



UNIVERSITÄT ZU LÜBECK  
INSTITUTE OF MATHEMATICS AND  
IMAGE COMPUTING

Bachelorarbeit

# Optimierungsverfahren in der Bildregistrierung

Jannes Quer

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

Abgabe: 23.09.2011



IM FOCUS DAS LEBEN





# Danksagung

Ich möchte mich ganz herzlich bei meinem Betreuer Björn Beuthien für die gute Zusammenarbeit, viele interessante Diskussionen und nützliche Anregungen bedanken.

Weiter möchte ich mich bei allen Mitarbeitern des MIC bedanken für die sehr nette Aufnahme ins Institut und dafür, dass ich auf alle gestellten Fragen eine konstruktive Antwort bekommen habe. Ein besonderer Dank geht dabei an Herrn Prof. Dr. Bernd Fischer.

Weiter danke ich meinen Eltern Margrit Möhle- Quer und Wolfgang Quer für jegliche Unterstützung beim meinem Studium.



# Eidesstattliche Erklärung

Ich versichere, die Bachelorarbeit selbstständig 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

<b>Danksagung</b>	<b>i</b>
<b>Eidesstattliche Erklärung</b>	<b>iii</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Nomenklatur und Formulierung des Problems</b>	<b>3</b>
2.1 Nomenklatur . . . . .	3
2.2 Formulierung des Problems . . . . .	5
<b>3 Lagrange-Funktion</b>	<b>7</b>
3.1 Vorüberlegung . . . . .	7
3.2 Lagrange-Funktion . . . . .	9
3.3 Notwendige Bedingungen . . . . .	10
3.4 Hinreichende Bedingungen . . . . .	10
<b>4 Strafterme</b>	<b>11</b>
4.1 Vorüberlegung . . . . .	11
4.2 Augmented-Lagrange-Funktion . . . . .	12
4.3 Modellalgorithmus . . . . .	14
4.4 Eigenschaften . . . . .	14
<b>5 Lösungsverfahren für unrestringierte Probleme</b>	<b>17</b>
5.1 Bestimmung der Abstiegsrichtung . . . . .	17
5.1.1 Gradienten-Verfahren . . . . .	17
5.1.2 Newton-Algorithmus . . . . .	19
5.1.3 Quasi-Newton-Verfahren . . . . .	19
5.2 Schrittlängenbestimmung . . . . .	22
5.2.1 Armijobedingungen . . . . .	22
5.3 Abbruchkriterien . . . . .	24
5.3.1 Abbruchkriterien für Probleme ohne Nebenbedingungen . . . . .	24
5.3.2 Abbruchkriterien für Probleme mit nichtlinearen Nebenbedingungen . . . . .	25

<b>6 Anwendung</b>	<b>27</b>
6.1 Bildregistrierung . . . . .	27
6.1.1 Transformationen . . . . .	27
6.1.2 Distanzmaße . . . . .	27
6.1.3 Regularisierer . . . . .	28
6.1.4 Volumenerhaltung . . . . .	28
6.1.5 Registrierungsfunktional . . . . .	28
6.2 Numerische Experimente . . . . .	29
6.2.1 Implementierung . . . . .	29
6.2.2 Daten . . . . .	30
6.3 Registrierung mit und ohne Volumenerhaltung . . . . .	31
6.4 Vergleich der Verfahren . . . . .	33
6.5 Parameterwahl . . . . .	35
<b>7 Fazit, Diskussion und Ausblick</b>	<b>39</b>
7.1 Fazit . . . . .	39
7.2 Diskussion . . . . .	39
7.3 Ausblick . . . . .	40
<b>Literaturverzeichnis</b>	<b>41</b>

# 1 Einleitung

Die Bildregistrierung ist ein bedeutender Bestandteil der modernen Bildverarbeitung und gerade im Bereich der Medizin hat die Bildregistrierung immer mehr an Bedeutung gewonnen. Die Registrierung zweier Bilder hat zum Ziel, eine Transformation, die eine möglichst gute Überlagerung der beiden Bilder realisiert, zu finden. Dieses Problem wird als ein Optimierungsproblem formuliert, wobei über die Transformationen optimiert wird. In einigen Fällen werden dabei Nebenbedingungen an das Bild gestellt, wie zum Beispiel, dass das Volumen des Bildes unverändert bleiben soll. Es gibt verschiedene Ansätze solch ein Optimierungsproblem mit Nebenbedingungen zu lösen. Einer davon ist der Augmented-Lagrange-Ansatz. Dieser wird in dieser Arbeit dargestellt. Dabei wird das Optimierungsproblem in mehrere Teilprobleme zerlegt, die mit Algorithmen für unrestringierte Probleme gelöst werden können. Einige dieser Algorithmen werden in dieser Arbeit vorgestellt. Am Ende der Arbeit wird der Augmented-Lagrange-Ansatz mit den unterschiedlichen Lösungsverfahren auf ein Beispiel aus der Bildregistrierung angewendet, um herauszufinden, ob es Unterschiede zwischen den eingesetzten Algorithmen gibt und inwieweit die Ergebnisse abhängig von der Wahl der Startparameter sind. Zum Schluss erfolgt eine Bewertung der Ergebnisse und es wird ein Ausblick gegeben.



## 2 Nomenklatur und Formulierung des Problems

Am Anfang der Arbeit werden Bezeichnungen festgelegt, um dann Optimierungsprobleme im Allgemeinen zu definieren.

### 2.1 Nomenklatur

- Sei  $\mathcal{C}^2(\mathbb{R}^n)$  der Raum der auf  $\mathbb{R}^n$  zweimal stetig differenzierbaren Funktionen.
- Mit  $f : \mathcal{C}^2(\mathbb{R}^n) \rightarrow \mathbb{R}$  wird im Folgenden eine Funktion bezeichnet, die optimiert werden soll. Diese Funktion heißt Zielfunktion.
- Mit  $c_i : \mathcal{C}^2(\mathbb{R}^n) \rightarrow \mathbb{R}$  werden die Nebenbedingungen (Nb.) bezeichnet. Die Nebenbedingungen lassen sich in Gleichheits- und Ungleichheitsnebenbedingungen unterscheiden.
- Alle  $c_i(x)$  für die gilt  $c_i(x) = 0$  werden als Gleichheitsnebenbedingungen bezeichnet. Die Indexmenge  $\mathcal{E}$  enthält alle  $i$  für die  $c_i$  eine Gleichheitsnebenbedingung ist.
- Alle  $c_i(x)$  für die gilt  $c_i(x) \geq 0$  werden als Ungleichheitsnebenbedingungen bezeichnet. Die Indexmenge  $\mathcal{I}$  enthält alle  $i$  für die  $c_i$  eine Ungleichheitsnebenbedingung ist.
- Der Gradient einer Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$

$$\nabla_x f = \left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T$$

ist ein Vektor, der alle partiellen Ableitungen enthält.

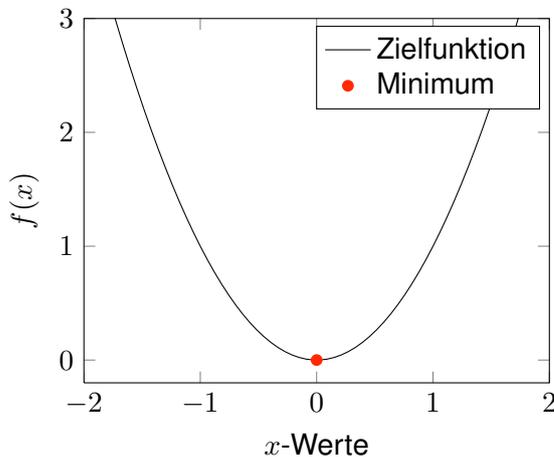
- Die Hessematrix einer Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$

$$\nabla_{xx}(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

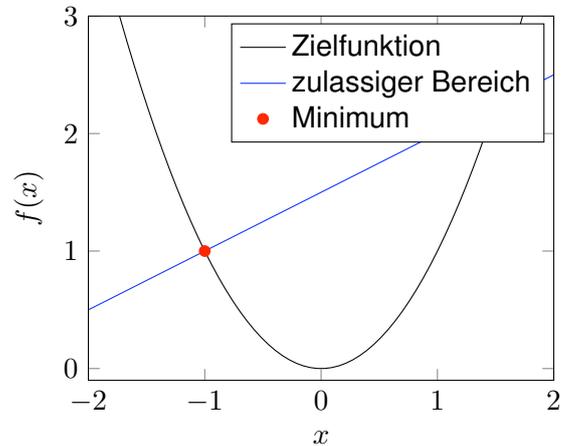
ist eine Matrix, die alle zweiten partiellen Ableitungen beinhaltet.

## 2 Nomenklatur und Formulierung des Problems

- Ein Optimierungsproblem heißt unrestringiert, wenn keine Nebenbedingungen an die Lösung gestellt werden, wenn also  $\mathcal{I} \cup \mathcal{E} = \emptyset$  gilt, ein Beispiel wird in Abbildung 2.1a gezeigt.
- Ein Optimierungsproblem heißt restringiert, wenn Nebenbedingungen an die Lösung gestellt werden, wenn also  $\mathcal{I} \cup \mathcal{E} \neq \emptyset$  gilt, ein Beispiel wird in Abbildung 2.1b gezeigt.



(a) unrestringiertes Problem  $f(x) = x^2$ ,  $x \in (-2, 2)$



(b) restringiertes Problem  $f(x) = x^2$ ,  $c(x) = 1/2 * x + 3/2$ ,  $x \in (-2, 2)$

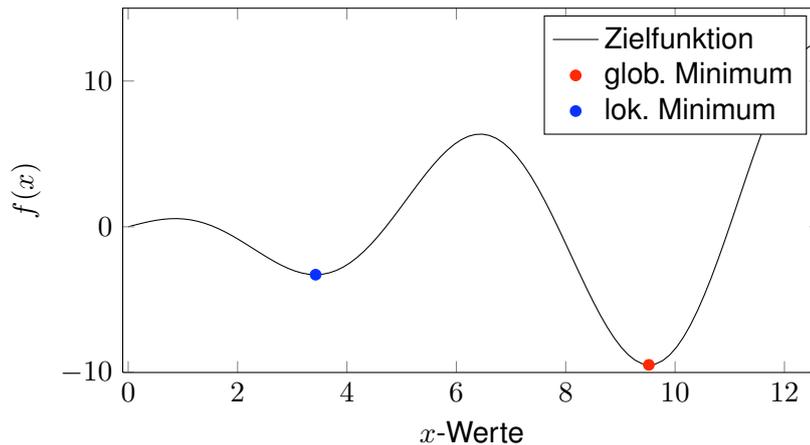
Abbildung 2.1: Beispiele für unrestringierte (a) und restringierte (b) Minimierungsprobleme

- Gilt für einen Funktionswert  $f(x^*) \leq f(x)$ ,  $\forall x \in \mathbb{R}^n$ , so heißt dieser Funktionswert  $f(x^*)$  globales Minimum und der Wert  $x^*$  globaler Minimierer, ein Beispiel wird in Abbildung 2.2 gegeben. Gilt  $f(x^*) < f(x)$ ,  $\forall x \in \mathbb{R}^n$  dann heißt  $f(x^*)$  striktes Minimum und  $x^*$  strikter Minimierer.
- Gilt diese Eigenschaft nur innerhalb einer bestimmten Umgebung  $\mathcal{U}(x^*)$ , d.h.
 
$$f(x^*) \leq f(x),$$

$$\forall x \in \mathcal{U},$$
 so heißt der Funktionswert  $f(x^*)$  lokales Minimum und der Wert  $x^*$  lokales Minimum, ein Beispiel wird in Abbildung 2.2 gegeben. Gilt  $f(x^*) < f(x)$ ,  $\forall x \in \mathcal{U}$  dann heißt  $f(x^*)$  strikt lokales Minimum und  $x^*$  strikt lokaler Minimierer.
- Das notwendige Kriterium für ein Extremum einer unrestringierter Funktionen ist, dass  $\nabla f(x^*) = 0$  gilt, vgl. [Arens et al., 2009].
- Um zu entscheiden, ob ein Extremum ein Minimum, ein Maximum oder ein Sattelpunkt ist, wird die Definitheit der Hessematrix betrachtet, vgl. [Arens et al., 2009].
 
$$x^* \text{ ist Minimum} \Leftrightarrow \nabla_{xx}(f(x^*)) \text{ positiv definit} \Leftrightarrow x^T \nabla_{xx} f(x^*) x > 0, \quad \forall x \in \mathbb{R}^n$$

$$x^* \text{ ist Maximum} \Leftrightarrow \nabla_{xx}(f(x^*)) \text{ negativ definit} \Leftrightarrow x^T \nabla_{xx} f(x^*) x < 0, \quad \forall x \in \mathbb{R}^n$$

$$x^* \text{ ist Sattelpunkt} \Leftrightarrow \nabla_{xx}(f(x^*)) \text{ indefinit} \Leftrightarrow x^T \nabla_{xx} f(x^*) x = 0, \quad \forall x \in \mathbb{R}^n$$

Abbildung 2.2: Minima für  $f(x) = \cos(x)x$ ,  $x \in (0, 4\pi)$ 

## 2.2 Formulierung des Problems

Mit Hilfe der so festgelegten Notation kann nun das Problem korrekt formuliert werden:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{u. d. Nb.} \quad \begin{cases} c_i(x) = 0, & \forall i \in \mathcal{E} \\ c_i(x) \geq 0, & \forall i \in \mathcal{I}. \end{cases}$$

Gesucht ist ein Minimum der Funktion  $f$ , sodass alle Nebenbedingungen erfüllt sind. Die Formulierung eines Maximierungsproblems würde äquivalent erfolgen. Da im weiteren Verlauf dieser Arbeit Minimierungsprobleme im Mittelpunkt stehen, wird auf eine genauere Betrachtung von Maximierungsproblemen verzichtet. Im Fokus dieser Arbeit soll eine Betrachtung von Gleichheitsnebenbedingungen liegen. Es wird das Problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{u. d. Nb.} \quad c_i(x) = 0, \quad \forall i \in \mathcal{E}$$

formuliert. Eine kompakte Formulierung des Problems kann durch  $\mathcal{M} = \{x | c_i(x) = 0, i \in \mathcal{E}\}$

$$\min_{x \in \mathcal{M}} f(x). \tag{2.1}$$

gegeben werden, wobei  $\mathcal{M}$  der sogenannte zulässige Bereich ist.



