . Es liegen Dirichlet Randbedingungen vor

$$\begin{aligned} \vec{f}_\Omega^{2D} &:= -2 \cdot \underbrace{((\nabla d)^t \cdot \nabla R^\perp)}_{\in \mathbb{R}} \cdot \underbrace{\nabla T^\phi}_{\in \mathbb{R}^2} = \vec{0} \quad \text{auf } \Omega \\ &\quad \vec{u} = \vec{0} \quad \text{auf } \partial\Omega. \end{aligned} \quad (\text{RWA zu } d \text{ im } \mathbb{R}^2 \text{ mit Dirichlet Randbedingung})$$

Dabei ist

$$\begin{aligned} \nabla d &= \nabla((\nabla T^\phi)^t \cdot D\phi \cdot \nabla R^\perp) \\ &= \left(\frac{\partial}{\partial x_1}((\nabla T^\phi)^t \cdot D\phi \cdot \nabla R^\perp), \frac{\partial}{\partial x_2}((\nabla T^\phi)^t \cdot D\phi \cdot \nabla R^\perp) \right)^t \\ &= \left((\nabla T^\phi)^t \cdot (D\phi)_{x_1} \cdot \nabla R^\perp \right) + \left(D^2 T^\phi \cdot D\phi \cdot D\phi \cdot \nabla R^\perp \right) + \left((\nabla T^\phi)^t \cdot D\phi \cdot D^2 R^\perp \right)^t. \end{aligned}$$

Hier ist zu sehen, dass \vec{f}_Ω invariant unter Farbinvertierung in R oder T ist. Wird das Vorzeichen bei ∇R oder ∇T invertiert, so hat ∇d umgekehrtes Vorzeichen. In der Gleichung von \vec{f}_Ω heben sich die beiden Vorzeichen gegenseitig auf. Ebenso ist $\vec{f}_{\partial\Omega}$ invariant unter Farbinvertierung.

2.1.2 Herleitung der RWA zum Abstandsmaß im dreidimensionalen Fall

Die Herleitung der Randwertaufgabe im dreidimensionalen Fall verläuft nach dem gleichen Prinzip, wie beim zweidimensionalen Fall. Die Rechnungen sind im Detail allerdings deutlich verschieden. Daher werden sie hier komplett dargestellt.

Sei

$$\phi(\vec{u}) : \Omega \rightarrow \mathbb{R}^3, \quad \phi(\vec{u})(\vec{x}) := \begin{pmatrix} \phi^1(u_1)(\vec{x}) \\ \phi^2(u_2)(\vec{x}) \\ \phi^3(u_3)(\vec{x}) \end{pmatrix} = \begin{pmatrix} x_1 - u_1(\vec{x}) \\ x_2 - u_2(\vec{x}) \\ x_3 - u_3(\vec{x}) \end{pmatrix}$$

die Koordinatentransformation des Templatebildes. Im folgenden wird der Übersichtlichkeit halber nur $\phi(\vec{x})$ statt $\phi(\vec{u})(\vec{x})$ geschrieben und $(\nabla T)(\phi(\vec{x})) = \nabla T^\phi(\vec{x})$ abgekürzt. Zur Wiederholung, das oben eingeführte Abstandsmaß lautet im dreidimensionalen Fall

$$\begin{aligned} d(R, T, \phi; \vec{x}) &:= \sqrt{\left((\nabla T(\phi(\vec{x})) \times \nabla R(\vec{x}))^t \cdot (\nabla T(\phi(\vec{x})) \times \nabla R(\vec{x})) \right)} \\ &= \sqrt{\underbrace{\left(R_{x_2} \cdot T(\phi)_{x_3} - R_{x_3} \cdot T(\phi)_{x_2} \right)^2}_{=: d_1(R, T, \phi)} + \underbrace{\left(R_{x_3} \cdot T(\phi)_{x_1} - R_{x_1} \cdot T(\phi)_{x_3} \right)^2}_{=: d_2(R, T, \phi)} + \underbrace{\left(R_{x_1} \cdot T(\phi)_{x_2} - R_{x_2} \cdot T(\phi)_{x_1} \right)^2}_{=: d_3(R, T, \phi)}} \end{aligned}$$

Das globale Abstandsfunktional $\mathcal{D}(\phi)$ im \mathbb{R}^3

Der Gesamtabstand der Bilder R und T entsteht durch Aufsummieren der lokalen Abstände mit dem Funktional

$$\mathcal{D}_{R, T}(\phi) := \int_\Omega d^2(R, T, \phi; \vec{x}) \, d\vec{x} = \int_\Omega \left((\nabla T(\phi(\vec{x})) \times \nabla R(\vec{x}))^t \cdot (\nabla T(\phi(\vec{x})) \times \nabla R(\vec{x})) \right) \, d\vec{x}.$$

Das Funktional $\mathcal{D}_{R, T}(\phi)$ ist bezüglich ϕ zu minimieren. Die weitere Behandlung des Funktionals lässt sich auf drei zweidimensionale Fälle zurückführen

$$\begin{aligned} \mathcal{D}_{R, T}(\phi) &:= \int_\Omega d^2(R, T, \phi; \vec{x}) \, d\vec{x} \\ &= \int_\Omega d_1^2(R, T, \phi; \vec{x}) + d_2^2(R, T, \phi; \vec{x}) + d_3^2(R, T, \phi; \vec{x}) \, d\vec{x} \\ &= \int_\Omega d_1^2(R, T, \phi; \vec{x}) \, d\vec{x} + \int_\Omega d_2^2(R, T, \phi; \vec{x}) \, d\vec{x} + \int_\Omega d_3^2(R, T, \phi; \vec{x}) \, d\vec{x}. \end{aligned}$$

Mit $\phi = (\phi^1, \phi^2, \phi^3)^t$ gilt

$$\begin{aligned}
d_1(R, T, \phi; x) &= \frac{\partial}{\partial x_2} R(x) \cdot \frac{\partial}{\partial x_3} T(\phi(x)) - \frac{\partial}{\partial x_3} R(x) \cdot \frac{\partial}{\partial x_2} T(\phi(x)) \\
&= \frac{\partial}{\partial x_2} R(x) \cdot (\nabla T)^t(\phi(x)) \cdot \frac{\partial}{\partial x_3} \phi(x) - \frac{\partial}{\partial x_3} R(x) \cdot (\nabla T)^t(\phi(x)) \cdot \frac{\partial}{\partial x_2} \phi(x) \\
&= R_{x_2} \cdot \left(T_{x_1}^\phi \cdot \frac{\partial}{\partial x_3} \phi^1 + T_{x_2}^\phi \cdot \frac{\partial}{\partial x_3} \phi^2 + T_{x_3}^\phi \cdot \frac{\partial}{\partial x_3} \phi^3 \right) - R_{x_3} \cdot \left(T_{x_1}^\phi \cdot \frac{\partial}{\partial x_2} \phi^1 + T_{x_2}^\phi \cdot \frac{\partial}{\partial x_2} \phi^2 + T_{x_3}^\phi \cdot \frac{\partial}{\partial x_2} \phi^3 \right) \\
&= (T_{x_1}^\phi, T_{x_2}^\phi, T_{x_3}^\phi) \cdot \begin{pmatrix} \frac{\partial}{\partial x_1} \phi^1 & \frac{\partial}{\partial x_2} \phi^1 & \frac{\partial}{\partial x_3} \phi^1 \\ \frac{\partial}{\partial x_1} \phi^2 & \frac{\partial}{\partial x_2} \phi^2 & \frac{\partial}{\partial x_3} \phi^2 \\ \frac{\partial}{\partial x_1} \phi^3 & \frac{\partial}{\partial x_2} \phi^3 & \frac{\partial}{\partial x_3} \phi^3 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ -R_{x_3} \\ R_{x_2} \end{pmatrix} \\
&= (\nabla T^\phi)^t \cdot D\phi \cdot (e_1 \times \nabla R)
\end{aligned}$$

$$\begin{aligned}
d_2(R, T, \phi; x) &= \frac{\partial}{\partial x_3} R(x) \cdot \frac{\partial}{\partial x_1} T(\phi(x)) - \frac{\partial}{\partial x_1} R(x) \cdot \frac{\partial}{\partial x_3} T(\phi(x)) \\
&= \frac{\partial}{\partial x_3} R(x) \cdot (\nabla T)^t(\phi(x)) \cdot \frac{\partial}{\partial x_1} \phi(x) - \frac{\partial}{\partial x_1} R(x) \cdot (\nabla T)^t(\phi(x)) \cdot \frac{\partial}{\partial x_3} \phi(x) \\
&= R_{x_3} \cdot \left(T_{x_1}^\phi \cdot \frac{\partial}{\partial x_1} \phi^1 + T_{x_2}^\phi \cdot \frac{\partial}{\partial x_1} \phi^2 + T_{x_3}^\phi \cdot \frac{\partial}{\partial x_1} \phi^3 \right) - R_{x_1} \cdot \left(T_{x_1}^\phi \cdot \frac{\partial}{\partial x_3} \phi^1 + T_{x_2}^\phi \cdot \frac{\partial}{\partial x_3} \phi^2 + T_{x_3}^\phi \cdot \frac{\partial}{\partial x_3} \phi^3 \right) \\
&= (T_{x_1}^\phi, T_{x_2}^\phi, T_{x_3}^\phi) \cdot \begin{pmatrix} \frac{\partial}{\partial x_1} \phi^1 & \frac{\partial}{\partial x_2} \phi^1 & \frac{\partial}{\partial x_3} \phi^1 \\ \frac{\partial}{\partial x_1} \phi^2 & \frac{\partial}{\partial x_2} \phi^2 & \frac{\partial}{\partial x_3} \phi^2 \\ \frac{\partial}{\partial x_1} \phi^3 & \frac{\partial}{\partial x_2} \phi^3 & \frac{\partial}{\partial x_3} \phi^3 \end{pmatrix} \cdot \begin{pmatrix} R_{x_3} \\ 0 \\ -R_{x_1} \end{pmatrix} \\
&= (\nabla T^\phi)^t \cdot D\phi \cdot (e_2 \times \nabla R)
\end{aligned}$$

$$\begin{aligned}
d_3(R, T, \phi; x) &= \frac{\partial}{\partial x_1} R(x) \cdot \frac{\partial}{\partial x_2} T(\phi(x)) - \frac{\partial}{\partial x_2} R(x) \cdot \frac{\partial}{\partial x_1} T(\phi(x)) \\
&= \frac{\partial}{\partial x_1} R(x) \cdot (\nabla T)^t(\phi(x)) \cdot \frac{\partial}{\partial x_2} \phi(x) - \frac{\partial}{\partial x_2} R(x) \cdot (\nabla T)^t(\phi(x)) \cdot \frac{\partial}{\partial x_1} \phi(x) \\
&= R_{x_1} \cdot \left(T_{x_1}^\phi \cdot \frac{\partial}{\partial x_2} \phi^1 + T_{x_2}^\phi \cdot \frac{\partial}{\partial x_2} \phi^2 + T_{x_3}^\phi \cdot \frac{\partial}{\partial x_2} \phi^3 \right) - R_{x_2} \cdot \left(T_{x_1}^\phi \cdot \frac{\partial}{\partial x_1} \phi^1 + T_{x_2}^\phi \cdot \frac{\partial}{\partial x_1} \phi^2 + T_{x_3}^\phi \cdot \frac{\partial}{\partial x_1} \phi^3 \right) \\
&= (T_{x_1}^\phi, T_{x_2}^\phi, T_{x_3}^\phi) \cdot \begin{pmatrix} \frac{\partial}{\partial x_1} \phi^1 & \frac{\partial}{\partial x_2} \phi^1 & \frac{\partial}{\partial x_3} \phi^1 \\ \frac{\partial}{\partial x_1} \phi^2 & \frac{\partial}{\partial x_2} \phi^2 & \frac{\partial}{\partial x_3} \phi^2 \\ \frac{\partial}{\partial x_1} \phi^3 & \frac{\partial}{\partial x_2} \phi^3 & \frac{\partial}{\partial x_3} \phi^3 \end{pmatrix} \cdot \begin{pmatrix} -R_{x_2} \\ R_{x_1} \\ 0 \end{pmatrix} \\
&= (\nabla T^\phi)^t \cdot D\phi \cdot (e_3 \times \nabla R)
\end{aligned}$$

Das lässt sich zu $d_j(R, T, \phi) = (\nabla T^\phi)^t \cdot D\phi \cdot (e_j \times \nabla R)$, $j \in \{1, 2, 3\}$ zusammenfassen.

Die Richtungsableitung des Abstandsfunktionals \mathcal{D} im \mathbb{R}^3

Das Abstandsfunktional $\mathcal{D}_{R,T}$ ist bezüglich ϕ zu minimieren. Dazu wird im folgenden die Abstiegsrichtung hergeleitet.

Analog zum Fall im \mathbb{R}^2 folgt mit $h \in H_2(\Omega)^3$ beliebig gewählt für $j \in \{1, 2, 3\}$

$$\begin{aligned}
\frac{\partial}{\partial \phi} \int_{\Omega} d_j(R, T, \phi; \vec{x}) \, d\vec{x} \, h &= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{d_j(R, T, \phi + \varepsilon \cdot h; \vec{x}) - d_j(R, T, \phi; \vec{x})}{\varepsilon} \, d\vec{x} \\
&= \int_{\Omega} ((\nabla T^\phi)^t \cdot Dh + h^t \cdot D^2 T^\phi \cdot D\phi) \cdot (e_j \times \nabla R) \, d\vec{x}.
\end{aligned}$$

$$\begin{aligned}
\mathcal{D}'_{R,T}(\phi)h &= \frac{\partial}{\partial \phi} \left(\int_{\Omega} d_1^2(R, T, \phi; \vec{x}) \, d\vec{x} + \int_{\Omega} d_2^2(R, T, \phi; \vec{x}) \, d\vec{x} + \int_{\Omega} d_3^2(R, T, \phi; \vec{x}) \, d\vec{x} \right) h \\
&= 2 \cdot \sum_{j=1}^3 \int_{\Omega} d_j \cdot ((\nabla T^\phi)^t \cdot Dh + h^t \cdot D^2 T^\phi \cdot D\phi) \cdot (e_j \times \nabla R) \, d\vec{x}
\end{aligned}$$

nach partieller Integration (vgl. Lemma 2.1 auf Seite 11)

$$\begin{aligned}
&= -2 \cdot \int_{\Omega} h^t \cdot \nabla T^\phi \cdot (d_{1,x_3} \cdot R_{x_2} - d_{1,x_2} \cdot R_{x_3} + d_{2,x_1} \cdot R_{x_3} - d_{2,x_3} \cdot R_{x_1} + d_{3,x_2} \cdot R_{x_1} - d_{3,x_1} \cdot R_{x_2}) \, d\vec{x} \\
&\quad + 2 \cdot \oint_{\partial\Omega} h^t \cdot \nabla T^\phi \cdot (\vec{d} \times \nabla R)^t \cdot \vec{n} \, ds
\end{aligned}$$

Die Randwertaufgabe zu \mathcal{D}

Ist ϕ ein (lokales) Minimum von \mathcal{D} , dann gilt $\mathcal{D}'_{R,T}(\phi)h = 0$ für jede Richtung $h \in H_2(\Omega)^3$. D.h. die beiden Integrale müssen unabhängig von einander und unabhängig von h verschwinden. Dazu gibt es zwei Möglichkeiten. Die Randwertaufgabe (RWA) zum Funktional \mathcal{D} ist also mit Neumann kompatibler Randbedingung

$$\begin{aligned} \vec{f}_\Omega^{3D} &:= -2 \cdot (d_{1,x_3} \cdot R_{x_2} - d_{1,x_2} \cdot R_{x_3} + d_{2,x_1} \cdot R_{x_3} - d_{2,x_3} \cdot R_{x_1} + d_{3,x_2} \cdot R_{x_1} - d_{3,x_1} \cdot R_{x_2}) \cdot \nabla T^\phi = 0 \quad \text{auf } \Omega \\ \vec{f}_{\partial\Omega}^{3D} &:= 2 \cdot (n^t \cdot (\vec{d} \times \nabla R)) \cdot \nabla T^\phi = 0 \quad \text{auf } \partial\Omega \end{aligned}$$

und im Fall mit $H_2^D(\Omega)^3$ folgt die Dirichlet Randbedingung

$$\begin{aligned} \vec{f}_\Omega^{3D} &:= -2 \cdot (d_{1,x_3} \cdot R_{x_2} - d_{1,x_2} \cdot R_{x_3} + d_{2,x_1} \cdot R_{x_3} - d_{2,x_3} \cdot R_{x_1} + d_{3,x_2} \cdot R_{x_1} - d_{3,x_1} \cdot R_{x_2}) \cdot \nabla T^\phi = \vec{0} \quad \text{auf } \Omega \\ \vec{u} &= \vec{0} \quad \text{auf } \partial\Omega. \end{aligned}$$

Hier ist zu erkennen, dass das dreidimensionale \vec{f}_Ω aus einer Summe der drei zweidimensionalen \vec{f}_Ω zu den Hyperebenen besteht, die von je zwei Einheitsvektoren aufgespannt werden und in Richtung des dritten Einheitsvektors verschoben sind. Daher kann die dreidimensionale Bildanpassung bei diesem Maß als zweidimensionale Bildanpassung, die über alle Ebenen zeitgleich erfolgt, aufgefasst werden.

Lemma 2.1. *Durch partielle Integration verschwindet der Term Dh .*

$$\begin{aligned} & \int_{\Omega} \sum_{j=1}^3 d^j \cdot (\nabla T)^t \cdot Dh \cdot (e_j \times \nabla R) \, dx dy dz \\ &= - \int_{\Omega} h^t \left(\nabla T \cdot (d_z^1 \cdot R_y - d_y^1 \cdot R_z + d_x^2 \cdot R_z - d_z^2 \cdot R_x + d_y^3 \cdot R_x - d_x^3 \cdot R_y) + D^2 T \cdot (\vec{d} \times \nabla R) \right) \, dx dy dz \\ & \quad + \oint_{\partial\Omega} h^t \cdot \nabla T \cdot (\vec{d} \times \nabla R)^t \cdot \vec{n} \, ds \end{aligned}$$

Beweis. Der Beweis erfolgt durch partielle Integration. Im folgenden wird die Abkürzung $Q^j := (e_j \times \nabla R)$ verwendet. $\vec{n} = (n_1, n_2, n_3)^t$ ist hier der Normalenvektor im jeweiligen Randpunkt.

$$\begin{aligned} & \int_{\Omega} \sum_{j=1}^3 d^j \cdot (\nabla T)^t \cdot Dh \cdot (e_j \times \nabla R) \, dx dy dz = \int_{\Omega} \sum_{j=1}^3 d^j \cdot (\nabla T)^t \cdot Dh \cdot Q^j \, dx dy dz \\ &= \int_{\Omega} \sum_{j=1}^3 d^j \left((T_x h_x^1 + T_y h_x^2 + T_z h_x^3) Q_1^j + (T_x h_y^1 + T_y h_y^2 + T_z h_y^3) Q_2^j + (T_x h_z^1 + T_y h_z^2 + T_z h_z^3) Q_3^j \right) \, dx dy dz \\ &= - \int_{\Omega} \sum_{j=1}^3 d_x^j (T_x h^1 + T_y h^2 + T_z h^3) Q_1^j + d_y^j (T_x h^1 + T_y h^2 + T_z h^3) Q_2^j + d_z^j (T_x h^1 + T_y h^2 + T_z h^3) Q_3^j \\ & \quad + d^j \left((T_{xx} h^1 + T_{yy} h^2 + T_{zz} h^3) Q_1^j + (T_{xy} h^1 + T_{yy} h^2 + T_{zy} h^3) Q_2^j + (T_{xz} h^1 + T_{yz} h^2 + T_{zz} h^3) Q_3^j \right) \\ & \quad + d^j \left((T_x h^1 + T_y h^2 + T_z h^3) (Q_1^j)_x + (T_x h^1 + T_y h^2 + T_z h^3) (Q_2^j)_y + (T_x h^1 + T_y h^2 + T_z h^3) (Q_3^j)_z \right) \, dx dy dz \\ & \quad + \oint_{\partial\Omega} \sum_{j=1}^3 d_j \cdot h^t \cdot \nabla T \cdot (e_j \times \nabla R)^t \cdot \vec{n} \, ds \\ &= - \int_{\Omega} \sum_{j=1}^3 h^t \cdot \nabla T \cdot (\nabla d^j)^t \cdot (e_j \times \nabla R) + h^t \cdot D^2 T \cdot (d_j \cdot e_j \times \nabla R) + h^t \cdot \nabla T \cdot d^j \cdot (\nabla^t (e_j \times \nabla R)) \, dx dy dz \\ & \quad + \oint_{\partial\Omega} h^t \cdot \nabla T \cdot (\vec{d} \times \nabla R)^t \cdot \vec{n} \, ds \end{aligned}$$

Zwei kleine Nebenrechnungen mit dem ersten und dem dritten Summanden, sowie $\vec{d} := (d^1, d^2, d^3)^t$ liefern die Behauptung.

$$\begin{aligned} & \sum_{j=1}^3 (\nabla d^j)^t \cdot (e_j \times \nabla R) = (\nabla d^1)^t \cdot (e_1 \times \nabla R) + (\nabla d^2)^t \cdot (e_2 \times \nabla R) + (\nabla d^3)^t \cdot (e_3 \times \nabla R) \\ &= (\nabla d^1)^t \cdot \begin{pmatrix} 0 \\ -R_z \\ R_y \end{pmatrix} + (\nabla d^2)^t \cdot \begin{pmatrix} R_z \\ 0 \\ -R_x \end{pmatrix} + (\nabla d^3)^t \cdot \begin{pmatrix} -R_y \\ R_x \\ 0 \end{pmatrix} \\ &= d_z^1 \cdot R_y - d_y^1 \cdot R_z + d_x^2 \cdot R_z - d_z^2 \cdot R_x + d_y^3 \cdot R_x - d_x^3 \cdot R_y \\ & \sum_{j=1}^3 d^j \cdot \nabla^t (e_j \times \nabla R) = \sum_{j=1}^3 d^j \cdot \left(\frac{\partial}{\partial x} (e_j \times \nabla R) + \frac{\partial}{\partial y} (e_j \times \nabla R) + \frac{\partial}{\partial z} (e_j \times \nabla R) \right) \\ &= d^1 \cdot (0 - R_{zy} + R_{yz}) + d^2 \cdot (R_{zx} + 0 - R_{xz}) + d^3 \cdot (-R_{yx} + R_{xy} + 0) \\ &= 0 \quad \text{da die Reihenfolge der Ableitungen vertauscht werden darf.} \end{aligned}$$

□

2.2 Ein alternatives Abstandsmaß

Das Maß d minimiert den Senkrechtanteil der Gradienten von R und T die sich überschneiden. Dadurch kann es passieren, dass die Kanten von T auf strukturlose Stellen in R registriert werden. Die Summe über die Abstände aller Punkte ist dann klein, obwohl die Bilder „falsch“ registriert sind. Dieses Phänomen veranlasst, das folgenden Maß \tilde{d} zu betrachten. Die Idee ist, nicht den Senkrechtanteil zu minimieren, sondern den nicht-parallel Anteil zu minimieren. Zu beachten ist, dass der Nullvektor per Definition auf jedem Vektor senkrecht steht.

$$\begin{aligned}\tilde{d}(R, T; \vec{x}) &:= \sqrt{\left(\frac{\|\nabla R(\vec{x})\|_2^2 + \|\nabla T(\vec{x})\|_2^2}{2}\right)^2 - (\nabla R(\vec{x})^t \cdot \nabla T(\vec{x}))^2} \\ &= \sqrt{\left(\frac{\nabla R(\vec{x})^t \cdot \nabla R(\vec{x}) + \nabla T(\vec{x})^t \cdot \nabla T(\vec{x})}{2}\right)^2 - (\nabla R(\vec{x})^t \cdot \nabla T(\vec{x}))^2}\end{aligned}$$

Der Term $\nabla R(\vec{x})^t \cdot \nabla T(\vec{x})$ ist der Parallel-Anteil der beiden Gradienten. Das Quadrat dient dazu, das Vorzeichen los zu werden, ohne dass die Differenzierbarkeit verloren geht, wie dies z.B. beim Betrag der Fall ist. $\frac{\nabla R(\vec{x})^t \cdot \nabla R(\vec{x}) + \nabla T(\vec{x})^t \cdot \nabla T(\vec{x})}{2}$ ist der Mittelwert der inneren Produkte jedes Gradienten mit sich selbst. Das Quadrat dient der Anpassung an den Subtraktions-Term. Dieses Abstandsmaß ist aufgrund der dimensionsunabhängigen Formulierung unmittelbar für beliebige Dimensionen verwendbar.

Das Abstandsmaß \tilde{d} besitzt einige nennenswerte Eigenschaften:

1. $\tilde{d}(R, R) = 0$, d.h. jedes Bild hat zu sich selbst den Abstand Null.
2. $\tilde{d}(R, T) = \tilde{d}(T, R)$, d.h. es ist symmetrisch in den Argumenten.
3. $\tilde{d}(R, C) = \frac{1}{2} \cdot \|\nabla R\|_2^2$ für jedes Bild C mit $\nabla C = 0$.
4. $\tilde{d}(R, T + C) = \tilde{d}(R, T)$ für jedes Bild C mit $\nabla C = 0$. \tilde{d} ist unabhängig von der absoluten Helligkeit.

Die dritte Eigenschaft macht den Unterschied zu dem ersten Maß d aus. Während $d(R, 0) = 0$ ist, gilt $\tilde{d}(R, 0) \neq 0$ falls $\nabla R \neq 0$. Liegt eine Kante im Templatebild T an einer Stelle, wo im Referenzbild R eine Äquipotentialfläche ist, so wird an dieser Stelle ein Abstand größer Null gemessen. Dieser hängt außerdem von der Stärke der Kante $\|\nabla T(\vec{x})\|_2$ ab, so dass durch Bildrauschen und Messungenauigkeit entstandene Kanten nicht so stark ins Gewicht fallen, wie deutlich erkennbare Umrisse eines Objektes. An den Kanten in R wird $\tilde{d}(R, T)$ genau dann minimal, wenn in T eine deckungsgleiche Kante verläuft. Umgekehrt wird bei Punkten aus Äquipotentialflächen in R $\tilde{d}(R, T) = \tilde{d}(0, T)$ genau dann klein, wenn $\|\nabla T\|_2$ klein wird, also in Bereichen mit schwachen Strukturen, also keine nennenswerte Kante.

Neben diesem Vorteil hat \tilde{d} auch einen Nachteil. Betrachten wir für $\mu \in \mathbb{R}$

$$\tilde{d}(R, \mu \cdot R) = \sqrt{\left(\frac{\|\nabla R\|_2^2 + \|\mu \cdot \nabla R\|_2^2}{2}\right)^2 - (\nabla R^t \cdot \mu \cdot \nabla R)^2} = \|\nabla R\|_2^2 \cdot \frac{1}{2} \cdot |\mu^2 - 1|.$$

Sind in beiden Bildern (R und T) die Kanten deckungsgleich und die Norm der Gradienten ($\|\nabla R\|_2$ und $\|\nabla T\|_2$) bedingt durch die Färbung verschieden, so wird \tilde{d} i.a. nicht Null. Dieses Phänomen wird später an Beispielbildern genauer untersucht. Zunächst wird die Randwertaufgabe zu \tilde{d} hergeleitet.

Die Richtungsableitung des Abstandsfunktionals \tilde{D}

Das Abstandsfunktional zum Maß \tilde{d} ist

$$\tilde{D}_{R,T}(\phi) := \int_{\Omega} \tilde{d}^2(R, T \circ \phi; \vec{x}) \, d\vec{x} = \int_{\Omega} \left(\frac{\|\nabla R(\vec{x})\|_2^2 + \|\nabla(T \circ \phi)(\vec{x})\|_2^2}{2} \right)^2 - (\nabla R(\vec{x})^t \cdot \nabla(T \circ \phi)(\vec{x}))^2 \, d\vec{x}.$$

Es wird die Richtungsableitung von $\tilde{D}_{R,T}$ hergeleitet. Sei $h \in H_2(\Omega)^n$ eine beliebig gewählte Richtung.

$$\begin{aligned}\frac{\partial}{\partial \phi} \tilde{D}_{R,T}(\phi)h &= \frac{\partial}{\partial \phi} \int_{\Omega} \left(\frac{1}{4}(\nabla R^t \nabla R)^2 + \frac{1}{2}(\nabla R^t \nabla(T \circ \phi))^2 + \frac{1}{4}(\nabla(T \circ \phi)^t \nabla(T \circ \phi))^2 - (\nabla R^t \nabla(T \circ \phi))^2 \right) \, d\vec{x} \, h \\ &= \frac{\partial}{\partial \phi} \int_{\Omega} \left(\frac{1}{4}(\nabla R^t \cdot \nabla R)^2 + \frac{1}{4}(\nabla(T \circ \phi)^t \cdot \nabla(T \circ \phi))^2 - \frac{1}{2}(\nabla R^t \cdot \nabla(T \circ \phi))^2 \right) \, d\vec{x} \, h\end{aligned}$$

Die Rechnung für den ersten Summanden.

$$\begin{aligned}
& \frac{\partial}{\partial \phi} \int_{\Omega} \left(\frac{1}{4} (\nabla(T \circ \phi))^t \cdot \nabla(T \circ \phi) \right)^2 d\vec{x} h \\
&= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{\left(\nabla(T \circ (\phi + \varepsilon \cdot h))^t \cdot \nabla(T \circ (\phi + \varepsilon \cdot h)) \right)^2 - \left(\nabla(T \circ \phi)^t \cdot \nabla(T \circ \phi) \right)^2}{4 \cdot \varepsilon} d\vec{x} \\
&= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{\left(((D\phi + \varepsilon \cdot Dh) \cdot \nabla T^{\phi + \varepsilon \cdot h})^t \cdot ((D\phi + \varepsilon \cdot Dh) \cdot \nabla T^{\phi + \varepsilon \cdot h}) \right)^2 - \left((D\phi \cdot \nabla T^{\phi})^t \cdot (D\phi \cdot \nabla T^{\phi}) \right)^2}{4 \cdot \varepsilon} d\vec{x} \\
&= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{\| (D\phi + \varepsilon \cdot Dh) \cdot \nabla T^{\phi + \varepsilon \cdot h} \|_2^4 - \| D\phi \cdot \nabla T^{\phi} \|_2^4}{4 \cdot \varepsilon} d\vec{x}
\end{aligned}$$

Taylorentwicklung um T^{ϕ}

$$\begin{aligned}
&= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{\| (D\phi + \varepsilon \cdot Dh) \cdot (\nabla T^{\phi} + \varepsilon \cdot D^2 T^{\phi} \cdot h + \mathcal{O}(\varepsilon^2)) \|_2^4 - \| D\phi \cdot \nabla T^{\phi} \|_2^4}{4 \cdot \varepsilon} d\vec{x} \\
&= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{\| D\phi \cdot \nabla T^{\phi} + \varepsilon \cdot Dh \cdot \nabla T^{\phi} + \varepsilon \cdot D\phi \cdot D^2 T^{\phi} \cdot h + \mathcal{O}(\varepsilon^2) \|_2^4 - \| D\phi \cdot \nabla T^{\phi} \|_2^4}{4 \cdot \varepsilon} d\vec{x}
\end{aligned}$$

ausmultiplizieren und vernachlässigen der Terme mit ε^2 und höher

$$= \int_{\Omega} \| D\phi \cdot \nabla T^{\phi} \|_2^2 \cdot \left((\nabla T^{\phi})^t \cdot Dh + h^t \cdot D^2 T^{\phi} \cdot D\phi \right) \cdot D\phi \cdot \nabla T^{\phi} d\vec{x}$$

Die Rechnung für den zweiten Summanden.

$$\begin{aligned}
& \frac{\partial}{\partial \phi} \int_{\Omega} \left(\frac{1}{2} (\nabla R^t \cdot \nabla(T \circ \phi))^2 \right) d\vec{x} h \\
&= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{\left(\nabla R^t \cdot \nabla(T \circ (\phi + \varepsilon \cdot h)) \right)^2 - \left(\nabla R^t \cdot \nabla(T \circ \phi) \right)^2}{2 \cdot \varepsilon} d\vec{x} \\
&= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{\left(\nabla R^t \cdot (D\phi + \varepsilon \cdot Dh) \cdot \nabla T^{\phi + \varepsilon \cdot h} \right)^2 - \left(\nabla R^t \cdot D\phi \cdot \nabla T^{\phi} \right)^2}{2 \cdot \varepsilon} d\vec{x} \\
&= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{\left(\nabla R^t \cdot (D\phi + \varepsilon \cdot Dh) \cdot (\nabla T^{\phi} + \varepsilon \cdot D^2 T^{\phi} \cdot h + \mathcal{O}(\varepsilon^2)) \right)^2 - \left(\nabla R^t \cdot D\phi \cdot \nabla T^{\phi} \right)^2}{2 \cdot \varepsilon} d\vec{x} \\
&= \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{\left(\nabla R^t \cdot (D\phi \cdot \nabla T^{\phi} + \varepsilon \cdot Dh \cdot \nabla T^{\phi} + \varepsilon \cdot D\phi \cdot D^2 T^{\phi} \cdot h + \mathcal{O}(\varepsilon^2)) \right)^2 - \left(\nabla R^t \cdot D\phi \cdot \nabla T^{\phi} \right)^2}{2 \cdot \varepsilon} d\vec{x}
\end{aligned}$$

ausmultiplizieren und vernachlässigen der Terme mit ε^2 und höher

$$\begin{aligned}
&= \int_{\Omega} \nabla R^t \cdot D\phi \cdot \nabla T^{\phi} \cdot \nabla R^t \cdot (Dh \cdot \nabla T^{\phi} + D\phi \cdot D^2 T^{\phi} \cdot h) d\vec{x} \\
&= \int_{\Omega} (\nabla T^{\phi})^t \cdot D\phi \cdot \nabla R \cdot (h^t \cdot D^2 T^{\phi} \cdot D\phi + (\nabla T^{\phi})^t \cdot Dh) \cdot \nabla R d\vec{x}
\end{aligned}$$

Damit ergibt sich

$$\begin{aligned}
\frac{\partial}{\partial \phi} \tilde{D}_{R,T}(\phi)h &= \int_{\Omega} \| D\phi \cdot \nabla T^{\phi} \|_2^2 \cdot \left((\nabla T^{\phi})^t \cdot Dh + h^t \cdot D^2 T^{\phi} \right) \cdot D\phi \cdot \nabla T^{\phi} \\
&\quad - (\nabla T^{\phi})^t \cdot D\phi \cdot \nabla R \cdot (h^t \cdot D^2 T^{\phi} \cdot D\phi + (\nabla T^{\phi})^t \cdot Dh) \cdot \nabla R d\vec{x}.
\end{aligned}$$

partielle Integration (vgl. Lemma 2.2 auf Seite 15) liefert

$$\begin{aligned}
&= \int_{\Omega} h^t \cdot \nabla T^{\phi} \cdot \left[((\nabla T^{\phi})^t \cdot D\phi \cdot \nabla R) \cdot \left(\sum_{k=1}^n R_{x_k x_k} \right) + \nabla((\nabla T^{\phi})^t \cdot D\phi \cdot \nabla R) \cdot \nabla R \right. \\
&\quad \left. - \| D\phi \cdot \nabla T^{\phi} \|_2^2 \cdot \left(\underbrace{(1, \dots, 1)}_n \cdot \nabla(D\phi \cdot \nabla T) \right) - \nabla((\nabla T^{\phi})^t \cdot D\phi \cdot D\phi \cdot \nabla T^{\phi}) \cdot D\phi \cdot \nabla T^{\phi} \right] d\vec{x} \\
&\quad + \oint_{\partial \Omega} h^t \cdot \nabla T^{\phi} \cdot (\nabla T^{\phi})^t \cdot D\phi \cdot \left(-\nabla R \cdot \nabla R^t + D\phi \cdot \nabla T^{\phi} \cdot (\nabla T^{\phi})^t \cdot D\phi \right) \cdot \vec{n} ds
\end{aligned}$$

Die Randwertaufgabe zu $\tilde{\mathcal{D}}_{R,T}$

Die Ableitung vom Funktional $\tilde{\mathcal{D}}_{R,T}$ verschwindet im Optimalpunkt $\tilde{\phi}$. D.h. die Integrale müssen unabhängig von h verschwinden. Damit ergibt sich die Randwertaufgabe

$$\begin{aligned} \vec{f}_{\Omega} := & \nabla T^{\phi} \cdot \left[((\nabla T^{\phi})^t \cdot D\phi \cdot \nabla R) \cdot \left(\sum_{k=1}^n R_{x_k x_k} \right) + \nabla((\nabla T^{\phi})^t \cdot D\phi \cdot \nabla R) \cdot \nabla R \right. \\ & \left. - \|D\phi \cdot \nabla T^{\phi}\|_2^2 \cdot \underbrace{((1, \dots, 1))}_n \cdot \nabla(D\phi \cdot \nabla T^{\phi}) - \nabla((\nabla T^{\phi})^t \cdot D\phi \cdot D\phi \cdot \nabla T^{\phi}) \cdot D\phi \cdot \nabla T^{\phi} \right] = \vec{0} \text{ auf } \Omega \end{aligned}$$

mit Neumann kompatibler Randbedingung

$$\vec{f}_{\partial\Omega} := \nabla T^{\phi} \cdot (\nabla T^{\phi})^t \cdot D\phi \cdot \left(-\nabla R \cdot \nabla R^t + D\phi \cdot \nabla T^{\phi} \cdot (\nabla T^{\phi})^t \cdot D\phi \right) \cdot \vec{n} = \vec{0} \text{ auf } \partial\Omega$$

oder Dirichlet Randbedingung

$$\vec{u} = \vec{0} \text{ auf } \partial\Omega.$$

Lemma 2.2. *Mit partieller Integration gilt die Identität*

$$\begin{aligned} & \int_{\Omega} (\nabla T^\phi)^t \cdot D\phi \cdot \nabla R \cdot (\nabla T^\phi)^t \cdot Dh \cdot \nabla R \, d\vec{x} \\ &= - \int_{\Omega} h^t \cdot \left[(D^2 T^\phi \cdot D\phi \cdot \nabla R + \nabla T^\phi \cdot \left(\sum_{k=1}^n R_{x_k x_k} \right)) \cdot (\nabla T^\phi)^t \cdot D\phi \cdot \nabla R + \nabla T^\phi \cdot \nabla \left((\nabla T^\phi)^t \cdot D\phi \cdot \nabla R \right)^t \cdot \nabla R \right] d\vec{x} \\ & \quad + \oint_{\partial\Omega} h^t \cdot \nabla T^\phi \cdot (\nabla T^\phi)^t \cdot D\phi \cdot \nabla R \cdot \nabla R^t \cdot \vec{n} \, ds. \end{aligned}$$

Beweis. Der Beweis erfolgt durch Ausmultiplizieren und Durchführen der partiellen Integration. Dh wird hochintegriert. $\vec{n} = (n_1, n_2, n_3)^t$ ist der Normalenvektor im jeweiligen Randpunkt. Um es halbwegs übersichtlich zu halten, wird hier genau der dreidimensionale Fall gerechnet. Dabei wird ersichtlich, dass die gleiche Rechnung für beliebige Dimensionen durchführbar ist.

$$\begin{aligned} & \int_{\Omega} (\nabla T^\phi)^t \cdot D\phi \cdot \nabla R \cdot (\nabla T^\phi)^t \cdot Dh \cdot \nabla R \, dx dy dz \\ &= \int_{\Omega} \left((\nabla T^\phi)^t D\phi \nabla R \right) \left((T_x h_x^1 + T_y h_x^2 + T_z h_x^3) R_x + (T_x h_y^1 + T_y h_y^2 + T_z h_y^3) R_y + (T_x h_z^1 + T_y h_z^2 + T_z h_z^3) R_z \right) dx dy dz \end{aligned}$$

partielle Integration, Dh wird hochintegriert

$$\begin{aligned} &= - \int_{\Omega} h^t \left[\left((T_{xx}, T_{yx}, T_{zx})^t R_x + \nabla T^\phi R_{xx} \right) \left((\nabla T^\phi)^t D\phi \nabla R \right) + \nabla T R_x \frac{\partial}{\partial x} \left((\nabla T^\phi)^t D\phi \nabla R \right) \right. \\ & \quad + \left((T_{xy}, T_{yy}, T_{zy})^t R_y + \nabla T^\phi R_{yy} \right) \left((\nabla T^\phi)^t D\phi \nabla R \right) + \nabla T R_y \frac{\partial}{\partial y} \left((\nabla T^\phi)^t D\phi \nabla R \right) \\ & \quad \left. + \left((T_{xz}, T_{yz}, T_{zz})^t R_z + \nabla T^\phi R_{zz} \right) \left((\nabla T^\phi)^t D\phi \nabla R \right) + \nabla T R_z \frac{\partial}{\partial z} \left((\nabla T^\phi)^t D\phi \nabla R \right) \right] dx dy dz \\ & \quad + \oint_{\partial\Omega} h^t \cdot \nabla T^\phi \cdot (\nabla T^\phi)^t \cdot D\phi \cdot \nabla R \cdot \nabla R^t \cdot \vec{n} \, ds \\ &= - \int_{\Omega} h^t \left[\left((T_{xx}, T_{xy}, T_{xz})^t R_x + (T_{yx}, T_{yy}, T_{yz})^t R_y + (T_{zx}, T_{zy}, T_{zz})^t R_z \right) \left((\nabla T^\phi)^t D\phi \nabla R \right) \right. \\ & \quad + \nabla T^\phi (R_{xx} + R_{yy} + R_{zz}) \left((\nabla T^\phi)^t D\phi \nabla R \right) \\ & \quad \left. + \nabla T^\phi \left(R_x \frac{\partial}{\partial x} \left((\nabla T^\phi)^t D\phi \nabla R \right) + R_y \frac{\partial}{\partial y} \left((\nabla T^\phi)^t D\phi \nabla R \right) + R_z \frac{\partial}{\partial z} \left((\nabla T^\phi)^t D\phi \nabla R \right) \right) \right] dx dy dz \\ & \quad + \oint_{\partial\Omega} h^t \cdot \nabla T^\phi \cdot (\nabla T^\phi)^t \cdot D\phi \cdot \nabla R \cdot \nabla R^t \cdot \vec{n} \, ds \\ &= - \int_{\Omega} h^t \left[D^2 T^\phi D\phi \nabla R \left((\nabla T^\phi)^t D\phi \nabla R \right) + \left((R_{xx} + R_{yy} + R_{zz}) \left((\nabla T^\phi)^t D\phi \nabla R \right) \right. \right. \\ & \quad \left. \left. + \nabla \left((\nabla T^\phi)^t D\phi \nabla R \right)^t \nabla R \right) \nabla T^\phi \right] dx dy dz \\ & \quad + \oint_{\partial\Omega} h^t \cdot \nabla T^\phi \cdot (\nabla T^\phi)^t \cdot D\phi \cdot \nabla R \cdot \nabla R^t \cdot \vec{n} \, ds \end{aligned}$$

□

3 Das Regularisierungsfunktional

Die Koordinatentransformation lässt sich durch das Verschiebungsfeld $\vec{u}(\vec{x})$ als $\phi(\vec{u})(\vec{x}) = \vec{x} - \vec{u}(\vec{x})$ darstellen. Ist das Bild nur leicht verzerrt, so werden die \vec{u} von benachbarten \vec{x} ähnliche Richtung und Länge haben. Die Stärke der Verzerrung kann auf verschiedene Weise gemessen werden:

1. mit dem Gradienten

$$S_1(\vec{u}) = \frac{1}{2} \int_{\Omega} \sum_{i=1}^n \|\nabla u_i(\vec{x})\|_2^2 d\vec{x} \quad (\text{harmonisch})$$

2. mit dem Laplace-Operator $\Delta = \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2}$

$$S_2(\vec{u}) = \frac{1}{2} \int_{\Omega} \sum_{i=1}^n (\Delta u_i(\vec{x}))^2 d\vec{x} \quad (\text{biharmonisch})$$

3. mit dem Navier-Lamé Operator aus der physikalischen Elastizitätstheorie, der auch die gemischten Ableitungen enthält

$$S_3(\vec{u}) = \frac{1}{2} \int_{\Omega} \sum_{i=1}^n \left(\Delta u_i(\vec{x}) + \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i(\vec{x}) \right)^2 d\vec{x} \quad (\text{Navier-Lamé})$$

Das vollständige Minimierungsproblem aus der Einleitung ist

$$\text{minimiere}_{\vec{u} \in \mathbb{R}^n \times \Omega} \mathcal{D}_{R,T}(\phi(\vec{u})) + \alpha \cdot S(\vec{u}).$$

In diesem Abschnitt wird nur der rechte Summand betrachtet, also

$$\text{minimiere}_{\vec{u} \in \mathbb{R}^n \times \Omega} S(\vec{u}).$$

Zu dem Funktional $S(\vec{u})$ gibt es eine äquivalente Randwertaufgabe. Diese wird jetzt für $S_1(\vec{u})$ hergeleitet.

$$S_1(\vec{u}) = \frac{1}{2} \int_{\Omega} \|\nabla \vec{u}(\vec{x})\|_2^2 d\vec{x} = \frac{1}{2} \int_{\Omega} (\nabla \vec{u}(\vec{x}))^t \cdot (\nabla \vec{u}(\vec{x})) d\vec{x}$$

Die Richtungsableitung von S_1 im Punkt $\vec{x} \in \Omega$ in Richtung von $\vec{h} = (h_1, \dots, h_n)^t \in H_2(\Omega)^n$ ist

$$\begin{aligned} \frac{\partial}{\partial \vec{u}} S_1(\vec{u}) h &= \lim_{\varepsilon \rightarrow 0} \frac{S_1(\vec{u} + \varepsilon \cdot \vec{h}) - S_1(\vec{u})}{\varepsilon} \\ &= \frac{1}{2} \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \sum_{i=1}^n (\nabla u_i(\vec{x}) + \varepsilon \cdot \nabla h_i(\vec{x}))^t \cdot (\nabla u_i(\vec{x}) + \varepsilon \cdot \nabla h_i(\vec{x})) - (\nabla u_i(\vec{x}))^t \cdot (\nabla u_i(\vec{x})) d\vec{x} \\ &= \frac{1}{2} \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \sum_{i=1}^n 2 \cdot \varepsilon \cdot (\nabla h_i(\vec{x}))^t \cdot (\nabla u_i(\vec{x})) + \varepsilon^2 \cdot (\nabla h_i(\vec{x}))^t \cdot (\nabla h_i(\vec{x})) d\vec{x} \\ &= \frac{1}{2} \int_{\Omega} \sum_{i=1}^n 2 \cdot (\nabla h_i(\vec{x}))^t \cdot (\nabla u_i(\vec{x})) d\vec{x} = \sum_{i=1}^n \int_{\Omega} (\nabla h_i(\vec{x}))^t \cdot (\nabla u_i(\vec{x})) d\vec{x} \end{aligned}$$

partielle Integration

$$= \sum_{i=1}^n - \int_{\Omega} h_i(\vec{x}) \cdot \Delta u_i(\vec{x}) d\vec{x} + \oint_{\partial\Omega} h_i(\vec{x}) \cdot (\nabla u_i(\vec{x}))^t \cdot \vec{n} ds$$

\vec{n} ist der Normalenvektor auf dem Rand von Ω . Ist \vec{u} ein lokales Minimum von S_1 , so ist jede Richtungsableitung Null. Entweder ist im Oberflächenintegral h_i oder $\frac{\partial}{\partial \vec{n}} u_i$ Null. Damit folgt für $i = 1, \dots, n$ die Randwertaufgabe mit der jeweiligen Randbedingung

$$\begin{aligned} -\Delta u_i &= 0 && \text{auf } \Omega \\ u_i &= 0 && \text{auf } \partial\Omega (\text{Dirichlet}) \\ \frac{\partial}{\partial \vec{n}} u_i &= 0 && (\text{Neumann}). \end{aligned} \quad (\text{harmonische RWA})$$

Das ist die homogene harmonische Randwertaufgabe mit Dirichlet bzw. Neumann Randbedingung.

