



UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS AND
IMAGE COMPUTING

Verwendung der mittleren Krümmung als Regularisierer in der Bildregistrierung Herleitung, Analyse und Erweiterung auf 3D

*Using the mean curvature model as a regularizer in
image registration*

Masterarbeit

im Rahmen des Studiengangs
Mathematik in Medizin und Lebenswissenschaften
der Universität zu Lübeck

vorgelegt von

Patricia Galuschka

ausgegeben und betreut von

Prof. Dr. Jan Modersitzki
Institute of Mathematics and Image Computing

mit Unterstützung von

Dr. Benjamin Wacker
Institute of Mathematics and Image Computing

Lübeck, den 10.04.2018

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Arbeit unterstützt haben.

Ganz besonders gilt dieser Dank meinem Betreuer Benjamin Wacker, der stets ein offenes Ohr für meine Fragen und Probleme hatte und mir immer wieder neue Denkanstöße gegeben hat.

Bei meinen Kommilitoninnen, die mit mir im Studenten-Pool des Instituts gearbeitet haben, möchte ich mich ebenfalls bedanken. Sie haben mich immer wieder motiviert und dank ihrer ist der Spaß an der Arbeit nicht zu kurz gekommen.

Ein weiteres Dankeschön geht an meine Familie, die sich Zeit für das Korrekturlesen genommen hat und immer für mich da ist.

Eidesstattliche Erklärung

Ich versichere an Eides statt, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Lübeck, den 10. April 2018

Patricia Galuschka

Kurzfassung

In dieser Arbeit wurde der 2016 von Ke Chen eingeführte neuartige Ansatz zum Lösen eines Bildregistrierungsproblems untersucht. Bei diesem Ansatz wird die mittlere Krümmung aus der Differentialgeometrie in dem zu minimierenden Registrierungsproblem als Regularisierer verwendet. Es wird der Behauptung nachgegangen, dass dadurch sehr gute Ergebnisse auf glatten wie auch auf nichtglatten Registrierungsbeispielen erhalten werden. Dazu werden die Grundlagen der Differentialgeometrie aufgearbeitet und der Term, der als Regularisierer verwendet wird, hergeleitet. Das Modell zum Lösen dieses Registrierungsproblems wird im Zweidimensionalen aufgestellt und theoretisch analysiert, indem die Euler-Lagrange-Gleichungen bestimmt und die Rotations- und Translationsinvarianz nachgewiesen wurden. Es wurde zusätzlich eine mögliche Erweiterung für drei Dimensionen entworfen. Ein weiterer Aspekt dieser Arbeit war die praktische Umsetzung und die Anwendung auf glatte und nichtglatte Registrierungsbeispiele in zwei, wie auch in drei Dimensionen. Dabei konnte, wie in der Arbeit von Ke Chen, demonstriert werden, dass die Differenz zwischen dem Referenzbild und dem transformierten Templatebild sehr klein wird. Zusätzlich wurden in dieser Arbeit auch die Deformationsgitter betrachtet, die jedoch nicht die gewünschten Eigenschaften haben, da sie Faltungen aufweisen. Eine Weiterentwicklung dieses Ansatzes, um Gitterfaltungen zu verhindern, wurde versucht, konnte aber im Rahmen dieser Arbeit nicht abgeschlossen werden.

Abstract

In 2016 Ke Chen proposed a new model for solving an image registration problem by using mean curvature from differential geometry as a regularizer. The objective of this thesis is to investigate if this model really delivers good results for both smooth and non-smooth registration problems. For this purpose, the fundamental differential geometry principles are prepared to derive the term that is used as a regularizer. Then the model for solving the registration problem is set up in two dimensions and a theoretical analysis is also done. In addition, a possible extension to three dimensions was elaborated. Another aspect of this thesis was the implementation and application to smooth and non-smooth registration examples in two and three dimensions respectively. It could be shown that with this model the distance between the reference image and the transformed template image can become very small. This thesis had an additional look at the deformation grids, whereby it turned out that they show mesh folding. Further steps were taken to avoid mesh folding but completing them went beyond the scope of this thesis.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Aufbau der Arbeit	2
2. Mathematische Grundlagen	3
2.1. Grundlagen der Differentialgeometrie	3
2.1.1. Kurven und ihre Krümmungen	3
2.1.2. Reguläre Flächen	5
2.1.3. Gauß-Abbildung	9
2.2. Funktionalanalytische Grundlagen	16
2.3. Grundlagen der Bildregistrierung	18
2.3.1. Problemstellung	19
2.3.2. Diskretisierung	19
2.3.3. Distanzmaße	21
2.3.4. Regularisierer	22
2.3.5. Lösen des Registrierungsproblems	24
3. Modellbeschreibung in 2D	27
3.1. Numerische Lösung des Registrierungsproblems	31
3.1.1. Diskretisierung	31
3.1.2. Lösen des diskreten Optimierungsproblems	36
3.2. Erläuterungen zur Implementierung in MATLAB	39
4. Erweiterung des Modells auf 3D	43
4.1. Diskretisierung	43
4.2. Erläuterungen zur Implementierung in MATLAB	50
5. Ergebnisse	53
5.1. Anwendung auf zweidimensionale Bilddaten	53
5.2. Anwendung auf dreidimensionale Bilddaten	60
6. Diskussion	65
6.1. Auswertung der Ergebnisse	65
6.2. Ausblick	67
A. Anhang	69
A.1. Zusammenhang Divergenz-Formel und mittlere Krümmung	69
A.2. Bestimmung der Euler-Lagrange-Gleichung	70
A.3. Bestimmung der Jakobimatrix des restringierten Optimierungsproblems	73
Literaturverzeichnis	75

1. Einleitung

Das Thema dieser Arbeit lässt sich der Bildregistrierung zuordnen. Bei der Bildregistrierung besteht die Aufgabe darin, zwei Bilder der gleichen Szene durch Deformation einander ähnlich zu machen. Es gibt unterschiedliche Herangehensweisen, dieses Problem zu lösen. Im Allgemeinen lassen sich diese in zwei verschiedene Arten unterteilen, den merkmalsbasierten und den intensitätsbasierten Verfahren [26]. Bei den merkmalsbasierten Verfahren erfolgt die Registrierung über ausgezeichnete Punkte, sogenannte Landmarken, die übereinander gelegt werden sollen [19]. Bei den intensitätsbasierten Verfahren hingegen wird versucht, über die Intensitätswerte in den zu registrierenden Bildern ein Distanzmaß zu minimieren. Da es sich dabei um ein schlecht gestelltes Problem handelt, wird ein Regularisierer benötigt [27].

In dieser Arbeit wird ein neuartiger Ansatz im Bereich der intensitätsbasierten Verfahren untersucht, der erstmals 2016 in [32] eingeführt wurde. Dabei wird die mittlere Krümmung aus der Differentialgeometrie in dem zu minimierenden Registrierungsproblem als Regularisierer verwendet. Es wird in [32] behauptet, dass dieser Registrierungsansatz auf glatten wie auf nichtglatten Registrierungsproblemen sehr gute Ergebnisse liefert. In dieser Arbeit wurde dieser Aussage nachgegangen, indem der dort eingeführte Registrierungsansatz theoretisch analysiert wurde. Es wurden die Euler-Lagrange-Gleichungen aufgestellt und die Rotations- und Translationsinvarianz bewiesen. Außerdem erfolgte die Implementierung dieses Ansatzes und die Anwendung auf praktische Registrierungsbeispiele. Zusätzlich wurde im Rahmen dieser Arbeit eine mögliche Erweiterung auf drei Dimensionen entwickelt, dessen Tauglichkeit an praktischen Beispielen exemplarisch aufgezeigt worden ist.

1.1. Motivation

Die Bildregistrierung ist ein sehr facettenreiches Anwendungsgebiet der Mathematik [27]. Es lassen sich damit Probleme in vielen unterschiedlichen Bereichen wie zum Beispiel der Kunst, der Astronomie, der Biologie, der Chemie oder der Physik angehen [26]. Ein sehr großes Anwendungsgebiet stellt die Medizin dar. Bildgebende Verfahren wie die Sonographie (US), die Magnetresonanztomographie (MRT), die Computertomographie (CT) oder die Positronen-Emissions-Tomographie (PET) helfen Ärzten bei klinischen Diagnostiken. Genau hier setzt die Bildregistrierung an.

In der Bildregistrierung werden die Informationen von zwei oder mehreren Bildern dergleichen Szene, die zu unterschiedlichen Zeiten, von unterschiedlichen Blickwinkeln oder mit unterschiedlichen Geräten aufgenommen wurden, kombiniert [4, 24, 30, 36]. Damit können noch mehr Informationen aus den Bildern gewonnen und die Diagnostik der Ärzte verfeinert werden. Wie in [16] nachzulesen ist, können zum Beispiel Informationen aus MRT-Bildern und PET-Bildern zusammengenommen werden. In MRT-Bildern lässt sich Weichteilgewebe besonders gut erkennen und in PET-Bildern sind Stoffwechselprozesse gut sichtbar. Überlagert man diese beiden Bilder, sodass

1. Einleitung

diese exakt aufeinander passen, lässt sich erkennen, in welchen Organbereichen bestimmte Stoffwechselprozesse stattfinden.

1.2. Aufbau der Arbeit

Das Kapitel 2 befasst sich mit den mathematischen Grundlagen, die für das Verständnis und die Analyse des Registrierungsansatzes von Bedeutung sind. Begonnen wird mit den Grundlagen der Differentialgeometrie. Diese werden soweit aufgearbeitet, bis die Formel der mittleren Krümmung hergeleitet werden kann. Es folgen Grundlagen der Funktionalanalysis, die für die theoretische Analyse des Registrierungsansatzes wichtig sind. Daran schließt der Abschnitt zu den Grundlagen der Bildregistrierung an, in dem das zu lösende Registrierungsproblem aufgestellt wird und Begriffe wie Diskretisierung, Distanzmaß und Regularisierer erläutert werden. Die Beschreibung des Registrierungsansatzes, wie er in [32] formuliert wird, wird in Kapitel 3 gegeben. Dies beinhaltet die Diskretisierung und das Lösen des Optimierungsproblems. Zusätzlich erfolgt in diesem Kapitel die theoretische Analyse dieses Ansatzes. In Kapitel 4 erfolgt die in dieser Arbeit entwickelte Erweiterung dieses Registrierungsansatzes auf drei Dimensionen, wobei im Mittelpunkt die Diskretisierung steht. Das Kapitel 5 beinhaltet die in dieser Arbeit durchgeführten Experimente. Es wurde untersucht, ob mit dem neuartigen Registrierungsansatz tatsächlich auf glatten und nichtglatten Registrierungsbeispielen sehr gute Ergebnisse erzielt werden können. Zusätzlich wurden auch Beispiele im Dreidimensionalen durchgeführt, um die entwickelte Erweiterung auf drei Dimensionen in der Praxis zu überprüfen. Diese Arbeit schließt mit Kapitel 6, in dem die praktischen Ergebnisse des vorherigen Kapitels ausgewertet und mögliche Verbesserungsvorschläge aufgezeigt werden.

2. Mathematische Grundlagen

2.1. Grundlagen der Differentialgeometrie

In diesem Kapitel werden die Grundlagen der Differentialgeometrie von Kurven und Flächen vorgestellt. Diese Darstellung beruht überwiegend auf [9] und soll eine anschauliche Einführung in das Themengebiet der Differentialgeometrie geben. Hierbei werden grundlegende Begriffe der linearen Algebra, wie sie beispielsweise in [2, 23] nachgelesen werden können, als bekannt vorausgesetzt.

Es werden in Abschnitt 2.1.1 Kurven eingeführt und anschließend dargestellt, wie deren Krümmung bestimmt wird. Im Zentrum des Abschnitts 2.1.2 stehen die Definitionen der regulären Fläche und der ersten Fundamentalform. In Abschnitt 2.1.3 werden die Definitionen der Gauß-Abbildung und der zweiten Fundamentalform präsentiert. Das Kapitel schließt mit den Definitionen der Gaußschen Krümmung und der mittleren Krümmung. Besonders die mittlere Krümmung ist für den in dieser Arbeit untersuchten Regularisierer von großer Bedeutung.

2.1.1. Kurven und ihre Krümmungen

In der Differentialgeometrie ist ein interessanter und repräsentativer Teil die Untersuchung von Flächen [9]. Da jedoch in Flächen einige lokale Eigenschaften von Kurven auftreten [9, Kap. 1.1], soll hier zunächst auf Kurven eingegangen werden. Mithilfe dieser Begriffe können im nächsten Abschnitt direkt reguläre Flächen eingeführt werden.

Definition 2.1 (reguläre parametrisierte Kurve, [9, Kap. 1.2, S.2 & Kap. 1.3, S.5]) Sei $I = (a, b)$ mit $a, b \in \mathbb{R} \cup \{-\infty, \infty\}$ ein offenes Intervall. Eine *parametrisierte differenzierbare Kurve* ist eine differenzierbare Abbildung $\alpha: I \rightarrow \mathbb{R}^n$ des Intervalls I in den \mathbb{R}^n mit $n \in \mathbb{N}$. Eine Kurve α heißt *regulär*, falls die komponentenweise Ableitung $\alpha'(t)$ ungleich Null für alle $t \in I$ ist.

Das Wort *differenzierbar* soll hier so verstanden werden, dass α eine Abbildung ist, die jedes $t \in I$ auf einen Punkt $\alpha(t) = (x_1(t), \dots, x_n(t)) \in \mathbb{R}^n$ abbildet, sodass die Funktionen $x_1(t), \dots, x_n(t)$ differenzierbar sind [9]. Die Variable t ist der *Parameter* der Kurve. Man spricht von *ebenen Kurven*, falls $n = 2$ ist, und von *Raumkurven*, falls $n = 3$ ist.

Im Folgenden soll nun der Krümmungsbegriff von Kurven vorgestellt werden. Betrachtet man eine nach der Bogenlänge t parametrisierte Kurve $\alpha: I \rightarrow \mathbb{R}^3$. Dann misst $|\alpha''(t)|$ die Änderungsrate des Winkels zwischen benachbarten Tangenten und der Tangente bei t , da der Tangentenvektor $\alpha'(t)$ die Länge 1 hat [9]. $|\alpha''(t)|$ drückt die Winkelgeschwindigkeit φ aus und gibt demnach hier an, wie schnell sich die Kurve in einer Umgebung von t von der Tangente bei t wegdreht [21].

2. Mathematische Grundlagen

Definition 2.2 (Krümmung einer Kurve an einem Punkt, [21, Kap. 1.3, S.42])

Sei $\alpha: I \rightarrow \mathbb{R}^3$ eine reguläre parametrisierte Kurve mit $\alpha \in C^2(I)$. Das Intervall $[t_1, t_2]$ sei ein Kurvenstück dieser Kurve und $\alpha'(t)$ die erste Ableitung am Punkt t . Es gilt wegen

$$\frac{|\alpha'(t_2) - \alpha'(t_1)|}{|t_2 - t_1|} = \frac{2 \sin \frac{\varphi}{2}}{|t_2 - t_1|} \approx \frac{\varphi}{|t_2 - t_1|} \quad \text{und} \quad \lim_{t_2 \rightarrow t_1} \frac{\alpha'(t_2) - \alpha'(t_1)}{t_2 - t_1} = \alpha''(t_1)$$

für die *Krümmung*

$$\kappa(t_1) = |\alpha''(t_1)| = \omega := \lim_{t_2 \rightarrow t_1} \frac{\varphi}{|t_2 - t_1|},$$

wobei mit φ der Winkel zwischen den beiden Vektoren $\alpha'(t_1)$ und $\alpha'(t_2)$ bezeichnet wird.

Eine Veranschaulichung der Definition 2.2 ist in Abbildung 2.1 zu sehen. Dabei sind an zwei beliebigen Punkten der Kurve jeweils der normierte Tangentenvektor $\alpha'(t)$ durch einen Pfeil mit ausgefüllter schwarzer Pfeilspitze und der Normaleneinheitsvektor durch einen schwarzen Pfeil mit nicht ausgefüllter Pfeilspitze dargestellt. Außerdem ist in rot $\alpha''(t)$ dargestellt, wobei die Länge dieses Vektors, das heißt $|\alpha''(t)|$, angibt, wie schnell sich die Kurve von der Tangente wegdreht und damit anzeigt, wie stark die Krümmung an diesem Punkt ist.

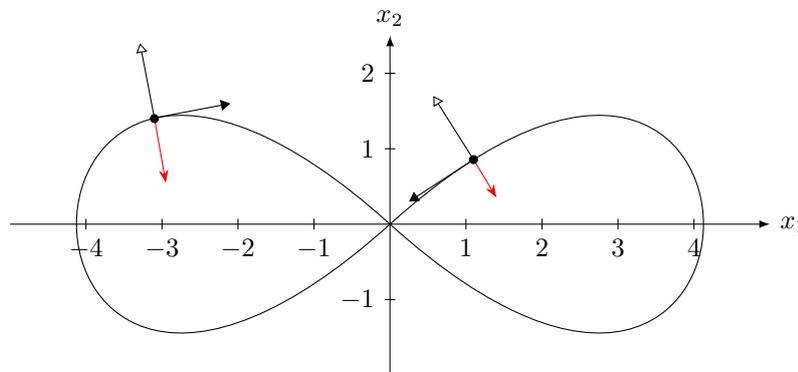


Abbildung 2.1: Skizzenhafte Visualisierung der Definition 2.2. An zwei beliebigen Punkten der Kurve ist jeweils der normierte Tangentenvektor $\alpha'(t)$ durch einen Pfeil mit ausgefüllter schwarzer Pfeilspitze und der Normaleneinheitsvektor durch einen schwarzen Pfeil mit nicht ausgefüllter Pfeilspitze dargestellt. In rot ist $\alpha''(t)$ dargestellt, wobei die Länge dieses Vektors $|\alpha''(t)|$ angibt, wie schnell sich die Kurve von der Tangente wegdreht und damit anzeigt, wie stark die Krümmung an diesem Punkt ist.

Mithilfe der Krümmung lässt sich der Krümmungsradius einer Kurve an einem Punkt $t \in I$ bestimmen.

Definition 2.3 (Krümmungsradius, [9, Kap. 1.5, S.16])

Sei $\alpha: I \rightarrow \mathbb{R}^3$ eine reguläre parametrisierte Kurve. Die Krümmung der Kurve am Punkt t sei $\kappa(t)$. Der Krümmungsradius R an der Stelle $t \in I$ ist definiert durch $R(t) = \frac{1}{\kappa(t)}$.

Es lässt sich anhand dieser Definition leicht nachvollziehen, dass der Krümmungsradius eines Kreises mit dem Radius dieses Kreises übereinstimmt.

2.1.2. Reguläre Flächen

Eine sehr anschauliche Definition von Flächen ist, diese als Teilmengen des \mathbb{R}^3 zu betrachten. Deswegen sollen Flächen hier zunächst auf diese Weise eingeführt werden.

Definition 2.4 (reguläre Fläche, [9, Kap. 2.2, Def. 1])

Eine Teilmenge $S \subset \mathbb{R}^3$ ist eine *reguläre Fläche*, wenn es für jedes $p \in S$ eine Umgebung V in \mathbb{R}^3 und eine Abbildung $\mathbf{x}: U \rightarrow V \cap S$ einer offenen Menge $U \subset \mathbb{R}^2$ auf $V \cap S \subset \mathbb{R}^3$ gibt, sodass gilt:

- (i) \mathbf{x} ist differenzierbar.
- (ii) \mathbf{x} ist ein Homöomorphismus, d. h.
 - a) \mathbf{x} ist bijektiv,
 - b) \mathbf{x} ist stetig,
 - c) \mathbf{x}^{-1} ist stetig.
- (iii) Es gilt die Regularitätsbedingung: Für jedes $q \in U$ ist das Differential $d\mathbf{x}_q: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ injektiv.

Die Abbildung \mathbf{x} heißt *Parametrisierung* oder (lokales) *Koordinatensystem* in (einer Umgebung von) p . Die Umgebung $V \cap S$ von p in S heißt *Koordinatenumgebung* [9]. Es folgt nun eine kurze Erläuterung zu Definition 2.4. Zuerst soll Bedingung (iii) in ähnlicher Weise wie in [9, Kap. 2.2, S.43-45] erläutert werden. Die Matrix der linearen Abbildung $d\mathbf{x}_q$ wird in den kanonischen Basen $e_1 = (1, 0), e_2 = (0, 1)$ von \mathbb{R}^2 mit den Koordinaten (u, v) und in $f_1 = (1, 0, 0), f_2 = (0, 1, 0), f_3 = (0, 0, 1)$ von \mathbb{R}^3 mit den Koordinaten (x, y, z) berechnet. Es sei $q = (u_0, v_0)$. Der Vektor e_1 ist tangential zur Kurve $u \rightarrow (u, v_0)$, deren Bild unter \mathbf{x} die Kurve

$$u \rightarrow (x(u, v_0), y(u, v_0), z(u, v_0))$$

ist. Die Bildkurve (auch *Koordinatenkurve* $v = v_0$ genannt) liegt auf S und hat in $\mathbf{x}(q)$ den Tangentenvektor

$$\left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right) = \frac{\partial \mathbf{x}}{\partial u}$$

2. Mathematische Grundlagen

mit Ableitungen in (u_0, v_0) . Ein Vektor wird durch seine Komponenten bezüglich der Basis f_1, f_2, f_3 dargestellt. Es gilt für das Differential

$$d\mathbf{x}_q(e_1) = \left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right) = \frac{\partial \mathbf{x}}{\partial u}.$$

Analog erhält man für die andere Koordinatenkurve $u = u_0$

$$d\mathbf{x}_q(e_2) = \left(\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v} \right) = \frac{\partial \mathbf{x}}{\partial v}.$$

Die Matrix

$$d\mathbf{x}_q = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{pmatrix}$$

beschreibt die lineare Abbildung $d\mathbf{x}_q$ bezüglich der ausgewählten Basen. Bedingung (iii) aus Definition 2.4 bedeutet nun, dass die beiden Spaltenvektoren dieser Matrix linear unabhängig sind. Dadurch wird die Existenz einer Tangentialebene in allen Punkten von S garantiert. Durch Bedingung (ii) werden Selbstdurchschneidungen verhindert, folglich existiert genau eine Tangentialebene an $p \in S$ [9]. Eine Veranschaulichung der Definition 2.4 ist in Abbildung 2.2 zu sehen.

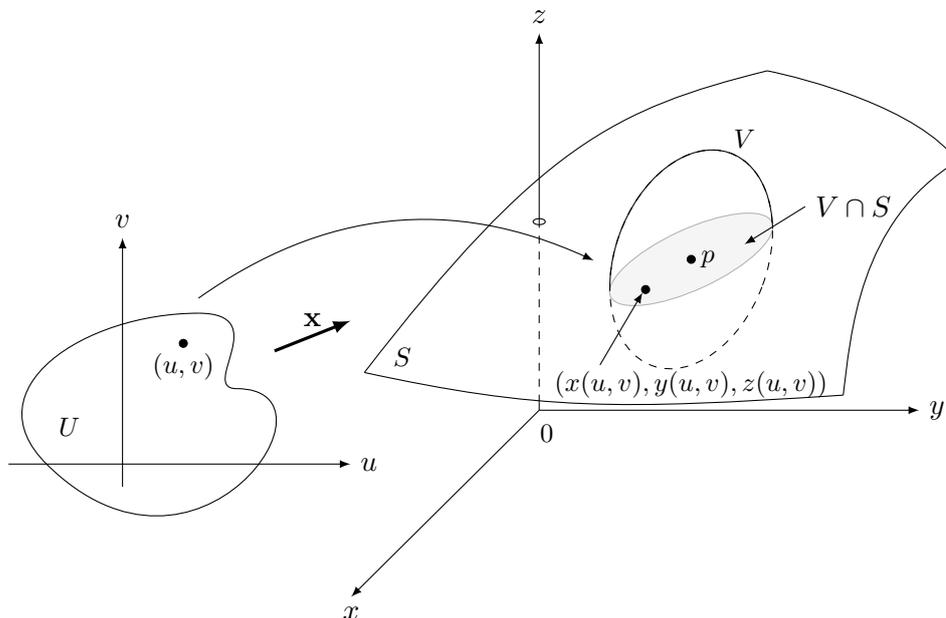


Abbildung 2.2: Bildliche Darstellung einer regulären Fläche nach Definition 2.4, Quelle: [9, Kap. 2.2, Bild 2.1].

An dieser Stelle soll im Hinblick auf Kapitel 5 auch auf den Diffeomorphismus eingegangen werden.

2.1. Grundlagen der Differentialgeometrie

Definition 2.5 (Diffeomorphismus, [22, Kap. 3.3, S. 104])

Es seien X und Y endlichdimensionale normierte \mathbb{K} -Vektorräume.

Eine bijektive C^1 -Abbildung $\Phi: U \rightarrow V$ einer offenen Menge $U \subset X$ auf eine offene Menge $V \subset Y$ heißt *Diffeomorphismus*, wenn die Umkehrung $\Phi^{-1}: V \rightarrow U$ ebenfalls eine C^1 -Abbildung ist.

Im folgenden Abschnitt soll der bereits genannte Begriff der Tangentialebene, ein zentraler Begriff der Differentialgeometrie, genauer erläutert werden. Die Tangentialebene wird über Tangentenvektoren definiert [9]. Ein *Tangentenvektor* an einer Fläche S soll hier – wie in [9] – als der Tangentenvektor $\alpha'(0)$ einer differenzierbaren parametrisierten Kurve $\alpha: (-\varepsilon, \varepsilon) \rightarrow S$ mit $\varepsilon > 0$ und $\alpha(0) = p$ verstanden werden.

Definition 2.6 (Tangentialebene, [9, Kap. 2.4, S.68/69])

Sei S eine reguläre Fläche und $p \in S$ ein Punkt in dieser Fläche. Eine *Tangentialebene* $T_p(S)$ ist eine Ebene, die durch den Punkt p geht und von den Tangentenvektoren an parametrisierte Kurven in S , die durch p gehen, aufgespannt wird.

Abbildung 2.3 veranschaulicht, wie eine Tangentialebene $T_p(S)$ einer regulären Fläche S an einem Punkt p gebildet wird.

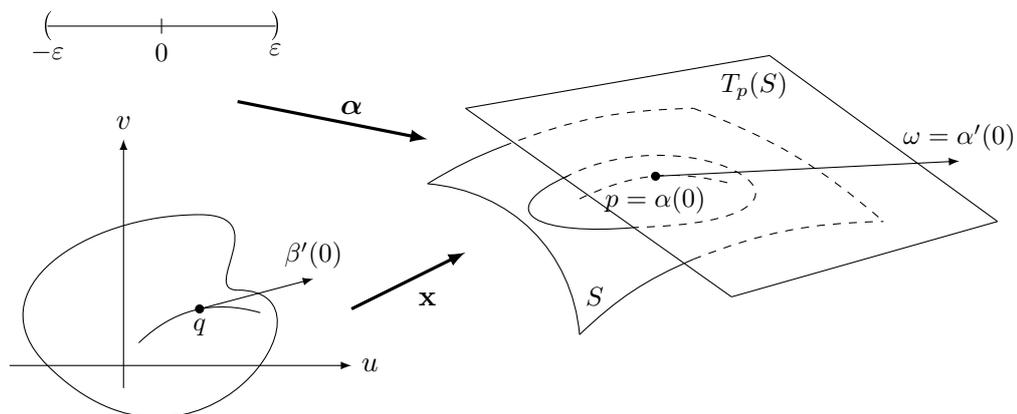


Abbildung 2.3: Bildliche Darstellung einer Tangentialebene $T_p(S)$ nach Definition 2.6, Quelle: [9, Kap. 2.4, Bild 2.23].

Der folgende Satz zeigt, dass die Tangentialebene nicht von der Parametrisierung \mathbf{x} abhängt.

Satz 2.1 (vgl. [9, Kap. 2.4, Prop. 1])

Sei $\mathbf{x}: U \subset \mathbb{R}^2 \rightarrow S$ eine Parametrisierung einer regulären Fläche S und $q \in U$. Dann stimmt der Untervektorraum der Dimension 2

$$d\mathbf{x}_q(\mathbb{R}^2) \subset \mathbb{R}^3$$

mit der Menge der Tangentenvektoren an S in $\mathbf{x}(q)$ überein.

2. Mathematische Grundlagen

Beweis. Siehe [9, Kap. 2.4, Prop. 1]. □

Mithilfe der Tangentialebene kann das Differential einer Abbildung zwischen Flächen definiert werden.

Definition 2.7 (Differential einer Abbildung, [9, Kap. 2.4, S.69/70])

Es seien S_1 und S_2 zwei reguläre Flächen und $\varphi: V \subset S_1 \rightarrow S_2$ eine differenzierbare Abbildung einer offenen Menge V von S_1 in S_2 . Falls $p \in V$ ist, ist jeder Tangentenvektor $\omega \in T_p(S_1)$ der Tangentenvektor $\alpha'(0)$ einer differenzierbaren parametrisierten Kurve $\alpha: (-\varepsilon, \varepsilon) \rightarrow V$ mit $\alpha(0) = p$.

Betrachtet man nun die Kurve $\beta = \varphi \circ \alpha$, so gilt $\beta(0) = \varphi(p)$. Damit ist $\beta'(0)$ ein Vektor in $T_{\varphi(p)}(S_2)$ und wird *Differential einer Abbildung* genannt.

In Abbildung 2.4 ist die in Definition 2.7 aufgezeigte Situation dargestellt.

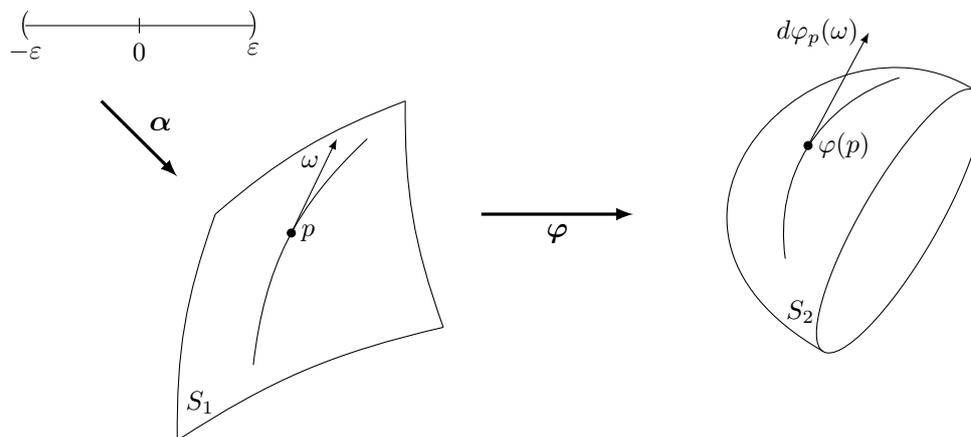


Abbildung 2.4: Bildliche Darstellung eines Differentials einer Abbildung nach Definition 2.7, Quelle: [9, Kap. 2.4, Bild 2.24].

Der folgende Satz erlaubt von der Linearität der Abbildung $d\varphi_p$, das heißt des Differentials von φ in $p \in S_1$, auszugehen.

Satz 2.2 (vgl. [9, Kap. 2.4, Prop. 2])

Sei $\omega \in T_p(S_1)$ gegeben, so hängt der Vektor $\beta'(0)$ nicht von der Wahl von α ab. Die Abbildung $d\varphi_p: T_p(S_1) \rightarrow T_{\varphi(p)}(S_2)$, definiert durch $d\varphi_p(\omega) = \beta'(0)$, ist linear.

Beweis. Siehe [9, Kap. 2.4, Prop. 2]. □

Zentrale Begriffe der Differentialgeometrie sind die Fundamentalformen. Zunächst soll hier auf die erste Fundamentalform eingegangen werden.

Definition 2.8 (Erste Fundamentalform, [9, Kap. 2.5, Def. 1])

Sei $\omega \in T_p(S) \subset \mathbb{R}^3$ ein Vektor der Tangentialebene einer regulären Fläche S . Das Standardskalarprodukt des \mathbb{R}^3 sei mit $\langle \cdot, \cdot \rangle_p$ bezeichnet. Die quadratische Form $I_p: T_p(S) \rightarrow \mathbb{R}$, die durch

$$I_p(\omega) = \langle \omega, \omega \rangle_p = |\omega|^2 \geq 0 \quad (2.1)$$

gegeben ist, heißt die *erste Fundamentalform* der regulären Fläche $S \subset \mathbb{R}^3$ in $p \in S$.