# 3 Lagrange-Funktion

## 3.1 Vorüberlegung

Das Minimieren von Funktionen ist nicht unproblematisch, da eine Funktion mehrere lokale und globale Minima haben kann.

Um ein Minimum einer unrestringierten Funktion zu bestimmen, wird zunächst das notwendige Kriterium verwendet, um alle Extrema zu bestimmen. Mit Hilfe des hinreichenden Kriteriums kann dann verifiziert werden, um welche Art von Extremum es sich handelt. Ein Problem, das sich bei der Formulierung (2.1) ergibt, ist, dass die Funktion auf einen zulässigen Bereich  $\mathcal{M}$  beschränkt wird und damit die Bedingungen aus der unrestringierten Optimierung nicht direkt übernommen werden können. Bei einem restringierten Problem muss z.B. der Gradient der Zielfunktion nicht zwingend Null sein, siehe Abbildung 3.1. Deshalb sollen im folgenden Abschnitt notwendige und hinreichende Bedingungen so formuliert werden, dass sie genutzt werden können, um ein restringiertes Minimierungsproblem zu lösen. Dies erfolgt exemplarisch für nur eine Nebenbedingung, kann aber auf beliebig viele erweitert werden.

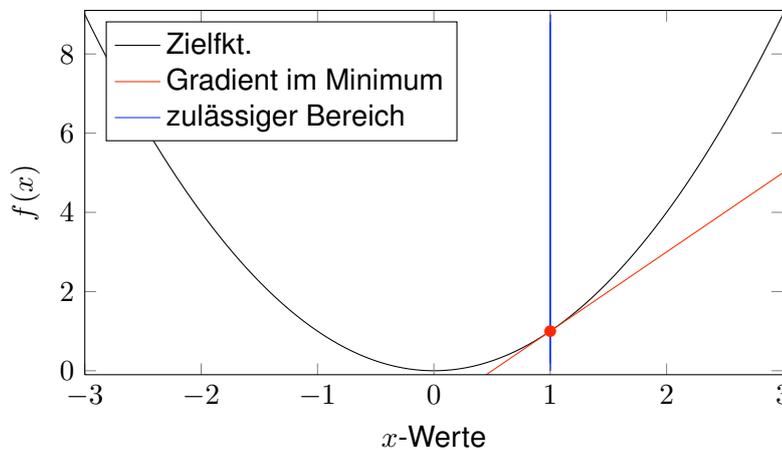


Abbildung 3.1:  $f(x) = x^2$  u.d.Nb.  $c(x) = x$   $c(x) = 1$   $x \in [-3, 3]$

Wird im zulässigen Bereich  $\mathcal{M}$  ein Wert  $f(x)$  betrachtet, der nicht minimal ist, dann ist ein kleiner Schritt  $d \neq 0$  gesucht, für den

$$f(x + d) < f(x) \tag{3.1}$$

### 3 Lagrange-Funktion

gilt. Der Ausdruck  $f(x+d)$  lässt sich mit der Hilfe des Satzes von Taylor, vgl. [Arens et al., 2009], durch

$$f(x+d) \approx f(x) + \nabla_x f(x)^T d$$

approximieren. Eingesetzt in (3.1) ergibt dies  $\nabla_x f(x)^T d < 0$ . Da im zulässigen Bereich  $\mathcal{M}$  gesucht wird, muss für  $(x+d)$  gelten, dass  $c(x+d) \in \mathcal{M}$  ist. Auch dies lässt sich mit dem Satz von Taylor mit

$$0 = c(x+d) \approx c(x) + \nabla_x c(x)^T d = \nabla_x c(x)^T d$$

approximieren. Zusammenfassend lassen sich die Bedingungen

$$\nabla_x c(x)^T d = 0 \quad \text{und} \quad \nabla_x f(x)^T d < 0 \tag{3.2}$$

an den Schritt  $d$  stellen, damit die Zielfunktion im zulässigen Bereich minimiert wird. Der Fall, dass diese beiden Bedingungen sich nicht gleichzeitig erfüllen lassen, tritt nur dann ein, wenn die beiden Gradienten  $\nabla_x f(x)$  und  $\nabla_x c(x)$  linear abhängig sind, also  $\nabla_x f(x) = \lambda \nabla_x c(x)$  gilt. Das bedeutet, dass ein Minimum gefunden worden ist und es keine Richtung im zulässigen Bereich mehr gibt, die die Zielfunktion minimiert, siehe dazu Abbildung 3.2. Die Forderungen aus (3.2) können auf Probleme mit mehreren Nebenbedingungen erweitert werden und es lässt sich zeigen, dass

$$\nabla_x f(x) = \sum_{i \in \mathcal{E}} \lambda_i \nabla_x c_i(x) \tag{3.3}$$

gelten muss vgl. [Arens et al., 2009],[Nocedal und Wright, 2006], wobei die  $\nabla_x c_i(x)$  linear unabhängig sein müssen, siehe Definition 1.

#### Definition 1

Wenn an einem gegebenen Punkt  $x$  für alle  $c_i(x), \forall i \in \mathcal{E}$  gilt, dass die Menge der Gradienten der Nebenbedingungen  $\{\nabla_x c_i(x) | \forall i \in \mathcal{E}\}$  linear unabhängig ist, so ist die Lineare Unabhängigkeitsbedingung - linear independence constraint qualification (LICQ) - erfüllt, vgl. [Nocedal und Wright, 2006].

Ist die LICQ verletzt, so ist es möglich, dass es  $\lambda_i$  gibt, für die  $\sum_{i \in \mathcal{E}} \lambda_i c_i(x) = 0$  ist. Dann wird (3.3) zu  $\nabla_x f(x) = 0$ . Da diese Bedingung im restringierten Fall nicht erfüllt sein muss, kann es sein, dass das Problem nicht gelöst werden kann. Sind die LICQ erfüllt, ist kein Gradient der Nebenbedingung gleich Null.

Eine Möglichkeit, eine Lösung des in (2.1) definierten Problems zu finden, kann die Verwendung einer Hilfsfunktion sein, die den oben gezeigten Zusammenhang (3.3) nutzt. Die Hilfsfunktion ist die Lagrange-Funktion.

## Beispiel

Wie in der Abbildung zu sehen, sind im Minimum  $x = (2.11, 0)$  die Gradienten linear abhängig, die notwendige Bedingung ist also erfüllt.

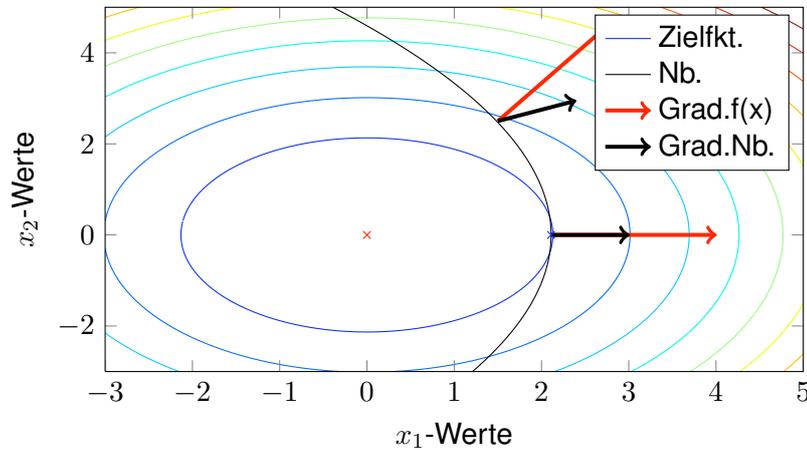


Abbildung 3.2:  $f(x) = x^2 + y^2$  u.d.Nb.  $c(x) = (x + 0.1y^2) - 2.11, x \in [-5, 5]$

## 3.2 Lagrange-Funktion

### Definition 2

Die durch

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x). \quad (3.4)$$

definierte Abbildung  $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{E}|} \rightarrow \mathbb{R}$  heißt Lagrange-Funktion für das in (2.1) definierte restringierte Minimierungsproblem, vgl [Geiger und Kanzow, 2002].

Die  $\lambda_i$  heißen Lagrange'sche Multiplikatoren. Für den Lagrangevektor  $\lambda$ , der alle  $\lambda_i$  enthält, gilt  $\lambda \in \mathbb{R}^{|\mathcal{E}|}$ .

Die Ableitung sowohl nach  $x$  als auch nach  $\lambda$  der Lagrange-Funktion

$$\nabla \mathcal{L}(x, \lambda) = \nabla f(x) - \sum_{i \in \mathcal{E}} \lambda_i \nabla c_i(x).$$

erfolgt komponentenweise.

Mit Hilfe der oben eingeführten Lagrange-Funktion werden nun die notwendigen Bedingungen für ein restringiertes Minimierungsproblem formuliert.

### 3.3 Notwendige Bedingungen

**Satz 1**

Sei  $x^* \in \mathcal{M}$  eine lokale Lösung des Minimierungsproblems (2.1) und die LICQ (1) an der Stelle  $x^*$  erfüllt, dann gibt es Lagrange'sche Multiplikatoren  $\lambda_1^*, \dots, \lambda_{|\mathcal{E}|}^* \in \mathbb{R}$ , sodass die  $n + |\mathcal{E}|$  Gleichungen mit

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0 \tag{3.5}$$

$$\nabla_\lambda \mathcal{L}(x^*, \lambda^*) = 0 \tag{3.6}$$

erfüllt sind, vgl. [Arens et al., 2009].

Ein Beweis dieser Aussage ist in [Nocedal und Wright, 2006] zu finden.

Der Vollständigkeit halber sei hier erwähnt, dass sich diese Bedingungen auch für Probleme mit Ungleichheitsnebenbedingungen erweitern lassen. Sie heißen dann Karush-Kuhn-Tucker (KKT-) Bedingungen, vgl [Nocedal und Wright, 2006].

Da mit der notwendigen Bedingung nicht verifiziert werden kann, ob der Punkt ein Minimum ist, wird eine hinreichende Bedingung benötigt.

### 3.4 Hinreichende Bedingungen

**Satz 2**

Angenommen  $x^*$  sei ein zulässiger Punkt aus  $\mathcal{M}$  und die LICQ erfüllt. Weiter sei  $\lambda^*$  der Lagrangevektor, für den die notwendigen Bedingungen, Satz (1) erfüllt seien. Dann ist  $x^*$  ein strikt lokaler Minimierer von (2.1), wenn

$$w^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w > 0, \quad \forall w \in \mathcal{B}(x^*, \lambda^*), w \neq 0$$

gilt. Dabei ist  $\mathcal{B}(x^*, \lambda^*)$  der so genannte kritische Kegel mit

$$w \in \mathcal{B}(x^*, \lambda^*) \Leftrightarrow \nabla_x c_i(x^*)^T w = 0, \quad \forall i \in \mathcal{E},$$

vgl. [Nocedal und Wright, 2006]

Ein Beweis dieser Aussage ist in [Nocedal und Wright, 2006] zu finden.

## 4 Strafterme

Nachdem die Lagrange-Funktion (3.4) bekannt ist, wird nun eine Herangehensweise vorgestellt, die eine Möglichkeit zur Lösung restringierter Minimierungsprobleme darstellt. Dabei wird das Problem in ein unrestringiertes Problem umformuliert, da es für diese Probleme Lösungsalgorithmen gibt. Ein anschauliches Beispiel am Ende des Kapitels wird den Zusammenhang zwischen den unterschiedlichen Funktionen verdeutlichen.

