



EFFEKT VERSCHIEDENER MODELLE FÜR OPTISCHEN FLUSS AUF DAS SUPERRESOLUTION-PROBLEM

MASTERARBEIT
zur Erlangung des akademischen Grades
MASTER OF SCIENCE

Westfälische Wilhelms-Universität Münster
Fachbereich Mathematik und Informatik

Betreuung:
Prof. Dr. Benedikt Wirth

Eingereicht von:
Felix Rehfeldt

Münster, den 13. Oktober 2015

Eidesstattliche Erklärung

Hiermit versichere ich, Felix Rehfeldt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Gedanklich, inhaltlich oder wörtlich Übernommenes habe ich durch Angabe von Herkunft und Text oder Anmerkung belegt bzw. kenntlich gemacht. Dies gilt in gleicher Weise für Bilder, Tabellen, Zeichnungen und Skizzen, die nicht von mir selbst erstellt wurden.

Münster, den 13. Oktober 2015

Unterschrift

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufbau der Arbeit	2
2	Mathematische Grundlagen	4
2.1	Konvexe Analysis	4
2.2	Differenzierbarkeit konvexer Funktionen	10
2.3	Umformung in die Primal duale Formulierung	16
2.4	Primal-Dual Algorithmus	19
3	Das Optical Flow Problem	23
3.1	Motivation	23
3.2	Mathematische Umsetzung	24
3.3	Lösungsverfahren	27
3.3.1	Lösungsverfahren mit dem Primal-Dual Algorithmus	27
3.3.2	Lösungsverfahren mit dem Gradientenabstiegsverfahren	29
3.4	Diskretisierung	30
3.5	Implementierung	33
3.6	Evaluation	34
3.6.1	Fehlermaße	34
3.6.2	Mögliche Herausforderungen	36
3.6.3	Experimentelle Erweiterungen	38
3.6.4	Vergleich mit anderen Algorithmen	46
4	Das Superresolution Problem	57
4.1	Motivation	57
4.2	Definition des Superresolution Problems	58
4.2.1	Die verwendeten Operatoren	58
4.3	Lösungsverfahren mit dem Primal-Dual Algorithmus	62
4.4	Diskretisierung und Implementierung	63
4.5	Evaluation	64
4.5.1	Fehlermaße	64
4.5.2	Testdaten	65
4.5.3	Effekt der verschiedenen optischen Flüsse	67
5	Fazit	75

1 Einleitung

1.1 Motivation

Es gibt in der Praxis viele Anwendungsgebiete, in denen hochauflösende Bilder benötigt werden. In manchen Fällen ist es technisch jedoch nicht immer möglich, die Bilder in der gewünschten Auflösung aufzunehmen. Für diesen Fall gibt es verschiedene mathematische Ansätze zur Erhöhung der Auflösung. In dem verwendeten Ansatz wird aus einer Reihe von niedrigauflösenden Bildern ein hochauflösendes Bild berechnet. Das mathematische Problem, das sich mit dieser Aufgabe befasst, nennt man Superresolution Problem.

Für die Berechnung sind zwei wesentliche Schritte notwendig. Als Erstes erfolgt die Registrierung. Dabei wird von einem ausgewählten Referenzbild zu allen anderen Eingabebildern jeweils eine Abbildung berechnet, die das Referenzbild möglichst exakt in das andere Bild transformiert. Eine hundertprozentige Überdeckung ist dabei normalerweise nur in künstlichen Beispielen zu erreichen. Nach der Registrierung wird dann ein Bild bestimmt, welches nach den jeweiligen Transformationen allen Eingabebildern gleichzeitig möglichst genau entsprechen soll. Das so berechnete Bild ist das hochauflösende Superresolution Bild.

Die Qualität des Superresolution Bildes hängt stark davon ab, wie gut die in der Registrierung berechneten Abbildungen das Referenzbild in die Eingabebilder überführen. In dieser Arbeit werde ich deshalb für einen Überlagerungsansatz, der auf einem Minimierungsproblem beruht, verschiedene Möglichkeiten für die Registrierung evaluieren. Ausgehend von einem Modell zur Registrierung von zwei Bildern, sollen verschiedene Fragestellungen geklärt werden. Zunächst stellt sich die Frage, welche Schritte für eine zufriedenstellende Lösung notwendig sind. Desweiteren muss untersucht werden, mit welchem Verfahren die Lösung berechnet werden kann und welche Modellerweiterungen vielversprechend sind. Für die Vergleichbarkeit muss abschließend noch der Vergleich zu bereits etablierten Verfahren gezogen werden.