Mit Hilfe der ersten Fundamentalform können Messungen auf einer Fläche durchgeführt werden. Beispielsweise lassen sich Längen von Kurven auf regulären Flächen bestimmen, der Winkel zwischen Tangentialvektoren kann ermittelt und der Flächeninhalt von Gebieten auf regulären Flächen berechnet werden [9].

Die erste Fundamentalform kann in der zu einer Parametrisierung $\mathbf{x}(u, v)$ bei p assoziierten Basis $\{\mathbf{x}_u, \mathbf{x}_v\}$ von der Tangentialebene $T_p(S)$ ausgedrückt werden. Ein Tangentenvektor $\omega \in T_p(S)$ ist ein Tangentenvektor an eine parametrisierte Kurve $\boldsymbol{\alpha}(t) = \mathbf{x}(u(t), v(t))$, $t \in (-\varepsilon, \varepsilon)$, mit $p = \boldsymbol{\alpha}(0) = \mathbf{x}(u_0, v_0)$ [9]. Damit erhält man

$$\begin{aligned} I_p(\boldsymbol{\alpha}'(0)) &= \langle \boldsymbol{\alpha}'(0), \boldsymbol{\alpha}'(0) \rangle_p \\ &= \langle \mathbf{x}_u u' + \mathbf{x}_v v', \mathbf{x}_u u' + \mathbf{x}_v v' \rangle_p \\ &= \langle \mathbf{x}_u, \mathbf{x}_u \rangle_p (u')^2 + 2\langle \mathbf{x}_u, \mathbf{x}_v \rangle_p u' v' + \langle \mathbf{x}_v, \mathbf{x}_v \rangle_p (v')^2 \\ &= E(u')^2 + 2F u' v' + G(v')^2, \end{aligned}$$

wobei die Werte der auftretenden Funktionen für $t = 0$ berechnet werden. Es lassen sich über

$$\begin{aligned} E(u_0, v_0) &= \langle \mathbf{x}_u, \mathbf{x}_u \rangle_p, \\ F(u_0, v_0) &= \langle \mathbf{x}_u, \mathbf{x}_v \rangle_p \quad \text{und} \\ G(u_0, v_0) &= \langle \mathbf{x}_v, \mathbf{x}_v \rangle_p \end{aligned} \quad (2.2)$$

die *Koeffizienten der ersten Fundamentalform* in der Basis $\{\mathbf{x}_u, \mathbf{x}_v\}$ von $T_p(S)$ bestimmen.

2.1.3. Gauß-Abbildung

In diesem Abschnitt soll die Bestimmung der Krümmung einer Kurve, wie in Abschnitt 2.1.1 erläutert, auf reguläre Flächen übertragen werden. Die Idee ist es zu messen, wie schnell sich eine Fläche S von der Tangentialebene $T_p(S)$ in einer Umgebung eines Punktes $p \in S$ entfernt. Das bedeutet, es muss die Veränderungsrate eines Einheitsnormalenvektorfeldes N auf einer Umgebung von p bezüglich der Stelle p gemessen werden. Dazu werden einige Begrifflichkeiten benötigt.

Zu einer gegebenen Parametrisierung $\mathbf{x}: U \subset \mathbb{R}^2 \rightarrow S$ einer regulären Fläche S bei einem Punkt $p \in S$ kann in jedem Punkt aus $\mathbf{x}(U)$ ein *Einheitsnormalenvektor* durch

$$N(q) = \frac{\mathbf{x}_u \times \mathbf{x}_v}{|\mathbf{x}_u \times \mathbf{x}_v|}(q), \quad q \in \mathbf{x}(U) \quad (2.3)$$

2. Mathematische Grundlagen

bestimmt werden. Es gibt also eine differenzierbare Abbildung $N: \mathbf{x}(U) \rightarrow \mathbb{R}^3$, die jedem $q \in \mathbf{x}(U)$ einen Einheitsnormalenvektor $N(q)$ zuordnet [9]. Darüber lässt sich das Einheitsnormalenvektorfeld definieren.

Definition 2.9 (Einheitsnormalenvektorfeld, [9, Kap. 3.2, S.99])

Sei $V \subset S$ eine offene Menge in S und $N: V \rightarrow \mathbb{R}^3$ eine differenzierbare Abbildung, die jedem $q \in V$ einen Einheitsnormalenvektor in q zuordnet. Dann ist N ein *differenzierbares Einheitsnormalenvektorfeld auf V* .

Über das Einheitsnormalenvektorfeld lässt sich eine Aussage über die Orientierung einer regulären Fläche machen.

Definition 2.10 (Orientierung, [9, Kap. 3.2, S.99])

Eine reguläre Fläche S heißt *orientierbar*, wenn es ein auf der ganzen Fläche definiertes differenzierbares Einheitsnormalenvektorfeld gibt. Die Wahl eines solchen Vektorfeldes N heißt *Orientierung* von S .

Nicht alle Flächen haben ein differenzierbares Einheitsnormalenvektorfeld, das auf der ganzen Fläche definiert ist. Ein Beispiel dafür ist das Möbiusband, auf dem kein solches Vektorfeld definiert werden kann. Anschaulich betrachtet, geht man einmal längs des Mittelkreises der Fläche herum. Nach einer Umdrehung kommt das Vektorfeld N als $-N$ zurück, was ein Widerspruch zur Stetigkeit von N ist. Folglich ist das Möbiusband auch keine orientierbare Fläche [9].

Im Folgenden bezeichnet S eine reguläre, orientierbare Fläche, für die eine Orientierung gewählt worden ist. Man spricht dann von einer Fläche S mit Orientierung N .

Definition 2.11 (Gauß-Abbildung, [9, Kap. 3.2, Def. 1])

Sei $S \subset \mathbb{R}^3$ eine Fläche mit einer Orientierung N und S^2 die Einheitssphäre mit $S^2 = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1\}$. Die Abbildung $N: S \rightarrow S^2$, die von der Fläche S in die Einheitssphäre abbildet, heißt die *Gauß-Abbildung* von S .

Eine Veranschaulichung der Gauß-Abbildung ist in Abbildung 2.5 zu sehen.

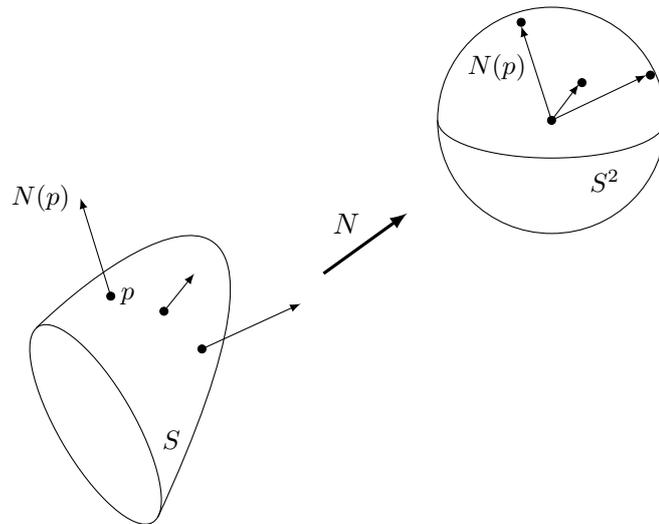


Abbildung 2.5: Bildliche Darstellung der Gauß-Abbildung nach Definition 2.11, Quelle: [9, Kap. 3.2, Bild 3.2].

Die Gauß-Abbildung ist differenzierbar [9, Kap. 3.2, S.100]. Das Differential dN_p von N bei $p \in S$ ist eine lineare Abbildung (siehe Satz 2.3) von $T_p(S)$ in $T_{N(p)}(S^2)$. Die Tangentialebenen $T_p(S)$ und $T_{N(p)}(S^2)$ sind parallele Ebenen. Aus diesem Grund kann dN_p als eine lineare Abbildung auf $T_p(S)$ aufgefasst werden.

Satz 2.3 (vgl. [9, Kap. 3.2, Prop. 1])

Das Differential $dN_p: T_p(S) \rightarrow T_p(S)$ der Gauß-Abbildung ist eine selbstadjungierte lineare Abbildung.

Beweis. Siehe [9, Kap. 3.2, Prop. 1]. □

Aufgrund dieses Satzes kann dem Differential dN_p eine quadratische Form Q auf $T_p(S)$ zugeordnet werden. Die quadratische Form ist durch $Q(v) = \langle dN_p(v), v \rangle_p$ mit $v \in T_p(S)$ gegeben. Dies führt direkt auf die nächste Definition.

Definition 2.12 (Zweite Fundamentalform, [9, Kap. 3.2, Def. 2])

Sei $T_p(S)$ die Tangentialebene einer regulären Fläche S , $v \in T_p(S)$ und dN_p das Differential der Gauß-Abbildung bei $p \in S$. Die quadratische Form II_p , definiert auf $T_p(S)$ durch $II_p(v) = -\langle dN_p(v), v \rangle_p$, heißt die *zweite Fundamentalform* von S bei p .

Die bisher gemachten Definitionen beziehen sich auf kein Koordinatensystem. Für einfache Fragestellungen ist dies ausreichend, doch um allgemeine Situationen behandeln zu können, ist es oft hilfreich, auf ein Koordinatensystem Bezug zu nehmen. Dies soll hier analog zu [9, Kap. 3.3, S.113-115] geschehen. Die folgenden Parametrisierungen $\mathbf{x}: U \subset \mathbb{R}^2 \rightarrow S$ werden als mit der Orientierung N von S verträglich

2. Mathematische Grundlagen

vorausgesetzt, das heißt, es gilt in $\mathbf{x}(U)$

$$N = \frac{\mathbf{x}_u \times \mathbf{x}_v}{|\mathbf{x}_u \times \mathbf{x}_v|}.$$

Es sei $\mathbf{x}(u, v)$ eine Parametrisierung einer Fläche S bei einem Punkt $p \in S$ und $\boldsymbol{\alpha}(t) = \mathbf{x}(u(t), v(t))$ eine parametrisierte Kurve auf S mit $\boldsymbol{\alpha}(0) = p$. Zur Vereinfachung der Notation sollen im Folgenden alle auftretenden Funktionen für ihren Wert im Punkt p stehen. Der Tangentenvektor an $\boldsymbol{\alpha}(t)$ in p ist $\boldsymbol{\alpha}' = \mathbf{x}_u u' + \mathbf{x}_v v'$ und

$$dN(\boldsymbol{\alpha}') = N'(u(t), v(t)) = N_u u' + N_v v'.$$

N_u und N_v liegen in $T_p(S)$. Deswegen kann man

$$N_u = a_{11}\mathbf{x}_u + a_{21}\mathbf{x}_v \quad \text{und} \quad N_v = a_{12}\mathbf{x}_u + a_{22}\mathbf{x}_v$$

schreiben, somit gilt

$$dN(\boldsymbol{\alpha}') = (a_{11}u' + a_{12}v')\mathbf{x}_u + (a_{21}u' + a_{22}v')\mathbf{x}_v. \quad (2.4)$$

In Matrixschreibweise lautet dies

$$dN \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} u' \\ v' \end{pmatrix}.$$

Das Differential dN kann also in der Basis $\{\mathbf{x}_u, \mathbf{x}_v\}$ durch die Matrix $(a_{ij})_{i,j=1,2}$ dargestellt werden mit $a_{ij} \in \mathbb{R}$. Die Darstellung der zweiten Fundamentalform in der Basis $\{\mathbf{x}_u, \mathbf{x}_v\}$ lautet

$$\begin{aligned} II_p(\boldsymbol{\alpha}') &= -\langle dN(\boldsymbol{\alpha}'), \boldsymbol{\alpha}' \rangle \\ &= -\langle N_u u' + N_v v', \mathbf{x}_u u' + \mathbf{x}_v v' \rangle \\ &= e(u')^2 + 2f u' v' + g(v')^2, \end{aligned}$$

wobei wegen $\langle N, \mathbf{x}_u \rangle = \langle N, \mathbf{x}_v \rangle = 0$

$$\begin{aligned} e &= -\langle N_u, \mathbf{x}_u \rangle = \langle N, \mathbf{x}_{uu} \rangle, \\ f &= -\langle N_v, \mathbf{x}_u \rangle = \langle N, \mathbf{x}_{uv} \rangle = \langle N, \mathbf{x}_{vu} \rangle = -\langle N_u, \mathbf{x}_v \rangle, \\ g &= -\langle N_v, \mathbf{x}_v \rangle = \langle N, \mathbf{x}_{vv} \rangle \end{aligned} \quad (2.5)$$

gilt. Aus den Gleichungen (2.4) und (2.5) können die Darstellungen für die a_{ij} mithilfe der Koeffizienten der ersten Fundamentalform aus Gleichung (2.2) bezüglich der Basis $\{\mathbf{x}_u, \mathbf{x}_v\}$ hergeleitet werden,

$$\begin{aligned} -f &= \langle N_u, \mathbf{x}_v \rangle = a_{11}F + a_{21}G, \\ -f &= \langle N_v, \mathbf{x}_u \rangle = a_{12}E + a_{22}F, \\ -e &= \langle N_u, \mathbf{x}_u \rangle = a_{11}E + a_{21}F, \\ -g &= \langle N_v, \mathbf{x}_v \rangle = a_{12}F + a_{22}G. \end{aligned} \quad (2.6)$$

Die Beziehungen in (2.6) können in Matrixschreibweise ausgedrückt werden als

$$-\begin{pmatrix} e & f \\ f & g \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}.$$

Durch Umschreiben erhält man eine explizite Formel für a_{ij}

$$\begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} = -\begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1},$$

wobei mit $(\cdot)^{-1}$ die inverse Matrix bezeichnet wird. Für die einzelnen Elemente $a_{ij} \in \mathbb{R}$ in der Matrix erhält man

$$\begin{aligned} a_{11} &= \frac{fF - eG}{EG - F^2}, \\ a_{12} &= \frac{gF - fG}{EG - F^2}, \\ a_{21} &= \frac{eF - fE}{EG - F^2}, \\ a_{22} &= \frac{fF - gE}{EG - F^2}. \end{aligned} \tag{2.7}$$

Die Gleichungen in (2.7) werden als *Gleichungen von Weingarten* bezeichnet [9].

Ein Begriff, auf dem die Gaußsche Krümmung und die mittlere Krümmung beruhen, ist die Normalkrümmung, die im Folgenden definiert werden soll.

Definition 2.13 (Normalkrümmung, [9, Kap. 3.2, Def. 3])
 Sei C eine reguläre Kurve in S , die durch $p \in S$ geht, κ die Krümmung von C in p und $\cos \theta = \langle n, N \rangle_p$, wobei n der Normalenvektor an C und N der Normalenvektor auf S in p ist. Die Zahl $\kappa_n = \kappa \cos \theta$ heißt die *Normalkrümmung* von $C \subset S$ in p .

Es soll nun nochmal auf die lineare Abbildung dN_p eingegangen werden. Dazu wird zuerst folgendes Theorem formuliert.

Theorem 2.1 (vgl. [8, Appendix zu Kap. 3, Theorem S.216])
 Sei $A: V \rightarrow V$ eine selbstadjungierte lineare Abbildung. Dann existiert eine Orthonormalbasis $\{e_1, e_2\}$ von V , sodass gilt $A(e_1) = \lambda_1 e_1, A(e_2) = \lambda_2 e_2$. Es seien mit e_1, e_2 Eigenvektoren und mit λ_1, λ_2 Eigenwerte bezeichnet. In der Basis $\{e_1, e_2\}$ sei die Matrix A eine Diagonalmatrix. Die Elemente λ_1, λ_2 auf der Diagonalen mit $\lambda_1 \geq \lambda_2$ sind das Maximum und das Minimum bezüglich der quadratischen Form $Q(v) = \langle Av, v \rangle$ auf dem Einheitskreis von V .

Beweis. Siehe [8, Appendix von Kap. 3, S. 216]. □

2. Mathematische Grundlagen

Angewandt auf die Abbildung dN_p bedeutet das, dass es zu jedem $p \in S$ eine Orthonormalbasis $\{e_1, e_2\}$ von $T_p(S)$ gibt, sodass $dN_p(e_1) = -\kappa_1 e_1$ und $dN_p(e_2) = -\kappa_2 e_2$ gilt. Zudem sind κ_1 und κ_2 , mit $\kappa_1 \geq \kappa_2$, das Maximum und Minimum der zweiten Fundamentalform II_p , eingeschränkt auf den Einheitskreis in $T_p(S)$. Damit sind sie die Extremwerte der Normalkrümmung bei p . Damit lassen sich nun die Hauptkrümmungen definieren.

Definition 2.14 (Hauptkrümmungen, [9, Kap. 3.2, Def. 4])

Die maximale Normalkrümmung κ_1 und die minimale Normalkrümmung κ_2 heißen *Hauptkrümmungen* bei $p \in \mathbb{R}^3$. Die dazugehörigen Richtungen, die durch die Eigenvektoren e_1, e_2 des Differentials dN_p gegeben sind, heißen *Hauptkrümmungsrichtungen* bei p .

Mithilfe der beiden Hauptkrümmungen werden die Gaußsche Krümmung und die mittlere Krümmung definiert. Dazu betrachte man einerseits die Determinante von dN , die über das Produkt der Hauptkrümmungen gebildet wird, was zu $(-\kappa_1)(-\kappa_2) = \kappa_1 \kappa_2$ führt. Andererseits ist die Spur von dN die Summe der Hauptkrümmungen und damit $-\kappa_1 + (-\kappa_2) = -(\kappa_1 + \kappa_2)$. Es sei hier angemerkt, dass, falls sich die Orientierung der Fläche ändert, die Determinante gleich bleibt, die Spur jedoch ihr Vorzeichen ändert.

Definition 2.15 (Gaußsche und mittlere Krümmung, [9, Kap. 3.2, Def. 6])

Sei $p \in S$ ein Punkt auf einer regulären Fläche S , $dN_p: T_p(S) \rightarrow T_p(S)$ das Differential der Gauß-Abbildung.

Die *Gaußsche Krümmung* K von S in p ist die Determinante von dN_p :

$$K = \kappa_1 \kappa_2. \quad (2.8)$$

Die *mittlere Krümmung* H von S in p ist das Negative der halben Spur:

$$H = \frac{\kappa_1 + \kappa_2}{2}. \quad (2.9)$$

Die beiden Krümmungen können über die Koeffizienten der ersten und der zweiten Fundamentalform ausgedrückt werden. Die Gaußsche Krümmung ist die Determinante der oben erwähnten Matrix a_{ij} . Durch einfaches Nachrechnen erhält man

$$K = \det(a_{ij}) = \frac{eg - f^2}{EG - F^2} \quad (2.10)$$

für die Gaußsche Krümmung. Die mittlere Krümmung kann genauso über die Koeffizienten der ersten und zweiten Fundamentalformen dargestellt werden. Dazu ist wichtig, dass die Hauptkrümmungen $-\kappa_1, -\kappa_2$ die Eigenwerte von dN sind. Die Krümmungen $-\kappa_1, -\kappa_2$ erfüllen also die Gleichung

$$dN(v) = -\kappa v = -\kappa I v \text{ für ein } v \in T_p(S), v \neq 0,$$

mit I der identischen Abbildung. Die Abbildung $dN + \kappa I$ ist demnach nicht invertierbar, das heißt, die Determinante dieser Abbildung ist Null. Damit gilt

$$\det \begin{pmatrix} a_{11} + \kappa & a_{12} \\ a_{21} & a_{22} + \kappa \end{pmatrix} = 0$$

oder

$$\kappa^2 + \kappa(a_{11} + a_{22}) + a_{11}a_{22} - a_{21}a_{12} = 0.$$

Mit den Lösungen dieser quadratischen Gleichung, κ_1 und κ_2 , folgt für die mittlere Krümmung

$$H = \frac{1}{2}(\kappa_1 + \kappa_2) = -\frac{1}{2}(a_{11} + a_{22}) = \frac{1}{2} \frac{eG - 2fF + gE}{EG - F^2}. \quad (2.11)$$

Den Zusammenhang zwischen den Koeffizienten der ersten und zweiten Fundamentalförmung und den Krümmungen, soll folgendes Beispiel verdeutlichen.

Beispiel 2.1 (vgl. [9, Kap. 3.3, Beispiel 5])

Sei eine Fläche S als Graph einer differenzierbaren Funktion $z = h(x, y)$ gegeben, wobei (x, y) in einer offenen Menge $U \subset \mathbb{R}^2$ liegt. Um Formeln für die oben definierten Begriffe zu erhalten, wird die Fläche durch

$$\mathbf{x}(u, v) = (u, v, h(u, v)), \quad (u, v) \in U,$$

mit $x = u, y = v$ parametrisiert. Für die benötigten Ableitungen gelten

$$\begin{aligned} \mathbf{x}_u &= (1, 0, h_u), \quad \mathbf{x}_v = (0, 1, h_v), \\ \mathbf{x}_{uu} &= (0, 0, h_{uu}), \quad \mathbf{x}_{uv} = (0, 0, h_{uv}), \quad \mathbf{x}_{vv} = (0, 0, h_{vv}). \end{aligned}$$

Für $N(x, y)$ gilt also

$$N(x, y) = \frac{\mathbf{x}_u \times \mathbf{x}_v}{|\mathbf{x}_u \times \mathbf{x}_v|} = \frac{(-h_x, -h_y, 1)}{\sqrt{1 + h_x^2 + h_y^2}},$$

was ein Einheitsnormalenfeld auf der Fläche ist. Die Koeffizienten der zweiten Fundamentalförmung sind in dieser Orientierung durch

$$\begin{aligned} e &= \frac{h_{xx}}{\sqrt{1 + h_x^2 + h_y^2}}, \\ f &= \frac{h_{xy}}{\sqrt{1 + h_x^2 + h_y^2}}, \\ g &= \frac{h_{yy}}{\sqrt{1 + h_x^2 + h_y^2}} \end{aligned} \quad (2.12)$$

2. Mathematische Grundlagen

gegeben. Die Formeln der Gaußschen Krümmung (2.10) und der mittleren Krümmung (2.11) können nun mithilfe der Koeffizienten der ersten Fundamentalförm (2.2) und den gerade gemachten Darstellungen der zweiten Fundamentalförm (2.12) durch Einsetzen auf folgende Form gebracht werden:

$$K = \frac{h_{xx}h_{yy} - h_{xy}^2}{(1 + h_x^2 + h_y^2)^2}, \quad (2.13)$$

$$H = \frac{1}{2} \frac{(1 + h_x^2)h_{yy} - 2h_xh_yh_{xy} + (1 + h_y^2)h_{xx}}{(1 + h_x^2 + h_y^2)^{3/2}}. \quad (2.14)$$

Diese Formeln sind für den Regularisierer, der im Kapitel 3 vorgestellt wird, von großer Bedeutung.

2.2. Funktionalanalytische Grundlagen

Dieses Kapitel widmet sich den Grundlagen der Analysis und der Variationsrechnung. Aus der Analysis soll hier auf den Gaußschen Integralsatz eingegangen und die Variationsrechnung bis zu Euler-Lagrange-Gleichungen aufgearbeitet werden. Diese Begriffe werden für die theoretische Analyse des in dieser Arbeit zu untersuchenden Regularisierers wichtig sein. Für elementare Begriffe der Analysis sei an dieser Stelle auf die Lehrbücher von Forster [13, 14, 15] verwiesen.

In diesem Abschnitt sollen nun Begriffe aus der Analysis aufgeführt werden, die für diese Arbeit von besonderer Bedeutung sind.

In der Integralrechnung im \mathbb{R}^n ist nach [22] einer der wichtigsten Sätze der *Gaußsche Integralsatz*. Dieser Satz ist das höherdimensionale Analogon des Fundamentalsatzes der Integral- und Differentialrechnung für Funktionen einer Veränderlichen [22]. Bevor der Satz hier formuliert wird, soll an den Begriff der Divergenz eines Vektorfeldes erinnert werden. Sei $U \subset \mathbb{R}^n$ offen und

$$F = (F_1, \dots, F_n): U \rightarrow \mathbb{R}^n$$

ein stetig differenzierbares Vektorfeld. Dann ist die *Divergenz* von F eine Funktion $\operatorname{div}F: U \rightarrow \mathbb{R}$, die durch

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

definiert ist (vgl. [15]). Damit kann nun der Gaußsche Integralsatz formuliert werden.

Satz 2.4 (Gaußscher Integralsatz, [15, Kap. 15, Satz 3])

Sei $\Omega \subset \mathbb{R}^n$ eine kompakte Teilmenge mit glattem Rand. Bezeichne $\nu: \partial\Omega \rightarrow \mathbb{R}^n$ das äußere Einheitsnormalenfeld und sei $U \supset \Omega$ eine offene Teilmenge von \mathbb{R}^n . Dann gilt für jedes stetig differenzierbare Vektorfeld $F: U \rightarrow \mathbb{R}^n$

$$\int_{\Omega} \operatorname{div}F(x) \, d^n x = \int_{\partial\Omega} \langle F(x), \nu(x) \rangle \, dS(x). \quad (2.15)$$

Beweis. Siehe [15, Kap. 15, S. 185/186]. □

Aus dem Integralsatz von Gauß lässt sich die Formel für die mehrdimensionale partielle Integration ableiten. Sei hierfür $F(x) = G(x) \cdot \varphi(x)$ mit $G(x) \in \mathbb{R}^n$ und $\varphi(x) \in \mathbb{R}$. Damit erhält Gleichung (2.15) folgende Form:

$$\int_{\partial\Omega} \langle G(x) \cdot \varphi(x), \nu(x) \rangle dS(x) = \int_{\Omega} \operatorname{div}(G(x) \cdot \varphi(x)) dx. \quad (2.16)$$

Es soll nun die Notation vereinfacht werden, indem folgende Schreibweise eingeführt wird: $G := G(x)$, $\varphi := \varphi(x)$ und $\nu := \nu(x)$. Im Folgenden wird der Term $\operatorname{div}(G \cdot \varphi)$ umgeschrieben:

$$\begin{aligned} \operatorname{div}(G \cdot \varphi) &= \operatorname{div} \begin{pmatrix} G_1 \cdot \varphi \\ \vdots \\ G_n \cdot \varphi \end{pmatrix} = \begin{pmatrix} \partial_1(G_1 \cdot \varphi) \\ \vdots \\ \partial_n(G_n \cdot \varphi) \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \partial_1 G_1 \cdot \varphi + G_1 \cdot \partial_1 \varphi \\ \vdots \\ \partial_n G_n \cdot \varphi + G_n \cdot \partial_n \varphi \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \\ &= \varphi \cdot \operatorname{div} G + \langle G, \nabla \varphi \rangle. \end{aligned}$$

Setzt man nun diesen Term in das zweite Integral der Gleichung (2.16) ein und teilt das Integral in zwei Integrale auf, so erhält man

$$\int_{\partial\Omega} \langle G(x) \cdot \varphi(x), \nu(x) \rangle dS(x) = \int_{\Omega} (\operatorname{div} G) \cdot \varphi dx + \int_{\Omega} \langle G, \nabla \varphi \rangle dx.$$

Betrachtet man nun den Spezialfall $G = \nabla u$, so erhält man

$$\int_{\Omega} \langle \nabla u, \nabla \varphi \rangle dx = - \int_{\Omega} \Delta u \cdot \varphi dx + \int_{\partial\Omega} \langle \varphi \cdot \nabla u, \nu \rangle dS(x),$$

was der allgemeinen Formel der mehrdimensionalen partiellen Integration entspricht und im Verlauf dieser Arbeit mehrmals angewandt wird.

Im Folgenden soll nun auf die Grundlagen der Variationsrechnung eingegangen werden. Es werden grundlegende Begriffe und Definitionen, die zum Beispiel in [17] und [25] nachgelesen werden können, genannt, die im weiteren Verlauf der Arbeit verwendet werden.

Die Variationsrechnung befasst sich mit Funktionalen, das heißt mit Abbildungen, die aus einem Gebiet V , das eine Teilmenge eines linearen Raumes X aus \mathbb{R} ist, in die reellen Zahlen \mathbb{R} abbildet [17]. Die betrachteten Funktionale in der Variationsrechnung sollen im Folgenden als Variationsintegrale definiert werden.

Definition 2.16 (Variationsintegral, [17, Kap. 1, S. 3])

Sei $\Omega \subset \mathbb{R}^n$, $n \geq 1$, und $u(x)$ eine in Ω definierte Funktion, die nach \mathbb{R}^N , $N \geq 1$, abbildet. Mit Du sei die erste Ableitung von u bezeichnet, das heißt $Du = (D_\alpha u^i)$, $i = 1, \dots, N$, $\alpha = 1, \dots, n$. Das Variationsintegral sei dann definiert über

$$\mathcal{F}(u) = \int_{\Omega} F(x, u(x), Du(x)) dx.$$

2. Mathematische Grundlagen

Mit den Begriffen Funktional und Variationsintegral lässt sich die erste Variation einführen. Dazu soll hier so vorgegangen werden wie in [17]. Man wähle einen Punkt $u_0 \in V$ und einen Vektor $\zeta \in X$ und nimmt an, dass das durch u_0 und ζ beschriebene Segment $\{u: u = u_0 + \varepsilon\zeta, |\varepsilon| < \varepsilon_0\}$ in V für $\varepsilon_0 > 0$ enthalten ist. Dann heißt

$$\delta\mathcal{F}(u_0, \zeta) := \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} [\mathcal{F}(u_0 + \varepsilon\zeta) - \mathcal{F}(u_0)] = \left. \frac{d}{d\varepsilon} \mathcal{F}(u_0 + \varepsilon\zeta) \right|_{\varepsilon=0}$$

erste Variation oder *Gâteaux-Differential* von \mathcal{F} bei u_0 in Richtung ζ , falls der Grenzwert existiert. Hieran können nun Bedingungen gestellt werden. Eine der wichtigsten Bedingungen in der Variationsrechnung ist das Verschwinden der ersten Variation, was zur *Euler-Lagrange-Gleichung* führt. Das Verschwinden der ersten Variation kann nach [17] folgendermaßen beschrieben werden. Falls u_0 das Funktional \mathcal{F} bezüglich der Variationen $\delta u_0 = \zeta$ minimiert, so gilt

$$\delta\mathcal{F}(u_0, \zeta) := \left. \frac{d}{d\varepsilon} \mathcal{F}(u_0 + \varepsilon\zeta) \right|_{\varepsilon=0} = 0$$

für alle ζ mit kompaktem Träger in Ω . Die Variationsrechnung ist das Teilgebiet der Mathematik, das sich mit dem Finden von Extrema von Funktionalen befasst. Aus diesem Grund geht es im Folgenden nun darum, kritische Punkte von \mathcal{F} wie Extrempunkte oder Sattelpunkte zu finden. Dafür betrachtet man folgenden Satz.

Satz 2.5 (Euler-Lagrange-Gleichung, [17, Kap. 1, S. 16])

Sei $u \in C^2(\Omega, \mathbb{R}^N)$. Wenn $u(x)$ die Gleichung

$$D_\alpha F_{p_\alpha^i}(x, u(x), Du(x)) - F_{u^i}(x, u(x), Du(x)) = 0, \quad 1 \leq i \leq N, \quad (2.17)$$

für alle $x \in \Omega$ erfüllt, so ist ein stationärer Punkt gefunden worden. Diese Gleichung bezeichnet man als *Euler-Lagrange-Gleichung*.

Hierbei bezeichnet $F_u(x, u, p)$ die partielle Ableitung und $F_{p_\alpha^i}$ ist die abkürzende Bezeichnung für $\frac{\partial F}{\partial u_{x^\alpha}^i}$. Für weitere Erläuterungen sei auf [17, Kap. 1, S. 11/12] verwiesen.

Beweis. Siehe [17, Kap. 1, S. 16]. □

Mit dem Ziel Extrempunkte zu finden, befasst sich auch die Bildregistrierung. Auf diesen Anwendungsbereich der Mathematik soll im folgenden Kapitel eingegangen werden.