### 4.1 Vorüberlegung

Ein Ansatz ist, eine Funktion zu formulieren, die sowohl Zielfunktion als auch Nebenbedingungen enthält. Dabei entsteht ein unrestringiertes Minimierungsproblem. Eine Möglichkeit, solch eine Funktion zu formulieren, ist, das Abweichen von den Nebenbedingungen zu bestrafen. Das kann durch einen quadratischen Strafterm ausgedrückt werden.

#### Definition 3

*Diese erweiterte Zielfunktion mit Strafterm ist definiert als*

$$\mathcal{P}(x; \mu) = f(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x).$$

Sie wurde von R. Courant als Erstes vorgestellt, vgl. [Courant, 1943]. Der Parameter  $\mu$  ist der Straf- oder Penalty-Parameter. Es lässt sich zeigen, dass eine Lösung dieses Hilfsproblems auch eine Lösung des Ausgangsproblems (2.1) ist, vgl.[Nocedal und Wright, 2006]. Allerdings kann auch gezeigt werden, dass der iterativ bestimmte Minimierer von  $\mathcal{P}(x, \mu)$  die Nebenbedingungen nicht genau erfüllt. Es kommt zu einer systematischen Störung von

$$c_i(x_k) \approx \frac{-\lambda_i^*}{\mu_k}, \quad \forall i \in \mathcal{E}. \quad (4.1)$$

Der Beweis dieser Aussage ist bei [Nocedal und Wright, 2006] zu finden. Diese Störung soll natürlich vermieden werden, was gleichzeitig bedeutet, dass die iterativen Lösungen die Nebenbedingungen besser erfüllen. Dies leistet die Augmented-Lagrange-Funktion.

## 4.2 Augmented-Lagrange-Funktion

### Definition 4

Die Augmented-Lagrange-Funktion ist definiert als

$$\mathcal{L}_{\mathcal{A}}(x, \lambda; \mu) = f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x). \quad (4.2)$$

Sie ist eine Erweiterung der Lagrange-Funktion mit einem quadratischen Strafterm, wie aus dem Penalty-Ansatz bekannt, vgl [Nocedal und Wright, 2006], [Geiger und Kanzow, 2002].

Im Weiteren wird ein Ansatz beschrieben, der das Minimierungsproblem (2.1) mit Hilfe der Augmented-Lagrange-Funktion löst. Dabei wird der Parameter  $\mu$  bei der  $k$ ten-Iteration auf einen Wert  $\mu_k > 0$  festgelegt, der Lagrangevektor  $\lambda$  an der aktuellen Schätzung  $\lambda^k$  fixiert, um dann den Wert  $x_k$  zu minimieren.

Da durch die Umformulierung nun ein unrestringiertes Problem gelöst wird, gilt für den Gradienten an der Stelle des approximierten Minimierers  $x_k^*$

$$\begin{aligned} 0 &\approx \nabla_x \mathcal{L}_{\mathcal{A}}(x_k, \lambda^k; \mu_k) = \nabla_x f(x_k) - \sum_{i \in \mathcal{E}} \lambda_i^k \nabla_x c_i(x_k) + \mu \sum_{i \in \mathcal{E}} c_i(x_k) \nabla_x c_i(x_k) \\ &= \nabla_x f(x_k) - \sum_{i \in \mathcal{E}} [\lambda_i^k - \mu_k c_i(x_k)] \nabla_x c_i(x_k). \end{aligned} \quad (4.3)$$

Für das restringierte Problem sind an der Stelle  $x_k$  auch die notwendigen Bedingungen erfüllt. Deshalb kann geschlossen werden, dass  $\lambda_i^* \approx \lambda_i^k - \mu_k c_i(x_k)$  gelten muss. Damit ergibt sich für den Lagrangevektor aus (4.3) diese Schätzungsregel

$$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k). \quad (4.4)$$

### Satz 3

Eine Lösung der Augmented-Lagrange-Funktion ist auch eine Lösung des restringierten Problems (2.1).

Der obige Satz (3) soll im Folgenden bewiesen werden.

### Beweis 1

Angenommen  $x^*$  sei ein lokales Minimum des Problems (2.1), an dem die LICQ und auch die hinreichenden Bedingungen, in Abschnitt 3.3 konkretisiert, für  $\lambda = \lambda^*$  erfüllt sind. Dann gibt es einen Schwellwert  $\hat{\mu}$ , sodass für alle  $\mu > \hat{\mu}$ ,  $x^*$  ein strikt lokaler Minimierer von  $\mathcal{L}_{\mathcal{A}}(x, \lambda^*; \mu)$  ist.

Zuerst wird gezeigt, dass das notwendige Kriterium  $\nabla_x \mathcal{L}_{\mathcal{A}}(x^*, \lambda^*; \mu) = 0$  erfüllt ist.

Da  $x^*$  ein lokales Minimum von (2.1), an dem die LICQ erfüllt sind, ist, werden die notwendigen Bedingungen genutzt um zu zeigen, dass  $\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$  und  $c_i(x^*) = 0, \forall i \in \mathcal{E}$ . Es gilt

$$\begin{aligned} \nabla_x \mathcal{L}_{\mathcal{A}}(x^*, \lambda^*; \mu) &= \nabla f(x^*) - \sum_{i \in \mathcal{E}} [\lambda_i^* - \mu c(x^*)] \nabla_x c(x^*) \\ &= \nabla_x f(x^*) - \sum_{i \in \mathcal{E}} \lambda_i^* \nabla_x c(x^*) = \nabla_x \mathcal{L}(x^*, \lambda^*) = 0 \end{aligned}$$

unabhängig von  $\mu$ . Es bleibt zu zeigen, dass die Hessematrix positiv definit ist. Diese kann geschrieben werden als

$$\nabla_{xx} \mathcal{L}(x^*, \lambda^*; \mu) = \nabla_{xx} \mathcal{L}(x^*, \lambda^*) + \mu \nabla_x c(x^*)^T \nabla_x c(x^*),$$

wobei  $\nabla_x c(x^*)^T \nabla_x c(x^*) = \|\nabla_x c(x^*)\|_2^2$  ist. Angenommen, die Hessematrix sei nicht positiv definit, dann kann für jede positive ganze Zahl  $k \geq 1$  ein Vektor  $v_k$  mit  $\|v_k\|_2 = 1$  gewählt werden, sodass

$$0 \geq v_k^T \nabla_{xx} \mathcal{L}_{\mathcal{A}}(x^*, \lambda^*; k) v_k = v_k^T \nabla_{xx} \mathcal{L}(x^*, \lambda^*) v_k + k \|\nabla_x c(x^*) v_k\|_2^2 \quad (4.5)$$

gilt. Im Limes für  $k \rightarrow \infty$  betrachtet, gilt

$$\|\nabla_x c(x^*) v_k\|_2^2 \leq -(1/k) v_k^T \nabla_{xx} \mathcal{L}(x^*, \lambda^*) v_k \rightarrow 0$$

Da die Elemente der Folge  $v_k$  auf der Einheitssphäre liegen, hat die Folge einen Grenzwert  $v$ . Die obige Betrachtung des Limes impliziert, dass  $Av = 0$  gilt. Durch eine Umformulierung von (4.5) kann gezeigt werden, dass

$$v_k^T \nabla_{xx} \mathcal{L}_{\mathcal{A}}(x^*, \lambda^*) v_k \leq -k \|\nabla_x c(x^*) v_k\|_2^2 \leq 0$$

gilt. Durch eine Betrachtung dieses Ausdruckes im Limes für  $k \rightarrow \infty$  ergibt sich, dass  $v^T \nabla_{xx} \mathcal{L}(x^*, \lambda^*) v \leq 0$  gilt. Diese Ungleichung verletzt die hinreichende Bedingung (Abschnitt 3.3), die besagt, dass  $v^T \mathcal{L}(x^*, \lambda^*) v > 0$  für alle von Null verschiedenen Vektoren  $v$  für die  $\nabla_x c(x^*)^T v = 0$  gilt. Damit ist die Annahme, die Hessematrix sei nicht positiv definit, zum Widerspruch gebracht, vgl. [Nocedal und Wright, 2006].

Mit der Augmented-Lagrange-Funktion (4.2) und der Erneuerungsregel (4.4) kann nun ein Modellalgorithmus erstellt werden, mit dessen Hilfe das Problem (2.1) gelöst werden kann.

### 4.3 Modellalgorithmus

Eine kurze Beschreibung des Modellalgorithmus für das mit Hilfe der Augmented-Lagrange-Funktion umformulierte Problem (2.1) kann wie folgt gegeben werden.

Wähle:  $\mu_0 > 0, \tau > 0$   
 Startpunkte  $x_0, \lambda_0$   
**while** Abbruchkriterien  $\neg$  OK **do**  
   Finde einen Minimierer  $x_k$  für  $\mathcal{L}_{\mathcal{A}}(x_k, \lambda^k; \mu_k)$   
   Stopp wenn  $\|\nabla_x \mathcal{L}_{\mathcal{A}}(x_k, \lambda^k; \mu_k)\| \leq \tau$   
   **if** Abbruchkriterien für (2.1) == OK **then**  
     Stopp mit Lösung  $x_k$   
   **end if**  
   Update Lagrange'sche Multiplikatoren mit (4.4)  
   Setze neuen Penalty-Parameter  $\mu_{k+1} \geq \mu_k$   
**end while**

In der inneren Schleife wird für fixiertes  $\lambda$  und  $\mu$  das unrestringierte Minimierungsproblem gelöst. In der äußeren Schleife wird dann getestet, ob ein Minimum gefunden wurde. Ist dies nicht der Fall, werden  $\lambda$  und  $\mu$  aktualisiert und die innere Schleife nochmal durchlaufen. Die einzelnen Bestandteile des Algorithmus werden in den folgenden Kapiteln näher erläutert.

### 4.4 Eigenschaften

Es lassen sich weitere Eigenschaften für die Augmented-Lagrange-Funktion zeigen.

#### Satz 4

Sei  $x^*$  eine lokale Lösung von (2.1), ab der die LICQ-Bedingungen (1) gelten und für  $\lambda = \lambda^*$  gilt  $\|\lambda^k - \lambda^*\| \leq \mu_k \delta, \quad \forall \mu_k \geq \hat{\mu}, \quad \delta \in \mathbb{R}^+,$  dann gibt es einen Wert  $\hat{\mu}$  für alle  $\mu > \hat{\mu}$ , sodass  $x^*$  ein strikt lokaler Minimierer von  $\mathcal{L}_{\mathcal{A}}(x, \lambda^*; \mu)$  ist. Für alle  $\lambda^k$  und  $\mu_k$ , die diesen Voraussetzungen genügen, können drei weitere Aussagen bewiesen werden.

#### 1. Das Problem

$$\min_x \mathcal{L}_{\mathcal{A}}(x, \lambda^k; \mu_k) \quad \text{u.d.N.} \quad \|x - x^*\| \leq \epsilon, \quad \epsilon \in \mathbb{R}^+$$

hat genau eine Lösung  $x_k$  und es gibt keine weitere.

#### 2. Für alle $\lambda_k$ gilt

$$\|\lambda^{k+1} - \lambda^*\| \leq M \|\lambda^k - \lambda^*\| / \mu_k, \quad M \in \mathbb{R}^+,$$

wobei  $\lambda^{k+1}$  der Formel (4.4) entspricht. Der Lagrangevektor konvergiert also linear gegen die exakte Lösung.

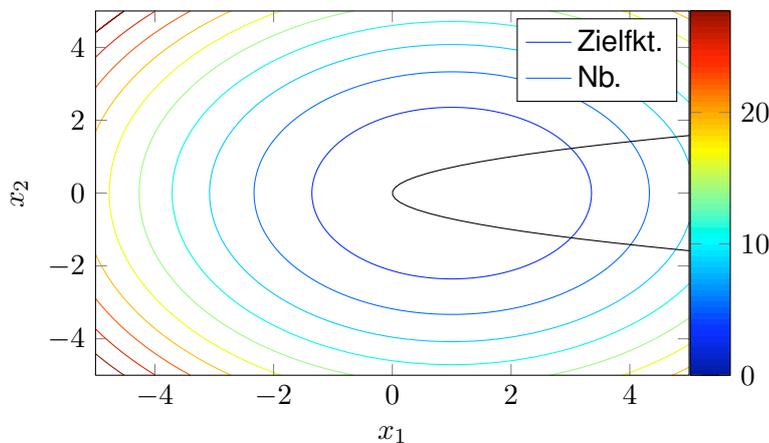
3. Die Hessematrix  $\nabla_{xx}\mathcal{L}_{\mathcal{A}}(x_k, \lambda^k; \mu_k)$  ist positiv definit, d.h. das hinreichende Kriterium an der approximierten Lösung  $x_k$  ist erfüllt. Weiter ist die LICQ erfüllt, was eine Voraussetzung für die notwendigen Kriterien (1) darstellt, vgl.[Nocedal und Wright, 2006].

