



UNIVERSITÄT ZU LÜBECK
INSTITUTE OF MATHEMATICS AND
IMAGE COMPUTING

Bachelorarbeit

Bildregistrierung mit Volumenbeschränkung

Alexander Derksen

Arbeit betreut von
Prof. Dr. Bernd Fischer und Dr. Stefan Heldmann
Institute of Mathematics and Image Computing

Abgabe: 28.02.2011



IM FOCUS DAS LEBEN

Eidesstattliche Erklärung

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

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

Ort, Datum

Unterschrift

Inhaltsverzeichnis

Eidesstattliche Erklärung	3
Einleitung und Aufbau	7
1 Bildregistrierung	9
1.1 Distanzmaße	10
1.2 Transformationen	11
1.3 Regularisierer	13
2 Diskretisierung	17
2.1 Bildmodellierung	17
2.2 Gitter	20
2.3 Distanzmaß	22
2.4 Regularisierer	22
3 Volumenbeschränkung	27
3.1 Kontinuierliche Formulierung	27
3.2 Diskretisierung	31
4 Optimierung	35
4.1 Log-Barrier-Ansatz	35
4.2 Gauss-Newton-Verfahren	37
4.3 Ableitungen der Zielfunktion	39
5 Numerische Experimente	43
5.1 Synthetische Daten	43
5.2 Lungenbilder	47
5.3 Fazit	51
Literaturverzeichnis	53

Einleitung und Aufbau

Bildregistrierung zählt heute zu den bedeutenden Feldern in der Bildverarbeitung. Das Registrierungsproblem besteht darin, in gegebenen Bildern korrespondierende Objekte durch geometrische Deformationen zur Überlagerung zu bringen. Das heißt, Bilder sollen möglichst „ähnlich“ werden. Speziell in der medizinischen Bildverarbeitung finden sich viele Möglichkeiten zum sinnvollen Einsatz, so z. B. beim Vergleich von pre- und postoperativen Aufnahmen oder auch beim Zusammenführen von Aufnahmen, die mit unterschiedlichen bildgebenden Verfahren erzeugt wurden. Typische Ansätze formulieren das Registrierungsproblem als Optimierungsaufgabe, bei der eine geeignete Zielfunktion minimiert wird. In dieser Arbeit erweitern wir diesen Ansatz, in dem wir zusätzlich Beschränkung hinsichtlich der Volumenänderung, hervorgerufen durch die Deformation, fordern. Diese Beschränkung der deformierten Volumina ist im Hinblick auf die Plausibilität der berechneten Deformation von Interesse.

Die vorliegende Arbeit gliedert sich in fünf Abschnitte. Das erste Kapitel behandelt die kontinuierliche Formulierung des hier verwendeten Registrierungs-Ansatzes. Im anschließenden Kapitel 2 wird eine Diskretisierung der kontinuierlichen Formulierung erörtert. Ein mit der Materie vertrauter Leser kann sich direkt Kapitel 3 und folgenden zuwenden. In Kapitel 3 wird sowohl die kontinuierliche Formulierung wie auch die Diskretisierung der Volumenbeschränkung behandelt. Im vorletzten Kapitel 4 wird ein Verfahren zur Lösung des diskretisierten Optimierungsproblems vorgestellt. Hier verwenden wir einen einfachen Log-Barrier-Ansatz, bei dem es sich um eine sog. Innere-Punkt-Methode handelt [6]. In Kapitel 5 werden numerische Experimente des vorgestellten Verfahrens betrachtet. Diese sollen die Vor- und Nachteile des Ansatzes herausstellen.

1 Bildregistrierung

Im folgenden Abschnitt wird das Problem der Bildregistrierung mathematisch formuliert. Wir betrachten *Bilder* als stetig differenzierbare Abbildungen

$$B : \mathbb{R}^d \rightarrow \mathbb{R}, \quad B \in C^1(\mathbb{R}^d, \mathbb{R}), \quad d \in \mathbb{N}, \quad (1.1)$$

so dass jedem Ort $\mathbf{x} \in \mathbb{R}^d$ ein Grauwert $B(\mathbf{x})$ zugeordnet wird. Dabei nehmen wir an, dass ein Bild einen beschränkten *Träger* (engl. Support)

$$\text{supp}(B) \subseteq \bar{\Omega} \subset \mathbb{R}^d \quad (1.2)$$

mit $\text{supp}(B) := \overline{\{\mathbf{x} \in \mathbb{R}^d \mid B(\mathbf{x}) \neq 0\}}$ in einem Gebiet Ω hat. Unter einem *Gebiet* verstehen wir dabei eine offene und zusammenhängende Menge. Im Folgenden wird Ω als rechteckig angenommen und als *Bildbereich* (engl. Region of Interest) bezeichnet.

Die *Verrückung* \mathbf{u} ist ein Vektorfeld

$$\mathbf{u} = (u_1, \dots, u_d)^T : \mathbb{R}^d \rightarrow \mathbb{R}^d$$

und beschreibt in welche Richtung ein gegebener Bildpunkt \mathbf{x} durch $\mathbf{u}(\mathbf{x})$ verschoben wird.

Unter einer *Deformation* φ versteht man schließlich die Abbildung

$$\varphi = (\varphi_1, \dots, \varphi_d)^T : \mathbb{R}^d \rightarrow \mathbb{R}^d, \quad \mathbf{x} \mapsto \varphi(\mathbf{x}) := \mathbf{x} + \mathbf{u}(\mathbf{x}) \quad , \quad \mathbf{x} \in \mathbb{R}^d \quad (1.3)$$

und für ein gegebenes Bild B , bezeichnet man $B \circ \varphi$ als *deformiertes Bild*.

Ein *Distanzmaß* ist ein Funktional

$$\mathcal{D} : V \times V \rightarrow \mathbb{R},$$

mit $V = V(\Omega) := \{B : \mathbb{R}^d \rightarrow \mathbb{R} \mid \text{supp}(B) \subseteq \bar{\Omega}, B \in C^1(\mathbb{R}^d, \mathbb{R})\}$, welches zwei Bildern ihren Abstand zuordnet.

Die Grundidee der Bildregistrierung lässt sich nun als Optimierungsaufgabe formulie-

ren. Bei einem gegebenen Referenz-Bild R und einem deformierbaren Template-Bild T , suchen wir eine Deformation φ , so dass R und $T \circ \varphi$ ähnlich sind, d. h. wir suchen eine Lösung des Minimierungsproblems

$$\min_{\varphi} \mathcal{D}(T \circ \varphi, R). \quad (1.4)$$

Der Einfachheit halber, wird von nun an, wenn nicht anders vermerkt, $d = 2$ angenommen. Alle hier vorgestellten Konzepte können analog auf höhere Dimensionen übertragen werden, insbesondere auch auf $d = 3$.

1.1 Distanzmaße

Der wesentliche Baustein eines jeden Registrierungs-Verfahrens ist das verwendete Distanzmaß. In den letzten Jahren wurde eine Vielzahl an Distanzmaßen für die unterschiedlichsten Einsatzgebiete entwickelt. Zwei essentielle Eigenschaften, die alle Maße gemeinsam haben, sind

- $\mathcal{D}(R, T) = \mathcal{D}(T, R)$,
- $\mathcal{D}(R, R) = \min_T \mathcal{D}(R, T)$.

Die naheliegende Annahme, dass solche Distanzmaße auch Metriken sind, ist allerdings im Allgemeinen nicht richtig. Beispielsweise wird man öfter eine Verletzung des Axioms

$$\mathcal{D}(R, T) = 0 \Rightarrow T = R$$

feststellen (siehe Normalized-Gradient-Fields).

Distanzmaße lassen sich grob in zwei Klassen unterteilen.

Monomodale Maße

Die Klasse der monomodalen Maße ist dafür konzipiert, Bilder, die mit gleichen Sensoren aufgenommen wurden, zu vergleichen. Dabei kann bei zwei gegebenen Bildern von einer vergleichbaren Kontrastausprägung und einem vergleichbaren Helligkeitsniveau ausgegangen werden. Diese Maße finden z. B. Anwendung bei zwei Aufnahmen die zwar vom selben Patienten stammen, allerdings zeitlich versetzt entstanden sind. Ein expliziter Anwendungsfall ist der Vergleich von prä- und postoperativen Magnetresonanztomographie-Aufnahmen zur Beurteilung des Erfolgs einer operativen Tumorentfernung.

Zwei der gebräuchlichsten monomodalen Maße sind

Sum-of-squared-differences $\text{SSD}(R, T) = \frac{1}{2} \int_{\Omega} (T(x, y) - R(x, y))^2 dx dy$

Sum-of-absolut-differences $SAD(R, T) = \int_{\Omega} |T(x, y) - R(x, y)| \, dx \, dy.$

Multimodale Maße

Die Klasse der multimodalen Maße erlaubt es, Bilder, die mit unterschiedlichen Sensoren aufgenommen wurden, zu vergleichen. Bei zwei derartigen Aufnahmen kann man keine allgemeinen Annahmen bezüglich vergleichbarer Kontrastausprägung oder vergleichbarem Helligkeitsniveau treffen. In der Medizin finden diese Maße vor allem Anwendung zum Vergleich von Bildern mit funktionellem und morphologischem Inhalt. Ein Beispiel ist die Registrierung einer Positronen-Emissions-Tomographie-Aufnahme mit einer Computertomographie-Aufnahme zur Lokalisierung eines Tumors.

Die mathematische Handhabung ist komplexer als die monomodaler Maße. Da der Schwerpunkt der Arbeit auf Volumenbeschränkung liegt, wird der Einfachheit halber SSD als Distanzmaß gewählt. Selbstverständlich lassen sich die vorgestellten Methoden zur Volumenbeschränkung auch mit einem beliebigen Distanzmaß kombinieren.

1.2 Transformationen

In diesem Abschnitt werden Transformations-Modelle behandelt. Das Transformations-Modell bestimmt die Art der zulässigen Deformationen und muss vor Beginn der Registrierung mit Bedacht gewählt werden. Es werden hier zwei Klassen von Transformationen vorgestellt. Zum einen die Klasse der parametrischen Transformationen, unter der die Auswahl der erlaubten Deformationen stark restringiert ist. Und zum anderen die Klasse der nicht-parametrischen Transformationen, die prinzipiell jede erdenkbare Deformation enthält.

Parametrische Transformationen

Parametrische Transformationen wirken global. Eine Transformation heißt *global*, wenn die Änderung eines Transformationsparameters zur Folge hat, dass das Bild als Ganzes verändert wird. Zwei übliche Transformationsarten sind die rigiden sowie die affinen Deformationen.

Rigide Deformationen Eine Deformation φ heißt *rigide*, falls

$$\varphi(\mathbf{x}) = \mathbf{Q}\mathbf{x} + \mathbf{b}, \quad \mathbf{x}, \mathbf{b} \in \mathbb{R}^d, \mathbf{Q} \in \mathbb{R}^{d \times d}, d \in \mathbb{N}$$

und \mathbf{Q} orthogonal. Durch \mathbf{Q} orthogonal ist sichergestellt, dass rigide Deformationen längen- und winkelerhaltend sind. Aus \mathbf{Q} orthogonal folgt, $\det(\mathbf{Q}^T \mathbf{Q}) = 1$ und mit der

Produktregel folgt, $\det(\mathbf{Q}) = \pm 1$. Falls $\det(\mathbf{Q}) = 1$, entspricht \mathbf{Q} einer Rotation. Falls $\det(\mathbf{Q}) = -1$, entspricht \mathbf{Q} einer Spiegelung an einer Ursprungsgeraden. Um Spiegelungen auszuschließen, wird nun

$$\det(\mathbf{Q}) = 1$$

gefordert. Im hier betrachteten Fall von $d = 2$, ergibt sich die Darstellung

$$\mathbf{Q} = \mathbf{Q}(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}, \quad \alpha \in [0, 2\pi). \quad (1.5)$$

Somit steuert man eine rigide Deformation, für $d = 2$, durch den Rotationsparameter α und die zwei Translationsparameter $(b_x, b_y)^T = \mathbf{b}$. Im Fall $d = 3$ ergeben sich analog 6 Parameter.

Affine Deformationen Eine Deformation φ heißt *affin*, falls

$$\varphi(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}, \quad \mathbf{x}, \mathbf{b} \in \mathbb{R}^d, \mathbf{A} \in \mathbb{R}^{d \times d}$$

mit \mathbf{A} invertierbar. Falls \mathbf{A} nicht invertierbar ist, besitzt \mathbf{A} einen nicht-trivialen Kern. Somit würde für unendlich viele Punkte $\mathbf{x} \in \mathbb{R}^d$ gelten, dass $\varphi(\mathbf{x}) = \mathbf{b}$. Dies würde eine unerwünschte Faltung des Bildes zur Folge haben.

Mit einer affinen Deformation lassen sich beliebige Hintereinanderausführungen von Elementardeformationen realisieren. Elementardeformationen besitzen für $d = 2$ die folgenden Darstellungen

- Translationen: $\varphi_\delta(x, y) = (x + \delta_x, y + \delta_y)^T$, mit zwei Freiheitsgraden $\delta_x, \delta_y \in \mathbb{R}$.
- Skalierungen: $\varphi_\sigma(x, y) = (\sigma_x x, \sigma_y y)^T = \begin{pmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$, mit zwei Freiheitsgraden $\sigma_x, \sigma_y \in \mathbb{R}$. Für σ_x bzw. σ_y gleich -1 , ist φ_σ eine Spiegelung an y bzw. x -Achse.
- Drehungen: $\varphi_\alpha(x, y) = \mathbf{Q} \begin{pmatrix} x \\ y \end{pmatrix}$, mit \mathbf{Q} wie in (1.5).
- Scherungen: $\varphi_s(x, y) = (x + s_y y, y)^T = \begin{pmatrix} 1 & s_y \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$ ist Scherung an y -Achse,
bzw. $\varphi_s(x, y) = (x, y + s_x x)^T = \begin{pmatrix} 1 & 0 \\ s_x & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$ ist Scherung an x -Achse.

Das heißt, 2-dim. affine Deformationen mit $\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$ und $\mathbf{b} = \begin{pmatrix} b_x \\ b_y \end{pmatrix}$,

können durch die 6 Parameter $a_{1,1}, \dots, a_{2,2}, b_x, b_y$ parametrisiert werden. Für $d = 3$ ergeben sich analog 12 Parameter.

Da diese Transformationen global wirken, ist es mit ihnen nicht möglich, lokale Veränderungen zu beschreiben. Doch gerade diese sind in medizinischen Fragestellungen von Interesse. Deswegen liegt der Fokus dieser Arbeit auf den sogenannten nicht-parametrischen Transformationen.

Nicht-parametrische Transformationen

Nicht-parametrische Transformationen können, im Gegensatz zu parametrischen Transformationen, beliebige lokale Deformationen des Bildes beschreiben. In diesem Fall ist das in (1.4) formulierte Problem im Sinne von Hadamard *schlecht gestellt*. Das heißt,

- die Lösung ist im Allgemeinen nicht eindeutig und
- die Lösung hängt nicht stetig von den Eingangsdaten ab.

Dies liegt unter anderem darin begründet, dass man zu jedem Ort $\mathbf{x} \in \mathbb{R}^d$ einen Ort $\varphi(\mathbf{x}) \in \mathbb{R}^d$ sucht, dabei aber lediglich auf skalarwertige Grauwertinformationen zurückgreifen kann [4]. Ein weiterer Grund ist: Ohne weitere Einschränkung ist jede denkbare Deformation gestattet und insbesondere gilt $\#\{\varphi : \mathcal{D}(R, T \circ \varphi) = \min\} > 1$. Um dieses Problem in den Griff zu bekommen, wird (1.4) um einen zusätzlichen Regularisierungsterm erweitert.