Die Rechnung wird analog für S_2 geführt.

$$\begin{aligned}
\frac{\partial}{\partial \vec{u}} S_2(\vec{u})h &= \lim_{\varepsilon \rightarrow 0} \frac{S_2(\vec{u} + \varepsilon \cdot \vec{h}) - S_2(\vec{u})}{\varepsilon} \\
&= \frac{1}{2} \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \sum_{i=1}^n (\Delta u_i(\vec{x}) + \varepsilon \cdot \Delta h_i(\vec{x})) \cdot (\Delta u_i(\vec{x}) + \varepsilon \cdot \Delta h_i(\vec{x})) - (\Delta u_i(\vec{x})) \cdot (\Delta u_i(\vec{x})) \, d\vec{x} \\
&= \frac{1}{2} \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \sum_{i=1}^n 2 \cdot (\varepsilon \cdot \Delta h_i(\vec{x})) \cdot (\Delta u_i(\vec{x})) + (\varepsilon \cdot \Delta h_i(\vec{x})) \cdot (\varepsilon \cdot \Delta h_i(\vec{x})) \, d\vec{x} \\
&= \frac{1}{2} \int_{\Omega} \sum_{i=1}^n 2 \cdot (\Delta h_i(\vec{x})) \cdot (\Delta u_i(\vec{x})) \, d\vec{x} = \sum_{i=1}^n \int_{\Omega} (\Delta h_i(\vec{x})) \cdot (\Delta u_i(\vec{x})) \, d\vec{x} \\
&\text{partielle Integration} \\
&= \sum_{i=1}^n \int_{\Omega} -(\nabla h_i(\vec{x}))^t \cdot (\nabla \Delta u_i(\vec{x}))^t \, d\vec{x} + \oint_{\partial\Omega} (\Delta u_i(\vec{x})) \cdot (\nabla h_i(\vec{x}))^t \cdot \vec{n} \, ds \\
&\text{nochmal partiell integriert} \\
&= \sum_{i=1}^n \int_{\Omega} (h_i(\vec{x}))^t \cdot (\Delta^2 u_i(\vec{x}))^t \, d\vec{x} + \oint_{\partial\Omega} (\Delta u_i(\vec{x})) \cdot (\nabla h_i(\vec{x}))^t \cdot \vec{n} \, ds - \oint_{\partial\Omega} (h_i(\vec{x})) \cdot (\nabla \Delta u_i(\vec{x}))^t \cdot \vec{n} \, ds
\end{aligned}$$

Damit die beiden Randintegrale verschwinden, gibt es vier Möglichkeiten. Zwei davon sind die homogene biharmonische Randwertaufgabe mit Dirichlet Randbedingung

$$\begin{aligned}
\Delta^2 u_i &= 0 \quad \text{auf } \Omega \\
\Delta u_i &= 0 \quad \text{auf } \partial\Omega \\
-u_i &= 0 \quad \text{auf } \partial\Omega
\end{aligned}
\quad (\text{biharmonische RWA mit Dirichlet Randbedingung})$$

und die homogene biharmonische Randwertaufgabe mit Neumann Randbedingung

$$\begin{aligned}
\Delta^2 u_i &= 0 \quad \text{auf } \Omega \\
-\frac{\partial}{\partial \vec{n}} \Delta u_i &= 0 \quad \text{auf } \partial\Omega \\
\frac{\partial}{\partial \vec{n}} u_i &= 0 \quad \text{auf } \partial\Omega
\end{aligned}
\quad (\text{biharmonische RWA mit Neumann Randbedingung})$$

für $i = 1, \dots, n$.

Die Randwertaufgabe und die Herleitung zum Navier-Lamé Operator S_3 steht im Anhang A auf Seite 81.

Bei der Rechnung wurde nur der Regularisierungsterm aus dem Funktional I betrachtet. Im vorigen Abschnitt wurde das Maß d angegeben und obige Rechnung für den Abstandssummanden \mathcal{D} bzw. $\tilde{\mathcal{D}}$ in I durchgeführt. Im nächsten Kapitel werden die beiden Teilergebnisse zur Randwertaufgabe von I zusammengesetzt. Aus diesem Grund wurden die Vorzeichen in der Randwertaufgabe oberhalb stehen gelassen.

4 Die vollständige Randwertaufgabe

In den beiden letzten Abschnitten wurde die Richtungsableitung zum Abstandsterm \mathcal{D} und dem Regularisierungsterm S aus I einzeln hergeleitet. Diese Aufteilung diente der Übersichtlichkeit. Hier werden die beiden Teilergebnisse nun zusammengesetzt zur nicht-linearen Randwertaufgabe vom Funktional I .

Zur Erinnerung, die eigentliche Aufgabe besteht darin, das Funktional

$$I(\vec{u}) := \mathcal{D}_{R,T}(\phi(\vec{u})) + \alpha \cdot S(\vec{u}) \, d\vec{x} \quad (\text{Funktional})$$

zu minimieren. Die bisherigen Ergebnisse lassen sich zu der Randwertaufgabe von I zusammensetzen. Sei wieder n die Dimension von Ω . Im Fall von Maß d und dem Regularisierungsterm S_1 muss im lokalen Minimum in jeder Richtung $\vec{h} = (h^1, \dots, h^n)^t \in H_2(\Omega)^n$ gelten

$$\begin{aligned} \frac{\partial}{\partial \vec{u}} \left((\mathcal{D}(\phi(\vec{u})) + \alpha \cdot S_1(\vec{u})) \vec{h} \right) &= 0 \\ \Leftrightarrow \frac{\partial}{\partial \vec{u}} (\mathcal{D}(\phi(\vec{u})) \vec{h} + \alpha \cdot \frac{\partial}{\partial \vec{u}} (S_1(\vec{u})) \vec{h}) &= 0 \end{aligned}$$

und die zugehörige Randwertaufgabe kann aus den früheren Überlegungen zu

$$\begin{aligned} (\vec{f}_\Omega)_i - \alpha \cdot \Delta u_i &= 0 \quad \text{auf } \Omega & -\alpha \cdot \Delta u_i &= -(\vec{f}_\Omega)_i \quad \text{auf } \Omega \\ (\vec{f}_{\partial\Omega})_i + \alpha \cdot u_i &= 0 \quad \text{auf } \partial\Omega & \Leftrightarrow \alpha \cdot u_i &= -(\vec{f}_{\partial\Omega})_i \quad \text{auf } \partial\Omega \quad \text{bei Dirichlet Randbedingung} \\ (\vec{f}_{\partial\Omega})_i + \alpha \cdot \frac{\partial}{\partial \vec{n}} u_i &= 0 \quad \text{auf } \partial\Omega & \alpha \cdot \frac{\partial}{\partial \vec{n}} u_i &= -(\vec{f}_{\partial\Omega})_i \quad \text{auf } \partial\Omega \quad \text{bei Neumann Randbedingung} \end{aligned}$$

zusammengesetzt werden.

Im Fall der Dirichlet Randbedingung gilt $\vec{f}_{\partial\Omega} = \vec{0}$. Also lautet die RWA dann

$$\begin{aligned} -\alpha \cdot \Delta u_i &= -(\vec{f}_\Omega)_i \quad \text{auf } \Omega \\ u_i &= 0 \quad \text{auf } \partial\Omega. \end{aligned} \quad (\text{vollständige harmonische RWA mit Dirichlet Randbedingung})$$

Sind die Bilder R und T mit der Wärmeleitungsgleichung mit Neumann Randbedingung (vgl. 5.2 auf Seite 22) geglättet, so gilt

$$\begin{aligned} (\nabla R)^t \cdot \vec{n} &= \frac{\partial}{\partial \vec{n}} R = 0 \quad \text{auf } \partial\Omega \\ (\nabla(T \circ \phi))^t \cdot \vec{n} &= \frac{\partial}{\partial \vec{n}} (T \circ \phi) = 0 \quad \text{auf } \partial\Omega \end{aligned}$$

D.h. im zweidimensionalen Fall mit Maß d kann es keinen Senkrechteil zwischen R und $T \circ \phi$ auf Randpunkten geben. Demzufolge ist $d(R, T \circ \phi; \vec{x}) = 0 \, \forall \vec{x} \in \partial\Omega$ und damit $\vec{f}_{\partial\Omega}^{2D} = 0$. Im dreidimensionalen Fall kommt der Faktor $(\vec{n}^t \cdot (\vec{d} \times \nabla R))$ in $\vec{f}_{\partial\Omega}^{3D}$ vor. Ist $\vec{n} = \pm \vec{n}_i$, so ist $d_j = 0 \, \forall j \neq i$. Durch Ausschreiben des Kreuzproduktes folgt $(\vec{d}_i \times \nabla R)_i = 0$ und damit gilt $(\vec{n}^t \cdot (\vec{d} \times \nabla R)) = 0$.

Bei Maß \vec{d} ist es offensichtlich, dass $\vec{f}_{\partial\Omega}^{3D} = 0$ gilt, denn dort kommt in jedem Summanden entweder $\nabla R^t \cdot \vec{n}$ oder $\nabla(T \circ \phi)^t \cdot \vec{n}$ vor.

Für die Rechnung vereinfacht sich die RWA mit Neumann Randbedingung dann zu

$$\begin{aligned} -\alpha \cdot \Delta u_i &= -(\vec{f}_\Omega)_i \quad \text{auf } \Omega \\ \frac{\partial}{\partial \vec{n}} u_i &= 0 \quad \text{auf } \partial\Omega. \end{aligned} \quad (\text{vollständige harmonische RWA mit Neumann Randbedingung})$$

Falls keine oder eine andere Glättung gewählt wird, so gilt diese Vereinfachung z.B. auch, falls die Bilder R und T im Randbereich 0 (schwarz) sind, so dass die Ableitung auf dem Rand Null ist und $\vec{f}_{\partial\Omega} = 0$ gilt.

Beim biharmonischen Operator ergibt sich mit analogen Überlegungen die Randwertaufgabe

$$\begin{aligned} \alpha \cdot \Delta^2 u_i &= -(\vec{f}_\Omega)_i \quad \text{auf } \Omega \\ \Delta u_i &= 0 \quad \text{auf } \partial\Omega \\ u_i &= 0 \quad \text{auf } \partial\Omega \end{aligned} \quad (\text{vollständige biharmonische RWA mit Dirichlet Randbedingung})$$

$$\begin{aligned} \alpha \cdot \Delta^2 u_i &= -(\vec{f}_\Omega)_i \quad \text{auf } \Omega \\ \frac{\partial}{\partial \vec{n}} \Delta u_i &= 0 \quad \text{auf } \partial\Omega \\ \frac{\partial}{\partial \vec{n}} u_i &= 0 \quad \text{auf } \partial\Omega \end{aligned} \quad (\text{vollständige biharmonische RWA mit Neumann Randbedingung})$$

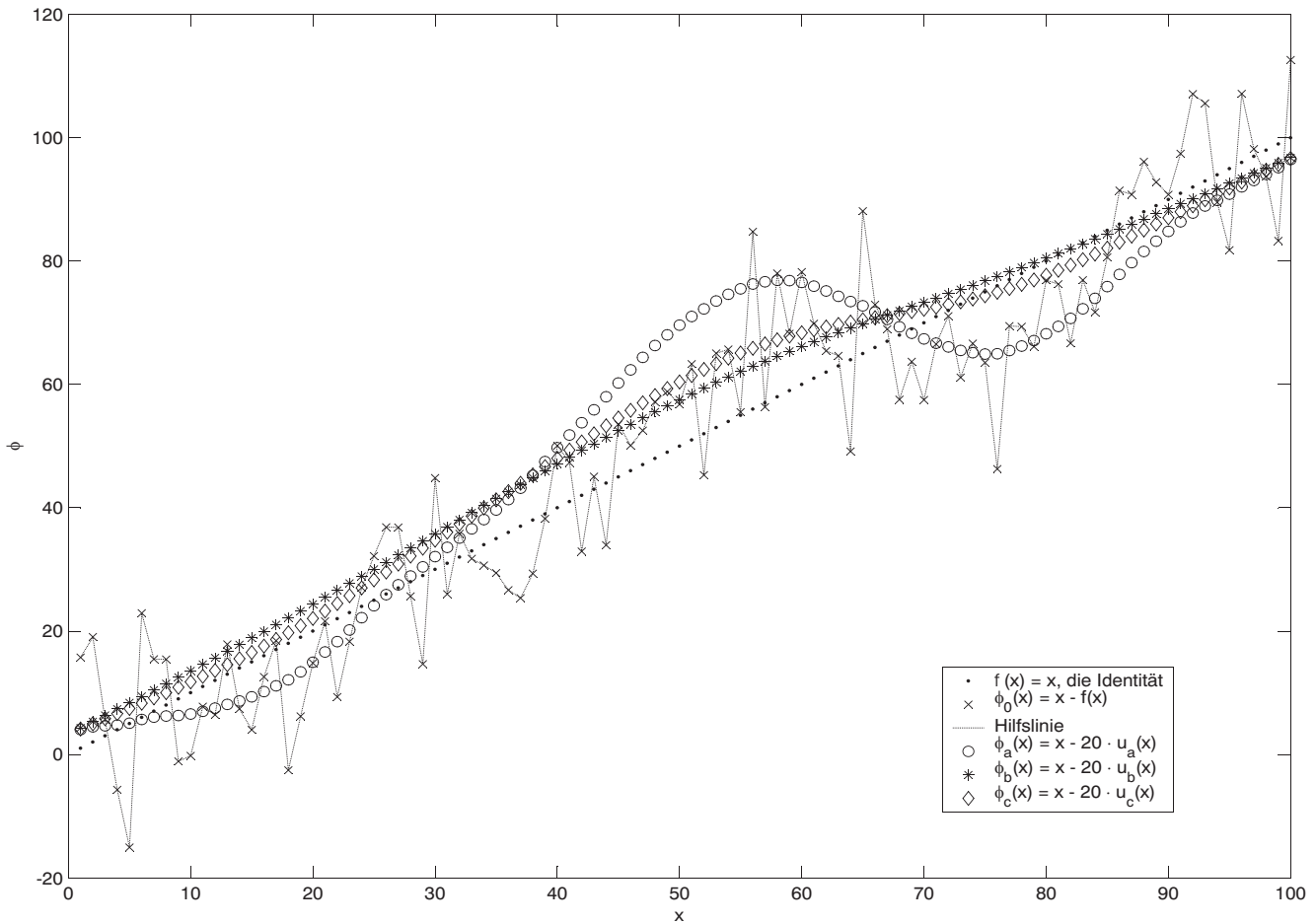


Abbildung 4.1: Wirkung der Regularisierung im eindimensionalen auf ein Bild mit 100 Punkten. u_a ist die Lösung der harmonischen und u_c der biharmonischen Randwertaufgabe mit Dirichlet-Randbedingung. Bei u_b wurden die Eigenwerte der harmonischen RWA im Frequenzraum hoch drei halbe genommen. Dementsprechend liegt die Wirkung zwischen a und c . Der Faktor 20 wurde nur zur besseren Übersicht eingefügt. Es ist deutlich zu erkennen, dass die regularisierte Kurve nur die wesentliche Form der Kräfte f wiedergibt. Das lokal starke gezitterte wird geglättet. Wird die Kraft f direkt als Suchrichtung gewählt, so verändert sich die Reihenfolge der Punkte. Mit der regularisierten Suchrichtung werden einige Punkte dichter aneinander gedrückt und andere weiter auseinander gezerrt, die Reihenfolge bleibt dabei allerdings erhalten.

Anmerkung zu den Randwertaufgaben

So wie die Randwertaufgabe hier formuliert ist, ist die Lösung $\vec{u} = (u_1, u_1, \dots, u_n)^t$ eine Anstiegsrichtung des Funktionals $I(\vec{u})$. Die benötigte Abstiegsrichtung ist $-\vec{u}$. $\alpha \cdot \Delta u$ lässt sich auch als $\Delta(\alpha \cdot u)$ schreiben. u ist also die Suchrichtung und α ist die Inverse der Schrittweite. Die Wahl von α wird im Abschnitt der Liniensuche 5.4.4 auf Seite 29 beschrieben.

Die Randwertaufgaben oberhalb können aufgrund der Randbedingung die Transformationen aus einem Unterraum der affin-linearen Transformationen nicht wiedergeben. Bei der Dirichlet Randbedingung sind dies alle affin-linearen Transformationen, also Verschiebung, Streckung, Scherung und Drehung. Mit der Neumann Randbedingung sind Verschiebungen möglich, die verbleibenden linearen Transformationen sind aber auch hier nicht mit der Randbedingung verträglich. Diese Transformationen werden von der RWA daher nur approximiert. Wie sich dieses Problem lösen lässt wird in 5.4.1 auf Seite 25 beschrieben.

Die numerische Rechnung

Bei beiden Maßen, d und \tilde{d} hängt die rechte Seite $\vec{f}_\Omega = \vec{f}_\Omega(\vec{u})$ vom aktuellen Verschiebungsfeld \vec{u} ab. Daher ist diese Randwertaufgabe hochgradig nicht-linear. Die numerische Rechnung kann folgendermaßen erfolgen. Zu einem Startwert u^0 wird die rechte Seite f_Ω^0 ausgerechnet. Mit der Lösung der Randwertaufgabe u^1 wird die rechte Seite f_Ω^1 erneut ausgerechnet und dazu erneut die Randwertaufgabe gelöst. Mit Startwert $u^0 = 0$ kann auf diese Weise die Lösung der Randwertaufgabe iterativ genähert werden. In jeder Iteration lässt sich die Randwertaufgabe zu festem f_Ω^k mit der Sinustransformation bei Dirichlet Randbedingung und bei Neumann Randbedingung mit der Cosinustransformation – wie im Abschnitt 5.4.3 auf Seite 26 beschrieben wird – ausrechnen.

5 Die Unterprogramme

5.1 Die Fouriertransformation

Die Fouriertransformation ist ein numerisch sehr effizientes Verfahren zum Lösen von bestimmten Differentialgleichungen. Für eine diskret definierte Funktion $\vec{x} := (x_1, \dots, x_n)^t \in \mathbb{R}^n$ lautet die Definition der eindimensionalen Fouriertransformation ohne Normierung für die j -te Frequenz

$$\text{DFT}_j^{1\text{D}}(\vec{x}) := \sum_{k=1}^n x_k \cdot \exp\left(2 \cdot \pi \cdot i \cdot \frac{(j-1) \cdot (k-1)}{n}\right) \quad \text{für } j = 1, \dots, n. \quad (\text{DFT})$$

Aufgrund der $2\pi i$ -Periodizität der Exponentialfunktion sind einige der Terme $\exp(\cdot)$ gleich. Daher lässt sich die Reihenfolge der Multiplikation und Addition durch Umklammern vertauschen, so dass die Fouriertransformation bereits mit einem Rechenaufwand in $\mathcal{O}(n \cdot \log(n))$ berechnet werden kann. Der Rechenaufwand ist besonders gering, falls $n = 2^l$ mit $l \in \mathbb{N}$ ist.

Falls \vec{x} eine gerade oder ungerade Funktion ist, so ist auch $\text{DFT}(\vec{x})$ eine gerade bzw. ungerade Funktion. Im folgenden werden diese zwei Spezialfälle genauer betrachtet.

5.1.1 Die Cosinustransformation

Sei $\vec{x} := (x_1, x_2, \dots, x_{n-2}, x_{n-1}, x_n, x_{n-1}, x_{n-2}, \dots, x_3, x_2)^t \in \mathbb{R}^{2n-2}$ eine gerade Funktion, dann gilt für $j = 1, 2, \dots, n$

$$\begin{aligned} \text{DFT}_j^{1\text{D}}(\vec{x}) &= \sum_{k=1}^{2n} x_k \cdot \exp\left(2 \cdot \pi \cdot i \cdot \frac{(j-1) \cdot (k-1)}{2 \cdot n - 2}\right) \\ &= x_1 + (-1)^j \cdot x_n + \sum_{k=2}^{n-1} x_k \cdot \left(\exp\left(2\pi i \cdot \frac{(j-1) \cdot (k-1)}{2 \cdot n - 2}\right) + \exp\left(-2\pi i \cdot \frac{(j-1) \cdot (k-1)}{2 \cdot n - 2}\right)\right) \\ &= x_1 + (-1)^j \cdot x_n + 2 \cdot \sum_{k=2}^{n-1} x_k \cdot \cos\left(\pi \cdot \frac{(j-1) \cdot (k-1)}{n-1}\right) \\ &=: \text{DCT}_j^{1\text{D}}((x_1, x_2, x_3, \dots, x_{n-2}, x_{n-1}, x_n)^t) \end{aligned} \quad (\text{DCT})$$

DCT bezeichnet die diskrete Cosinustransformation vom Typ I. Obwohl sich die Rechnung für jedes $n \in \mathbb{N}$ mit einem Aufwand in $\mathcal{O}(n \cdot \log(n))$ berechnen lässt, ist die numerische Rechnung bei $n = 2^l + 1$ mit $l \in \mathbb{N}$ besonders schnell, da dabei im Nenner der Cosinusfunktion eine Zweierpotenz steht und dadurch viele Cosinusterme gleich sind.

Die Umkehrtransformation von der DCT ist wieder die DCT bis auf einen Faktor von $2 \cdot (n-1)$. Also gilt

$$(\text{DCT}^{1\text{D}})^{-1} = \frac{\text{DCT}^{1\text{D}}}{2 \cdot (n-1)}.$$

5.1.2 Die Sinustransformation

Sei $\vec{x} := (0, x_1, x_2, \dots, x_{n-2}, x_{n-1}, x_n, 0, -x_n, -x_{n-1}, -x_{n-2}, \dots, -x_3, -x_2, -x_1)^t \in \mathbb{R}^{2n+2}$ eine ungerade Funktion, dann gilt für $j = 1, \dots, n$

$$\begin{aligned} \text{DFT}_j^{1\text{D}}(\vec{x}) &= \sum_{k=0}^{2n+1} x_k \cdot \exp\left(2 \cdot \pi \cdot i \cdot \frac{j \cdot k}{2 \cdot n + 2}\right) \\ &= \sum_{k=1}^n x_k \cdot \left(\exp\left(2 \cdot \pi \cdot i \cdot \frac{j \cdot k}{2 \cdot n + 2}\right) - \exp\left(-2 \cdot \pi \cdot i \cdot \frac{j \cdot k}{2 \cdot n + 2}\right)\right) \\ &= \frac{2}{i} \cdot \sum_{k=1}^n x_k \cdot \sin\left(\pi \cdot \frac{j \cdot k}{n+1}\right) \\ &=: -i \cdot \text{DST}_j^{1\text{D}}((x_1, x_2, x_3, \dots, x_{n-3}, x_{n-1}, x_n)^t). \end{aligned} \quad (\text{DST})$$

Der Imaginärteil davon wird die diskrete Sinustransformation (DST) genannt. In der Literatur wird diese Variante der Sinustransformation als Type-I-DST bezeichnet. Auch diese Rechnung kann mit einem Aufwand in $\mathcal{O}(n \cdot \log(n))$ berechnet werden. Für $n = 2^l - 1$ mit $l \in \mathbb{N}$ ist die Rechnung der DST besonders schnell.

Die Umkehrtransformation von der DST ist wieder die DST bis auf einen Faktor von $2 \cdot (n + 1)$. Also gilt

$$(\text{DST}^{1\text{D}})^{-1} = \frac{\text{DST}^{1\text{D}}}{2 \cdot (n + 1)}.$$

5.1.3 Trigonometrische Transformation im mehrdimensionalen

Zuvor wurde die diskrete Fourier-, Sinus- und Cosinustransformation im eindimensionalen beschrieben. Im mehrdimensionalen Fall wird die eindimensionale Transformation sukzessive entlang jeder Raumrichtung angewendet. Im dreidimensionalen sieht das dann folgendermaßen aus. Sei $x \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_3}$, dann gilt für $j \in \{1, \dots, n_1\}$, $k \in \{1, \dots, n_2\}$ und $l \in \{1, \dots, n_3\}$

$$\text{DFT}_{j,k,l}^{3\text{D}}(x) := \sum_{a=1}^{n_1} \sum_{b=1}^{n_2} \sum_{c=1}^{n_3} x_{a,b,c} \exp\left(2\pi i \frac{(c-1)(l-1)}{n_3}\right) \exp\left(2\pi i \frac{(b-1)(k-1)}{n_2}\right) \exp\left(2\pi i \frac{(a-1)(j-1)}{n_1}\right).$$

5.2 Glätten mit der Wärmeleitungsgleichung

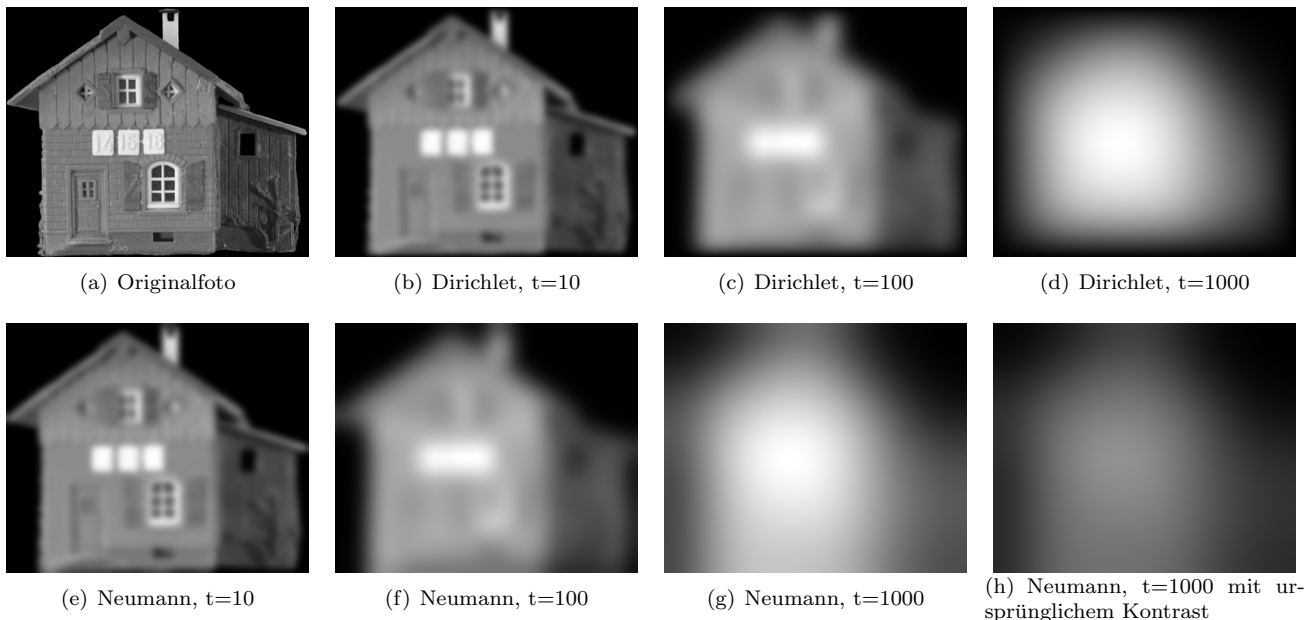


Abbildung 5.1: Glättung mit der Wärmeleitungsgleichung zu verschiedenen Zeitwerten t mit Dirichlet- und Neumannrandbedingung. Zum besseren Erkennen ist der Kontrast bei (b) bis (g) nach dem Glätten erhöht worden. Die Wahl der Randbedingung macht im Randbereich einen erkennbaren Unterschied. So ist in (d) der Rand weiterhin schwarz. Dagegen ist in (g) das Bild über den Rand hinaus verlaufen. Mit zunehmenden t werden Details immer schwächer. Dagegen werden bei $t = 100$ die groben Formen wie der Umriss des Hauses und die Nummern-Tafeln sehr deutlich.

Die Bildregistrierung soll zuerst einmal die groben Strukturen aufeinander ausrichten. Sobald die groben Strukturen übereinstimmen, können die lokalen Strukturen aufeinander angepasst werden. Zu diesem Zweck werden die lokalen Strukturen anfangs durch glätten des Bildes unsichtbar gemacht.

Eine Möglichkeit der Bildglättung bietet die Wärmeleitungsgleichung. Wie der Name schon sagt, beschreibt diese Differentialgleichung die Wärmeleitung in Materie. Helle Bildbereiche werden als warm aufgefasst und dunkle kühl. Zum Zeitpunkt $t = 0$ ist die Wärmeverteilung wie im gegebenen Bild B . Mit Fortschreiten der Zeit t wird sich die Wärme im Bild gleichmäßig verteilen, bis das Bild bei $t = \infty$ einheitlich grau gefärbt ist. Für die zweite Randbedingung gibt es zwei naheliegende Möglichkeiten:

1. Der Rand des Bildes wird auf Temperatur 0 (schwarz) gehalten. Damit ist die Randbedingung auf Dirichlet festgelegt.
2. Das Bild wird entlang des Randes symmetrisch fortgesetzt. Damit ist die Randbedingung Neumannscher Art.

Die Randwertaufgabe der Wärmeleitungsgleichung ist

$$\begin{aligned} \frac{\partial}{\partial t} g(t, \vec{x}) &= \Delta g(t, \vec{x}) \quad \text{auf } \Omega \\ g(t=0, \vec{x}) &= B(\vec{x}) \quad \forall \vec{x} \in \Omega \\ g(\cdot, \vec{x}) &= 0 \quad \forall \vec{x} \in \partial\Omega \quad \text{Dirichlet} \\ \frac{\partial}{\partial \vec{n}} g(\cdot, \vec{x}) &= 0 \quad \forall \vec{x} \in \partial\Omega \quad \text{Neumann.} \end{aligned} \quad (\text{Wärmeleitungsgleichung})$$

Die allgemeine Lösung lässt sich im dreidimensionalen mit dem Separationsansatz

$$g(t, x, y, z) = \varphi(t) \cdot a(x) \cdot b(y) \cdot c(z)$$

herleiten. Sei $N_1 := \{1, 2, 3, \dots, n_1\}$, $N_2 := \{1, 2, 3, \dots, n_2\}$, $N_3 := \{1, 2, 3, \dots, n_3\}$ und $\Omega := N_1 \times N_2 \times N_3$ der Definitionsbereich des dreidimensionalen Bildes.

$$\begin{aligned} g_t = \Delta g &\Leftrightarrow \varphi' \cdot a \cdot b \cdot c = \varphi \cdot a'' \cdot b \cdot c + \varphi \cdot a \cdot b'' \cdot c + \varphi \cdot a \cdot b \cdot c'' \\ &\Rightarrow \frac{\varphi'}{\varphi} = \frac{a''}{a} + \frac{b''}{b} + \frac{c''}{c} =: -\lambda \end{aligned}$$

Mit der Dirichlet Randbedingung sind die Eigenwerte

$$\lambda_{p,q,r}^D = \pi^2 \cdot \left(\left(\frac{p}{n_1 + 1} \right)^2 + \left(\frac{q}{n_2 + 1} \right)^2 + \left(\frac{r}{n_3 + 1} \right)^2 \right), \quad p \in N_1, q \in N_2, r \in N_3$$

und die Eigenfunktionen

$$\begin{aligned} a_p(x) &= \sin\left(\frac{\pi \cdot p \cdot x}{n_1 + 1}\right) \\ b_q(y) &= \sin\left(\frac{\pi \cdot q \cdot y}{n_2 + 1}\right) \\ c_r(z) &= \sin\left(\frac{\pi \cdot r \cdot z}{n_3 + 1}\right) \\ \varphi_{p,q,r}(t) &= \exp(-\lambda_{p,q,r}^D \cdot t) = \exp\left(-\pi^2 \cdot \left(\left(\frac{p}{n_1 + 1} \right)^2 + \left(\frac{q}{n_2 + 1} \right)^2 + \left(\frac{r}{n_3 + 1} \right)^2 \right) \cdot t\right). \end{aligned}$$

Bei der Neumann Randbedingung sind die Eigenwerte

$$\lambda_{p,q,r}^N = \pi^2 \cdot \left(\left(\frac{p-1}{n_1-1} \right)^2 + \left(\frac{q-1}{n_2-1} \right)^2 + \left(\frac{r-1}{n_3-1} \right)^2 \right), \quad p \in N_1, q \in N_2, r \in N_3$$

und die Eigenfunktionen

$$\begin{aligned} a_p(x) &= \cos\left(\frac{\pi \cdot (p-1) \cdot (x-1)}{n_1-1}\right) \\ b_q(y) &= \cos\left(\frac{\pi \cdot (q-1) \cdot (y-1)}{n_2-1}\right) \\ c_r(z) &= \cos\left(\frac{\pi \cdot (r-1) \cdot (z-1)}{n_3-1}\right) \\ \varphi_{p,q,r}(t) &= \exp(-\lambda_{p,q,r}^D \cdot t) = \exp\left(-\pi^2 \cdot \left(\left(\frac{p-1}{n_1-1} \right)^2 + \left(\frac{q-1}{n_2-1} \right)^2 + \left(\frac{r-1}{n_3-1} \right)^2 \right) \cdot t\right). \end{aligned}$$

Damit ist die allgemeine Lösung im Dirichlet Fall

$$g^D(t=0, x, y, z) = \sum_{p,q,r} \alpha_{p,q,r} \cdot \sin\left(\frac{\pi \cdot p \cdot x}{n_1 + 1}\right) \cdot \sin\left(\frac{\pi \cdot q \cdot y}{n_2 + 1}\right) \cdot \sin\left(\frac{\pi \cdot r \cdot z}{n_3 + 1}\right)$$

bzw. im Neumann Fall

$$g^N(t=0, x, y, z) = \sum_{p,q,r} \beta_{p,q,r} \cdot \cos\left(\frac{\pi \cdot (p-1) \cdot (x-1)}{n_1-1}\right) \cdot \cos\left(\frac{\pi \cdot (q-1) \cdot (y-1)}{n_2-1}\right) \cdot \cos\left(\frac{\pi \cdot (r-1) \cdot (z-1)}{n_3-1}\right).$$

Wegen der ersten Randbedingung lassen sich die Koeffizienten $\alpha_{p,q,r}$ mit der Sinustransformation (vgl. 5.1.2 auf Seite 21) aus dem gegebenen Bild B errechnen:

$$\alpha_{p,q,r} = \text{DST}_{p,q,r}^{3D}(B) \cdot \frac{1}{2^3 \cdot (n_1 + 1) \cdot (n_2 + 1) \cdot (n_3 + 1)} = (\text{DST}_{p,q,r}^{3D}(B))^{-1}$$

Die $\beta_{p,q,r}$ werden mit der Cosinustransformation

$$\beta_{p,q,r} = \text{DCT}_{p,q,r}^{3D}(B) \cdot \frac{1}{2^3 \cdot (n_1 - 1) \cdot (n_2 - 1) \cdot (n_3 - 1)} = (\text{DCT}_{p,q,r}^{3D}(B))^{-1}$$

berechnet.

Das verschwommene Bild zum Zeitpunkt t lässt sich schließlich mit

$$\begin{aligned} g^D(t, x, y, z) &= \sum_{p,q,r} \alpha_{p,q,r} \cdot \sin\left(\frac{\pi \cdot p \cdot x}{n_1 + 1}\right) \cdot \sin\left(\frac{\pi \cdot q \cdot y}{n_2 + 1}\right) \cdot \sin\left(\frac{\pi \cdot r \cdot z}{n_3 + 1}\right) \cdot \exp(-\pi^2 \cdot \lambda_{p,q,r}^D \cdot t) \\ &= \text{DST}_{x,y,z}^{3D} \left(\left((\text{DST}_{p,q,r}^{3D}(B))^{-1} \cdot \exp(-\pi^2 \cdot \lambda_{p,q,r}^D \cdot t) \right)_{p=1,\dots,n_1, q=1,\dots,n_2, r=1,\dots,n_3} \right) \end{aligned}$$

bzw.

$$\begin{aligned} g^N(t, x, y, z) &= \sum_{p,q,r} \beta_{p,q,r} \cdot \cos\left(\frac{\pi(p-1)(x-1)}{n_1-1}\right) \cdot \cos\left(\frac{\pi(q-1)(y-1)}{n_2-1}\right) \cdot \cos\left(\frac{\pi(r-1)(z-1)}{n_3-1}\right) \cdot \exp(-\pi^2 \lambda_{p,q,r}^N \cdot t) \\ &= \text{DCT}_{x,y,z}^{3D} \left(\left((\text{DCT}_{p,q,r}^{3D}(B))^{-1} \cdot \exp(-\pi^2 \cdot \lambda_{p,q,r}^N \cdot t) \right)_{p=1,\dots,n_1, q=1,\dots,n_2, r=1,\dots,n_3} \right) \end{aligned}$$

berechnen. Anschaulich gesprochen werden die hohen Frequenzen des Bildes durch die Gewichtung mit der abklingenden Exponentialfunktion gedämpft.

5.3 Interpolation des verformten Bildes

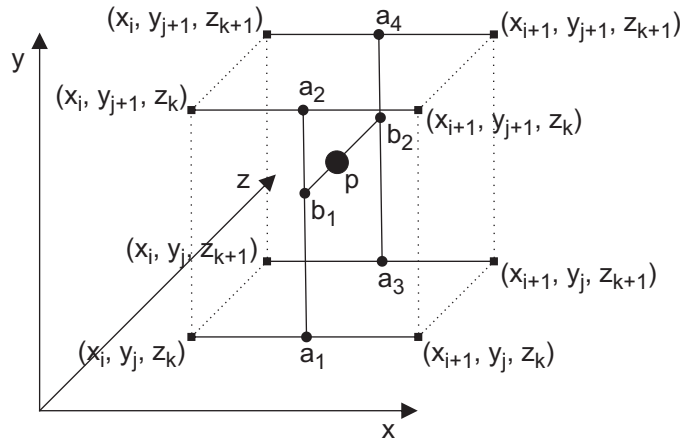


Abbildung 5.2: Trilineare Interpolation als sukzessive eindimensionale lineare Interpolation

In diesem Abschnitt wird beschrieben, wie sich $T(\phi(\vec{u}))$ berechnen lässt. T ist das Template-Bild und $\phi(\vec{u}) : \Omega \rightarrow \mathbb{R}^n$ ist eine stetige Koordinatentransformation. Nun ist $\phi(\vec{u})(\vec{x})$ im allgemeinen kein Gitterpunkt. Somit existiert im diskreten Bild erstmal kein Wert an dieser Stelle. Mit Hilfe der Interpolation lässt sich das diskrete Bild kontinuierlich machen. Allerdings sind die so berechneten Punkte zwischen den Gitterknoten nach einer Regel „ausgedacht“, die sich aus dem Interpolationsalgorithmus ergibt. Ideal wäre hier die Interpolation mit trikubischen Splines. Sie hat den Vorteil, dass die erste und zweite Ableitung des interpolierten Bildes stetig ist. Im dreidimensionalen ist der Rechenaufwand beim Lösen des dabei auftretenden Gleichungssystems allerdings enorm hoch. Aus diesem Grunde wird hier vorerst nur die trilineare Interpolation eingesetzt. Die Ableitungen sind dann zwar unstetig, in der Praxis ist das aber kein Problem, da die Ableitungen dort mit finiten Differenzen gebildet werden.

Die trilineare Interpolation wird praktisch durch mehrere hintereinander geschaltete eindimensionale Interpolationen implementiert. Man stelle sich dazu einen Würfel vor, dessen Ecken die nächstgelegenen Gitterpunkte sind. Dies sind die eckigen Punkte in Abbildung 5.2. Man interpoliere in x -Richtung entlang der vier dick gedruckten Kanten die Hilfspunkte a_i , $i \in \{1, 2, 3, 4\}$. Zwischen a_1 und a_2 wird dann b_1 in y -Richtung linear interpoliert. Analog entsteht b_2 aus a_3 und a_4 . Schließlich wird der gesuchte Punkt p zwischen b_1 und b_2 längs der z -Richtung linear interpoliert.

5.4 Die Suchrichtung

Im Abschnitt 2 auf Seite 5 wurden die „Kräfte“ $\vec{f} = (f^1, f^2, f^3)^t$ hergeleitet, die am Templatebild T greifen und es in Richtung der Form vom Referenzbild R zerren. Im Prinzip ist dieses Kraftfeld die Suchrichtung. Es muss lediglich regularisiert werden, damit die Transformation die Topologie des Bildes beibehält. Dazu wurde bereits das Prinzip der Regularisierung vorgestellt. Dabei wird als Suchrichtung die Lösung einer Randwertaufgabe verwendet, deren rechte Seite die Kräfte f sind. Bedingt durch die Randbedingung befindet sich ein Unterraum der affin-linearen Transformationen nicht im Bildraum der Randwertaufgabe. Die Randwertaufgabe wird diesen Anteil der Transformation der Kräfte \vec{f} daher lediglich approximieren, so weit dies im Bezug auf die Randbedingung möglich ist. Affin-lineare Transformationen sind bezüglich der Schlechtgestellttheit der Bildanpassung vergleichsweise unbedenklich. Daher soll diese Transformationskomponente auch genutzt werden. Um dies zu realisieren wird $f = f_l + f_c$ zerlegt. Dabei ist f_l die Projektion der Kraft f auf den Unterraum der affin-linearen Transformationen \mathcal{L} , der mit der Randbedingung unverträglich ist. Im Fall von Dirichlet Randbedingungen sind das alle affin-linearen Transformationen, wie die Verschiebung, die Drehung, die Streckung und die Scherung. Bei Neumann und periodischer Randbedingung sind es die linearen Funktionen, also die zuvor genannten ohne die Verschiebung.

5.4.1 Die lineare Transformation

Aus den Kräften $f(\vec{x})$ soll der lineare Anteil bestimmt werden. Dazu wird zunächst eine Basis L der linearen Transformationen gebraucht. Diese soll auch numerisch stabil sein. Die naheliegende Variante ist

$$L := [x_1, \dots, x_n, 1]$$

bzw. diskret mit Gitterkonstante h

$$L_h := [x_{1,h}, \dots, x_{n,h}, 1_h] \in \mathbb{R}^{n \times n+1}.$$

Die Projektion P von f_i auf L ist (ohne Normierung)

$$P_L(f_i) := L \cdot L^t \cdot f_i.$$

Die konkrete Rechnung sieht folgendermaßen aus

$$\theta_{i,j} := (L^t \cdot f_i)_j = \frac{\int_{\Omega} f_i(\vec{x}) \cdot x_j \, d\vec{x}}{\int_{\Omega} x_j \cdot x_j \, d\vec{x}}, \quad i, j \in \{1, \dots, n\}$$

und für den Verschiebungsanteil gilt

$$\theta_{i,n+1} := (L^t \cdot f_i)_{n+1} = \frac{\int_{\Omega} f_i(\vec{x}) \, d\vec{x}}{\int_{\Omega} 1 \, d\vec{x}}, \quad i \in \{1, \dots, n\}.$$

Die Integrale im Nenner des Bruches dienen der Normierung, damit lineare Funktionen genau auf sich selbst projiziert werden.

Das Verschiebungsfeld \vec{u}_l der linearen Transformation berechnet sich damit wie folgt

$$\vec{u}_l = \begin{pmatrix} u_{l,1} \\ u_{l,2} \\ \vdots \\ u_{l,n} \end{pmatrix} = \begin{pmatrix} \theta_{1,1} \cdot x_1 + \theta_{1,2} \cdot x_2 + \dots + \theta_{1,n} \cdot x_n + \theta_{1,n+1} \\ \theta_{2,1} \cdot x_1 + \theta_{2,2} \cdot x_2 + \dots + \theta_{2,n} \cdot x_n + \theta_{2,n+1} \\ \vdots \\ \theta_{n,1} \cdot x_1 + \theta_{n,2} \cdot x_2 + \dots + \theta_{n,n} \cdot x_n + \theta_{n,n+1} \end{pmatrix} = \theta \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{pmatrix}.$$

Die Regularisierung ist hier automatisch gegeben, da sich die Koeffizienten $\theta_{i,j}$ durch die Integralformel als (gewichtete) Mittelwerte berechnen.

Im Prinzip lassen sich auf die gleiche Weise auch Polynome höheren Grades aus den f_i entwickeln. Dabei treten dann allerdings die aus der Polynominterpolation bekannten Schwingungsausreißer auf. Wird der Grad der Polynome so groß wie die Dimension der f_i , so erhält man die f_i zurück. Unter dem Aspekt der Regularisierung sind Polynome höheren Grades daher eine schlechte Wahl. Eine weitere Möglichkeit besteht darin, die Projektion auf einen von Splines aufgespannten Raum zu erweitern. Dabei bestimmt die Wahl der Splines und ihrer Knoten die Möglichkeiten der Transformation. Ist die Anzahl der Knoten klein, so können lokal verzerrte Bildstrukturen meist nicht richtig ausgerichtet werden, ist dagegen die Anzahl der Knoten zu groß gewählt, so werden i.a. zu starke lokale Verzerrungen auftreten. Die Wahl der Splines und der Knotenanzahl muss gut auf die erwartete Transformation angepasst werden. Die Spline-Variante wird in dieser Arbeit nicht weiter verfolgt. Sie soll hier nur der Vollständigkeit halber erwähnt sein.

5.4.2 Die nicht-lineare Transformation

An die nicht-lineare Transformation wird der Anspruch gestellt, dass sie zwar nicht-lineare Transformationen zulässt, dabei aber – anschaulich gesprochen – das Bild „nur“ verzerrt und keine Bildpunkte kreuz und quer vertauscht. Benachbarte Punkte im Templatebild T müssen im transformierten Bild T^ϕ ebenfalls benachbart sein. Dies lässt sich mathematisch durch beschränken der Ableitung(en) der Transformationsabbildung ϕ bzw. des Verschiebungsfeldes \vec{u} erreichen. In Kapitel 4 auf Seite 19 wurden dazu bereits geeignete Funktionale angegeben und die zugehörigen Randwertaufgaben hergeleitet. Die Randbedingung verhindert dabei bestimmte Transformationen aus dem Raum der affin-linearen Transformationen. Bei Dirichlet Randbedingungen sind dies alle affin-linearen Transformation inklusive Verschiebung und bei Neumann oder periodischer Randbedingung sind es nur die linearen Transformationen, also ohne die konstante Verschiebung. Der Raum dieser Transformation wird im folgenden mit \mathcal{L} bezeichnet.

Der Anteil von den f_i , der in \mathcal{L} liegt wird im folgenden mit $u_{i,l}$ bezeichnet. $u_{i,l}$ ist also die Projektion von f_i auf \mathcal{L} und lässt sich durch Entwicklung der f_i entlang einer Basis von \mathcal{L} – wie im vorangegangenen Abschnitt beschrieben – berechnen. Damit die Lösung der Randwertaufgabe nicht versucht, den Transformationsanteil aus \mathcal{L} zu approximieren, muss dieser zuvor von den f_i abgezogen werden, d.h. der Orthogonalraum zu \mathcal{L} , also $f_{i,c} := f_i - u_{i,l} = (I - P_{\mathcal{L}})(f_i)$ wird als rechte Seite für die Randwertaufgabe verwendet und die Lösung mit $u_{i,c}$ bezeichnet. Die regularisierte Suchrichtung ist dann $u_i := u_{i,c} + u_{i,l} = u_{i,c} + f_{i,l}$.

5.4.3 Lösen der Randwertaufgabe

Nun wird besprochen, wie sich die verschiedenen Randwertaufgaben konkret lösen lassen. Zuerst wird der harmonische Operator im dreidimensionalen Fall untersucht. Der zweidimensionale Fall lässt sich daraus unmittelbar ablesen und die Verallgemeinerung auf höhere Dimensionen ist ebenfalls ersichtlich. Für diesen Abschnitt werden folgende Definitionen eingeführt:

$$N_1 := \{1, \dots, n_1\}, N_2 := \{1, \dots, n_2\}, N_3 := \{1, \dots, n_3\}$$

$$\Omega := N_1 \times N_2 \times N_3$$

$$\partial\Omega_D := ((N_1 \cup \{0, n_1 + 1\}) \times (N_2 \cup \{0, n_2 + 1\}) \times (N_3 \cup \{0, n_3 + 1\})) \setminus \Omega \text{ (Rand von } \Omega \text{ bei Dirichlet Randbedingung)}$$

$$\partial\Omega_N := (N_1 \times N_2 \times N_3) \setminus (\{2, \dots, n_1 - 1\} \times \{2, \dots, n_2 - 1\} \times \{2, \dots, n_3 - 1\}) \text{ (Rand von } \Omega \text{ bei Neumann Randbedingung)}$$

Weiter sei $\vec{u} := (u_1, u_2, u_3)^t$ und $\vec{f} := \vec{f}_\Omega - \vec{f}_l$ mit dem \vec{f}_Ω aus der Herleitung der rechten Seite zu dem verwendeten Abstandsmaß und der Projektion \vec{f}_l von \vec{f} auf \mathcal{L} , also auf den Unterraum der affin-linearen Transformationen, der mit der Randbedingung unverträglich ist.

Die Randwertaufgabe mit dem harmonischen und dem biharmonischen Operator lässt sich bei Dirichlet Randbedingung mit der Sinus- und bei Neumann Randbedingung mit der Cosinustransformation lösen. Der Navier-Lamé Operator kann bei periodischer Randbedingung mit der Fouriertransformation gelöst werden.

Die harmonische Randwertaufgabe mit Dirichlet Randbedingung

Sie lautet für die i -te Komponente von \vec{u}

$$\begin{aligned} -\Delta u_i &= -f_i && \text{auf } \Omega \\ u_i &= 0 && \text{auf } \partial\Omega_D. \end{aligned}$$

Die Funktionen $\varphi_{j,k,l}(x, y, z) := \sin\left(\frac{\pi j x}{n_1 + 1}\right) \cdot \sin\left(\frac{\pi k y}{n_2 + 1}\right) \cdot \sin\left(\frac{\pi l z}{n_3 + 1}\right)$ mit $j \in N_1$, $k \in N_2$ und $l \in N_3$ erfüllen die Randbedingung und sind die Eigenfunktionen des Laplace-Operators

$$-\Delta \varphi_{j,k,l}(x, y, z) = \left(\left(\frac{\pi j}{n_1 + 1} \right)^2 + \left(\frac{\pi k}{n_2 + 1} \right)^2 + \left(\frac{\pi l}{n_3 + 1} \right)^2 \right) \cdot \varphi_{j,k,l}(x, y, z) =: \lambda_{j,k,l} \cdot \varphi_{j,k,l}(x, y, z).$$

Die rechte Seite der Randwertaufgabe lässt sich mit der Sinustransformation nach den $\varphi_{j,k,l}$ entwickeln:

$$f_i(x, y, z) = \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \alpha_{i,j,k,l} \cdot \sin\left(\frac{\pi j x}{n_1 + 1}\right) \cdot \sin\left(\frac{\pi k y}{n_2 + 1}\right) \cdot \sin\left(\frac{\pi l z}{n_3 + 1}\right) = \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \alpha_{i,j,k,l} \cdot \varphi_{j,k,l}(x, y, z)$$

wobei $\alpha_{i,j,k,l} := (\text{DST}_{j,k,l}^{3D})^{-1}(f_i)$. Die inverse Sinustransformation wird hier wegen dem Normierungsfaktor gewählt.