2.3. Grundlagen der Bildregistrierung

In diesem Kapitel erfolgt eine kurze Einführung in das Thema der Bildregistrierung. Es werden grundlegende Begrifflichkeiten eingeführt und das Registrierungsproblem der nicht-parametrischen Bildregistrierung dargestellt. Diese Grundlagen stammen aus den Werken [26] und [27].

2.3.1. Problemstellung

In der Bildregistrierung werden Bilder betrachtet. Ein Bild soll im Folgenden als eine kontinuierliche Abbildung $T: \mathbb{R}^n \rightarrow \mathbb{R}^n$ verstanden werden, wobei $n \in \mathbb{N}$ die räumliche Dimension der gegebenen Daten angibt [26]. Die Bildregistrierung beschäftigt sich mit zwei Bildern, einem Referenzbild R und einem Templatebild T . Gesucht wird eine Transformation $\varphi: \mathbb{R}^n \rightarrow \mathbb{R}^n$ mit

$$\varphi(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x}),$$

die von einem unbekanntem Deformationsfeld $\mathbf{u}, \mathbf{u}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ abhängt, sodass das transformierte Templatebild $T_\varphi(\mathbf{x}) := T(\varphi(\mathbf{x}))$ ähnlich zum Referenzbild R wird. Um das Registrierungsproblem zu lösen, muss nun ein geeignetes Distanzmaß angewandt werden, um die Differenz zwischen R und T bezüglich \mathbf{u} zu minimieren,

$$\mathcal{D}[R, T; \mathbf{u}] \xrightarrow{\mathbf{u}} \min. \quad (2.18)$$

Dieses Problem ist nach Hadamard [18] ein schlecht gestelltes Problem. Bei einem korrekt gestellten Problem muss gewährleistet sein, dass das Problem eine Lösung besitzt, dass diese Lösung eindeutig ist und dass die Lösung stetig von den Eingabewerten abhängt. Ist eine dieser Bedingungen nicht erfüllt, so handelt es sich um ein schlecht gestelltes Problem. Das Registrierungsproblem (2.18) ist schlecht gestellt, da kleine Veränderungen der Eingabedaten zu großen Veränderungen der Ausgabe führen können. Außerdem ist die Lösung nicht eindeutig, da das Problem nicht unbedingt konvex ist und die Deformation nicht kontinuierlich sein muss [27]. Aus diesem Grund muss eine a priori Information an \mathbf{u} über einen Regularisierungsterm \mathcal{R} gestellt werden. Das Minimierungsproblem kann damit folgendermaßen erweitert werden

$$\min_{\mathbf{u} \in \mathbb{R}^n} \{ \mathcal{J}_\alpha[\mathbf{u}] = \mathcal{D}[R, T; \mathbf{u}] + \alpha \mathcal{R}[\mathbf{u}] \}. \quad (2.19)$$

Der Parameter α sei größer als Null und \mathbf{u} wird in einer Menge \mathcal{U} von zulässigen Lösungen, die \mathcal{J}_α minimieren, gesucht.

2.3.2. Diskretisierung

In der Theorie können Daten kontinuierlich betrachtet werden, während dies in der Praxis aufgrund von begrenztem Speicherplatz und endlicher Zeit nicht möglich ist. Aus diesem Grund werden die Daten durch Abtastung diskretisiert. Dies erfolgt, indem den gegebenen Daten Punkte zugeordnet werden, die sich auf einem Gitter eines n -dimensionalen Intervalls $\Omega = (\omega_1, \omega_2) \times \dots \times (\omega_{2n-1}, \omega_{2n}) \subset \mathbb{R}^n$ befinden. Das Standardgitter ist das zell-zentrierte Gitter, bei dem über die Zellmittelpunkte interpoliert wird. Darüber hinaus existieren das nodale Gitter und das gestaffelte Gitter, bei denen über die Eckpunkte beziehungsweise über die Mittelpunkte der Kanten

2. Mathematische Grundlagen

interpoliert wird. Da in dieser Arbeit das zell-zentrierte Gitter verwendet wurde, soll im Folgenden auf dieses genauer eingegangen werden. Für genauere Ausführungen der anderen Gitterformen wird auf [26] verwiesen.

Definition 2.17 (Zell-zentriertes Gitter, [26, Kap. 3.1, S. 20])

Sei $\Omega = (\omega_1, \omega_2) \times \dots \times (\omega_{2n-1}, \omega_{2n}) \subset \mathbb{R}^n$ das Bildgebiet und $m = [m_1, \dots, m_n]$ die Datengröße. Die Gitterweite lässt sich in einem Vektor $h = [h_1, \dots, h_n]$ mit

$$h_i = \frac{(\omega_{2i} - \omega_{2i-1})}{m_i}$$

zusammenfassen. Sei $j = [j_1, \dots, j_n]$ ein Multiindex, dann definiert man

$$\xi_j^i = \omega_{2i-1} + (j - 0.5)h_i, \text{ wobei } \xi^i = [\xi_1^i, \dots, \xi_{m_i}^i] \in \mathbb{R}^{m_i}.$$

Diese Punkte werden in einem Vektor x_j folgendermaßen

$$x_j = [\xi_{j_1}^1, \dots, \xi_{j_n}^n], \text{ mit } j_i = 1, \dots, m_i \text{ und } i = 1, \dots, n,$$

zusammengefasst, der dann die Punkte des gesamten zell-zentrierten Gitters enthält.

Für die praktische Verwendung ist es wichtig, dass der Vektor in einer handhabbaren Form geordnet ist. In dieser Arbeit wurde die lexikographische Anordnung verwendet. Die Umsetzung dieser Sortierung kann, wie in [26] beschrieben, im Zweidimensionalen über das Schema $j \mapsto j_1 + m_1(j_2 - 1)$ erfolgen. Eine Veranschaulichung des zell-zentrierten Gitters und der lexikographischen Ordnung der Gitterpunkte ist in Abbildung 2.6 zu sehen.

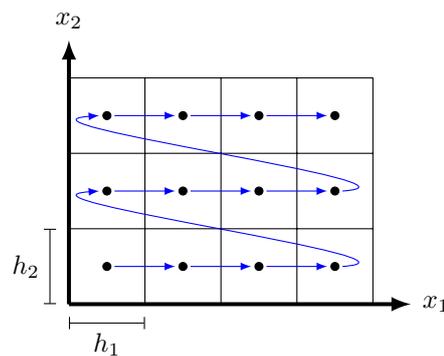


Abbildung 2.6: Darstellung der Diskretisierung eines zweidimensionalen Bildgebietes über ein zell-zentriertes Gitter mit $m_1 = 4$ und $m_2 = 3$. Die Pfeile zeigen die lexikographische Anordnung der Gitterpunkte an.

Um die Implementierung einer Diskretisierung effizienter zu gestalten, ist es oftmals sinnvoll, das Kroneckerprodukt zu verwenden, das folgendermaßen definiert ist.

Definition 2.18 (Kroneckerprodukt, [26, Kap. 2.3.8, S. 17])

Gegeben seien zwei Matrizen $A \in \mathbb{R}^{m \times n}$ und $B \in \mathbb{R}^{p \times q}$.

Das *Kroneckerprodukt* ist dann definiert über

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \dots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \dots & a_{m,q}B \end{pmatrix} \in \mathbb{R}^{mp \times nq}.$$

2.3.3. Distanzmaße

Wie in Kapitel 2.3.1 angesprochen, muss ein geeignetes Distanzmaß angewandt werden, um die Differenz zwischen dem Referenzbild R und dem Templatebild T minimieren und damit die Ähnlichkeit zweier Bilder messen zu können. In dieser Arbeit wurde als Maß, das in [26] näher beschriebene Maß über die Summe der quadrierten Differenzen – auch *sum of squared differences (SSD)* genannt – verwendet. Dieses Maß misst die Energie, die im Differenzbild enthalten ist. Deshalb ist es wichtig, dass die Intensitätswerte der beiden Bilder vergleichbar sind. Das SSD-Maß lässt sich wie folgt definieren.

Definition 2.19 (Summe der quadrierten Differenzen (SSD), [26, Kap. 6.2, S. 71])

Seien das Templatebild T und das Referenzbild R gegeben. Das SSD-Maß wird dann definiert durch

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

Neben dem SSD-Maß existieren noch weitere Maße wie beispielsweise das Maß über die normalisierte Kreuzkorrelation oder über das normalisierte Gradientenfeld. Für weitere Information sei hier auf [26] verwiesen.

Für ein Distanzmaß \mathcal{D} kann die erste Variation angegeben werden. Dabei sei \mathcal{H} ein Hilbertraum [7]. Die erste Variation des Distanzmaßes \mathcal{D} sei nach [7] für alle Variationsrichtungen $\boldsymbol{\eta} \in \mathcal{H}$ gegeben über

$$\delta \mathcal{D}(\mathbf{u}, \boldsymbol{\eta}) = \langle \mathbf{f}(\mathbf{u}), \boldsymbol{\eta} \rangle_{\mathcal{H}},$$

wobei

$$\mathbf{f}(\mathbf{u}) = (f_1(\mathbf{u}), f_2(\mathbf{u}))^\top = (T(\mathbf{u}) - R) \nabla_{\mathbf{u}} T(\mathbf{u})$$

eine nichtlineare Funktion ist. Mit $\nabla_{\mathbf{u}} F = (\partial F / \partial u_1, \partial F / \partial u_2)^\top$ sei der Gradient eines Funktionals F bezüglich $\mathbf{u}(\mathbf{x})$ bezeichnet. Mit dieser Notation sollte die Formulierung der Euler-Lagrange-Gleichungen im nächsten Abschnitt nachvollziehbar sein.

2. Mathematische Grundlagen

Wie in Kapitel 2.3.1 angesprochen, reicht ein Distanzmaß zum Lösen des Bildregistrierungsproblems nicht aus, sondern es wird ein zusätzlicher Regularisierungsterm benötigt. Das folgende Kapitel beschäftigt sich mit den existierenden und üblicherweise verwendeten Regularisierern.

2.3.4. Regularisierer

Im Folgenden sollen die hauptsächlich verwendeten Registrierungsmodelle und die dazugehörigen Euler-Lagrange-Gleichungen kurz vorgestellt werden.

Elastischer Regularisierer

Die elastische Registrierung wurde zuerst von Broit im Jahre 1981 in [3] eingeführt. Diese Registrierung basiert auf einem quantitativen Maß für die Deformation, einem quantitativen Maß für die Ähnlichkeit zwischen deformierten Bildern und einer Methode, die die beiden Maße verwendet, um eine optimale Abbildung zu erhalten [27]. Diese Herangehensweise führt auf folgenden Regularisierungsterm

$$\mathcal{R}^{elas}(\mathbf{u}) = \int_{\Omega} \frac{\mu}{4} \sum_{l,m=1}^2 (\partial_{x_l} u_m + \partial_{x_m} u_l)^2 + \frac{\lambda}{2} (\nabla \cdot \mathbf{u})^2 d\Omega,$$

was nach [7] zu folgendem Euler-Lagrange System von nichtlinearen partiellen Differentialgleichungen zweiter Ordnung führt

$$\begin{cases} f_1(\mathbf{u}) - \alpha((\lambda + 2\mu)\partial_{x_1x_2}u_1 + \mu\partial_{x_2x_2}u_1 + (\lambda + \mu)\partial_{x_1x_2}u_2) = 0 \\ f_2(\mathbf{u}) - \alpha((\lambda + \mu)\partial_{x_1x_2}u_1 + \mu\partial_{x_1x_1}u_2 + (\lambda + 2\mu)\partial_{x_2x_2}u_2) = 0, \end{cases}$$

mit den Randbedingungen $\langle \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top) + \lambda \text{diag}(\nabla \cdot \mathbf{u}), \mathbf{n} \rangle_{\mathbb{R}^2} = 0$ auf $\partial\Omega$. Die Parameter $\mu > 0$ und $\lambda \leq 0$ sind die Lamé Konstanten und $\mathbf{n} = (n_{x_1}, n_{x_2})^\top$ ist der nach außen zeigende Einheitsnormalenvektor des Bildrandes $\partial\Omega$.

Diffusiver Regularisierer

Bei der diffusiven Registrierung basiert der Regularisierungsterm \mathcal{R} auf der L^2 Norm von ∇u_l . Der Regularisierungsterm ist nach [26] gegeben durch

$$\mathcal{R}^{diff}(\mathbf{u}) = \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} |\nabla u_l|^2 d\Omega,$$

was durch Einsetzen in die Definition zur Bestimmung von Euler-Lagrange-Gleichungen oder Nachlesen in [7] zu folgendem Euler-Lagrange System von nichtlinearen partiellen Differentialgleichungen zweiter Ordnung führt

$$\begin{cases} f_1(\mathbf{u}) - \alpha \Delta u_1 = 0 \\ f_2(\mathbf{u}) - \alpha \Delta u_2 = 0, \end{cases}$$

mit den Randbedingungen $\langle \nabla u_l, \mathbf{n} \rangle_{\mathbb{R}^2} = 0$ auf $\partial\Omega$. Dieses Modell kann also als ein Spezialfall des elastischen Modells mit $\mu = 1$ und $\lambda = -1$ angesehen werden.

Regularisierer basierend auf der Totalen Variation (TV)

Der TV-Regularisierer wurde erstmals in [20] eingeführt und dort basierend auf der Semi-Norm der beschränkten Variation formuliert. Dadurch wird folgender Regularisierungsterm erhalten

$$\mathcal{R}^{\beta TV}(\mathbf{u}) = \sum_{l=1}^2 \int_{\Omega} |\nabla u_l|_{\beta} d\mathbf{x} = \sum_{l=1}^2 \int_{\Omega} \sqrt{u_{l_{x_1}}^2 + u_{l_{x_2}}^2 + \beta} d\Omega,$$

wobei der Parameter β , der klein aber größer als Null gewählt wird, hinzugefügt wird, um Singularitäten von $|\nabla u_l| = 0$ zu vermeiden. Durch Nachrechnen oder Nachschlagen in [7] führt dies zu folgendem Euler-Lagrange System von nichtlinearen partiellen Differentialgleichungen zweiter Ordnung

$$\begin{cases} f_1(\mathbf{u}) - \alpha \nabla \cdot \left(\frac{\nabla u_1}{|\nabla u_1|_{\beta}} \right) = 0 \\ f_2(\mathbf{u}) - \alpha \nabla \cdot \left(\frac{\nabla u_2}{|\nabla u_2|_{\beta}} \right) = 0 \end{cases}$$

mit den Randbedingungen $\langle \nabla u_l, \mathbf{n} \rangle_{\mathbf{R}^2} = 0$ auf $\partial\Omega$. Anhand der Formel sieht man, dass jede Deformationsvariable u_1 und u_2 separat regularisiert wird, damit ist die Entkopplung des nichtlinearen Diffusionsprozesses ersichtlich. Des Weiteren fehlt diesem Regularisierer die Rotationsinvarianz, da hier eine nicht-vektorielle Regularisierung vorliegt. Aus diesem Grund sind für diesen Regularisierer Registrierungsprobleme mit nicht-achsen ausgerichteten Unstetigkeiten schwierig zu lösen [7].

Fischer-Modersitzki Krümmungsregularisierer

Der Fischer-Modersitzki-Regularisierer kann als eine Approximation der mittleren Krümmung der Fläche u_l betrachtet werden. Er ist gegeben durch

$$\mathcal{R}^{FMcurv}(\mathbf{u}) = \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} (\hat{\kappa}(u_l))^2 d\Omega = \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} (\Delta u_l)^2 d\Omega.$$

Durch Einsetzen in die Definition zur Bestimmung von Euler-Lagrange-Gleichungen oder Nachlesen in [7] führt dies zu folgendem Euler-Lagrange System von nichtlinearen partiellen Differentialgleichungen vierter Ordnung

$$\begin{cases} f_1(\mathbf{u}) + \alpha \Delta^2 u_1 = 0 \\ f_2(\mathbf{u}) + \alpha \Delta^2 u_2 = 0, \end{cases}$$

mit den Randbedingungen $\nabla u_l = 0, \nabla \Delta u_l = 0$ auf $\partial\Omega$ für $l = 1, 2$.

Blickt man mit den Methoden der Differentialgeometrie auf diesen Regularisierer, kann u_l hier als Fläche im \mathbb{R}^3 verstanden werden, die durch $(x_1, x_2, u_l(x_1, x_2))$ mit $u_l(x_1, x_2) = 0$ gegeben ist. An dieser Stelle soll noch einmal die Definition der mittleren Krümmung (siehe Formel (2.14)) aus dem Differentialgeometriekapitel 2.1 be-

2. Mathematische Grundlagen

trachtet werden. Wendet man diese Definition der mittleren Krümmung auf die Fläche u_l an, so erhält man

$$\kappa(u_l) = \frac{(1 + u_{l_{x_1}}^2)u_{l_{x_2x_2}} - 2u_{l_{x_1}}u_{l_{x_2}}u_{l_{x_1x_2}} + (1 + u_{l_{x_2}}^2)u_{l_{x_1x_1}}}{(1 + u_{l_{x_1}}^2 + u_{l_{x_2}}^2)^{3/2}}.$$

Unter der Annahme, dass $|\nabla u_l| = \sqrt{u_{l_{x_1}}^2 + u_{l_{x_2}}^2} \approx 0$ ist, und es somit gilt, dass $|\nabla u_l|^2 = u_{l_{x_1}}^2 + u_{l_{x_2}}^2 \approx 0$ ist, ergibt sich

$$\kappa(u_l) \approx u_{l_{x_2x_2}} + u_{l_{x_1x_1}} =: \Delta u_l,$$

was der Laplace-Operator ist, der im Fischer-Modersitzki-Krümmungsregularisierer verwendet wird und dort wegen der Verwandtschaft mit der mittleren Krümmung als $\hat{\kappa}(u_l)$ bezeichnet wird. Aufgrund dieser Tatsache lässt sich der Fischer-Modersitzki-Regularisierer als eine Approximation der mittleren Krümmung aus der Differentialgeometrie auffassen.

2.3.5. Lösen des Registrierungsproblems

Das Bildregistrierungsproblem (2.19) kann mit gewöhnlichen Verfahren der Optimierung, die zum Beispiel in [29] aufgeführt werden, gelöst werden.

In dieser Arbeit wird das Gauß-Newton Verfahren angewandt. Dieses kann als eine modifizierte Form des Newton Verfahrens angesehen werden. Das Newton Verfahren ist, wie in [29] nachzulesen ist, ein iteratives Verfahren, das eine Approximation der Taylorentwicklung der zu minimierenden Funktion nutzt. Die Suchrichtung p_k in der k -ten Iteration wird über die Newton-Gleichungen $\nabla^2 f(x_k)p_k = -\nabla f(x_k)$ erhalten, wobei $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ eine differenzierbare Funktion ist. Mit ∇f sei der Gradient und mit $\nabla^2 f$ die Hessematrix von f bezeichnet.

Beim Gauß-Newton Verfahren wird nach [29] das Residuum r_k der zu minimierenden Funktion mit der Methode der kleinsten Quadrate minimiert. Dabei wird ausgenutzt, dass die Hesse-Matrix über die Jakobi-Matrix durch $\nabla^2 f_k \approx J_k^\top J_k$ approximiert werden kann, wobei J_k die Jakobi-Matrix bezeichnet. Die Suchrichtung wird bei diesem Verfahren durch Lösen von $J_k^\top J_k p_k = -J_k^\top r_k$ erhalten. Bei den Liniensuchalgorithmen ist gemein, dass an diese Algorithmen Abbruchbedingungen gestellt werden, um einen hinreichenden Abstieg der Zielfunktion zu gewährleisten. Hier wird die Armijo-Bedingung verwendet, die folgendermaßen lautet: $f(x_k + \alpha_k p_k) \leq f(x_k) + c \alpha_k \nabla f_k(x_k)^\top p_k$ mit $c \in (0, 1)$. Mit dem Gauß-Newton Verfahren erhält man ein ähnliches Ergebnis wie über die Newton Methode, doch durch die im Gauß-Newton Verfahren verwendete Approximation muss im Gegensatz zum Newton Verfahren nicht die Hesse-Matrix berechnet werden. Dies verringert den Rechenaufwand deutlich und ist damit für die Implementierung eines Optimierungsproblems geeignet. Aus diesem Grund wird das Gauß-Newton Verfahren zum

Lösen des in dieser Arbeit gestellten Bildregistrierungsproblems verwendet. In diesem Kapitel wurde zunächst eine anschauliche Einführung in das Themengebiet der Differentialgeometrie gegeben, wobei diese mit der Herleitung der Formel für die mittlere Krümmung geschlossen hat. Diese Formel wurde bei der Beschreibung des Fischer-Modersitzki-Regularisierers wieder aufgegriffen, wobei gezeigt werden konnte, dass dieser als eine Approximation der mittleren Krümmung aus der Differentialgeometrie aufgefasst werden kann. Im nächsten Kapitel wird ein Regularisierungsmodell vorgestellt, welches diese Approximationen nicht nutzt, sondern die vollständige Form der mittleren Krümmung verwendet.

3. Modellbeschreibung in 2D

In diesem Kapitel soll der Regularisierer eingeführt werden, der in dieser Arbeit genauer untersucht wird. Dieser Regularisierer beruht auf der Differentialgeometrie. Die daraus benötigten Konzepte, die zum Verständnis dieses Regularisierers wichtig sind, wurden in Kapitel 2.1 dargestellt. Der Regularisierer, der hier im Zentrum steht, ist der Krümmungsregularisierer, der in [32] vorgestellt wird. Dieser Regularisierer setzt die Idee fort, die im Fischer-Modersitzki-Regularisierungsmodell eingeführt [10, 11, 12] und in [6, 7] als neues Krümmungsregularisierungsmodell weiterentwickelt wurde.

Das krümmungsbasierte Regularisierungsmodell, das in [7] eingeführt wird, wird dort über das Fischer-Modersitzki-Modell motiviert und als eine Erweiterung dessen angesehen. Ziel von Chumchob et al. in [7] war es, ein Registrierungsmodell zu entwerfen, das die gleichen positiven Eigenschaften wie das Fischer-Modersitzki-Modell hat und dieses noch übertrifft, indem es auf glatte und nichtglatte Registrierungsprobleme angewandt werden kann. Um dies zu erreichen, wurde das vollständige Krümmungsmodell der mittleren Krümmung verwendet, das heißt die Approximationen, die im Fischer-Modersitzki-Modell verwendet werden, werden hier nicht vorgenommen. Nach [7] lautet dann das Modell folgendermaßen:

$$\kappa(u_l) = \frac{(\beta + u_{l_{x_1}}^2)u_{l_{x_1x_1}} - 2u_{l_{x_1}}u_{l_{x_2}}u_{l_{x_1x_2}} + (\beta + u_{l_{x_2}}^2)u_{l_{x_2x_2}}}{(\beta + u_{l_{x_1}}^2 + u_{l_{x_2}}^2)^{3/2}} \quad (3.1)$$

mit $|\nabla u_l|_\beta = \sqrt{u_{l_{x_1}}^2 + u_{l_{x_2}}^2 + \beta}$ und $\beta > 0$. Außerdem wird behauptet, dass das Modell (3.1) über die Divergenz auf folgende Weise ausgedrückt werden kann:

$$\kappa(u_l) = \nabla \cdot \frac{\nabla u_l}{|\nabla u_l|_\beta}. \quad (3.2)$$

An dieser Stelle sind einige Anmerkungen nötig. Das in [7] formulierte Modell (3.1) ist nicht äquivalent zu der Formel (2.14) für die mittlere Krümmung, die man aus der Differentialgeometrie erhält. Der Unterschied besteht darin, dass im Zähler der Formel (3.1) im ersten und im letzten Summanden jeweils die gleichen partiellen Ableitungen enthalten sind, wohingegen in Formel (2.14) in diesen Summanden jeweils verschiedene partielle Ableitungen stehen. Da jedoch nach [7] gleichzeitig die Divergenzformel (3.2) gelten soll, wurde in dieser Arbeit diese Formel auf die Form der mittleren Krümmung gebracht. Die Berechnungen dazu können im Anhang in Abschnitt A.1 nachgelesen werden. Dabei wurde als Ergebnis eine Formel der Form (2.14) erhalten. Das lässt darauf schließen, dass sich in [6] und in [7], in denen dieses Modell aufgestellt wurde, ein Schreibfehler eingeschlichen hat und das Modell richtigerweise folgendermaßen heißen müsste:

$$\kappa(u_l) = \frac{(\beta + u_{l_{x_1}}^2)u_{l_{x_2x_2}} - 2u_{l_{x_1}}u_{l_{x_2}}u_{l_{x_1x_2}} + (\beta + u_{l_{x_2}}^2)u_{l_{x_1x_1}}}{(\beta + u_{l_{x_1}}^2 + u_{l_{x_2}}^2)^{3/2}} = \nabla \cdot \frac{\nabla u_l}{|\nabla u_l|_\beta}.$$

3. Modellbeschreibung in 2D

Da die Implementierung über die Divergenzformel erfolgt ist, hat dies keine Auswirkungen auf die praktischen Ergebnisse.

Der vollständige Regularisierungsterm hat damit die Form

$$\mathcal{R}^{CCB}(\mathbf{u}) = \sum_{l=1}^2 \int_{\Omega} \Phi(\kappa(u_l)) \, d\Omega \quad (3.3)$$

mit $\Phi(s) = \frac{1}{2}s^2$. Das Registrierungsmodell mit Regularisierungsterm (3.3) führt nach [7] zu folgendem System von Euler-Lagrange-Gleichungen

$$\begin{cases} f_1(\mathbf{u}) + \alpha \nabla \cdot \left(\frac{1}{|\nabla u_1|^\beta} \nabla \kappa(u_1) - \frac{\nabla u_1 \nabla \kappa(u_1)}{(|\nabla u_1|^\beta)^3} \nabla u_1 \right) = 0 \\ f_2(\mathbf{u}) + \alpha \nabla \cdot \left(\frac{1}{|\nabla u_2|^\beta} \nabla \kappa(u_2) - \frac{\nabla u_2 \nabla \kappa(u_2)}{(|\nabla u_2|^\beta)^3} \nabla u_2 \right) = 0 \end{cases}$$

mit den Randbedingungen $\langle \nabla u_l, \mathbf{n} \rangle_{\mathbb{R}^2} = \langle \nabla \kappa(u_l), \mathbf{n} \rangle_{\mathbb{R}^2} = 0$ auf $\partial\Omega$. Zhang und Chen haben in [32] festgestellt, dass das Modell (3.3) sehr gute Ergebnisse liefert, falls u_1 und u_2 nicht stark gekoppelt sind. Der Grund dafür ist der nichtlineare Diffusionsprozess, der sich aus der ersten Variation von \mathcal{R}^{CCB} ergibt und der die Kopplung der Komponenten des Deformationsfeldes u_1 und u_2 nicht erzwingt. Das bedeutet, es gibt Fälle, in denen dieses Modell kein gutes Registrierungsergebnis liefert, zum Beispiel bei nichtglatten Registrierungsproblemen mit nicht-achsen-ausgerichteter Unstetigkeiten. Aus diesem Grund wurde in [32] ein neuer krümmungsbasierter Regularisierer eingeführt, der sehr ähnlich zu Modell (3.3) ist. Der Regularisierer basiert, wie der in [6, 7] beschriebene Regularisierer, auf der mittleren Krümmung der Differentialgeometrie. Zusätzlich erlaubt dieser eine Koppelung der beiden Komponenten u_1 und u_2 des Deformationsfeldes. Dies soll die Qualität der Registrierung, für glatte und nichtglatte Registrierungsprobleme steigern. Die erste Idee, die in [32] vorgeschlagen wird, ist das Modell

$$\min_{\mathbf{u}} \mathcal{J}_\alpha(\mathbf{u}) = \mathcal{D}(\mathbf{u}) + \alpha \mathcal{R}(\mathbf{u}), \quad \mathcal{R}(\mathbf{u}) = \int_{\Omega} \sqrt{\kappa(u_1)^2 + \kappa(u_2)^2} \, d\Omega$$

zu verwenden, wobei \mathbf{u} aus einem Funktionenraum stammt. Allerdings erzwingt dieses Modell extrem stark die Kopplung, weshalb es nicht das bevorzugte Modell ist. Aus diesem Grund wird in [32] eine modifizierte Form der mittleren Krümmung $\kappa_1(\mathbf{u}), \kappa_2(\mathbf{u})$ verwendet. Bevor der eigentliche Regularisierungsterm eingeführt wird, wird die TV-Halbnorm betrachtet. Auf diese soll hier nicht weiter eingegangen werden, für eine genauere Ausführung sei auf [20] verwiesen. Die Minimierung des Terms

$$\int_{\Omega} |\nabla u| \, d\Omega$$

für eine skalare Funktion u , führt zur mittleren Krümmung

$$\kappa(u) = \kappa_{CCB}(u) = \nabla \cdot \frac{\nabla u}{|\nabla u|}.$$

Wird die vektorisierte TV-Halbnorm zu

$$\int_{\Omega} |\nabla \mathbf{u}| d\Omega \equiv \int_{\Omega} \sqrt{|u_1|^2 + |u_2|^2} d\Omega$$

minimiert, so führt dies zur mittleren Krümmung in der folgenden Form

$$\kappa_1(\mathbf{u}) = \nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|}, \quad \kappa_2(\mathbf{u}) = \nabla \cdot \frac{\nabla u_2}{|\nabla \mathbf{u}|}, \quad (3.4)$$

wobei durch den Term $|\nabla \mathbf{u}|$ die Kopplung miteinbezogen ist. Um nun die Kopplungsinformation auszunutzen, wird in [32] folgendes Variationsmodell

$$\min_{\mathbf{u}} \mathcal{J}_{\alpha}(\mathbf{u}) = \mathcal{D}(\mathbf{u}) + \alpha \mathcal{R}^{NewC}(\mathbf{u}), \quad \text{mit} \quad \mathcal{R}^{NewC}(\mathbf{u}) = \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_l}{|\nabla \mathbf{u}|_{\beta}} \right)^2 d\Omega,$$

das auf Gleichung (3.4) basiert, eingeführt, wobei $|\nabla \mathbf{u}|_{\beta} = \sqrt{|\nabla u_1|^2 + |\nabla u_2|^2 + \beta}$ ist und β klein und positiv gewählt wird, um Singularitäten zu vermeiden. Das vektorielle Krümmungsmodell hat somit die Form

$$\min_{\mathbf{u}} \left\{ \mathcal{J}_{\alpha}(\mathbf{u}) = \frac{1}{2} \int_{\Omega} (T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - R(\mathbf{x}))^2 d\Omega + \frac{\alpha}{2} \sum_{l=1}^2 \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_l}{|\nabla \mathbf{u}|_{\beta}} \right)^2 d\Omega \right\}. \quad (3.5)$$

Diese Wahl des Regularisierers hat nach [32] einige Vorteile. Es basiert auf dem krümmungsbasierten Fischer-Modersitzki-Regularisierer und dem Regularisierer \mathcal{R}^{CCB} aus [7] und teilt mit diesen die positiven Eigenschaften, die in dieser Arbeit für den neuen Regularisierungsterm in Lemma 3.1 und der anschließenden Folgerung 3.1 bewiesen wurden.

Lemma 3.1

Sei $\mathbf{u}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ mit $n \in \mathbb{N}$ das Verschiebungsfeld. Es sei \mathbf{u} eine affin-lineare Transformation, das heißt $\mathbf{u}(\mathbf{x}) = \mathbf{C}\mathbf{x} + \mathbf{b}$ mit $\mathbf{C} \in \mathbb{R}^{n \times n}$ und $\mathbf{b} \in \mathbb{R}^{n \times 1}$, dann gilt $\mathcal{R}^{NewC}(\mathbf{u}) = 0$.