1.3 Regularisierer

Regularisierung ist ein Ansatz, bei dem man versucht, ein schlecht gestelltes Problem in ein korrekt gestelltes zu überführen. Dies geschieht durch Hinzunahme von Informationen zur Zielfunktion. Das aus (1.4) bekannte Problem wird somit um einen zusätzlichen Term \mathcal{S} erweitert. Man erhält

$$\min_{\varphi} \mathcal{D}(T \circ \varphi, R) + \alpha \mathcal{S}(\mathbf{u}) \quad , \alpha \in \mathbb{R}^+ \quad (1.6)$$

mit dem *Regularisierer* \mathcal{S} (im Kontext von Registrierung auch *Smoother* genannt) und einem Regularisierungsparameter α . Dabei soll \mathcal{S} nicht-glatte Deformationen φ auf Grundlage der Verrückung \mathbf{u} bestrafen. Eine Deformation φ heißt *glatt*, wenn sie stetig-differenzierbar ist und kleine Ableitungen hat. \mathbf{u} ist als Parameter für \mathcal{S} hinreichend, da aus (1.3) folgt

$$\mathbf{u}(\mathbf{x}) = \varphi(\mathbf{x}) - \mathbf{x} \quad , \mathbf{x} \in \mathbb{R}^d$$

und somit alle Information über φ implizit in \mathbf{u} enthalten ist. Der *Regularisierungsparameter* α aus (1.6) gewichtet den Einfluss des Regularisierers gegenüber dem Distanzmaß und ist somit die zentrale Stellgröße des Optimierungsproblems.

Neben den verschiedensten Distanzmaßen wurden in den letzten Jahren auch zahlreiche Regularisierer vorgeschlagen. Im Folgenden geben wir einen kurzen Überblick über drei der gängigsten Regularisierer. Dafür sei $\mathbf{x} = (x, y)^T \in \mathbb{R}^2$ und $\mathbf{u} = (u, v)^T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$.

Diffusiver Regularisierer Der diffusive Regularisierer fordert Glattheit der Deformation durch Bestrafen von hohen Ableitungen, d. h. der Regularisierungsterm $\mathcal{S}^{\text{diff}}(\mathbf{u})$ mit

$$\begin{aligned} \mathcal{S}^{\text{diff}}(\mathbf{u}) &:= \frac{1}{2} \int_{\Omega} \|\nabla u\|_2^2 + \|\nabla v\|_2^2 \, dx \, dy \\ &= \frac{1}{2} \int_{\Omega} (\partial_x u)^2 + (\partial_y u)^2 + (\partial_x v)^2 + (\partial_y v)^2 \, dx \, dy \end{aligned}$$

ist groß für große Ableitungen von \mathbf{u} und klein für kleine Ableitungen von \mathbf{u} [9].

Elastischer Regularisierer Der elastische Regularisierer ist physikalisch motiviert, d. h. er modelliert ein elastisches Potential. Er gestattet Deformationen, welche das Bild „gummiartig“ verformen. Mit Hilfe der *Divergenz* $\text{div } \mathbf{u} := \partial_x u + \partial_y v$, ist er durch

$$\mathcal{S}^{\text{elastic}}(\mathbf{u}) := \frac{1}{2} \int_{\Omega} \mu (\|\nabla u\|_2^2 + \|\nabla v\|_2^2) + (\lambda + \mu) (\text{div } \mathbf{u})^2 \, dx \, dy$$

gegeben.

Curvature Regularisierer Der curvature Regularisierer setzt sich, im Gegensatz zum diffusiven Regularisierer, aus zweiten Ableitungen von \mathbf{u} zusammen. Er ist gegeben durch

$$\mathcal{S}^{\text{curv}}(\mathbf{u}) := \frac{1}{2} \int_{\Omega} (\Delta u)^2 + (\Delta v)^2 \, dx \, dy,$$

mit Laplace-Operator $\Delta u := \partial_x \partial_x u + \partial_y \partial_y u$ [8].

Der Einfachheit halber betrachten wir im Folgenden den diffusiven Regularisierer.

Das kontinuierliche Optimierungsproblem hat nun, durch die Wahl des SSD-Maßes und

des diffusiven Regularisierers, die Zielfunktion

$$\mathcal{J}(\varphi) = \frac{1}{2} \int_{\Omega} (T \circ \varphi - R)^2 \, d\mathbf{x} + \frac{\alpha}{2} \int_{\Omega} \|\nabla u\|_2^2 + \|\nabla v\|_2^2 \, d\mathbf{x}. \quad (1.7)$$

Damit lautet das Optimierungsproblem

$$\min_{\varphi} \mathcal{J}(\varphi).$$

Da eine analytische Lösung nicht berechenbar ist, wird hier der sog. *Discretize-then-Optimize*-Ansatz [3] verwendet. Dazu wird die Zielfunktion zunächst diskretisiert und dann eine Lösung des diskretisierten Problems mit Optimierungsmethoden berechnet. Im folgenden Kapitel beschäftigen wir uns mit der Diskretisierung des Registrierungs-Problems.

2 Diskretisierung

In diesem Kapitel wird eine Diskretisierung der kontinuierlichen Zielfunktion erörtert.

2.1 Bildmodellierung

Bilder liegen typischerweise nicht als Funktionen, wie in (1.1) und (1.2) gefordert, vor. Sie liegen stattdessen in diskreter Form vor, als Messungen von Sensoren die ein Signal nur an endlich vielen Stellen abtasten. Um dennoch eine kontinuierliche Darstellung der Bilddaten zu erhalten, wird diese durch Interpolation erzeugt. *Interpolation* bedeutet, dass zu gegebenen diskreten Daten eine kontinuierliche Funktion gesucht wird, die diese Daten abbildet. Im eindimensionalen Fall lässt sich das Interpolationsproblem folgendermaßen formulieren:

Gegeben seien m Datenpunkte $x_i = x_0 + (i - 1)h$, $i = 1, \dots, m$, mit konstanter Schrittweite h . Ein gegebener Farbwert $p_i \in \mathbb{R}$ soll am Ort x_i liegen. Damit ist eine *Interpolante* $I : \mathbb{R} \rightarrow \mathbb{R}$ mit der Forderung

$$I(x_i) = p_i \quad , i = 1, \dots, m \quad (2.1)$$

gesucht. Gängige Interpolationsansätze wie Nearest-Neighbour-, Lineare- oder kubische B-Spline-Interpolation, stellen die Interpolante als Linearkombination translaterter Basisfunktionen dar, d. h. I hat die Form

$$I(x) = \sum_{i=1}^m c_i b(x - x_i).$$

Die gängigen Interpolationsansätze mit ihren Vor- und Nachteilen werden in Tabelle 2.1 aufgeführt. In Abbildung 2.1 werden die zugehörigen Basis-Funktionen aufgeführt. Im

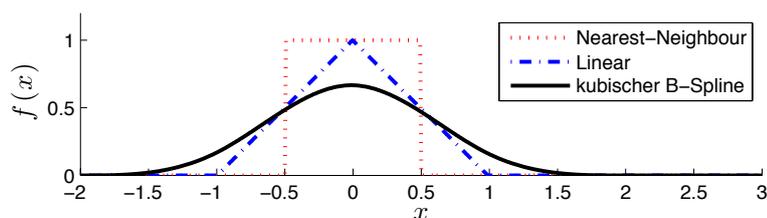


ABBILDUNG 2.1: Basisfunktionen der Spline-Interpolation vom Grad 0,1 und 3.

Weiteren wird kubische B-Spline-Interpolation betrachtet, da wir verlangt haben, dass Bilder stetig differenzierbar sein müssen. Diese Eigenschaft ist für das später verwendete ableitungsbasierte Optimierungsverfahren nötig.

Interpolante	Vorteile	Nachteile
Nearest-Neighbour	<ul style="list-style-type: none"> – erfüllt Min/Max-Prinzip – einfach zu bestimmen 	<ul style="list-style-type: none"> – nur stückweise differenzierbar – nur stückweise stetig
Linear	<ul style="list-style-type: none"> – erfüllt Min/Max-Prinzip – einfach zu bestimmen – stetig 	<ul style="list-style-type: none"> – nur stückweise differenzierbar
kubische B-Spline	<ul style="list-style-type: none"> – zweimal stetig differenzierbar – es ex. schnelle Auswertung – beschränkter Träger 	<ul style="list-style-type: none"> – erfüllt nicht Min/Max-Prinzip
Polynom	<ul style="list-style-type: none"> – eindeutig bestimmt – stetig 	<ul style="list-style-type: none"> – Grad des Pol. = Anz. Datenpkt. – 1 – erfüllt nicht Min/Max-Prinzip – stark oszillierend

TABELLE 2.1: Gängige Interpolanten mit Vor- und Nachteilen.
 Erklärung: Min/Max-Prinzip bedeutet, dass die Interpolante durch das minimale/maximale p_i nach unten/oben beschränkt ist.

kubische B-Spline-Interpolation

Ein B-Spline $I : \mathbb{R} \rightarrow \mathbb{R}$ vom Grad n ist eine auf $(0, m)$ $(n - 1)$ mal stetig differenzierbare Funktion. Für jedes Intervall $(k, k + 1)$ aus $(0, m)$ ist sie gleich einem Polynom vom Grad n . B-Spline-Interpolation mit $n = 3$ hat sich als ein Standard-Verfahren durchgesetzt. Dabei ist ein kubischer B-Spline

$$b^3(x) = \begin{cases} \frac{1}{6}x^3 & , x \in [0, 1[\\ \frac{1}{6} + \frac{1}{2}(x-1) + \frac{1}{2}(x-1)^2 - \frac{1}{2}(x-1)^3 & , x \in [1, 2[\\ \frac{2}{3} - (x-2)^2 + \frac{1}{2}(x-2)^3 & , x \in [2, 3[\\ \frac{1}{6} - \frac{1}{2}(x-3) + \frac{1}{2}(x-3)^2 - \frac{1}{6}(x-3)^3 & , x \in [3, 4[\\ 0 & , x \notin [0, 4[\end{cases}$$

stückweise durch Polynome dritten Grades gegeben. Sei $x_i = x_0 + (i - 1)h$, $i = 1, \dots, m$ ein äquidistantes eindimensionales Gitter mit m Stützstellen und dazugehörigen p_i . Davon ausgehend suchen wir nun eine Interpolante der Form

$$I(x) = \sum_{k=1}^m c_k b_{k,h}^3(x),$$

mit sog. kardinalen B-Splines

$$b_{k,h}^3(x) = b^3\left(\frac{x - x_0}{h} - k + 2\right).$$

Mit Forderung (2.1) erhält man m Gleichungen

$$\begin{aligned} p_i = I(x_i) &= \sum_{k=1}^m c_k b_{k,h}^3(x_i) \quad , i = 1, \dots, m \\ &= \sum_{k=1}^m c_k b^3\left(\frac{x_i - x_0}{h} - k + 2\right) \\ &= \sum_{k=1}^m c_k b^3(i - k + 1) \end{aligned}$$

zur Bestimmung der Koeffizienten c_k .

Dass heißt, die Koeffizienten $\mathbf{c} = (c_1, \dots, c_m)^T$ zu den gegebenen Bildpunkten $\mathbf{p} = (p_1, \dots, p_m)^T$, bestimmen sich durch

$$\mathbf{B}\mathbf{c} = \mathbf{p} \quad , \mathbf{c}, \mathbf{p} \in \mathbb{R}^m,$$

mit

$$\mathbf{B} = \frac{1}{6} \begin{pmatrix} 4 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & \ddots & \ddots & & \\ & & \ddots & 4 & 1 & \\ & & & 1 & 4 & \end{pmatrix} \quad , \mathbf{B} \in \mathbb{R}^{m \times m}.$$

Die eindimensionale Interpolante lässt sich einfach durch sog. Tensor-Produkt-Bildung auf beliebige Dimensionen verallgemeinern. Betrachten wir nun ein zweidimensionales Gitter bestehend aus den Gitterpunkten $(x_i, y_j) \in \mathbb{R}^2$, mit $x_i = x_0 + h_x i$, $y_j = y_0 + h_y j$ und $i = 1, \dots, M$, $j = 1, \dots, N$. Gesucht ist nun $I(x, y)$, mit

$$p_{i,j} = I(x_i, y_j) = \sum_{\mu=1}^N \sum_{\lambda=1}^M c_{\lambda,\mu} b_{\lambda,\mu,h}^3(x_i, y_j),$$

wobei der zweidimensionale kardinale B-Spline gegeben ist als

$$b_{\lambda,\mu,h}^3(x, y) := b_{\lambda,h_x}^3(x) b_{\mu,h_y}^3(y).$$

Das heißt, höher dimensionale B-Spline-Basen sind das Produkt der eindimensionalen Basisfunktionen. Damit folgt

$$p_{i,j} = I(x_i, y_j) = \sum_{\mu=1}^N \sum_{\lambda=1}^M c_{\lambda,\mu} b_{\lambda,h_x}^3(x_i) b_{\mu,h_y}^3(y_j)$$

als Forderung für die gesuchte Spline-Interpolante $I(x, y)$. Aufgrund der Produktbildung der eindimensionalen Splines lassen sich die Koeffizienten $c_{\lambda,\mu}$ effizient berechnen. Zur Bestimmung der Koeffizienten $c_{\lambda,\mu}$ müssen nun zuerst M -viele $N \times N$ Gleichungssysteme

und anschließend N -viele $M \times M$ Gleichungssysteme gelöst werden.

2.2 Gitter

Für den Discretize-then-Optimize-Ansatz wird das gegebene Bild an äquidistanten Stellen ausgewertet. Typischerweise an maximal so vielen Stellen, wie die Bildmodalität Bilddaten zur Verfügung stellt. Die Menge dieser Punkte wird nun als *Gitter* bezeichnet. Ausgehend vom Bildbereich $\Omega = (\omega_1, \omega_2) \times (\omega_3, \omega_4)$ und der Anzahl der Zellen $\mathbf{m} = (m_1, m_2)$ werden drei gängige Gittertypen vorgestellt.

Cell-centered Gitter Bei einem cell-centered Gitter liegen die Gitterpunkte jeweils zentriert in einer Zelle, d. h. das Gitter ist gegeben durch die Schrittweite $\mathbf{h} = (h_1, h_2)$, mit $h_1 = \frac{\omega_2 - \omega_1}{m_1}$, $h_2 = \frac{\omega_4 - \omega_3}{m_2}$, sowie durch die Punkte $\mathbf{x}_{i,j} = (x_i, y_j) \in \mathbb{R}^2$, mit $x_i = \omega_1 + (i - \frac{1}{2})h_1$, $y_j = \omega_3 + (j - \frac{1}{2})h_2$ und $i = 1, \dots, m_1, j = 1, \dots, m_2$ (siehe Abbildung 2.2, links).

Nodales Gitter Bei einem nodalen Gitter liegen die Gitterpunkte jeweils auf den Ecken einer Zelle, d. h. das Gitter ist gegeben durch die Schrittweite \mathbf{h} , wie beim cell-centered Gitter, sowie durch die Punkte $\mathbf{x}_{i,j} = (x_i, y_j) \in \mathbb{R}^2$, mit $x_i = \omega_1 + (i - 1)h_1$, $y_j = \omega_3 + (j - 1)h_2$ und $i = 1, \dots, m_1 + 1, j = 1, \dots, m_2 + 1$ (siehe Abbildung 2.2, Mitte).