Ein Beweis dieser Aussagen kann in [Nocedal und Wright, 2006] gefunden werden. Diese Eigenschaften sind nützlich, da jetzt bekannt ist, dass

1. die Lösung einzigartig ist,
2.  $\lambda_k$  gegen das exakte  $\lambda^*$  konvergiert und
3. die Lösung ein Minimum ist.

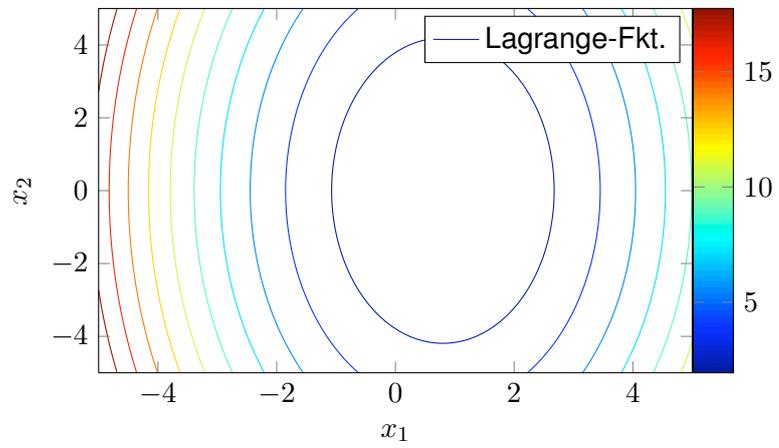
## Beispiel

Im Folgenden soll ein weiteres Beispiel veranschaulichen, wie die Funktion und die Nebenbedingungen, die Lagrange-Funktion und die Augmented-Lagrange-Funktion zusammenhängen. Dafür wird die Funktion  $f(x) = (x_1 - 1)^2 + x_2^2$  mit der Nebenbedingung  $c(x) = (x_1 - 1)^2 + x_2^2$  betrachtet, deren Minimum bei  $(0, 1)$  liegt.

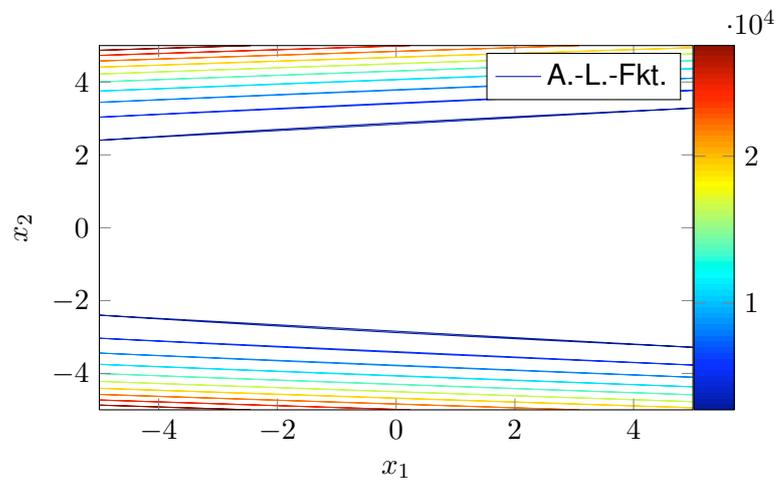


(a) Funktion mit Nebenbedingungen

#### 4 Strafterme



(b) Lagrange-Funktion des Problems mit  $\lambda = 0.2$



(c) Augmented-Lagrange-Funktion mit  $\lambda = 0.2, \mu = 10$

Abbildung 4.1: Die Abbildungen zeigen die Höhenlinien der unterschiedlichen Funktionen.

# 5 Lösungsverfahren für unrestringierte Probleme

Im folgenden Kapitel sollen die Algorithmen näher beschrieben werden, die zum Lösen eines unrestringierten Minimierungsproblems genutzt werden. Diese Algorithmen können für den umformulierten Ansatz des Problems (2.1) mit der Augmented-Lagrange-Funktion herangezogen werden. Alle vorgestellten Algorithmen sind iterative Abstiegsverfahren, die sich in der Bestimmung der Suchrichtung unterscheiden. Der allgemeine Aufbau lässt sich durch

```
while Abbruchkriterien  $\neg$  OK do  
     $s_k$  = bestimme Suchrichtung  
    bestimme Schrittlänge  $t_k$   
     $x_{k+1}$  = addiere auf  
end while
```

beschreiben.

Es gibt diverse Methoden für die Bestimmung einer Abstiegsrichtung, für die die  $f(x + ts) < f(x)$  gilt. Die meistgenutzten Methoden werden hier vorgestellt. Die meisten vorgestellten Verfahren werden mit einer Schrittlängenbestimmung durchgeführt, um einen effizienten Abstieg zu gewährleisten, siehe Abbildung 5.1. Zwei werden näher erläutert. Am Ende des Kapitels werden die Abbruchbedingungen konkretisiert.

## 5.1 Bestimmung der Abstiegsrichtung

### 5.1.1 Gradienten-Verfahren

Das Gradientenverfahren ist das bekannteste aller Verfahren zum Lösen unrestringierter Minimierungsprobleme und ist sehr leicht zu implementieren, vgl.[Arens et al., 2009]. Die Grundidee des Algorithmus ist, dass die Richtung des steilsten Abstiegs auf den aktuellen Wert  $x_k$  aufaddiert wird. Die Richtung des steilsten Abstiegs ist die Richtung des negativen Gradienten, was sich leicht herleiten lässt. Zunächst lässt sich sehen, dass der Gradient in

## 5 Lösungsverfahren für unrestringierte Probleme

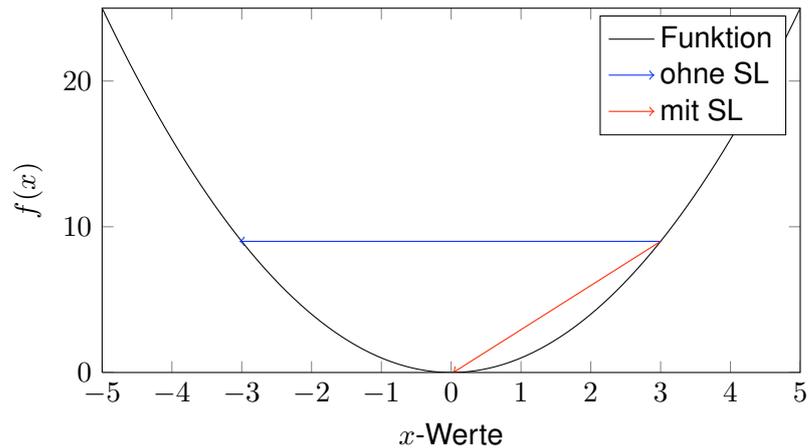


Abbildung 5.1: Minimierung mit und ohne Schrittweitenbestimmung

Richtung des steilsten Anstiegs weist. Daraus kann dann geschlossen werden, dass die Richtung des steilsten Abstiegs der negative Gradient ist.

Die Richtungsableitung einer stetig differenzierbaren Funktion  $f$  in Richtung  $h$ ,  $\|h\|_2 = 1$ , die sich durch  $\partial_h f(x) = \langle \nabla_x f(x), h \rangle_2$  berechnet, muss dafür maximal werden. Die Richtungsableitung kann mit der Cauchy-Schwarzschen-Ungleichung abgeschätzt werden,

$$\langle \nabla_x f(x), h \rangle_2 \leq \|h\|_2 \|\nabla_x f(x)\|_2.$$

Maximal wird dieser Ausdruck genau dann, wenn Gleichheit gilt. Dies ist genau dann der Fall, wenn  $d = \frac{\nabla_x f(x)}{\|\nabla_x f(x)\|}$ . Damit ist ersichtlich, dass der Gradient in Richtung des steilsten Anstiegs weist, beziehungsweise der negative Gradient in Richtung des steilsten Abstiegs zeigt, vgl.[Arens et al., 2009].

Der Algorithmus des Gradientenverfahrens lässt sich wie folgt formulieren:

**while** Abbruchkriterien  $\neg$  OK **do**

$$s_k = (-\nabla_x f(x_k))$$

bestimme Schrittweite  $t_k$

$$x_{k+1} = x_k + t_k s_k$$

**end while**

Der Algorithmus muss mit einer Schrittweitenbestimmung durchgeführt werden, um einen möglichst effizienten Abstieg zu gewährleisten. Das Gradientenverfahren hat ein nur lineares Konvergenzverhalten, da die Schritte, je dichter  $x_k$  am Minimum ist, immer kleiner werden, vgl. [Nocedal und Wright, 2006].

### 5.1.2 Newton-Algorithmus

Ein Verfahren mit besserem Konvergenzverhalten als das Gradientenverfahren ist das Newtonverfahren, vgl. [Nocedal und Wright, 2006]. Mit diesem kann bei geeigneter Wahl eines Startwertes lokal quadratische Konvergenz erzielt werden. Die Suchrichtung lässt sich für dieses Verfahren über eine Taylorapproximation und die notwendige Bedingung herleiten. Die zweite Taylorapproximation für  $f(x_k + s_k)$  ist durch

$$f(x_k + s_k) \approx f(x_k) + \nabla_x f(x_k)^T s_k + \frac{1}{2} s_k^T \nabla_{xx} f(x_k) s_k := m_k(s)$$

gegeben. Es ist nun eine Richtung  $s_k$  gesucht, die die Funktion  $m_k(s)$  minimiert. Wird  $m_k(s)$  abgeleitet und gleich Null gesetzt, ergibt sich

$$-(\nabla_{xx} f(x_k)) s_k = \nabla_x f(x_k). \quad (5.1)$$

Die so bestimmte Abstiegsrichtung heißt Newton-Richtung. Der Newton-Algorithmus lässt sich durch

```

while Abbruchkriterien  $\neg$  OK do
   $-(\nabla_{xx} f(x_k)) s_k = \nabla_x f(x_k)$ 
   $x_{k+1} = x_k + t_k s_k$ 
end while

```

beschreiben.

Beim Newton-Verfahren wird die Schrittlänge  $t = 1$  gesetzt. Es kann gezeigt werden, dass wenn der Startwert nah am exakten Minimum liegt, die durch den Algorithmus approximier- te Lösung gegen die exakte Lösung strebt, vgl. [Arens et al., 2009]. Nachteile des Newton- Verfahrens sind, dass die Abstiegsrichtung über die zweiten Ableitungen bestimmt wird. Das kann sehr aufwendig sein. Außerdem ist die aus (5.1) bestimmte Richtung nur dann eine Abstiegsrichtung, wenn die Hessematrix der Funktion positiv definit ist.

### 5.1.3 Quasi-Newton-Verfahren

Das Quasi-Newton-Verfahren ist ein Verfahren, das dem Newton-Verfahren ähnelt, vgl. [Nocedal und Wright, 2006]. Der Unterschied ist, dass bei diesem Verfahren die Hesse- matrix approximiert wird und dadurch die zweiten Ableitungen nicht exakt bestimmt werden müssen. Die Approximation der Hessematrix kann durch unterschiedliche Ansätze erfolgen, einige davon werden vorgestellt. Der Modellalgorithmus ist durch

```

while Abbruchkriterien  $\neg$  OK do
  Approximiere Hessematrix  $\mathcal{A}_k$ 
  bestimme Schrittlänge  $t_k$ 
   $-\mathcal{A}_k s_k = \nabla_x f(x_k)$ 
   $x_{k+1} = x_k + t_k s_k$ 
end while
    
```

gegeben.

Die approximierte Hessematrix muss auch wieder positiv definit sein, damit  $s_k$  eine Abstiegsrichtung ist. Es gibt Quasi-Newton-Verfahren, die eine feste Approximation der Hessematrix verwenden, die nicht aktualisiert wird. Damit ist allerdings immer ein Informationsverlust für die damit bestimmte Suchrichtung verbunden, da die aktuelle Krümmung, die in der exakten zweiten Ableitung bestimmt wird, nicht mit in die Bestimmung der Suchrichtung mit einbezogen wird. Zwei Quasi-Newton-Verfahren, die eine iterative Approximation der Hessematrix nutzen, werden nun vorgestellt.