Beweis. Da \mathbf{u} darstellbar ist als $\mathbf{u} = \mathbf{C}\mathbf{x} + \mathbf{b}$ folgt für den Regularisierer

$$\mathcal{R}^{NewC}(\mathbf{u}) = \sum_{l=1}^n \frac{1}{2} \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_l}{|\nabla \mathbf{u}|_{\beta}} \right)^2 d\Omega = \mathcal{R}^{NewC}(\mathbf{C}\mathbf{x} + \mathbf{b}).$$

u_l kann geschrieben werden als $u_l = [\mathbf{C}\mathbf{x} + \mathbf{b}]_l = \sum_{k=1}^n c_{lk} \cdot x_k + b_l$. Betrachtet man nun den Integranden ohne die Potenz, erhält man

$$\nabla \cdot \frac{\nabla u_l}{|\nabla \mathbf{u}|_{\beta}} = \nabla \cdot \left(\frac{1}{|\nabla \mathbf{u}|_{\beta}} \nabla \left[\sum_{k=1}^n c_{lk} x_k + b_l \right] \right) = \nabla \cdot \begin{pmatrix} c_{l1} \\ \vdots \\ c_{ln} \end{pmatrix} \frac{1}{|\nabla \mathbf{u}|_{\beta}}$$

3. Modellbeschreibung in 2D

$$\begin{aligned}
&= \nabla \cdot \frac{\begin{pmatrix} c_{l1} \\ \vdots \\ c_{ln} \end{pmatrix}}{\sqrt{\sum_{k=1}^n |\nabla u_k|^2 + \beta}} = \nabla \cdot \frac{\begin{pmatrix} c_{l1} \\ \vdots \\ c_{ln} \end{pmatrix}}{\sqrt{\sum_{k=1}^n \left| \nabla \left(\sum_{j=1}^n c_{lj} x_j + b_l \right) \right|^2 + \beta}} \\
&= \nabla \cdot \frac{\begin{pmatrix} c_{l1} \\ \vdots \\ c_{ln} \end{pmatrix}}{\sqrt{\sum_{k=1}^n \left| \begin{pmatrix} c_{l1} \\ \vdots \\ c_{ln} \end{pmatrix} \right|^2 + \beta}} = \mathbf{0}.
\end{aligned}$$

Das letzte Gleichheitszeichen gilt, da der gesamte Term nun unabhängig von x und u ist und somit eine Konstante darstellt, die beim Ableiten verschwindet. Damit erhält man $\mathcal{R}^{NewC}(\mathbf{u}) = \mathcal{R}^{NewC}(\mathbf{C}\mathbf{x} + \mathbf{b}) = 0$. \square

Folgerung 3.1

Wählt man in Lemma 3.1 eine Rotationsmatrix $Q \in \mathbb{R}^{n \times n}$ mit $Q^\top Q = I_{n \times n}$ die Einheitsmatrix und $n \in \mathbb{N}$, so lässt sich analog zu Lemma 3.1 die Rotations- und Translationsinvarianz des Regularisierers beweisen.

Mit Lemma 3.1 und Folgerung 3.1 sind nun die in [32] behaupteten Aussagen, dass das Modell affin-lineare Transformationen nicht bestraft und invariant unter Rotation und Translation ist, bewiesen.

Zur Untersuchung des Regularisierers gehört es, dessen Euler-Lagrange-Gleichungen aufzustellen. Aus diesem Grund wurden die zum Minimierungsproblem (3.5) gehörigen Euler-Lagrange-Gleichungen berechnet. Das Aufstellen dieser Euler-Lagrange-Gleichungen lässt sich im Anhang in Abschnitt A.2 nachlesen. Als Ergebnis erhält man folgende Terme, die gleich Null zu setzen sind:

$$\begin{cases} f_1(\mathbf{u}) + \alpha \varphi_1 \nabla \cdot \frac{\nabla \kappa(u_1)}{|\nabla \mathbf{u}|_\beta} - \varphi_1 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_\beta^3} \nabla u_1 \right) + \varphi_2 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_\beta^3} \nabla u_2 \right) \\ f_2(\mathbf{u}) + \alpha \varphi_2 \nabla \cdot \frac{\nabla \kappa(u_2)}{|\nabla \mathbf{u}|_\beta} - \varphi_2 \nabla \cdot \left(\nabla \kappa(u_2) \frac{\nabla u_2}{|\nabla \mathbf{u}|_\beta^3} \nabla u_2 \right) + \varphi_1 \nabla \cdot \left(\nabla \kappa(u_2) \frac{\nabla u_2}{|\nabla \mathbf{u}|_\beta^3} \nabla u_1 \right) \end{cases}$$

mit den Randbedingungen $\langle \nabla u_l, \vec{n} \rangle_{\mathbb{R}^2} = \langle \nabla \kappa(u_l), \vec{n} \rangle_{\mathbb{R}^2} = \langle \varphi_l, \vec{n} \rangle_{\mathbb{R}^2} = 0$ für $l = 1, 2$.

Um nun das hier vorliegende Optimierungsproblem (3.5) zu lösen, gibt es mehrere Möglichkeiten. In dieser Arbeit wurde der Ansatz verwendet, bei dem zuerst das Problem diskretisiert wird und im Anschluss die Optimierung erfolgt. Dieser Ansatz

wurde auch in [32] genutzt. Dabei muss zuerst eine geeignete Diskretisierung für jeden Term des Optimierungsproblems gefunden werden. Anschließend wird das komplett diskretisierte Optimierungsproblem mit numerischen Algorithmen gelöst. Mithilfe der hier aufgestellten Euler-Lagrange-Gleichungen könnte auch der umgekehrte Ansatz, bei dem zuerst optimiert und dann diskretisiert wird, verwendet werden. Dieser Ansatz wurde in dieser Arbeit nicht mehr weiter verfolgt, diesem könnte aber in weiterführenden Arbeiten nachgegangen werden.

3.1. Numerische Lösung des Registrierungsproblems

In diesem Abschnitt wird zuerst die in [32] verwendete Diskretisierung beschrieben und im Anschluss daran die numerischen Algorithmen erläutert, die in [32] zur Lösung des Optimierungsproblems verwendet wurden.

3.1.1. Diskretisierung

Finite Differenzen Diskretisierung

Es wird angenommen, dass die diskreten Bilder $m_1 \times m_2$ Pixel haben. Des Weiteren sollen hier nur 2D Bilder betrachtet werden, damit gilt für den Bildbereich $\Omega = [0, \omega_1] \times [0, \omega_2] \subset \mathbb{R}^2$, wobei die Gitterweite über $h_i = \omega_i/m_i$, $i = 1, 2$ bestimmt ist. Es wird ein zell-zentriertes Gitter angenommen, deswegen wird das diskrete Gebiet in Ω durch die Menge

$$\Omega_h = \left\{ \mathbf{x} \mid \mathbf{x} = \begin{pmatrix} x_{1j_1} \\ x_{2j_2} \end{pmatrix} = \begin{pmatrix} (j_1 - 0.5)h_1 \\ (j_2 - 0.5)h_2 \end{pmatrix}, j_1 = 1, 2, \dots, m_1; j_2 = 1, 2, \dots, m_2 \right\}.$$

beschrieben. Punkte in dieser Menge können in Vektorform gebracht werden

$$\begin{aligned} \mathbf{x}_{c,1} &= [x_{1,1,1}, x_{1,2,1}, \dots, x_{1,m_1,1}, x_{1,1,2}, x_{1,2,2}, \dots, x_{1,m_1,2}, \dots, x_{1,1,m_2}, x_{1,2,m_2}, \dots, x_{1,m_1,m_2}]^\top, \\ \mathbf{x}_{c,2} &= [x_{2,1,1}, x_{2,2,1}, \dots, x_{2,m_1,1}, x_{2,1,2}, x_{2,2,2}, \dots, x_{2,m_1,2}, \dots, x_{2,1,m_2}, x_{2,2,m_2}, \dots, x_{2,m_1,m_2}]^\top \end{aligned}$$

und in einem Vektor $\mathbf{X}_c^h = [\mathbf{x}_{c,1}; \mathbf{x}_{c,2}]$ zusammengefasst werden.

Diskretisierung des neuen Regularisierers $\mathcal{R}^{NewC}(\mathbf{u})$

Das kontinuierliche Verschiebungsfeld wird mit $\mathbf{u} = (u_1, u_2)^\top$ bezeichnet. Das diskrete Analogon dazu wird im diskreten Gebiet Ω_h mit $\mathbf{u}^h = (u_1^h, u_2^h)^\top$ bezeichnet, wobei u_1^h und u_2^h Gitterfunktionen sind. Für diese gilt $(u_l^h)_{j_1, j_2} = u_l^h(x_{1j_1}, x_{2j_2})$, $j_1 = 1, 2, \dots, m_1$; $j_2 = 1, 2, \dots, m_2$ und $l = 1, 2$. Die diskreten Gradienten- und Divergenzoperatoren sollen im Folgenden als ∇^h und $\nabla^h \cdot$ bezeichnet werden. Der diskrete Gradientenoperator ∇^h an jedem Pixel (j_1, j_2) ist definiert über

$$(\nabla^h \mathbf{u}^h)_{j_1, j_2} = ((\nabla^{h,1} u_1^h)_{j_1, j_2}, (\nabla^{h,2} u_2^h)_{j_1, j_2})^\top,$$

3. Modellbeschreibung in 2D

wobei

$$\begin{aligned} (\nabla^{h,l} u_l^h)_{j_1, j_2} &= ((\partial_1^{h,l} u_l^h)_{j_1, j_2}, (\partial_2^{h,l} u_l^h)_{j_1, j_1})^\top, \\ (\partial_1^{h,l} u_l^h)_{j_1, j_2} &= \begin{cases} (u_l^h)_{j_1+1, j_2} - (u_l^h)_{j_1, j_2}, & \text{falls } j_1 < m_1 \\ 0, & \text{falls } j_1 = m_1, \end{cases} \\ (\partial_2^{h,l} u_l^h)_{j_1, j_2} &= \begin{cases} (u_l^h)_{j_1, j_2+1} - (u_l^h)_{j_1, j_2}, & \text{falls } j_2 < m_2 \\ 0, & \text{falls } j_2 = m_2 \end{cases} \end{aligned}$$

gilt und Neumann-Randbedingungen $\frac{\partial u_l}{\partial \nu} = 0$, $l = 1, 2$ auf $\partial\Omega$ angenommen werden. Der Divergenzoperator wird berechnet, indem ausgenutzt wird, dass dieser über den negativen adjungierten Gradienten ausgedrückt werden kann. Darauf soll im Folgenden näher eingegangen werden. Es gilt

$$\langle \nabla \cdot u, v \rangle_{\mathcal{L}^2(\Omega)} = \int_{\Omega} (\nabla \cdot u) v \, dx.$$

Durch die Anwendung der mehrdimensionalen partiellen Integration erhält man

$$\begin{aligned} \int_{\Omega} (\nabla \cdot u) v \, dx &= - \int_{\Omega} u \nabla v \, dx + \int_{\partial\Omega} v u \, dS \\ &= - \int_{\Omega} u \nabla v \, dx + 0 \\ &= - \langle u, \nabla v \rangle_{\mathcal{L}^2(\Omega)} = \langle u, -\nabla v \rangle_{\mathcal{L}^2(\Omega)} = \langle -\nabla v, u \rangle_{\mathcal{L}^2(\Omega)}. \end{aligned}$$

Nach der Definition eines adjungierten Operators (siehe [31, Def. V.5.1]) gilt

$$\langle -\nabla v, u \rangle_{\mathcal{L}^2(\Omega)} = \langle v, -\nabla^* u \rangle_{\mathcal{L}^2(\Omega)}.$$

Folglich kann der Divergenzoperator $\nabla \cdot$ über den negativen adjungierten Gradienten $-\nabla^*$ formuliert werden. Deshalb kann er durch

$$\begin{aligned} (\nabla \cdot \omega_l)_{j_1, j_2} &= \begin{cases} (\omega_l^1)_{j_1, j_2} - (\omega_l^1)_{j_1-1, j_2} \\ (\omega_l^1)_{j_1, j_2} \\ -(\omega_l^1)_{j_1-1, j_2} \end{cases} \\ &+ \begin{cases} (\omega_l^2)_{j_1, j_2} - (\omega_l^2)_{j_1, j_2-1}, & \text{falls } 1 < j_i < m_i, \, i = 1, 2 \\ (\omega_l^2)_{j_1, j_2} & \text{falls } j_1 = j_2 = 1, \\ -(\omega_l^2)_{j_1, j_2-1}, & \text{falls } j_1 = m_1, j_2 = m_2 \end{cases} \end{aligned}$$

ausgedrückt werden. Die Gitterfunktionen u_1^h und u_2^h werden als Spaltenvektoren \mathbf{u}_1^h und \mathbf{u}_2^h in lexikographischer Reihenfolge angeordnet, und zwar

$$\begin{aligned} \mathbf{u}_1^h &= (u_{1,1,1}^h, u_{1,2,1}^h, \dots, u_{1,m_1,1}^h, u_{1,1,2}^h, u_{1,2,2}^h, \dots, u_{1,m_1,2}^h, \dots, u_{1,1,m_2}^h, u_{1,2,m_2}^h, \dots, u_{1,m_1,m_2}^h)^\top, \\ \mathbf{u}_2^h &= (u_{2,1,1}^h, u_{2,2,1}^h, \dots, u_{2,m_1,1}^h, u_{2,1,2}^h, u_{2,2,2}^h, \dots, u_{2,m_1,2}^h, \dots, u_{2,1,m_2}^h, u_{2,2,m_2}^h, \dots, u_{2,m_1,m_2}^h)^\top, \end{aligned}$$

3.1. Numerische Lösung des Registrierungsproblems

mit $\mathbf{u}_1^h \in \mathbb{R}^N$, $\mathbf{u}_2^h \in \mathbb{R}^N$. Diese werden in einem Vektor $\mathbf{U}^h = (\mathbf{u}_1^h; \mathbf{u}_2^h) \in \mathbb{R}^{2N}$ zusammengefasst, wobei $N = m_1 m_2$ gilt. Der diskrete Gradient $(\nabla^{h,l} \mathbf{u}_l^h)_{j_1, j_2}$ kann über eine Multiplikation einer Matrix $A_k^\top \in \mathbb{R}^{2 \times N}$ ($k = 1, 2, \dots, N$) mit dem Vektor \mathbf{u}_l^h ($l = 1, 2$) ausgedrückt werden:

$$A_k^\top \mathbf{u}_l^h = \begin{cases} ((\mathbf{u}_l^h)_{k+1} - (\mathbf{u}_l^h)_k; (\mathbf{u}_l^h)_{k+m_2} - (\mathbf{u}_l^h)_k), & \text{falls } k \bmod m_1 \neq 0 \text{ und } k + m_2 \leq N \\ (0; (\mathbf{u}_l^h)_{k+m_2} - (\mathbf{u}_l^h)_k), & \text{falls } k \bmod m_1 = 0 \text{ und } k + m_2 \leq N \\ ((\mathbf{u}_l^h)_{k+1} - (\mathbf{u}_l^h)_k; 0), & \text{falls } k \bmod m_1 \neq 0 \text{ und } k + m_2 > N \\ (0; 0), & \text{falls } k \bmod m_1 = 0 \text{ und } k + m_2 > N. \end{cases} \quad (3.6)$$

Die Matrizen A_k , $k = 1, 2, \dots, N$ werden verknüpft und man erhält

$$A = (A_1, A_2, \dots, A_N) = (A_{1,1}, A_{1,2}, \dots, A_{N,1}, A_{N,2}) \in \mathbb{R}^{N \times 2N}.$$

Daraus kann man die Matrizen A_{x_1} und A_{x_2} wählen

$$A_{x_1} = (A_{1,1}, A_{2,1}, \dots, A_{N,1}) \in \mathbb{R}^{N \times N},$$

$$A_{x_2} = (A_{1,2}, A_{2,2}, \dots, A_{N,2}) \in \mathbb{R}^{N \times N}.$$

Über diese Matrizen kann der Gradient beschrieben werden, der im Folgenden als Matrix B bezeichnet wird:

$$\nabla^{h,1} \mathbf{u}_1^h = \begin{bmatrix} A_{x_1}^\top \\ A_{x_2}^\top \end{bmatrix} \mathbf{u}_1^h =: B \mathbf{u}_1^h,$$

$$\nabla^{h,2} \mathbf{u}_2^h = \begin{bmatrix} A_{x_1}^\top \\ A_{x_2}^\top \end{bmatrix} \mathbf{u}_2^h =: B \mathbf{u}_2^h.$$

Eine Veranschaulichung der Gradienten, die im Zweidimensionalen berechnet werden, ist in Abbildung 3.1 zu sehen. Dabei zeigen die blauen Pfeile die Gradienten in x_1 -Richtung und die roten Pfeile die Gradienten in x_2 -Richtung an.

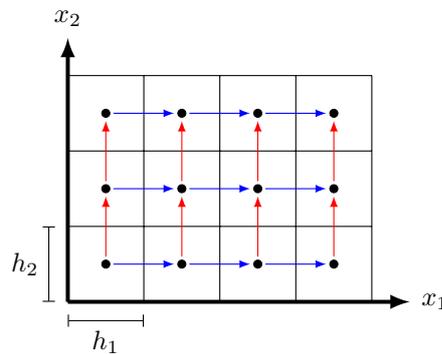


Abbildung 3.1: Darstellung der in 2D zu berechnenden Gradienten am Beispiel des Bildgebietes $m_1 = 4$ und $m_2 = 3$. Blaue Pfeile: Gradienten in x_1 -Richtung, rote Pfeile: Gradienten in x_2 -Richtung.

3. Modellbeschreibung in 2D

Für den gesamten diskreten Gradientenoperator ∇^h gilt

$$\nabla^h \mathbf{U}^h = \begin{bmatrix} \nabla^{h,1} & 0 \\ 0 & \nabla^{h,2} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^h \\ \mathbf{u}_2^h \end{bmatrix} = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^h \\ \mathbf{u}_2^h \end{bmatrix} =: \mathbf{A} \mathbf{U}^h.$$

Eine Veranschaulichung der Matrix \mathbf{A} am Beispiel $m_1 = 4$ und $m_2 = 3$ ist in Abbildung 3.2 zu sehen. Unter Verwendung des MATLAB-Befehls `spy` sind die von Null verschiedenen Einträge dargestellt.

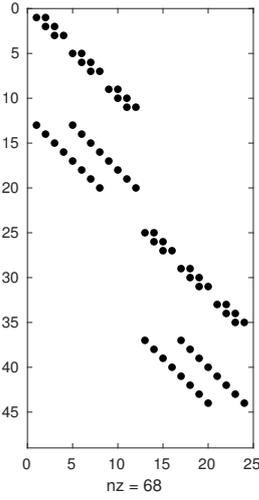


Abbildung 3.2: Darstellung der von Null verschiedenen Einträge der Gradientenmatrix \mathbf{A} unter Verwendung des MATLAB-Befehls `spy` am Beispiel $m_1 = 4, m_2 = 3$.

Der zu diskretisierende Regularisierer lautet

$$\mathcal{R}^{NewC}(\mathbf{u}) = \frac{1}{2} \int_{\Omega} \mathcal{B}[\mathbf{u}] \, d\Omega \quad \text{mit} \quad \mathcal{B}[\mathbf{u}] = \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right)^2 + \left(\nabla \cdot \frac{\nabla u_2}{|\nabla \mathbf{u}|_{\beta}} \right)^2.$$

Die diskrete Form des Regularisierer sieht demnach folgendermaßen aus

$$\mathcal{R}^{hNewC}(\mathbf{U}^h) = \frac{1}{2} h_d \sum_{i,j} (\mathcal{B}[\mathbf{u}^h])_{i,j} = \frac{1}{2} h_d \mathbb{B}^h[\mathbf{U}^h]$$

mit $h_d = h_1 h_2$ und es gilt

$$\begin{aligned} \mathbb{B}^h[\mathbf{U}^h] &= \left| \frac{-B^{\top} B \mathbf{u}_1^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}} \right|^2 + \left| \frac{-B^{\top} B \mathbf{u}_2^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}} \right|^2 \\ &= \frac{(\mathbf{u}_1^h)^{\top} B^{\top} B B^{\top} B \mathbf{u}_1^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}^2} + \frac{(\mathbf{u}_2^h)^{\top} B^{\top} B B^{\top} B \mathbf{u}_2^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}^2} \\ &= \frac{1}{|\mathbf{A} \mathbf{U}^h|_{\beta}^2} \left((\mathbf{u}_1^h)^{\top} B^{\top} B B^{\top} B \mathbf{u}_1^h + (\mathbf{u}_2^h)^{\top} B^{\top} B B^{\top} B \mathbf{u}_2^h \right) \\ &= \frac{1}{|\mathbf{A} \mathbf{U}^h|_{\beta}^2} \left((\mathbf{u}_1^h)^{\top}, (\mathbf{u}_2^h)^{\top} \right) \begin{bmatrix} B^{\top} B B^{\top} B & 0 \\ 0 & B^{\top} B B^{\top} B \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^h \\ \mathbf{u}_2^h \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{|\mathbf{A}\mathbf{U}^h|_\beta^2} (\mathbf{U}^h)^\top \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}^\top \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}^\top \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \mathbf{U}^h \\
 &= \frac{(\mathbf{U}^h)^\top \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{A} \mathbf{U}^h}{|\mathbf{A}\mathbf{U}^h|_\beta^2}.
 \end{aligned}$$

Damit lässt sich der Regularisierungsterm als

$$\mathcal{R}^{hNewC}(\mathbf{U}^h) = \frac{h_d (\mathbf{U}^h)^\top \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{A} \mathbf{U}^h}{2 |\mathbf{A}\mathbf{U}^h|_\beta^2} \quad (3.7)$$

schreiben.

Diskretisierung des Referenz- und Templatebildes

Die diskreten Bilder sollen aus den glatten interpolierten Approximationen des Referenzbildes R und des Templatebildes T gewonnen werden. Diese sollen über eine B-Spline-Interpolation erhalten werden und mit \mathcal{R} beziehungsweise \mathcal{T} bezeichnet werden. Die diskreten Bilder können über

$$R^h = \mathcal{R}(\mathbf{X}_c^h) \text{ bzw. } T^h = \mathcal{T}(\mathbf{X}_c^h + \mathbf{U}^h)$$

erhalten werden. Die Jakobi-Matrix von T^h ist

$$T_{\mathbf{U}^h}^h = \frac{\partial T^h}{\partial \mathbf{U}^h}(\mathbf{U}^h) = \frac{\partial \mathcal{T}}{\partial \mathbf{U}_c^h}(\mathbf{U}_c^h),$$

mit $\mathbf{U}_c^h = \mathbf{X}_c^h + \mathbf{U}^h$.

Diskretisierung des Distanzmaßes \mathcal{D}

Die Diskretisierung des Distanzmaßes \mathcal{D} wird über

$$\mathcal{D}^h(\mathbf{U}^h) = \frac{h_d}{2} (T^h(\mathbf{U}^h) - R^h)^\top (T^h(\mathbf{U}^h) - R^h)$$

bestimmt. Die erste Ableitung des diskreten Distanzmaßes bezüglich \mathbf{U}^h wird über

$$d\mathcal{D}^h(\mathbf{U}^h) = h_d (T^h(\mathbf{U}^h) - R^h)^\top T_{\mathbf{U}^h}^h$$

berechnet, denn es gilt

$$\begin{aligned}
 \frac{\partial}{\partial \mathbf{U}^h} \mathcal{D}^h(\mathbf{U}^h) &= \frac{\partial}{\partial \mathbf{U}^h} \left[\frac{h_d}{2} (T^h(\mathbf{U}^h) - R^h)^\top (T^h(\mathbf{U}^h) - R^h) \right] \\
 &= \frac{h_d}{2} \left(\frac{\partial}{\partial \mathbf{U}^h} \left[(T^h(\mathbf{U}^h) - R^h)^\top \right] (T^h(\mathbf{U}^h) - R^h) \right. \\
 &\quad \left. + (T^h(\mathbf{U}^h) - R^h)^\top \frac{\partial}{\partial \mathbf{U}^h} \left[T^h(\mathbf{U}^h) - R^h \right] \right) \\
 &= \frac{h_d}{2} \left((T_{\mathbf{U}^h}^h)^\top (T^h(\mathbf{U}^h) - R^h) + ((T^h(\mathbf{U}^h))^\top - (R^h)^\top) T_{\mathbf{U}^h}^h \right)
 \end{aligned}$$

3. Modellbeschreibung in 2D

$$= h_d T_{\mathbf{U}^h}^h (T^h(\mathbf{U}^h) - R^h).$$

Die Formel für die zweite Ableitung $\partial^2 \mathcal{D}^h(\mathbf{U}^h)$ des Distanzmaßes \mathcal{D} ist in [32] als

$$\partial^2 \mathcal{D}^h(\mathbf{U}^h) = h_d (T_{\mathbf{U}^h}^h)^\top T_{\mathbf{U}^h}^h$$

gegeben. Dies stellt jedoch nur eine Approximation der zweiten Ableitung dar. Auf die exakte Form der zweiten Ableitung des Distanzmaßes, die man durch direktes Nachrechnen erhält, wird erst in den später veröffentlichten Publikationen [33, 34] eingegangen. Die exakte Form lautet

$$\partial^2 \mathcal{D}^h(\mathbf{U}^h) = h_d (T_{\mathbf{U}^h}^h)^\top T_{\mathbf{U}^h}^h + h_d \sum_{i=1}^{m_1 m_2} d_i(\mathbf{U}^h) \nabla^2 d_i(\mathbf{U}^h)$$

mit $d(\mathbf{U}^h) = T(\mathbf{U}^h) - R \in \mathbb{R}^{m_1 m_2}$. Durch Verwendung der Approximation müssen keine Ableitungen höherer Ordnung bestimmt werden, deren Berechnung für ein Registrierungsproblem in praktischen Anwendungen rechenaufwendig und numerisch instabil ist [33, 34]. Außerdem soll durch Lösen des Registrierungsproblems die Differenz zwischen $T(\mathbf{U}^h)$ und R ohnehin immer kleiner werden, sodass dieser Teil vernachlässigbar ist. Des Weiteren wird für die numerischen Algorithmen zur Lösung des Optimierungsproblems eine positiv semidefinite Matrix benötigt, was nur durch die Verwendung der approximierten Form stets gewährleistet ist.

3.1.2. Lösen des diskreten Optimierungsproblems

Das diskrete Analogon zum Bildregistrierungsproblem (3.5) mit $\mathbf{U}^h \in \mathbb{R}^{2m_1 m_2}$ lautet

$$\min_{\mathbf{U}^h} \{ \mathcal{J}_\alpha(\mathbf{U}^h) = \mathcal{D}^h(\mathbf{U}^h) + \alpha \mathcal{R}^{hNewC}(\mathbf{U}^h) \}$$

und mit eingesetztem Regularisierungsterm

$$\min_{\mathbf{U}^h} \left\{ \mathcal{J}_\alpha(\mathbf{U}^h) = \mathcal{D}^h(\mathbf{U}^h) + \frac{\alpha}{2} h_d \frac{(\mathbf{U}^h)^\top \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{A} \mathbf{U}^h}{|\mathbf{A} \mathbf{U}^h|_\beta^2} \right\}. \quad (3.8)$$

Das Funktional in der algebraischen Form ist nichtlinear. Um eine Lösung des Optimierungsproblems zu finden, muss dieses zweimal differenziert werden. Die Berechnung der Ableitung kann vereinfacht werden, indem die Lösung aus dem vorherigen Iterationsschritt im neuen Regularisierungsterm verwendet wird. Dies wird dadurch erreicht, dass die vorher berechnete und damit bekannte Iterierte $\mathbf{U}^{h(k)}$ im Nenner des Regularisierungsterms eingesetzt wird. Damit erhält man das Optimierungsproblem

$$\min_{\mathbf{U}^h} \left\{ \hat{\mathcal{J}}_\alpha(\mathbf{U}^h) = \mathcal{D}^h(\mathbf{U}^h) + \frac{\alpha}{2} h_d \frac{(\mathbf{U}^h)^\top \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{A} \mathbf{U}^h}{|\mathbf{A} \mathbf{U}^{h(k)}|_\beta^2} \right\}. \quad (3.9)$$

3.1. Numerische Lösung des Registrierungsproblems

Dies ist äquivalent dazu, dass \mathcal{J}_α durch ein quasi-quadratisches $\hat{\mathcal{J}}_\alpha$ ersetzt wird, das in einer Taylorentwicklung

$$\mathcal{J}_\alpha(\mathbf{U}^{h^{(k)}} + \delta_{\mathbf{U}^h}) \approx \hat{\mathcal{J}}_\alpha(\mathbf{U}^{h^{(k)}} + \delta_{\mathbf{U}^h}) = \hat{\mathcal{J}}_\alpha(\mathbf{U}^{h^{(k)}}) + d\hat{\mathcal{J}}_\alpha(\mathbf{U}^{h^{(k)}})\delta_{\mathbf{U}^h} + \frac{1}{2}\delta_{\mathbf{U}^h}^\top \mathbf{H}\delta_{\mathbf{U}^h},$$

erhalten werden kann. Hierbei gilt für den Gradienten $d\hat{\mathcal{J}}_\alpha(\mathbf{U}^{h^{(k)}})$ an der bekannten Iterierten $\mathbf{U}^{h^{(k)}}$

$$d\hat{\mathcal{J}}_\alpha(\mathbf{U}^{h^{(k)}}) = d\mathcal{D}^h(\mathbf{U}^{h^{(k)}}) + \alpha h_d \frac{(\mathbf{U}^{h^{(k)}})^\top \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{A}}{|\mathbf{A}\mathbf{U}^{h^{(k)}}|_\beta^2}. \quad (3.10)$$

Für die Matrix \mathbf{H} , die positiv und semidefinit ist, wird eine durch die verwendete Verzögerung hervorgerufene Approximation der Hesse-Matrix

$$\mathbf{H} = d^2\mathcal{D}^h(\mathbf{U}^{h^{(k)}}) + \alpha h_d \frac{\mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{A}}{|\mathbf{A}\mathbf{U}^{h^{(k)}}|_\beta^2} \quad (3.11)$$

gewählt.

Um das diskrete Registrierungsproblem (3.8) zu lösen, wird, wie in Kapitel 2.3.5 angesprochen, das Gauß-Newton Verfahren verwendet. Dieses hat hier nun die Form:

$$\mathbf{U}^{h^{(k+1)}} = \mathbf{U}^{h^{(k)}} + \delta_{\mathbf{U}^h}, \quad \mathbf{H}\delta_{\mathbf{U}^h} = -d\hat{\mathcal{J}}_\alpha(\mathbf{U}^{h^{(k)}}). \quad (3.12)$$

Um einen hinreichenden Abstieg der Zielfunktion zu gewährleisten, wird die Armijo-Abbruchbedingung genutzt, wodurch Gleichung (3.12) die Form

$$\mathbf{U}^{h^{(k+1)}} = \mathbf{U}^{h^{(k)}} + t\delta_{\mathbf{U}^h}, \quad \mathbf{H}\delta_{\mathbf{U}^h} = -d\hat{\mathcal{J}}_\alpha(\mathbf{U}^{h^{(k)}}).$$

annimmt. Der Algorithmus iteriert über diese Gleichungen so oft, bis das Abbruchkriterium erfüllt ist. Dabei läuft die Armijo-Liniensuche so ab, dass mit $t = 1$ begonnen wird. Die Zielfunktion $\mathcal{J}_\alpha(\mathbf{U}^{h^{(k+1)}})$ der neuen Iterierten $\mathbf{U}^{h^{(k+1)}} + t\delta_{\mathbf{U}^h}$ wird mit dem alten Wert $\mathcal{J}_\alpha(\mathbf{U}^{h^{(k)}})$ verglichen. Falls die Armijo-Bedingung nicht erfüllt ist, wird t durch $\frac{1}{2}t$ ersetzt und das Verfahren wiederholt. Die Armijo-Bedingung hat hier die folgende Form

$$\mathcal{J}_\alpha(\mathbf{U}^{h^{(k+1)}}) < \mathcal{J}_\alpha(\mathbf{U}^{h^{(k)}}) + t((d\hat{\mathcal{J}}_\alpha(\mathbf{U}^{h^{(k)}}))^\top \mathbf{U}^{h^{(k)}})tol,$$

wobei standardmäßig $tol = 10^{-4}$ gewählt wird. Die genaue Beschreibung des hier verwendeten Gauß-Newton Verfahrens lässt sich in Algorithmus 1 und die der Armijo-Liniensuche in Algorithmus 2 nachlesen. Es sei hier angemerkt, dass sich in der Implementierung des Gauß-Newton Verfahrens eine vorkonditionierte konjugierte Gradientenmethode als nützlich erwiesen hat und angewandt wurde. Für weitere Erläuterungen dazu sei auf [29] verwiesen.