Staggered Gitter Bei einem staggered Gitter liegen die Gitterpunkte bzgl. der einen Achse cell-centered und bzgl. der anderen Achse nodal. Somit existieren zwei Möglichkeiten, dieses Gitter zu definieren. Hier folgt eine Variante. Sei \mathbf{h} wie oben gegeben, dann werden die Gitterpunkte $(x_i, y_j)^T \in \mathbb{R}^2$ mit $x_i = \omega_1 + (i - \frac{1}{2})h_1$, $y_j = \omega_3 + (j - 1)h_2$ und $i = 1, \dots, m_1, j = 1, \dots, m_2 + 1$ konstruiert (siehe Abbildung 2.2, rechts).

Für das Folgende Optimierungsverfahren ist es notwendig, das gewählte Gitter als Vektor aufzufassen. Hierzu werden die Gitterpunkte *lexikographisch* als Spaltenvektor angeordnet. Das heißt für ein cell-centered Gitter entsprechen die ersten $m_1 m_2$ Einträge

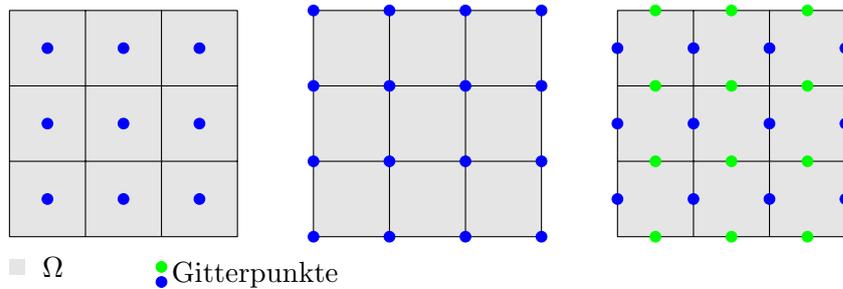


ABBILDUNG 2.2: links: Cell-Centered Gitter, Mitte: Nodal Gitter, rechts: Zwei Staggered Gitter.

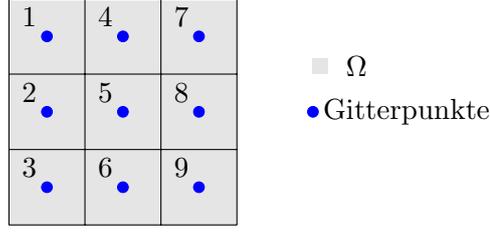


ABBILDUNG 2.3: Lexikographische Ordnung am Beispiel eines cell-Centered Gitters.

den x -Komponenten, zuerst nach Zeilen und anschließend nach Spalten sortiert (siehe Abbildung 2.3). Die restlichen $m_1 m_2$ -Einträge sind die entsprechend sortierten y -Komponenten. Damit liegt das Gitter als Vektor

$$\bar{\mathbf{x}} = \begin{pmatrix} \mathbf{e}_{m_2} \otimes \bar{\mathbf{x}}^1 \\ \bar{\mathbf{x}}^2 \otimes \mathbf{e}_{m_1} \end{pmatrix} \in \mathbb{R}^{2m_1 m_2}, \quad (2.2)$$

mit $\bar{\mathbf{x}}^1 = (x_1, \dots, x_{m_1})^T$, $\bar{\mathbf{x}}^2 = (y_1, \dots, y_{m_2})^T$ und $\mathbf{e}_m = (1, \dots, 1)^T \in \mathbb{R}^m$, vor.

Der Wechsel von einem nodalen Gitter auf ein cell-centered Gitter lässt sich im Eindimensionalen durch die Mittelwertbildung

$$\hat{\mathbf{G}}_m = \frac{1}{2} \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & & 1 & 1 \end{pmatrix} \in \mathbb{R}^{(m-1) \times m}$$

realisieren. Für den Fall $d = 2$ ergibt sich damit

$$\mathbf{G} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \left(\hat{\mathbf{G}}_{m_2+1} \otimes \hat{\mathbf{G}}_{m_1+1} \right), \quad \mathbf{G} \in \mathbb{R}^{2m_1 m_2 \times 2(m_1+1)(m_2+1)},$$

so dass der Gitterwechsel durch

$$\bar{\mathbf{x}}^{\text{cc}} = \mathbf{G} \bar{\mathbf{x}}^{\text{nodal}}$$

gegeben ist. Analog sind durch

$$\begin{aligned} \mathbf{G}_y &= \hat{\mathbf{G}}_{m_2+1} \otimes \mathbf{E}_{m_1}, & \mathbf{G}_y &\in \mathbb{R}^{m_1 m_2 \times m_1(m_2+1)}, \\ \mathbf{G}_x &= \mathbf{E}_{m_2} \otimes \hat{\mathbf{G}}_{m_1+1}, & \mathbf{G}_x &\in \mathbb{R}^{m_1 m_2 \times (m_1+1)m_2}, \end{aligned}$$

mit $\mathbf{E}_m \in \mathbb{R}^{m \times m}$ ist die Einheitsmatrix, die Mittelungs-Operatoren gegeben, die einen Gitterwechsel von staggered nach cell-centered realisieren.

2.3 Distanzmaß

Die Diskretisierung des Distanzmaßes basiert auf der Wahl eines cell-centered Gitters und der Mittelpunktsregel. Sei $\bar{\mathbf{x}} \in \mathbb{R}^{2m_1m_2}$ cell-centered auf $\Omega = (\omega_1, \omega_2) \times (\omega_3, \omega_4)$ gegeben, mit Schrittweite $\mathbf{h} = (h_1, h_2)$. Dann gilt mit der Mittelpunktsregel

$$\begin{aligned} \text{SSD}(R, T) &= \frac{1}{2} \int_{\Omega} (T(x, y) - R(x, y))^2 dx dy \\ &= \frac{h_1 h_2}{2} \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} (T(x_i, y_j) - R(x_i, y_j))^2 + \mathcal{O}(h_1^2 + h_2^2). \end{aligned} \quad (2.3)$$

Aus (2.2) folgt, dass $\bar{x}_1, \dots, \bar{x}_{m_1 m_2}$ alle x -Werte und $\bar{x}_{m_1 m_2 + 1}, \dots, \bar{x}_{m_1 m_2 + m_1 m_2}$ alle y -Werte des Gitters $\bar{\mathbf{x}}$ sind. Sei

$$\bar{\mathbf{R}} := (R(\bar{x}_1, \bar{x}_{m_1 m_2 + 1}), \dots, R(\bar{x}_{m_1 m_2}, \bar{x}_{m_1 m_2 + m_1 m_2}))^T \in \mathbb{R}^{m_1 m_2}$$

für $\bar{\mathbf{x}}$ fest gewählt. Analog definieren wir die Abbildung

$$\bar{\mathbf{T}} : \mathbb{R}^{2m_1 m_2} \rightarrow \mathbb{R}^{m_1 m_2}, \bar{\mathbf{x}} \mapsto \bar{\mathbf{T}}(\bar{\mathbf{x}}) := (T(\bar{x}_1, \bar{x}_{m_1 m_2 + 1}), \dots, T(\bar{x}_{m_1 m_2}, \bar{x}_{m_1 m_2 + m_1 m_2}))^T. \quad (2.4)$$

Mit diesen beiden Definitionen und durch vernachlässigen des Fehlerterms aus (2.3) ist

$$\text{SSD}^h : \mathbb{R}^{m_1 m_2} \rightarrow \mathbb{R}, \bar{\mathbf{T}} \mapsto \text{SSD}^h(\bar{\mathbf{T}}) := \frac{h_1 h_2}{2} \|\bar{\mathbf{T}} - \bar{\mathbf{R}}\|^2, \quad (2.5)$$

eine diskrete Darstellung des SSD-Maßes, mit

$$\text{SSD}^h(\bar{\mathbf{R}}, \bar{\mathbf{T}}) + \mathcal{O}(h_1^2 + h_2^2) = \text{SSD}(R, T).$$

2.4 Regularisierer

Für die Diskretisierung des diffusiven Regularisierers wird von einem nodalen Startgitter ausgegangen. Das heißt, $\bar{\mathbf{x}} \in \mathbb{R}^{2(m_1+1)(m_2+1)}$ ist nodal auf $\Omega = (\omega_1, \omega_2) \times (\omega_3, \omega_4)$ mit Schrittweite $\mathbf{h} = (h_1, h_2)$ gegeben. Weiter sei

$$\hat{\mathbf{D}}_m = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(m-1) \times m}$$

die Matrix zur Approximation der ersten Ableitung für $d = 1$. Für den zweidimensionalen Fall ergibt sich damit in x - bzw. y -Richtung

$$\begin{aligned} \mathbf{D}_x &= \frac{1}{h_1} \left(\mathbf{E}_{m_2+1} \otimes \hat{\mathbf{D}}_{m_1+1} \right) \in \mathbb{R}^{m_1(m_2+1) \times (m_1+1)(m_2+1)}, \\ \mathbf{D}_y &= \frac{1}{h_2} \left(\hat{\mathbf{D}}_{m_2+1} \otimes \mathbf{E}_{m_1+1} \right) \in \mathbb{R}^{m_2(m_1+1) \times (m_1+1)(m_2+1)}. \end{aligned}$$

Analog zu (2.4) sei die Abbildung $\bar{\varphi} : \mathbb{R}^{2(m_1+1)(m_2+1)} \rightarrow \mathbb{R}^{2(m_1+1)(m_2+1)}$ mit $\bar{\mathbf{x}} \mapsto \bar{\varphi}(\bar{\mathbf{x}})$ die an $\bar{\mathbf{x}}$ ausgewertete Deformation φ und $\bar{\mathbf{u}} : \mathbb{R}^{2(m_1+1)(m_2+1)} \rightarrow \mathbb{R}^{2(m_1+1)(m_2+1)}$ mit $\bar{\mathbf{x}} \mapsto \bar{\mathbf{u}}(\bar{\mathbf{x}}) := \bar{\varphi}(\bar{\mathbf{x}}) - \bar{\mathbf{x}}$ die Verrückung des Gitters. Mit $\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2 \in \mathbb{R}^{(m_1+1)(m_2+1)}$ werden die x - bzw. y -Komponenten in $\bar{\mathbf{u}}$ bezeichnet, so dass $\bar{\mathbf{u}} = (\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2)^T$.

Wir betrachten nun den diffusiven Regularisierer. Unter Verwendung der Mittelpunktsregel erhalten wir analog zum SSD-Maß

$$\begin{aligned} \mathcal{S}(\mathbf{u}) &= \frac{1}{2} \int_{\Omega} \|\nabla u\|_2^2 + \|\nabla v\|_2^2 \, dx \, dy, \\ &= \frac{1}{2} \int_{\Omega} (\partial_x u)^2 + (\partial_y u)^2 \, dx \, dy + \frac{1}{2} \int_{\Omega} (\partial_x v)^2 + (\partial_y v)^2 \, dx \, dy \\ &= \frac{h_1 h_2}{2} \sum_{i=1}^{m_1+1} \sum_{j=1}^{m_2+1} (\partial_x u(x_i, y_j))^2 + (\partial_y u(x_i, y_j))^2 + \\ &\quad \frac{h_1 h_2}{2} \sum_{i=1}^{m_1+1} \sum_{j=1}^{m_2+1} (\partial_x v(x_i, y_j))^2 + (\partial_y v(x_i, y_j))^2 + \mathcal{O}(h_1^2 + h_2^2), \quad (2.6) \end{aligned}$$

wobei die Ableitungen $\partial_x u, \partial_y u, \partial_x v$ und $\partial_y v$ an den cell-centered Punkten benötigt werden. Durch Anwendung der Operatoren $\mathbf{D}_x, \mathbf{D}_y$ werden die Ableitungen zunächst auf einem staggered Gitter genähert (Abbildung 2.4 (links oben)). Danach werden sie quadriert (Abbildung 2.4 (rechts oben)) und mit den Operatoren $\mathbf{G}_x, \mathbf{G}_y$ in die Zellmitte gemittelt (Abbildung 2.4 (links unten)). Und durch Weglassen des Fehlerterms aus (2.6) erhält man nun

$$\begin{aligned} &\int_{\Omega} (\partial_x u)^2 + (\partial_y u)^2 \, dx \, dy \\ &\approx h_1 h_2 \left(\frac{1}{h_1^2} \mathbf{e}^T \mathbf{G}_y \mathbf{diag}(\mathbf{D}_x \bar{\mathbf{u}}_1) \mathbf{D}_x \bar{\mathbf{u}}_1 + \frac{1}{h_2^2} \mathbf{e}^T \mathbf{G}_x \mathbf{diag}(\mathbf{D}_y \bar{\mathbf{u}}_1) \mathbf{D}_y \bar{\mathbf{u}}_1 \right) \\ &= h_1 h_2 \left(\frac{1}{h_1^2} \bar{\mathbf{u}}_1^T \mathbf{D}_x^T \mathbf{diag}(\mathbf{G}_y^T \mathbf{e}) \mathbf{D}_x \bar{\mathbf{u}}_1 + \frac{1}{h_2^2} \bar{\mathbf{u}}_1^T \mathbf{D}_y^T \mathbf{diag}(\mathbf{G}_x^T \mathbf{e}) \mathbf{D}_y \bar{\mathbf{u}}_1 \right) \\ &= h_1 h_2 \bar{\mathbf{u}}_1^T \underbrace{\left(\frac{1}{h_1^2} \mathbf{D}_x^T \mathbf{diag}(\mathbf{G}_y^T \mathbf{e}) \mathbf{D}_x + \frac{1}{h_2^2} \mathbf{D}_y^T \mathbf{diag}(\mathbf{G}_x^T \mathbf{e}) \mathbf{D}_y \right)}_{=:\hat{\mathbf{A}}^h} \bar{\mathbf{u}}_1 \\ &= h_1 h_2 \bar{\mathbf{u}}_1^T \hat{\mathbf{A}}^h \bar{\mathbf{u}}_1, \end{aligned}$$

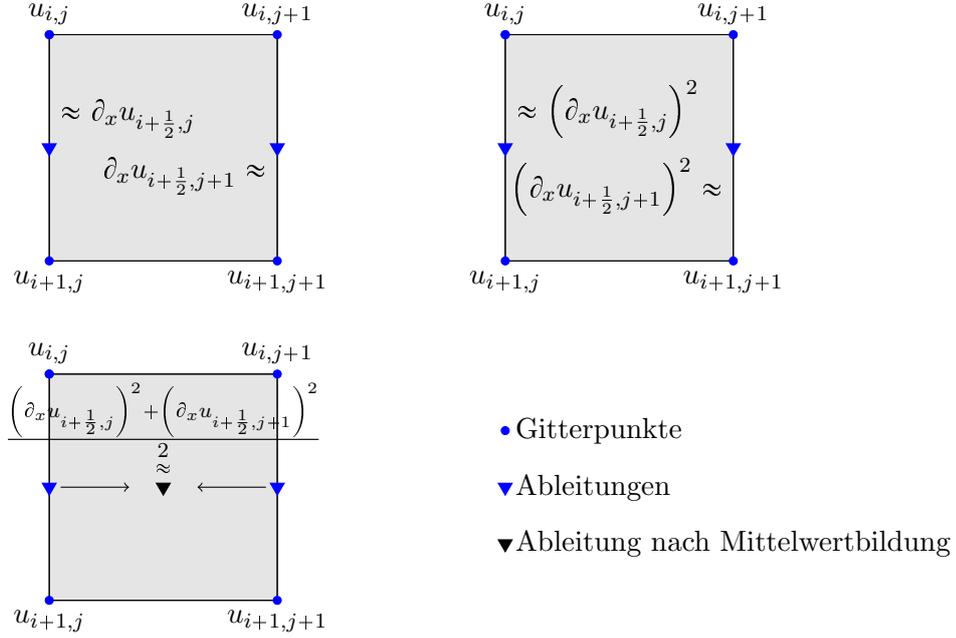


ABBILDUNG 2.4: Cell-centered Ableitung in x -Richtung. Ableitung in y -Richtung folgt analog.
 links oben: Ableitungen auf staggered Gitter in x -Richtung durch Operator \mathbf{D}_x .
 rechts oben: Quadrieren der approximierten Ableitungen durch Operator $\mathbf{diag}(\cdot)$.
 links unten: Mittelwertbildung mit \mathbf{G}_y .