Nachdem erforscht wurde, wie die Registrierung optimal gelöst werden kann, ist es sinnvoll, die verschiedenen Lösungen zur Registrierung im Weiteren auch zu testen. Dazu

wird aufbauend auf den Ergebnissen der Registrierung untersucht, welche Auswirkungen die verschiedenen Ansätze auf das Superresolution Problem haben, und welche Probleme die optimale Lösungsfindung beeinträchtigen.

1.2 Aufbau der Arbeit

Zu Beginn der Arbeit werden die, für die Modellherleitung und Lösung, notwendigen Grundlagen zusammengefasst. Da die behandelten Minimierungsprobleme konvex sind, werden in Abschnitt 2.1 zunächst die Grundlagen der konvexen Analysis aufbereitet. Ein wichtiger Schritt innerhalb dieses Abschnittes, ist die Legendre-Fenchel Transformation und der Dualitätssatz für bestimmte Funktionen. Mit ihrer Hilfe werden die Minimierungsprobleme zu Sattelpunktproblemen umgeformt. In Abschnitt 2.2 wird dann ein der Differentialrechnung ähnliches Konzept für konvexe Funktionen eingeführt und der Zusammenhang mit der gewöhnlichen Differentialrechnung erläutert. Dies ist notwendig, da der einfachste Weg Minimierungsprobleme zu lösen, über die notwendige Bedingung für deren Ableitung führt. Im Anschluss daran wird in Abschnitt 2.3 die Umformung in ein Sattelpunktproblem beispielhaft aufgezeigt. Im letzten Teil der mathematischen Grundlagen wird schließlich ein allgemeines Verfahren zur Lösung von Sattelpunktproblemen hergeleitet. Dieses Verfahren wird später für die Registrierung und Überlagerung zu einem Superresolution Bild genutzt.

Im Kapitel 3 geht es um die Registrierung. In der Literatur hat sich für die Berechnung der Veränderung von einem Bild zum nächsten der Begriff Optical Flow etabliert. Das Kapitel beginnt mit einer kurzen Motivation und einer mathematischen Definition des Modells. Dieses wird dabei von der ersten Idee ausgehend hergeleitet. In Abschnitt 3.2 wird darüber hinaus die Motivation für ein Implementierungsdetail geliefert, ohne das das Optical Flow Problem nicht exakt gelöst werden kann. In Abschnitt 3.3 werden zwei mögliche Lösungsverfahren für das Problem hergeleitet, bevor in Kapitel 3.4 die Diskretisierung vorgestellt wird. Anschließend folgen noch ein Abschnitt für die Implementierung des Lösungsverfahrens und ein Abschnitt für die Evaluation. Im Abschnitt der Evaluation werden Fehlermaße für die objektive Bewertung der Ergebnisse eingeführt, die Herausforderungen des Optical Flow Problems vorgestellt und verschiedene Lösungen für das Problem verglichen.

Der vierte Teil dieser Arbeit beinhaltet eine Motivation für das Superresolution Problem mit einer kurzen Beschreibung von zwei Anwendungsbeispielen. Im folgenden Abschnitt

wird ein Modell für die Lösung des Problems angegeben, bei dem eine vorherige Registrierung vorausgesetzt wird. Der Teil des Problems, der sich mit der Überlagerung befasst, wird deshalb im Laufe der Arbeit manchmal auch stellvertretend Superresolution Problem genannt. In Abschnitt 4.3 wird für dieses Modell ein Lösungsverfahren mit dem Algorithmus aus den mathematischen Grundlagen genauer erläutert. Es folgen ein Abschnitt zur Diskretisierung und Implementierung des Verfahrens und ein Abschnitt zur Evaluation. Der Abschnitt der Evaluation wird aufgeteilt in die Einführung von Fehlermaßen für die Auswertung, einen Abschnitt zu den Testfällen und einem Vergleich wie sich die verschiedenen Lösungen zur Registrierung auf das hochauflösende Bild auswirken.