Da die $\varphi_{j,k,l}$ eine Basis für den Bildraum der RWA bilden, gilt $u_i = \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \gamma_{i,j,k,l} \cdot \varphi_{j,k,l}$ mit den noch unbekanntenen Koeffizienten $\gamma_{i,j,k,l}$. Aus den bisher betrachteten Einzelteilen ergibt sich

$$\sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \gamma_{i,j,k,l} \cdot \lambda_{j,k,l} \cdot \varphi_{j,k,l} = -\Delta \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \gamma_{i,j,k,l} \cdot \varphi_{j,k,l} = -\Delta u_i = -f_i = -\sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \alpha_{i,j,k,l} \cdot \varphi_{j,k,l}$$

und da die $\varphi_{j,k,l}$ paarweise orthogonal stehen, kann die Gleichung gliedweise gelöst werden. Es gilt insbesondere, dass $\lambda_{j,k,l} > 0$ und daher

$$\gamma_{i,j,k,l} \cdot \lambda_{j,k,l} = -\alpha_{i,j,k,l} \quad \Rightarrow \quad \gamma_{i,j,k,l} = \frac{-\alpha_{i,j,k,l}}{\lambda_{j,k,l}}.$$

Also gilt

$$u_{i,c}(x, y, z) = - \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \frac{\alpha_{i,j,k,l}}{\lambda_{j,k,l}} \cdot \varphi_{j,k,l}(x, y, z) = - \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \frac{(\text{DST}_{j,k,l}^{3D})^{-1}(f_i)}{\lambda_{j,k,l}} \cdot \varphi_{j,k,l}(x, y, z).$$

Die regularisierte Abstiegsrichtung ist in i -ter Raumrichtung $u_i := u_{i,c} + u_{i,l} = u_{i,c} + f_{i,l}$ für $i = 1, \dots, n$.

Die harmonische Randwertaufgabe mit Neumann Randbedingung

Für die i -te Komponente von \vec{u} lautet sie

$$\begin{aligned} -\Delta u_i &= -f_i & \text{auf } \Omega \\ \frac{\partial}{\partial \vec{n}} u_i &= 0 & \text{auf } \partial\Omega_N. \end{aligned}$$

Diesmal erfüllen die Cosinusfunktionen

$$\psi_{j,k,l}(x, y, z) := \cos\left(\frac{\pi(j-1)(x-1)}{n_1-1}\right) \cdot \cos\left(\frac{\pi(k-1)(y-1)}{n_2-1}\right) \cdot \cos\left(\frac{\pi(l-1)(z-1)}{n_3-1}\right) \quad \text{mit } j \in N_1, k \in N_2, l \in N_3$$

die Randbedingung und bilden eine Basis aus Eigenfunktionen mit den Eigenwerten

$$\mu_{j,k,l} := \left(\left(\frac{\pi(j-1)}{n_1-1} \right)^2 + \left(\frac{\pi(k-1)}{n_2-1} \right)^2 + \left(\frac{\pi(l-1)}{n_3-1} \right)^2 \right).$$

Nun ist der Eigenwert $\mu_{1,1,1} = 0$. Daher kann nicht wie im Dirichlet-Fall durch μ dividiert werden. Stattdessen muss die Randwertaufgabe selbst noch einmal regularisiert werden:

$$\begin{aligned} -\Delta u_i + c_\mu \cdot u_i &= -f_i + c_\mu \cdot u_i & \text{auf } \Omega \\ \frac{\partial}{\partial \vec{n}} u_i &= 0 & \text{auf } \partial\Omega_N. \end{aligned}$$

mit $c_\mu > 0$. Die so modifizierte RWA hat die Eigenwerte $\mu_{j,k,l} + c_\mu$. Also lässt sich

$$\hat{\mu}_{j,k,l} := \frac{1}{c_\mu + \mu_{j,k,l}}$$

definieren. Numerisch lässt sich das iterativ lösen

$$-\Delta u_i^{k+1} + c_\mu \cdot u_i^{k+1} = -f_i^k + c_\mu \cdot u_i^k,$$

wobei das u_i^k bereits bekannt ist. Allerdings hat sich in der Praxis herausgestellt, dass die Bildregistrierung am besten funktioniert, wenn der Summand $c_\mu \cdot u_i^k$ auf der rechten Seite weggelassen wird, also

$$-\Delta u_i^{k+1} + c_\mu \cdot u_i^{k+1} = -f_i^k.$$

Der Parameter c_μ steuert in diesem Fall eine Translation der Gewichtungsfunktion im Frequenzraum. Das Weglassen von $c_\mu \cdot u_i^{k+1}$ wird in gewisser Weise durch das CG-Verfahren (siehe 5.4.4 auf Seite 33) wieder ausgeglichen.

Wird c_μ groß gewählt, so ist die Regularisierung schwach. Wird c_μ zu klein gewählt, dann werden die Eigenwerte singulär. Eine sinnvolle Wahl liegt in der Größenordnung

$$c_\mu \approx \frac{1}{(\max\{n_1, n_2, n_3\} - 1)^2}.$$

Der Eigenwert $\mu_{1,1,1}$ gehört zu der konstanten Verschiebung. Sein Gewicht wird auf 1 gesetzt. Damit entspricht dies genau dem Verschiebungs-Teil der affin-linearen Transformation. Also

$$\tilde{\mu}_{j,k,l} := \begin{cases} 1 & \text{falls } (j, k, l) = (1, 1, 1) \\ \hat{\mu}_{j,k,l} & \text{sonst.} \end{cases}$$

Setze jetzt

$$u_{i,c}(x, y, z) := \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \tilde{\mu}_{j,k,l} \cdot (\text{DCT}_{j,k,l}^{3D})^{-1}(f_i) \cdot \psi_{j,k,l}(x, y, z)$$

und $u_i := u_{i,c} + u_{i,l} = u_{i,c} + f_{i,l}$ ist die i -te regularisierte Abstiegsrichtung für $i = 1, \dots, n$.

Die biharmonische Randwertaufgabe mit Dirichlet Randbedingung

Sie hat die Gestalt

$$\begin{aligned}\Delta^2 u_i &= f_i && \text{auf } \Omega \\ \Delta u_i &= 0 && \text{auf } \partial\Omega_D \\ u_i &= 0 && \text{auf } \partial\Omega_D.\end{aligned}$$

Offenbar erfüllen genau die $\varphi_{j,k,l}$ diese RWA und daher kann die Lösung erhalten werden, indem im Frequenzraum zweimal durch die Eigenwerte $\lambda_{j,k,l}$ dividiert wird

$$u_i(x, y, z) = \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \frac{\alpha_{i,j,k,l}}{\lambda_{j,k,l}^2} \cdot \varphi_{j,k,l}(x, y, z) = \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \frac{(\text{DST}_{j,k,l}^{3D})^{-1}(f_i)}{\lambda_{j,k,l}^2} \cdot \varphi_{j,k,l}(x, y, z),$$

bzw. zweimal mit $\tilde{\lambda}_{j,k,l}$ multipliziert wird

$$u_{i,c}(x, y, z) := \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \tilde{\lambda}_{j,k,l}^2 \cdot (\text{DST}_{j,k,l}^{3D})^{-1}(f_i) \cdot \varphi_{j,k,l}(x, y, z).$$

Die biharmonische Randwertaufgabe mit Neumann Randbedingung

Die RWA ist

$$\begin{aligned}\Delta^2 u_i &= f_i && \text{auf } \Omega \\ \frac{\partial}{\partial \vec{n}} \Delta u_i &= 0 && \text{auf } \partial\Omega_N \\ \frac{\partial}{\partial \vec{n}} u_i &= 0 && \text{auf } \partial\Omega_N.\end{aligned}$$

Hier erfüllen die $\psi_{j,k,l}$ wieder die Randbedingungen und die Lösung lautet mit

$$\tilde{\mu}_{j,k,l} := \begin{cases} 1 & \text{falls } (j, k, l) = (1, 1, 1) \\ \frac{1}{c_\mu + \mu_{j,k,l}^2} & \text{sonst} \end{cases}$$

$$u_{i,c}(x, y, z) := \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \tilde{\mu}_{j,k,l} \cdot (\text{DCT}_{j,k,l}^{3D})^{-1}(f_i) \cdot \psi_{j,k,l}(x, y, z).$$

Der Navier-Lamé Operator mit periodischer Randbedingung

Der Navier-Lamé Operator

$$Lu = -\Delta u - \frac{\beta}{2} \cdot \sum_{i=1}^n \sum_{j=1}^n \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} u$$

kann bei periodischer Randbedingung mit Hilfe der Fouriertransformation gelöst werden. Die periodische Randbedingung bedeutet, dass das Bild an allen Seiten nochmal dran gelegt wird. Beim Verlassen des Bildes z.B. nach rechts, wird das Bild auf der linken Seite wieder betreten. Mathematisch bedeutet das für ein zweidimensionales Bild B auf dem Gitter $\Omega = \{1, \dots, n_1\} \times \{1, \dots, n_2\}$, dass $B_{i,j} = B_{i+n_1 \cdot \mathbb{Z}, j+n_2 \cdot \mathbb{Z}}$ gilt.

Durch die Periodizität sind genau die komplexen Exponentialfunktionen die Eigenfunktionen des Navier-Lamé Operators. Im dreidimensionalen Fall ($n = 3$) sind es

$$\omega_{j,k,l}(x, y, z) := \exp\left(-2 \cdot \pi \cdot i \cdot \frac{j-1}{n_1} \cdot x\right) \cdot \exp\left(-2 \cdot \pi \cdot i \cdot \frac{k-1}{n_2} \cdot y\right) \cdot \exp\left(-2 \cdot \pi \cdot i \cdot \frac{l-1}{n_3} \cdot z\right)$$

für $j \in N_1$, $k \in N_2$, $l \in N_3$. Um die Eigenwerte zu bestimmen, wird die Eigenwertgleichung für die Eigenfunktionen $\omega_{j,k,l}(x, y, z)$ durch gerechnet.

$$\begin{aligned}L \omega_{j,k,l}(x, y, z) &= -\Delta \omega_{j,k,l}(x, y, z) - \beta \cdot \left(\frac{\partial^2}{\partial x \partial y} \omega_{j,k,l}(x, y, z) + \frac{\partial^2}{\partial x \partial z} \omega_{j,k,l}(x, y, z) + \frac{\partial^2}{\partial y \partial z} \omega_{j,k,l}(x, y, z) \right) \\ &= -(2\pi i)^2 \cdot \left(\left(\frac{j-1}{n_1} \right)^2 + \left(\frac{k-1}{n_2} \right)^2 + \left(\frac{l-1}{n_3} \right)^2 + \frac{(j-1)(k-1)}{n_1 \cdot n_2} + \frac{(j-1)(l-1)}{n_1 \cdot n_3} + \frac{(k-1)(l-1)}{n_2 \cdot n_3} \right) \omega_{j,k,l}(x, y, z) \\ &= \nu_{j,k,l} \cdot \omega_{j,k,l}(x, y, z)\end{aligned}$$

mit den Eigenwerten

$$\nu_{j,k,l} = 4 \cdot \pi^2 \cdot \left(\left(\frac{j-1}{n_1} \right)^2 + \left(\frac{k-1}{n_2} \right)^2 + \left(\frac{l-1}{n_3} \right)^2 + \frac{(j-1)(k-1)}{n_1 \cdot n_2} + \frac{(j-1)(l-1)}{n_1 \cdot n_3} + \frac{(k-1)(l-1)}{n_2 \cdot n_3} \right).$$

Also gibt es Koeffizienten $\gamma_{i,j,k,l} \in \mathbb{R}$, so dass

$$\sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \gamma_{i,j,k,l} \cdot \nu_{j,k,l} \cdot \omega_{j,k,l} = L \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \gamma_{i,j,k,l} \cdot \omega_{j,k,l} = L u_i = -f_i = - \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \alpha_{i,j,k,l} \cdot \omega_{j,k,l}$$

und die $\alpha_{i,j,k,l}$ lassen sich mit der inversen Fouriertransformation berechnen. Es gilt $\nu_{1,1,1} = 0$ und sonst $\nu_{j,k,l} > 0$. Der Eigenwert $\nu_{1,1,1}$ gehört zur konstanten Verschiebung. Daher kann er in dieser Arbeit auf 1 definiert werden. Weiter ist zu beachten, dass bei Indices zwischen $\frac{n_m}{2}$ und n_m ($m \in \{1, 2, 3\}$) die Frequenz wieder sinkt, da diese Frequenzen modulo 2π den negativen Frequenzen aus dem Intervall $-\frac{n_m}{2} - 1$ bis 0 entsprechen. Da es sich bei der Regularisierung um die Dämpfung der hohen Frequenzen handelt, wird die Gewichtungsfunktion $\tilde{\nu}$ nur über die erste Hälfte der Frequenzindices berechnet und dann gespiegelt fortgesetzt. Die Gewichte sind also

$$\tilde{\nu}_{j,k,l} := \begin{cases} 1 & \text{falls } (j, k, l) = (1, 1, 1) \\ \frac{1}{\nu_{\min\{j, n_1-j+2\}, \min\{k, n_2-k+2\}, \min\{l, n_3-l+2\}}} & \text{sonst.} \end{cases}$$

Da die $\omega_{j,k,l}$ paarweise orthogonal sind kann die Summengleichung gliedweise gelöst werden

$$\gamma_{i,j,k,l} = \tilde{\nu}_{j,k,l} \cdot \alpha_{i,j,k,l} = \tilde{\nu}_{j,k,l} \cdot (\text{FFT}_{j,k,l}^{3D})^{-1}(f_i)$$

und damit ist die regularisierte Suchrichtung $u_{i,c}$ zur Kraft f_i

$$u_{i,c}(x, y, z) = \sum_{l=1}^{n_3} \sum_{k=1}^{n_2} \sum_{j=1}^{n_1} \tilde{\nu}_{j,k,l} \cdot (\text{FFT}_{j,k,l}^{3D})^{-1}(f_i) \cdot \omega_{j,k,l}(x, y, z).$$

Die inverse Fouriertransformation bekommt zwar ausschließlich reelle Werte in der Eingabe f_i , die Frequenzintensitäten sind dagegen komplexwertig. Das ist zwar erstmal die doppelte Datenmenge im Vergleich zu f_i , allerdings liegen Redundanzen vor, so dass die tatsächliche Information nur halb so groß ist. Die Gewichte werden – wie bei komplexen Zahlen üblich – sowohl auf den Realteil, als auch auf den Imaginärteil multipliziert. Die Fouriertransformation nach der Gewichtung ergibt aufgrund der redundanten Struktur bei exakter Rechnung wieder rein reelle Zahlen. Aufgrund von Rechenungenauigkeiten muss der Imaginärteil in der Praxis jedoch explizit auf Null gesetzt werden.

Die Regularisierung mit dieser Randwertaufgabe lässt sich auch mit anderen Potenzen von $\tilde{\nu}$ berechnen. Bei $\tilde{\nu}^2$ entspricht dies der biharmonischen Randwertaufgabe mit periodischer Randbedingung, nur dass zusätzlich die beiden Ableitungsrichtungen für $\beta > 0$ gekoppelt sind. Die Wahl für β ist 1 falls die gemischte Ableitung das gleiche Gewicht bekommen soll wie die reine Ableitung und 2 um eine Gewichtung zu erhalten, die $(j+k+l)^2$ entspricht. Sinnvolle Wahlen liegen in dieser Größenordnung.

Die Rechnung besteht bei allen zuvor beschriebenen Randwertaufgaben darin, die inverse Fourier-, Sinus- bzw. Cosinustransformation zu berechnen, die erhaltenen Frequenzen mit den invertierten Eigenwerten zu gewichten und das dann wieder mit der trigonometrischen Transformation zurück zu transformieren.

Bei der Implementierung des Programms brauchen die inversen Eigenwerte $\tilde{\nu}_{j,k,l}$, $\tilde{\lambda}_{j,k,l}$ bzw. $\tilde{\mu}_{j,k,l}$ im periodischen Fall und harmonischen Fall bzw. die Quadrate davon im biharmonischen Fall nur einmal berechnet zu werden. Der Gesamtaufwand zum Lösen der RWA entspricht also einer Fourier-, Sinus- oder Cosinustransformation vorwärts, rückwärts und $n_1 \cdot n_2 \cdot n_3$ Multiplikationen für jede Komponente $u_{i,c}$. Der Aufwand liegt also im dreidimensionalen Fall in $\mathcal{O}(n_1 \cdot n_2 \cdot n_3 \cdot \log(n_1 \cdot n_2 \cdot n_3))$.

5.4.4 Liniensuche

Es ist eine Abstiegsrichtung $\vec{u} = (u_1, \dots, u_n)^t = \vec{u}_c + \vec{u}_l$ nach dem zuvor beschriebenen Algorithmus gefunden und das Verschiebungsfeld \vec{u} wird die Suchrichtung bei der Liniensuche. Die Randwertaufgabe enthält noch den Parameter $\alpha \geq 0$ mit dem sich steuern lässt, wie stark die Größe des Regularisierungsterms S im Funktional gewichtet wird (vgl. Kapitel 3 auf Seite 17 und 4 auf Seite 19). Dort wurde bereits angesprochen, dass die Länge der Verschiebungsvektoren \vec{u} von der Wahl des Parameters α abhängt und zwar gilt $\frac{1}{\alpha} \propto \|\vec{u}\|$.

Die Schrittweite ρ_0 wird so gewählt, dass $\|\rho_0 \cdot \vec{u}\|_\infty = 1$ gilt. Das Pixel bzw. Voxel mit der stärksten Verschiebung wird auf ein Nachbarpixel bzw. -voxel verschoben. Es gilt also $\alpha = \frac{1}{\rho_0}$. Nach dem Prinzip der Armijo-Liniensuche wird die

Schrittweite solange halbiert, bis sich der Abstand der beiden Bilder nicht mehr verringert. Die Schrittweite zu dem kleinsten Abstand unter allen getesteten wird schließlich genommen. Gegebenenfalls bietet sich noch Interpolation mit quadratischen oder kubischen Polynomen an, deren Minimalstelle als neue Schrittweite ρ_{k+1} dient.

Beschleunigung der Liniensuche mit kubischer Interpolation

Der Abstand zur Schrittweite $\rho = 0 =: q_0$ ist aus der vorangegangenen Iteration bekannt. Die Abstände der Bilder zu $q_3 := \rho_0$, $q_2 := \frac{\rho_0}{2}$ und $q_1 := \frac{\rho_0}{4}$ werden durch Auswerten des Abstandsfunktional

$$p_i := \mathcal{D}_{R,T}(\phi(\vec{u}_{\text{alt}} + q_i \cdot \vec{u}_{\text{such}})), \quad i = 1, 2, 3$$

ausgerechnet. Damit sind die vier Punkte (q_0, p_0) , (q_1, p_1) , (q_2, p_2) und (q_3, p_3) bekannt und Newtons Verfahren der *dividierten Differenzen* ist anwendbar. Das Rechenschema lautet

$$\begin{array}{rcccccc} q_3 & p_3 & & & & & \\ & & & & & & \\ q_2 & p_2 & \frac{p_3 - p_2}{q_3 - q_2} =: p_{23} & & & & \\ & & & & & & \\ q_1 & p_1 & \frac{p_2 - p_1}{q_2 - q_1} =: p_{12} & \frac{p_{23} - p_{12}}{q_3 - q_1} =: p_{123} & & & \\ & & & & & & \\ q_0 & p_0 & \frac{p_1 - p_0}{q_1 - q_0} =: p_{01} & \frac{p_{12} - p_{01}}{q_2 - q_0} =: p_{012} & \frac{p_{123} - p_{012}}{q_3 - q_0} =: p_{0123} & & \end{array}$$

damit ergibt sich das interpolierende kubische Polynom

$$\begin{aligned} \bar{p}(q) &= p_0 + p_{01} \cdot (q - q_0) + p_{012} \cdot (q - q_0) \cdot (q - q_1) + p_{0123} \cdot (q - q_0) \cdot (q - q_1) \cdot (q - q_2) \\ &= p_0 + (q - q_0) \cdot \left(p_{01} + (q - q_1) \cdot (p_{012} + (q - q_2) \cdot p_{0123}) \right). \end{aligned}$$

Gegenüber dem Aufstellen der Bedingungen und Lösen des daraus resultierenden Gleichungssystems hat dieses Verfahren den entscheidenden Vorteil, dass es auch quadratische Funktionen und Geraden ausrechnen kann, falls die Punkte (q_i, p_i) dementsprechend gewählt sind. Das Gleichungssystem würde dabei singular werden. Die einzige Voraussetzung an die dividierten Differenzen ist die Forderung $q_i \neq q_j \forall i \neq j$. Diese ist beim Startwert automatisch erfüllt. Bei der weiteren Iteration wird bei der Verletzung dieser Bedingung abgebrochen, da das Minimum der Suchrichtung dann entweder bereits hinreichend genau approximiert ist, oder das Suchverfahren osziliert. Letzteres wurde in der Praxis bisher nie beobachtet.

Das gesuchte Minimum von $p(q)$ liegt an einer der beiden Nullstellen der Ableitung

$$p'(q) = \underbrace{p_{01} - p_{012} \cdot (q_0 + q_1) + p_{0123} \cdot (q_1 \cdot q_2 + q_0 \cdot q_2 + q_0 \cdot q_1)}_{=:c} + 2 \cdot \underbrace{(p_{012} - p_{0123} \cdot (q_0 + q_1 + q_2))}_{=:b} \cdot q + \underbrace{3 \cdot p_{0123}}_{=:a} \cdot q^2.$$

Die Ableitung ist eine quadratische Gleichung. Die Nullstellen Q_1, Q_2 lassen sich direkt mit der so genannten p-q-Formel ausrechnen.

$$Q_{1,2} = \frac{-b}{2 \cdot a} \pm \sqrt{\left(\frac{b}{2 \cdot a}\right)^2 - \frac{c}{a}}$$

Durch Vergleich der Funktionswerte $p(Q_1)$, $p(Q_2)$ lässt sich feststellen, welche der beiden Nullstellen Q_1 , Q_2 das Minimum von p ist. Dies geht sogar ohne Auswerten des Polynoms, denn ist p_{0123} positiv, so muss $\max\{Q_1, Q_2\}$ die Minimalstelle von p sein. Bei negativem p_{0123} ist es $\min\{Q_1, Q_2\}$. Falls $p_{0123} = 0$ gilt, so ist $p(q)$ eine Parabel. Diese hat entweder ein Minimum, oder ein Maximum, was sich anhand des Vorzeichens von b entscheiden lässt. Liegt ein Maximum vor, so wird $Q := 0$ definiert und andernfalls die Nullstelle der Parabel verwendet.

Die so gefundene Nullstelle Q ist das vorhergesagte Minimum der Liniensuche. Es wird dann

$$p_4 := \mathcal{D}_{R,T}(\phi(\vec{u}_{\text{alt}} + Q \cdot \vec{u}_{\text{such}}))$$

berechnet um den tatsächlichen Abstand zur Schrittweite Q zu erhalten.

Es sind einige Ausnahmefälle zu berücksichtigen.

1. Es kann passieren, dass $Q < 0$ gilt. In diesem Fall wird die Schrittweite auf Null gesetzt.
2. Falls $Q > q_3$ so ist dies ein Extrapolationswert. Dies ist zum einen eine ungenau Vorhersage an den Bildabstand zu dieser Schrittweite und zum anderen unter dem Aspekt der Regularisierung sehr gefährlich. In diesem Fall kann $Q := q_3$ gesetzt werden. Das ist der größte erlaubte Schritt.

Ist nun $q_3 > Q > 0$, dann wird $q_4 := Q$ als neue Schrittweite getestet, um den tatsächlichen Abstand p_4 der Bilder zu ermitteln. Ist dieser echt kleiner als p_0, \dots, p_3 , so wird er in $q_0 < q_1 < q_2 < q_3$ eingereiht. Der am weitesten von q_4 entfernte Punkt wird weggelassen und die Interpolation beginnt vom neuen. Dieser Vorgang wird solange wiederholt, bis das Minimum entlang der Suchrichtung hinreichend genau gefunden ist.

Algorithmus 1 : Minimumsuche entlang der Suchrichtung mit kubischer Interpolation

Gegeben sei : das bisher errechnete Verschiebungsfeld \vec{u}_{alt} , eine Suchrichtung \vec{u}_{such} , eine Startschrittweite ρ_0 und eine Mindestabstiegsforderung $\varepsilon > 0$

```

1   $q_0 := 0$      $q_1 := \frac{\rho_0}{4}$      $q_2 := \frac{\rho_0}{2}$      $q_3 := \frac{\rho_0}{1}$ 
2   $p_0 := \mathcal{D}_{R,T}(\phi(\vec{u}_{\text{alt}}))$ 
3   $p_1 := \mathcal{D}_{R,T}(\phi(\vec{u}_{\text{alt}} + q_1 \cdot \vec{u}_{\text{such}}))$ 
4   $p_2 := \mathcal{D}_{R,T}(\phi(\vec{u}_{\text{alt}} + q_2 \cdot \vec{u}_{\text{such}}))$ 
5   $p_3 := \mathcal{D}_{R,T}(\phi(\vec{u}_{\text{alt}} + q_3 \cdot \vec{u}_{\text{such}}))$ 
   /* Berechnung des kubischen Polynoms mit dem Verfahren der dividierten Differenzen. */
6  for  $i := 0$  step 1 until 3 do
7  |    $t[i] := p[i]$ 
8  |   for  $j := i - 1$  step -1 until 0 do
9  |   |    $t[j] := \frac{t[j+1] - t[j]}{q_i - q_j}$ 
10 |   |    $p[i] := t[0]$ 
11 |   end
12 end
   /* Berechnung des Minimums vom kubischen Polynom. */
13  $a := 3 \cdot p[3]$ 
14  $b := 2 \cdot (p[2] - p[3] \cdot (q_0 + q_1 + q_2))$ 
15  $c := p[1] - p[2] \cdot (q_0 + q_1) + p[3] \cdot (q_1 \cdot q_2 + q_0 \cdot q_2 + q_0 \cdot q_1)$ 
16 if  $a > 0$  then
17 |    $Q := \frac{-b}{2 \cdot a} + \sqrt{\left(\frac{b}{2 \cdot a}\right)^2 - \frac{c}{a}}$ 
18 end
19 if  $a < 0$  then
20 |    $Q := \frac{-b}{2 \cdot a} - \sqrt{\left(\frac{b}{2 \cdot a}\right)^2 - \frac{c}{a}}$ 
21 end
22 if  $a = 0$  then
23 |   if  $b > 0$  then
24 |   |   /* Es handelt sich um eine nach oben geöffnete Parabel. Ihr Minimum wird verwendet. */
25 |   |    $Q := -\frac{c}{b}$ 
26 |   else
27 |   |   if  $b < 0$  then
28 |   |   |   /* Es handelt sich um eine nach unten geöffnete Parabel. Diese hat kein Minimum. */
29 |   |   |    $Q := 0$ 
30 |   |   end
31 |   |   if  $b = 0$  then
32 |   |   |   /* Es handelt sich um eine Gerade */
33 |   |   |   if  $c \geq 0$  then
34 |   |   |   |   /* Die Gerade steigt oder ist konstant. */
35 |   |   |   |    $Q := 0$ 
36 |   |   |   else
37 |   |   |   |   /* Die Gerade fällt. Es wird die maximal zulässige Schrittweite gewählt. */
38 |   |   |   |    $Q := q_3$ 
39 |   |   |   end
40 |   |   end
41 |   end
42 end
43  $q_4 := Q$ 
44  $p_4 := \mathcal{D}_{R,T}(\phi(\vec{u}_{\text{alt}} + q_4 \cdot \vec{u}_{\text{such}}))$ 
45 if  $p_4 + \varepsilon \leq \min\{p_0, p_1, p_2, p_3\}$  then
46 |   Sortiere die Menge  $\{(q_0, p_0), (q_1, p_1), (q_2, p_2), (q_3, p_3), (q_4, p_4)\}$  nach der Größe des ersten Elementes in jedem
47 |   Tupel.
48 |   Entferne das Tupel, zu dem  $q_4$  maximal weit entfernt ist.
49 |   Nummeriere die verbleibenden Tupel der Reihe nach mit 0, 1, 2, 3 durch.
50 |   if  $q_0 < q_1 < q_2 < q_3$  then
51 |   |   /* Das Verfahren der Dividierten Differenzen ist anwendbar. */
52 |   |   Gehe zu Zeile 6.
53 |   end
54 |   /* Sonst gibt es  $i$  und  $j$  mit  $i \neq j$  und  $q_i = q_j$ , d.h. die Nullstelle der Interpolation ist eine bereits getestete
55 |   Schrittweite. */
56 end
57 Die Schrittweite ist das  $q_i$  zum kleinsten  $p_i$  ( $i = 0, \dots, 4$ ). Das neue Verschiebungsfeld ist  $\vec{u}_{\text{alt}} + q_i \cdot \vec{u}_{\text{such}}$ .

```

Das nicht-lineare konjugierte Gradientenverfahren (cg)

Das nicht-lineare konjugierte Gradientenverfahren (cg) ist eine verhältnismäßig einfache Möglichkeit, die Suchrichtung so zu optimieren, dass die Schrittweiten größer ausfallen und somit das gleiche Ziel mit weniger Iterationen erreicht wird. Dabei greift das Verfahren lediglich auf die Suchrichtung der vorangegangenen Iteration zurück.

Beim Verfahren des steilsten Abstiegs kommt es meist vor, dass das Abstiegsverfahren einen Zickzack-Weg (vgl. Abbildung 5.3 (a)) mit vielen kleinen Schrittweiten durchläuft. Die Suchrichtung muss entsprechend oft neu berechnet werden und die Konvergenz des Verfahrens ist langsam. Nun braucht beim Abstiegsverfahren nicht unbedingt genau in die Richtung des negativen Gradienten gegangen zu werden, es kann jede Richtung gewählt werden, die im spitzen Winkel zum negativen Gradienten verläuft. Die Wahl der geschicktesten Suchrichtung ist ebenso schwierig, wie die ursprüngliche Minimierungsaufgabe. Bei quadratischen Funktionen konnte gezeigt werden, dass die Wahl der Suchrichtung s_k zu dem Gradienten g_k mit

$$\begin{aligned} s_k &:= -g_k + \gamma_k \cdot s_{k-1} \\ &= -g_k + \frac{g_k^t \cdot g_k}{g_{k-1}^t g_{k-1}} \cdot s_{k-1} \end{aligned} \quad (\text{Fletcher und Reeves})$$

bei exakter Rechnung in endlich vielen Schritten das exakte Minimum findet. Der vollständige Beweis dazu befindet sich z.B. in [9].

Anschaulich bedeutet das CG-Verfahren, dass dem aktuellen negativen Gradienten ein gewichteter Anteil der zuvor verwendeten Suchrichtung hinzu addiert wird. Das führt bei Zickzack-Wegen, wo zwei aufeinander folgende Schritte nahezu antiparallel verlaufen, dazu, dass eine Suchrichtung resultiert, die im wesentlichen senkrecht zu der aktuellen und der letzten Zickzack-Strecke steht. In Abbildung 5.3 ist dies bei Vergleich der Suchrichtungen von (a) und (b) zu erkennen.

Für den nicht-quadratischen Fall lässt sich in der Literatur nur wenig finden. Nach der anschaulichen Beschreibung ist zu vermuten, dass es einen Vorteil bringt, auch wenn eine Konvergenz nach endlich vielen Schritten nicht zu erwarten ist. Das oben zitierte Beispiel ist bereits nicht-quadratisch und das CG-Verfahren bringt ein deutlich besseres Ergebnis in wenigen Iterationen. Für die Bildanpassung eignet sich die postkonditionierte Variante des CG-Verfahrens von Fletcher und Reeves. Dieses wird im folgenden beschrieben.

Das Postkonditionierte nicht-lineare CG-Verfahren

Die nicht-lineare Funktion $g : \mathbb{R}^n \rightarrow \mathbb{R}$ soll minimiert werden. Dies lässt sich mit der Methode der konjugierten Gradienten für nicht-lineare Funktionen bewerkstelligen.

Um die Konvergenz zu beschleunigen wird ein Prinzip verwendet, dass dem Präkonditionieren bei quadratischen Funktionen entspricht. Im nicht-linearen Fall gibt es zwar keine Matrix, aber es gibt trotzdem einen Weg diese Methode anzuwenden. Anstelle von x und $g(x)$ wird

$$x = M \cdot y$$

verwendet. Was die Matrix M ist, wird später klar. Zunächst wird die Funktion

$$\tilde{g}(y) := g(M \cdot y)$$

definiert. Ihr Gradient ist

$$\nabla \tilde{g}(y) = M^t \cdot \nabla g(x)$$

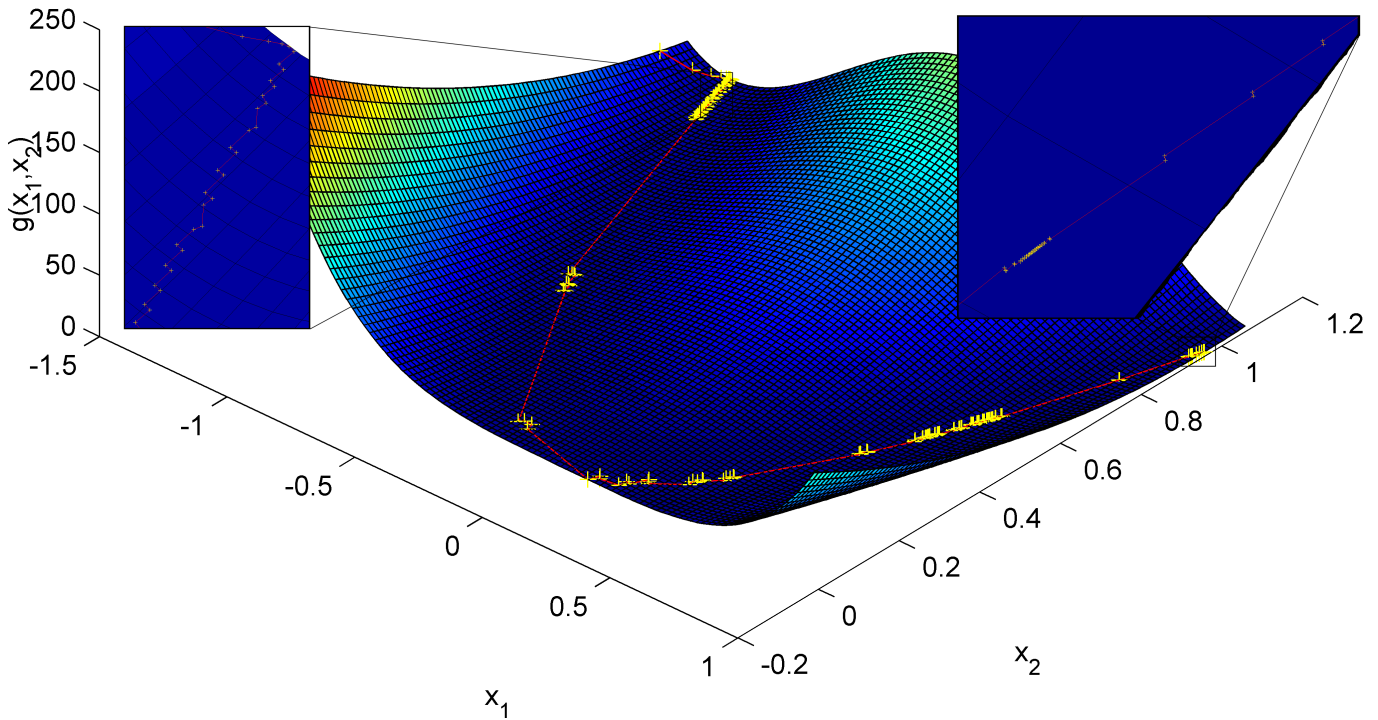
und die zweite Ableitung, die Hessematrix H ist

$$H(\tilde{g}(y)) = M^t \cdot H(g(x)) \cdot M.$$

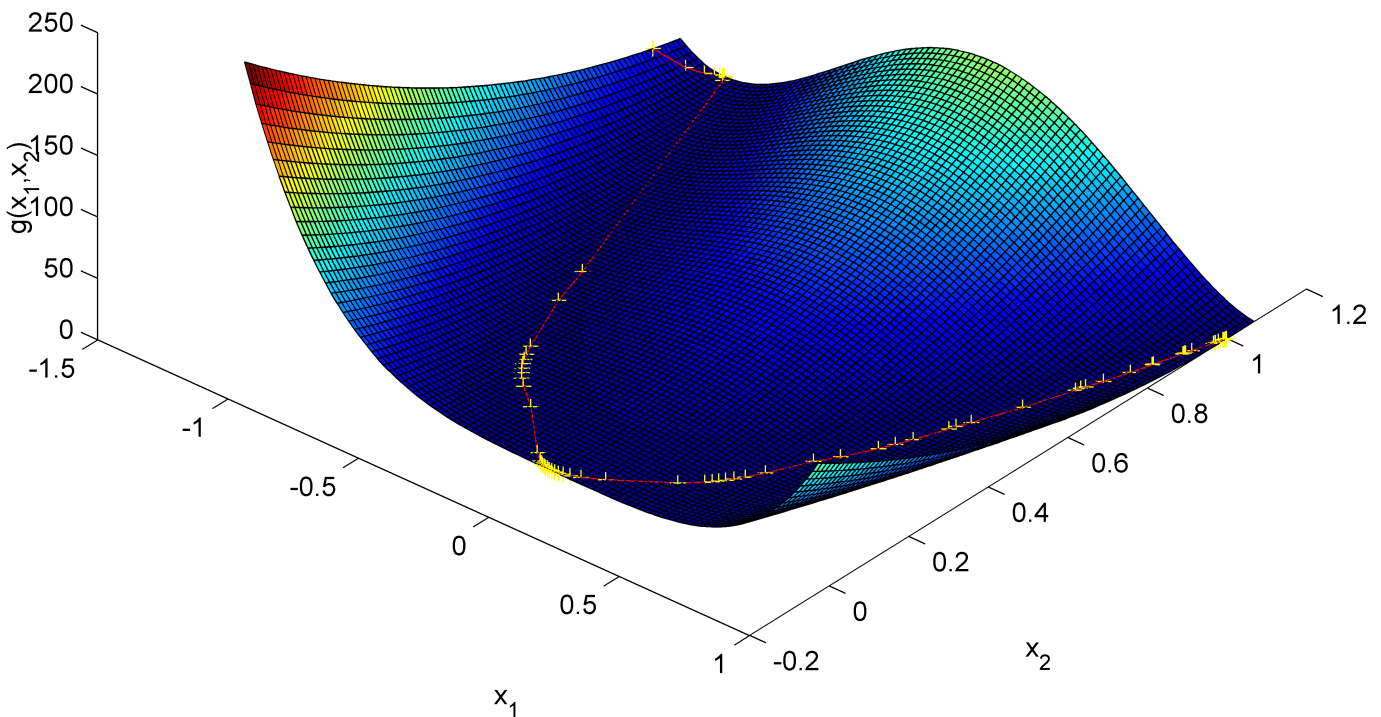
Wird nun das CG-Verfahren nach Fletcher und Reeves auf \tilde{g} angewendet, so ergibt sich:

$$\begin{aligned} C &:= M \cdot M^t \\ f_0 &= -\nabla g(x_0) \\ s_0 &= C \cdot f_0 \\ w_0 &= s_0 \\ \delta_0 &= f_0^t \cdot s_0 = (M^t \cdot f_0)^t \cdot (M^t \cdot f_0) \end{aligned}$$

Für $k = 0, 1, 2, \dots$ wird die Schrittweite $\rho_k > 0$ gesucht, so dass $g(u_k + \rho_k \cdot w_k)$ (annähernd) das Minimum entlang der Suchrichtung w_k ist.



(a) Abstiegsverfahren entlang des negativen Gradienten. In den beiden Vergrößerungen ist ein Zickzack-Weg zu erkennen.



(b) Abstiegsverfahren entlang des konjugierten negativen Gradienten

Abbildung 5.3: Das Minimum der Funktion $g(x_1, x_2) = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2$ wird mit dem Gradientenverfahren (a) und dem Verfahren der konjugierten Gradienten (b) bestimmt. Das exakte Minimum befindet sich in $g(1, 1) = 0$. Bei (a) beträgt die Höhe nach 200 Iterationen $g(0.98348764024707, 0.96719445548042) = 2.729 \cdot 10^{-4}$ und nach 400 Iterationen $g(0.98821389654450, 0.97650878572053) = 1.392 \cdot 10^{-4}$. Mit dem CG-Verfahren (b) ist das Minimum nach bereits 112 Iterationen mit $g(0.99999955217203, 0.99999910135431) = 2.014 \cdot 10^{-13}$ hinreichend genau gefunden. Im Schaubild sind die einzelnen Iterierten durch ein gelbes Kreuz markiert. Es ist zu erkennen, dass die Schrittweiten mit den konjugierten Gradienten deutlich größer ausfallen.

$$\begin{aligned}
x_{k+1} &= x_k + \rho_k \cdot w_k \\
f_{k+1} &= -\nabla g(x_{k+1}) && \text{direkte Abstiegsrichtung} \\
s_{k+1} &= C \cdot f_{k+1} && \text{regularisierte Abstiegsrichtung} \\
\delta_{k+1} &= f_{k+1}^t \cdot s_{k+1} && = (M^t \cdot f_{k+1})^t \cdot (M^t \cdot f_{k+1}) \\
\gamma_k &= \frac{\delta_{k+1}}{\delta_k} \\
w_{k+1} &= s_{k+1} + \gamma_k \cdot w_k
\end{aligned}$$

In diesem Algorithmus wird nur das Produkt der Matrix $C = M \cdot M^t$ verwendet. $H(g)$ bezeichne die Hessematrix von g . Eine obere Schranke für die Konvergenzgeschwindigkeit der Funktion g selbst ist $\sqrt{\kappa(H(g))}$ und für die transformierte Funktion \tilde{g} ist $\sqrt{\kappa(M \cdot H(g) \cdot M^t)}$. Damit ist die Konvergenz optimal beschleunigt, falls $C \approx H(g)^{-1}$ gilt.

Bei der Bildanpassung nimmt die Regularisierung, d.h. das Lösen der Randwertaufgabe mit rechter Seite f_k die Rolle der Matrix C ein. Sei \vec{f}^{k+1} das Kraftfeld und \vec{s}^{k+1} die Lösung der Randwertaufgabe. Dann lautet die Rechnung speziell auf das Minimierungsproblem der Bildanpassung ausgelegt:

$$\begin{aligned}
\vec{w}^0 &:= \vec{s}^0 \\
\text{und für } k &= 1, 2, 3, \dots \\
\gamma_k &:= \frac{\int_{\Omega} \vec{f}^{k+1}(\vec{x})^t \cdot \vec{s}^{k+1}(\vec{x}) \, d\vec{x}}{\int_{\Omega} \vec{f}^k(\vec{x})^t \cdot \vec{w}^k(\vec{x}) \, d\vec{x}} && \text{(CG)} \\
\vec{w}^{k+1} &:= \vec{s}^{k+1} + \gamma \cdot \vec{w}^k
\end{aligned}$$

\vec{w}^{k+1} ist dann die konjugierte Suchrichtung. Die Konvergenzschranken sind hier nicht anwendbar, da $C \neq H^{-1}(\mathcal{D})$. Die Praxis wird zeigen, in wie weit die Konvergenz verbessert wird. Siehe dazu auf Seite 73.

5.5 Veränderung der Auflösung

5.5.1 Halbierung der Auflösung

Hier wird beschrieben, wie sich die Auflösung von dreidimensionalen Bildern halbieren lässt. Die Bildregistrierung kann eine grobe Anpassung auf groben Gittern durchführen. Dabei verringert sich die Datenmenge der Bilder erheblich. Details verschwinden dabei, aber grobe Strukturen sind weiterhin erkennbar. Eine ausführliche Diskussion dazu folgt im Kapitel 6 auf Seite 41.

Bei der Vergrößerung der Auflösung soll die Anzahl der Bildpunkte je Raumrichtung in etwa halbiert werden. Im dreidimensionalen Raum gibt es mehrere Möglichkeiten die umliegenden Punkte zu gewichten. Die symmetrischen Varianten sind die *injection*, das *half weighting* und das *full weighting*. Die Differenzensterne dazu lauten im dreidimensionalen

$$\begin{aligned}
\frac{1}{1} & \left[\begin{bmatrix} 0 & & \\ 0 & 0 & 0 \\ 0 & & \end{bmatrix} \begin{bmatrix} 0 & & \\ 0 & 1 & 0 \\ 0 & & \end{bmatrix} \begin{bmatrix} 0 & & \\ 0 & 0 & 0 \\ 0 & & \end{bmatrix} \right] && \text{(injection)} \\
\frac{1}{12} & \left[\begin{bmatrix} 0 & & \\ 0 & 1 & 0 \\ 0 & & \end{bmatrix} \begin{bmatrix} 1 & & \\ 1 & 6 & 1 \\ 1 & & \end{bmatrix} \begin{bmatrix} 0 & & \\ 0 & 1 & 0 \\ 0 & & \end{bmatrix} \right] && \text{(half weighting)} \\
\frac{1}{64} & \left[\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \\ 4 & 8 & 4 \\ 2 & 4 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right] && \text{(full weighting)}
\end{aligned}$$

Bei der *injection* besteht die Gefahr, dass eine unrepräsentative Auswahl der Punkte verwendet wird. Bei den anderen beiden Gewichtungen werden Mittelwerte der umliegenden Punkte gebildet. In sofern kann das *halfweighting* und das *fullweighting* als Injektion mit vorgeschalteter Glättung verstanden werden.

Soll die Bildauflösung auf ein viertel reduziert werden, so wird die Auflösung halbiert und die halbierte Auflösung nochmal halbiert.

In Abbildung 5.4 und 5.5 auf der nächsten Seite wird das *full weighting* für den eindimensionalen Fall mit verschiedenen Randbedingungen grafisch dargestellt. Der zwei- bzw. dreidimensionale Fall folgt durch sukzessives Anwenden entlang jeder Raumrichtung.

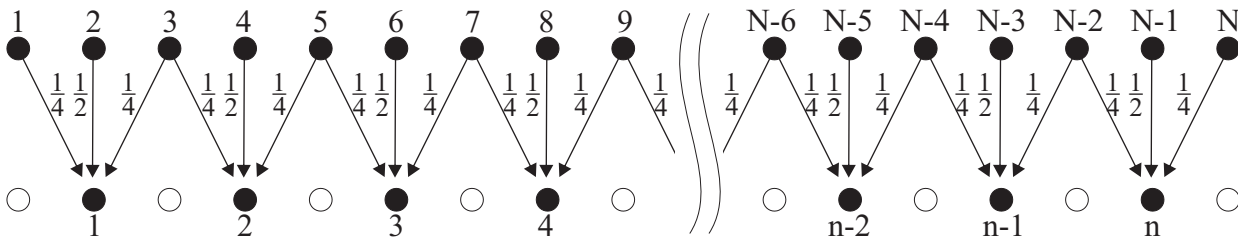


Abbildung 5.4: Halbieren der Auflösung im eindimensionalen, optimiert für die Sinustransformation. Die oberen Punkte repräsentieren das Bild mit der vollen Auflösung $N = 2^k - 1$, $k \in \mathbb{N}$ und die unteren Punkte das Bild mit etwa halbiertes Auflösung $n = 2^{k-1} - 1$. Die Brüche an den Pfeilen geben die Gewichtung der Farbwerte an.

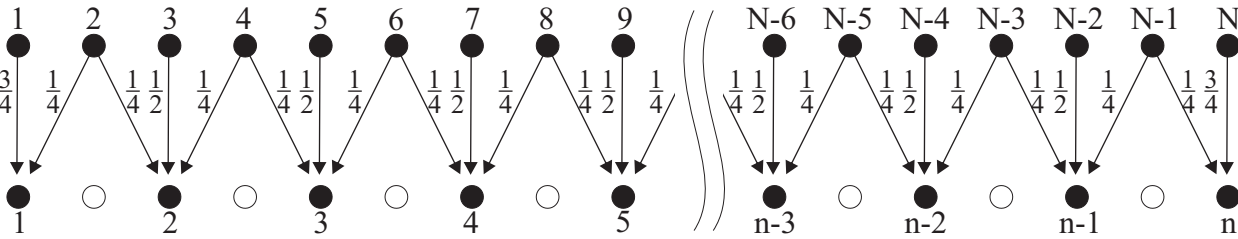


Abbildung 5.5: Halbieren der Auflösung im eindimensionalen, optimiert für die Cosinustransformation. Die oberen Punkte repräsentieren das Bild mit der vollen Auflösung $N = 2^k + 1$, $k \in \mathbb{N}$ und die unteren Punkte das Bild mit etwa halbiertes Auflösung $n = 2^{k-1} + 1$. Die Brüche an den Pfeilen geben die Gewichtung der Farbwerte an.