mit $\mathbf{e} = (1, \dots, 1)^T \in \mathbb{R}^{m_1 m_2}$ und Diagonalmatrix $\mathbf{diag}(\mathbf{w}) = \begin{pmatrix} \omega_1 & & \\ & \ddots & \\ & & \omega_m \end{pmatrix} \in \mathbb{R}^{m \times m}$. Der noch fehlende Term wird identisch diskretisiert. Damit folgt für die diskrete Darstellung des diffusiven Regularisierers

$$\begin{aligned}
 \mathcal{S}(\mathbf{u}) \approx \mathcal{S}^h(\bar{\mathbf{u}}) &:= \frac{h_1 h_2}{2} \left(\bar{\mathbf{u}}_1^T \hat{\mathbf{A}}^h \bar{\mathbf{u}}_1 + \bar{\mathbf{u}}_2^T \hat{\mathbf{A}}^h \bar{\mathbf{u}}_2 \right) \\
 &= h_1 h_2 \frac{1}{2} \bar{\mathbf{u}}^T \underbrace{\begin{pmatrix} \hat{\mathbf{A}}^h & \\ & \hat{\mathbf{A}}^h \end{pmatrix}}_{=:\mathbf{A}^h} \bar{\mathbf{u}} \\
 &= h_1 h_2 \frac{1}{2} \bar{\mathbf{u}}^T \mathbf{A}^h \bar{\mathbf{u}}, \tag{2.7}
 \end{aligned}$$

mit $\mathbf{A}^h \in \mathbb{R}^{2(m_1+1)(m_2+1) \times 2(m_1+1)(m_2+1)}$ als die diskrete Darstellung des diffusiven Regularisierers. Man beachte, dass bei der gewählten Diskretisierung aufgrund der Wahl eines nodalen Gitters keine explizite Modellierung von Randbedingungen erforderlich ist.

Mit (2.5) und (2.7) erhält man schließlich

$$\begin{aligned}\mathcal{J}^h(\bar{\mathbf{u}}) &= \text{SSD}^h(\bar{\mathbf{T}}(\bar{\mathbf{u}})) + \alpha \mathcal{S}^h(\bar{\mathbf{u}}) \\ &= \frac{h_1 h_2}{2} \|\bar{\mathbf{T}}(\bar{\mathbf{u}}) - \bar{\mathbf{R}}\|^2 + \alpha \frac{h_1 h_2}{2} \bar{\mathbf{u}}^T \mathbf{A}^h \bar{\mathbf{u}}\end{aligned}$$

die diskretisierte Zielfunktion für die Registrierung.

3 Volumenbeschränkung

Mit dem Framework aus Kapitel 1 und 2 ist bei guter Wahl von α eine glatte Deformation gewährleistet. Jedoch ist eine glatte Deformation nicht zwangsläufig auch plausibel. Im schlimmsten Fall können trotz ausreichender Glättung der Zielfunktion (1.7) starke Stauchungen, Streckungen oder sogar Falten im Gitter entstehen (siehe Abbildung 3.1). Dies liegt darin begründet, dass der Regularisierer ein Energie-Term auf ganz Ω ist und somit lokale Ausreißer im Gitter nicht verhindern kann. Insbesondere für Faltungen gibt es im medizinischen Kontext keine plausible Begründung. Im Rahmen dieser Arbeit wird nun ein erweiterter Ansatz zur Vermeidung von Falten vorgestellt.

Durch den hier vorgestellten Ansatz wird die Forderung nach Faltenfreiheit direkt in das Framework aus den vorherigen Kapiteln implementiert. Dies geschieht durch Hinzufügen von Nebenbedingungen an die Registrierung. Im Folgenden wird der Begriff *Volumen* als Verallgemeinerung einer d -dimensionalen Fläche verwendet.

3.1 Kontinuierliche Formulierung

Eine Faltung im Gitter bedeutet übertragen ins Kontinuierliche, dass φ keine bijektive Abbildung mehr ist. Das heißt, um Faltenbildung zu vermeiden, muss eine Forderung nach Bijektivität von φ eingebunden werden. Um dies zu modellieren, nutzen wir folgenden Satz:

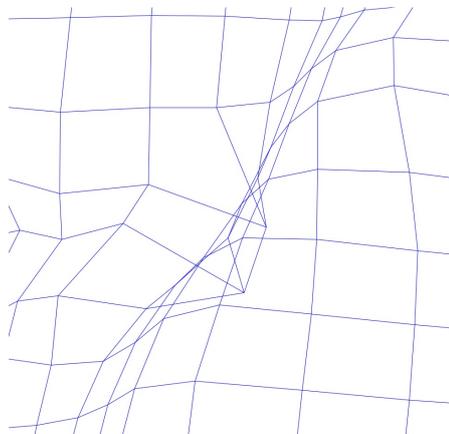


ABBILDUNG 3.1: Ausschnitt eines Gitters mit Faltung.

Satz (Umkehrabbildung, siehe [5]). Sei $U, V \subseteq \mathbb{R}^d$ offen und $\varphi : U \rightarrow V$ mit $\varphi \in C^1(U, V)$. Sei $\mathbf{a} \in U$ und $\mathbf{b} := \varphi(\mathbf{a}) \in V$. Wenn für die Funktional-Determinante gilt

$$\det \nabla \varphi(\mathbf{a}) \neq 0,$$

dann gibt es eine offene Umgebung $U_{\mathbf{a}} \subseteq U$ von \mathbf{a} und eine offene Umgebung $V_{\mathbf{b}} \subseteq V$ von \mathbf{b} , so dass $\varphi^{-1} \in C^1(V_{\mathbf{b}}, U_{\mathbf{a}})$ und φ die Menge $U_{\mathbf{a}}$ bijektiv auf $V_{\mathbf{b}}$ abbildet.

Basierend auf diesem Satz fordern wir nun

$$\det \nabla \varphi(\mathbf{x}) \neq 0, \quad \forall \mathbf{x} \in \Omega.$$

Aufgrund der Stetigkeit von φ muss die Funktionaldeterminante nun entweder überall positiv oder überall negativ sein. Da für $\varphi(\mathbf{x}) = \mathbf{x}$ gilt $\det \nabla \varphi = 1 > 0$, fordern wir

$$0 < \det \nabla \varphi(\mathbf{x}) < \infty, \quad \forall \mathbf{x} \in \Omega, \quad (3.1)$$

so dass φ bijektiv auf Ω ist.

Neben der Forderung nach Bijektivität der Deformation sind wir im Folgenden daran interessiert die lokale Änderung des Volumens, hervorgerufen durch die Deformation, zu beschreiben. Das heißt, für beliebige $S \subset \Omega$ und gegebene Konstanten v_{\min}, v_{\max} soll gelten

$$0 < v_{\min} \leq \frac{\text{vol}(\varphi(S))}{\text{vol}(S)} \leq v_{\max} < \infty, \quad (3.2)$$

wobei

$$\text{vol}(\varphi(S)) = \int_{\varphi(S)} d\mathbf{y} = \int_S \det \nabla \varphi d\mathbf{x}, \quad \forall S \subset \Omega$$

aus dem Transformationssatz [2] folgt.

Es wird nun gezeigt, dass die Funktionaldeterminante der punktweisen Forderung aus (3.2) entspricht.

Satz. Sei $\Omega \subset \mathbb{R}^d$ offen und φ eine zweimal stetig differenzierbare Funktion. Dann gilt

$$0 < v_{\min} \leq \frac{\text{vol}(\varphi(S))}{\text{vol}(S)} \leq v_{\max} < \infty, \quad \forall S \subset \Omega,$$

genau dann, wenn

$$0 < v_{\min} \leq \det \nabla \varphi(\mathbf{x}) \leq v_{\max} < \infty, \quad \forall \mathbf{x} \in \Omega.$$

Beweis. Zuerst die Rückrichtung: Falls $0 < v_{\min} \leq \det \nabla \varphi(\mathbf{x}) \leq v_{\max} < \infty, \quad \forall \mathbf{x} \in \Omega,$

dann gilt für $S \subset \Omega$

$$\text{vol}(\varphi(S)) = \int_S \det \nabla \varphi \, d\mathbf{x} \geq v_{\min} \int_S d\mathbf{x} = v_{\min} \text{vol}(S) \Rightarrow \frac{\text{vol}(\varphi(S))}{\text{vol}(S)} \geq v_{\min}.$$

Der Beweis für die obere Schranke erfolgt analog.

Es folgt die Hinrichtung: Sei $\mathbf{x}_0 \in \Omega$ beliebig, d. h. $\exists \varepsilon_0 > 0$, so dass $B_\varepsilon(\mathbf{x}_0) := \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_0\|_2 < \varepsilon\} \subset \Omega$, für alle $\varepsilon < \varepsilon_0$. Wir zeigen nun

$$\lim_{\varepsilon \rightarrow 0} \frac{\text{vol}(\varphi(B_\varepsilon(\mathbf{x}_0)))}{\text{vol}(B_\varepsilon(\mathbf{x}_0))} = \det \nabla \varphi(\mathbf{x}_0).$$

Da für alle ε der Quotient nach Voraussetzung beschränkt ist, gilt dies auch für den Grenzwert. Mit $f(\mathbf{x}) := \det \nabla \varphi(\mathbf{x})$ und seinem ersten Taylorpolynom mit Lagrange-Restglied folgt

$$\begin{aligned} & \left| \frac{\int_{B_\varepsilon(\mathbf{x}_0)} f(\mathbf{x}) \, d\mathbf{x}}{\text{vol}(B_\varepsilon(\mathbf{x}_0))} - f(\mathbf{x}_0) \right| \\ &= \left| \frac{\int_{B_\varepsilon(\mathbf{x}_0)} f(\mathbf{x}_0) + \nabla f(\mathbf{z})^T (\mathbf{x} - \mathbf{x}_0) \, d\mathbf{x}}{\text{vol}(B_\varepsilon(\mathbf{x}_0))} - f(\mathbf{x}_0) \right|, \quad \mathbf{z} := \mathbf{z}(\mathbf{x}) \in B_\varepsilon(\mathbf{x}_0) \\ &= \left| \frac{\int_{B_\varepsilon(\mathbf{x}_0)} f(\mathbf{x}_0) + \nabla f(\mathbf{z})^T (\mathbf{x} - \mathbf{x}_0) \, d\mathbf{x} - f(\mathbf{x}_0) \text{vol}(B_\varepsilon(\mathbf{x}_0))}{\text{vol}(B_\varepsilon(\mathbf{x}_0))} \right| \\ &= \left| \frac{f(\mathbf{x}_0) \text{vol}(B_\varepsilon(\mathbf{x}_0)) + \int_{B_\varepsilon(\mathbf{x}_0)} \nabla f(\mathbf{z})^T (\mathbf{x} - \mathbf{x}_0) \, d\mathbf{x} - f(\mathbf{x}_0) \text{vol}(B_\varepsilon(\mathbf{x}_0))}{\text{vol}(B_\varepsilon(\mathbf{x}_0))} \right| \\ &= \left| \frac{\int_{B_\varepsilon(\mathbf{x}_0)} \nabla f(\mathbf{z})^T (\mathbf{x} - \mathbf{x}_0) \, d\mathbf{x}}{\text{vol}(B_\varepsilon(\mathbf{x}_0))} \right| \end{aligned}$$

$$\begin{aligned}
& \leq \frac{\int_{B_\varepsilon(\mathbf{x}_0)} |\nabla f(\mathbf{z})^T (\mathbf{x} - \mathbf{x}_0)| \, d\mathbf{x}}{|\text{vol}(B_\varepsilon(\mathbf{x}_0))|} \\
& \stackrel{\text{C.S. Ungl.}}{\leq} \frac{\int_{B_\varepsilon(\mathbf{x}_0)} \|\nabla f(\mathbf{z})^T\|_2 \|\mathbf{x} - \mathbf{x}_0\|_2 \, d\mathbf{x}}{|\text{vol}(B_\varepsilon(\mathbf{x}_0))|}
\end{aligned}$$

Da φ nach Voraussetzung zweimal stetig differenzierbar ist, können wir $\|\nabla f(\mathbf{z})^T\|_2$ durch $M := \sup_{\mathbf{z} \in \Omega} \|\nabla f(\mathbf{z})^T\|_2$ abschätzen. Ebenfalls können wir $\|\mathbf{x} - \mathbf{x}_0\|_2 < \varepsilon$ abschätzen. Damit erhalten wir

$$\left| \frac{\text{vol}(\varphi(B_\varepsilon(\mathbf{x}_0)))}{\text{vol}(B_\varepsilon(\mathbf{x}_0))} - \det \nabla \varphi(\mathbf{x}_0) \right| \leq M\varepsilon \xrightarrow{\varepsilon \rightarrow 0} 0.$$

□

Somit ist die Forderung

$$v_{\min} \leq \det \nabla \varphi(\mathbf{x}) \leq v_{\max}, \quad \forall \mathbf{x} \in \Omega, \tag{3.3}$$

mit $0 < v_{\min} \leq v_{\max}$, äquivalent zu (3.2). Es stellt sich also heraus, dass die Forderungen nach Bijektivität und Volumenbeschränkung gemeinsam abgedeckt werden durch (3.3).

Damit lautet das Optimierungsproblem mit der Zielfunktion \mathcal{J} aus (1.7) nun

$$\begin{aligned}
& \min_{\varphi} \mathcal{J}(\varphi), \\
& \text{u. d. Nb.:} \quad v_{\min} \leq \det \nabla \varphi(\mathbf{x}) \leq v_{\max}, \quad \forall \mathbf{x} \in \Omega.
\end{aligned}$$

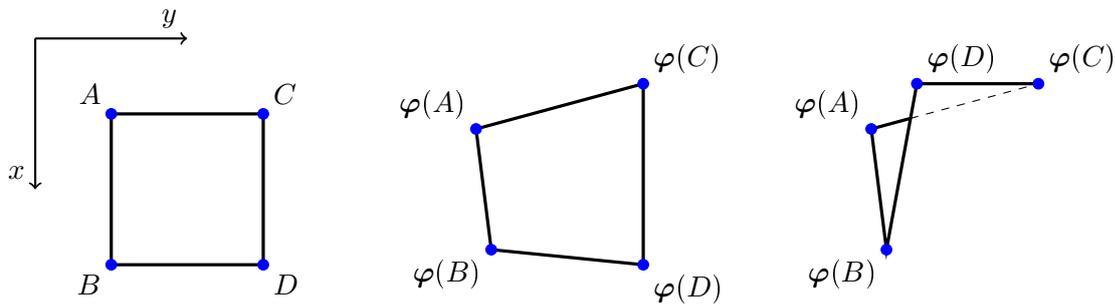


ABBILDUNG 3.2: links: Undeformierte Zelle
Mitte: Deformierte Zelle ohne Faltung
rechts: Deformierte Zelle mit Faltung

3.2 Diskretisierung

Im vorhergehenden Abschnitt wurde eine punktweise Forderung an die Deformation hergeleitet, so dass Bijektivität und Volumenbeschränkung gewährleistet sind. Diese Nebenbedingung wird nun diskretisiert. Das heißt, die Einhaltung von (3.3) wird nun an den als Kontrollpunkten gewählten Gitterpunkten geprüft. Wir wählen als Diskretisierungsansatz ein nodales Startgitter und betrachten nun eine Zelle wie in Abbildung 3.2 (links). Es bezeichnen hier, typisch für geometrische Betrachtungen, Großbuchstaben die Gitterpunkte. Als Faltung einer Zelle wird hier der Fall bezeichnet, bei dem ein Eckpunkt der Zelle eine der ihm gegenüberliegenden Kanten überschreitet (vgl. Abbildung 3.2 (rechts)).