Zu guter Letzt werden die Ergebnisse aus Kapitel 3 und 4 im Fazit noch einmal kurz zusammengefasst und ein kurzer Ausblick gegeben, wie die Erforschung des Problems fortgesetzt werden könnte.

2 Mathematische Grundlagen

2.1 Konvexe Analysis

Ein kurzer Hinweis vorab: Die Theorie für konvexe Analysis wird häufig auf der Menge der Funktionen $\{f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\pm\infty\}\}$ aufgebaut. Da diese Erweiterung in unserem Fall größtenteils nicht benötigt wird, wird auf die Unterschiede an den entscheidenden Stellen hingewiesen. Generell beruht dieses Kapitel größtenteils auf den Ausführungen von [Roc70] und [Mar77].

Definition 2.1.1. Eine Menge $X \subset \mathbb{R}^n$ heißt konvex, falls gilt:

$$x, y \in X \quad \Rightarrow \quad \lambda x + (1 - \lambda)y \in X \quad \forall \lambda \in [0, 1]$$

Definition 2.1.2. Sei $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine Funktion. Dann ist der Epigraph von f die Menge

$$\text{epi } f = \{(x, h) \in U \times \mathbb{R} \mid f(x) \leq h\}$$

Definition 2.1.3. 1. Sei $f : X \subset \mathbb{R}^n \mapsto \mathbb{R}$ eine Funktion. Dann heißt f konvex, falls der Epigraph von f eine konvexe Menge ist.

2. f heißt affin, falls f und $-f$ konvex sind.

Korollar 2.1.4. Aus der Definition folgt direkt, dass die Menge X konvex sein muss, da sonst der Epigraph von X nicht konvex sein kann. Außerdem folgen diese Äquivalenzen:

$$\begin{aligned} f \text{ konvex} &\Leftrightarrow f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \quad \forall t \in [0, 1] \wedge \forall x, y \in X \\ &\stackrel{!}{\Leftrightarrow} f\left(\sum_{i=1}^n t_i x_i\right) \leq \sum_{i=1}^n t_i f(x_i) \quad \text{für } t_i \in [0, 1] \text{ mit } \sum_{i=1}^n t_i = 1 \wedge x_i \in X \forall i \end{aligned}$$

$$\begin{aligned} f \text{ affin} &\Leftrightarrow f(x) = x \cdot v + r, \quad \text{mit } v \in \mathbb{R}^n \wedge r \in \mathbb{R} \\ &\Leftrightarrow f(tx + (1-t)y) = tf(x) + (1-t)f(y) \quad \forall t \in [0, 1] \wedge \forall x, y \in X \end{aligned}$$

Beweis. Zeige: f konvex $\Rightarrow f$ erfüllt die Jensen Ungleichung.

IV: Für alle $n \in \mathbb{N}$ gelte: Falls $t_i \in [0, 1] \forall i$ mit $\sum_{i=1}^n t_i = 1$ und $x_i \in X \forall i$, so gilt:

$$f\left(\sum_{i=1}^n t_i x_i\right) \leq \sum_{i=1}^n t_i f(x_i) \quad .$$

IA: Die Aussage gilt für $n = 1, 2$ da f konvex.

IS:

$$\begin{aligned} f\left(\sum_{i=1}^{n+1} t_i x_i\right) &= f\left(t_{n+1} x_{n+1} + (1 - t_{n+1}) \left(\sum_{i=1}^n \frac{t_i}{1 - t_{n+1}} x_i\right)\right) \\ &\leq t_{n+1} f(x_{n+1}) + (1 - t_{n+1}) f\left(\sum_{i=1}^n \frac{t_i}{1 - t_{n+1}} x_i\right) \\ &\stackrel{IV}{\leq} t_{n+1} f(x_{n+1}) + (1 - t_{n+1}) \sum_{i=1}^n \frac{t_i}{1 - t_{n+1}} f(x_i) = \sum_{i=1}^{n+1} t_i f(x_i) \end{aligned}$$

□

Satz 2.1.5. Sei $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ konvex. Falls f an der Stelle x_m ein lokales Minimum hat, so ist das Minimum auch ein globales Minimum.

Beweis. Angenommen es ex. $y \in X$ mit $f(y) < f(x_m)$.