3. Modellbeschreibung in 2D

Algorithmus 1 Gauß-Newton

$\mathbf{u} \leftarrow \text{GaussNewton}(\alpha, \mathbf{u})$

```

1: Setze  $\text{iter} = 0$ 
2: while wahr do
3:   Berechne  $\mathcal{J}_\alpha(\mathbf{u})$ ,  $d\hat{\mathcal{J}}_\alpha(\mathbf{u})$  und  $\mathbf{H}$  mithilfe von (3.9), (3.10) und (3.11);
4:   Überprüfe die Abbruchkriterien;
5:   Erhöhe den Iterationsschritt  $\text{iter} \leftarrow \text{iter} + 1$ ;
6:   Berechne die Gauß-Newton-Gleichung:  $\mathbf{H}\delta_{\mathbf{u}} = -d\hat{\mathcal{J}}_\alpha(\mathbf{u})$  unter Verwendung
   einer vorkonditionierten konjugierten Gradientenmethode;
7:   Führe eine Armijo-Liniensuche durch:  $\mathbf{u}_c \leftarrow \text{Armijo}(\alpha, \delta_{\mathbf{u}}, \mathbf{u})$ ;
8:   if Liniensuche fehlschlägt then
9:     break;
10:  else
11:    Erhöhe aktuelle Werte:  $\mathbf{u} \leftarrow \mathbf{u}_t$ ;
12:  end if
13: end while

```

Algorithmus 2 Armijo Liniensuche

$\mathbf{u} \leftarrow \text{Armijo}(\alpha, \delta_{\mathbf{u}}, \mathbf{u})$

```

1: Berechne  $\mathcal{J}_\alpha(\mathbf{u})$ ,  $d\hat{\mathcal{J}}_\alpha(\mathbf{u})$  und  $\mathbf{H}$  mithilfe von (3.9), (3.10) und (3.11);
2: Setze  $\mathbf{k} \leftarrow 0$ ,  $\mathbf{t} \leftarrow 1$ ,  $\text{MaxIter} \leftarrow 10$  und  $\eta \leftarrow 10\text{e-}4$ ;
3: while wahr do
4:   Setze  $\mathbf{u}_t \leftarrow \mathbf{u} + \mathbf{t} * \delta_{\mathbf{u}}$ ;
5:   Berechne  $\mathcal{J}_\alpha(\mathbf{u}_t)$  unter Verwendung von (3.9);
6:   if  $\mathcal{J}_\alpha(\mathbf{u}_t) < \mathcal{J}_\alpha(\mathbf{u}) + \mathbf{t} * \eta(d\hat{\mathcal{J}}_\alpha(\mathbf{u}))^\top \delta_{\mathbf{u}}$  ||  $\mathbf{k} > \text{MaxIter}$  then
7:     break;
8:   end if
9:   Setze  $\mathbf{t} \leftarrow \frac{\mathbf{t}}{2}$ ;
10: end while
11: Setze  $\mathbf{u} \leftarrow \mathbf{u}_t$ ;

```

Als Abbruchkriterien wurden in dieser Arbeit etwas andere Kriterien verwendet, als in [32] angegeben sind. Dies sollte jedoch das Ergebnis nicht wesentlich ändern. Die Abbruchkriterien, die verwendet wurden, werden im Folgenden dargestellt. Dabei sei $\mathcal{J}_c = \mathcal{J}_\alpha(\mathbf{U}^{h^{(k+1)}})$, $\mathcal{J}_{old} = \mathcal{J}_\alpha(\mathbf{U}^{h^{(k)}})$ und \mathbf{u}_0 sei der Initialwert auf Level k . Für die benötigten Toleranzen wurden die Werte $\text{tolJ} = 1\text{e-}3$, $\text{tolY} = 1\text{e-}2$ und $\text{tolG} = 1\text{e-}2$ gewählt, wobei tolJ die Toleranz der Zielfunktion, tolY die Toleranz des aktuellen Wertes und tolG die Toleranz der Norm des Gradienten angibt.

```

1: STOP(1):  $(\text{iter} > 0) \ \&\& \ (|\mathcal{J}_{old} - \mathcal{J}_c| \leq \text{tolJ} * (1 + |\mathcal{J}_{stop}|)$ ;
2: STOP(2):  $(\text{iter} > 0) \ \&\& \ (\text{norm}(\mathbf{u}_c - \mathbf{u}_{old}) \leq \text{tolY} * (1 + \text{norm}(\mathbf{u}_0)))$ ;
3: STOP(3):  $\text{norm}(d\mathcal{J}) \leq \text{tolG} * (1 + |\mathcal{J}_{stop}|)$ ;

```

3.2. Erläuterungen zur Implementierung in Matlab

```
4: STOP(4): norm(dJ) ≤ 1e6*eps;  
5: STOP(5): (iter ≥ maxIter);  
6: if all (STOP(1:3)) || any (STOP(4:5)), break; end;
```

Um Rechenzeit zu sparen und die Konvergenz des Algorithmus zu beschleunigen, wurde wie in [32] das Gauß-Newton Verfahren mit einem Multilevelansatz verbunden. Bei einem Multilevelansatz wird, wie in [26] nachzulesen ist, das Registrierungsproblem auf mehreren Leveln berechnet, wobei sich die Level in der Grobheit der Diskretisierung unterscheiden. Man beginnt die Berechnungen auf dem größten Level, was ein relativ kleines Problem ist und meistens schnell durchgeführt ist. Die erhaltene Lösung wird dann auf das nächstfeinere Level prolongiert und für dieses Level als Startwert genutzt. Dadurch ist gewährleistet, dass der Startwert in der Nähe des Minimierers des feineren Levels liegt. Durch diese Herangehensweise sinkt das Risiko, in einem lokalen Minimum stecken zu bleiben. Wird der Multilevelansatz angewandt, sind wenige Schritte zur Konvergenz nötig und Rechenzeit wird eingespart. Die genaue Beschreibung des Multilevelansatzes ist in Algorithmus 3 nachzulesen.

Algorithmus 3 Multilevelregistrierung $\mathbf{u} \leftarrow \text{MLIR}(\text{MLData})$

```
1: MaxLevel ← L      % das feinste Level;  
2: MinLevel ← 3      % das größte Level;  
3: Input MLData      % Multilevel Darstellung der gegebenen Bilder R und T;  
4: for l = MinLevel : MaxLevel do  
5:   if l == MinLevel then  
6:     Erhalte erste Approximation für  $\mathbf{u}_0$  aus einer Multilevel affin linearen Vor-  
       registrierung;  
7:      $\mathbf{u}_0 \leftarrow \mathbf{u}_0$ ;  
8:   else  
9:      $\mathbf{u} \leftarrow I_H^h(\mathbf{u})$ ;      %  $I_H^h(\mathbf{u})$  Interpolationsoperator  
10:  end if  
11:   $\mathbf{u} \leftarrow \text{GaussNewton}(\alpha, \mathbf{u}_0)$ ;  
12: end for
```

Für eine genaue Ausführung des Interpolationsoperators sei an dieser Stelle auf [26, Kap. 3] verwiesen.

3.2. Erläuterungen zur Implementierung in Matlab

Die Implementierung erfolgte in MATLAB, wobei als Grundlage die FAIR-Toolbox [28] verwendet wurde. Für die Einbettung des in dieser Arbeit untersuchten Regularisierungsterms wurden neue Funktionen erstellt, die die Formulierung der Gradientenmatrix, die Diskretisierung und die Berechnung des Regularisierungsterms enthalten. Außerdem mussten bereits bestehende Funktionen verändert werden, da die

3. Modellbeschreibung in 2D

Implementierung über das Minimierungsproblem (3.9) erfolgte, das Informationen aus einem vorher berechneten Schritt benötigt. Im Zuge dessen mussten die Funktionen `NPIRobjFct.m`, `MLIR.m`, `GaussNewton.m` und `Armijo.m` verändert werden, um die Speicherung und Wiederverwendung bereits erfolgter Berechnungen zu gewährleisten. Auf Einzelheiten der Implementierung soll hier nicht eingegangen werden, allerdings sollen im Folgenden ein paar wesentliche Dinge angesprochen werden, die die Art und Weise der Implementierung betreffen.

Einerseits betrifft es die Berechnung des Regularisierungsterms (3.7). Der Zähler dieses Terms besteht ausschließlich aus Matrixmultiplikationen. Diese können direkt in MATLAB implementiert werden. Der Nenner jedoch enthält eine Norm, für die eine geeignete Norm gewählt werden muss. In der Implementierung dieser Arbeit wurde eine β -Frobeniusnorm gewählt, die für eine Matrix $A \in \mathbb{R}^{n \times m}$ mit Matrixelementen a_{ij} die folgende Gestalt hat:

$$|A|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 + \beta},$$

wobei für β eine positive Zahl größer als Null gewählt wird, um Singularitäten zu vermeiden.

Andererseits betrifft es die Implementierung der Gradientenmatrix (3.6), bei der bei genauerer Untersuchung mehrere Dinge aufgefallen sind. Zum einen ist die dort formulierte Matrix nicht allgemeingültig, das heißt, über die dort aufgeführten Fallunterscheidungen wird nur eine korrekte Gradientenmatrix bestimmt, falls $m_1 = m_2$ gilt. Da dies jedoch nicht immer der Fall ist, wurde zunächst die Matrix allgemeingültig formuliert.

$$A_k^\top \mathbf{u}_l^h = \begin{cases} ((\mathbf{u}_l^h)_{k+1} - (\mathbf{u}_l^h)_k; (\mathbf{u}_l^h)_{k+m_2} - (\mathbf{u}_l^h)_k), & \text{falls } k \bmod m_1 \neq 0 \text{ und } k + m_1 \leq N \\ (0; (\mathbf{u}_l^h)_{k+m_2} - (\mathbf{u}_l^h)_k), & \text{falls } k \bmod m_1 = 0 \text{ und } k + m_1 \leq N \\ ((\mathbf{u}_l^h)_{k+1} - (\mathbf{u}_l^h)_k; 0), & \text{falls } k \bmod m_1 \neq 0 \text{ und } k + m_1 > N \\ (0; 0), & \text{falls } k \bmod m_1 = 0 \text{ und } k + m_1 > N. \end{cases}$$

Eine direkte Implementierung dieser Matrix über die hier aufgestellten Fallunterscheidungen ist möglich, aber nicht sehr effizient. Aus diesem Grund wurde die Matrix vor der Implementierung über eine kürzere und effizienter zu berechnende Form ausgedrückt. Dabei wurden das Kroneckerprodukt (siehe Definition 2.18 in Kapitel 2.3.2) und die MATLAB internen Funktionen `spdiags` für dünnbesetzte Diagonalmatrizen und `speye` für die speicherarme Implementierung der Einheitsmatrix ausgenutzt.

Für die Matrix A_{x_1} wird eine Diagonalmatrix \mathbf{D}_x erstellt, die auf der Hauptdiagonalen -1 und auf ihrer rechten Nebendiagonalen 1 stehen hat. Da Neumann-Randbedingungen angenommen werden, wird die letzte Zeile gleich Null gesetzt.

3.2. Erläuterungen zur Implementierung in Matlab

Um alle Gradienten in x_1 -Richtung zu erhalten, wird das Kroneckerprodukt einer Einheitsmatrix der Größe $m_2 \times m_2$ mit der Matrix Dx berechnet.

```
1: % Matrix  $A_{x_1}$  (Ableitungen nach rechts)
2:  $Dx = \text{spdiags}(\text{ones}(m(1),1)*[-1,1],0:1,m(1),m(1));$ 
3: % Neumann-Randbedingung
4:  $Dx(\text{end},:) = 0;$ 
5:  $Ax1 = \text{kron}(\text{speye}(m(2)),Dx);$ 
```

Für die Matrix A_{x_2} wird eine Diagonalmatrix Dy erstellt, die auf ihrer Hauptdiagonalen -1 und auf ihrer m_1 -ten rechten Nebendiagonalen 1 stehen hat. Eine Nullmatrix, die hier mit `rest` bezeichnet wird, wird an die Matrix Dy gesetzt. Dies gewährleistet, dass die Neumann-Randbedingungen nach oben eingehalten werden.

```
1: % Matrix  $A_{x_2}$  (Ableitungen nach oben)
2:  $Dy = \text{spdiags}(\text{ones}(m(1)*m(2)-m(1),1)*[-1,1],\dots$ 
3:    $0:m(1):m(1),m(1)*m(2)-m(1),m(1)*m(2));$ 
4:  $\text{rest} = \text{sparse}((m(1)*m(2))-size(Dy,1),size(Dy,2));$ 
5:  $Ax2 = [Dy;\text{rest}];$ 
```

In diesem Kapitel erfolgte zum einen die Beschreibung des in [32] eingeführten Regularisierungsmodells. Zum anderen wurde dieser theoretisch analysiert, indem die Rotations- und Translationsinvarianz bewiesen und die Euler-Lagrange-Gleichungen bestimmt worden sind. Außerdem wurde über die exakte Formulierung des Regularisierers diskutiert und für die Implementierung eine bessere Form gefunden.

Im nächsten Kapitel wird der Aussage in [32], dass dieser Regularisierer ohne Weiteres auf drei Dimensionen erweitert werden kann, nachgegangen.

4. Erweiterung des Modells auf 3D

Dieses Kapitel befasst sich mit der Anmerkung in [32], dass das dort entwickelte Regularisierungsmodell ohne Weiteres auf drei Dimensionen erweitert werden kann. Diese Aussage wird in diesem Kapitel überprüft, indem jeder einzelne Schritt, der im Kapitel 3.1.1 zum diskreten zweidimensionalen Regularisierungsterm geführt hat, um eine Dimension erweitert wird.

4.1. Diskretisierung

Zunächst soll hier darauf eingegangen werden, wie ein dreidimensionales Bildgebiet über ein zell-zentriertes Gitter diskretisiert werden kann. Anschließend wird der diskrete Regularisierungsterm analog zum zweidimensionalen Fall hergeleitet.

Finite Differenzen Diskretisierung

Die dreidimensionalen diskreten Bilder sollen $m_1 \times m_2 \times m_3$ Pixel haben. Für den Bildbereich gelte $\Omega = [0, \omega_1] \times [0, \omega_2] \times [0, \omega_3] \subset \mathbb{R}^3$ mit Gitterweite $h_i = \omega_i/m_i, i = 1, 2, 3$. Wie in der zweidimensionalen Diskretisierung wird ein zell-zentriertes Gitter mit lexicographischer Anordnung der Gitterpunkte angenommen. Das diskrete Gebiet in Ω wird dann über

$$\Omega_h = \left\{ \mathbf{x} \mid \mathbf{x} = \begin{pmatrix} x_{1_{j_1}} \\ x_{2_{j_2}} \\ x_{3_{j_3}} \end{pmatrix} = \begin{pmatrix} (j_1 - 0.5)h_1 \\ (j_2 - 0.5)h_2 \\ (j_3 - 0.5)h_3 \end{pmatrix}, \begin{matrix} j_1 = 1, 2, \dots, m_1 \\ j_2 = 1, 2, \dots, m_2 \\ j_3 = 1, 2, \dots, m_3 \end{matrix} \right\} \quad (4.1)$$

beschrieben. Eine Veranschaulichung dieser Diskretisierung ist in Abbildung 4.1 zu sehen. Die beiden Darstellungen werden hintereinander betrachtet. Zunächst wird, wie links in Abbildung 4.1 zu sehen ist, $j_3 = 1$ festgehalten und eine zell-zentrierte Diskretisierung in der x_1x_2 -Ebene wie im Zweidimensionalen durchgeführt. Anschließend wird j_3 um eins erhöht und es wird, wie rechts in Abbildung 4.1 zu sehen ist, wieder eine zell-zentrierte Diskretisierung in der x_1x_2 -Ebene durchgeführt. Dies wird so lange wiederholt, bis man bei $j_3 = m_3$ angekommen ist. Tritt dieser Fall ein, wird ein letztes Mal eine zell-zentrierte Diskretisierung in der x_1x_2 -Ebene durchgeführt. Daraufhin ist die Diskretisierung des gesamten dreidimensionalen Bildgebietes abgeschlossen.

4. Erweiterung des Modells auf 3D

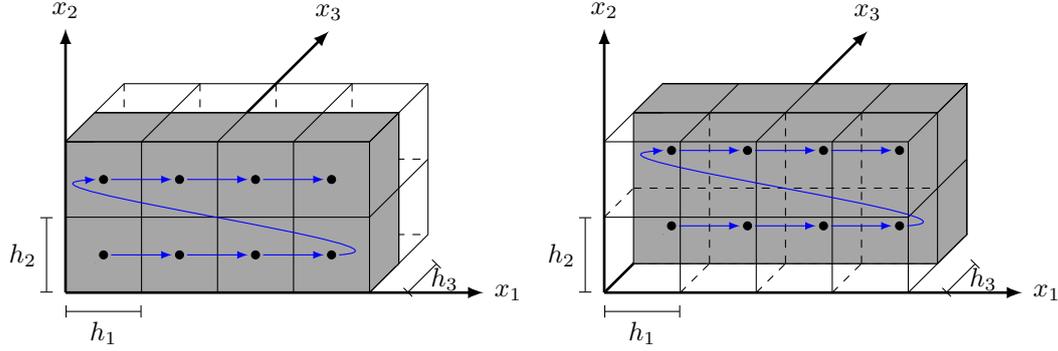


Abbildung 4.1: Darstellung der Diskretisierung eines dreidimensionalen Bildgebietes über ein zell-zentriertes Gitter mit $m_1 = 4$, $m_2 = 2$ und $m_3 = 2$. Die Pfeile zeigen die lexikographische Anordnung der Gitterpunkte an. Die beiden Darstellungen müssen hintereinander betrachtet werden. Links: Zunächst wird eine zell-zentrierte Diskretisierung der vorderen Schicht durchgeführt. Rechts: Anschließend wird eine zell-zentrierte Diskretisierung der hinteren Schicht durchgeführt.

Punkte, die in der Menge (4.1) liegen, können analog zum zweidimensionalen Fall in Vektorform gebracht werden,

$$\begin{aligned} \mathbf{x}_{c,1} &= [x_{1,1,1,1}, x_{1,2,1,1}, \dots, x_{1,m_1,1,1}, x_{1,1,2,1}, x_{1,2,2,1}, \dots, x_{1,m_1,2,1}, x_{1,1,m_2,1}, x_{1,2,m_2,1}, \dots, \\ &\quad x_{1,m_1,m_2,1}, x_{1,1,1,m_3}, x_{1,2,1,m_3}, \dots, x_{1,m_1,1,m_3}, x_{1,1,2,m_3}, x_{1,2,2,m_3}, \dots, x_{1,m_1,2,m_3}, \\ &\quad x_{1,1,m_2,m_3}, x_{1,2,m_2,m_3}, \dots, x_{1,m_1,m_2,m_3}]^\top, \\ \mathbf{x}_{c,2} &= [x_{2,1,1,1}, x_{2,2,1,1}, \dots, x_{2,m_1,1,1}, x_{2,1,2,1}, x_{2,2,2,1}, \dots, x_{2,m_1,2,1}, x_{2,1,m_2,1}, x_{2,2,m_2,1}, \dots, \\ &\quad x_{2,m_1,m_2,1}, x_{2,1,1,m_3}, x_{2,2,1,m_3}, \dots, x_{2,m_1,1,m_3}, x_{2,1,2,m_3}, x_{2,2,2,m_3}, \dots, x_{2,m_1,2,m_3}, \\ &\quad x_{2,1,m_2,m_3}, x_{2,2,m_2,m_3}, \dots, x_{2,m_1,m_2,m_3}]^\top, \\ \mathbf{x}_{c,3} &= [x_{3,1,1,1}, x_{3,2,1,1}, \dots, x_{3,m_1,1,1}, x_{3,1,2,1}, x_{3,2,2,1}, \dots, x_{3,m_1,2,1}, x_{3,1,m_2,1}, x_{3,2,m_2,1}, \dots, \\ &\quad x_{3,m_1,m_2,1}, x_{3,1,1,m_3}, x_{3,2,1,m_3}, \dots, x_{3,m_1,1,m_3}, x_{3,1,2,m_3}, x_{3,2,2,m_3}, \dots, x_{3,m_1,2,m_3}, \\ &\quad x_{3,1,m_2,m_3}, x_{3,2,m_2,m_3}, \dots, x_{3,m_1,m_2,m_3}]^\top. \end{aligned}$$

Durch Zusammenfassen der Vektoren $\mathbf{x}_{c,1}, \mathbf{x}_{c,2}, \mathbf{x}_{c,3}$ in einen Vektor vergrößert sich der Vektor \mathbf{X}_c^h im dreidimensionalen Fall zu dem Vektor $\mathbf{X}_c^h = [\mathbf{x}_{c,1}; \mathbf{x}_{c,2}; \mathbf{x}_{c,3}]$.

Diskretisierung des Krümmungsregularisierers in 3D

Wie im zweidimensionalen Fall wird analog dazu im Dreidimensionalen das kontinuierliche Verschiebungsfeld mit $\mathbf{u} = (u_1, u_2, u_3)^\top$ bezeichnet. Das diskrete Analogon dazu wird entsprechend im diskreten Gebiet Ω_h mit $\mathbf{u}^h = (u_1^h, u_2^h, u_3^h)^\top$ bezeichnet, wobei u_1^h, u_2^h und u_3^h Gitterfunktionen sind. Für diese gilt $(u_l^h)_{j_1, j_2, j_3} = u_l^h(x_{1,j_1}, x_{2,j_2}, x_{3,j_3}), j_1 = 1, 2, \dots, m_1; j_2 = 1, 2, \dots, m_2; j_3 = 1, 2, \dots, m_3$ mit $l = 1, 2, 3$. Der diskrete Gradientenoperator ∇^h an jedem Voxel (j_1, j_2, j_3) ist definiert

über

$$(\nabla^h \mathbf{u}^h)_{j_1, j_2, j_3} = ((\nabla^{h,1} u_1^h)_{j_1, j_2, j_3}, (\nabla^{h,2} u_2^h)_{j_1, j_2, j_3}, (\nabla^{h,3} u_3^h)_{j_1, j_2, j_3})^\top,$$

wobei

$$\begin{aligned} (\nabla^{h,l} u_l^h)_{j_1, j_2, j_3} &= ((\partial_1^{h,l} u_l^h)_{j_1, j_2, j_3}, (\partial_2^{h,l} u_l^h)_{j_1, j_2, j_3}, (\partial_3^{h,l} u_l^h)_{j_1, j_2, j_3})^\top, \\ (\partial_1^{h,l} u_l^h)_{j_1, j_2, j_3} &= \begin{cases} (u_l^h)_{j_1+1, j_2, j_3} - (u_l^h)_{j_1, j_2, j_3}, & \text{falls } j_1 < m_1 \\ 0, & \text{falls } j_1 = m_1, \end{cases} \\ (\partial_2^{h,l} u_l^h)_{j_1, j_2, j_3} &= \begin{cases} (u_l^h)_{j_1, j_2+1, j_3} - (u_l^h)_{j_1, j_2, j_3}, & \text{falls } j_2 < m_2 \\ 0, & \text{falls } j_2 = m_2, \end{cases} \\ (\partial_3^{h,l} u_l^h)_{j_1, j_2, j_3} &= \begin{cases} (u_l^h)_{j_1, j_2, j_3+1} - (u_l^h)_{j_1, j_2, j_3}, & \text{falls } j_3 < m_3 \\ 0, & \text{falls } j_3 = m_3. \end{cases} \end{aligned}$$

mit Neumann-Randbedingungen $\frac{\partial u_l}{\partial \nu} = 0$, $l = 1, 2, 3$ auf $\partial\Omega$ gilt. Analog zum zweidimensionalen Fall wird auch im dreidimensionalen Fall ausgenutzt, dass der Divergenzoperator über den adjungierten Gradienten ausgedrückt werden kann. Der Divergenzoperator kann im Dreidimensionalen analog zum zweidimensionalen Fall geschrieben werden als

$$\begin{aligned} (\nabla \cdot \omega_l)_{j_1, j_2, j_3} &= \begin{cases} (\omega_l^1)_{j_1, j_2, j_3} - (\omega_l^1)_{j_1-1, j_2, j_3} \\ (\omega_l^1)_{j_1, j_2, j_3} \\ -(\omega_l^1)_{j_1-1, j_2, j_3} \end{cases} \\ &+ \begin{cases} (\omega_l^2)_{j_1, j_2, j_3} - (\omega_l^2)_{j_1, j_2-1, j_3} \\ (\omega_l^2)_{j_1, j_2, j_3} \\ -(\omega_l^2)_{j_1, j_2-1, j_3} \end{cases} \\ &+ \begin{cases} (\omega_l^3)_{j_1, j_2, j_3} - (\omega_l^3)_{j_1, j_2, j_3-1}, & \text{falls } 1 < j_i < m_i, \quad i = 1, 2, 3 \\ (\omega_l^3)_{j_1, j_2, j_3}, & \text{falls } j_1 = j_2 = j_3 = 1, \\ -(\omega_l^3)_{j_1, j_2, j_3-1}, & \text{falls } j_1 = m_1, j_2 = m_2, j_3 = m_3. \end{cases} \end{aligned}$$

Die Gitterfunktionen u_1^h , u_2^h und u_3^h werden als Spaltenvektoren \mathbf{u}_1^h , \mathbf{u}_2^h und \mathbf{u}_3^h in lexikographischer Reihenfolge angeordnet, und zwar

$$\begin{aligned} \mathbf{u}_1^h &= [u_{1,1,1}^h, u_{1,2,1}^h, \dots, u_{1,m_1,1,1}^h, u_{1,1,2,1}^h, u_{1,2,2,1}^h, \dots, u_{1,m_1,2,1}^h, u_{1,1,m_2,1}^h, u_{1,2,m_2,1}^h, \dots, \\ &u_{1,m_1,m_2,1}^h, u_{1,1,m_3}^h, u_{1,2,m_3}^h, \dots, u_{1,m_1,1,m_3}^h, u_{1,1,2,m_3}^h, u_{1,2,2,m_3}^h, \dots, u_{1,m_1,2,m_3}^h, \\ &u_{1,1,m_2,m_3}^h, u_{1,2,m_2,m_3}^h, \dots, u_{1,m_1,m_2,m_3}^h]^\top, \\ \mathbf{u}_2^h &= [u_{2,1,1,1}^h, u_{2,2,1,1}^h, \dots, u_{2,m_1,1,1}^h, u_{2,1,2,1}^h, u_{2,2,2,1}^h, \dots, u_{2,m_1,2,1}^h, u_{2,1,m_2,1}^h, u_{2,2,m_2,1}^h, \dots, \\ &u_{2,m_1,m_2,1}^h, u_{2,1,m_3}^h, u_{2,2,m_3}^h, \dots, u_{2,m_1,1,m_3}^h, u_{2,1,2,m_3}^h, u_{2,2,2,m_3}^h, \dots, u_{2,m_1,2,m_3}^h, \\ &u_{2,1,m_2,m_3}^h, u_{2,2,m_2,m_3}^h, \dots, u_{2,m_1,m_2,m_3}^h]^\top, \end{aligned}$$

4. Erweiterung des Modells auf 3D

$$\mathbf{u}_3^h = [u_{3_{1,1,1}}^h, u_{3_{2,1,1}}^h, \dots, u_{3_{m_1,1,1}}^h, u_{3_{1,2,1}}^h, u_{3_{2,2,1}}^h, \dots, u_{3_{m_1,2,1}}^h, u_{3_{1,m_2,1}}^h, u_{3_{2,m_2,1}}^h, \dots, u_{3_{m_1,m_2,1}}^h, u_{3_{1,1,m_3}}^h, u_{3_{2,1,m_3}}^h, \dots, u_{3_{m_1,1,m_3}}^h, u_{3_{1,2,m_3}}^h, u_{3_{2,2,m_3}}^h, \dots, u_{3_{m_1,2,m_3}}^h, u_{3_{1,m_2,m_3}}^h, u_{3_{2,m_2,m_3}}^h, \dots, u_{3_{m_1,m_2,m_3}}^h]^\top,$$

wobei $\mathbf{u}_1^h, \mathbf{u}_2^h, \mathbf{u}_3^h \in \mathbb{R}^N$ wieder in einem Vektor zusammengefasst werden. Es gilt $\mathbf{U}^h = (\mathbf{u}_1^h; \mathbf{u}_2^h; \mathbf{u}_3^h) \in \mathbb{R}^{3N}$ mit $N = m_1 m_2 m_3$. Der diskrete Gradient $(\nabla^{h,l} u_l^h)_{j_1, j_2, j_3}$ kann über eine Multiplikation einer Matrix $A_k^\top \in \mathbb{R}^{3 \times N}$, $k = 1, 2, \dots, N$ mit dem Vektor \mathbf{u}_l^h ($l = 1, 2, 3$) ausgedrückt werden. An dieser Stelle muss aufgepasst werden, denn die Formulierung der Gradientenmatrix, wie sie in [32] für den zweidimensionalen Fall vorgeschlagen wurde, lässt sich nicht ohne Weiteres auf den dreidimensionalen Fall übertragen. Zum einen soll hier direkt beachtet werden, dass m_1, m_2 und m_3 beliebige Werte annehmen können. Zum anderen müssen die Neumann-Randbedingungen nicht nur wie im Zweidimensionalen rechts, sondern auch nach oben und nach hinten berücksichtigt werden. Dazu gehört, dass sichergestellt wird, dass der Übergang zwischen zwei Schichten, beispielsweise Schicht $j_3 = 1$ und Schicht $j_3 = 2$, korrekt berücksichtigt wird. Welche Gradienten an welcher Stelle berechnet werden müssen, ist in Abbildung 4.2 dargestellt. Dabei zeigen die blauen/roten/grünen Pfeile die Gradienten in $x_1/x_2/x_3$ -Richtung an.

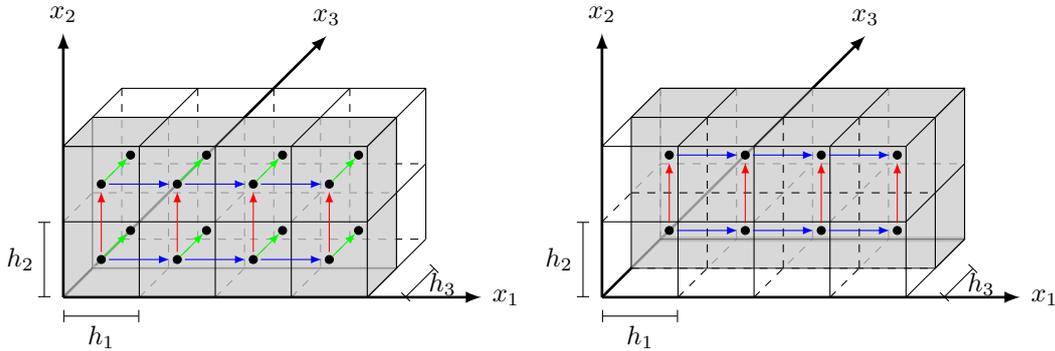


Abbildung 4.2: Darstellung der in 3D zu berechnenden Gradienten am Beispiel des Bildgebietes $m_1 = 4, m_2 = 2, m_3 = 2$. Blaue Pfeile: Gradienten in x_1 -Richtung, rote Pfeile: Gradienten in x_2 -Richtung, grüne Pfeile: Gradienten in x_3 -Richtung.