Um nun die Funktionaldeterminante $C(\varphi) := \det \nabla \varphi$, am Gitterpunkt $\varphi(A)$ auszuwerten, wird der Differentialoperator ∇ approximiert. Durch die äquidistante Konstruktion des nodalen Gitters gilt $B = A + h_x e_1$ und $C = A + h_y e_2$ und es folgt mit Vorwärts-Differenzenquotienten

$$C_1^h(\varphi(A)) := \det \left[\frac{\varphi(B) - \varphi(A)}{h_x}, \frac{\varphi(C) - \varphi(A)}{h_y} \right] \approx \det \nabla \varphi(A)$$

als Approximation an die Funktionaldeterminante. Es wird jetzt gezeigt, dass dieser Diskretisierungsansatz auch eine Eigenschaft zum Messen von Volumen-Änderungen besitzt. Seien Punkte A, B, C wie oben gegeben. Dann ist der Flächeninhalt des durch die deformierten Punkte gegeben Dreiecks durch

$$\text{vol}(\varphi(A, B, C)) := \frac{1}{2} |(\varphi_x(B) - \varphi_x(A))(\varphi_y(C) - \varphi_y(A)) - (\varphi_x(C) - \varphi_x(A))(\varphi_y(B) - \varphi_y(A))|,$$

bestimmt. Und der Flächeninhalt des undeformierten Dreiecks bestimmt sich durch

$$\text{vol}(A, B, C) := \frac{1}{2}h_x h_y.$$

In Anlehnung an (3.2) aus der kontinuierlichen Formulierung wird auch hier der Quotient dieser Volumina betrachtet. Es gilt,

$$\begin{aligned} & \frac{\text{vol}(\varphi(A, B, C))}{\text{vol}(A, B, C)} \\ = & \frac{\frac{1}{2}|(\varphi_x(B) - \varphi_x(A))(\varphi_y(C) - \varphi_y(A)) - (\varphi_x(C) - \varphi_x(A))(\varphi_y(B) - \varphi_y(A))|}{\frac{1}{2}h_x h_y} \\ = & \frac{|(\varphi_x(B) - \varphi_x(A))(\varphi_y(C) - \varphi_y(A)) - (\varphi_x(C) - \varphi_x(A))(\varphi_y(B) - \varphi_y(A))|}{h_x h_y} \\ = & \left| \det \begin{bmatrix} \frac{\varphi_x(B) - \varphi_x(A)}{h_x} & \frac{\varphi_x(C) - \varphi_x(A)}{h_y} \\ \frac{\varphi_y(B) - \varphi_y(A)}{h_x} & \frac{\varphi_y(C) - \varphi_y(A)}{h_y} \end{bmatrix} \right| \\ = & |C_1^h(\varphi(A))|. \end{aligned}$$

Somit messen wir mit dieser Diskretisierung die Volumen-Änderung des Dreiecks, welches durch die Punkte $\varphi(A)$, $\varphi(B)$, $\varphi(C)$ aufgespannt wird. Insbesondere folgt daraus auch, dass für $C_1^h(\varphi(A)) = 1$ keine Volumenänderung durch φ in A stattgefunden hat. Für die an $C_1^h(\varphi(A))$ beteiligten Punkte lässt sich zeigen [11], dass wenn

$$C_1^h(\varphi(A)) > 0 \Rightarrow \text{Orientierungserhaltung dieser Punkte.}$$

Mit *Orientierungserhaltung* ist gemeint, dass man beim Durchlaufen der Punkte des Dreiecks im Uhrzeigersinn, nach der Deformation durch φ , die Punkte noch immer in derselben Reihenfolge durchläuft. Dieses Dreieck ist somit faltenfrei. Jedoch stellt sich schnell heraus, dass bei diesem Ansatz keine Faltenfreiheit des Gitters gegeben ist (vgl. Abbildung 3.3 (rechts)). Dies motiviert dazu, auch die zu Abbildung 3.3 (links) komplementären Dreiecke in die Berechnung mit einfließen zu lassen. Für die betrachtete Beispielzelle, bedeutet dies mit Rückwärts-Differenzenquotienten

$$C_2^h(\varphi(D)) := \det \left[\frac{\varphi(C) - \varphi(D)}{h_x}, \frac{\varphi(B) - \varphi(D)}{h_y} \right] \approx \det \nabla \varphi(D).$$

Zur Vermeidung von Vorzugsrichtungen betrachten wir die Funktionaldeterminante auch an den Punkten B und C . Mit gemischten Differenzenquotienten ergibt sich

$$\begin{aligned} C_3^h(\varphi(B)) & := \det \left[\frac{\varphi(D) - \varphi(B)}{h_y}, \frac{\varphi(A) - \varphi(B)}{h_x} \right] \approx \det \nabla \varphi(B), \\ C_4^h(\varphi(C)) & := \det \left[\frac{\varphi(A) - \varphi(C)}{h_y}, \frac{\varphi(D) - \varphi(C)}{h_x} \right] \approx \det \nabla \varphi(C). \end{aligned}$$

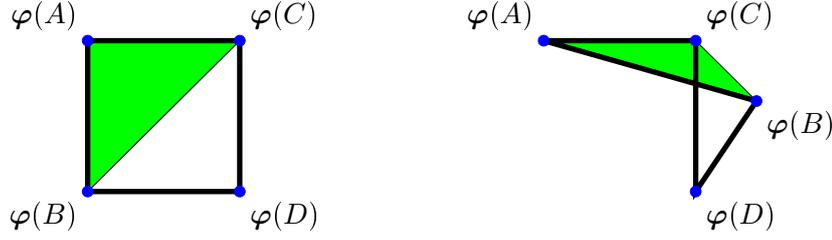


ABBILDUNG 3.3: links: Zelle mit dem von $C_1^h(\varphi(A))$ berücksichtigten Dreieck (grün)
rechts: Mögliche Faltung, trotz Orientierungserhaltung des grünen Dreiecks

Ein interessante Eigenschaft, die durch die Berechnung aller vier Determinanten zustande kommt, ist die sichergestellte Konvexität aller Zellen. Dies ergibt sich aus der Tatsache, dass kein Punkt die ihm gegenüberliegende Zell-Diagonale überschreiten darf. Im Fall $d = 3$ gilt dies nicht mehr.

Wir übertragen diese Forderungen an eine Zelle jetzt auf das gesamte Gitter. Da die Anzahl der Zellen im Gitter $m_1 m_2$ beträgt, erhalten wir $4m_1 m_2$ diskretisierte Nebenbedingungen. Das heißt, durch Einhaltung von

$$\begin{aligned}
 0 < v_{\min} &\leq C_1^h(\varphi(A_i)) \leq v_{\max} < \infty, \\
 0 < v_{\min} &\leq C_2^h(\varphi(D_i)) \leq v_{\max} < \infty, \\
 0 < v_{\min} &\leq C_3^h(\varphi(B_i)) \leq v_{\max} < \infty, \\
 0 < v_{\min} &\leq C_4^h(\varphi(C_i)) \leq v_{\max} < \infty,
 \end{aligned}$$

mit $i = 1, \dots, m_1 m_2$, ist das Gitter faltenfrei und volumenbeschränkt. Es ist hier allerdings anzumerken, dass durch die Diskretisierung lediglich „Gutartigkeit“ des Gitters gewährleistet wird. Eine Verletzung der Bijektivität von φ kann, abhängig vom Interpolationsmodell, nicht ausgeschlossen werden.

Zur Vereinfachung der Notation werden die Funktionaldeterminanten nun als eine Sequenz von c_i , $i = 1, \dots, M$, mit $M := 4m_1 m_2$, bezeichnet. Dabei sind die ersten $m_1 m_2$ vielen c_i , die lexikographisch sortierten C_1^h und die nächsten $m_1 m_2$ vielen c_i , die lexikographisch sortierten C_2^h , usw.

4 Optimierung

Zur Lösung des bedingten Optimierungsproblems

$$\begin{aligned} \min_{\bar{\mathbf{u}}} \quad & \mathcal{J}^h(\bar{\mathbf{u}}) = \text{SSD}^h(\bar{\mathbf{T}}(\bar{\mathbf{u}})) + \alpha \mathcal{S}^h(\bar{\mathbf{u}}), \\ \text{u.d. Nb.} \quad & 0 < v_{\min} \leq c_i \leq v_{\max} < \infty, \quad i = 1, \dots, M \end{aligned} \quad (4.1)$$

wird in diesem Kapitel ein Log-Barrier-Ansatz vorgestellt. Durch dieses Verfahren werden die Nebenbedingungen zunächst in die diskrete Zielfunktion eingebunden. Der daraus resultierende Vorteil ist, dass Löser für unrestringierte Probleme verwendet werden können.

4.1 Log-Barrier-Ansatz

Die Idee des hier verwendeten Log-Barrier-Ansatzes ist, dass man die punktweisen Nebenbedingungen in Form eines zusätzlichen Strafterms implementiert. Dazu benutzen wir die Singularitäten speziell konstruierter Logarithmus-Funktionen aus, um die Zielfunktion immer im zulässigen Bereich zu halten. Der zulässige Bereich für die c_i ist durch

$$\mathcal{F} := \{ \mathbf{x} \in \mathbb{R}^2 \mid 0 \leq v_{\min} \leq c_i(\mathbf{x}) \leq v_{\max} \leq \infty, i = 1, \dots, M \}$$

definiert. Die verwendeten Logarithmus-Funktionen nennt man auch Log-Barrier-Funktionen und sie haben folgende Eigenschaften: Eine Barriere-Funktion $b : \mathbb{R} \rightarrow \mathbb{R}^+ \cup \{\infty\}$, $x \mapsto b(x)$, muss überall unendlich sein, außer in $c_i(\mathcal{F} \setminus \partial\mathcal{F}) := \{c_i(\mathbf{x}) \mid \mathbf{x} \in \mathcal{F} \setminus \partial\mathcal{F}\}$, mit $\partial\mathcal{F}$ ist Rand von \mathcal{F} . Des Weiteren soll sie auf $c_i(\mathcal{F} \setminus \partial\mathcal{F})$ zweimal stetig differenzierbar sein. Das heißt, $b(c_i)$ soll nach Konstruktion für $c_i \rightarrow v_{\min}$ bzw. $c_i \rightarrow v_{\max}$ gegen unendlich gehen. Ein Verlassen des zulässigen Bereichs wird mit einem Funktionswert von ∞ bestraft.

Die Idee der Barriere-Funktion, die hier zum Einsatz kommt, basiert auf der Konstruktion einer Logarithmus-Funktion mit zwei Singularitäten. Mit ihr sind v_{\min}, v_{\max} sowie das globale Minimum $v_0 \in (v_{\min}, v_{\max})$ frei wählbar. Sie hat die Darstellung

$$b(x) := \begin{cases} -\alpha \log(x - v_{\min}) - \beta \log(v_{\max} - x) - b_0, & \text{wenn } x \in c_i(\mathcal{F} \setminus \partial\mathcal{F}), \\ \infty & \text{sonst} \end{cases}$$

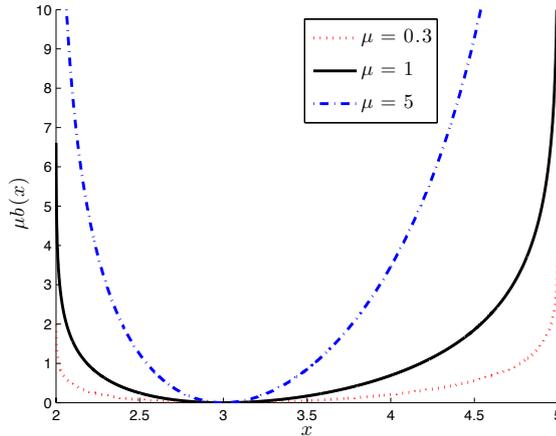


ABBILDUNG 4.1: Barriere-Funktion mit Vorfaktor μ und Parametern $v_{\min} = 2$, $v_{\max} = 5$ und $v_0 = 3$.

mit $\alpha = v_0 - v_{\min}$, $\beta = v_{\max} - v_0$ und $b_0 = -\alpha \log(\alpha) - \beta \log(\beta)$ (vgl. Abbildung 4.1). Durch diese Konstruktion ergeben sich folgende Eigenschaften

- für $x \rightarrow v_{\min}$, sowie $x \rightarrow v_{\max}$, geht $b(x) \rightarrow \infty$
- das globale Minimum v_0 ist frei wählbar, mit $b(v_0) = 0$
- die Funktion ist streng konvex.

Mit Hilfe dieser Log-Barrier-Funktion erweitern wir die diskrete Zielfunktion aus (4.1) um

$$\mathcal{V}^h(\bar{\mathbf{u}}) := \sum_{i=1}^M b(c_i)$$

und betrachten nun das unrestringierte Optimierungsproblem

$$\min_{\bar{\mathbf{u}}} \mathcal{J}_\mu^h(\bar{\mathbf{u}}) = \min_{\bar{\mathbf{u}}} \left\{ \text{SSD}^h(\bar{\mathbf{T}}(\bar{\mathbf{u}})) + \alpha \mathcal{S}^h(\bar{\mathbf{u}}) + \mu \mathcal{V}^h(\bar{\mathbf{u}}) \right\}, \quad (4.2)$$

mit $\mu > 0$. Die Singularitäten von $b(x)$ bleiben unbeeinflusst von der konkreten Wahl von μ , jedoch lässt sich das Verhalten der Funktion im offenen Intervall (v_{\min}, v_{\max}) kontrollieren. Das heißt, für ein kleines μ ist der Strafterm in der Nähe von v_{\min}, v_{\max} niedriger als für ein großes μ (vgl. Abbildung 4.1). Ein Minimierer $\bar{\mathbf{u}}(\mu)$ dieser Zielfunktion ist allerdings nur eine Näherung an die Lösung von (4.1). Es lässt sich jedoch zeigen, dass $\bar{\mathbf{u}}(\mu)$ für $\mu \rightarrow 0$ gegen eine Lösung des bedingten Problems (4.1) konvergiert [6].