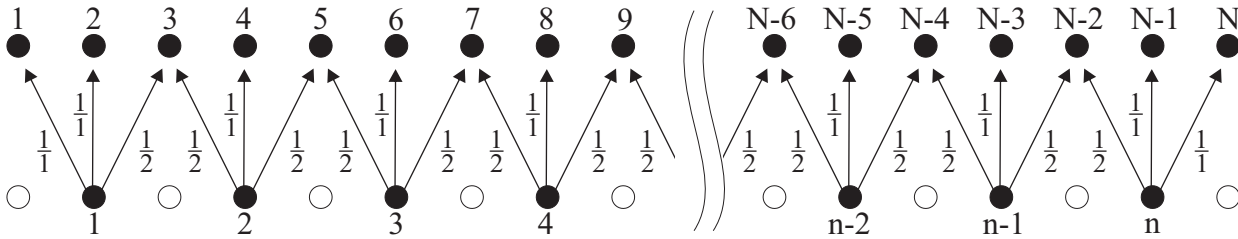


Abbildung 5.6: Interpolation des Verschiebungsfeldes auf die nächst höhere Auflösungsstufe, hier im eindimensionalen dargestellt und für die Sinustransformation optimiert. Die oberen Punkte repräsentieren das Bild mit der doppelten Auflösung $N = 2^{k+1} - 1$, $k \in \mathbb{N}$ und die unteren Punkte das Bild mit der derzeitigen Auflösung $n = 2^k - 1$. Die Brüche an den Pfeilen geben die Gewichtung der Farbwerte an.

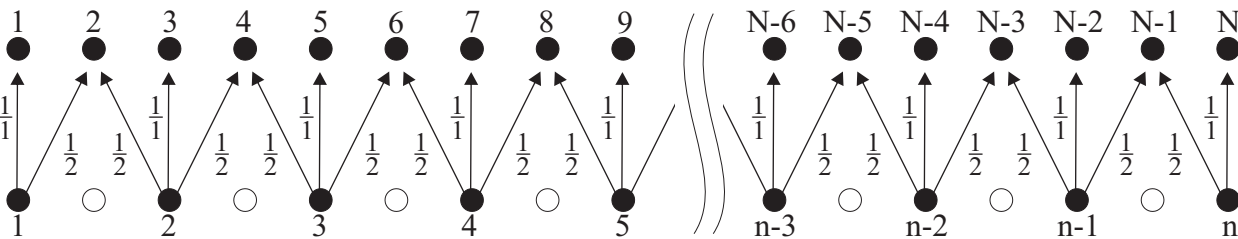


Abbildung 5.7: Interpolation des Verschiebungsfeldes auf die nächst höhere Auflösungsstufe, hier im eindimensionalen dargestellt und für die Cosinustransformation optimiert. Die oberen Punkte repräsentieren das Bild mit der doppelten Auflösung $N = 2^{k+1} + 1$, $k \in \mathbb{N}$ und die unteren Punkte das Bild mit der derzeitigen Auflösung $n = 2^k + 1$. Die Brüche an den Pfeilen geben die Gewichtung der Farbwerte an.

5.5.2 Hochinterpolation der Verformung

Die Bildregistrierung kann die Anpassung zunächst auf groben Gittern vornehmen. Anschließend wird die gefundene Koordinatentransformation $\phi(\vec{u})$ auf das nächst feinere Gitter hoch interpoliert und dort als Startwert für die weitere Iteration verwendet. Dazu muss das Verschiebungsfeld \vec{u} auf das feinere Gitter übertragen werden. Die Komponenten u_i von \vec{u} können als je ein n -dimensionales Bild aufgefasst werden. Es wird also eine Abbildung gebraucht, die die Auflösung von n -dimensionalen Bildern verdoppelt. Dies lässt sich mittels Interpolation lösen. Bei zwei- bzw. dreidimensionalen Bildern ist die bilineare bzw. trilineare Interpolation ein guter Kompromiss zwischen Qualität und Aufwand. Sie passt außerdem am besten zu dem oben beschriebenen Vergrößerungsvorgang. Werden die Indices der Bildmatrix für \vec{x} verwendet, so muss das \vec{u} in $\phi(\vec{u})(\vec{x}) = \vec{x} - \vec{u}(\vec{x})$ nach dem Hochinterpolieren mit dem Faktor 2 multipliziert werden, da die Matrixindices dann die doppelte Laufweite haben.

Die feineren Bilder zu R und T werden sinnvoller Weise aus den Originaldaten heruntergerechnet und bei voller Auflösung sind es die Originaldaten selbst.

5.6 Einbinden von Kontrollpunkten

Ein Kontrollpunkt (oder Landmarke) P ist ein Paar $P = (\vec{a}, \vec{b}) \in \Omega \times \Omega$ von Koordinaten zu gleichen Objekten im Referenzbild R und Templatebild T . Im Allgemeinen gilt $\vec{a} = (a_1, a_2, \dots, a_n)^t \neq (b_1, b_2, \dots, b_n)^t = \vec{b}$. Der Kontrollpunkt stellt die Anforderung $\vec{b} = \phi(\vec{u})(\vec{a}) = \vec{a} - \vec{u}(\vec{a})$ an die Koordinatentransformation $\phi(\vec{u})$. Um die Schreibweise im folgenden übersichtlich zu halten, wird die δ -Funktion mit der Eigenschaft

$$\delta(\vec{x}) := \begin{cases} 1 & \text{falls } \vec{x} = \vec{0} \\ 0 & \text{sonst} \end{cases} \quad \forall \vec{x} = (x_1, \dots, x_n)^t \in \Omega$$

definiert.

Aus der Menge $\mathcal{P} = \{P_1, P_2, \dots, P_r\}$ mit $P_i = (\vec{a}^i, \vec{b}^i) = ((a_1^i, \dots, a_n^i)^t, (b_1^i, \dots, b_n^i)^t)$ von r Kontrollpunkten lässt sich das Kraftfeld $\vec{f} = (f_1, \dots, f_n)^t$ durch

$$f_k(\vec{x}) := \sum_{i=1}^r \delta(\vec{x} - \vec{a}^i) \cdot (a_k^i - b_k^i)$$

explizit ausrechnen. Nun ist es unsinnig einzelne Punkte zu verschieben. Daher muss die Kraft \vec{f} – wie bereits in früheren Abschnitten beschrieben – regularisiert werden. Nun gilt für die regularisierte Kraft \vec{u} i.a.

$$\vec{u}(\vec{a}^i) \neq \vec{a}^i - \vec{b}^i.$$

Es existieren aber bei „sinnvollem“ \mathcal{P} Faktoren $\beta_i \in \mathbb{R}$, so dass mit

$$\tilde{f}_k(\vec{x}) := \sum_{i=1}^r \beta_i^k \cdot \delta(\vec{x} - \vec{a}^i)$$

die regularisierte Kraft \tilde{u} die Bedingung $\tilde{u}(\vec{a}^i) = \vec{a}^i - \vec{b}^i$ erfüllt.

Um die β_i^k zu bestimmen wird das regularisierte Verschiebungsfeld v_i zu den benötigten Basis δ -Peaks

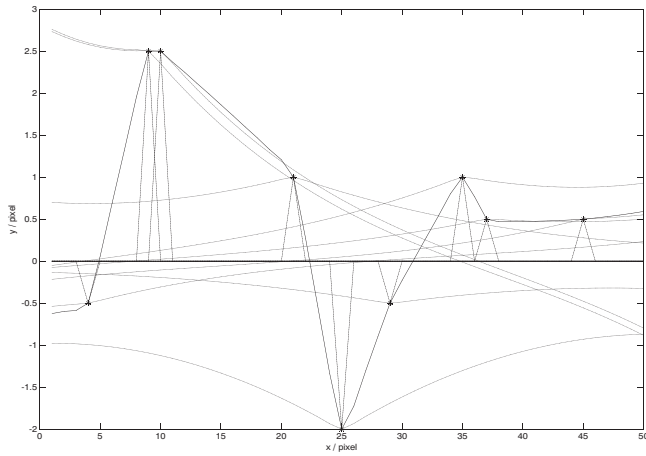
$$f_i(\vec{x}) := \delta(\vec{x} - \vec{a}^i)$$

für $i = 1, \dots, r$ berechnet. Aus

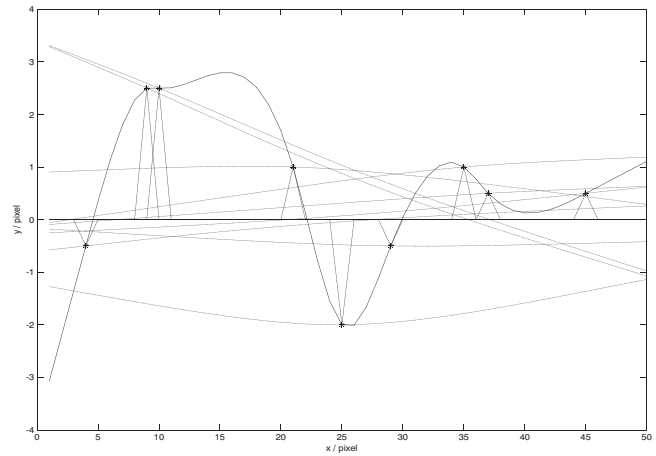
$$u_k(\vec{a}^j) := \sum_{i=1}^r \beta_i^k \cdot v_i(\vec{a}^j) = a_k^j - b_k^j = f_k(\vec{a}^j) \quad \forall j = 1, \dots, r \text{ und } k = 1, \dots, n$$

folgen die β_i^k durch Lösen von n vielen linearen Gleichungssystemen $M \cdot \vec{\beta}^k = \vec{b}^k$ mit Matrix $M_{j,i} := v_i(\vec{a}^j)$ und Vektor $b_i^k := a_k^i - b_k^i$. Hier wird jetzt klar, dass ein Satz an Kontrollpunkten \mathcal{P} aus mathematischer Sicht genau dann „sinnvoll“ ist, wenn die zugehörige Matrix M regulär ist. Aus anschaulicher Sicht müssen dabei die Punkte nicht nur richtig zugeordnet sein, sondern die Zuordnung muss auch mit der Regularisierung kompatibel sein. D.h. falls Kontrollpunkte in einem Bild dicht zusammen liegen, dürfen sie im anderen Bild nur so weit voneinander entfernt sein, wie es die Regularisierung zulässt. Ein stetiges Maß dazu liefert die Konditionszahl $\kappa(M)$.

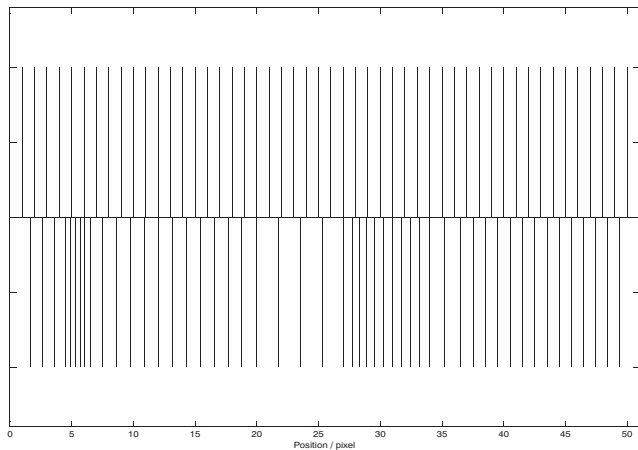
Das aus den Kontrollpunkten $\vec{u} = (u_1, u_2, \dots, u_n)$ berechnete Verschiebungsfeld wird als Startwert für die Bildanpassung verwendet. Die Kontrollpunkte sind dann bereits richtig ausgerichtet. Im weiteren Iterationsverlauf gibt es drei Möglichkeiten mit den Kontrollpunkten zu verfahren:



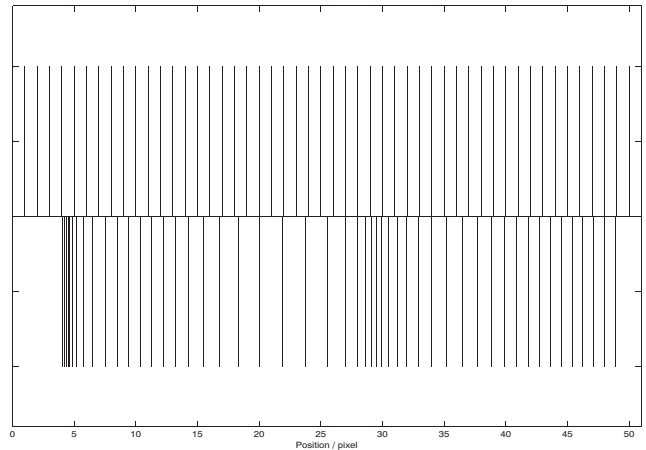
(a) Die Kontrollpunkte (*), ihr jeweiliges Kraftfeld (gestrichelt), resultierendes Verschiebungsfeld (gepunktet) und die Linearkombination der Verschiebungsfelder (durchgezogene Linie) bei harmonischer Regularisierung.



(b) Die Kontrollpunkte (*), ihr jeweiliges Kraftfeld (gestrichelt), resultierendes Verschiebungsfeld (gepunktet) und die Linearkombination der Verschiebungsfelder (durchgezogene Linie) bei biharmonischer Regularisierung.



(c) Lage der Pixel vor (oben) und nach (unten) der Transformation mit harmonischer Regularisierung.



(d) Lage der Pixel vor (oben) und nach (unten) der Transformation mit biharmonischer Regularisierung.

Abbildung 5.8: Ein eindimensionales Beispiel zur Verwendung von Kontrollpunkten mit zwei verschiedenen Regularisierungen. Die Liste der Kontrollpunkte ist $\mathcal{P} := \{(4, 3.5), (9, 11.5), (10, 12.5), (21, 22), (25, 23), (29, 28.5), (35, 36), (37, 37.5), (45, 45.5)\}$. In der Linken Spalte wurde die harmonische und in der rechten Spalte die biharmonische Regularisierung verwendet. Die Lage der Punkte aus \mathcal{P} ist verständlicher Weise gleich, aber die Lage der übrigen Punkte ist echt verschieden. Die Reihenfolge der Punkte ist in beiden Fällen erhalten geblieben, so dass auf eine Nummerierung der Linien in (c) und (d) verzichtet werden kann. Es muss noch gesagt werden, dass die gepunkteten Verschiebungsfelder in (a) und (b) so normiert sind, dass sie durch den zugehörigen Kontrollpunkt (*) verlaufen. Normalerweise sind sie viel dichter an der Nulllinie.

1. Verwendung der Kontrollpunkte ausschließlich zur Berechnung des Startwertes.
2. Aus den Kontrollpunkten wird ein Startwert \vec{u} berechnet und im weiteren Verlauf der Bildanpassung werden die Punkte streng auf ihrem Platz festgehalten („feste Kontrollpunkte“).
3. Die Kräfte \vec{f} werden anhand der Kontrollpunkte modifiziert („weiche Kontrollpunkte“).

Der Fall (1.) ist bereits oben abgehandelt.

Um im Fall von (2.) sicher zu stellen, dass die Kontrollpunkte im Verlauf der Iterationen ihren Platz nicht mehr verlassen, muss im Abstiegsschritt dafür gesorgt werden, dass für die Suchrichtung \vec{w}

$$\vec{w}(\vec{a}^j) = \vec{0} \quad \forall j = 1, \dots, r$$

gilt. Dies beschreibt die Gleichung

$$0 \stackrel{!}{=} w_k(\vec{a}^j) = (w_l(\vec{a}^j))_k + (w_c(\vec{a}^j))_k + \sum_{i=1}^r \beta_i^k \cdot v_i(\vec{a}^j) \quad \forall j = 1, \dots, r \text{ und } k = 1, \dots, n \quad (\text{feste Kontrollpunkte})$$

Es ist wieder das Gleichungssystem $M \cdot \vec{\beta}^k = \vec{b}^k$ diesmal mit $b_j^k := -(w_l(\vec{a}^j))_k - (w_c(\vec{a}^j))_k$ zu lösen.

Es bietet sich hier an, die Matrix M einmal zu invertieren, da ihre Inverse M^{-1} innerhalb einer Auflösungsstufe wiederverwendet werden kann.

Im Fall (3.) werden die Kräfte $\vec{f} = (f_1, \dots, f_n)^t$ wie gewöhnlich berechnet und anschließend in den Kontrollpunkten durch

$$f_k(\vec{a}^j) := \omega \cdot ((a_k^j - b_k^j) - u_k^{\text{alt}}(\vec{a}^j)) \quad \forall j = 1, \dots, r \text{ und } k = 1, \dots, n \quad (\text{weiche Kontrollpunkte})$$

ersetzt, wobei ω ein Gewichtungsmultiplikator und \vec{u}^{alt} das Verschiebungsfeld aus der vorangegangenen Iteration ist.

Auf den ersten Blick ist $\omega = 1$ die richtige Wahl, denn dann wird das nicht-regularisierte Kraftfeld die Kontrollpunkte genau in ihre Position schieben. Dies ist i.a. bereits nach der Regularisierung nicht mehr der Fall. Eine geeignete Wahl ist

$$\omega \approx \frac{\max\{\|\vec{f}(\vec{x})\|_\infty \mid \vec{x} \in \Omega \setminus \mathcal{P}\}}{\max\{\|\vec{f}(\vec{a}^j)\|_\infty \mid j = 1, \dots, r\}}.$$

Diese Wahl von ω sorgt dafür, dass die aus den Kontrollpunkten eingefügten Kräfte in einem ausgewogenen Größenverhältnis zu den übrigen Kräften stehen.

Das erweiterte Funktional

$$J(\vec{u}) := \int_{\Omega} (1 - \chi_{\mathcal{P}}(\vec{x})) \cdot d(R(\vec{x}), T(\vec{x} - \vec{u}(\vec{x})))^2 + \alpha \cdot S(\vec{u}(\vec{x})) \, d\vec{x} + \omega \cdot \sum_{k=1}^n \sum_{j=1}^r (u_k(\vec{a}^j) - (a_k^j - b_k^j))^2$$

mit

$$\chi_{\mathcal{P}}(\vec{x}) := \begin{cases} 1 & \text{falls } \exists P = (\vec{x}, \vec{b}) \in \mathcal{P} \\ 0 & \text{sonst} \end{cases}$$

hat in den Kontrollpunkten genau das oben aufgeführte Kraftfeld der weichen Kontrollpunkte. In den übrigen Punkten ist es mit den Kräften von I identisch.

Zur Erinnerung, der erste Summand beschreibt den Abstand der beiden Bilder, der zweite die Stärke der Verzerrung und der neu hinzu gekommene dritte Term misst wie gut die Kontrollpunkte ausgerichtet sind. Durch die Parameter $\alpha \geq 0$ und $\omega \geq 0$ lassen sich die einzelnen Aspekte unterschiedlich stark gewichten. Im ersten Summand wird wegen $1 - \chi_{\mathcal{P}}(\vec{x})$ der Abstand genau der Punkte nicht gemessen, zu denen Kontrollpunkte gegeben sind. Zu

$$\begin{aligned} \tilde{J}(\vec{u}) &:= \int_{\Omega} d(R(\vec{x}), T(\vec{u}(\vec{x})))^2 + \alpha \cdot S(\vec{u}(\vec{x})) \, d\vec{x} + \omega \cdot \sum_{k=1}^n \sum_{j=1}^r (u_k(\vec{a}^j) - (a_k^j - b_k^j))^2 \\ &= I(\vec{u}) + \omega \cdot \sum_{k=1}^n \sum_{j=1}^r (u_k(\vec{a}^j) - (a_k^j - b_k^j))^2 \end{aligned}$$

lautet das Kraftfeld in den Kontrollpunkten dagegen

$$\tilde{f}_k(\vec{a}^j) := \hat{f}_k(\vec{a}^j) + \omega \cdot ((a_k^j - b_k^j) - u_k^{\text{alt}}(\vec{a}^j)) \quad \forall j = 1, \dots, r \text{ und } k = 1, \dots, n$$

wobei \hat{f}_k das Kraftfeld zum Funktional I ist. Im Kraftfeld ist der Vorzug von J gegenüber \tilde{J} sichtbar, denn $\hat{f}_k(\vec{a}^j)$ sollte zwar mit der Richtung aus dem Kontrollpunkt übereinstimmen, allerdings gibt es keine Garantie dafür – vor allem auch unter dem Gesichtspunkt, dass Kontrollpunkte genau dann nötig sind, wenn die Bilder so stark abweichen, dass sich mit dem Abstandsmaß d alleine keine befriedigende Bildanpassung erzielen lässt.

6 Der Algorithmus

Der Algorithmus zur multimodalen Bildregistrierung ist insgesamt recht lang. Im vorangegangenen Kapitel wurden bereits die Details der Unterprogramme geklärt. Hier wird zuerst eine sehr grobe Formulierung angegeben, diese wird dann besprochen und anschließend werden weitere Details genannt.

6.1 Die Iteration über die Auflösungsstufen

Algorithmus 2 : Die Iteration über die Auflösungsstufen

Gegeben sei : ein Startwert für die Koordinatentransformation, z.B. $\vec{u} = 0$ oder ein aus Kontrollpunkten errechnetes Verschiebungsfeld.

/ Der Auflösungsparameter r gibt an, wie oft die Auflösung des Originalbildes R bzw. T halbiert ist. */*

```

1 foreach Auflösung  $r$  von der gröbsten bis zur feinsten do */
  /* vergrößere  $R$  und  $T$  auf die aktuelle Auflösung. */
2    $R_r := R$ 
3    $T_r := T$ 
4   for  $i := 1$  step 1 until  $i = r$  do
5      $R_r :=$  Auflösungshalbierung( $R_r$ ) /* mit dem fullweighting wie in 5.5.1 auf Seite 35 beschrieben. */
6      $T_r :=$  Auflösungshalbierung( $T_r$ )
7   end
8   führe die Bildregistrierung auf dieser Auflösung durch. /* wird später beschrieben */
  /* Interpoliere das Verschiebungsfeld  $\vec{u}$  auf das nächst feinere Gitter (vgl. 5.5.2 auf Seite 37), damit es dort als Startwert verwendet werden kann: */
9   for  $i := 1$  step 1 step  $n$  do
10     $(\vec{u})_i :=$  Hochinterpolation( $(\vec{u})_i$ )
11  end
12 end

```

Es gibt zwei wesentliche Gründe dafür, dass mit niedriger Auflösung begonnen wird. Der erste ist von praktischer Natur. Bei groben Auflösungen ist die Datenmenge viel geringer. Damit sinkt der Rechenaufwand erheblich und die einzelne Iteration wird deutlich schneller. Jede Halbierung der Auflösung verringert die Datenmenge um den Faktor 2^n wobei n die Dimension der Bilder ist. Dazu kommt noch die Tatsache, dass eine Verschiebung von 1 Pixel in einem Bild der Auflösung $\frac{1}{8}$ in der vollen Auflösung eine Verschiebung von etwa 8 Pixel ergibt. Damit können grobe Strukturen auf der niedrigen Auflösung schneller über weite Strecken verschoben werden. Das ist analog zu Mehrgitterverfahren, bei denen die niedrigen Frequenzen auf groben Gittern geglättet werden, weil sie dort relativ gesehen höher frequent sind.

Der zweite Grund liegt in der Tatsache, dass zuerst grobe Anpassungen vorgenommen werden müssen, damit nicht lokal Strukturen aufeinander ausgerichtet werden, die global gesehen nicht zusammen gehören. Sind die groben Strukturen ausgerichtet, dann können Details korrigiert werden. Bei den Gehirnaufnahmen z.B. muss zuerst das gesamte Hirn zurecht gedreht werden. Ist dies geschehen, so können die Falten auf einander ausgerichtet werden. Wird der Registrierungsvorgang dagegen auf feiner Auflösung begonnen, so kann es passieren, dass Falte a auf Falte b registriert wird, weil sie sich ähnlich sehen und zufälliger Weise dicht bei einander liegen.

Ein weiterer Grund, der die Registrierung zunächst auf groben Auflösungen motiviert liegt in der Beschaffenheit des Abstandsmaßes. Dieses vergleicht den Verlauf der Kanten bzw. Höhenlinien von R und T punktweise. Das verlangt, dass sich die Höhenlinien der beiden Bilder in vielen Punkten berühren. Die Gradienten werden numerisch mit dem Differenzenverfahren 2. Ordnung berechnet. Dieses nimmt die Hälfte der Grauwertdifferenz zwischen dem vorigen und dem nächsten Bildpunkt. Daraus folgt, dass sich die Gradienten ∇R und ∇T nicht mehr überlappen, sobald die Höhenlinien mehr als ein Pixel Abstand haben. Aus diesem Grund werden die Kanten durch Glättung künstlich verbreitert. Bei schlecht ausgerichteten Bildern müssen die Kanten sehr breit werden, was einer starken Glättung entspricht. Bei gut ausgerichteten Bildern reichen dünnere Kanten, also geringe Glättung. Sind die Kanten stark verbreitert, so ist die Höhe in benachbarten Bildpunkten annähernd identisch. Daher genügt es das Bild auf jeden zweiten oder vierten Bildpunkt zu reduzieren. Der Vorgang des Glättens und Auswählens einzelner Punkt ist genau im

half- bzw. noch ausgeprägter im *fullweighting* enthalten.

Ein Beispiel dazu: R enthalte eine gerade Linie und T eine sinusförmige Linie, die etwa symmetrisch um die Linie in R osziliert. Auf feiner Auflösung schneiden sich die beiden Linien nur in einzelnen Punkten. Auf gröberer Auflösung kann die Oszillation nicht mehr vollständig wiedergegeben werden. Demzufolge liegen die groben Pixel dichter beieinander und die Kanten beginnen sich in verhältnismäßig mehr Punkten zu überlappen. Ein Beispiel mit zwei sich kreuzenden Geraden beginnt auf Seite 48.

6.2 Die Iterationen über die Glättungsstufen

Algorithmus 3 : Die Iteration über die Glättungsstufen

Gegeben sei : eine Höchstglättung t_a , eine Mindestglättung $0 \leq t_e \leq t_a$ und ein Glättungsreduzierungsfaktor $0 < t_r < 1$. Weiter seien die Bilder R_r und T_r und das bisher errechnete Verschiebungsfeld \vec{u} in gleicher Auflösung gegeben.

```

1 while  $t \geq t_e$  do
    /* glätte die Bilder  $R_r$  und  $T_r$ , indem z.B. die Wärmeleitungsgleichung zu beiden Bildern im Zeitpunkt  $t$ 
       ausgewertet wird. */
2    $R_{r,t} := \text{Glättung}(R_r, t)$ 
3    $T_{r,t} := \text{Glättung}(T_r, t)$ 
4   führe die Bildregistrierung mit den geglätteten Bildern  $T_{r,t}, R_{r,t}$  durch.
5   setze  $t := t \cdot t_r$  /* Die Glättung wird reduziert. */
6 end

```

Eine gewisse Mindestglättung ist immer erforderlich, damit die Höhenlinien der beiden Bilder aufeinander wirken können. Wie zuvor bereits erwähnt vergleicht das Abstandsmaß die Bilder punktweise. Daher ist es notwendig, dass sich die Kanten der beiden Bilder überschneiden. Durch den Glättungsvorgang werden die Kanten künstlich verbreitert. Das Beispiel auf Seite 49 veranschaulicht diesen Prozess.

Die stärkere Glättung zu Anfang soll den Wechsel der Auflösung und die dadurch verloren gegangene Glättung ausgleichen. Zum einen muss die lineare Hochinterpolation des Verschiebungsfeldes \vec{u} nachkorrigiert werden und zum anderen können feinere Kanten sichtbar werden, die auf der gröberen Auflösung unsichtbar gewesen sind. Die Glättung wird dann nach und nach zurückgefahren, so dass die Höhenlinien immer genauer an ihren Zielplatz lokalisiert werden. Die schwächste Glättung ist immer noch größer Null, damit bereits gut passende Kanten stabilisiert bleiben. Mit Glättung $t = 0$ kann es passieren, dass die Kanten nebeneinander rutschen und dadurch vom Maß nicht mehr in Beziehung gesetzt werden. Ein sehr großes Problem sind krümmungsfreie Kanten. Liegen diese leicht nebeneinander, so kann Maß d diesen Fehler nur anhand der durch Glättung auftretenden halbkreisförmigen Enden bemerken (vgl. 6.6 auf Seite 54).

Zu Beachten ist noch, dass Bilder mit reduzierter Auflösung bereits eine gewisse Glättung beinhalten. Daher müssen t_a und t_e bei niedrigeren Auflösungen kleiner gewählt werden als auf hohen Auflösungen. Es hat sich in der Praxis bewährt, zwei Auflösungsunabhängige Werte t_a^{init} und t_e^{init} zu wählen und

$$t_a := t_a^{\text{init}} \cdot \min\{n_{r,1}, \dots, n_{r,n}\}$$

$$t_e := t_e^{\text{init}} \cdot \min\{n_{r,1}, \dots, n_{r,n}\}$$

zu setzen. Dies entspricht einer Verdoppelung beim Wechsel auf das nächst feinere Gitter. $n_{r,i}$ ist auch hier die Anzahl der Pixel entlang der i -ten Bildraumrichtung auf Auflösungsstufe r . Das Minimum der Dimensionslängen zu wählen ist hier von Vorteil, da die Glättungsintensität dann auf die Raumrichtung mit der geringsten Detailtiefe angepasst ist.

Im Abschnitt der Unterprogramme wurden zwei Möglichkeiten zum Glätten der Bilder vorgestellt:

1. die Wärmeleitungsgleichung
2. das Half- bzw. Fullweighting.

Letzteres ist die diskrete Gaußglättung mit kleiner Standardabweichung in der Gaußglockenkurve. Für stärkere Glättung, was einer größeren Standardabweichung entspricht, kann das Half- bzw. Fullweighting mehrmals hintereinander angewendet werden. Damit ergibt sich auch schon der Rechenaufwand als $\mathcal{O}(g \cdot n_1 \cdot \dots \cdot n_n)$ wobei $g \in \mathbb{N}$

angibt, wie oft das Half- bzw. Fullweighting angewendet wird. Für kleine g ist der Aufwand damit linear, bei größerem g quadratisch zur Bildgröße.

Bei der Wärmeleitungsgleichung liegt der Rechenaufwand unabhängig vom Glättungsparameter t – der in dem physikalischen Modell der Zeit entspricht – wegen der schnellen Fouriertransformation in $\mathcal{O}(n_1 \cdot \dots \cdot n_n \cdot \log(n_1 \cdot \dots \cdot n_n))$. D.h. Bei geringer Bildglättung ist das Half- bzw. Fullweighting effizienter, bei starker Glättung dagegen die Wärmeleitungsgleichung.

Nun ist die Fouriertransformierte einer Gaußkurve im Frequenzraum wieder eine Gaußkurve mit reziproker Standardabweichung $\sigma > 0$. Die gaußsche Glättung ist mathematisch die Faltung des Bildes B mit der Gaußfunktion. Bis auf Normierung gilt also

$$\begin{aligned} B * \exp\left(-\frac{(\cdot)^2}{2 \cdot \sigma^2}\right) &= \int_{\Omega} B(\cdot - \vec{y}) \cdot \exp\left(-\frac{\vec{y}^t \cdot \vec{y}}{2 \cdot \sigma^2}\right) d\vec{y} \\ &= \mathcal{F}^{-1}\left(\mathcal{F}(B) \cdot \mathcal{F}\left(\exp\left(-\frac{(\cdot)^2}{2 \cdot \sigma^2}\right)\right)\right) \\ &= \mathcal{F}^{-1}\left(\mathcal{F}(B) \cdot \exp\left(-\frac{(\cdot)^2 \cdot \sigma^2}{2}\right)\right) \end{aligned}$$

wobei \mathcal{F} die Fouriertransformation bezeichnet.

In sofern unterscheidet sich die Gaußglättung zur Wärmeleitungsgleichung nur durch die Randbedingung. Während bei der Wärmeleitungsgleichung durch Dirichlet oder Neumann Randbedingung die trigonometrische Transformation auf Sinus- bzw. Cosinusfunktionen eingeschränkt ist, so ist es bei der Gaußglättung die volle Fouriertransformation. Das ist auch der Grund, weshalb bei Gaußglättung die Gewichtungsfunktion im Frequenzraum aus der vollen Glockenkurve besteht und bei der Wärmeleitungsgleichung nur die rechte Hälfte vorkommt.

Die praktische Auswirkung der Glättungswahl wird im nächsten Kapitel besprochen.

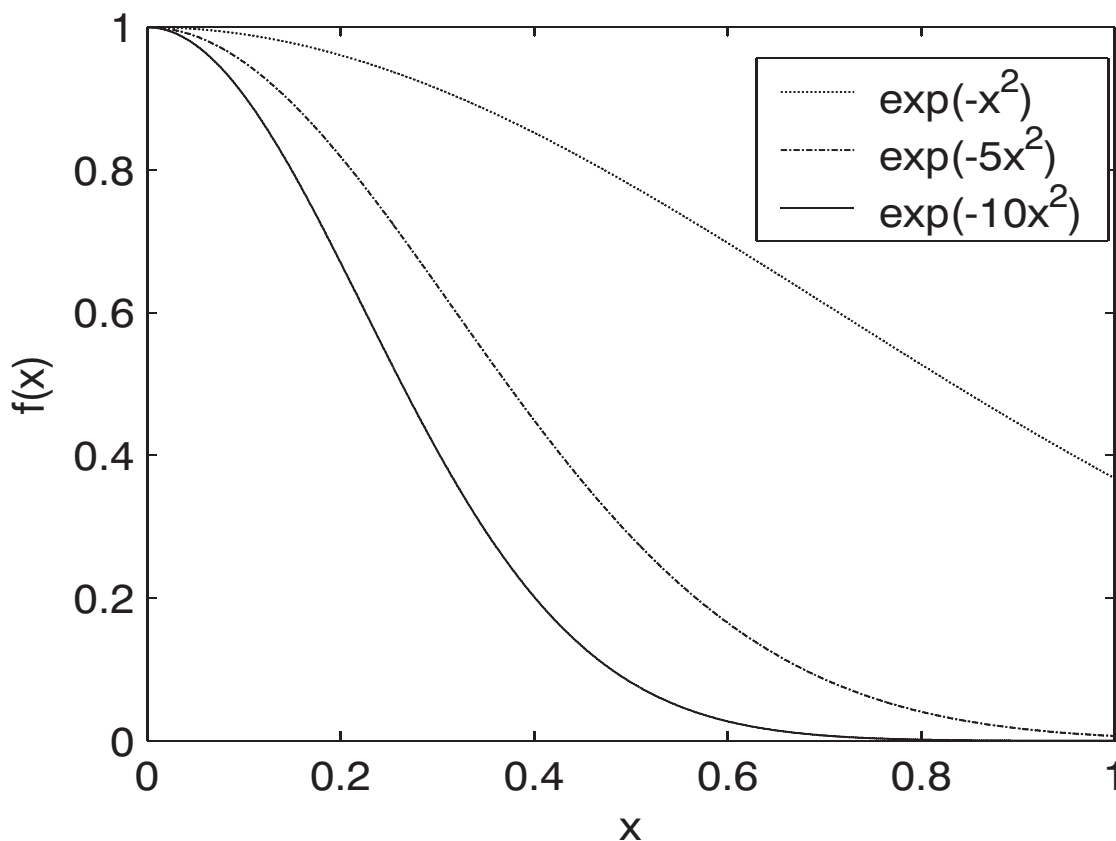


Abbildung 6.1: Der Verlauf der Gewichtungsfunktionen im Frequenzraum für drei verschieden starke Glättung mit der Wärmeleitungsgleichung. Bei der Gaußglättung wird die Funktion nach links symmetrisch fortgesetzt. x ist dabei die relative Frequenzhöhe und $f(x)$ die Gewichtung. Je schneller die Gewichtungsfunktionen gegen Null gehen, um so verschwommener ist das resultierende Bild, da die hohen Frequenzanteile stark abgedämpft sind.

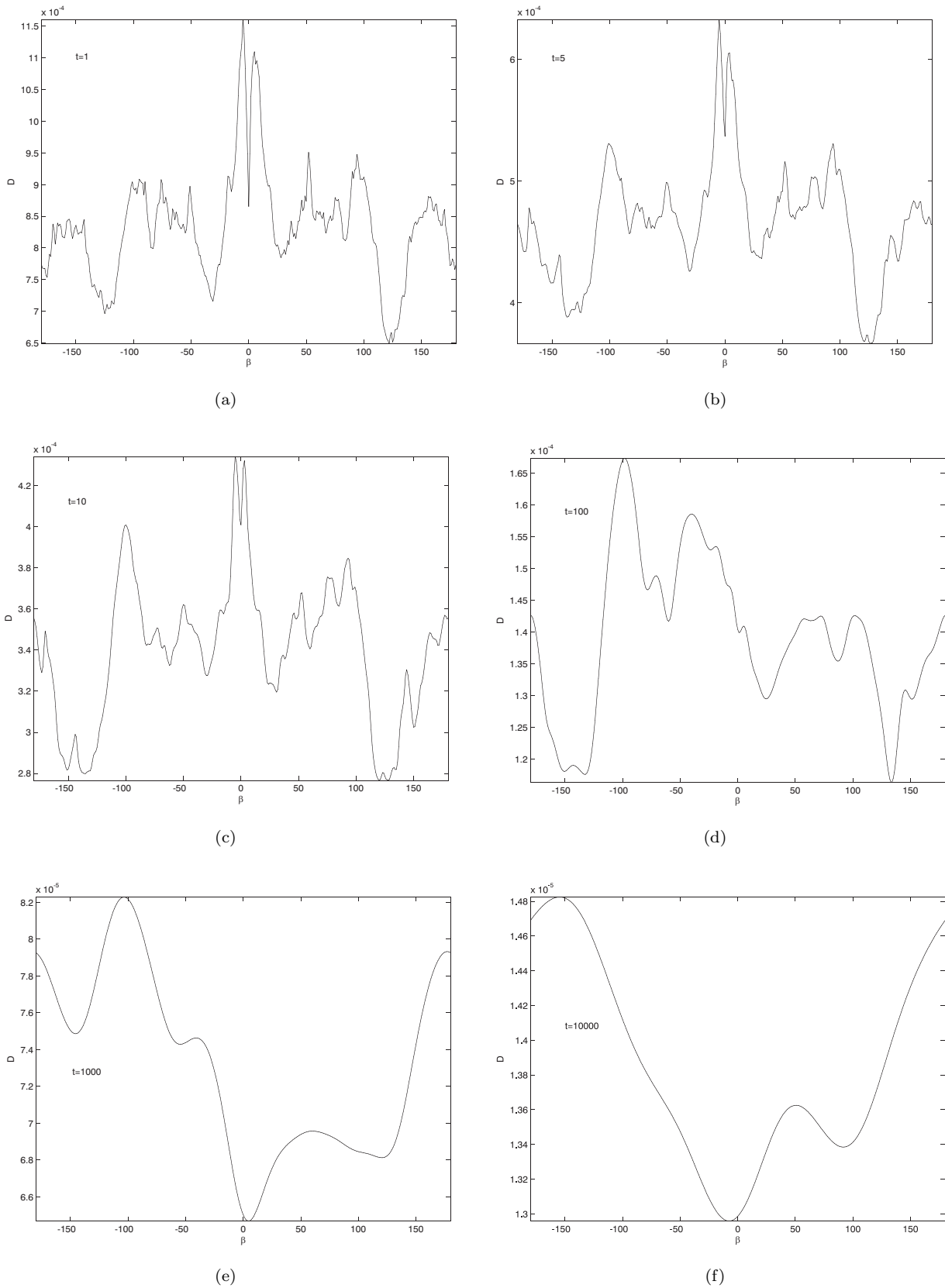


Abbildung 6.2: Der Abstand der Bilder $\mathcal{D}_{R,T}(\phi(\beta)) = \sqrt{\frac{\int_{\Omega} d(R, T \circ \phi(\beta); \vec{x})^2 d\vec{x}}{n_1 \cdot n_2}}$ bei Drehung um den Mittelpunkt mit Drehwinkel β und verschieden starker Glättung t in der Wärmeleitungsgleichung. R ist das Bild in Abb. 7.1 auf Seite 58 (a) und T ist die unverzerrte Version von (b), also das Ergebnis der idealen Transformation. Die Bilder sind mit Drehwinkel $\beta = 0$ deckungsgleich. Bei geringer Glättung existieren viele lokale Minima. Bei starker Glättung existieren nur noch einzelne lokale Minima. Allerdings ist das globale Minimum etwas verschoben. Der Bildanpassungsalgorithmus fängt mit (f) an. Dort wird das globale Minimum leicht gefunden. Dann wird die Glättung etwas reduziert. Durch die Nähe zum globalen Minimum wird bei vorsichtiger Zurücknahme der Glättung das gewünschte Minimum $\beta = 0$ in (a) gefunden werden.

6.3 Die innere Iteration der Bildanpassung

Algorithmus 4 : Die innere Iteration der Bildanpassung

Gegeben seien : zwei Bilder $R_{r,t}$, $T_{r,t}$ und ein Verschiebungsfeld \vec{u}^0 mit gleicher Auflösung, sowie eine Mindestabstiegsforderung $\varepsilon > 0$.

```

1  $k := 0$ 
2  $\vec{f}^k := \text{forces}_d\left(R_{r,t}, (T \circ \phi(\vec{u}^k))_{r,t}\right)$  /* die Kräfte  $\vec{f}_\Omega$  der aktuellen Bilder mit Maß  $d$  oder  $\tilde{d}$  */
3 if weiche Kontrollpunkte sind gegeben then
4 |   modifiziere  $\vec{f}^k$  wie in 5.6 auf Seite 39 beschrieben.
5 end
6 for  $i = 1$  step 1 until  $n$  do
7 |    $(\vec{f}_i^k)_i := P_{\mathcal{L}}((\vec{f}^k)_i)$  /* die Projektion von  $\vec{f}^k$  auf den Raum der affin-linearen Funktionen  $\mathcal{L}$  wie es in 5.4.1
   |   auf Seite 25 beschrieben ist. */
8 end
9  $\vec{f}_c^k := \vec{f}^k - \vec{f}_i^k$  /* das orthogonale Komplement zur Projektion auf  $\mathcal{L}$ . */
10  $\vec{u}_i^k := \vec{f}_i^k$  /* Bei dem linearen Anteil ist keine Regularisierung notwendig. */
11 for  $i = 1$  step 1 until  $n$  do
12 |    $(\vec{u}_c^k)_i := \text{rwa}((\vec{f}_c^k)_i)$  /* Der Anteil der Kräfte aus dem Orthogonalraum von  $\mathcal{L}$  wird durch Lösen der
   |   Randwertaufgabe regularisiert. Die genaue Rechnung steht in 5.4.2 auf Seite 26. */
13 end
14  $\vec{w}^k := \vec{u}_i^k + \vec{u}_c^k$  /* die direkte Suchrichtung */
15 if feste Kontrollpunkte sind gegeben then
16 |   modifiziere  $\vec{w}^k$  wie in 5.6 auf Seite 39 beschrieben.
17 end
18 if  $k \geq 1$  then
   |   /* Verfahren der konjugierten Gradienten: */
19    $\gamma_k := \frac{\int_{\Omega} \vec{f}^k(\vec{x})^t \cdot \vec{w}^k(\vec{x}) d\vec{x}}{\int_{\Omega} \vec{f}^{k-1}(\vec{x})^t \cdot \vec{v}^{k-1}(\vec{x}) d\vec{x}}$ 
20    $\vec{v}^k := \vec{w}^k + \gamma_k \cdot \vec{v}^{k-1}$  /* die konjugierte Suchrichtung. Das rechts verwendete  $\vec{v}^{k-1}$  ist die Suchrichtung aus
   |   der vorangegangenen Iteration. */
21 else
22 |    $\vec{v}^k := \vec{w}^k$  /* im ersten Durchlauf ( $k = 0$ ) */
23 end
24  $\rho_0 := \frac{1}{\max_{\vec{x} \in \Omega} \|\vec{v}(\vec{x})\|_{\infty}}$  /* Bei Schrittweite  $\rho_0$  beträgt die Verschiebung an der stärksten Stelle genau ein Pixel. */
25  $\rho := \text{argmin}_{0 \leq \rho \leq \rho_0} \mathcal{D}_{R_{r,t}, T_{r,t}}(\phi(\vec{u}^k + \rho \cdot \vec{v}^k))$  /* Bestimmung der Schrittweite in der Liniensuche. Praktisch kann
   |   das argmin mit der kubischen Interpolation wie in 5.4.4 auf Seite 32 gefunden werden. */
26 if  $\rho > 0$  AND  $\mathcal{D}_{R_{r,t}, T_{r,t}}(\phi(\vec{u}^k + \rho \cdot \vec{v}^k)) + \varepsilon \leq \mathcal{D}_{R_{r,t}, T_{r,t}}(\phi(\vec{u}^k))$  then
27 |    $\vec{u}^{k+1} := \vec{u}^k + \rho \cdot \vec{v}^k$ 
28 |    $k := k + 1$ 
29 |   Gehe zu Zeile 2.
30 else
31 |   Die Iteration ist auf der Glättungsstufe  $t$  zu Ende.
32 end

```

Die Schreibweise $(T \circ \phi(\vec{u}^k))_{r,t}$ deutet bereits an, dass es notwendig ist, das ungeglättete Bild zu verformen und anschließend zu glätten, anstatt einfach das geglättete Bild zu verformen was sich so schreiben lässt $(T_{r,t} \circ \phi(\vec{u}^k))$. Folgende Überlegung erklärt den Grund dafür. Nehme an R und T seien bis auf eine Streckung in T identisch. Dann

wird der Registrierungsalgorithmus eine Stauchung einleiten. Wird die Stauchung auf das geglättete Bild angewendet, so ist die Glättung im transformierten Bild T_t^ϕ entlang der Stauchungsrichtung deutlich geringer als es in R_t der Fall ist. Ungeglättet sind die Bilder dann zwar identisch, in der geglätteten Version sind die Kanten dagegen unterschiedlich breit. Damit ist eine exakte Registrierung i.a. nicht möglich.

Der lineare Anteil \vec{u}_l braucht nicht regulärisiert zu werden, da lineare Transformationen das Bild immer global verändern. Bzw. anderes herum betrachtet ist die Projektion auf den Raum der linearen Transformationen bereits regulärisiert, denn die konkrete Rechnung beinhaltet eine (gewichtete) Mittelwertbildung der Kräfte \vec{f} (vgl. 5.4.1 auf Seite 25).

Die konjugierten Gradienten werden bei jedem Wechsel der Glättung bzw. Auflösung wieder vom neuen aufgebaut, da jede Wahl der Parameter r und t streng genommen ein eigenes Minimierungsproblem stellt. Konjugierte Suchrichtungen sind damit erst im zweiten Durchlauf der inneren Iteration möglich.

Bei der Bestimmung der Schrittweite mit Hilfe der kubischen Interpolation werden die Schrittweiten $\rho_0, \rho_1 := \frac{\rho_0}{2}$ und $\rho_2 := \frac{\rho_0}{4}$ ohnehin ausgerechnet. Dort kann nach dem Prinzip der Armijo-Liniensuche vorgegangen werden. Falls also

$$\mathcal{D}_{R_{r,t}, T_{r,t}}(\phi(\vec{u} + \rho_i \cdot \vec{v})) > \mathcal{D}_{R_{r,t}, T_{r,t}}(\phi(\vec{u} + \rho_{i-1} \cdot \vec{v}))$$

für $i = 1$ oder $i = 2$ gilt, so wird ρ_i als Schrittweite genommen und die kubische Interpolation entfällt in dieser Iteration.

Die Liniensuche bedarf besonderer Beachtung, da sie der rechenaufwändigste Teil ist. Jede Auswertung des Funktionals \mathcal{D} bedarf

1. der Berechnung des ungeglätteten verformten Bildes $T_r \circ \phi$
2. der Glättung von $T_r \circ \phi$ zu $(T_r \circ \phi)_s$
3. der Berechnung des Gradienten
4. dem Aufsummieren des punktweise berechneten Kreuzproduktes $(\nabla R_{r,s}(\vec{x}) \times \nabla (T_r \circ \phi)_s(\vec{x}))^2$.

So betrachtet ist der Rechenaufwand für die kubische Interpolation mit Berücksichtigung aller Spezialfälle vernachlässigbar, wenn dadurch bereits eine Abstandsauswertung \mathcal{D} eingespart werden kann. Es ist sinnvoll die Frequenzraumgewichte $\exp(-\pi^2 \cdot \lambda \cdot t)$ zur Wärmeleitungsgleichung zwischen zu speichern, da sie unter anderem wegen (2.) sehr häufig gebraucht werden. Sie ändern sich erst beim Wechsel der Glättungs- bzw. Auflösungsstufe.

Zum Lösen der Randwertaufgabe mit der Sinus- oder der Cosinustransformation empfiehlt es sich, die inversen Eigenwerte ebenfalls zwischen zu speichern, da sie in jeder inneren Iteration n Mal verwendet werden und sich nur beim Wechsel der Auflösungsstufe ändern.