Damit dies alles gewährleistet ist und die Form einer Fallunterscheidungsmatrix, wie sie für den zweidimensionalen Fall in [32] eingeführt wurde, beibehalten wird, wird in dieser Arbeit ein Hilfsvektor eingeführt. Dieser Hilfsvektor wird über

$$p = [0, m_1 * m_2 - 1 : -1 : m_1 * m_2 - m_1 * m_2 - m_1 + 1]$$

beschrieben und in den Fällen der Gradientenmatrix verwendet. Wie die Matrix, die das Analogon zur Gradientenmatrix (3.6) (siehe Kapitel 3.1.1) im Zweidimensionalen darstellt, im dreidimensionalen Fall im Einzelnen aufgebaut werden kann, ist auf der nächsten Seite dargestellt.

$$\begin{aligned}
A_k^\top \mathbf{u}_l^h = & \left\{ \begin{array}{l}
((\mathbf{u}_l^h)_{k+1} - (\mathbf{u}_l^h)_k; (\mathbf{u}_l^h)_{k+m_1} - (\mathbf{u}_l^h)_k; (\mathbf{u}_l^h)_{k+(m_1 m_2)} - (\mathbf{u}_l^h)_k), \text{ falls } k \bmod m_1 \neq 0, \text{ und } (k \bmod m_1 m_2) \cap p = \emptyset \\
\text{und } k + (m_1 m_2) \leq N, \\
(\quad 0 \quad ; (\mathbf{u}_l^h)_{k+m_1} - (\mathbf{u}_l^h)_k; (\mathbf{u}_l^h)_{k+(m_1 m_2)} - (\mathbf{u}_l^h)_k), \text{ falls } k \bmod m_1 = 0, \text{ und } (k \bmod m_1 m_2) \cap p = \emptyset, \\
\text{und } k + (m_1 m_2) \leq N, \\
((\mathbf{u}_l^h)_{k+1} - (\mathbf{u}_l^h)_k; \quad 0 \quad ; (\mathbf{u}_l^h)_{k+(m_1 m_2)} - (\mathbf{u}_l^h)_k), \text{ falls } k \bmod m_1 \neq 0, \text{ und } (k \bmod m_1 m_2) \cap p \neq \emptyset, \\
\text{und } k + (m_1 m_2) \leq N, \\
((\mathbf{u}_l^h)_{k+1} - (\mathbf{u}_l^h)_k; (\mathbf{u}_l^h)_{k+m_1} - (\mathbf{u}_l^h)_k; \quad 0 \quad), \text{ falls } k \bmod m_1 \neq 0, \text{ und } (k \bmod m_1 m_2) \cap p = \emptyset, \\
\text{und } k + (m_1 m_2) > N, \\
(\quad 0 \quad ; \quad 0 \quad ; (\mathbf{u}_l^h)_{k+(m_1 m_2)} - (\mathbf{u}_l^h)_k), \text{ falls } k \bmod m_1 = 0, \text{ und } (k \bmod m_1 m_2) \cap p \neq \emptyset, \\
\text{und } k + (m_1 m_2) \leq N, \\
(\quad 0 \quad ; (\mathbf{u}_l^h)_{k+m_1} - (\mathbf{u}_l^h)_k; \quad 0 \quad), \text{ falls } k \bmod m_1 = 0, \text{ und } (k \bmod m_1 m_2) \cap p = \emptyset, \\
\text{und } k + (m_1 m_2) > N, \\
((\mathbf{u}_l^h)_{k+1} - (\mathbf{u}_l^h)_k; \quad 0 \quad ; \quad 0 \quad), \text{ falls } k \bmod m_1 \neq 0, \text{ und } (k \bmod m_1 m_2) \cap p \neq \emptyset, \\
\text{und } k + (m_1 m_2) > N, \\
(\quad 0 \quad ; \quad 0 \quad ; \quad 0 \quad), \text{ falls } k \bmod m_1 = 0, \text{ und } (k \bmod m_1 m_2) \cap p \neq \emptyset, \\
\text{und } k + (m_1 m_2) > N,
\end{array} \right.
\end{aligned} \tag{4.2}$$

mit $p = [0, m_1 * m_2 - 1 : -1 : m_1 * m_2 - m_1 * m_2 - m_1 + 1]$.

4. Erweiterung des Modells auf 3D

Daraus können die Matrizen $A_k, k = 1, 2, 3, \dots, N$ ausgelesen und verknüpft werden. Man erhält

$$A = (A_1, A_2, \dots, A_N) = (A_{1,1}, A_{1,2}, A_{1,3}, \dots, A_{N,1}, A_{N,2}, A_{N,3}) \in \mathbb{R}^{N \times 3N}.$$

Aus diesen werden wiederum die Matrizen A_{x_1}, A_{x_2} und A_{x_3} gewählt:

$$A_{x_1} = (A_{1,1}, A_{2,1}, A_{3,1}, \dots, A_{N,1}) \in \mathbb{R}^{N \times N},$$

$$A_{x_2} = (A_{1,2}, A_{2,2}, A_{3,2}, \dots, A_{N,2}) \in \mathbb{R}^{N \times N},$$

$$A_{x_3} = (A_{1,3}, A_{2,3}, A_{3,3}, \dots, A_{N,3}) \in \mathbb{R}^{N \times N},$$

die die Gradienten in die drei Richtungen enthalten. Über diese Matrizen können die einzelnen Gradienten folgendermaßen beschrieben werden:

$$\nabla^{h,1} \mathbf{u}_1^h = \begin{bmatrix} A_{x_1}^\top \\ A_{x_2}^\top \\ A_{x_3}^\top \end{bmatrix} \mathbf{u}_1^h =: B \mathbf{u}_1^h,$$

$$\nabla^{h,2} \mathbf{u}_2^h = \begin{bmatrix} A_{x_1}^\top \\ A_{x_2}^\top \\ A_{x_3}^\top \end{bmatrix} \mathbf{u}_2^h =: B \mathbf{u}_2^h,$$

$$\nabla^{h,3} \mathbf{u}_3^h = \begin{bmatrix} A_{x_1}^\top \\ A_{x_2}^\top \\ A_{x_3}^\top \end{bmatrix} \mathbf{u}_3^h =: B \mathbf{u}_3^h.$$

Für den gesamten diskreten Gradientenoperator ∇^h gilt

$$\nabla^h \mathbf{U}^h = \begin{bmatrix} \nabla^{h,1} & 0 & 0 \\ 0 & \nabla^{h,2} & 0 \\ 0 & 0 & \nabla^{h,3} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^h \\ \mathbf{u}_2^h \\ \mathbf{u}_3^h \end{bmatrix} = \begin{bmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^h \\ \mathbf{u}_2^h \\ \mathbf{u}_3^h \end{bmatrix} =: \mathbf{A} \mathbf{U}^h.$$

Eine Veranschaulichung der Matrix \mathbf{A} am Beispiel $m_1 = m_2 = m_3 = 2$ ist in Abbildung 4.3 zu sehen. Es sind die von Null verschiedenen Einträge unter Verwendung des MATLAB-Befehls `spy` dargestellt.

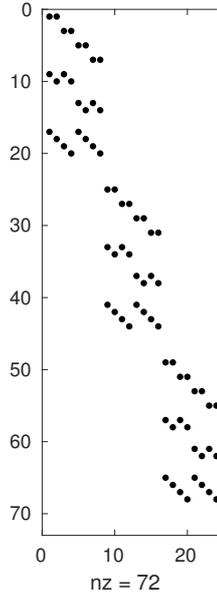


Abbildung 4.3: Darstellung der von Null verschiedenen Einträge der Matrix \mathbf{A} unter Verwendung des MATLAB-Befehls `spy` am Beispiel $m_1 = m_2 = m_3 = 2$.

Der zu diskretisierende Regularisierer erhält im Dreidimensionalen einen Summanden für die dritte Dimension mehr. Damit lautet dieser

$$\mathcal{R}^{NewC}(\mathbf{u}) = \frac{1}{2} \int_{\Omega} \mathcal{B}[\mathbf{u}] \, d\Omega$$

$$\text{mit } \mathcal{B}[\mathbf{u}] = \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right)^2 + \left(\nabla \cdot \frac{\nabla u_2}{|\nabla \mathbf{u}|_{\beta}} \right)^2 + \left(\nabla \cdot \frac{\nabla u_3}{|\nabla \mathbf{u}|_{\beta}} \right)^2.$$

Die diskrete Form des Regularisierers sieht demnach folgendermaßen aus

$$\mathcal{R}^{hNewC}(\mathbf{U}^h) = \frac{1}{2} h_d \sum_{i,j,k} (\mathcal{B}[\mathbf{u}^h])_{i,j,k} = \frac{1}{2} h_d \mathbb{B}^h[\mathbf{U}^h]$$

mit $h_d = h_1 h_2 h_3$.

Die Berechnungen, die im zweidimensionalen Fall durchgeführt wurden, um eine Form zu erhalten, die mit geringem Aufwand implementiert werden kann, können, analog zum Zweidimensionalen, im Dreidimensionalen durchgeführt werden. Dabei muss beachtet werden, dass ein weiterer Summand für die dritte Dimension mitgeführt wird. Dies führt dazu, dass sich die Dimension der Matrix \mathbf{A} in beide Richtungen um eins erhöht. Demnach gilt

$$\begin{aligned} \mathbb{B}^h[\mathbf{U}^h] &= \left| \frac{-B^{\top} B \mathbf{u}_1^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}} \right|^2 + \left| \frac{-B^{\top} B \mathbf{u}_2^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}} \right|^2 + \left| \frac{-B^{\top} B \mathbf{u}_3^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}} \right|^2 \\ &= \frac{(\mathbf{u}_1^h)^{\top} B^{\top} B B^{\top} B \mathbf{u}_1^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}^2} + \frac{(\mathbf{u}_2^h)^{\top} B^{\top} B B^{\top} B \mathbf{u}_2^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}^2} + \frac{(\mathbf{u}_3^h)^{\top} B^{\top} B B^{\top} B \mathbf{u}_3^h}{|\mathbf{A} \mathbf{U}^h|_{\beta}^2} \end{aligned}$$

4. Erweiterung des Modells auf 3D

$$\begin{aligned}
&= \frac{1}{|\mathbf{AU}^h|_\beta^2} \left((\mathbf{u}_1^h)^\top B^\top B B^\top B \mathbf{u}_1^h + (\mathbf{u}_2^h)^\top B^\top B B^\top B \mathbf{u}_2^h + (\mathbf{u}_3^h)^\top B^\top B B^\top B \mathbf{u}_3^h \right) \\
&= \frac{1}{|\mathbf{AU}^h|_\beta^2} \left((\mathbf{u}_1^h)^\top, (\mathbf{u}_2^h)^\top, (\mathbf{u}_3^h)^\top \right) \begin{bmatrix} B^\top B B^\top B & 0 & 0 \\ 0 & B^\top B B^\top B & 0 \\ 0 & 0 & B^\top B B^\top B \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^h \\ \mathbf{u}_2^h \\ \mathbf{u}_3^h \end{bmatrix} \\
&= \frac{1}{|\mathbf{AU}^h|_\beta^2} (\mathbf{U}^h)^\top \begin{bmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{bmatrix}^\top \begin{bmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{bmatrix} \begin{bmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{bmatrix}^\top \begin{bmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{bmatrix} \mathbf{U}^h \\
&= \frac{(\mathbf{U}^h)^\top \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{AU}^h}{|\mathbf{AU}^h|_\beta^2}.
\end{aligned}$$

Damit lässt sich der Regularisierungsterm analog zum zweidimensionalen Fall als

$$\mathcal{R}^{hNewC}(\mathbf{U}^h) = \frac{h_d}{2} \frac{(\mathbf{U}^h)^\top \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{AU}^h}{|\mathbf{AU}^h|_\beta^2}$$

schreiben, wobei sich hier im dreidimensionalen Fall, im Unterschied zum zweidimensionalen Fall, natürlich die Dimensionen von \mathbf{A} und \mathbf{U}^h erhöht haben.

4.2. Erläuterungen zur Implementierung in Matlab

Aufgrund begrenzter Arbeitsspeicherkapazität ist es gerade für die dreidimensionale Implementierung sinnvoll, dies nicht über die Fallunterscheidungen zu implementieren, die in (4.2) angegeben sind. In dieser Arbeit wurde die Implementierung der Diskretisierung etwas effizienter gestaltet, indem das Kroneckerprodukt (siehe Definition 2.18 in Kapitel 2.3.2) und die MATLAB internen Funktionen `spdiags` für dünnbesetzte Diagonalmatrizen und `speye` für die speicherarme Implementierung der Einheitsmatrix verwendet wurden.

Für die Matrix A_{x_1} wird wieder eine Diagonalmatrix \mathbf{Dx} erstellt, die auf ihrer Hauptdiagonalen -1 und auf ihrer rechten Nebendiagonalen 1 stehen hat. Da Neumann-Randbedingungen angenommen werden, wird die letzte Zeile gleich Null gesetzt. Das Kroneckerprodukt wird jetzt aber von einer Einheitsmatrix der Größe $m_2 \times m_3$ mit der Matrix \mathbf{Dx} berechnet, um alle Gradienten in x_1 -Richtung zu erhalten.

```

1: % Matrix  $A_{x_1}$  (Ableitungen nach rechts)
2:  $\mathbf{Dx} = \text{spdiags}(\text{ones}(m(1), 1) * [-1, 1], 0 : 1, m(1), m(1));$ 
3: % Neumann-Randbedingung
4:  $\mathbf{Dx}(\text{end}, :) = 0;$ 
5:  $\mathbf{Ax1} = \text{kron}(\text{speye}(m(2) * m(3)), \mathbf{Dx});$ 

```

Für die Matrix A_{x_2} wird, wie auch im Zweidimensionalen, eine Diagonalmatrix \mathbf{Dy} erstellt, die auf ihrer Hauptdiagonalen -1 und auf ihrer m_1 -ten rechten Nebendiagonalen 1 stehen hat. Die Nullmatrix, die mit `rest` bezeichnet wird, wird an die Matrix

4.2. Erläuterungen zur Implementierung in Matlab

Dy gesetzt und gewährleistet, dass die Neumann-Randbedingungen nach oben eingehalten werden. Hier im dreidimensionalen Fall wird nun das Kroneckerprodukt einer Einheitsmatrix der Größe $m_3 \times m_3$ mit der Matrix Dy berechnet, um alle Gradienten in m_2 -Richtung zu erhalten.

```
1: % Matrix  $A_{x_2}$  (Ableitungen nach oben)
2: Dy = spdiags(ones(m(1)*m(2)-m(1),1)*[-1,1],...
3:     0:m(1):m(1),m(1)*m(2)-m(1),m(1)*m(2));
4: rest = sparse((m(1)*m(2))-size(Dy,1),size(Dy,2));
5: Dy = [Dy;rest];
6: Ax2 = kron(speye(m(3)),Dy);
```

Für die Matrix A_{x_3} wird nun eine Diagonalmatrix Dz erstellt, die auf ihrer Hauptdiagonalen -1 und auf ihrer $(m_1 * m_2)$ -ten rechten Nebendiagonalen 1 stehen hat. Wiederum wird eine Matrix mit Nullen erstellt, die mit **rest** bezeichnet wird und an die Matrix Dz gesetzt wird. Diese Matrix gewährleistet, dass die Neumann-Randbedingungen nach hinten eingehalten werden.

```
1: % Matrix  $A_{x_3}$  (Ableitungen nach hinten)
2: Dz = spdiags(ones(m(1)*m(2)*(m(3)-1),1)*[-1,1],...
3:     0:m(1)*m(2):m(1)*m(2),m(1)*m(2)*(m(3)-1),m(1)*m(2)*m(3));
4: rest = sparse(m(1)*m(2)*m(3)-size(Dz,1),size(Dz,2));
5: Ax3 = [Dz;rest];
```

In diesem Kapitel wurde der Regularisierer auf drei Dimensionen erweitert. Dabei wurde festgestellt, dass vor allem bei der Bestimmung der Gradientenmatrix sehr aufgepasst werden muss, dass die Gradienten für ein zell-zentriertes Gitter, das hier verwendet wurde, korrekt berechnet werden und die Neumann-Randbedingungen eingehalten werden.

5. Ergebnisse

In diesem Kapitel erfolgt die praktische Untersuchung des in dieser Arbeit aufgearbeiteten Regularisierers. Dabei wurde dieser auf Bildregistrierungsprobleme angewandt, die in MATLAB (Version 9.2 (R2017)) implementiert wurden, wobei als Grundlage die FAIR-Toolbox [28] verwendet wurde. Im Zentrum der Analyse steht besonders das Verhalten des Regularisierers auf glatten und nichtglatten Registrierungsproblemen, da nach [32] der Vorteil dieses Regularisierers besonders darin besteht, dass dieser auf glatten wie auf nichtglatten Registrierungsproblemen sehr gute Ergebnisse liefert. Des Weiteren galt vor allem den Deformationsfeldern ein besonderes Interesse, da diese in [32] nicht gezeigt wurden.

Zunächst wurden die Experimente auf zweidimensionalen Bilddaten durchgeführt. Im Anschluss daran wurde gezeigt, dass der Regularisierer mit der in dieser Arbeit entwickelten Erweiterung für 3D auch auf dreidimensionale Bilddaten angewandt werden kann.

5.1. Anwendung auf zweidimensionale Bilddaten

Für die Untersuchung des Regularisierers auf glatte zweidimensionale Bilder wurden die Röntgenbilder der Hand aus [1] verwendet. Diese sind in Abbildung 5.1 zu sehen. Für die Registrierung wurde das linke Bild als Referenzbild und das rechte Bild als Templatebild verwendet.

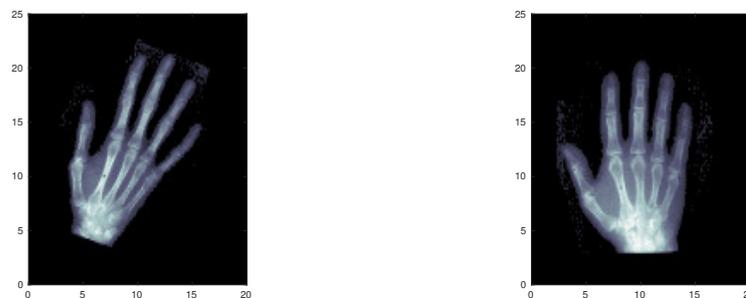


Abbildung 5.1: Verwendete Bilddaten zur Untersuchung des Regularisierers auf ein glattes zweidimensionales Registrierungsbeispiel. Links: Referenzbild. Rechts: Templatebild.

Das Ergebnis der Registrierung des Handdatensatzes unter Verwendung des in dieser Arbeit untersuchten Regularisierers ist in Abbildung 5.2 zu sehen. In dieser Abbildung ist links das Referenzbild zum Vergleich abgebildet, in der Mitte das Templatebild mit dem dazugehörigen deformierten Gitter und rechts das Ergebnis der Registrierung dargestellt. Ziel dieser Abbildung war es, das in [32] gezeigte Registrierungsergebnis nachzubauen, dabei aber zusätzlich das Deformationsfeld zu betrachten. Es ist sehr wichtig, gleichzeitig zum deformierten Templatebild das Deformationsfeld zu

5. Ergebnisse

überprüfen, denn erst damit lässt sich die Qualität des Regularisierers beurteilen. In diesem Registrierungsbeispiel fällt auf, dass das Deformationsfeld nicht diffeomorph (siehe Definition 2.5 in Kapitel 2.1.2) ist, da es Faltungen aufweist. Aufgrund dieser Faltungen werden die Nachbarschaftsbeziehungen der einzelnen Pixel verletzt, was durch Anwendung eines Regularisierers nicht passieren sollte. Für die Durchführung der Registrierungsbeispiele mussten Werte für die Parameter α, β und θ gewählt werden. Der Parameter α gibt die Gewichtung des Regularisierers an. Der Parameter β wird für den hier untersuchten Regularisierer für die Berechnung des Nenners benötigt, um Singularitäten zu vermeiden. Der Parameter θ wird für die hier verwendete Spline-Interpolation benötigt. Dieser kann als Glättungsfaktor angesehen werden, mit dem ein Kompromiss zwischen einer perfekten Anpassung an die Daten und Glättung der Daten gefunden werden kann. Für weitere Ausführungen sei auf [26] verwiesen.

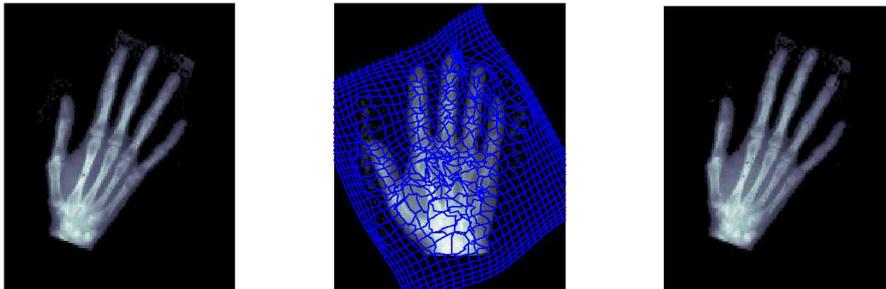


Abbildung 5.2: Ergebnis des glatten zweidimensionalen Registrierungsbeispiels mit $\theta = 1e - 2, \beta = 1e - 6$ und $\alpha = 16000$. Links: Referenzbild zum Vergleich. Mitte: Templatebild und Deformationsfeld. Rechts: Transformiertes Templatebild.

Wie in Abschnitt 3.1.2 erwähnt, wurde für die Implementierung der Registrierung ein Multilevelansatz verwendet. Die Handdaten wurden auf fünf Leveln registriert. Auf dem größten Level hatten die Handdaten die Größe 8×8 . Es folgte die Abstufung $16 \times 16, 32 \times 32, 64 \times 64$ und als feinstes Level 128×128 . Die Entwicklung des Zielfunktional zu jeder Iteration über alle Level des glatten zweidimensionalen Registrierungsbeispiels der Hand ist in Abbildung 5.3 zu sehen. Dabei trennen die gestrichelten vertikalen Linien die einzelnen Level voneinander. Die Quadrate zeigen den Referenzwert auf jedem Level an und die Werte des Zielfunktional sind als Kreuze dargestellt.

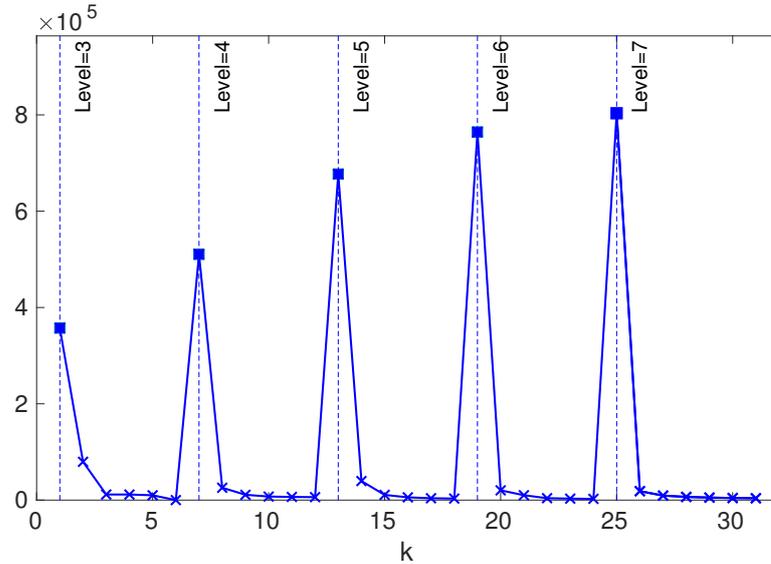


Abbildung 5.3: Entwicklung des Zielfunktional zu jeder Iteration über alle Level des glatten zweidimensionalen Registrierungsbeispiels. Vertikale Linien: Trennen der Level. Quadrate: Referenzwerte. Kreuze: Werte des Zielfunktional.

Beim Suchen eines geeigneten Wertes für den Parameter α wurde folgende Untersuchung gemacht. Die Registrierung wurde mehrmals mit unterschiedlichem α durchgeführt, wobei α Werte zwischen 0 und $1e11$ angenommen hat. Jedes Mal wurde auf dem feinsten Level die Irregularität und die Distanz berechnet und zusammen in einer Graphik geplottet (siehe Abbildung 5.4). Für die Irregularität wurde das Reziproke des Regularisierers verwendet, das heißt der Wert $\frac{1}{\text{Regularisierer}}$, und für die Distanz der Wert $\frac{J_c}{J_{stop}}$. Es lässt sich erkennen, dass die Irregularität (in rot) bei kleinem α sehr groß ist und mit zunehmendem α abnimmt. Im Gegensatz dazu ist die Distanz (in blau) bei kleinem α klein und wird mit zunehmendem α größer. Der Wert für α sollte in dem Bereich liegen, in dem die Irregularität und die Distanz sich am nächsten sind.

5. Ergebnisse

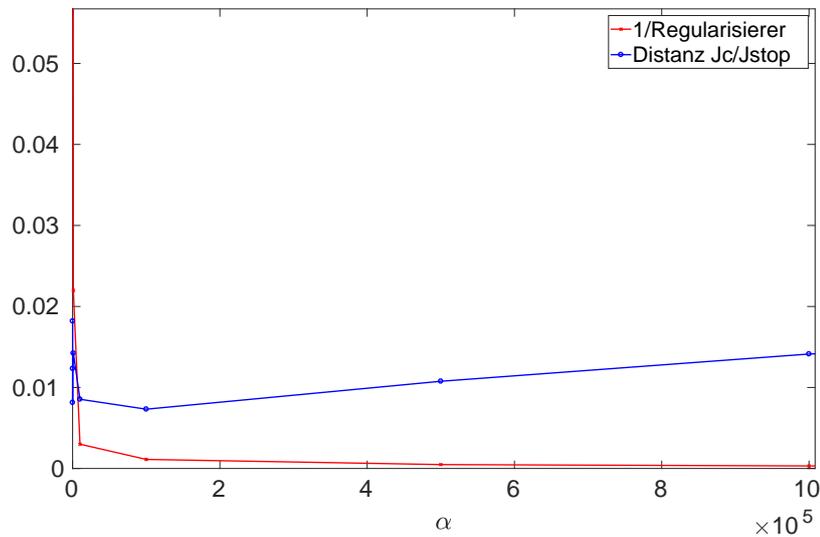


Abbildung 5.4: Darstellung der Irregularität (in rot) und der Distanz (in blau) für Werte zwischen 0 und $1e6$ für α für das glatte zweidimensionale Registrierungsbeispiel.

In Abbildung 5.4 wurden die Irregularität und die Distanz an wenigen Punkten ausgewertet. Dies soll einen groben Überblick darüber geben, wie sich die Irregularität und die Distanz in Abhängigkeit von α verhalten und wo sie sich am nächsten sind. In der nächsten Abbildung 5.5 wird diese Stelle genauer betrachtet, indem die Irregularität und die Distanz für Werte zwischen 0 und 20000 für α geplottet wurden. Hierbei lässt sich erkennen, dass der Wert $\alpha = 16000$, der für dieses Registrierungsbeispiel gewählt wurde, in dem Bereich liegt, in dem sich die Irregularität und die Distanz am nächsten liegen und damit eine geeignete Wahl für α ist.

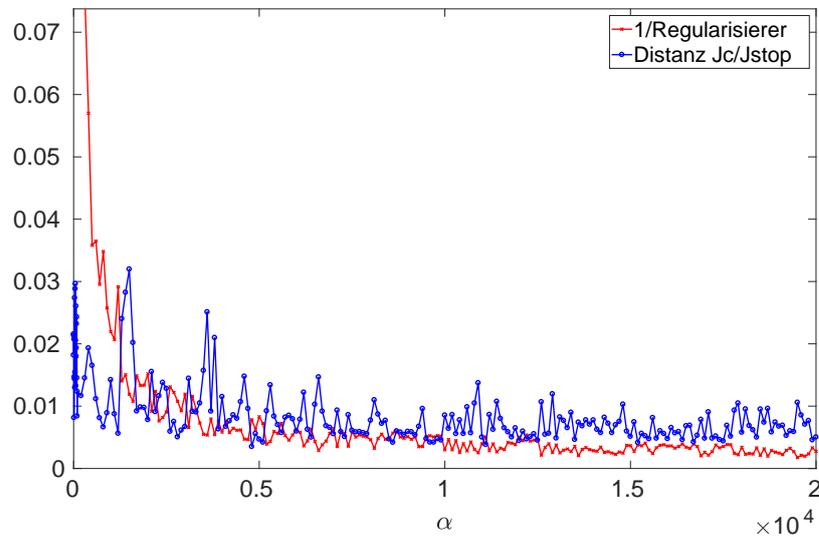


Abbildung 5.5: Darstellung der Irregularität (in rot) und der Distanz (in blau) für Werte zwischen 0 und 20000 für α für das glatte zweidimensionale Registrierungsbeispiel.

Um zu zeigen, was passiert, wenn α zu klein oder zu groß gewählt wird, wurde die folgende Abbildung erstellt. Auf der linken Seite in Abbildung 5.6 wurde $\alpha = 100$ gewählt. Es ist links das Templatebild mit Gitter und rechts das transformierte Templatebild auf dem feinsten Level dargestellt. Auf der rechten Seite wurde $\alpha = 1e11$ gewählt. Links sieht man wieder das Templatebild mit Gitter und rechts das transformierte Templatebild nach der Registrierung. Es ist zu erkennen, dass bei sehr kleinem α ($\alpha = 100$) das Gitter sehr viele Faltungen aufweist und im transformierten Templatebild Löcher auftreten. Bei sehr großem α ($\alpha = 1e11$) hingegen ist das Gitter rigide und es wird im Grunde nur eine Rotation durchgeführt.

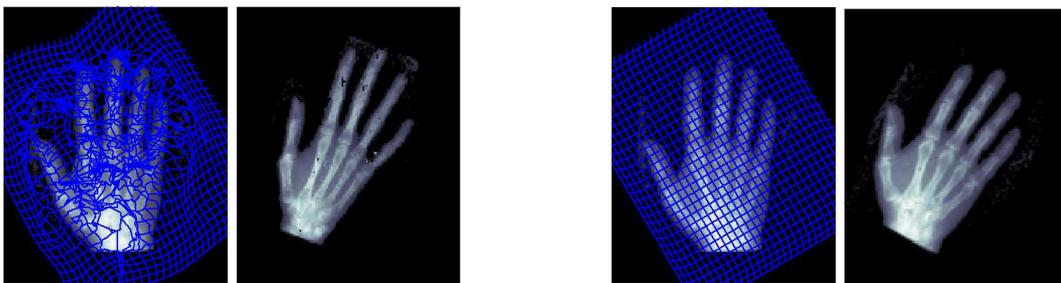


Abbildung 5.6: Ergebnis des glatten zweidimensionalen Registrierungsbeispiels mit $\theta = 1e - 2, \beta = 1e - 6$ bei zu kleinem und zu großem α . Links: Darstellung, Templatebild mit deformiertem Gitter und transformiertes Templatebild bei $\alpha = 100$. Rechts: Darstellung, Templatebild mit deformiertem Gitter und transformiertes Templatebild bei $\alpha = 1e11$.

5. Ergebnisse

Für die Untersuchung des Regularisierers auf ein nichtglattes Registrierungsbeispiel wurde in dieser Arbeit ein akademisches nichtglattes Beispiel mithilfe von MATLAB erstellt. Dieses ist in Abbildung 5.7 zu sehen. Es zeigt zwei nicht-achsen-ausgerichtete Rechtecke auf schwarzem Hintergrund, wobei das eine Rechteck weiß und das andere grau ist. Im Referenzbild sind die beiden Rechtecke gegeneinander verschoben, im Templatebild liegen sie exakt nebeneinander.

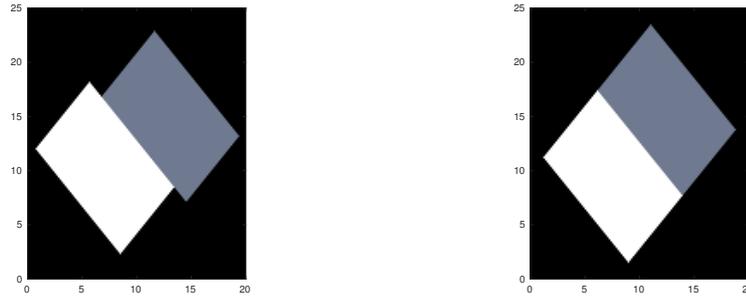


Abbildung 5.7: Verwendete Bilddaten zur Untersuchung des Regularisierers auf ein nichtglattes zweidimensionales Registrierungsbeispiel. Links: Referenzbild. Rechts: Templatebild.