Modell-Algorithmus

Die algorithmische Umsetzung der Forderung nach $\mu \rightarrow 0$ wurde folgendermaßen realisiert:

Wähle Startparameter $\alpha, \mu_0, \mu_{\min}, \tau \in (0, 1), \bar{\mathbf{u}}_0, k = 0$
while $\mu_k \geq \mu_{\min}$ **do**

Bestimme $\bar{\mathbf{u}}_{k+1} \in \arg \min_{\bar{\mathbf{u}}} \{ \text{SSD}^h(\bar{\mathbf{T}}(\bar{\mathbf{u}})) + \alpha \mathcal{S}^h(\bar{\mathbf{u}}) + \mu \mathcal{V}^h(\bar{\mathbf{u}}) \}$ ausgehend von Startwert $\bar{\mathbf{u}}_k$

$\mu_{k+1} \leftarrow \tau \mu_k$

end while

Für das unbeschränkte Optimierungsproblem

$$\min_{\bar{\mathbf{u}}} \mathcal{J}_\mu^h(\bar{\mathbf{u}})$$

aus (4.2) wird im folgenden Abschnitt ein ableitungsbasiertes Optimierungsverfahren vorgestellt.

4.2 Gauss-Newton-Verfahren

Newton-Verfahren

Zunächst wird der Ansatz des klassischen Newton-Verfahrens betrachtet, da dieses die Grundlage für das im Folgenden verwendete Gauss-Newton-Verfahren bildet. Ausgehend von einer zweimal stetig differenzierbaren Funktion f und ihrer quadratischen Taylor-Approximation

$$f(\mathbf{x}_k + \mathbf{p}_k) \approx f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p}_k + \mathbf{p}_k^T \nabla^2 f(\mathbf{x}_k) \mathbf{p}_k,$$

erhält man, durch Ableiten und Nullsetzen, die Newtongleichung

$$\nabla^2 f(\mathbf{x}_k) \mathbf{p}_k = -\nabla f(\mathbf{x}_k). \quad (4.3)$$

Dabei ist \mathbf{p}_k eine sog. Abstiegsrichtung von f im Punkt \mathbf{x}_k , für die gilt

$$\exists t > 0, \text{ so dass } f(\mathbf{x}_k + \alpha \mathbf{p}_k) < f(\mathbf{x}_k), \forall \alpha \in (0, t).$$

Mit Hilfe einer Taylorentwicklung kann man zeigen [6], dass dies äquivalent ist zu

$$\nabla f(\mathbf{x}_k)^T \mathbf{p}_k < 0.$$

Durch Lösen des linearen Gleichungssystems (4.3) erhält man \mathbf{p}_k , so dass sich \mathbf{x}_{k+1} durch

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k,$$

mit geeigneter Schrittweite $\alpha > 0$ bestimmt.

Für die Lösung des diskreten Optimierungsproblem aus dem vorhergehenden Abschnitt wird nun das Gauss-Newton-Verfahren vorgestellt.

Das Gauss-Newton-Verfahren verwendet das oben vorgestellte Schema. Jedoch wird statt der Hesse-Matrix eine approximierte zweite Ableitung verwendet. Der genaue Aufbau der Ableitungen wird in Abschnitt 4.3 besprochen.

Bestimmung der Abstiegsrichtung

Durch Lösen des Gleichungssystems

$$\mathbf{H}_k \mathbf{p}_k = -\mathbf{g}_k, \quad (4.4)$$

mit $\mathbf{H}_k \approx \nabla^2 \mathcal{J}_\mu^h(\bar{\mathbf{u}}_k)$ und $\mathbf{g}_k = \nabla \mathcal{J}_\mu^h(\bar{\mathbf{u}}_k)$, erhält man \mathbf{p}_k als Gauss-Newton-Abstiegsrichtung. Die Matrix \mathbf{H}_k wird im Folgenden so konstruiert, dass sie symmetrisch-positivdefinit ist, so dass sichergestellt ist, dass \mathbf{p}_k eine Abstiegsrichtung ist [6]. Damit lässt sich die nächste Iterierte $\bar{\mathbf{u}}_{k+1}$ durch

$$\bar{\mathbf{u}}_{k+1} = \bar{\mathbf{u}}_k + \alpha \mathbf{p}_k, \quad (4.5)$$

mit geeigneter Schrittweite α , bestimmen.

Bestimmung der Schrittweite

Zur Bestimmung der Schrittweite α wird hier die Armijo-Bedingung

$$\mathcal{J}_\mu^h(\bar{\mathbf{u}}_k + \alpha \mathbf{p}_k) \leq \mathcal{J}_\mu^h(\bar{\mathbf{u}}_k) + c\alpha \nabla \mathcal{J}_\mu^h(\bar{\mathbf{u}}_k)^T \mathbf{p}_k,$$

mit Konstantem $c = 10^{-4}$ verwendet. Ausgehend von $\alpha = 1$ wird geprüft, ob die Armijo-Bedingung erfüllt ist. Falls nicht, wird α so lange halbiert, bis die Bedingung erfüllt ist. Weitere Details finden sich in [6].

Abbruchkriterien

Die oben angegebene Iteration wird so lange durchgeführt, bis man hinreichend genau am Minimum angelangt ist. Zur klaren Eingrenzung dieser schwammigen Formulierung werden hier die Kriterien nach Gill, Murray und Wright verwendet. Sie lauten

$$(K1) \quad \|\mathcal{J}_\mu^h(\bar{\mathbf{u}}_{k+1}) - \mathcal{J}_\mu^h(\bar{\mathbf{u}}_k)\|_2 < \tau \left(1 + \|\mathcal{J}_\mu^h(\bar{\mathbf{u}}_k)\|_2\right)$$

(Funktionswerte dürfen sich kaum noch ändern)

$$(K2) \quad \|\bar{\mathbf{u}}_{k+1} - \bar{\mathbf{u}}_k\|_2 < \sqrt{\tau} (1 + \|\bar{\mathbf{u}}_k\|_2)$$

(Schrittweite muss klein genug sein)

$$(K3) \quad \|\nabla \mathcal{J}_\mu^h(\bar{\mathbf{u}}_k)\|_2 < \sqrt[3]{\tau} \left(1 + \|\mathcal{J}_\mu^h(\bar{\mathbf{u}}_k)\|_2\right)$$

(Norm des Gradienten muss klein sein)

$$(K4) \quad \|\nabla \mathcal{J}_\mu^h(\bar{\mathbf{u}}_k)\|_2 < \varepsilon$$

(Norm des Gradienten darf nicht unterhalb der Maschinengenauigkeit liegen)

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

(Maximale Iterationszahl darf nicht überschritten werden)

mit $\tau = 10^{-5}$ und der Maschinengenauigkeit ε . Die Iteration wird abgebrochen, sobald (K1) \wedge (K2) \wedge (K3) erfüllt sind bzw. wenn (K4) \vee (K5) erfüllt sind.

4.3 Ableitungen der Zielfunktion

Es wird nun eine Zielfunktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ mit $f(\mathbf{x}) := \frac{1}{2} \|\mathbf{r}(\mathbf{x})\|^2$ und Residuum $\mathbf{r} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ betrachtet. Das Gauss-Newton-Verfahren verwendet dessen spezielle Struktur zur Approximation der zweiten Ableitung von f . Die erste Ableitung ist durch

$$\nabla f(\mathbf{x}) = \mathbf{g} = \nabla \mathbf{r}(\mathbf{x})^T \mathbf{r}(\mathbf{x}), \quad (4.6)$$

mit $\nabla \mathbf{r}(\mathbf{x})$ ist Jacobi-Matrix von $\mathbf{r}(\mathbf{x})$ gegeben. Für die zweite Ableitung von f gilt

$$\nabla^2 f(\mathbf{x}) = \left(\nabla \mathbf{r}(\mathbf{x})^T \nabla \mathbf{r}(\mathbf{x}) + \sum_{i=1}^m \nabla^2 r_i r_i \right),$$

mit der i -ten Komponente r_i von \mathbf{r} und zugehöriger Hesse-Matrix $\nabla^2 r_i$. Es lässt sich nun zeigen, dass nahe des Minimums von f der zweite Summand vernachlässigt werden kann [6]. Als Approximation an die Hesse-Matrix betrachten wir nun

$$\nabla^2 f(\mathbf{x}) \approx \mathbf{H} = \nabla \mathbf{r}(\mathbf{x})^T \nabla \mathbf{r}(\mathbf{x}). \quad (4.7)$$

Durch Einsetzen dieser Approximation in (4.3) erhält man

$$\left(\nabla \mathbf{r}(\mathbf{x}_k)^T \nabla \mathbf{r}(\mathbf{x}_k) \right) \mathbf{p}_k = -\nabla \mathbf{r}(\mathbf{x}_k)^T \mathbf{r}(\mathbf{x}_k),$$

als Gleichungssystem zur Bestimmung der Gauss-Newton-Abstiegsrichtung \mathbf{p}_k .

Erste Ableitung

Nun werden die ersten Ableitungen der in (4.2) verwendeten Terme bestimmt. Für das SSD-Maß

$$\text{SSD}^h(\bar{\mathbf{T}}(\bar{\mathbf{u}})) = \frac{h_1 h_2}{2} \|\bar{\mathbf{T}}(\bar{\mathbf{u}}) - \bar{\mathbf{R}}\|^2,$$

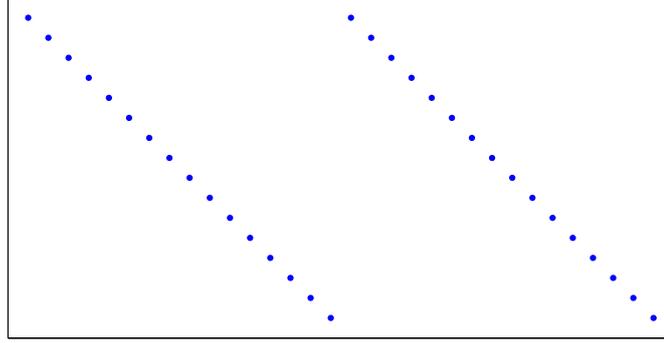


ABBILDUNG 4.2: Struktur der Jacobi-Matrix $\nabla \bar{T}(\bar{u})$.

gilt mit (4.6)

$$\nabla \text{SSD}^h(\bar{T}(\bar{u})) = h_1 h_2 \nabla \bar{T}(\bar{u})^T [\bar{T}(\bar{u}) - \bar{\mathbf{R}}].$$

Dabei hat $\nabla \bar{T}(\bar{u})$ eine Struktur wie in Abbildung 4.2.

Die linke Diagonale entspricht dabei den Ableitungen nach x und die rechte Diagonale den Ableitungen nach y . Für den diffusiven Regularisierer

$$\mathcal{S}^h(\bar{u}) = \frac{h_1 h_2}{2} \bar{u}^T \mathbf{A}^h \bar{u},$$

folgt für die erste Ableitung

$$\nabla \mathcal{S}^h(\bar{u}) = h_1 h_2 \mathbf{A}^h \bar{u}.$$

Ausgehend vom Volumen-Strafterm

$$\mathcal{V}^h(\bar{u}) = \sum_{i=1}^M b(c_i) = \mathbf{e}^T \mathbf{b}(\mathbf{c}),$$

mit $\mathbf{b}(\mathbf{c}) \in \mathbb{R}^M$ ist Vektor der $b(c_i)$'s und $\mathbf{e} := (1, \dots, 1)^T \in \mathbb{R}^M$ gilt für die erste Ableitung

$$\nabla \mathcal{V}^h(\bar{u}) = \nabla \mathbf{c}^T \nabla \mathbf{b}(\mathbf{c}) \mathbf{e}.$$

Dabei ist $\nabla \mathbf{c}$ die Jacobi-Matrix der am Gitter ausgewerteten Näherungen an die Funktionaldeterminanten und hat die Struktur wie in Abbildung 4.3. Man sieht in ihr, dass pro Zeile jeweils sechs Nicht-Null-Einträge vorhanden sind aufgrund der jeweils drei beteiligten Gitterpunkte pro Nebenbedingung. Weiter sind in ihr vier Blöcke zu erkennen (von oben nach unten), die aus den vier betrachteten Dreieckstypen resultieren. Die Matrix $\nabla \mathbf{b}(\mathbf{c})$ hat Diagonalgestalt mit den differenzierten $b(c_i)$'s als Einträgen. Daraus folgt

$$\nabla \mathcal{V}^h(\bar{u}) = \nabla \mathbf{c}^T \mathbf{b}'(\mathbf{c})$$

mit $\mathbf{b}'(\mathbf{c}) = (b'(c_1), \dots, b'(c_M))^T$.

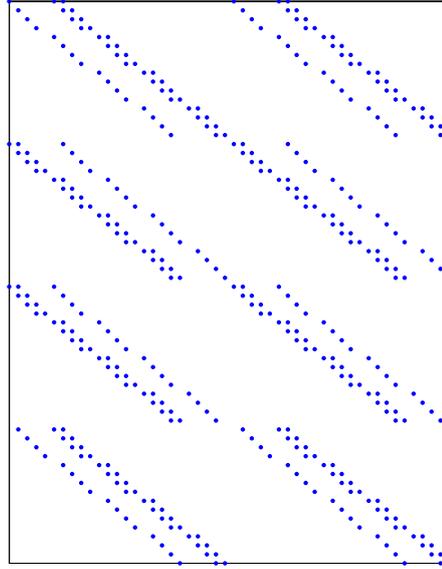


ABBILDUNG 4.3: Struktur der Jacobi-Matrix von \mathbf{c} .

Für die hier verwendete Barriere-Funktion

$$b(x) = \begin{cases} -\alpha \log(x - v_{\min}) - \beta \log(v_{\max} - x) - b_0 & \text{für } x \in (v_{\min}, v_{\max}), \\ \infty & \text{sonst} \end{cases}$$

ist die erste Ableitung durch

$$b'(x) = \begin{cases} -\alpha \frac{1}{(x - v_{\min})} + \beta \frac{1}{(v_{\max} - x)} & \text{für } x \in (v_{\min}, v_{\max}), \\ \infty & \text{sonst} \end{cases}$$

gegeben.

Damit erhalten wir den Gradienten

$$\begin{aligned} \nabla \mathcal{J}_\mu^h(\bar{\mathbf{u}}) &= \nabla \text{SSD}^h(\bar{\mathbf{T}}(\bar{\mathbf{u}})) + \alpha \nabla \mathcal{S}^h(\bar{\mathbf{u}}) + \mu \nabla \mathcal{V}^h(\bar{\mathbf{u}}) \\ &= h_1 h_2 \nabla \bar{\mathbf{T}}(\bar{\mathbf{u}})^T [\bar{\mathbf{T}}(\bar{\mathbf{u}}) - \bar{\mathbf{R}}] + \alpha h_1 h_2 \mathbf{A}^h \bar{\mathbf{u}} + \mu \nabla \mathbf{c}^T \mathbf{b}'(\mathbf{c}), \end{aligned}$$

der diskreten Zielfunktion.