Werden die drei in diesem Kapitel angegebenen Algorithmen ineinander eingesetzt, so ergeben sich drei ineinander verschachtelte Iterationen. Die äußere iteriert über die Auflösung der Bilder, die mittlere iteriert über die Glättungsstufen und die innere registriert die Bilder auf der jeweiligen Auflösungs- und Glättungsstufe. Die Liniensuche ist eine vierte noch weiter innen liegende Iteration.

6.4 Der Speicherbedarf

Dem Speicherbedarf wird aus dem Grunde hier ein Abschnitt gewidmet, da er ein vielfaches der Bilddatenmenge ergibt. Bereits bei dreidimensionalen Bildern wird schnell die Obergrenze der Rechenmaschine erreicht.

Den wesentlichen Speicherbedarf macht aus

1. die gegebenen Bilder R und T .
2. die in der Auflösung reduzierten Bilder R_r und T_r .
3. die geglätteten Bilder $R_{r,s}$ und $(T_r \circ \phi)_s$.
4. das Verschiebungsfeld \vec{u} , welches von der Datenmenge her betrachtet n Bildern entspricht.
5. die Gradienten $\nabla R_{r,s}, \nabla (T_r \circ \phi)_s$ und für die Berechnung der Kräfte wird auch noch $\nabla T_{r,s} \circ \phi$ gebraucht.
6. das Verschiebungsfeld \vec{w} zur Suchrichtung, beim CG-Verfahren auch noch das von der vorangegangenen Iteration.
7. die Kräfte \vec{f}, \vec{f}_l und \vec{f}_c deren Datenmenge jeweils n Bildern entspricht.

8. die Gewichte für die Wärmeleitungsgleichung und die Randwertaufgabe entsprechen je der Größe eines Bildes.

Die Eingangsbilder werden zu Anfang bis auf die größte Auflösungsstufe herunter gerechnet. Die Bilder zu den Zwischenauflösungen können dabei ausgelagert werden. Bei jedem Wechsel der Auflösungsstufe wird dann das benötigte Bild wieder in den Arbeitsspeicher eingelagert.

Die Suchrichtung kann sich den Speicherplatz mit dem Kräftevektor teilen. Andererseits benötigt die trigonometrische Transformation auch noch zusätzlichen Speicher.

Bei der Liniensuche wird das aktuell getestete Verschiebungsfeld berechnet, da in dem Algorithmus die nächste Schrittweite feststellt, ob die vorangegangene die optimale ist, muss das Verschiebungsfeld aus der vorangegangenen entweder aufgehoben oder neu berechnet werden. Dies gilt für das Verschiebungsfeld, das daraus berechnete verschobene und geglättete Bild sowie für die Gradienten davon.

Damit kommt der Arbeitsspeicherbedarf auf etwa die 40-fache Bildgröße. Bei einem dreidimensionalen Bild von 128^3px^3 benötigt der Algorithmus bei einer Rechengenauigkeit mit 64-Bit Gleitkommazahlen (double precision) in der Praxis 600 MB Arbeitsspeicher und bei 256^3px^3 sind es bereits 4800 MB, was den Adressraum eines 32-Bit-Systems bereits übersteigt.

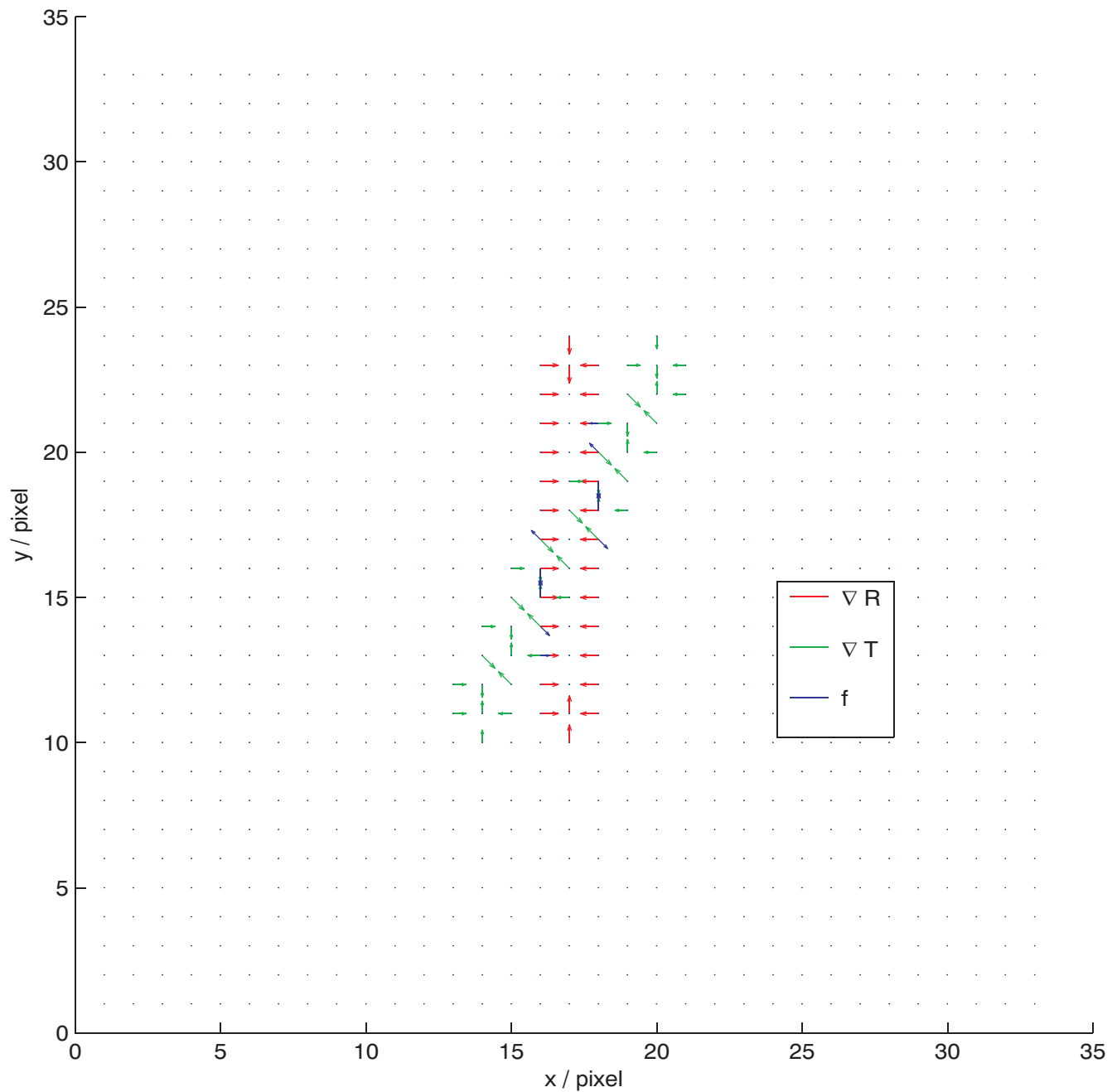
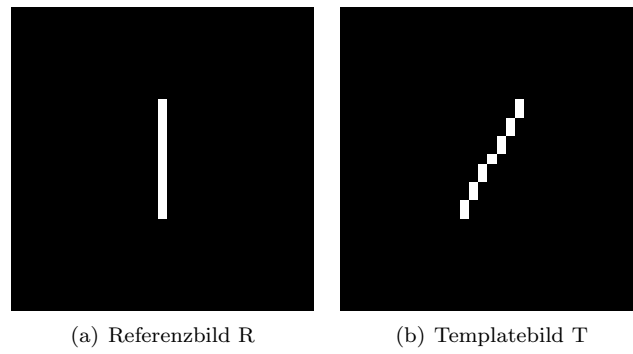
6.5 Veranschaulichung des Registrierungsablaufes an einem einfachen zweidimensionalen Beispiel

Das Prinzip und der Ablauf der morphologischen multimodalen Bildanpassung wird im folgenden anhand zweier ganz einfachen Beispielbilder veranschaulicht. Das Referenzbild R enthält eine senkrechte Strecke und das Templatebild T die gleiche Strecke etwas verdreht. Der Mittelpunkt der beiden Strecken befindet sich im gleichen Pixel. Siehe dazu in Abbildung 6.5 auf der nächsten Seite (a) und (b). In (c) ist das Gradientenfeld ∇R (rot) und ∇T (grün), sowie die rechte Seite der Randwertaufgabe, also die Kräfte \vec{f} (blau) unter Verwendung von Maß d (vgl. 2.1.1 auf Seite 6) eingezeichnet. Deutlich zu erkennen ist, dass \vec{f} nur in den Pixeln von Null verschieden ist, wo weder ∇R noch ∇T verschwinden. Die Pixel außerhalb des Schnittpunktes müssen verschoben werden, es treten aber im Randbereich der Strecken keine Kräfte \vec{f} auf. Dies ist darauf zurück zu führen, dass das Abstandsmaß $d(R, T; \vec{x})$ lokal definiert ist. Die Randpixel der Strecke sind nicht unmittelbar benachbart. Somit haben sie keinen gegenseitigen Einfluss.

Um dieses Problem zu lösen, werden die Bilder geglättet und somit die Kanten verbreitert. Dies ist in Abbildung 6.5 auf Seite 49 gezeigt. Die Gradientenfelder sind nun über ganz Ω verteilt, werden im Bereich der Strecken allerdings stärker. Die Vektoren der Kräfte \vec{f} haben jetzt eine sinnvolle Anordnung. In Abbildung 6.5 auf Seite 50 ist in (b) dargestellt, wie die Kräfte \vec{f} das Bild verformen. Dabei bleiben die Randpixel der Strecke stehen, während sich die Punkte zwischen Mittelpunkt und Rand zur richtigen Position schieben. Dieses „Zerreißen“ des Bildes ist nicht gewollt. Daher wird das Kraftfeld \vec{f} durch Lösen der Randwertaufgabe und hinzu addieren des linearen Anteils regularisiert. In (c) und (d) ist dargestellt, wie sich das Templatebild T mit verschiedenen Schrittweiten der regularisierten Kräfte \vec{u} verformt. Der Zusammenhalt der Strecke ist bei dieser Verformung sichergestellt.

Eine zweite Möglichkeit, die Bildobjekte dichter an einander zu bringen besteht darin, die Auflösung zu reduzieren. Das Fullweighting beinhaltet dabei ebenfalls ein leichtes Glätten. Darüber hinaus sind die Pixel dichter an einander gerückt und die Bilder somit ähnlicher. Dies ist in 6.5 auf Seite 51 ff. zu sehen. Sind die Bilder auf der niedrigen Auflösung hinreichend genau registriert, dann wird das Verschiebungsfeld hochinterpoliert und als Startlösung für die nächsthöhere Auflösungsstufe genommen. Durch die reduzierte Auflösung werden außerdem – auf die Originalauflösung bezogen – globalere Verschiebungen gemacht, diese dafür in größeren Schrittweiten. Dies ist das gleiche Prinzip wie bei Mehrgitterverfahren.

Im Gegensatz zu Mehrgitterverfahren wird bei der Bildregistrierung monoton vom größten zum feinsten Gitter gearbeitet. Der Grund dafür ist, dass es keinen Sinn macht Detailstrukturen auszurichten, solange nicht sichergestellt ist, dass das Bild in seinen Grobstrukturen richtig sitzt. Angenommen die Bildanpassung durchläuft einen W-Zyklus über drei Auflösungsstufen. Dann werden die Grobstrukturen ein paar Iterationen lang auf grober Auflösung ausgerichtet. Anschließend wird mit diesem Ergebnis auf mittlerer Auflösung iteriert. Dabei werden die Bilddetails von T^ϕ anhand einer kleinen Umgebung der korrespondierenden Stelle in R ausgerichtet. Nun stünde an, die Bildregistrierung auf grober Auflösung nach zu korrigieren. Hier liegt der Widerspruch. Dadurch dass die Details auf mittlerer Auflösung ausgerichtet werden, wird angenommen, dass die Grobausrichtung bereits stimmt. D.h. es macht keinen Sinn anschließend wieder zur größeren Auflösung zurück zu kehren. Wird andererseits angenommen, dass die Grobausrichtung noch nicht stimmt, so macht es keinen Sinn auf mittlerer Auflösung feinere Strukturen anzupassen, denn diese werden nach den Details in unmittelbarer Umgebung des Punktes im Referenzbild R ausgerichtet und bei unzureichender Grobanpassung gehören die Details zu zwei verschiedenen Objekten. Aus Sicht der Inversen Problemstellung beruht dieser Effekt auf den vielen lokalen Minima der Funktion $\mathcal{D}(\phi)$.



(c)

Abbildung 6.3: Die Ausgangsbilder R und T mit einer Auflösung von 33×33 Pixel sind in (a) bzw. (b) gezeigt. Das Vektorfeld der Gradienten ∇R (rot), ∇T (grün) und der Kräfte $\vec{f}(R, T)$ (blau) die sich mit Maß d ergeben, ist in (c). Es ist zu sehen, dass Kräfte nur in Punkten auftreten, wo weder ∇R noch ∇T verschwindet.

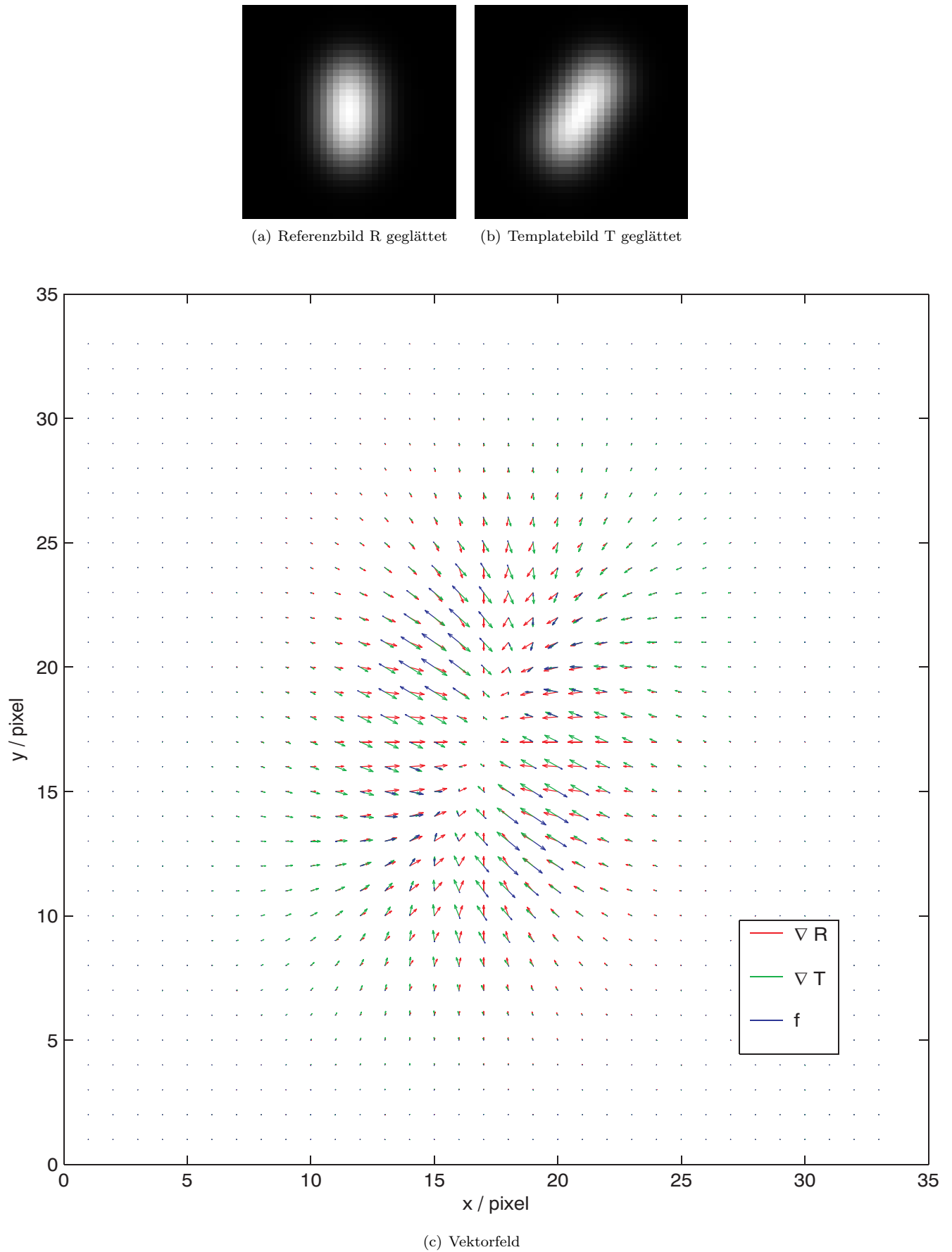
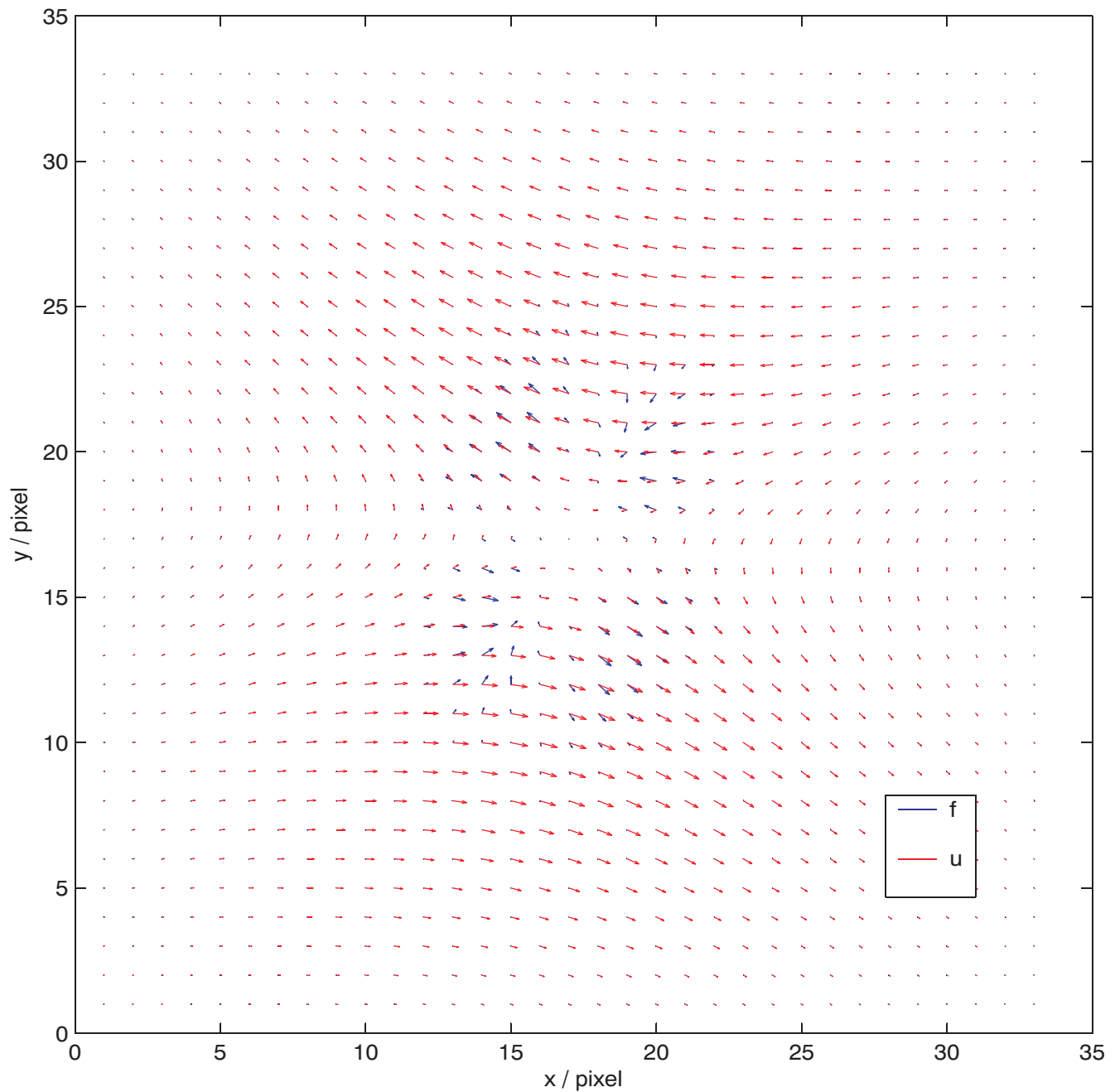


Abbildung 6.4: Die geglätteten Bilder R_s und T_s mit einer Auflösung von 33×33 Pixel sind in (a) bzw. (b) gezeigt. Das Vektorfeld der Gradienten ∇R_s (rot), ∇T_s (grün) und der Kräfte $\vec{f}(R_s, T_s)$ (blau) die sich mit Maß d ergeben, ist in (c) gezeichnet. Durch das Glätten ist der Überlappungsbereich der Gradientenfelder von R und T größer. Dementsprechend treten auch an mehr Stellen Kräfte $\vec{f}(R_s, T_s)$ auf.



(a) Vektorfeld

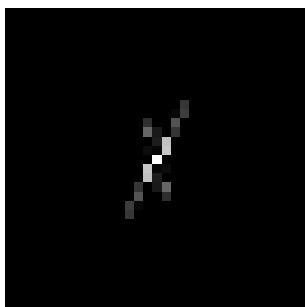
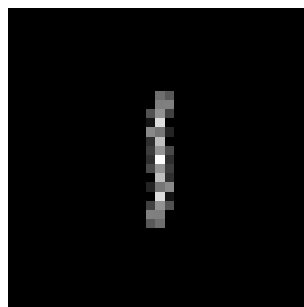
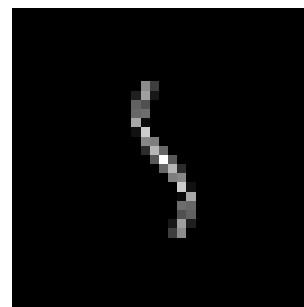
(b) verformtes Bild $T \circ \phi(\tau \cdot \vec{f})$ (c) verformtes Bild $T \circ \phi(\rho \cdot \vec{u})$ (d) verformtes Bild $T \circ \phi(2 \cdot \rho \cdot \vec{u})$

Abbildung 6.5: Das Vektorfeld der Kräfte \vec{f} und der biharmonisch regularisierten Kräfte \vec{u} , so wie das durch die Kräfte \vec{f} selbst (b) und durch die regularisierten Kräfte \vec{u} (c) verformte Templatebild. Hier wurde nur eine einzige Iteration durchgeführt und die Schrittweite $\tau = 4.6 \cdot 10^7$ und $\rho = 1.2 \cdot 10^6$ manuell bestimmt. Während das Bild in (b) bereits zerstückelt ist, bleibt der Zusammenhalt des Objektes mit der regularisierten Suchrichtung auch bei größeren Schrittweiten gewährleistet (d).

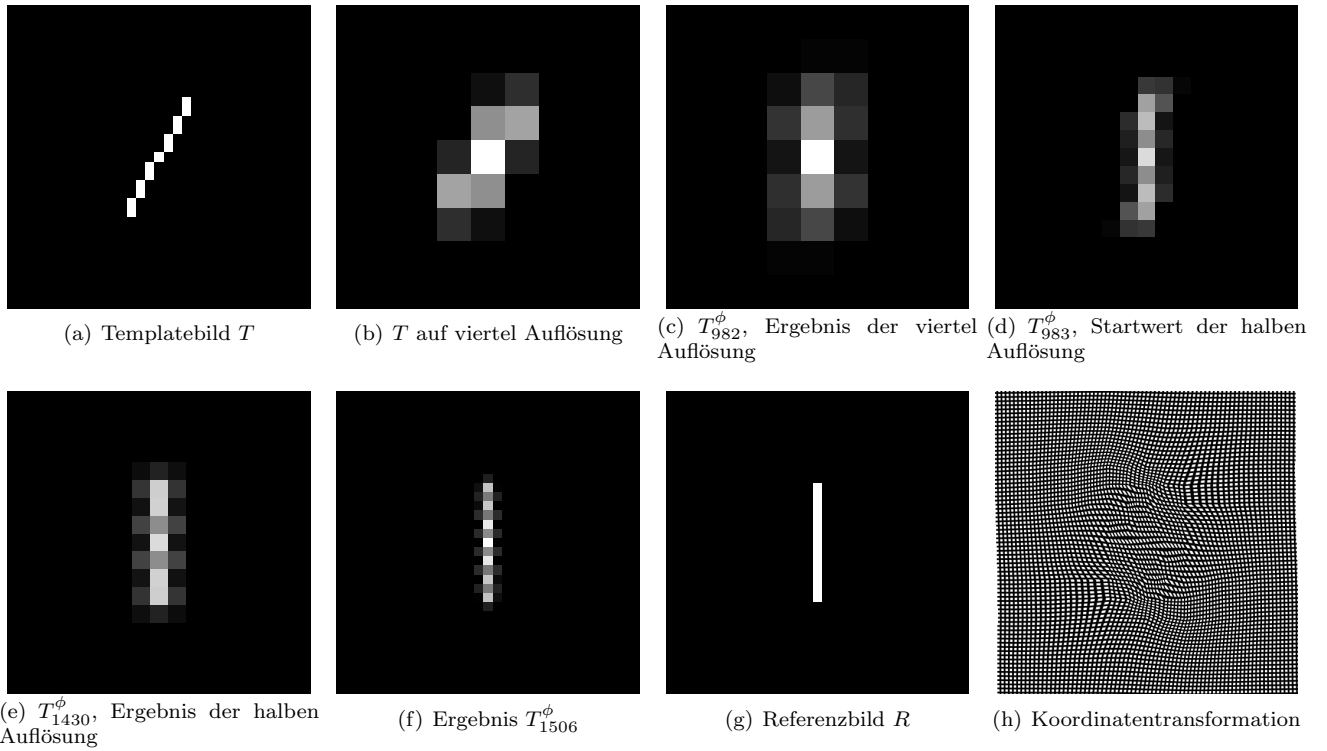


Abbildung 6.6: Ein Programmdurchlauf über 3 Auflösungsstufen mit Maß d und harmonischer Randwertaufgabe mit Dirichlet Randbedingung. In der Koordinatentransformation (h) lässt sich die Transformation vom größten Gitter noch erkennen.

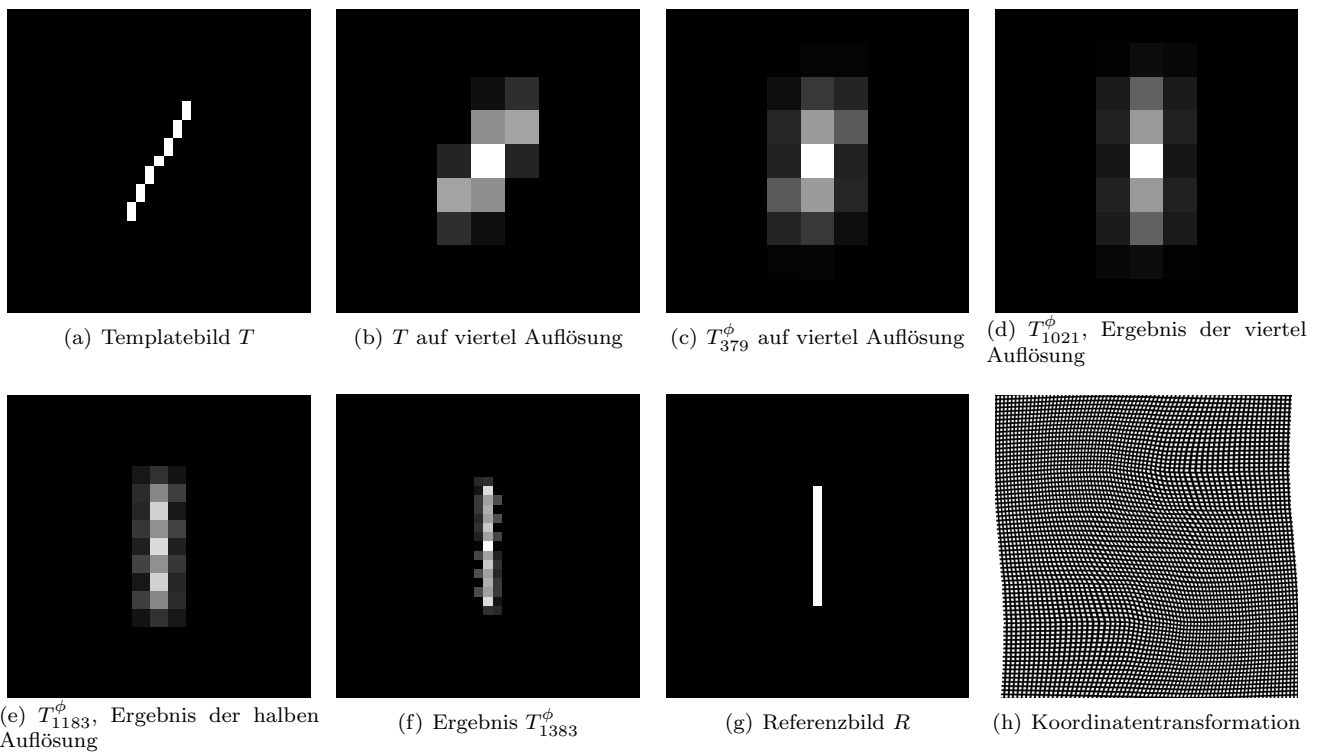


Abbildung 6.7: Ein Programmdurchlauf über 3 Auflösungsstufen mit Maß d und biharmonischer Randwertaufgabe mit Dirichlet Randbedingung. Die Koordinatentransformation ist deutlich glatter als bei der harmonischen Regularisierung.

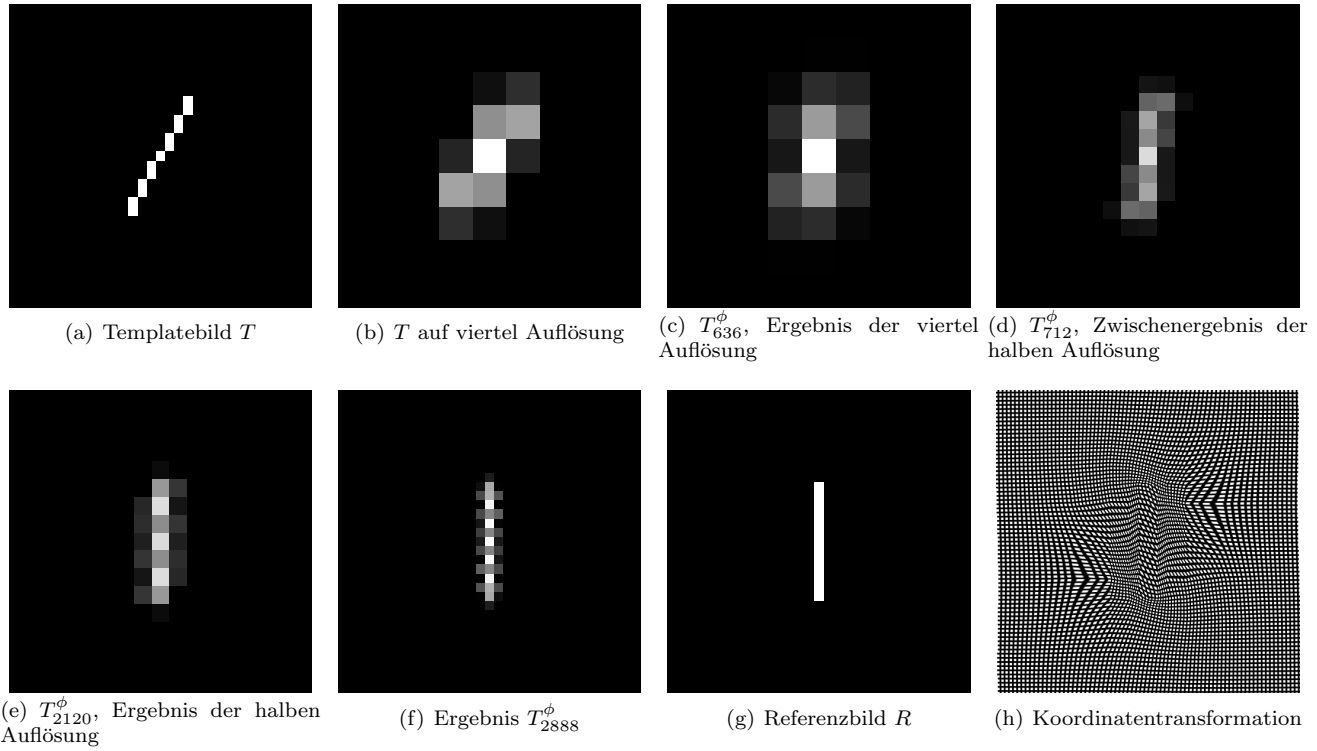


Abbildung 6.8: Ein Programmdurchlauf über 3 Auflösungsstufen mit Maß \tilde{d} und harmonischer Randwertaufgabe mit Dirichlet Randbedingung.

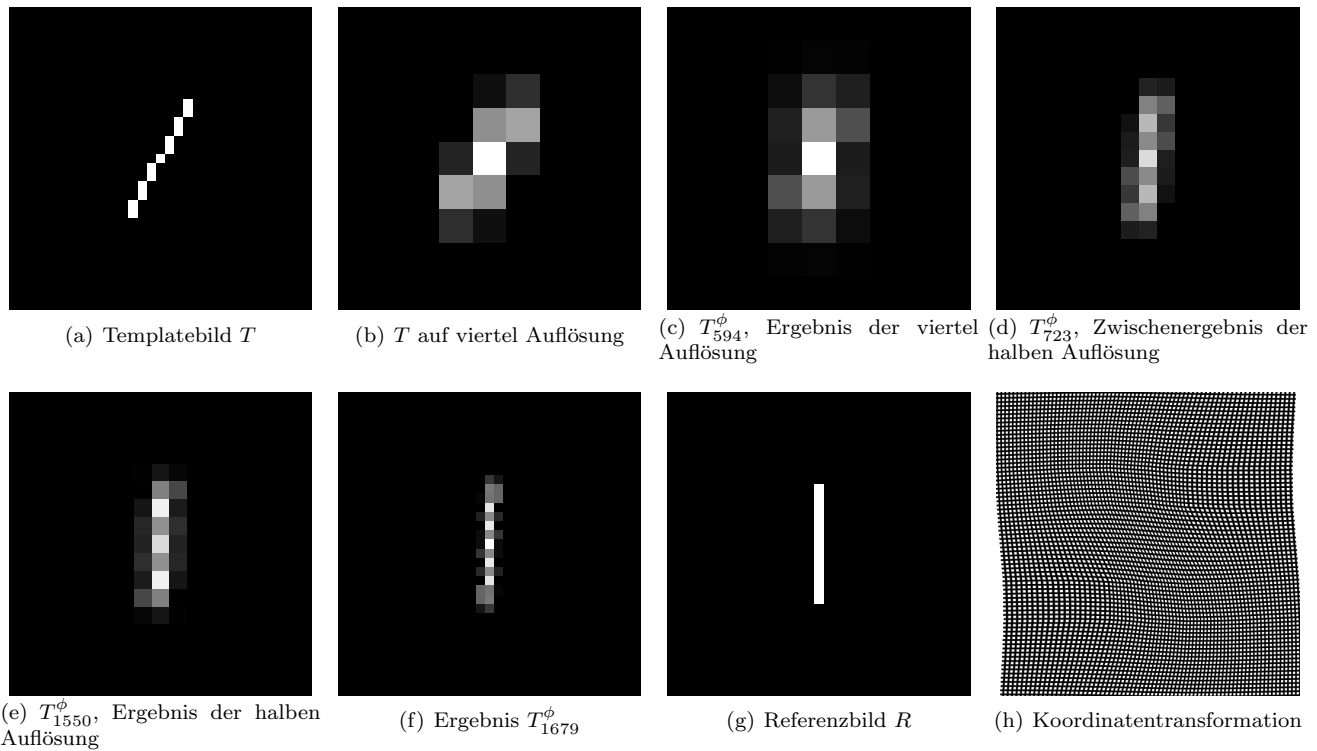
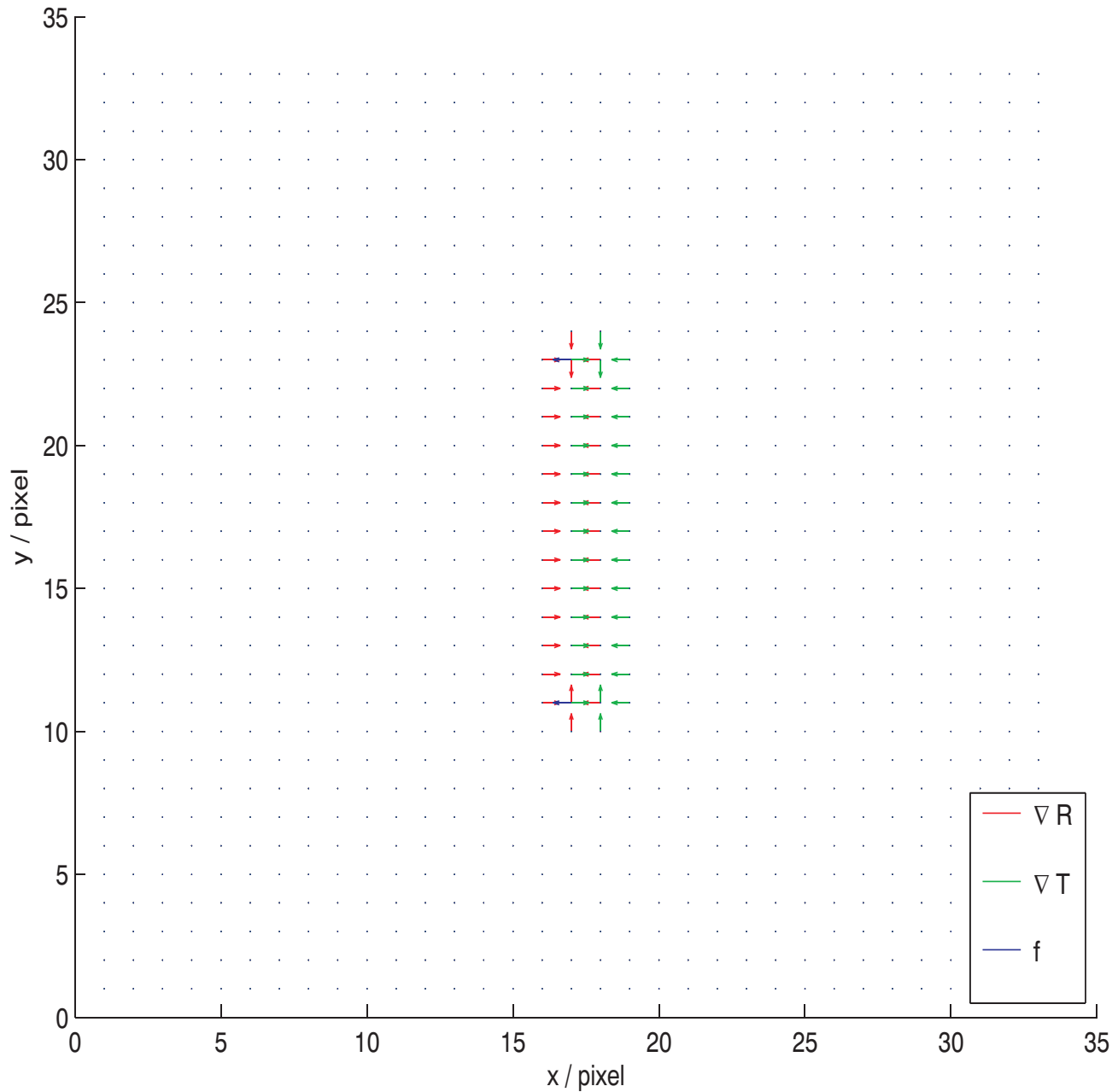
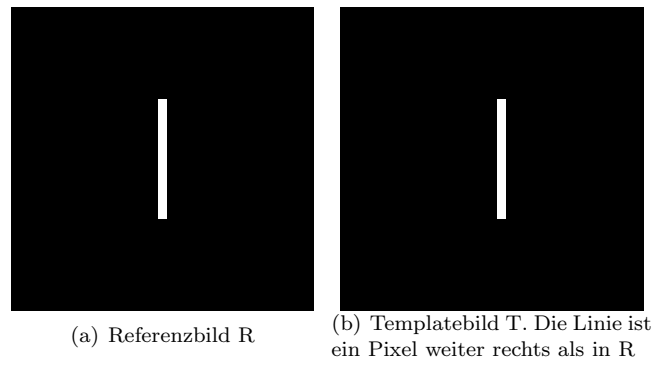


Abbildung 6.9: Ein Programmdurchlauf über 3 Auflösungsstufen mit Maß \tilde{d} und biharmonischer Randwertaufgabe mit Dirichlet Randbedingung. Auch hier ist die Koordinatentransformation (h) deutlich glatter als bei der harmonischen Regularisierung.

6.6 Veranschaulichung des Registrierungsablaufes an einem zweiten Beispiel

Auf den nächsten drei Seiten wird die Bildanpassung mit einem anderen Templatebild T wiederholt. Diesmal ist die Strecke in T lediglich um ein Pixel gegenüber R nach rechts verschoben. Ungeglättet gibt es nur an den Endpunkten nicht-parallele Gradienten. Dementsprechend treten nur zwei nicht-verschwindende Kräftevektoren auf. Erst die Glättung der Bilder sorgt dafür, dass längs der Strecke Verschiebungskräfte entstehen. Nach der Regularisierung haben diese sogar genau die richtige Richtung. Lediglich die Länge der regularisierten Verschiebungsrichtung variiert leicht.



(c)

Abbildung 6.10: Die Ausgangsbilder R und T mit einer Auflösung von 33×33 Pixel sind in (a) bzw. (b) gezeigt. Das Vektorfeld der Gradienten ∇R (rot), ∇T (grün) und der Kräfte $\vec{f}(R, T)$ (blau) die sich mit Maß d ergeben, ist in (c). Es ist zu sehen, dass lediglich zwei Kräfte in den beiden Endpunkten auftreten.

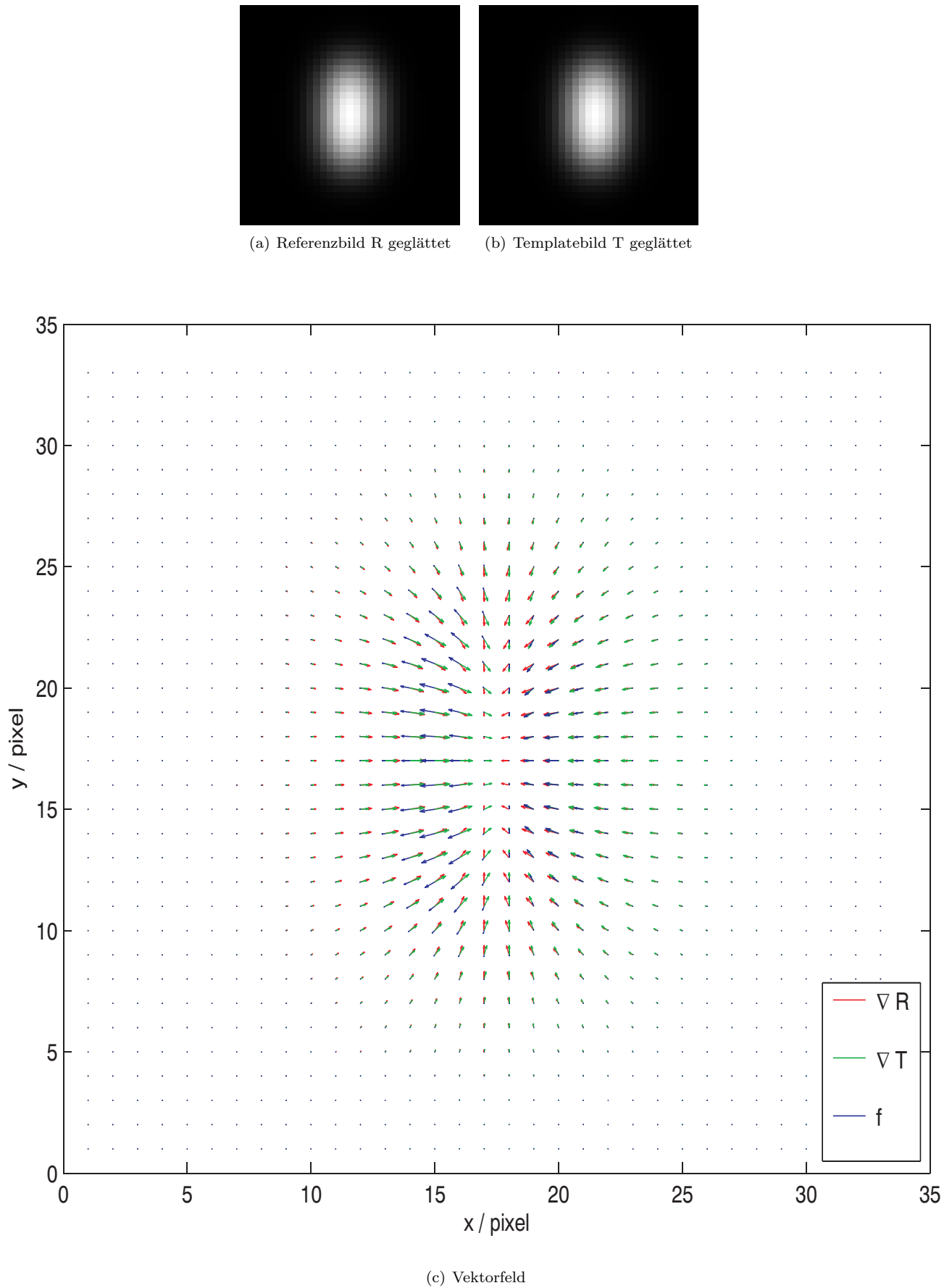
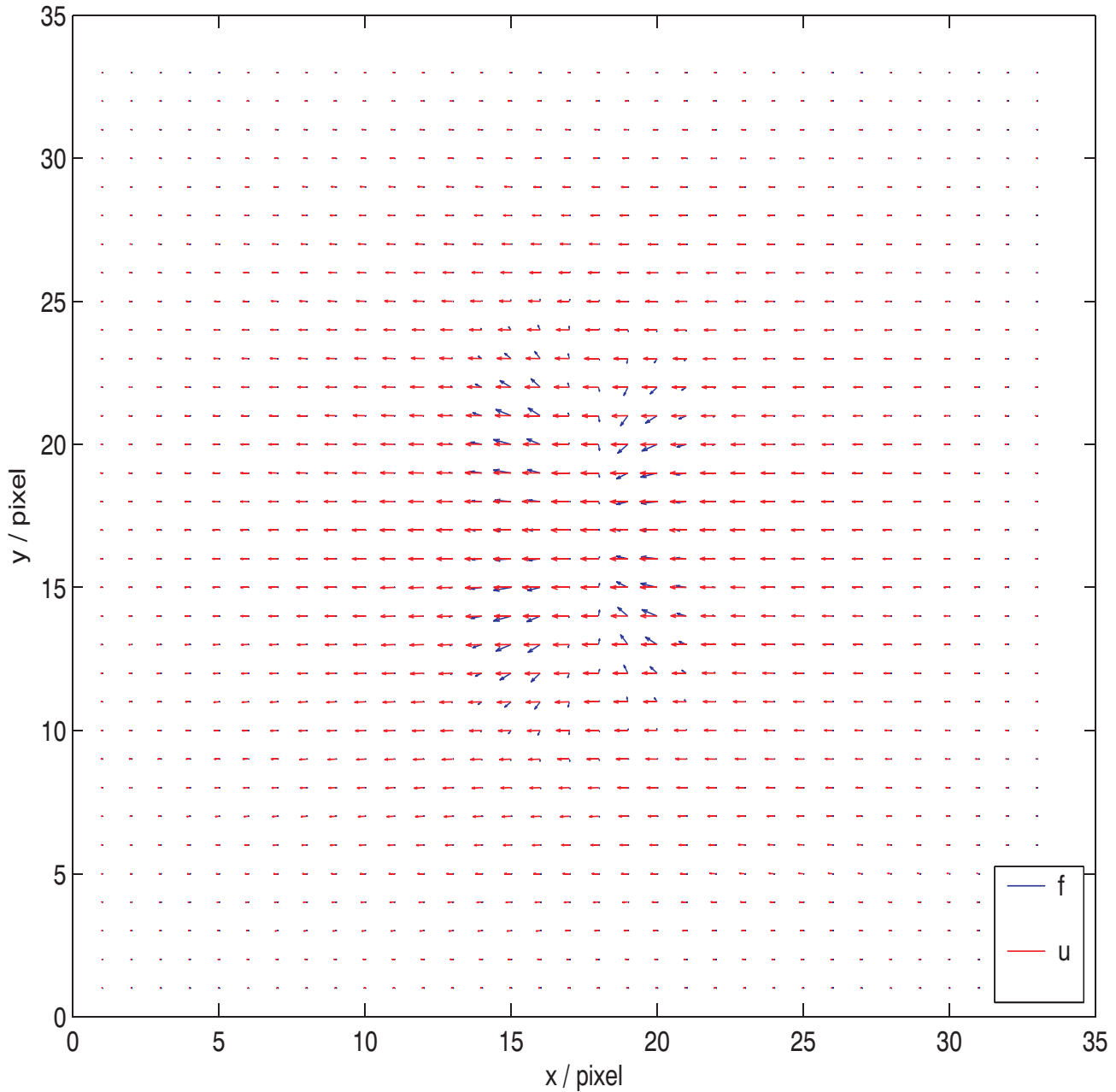


Abbildung 6.11: Die geglätteten Bilder R_s und T_s mit einer Auflösung von 33×33 Pixel sind in (a) bzw. (b) gezeigt. Das Vektorfeld der Gradienten ∇R_s (rot), ∇T_s (grün) und der Kräfte $\vec{f}(R_s, T_s)$ (blau) die sich mit Maß d ergeben, ist in (c) gezeichnet. Durch das Glätten gibt es mehr nicht-parallele Gradienten. Dementsprechend treten auch an mehr Stellen Kräfte $\vec{f}(R_s, T_s)$ auf.