In Abbildung 5.8 ist das Ergebnis der Registrierung des nichtglatten Registrierungsbeispiels unter Verwendung des in dieser Arbeit untersuchten Regularisierers dargestellt. Es ist links in der Abbildung das Referenzbild zum Vergleich abgebildet, in der Mitte das Templatebild mit dazugehörigem deformiertem Gitter und rechts das Ergebnis der Registrierung. Auch hier ist bei genauer Betrachtung des Deformationsgitters zu erkennen, dass dieses nicht diffeomorph ist, da es Faltungen an den Stellen im Bild aufweist, an denen die größte Bewegung im Bild stattfindet. Diese befinden sich auf der Linie, an der die beiden Rechtecke aneinander gleiten und an dieser Linie an den Stellen, an denen sich die Rechtecke gerade noch berühren.

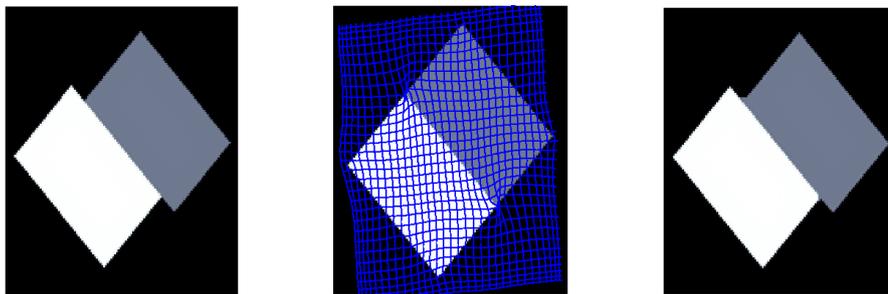


Abbildung 5.8: Ergebnis des nichtglatten zweidimensionalen Registrierungsbeispiels mit $\theta = 1e - 2$, $\beta = 5$ und $\alpha = 4e5$. Links: Referenzbild zum Vergleich. Mitte: Templatebild und Deformationsfeld. Rechts: Transformiertes Templatebild.

5.1. Anwendung auf zweidimensionale Bilddaten

Auch dieses Registrierungsbeispiel wurde über einen Multilevelansatz mit fünf Leveln implementiert. Das größte Level hatte die Größe 8×8 , danach folgte die Abstufung 16×16 , 32×32 , 64×64 und als feinstes Level 128×128 . In Abbildung 5.9 ist die Entwicklung des Zielfunktional zu jeder Iteration über alle Level des nichtglatten zweidimensionalen Registrierungsbeispiels dargestellt. Die gestrichelten vertikalen Linien trennen wiederum die einzelnen Level voneinander, die Quadrate zeigen den Referenzwert auf jedem Level an und die Werte des Zielfunktional sind als Kreuze dargestellt.

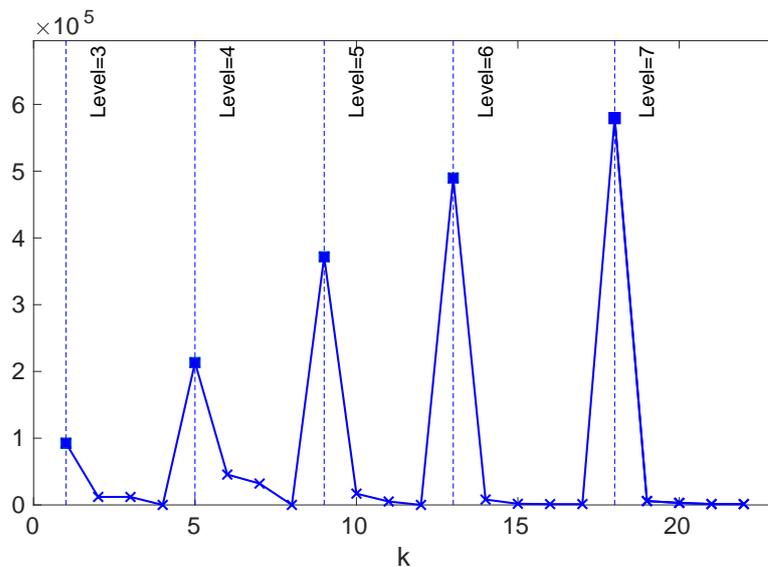


Abbildung 5.9: Entwicklung des Zielfunktional zu jeder Iteration über alle Level des nichtglatten zweidimensionalen Registrierungsbeispiels. Vertikale Linien: Trennen der Level. Quadrate: Referenzwerte. Kreuze: Werte des Zielfunktional.

Eine geeignete Wahl des Parameters α wurde wieder mithilfe der Irregularität und der Distanz gefunden. Dabei hat sich herausgestellt, dass für α geeignete Werte zwischen $1e6$ und $2e6$ liegen, was in Abbildung 5.10 zu sehen ist. Damit ist der Wert $\alpha = 4e5$, der für das nichtglatte Registrierungsbeispiel gewählt worden ist, in dem Bereich, in dem sich die Irregularität und die Distanz am nächsten sind und damit eine geeignete Wahl für α .

5. Ergebnisse

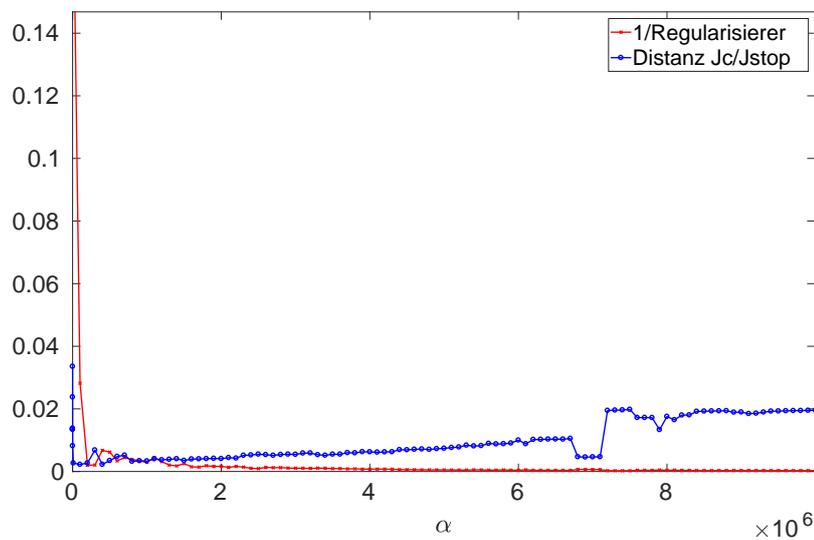


Abbildung 5.10: Darstellung der Irregularität (in rot) und der Distanz (in blau) für Werte zwischen 0 und $1e7$ für α für das nichtglatte zweidimensionale Registrierungsbeispiel.

Auch für das nichtglatte Registrierungsbeispiel wurde in Abbildung 5.11 gezeigt, was passiert, wenn α zu groß oder zu klein gewählt wird. Auf der linken Seite wurde $\alpha = 0,1$ gewählt und auf der rechten Seite $\alpha = 1e11$. Man sieht jeweils links das Templatebild mit Deformationsgitter und rechts das transformierte Templatebild nach der Registrierung. Es ist zu erkennen, dass das Deformationsgitter bei kleinem α sehr viele Faltungen enthält und bei sehr großem α rigide wird.

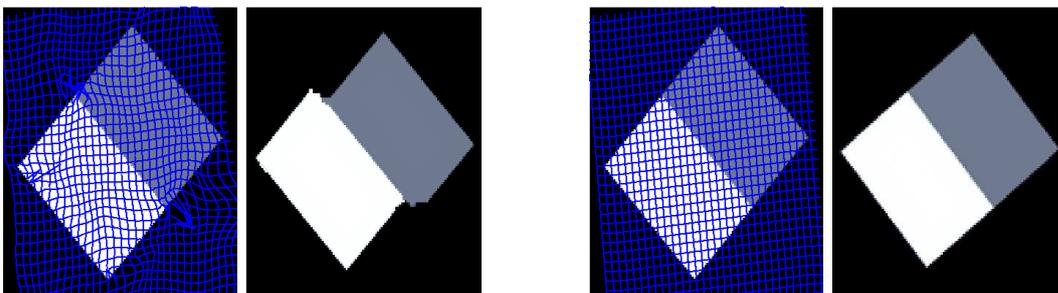


Abbildung 5.11: Ergebnis des nichtglatten zweidimensionalen Registrierungsbeispiels mit $\theta = 1e - 2, \beta = 5$ bei zu kleinem und zu großem α . Links: Darstellung, Templatebild mit deformiertem Gitter und transformiertes Templatebild bei $\alpha = 0,1$. Rechts: Darstellung, Templatebild mit deformiertem Gitter und transformiertes Templatebild bei $\alpha = 1e11$.

5.2. Anwendung auf dreidimensionale Bilddaten

Im Dreidimensionalen wurde der Regularisierer ebenfalls auf ein glattes und ein nicht-glatte Registrierungsbeispiel angewandt. Für das nichtglatte Registrierungsbeispiel

5.2. Anwendung auf dreidimensionale Bilddaten

wurde in dieser Arbeit ein akademisches Beispiel mithilfe von MATLAB erstellt, das in Abbildung 5.12 dargestellt ist. Es zeigt zwei Quader, wobei im Referenzbild die beiden Quader direkt nebeneinander stehen und im Templatebild die beiden Quader gegeneinander nach hinten und oben verschoben sind.

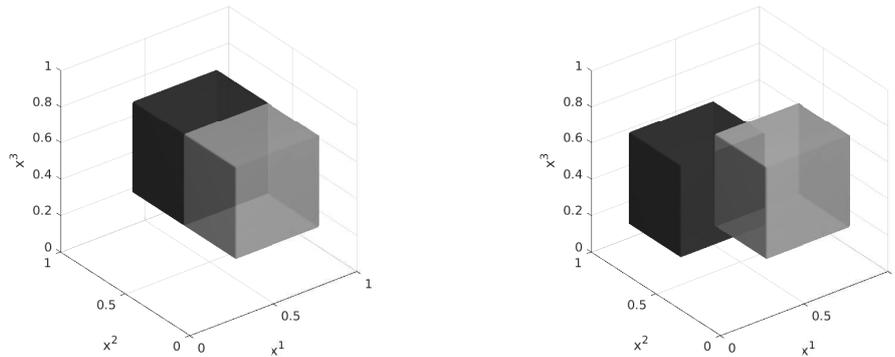


Abbildung 5.12: Verwendete Bilddaten zur Untersuchung des Regularisierers auf ein nicht-glattes dreidimensionales Registrierungsbeispiel. Links: Referenzbild. Rechts: Templatebild.

Das Ergebnis der Registrierung des nichtglatten dreidimensionalen Registrierungsbeispiels unter Verwendung des in dieser Arbeit untersuchten Regularisierers ist in Abbildung 5.13 dargestellt. Diese Darstellung ist so zu verstehen, dass die Quader horizontal von oben nach unten mehrmals geschnitten werden. Diese einzelnen Schnitte sind in der folgenden Abbildung dargestellt, wobei links das Referenzbild zum Vergleich und rechts das transformierte Templatebild zu sehen ist.

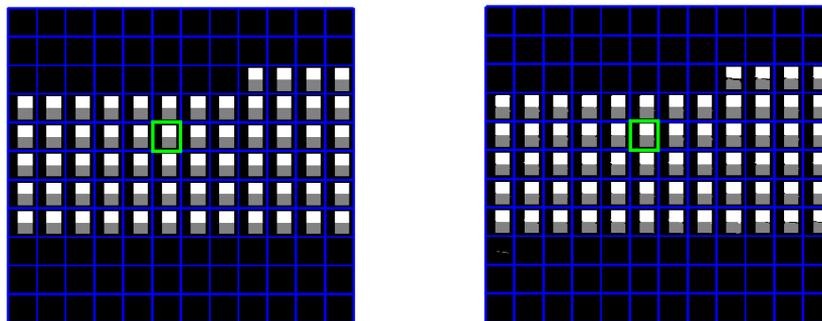


Abbildung 5.13: Ergebnis des nichtglatten dreidimensionalen Registrierungsbeispiels mit $\theta = 1e - 2$, $\beta = 1e - 6$ und $\alpha = 900000$. Links: Referenzbild zum Vergleich. Rechts: Transformiertes Templatebild.

Da in dieser Darstellung keine Details zu sehen sind, sind in der nächsten Abbildung 5.14, die grün eingekästelten Schnitte vergrößert dargestellt.

5. Ergebnisse



Abbildung 5.14: Betrachtung eines Schnitts des nichtglatten dreidimensionalen Registrierungsbeispiels. Links: Referenzbild zum Vergleich. Rechts: Transformiertes Templatebild.

Für die Implementierung des dreidimensionalen nichtglatten Registrierungsbeispiels wurde ein Multilevelansatz mit fünf Leveln verwendet. Das größte Level hatte die Größe 8×8 , danach folgte die Abstufung 16×16 , 32×32 , 64×64 und als feinstes Level 128×128 . Die Entwicklung des Zielfunktional zu jeder Iteration über alle Level des nichtglatten dreidimensionalen Registrierungsbeispiels, ist in Abbildung 5.15 dargestellt. Die gestrichelten vertikalen Linien trennen die einzelnen Level voneinander, die Quadrate zeigen den Referenzwert auf jedem Level an und die Werte des Zielfunktional sind als Kreuze dargestellt.

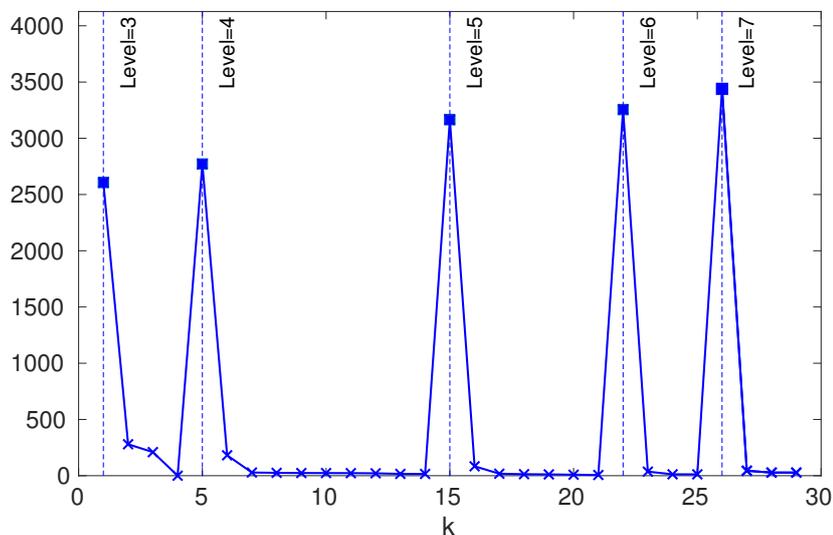


Abbildung 5.15: Entwicklung des Zielfunktional zu jeder Iteration über alle Level des nichtglatten dreidimensionalen Registrierungsbeispiels. Vertikale Linien: Trennen der Level. Quadrate: Referenzwerte. Kreuze: Werte des Zielfunktional.

Für das glatte Registrierungsbeispiel wurde das menschliche Knie untersucht. Ein dreidimensionaler Datensatz dazu ist in der FAIR-Toolbox vorhanden. Dieser ist in Abbildung 5.16 dargestellt, wobei als Referenzbild das durchgestreckte Knie und als Templatebild das angewinkelte Knie verwendet wurde.

5.2. Anwendung auf dreidimensionale Bilddaten

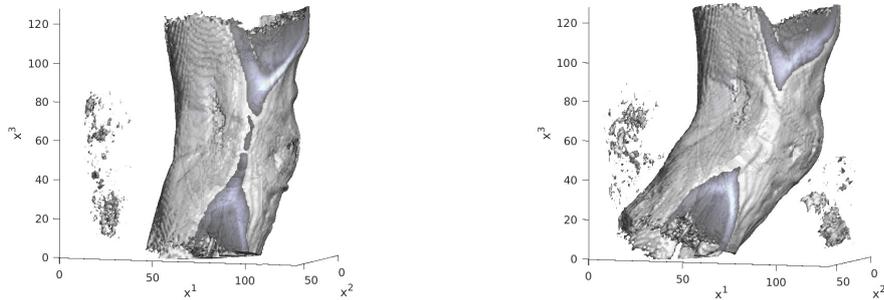


Abbildung 5.16: Verwendete Bilddaten zur Untersuchung des Regularisierers auf ein glattes dreidimensionales Registrierungsbeispiel. Links: Referenzbild. Rechts: Templatebild.

In Abbildung 5.17 ist das Ergebnis der Registrierung des glatten dreidimensionalen Registrierungsbeispiels unter Verwendung des in dieser Arbeit untersuchten Regularisierers dargestellt. In dieser Darstellung sind, wie beim nichtglatten Beispiel, die einzelnen Schnitte dargestellt, die entstehen, wenn man die Daten horizontal von oben nach unten mehrmals schneidet. Es ist wieder links das Referenzbild zum Vergleich und rechts das transformierte Templatebild abgebildet.

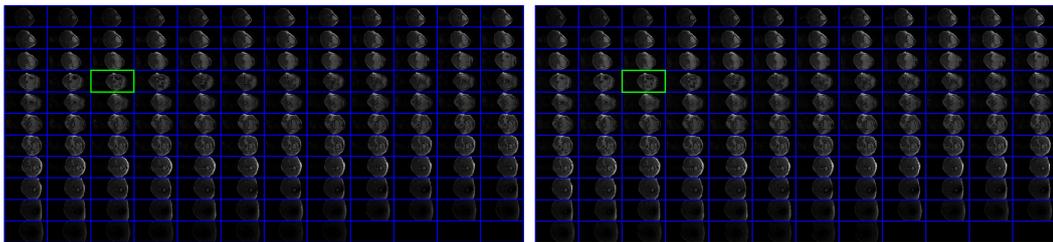


Abbildung 5.17: Ergebnis des glatten dreidimensionalen Registrierungsbeispiels mit $\theta = 1e - 2$, $\beta = 0.01$ und $\alpha = 7000$. Links: Referenzbild zum Vergleich. Rechts: Transformiertes Templatebild.

Für eine detailreichere Ansicht wurden wieder die grün eingekästelten Schnitte vergrößert und in Abbildung 5.18 dargestellt.



Abbildung 5.18: Betrachtung eines Schnitts des glatten dreidimensionalen Registrierungsbeispiels. Links: Referenzbild zum Vergleich. Rechts: Transformiertes Templatebild.

5. Ergebnisse

Für die Implementierung des dreidimensionalen glatten Registrierungsbeispiels wurde ein Multilevelansatz mit vier Leveln angewandt. Das größte Level hatte die Größe $16 \times 8 \times 16$, danach folgte die Abstufung $32 \times 16 \times 32$, $64 \times 32 \times 64$ und als feinstes Level $128 \times 64 \times 128$. Die Entwicklung des Zielfunktionalis zu jeder Iteration über alle Level des glatten dreidimensionalen Registrierungsbeispiels, ist in Abbildung 5.19 dargestellt. Dabei sind die gestrichelten vertikalen Linien wieder die Trennlinien zwischen den einzelnen Leveln, die Quadrate zeigen den Referenzwert auf jedem Level an und die Kreuze die Werte des Zielfunktionalis.

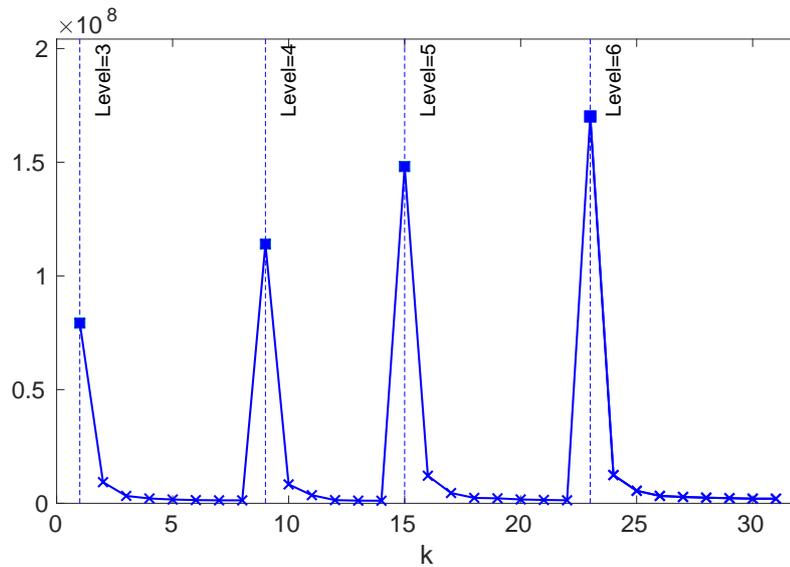


Abbildung 5.19: Entwicklung des Zielfunktionalis zu jeder Iteration über alle Level des glatten dreidimensionalen Registrierungsbeispiels. Vertikale Linien: Trennen der Level. Quadrate: Referenzwerte. Kreuze: Werte des Zielfunktionalis.

6. Diskussion

In diesem Kapitel erfolgt eine Beurteilung der in Kapitel 5 durchgeführten Experimente. Des Weiteren werden Möglichkeiten aufgezeigt, wie die erhaltenen Ergebnisse weiter verbessert werden könnten.

6.1. Auswertung der Ergebnisse

In Kapitel 5 ist zu sehen, dass es mit dem in dieser Arbeit untersuchten Regularisierer möglich ist, ein transformiertes Templatebild zu erhalten, das dem Referenzbild sehr ähnelt. Diese Tatsache wird in [32] verwendet, um die Behauptung, dass der Regularisierer gute Ergebnisse liefert, zu begründen. Die Deformationsfelder werden dort nicht gezeigt. Doch gerade das Deformationsfeld einer Registrierung liefert sehr viele Informationen, die auf die Qualität der erhaltenen Lösung schließen lassen, da dieses aus der Lösung \mathbf{u} , die über das Registrierungsproblem

$$\min_{\mathbf{u} \in \mathbb{R}^d} \{ \mathcal{J}_\alpha[\mathbf{u}] = \mathcal{D}(\mathbf{u}) + \alpha \mathcal{R}(\mathbf{u}) \}$$

erhalten wird, gebildet wird.

Es ist in dieser Arbeit nicht gelungen, Parameterkombinationen für die durchgeführten Registrierungsbeispiele zu finden, mit denen nach der Registrierung die Differenz zwischen Referenzbild und transformiertem Templatebild möglichst klein ist und gleichzeitig das Deformationsgitter keine Faltungen aufweist. Einerseits kann natürlich nicht ausgeschlossen werden, dass Implementierungsfehler im MATLAB-Code, der im Rahmen dieser Arbeit erstellt worden ist, vorliegen. Andererseits weist die Tatsache, dass das zweidimensionale glatte Handregistrierungsbeispiel in dieser Arbeit nachgebaut werden konnte und das transformierte Templatebild (siehe Abbildung 5.2) sehr ähnlich zu dem in [32] ist, darauf hin, dass die Implementierung nicht grundsätzlich falsch sein sollte.

In dieser Arbeit ist nun noch ein Schritt weitergegangen worden und es wurde versucht, eine Möglichkeit zu finden, die Faltungen im Deformationsfeld zu verhindern. Zu diesem Zweck sollte die Publikation [35] angewandt werden, indem das Registrierungsproblem zu einem restringierten Optimierungsproblem erweitert wird. Leider hat sich herausgestellt, dass in dieser Publikation Unstimmigkeiten enthalten sind, die zu Dimensionskonflikten führten. Die erste Herausforderung war daher, die genaue Problemstelle zu finden und zu berichtigen. Dennoch blieb eine weitere Stelle unklar, sodass im Rahmen dieser Arbeit eine praktische Umsetzung dieses Ansatzes nicht möglich war. Es soll im Folgenden dennoch dieser Ansatz kurz umrissen und die Problemstellen aufgezeigt werden.

Bei dem restringierten Optimierungsproblem wird eine Bedingung an die Determinante der Jakobi-Matrix der Transformation $\varphi(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ gestellt, die größer als

6. Diskussion

Null sein soll. Das restringierte Optimierungsproblem lautet demnach

$$\begin{aligned} \min_{\mathbf{u} \in \mathbb{R}^n} \{ \mathcal{J}_\alpha[\mathbf{u}] = \mathcal{D}[R, T; \mathbf{u}] + \alpha \mathcal{R}[\mathbf{u}] \} \\ \text{sodass } \mathcal{F}(\mathbf{u}) > 0, \end{aligned}$$

wobei für

$$\begin{aligned} \mathcal{F}(\mathbf{u}) &= \det(J(\varphi(\mathbf{u}))) = \begin{vmatrix} 1 + (u_1)_{x_1} & (u_1)_{x_2} \\ (u_2)_{x_1} & 1 + (u_2)_{x_2} \end{vmatrix} \\ &= (1 + (u_1)_{x_1})(1 + (u_2)_{x_2}) - (u_1)_{x_2}(u_2)_{x_1} \end{aligned}$$

gilt. Nach der Diskretisierung des restringierten Optimierungsproblems wird dieses in [35] über das Augmented-Lagrangian Verfahren gelöst. Dabei hat sich in [35] ein Fehler eingeschlichen, wodurch es dort zu Dimensionskonflikten kommt. Das zu lösende Optimierungsproblem lautet

$$\psi(\mathbf{U}^h, \lambda, \sigma) = \mathcal{J}_\alpha(\mathbf{U}^h) + \frac{1}{2\sigma} \sum_{i=1}^N ([\min\{0, \sigma F_i^h(\mathbf{U}^h) - \lambda_i\}]^2 - \lambda_i^2),$$

wobei F^h das diskretisierte Funktional der Bedingung im Optimierungsproblem ist. Die Funktion ψ wird linearisiert, indem es durch ein quadratisches $\hat{\psi}$ ersetzt wird, das über eine Taylorentwicklung erhalten wird:

$$\psi(\mathbf{U}^{h(k)} + \delta_{\mathbf{U}^h}) \approx \hat{\psi}(\mathbf{U}^{h(k)} + \delta_{\mathbf{U}^h}) = \psi(\mathbf{U}^{h(k)}) + d\psi(\mathbf{U}^{h(k)})\delta_{\mathbf{U}^h} + \frac{1}{2}\delta_{\mathbf{U}^h}^\top \mathbf{H} \delta_{\mathbf{U}^h},$$

wobei $d\psi(\mathbf{U}^{h(k)})$ die Jakobimatrix und \mathbf{H} die Hessematrix von ψ an $\mathbf{U}^{h(k)}$ ist. Die Berechnung der Jakobimatrix kann im Anhang A.3 nachgelesen werden. Als Ergebnis erhält man für die Jakobimatrix

$$d\psi(\mathbf{U}^{h(k)}) = d\mathcal{J}_\alpha(\mathbf{U}^{h(k)}) + \underbrace{\begin{pmatrix} \partial_1 F_1^h(\mathbf{U}^h) & \dots & \partial_1 F_N^h(\mathbf{U}^h) \\ \vdots & \ddots & \vdots \\ \partial_{2N} F_1^h(\mathbf{U}^h) & \dots & \partial_{2N} F_N^h(\mathbf{U}^h) \end{pmatrix}}_{\in \mathbb{R}^{2N \times N}} \cdot \underbrace{\begin{pmatrix} \sigma F_1^h(\mathbf{U}^h) - \lambda_1 \\ \sigma F_2^h(\mathbf{U}^h) - \lambda_2 \\ \vdots \\ \sigma F_N^h(\mathbf{U}^h) - \lambda_N \end{pmatrix}}_{\in \mathbb{R}^{N \times 1}}.$$

Das heißt, es handelt sich hier um eine Matrix-Vektor-Multiplikation und nicht um eine punktweise Multiplikation, wie es in [35] angegeben ist. Damit ergibt sich für die Hessematrix

$$\begin{aligned} \mathbf{H} &= d^2 \mathcal{J}_\alpha(\mathbf{U}^{h(k)}) \\ &+ \begin{pmatrix} \partial_1 F_1^h(\mathbf{U}^h) & \dots & \partial_1 F_N^h(\mathbf{U}^h) \\ \vdots & \ddots & \vdots \\ \partial_{2N} F_1^h(\mathbf{U}^h) & \dots & \partial_{2N} F_N^h(\mathbf{U}^h) \end{pmatrix} \cdot \begin{pmatrix} \partial_1 F_1^h(\mathbf{U}^h) & \dots & \partial_1 F_N^h(\mathbf{U}^h) \\ \vdots & \ddots & \vdots \\ \partial_{2N} F_1^h(\mathbf{U}^h) & \dots & \partial_{2N} F_N^h(\mathbf{U}^h) \end{pmatrix}^\top \end{aligned}$$

und zu lösen ist das Gleichungssystem

$$\mathbf{H}\delta_{\mathcal{U}^h} = -d\psi(\mathbf{U}^{h^{(k)}}).$$

Bei der Berechnung der ersten Variation des Funktionals $\mathcal{F}(\mathbf{u})$ nach [35] tritt das zweite Problem auf. Es kann nachgerechnet werden, dass die erste Variation des kontinuierlichen Funktionals $\mathcal{F}(\mathbf{u})$ bezüglich des Deformationsfeldes \mathbf{u} die folgende Form hat:

$$d\mathcal{F}(\mathbf{u}) = ((u_2)_{x_1x_2}) - (u_2)_{x_2x_1}, (u_1)_{x_2x_1} - (u_1)_{x_1x_2})^\top.$$

Doch an dieser Stelle muss irgendwo ein Fehler stecken, denn bei zweimal stetigen Funktionen lässt sich der Satz von Schwarz anwenden, sodass die hier formulierte erste Variation des Funktionals $\mathcal{F}(\mathbf{u})$ verschwindet und somit bei der Implementierung der gesamte restringierte Ansatz keinen Einfluss auf das Registrierungsproblem hat. Wo genau an dieser Stelle der Fehler liegt, konnte im Rahmen dieser Arbeit nicht mehr herausgefunden werden.

6.2. Ausblick

In weiteren Arbeiten könnte zuallererst versucht werden, das gerade beschriebene Problem des Verschwindens der ersten Variation des Funktionals $\mathcal{F}(\mathbf{u})$ beim Ansatz des restringierten Optimierungsproblem zu lösen. Dann könnte dieser Ansatz nach [35] und den hier gemachten Ergänzungen komplett implementiert und auf den in dieser Arbeit untersuchten Regularisierer angewandt werden. Es wäre dann zu untersuchen, wie sich dieser Ansatz einerseits auf die Deformationsfelder von Registrierungsbeispielen auswirkt und ob sich durch diesen Ansatz tatsächlich diffeomorphe Deformationsfelder ergeben. Andererseits darf auch das transformierte Templatebild nicht außer Acht gelassen werden und es muss untersucht werden, wie sich dieses bei unterschiedlicher Parameterwahl verhält. Erhält man dann nicht nur sehr gute transformierte Templatebilder, die den Referenzbildern sehr ähneln, sondern zusätzlich ein Deformationsfeld ohne Faltungen, so kann man davon sprechen, dass ein sehr guter Regularisierer gefunden wurde, der die gewünschten Eigenschaften hat und auf glatten und nichtglatten Registrierungsbeispielen sehr gute Ergebnisse liefert. Zusätzlich sollte die Erweiterung auf drei Dimensionen keine großen Mühen bereiten, da die größte Arbeit hier bereits erledigt wurde.

Eine andere Herangehensweise wäre, sich intensiv mit der Publikation [5] zu beschäftigen. Dieses widmet sich ebenfalls der Bestrafung der Determinante der Jakobimatrix der Transformation. Jedoch wird dort die Diskretisierung über eine Triangulierung vorgenommen. Das heißt, die gesamte Diskretisierung, die sich durch diese Arbeit zieht, müsste geändert werden. Aus diesem Grund ist in dieser Arbeit auf diesen Ansatz nicht weiter eingegangen worden.

Zudem könnte ein anderer Ansatz zum Lösen des Registrierungsproblem gewählt

6. Diskussion

werden. In dieser Arbeit wurde der Ansatz gewählt, bei dem das Optimierungsproblem zuerst diskretisiert und dann optimiert wird. Allerdings wäre es mit den hier aufgestellten Euler-Lagrange-Gleichungen nun auch möglich, den Ansatz zu wählen, bei dem zuerst optimiert und dann diskretisiert wird. Ein mögliches Herangehen an dieses Problem, kann zum Beispiel in [6] oder [7] nachgelesen werden.