Zweite Ableitung

Zur Bestimmung der Gauss-Newton-Abstiegsrichtung werden im Folgenden die Gauss-Newton-Approximierten zweiten Ableitungen hergeleitet. Ausgehend vom Gradienten $\nabla \text{SSD}^h(\bar{\mathbf{T}}(\bar{\mathbf{u}}))$ und der Approximation aus (4.7) ergibt sich

$$\nabla^2 \text{SSD}^h(\bar{\mathbf{T}}(\bar{\mathbf{u}})) \approx h_1 h_2 \nabla \bar{\mathbf{T}}(\bar{\mathbf{u}})^T \nabla \bar{\mathbf{T}}(\bar{\mathbf{u}}),$$

als Näherung an die Hesse-Matrix für das Distanzmaß. Für den Glättungsterm $\mathcal{S}^h(\bar{\mathbf{u}})$ folgt

$$\nabla^2 \mathcal{S}^h(\bar{\mathbf{u}}) = h_1 h_2 \mathbf{A}^h.$$

Analog zu (4.7) verwenden wir folgende Näherung für die Hesse-Matrix $\nabla^2 \mathcal{V}^h$ von \mathcal{V}^h :

$$\begin{aligned} \nabla^2 \mathcal{V}^h(\bar{\mathbf{u}}) &= \nabla [\nabla \mathbf{c}^T \mathbf{b}'(\mathbf{c})] \\ &= \nabla \left[\sum_{i=1}^M b'(c_i) \nabla c_i \right] \\ &= \sum_{i=1}^M \nabla c_i^T b''(c_i) \nabla c_i + \sum_{i=1}^M b'(c_i) \nabla^2 c_i \\ &= \nabla \mathbf{c}^T \mathbf{b}''(\mathbf{c}) \nabla \mathbf{c} + \sum_{i=1}^M b'(c_i) \nabla^2 c_i \\ &\approx \nabla \mathbf{c}^T \mathbf{b}''(\mathbf{c}) \nabla \mathbf{c}. \end{aligned}$$

Dabei ist $\mathbf{b}''(\mathbf{c}) := \begin{pmatrix} b''(c_1) & & \\ & \ddots & \\ & & b''(c_M) \end{pmatrix}$ eine Diagonalmatrix mit den zweiten Ableitungen

$$b''(x) = \begin{cases} \alpha \frac{1}{(x-v_{\min})^2} + \beta \frac{1}{(v_{\max}-x)^2} & , \text{ wenn } x \in (v_{\min}, v_{\max}), \\ \infty & , \text{ sonst} \end{cases}$$

von b .

Damit erhalten wir die folgende Approximation

$$\nabla^2 \mathcal{J}_\mu^h(\bar{\mathbf{u}}) \approx h_1 h_2 \nabla \bar{\mathbf{T}}(\bar{\mathbf{u}})^T \nabla \bar{\mathbf{T}}(\bar{\mathbf{u}}) + \alpha h_1 h_2 \mathbf{A}^h + \mu \nabla \mathbf{c}^T \mathbf{b}''(\mathbf{c}) \nabla \mathbf{c}$$

an die Hesse-Matrix der diskreten Zielfunktion.

5 Numerische Experimente

Wir betrachten in diesem Abschnitt numerische Experimente, welche die Auswirkungen der Volumenbeschränkung verdeutlichen. Es wurde für alle Tests das SSD-Maß mit diffusivem Regularisierer verwendet. Im Folgenden werden die gestatteten Volumenänderungen v_{\min}, v_{\max} durch den Ausdehnungsfaktor $s > 1$ angegeben. Es gilt $v_{\min} = v_0 s^{-1}$ und $v_{\max} = v_0 s$, wobei v_0 das Volumen eines undeformierten Pixels ist.

5.1 Synthetische Daten

Zunächst betrachten wir Beispiele auf synthetischen Daten. Dabei wurden Template und Referenz-Bild so gewählt, dass mit kleinem Regularisierungsparameter α und ohne Volumenbeschränkung Faltungen im deformierten Gitter auftreten. Für das Referenz-Bild wurde ein ausgefüllter Kreis in der Bildmitte gewählt. Und für das Template-Bild ein kleinerer, komplett innenliegender Ring (vgl. Abbildung 5.1). Zur besseren Präsentation der Ergebnisse haben beide lediglich eine Auflösung von 32×32 Pixeln. Für alle Berechnungen auf den synthetischen Daten wurde $\alpha = 1E - 03$ gewählt.

Registrierung ohne Volumenbeschränkung

Die Registrierung ohne Nebenbedingungen liefert ein nahezu perfektes Ergebnis. Es wurde eine sehr hohe SSD-Reduktion von 99.98 % erzielt. Auch durch eine Evaluation mit

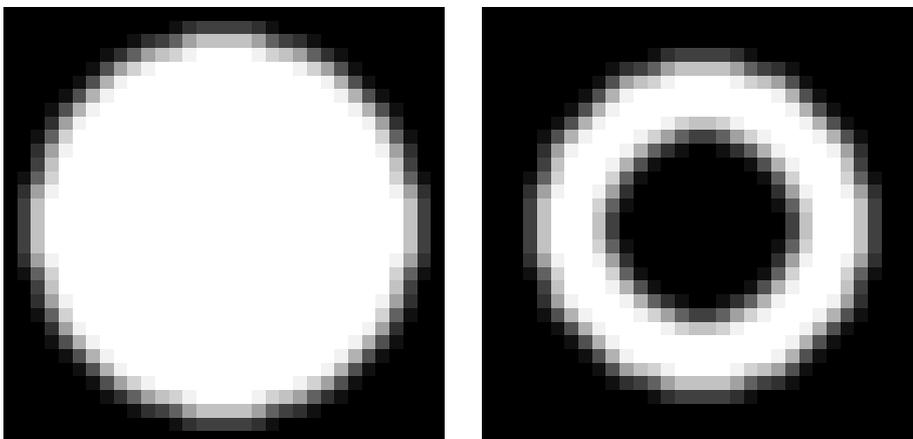


ABBILDUNG 5.1: links: Referenz-Bild, rechts: Template-Bild

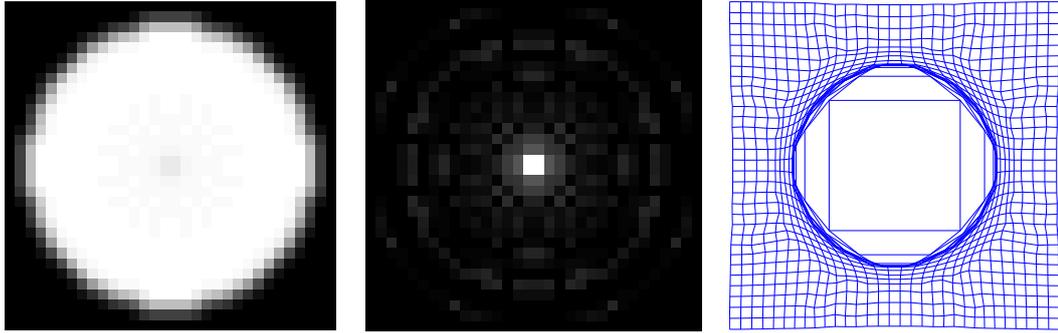


ABBILDUNG 5.2: Keine Volumenbeschränkung.
links: deformiertes Template, Mitte: Differenz-Bild, rechts: deformiertes Gitter.

dem bloßen Auge lassen sich nur sehr kleine Unterschiede ausmachen (vgl. Abbildung 5.1 links und 5.2 links). Bei genauerer Inspektion sind jedoch Falten im Gitter zu erkennen sowie eine extreme Expansion in der Mitte des Bildes (Abbildung 5.2 rechts). Bei gegebenem $v_0 = 9.7656E - 04$ wurden $v_{\min} = -1.3124E - 06$ und $v_{\max} = 0.1166$ gemessen.

Registrierung mit Volumenbeschränkung

Für den volumenbeschränkten Ansatz wurden die Parameter $\mu_0 = 10^{-2}$, $\mu_{\min} = 10^{-4}$, $\tau = 10^{-2}$ und $s = 2, 5, 10$ gewählt. In Tabelle 5.1 sind die erzielten SSD-Reduktionen abhängig vom Ausdehnungsfaktor aufgeführt. Eine direkte Betrachtung des deformierten Bildes (Abbildung 5.3 linke Spalte) und des Differenzbildes (Abbildung 5.3 mittlere Spalte) zeigt, dass für größere s eine gute Näherung an das Referenz-Bild erreicht wurde. Insbesondere sind die Gitter faltenfrei und glatt (vgl. Abbildung 5.3 rechte Spalte). Tabelle 5.2 zeigt die gestatteten und tatsächlich gemessenen Volumina in Abhängigkeit von s an. In Abbildung 5.4 sind die Volumina des Gitters farbig dargestellt. Eine Überschreitung der vorgegeben Grenzen v_{\min}, v_{\max} wurde vermieden.

	$s = 2$	$s = 5$	$s = 10$	ohne Volumenbeschr.
SSD-Reduktion	68.84 %	87.82 %	93.99 %	99.98 %

TABELLE 5.1: Synthetische-Daten. SSD-Reduktionen abhängig vom gewählten s .

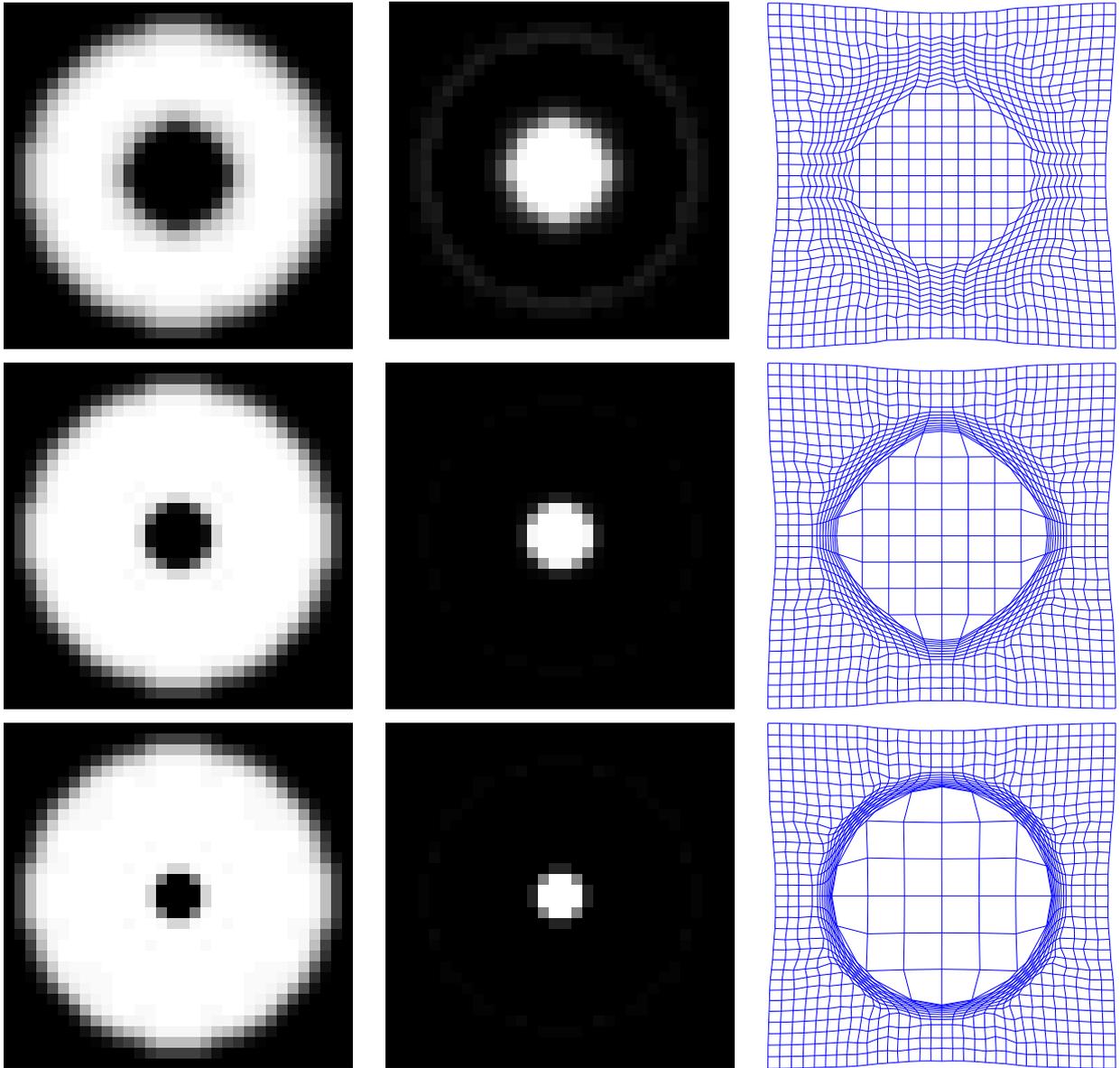


ABBILDUNG 5.3: Erste Zeile: Reg. mit $s = 2$, zweite Zeile: Reg. mit $s = 5$, dritte Zeile: Reg. mit $s = 10$
 links: deformiertes Template, Mitte: Differenz-Bild $|T \circ \varphi - R|$, rechts: deformiertes Gitter.

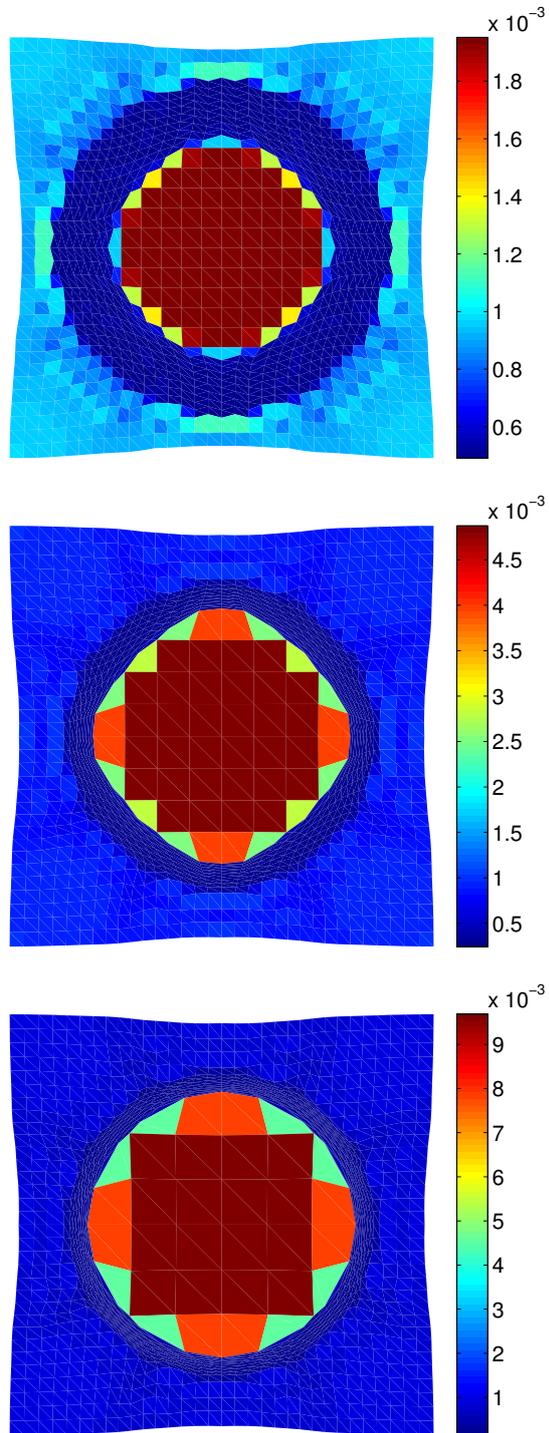


ABBILDUNG 5.4: Plot der Volumina im Gitter. Normiert auf das jeweilige v_{\min}, v_{\max} .
oben: $s = 2$, Mitte: $s = 5$, unten: $s = 10$.

s	v_{\min}		v_{\max}	
	original	gemessen	original	gemessen
2	$4.8828E - 04$	$4.9498E - 04$	0.002	$1.998E - 03$
5	$1.9531E - 04$	$1.9533E - 04$	0.0049	0.0048
10	$9.7656E - 05$	$9.7665E - 05$	$9.7656E - 3$	$9.7650E - 3$

TABELLE 5.2: Synthetische Daten. Maximale/Minimale Zell-Volumina mit $v_0 = 9.7656E - 04$.