(a) Vektorfeld

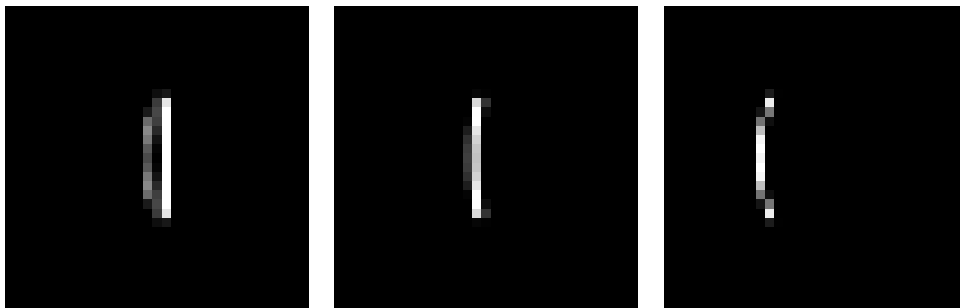
(b) verformtes Bild $T \circ \phi(\tau \cdot \vec{f})$ (c) verformtes Bild $T \circ \phi(\rho \cdot \vec{u})$ (d) verformtes Bild $T \circ \phi(4 \cdot \rho \cdot \vec{u})$

Abbildung 6.12: Das Vektorfeld der Kräfte \vec{f} und der biharmonisch regularisierten Kräfte \vec{u} , so wie das durch die Kräfte \vec{f} selbst (b) und durch die regularisierten Kräfte \vec{u} (c) verformte Bild. Hier wurde nur eine einzige Iteration durchgeführt und die Schrittweite $\tau = 1.3 \cdot 10^8$ und $\rho = 4.2 \cdot 10^5$ manuell bestimmt. Die leichte Krümmung in (c) wird normalerweise von den nachfolgenden Iterationen kompensiert.

7 Ergebnisse

In diesem Kapitel werden Bildanpassungen von Testbildern gezeigt und besprochen. Es werden also Resultate aus der Praxis vorgestellt und versucht theoretisch nachzuvollziehen. Die meisten Beispiele sind zweidimensional. Zum einen lassen sich zweidimensionale Beispielbilder recht einfach mit Grafik- und Fotobearbeitungsprogrammen anfertigen, zum anderen können sie vollständig auf Papier dargestellt werden. Bei dreidimensionalen Bildern kann immer nur eine Ebene gezeigt werden. Für einen Eindruck der dreidimensionalen Bilder sind entsprechend viele Ebenen notwendig.

Der Algorithmus zur Bildanpassung wurde in Matlab implementiert. Einige Unterrountinen wie die Interpolation und die Auflösungswechsel sind in C geschrieben und über das mex-Interface eingebunden. Aus Performance Gründen wird für die (Co-) Sinustransformation das Programmpaket von FFTW in der Version 3.1.2 verwendet. Die zweidimensionalen Beispiele sind auf einem 32-Bit Pentium-4TM Prozessor unter Matlab 6.5 gerechnet. Die dreidimensionalen Beispiele wurden wegen des Speicherbedarfs auf einem 64-Bit AMD OpteronTM Prozessor 246 unter Matlab 7.3 gerechnet. Der Programmcode und dreidimensionale Bildanpassungen sind auf der beiliegenden CD (Seite 83) zu finden.

Im folgenden wird mit d das auf Seite 6 beschriebene Maß und mit \tilde{d} das Maß von Seite 12 bezeichnet.

7.1 Die Wirkung der Regularisierung

Die Regularisierung hat die Aufgabe die Topologie des Bildes bei der Transformation zu erhalten. Zuvor wurde dazu die harmonische und die biharmonische Randwertaufgabe vorgestellt. Nun soll überprüft werden, in wie weit diese Forderung in der Praxis erfüllt ist. Dazu wird das in Abbildung 7.2 auf Seite 59 gezeigte Referenzbild R (a) und das Templatebild T (b) verwendet. Das Ergebnis T^ϕ der Bildanpassung mit biharmonischer Regularisierung ist in (g) gezeigt. Bis auf ein paar Kleinigkeiten stimmen T^ϕ und R überein. Der untere Indices bei T_{xxx}^ϕ ist die gesamte Anzahl der inneren Iterationen, die über alle Auflösungs- und Glättungsstufen gezählt ist. In Abbildung 7.3 auf Seite 60 wurde die gleiche Aufgabe mit der harmonischen Randwertaufgabe wiederholt. Die grobe Form von T^ϕ stimmt auch hier mit dem Referenzbild R überein. Es treten allerdings an vielen Stellen lokale artifizielle Deformationen auf, die sich nicht sinnvoll mit Bildmerkmalen erklären lassen.

Theoretisch lässt sich diese Erscheinung durch Vergleich der Ordnung der höchsten Ableitung im Abstands- (\mathcal{D}) und Regularisierungsfunktional (S) im Minimierungsproblem erklären. In \mathcal{D} kommt bereits die erste Ableitung der Transformationsabbildung vor. In so fern reicht es nicht aus, im Strafterm S die Größe des Gradienten, also auch der ersten Ableitung zu verwenden. Damit die erste Ableitung der Transformation nicht stellenweise zu groß wird, muss im Strafterm mit mindestens der Größe der nächsthöheren Ableitung gemessen werden. Das ist bei S_2 mit dem Laplaceoperator erfüllt.

Der Vergleich der maximal verwendeten Ableitung kann auch bei der Randwertaufgabe durchgeführt werden. Auf der rechten Seite kommen in den Kräften \tilde{f} Ableitungen der Transformationsabbildung bis zur zweiten Ordnung vor. D.h. auf der linken Seite muss ein Operator stehen, der höhere als die zweiten Ableitungen der Transformation enthält. Bei der harmonischen Regularisierung steht links in der RWA der Laplace Operator. Das sind genau die zweiten Ableitungen und damit reicht der Operator hier nicht aus. Bei der biharmonischen Regularisierung kommen die vierten Ableitungen auf der linken Seite der RWA vor, so dass er ausreicht.

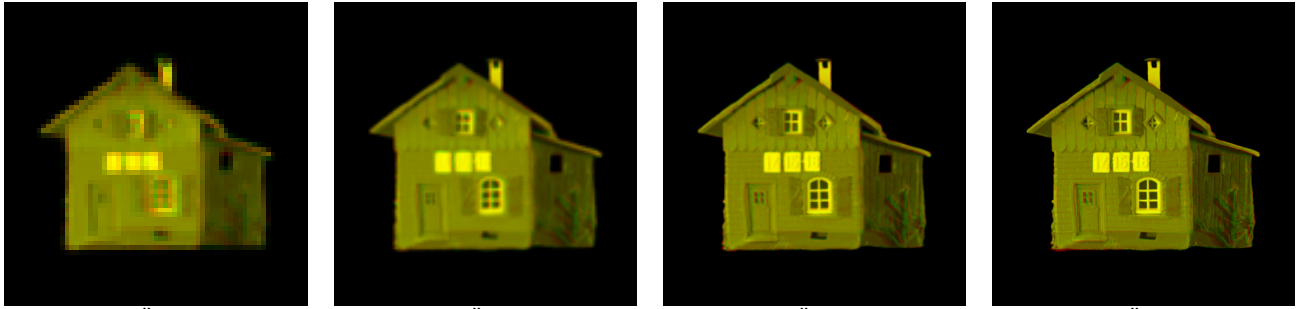
Es stellt sich hier die Frage, nach einer RWA bei der auf der linken Seite die dritten Ableitungen vorkommen. Die (bi-) harmonische Randwertaufgabe kann, wie bereits beschrieben, mit der Sinus- bzw. Cosinustransformation ausgerechnet werden. Dabei wird im Frequenzraum durch die Eigenwerte dividiert. Nun sind die Eigenwerte des biharmonischen Operators genau die Quadrate von den Eigenwerten λ des harmonischen Operators. Insofern lassen sich Regularisierungen bilden, die zwischen der harmonischen und der biharmonischen liegen. Dazu wird im Frequenzraum durch λ^a ($1 < a < 2$) dividiert. Der Zwischen-Operator zu $a = \frac{3}{2}$ wird in dieser Arbeit als „sesqui-harmonischer“ Operator bezeichnet, da er genau der Mitte vom einfach-harmonischen und dem biharmonischen Operator entspricht. Die Zwischenoperatoren sind nur im Frequenzraum sinnvoll definierbar, denn sie entsprächen so etwas wie Δ^a . Bei der praktischen numerischen Rechnung kann der Exponent $a \geq 1$ beliebig gewählt werden. Die Wahl $a \geq 3$ ist allerdings so streng, dass in Beispielen keine nennenswerte Transformation stattfindet.

In Abbildung 7.4 auf Seite 60 wird die Bildanpassung mit der „sesqui-harmonischen“ Randwertaufgabe wiederholt. Im Gegensatz zur harmonischen ist nur eine leichte Deformation am Dachgiebel zu erkennen. An mehreren Stellen sind im transformierten Gitter leichte Verwacklungen wahrzunehmen. Der einzige Vorteil gegenüber dem biharmonischen Operator ist in diesem Beispiel, dass der linke Teil des Türrahmens etwas besser zu R passt.

Der Sesqui-harmonische Operator regularisiert damit auch weniger als notwendig ist, um hinreichend glatte Transformationen zu erhalten. Das ist wieder mit dem ursprünglichen Minimierungsproblem zu erklären. Dort entspricht der sesqui-harmonische Operator der anderthalben Ableitungen im Straffunktional S . Das ist zwar mehr, als im Abstandsunktional \mathcal{D} , aber mit weniger als einer Ableitungsordnung Unterschied nicht genug.

(a) Referenzbild R (b) Templatebild T (c) $1 - \|\nabla R\|_2$ (d) $1 - \|\nabla T\|_2$

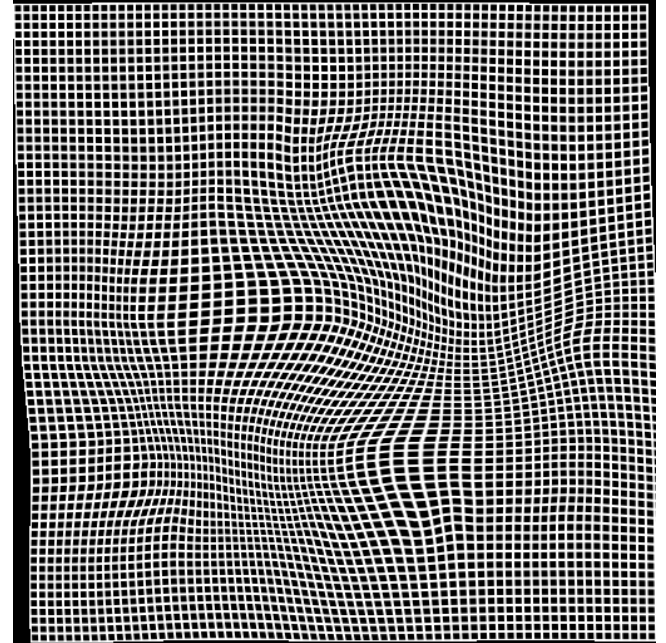
Abbildung 7.1: Das Modellproblem. Gegeben sind das Referenzbild R und das Templatebild T . Die Aufgabe besteht darin, das Templatebild T so zu verformen, dass seine Gestalt zum Referenzbild R passt. Dabei werden nur Informationen aus den Gradienten verwendet. Die Höhenlinien sind in der zweiten Zeile zu sehen. Die Auflösung aller vier Bilder beträgt $513 \times 513\text{px}^2$.



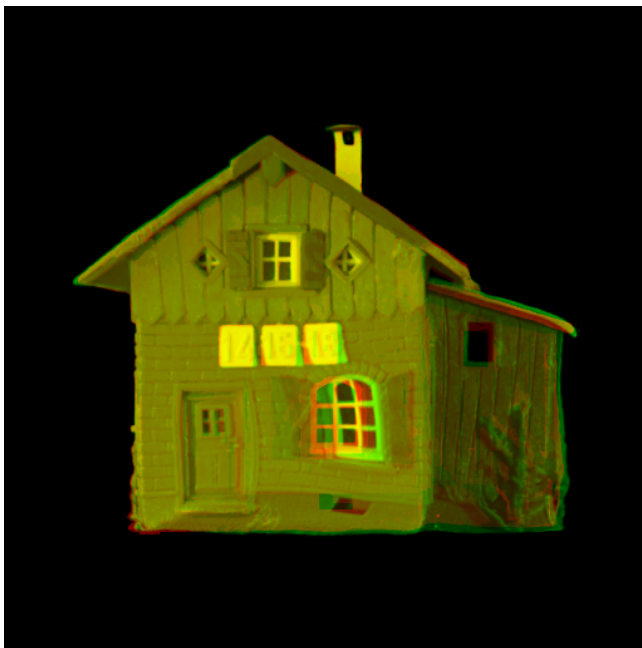
(a) Rot-Grün-Überlagerung von R und T_{227}^ϕ mit CG, Startwert zu 65^2px^2 (b) Rot-Grün-Überlagerung von R und T_{374}^ϕ mit CG, Startwert zu 129^2px^2 (c) Rot-Grün-Überlagerung von R und T_{462}^ϕ mit CG, Startwert zu 257^2px^2 (d) Rot-Grün-Überlagerung von R und T_{513}^ϕ mit CG, Startwert zu 513^2px^2



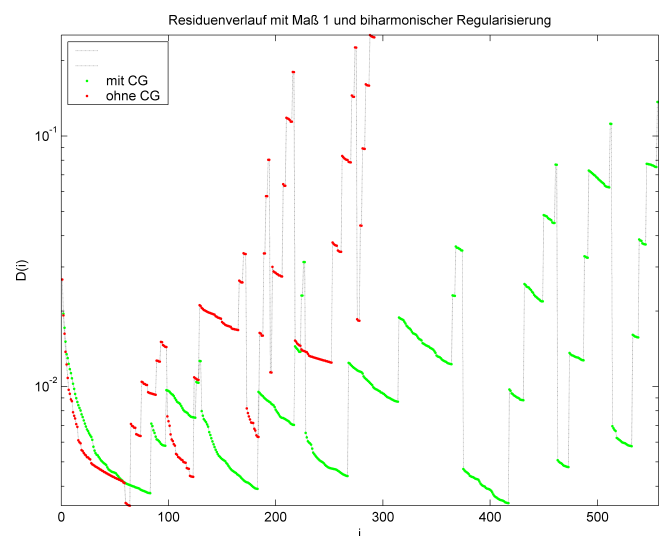
(e) Rot-Grün-Überlagerung von R und Ergebnis T_{557}^ϕ mit Maß d und CG-Verfahren



(f) Koordinatentransformation ϕ zu T_{557}^ϕ



(g) Rot-Grün-Überlagerung von R und Ergebnis T_{292}^ϕ mit Maß d ohne CG-Verfahren

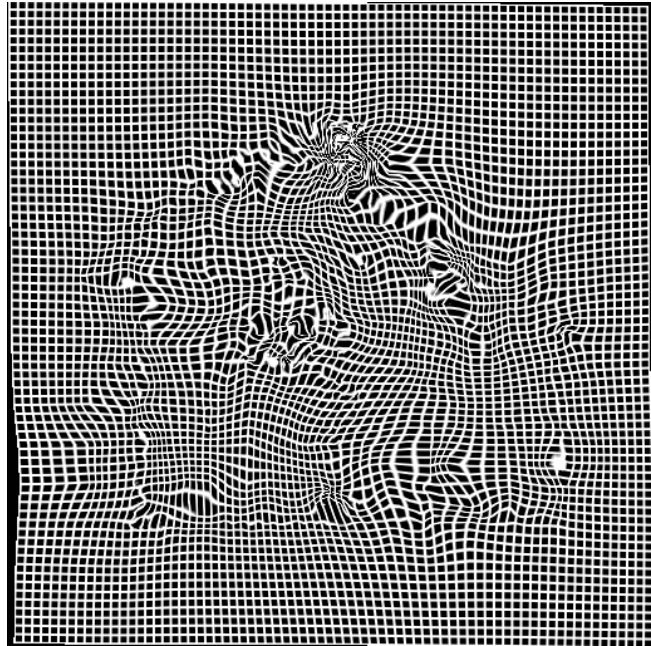


(h) Residuenverlauf

Abbildung 7.2: Ein Beispiel mit Maß d (vgl. 2.1.1 auf Seite 6). Gegeben sind die Bilder R und T aus Abb. 7.1 auf der vorherigen Seite mit einer Auflösung von $513 \times 513 \text{px}^2$. Die Registrierung wurde auf einer Auflösung von $17 \times 17 \text{px}^2$ begonnen und schrittweise verdoppelt bis zur vollen Auflösung. Die Rechenzeit betrug 2 Minuten. Die Glättung erfolgt mit der Cosinustransformation und die Regularisierung mit der Sinustransformation.



(a) harmonisch regularisiertes T_{1046}^ϕ mit Verwendung des CG-Verfahrens

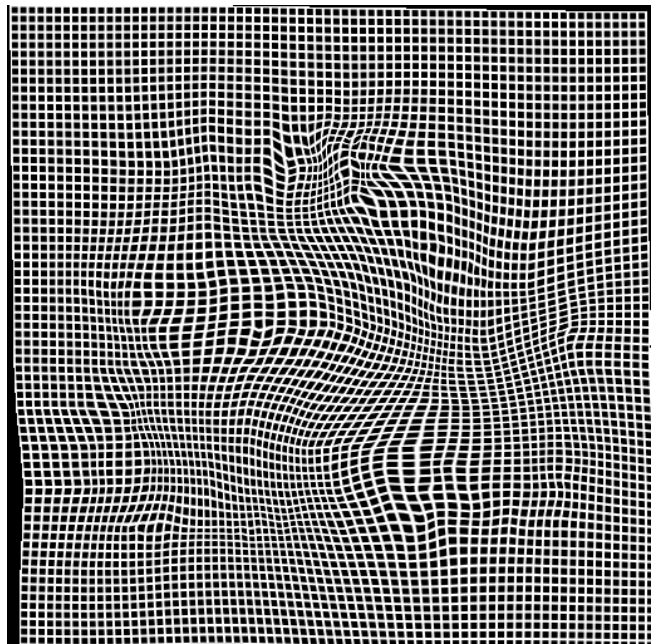


(b) Koordinatentransformation ϕ zu T_{1046}^ϕ

Abbildung 7.3: Gegeben sind wieder R und T aus Abb. 7.1 auf Seite 58. Diesmal wird die harmonische Regularisierung verwendet. Alle übrigen Parameter sind die gleichen wie in Abbildung 7.2 auf der vorherigen Seite. Es treten lokal starke artifizielle Deformationen auf, die sich nicht aus den Strukturen des Bildes nachvollziehen lassen. Die Residuen im Iterationsverlauf sind in Abb. 7.15 auf Seite 74 aufgetragen.

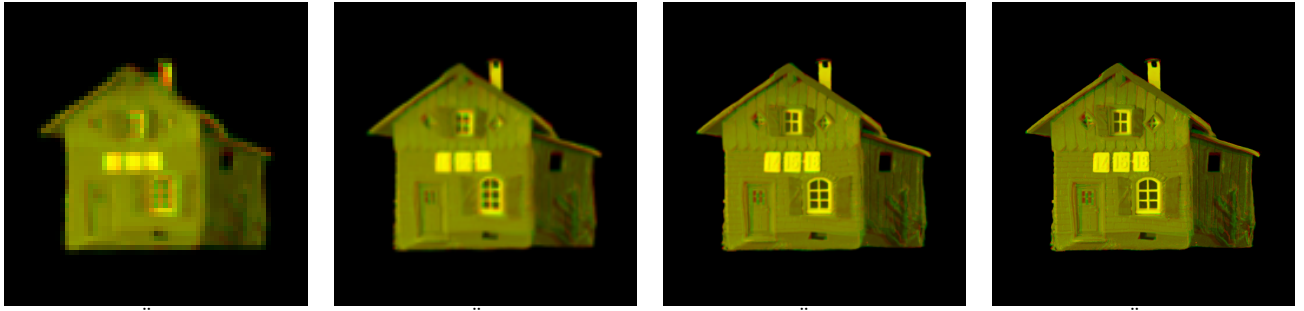


(a) sesquiharmonisch regularisiertes Templatebild T_{820}^ϕ mit Verwendung des CG-Verfahrens

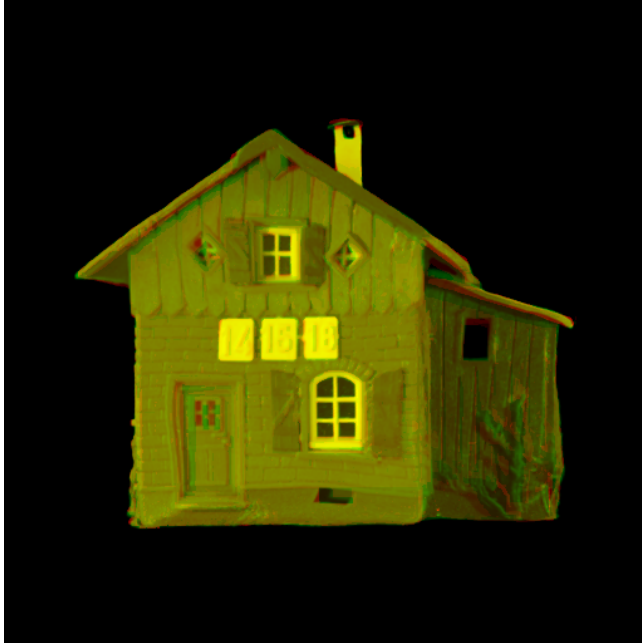


(b) Koordinatentransformation ϕ zu T_{820}^ϕ

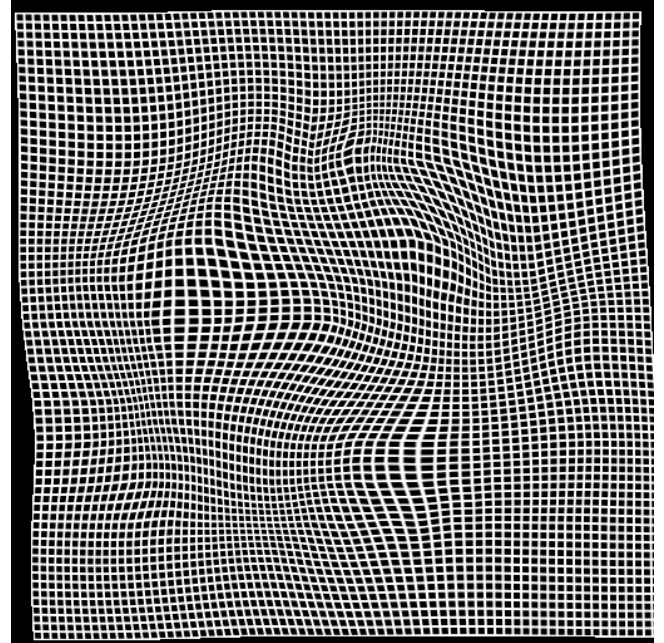
Abbildung 7.4: Gegeben sind wieder R und T aus Abb. 7.1 auf Seite 58. Diesmal wird die sesqui-harmonische Regularisierung verwendet. Alle übrigen Parameter sind die gleichen wie in Abbildung 7.2 auf der vorherigen Seite. Gegenüber der einfach-harmonischen Regularisierung in Abbildung 7.3 sind die artifizellen Deformationen hier deutlich geringer. Doch am Dachgiebel ist dieser Fehler immer noch zu erkennen. Die Residuen im Iterationsverlauf sind in Abb. 7.15 auf Seite 74 aufgetragen.



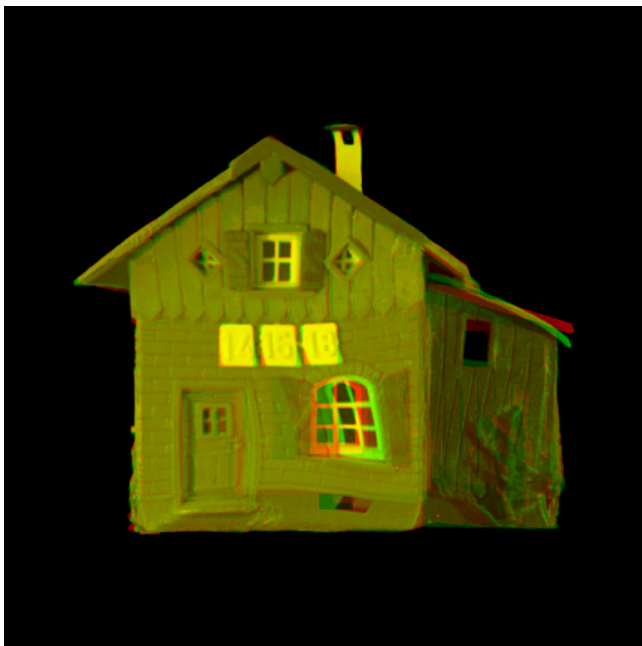
(a) Rot-Grün-Überlagerung von R und T_{112}^ϕ mit CG, Startwert zu 65^2px^2 (b) Rot-Grün-Überlagerung von R und T_{176}^ϕ mit CG, Startwert zu 129^2px^2 (c) Rot-Grün-Überlagerung von R und T_{222}^ϕ mit CG, Startwert zu 257^2px^2 (d) Rot-Grün-Überlagerung von R und T_{277}^ϕ mit CG, Startwert zu 513^2px^2



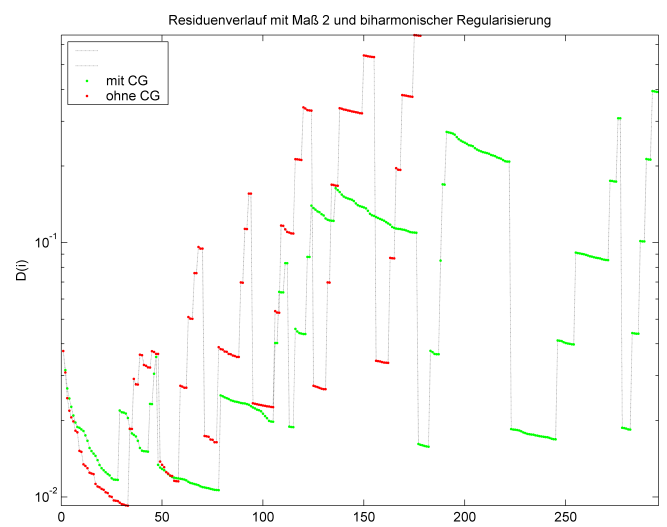
(e) Rot-Grün-Überlagerung von R und Ergebnis T_{296}^ϕ mit Maß \tilde{d} mit CG-Verfahren



(f) Koordinatentransformation ϕ zu T_{296}^ϕ



(g) Rot-Grün-Überlagerung von R und Ergebnis T_{178}^ϕ mit Maß \tilde{d} ohne CG-Verfahren



(h) Residuenverlauf

Abbildung 7.5: Gegeben sind R und T aus Abb. 7.1 auf Seite 58. Diesmal wird Maß \tilde{d} (vgl. 2.2 auf Seite 12) verwendet. Alle weiteren Parameter sind die gleichen wie bei Abbildung 7.2 auf Seite 59. Die Registrierung wurde auf einer Auflösung von $17 \times 17 \text{px}^2$ begonnen und schrittweise verdoppelt bis zur vollen Auflösung. Die Rechenzeit betrug 2 Minuten.

Eine Sache die beim Vergleich der Ergebnisse zu den drei Operatoren auffällt ist die Anzahl der Iterationen. Diese ist beim harmonischen Operator mit insgesamt 1046 inneren Iterationen am größten und nimmt über den sesquiharmonischen (820) bis zum biharmonischen Operator (557) ab. Beim triharmonischen Operator erfolgen sehr schnell Nullschritte, so dass die Iterationszahl noch niedriger ist. Diese Beobachtung lässt sich folgendermaßen erklären. Passt ein großer Bereich von $T \circ \phi_i$ schon zu R , so treten dort keine Kräfte f auf. An Stellen, an denen die Bilder noch abweichen treten nicht-triviale Kräfte auf. Bei einer lokal wirkenden Regularisierung verschiebt die regularisierte Suchrichtung im wesentlichen das Bild nur in letztgenanntem Bereich. Bei strengerer Regularisierung treten auch Verschiebungsvektoren im Bereich der Bildstellen auf, die bereits passen. Die Liniensuche findet dann das Optimum zwischen der Verbesserung in dem einen und der Verschlechterung in dem anderen Bereich. Diese Zwietracht lässt sich unmittelbar an Schrittweitenwahlen, die weniger als 0.1 Pixel verschieben, ablesen. Die Mindestabstiegsforderung ist damit recht bald unterschritten und die Glättungsstufe wird verlassen. Dazu kommt auch noch die Tatsache, dass die Testbilder mit einer Verzerrung versehen sind, die keine allzu lokalen Verzerrungen beinhaltet.

Bei der monomodalen Bildanpassung mit

$$d_{\text{mono}}(R, T \circ \phi; \vec{x}) = R(\vec{x}) - (T \circ \phi)(\vec{x})$$

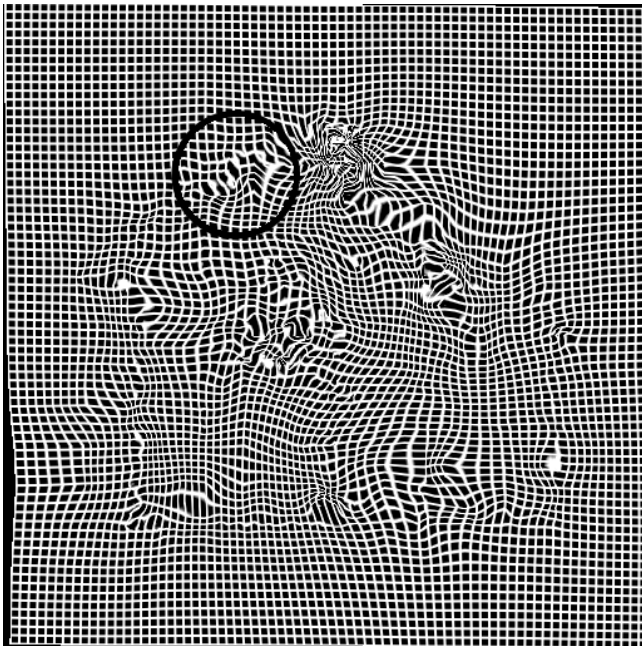
kommen dagegen im Abstandsfunktional

$$\mathcal{D}_{R,T}^{\text{mono}}(\phi) = \int_{\Omega} d_{\text{mono}}^2(R, T \circ \phi; \vec{x}) d\vec{x} = \int_{\Omega} \left(R(\vec{x}) - T(\phi(\vec{x})) \right)^t \cdot \left(R(\vec{x}) - T(\phi(\vec{x})) \right) d\vec{x}$$

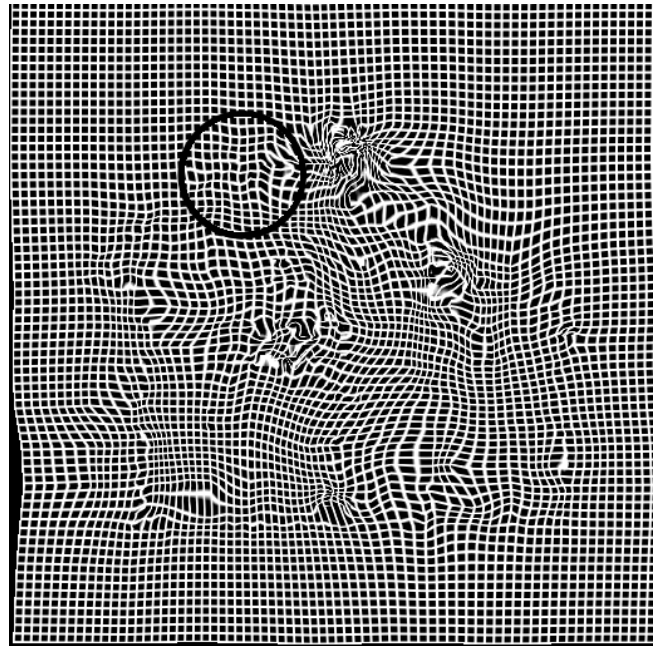
keine Ableitungen der Transformationsabbildung vor. In den Kräften \vec{f}_{mono} kommt nur die erste Ableitung von ϕ vor. Demzufolge reicht dann die einfach-harmonische Regularisierung vollkommen aus. Die biharmonische kann verwendet werden, wenn a priori feststeht, dass die zu erwartende Transformation extrem glatt ist.

Die für die morphologische multimodale Bildanpassung nötige Regularisierung schränkt die möglichen Transformationen damit ein.

7.1.1 Elastische Verformung



(a) harmonischer Operator mit Dirichlet Randbedingung



(b) Navier-Lamé mit periodischer Randbedingung

Abbildung 7.6: Vergleich der harmonischen und der elastischen Regularisierung. Letztere ist der Navier-Lamé Operator mit $\beta = 2$. Der wesentliche Unterschied lässt sich im eingekreisten Bereich erkennen. Im harmonischen Fall laufen ursprünglich senkrechte Gitterlinien stellenweise parallel. Der Navier-Lamé Operator vermeidet dies. Referenz und Templatebild sind die gleichen wie in Abb. 7.1 auf Seite 58. Aufgrund des hinreichend breiten Randes stört die periodische Randbedingung nicht.

Im vorangegangenen Abschnitt wurde festgestellt, dass mindestens die biharmonische Regularisierung notwendig ist, um keine artifiziellen Deformationen zu erhalten. Dies kann je nach Bilddaten auch zu streng sein. Z.B. muss bei der Anpassung eines Patientengehirns an einen Atlas stärker verzerrt werden, als bei der Anpassung eines CT- an ein MR-Bild vom gleichen Individuum.

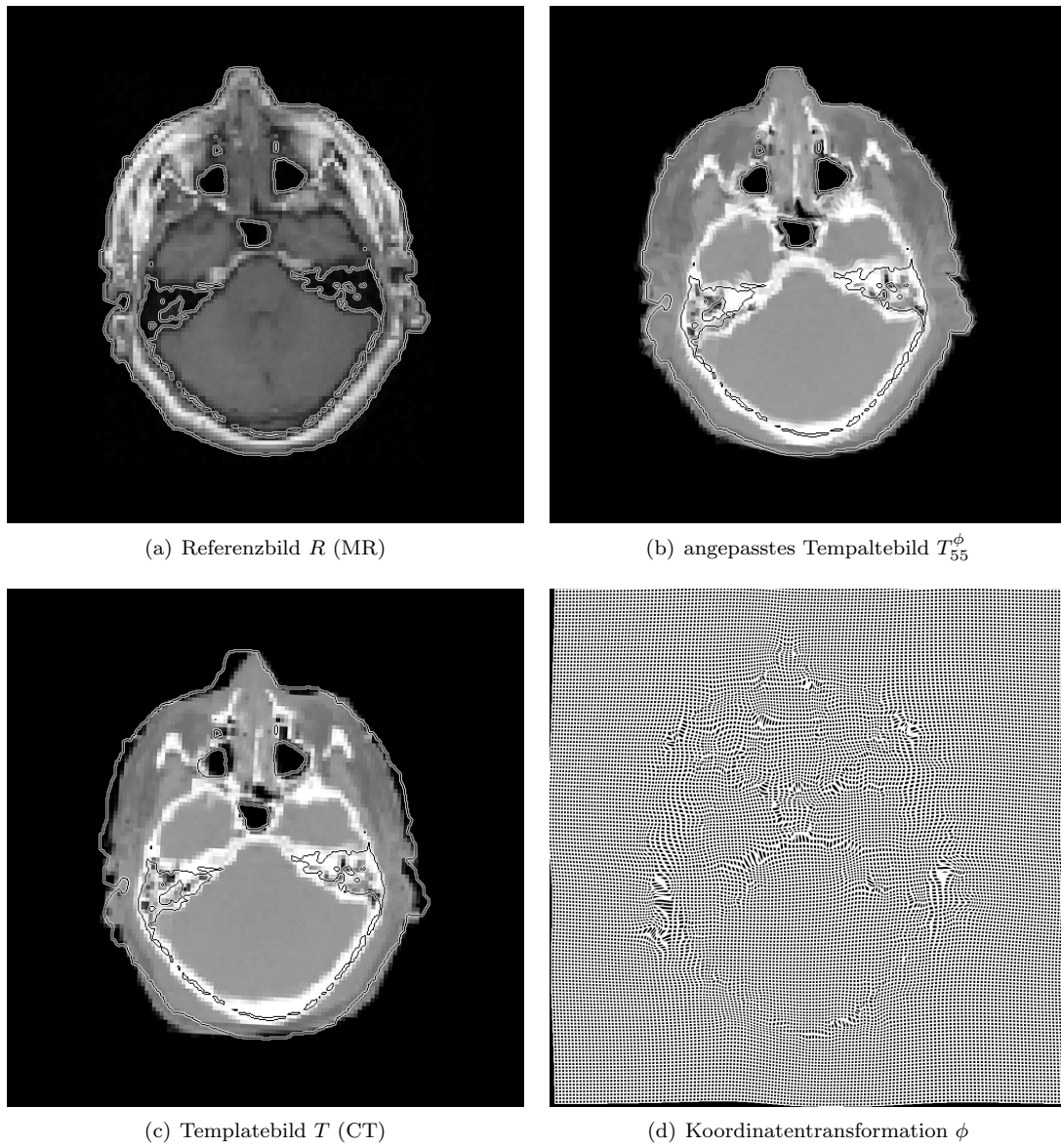


Abbildung 7.7: Das CT-Bild wird an das MR-Bild des gleichen Gehirns angepasst. Als Orientierungshilfe für den Leser ist hier die Kontour des MR-Bildes in allen Bildern eingefügt. Die Regularisierung erfolgt nach dem elastischen Modell des Navier-Lamé Operators mit $\beta = 2$.

Um den Anpassungsvorgang stabil zu halten wird das Modell der elastischen Verformung, also der Navier-Lamé Operator ausprobiert. Dieser enthält im Gegensatz zum harmonischen Operator auch die gemischten Ableitungen und tendiert daher zum Volumenerhalt. Das ist genau der Effekt der bei der Deformation von Körpern in der Physik auftritt. Für eine effiziente Rechnung wird die periodische Randbedingung gewählt. Diese ist zwar eher unpassend zur Aufgabe, hat aber den Vorteil, dass die Lösung mit FFT erfolgen kann. Die genaue Rechnung befindet sich im Abschnitt „Unterprogramme“ auf Seite 28. Ist an den Bildrändern genug Rand vorhanden, so stellt die periodische Randbedingung kein Problem dar.

In Abbildung 7.6 auf Seite 62 ist die Transformation des harmonischen und des elastischen Operators zu sehen. Dabei fällt auf, dass beim harmonischen Operator an einigen Stellen die ursprünglich senkrechten Gitterlinien kurzzeitig parallel verlaufen. Dies vermeidet der elastische Operator aufgrund der gemischten Ableitung. Eine solche Stelle ist eingekreist.

Auch beim Navier-Lamé Operator tauchen artifizielle Deformationen auf. Diese verringern sich, wenn die Eigenwerte hoch $\frac{3}{2}$ genommen werden, was der Stärke des sesqui-harmonischen Operators entspricht und verschwinden bei quadratischen Eigenwerten, was bis auf die Koppelung der Richtung dem biharmonischen Operator gleich kommt. Die gemischte Ableitung im Navier-Lamé vermeidet sehr stark, einzelne Strukturen zusammen zu schieben bis sie unsichtbar werden. Dies ist wieder die Eigenschaft des Volumenerhaltes. Sie kann aber auch von Nachteil sein. Bei dem Beispiel mit dem Modellhäuschen muss z.B. im Bereich des Erdgeschossfensters die Fläche verkleinert werden, damit es zum Referenzbild passt. An dieser Stelle gibt es mit der Navier-Lamé-Regularisierung Schwierigkeiten.

Zusammenfassend lässt sich sagen, dass der Navier-Lamé Operator bei stark abweichenden Bildern eine gute Wahl ist um eine stabile Bildanpassung durchzuführen. Mit quadratischen Eigenwerten ist der Navier-Lamé Operator aufgrund seiner Strenge dagegen nur bei sehr ähnlichen Bildern einsetzbar.

7.1.2 Der Einfluss der Randbedingung

Die (bi-) harmonische Randwertaufgabe kann mit Dirichlet oder Neumann Randbedingung gewählt werden. Die beiden Randtypen bedingen, dass nicht alle affin-linearen Transformationen möglich sind. Diese Eigenschaft wird – wie bereits erwähnt – durch Zerlegung der $\vec{f} = \vec{f}_l + \vec{f}_c$ in den affin-linearen Anteil und das orthogonale Komplement, ausgeglichen. Beim Lösen der jeweiligen Randwertaufgabe mit der Sinus- bzw. Cosinustransformation wird im Frequenzraum durch die Eigenwerte dividiert. Dies bedeutet eine Abdämpfung der hohen Frequenzen. Damit wird der Einfluss der Frequenzanteile mit zunehmender Frequenz geringer.

Die niedrigste Frequenz der Sinustransformation ist $\varphi_1(x) = \sin\left(\frac{\pi \cdot x}{n+1}\right)$. Die Dirichlet Randbedingung lässt also starke Verschiebungen des Mittelbereiches im Bilde zu. Um Verschiebungen im Randbereich zu erzeugen bedarf es dort entsprechend starker Kräfte. Diese nehmen dann indirekt wieder Einfluss auf die Intensität der niedrigsten Frequenz, da durch die Dämpfung der hohen Frequenzen die niedrige Frequenz verhältnismäßig stärker wird. Somit versuchen Anpassungen im Randbereich die Bildmitte negativ zu beeinflussen.

Bei der Cosinustransformation ist die niedrigste nicht konstante Frequenz $\psi_2(x) = \cos\left(\frac{\pi \cdot x}{n-1}\right)$. Die Neumann Randbedingung kann randnahe Verschiebungen somit nachbilden. Diese Frequenz führt bezogen auf den Bildmittelpunkt antisymmetrische Verschiebungen durch. D.h. wird die Linke Seite des Bildes nach unten verschoben, so verschiebt sich die rechte Seite des Bildes tendenziell nach oben. Die nächste Frequenz $\psi_3(x) = \cos\left(\frac{2 \cdot \pi \cdot x}{n-1}\right)$ ist für symmetrische Verschiebungen zuständig. Dabei wird der Mittelbereich des Bildes gegenüber der Randzone entgegengesetzt verschoben.

Nun muss beachtet werden, dass die Frequenzen ν mit zunehmender Höhe sehr schnell abgedämpft werden. Beim biharmonischen Operator ist die Dämpfung proportional zu $\frac{1}{\nu^4}$. Dadurch ist in Rechenbeispielen oft zu beobachten, dass die Randzonen des Templatebildes in aufeinanderfolgenden Iterationen in der niedrigen Frequenz hin- und her schwingt. Da sich die Verschiebung innerhalb einer Iteration höchstens um ein Pixel ändert und der Abstand \mathcal{D} dabei streng monoton fällt, gibt es auch noch höherfrequente Verschiebungsanteile. Diese sind Verhältnismäßig kleiner, so dass sie dem menschlichen Auge nicht unmittelbar auffallen.

Es sei daran erinnert, dass auf die Abdämpfung der höheren Frequenzen im Kraftfeld \vec{f}_c nicht verzichtet werden kann, da dies den Verzicht auf das Regularisieren bedeutet. In Abbildung 6.5 auf Seite 50 (b) ist gezeigt, dass die Topologie des Bildes bei Verzicht auf die Regularisierung verloren geht. Wird hingegen erst ab der k -ten Frequenz mit der Dämpfung begonnen, so resultieren periodische Muster in \vec{u}_c .

Es besteht die Möglichkeit, den Eigenwerten der biharmonischen Randwertaufgabe vor der Division eine Konstante $c \approx 1$ hinzu zu addieren. Dies entspricht einer Verschiebung der Dämpfungsfunktion im Frequenzraum. Dadurch wird die Abdämpfungsfunktion in den ersten Frequenzen flacher. Praktische Ergebnisse zeigen in der Tat eine etwas großzügigere Verformung. Allerdings werden bei $c > 1$ zunehmend die von der einfach-harmonischen Regularisierung bekannten Deformationen sichtbar.

Es lässt sich kaum eine allgemein gültige Aussage treffen, wann welche Randbedingung besser geeignet ist. Ergibt sich aus der Bildanpassungsaufgabe, dass die Randzonen von R und T gut passen, dann empfiehlt sich die Dirichlet Randbedingung. Bei Verwendung der Neumann Randbedingung ist zu Beachten, dass der Rand echt konstant gefärbt sein muss – sofern nicht mit Neumann-Randbedingung geglättet wird – damit die in Kapitel 4 auf Seite 19 vorgenommene Vereinfachung der Randbedingungsgleichung erlaubt ist. Ist die Normalenableitung auf dem Rand nicht Null, so sind die Ergebnisse in der Tat am Rand schlecht angepasst. Ohne diese Vereinfachung lässt sich die RWA nicht mehr mit der Cosinustransformation lösen, stattdessen sind Mehrgitterverfahren notwendig. Da kein hinreichend effizient implementiertes Mehrgitterverfahren zur Verfügung stand, wurde diese Variante nicht getestet.

Des Weiteren besteht die Möglichkeit die Bilder mit zusätzlichem Rand aus der Hintergrundfarbe zu umgeben, so dass die Auswirkung der Randbedingung herabgesetzt wird. Diese Methode vergrößert allerdings die Bilddatenmenge. Ihr Einsatz empfiehlt sich daher nur auf geringer Auflösungsstufe.

7.2 Regularisierung in der Liniensuche

Bisher wurde im Algorithmus das Verschiebungsfeld der Suchrichtung regularisiert. Die jeweilige Suchrichtung ist damit hinreichend glatt, aber die Summe der Suchschritte kann recht unglatt werden, wenn in vielen Iterationen das Verschiebungsfeld in einem Bereich ähnlich ausfällt. Aus diesem Grund scheint es sinnvoll das Straffunktional auch bei der Bestimmung der Schrittweite in der Liniensuche zu verwenden. Der erste Ansatz ist die Liniensuche von

$$\rho^{\text{opt}} := \operatorname{argmin}_{0 \leq \rho \leq \rho_0} \mathcal{D}_{R,T}(\phi(\vec{u} + \rho \cdot \vec{w}))$$

auf

$$\begin{aligned} \rho^{\text{opt}} &:= \operatorname{argmin}_{0 \leq \rho \leq \rho_0} \mathcal{D}_{R,T}(\phi(\vec{u} + \rho \cdot \vec{w})) + \alpha \cdot S(\rho \cdot \vec{w}) \\ &= \operatorname{argmin}_{0 \leq \rho \leq \rho_0} \mathcal{D}_{R,T}(\phi(\vec{u} + \rho \cdot \vec{w})) + \rho^2 \cdot (\alpha \cdot S(\vec{w})) \end{aligned}$$

abzuändern. Das ist nicht sonderlich sinnvoll, da die Größe des aktuellen Suchrichtungsverschiebungsfeldes bereits durch die Obergrenze für ρ , so dass $\|\rho_0 \cdot \vec{w}\|_\infty = 1$, hinreichend klein gehalten wird.