### BFGS-Verfahren

Das Verfahren von Brodyen-Fletcher-Goldfarb-Shanno (BFGS) ist ein Quasi-Newton-Verfahren, das die Hessematrix in jedem Schritt iterativ approximiert, vgl. [Gill et al., 1981], [Nocedal und Wright, 2006]. Auf die exakte Herleitung soll an dieser Stelle verzichtet werden, kann aber in [Nocedal und Wright, 2006] nachgelesen werden.

```

while Abbruchkriterien  $\neg$  OK do
   $-\mathcal{A}_k s_k = \nabla_x f(x_k)$ 
  bestimme Schrittlänge  $t_k$ 
   $x_{k+1} = x_k + t_k s_k$ 
   $p_k = x_{k+1} - x_k$ 
   $y_k = \nabla_x f(x_{k+1}) - \nabla_x f(x_k)$ 
   $\mathcal{A}_{k+1} = \mathcal{A}_k - \frac{\mathcal{A}_k p_k p_k^T \mathcal{A}_k}{p_k^T \mathcal{A}_k p_k} + \frac{y_k y_k^T}{y_k^T p_k}$ 
end while
    
```

Als Startwert für  $\mathcal{A}_k$  kann ein positives Vielfaches der Einheitsmatrix genommen werden. Es ist möglich zu zeigen, dass ein BFGS-Update, wenn  $p_k^T y_k > 0$  gilt, eine positiv definite Approximation der Hessematrix liefert, vgl. [Nocedal und Wright, 2006]. Damit ist  $s_k$  eine Abstiegsrichtung. Ist dies nicht der Fall, kann wieder eine positives Vielfaches der Einheitsmatrix genommen werden, um die nächste Suchrichtung zu bestimmen. Die Updateformel für  $\mathcal{A}_{k+1}$  kann auch so umformuliert werden, dass die inverse Hessematrix berechnet wird. Das ist insofern von Vorteil, dass das Lösen des linearen Gleichungssystems  $\mathcal{A}_k s_k = \nabla_x f(x_k)$  zu einer

Matrix-Vektor-Multiplikation wird. Dieses Verfahren heißt dann inverses BFGS-Verfahren. Ein Nachteil des Verfahrens ist der hohe Rechenaufwand, der durch die Berechnung der approximierten Hessematrix zustande kommt.

### Gauß-Newton-Verfahren

Das Gauß-Newton-Verfahren approximiert die Hessematrix auf eine andere Art und Weise. Die grundlegende Idee des Verfahrens ist die Minimierung des Residuums durch die Methode der kleinsten Quadrate. Dazu wird eine Zielfunktion  $f(x) = \frac{1}{2}\|r(x)\|^2$  mit Residuum  $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$  betrachtet. Der Gradient kann mit  $\nabla_x f(x) = J(x)^T r(x)$  angegeben werden, wobei  $J(x)$  die Jacobi-Matrix von  $r(x)$  ist, die definiert ist als

$$J(x) = \begin{bmatrix} \nabla_x r_1(x)^T \\ \nabla_x r_2(x)^T \\ \vdots \\ \nabla_x r_m(x)^T \end{bmatrix}.$$

Für die Hessematrix der Zielfunktion gilt dann

$$\nabla_{xx} f(x) = J(x)^T J(x) + \sum_{i=1}^m r_i(x) \nabla_{xx} r_i(x).$$

Der erste Teil des Terms kann berechnet werden, ohne dass eine zweite Ableitung gebildet werden muss. Darüber hinaus kann gezeigt werden, dass nahe des Minimums sowohl das Residuum als auch die Hessematrix klein ist und der hintere Teil deshalb vernachlässigt werden kann. Es macht deshalb Sinn, die Hessematrix über  $J(x)^T J(x)$  zu approximieren. Ein Modellalgorithmus kann mit

**while** Abbruchkriterien  $\neg$  OK **do**

$$-J(x)^T J(x) s_k = \nabla_x f(x_k)$$

bestimme Schrittlänge  $t_k$

$$x_{k+1} = x_k + t_k s_k$$

**end while**

beschrieben werden. Das Gauß-Newton-Verfahren hat ein ähnliches Konvergenzverhalten, wie das Newton-Verfahren, wenn das Residuum klein ist und somit  $J(x)^T J(x)$  eine gute Approximation der exakten Hessematrix ist. Es kann weiter gezeigt werden, dass wenn die Jacobimatrix  $J(x_k)$  vollen Rang hat und der Gradient  $\nabla_x f(x_k)$  von Null verschieden ist, dann ist die Suchrichtung  $s_k$  immer eine Abstiegsrichtung, vgl. [Nocedal und Wright, 2006].

## Zusammenfassung

In der nachfolgenden Tabelle 1 sind die Vor- und Nachteile der hier vorgestellten Verfahren noch einmal zusammenfassend dargestellt.

**Tabelle 1**

Verfahren	Vorteil	Nachteil	Konvergenz
Gradienten	leicht zu berechnen	ineffizient	linear
Newton	effiziente Richtung	2. Ableitung nötig	quadratisch
Quasi-Newton	keine 2. Ableitungen nötig	Informationsverlust	linear
BFGS-	keine 2. Ableitungen nötig	hoher Rechenaufwand	superlinear
Gauß-Newton	gute Approx. von $\nabla_{xx}f(x)$	spez. Struktur von $f(x)$ nötig	(super-)linear

## 5.2 Schrittweitenbestimmung

Da durch die oben vorgestellten Lösungsverfahren eine Abstiegsrichtung angegeben wird, die in jeder Iteration in der Länge variiert, bleibt die Notwendigkeit die Schrittweitenbestimmung zu ergänzen. Die Schrittweitenbestimmung gibt an, wie weit in die entsprechende Richtung gegangen wird, um zu garantieren, dass  $f(x + ts) < f(x)$  gilt. Die exakte Schrittweite könnte bestimmt werden, indem die Funktion

$$\varphi(t) = f(x + ts)$$

minimiert würde. Dies benötigt viel Zeit und zu viele Funktionsauswertungen. Deshalb wird in der Praxis eine inexacte Schrittweitenbestimmung durchgeführt, die eine Minimierung der Funktion mit möglichst geringem Aufwand realisiert. Dafür werden Bedingungen an die Funktion  $\varphi(t)$  gestellt.

### 5.2.1 Armijobedingungen

Die Armijobedingung ist eine inexacte Schrittweitenbedingung, die durch die Ungleichung ausgedrückt werden

$$f(x_k + t_k s_k) \leq f(x_k) + c_1 t_k \nabla_x f(x_k)^T s_k$$

kann, wobei die Konstante  $c_1 \in (0, 1]$  ist, vgl. [Nocedal und Wright, 2006]. Die Ungleichung sagt aus, dass die Reduktion der Zielfunktion sowohl proportional zur Schrittweite  $t_k$  als auch

zur Richtungsableitung  $\nabla_x f(x_k)^T s_k$  ist. Gilt die Ungleichung, kann eine hinreichende Abnahme von  $f(x)$  garantiert werden. Ein Nachteil kann sein, dass die so bestimmte Schrittweite  $t_k$  sehr klein ist und somit die Zielfunktion nur sehr gering minimiert wird. Ein Algorithmus gemäß der vorgenannten Bedingungen kann mit

```

 $t_1 =$  wähle Startwert, wähle  $\beta \in (0, 1)$ 
while  $\varphi(t_k) \leq f(x_k) + c_1 t_k \nabla_x f(x_k)^T s_k \neg$  OK
do
     $t_{k+1} = \beta t_k$ 
end while

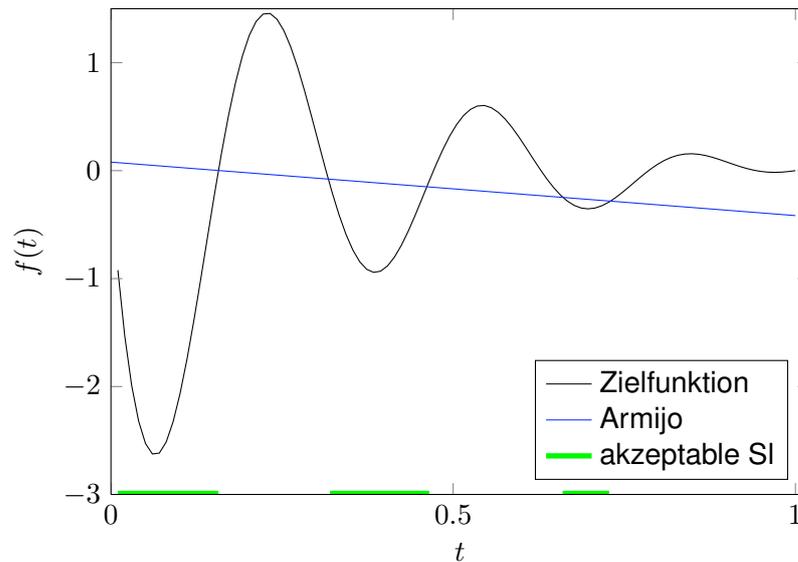
```

dargestellt werden.

Die Konstante  $c_1$  wird in der Praxis klein gewählt, zum Beispiel mit  $c_1 = 10^{-4}$ . Dieser Algorithmus heißt auch Backtracking-Algorithmus, weil bei diesem Algorithmus von einer großen Schrittweite zu einer kleinen geschaut wird, ob die Armijobedingung erfüllt ist. Es ist sinnvoll einen Backtracking-Algorithmus zu verwenden, da dadurch die Suchrichtung am besten ausgenutzt wird. Ein Beispiel für die durch die Armijobedingungen bestimmten Schrittweiten ist in Abbildung zu sehen.

Es gibt weitere Verfahren, um eine Schrittweite zu bestimmen, wie zum Beispiel das Goldsteinverfahren. Hier wird versucht, durch Hinzunahme einer weiteren Bedingung kleine Schrittweiten auszuschließen. Ein Nachteil dabei ist, dass die Zielfunktion häufiger ausgewertet werden muss. Deshalb wird dieser Schrittweitenalgorithmus in der Praxis meist nicht genutzt. In dieser Arbeit wird nur der Armijoalgorithmus verwendet, da der Rechenaufwand sonst zu groß wird.

## Beispiel



(a) Armijoschrittlänge

Abbildung 5.2: In grün sind die Schrittlängen eingezeichnet, die nach der Armijobedingung zulässig sind

## 5.3 Abbruchkriterien

An dieser Stelle sollen die Abbruchkriterien für die oben vorgestellten Algorithmen konkretisiert werden. Dabei werden die Abbruchkriterien von [Gill et al., 1981] verwendet, die jeweils für unrestringierte und restringierte Probleme mit nichtlinearen Nebenbedingungen formuliert werden.

### 5.3.1 Abbruchkriterien für Probleme ohne Nebenbedingungen

Wenn unrestringierte Probleme behandelt werden, so werden in der Regel folgende Abbruchbedingungen verwendet, vgl. [Gill et al., 1981]:

$$\|f(x_{k-1}) - f(x_k)\| < \tau_F(1 + |f(x_k)|) \quad (5.2)$$

$$\|x_{k-1} - x_k\| < \sqrt{\tau_F}(1 + |f(x_k)|) \quad (5.3)$$

$$\|\nabla_x f(x_k)\| \leq \sqrt[3]{\tau_F}(1 + \|f(x_k)\|) \quad (5.4)$$

$$\|\nabla_x f(x_k)\| < \epsilon \quad (5.5)$$

Ein erfolgreicher Abbruch soll dann erfolgen, wenn die Bedingungen (5.2)-(5.4) gleichzeitig erfüllt sind oder (5.5) allein erfüllt ist. Dabei werden durch die ersten beiden Bedingungen geprüft, ob die Folge der  $\{x_k\}$  konvergiert. Die dritte Bedingung ist durch das notwendige Kriterium motiviert. Für ein gut gestelltes Problem impliziert das Zutreffen von (5.2) auch das Eintreten von (5.3). Für ein schlecht gestelltes Problem wird der Algorithmus durch (5.3) gezwungen einen besseren Punkt zu finden, vgl. [Gill et al., 1981]. Der Parameter  $\epsilon$  wird als Maschinengenauigkeit gewählt, da bei einem Gradienten nahe Null eine Extremstelle erreicht ist und keine effiziente Abstiegsrichtung mehr bestimmt werden kann. Der Parameter  $\tau_F$  wird gewählt als  $10^{-q}$ . Das  $q$  kann vom Benutzer frei gewählt werden und gibt an, auf wie viele Stellen nach dem Komma die Lösung korrekt sein soll.

### 5.3.2 Abbruchkriterien für Probleme mit nichtlinearen Nebenbedingungen

Für den restringierten Fall, dass der Lösungsraum durch nichtlineare Nebenbedingungen beschränkt wird, werden andere Abbruchkriterien von [Gill et al., 1981] vorgeschlagen.

$$\|c(x_k)\| \leq \tau_C \quad (5.6)$$

$$\|g_z\| \leq \sqrt{\tau_F} \|\nabla_x f(x_k)\| \quad (5.7)$$

$$\|x_{k-1} - x_k\| \leq \sqrt{\tau_F} (1 + \|x_k\|) \quad (5.8)$$

Der Parameter  $\tau_C$  legt fest, um welche Größe die Nebenbedingungen verletzt werden dürfen. Auch  $\tau_C$  ist frei wählbar und bewegt sich in einer ähnlichen Größenordnung wie  $\tau_F$ .

Ein wesentlicher Unterschied zu den Abbruchkriterien unrestringierter Probleme ist die Betrachtung des projizierten Gradienten  $g_z$ . Dieser wird definiert als  $Z^T \nabla_x f(x_k) = 0$ , wobei die Spalten der Matrix  $Z$  die Basis des Unterraumes aufspannt, der alle Abstiegsrichtungen enthält.