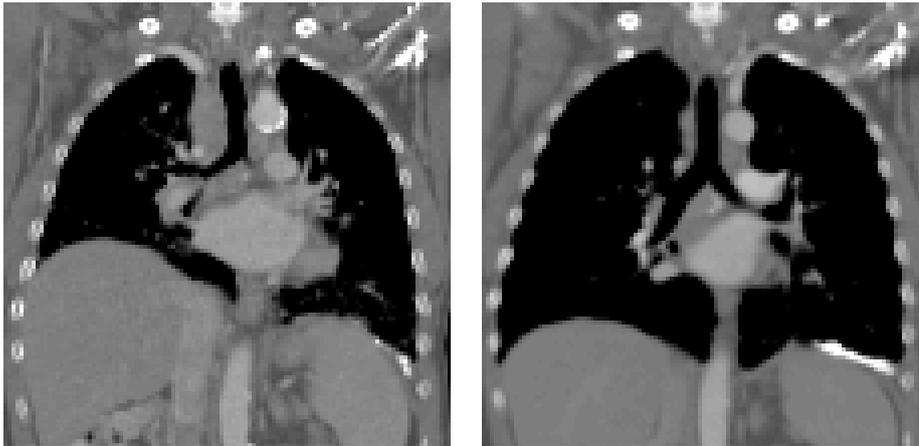


ABBILDUNG 5.5: links: Template-Bild T , rechts: Referenz-Bild R .

5.2 Lungenbilder

Es werden nun zwei Schichten aus CT-Bildern des Thorax (vgl. Abbildung 5.5) zur Evaluation der volumenbeschränkten Registrierung betrachtet. Diese zeigen die Lunge im Zustand maximaler Einatmung und maximaler Ausatmung. Wie auch auf den synthetischen Daten wurde α mit $\alpha = 10^2$ fest gewählt. Beide Bilder haben eine Auflösung von 128×128 Pixeln.

Registrierung ohne Volumenbeschränkung

Ohne Volumenbeschränkung wurde wieder eine sehr gute SSD-Reduktion von 97.89 % erreicht. Wie auch auf den synthetischen Daten ist das Gitter allerdings stark unregulär und weist Falten auf. Insbesondere fällt bei einer direkten Betrachtung der linken unteren Ecke des deformierten Templates auf, dass ein „Bogen“ entstanden ist. Dieser ist das Ergebnis einer besonders starken Faltung des Gitters in dem Bereich (vgl. Abbildung 5.6). Bei gegebenem $v_0 = 9.7656E - 04$ wurden $v_{\min} = -0.0011$ und $v_{\max} = 0.0019$ gemessen.

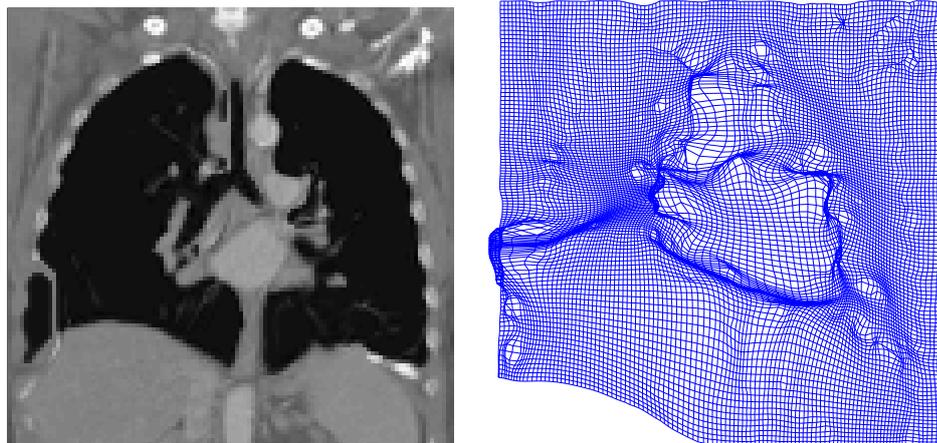


ABBILDUNG 5.6: links: deformiertes Template $T \circ \varphi$, rechts: deformiertes Gitter.

Registrierung mit Volumenbeschränkung

Für den volumenbeschränkten Ansatz wurden $\mu_0 = 10^3$, $\mu_{\min} = 10^{-3}$, und $\tau = 10^{-3}$ gewählt. In Tabelle 5.3 sind die erzielten SSD-Reduktionen gegen den verwendeten Streckungsfaktor s aufgeführt. Für die drei betrachteten s ist das deformierte Gitter (Abbildung 5.7 (rechts)) jeweils glatt und faltenfrei. Und auch an den deformierten Templates lassen sich keine unplausiblen Deformationen feststellen. Tabelle 5.4 zeigt die gestatteten und tatsächlich gemessenen Volumina in Abhängigkeit von s an. Eine Überschreitung der vorgegeben Grenzen v_{\min}, v_{\max} wurde auch hier vermieden. In Abbildung 5.8 sind die Volumina der deformierten Gitter zu sehen.

	$s = 1.1$	$s = 2$	$s = 10$	ohne Volumenbeschr.
SSD-Reduktion	36.78 %	81.35 %	92.63 %	97.89 %

TABELLE 5.3: Lungenbilder. SSD-Reduktionen abhängig vom gewählten s .

s	v_{\min}		v_{\max}	
	original	gemessen	original	gemessen
1.1	$5.5486E - 05$	$5.5487E - 05$	$6.7139E - 05$	$6.7138E - 05$
2	$3.0518E - 05$	$3.1594E - 05$	$1.2207E - 04$	$1.1931E - 04$
10	$6.1035E - 06$	$6.2137E - 06$	$6.1035E - 04$	$3.3580E - 04$

TABELLE 5.4: Lungen-Daten. Maximale/Minimale Zell-Volumina mit $v_0 = 9.7656E - 04$.

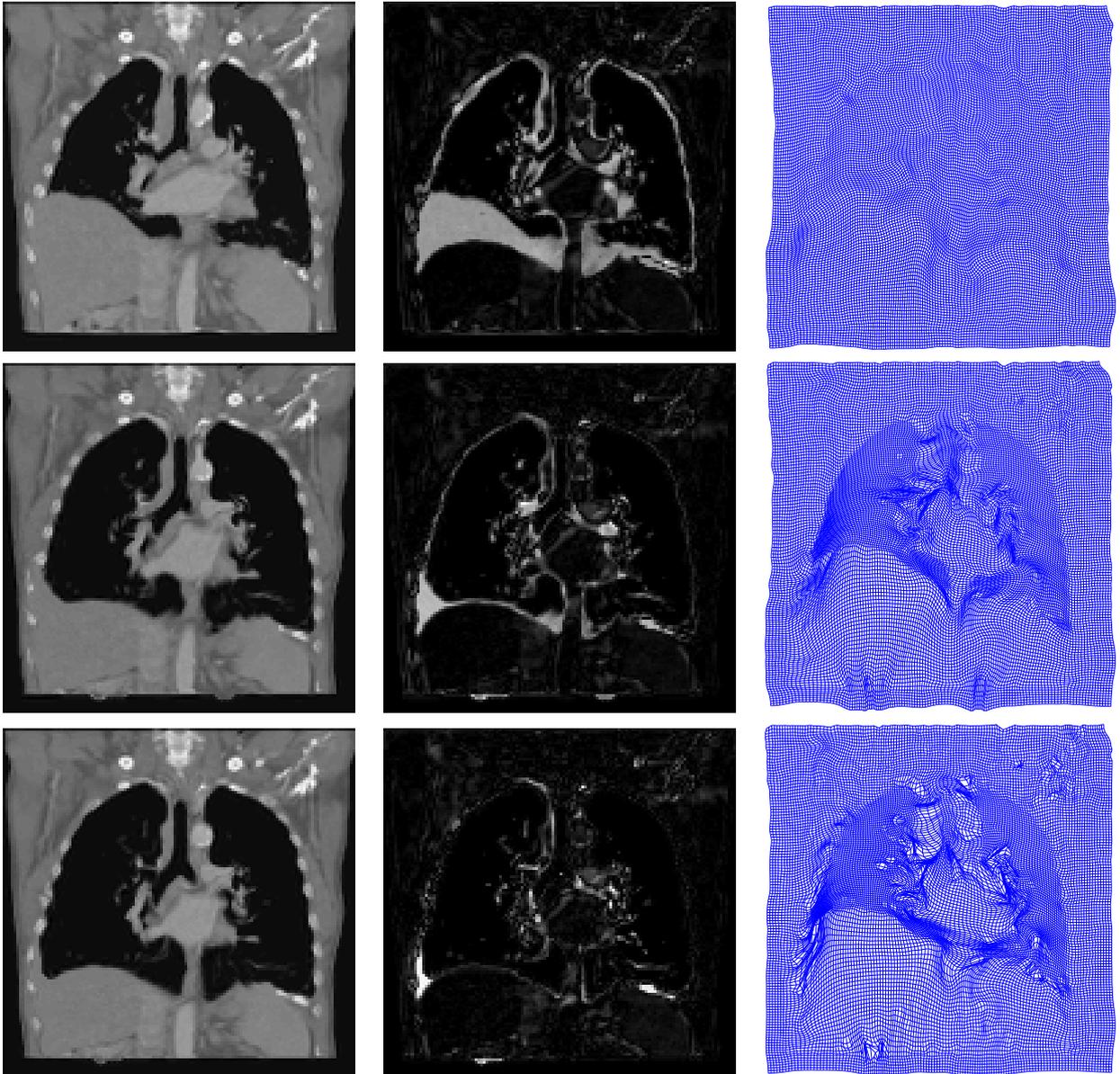


ABBILDUNG 5.7: Erste Zeile: Reg. mit $s = 1.1$, zweite Zeile: Reg. mit $s = 2$, dritte Zeile: Reg. mit $s = 10$
 links: deformiertes Template, Mitte: Differenz-Bild $|T \circ \varphi - R|$, rechts: deformiertes Gitter.

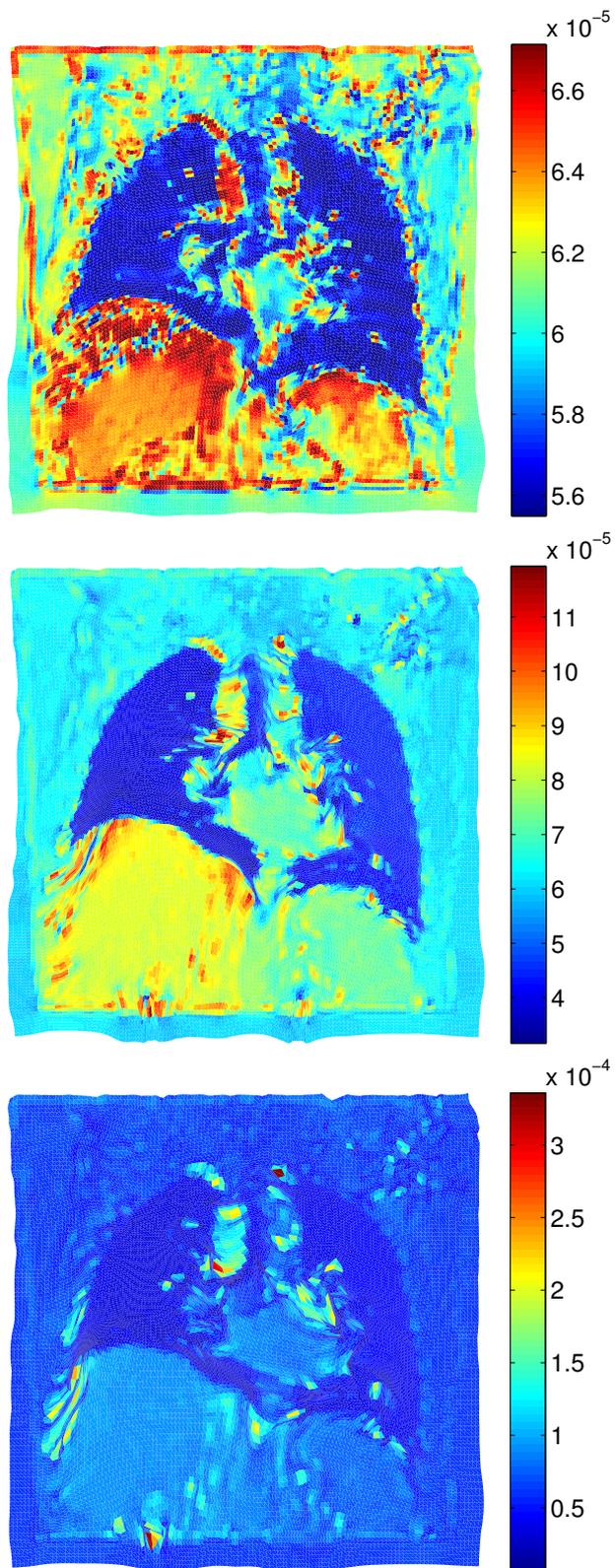


ABBILDUNG 5.8: Plot der Volumina im Gitter. Normiert auf das jeweilige v_{\min}, v_{\max} .
oben: $s = 1.1$, Mitte: $s = 2$, unten: $s = 10$.

5.3 Fazit

Es wurde gezeigt, dass durch Einsatz von Volumenbeschränkung mit der Log-Barrier-Methode glatte und faltenfreie Deformationen realisierbar sind. Diese beiden Eigenschaften sind für eine sinnvolle Registrierung unablässig. Dabei garantiert der Log-Barrier-Ansatz, dass die Deformation volumenbeschränkt und insbesondere injektiv (faltensfrei) ist für eine beliebige Wahl des Regularisierungs-Parameters α .

Literaturverzeichnis

- [1] E. Haber and J. Modersitzki, “Image registration with guaranteed displacement regularity,” *International Journal of Computer Vision*, vol. V71, no. 3, pp. 361–372, 2007.
- [2] O. Forster, *Analysis 3*. Vieweg, 6. Auflage, 2011.
- [3] J. Modersitzki, *FAIR: Flexible Algorithms for Image Registration*. SIAM, 2009.
- [4] B. Fischer and J. Modersitzki, “Ill-posed medicine — an introduction to image registration,” *Inverse Problems*, vol. 24, no. 3, p. 034008 (16pp), 2008.
- [5] O. Forster, *Analysis 2*. Vieweg, 2008.
- [6] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer Series in Operations Research, Springer, 2nd ed., 2006.
- [7] J. Modersitzki, *Numerical Methods for Image Registration*. Numerical Mathematics and Scientific Computation, Oxford University Press, 2003.
- [8] B. Fischer and J. Modersitzki, “Curvature based image registration,” *JMIV*, vol. 18, no. 1, 2003.
- [9] B. Fischer and J. Modersitzki, “Fast diffusion registration,” in *Inverse Problems, Image Analysis, and Medical Imaging* (M. Nashed and O. Scherzer, eds.), vol. 313 of *Contemporary Mathematics*, AMS, 2002.
- [10] R. Bajcsy and C. Broit, “Matching of deformed images,” in *IEEE Proceedings of the Sixth International Conference on Pattern Recognition*, pp. 351–353, 1982.
- [11] G. Fischer, *Lineare Algebra*. Vieweg Studium Grundkurs Mathematik, 10. Auflage