Im zweiten Ansatz wird das resultierende Verschiebungsfeld mit dem Straffunktional gemessen:

$$\rho^{\text{opt}} := \operatorname{argmin}_{0 \leq \rho \leq \rho_0} \mathcal{D}_{R,T}(\phi(\vec{u} + \rho \cdot \vec{w})) + \alpha \cdot S(\vec{u} + \rho \cdot \vec{w})$$

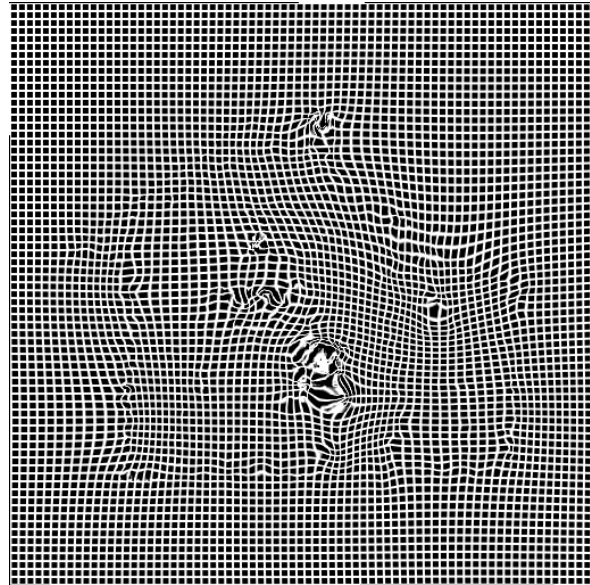
Bei dieser Variante ist S_2 und S_3 viel zu streng wie in Abbildung 4.1 auf Seite 20 zu erkennen ist. Daher wird hier nur S_1 verwendet.

$$\begin{aligned} S_1(\vec{u} + \rho \cdot \vec{w}) &= \int_{\Omega} \sum_{k=1}^n \|\nabla(u_k + \rho \cdot w_k)\|_2^2 d\vec{x} \\ &= \int_{\Omega} \sum_{k=1}^n (\nabla(u_k + \rho \cdot w_k))^t \cdot (\nabla(u_k + \rho \cdot w_k)) d\vec{x} \\ &= \int_{\Omega} \sum_{k=1}^n (\nabla u_k + \rho \cdot \nabla w_k)^t \cdot (\nabla u_k + \rho \cdot \nabla w_k) d\vec{x} \\ &= \int_{\Omega} \sum_{k=1}^n (\nabla u_k)^t \cdot (\nabla u_k) + \rho \cdot 2 \cdot (\nabla u_k)^t \cdot (\nabla w_k) + \rho^2 \cdot (\nabla w_k)^t \cdot (\nabla w_k) d\vec{x} \\ &= \int_{\Omega} \sum_{k=1}^n (\nabla u_k)^t \cdot (\nabla u_k) d\vec{x} + \rho \cdot 2 \cdot \int_{\Omega} \sum_{k=1}^n (\nabla u_k)^t \cdot (\nabla w_k) d\vec{x} + \rho^2 \cdot \int_{\Omega} \sum_{k=1}^n (\nabla w_k)^t \cdot (\nabla w_k) d\vec{x} \\ &= \underbrace{\int_{\Omega} \sum_{k=1}^n \|\nabla u_k\|_2^2 d\vec{x}}_{=:c} + \rho \cdot 2 \cdot \underbrace{\int_{\Omega} \sum_{k=1}^n (\nabla u_k)^t \cdot (\nabla w_k) d\vec{x}}_{=:b} + \rho^2 \cdot \underbrace{\int_{\Omega} \sum_{k=1}^n \|\nabla w_k\|_2^2 d\vec{x}}_{=:a} \\ &= a \cdot \rho^2 + b \cdot \rho + c =: p(\rho) \end{aligned}$$

Für jede Suchrichtung sind einmalig die Werte a , b und c zu berechnen. Bei den Schrittweitentests ist dann nur noch das quadratische Polynom $p(\rho)$ auszuwerten. Es kann $a > 0$ angenommen werden, da $a = 0$ bedeutet, dass alle Kräfte $f = 0$ sind. Damit handelt es sich bei p um eine nach oben geöffnete Parabel. Diese hat ihr Minimum in $\rho^{\text{Sopt}} = \frac{-b}{2 \cdot a}$. Damit ist ρ^{Sopt} nur dann größer Null, wenn die ∇w_k in genügend vielen Punkten $\vec{x} \in \Omega$ hinreichend antiparallel zu ∇u_k sind. Dieser Fall bedeutet anschaulich, dass das aktuelle Suchrichtungsverschiebungsfeld \vec{w} versucht die bisherige Verzerrung aus \vec{u} rückgängig zu machen. Der Algorithmus der Bildanpassung ist dagegen so aufgebaut, dass die Schritte so vorsichtig gemacht werden, dass im großen keine Verschiebung rückgängig gemacht werden braucht. Da



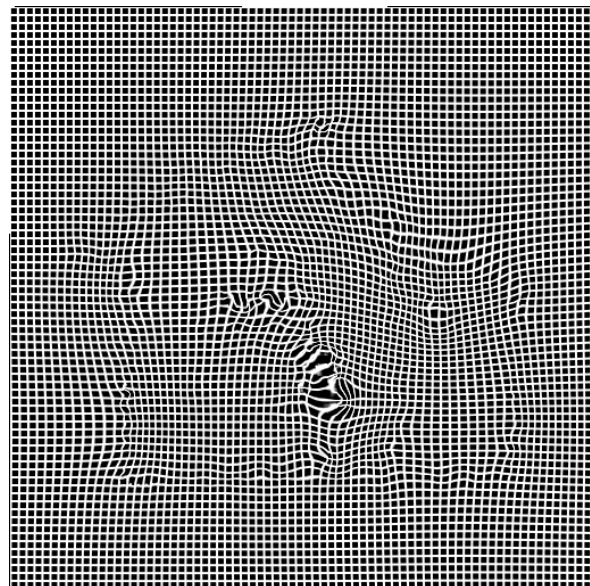
(a) Ergebnis T_{1045}^ϕ mit harmonischer Regularisierung bei Suchrichtung und Schrittweite mit CG



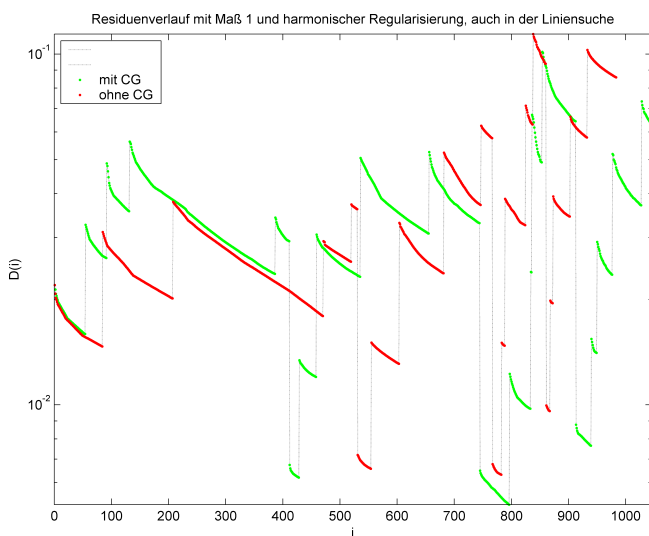
(b) Koordinatentransformation ϕ zu T_{1045}^ϕ



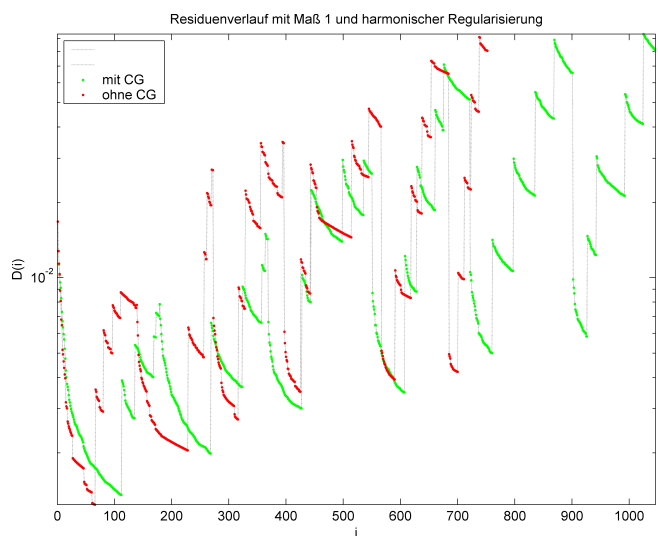
(c) Ergebnis T_{983}^ϕ mit harmonischer Regularisierung bei Suchrichtung und Schrittweite ohne CG



(d) Koordinatentransformation ϕ zu T_{983}^ϕ



(e) Residuenverlauf mit $\alpha = 0.001$



(f) Residuenverlauf bei $\alpha = 0$

Abbildung 7.8: Mit harmonischer Regularisierung in der Schrichtung und $\alpha = 0.001$ in der Liniensuche. Die Residuen $D(i) := \sqrt{\mathcal{D}(\phi(\vec{u}^i))}$ reduzieren sich dabei deutlich langsamer als mit $\alpha = 0$.

die Schrittweite nach unten durch Null beschränkt ist, lässt sich sagen, dass der Strafterm S_1 versucht, nach einer gewissen Verschiebung \vec{u} die Schrittweite ρ gegen Null zu ziehen.

Es wurden Beispiele mit dem Strafterm S_1 in der Liniensuche und der harmonischen Regularisierung durchgeführt. Als Startwert für den Algorithmus wird das triviale Verschiebungsfeld $\vec{u}_0 = 0$ verwendet. Es zeigt sich, dass in den ersten Iterationen der Strafterm so klein ist, dass die Schrittweiten fast ausschließlich vom Abstandsfunktional \mathcal{D} bestimmt werden. Nach einigen Iterationen liegt eine gewisse Verzerrung in \vec{u} vor. Dann erreicht der Term c im Straffunktional eine gewisse Größe und die Schrittweite wird zunehmend vom Straffunktional S bestimmt. Innerhalb weniger Iterationen überwiegt der Strafterm bei der Liniensuche sehr stark und die Schrittweiten werden sehr klein. Oft ist $\rho^{\text{opt}} \approx 0.001 \cdot \rho_0$. Das führt dann wiederum dazu, dass die Mindestabstiegsforderung $\varepsilon > 0$ nicht mehr erfüllt ist und die Glättungsstufe wird verlassen. Bei großzügigeren Mindestabstiegsforderungen erfolgen viele Kurzschritte. Aufgrund der Kurzschritte ändert sich die Kraft in der nächsten Iteration nur minimal und die Suchrichtung ist fast identisch.

Die endgültige Verzerrung enthält zwar weniger aber immer noch zu viele artifizielle Deformationen gegenüber $\alpha = 0$. Das Problem ist, dass einzelne starke Verzerrungen in S_1 trotz der quadratischen Aufsummierung aufgrund der großen Anzahl an Punkten nicht so stark ins Gewicht fallen. Andererseits ist die Bestrafung bei globaleren Verformungen wiederum zu streng, da sich das Abstandsfunktional mit zunehmender Anpassung aufgrund der bereits recht parallelen Kanten nicht sehr stark reduziert. Beim Modellproblem wird daher das Erdgeschossfenster nicht weit genug verschoben. Die genaue Wahl von α bestimmt dabei, ab welcher Iteration die Schrittweite gedämpft wird. Das bedeutet in der Praxis, dass mit geringem α die artifiziellen Deformationen zu stark sind und bei größerem α wird das Bild zu wenig angepasst.

7.3 Vergleich der beiden Maße

Bei der Einführung der Abstandsmaße d und \tilde{d} wurden bereits einige Merkmale aufgezählt, die sich mathematisch unmittelbar aus der Definition der Maße ableiten. In diesem Abschnitt wird untersucht, in wie weit sich diese Eigenschaften in der Praxis bemerkbar machen, vor allem wo sie problematisch werden.

In Abbildung 7.2 auf Seite 59 und 7.5 auf Seite 61 ist die gleiche Aufgabe einmal mit Maß d (siehe Seite 8) und einmal mit dem auf Seite 12 eingeführten Maß \tilde{d} durchgerechnet. In diesen Beispielen eignen sich beide Maße gleichermaßen gut. Bis auf kleine Ungenauigkeiten stimmen beide Bilder mit dem Referenzbild überein.

Nun werden die Farben im Referenzbild R geändert. Maß d liefert immer noch ein gutes Ergebnis. Siehe dazu in Abbildung 7.4 auf der nächsten Seite. Die gleiche Aufgabe mit Maß \tilde{d} läuft völlig daneben. Daher wird hier nur kurz beschrieben, was in etwa geschieht: Der Umriss zwischen dem schwarzen Hintergrund und dem Objekt ist das einzige, was auf dem richtigen Platz landet. Die weißen Tafeln mit den Nummern werden auf den weißen Bereich links neben der Haustür verschoben. Die übrigen Bereiche werden mitgezerrt.

Die geringen Farbabweichungen durch die Detailreduzierung in R (Abb. 7.5 auf Seite 61) stört bei \tilde{d} nicht so stark. Aber die Umfärbung eines Bildes wie es in Abbildung 7.4 vollzogen ist, macht Maß \tilde{d} unbrauchbar.

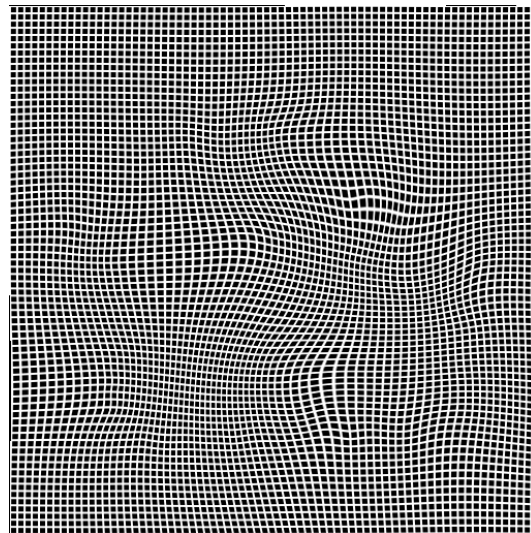
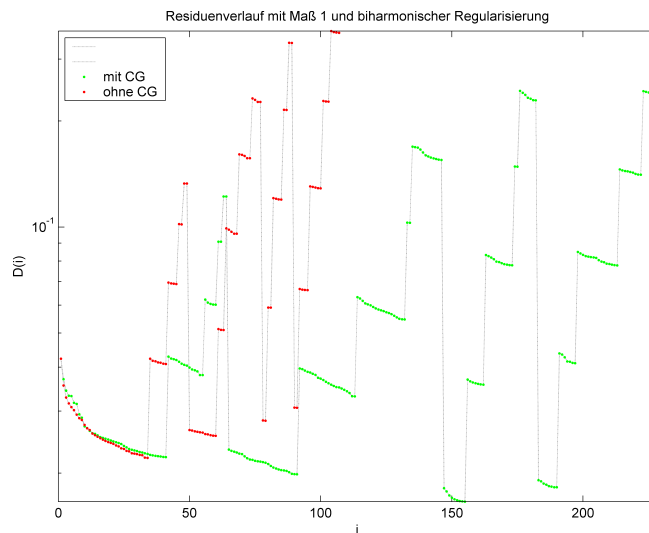
So gut wie in Beispiel 7.4 funktioniert die Bildanpassung auch mit Maß d nicht immer. Der Grund dafür lässt sich beim Vergleich der Gradienten-Bilder $\|\nabla R\|_2$ und $\|\nabla T\|_2$ erkennen. In Abbildung 7.4 auf Seite 70 sind drei Fotos der gleichen Landschaft aufgenommen – eines bei Sonnenschein, das zweite bei bewölktem Himmel und das letzte nachdem es geschneit hatte. Diese Fotos passen bis auf die paar Autos die zwischenzeitlich verändert wurden nach minimaler Verzerrung aufeinander. Trotzdem zeigt sich die Bildanpassung dabei sehr schwierig. Der Grund besteht darin, dass der Farbunterschied an Kanten zu wesentlichen Objektteilen, wie die Umrisse der Häuser aufgrund des geringen Farbunterschiedes sehr schwach ausgeprägt sind. Dagegen sind Kanten von Details, wie z.B. die Fenster verhältnismäßig stark. Die Grob-Anpassung ist damit schlecht möglich, da weder d noch \tilde{d} die Kanten sicher zuordnen können. Diese Problematik wird im nachfolgenden Abschnitt im Zusammenhang mit der Glättung noch ausführlicher besprochen.

Es lässt sich zusammenfassen, dass Maß \tilde{d} nur bei Bildern mit ähnlichen Helligkeitsverhältnissen ein brauchbares Ergebnis liefert. Maß d benötigt dagegen nur einen klaren Verlauf der Höhenlinien zu den wesentlichen Objektkanten.

7.4 Glättung und abweichende Farben

In früheren Kapiteln wurde für die Glättung die Wärmeleitungsgleichung mit Dirichlet und Neumann Randbedingung, sowie die Gaußglättung vorgestellt.

Die Wärmeleitungsgleichung mit Dirichlet Randbedingung hat immer dunklen Rand, da der erste außerhalb liegende Bildpunkt durch die Randbedingung immer schwarz ist. Das führt bei starker Glättung dazu, dass jedes Bild in der Mitte hell ist und zum Rand hin dunkler wird. Bei extrem starker Glättung sind damit alle Bilder bis auf Skalierung

(a) Referenzbild R (b) Templatebild T (c) registriertes Templatebild T_{228}^{ϕ} mit biharmonischer Regularisierung und CG-Verfahren(d) Koordinatentransformation ϕ zu T_{228}^{ϕ} 

(e) Residuenverlauf

Abbildung 7.9: Eine multimodale Registrierung. Gegeben sind die Bilder R (a) und T (b) mit einer Auflösung von $513 \times 513 \text{px}^2$. Begonnen wird mit $65 \times 65 \text{px}^2$ und dann wird die Auflösung schrittweise verdoppelt. Die Tür, das Erdgeschossfenster, die Nummernschilder, das Schuppendach und der Schornstein sind an die richtige Position geschoben worden. Lediglich die Brettterritzen vom Schuppen und das Kellerfenster sind noch an verkehrter Stelle. Die Koordinatentransformation ϕ ist glatt. Die Rechenzeit beträgt 3 Minuten. Bei der gleichen Rechnung ohne CG-Verfahren wird nur die Abweichung an der linken Hausecke angepasst und das übrige T^{ϕ} sieht dem T ähnlicher als dem R . Das gleiche Beispiel funktioniert auf Grund der stark verschiedenen Farbdifferenzen mit Maß \tilde{d} überhaupt nicht.

der absoluten Helligkeit gleich. Da der Rand der dirichlet-geglätteten Bilder zwangsläufig ähnlich ist, macht diese Form der Glättung nur Sinn, wenn der Rand der Bilder bereits richtig registriert ist, denn randnah treten auch bei verschiedenen Bildern nur minimale Kräfte auf.

Die Wärmeleitungsgleichung mit der Neumann Randbedingung nimmt an, dass sich die Bilder außerhalb zum Rand gespiegelt fortsetzen. Dadurch wird das geglättete Bild im Randbereich in der Normalenrichtung etwa konstant und die Randzone erzeugt zum Rand parallel gerichtete Verschiebungskräfte deren Größe im wesentlichen von dem Inhalt der Bildpunkte abhängt und nicht vom Randabstand.

Die Gaußglättung ist eher der Neumann als der Dirichlet Randbedingung ähnlich. Wird z.B. beim Fullweighting

$$\frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

im Randpunkt der Punkt, der über die Bildfläche Ω hinausgeht umgeklappt, wie z.B. für den linken Rand

$$\frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 4 & 2 \\ 0 & 1 & 0 \end{bmatrix}$$

dann entspricht das dem impliziten symmetrischen Fortsetzen bei der Neumann Randbedingung aus der Wärmeleitungsgleichung.

Bei multimodalen Bildern kann zu viel Glättung schaden.

Bei starker Glättung schafft es die Bildregistrierung, Strukturen, die mehrere Pixel voneinander abweichen, in Deckung zu bringen. In sofern sollte die Registrierung mit entsprechend starker Glättung immer zum richtigen Ergebnis finden. Dies ist tatsächlich nicht der Fall. Es hat sich gezeigt, dass dies nur gilt, wenn die Gradientenstrukturen und Intensitäten von Referenzbild und Templatebild ähnlich sind. In Abbildung 7.4 auf der nächsten Seite ist dazu ein Beispiel. Die drei Fotos wurden vom annähernd gleichen Ort zu verschiedenen Zeiten fotografiert. Einmal mit Sonnenschein, dann bei bewölktem Himmel und schließlich nachdem es geschneit hatte. Die Norm der Gradienten ist jeweils rechts daneben abgebildet, wobei dunkle Farbe großen Gradienten und weiß kleinen Gradienten entspricht. Obwohl die Fotos bereits fast deckungsgleich sind, ist die Bildanpassung in der Praxis äußerst schwierig. Die Tatsache, dass sich die Lage einiger Autos auf diesen Bildern verändert hat, ist hier nicht das Problem, denn diese Stellen können mit einer Gewichtungsmaske (vgl. 7.5 auf Seite 72) aus dem Bildanpassungsvorgang ausgeblendet werden.

Die Schwierigkeit besteht darin, dass die Kanten zu den wesentlichen Strukturen wie den Umrissen der Gebäude teilweise sehr schwach sind und beim Glätten untergehen. Die Gradienten von Details, wie Fenstern sind dagegen viel stärker ausgeprägt. Eine robuste Grobanpassung ist deshalb nicht möglich. Lassen sich zwei Bilder aufeinander registrieren, so funktioniert der Vorgang gleichermaßen gut, wenn die Farben in einem der beiden Bilder invertiert sind. Bei Einbettung in ein größeres Bild ist darauf zu achten, dass die Hintergrundfarbe ebenfalls invertiert wird. Andernfalls laufen die Bilder beim Glätten unterschiedlich weit nach außen, so dass die Kante am Bildrahmen von R und T unterschiedlich breit wird und eine genaue Lokalisierung der Randregion unmöglich ist.

Bild 7.4 (a) und 7.4 (b) lassen sich registrieren, wenn die Glättung so gering eingestellt ist, dass sie nur maximal drei Nachbarpixel breit wirkt und die Auflösung bei $\frac{1}{32}$ begonnen wird.

Das verschneite Bild auf eines der zuvor genannten zu registrieren gelingt nur durch Hinzunahme von weichen Kontrollpunkten oder einer Transformation der Gradienten, wie sie im nächsten Absatz vorgestellt wird. Die Kontrollpunkte werden entlang des Dachfirsts und der Regenrinne der zentralen Häuserreihe gesetzt. Andernfalls versucht das Programm die Dachrinne des Schneebedeckten Hauses auf den Dachfirst im anderen Bild zu setzen, da dort jeweils eine deutliche Kante verläuft.

In einigen Fällen von abweichenden Helligkeitsdifferenzen hilft es, die Helligkeit der Bilder zu transformieren. Das kann durch eine Histogrammequilisierung geschehen oder durch die Gamma-Transformation mit geeignetem $\Gamma > 0$, d.h. $R_{\text{angepasst}}(\vec{x}) := R(\vec{x})^\Gamma \forall \vec{x} \in \Omega$. Sind die Kanten der Hauptorientierungsmerkmale zu schwach ausgeprägt, so hilft es meist, die Länge der Gradienten auf Eins zu normieren.

$$\nabla R_{\text{angepasst}} := \frac{\nabla R}{\varepsilon + \|\nabla R\|_2^q}$$

Das $0 < \varepsilon \approx 0.001$ ist ein Regularisierungsparameter, der die Abbildung in Null stetig macht. Mit $0 \leq q \leq 1$ kann gesteuert werden, wie stark die Länge der Gradienten normiert ist. Die Wirkung wird in Abb. 7.11 gezeigt. Bei $q = 1$ geht die Information aus der Gradientenlänge vollständig verloren, es bleibt nur noch die Richtung. Bei $q = 0$ ist die Transformation bis auf das ε die Identität. Die Transformation wird analog für T bzw. $T \circ \phi$ durchgeführt. Ist q ungefähr 1 gewählt, so werden die Gradienten von Bildstörungen wie z.B. Rauschen zu stark. Es hat sich in der Praxis



(a) mit Sonnenschein



(b) Gradientennorm



(c) bei bewölktem Himmel



(d) Gradientennorm



(e) mit Schnee bedeckt



(f) Gradientennorm

Abbildung 7.10: In der Linken Spalte sind drei Fotos des im wesentlichen gleichen Objektes mit unterschiedlichen Beleuchtungseigenschaften. Rechts daneben ist die euklidische Norm des Gradienten aufgezeichnet. An weißen Stellen ist der Gradient Null. Je dunkler ein Bildpunkt ist, umso größer ist dort die Norm. Die Stärke der Gradienten an einigen Objekten ändert sich recht stark zwischen den drei Bildern. So verläuft eine stark ausgeprägte Kante entlang des Dachfirsts in (a) und (b). In (c) hingegen ist die Kante entlang der Regenrinne am stärksten. In Bild (a) sind die Umrisse der meisten Fenster stärker als in (b) oder (c).

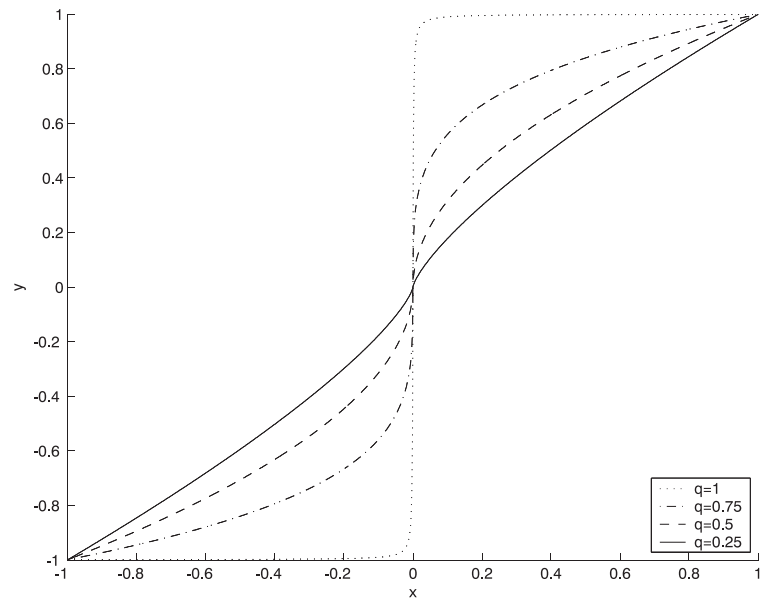


Abbildung 7.11: Gradiententransformation. x entspricht $\|\nabla R\|_2$ und y entspricht $\|\nabla R_{\text{angepasst}}\|_2$.



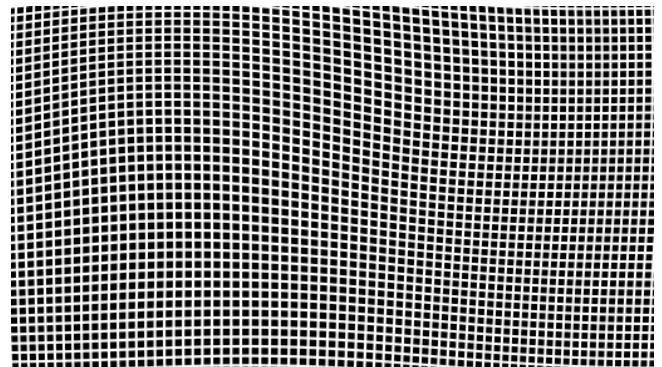
(a) Referenzbild R



(b) Templatebild T



(c) Ergebnis T_{65}^{ϕ}



(d) Koordinatentransformation ϕ

Abbildung 7.12: Ein Beispiel mit normiertem Gradient ($q = 0.9$). Des Weiteren ist die biharmonische Regularisierung mit Dirichlet Randbedingung und Maß d verwendet. Außerdem wird die in Abbildung 7.14 auf der nächsten Seite gezeigte Gewichtungsmaske eingesetzt, um die Kante rings herum am Bildrand und die Autos am unteren Bildrand für das Programm unsichtbar zu machen. Die Bilder sind aus dem auf $513 \times 513\text{px}^2$ vergrößerten Bild ausgeschnitten. Im wesentlichen ist T^{ϕ} richtig zurecht gebogen. Aufgrund der Gradientennormierung sind einige Kanten, wie z.B. der Dachfirst, bis zu 5 px voneinander entfernt.

ein Wert von $q \approx 0.5$ bewährt. Gegebenenfalls muss ein Schwellenwert für die Normierung eingeführt werden, so dass Gradienten mit geringer Länge klein bleiben, oder sogar auf 0 reduziert werden.

Bei nicht-normierten Gradienten kann der sich über mehrere Pixel ausbreitende Anstieg der Kante helfen, die Kanten von T auf die von R zu zentrieren. Bei der Normierung der Gradienten sind die Kanten dagegen sehr steil. Deshalb muss die Glättung besonders vorsichtig zurückgefahren werden, um leicht verschobene Kanten nach und nach genauer zu lokalisieren. Bei zu schnellem zurücknehmen der Glättung passiert es leicht, dass die Kanten in der nächstniedrigeren Glättung zu weit nebeneinander liegen, so dass sie im Maß nicht mehr aufeinander wirken.

Es lässt sich zusammenfassen, dass eine robuste Bildanpassung ähnlich stark ausgebildete Kanten zu den wesentlichen Objekten im Referenz- und Templatebild erfordert. Das besagt wiederum, dass zwar nicht die Farben selbst, aber die Helligkeitsdifferenzen bei den wesentlichen Strukturen ähnlich sein müssen. Je stärker diese Bedingung erfüllt ist, desto robuster wird die Bildanpassung durch erhöhen der Ausgangsglättung. Andernfalls muss das Optimum einer geringeren Glättung abhängig von den Bilddaten ermittelt werden. Gegebenenfalls hilft auch die Normierung der Gradienten.

7.5 Die Verwendung einer Gewichtungsmaske

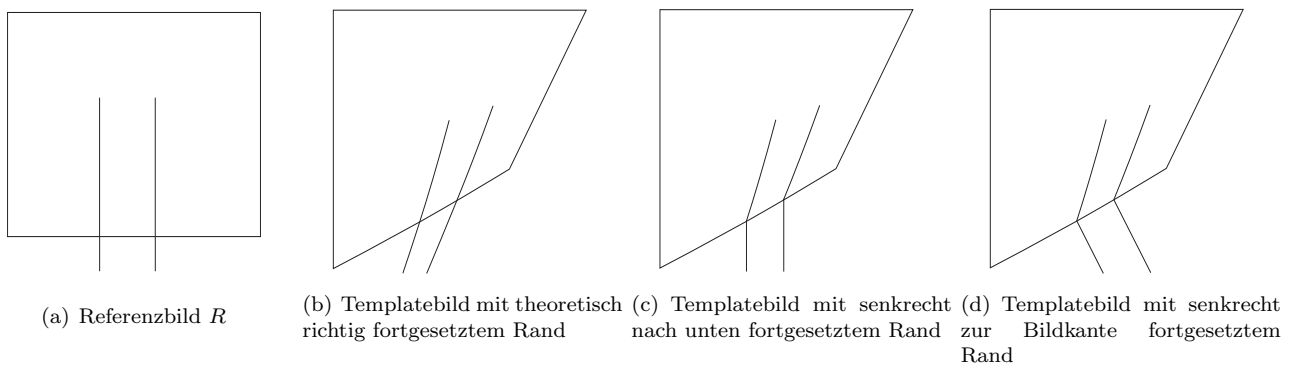


Abbildung 7.13: Verschiedene Möglichkeiten, den Rand der Bilder zu ergänzen. Das Rechteck in (a) soll das Referenzbild darstellen. Dieses enthält zwei Linien, die am unteren Rand fortgesetzt sind. In (b) bis (d) ist das perspektivisch verzerrte Templatebild mit verschiedenen Möglichkeiten, den Rand fort zu setzen. In (b) ist die Version, die bei Umkehrung der Verzerrung (a) ergibt. Diese ist in der Praxis erst nach erfolgreicher Bildregistrierung bekannt. In (c) und (d) sind zwei Varianten, die sich ohne Kenntnis der Transformationsabbildung realisieren lassen.



Abbildung 7.14: Das ist die Gewichtungsmaske für Beispiel 7.4 auf Seite 70 und 7.12. Die Kante am Rand der Fotos und die Autos im unteren Bildbereich werden damit für das Bildanpassungsprogramm unsichtbar. Der schwarze Bereich wird dann ausschließlich durch die Regularisierung überbrückt.

Es gibt zwei Fälle, in denen es sinnvoll ist, in bestimmten Regionen den Einfluss der Kräfte \vec{f} und des Abstandes d abzugewichten. Der eine sind Regionen in Bildern, die in R und T unterschiedliche Objekte beinhalten und die nur Anhand der Umgebung registriert werden sollen. Der andere Fall sind die Bildränder, die bei dem abrupten Wechsel auf die Hintergrundfarbe eine starke Kante erzeugen, wie dies in Beispiel 7.4 auf Seite 70 im Bereich des Himmels zur Hintergrundfarbe Schwarz der Fall ist. Da beide Bilder echt unterschiedlich groß sind und die Ränder auch nicht genau aufeinander gehören, ist es unerlässlich auf die Einbettung in ein größeres Bild mit konstanter Hintergrundfarbe zu verzichten. In Abbildung 7.5 sind Versuche unternommen, den Rand des ursprünglichen Bildes nach der Einbettung

in das größere Bild zu wiederholen. Dabei ist zu sehen, dass alle praktisch durchführbaren Methoden aufgrund des entstehenden Knickes mit der Regularisierung unverträglich sind.

Die Gewichtungsmaske ist ein Bild M aus Werten in $[0, 1]$. Mit den Punktwerten dieses Bildes wird die Berechnung des Abstandes \mathcal{D}

$$\mathcal{D}_{R,T}(\phi) := \int_{\Omega} M(\vec{x}) \cdot d^2(R, T \circ \phi; \vec{x}) \, d\vec{x}$$

und der Kräfte \vec{f}

$$(\vec{f}(\vec{x}))_i := M(\vec{x}) \cdot (\vec{f}_{\Omega}(\vec{x}))_i \quad \forall \vec{x} \in \Omega \text{ und } i = 1, \dots, n$$

gewichtet. Damit ist es möglich, ausgewählte Bereiche für das Bildanpassungsprogramm unsichtbar (schwarz) zu machen oder geringer zu gewichten (grau). Ein komplett weißes Gewichtungsbild ist hierbei das neutrale Element.

7.6 Der Einsatz des CG-Verfahrens

Mit dem nicht-linearen CG-Verfahren konnte in der Praxis im allgemeinen eine bessere Bildanpassung erzielt werden. Die Geschwindigkeit erhöht sich dadurch nicht. Im Gegenteil, meist sind es insgesamt mehr Iterationen. Das ist auf die Forderung des Mindestabstieges zurück zu führen. Ohne das CG-Verfahren werden die einzelnen Glättungs- bzw. Auflösungsstufen aufgrund zu geringer Verbesserung früher beendet, als mit den konjugierten Gradienten. Der Parameter γ liegt dabei meist in der Größenordnung $0.1, \dots, 1$.

In Abbildung 7.15 auf der nächsten Seite sind die Residuen

$$D(i) := \sqrt{\mathcal{D}(\phi(\vec{u}^i))}$$

im Iterationsverlauf mit und ohne CG-Verfahren bei verschiedenen Regularisierungsstärken, Randbedingungen und den Maßen d (Maß 1) und \tilde{d} (Maß 2) eingezeichnet. Abbildung 7.2 auf Seite 59 ist die CG-Variante aus (a), Abbildung 7.4 auf Seite 60 aus (b), Abbildung 7.3 auf Seite 60 aus (c) und Abbildung 7.5 auf Seite 61 aus (d). Alle nicht genannten Parameter sind jeweils identisch.

An den Aufwärtssprüngen findet eine Glättungsreduzierung statt. Die Abwärtssprünge bei den Auflösungswechseln kommen durch die wieder erhöhte Glättung zustande. Es fällt auf, dass die Iterationsanzahl ohne CG-Verfahren deutlich kleiner ausfällt und der endgültige Abstand größer ist. Dieser Effekt nimmt bei strengeren Regularisierungen (sesqui-, bi-harmonisch) zu. Anfangs ist der Bildabstand zwar ohne CG-Verfahren geringer, beim Betrachten des Anpassungsablaufes sieht das Bild jedoch mit CG-Verfahren richtiger aus. Bei (d) ist das Erdgeschossfenster z.B. ohne CG-Verfahren zu weit rechts und mit CG-Verfahren an der richtigen Stelle. Die Zwischenergebnisse und die Iterationsverläufe auch zu den anderen 6 Kombinationen sind auf der CD (siehe Anhang B auf Seite 83) zu finden.

Auffallend ist auch, dass die Residuen ohne CG-Verfahren anfangs schneller abnehmen und das hin und wieder in einem größeren Sprung. Beim Betrachten der Zwischenergebnisse sieht es danach aus, dass bestimmte Stellen vorrangig eingepasst werden. Beim Modellproblem wird die linke Hausecke recht schnell an ihre Position geschoben. Das Erdgeschossfenster hingegen bleibt auf halben Wege hängen. Sobald die Glättung zu gering ist, wird die Zuordnung vom Abstandsmaß nicht mehr gesehen und dementsprechend bleiben die notwendigen Kräfte aus.

Das CG-Verfahren sorgt mit dem hinzu addieren der gewichteten vorangegangenen Suchrichtung dafür, dass einmal in Bewegung gesetzte Bildbereiche in nachfolgenden Iterationen weiter rücken, bis die Kräfte in die entgegengesetzte Richtung zeigen. Denn der Gewichtungsfaktor γ wird aus dem ganzen Bild ermittelt und ist in der Praxis in der Nähe von Eins. Das Einpassen der linken Hausecke im Beispiel sorgt dann dafür, dass auch das Erdgeschossfenster weiter nach links rutscht. Zu dem Zeitpunkt an dem die linke Hausecke ihr Ziel erreicht hat, ist das Erdgeschossfenster schon so dicht an seinem Platz angekommen, dass die Kräfte auch bei geringer Glättung die richtige Richtung vorgeben.

Problematisch wird der Einsatz des CG-Verfahrens genau dann, wenn es keine deutlich erkennbare Grobstruktur in den Bildern gibt. In diesem Fall wirkt sich eine einmalige Verwechslung einer Kante auf die nachfolgenden Iterationen aus, so dass eine oder mehrere Bildstellen ihren Abstand zur Idealposition über einige Iteration verschlechtern, anstatt zu verbessern. Dieses Phänomen trat z.B. bei den Bildern aus Abbildung 7.4 auf Seite 70 auf, sofern die Gradientenlänge nicht normiert wurde. In diesen Fällen ist es angebracht die Bildanpassung ohne konjugierte Gradienten durchzuführen.

7.7 Die Auswirkung von Bildstörungen

Bildstörungen sind zum einen das Rauschen und zum anderen Objekte, die es nur in einem der beiden Bilder gibt. Das Rauschen stört erst, wenn die Kanten der wesentlichen Objekte zu schwach werden. Dann besteht die Gefahr, dass vom Rauschen herrührende Kanten registriert werden. In Abbildung 7.16 auf Seite 75 ist der Übersichtlichkeit halber lediglich ein Kreis auf ein Quadrat angepasst. Das Referenzbild wird mit verschieden starkem gaußschen und Impulsrauschen („Salt & Pepper“) gestört. Die Anpassung der groben Form stellt dabei kein Problem dar. Treten

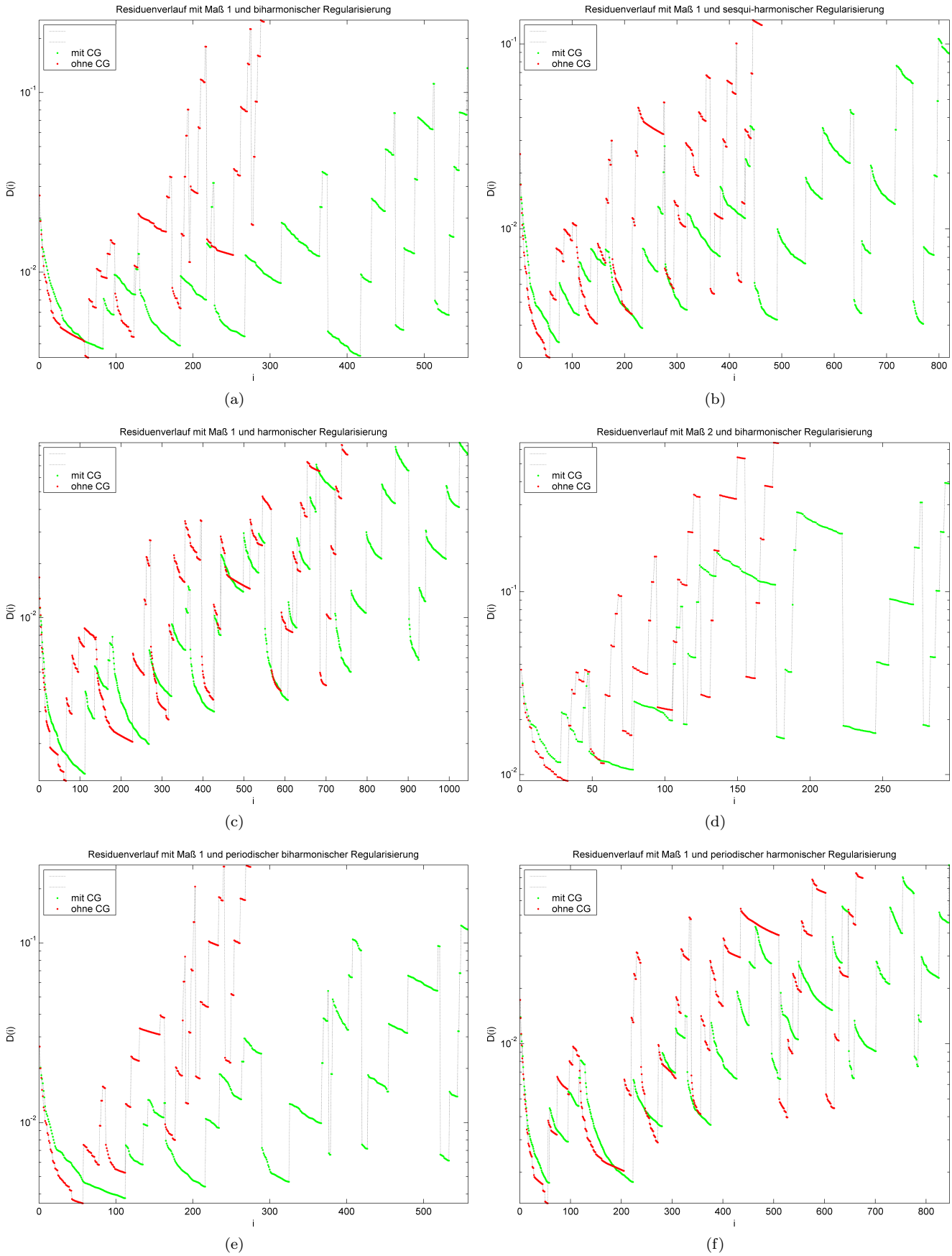
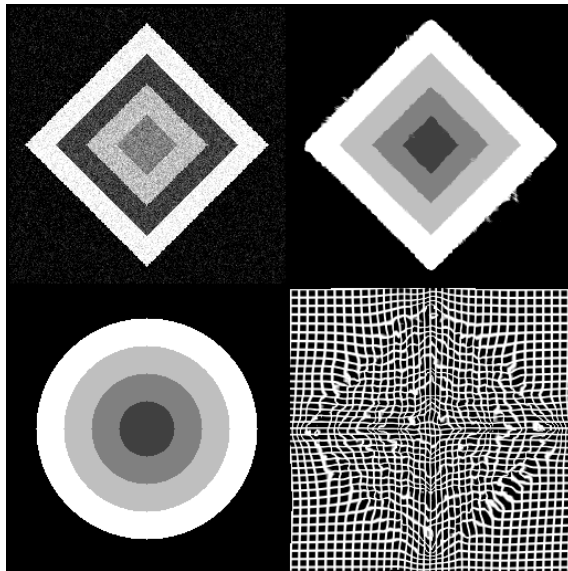
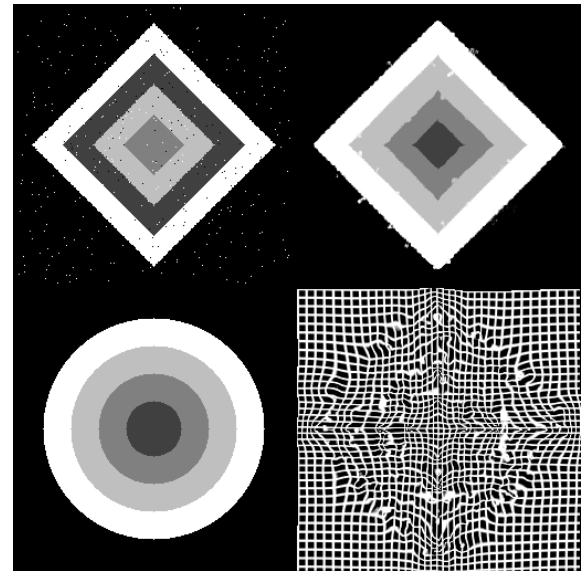
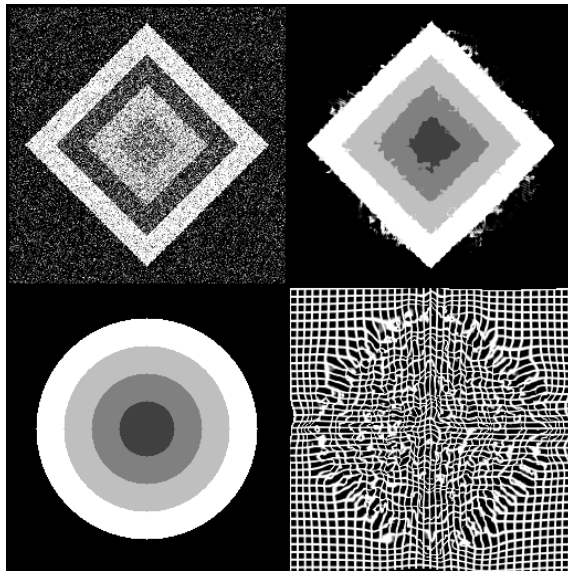
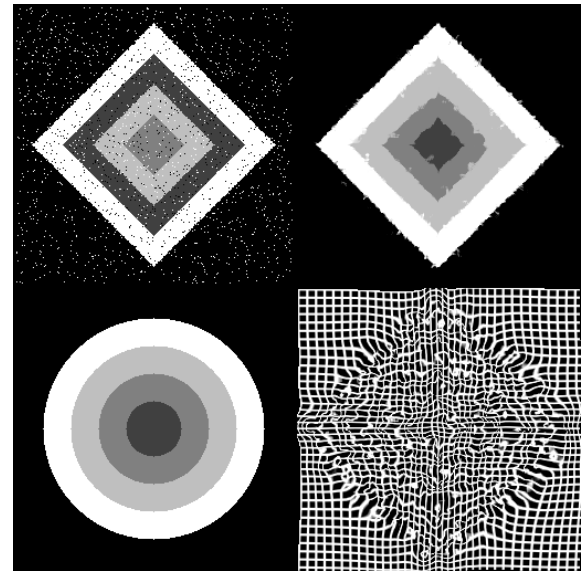


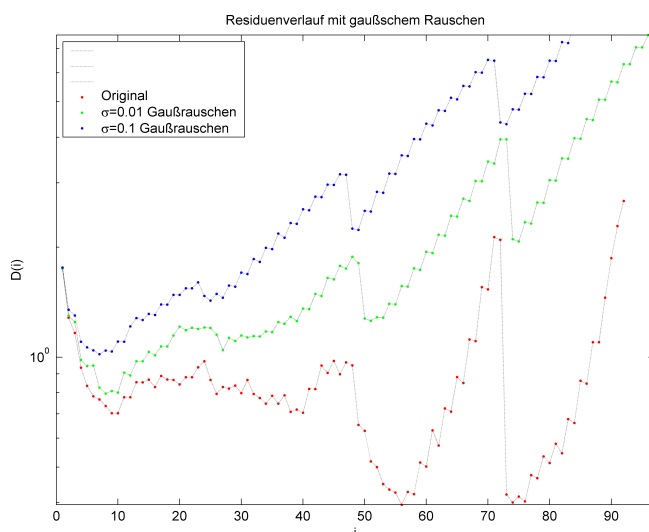
Abbildung 7.15: Die Residuen $D(i) := \sqrt{\mathcal{D}(\phi(\vec{u}^i))}$ im Iterationsverlauf mit und ohne CG-Verfahren bei verschiedenen Regularisierungsstärken, Randbedingungen und den Maßen d (Maß 1) und \tilde{d} (Maß 2). Zu beachten ist, dass die D -Achse logarithmisch skaliert ist. Bei linearer Skalierung ist die Abflachung der Teilkurven noch deutlicher zu erkennen. Weitere Informationen befinden sich im Text.