Da x_m ein lokales Minimum ist, existiert ein $r > 0$ mit $f(x) \leq f(z) \quad \forall z \in \mathbf{B}_r(x)$.

Sei nun $z_y := x + \frac{s}{|y-x|}(y-x)$ mit $s \in (0, r)$. $\Rightarrow z_y \in \mathbf{B}_r(x)$

$$\begin{aligned} \Rightarrow f(x) &\leq f(z_y) = f\left(x + \frac{s}{|y-x|}(y-x)\right) = f\left(\left(1 - \frac{s}{|y-x|}\right)x + \frac{s}{|y-x|}y\right) \\ &\leq \left(1 - \frac{s}{|y-x|}\right) f(x) + \frac{s}{|y-x|} f(y) \\ &< \left(1 - \frac{s}{|y-x|}\right) f(x) + \frac{s}{|y-x|} f(x) = f(x) \end{aligned}$$

Widerspruch. □

Satz 2.1.6. Sei $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ konvex. Dann gilt:

$$f(x) = \sup\{\alpha(x) \mid \alpha \text{ affin, } \alpha(z) \leq f(z) \forall z \in X\} \quad \forall x \in X$$

Beweis. Sei $x \in X$ beliebig.

I. $\sup_{\alpha} \alpha(x) \leq f(x)$:

Aus der Voraussetzung $\alpha(z) \leq f(z) \forall z \in X$ folgt $\alpha(x) \leq f(x)$. Also gilt auch $\sup_{\alpha} \alpha(x) \leq f(x)$.

II. $\sup_{\alpha} \alpha(x) \geq f(x)$:

Nach der Definition von $\text{epi } f$ gilt für eine reelle Funktion ($f \not\equiv -\infty$) $\text{epi } f^C \neq \emptyset$. Da $\overline{\text{epi } f}$ abgeschlossen und konvex ist existiert, da $\begin{pmatrix} x \\ f(x) \end{pmatrix} \in \overline{\text{epi } f}$, nach dem Trennungssatz ein Halbraum H_x mit $\text{epi } f \subset \overline{\text{epi } f} \subset H_x$ und $\begin{pmatrix} x \\ f(x) \end{pmatrix} \in \partial H_x$ (vgl. [Mar77] V.1.4).

Da ∂H_x eine nicht vertikale Hyperfläche ist, ex. (mit Skalierung) $\begin{pmatrix} n_x \\ 1 \end{pmatrix} \perp \partial H_x$. Die Hyperfläche kann also charakterisiert werden durch folgende Menge:

$$\partial H_x = \left\{ z \in X \times \mathbb{R} \mid \left\langle \begin{pmatrix} n_x \\ 1 \end{pmatrix}, z - \begin{pmatrix} x \\ f(x) \end{pmatrix} \right\rangle = 0 \right\}$$

Damit ergibt sich aus einer kleinen Rechnung eine affine Abbildung α_x (vgl. Abbildung 2.1):