A. Anhang

A.1. Zusammenhang Divergenz-Formel und mittlere Krümmung

In diesem Abschnitt soll gezeigt werden, dass die Formel der mittleren Krümmung

$$\kappa(u_l) = \frac{(\beta + u_{l_y}^2)u_{l_{xx}} - 2u_{l_x}u_{l_y}u_{l_{xy}} + (\beta + u_{l_x}^2)u_{l_{yy}}}{(\beta + u_{l_x}^2 + u_{l_y}^2)^{3/2}}$$

und die Divergenz Formel

$$\kappa(u_l) = \nabla \cdot \frac{\nabla u_l}{|\nabla u_l|_\beta}$$

äquivalent sind.

Beweis.

$$\begin{aligned} \kappa(u_l) &= \nabla \cdot \frac{\nabla u_l}{|\nabla u_l|_\beta} = \operatorname{div} \left(\frac{u_{l_x}}{|\nabla u_l|_\beta} \right) = \operatorname{div} \left(\frac{\frac{u_{l_x}}{\sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta}}}{\frac{u_{l_y}}{\sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta}}} \right) \\ &= \frac{\partial}{\partial x} \left[\frac{u_{l_x}}{\sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta}} \right] + \frac{\partial}{\partial y} \left[\frac{u_{l_y}}{\sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta}} \right] \\ &= \frac{u_{l_{xx}} \sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta} - u_{l_x} \frac{u_{l_y} u_{l_{yx}} + u_{l_x} u_{l_{xx}}}{\sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta}}}{u_{l_x}^2 + u_{l_y}^2 + \beta} + \frac{u_{l_{yy}} \sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta} - u_{l_y} \frac{u_{l_x} u_{l_{xy}} + u_{l_y} u_{l_{yy}}}{\sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta}}}{u_{l_x}^2 + u_{l_y}^2 + \beta} \\ &= \frac{u_{l_{xx}} \sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta}}{u_{l_x}^2 + u_{l_y}^2 + \beta} - \frac{u_{l_x} u_{l_y} u_{l_{yx}} + u_{l_x} u_{l_x} u_{l_{xx}}}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} \\ &\quad + \frac{u_{l_{yy}} \sqrt{u_{l_x}^2 + u_{l_y}^2 + \beta}}{u_{l_x}^2 + u_{l_y}^2 + \beta} - \frac{u_{l_y} u_{l_x} u_{l_{xy}} + u_{l_y} u_{l_y} u_{l_{yy}}}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} \\ &= \frac{u_{l_{xx}} (u_{l_x}^2 + u_{l_y}^2 + \beta) - u_{l_x} u_{l_y} u_{l_{yx}} - u_{l_x} u_{l_x} u_{l_{xx}}}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} \\ &\quad + \frac{u_{l_{yy}} (u_{l_x}^2 + u_{l_y}^2 + \beta) - u_{l_y} u_{l_x} u_{l_{xy}} - u_{l_y} u_{l_y} u_{l_{yy}}}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} \\ &= \frac{u_{l_{xx}} u_{l_x}^2 + u_{l_{xx}} u_{l_y}^2 + u_{l_{xx}} \beta - u_{l_x} u_{l_y} u_{l_{yx}} - u_{l_{xx}} u_{l_x}^2}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} \\ &\quad + \frac{u_{l_{yy}} u_{l_x}^2 + u_{l_{yy}} u_{l_y}^2 + u_{l_{yy}} \beta - u_{l_y} u_{l_x} u_{l_{xy}} - u_{l_{yy}} u_{l_y}^2}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} \\ &= \frac{u_{l_{xx}} (u_{l_y}^2 + \beta) - u_{l_x} u_{l_y} u_{l_{yx}}}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} + \frac{u_{l_{yy}} (u_{l_x}^2 + \beta) - u_{l_y} u_{l_x} u_{l_{xy}}}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} \\ &= \frac{u_{l_{xx}} (u_{l_y}^2 + \beta) + u_{l_{yy}} (u_{l_x}^2 + \beta) - u_{l_x} u_{l_y} u_{l_{yx}} - u_{l_y} u_{l_x} u_{l_{xy}}}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} \\ &= \frac{u_{l_{xx}} (u_{l_y}^2 + \beta) + u_{l_{yy}} (u_{l_x}^2 + \beta) - 2u_{l_x} u_{l_y} u_{l_{xy}}}{(u_{l_x}^2 + u_{l_y}^2 + \beta)^{3/2}} \end{aligned}$$

A. Anhang

$$= \frac{(\beta + u_y^2)u_{l_{xx}} - 2u_{l_x}u_{l_y}u_{l_{xy}} + (\beta + u_{l_x}^2)u_{l_{yy}}}{(\beta + u_{l_x}^2 + u_{l_y}^2)^{3/2}}$$

□

A.2. Bestimmung der Euler-Lagrange-Gleichung

Aufstellen der Euler-Lagrange-Gleichungen des neuen Regularisierungsterms:

$$\begin{aligned} \mathcal{R}^{NewC}(\mathbf{u}) &= \sum_{l=1}^2 \int_{\Omega} \Phi(\kappa(u_l)) \, d\Omega \\ &= \int_{\Omega} \frac{1}{2} \kappa(u_1)^2 \, d\Omega + \int_{\Omega} \frac{1}{2} \kappa(u_2)^2 \, d\Omega \\ &= \int_{\Omega} \frac{1}{2} \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right) \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right) \, d\Omega \\ &\quad + \int_{\Omega} \frac{1}{2} \left(\nabla \cdot \frac{\nabla u_2}{|\nabla \mathbf{u}|_{\beta}} \right) \left(\nabla \cdot \frac{\nabla u_2}{|\nabla \mathbf{u}|_{\beta}} \right) \, d\Omega \end{aligned}$$

$$\begin{aligned} d_{\varphi} \mathcal{R}^{NewC}(\mathbf{u}) &= \lim_{h \rightarrow 0} \frac{\mathcal{R}^{NewC}(\mathbf{u} + h\varphi) - \mathcal{R}^{NewC}(\mathbf{u})}{h} \\ &= \lim_{h \rightarrow 0} \frac{1}{2h} \int_{\Omega} \left(\nabla \cdot \frac{\nabla(u_1 + h\varphi_1)}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right)^2 + \left(\nabla \cdot \frac{\nabla(u_2 + h\varphi_2)}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right)^2 \\ &\quad - \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right)^2 - \left(\nabla \cdot \frac{\nabla u_2}{|\nabla \mathbf{u}|_{\beta}} \right)^2 \, d\Omega \end{aligned}$$

Für die weitere Berechnung wird nur der Fall für $l = 1$ betrachtet. Der Fall für $l = 2$ läuft analog.

$$\lim_{h \rightarrow 0} \frac{1}{2h} \int_{\Omega} \left(\nabla \cdot \frac{\nabla(u_1 + h\varphi_1)}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right)^2 - \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right)^2 \, d\Omega$$

Die Divergenz ist ein linearer Operator, somit gilt die Additivität:

$$\begin{aligned} &= \lim_{h \rightarrow 0} \frac{1}{2h} \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} + \nabla \cdot \frac{h\varphi_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right)^2 - \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right)^2 \, d\Omega \\ &= \lim_{h \rightarrow 0} \frac{1}{2h} \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right)^2 + 2 \left(\nabla \cdot \frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right) \left(\nabla \cdot \frac{\nabla \varphi_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right) \\ &\quad + \left(\nabla \cdot \frac{\nabla \varphi_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right)^2 - \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right)^2 \, d\Omega \\ &= \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right) \left(\nabla \cdot \frac{\nabla \varphi_1}{|\nabla \mathbf{u}|_{\beta}} \right) \, d\Omega \tag{A.1} \\ &\quad + \lim_{h \rightarrow 0} \frac{1}{2h} \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right)^2 - \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right)^2 \, d\Omega. \end{aligned}$$

Im Folgenden wird nur der zweite Teil von (A.1) betrachtet. Der erste wird später wieder hinzugenommen.

$$\lim_{h \rightarrow 0} \frac{1}{2h} \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right)^2 - \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right)^2 \, d\Omega \tag{A.2}$$

A.2. Bestimmung der Euler-Lagrange-Gleichung

$$\begin{aligned}
&= \lim_{h \rightarrow 0} \frac{1}{2h} \int_{\Omega} \left[\nabla \cdot \frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} - \nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right] \cdot \left[\nabla \cdot \frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} + \nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right] d\Omega \\
&= \lim_{h \rightarrow 0} \frac{1}{2h} \int_{\Omega} \left[\nabla \cdot \left(\frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} - \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right) \right] \cdot \left[\nabla \cdot \frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} + \nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right] d\Omega \\
&= \lim_{h \rightarrow 0} \frac{1}{2h} \int_{\Omega} \left[\nabla \cdot \left(\frac{\nabla u_1 |\nabla \mathbf{u}|_{\beta} - \nabla u_1 |\nabla(\mathbf{u} + h\varphi)|_{\beta}}{|\nabla(\mathbf{u} + h\varphi)|_{\beta} |\nabla \mathbf{u}|_{\beta}} \right) \right] \cdot \left[\nabla \cdot \left(\frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} + \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right) \right] d\Omega \\
&= \lim_{h \rightarrow 0} -\frac{1}{2h} \int_{\Omega} \left[\nabla \cdot \left(\nabla u_1 \frac{|\mathbf{u} + h\varphi|_{\beta} - |\nabla \mathbf{u}|_{\beta}}{|\nabla \mathbf{u}|_{\beta} |\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right) \right] \cdot \left[\nabla \cdot \left(\frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi)|_{\beta}} + \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right) \right] d\Omega.
\end{aligned} \tag{A.3}$$

Im Folgenden wird nur der erste Teil im Integral von Gleichung (A.3) betrachtet. Der zweite wird später wieder hinzugenommen.

$$\begin{aligned}
&\lim_{h \rightarrow 0} \left[\nabla \cdot \left(\nabla u_1 \frac{|\mathbf{u} + h\varphi|_{\beta} - |\nabla \mathbf{u}|_{\beta}}{|\nabla \mathbf{u}|_{\beta} |\nabla(\mathbf{u} + h\varphi)|_{\beta}} \right) \right] \\
&= \nabla \cdot \left(\nabla u_1 \lim_{h \rightarrow 0} \frac{|\nabla(\mathbf{u} + h\varphi)|_{\beta} - |\nabla \mathbf{u}|_{\beta}}{h(|\nabla \mathbf{u}|_{\beta} |\nabla(\mathbf{u} + h\varphi)|_{\beta})} \right) \\
&= \nabla \cdot \left(\nabla u_1 \lim_{h \rightarrow 0} \frac{|\nabla(\mathbf{u} + h\varphi)|_{\beta}^2 - |\nabla \mathbf{u}|_{\beta}^2}{h(|\nabla \mathbf{u}|_{\beta} |\nabla(\mathbf{u} + h\varphi)|_{\beta} (|\nabla(\mathbf{u} + h\varphi)|_{\beta} + |\nabla \mathbf{u}|_{\beta}))} \right) \\
&= \nabla \cdot \left(\nabla u_1 \lim_{h \rightarrow 0} \frac{|\nabla(\mathbf{u} + h\varphi)|_{\beta}^2 - |\nabla \mathbf{u}|_{\beta}^2}{h(|\nabla \mathbf{u}|_{\beta} |\nabla(\mathbf{u} + h\varphi)|_{\beta} |\nabla(\mathbf{u} + h\varphi)|_{\beta} + |\nabla \mathbf{u}|_{\beta} |\nabla(\mathbf{u} + h\varphi)|_{\beta} |\nabla \mathbf{u}|_{\beta})} \right) \\
&= \nabla \cdot \left(\nabla u_1 \frac{1}{|\nabla \mathbf{u}|_{\beta}^3 + |\nabla \mathbf{u}|_{\beta}^3} \lim_{h \rightarrow 0} \frac{|\nabla(\mathbf{u} + h\varphi)|_{\beta}^2 - |\nabla \mathbf{u}|_{\beta}^2}{h} \right) \\
&= \nabla \cdot \left(\nabla u_1 \frac{1}{2|\nabla \mathbf{u}|_{\beta}^3} \lim_{h \rightarrow 0} \frac{|\nabla(\mathbf{u} + h\varphi)|_{\beta}^2 - |\nabla \mathbf{u}|_{\beta}^2}{h} \right)
\end{aligned} \tag{A.4}$$

Im Folgenden wird nur der zweite Teil in der Divergenz von Gleichung (A.4) betrachtet. Der erste wird später wieder hinzugenommen. An dieser Stelle sei angemerkt, dass hier [31, Anhang A.3 Korollar A.3.3] anwendbar ist und deswegen die Grenzwert- und Integrationsprozesse vertauscht werden dürfen.

$$\begin{aligned}
&\lim_{h \rightarrow 0} \frac{|\nabla(\mathbf{u} + h\varphi)|_{\beta}^2 - |\nabla \mathbf{u}|_{\beta}^2}{h} \\
&= \lim_{h \rightarrow 0} \frac{(\sqrt{\beta + (u_{1x} + h\varphi_{1x})^2 + (u_{1y} + h\varphi_{1y})^2 + (u_{2x} + h\varphi_{2x})^2 + (u_{2y} + h\varphi_{2y})^2})^2}{h} \\
&\quad - \frac{(\sqrt{\beta + (u_{1x})^2 + (u_{1y})^2 + (u_{2x})^2 + (u_{2y})^2})^2}{h} \\
&= \lim_{h \rightarrow 0} \frac{(u_{1x})^2 + 2u_{1x}h\varphi_{1x} + (h\varphi_{1x})^2 + (u_{1y})^2 + 2u_{1y}h\varphi_{1y} + (h\varphi_{1y})^2 + (u_{2x})^2 + 2u_{2x}h\varphi_{2x} + (h\varphi_{2x})^2 + (u_{2y})^2 + 2u_{2y}h\varphi_{2y} + (h\varphi_{2y})^2 - (u_{1x})^2 - (u_{1y})^2 - (u_{2x})^2 - (u_{2y})^2}{h} \\
&\quad + \frac{2u_{1x}h\varphi_{1x} + 2u_{1y}h\varphi_{1y} + 2u_{2x}h\varphi_{2x} + 2u_{2y}h\varphi_{2y}}{h} \\
&= 2(\nabla u_1 \nabla \varphi_1 + \nabla u_2 \nabla \varphi_2).
\end{aligned} \tag{A.5}$$

A. Anhang

Zurück zu (A.2):

$$\begin{aligned} & \lim_{h \rightarrow 0} \frac{1}{2} \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_1}{|\nabla(\mathbf{u} + h\varphi|_{\beta})} \right)^2 - \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right)^2 d\Omega \\ &= \lim_{h \rightarrow 0} -\frac{1}{2} \int_{\Omega} \left[\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u} + h\varphi|_{\beta}} - \nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right] \cdot \left[\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u} + h\varphi|_{\beta}} + \nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right] d\Omega. \end{aligned}$$

Ersetzen des ersten Faktors durch (A.4) und (A.5) und bilden des Grenzwertes über den zweiten Faktor liefert:

$$\begin{aligned} &= -\frac{1}{2} \int_{\Omega} \nabla \cdot \left(\frac{\nabla u_1}{2|\nabla \mathbf{u}|_{\beta}^3} (2(\nabla u_1 \nabla \varphi_1 + \nabla u_2 \nabla \varphi_2)) \right) 2 \nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} d\Omega \\ &= - \int_{\Omega} \nabla \cdot \left(\frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} (\nabla u_1 \nabla \varphi_1 + \nabla u_2 \nabla \varphi_2) \right) \kappa(u_1) d\Omega. \end{aligned}$$

Nach partieller Integration folgt:

$$\begin{aligned} &= - \int_{\partial\Omega} \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} (\nabla u_1 \nabla \varphi_1 + \nabla u_2 \nabla \varphi_2) \vec{n} dS \\ &\quad + \int_{\Omega} \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} (\nabla u_1 \nabla \varphi_1 + \nabla u_2 \nabla \varphi_2) \nabla \kappa(u_1) d\Omega \\ &= \int_{\Omega} \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} (\nabla u_1 \nabla \varphi_1 + \nabla u_2 \nabla \varphi_2) \nabla \kappa(u_1) d\Omega + 0 \\ &= \int_{\Omega} \nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \nabla \varphi_1 + \nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \nabla \varphi_2 d\Omega \\ &= \int_{\Omega} \nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \nabla \varphi_1 d\Omega + \int_{\Omega} \nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \nabla \varphi_2 d\Omega. \end{aligned}$$

Partielle Integration des ersten Integrals:

$$\begin{aligned} & \int_{\Omega} \nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \nabla \varphi_1 d\Omega \\ &= - \int_{\Omega} \varphi_1 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \right) d\Omega + \int_{\partial\Omega} \varphi_1 \nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \vec{n} dS \\ &= - \int_{\Omega} \varphi_1 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \right) d\Omega + 0. \end{aligned}$$

Partielle Integration des zweiten Integrals:

$$\begin{aligned} & \int_{\Omega} \nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \nabla \varphi_2 d\Omega \\ &= - \int_{\Omega} \varphi_2 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \right) d\Omega + \int_{\partial\Omega} \varphi_2 \nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \vec{n} dS \\ &= - \int_{\Omega} \varphi_2 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \right) d\Omega + 0. \end{aligned}$$

A.3. Bestimmung der Jakobimatrix des restringierten Optimierungsproblems

Beide Integrale zusammen liefern demnach:

$$\begin{aligned}
 & - \int_{\Omega} \varphi_1 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \right) d\Omega - \int_{\Omega} \varphi_2 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \right) d\Omega \\
 & = - \int_{\Omega} \varphi_1 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \right) + \varphi_2 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \right) d\Omega. \tag{A.6}
 \end{aligned}$$

Betrachtet man den ersten Summanden von (A.1)

$$\begin{aligned}
 & \int_{\Omega} \left(\nabla \cdot \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}} \right) \left(\nabla \cdot \frac{\nabla \varphi_1}{|\nabla \mathbf{u}|_{\beta}} \right) d\Omega \\
 & = \int_{\Omega} \kappa(u_1) \nabla \cdot \frac{\nabla \varphi_1}{|\nabla \mathbf{u}|_{\beta}} d\Omega,
 \end{aligned}$$

dann liefert die partielle Integration folgendes:

$$\begin{aligned}
 & \int_{\partial\Omega} \kappa(u_1) \frac{\nabla \varphi_1}{|\nabla \mathbf{u}|_{\beta}} \vec{n} dS - \int_{\Omega} \frac{\nabla \varphi_1}{|\nabla \mathbf{u}|_{\beta}} \nabla \kappa(u_1) d\Omega \\
 & = - \int_{\Omega} \frac{\nabla \varphi_1}{|\nabla \mathbf{u}|_{\beta}} \nabla \kappa(u_1) d\Omega.
 \end{aligned}$$

Eine nochmalige partielle Integration liefert:

$$\begin{aligned}
 & \int_{\Omega} \varphi_1 \nabla \cdot \frac{\nabla \kappa(u_1)}{|\nabla \mathbf{u}|_{\beta}} d\Omega - \int_{\partial\Omega} \varphi_1 \frac{\nabla \kappa(u_1)}{|\nabla \mathbf{u}|_{\beta}} \vec{n} dS \\
 & = \int_{\Omega} \varphi_1 \nabla \cdot \frac{\nabla \kappa(u_1)}{|\nabla \mathbf{u}|_{\beta}} d\Omega. \tag{A.7}
 \end{aligned}$$

Mit den Gleichungen (A.6) und (A.7) erhält man die Euler-Lagrange-Gleichungen

$$\begin{cases} f_1(\mathbf{u}) + \alpha \varphi_1 \nabla \cdot \frac{\nabla \kappa(u_1)}{|\nabla \mathbf{u}|_{\beta}} - \varphi_1 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \right) + \varphi_2 \nabla \cdot \left(\nabla \kappa(u_1) \frac{\nabla u_1}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \right) = 0 \\ f_2(\mathbf{u}) + \alpha \varphi_2 \nabla \cdot \frac{\nabla \kappa(u_2)}{|\nabla \mathbf{u}|_{\beta}} - \varphi_2 \nabla \cdot \left(\nabla \kappa(u_2) \frac{\nabla u_2}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_2 \right) + \varphi_1 \nabla \cdot \left(\nabla \kappa(u_2) \frac{\nabla u_2}{|\nabla \mathbf{u}|_{\beta}^3} \nabla u_1 \right) = 0 \end{cases}$$

mit den Randbedingungen $\langle \nabla u_l, \vec{n} \rangle_{\mathbb{R}^2} = \langle \nabla \kappa(u_l), \vec{n} \rangle_{\mathbb{R}^2} = \langle \varphi_l, \vec{n} \rangle_{\mathbb{R}^2} = 0$ für $l = 1, 2$.

A.3. Bestimmung der Jakobimatrix des restringierten Optimierungsproblems

Das zu lösende Optimierungsproblem lautet

$$\psi(\mathbf{U}^h, \lambda, \sigma) = \mathcal{J}_{\alpha}(\mathbf{U}^h) + \underbrace{\frac{1}{2\sigma} \sum_{i=1}^N ([\min\{0, \sigma F_i^h(\mathbf{U}^h) - \lambda_i\}]^2 - \lambda_i^2)}_{=: G(\mathbf{U}^h)},$$

wobei F^h das diskretisierte Funktional der Bedingung im Optimierungsproblem ist. Es soll nun die Jakobimatrix von $G(\mathbf{U}^h)$ bestimmt werden. Es sei $j \in 1, \dots, 2N$, dann gilt

$$d_j G(\mathbf{U}^h) = d_j \left[\frac{1}{2\sigma} \sum_{i=1}^N ([\min\{0, \sigma F_i^h(\mathbf{U}^h) - \lambda_i\}]^2 - \lambda_i^2) \right]$$

$$\begin{aligned}
&= \frac{1}{2\sigma} d_j \left[\sum_{i=1}^N ([\min\{0, \sigma F_i^h(\mathbf{U}^h) - \lambda_i\}]^2 - \lambda_i^2) \right] \\
&= \frac{1}{2\sigma} d_j \left[\sum_{i=1}^N \frac{\sigma F_i^h(\mathbf{U}^h - \lambda_i)^2 - (\sigma F_i^h(\mathbf{U}^h) - \lambda_i) \cdot |\sigma F_i^h(\mathbf{U}^h) - \lambda_i|}{2} \right] \\
&= \begin{cases} 0, & \text{falls } (\sigma F_i^h(\mathbf{U}^h) - \lambda_i) \geq 0, \\ \frac{1}{2\sigma} d_j \left[\sum_{i=1}^N \frac{\sigma F_i^h(\mathbf{U}^h - \lambda_i)^2 + (\sigma F_i^h(\mathbf{U}^h) - \lambda_i)^2}{2} \right], & \text{falls } (\sigma F_i^h(\mathbf{U}^h) - \lambda_i) < 0. \end{cases}
\end{aligned}$$

Betrachtet man nun den Fall $(\sigma F_i^h(\mathbf{U}^h) - \lambda_i) < 0$ folgt:

$$\begin{aligned}
&\frac{1}{2\sigma} d_j \left[\sum_{i=1}^N \frac{2 \cdot (\sigma F_i^h(\mathbf{U}^h) - \lambda_i)^2}{2} \right] = \frac{1}{2\sigma} d_j \left[(\sigma F_i^h(\mathbf{U}^h) - \lambda_i)^2 \right] \\
&= \sum_{i=1}^N (\sigma F_i^h(\mathbf{U}^h) - \lambda_i) d_j F_i^h(\mathbf{U}^h) = \sum_{i=1}^N d_j F_i^h(\mathbf{U}^h) \cdot (\sigma F_i^h(\mathbf{U}^h) - \lambda_i).
\end{aligned}$$

Das heißt, es gilt für $\nabla G(\mathbf{U}^h)$

$$\begin{aligned}
\nabla G(\mathbf{U}^h) &= \begin{pmatrix} \sum_{i=1}^N \partial_1 F_i^h(\mathbf{U}^h) \cdot (\sigma F_i^h(\mathbf{U}^h) - \lambda_i) \\ \vdots \\ \sum_{i=1}^N \partial_{2N} F_i^h(\mathbf{U}^h) \cdot (\sigma F_i^h(\mathbf{U}^h) - \lambda_i) \end{pmatrix} \\
\iff \nabla G(\mathbf{U}^h) &= \begin{pmatrix} \partial_1 F_1^h(\mathbf{U}^h) & \dots & \partial_1 F_N^h(\mathbf{U}^h) \\ \vdots & \ddots & \vdots \\ \partial_{2N} F_1^h(\mathbf{U}^h) & \dots & \partial_{2N} F_N^h(\mathbf{U}^h) \end{pmatrix} \cdot \begin{pmatrix} \sigma F_1^h(\mathbf{U}^h) - \lambda_1 \\ \sigma F_2^h(\mathbf{U}^h) - \lambda_2 \\ \vdots \\ \sigma F_N^h(\mathbf{U}^h) - \lambda_N \end{pmatrix}.
\end{aligned}$$

Literaturverzeichnis

- [1] AMIT, YALI: *A nonlinear variational problem for image matching*. SIAM Journal on Scientific Computing, 15(1):207–224, 1994.
- [2] BOSCH, SIEGFRIED: *Lineare Algebra*. Springer-Spektrum-Verlag, Berlin Heidelberg, 5. Auflage, 2014.
- [3] BROIT, CHAIM: *Optimal registration of deformed images*. PhD Thesis University of Pennsylvania, 1981.
- [4] BROWN, LISA GOTTESFELD: *A survey of image registration techniques*. ACM computing surveys (CSUR), 24(4):325–376, 1992.
- [5] BURGER, MARTIN, JAN MODERSITZKI und LARS RUTHOTTO: *A hyperelastic regularization energy for image registration*. SIAM Journal on Scientific Computing, 35(1):B132–B148, 2013.
- [6] CHUMCHOB, NOPPADOL und KE CHEN: *Improved variational image registration model and a fast algorithm for its numerical approximation*. Numerical Methods for Partial Differential Equations, 28(6):1966–1995, 2012.
- [7] CHUMCHOB, NOPPADOL, KE CHEN und CARLOS BRITO-LOEZA: *A fourth-order variational image registration model and its fast multigrid algorithm*. Multiscale Modeling & Simulation, 9(1):89–128, 2011.
- [8] DO CARMO, MANFREDO: *Differential Geometry of Curves and Surfaces*. Prentice-Hall Inc., Engelwood Cliffs, New Jersey, 1976.
- [9] DO CARMO, MANFREDO: *Differentialgeometrie von Kurven und Flächen*. Vieweg Verlag, Braunschweig, 3. Auflage, 1993.
- [10] FISCHER, BERND und JAN MODERSITZKI: *Fast curvature based registration of MR-mammography images*. In: *Bildverarbeitung für die Medizin 2002*, Seiten 139–142. Springer-Verlag, 2002.
- [11] FISCHER, BERND und JAN MODERSITZKI: *Curvature based image registration*. Journal of Mathematical Imaging and Vision, 18(1):81–85, 2003.
- [12] FISCHER, BERND und JAN MODERSITZKI: *A unified approach to fast image registration and a new curvature based registration technique*. Linear Algebra and its applications, 380:107–124, 2004.
- [13] FORSTER, OTTO: *Analysis 1: Differential- und Integralrechnung einer Veränderlichen*. Springer-Spektrum-Verlag, Wiesbaden, 12. Auflage, 2016.

LITERATURVERZEICHNIS

- [14] FORSTER, OTTO: *Analysis 2: Differentialrechnung im \mathbb{R}^n , gewöhnliche Differentialgleichungen*. Springer-Spektrum-Verlag, Wiesbaden, 11. Auflage, 2017.
- [15] FORSTER, OTTO: *Analysis 3: Maß- und Integrationstheorie, Integralsätze im \mathbb{R}^n und Anwendungen*. Springer-Spektrum-Verlag, Wiesbaden, 8. Auflage, 2017.
- [16] GHOLIPOUR, ALI, NASSER KEHTARNAVAZ, RICHARD BRIGGS, MICHAEL DEVOUS und KAUNDINYA GOPINATH: *Brain functional localization: a survey of image registration techniques*. IEEE transactions on medical imaging, 26(4):427–451, 2007.
- [17] GIAQUINTA, MARIANO und STEFAN HILDEBRANDT: *Calculus of Variations I*. Springer-Verlag, Berlin Heidelberg, 2004.
- [18] HADAMARD, JACQUES: *Sur les problèmes aux dérivées partielles et leur signification physique*. Bulletin of Princeton University, 13:49–52, 1902.
- [19] HANDELS, HEINZ: *Medizinische Bildverarbeitung: Bildanalyse, Mustererkennung und Visualisierung für die computergestützte ärztliche Diagnostik und Therapie*. Springer-Verlag, 2009.
- [20] HÖMKE, LARS, CLAUDIA FROHN-SCHAUF, STEFAN HENN und KRISTIAN WITSCH: *Total variation based image registration*. In: *Image Processing Based on Partial Differential Equations*, Seiten 343–361. Springer-Verlag, 2007.
- [21] KLOTZEK, BENNO: *Einführung in die Differentialgeometrie*. Verlag Harri Deutsch, Frankfurt am Main, 3. Auflage, 1997.
- [22] KÖNIGSBERGER, KONRAD: *Analysis 2*. Springer-Verlag, Berlin Heidelberg, 5. Auflage, 2004.
- [23] LAU, DIETLINDE: *Algebra und Diskrete Mathematik 1: Grundbegriffe der Mathematik, Algebraische Strukturen 1, Lineare Algebra und Analytische Geometrie, Numerische Algebra und Kombinatorik*. Springer-Verlag, Berlin Heidelberg, 3. Auflage, 2011.
- [24] MAINTZ, JB ANTOINE und MAX A VIERGEVER: *A survey of medical image registration*. Medical image analysis, 2(1):1–36, 1998.
- [25] MEYBERG, KURT und PETER VACHENAUER: *Höhere Mathematik 2*. Springer-Verlag, Berlin Heidelberg, 3. Auflage, 1999.
- [26] MODERSITZKI, JAN: *FAIR: flexible algorithms for image registration*. SIAM, Philadelphia, 2009.
- [27] MODERSITZKI, JAN: *Numerical methods for image registration*. Oxford University, Oxford, 2009.

- [28] MODERSITZKI, JAN, LARS RUTHOTTO und FABIAN GIGENGACK: *FAIR (Toolbox): Flexible Algorithms for Image Registration*. <https://github.com/C4IR/FAIR.m>. [letzter Aufruf: 21.2.2018].
- [29] NOCEDAL, JORGE und STEPHEN WRIGHT: *Numerical optimization*. Springer-Verlag, New York, 2. Auflage, 2006.
- [30] OLIVEIRA, FRANCISCO und JOAO MANUEL TAVARES: *Medical image registration: a review*. Computer methods in biomechanics and biomedical engineering, 17(2):73–93, 2014.
- [31] WERNER, DIRK: *Funktionalanalysis*. Springer-Verlag, Berlin Heidelberg, 7. Auflage, 2011.
- [32] ZHANG, JIN und KE CHEN: *A new curvature-based image registration model and its fast algorithm*. International Journal of Numerical Analysis and Modeling, 13(6):969–985, 2016.
- [33] ZHANG, JIN, KE CHEN, FANG CHEN und BO YU: *An Efficient Numerical Method for Mean Curvature-Based Image Registration Model*. East Asian Journal on Applied Mathematics, 7(1):125–142, 2017.
- [34] ZHANG, JIN, KE CHEN und BO YU: *An Improved Discontinuity-Preserving Image Registration Model and Its Fast Algorithm*. Applied Mathematical Modelling (c), 40(23/24):10740–10759, 2016.
- [35] ZHANG, JIN, KE CHEN und BO YU: *A novel high-order functional based image registration model with inequality constraint*. Computers & Mathematics with Applications, 72(12):2887–2899, 2016.
- [36] ZITOVA, BARBARA und JAN FLUSSER: *Image registration methods: a survey*. Image and vision computing, 21(11):977–1000, 2003.