Da es passieren kann, dass beispielsweise die Hessematrix schlecht konditioniert ist und somit keine Abstiegsrichtung mehr bestimmt werden kann, ist es sinnvoll ein zusätzliches Abbruchkriterium einzubauen, das die Anzahl der Iterationen begrenzt. Die Begrenzung sollte in die Abbruchkriterien 5.3.1 und 5.3.2 aufgenommen werden.



# 6 Anwendung

## 6.1 Bildregistrierung

Da die oben vorgestellten Algorithmen auf Bildregistrierungsprobleme mit Volumenerhaltung angewendet werden sollen, wird dieses Problem nun skizziert, eine detaillierte Beschreibung ist in [Modersitzki, 2009] zu finden.

Im Allgemeinen lässt sich feststellen, dass die Registrierung zweier Bilder  $T, R$  das Ziel hat, eine Transformation  $\varphi$  zu finden, um diese Bilder gut aufeinander abzubilden. Dazu wird ein Funktional

$$\min_{\varphi} \mathcal{D}(\varphi(T) - R) + \frac{\alpha}{2} \mathcal{S}(\varphi) \quad \text{u.d.Nb.} \quad c(\varphi) = 0, \quad (6.1)$$

betrachtet, welches sich aus einem Distanzmaß  $\mathcal{D}$ , einem Regularisierer  $\mathcal{S}$ , einer Transformation  $\varphi$  und den Nebenbedingungen zusammensetzt. Diese einzelnen Bestandteile werden im Folgenden kurz beschrieben.

### 6.1.1 Transformationen

Eine Transformation  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  ist eine Abbildung aus dem Bildbereich  $\Omega$  wieder in den Bildbereich  $\Omega$ . Es gibt zwei unterschiedliche Klassen von Transformationen, die parametrischen und die nichtparametrischen. Die parametrischen Transformation verändern immer das gesamte Bild. Zu diesen Transformationen gehören Translation, Rotation, Skalierung und Scherung. Mit nicht parametrischen Transformationen können alle denkbaren Transformationen beschrieben werden. Dabei sind Transformationen möglich, die nur einzelne Pixel betreffen. Dies ist nach der Definition Hardamards ein schlecht konditioniertes Problem, deshalb wird ein Regularisierer  $\mathcal{S}$  benutzt, um die Kondition zu verbessern.

### 6.1.2 Distanzmaße

Distanzmaße sind Maße, die einen Abstand zwischen zwei Bildern messen. Es gibt unterschiedliche Ansätze, wie dieser Abstand gemessen werden kann.

## 6 Anwendung

Das intuitivste Distanzmaß ist die Summe der quadrierten Differenzen (Sum of squared Differences (SSD)), vgl. [Modersitzki, 2009]. Dabei werden die Grauwerte der einzelnen Bilder punktweise voneinander subtrahiert, quadriert und aufsummiert über alle Pixel. Mathematisch kann dies durch

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

formuliert werden. Es gibt noch etliche weitere Distanzmaße, die unterschiedliche Eigenschaften der Bilder berücksichtigen, auf die an dieser Stelle aber nicht weiter eingegangen wird. Für die Auswertung der Daten wird nur das  $SSD$  - Maß betrachtet.

### 6.1.3 Regularisierer

Durch Hinzunahme von Informationen zum Registrierungsfunctional soll ein schlecht gestelltes Problem in ein besser gestelltes überführt werden. Es gibt unterschiedliche Regularisierer, die gewisse Transformationen bevorzugen. Einer der am häufigsten genutzten ist der diffuse Regularisierer, der durch

$$S^{diff}(\varphi) = \frac{1}{2} \int_{\Omega} \sum_i \|\nabla_x \varphi_i\|^2 dx.$$

definiert ist, [Modersitzki, 2009] Der Einfachheit halber wird in der Registrierungsfunktion nur ein diffusiver Regularisierer benutzt. Auf die Darstellung weiterer Regularisierer wird deshalb verzichtet, können aber in [Modersitzki, 2009] nachgelesen werden.

### 6.1.4 Volumenerhaltung

Bei der Registrierung der Bilder soll das Volumen des Templatebildes möglichst nicht verändert werden. Im Verlauf der Registrierung wird deshalb die Differenz des aktuellen Volumens und des Startvolumens  $V_0$ ,  $(V_i - V_0)$ , betrachtet, welche nach Möglichkeit Null sein soll. Diese wird als Nebenbedingung für jedes Pixel an das Zielfunktional gestellt.

### 6.1.5 Registrierungsfunctional

Das Registrierungsfunctional

$$\mathcal{J}(\varphi) = \frac{1}{2} \int_{\Omega} (\varphi(T) - R)^2 dx + \frac{\alpha}{2} \int_{\Omega} \sum_i \|\nabla_x \varphi_i\|^2 dx \text{ u.d.NB. } (V_i - V_0) = 0.$$

wird minimiert, um zwei Bilder volumenerhaltend aufeinander zu registrieren, wobei  $V_i$  die Volumen der einzelnen Pixel sind. Dabei ist  $V_0$  das Startvolumen der Pixel im Templatebild.

Da ein Discretize-then-Optimize-Ansatz verwendet wird, werden die obigen kontinuierlichen Terme zuerst diskretisiert, damit das Minimierungsproblem mit Hilfe des Augmented-Lagrange-Ansatzes gelöst werden kann. Auf die Diskretisierung soll im Weiteren nicht eingegangen werden, da dieses über den Rahmen der vorliegenden Arbeit hinausgeht, es sei an dieser Stelle auf die diesbezüglichen Erläuterungen bei Modersitzki verwiesen, vgl. [Modersitzki, 2009].

## 6.2 Numerische Experimente

Im folgenden Abschnitt sollen der in Abschnitt 4.3 vorgestellte Algorithmus für den Augmented-Lagrange-Ansatz getestet werden. Dafür werden zunächst synthetische Daten genutzt. Außerdem werden die unterschiedlichen Algorithmen zum Lösen unrestringierter Minimierungsprobleme zur Berechnung der inneren Schleife in 4.3 genutzt und verglichen.

### 6.2.1 Implementierung

Die Algorithmen wurden in Matlab (Version 2010 b) implementiert. Insgesamt werden drei der oben vorgestellten Algorithmen gegeneinander getestet, das Gradienten-Verfahren, das BFGS-Verfahren und das Quasi-Newton-Verfahren. Diese drei Verfahren werden in Laufzeit und den gelieferten Ergebnissen verglichen. Das Gradienten-Verfahren ist leicht zu implementieren und wird häufig in der Praxis genutzt. Das BFGS-Verfahren hat bei geeigneter Wahl der Startwerte ein superlineares Konvergenzverhalten und sollte bessere Ergebnisse in der Laufzeit gegenüber dem Gradienten-Verfahren liefern. Das Gauß-Newton-Verfahren wird wahrscheinlich von allen getesteten Algorithmen die besten Ergebnisse in der Laufzeit haben. Eine Approximation der Hessematrix der Nebenbedingungen erfolgt über

$$\begin{aligned}
 C(x) &= \lambda^T c(x) + \mu/2 c(x)^T c(x) \\
 \nabla_x C(x) &= \lambda^T \nabla_x c(x) + \mu \nabla_x c(x)^T c(x) \\
 \nabla_{xx} C(x) &= \sum_i \nabla_{xx} c_i(x) + \mu (\nabla_x c(x)^T \nabla_x c(x) + \sum_i c_i(x) \nabla_{xx} c_i(x)) \\
 &= \mu \nabla_x c(x)^T \nabla_x c(x) + \sum_i (\lambda_i + c_i(x)) \nabla_{xx} c_i(x) \\
 &\approx \mu \nabla_x c(x)^T \nabla_x c(x),
 \end{aligned}$$

da die Hessematrix im Minimum klein ist. Das Newton-Verfahren wurde nicht implementiert, da das Bestimmen der exakten 2. Ableitung gerade in der Bildregistrierung zu aufwendig ist. Als Abbruchkriterien in der äußeren Schleife des Algorithmus' aus 4.3 wurden die in Abschnitt 5.3.1 vorgestellten Kriterien genommen, mit einem  $\tau_F = 1e - 4$  und maximal 10 Iterationen. Außerdem sollten alle Funktionswertvergleiche und auch die Betrachtung des Gradienten auf

## 6 Anwendung

die Lagrange-Funktion bezogen werden. Dies ist im restringierten Fall die Zielfunktion, für die an einem Minimum die notwendigen Bedingungen erfüllt sein müssen. Ein Abbruch der inneren Schleife des Algorithmus' 4.3 erfolgt, wenn der Gradient der Augmented-Lagrange-Funktion  $\|\nabla_x \mathcal{L}_A\| \leq 1e-4$  ist oder 5 Iterationen erfolgt sind. Zur Schrittlängenbestimmung wird der Armijoalgorithmus verwendet, mit  $c_1 = 1e-4$ .

### 6.2.2 Daten

Zum Testen der unterschiedlichen Algorithmen wird versucht, ein Quadrat und ein Rechteck mit der doppelten Seitenlänge des Quadrats aufeinander zu registrieren. Die genaue Erzeugung der synthetischen Daten kann im Anhang eingesehen werden. Dabei soll das Volumen des Templatebildes nach Möglichkeit erhalten bleiben. Für die Testrechnung mit den synthetischen Daten wurde  $\alpha = 1e3$  gewählt. Die Auflösung der Bilder beträgt  $64 \times 64$ . Es wird eine Armijoschrittlängenbestimmung, Abschnitt 5.2.1, benutzt und als Abbruchkriterien werden die in Abschnitt 5.3.1 vorgestellten Kriterien für Probleme ohne Nebenbedingungen benutzt. Als Startwerte bei der Registrierung wurden  $\mu = 1e6$  und  $\lambda_i = 1, \forall i \in \mathcal{E}$  gewählt. Die Distanz der Bilder vor der Registrierung beträgt  $1.1286e3$ . Als Zielfunktionswert wird Distanzmaß plus Regularisierer betrachtet. Welche Abbruchkriterien erfüllt waren am Ende der Registrierung ist in den letzten Zeilen der Tabellen zu sehen. Dabei stehen die ersten vier Zahlen für die Bedingungen aus Abschnitt 5.3.1 und die fünfte Zahl gibt an, ob das Iterationmaximum der äußeren erreicht worden ist, dabei gibt 1 an, dass es zutrifft, und 0, dass es nicht zutrifft.

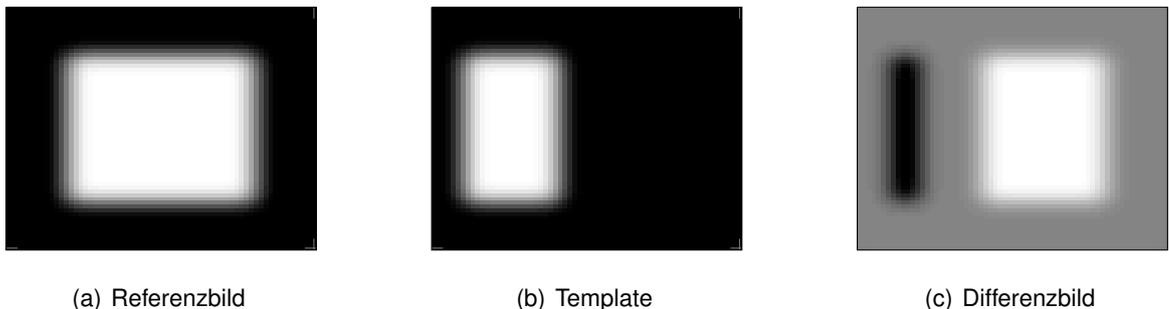


Abbildung 6.1: Synthetische Daten

## 6.3 Registrierung mit und ohne Volumenerhaltung

Für die Registrierung ohne Nebenbedingungen wurde das Gauß-Newton-Verfahren verwendet. Der Regularisierungsparameter  $\alpha$  wurde  $1e3$  gesetzt. Die maximale Anzahl an Iterationen bei der Registrierung ohne Nebenbedingungen wurde auf 20 gesetzt.

Bei der Registrierung ohne Nebenbedingungen konnte eine Reduktion des Distanzmaßes von 99.56% erzielt werden. Die Reduktion des Distanzmaßes bei der Registrierung mit Nebenbedingungen beträgt 53.75%, allerdings bleiben die Volumen der Pixel in der Summe um den Faktor  $1e7$  besser erhalten.