(a) gaußsches Rauschen mit $\sigma = 0.01$ 

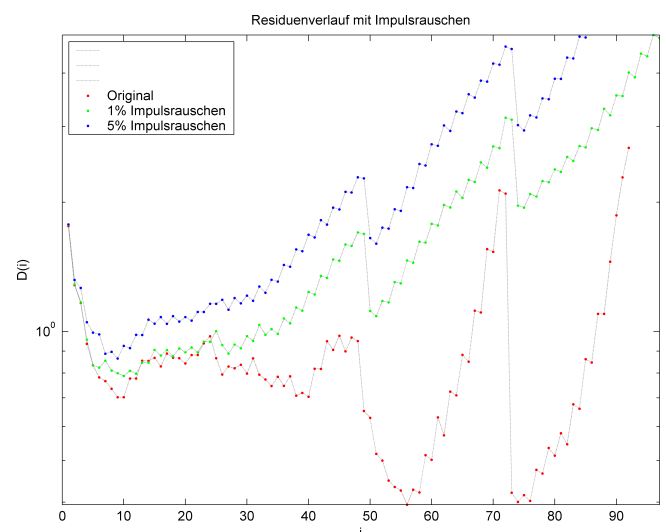
(b) mit 1% Impulsrauschen

(c) gaußsches Rauschen mit $\sigma = 0.1$ 

(d) mit 5% Impulsrauschen



(e) Residuenverlauf zum gaußschen Rauschen



(f) Residuenverlauf zum Impulsrauschen

Abbildung 7.16: Die Auswirkung von Bildrauschen auf den Anpassungsvorgang mit Maß d , Glättung mit der Wärmeleitungsgleichung, Regularisierung mit Navier-Lamé und CG-Verfahren. Das Referenzbild ist jeweils oben und das Templatebild unten links. Rechts daneben befindet sich das Ergebnis der Anpassung. Die Bildanpassung klappt im wesentlichen auch bei verrauschtem Referenzbild. Allerdings werden auch durch Rauschen entstandene Kanten nach geformt.

farbähnliche Rausch-Pixel dicht beieinander auf, so bilden sie eine Kante, auf die das Templatebild eingeht. Bei ähnlicheren Bildern lässt sich eine strengere Regularisierung wählen, die solch starke Deformation nicht zulässt.

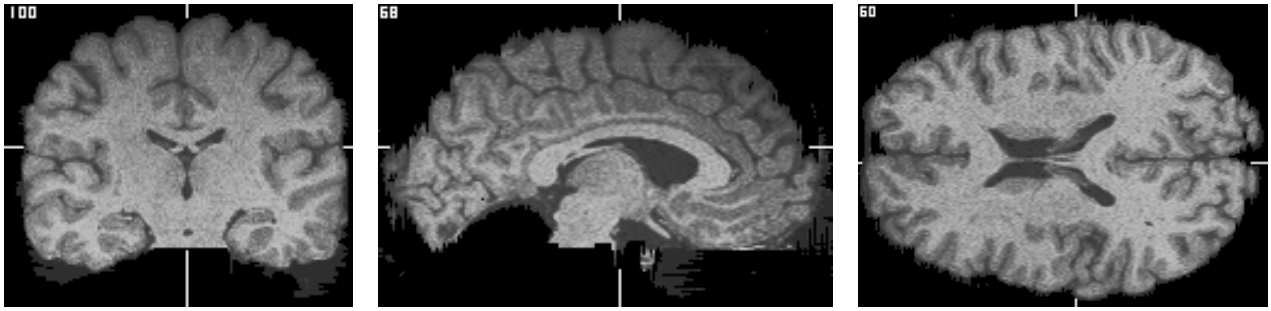
Objekte, die es nur in einem der beiden Bilder gibt, können zu Verwechslung führen, wenn sie nach der Glättung mit Nachbarobjekten verschmelzen. In diesem Fall ist der Einsatz von Kontrollpunkten sinnvoll. Liegt das Objekt isoliert im Bild, kann auch das Ausblenden dieser Stelle mit einer Gewichtungsmaske helfen.

7.8 Der Einsatz von Kontrollpunkten

Bei stark verzerrten Bildern eignet sich ein Satz \mathcal{P} an Kontrollpunkten – sofern sie richtig gesetzt sind – als Startlösung. Auch der Einsatz der schwachen Kontrollpunkte eignet sich sowohl als Fortsetzung, als auch alleine. Die Platzierung der gewählten Punkte sollte sich halbwegs gleichmäßig über die Bildfläche verteilen, damit die Regularisierung keine große Fehlverzerrung in entfernten Bereichen verursacht.

Der Einsatz der festen Kontrollpunkte hat sich in der Praxis nicht bewährt. Bei der Rechnung erfolgen ausnahmslos Nullschritte. Die Regularisierung bewirkt immer – wenn auch nur leicht – globale Verschiebungen. Das Festhalten der Kontrollpunkte wird realisiert, indem ein Korrektur-Verschiebungsfeld \vec{v} dem aus den regularisierten Kräften \vec{u} hinzu addiert wird, so dass $\vec{u} + \vec{v}$ in den Kontrollpunkten verschwindet. Das Suchrichtungs-Verschiebungsfeld muss in den Kontrollpunkten verschwinden, da sich diese Punkte aufgrund der Verwendung der Kontrollpunkte als Startlösung bereits auf dem richtigen Platz befinden. Nun beinhaltet die Berechnung von \vec{v} ebenfalls die Regularisierung. Dadurch überlagert \vec{v} das \vec{u} auch außerhalb der Kontrollpunkte so sehr, dass $\vec{u} + \vec{v}$ im allgemeinen keine Abstiegsrichtung ist. Das strenge Festhalten der gewählten Punkte verträgt sich also nicht mit der Regularisierung.

Schnittebenen von einer Kernspinaufnahme eines menschlichen Gehirns:

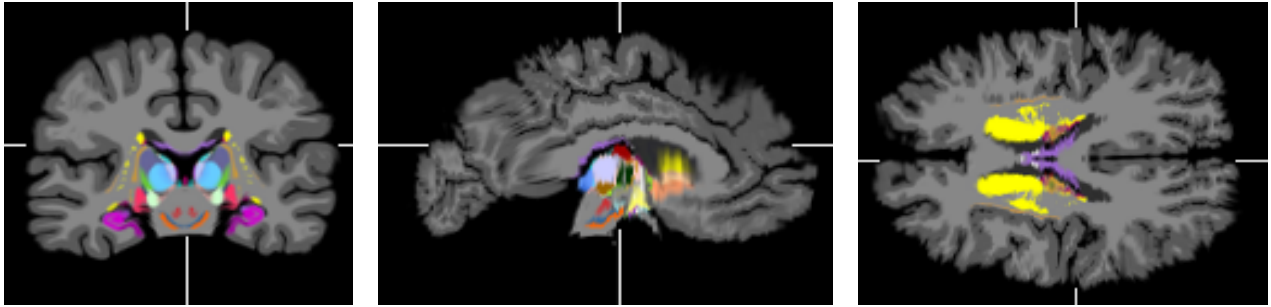


(a) MR coronal

(b) MR sagittal

(c) MR horizontal

„Atlas of the Human Brain“ sehr starr transformiert, um die zugehörigen Ebenen zu erhalten:

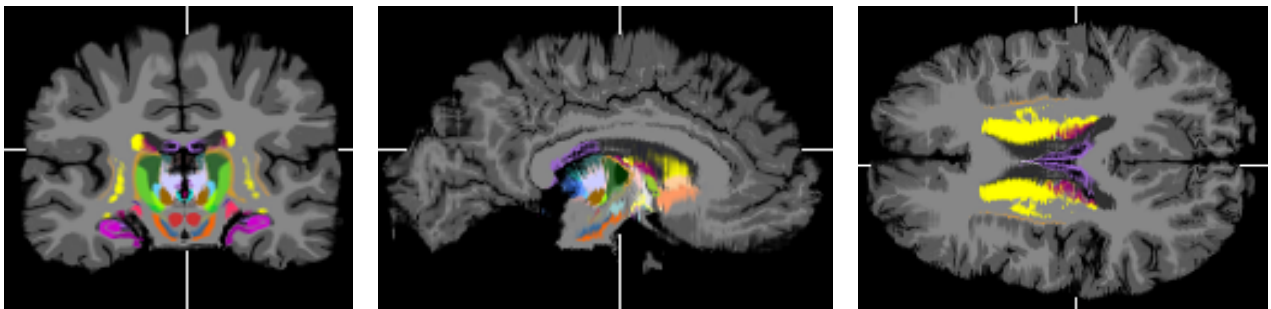


(d) Atlas coronal

(e) Atlas sagittal

(f) Atlas horizontal

Mit elastischer Regularisierung angepasster Atlas:

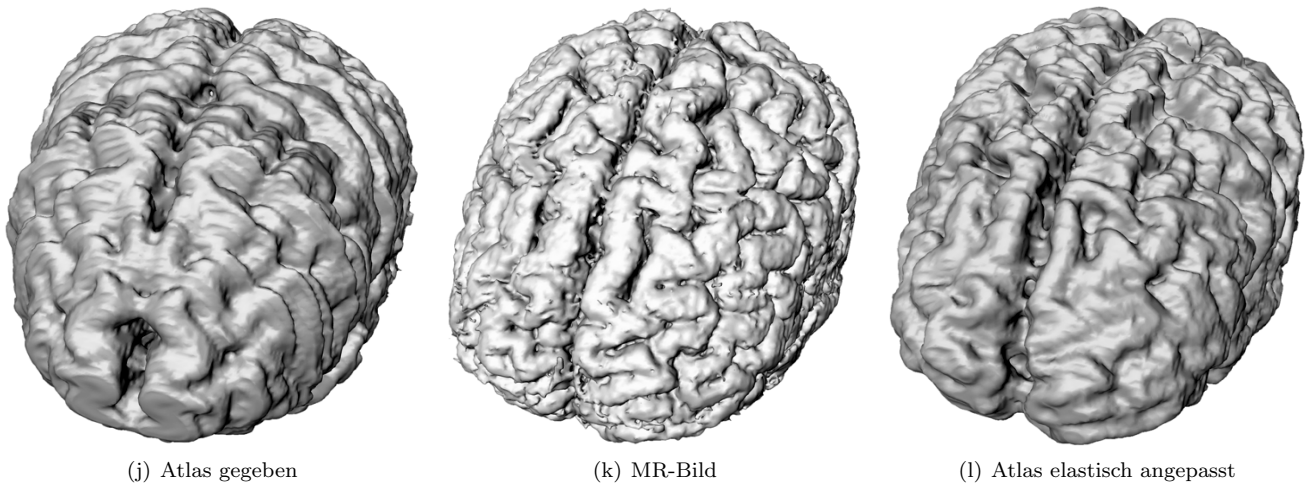


(g) Atlas coronal

(h) Atlas sagittal

(i) Atlas horizontal

Rekonstruktion der Oberfläche:



(j) Atlas gegeben

(k) MR-Bild

(l) Atlas elastisch angepasst

Abbildung 7.17: Eine Graustufenvariante des „Atlas of the Human Brain“ [10] wurde auf das Magnetresonanzbild eines fremden Gehirns (erste Zeile) angepasst. Die so erhaltene Transformation ϕ wurde auf die drei Farbkanäle des gefärbten Atlases angewendet (zweite und dritte Zeile). Auf diese Weise ist es möglich, anhand der Farben die Funktion vom jeweiligen Hirnareal des Patienten (erste Zeile) in der Legende des Atlases nach zu schlagen. In der zweiten Zeile ist das Ergebnis mit biharmonischer Regularisierung. In Zeile drei wurde der Navier-Lamé Operator mit $\beta = 2$ eingesetzt. Dieser liefert hier das beste Ergebnis. Es sind jeweils drei Schnittebenen aus verschiedenen Raumrichtungen abgebildet. Die weißen Striche markieren die Stelle, an der die beiden anderen Ebenen durchstoßen. Die Bildauflösung ist aus coronaler Sicht $142 \times 128 \times 189$ Voxel (Breite \times Höhe \times Tiefe). In der vierten Zeile ist die Oberfläche rekonstruiert. Die Strukturen vom angepassten Atlas haben dabei deutliche Ähnlichkeit zum MR-Bild.

8 Schlussfolgerung

In dieser Arbeit wurde eine Möglichkeit vorgestellt, zwei Bilder auf einander anzupassen, bei denen nur die Gestalt der Objekte ähnlich ist, die Farben selbst jedoch i.a. sehr verschieden sind. Die Aufgabe wurde als Minimierungsproblem formuliert. Dabei ist die Nebenbedingung, dass der Zusammenhalt, also die Topologie des Bildes, erhalten bleibt.

Dazu wurden zwei Abstandsmaße vorgestellt, die nur die Informationen aus den Gradienten verwenden. Es stellte sich heraus, dass das eine Maß bei unterschiedlichen Farbdifferenzen zwischen dem Referenzbild und dem Templatebild keine zufrieden stellenden Ergebnisse bringt. Das andere Maß zeigt keine Farbabhängigkeiten.

Die Nebenbedingung wurde durch Messen der Ableitung der Transformationsabbildung als Strafterm eingeführt. Zu dem so entstandenen Funktional wurde die zugehörige Randwertaufgabe für das Minimum hergeleitet. Diese hochgradig nicht-lineare Randwertaufgabe wurde dann iterativ gelöst, was wiederum einem Abstiegsverfahren entspricht. Es wurde die harmonische und die biharmonische Randwertaufgabe sowie der Navier-Lamé Operator untersucht. Dabei stellte sich heraus, dass erstere lokale artifizielle Verzerrungen nicht ausreichend bestraft, die zweite gute Ergebnisse bringt, wenn die Bilder nicht zu lokal verzerrt sind und letzterer recht stabil ist.

Der Eigenschaft, dass die Bildanpassung ein extrem schlecht gestelltes Problem ist, wurde entgegengewirkt, indem die Bilder anfangs stark geglättet werden und zum anderen mit der strengen Regularisierung der Suchrichtung mittels der biharmonischen Randwertaufgabe oder dem Volumen erhaltenden Navier-Lamé Operator. Um die Performance des Programms zu erhöhen wurde die Möglichkeit der Mehrgitteriteration vorgestellt. Eine Grobanpassung ist bereits auf niedrigen Auflösungen möglich. Auf den höheren Auflösungen brauchen dann nur noch Kleinigkeiten nachkorrigiert zu werden. Eine weitere Beschleunigung und bessere Ergebnisse konnten mit den konjugierten Gradienten erzielt werden.

Es wurden verschiedene Möglichkeiten vorgestellt, wie mit Bildstörungen und schlecht erkennbaren Kanten dennoch eine Anpassung gelingt. Objekte die keine Entsprechung im anderen Bild finden können mit einer Gewichtungsmaske ausgeblendet werden. Schlecht erkennbare Kanten können durch Transformation der Gradientenlänge hervorgehoben werden. Notfalls können Kontrollpunkte von Hand hinzugefügt werden.

Es hat sich gezeigt, dass die morphologische multimodale Bildanpassung im Prinzip möglich ist, aber die Wahl der optimalen Parameter lässt sich nur grob voraussagen und muss gegebenenfalls empirisch ermittelt werden. Weiterhin ist erforderlich, dass die notwendige Transformation mit der gewählten Regularisierung verträglich ist. Bei zu gering gewählter Regularisierung treten artifizielle Deformationen auf und bei zu strenger Wahl können stark lokale Anpassungen nicht vorgenommen werden. Im Prinzip lassen sich auch höherdimensionale Bilder aufeinander anpassen. Allerdings stoßen heutige Computer bereits mit dreidimensionalen Bilddaten recht schnell an ihre Kapazitätsgrenze.

A Die Randwertaufgabe zum Navier-Lamé Operator

Die Richtungsableitung des Navier-Lamé Operators

$$S_3(\vec{u}) = \frac{1}{2} \int_{\Omega} \sum_{i=1}^n \left(\Delta u_i(\vec{x}) + \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i(\vec{x}) \right)^2 d\vec{x} \quad (\text{Navier-Lamé})$$

in Richtung von $\vec{h} \in H_2(\Omega)^n$ ist

$$\begin{aligned} \frac{\partial}{\partial \vec{u}} S_3(\vec{u}) \vec{h} &= \lim_{\varepsilon \rightarrow 0} \frac{S_3(\vec{u} + \varepsilon \cdot \vec{h}) - S_3(\vec{u})}{\varepsilon} \\ &= \frac{1}{2} \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \sum_{i=1}^n \left(\Delta(u_i + \varepsilon \cdot h_i) + \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} (u_i + \varepsilon \cdot h_i) \right)^2 - \left(\Delta u_i + \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right)^2 d\vec{x} \\ &= \frac{1}{2} \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \sum_{i=1}^n (\Delta(u_i + \varepsilon \cdot h_i))^2 + 2 \cdot \frac{\beta}{2} \cdot \Delta(u_i + \varepsilon \cdot h_i) \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} (u_i + \varepsilon \cdot h_i) \\ &\quad + \left(\frac{\beta}{2} \right)^2 \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} (u_i + \varepsilon \cdot h_i) \right)^2 - \left(\Delta u_i + \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right)^2 d\vec{x} \\ &= \frac{1}{2} \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \sum_{i=1}^n (\Delta u_i)^2 + 2 \cdot \varepsilon \cdot \Delta u_i \cdot \Delta h_i + \varepsilon^2 \cdot (\Delta h_i)^2 \\ &\quad + 2 \cdot \frac{\beta}{2} \cdot (\Delta u_i + \varepsilon \cdot \Delta h_i) \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i + \varepsilon \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} h_i \right) \\ &\quad + \left(\frac{\beta}{2} \right)^2 \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i + \varepsilon \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} h_i \right)^2 - (\Delta u_i)^2 \\ &\quad - 2 \cdot \frac{\beta}{2} \cdot \Delta u_i \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i - \left(\frac{\beta}{2} \right)^2 \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right)^2 d\vec{x} \\ &= \frac{1}{2} \int_{\Omega} \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \sum_{i=1}^n 2 \cdot \varepsilon \cdot \Delta u_i \cdot \Delta h_i + 2 \cdot \varepsilon \cdot \frac{\beta}{2} \cdot \left(\Delta u_i \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} h_i + \Delta h_i \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) \\ &\quad + 2 \cdot \varepsilon \cdot \left(\frac{\beta}{2} \right)^2 \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} h_i \right) + \mathcal{O}(\varepsilon^2) d\vec{x} \\ &= \int_{\Omega} \sum_{i=1}^n \Delta u_i \cdot \Delta h_i + \frac{\beta}{2} \cdot \left(\Delta u_i \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} h_i + \Delta h_i \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) \\ &\quad + \left(\frac{\beta}{2} \right)^2 \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} h_i \right) d\vec{x} \end{aligned}$$

partielle Integration

$$\begin{aligned} &= - \int_{\Omega} \sum_{i=1}^n (\nabla h_i)^t \cdot (\nabla \Delta u_i) + \frac{\beta}{2} \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_k} \Delta u_i \cdot \frac{\partial}{\partial x_j} h_i + (\nabla h_i)^t \cdot \nabla \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) \right) \\ &\quad + \left(\frac{\beta}{2} \right)^2 \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial^2}{\partial x_k^2} u_i \right) \cdot \left(\sum_{j=1}^n \frac{\partial}{\partial x_j} h_i \right) d\vec{x} \\ &\quad + \oint_{\partial\Omega} \sum_{i=1}^n (\nabla h_i)^t \cdot \vec{n} \cdot \Delta u_i + \frac{\beta}{2} \cdot \left(\Delta u_i \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} h_i \cdot \vec{n}_k + (\nabla h_i)^t \cdot \vec{n} \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) \right) \\ &\quad + \left(\frac{\beta}{2} \right)^2 \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} h_i \cdot \vec{n}_k \right) ds \end{aligned}$$

nochmal partiell integriert

$$\begin{aligned}
&= + \int_{\Omega} \sum_{i=1}^n h_i \cdot \Delta^2 u_i + \frac{\beta}{2} \cdot h_i \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} \Delta u_i + \Delta \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) \right) + \left(\frac{\beta}{2} \right)^2 \cdot h_i \cdot \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial^2}{\partial x_j^2} \frac{\partial^2}{\partial x_k^2} u_i \right) d\vec{x} \\
&+ \oint_{\partial\Omega} \sum_{i=1}^n (\nabla h_i)^t \cdot \vec{n} \cdot \left(\Delta u_i + \frac{\beta}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) + \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} h_i \cdot \vec{n}_k \right) \cdot \left(\frac{\beta}{2} \cdot \Delta u_i + \left(\frac{\beta}{2} \right)^2 \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) ds \\
&- \oint_{\partial\Omega} \sum_{i=1}^n h_i \left(\vec{n}^t \cdot (\nabla \Delta u_i) + \frac{\beta}{2} \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_k} \Delta u_i \vec{n}_j + \vec{n}^t \cdot \nabla \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) \right) + \left(\frac{\beta}{2} \right)^2 \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial^2}{\partial x_k^2} u_i \vec{n}_j \right) \right) ds \\
&= + \int_{\Omega} \sum_{i=1}^n h_i \cdot \left(\left(1 + \left(\frac{\beta}{2} \right)^2 \right) \cdot \Delta^2 u_i + 2 \cdot \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} \Delta u_i \right) d\vec{x} \\
&+ \oint_{\partial\Omega} \sum_{i=1}^n (\nabla h_i)^t \cdot \vec{n} \cdot \left(\Delta u_i + \frac{\beta}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) + \left(\sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} h_i \cdot \vec{n}_k \right) \cdot \left(\frac{\beta}{2} \cdot \Delta u_i + \left(\frac{\beta}{2} \right)^2 \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) ds \\
&- \oint_{\partial\Omega} \sum_{i=1}^n h_i \cdot \left(\vec{n}^t \cdot \nabla \left(\Delta u_i + \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) + \left(\frac{\beta}{2} + \left(\frac{\beta}{2} \right)^2 \right) \cdot \left(\sum_{j=1}^n \frac{\partial}{\partial x_j} \Delta u_i \cdot \vec{n}_j \right) \right) ds
\end{aligned}$$

Da die Richtung $\vec{h} \in H_2(\Omega)^n$ beliebig gewählt war, müssen die Integrale unabhängig von h_i und ∇h_i im Minimum von S_3 verschwinden. Die Randwertaufgabe zum Navier-Lamé Operator ist also für $i = 1, \dots, n$

$$\left\{ \begin{array}{l} \left(1 + \left(\frac{\beta}{2} \right)^2 \right) \cdot \Delta^2 u_i + 2 \cdot \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} \Delta u_i = 0 \quad \text{auf } \Omega \\ \vec{n} \cdot \left(\Delta u_i + \frac{\beta}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) + \vec{n} \cdot \left(\frac{\beta}{2} \cdot \Delta u_i + \left(\frac{\beta}{2} \right)^2 \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) = \vec{0} \quad \text{auf } \partial\Omega \\ \vec{n}^t \cdot \nabla \left(\Delta u_i + \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) + \left(\frac{\beta}{2} + \left(\frac{\beta}{2} \right)^2 \right) \cdot \left(\sum_{j=1}^n \frac{\partial}{\partial x_j} \Delta u_i \cdot \vec{n}_j \right) = \vec{0} \quad \text{auf } \partial\Omega. \end{array} \right.$$

$$\iff \left\{ \begin{array}{l} \left(1 + \left(\frac{\beta}{2} \right)^2 \right) \cdot \Delta^2 u_i + 2 \cdot \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} \Delta u_i = 0 \quad \text{auf } \Omega \\ \left(1 + \frac{\beta}{2} \right) \cdot \vec{n} \cdot \left(\Delta u_i + \frac{\beta}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) = \vec{0} \quad \text{auf } \partial\Omega \\ \vec{n}^t \cdot \nabla \left(\Delta u_i + \frac{\beta}{2} \cdot \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} u_i \right) + \left(\frac{\beta}{2} + \left(\frac{\beta}{2} \right)^2 \right) \cdot \left(\sum_{j=1}^n \frac{\partial}{\partial x_j} \Delta u_i \cdot \vec{n}_j \right) = \vec{0} \quad \text{auf } \partial\Omega. \end{array} \right.$$

Diese Randwertaufgabe lässt sich aufgrund der Koppelung der Richtungen nicht mit Fourierentwicklung lösen. Eine Möglichkeit für effiziente Löser sind Mehrgitterverfahren.

B CD mit Beispielbildern und Programmcode

Auf der beiliegenden CD befindet sich der Quelltext des Programms zur morphologischen multimodalen Bildanpassung sowie einige 3D Beispielbilder. Dem Programm stehen die Abstandsmaße d , \tilde{d} , eine Konvexkombination der beiden und das monomodale Abstandsmaß $R - T$ zur Verfügung.

Inhalt der CD

- /examples/
 - erg2D115/ enthält die Beispiele zum Modellhaus, welche auszugsweise oberhalb abgedruckt sind. Es empfiehlt sich die Datei erg2D115/index.htm mit einem Webbrowser zu öffnen.
 - mr_mr/ zwei dreidimensionale Magnet-Spin-Resonanz-Aufnahmen von zwei verschiedenen Menschen werden aufeinander angepasst.
 - mr_atlas/ eine Graustufenversion des Gehirnatlasses „Atlas of the Human Brain“ von J. K. Mai, T. Voß und anderen (siehe auch <http://www.thehumanbrain.info/>) ist auf das freigestellte Gehirn einer Magnet-Spin-Resonanz-Aufnahme angepasst.
- /software/
 - img_reg1d/ Programmcode für eindimensionale Tests.
 - img_reg2d/ Programmcode für die zweidimensionale Bildanpassung.
 - img_reg3d/ Programmcode für die dreidimensionale Bildanpassung.
 - fftw/ die schnelle Sinus- und Cosinustransformation von FFTW (siehe auch <http://www.fftw.org/>).

Weitere Informationen und eine Installationsanleitung befinden sich in /software/install.txt.

Bei der Verwendung des Programms ist das Urheberrecht zu beachten.

Hier sollte sich eine CD mit
oben genanntem Inhalt befinden.

Erklärung der Originalität

Die hier vorgelegte Diplomarbeit habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Alle verwendeten Quellen und Hilfsmittel sind im Literaturverzeichnis aufgelistet bzw. im Text genannt. Zitate sind kenntlich gemacht. Diese Arbeit wurde weder in dieser noch in ähnlicher Form bei keiner anderen Institution eingereicht.

Düsseldorf, den 19. März 2007

(Dominik Löchel)

Literatur

- [1] FFTW (<http://www.fftw.org/>) Version 3.1.2
- [2] Hanselman, D., Littlefield, B.: *Mastering Matlab 7*. Pearson (2005)
- [3] Hömke, L.: *A multigrid method for elastic image registration with additional structural constraints*. Inaugural-Dissertation, Heinrich-Heine-Universität Düsseldorf (2006)
- [4] Henn, S.: *Numerische Lösung und Modellierung eines inversen Problems zur Assimilation digitaler Bilddaten*. Logos (2001).
- [5] Henn, S., Jarre, F., Witsch, K.: *Mathematische Bildverarbeitung - Ein Überblick über verschiedene Modelle und Methoden zur Registrierung digitaler Bilddaten*. Jahrbuch der Heinrich-Heine-Universität Düsseldorf 2003.
- [6] Henn, S., Witsch, K.: *Multimodal Image Registration using a Variational Approach*. SIAM NEWS 25 (4), pp. 1429-1447, (2003).
- [7] Henn, S., Witsch, K.: *Curv-Flat Image Registration by Using Topographic Surface Structure from 2D Images*. Preprint.
- [8] Henn, S., Witsch, K.: *Multimodal Image Registration*. Preprint.
- [9] Jarre, F., Stoer, J.: *Optimierung*. Springer (2004)
- [10] Mai, J. und andere: *Atlas of the Human Brain*. (<http://www.thehumanbrain.info/>)
- [11] Stoer, J.: *Numerische Mathematik 1*. Springer (2005)
- [12] Witsch, K.: *Postconditioned Nonlinear CG*. Preprint (2006)

Abbildungsverzeichnis

- 4.1 Wirkung der Regularisierung im eindimensionalen auf ein Bild mit 100 Punkten. u_a ist die Lösung der harmonischen und u_c der biharmonischen Randwertaufgabe mit Dirichlet-Randbedingung. Bei u_b wurden die Eigenwerte der harmonischen RWA im Frequenzraum hoch drei halbe genommen. Dementsprechend liegt die Wirkung zwischen a und c . Der Faktor 20 wurde nur zur besseren Übersicht eingefügt. Es ist deutlich zu erkennen, dass die regularisierte Kurve nur die wesentliche Form der Kräfte f wiedergibt. Das lokal starke gezitterte wird geglättet. Wird die Kraft f direkt als Suchrichtung gewählt, so verändert sich die Reihenfolge der Punkte. Mit der regularisierten Suchrichtung werden einige Punkte dichter aneinander gedrückt und andere weiter auseinander gezerrt, die Reihenfolge bleibt dabei allerdings erhalten. 20
- 5.1 Glättung mit der Wärmeleitungsgleichung zu verschiedenen Zeitwerten t mit Dirichlet- und Neumannrandbedingung. Zum besseren Erkennen ist der Kontrast bei (b) bis (g) nach dem Glätten erhöht worden. Die Wahl der Randbedingung macht im Randbereich einen erkennbaren Unterschied. So ist in (d) der Rand weiterhin schwarz. Dagegen ist in (g) das Bild über den Rand hinaus verlaufen. Mit zunehmenden t werden Details immer schwächer. Dagegen werden bei $t = 100$ die groben Formen wie der Umriss des Hauses und die Nummern-Tafeln sehr deutlich. 22
- 5.2 Trilineare Interpolation als sukzessive eindimensionale lineare Interpolation 24
- 5.3 Das Minimum der Funktion $g(x_1, x_2) = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2$ wird mit dem Gradientenverfahren (a) und dem Verfahren der konjugierten Gradienten (b) bestimmt. Das exakte Minimum befindet sich in $g(1, 1) = 0$. Bei (a) beträgt die Höhe nach 200 Iterationen $g(0.98348764024707, 0.96719445548042) = 2.729 \cdot 10^{-4}$ und nach 400 Iterationen $g(0.98821389654450, 0.97650878572053) = 1.392 \cdot 10^{-4}$. Mit dem CG-Verfahren (b) ist das Minimum nach bereits 112 Iterationen mit $g(0.99999955217203, 0.99999910135431) = 2.014 \cdot 10^{-13}$ hinreichend genau gefunden. Im Schaubild sind die einzelnen Iterierten durch ein gelbes Kreuz markiert. Es ist zu erkennen, dass die Schrittweiten mit den konjugierten Gradienten deutlich größer ausfallen. 34
- 5.4 Halbieren der Auflösung im eindimensionalen, optimiert für die Sinustransformation. Die oberen Punkte repräsentieren das Bild mit der vollen Auflösung $N = 2^k - 1$, $k \in \mathbb{N}$ und die unteren Punkte das Bild mit etwa halbiertes Auflösung $n = 2^{k-1} - 1$. Die Brüche an den Pfeilen geben die Gewichtung der Farbwerte an. 36
- 5.5 Halbieren der Auflösung im eindimensionalen, optimiert für die Cosinustransformation. Die oberen Punkte repräsentieren das Bild mit der vollen Auflösung $N = 2^k + 1$, $k \in \mathbb{N}$ und die unteren Punkte das Bild mit etwa halbiertes Auflösung $n = 2^{k-1} + 1$. Die Brüche an den Pfeilen geben die Gewichtung der Farbwerte an. 36
- 5.6 Interpolation des Verschiebungsfeldes auf die nächst höhere Auflösungsstufe, hier im eindimensionalen dargestellt und für die Sinustransformation optimiert. Die oberen Punkte repräsentieren das Bild mit der doppelten Auflösung $N = 2^{k+1} - 1$, $k \in \mathbb{N}$ und die unteren Punkte das Bild mit der derzeitigen Auflösung $n = 2^k - 1$. Die Brüche an den Pfeilen geben die Gewichtung der Farbwerte an. 36
- 5.7 Interpolation des Verschiebungsfeldes auf die nächst höhere Auflösungsstufe, hier im eindimensionalen dargestellt und für die Cosinustransformation optimiert. Die oberen Punkte repräsentieren das Bild mit der doppelten Auflösung $N = 2^{k+1} + 1$, $k \in \mathbb{N}$ und die unteren Punkte das Bild mit der derzeitigen Auflösung $n = 2^k + 1$. Die Brüche an den Pfeilen geben die Gewichtung der Farbwerte an. 36
- 5.8 Ein eindimensionales Beispiel zur Verwendung von Kontrollpunkten mit zwei verschiedenen Regularisierungen. Die Liste der Kontrollpunkte ist $\mathcal{P} := \{(4, 3.5), (9, 11.5), (10, 12.5), (21, 22), (25, 23), (29, 28.5), (35, 36), (37, 37.5), (45, 45.5)\}$. In der Linken Spalte wurde die harmonische und in der rechten Spalte die biharmonische Regularisierung verwendet. Die Lage der Punkte aus \mathcal{P} ist verständlicher Weise gleich, aber die Lage der übrigen Punkte ist echt verschieden. Die Reihenfolge der Punkte ist in beiden Fällen erhalten geblieben, so dass auf eine Nummerierung der Linien in (c) und (d) verzichtet werden kann. Es muss noch gesagt werden, dass die gepunkteten Verschiebungsfelder in (a) und (b) so normiert sind, dass sie durch den zugehörigen Kontrollpunkt (*) verlaufen. Normalerweise sind sie viel dichter an der Nulllinie. 38
- 6.1 Der Verlauf der Gewichtungsfunktionen im Frequenzraum für drei verschiedene starke Glättung mit der Wärmeleitungsgleichung. Bei der Gaußglättung wird die Funktion nach links symmetrisch fortgesetzt. x ist dabei die relative Frequenzhöhe und $f(x)$ die Gewichtung. Je schneller die Gewichtungsfunktionen gegen Null gehen, um so verschwommener ist das resultierende Bild, da die hohen Frequenzanteile stark abgedämpft sind. 43

- 6.2 Der Abstand der Bilder $\mathcal{D}_{R,T}(\phi(\beta)) = \frac{\sqrt{\int_{\Omega} d(R, T \circ \phi(\beta); \bar{x})^2 d\bar{x}}}{n_1 \cdot n_2}$ bei Drehung um den Mittelpunkt mit Drehwinkel β und verschieden starker Glättung t in der Wärmeleitungsgleichung. R ist das Bild in Abb. 7.1 auf Seite 58 (a) und T ist die unverzerrte Version von (b), also das Ergebnis der idealen Transformation. Die Bilder sind mit Drehwinkel $\beta = 0$ deckungsgleich. Bei geringer Glättung existieren viele lokale Minima. Bei starker Glättung existieren nur noch einzelne lokale Minima. Allerdings ist das globale Minimum etwas verschoben. Der Bildanpassungsalgorithmus fängt mit (f) an. Dort wird das globale Minimum leicht gefunden. Dann wird die Glättung etwas reduziert. Durch die Nähe zum globalen Minimum wird bei vorsichtiger Zurücknahme der Glättung das gewünschte Minimum $\beta = 0$ in (a) gefunden werden. 44
- 6.3 Die Ausgangsbilder R und T mit einer Auflösung von 33×33 Pixel sind in (a) bzw. (b) gezeigt. Das Vektorfeld der Gradienten ∇R (rot), ∇T (grün) und der Kräfte $\vec{f}(R, T)$ (blau) die sich mit Maß d ergeben, ist in (c). Es ist zu sehen, dass Kräfte nur in Punkten auftreten, wo weder ∇R noch ∇T verschwindet. 48
- 6.4 Die geglätteten Bilder R_s und T_s mit einer Auflösung von 33×33 Pixel sind in (a) bzw. (b) gezeigt. Das Vektorfeld der Gradienten ∇R_s (rot), ∇T_s (grün) und der Kräfte $\vec{f}(R_s, T_s)$ (blau) die sich mit Maß d ergeben, ist in (c) gezeichnet. Durch das Glätten ist der Überlappungsbereich der Gradientenfelder von R und T größer. Dementsprechend treten auch an mehr Stellen Kräfte $\vec{f}(R_s, T_s)$ auf. 49
- 6.5 Das Vektorfeld der Kräfte \vec{f} und der biharmonisch regularisierten Kräfte \vec{u} , so wie das durch die Kräfte \vec{f} selbst (b) und durch die regularisierten Kräfte \vec{u} (c) verformte Templatebild. Hier wurde nur eine einzige Iteration durchgeführt und die Schrittweite $\tau = 4.6 \cdot 10^7$ und $\rho = 1.2 \cdot 10^6$ manuell bestimmt. Während das Bild in (b) bereits zerstückelt ist, bleibt der Zusammenhalt des Objektes mit der regularisierten Suchrichtung auch bei größeren Schrittweiten gewährleistet (d). 50
- 6.6 Ein Programmdurchlauf über 3 Auflösungsstufen mit Maß d und harmonischer Randwertaufgabe mit Dirichlet Randbedingung. In der Koordinatentransformation (h) lässt sich die Transformation vom größten Gitter noch erkennen. 51
- 6.7 Ein Programmdurchlauf über 3 Auflösungsstufen mit Maß d und biharmonischer Randwertaufgabe mit Dirichlet Randbedingung. Die Koordinatentransformation ist deutlich glatter als bei der harmonischen Regularisierung. 51
- 6.8 Ein Programmdurchlauf über 3 Auflösungsstufen mit Maß \tilde{d} und harmonischer Randwertaufgabe mit Dirichlet Randbedingung. 52
- 6.9 Ein Programmdurchlauf über 3 Auflösungsstufen mit Maß \tilde{d} und biharmonischer Randwertaufgabe mit Dirichlet Randbedingung. Auch hier ist die Koordinatentransformation (h) deutlich glatter als bei der harmonischen Regularisierung. 52
- 6.10 Die Ausgangsbilder R und T mit einer Auflösung von 33×33 Pixel sind in (a) bzw. (b) gezeigt. Das Vektorfeld der Gradienten ∇R (rot), ∇T (grün) und der Kräfte $\vec{f}(R, T)$ (blau) die sich mit Maß d ergeben, ist in (c). Es ist zu sehen, dass lediglich zwei Kräfte in den beiden Endpunkten auftreten. 54
- 6.11 Die geglätteten Bilder R_s und T_s mit einer Auflösung von 33×33 Pixel sind in (a) bzw. (b) gezeigt. Das Vektorfeld der Gradienten ∇R_s (rot), ∇T_s (grün) und der Kräfte $\vec{f}(R_s, T_s)$ (blau) die sich mit Maß d ergeben, ist in (c) gezeichnet. Durch das Glätten gibt es mehr nicht-parallele Gradienten. Dementsprechend treten auch an mehr Stellen Kräfte $\vec{f}(R_s, T_s)$ auf. 55
- 6.12 Das Vektorfeld der Kräfte \vec{f} und der biharmonisch regularisierten Kräfte \vec{u} , so wie das durch die Kräfte \vec{f} selbst (b) und durch die regularisierten Kräfte \vec{u} (c) verformte Bild. Hier wurde nur eine einzige Iteration durchgeführt und die Schrittweite $\tau = 1.3 \cdot 10^8$ und $\rho = 4.2 \cdot 10^5$ manuell bestimmt. Die leichte Krümmung in (c) wird normalerweise von den nachfolgenden Iterationen kompensiert. 56
- 7.1 Das Modellproblem. Gegeben sind das Referenzbild R und das Templatebild T . Die Aufgabe besteht darin, das Templatebild T so zu verformen, dass seine Gestalt zum Referenzbild R passt. Dabei werden nur Informationen aus den Gradienten verwendet. Die Höhenlinien sind in der zweiten Zeile zu sehen. Die Auflösung aller vier Bilder beträgt $513 \times 513\text{px}^2$ 58
- 7.2 Ein Beispiel mit Maß d (vgl. 2.1.1 auf Seite 6). Gegeben sind die Bilder R und T aus Abb. 7.1 auf Seite 58 mit einer Auflösung von $513 \times 513\text{px}^2$. Die Registrierung wurde auf einer Auflösung von $17 \times 17\text{px}^2$ begonnen und schrittweise verdoppelt bis zur vollen Auflösung. Die Rechenzeit betrug 2 Minuten. Die Glättung erfolgt mit der Cosinustransformation und die Regularisierung mit der Sinustransformation. 59
- 7.3 Gegeben sind wieder R und T aus Abb. 7.1 auf Seite 58. Diesmal wird die harmonische Regularisierung verwendet. Alle übrigen Parameter sind die gleichen wie in Abbildung 7.2 auf Seite 59. Es treten lokal starke artifizielle Deformationen auf, die sich nicht aus den Strukturen des Bildes nachvollziehen lassen. Die Residuen im Iterationsverlauf sind in Abb. 7.15 auf Seite 74 aufgetragen. 60

- 7.4 Gegeben sind wieder R und T aus Abb. 7.1 auf Seite 58. Diesmal wird die sesqui-harmonische Regularisierung verwendet. Alle übrigen Parameter sind die gleichen wie in Abbildung 7.2 auf Seite 59. Gegenüber der einfach-harmonischen Regularisierung in Abbildung 7.3 auf Seite 60 sind die artifizialen Deformationen hier deutlich geringer. Doch am Dachgiebel ist dieser Fehler immer noch zu erkennen. Die Residuen im Iterationsverlauf sind in Abb. 7.15 auf Seite 74 aufgetragen. 60
- 7.5 Gegeben sind R und T aus Abb. 7.1 auf Seite 58. Diesmal wird Maß \tilde{d} (vgl. 2.2 auf Seite 12) verwendet. Alle weiteren Parameter sind die gleichen wie bei Abbildung 7.2 auf Seite 59. Die Registrierung wurde auf einer Auflösung von $17 \times 17 \text{px}^2$ begonnen und schrittweise verdoppelt bis zur vollen Auflösung. Die Rechenzeit betrug 2 Minuten. 61
- 7.6 Vergleich der harmonischen und der elastischen Regularisierung. Letztere ist der Navier-Lamé Operator mit $\beta = 2$. Der wesentliche Unterschied lässt sich im eingekreisten Bereich erkennen. Im harmonischen Fall laufen ursprünglich senkrechte Gitterlinien stellenweise parallel. Der Navier-Lamé Operator vermeidet dies. Referenz und Templatebild sind die gleichen wie in Abb. 7.1 auf Seite 58. Aufgrund des hinreichend breiten Randes stört die periodische Randbedingung nicht. 62
- 7.7 Das CT-Bild wird an das MR-Bild des gleichen Gehirns angepasst. Als Orientierungshilfe für den Leser ist hier die Kontour des MR-Bildes in allen Bildern eingefügt. Die Regularisierung erfolgt nach dem elastischen Modell des Navier-Lamé Operators mit $\beta = 2$ 63
- 7.8 Mit harmonischer Regularisierung in der Schrichtung und $\alpha = 0.001$ in der Liniensuche. Die Residuen $D(i) := \sqrt{\mathcal{D}(\phi(\vec{u}^i))}$ reduzieren sich dabei deutlich langsamer als mit $\alpha = 0$ 66
- 7.9 Eine multimodale Registrierung. Gegeben sind die Bilder R (a) und T (b) mit einer Auflösung von $513 \times 513 \text{px}^2$. Begonnen wird mit $65 \times 65 \text{px}^2$ und dann wird die Auflösung schrittweise verdoppelt. Die Tür, das Erdgeschossfenster, die Nummernschilder, das Schuppendach und der Schornstein sind an die richtige Position geschoben worden. Lediglich die Bretterritzen vom Schuppen und das Kellerfenster sind noch an verkehrter Stelle. Die Koordinatentransformation ϕ ist glatt. Die Rechenzeit beträgt 3 Minuten. Bei der gleichen Rechnung ohne CG-Verfahren wird nur die Abweichung an der linken Hausecke angepasst und das übrige T^ϕ sieht dem T ähnlicher als dem R . Das gleiche Beispiel funktioniert auf Grund der stark verschiedenen Farbdifferenzen mit Maß \tilde{d} überhaupt nicht. 68
- 7.10 In der Linken Spalte sind drei Fotos des im wesentlichen gleichen Objektes mit unterschiedlichen Beleuchtungseigenschaften. Rechts daneben ist die euklidische Norm des Gradienten aufgezeichnet. An weißen Stellen ist der Gradient Null. Je dunkler ein Bildpunkt ist, umso größer ist dort die Norm. Die Stärke der Gradienten an einigen Objekten ändert sich recht stark zwischen den drei Bildern. So verläuft eine stark ausgeprägte Kante entlang des Dachfirsts in (a) und (b). In (c) hingegen ist die Kante entlang der Regenrinne am stärksten. In Bild (a) sind die Umrisse der meisten Fenster stärker als in (b) oder (c). 70
- 7.11 Gradiententransformation. x entspricht $\|\nabla R\|_2$ und y entspricht $\|\nabla R_{\text{angepasst}}\|_2$ 71
- 7.12 Ein Beispiel mit normiertem Gradient ($q = 0.9$). Des weiteren ist die biharmonische Regularisierung mit Dirichlet Randbedingung und Maß d verwendet. Außerdem wird die in Abbildung 7.14 auf Seite 72 gezeigte Gewichtungsmaske eingesetzt, um die Kante rings herum am Bildrand und die Autos am unteren Bildrand für das Programm unsichtbar zu machen. Die Bilder sind aus dem auf $513 \times 513 \text{px}^2$ vergrößerten Bild ausgeschnitten. Im wesentlichen ist T^ϕ richtig zurecht gebogen. Aufgrund der Gradientennormierung sind einige Kanten, wie z.B. der Dachfirst, bis zu 5 px voneinander entfernt. 71
- 7.13 Verschiedene Möglichkeiten, den Rand der Bilder zu ergänzen. Das Rechteck in (a) soll das Referenzbild darstellen. Dieses enthält zwei Linien, die am unteren Rand fortgesetzt sind. In (b) bis (d) ist das perspektivisch verzerrte Templatebild mit verschiedenen Möglichkeiten, den Rand fort zu setzen. In (b) ist die Version, die bei Umkehrung der Verzerrung (a) ergibt. Diese ist in der Praxis erst nach erfolgreicher Bildregistrierung bekannt. In (c) und (d) sind zwei Varianten, die sich ohne Kenntnis der Transformationsabbildung realisieren lassen. 72
- 7.14 Das ist die Gewichtungsmaske für Beispiel 7.4 auf Seite 70 und 7.12. Die Kante am Rand der Fotos und die Autos im unteren Bildbereich werden damit für das Bildanpassungsprogramm unsichtbar. Der schwarze Bereich wird dann ausschließlich durch die Regularisierung überbrückt. 72
- 7.15 Die Residuen $D(i) := \sqrt{\mathcal{D}(\phi(\vec{u}^i))}$ im Iterationsverlauf mit und ohne CG-Verfahren bei verschiedenen Regularisierungsstärken, Randbedingungen und den Maßen d (Maß 1) und \tilde{d} (Maß 2). Zu beachten ist, dass die D -Achse logarithmisch skaliert ist. Bei linearer Skalierung ist die Abflachung der Teilkurven noch deutlicher zu erkennen. Weitere Informationen befinden sich im Text. 74
- 7.16 Die Auswirkung von Bildrauschen auf den Anpassungsvorgang mit Maß d , Glättung mit der Wärmeleitungsgleichung, Regularisierung mit Navier-Lamé und CG-Verfahren. Das Referenzbild ist jeweils oben und das Templatebild unten links. Rechts daneben befindet sich das Ergebnis der Anpassung. Die Bildanpassung klappt im wesentlichen auch bei verrauschtem Referenzbild. Allerdings werden auch durch Rauschen entstandene Kanten nach geformt. 75

- 7.17 Eine Graustufenvariante des „Atlas of the Human Brain“ [10] wurde auf das Magnetresonanzbild eines fremden Gehirns (erste Zeile) angepasst. Die so erhaltene Transformation ϕ wurde auf die drei Farbk채n채le des gef채rbbten Atlases angewendet (zweite und dritte Zeile). Auf diese Weise ist es moglich, anhand der Farben die Funktion vom jeweiligen Hirnareal des Patienten (erste Zeile) in der Legende des Atlases nach zu schlagen. In der zweiten Zeile ist das Ergebnis mit biharmonischer Regularisierung. In Zeile drei wurde der Navier-Lam e Operator mit $\beta = 2$ eingesetzt. Dieser liefert hier das beste Ergebnis. Es sind jeweils drei Schnittebenen aus verschiedenen Raumrichtungen abgebildet. Die weien Striche markieren die Stelle, an der die beiden anderen Ebenen durchstoen. Die Bildauflosung ist aus coronaler Sicht $142 \times 128 \times 189$ Voxel (Breite \times Hohe \times Tiefe). In der vierten Zeile ist die Oberflache rekonstruiert. Die Strukturen vom angepassten Atlas haben dabei deutliche hnlichkeit zum MR-Bild. 77

Liste der Algorithmen

- | | | |
|---|---|----|
| 1 | Minimumsuche entlang der Suchrichtung mit kubischer Interpolation | 32 |
| 2 | Die Iteration uber die Auflosungsstufen | 41 |
| 3 | Die Iteration uber die Glattungsstufen | 42 |
| 4 | Die innere Iteration der Bildanpassung | 45 |

Index

$H_2(\Omega)$, 2

$H_2^D(\Omega)$, 2

$H_2^N(\Omega)$, 2

T^ϕ , 6, 8

Ω , 2

$\delta(\bar{x})$, 37

$\partial\Omega$, 2

$\partial\Omega_D$, 26

$\partial\Omega_N$, 26

DFT^{3D}, 22

\vec{n} , 17

\mathcal{L} , 26

\mathcal{P} , 37

CG-Verfahren, 33

DCT, 21

DFT, 21

dividierte Differenzen, 30

DST, 21

feste Kontrollpunkte, 39

full weighting, 35

half weighting, 35

ill-posed, 1

injection, 35

inverses Problem, 1

line search, 29

Liniensuche, 29

Navier-Lamé, 28

P (Kontrollpunkt), 37

schlecht gestelltes Problem, 1

sesqui-harmonischer Operator, 57

Sobolevraum, 2

Topologieerhaltung, 3

Wärmeleitungsgleichung, 23

weiche Kontrollpunkte, 39