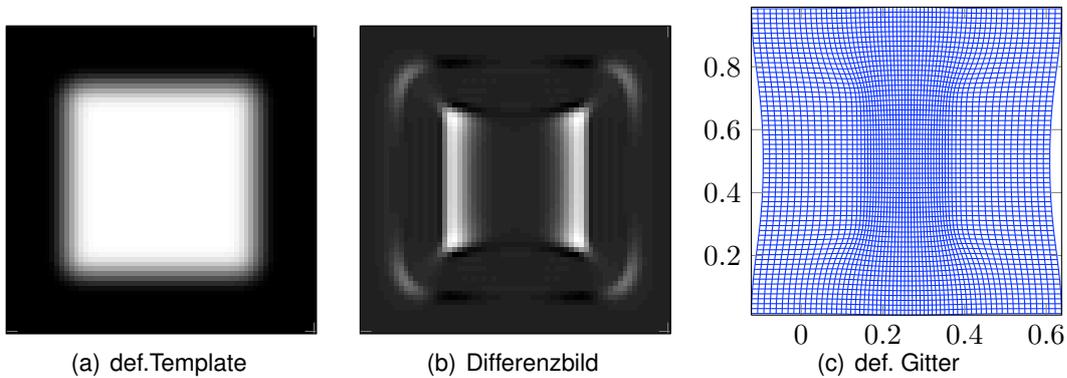


Abbildung 6.2: Ergebnisse für die Registrierung ohne Nebenbedingungen

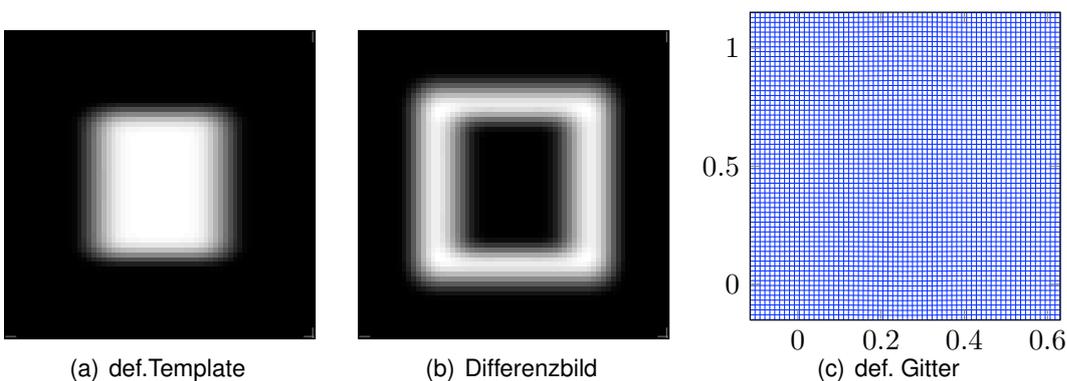


Abbildung 6.3: Ergebnisse für die Registrierung mit Nebenbedingungen

## 6 Anwendung

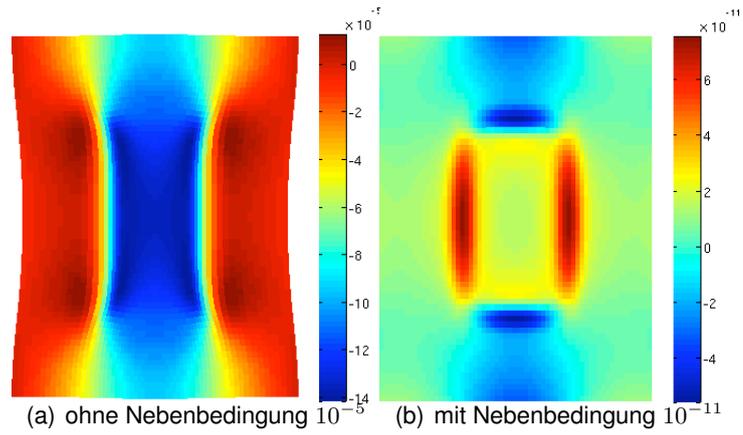


Abbildung 6.4: Volumenveränderung für die Registrierung mit und ohne Nebenbedingungen

**Tabelle 2**

	ohne Nebenbedingungen	mit Nebenbedingungen
Zielfunktionswert	68.4892	545.3951
SSD	5.1135	522.7146
$\alpha$ Regularisierer	63.3767	22.6805
Funktionsauswert.	21	50
Zeit in s	3.3978	11.8501
$\sum_i (V_i - V_0)$	-0.5306	6.0336e-8
Iterationen	7	7
Abbruchkriterien	1 1 1 0 0	1 1 1 0 0

## 6.4 Vergleich der Verfahren

Die Verfahren wurde an dem obigen Beispiel gegeneinander getestet. Die Parameter sind, wie oben schon erwähnt auf  $\alpha = 1e3$  und  $\mu = 1e6$  gewählt worden. Es wurde eine Armijo-schrittlängenbestimmung genutzt.

Das Gauß-Newton-Verfahren erzielt im Vergleich zu allen anderen getesteten Verfahren die beste Volumenerhaltung und einer Reduktion des Distanzmaßes um 53.75%. Die anderen Verfahren erreichen im Vergleich zum Gauß-Newton-Verfahren keine so gute Reduktion des Distanzmaßes (13.26% Gradienten-Verfahren , 22.74% BFGS-Verfahren), einhergehend mit einer stärkeren Verletzung der Nebenbedingungen. Außerdem benötigen sie mehr Zeit zur Auswertung. Das Gauß-Newton-Verfahren verändert das Gitter so, dass es nach der Registrierung rechteckig ist, die Transformation ist aber trotzdem zulässig. Bei den anderen beiden Verfahren bleibt das Gitter quadratisch.

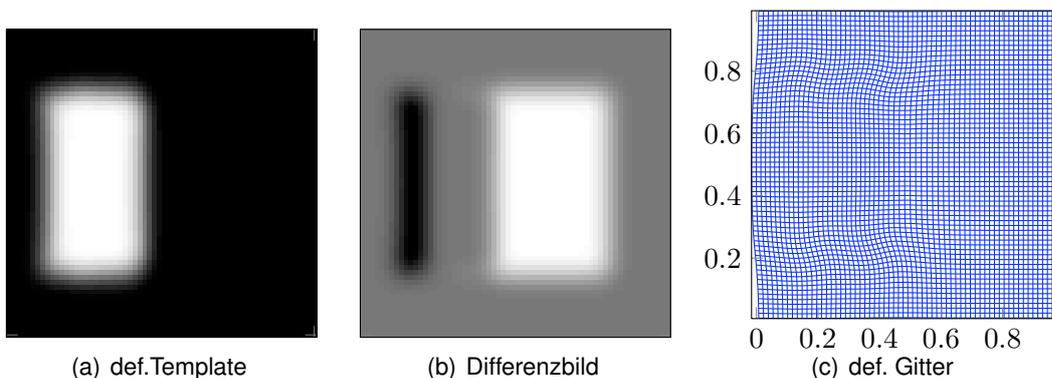


Abbildung 6.5: Ergebnisse für die Registrierung mit dem Gradienten-Verfahren

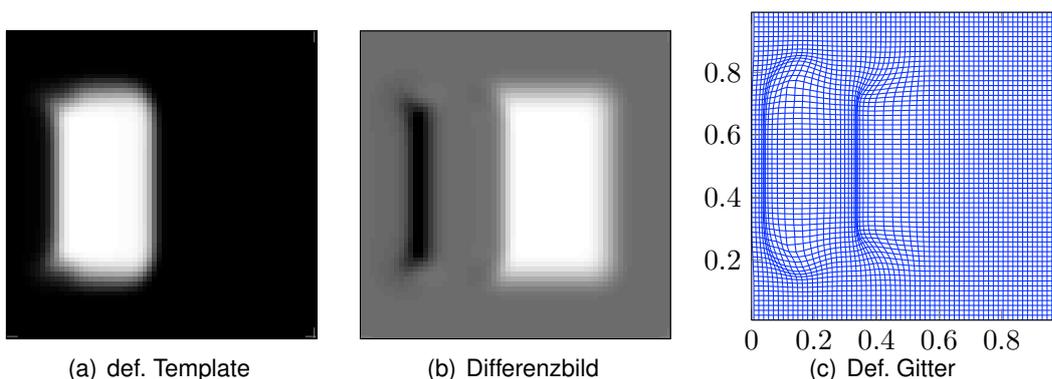


Abbildung 6.6: Ergebnisse für eine Registrierung mit dem BFGS-Verfahren

## 6 Anwendung

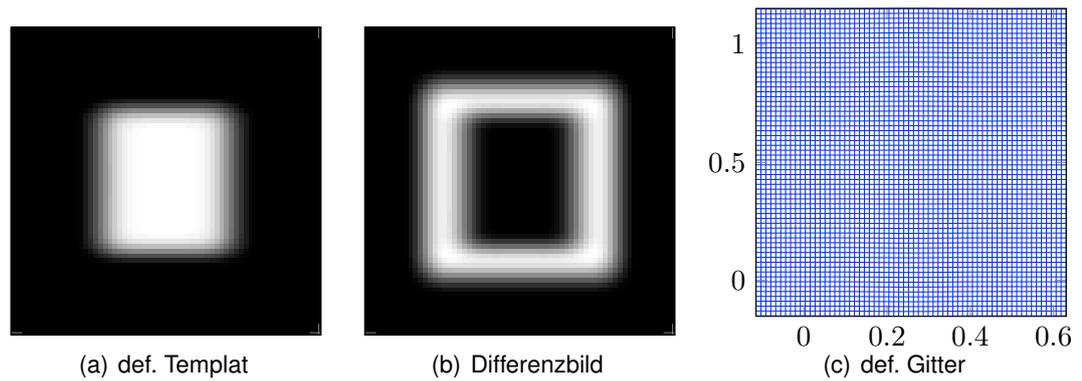


Abbildung 6.7: Ergebnisse für eine Registrierung mit dem Gauß-Newton-Verfahren

**Tabelle 3**

Verfahren	Gradienten	BFGS	Gauß-Newton
Zielfunktionswert	985.1546		545.3951
SSD	979.8372	0.0089	872.664
$\alpha$ Regularisierer	5.3174	55.021	22.6805
Funktionsauswert .	1597	255	50
Zeit in s	83.9471	506.4698	11.8501
$\sum_i (V_i - V_0)$	0.02775	0.0075	6.0336e-8
Iterationen	10	10	7
Abbruchkriterien	0 1 0 0 1	1 1 0 0 1	1 1 1 0 0

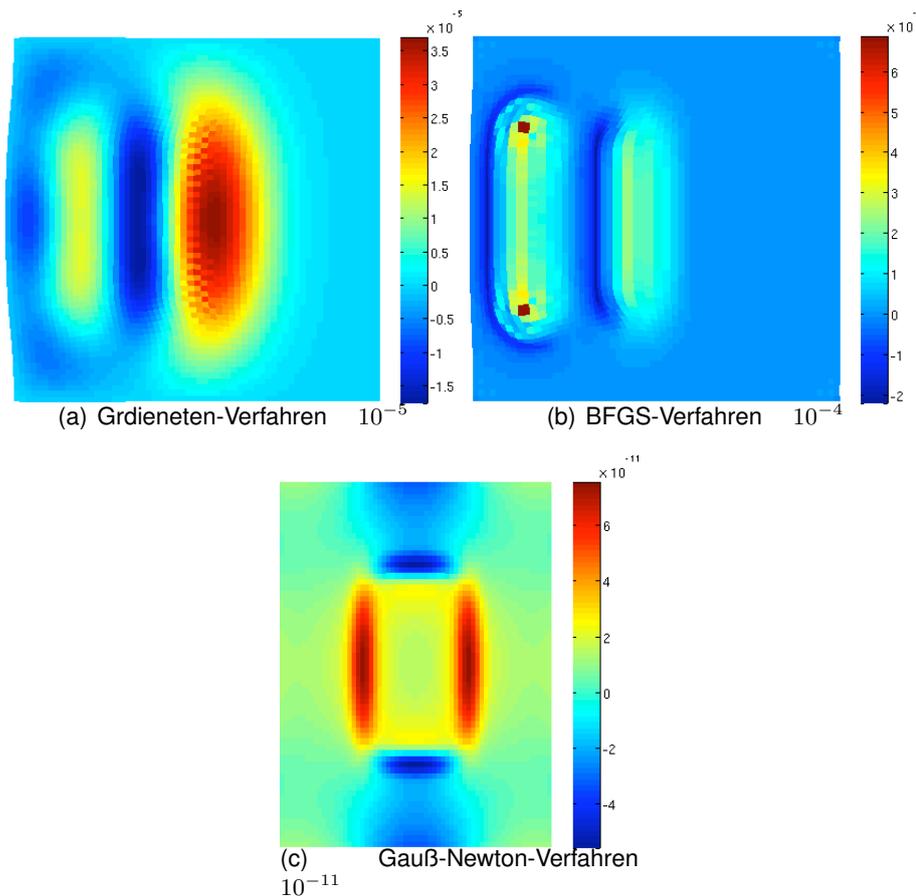


Abbildung 6.8: Volumenveränderung für die Registrierung mit Gradienten-, BFGS- und Gauß-Newton-Verfahren

## 6.5 Parameterwahl

Hier wird gezeigt, dass das Ergebnis der Registrierung abhängig ist von der Wahl des Parameters  $\mu$ . Es werden drei unterschiedliche Startwerte gegeneinander getestet. Der Regularisierungsparameter  $\alpha$  wird mit  $1e3$  gewählt. Es wird eine Armijoschrittlängenbestimmung genutzt.

Es lässt sich erkennen, dass die Wahl des Startparameters einen maßgeblichen Einfluss auf das Ergebnis des Algorithmus' hat. Ein zu niedriger Startwert hat zur Folge, dass die Nebenbedingungen nicht genügend Gewicht in der Zielfunktion haben und deshalb stärker verletzt werden, als wenn ein höherer Startparameter gewählt wird.

## 6 Anwendung

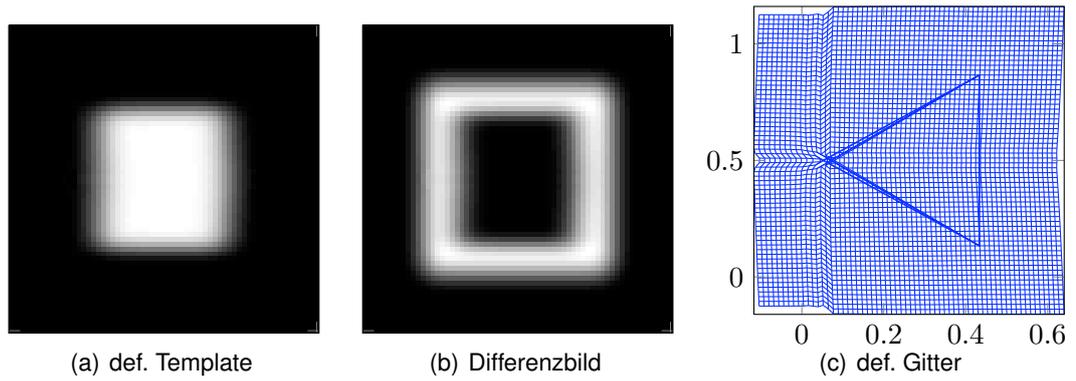


Abbildung 6.9: Ergebnisse für  $\mu = 1e3$

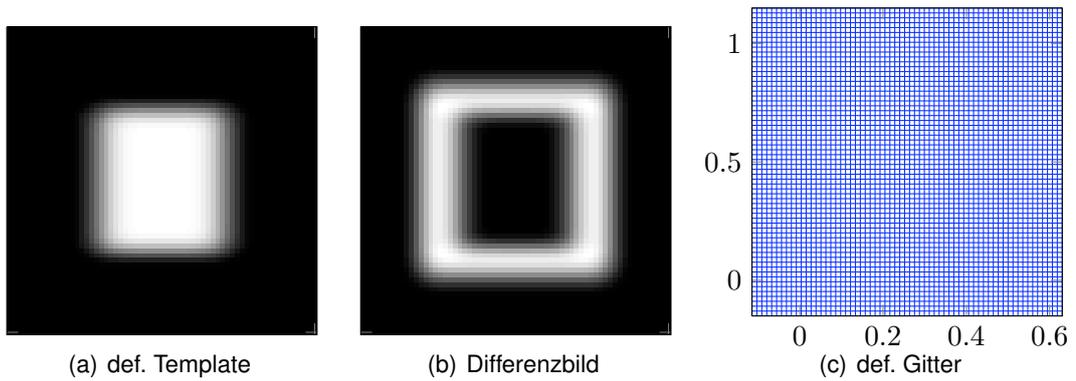


Abbildung 6.10: Ergebnisse für  $\mu = 1e9$

**Tabelle 4**

$\mu$	1e3	1e9	1e15
Zielfunktionswert	68.476	545.56	682.534
SSD	5.0765	522.9947	659.9536
$\alpha$ Regularisierer	63.3995	22.5653	22.5804
Funktionsauswert.	14	36	425
Zeit in s	3.6330	8.0549	39.6342
$\sum_i (V_i - V_0)$	-0.5274	-4.1371e-9	-4.2542e-06
Iterationen	2	5	10
Abbruchkriterien	1 1 1 0 0	1 1 1 0 0	1 1 0 0 1

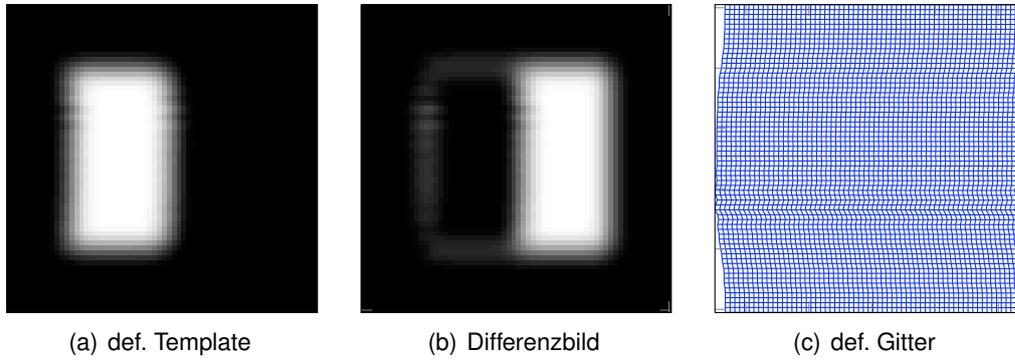


Abbildung 6.11: Ergebnisse für  $\mu = 1e15$

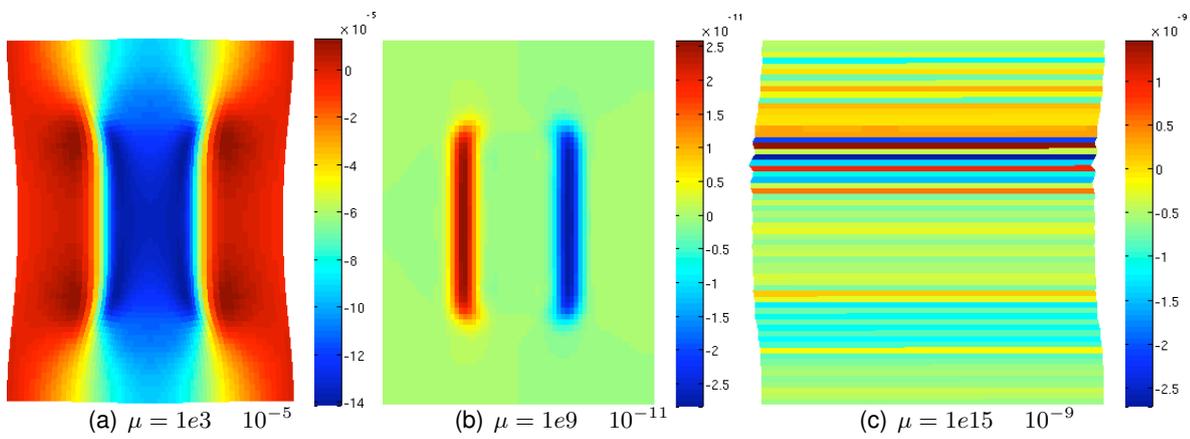


Abbildung 6.12: Volumenveränderung beim Gauß-Newton-Verfahren für unterschiedliche  $\mu$



# 7 Fazit, Diskussion und Ausblick

## 7.1 Fazit

In dieser Arbeit wurde die Augmented-Lagrange-Funktion vorgestellt und ein Modellalgorithmus entworfen, mit dem Minimierungsprobleme gelöst werden können. Weiter wurden unterschiedliche Ansätze zum Lösen unrestringierter Optimierungsprobleme vorgestellt, da mit Hilfe der Augmented-Lagrange-Funktion restringierte Optimierungsprobleme zu unrestringierten umformuliert werden und diese zum Lösen des Problems genutzt werden können. Diese Lösungsmethoden wurden implementiert und auf Fragestellungen aus der Bildregistrierung angewendet. Die Auswertung erfolgte mit synthetischen Daten, um die Unterschiede und Gemeinsamkeiten der verschiedenen Algorithmen aufzuzeigen.

- Es konnte gezeigt werden, dass bei der Registrierung von unterschiedlichen Objekten, die unterschiedliche Volumina haben, eine bessere Volumenerhaltung durch den Augmented-Lagrange-Ansatz erzielt werden konnte als bei einer Nichtbeachtung der Volumenerhaltung.
- Weiter konnte gezeigt werden, dass die Volumenerhaltung auch durch das Lösungsverfahren beeinflusst wird. Das Gauß-Newton-Verfahren erzielte hier die besten Ergebnisse. Sowohl das Gradienten-Verfahren und das BFGS-Verfahren waren langsamer in der Laufzeit und erzielten im Vergleich nicht so gute Ergebnisse.
- Bei allen getesteten Algorithmen wurden in der inneren Schleife die maximale Anzahl an Iterationen ausgenutzt, d.h. die anderen Abbruchbedingungen waren nie erfüllt.

## 7.2 Diskussion

Die unterschiedlichen Ergebnisse der Verfahren könnten auf das gewählte Beispiel zurückgehen.

Die Wahl des Parameters  $\mu$  beeinflusst die Ergebnisse maßgeblich. Jedoch konnte keine Aussage gemacht werden, wie dieser Parameter am Anfang genau zu wählen ist. Wird der Parameter zu hoch gewählt, kann es passieren, dass die zulässigen Transformationen stark

eingeschränkt sind, siehe 6.11. Es kommt deshalb zu einer sehr geringen Veränderung des Templates einhergehend mit einer geringen Reduktion des Distanzmaßes. Wird der Parameter zu niedrig gewählt, so kann es sein, dass die Nebenbedingungen sehr stark verletzt werden, siehe 6.9. Trotz der Erhöhung des Parameters  $\mu$  nach dem Durchlauf der inneren Schleife kann dies nicht wieder rückgängig gemacht werden.

Die in Abschnitt 5.3.2 vorgestellten Abbruchkriterien konnten nicht implementiert werden, da die Matrix, die benötigt wird, um den projizierten Gradienten zu berechnen, nicht realisiert werden konnte.

### 7.3 Ausblick

Der Augmented-Lagrange-Ansatz sollte für andere Beispiele getestet werden. Hier wird sich auch zeigen, ob der Unterschied zwischen den Lösungsverfahren immer noch vorhanden ist.

Ein weiterer Schritt wäre es die Abbruchkriterien für nicht lineare Nebenbedingungen zu implementieren. Der Fokus der Abbruchkriterien liegt eher auf den Nebenbedingungen. Eventuell könnte damit eine bessere Einhaltung der Nebenbedingungen realisiert werden.

Ein anderer Regularisierer sollte getestet werden, um Informationen zu bekommen, inwieweit dieser Einfluss auf die Nebenbedingungen hat.

Ein Test mit einem Multi-Level-Ansatz könnte gemacht werden, um zu schauen, wie der Algorithmus sich verhält und Aufschluss darüber geben, in wie weit der Augmented-Lagrange-Ansatz dabei zu verwenden ist .

Da die Abbruchkriterien der inneren Schleife nie erfüllt waren und deshalb immer die maximale Anzahl an Iterationen in der Inneren Schleife durchlaufen wurden, ist es vielleicht besser andere Abbruchkriterien zu wählen.

# Literaturverzeichnis

[Arens et al., 2009] Arens, T., Hettlich, F., Karpfinger, C., Kockelkorn, U., Lichtenegger, K., und Stachel, H. (2009). *Mathematik*. Spektrum Akademischer Verlag.

[Courant, 1943] Courant, R. (1943). *Variational methods for the solution of problems of the equilibrium and vibrations*. Bulletin of the Americal Mathematical Society.

[Geiger und Kanzow, 2002] Geiger, C. und Kanzow, C. (2002). *Theorie und Numerik restrin-gierter Optimierungsaufgaben*. Springer Verlag.

[Gill et al., 1981] Gill, P., Murray, W., und Wright, M. (1981). *Practical optimization*. Academic press.

[Modersitzki, 2009] Modersitzki, J. (2009). *FAIR: Flexible Algorithms for Image Registration*. SIAM, Philadelphia.

[Nocedal und Wright, 2006] Nocedal, J. und Wright, S. (2006). *Numerical Optimization*. Springer Verlag.



# Anhang

## Matlab File für die synthetischen Daten

```
clear all
close all
n=64;
x=linspace(-1,1,n);
[X,Y] = meshgrid(x,x);
kantenl=0.3;

T = abs(X+.5)<kantenl & abs(Y)<2*kantenl;
R = abs(X)<2*kantenl & abs(Y)<2*kantenl;

T=100*T;
R=100*R;

smooth = @(X) convn(convn(X,[1 2 1]'/4,'same'),[1 2 1]/4,'same');
nsmooth=10;
for i=1:nsmooth
    R = smooth(R);
    T = smooth(T);
end
figure(1)
imagesc(R);colormap(gray)
figure(2)
imagesc(T);colormap(gray)
figure(3)
imagesc(R-T);colormap(gray)
sum(sum(R-T))
```