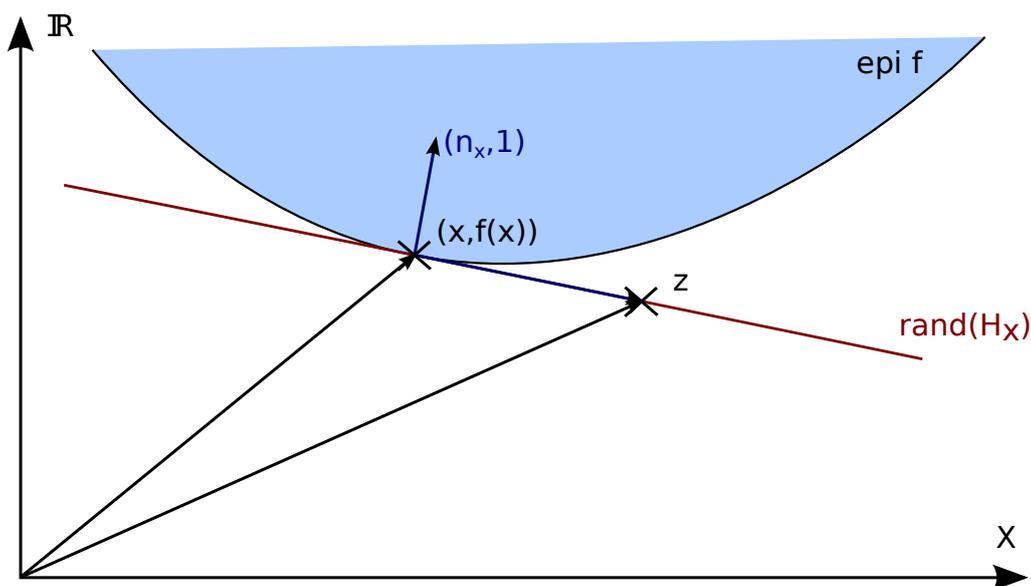


Abbildung 2.1: Konstruktion von $(n_x, 1)$

$$\begin{aligned}
 0 &= n_x \cdot (z_X - x) + z_{\mathbb{R}} - f(x) \\
 \Leftrightarrow \quad z_{\mathbb{R}} &= n_x \cdot x + f(x) - n_x \cdot z_X \\
 \text{also} \quad \alpha_x(z_X) &:= -n_x \cdot z_X + f(x) + n_x \cdot x = f(x) + \langle n_x, x - z_X \rangle \quad (2.1)
 \end{aligned}$$

Nun wurde α_x gerade so definiert, dass für $y \in X$ beliebig $\begin{pmatrix} y \\ \alpha_x(y) \end{pmatrix} \in \partial H_x$. Also gilt

insbesondere $\begin{pmatrix} y \\ \alpha_x(y) \end{pmatrix} \in \overline{\text{epi } f^C}$. Daraus folgt wiederum $\alpha_x(y) \leq f(y) \forall y \in X$.

Damit habe ich eine affine Funktion $\alpha_x \leq f$ gefunden mit $\alpha_x(x) = -n_x \cdot x + f(x) + n_x \cdot x = f(x)$. Also $\sup_{\alpha} \alpha(x) \geq f(x)$. \square

Definition 2.1.7. Eine reelle Funktion $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ heißt unterhalbstetig, falls für jeden Punkt $x \in U$ gilt:

$$f(x) \leq \lim_{i \rightarrow \infty} f(x_i)$$

für jede Folge x_i aus U mit $x_i \rightarrow x$ für $i \rightarrow \infty$.

Definition 2.1.8. Sei $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine konvexe Funktion. Dann ist der Abschluss \bar{f} von f die existierende, größte, unterhalbstetige Funktion $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ mit

$$\bar{f}(x) \geq g(x) \quad \forall x \in \mathbb{R}^n$$

Bemerkung. Für eine Funktion $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ mit U abgeschlossen, ist die Unterhalbstetigkeit von f äquivalent zur Abgeschlossenheit vom Epigraph von f . Deshalb werden unterhalbstetige konvexe Funktionen auch abgeschlossene konvexe Funktionen genannt.

Definition 2.1.9. Sei $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine reelle Funktion. Dann ist die Legendre-Fenchel Transformierte f^* (kurz: LF-Transformierte) von f gegeben durch:

$$f^*(p) = \sup_{z \in U} \{p \cdot z - f(z)\} \quad \forall p \in U_{LF}(f) := \{p \in \mathbb{R}^n \mid \sup_{x \in U} \{p \cdot x - f(x)\} < \infty\}$$

Bemerkung. Man kann den Definitionsbereich auch auf \mathbb{R}^n erweitern, wenn man den Wertebereich auf $\mathbb{R}^n \cup \{+\infty\}$ erweitert.

Satz 2.1.10. Sei $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ konvex und unterhalbstetig. Dann gilt:

- (1) Die LF-Transformierte von f ist konvex und unterhalbstetig
- (2) $f(x) = \sup_{y \in U_{LF}(f)} \{x \cdot y - f^*(y)\} \quad \forall x \in X$
- (3) $f^* = \bar{f}^*$

Beweis. (1) Aus der Ungleichung

$$\begin{aligned} \sup_{z \in X} \langle tp + (1-t)q, z \rangle - f(z) &= \sup_{z \in X} t(p \cdot z - f(z)) + (1-t)(q \cdot z - f(z)) \\ &\leq t \sup_{z \in X} p \cdot z - f(z) + (1-t) \sup_{z \in X} (q \cdot z - f(z)) \end{aligned}$$

folgt direkt die Konvexität der Menge $U_{LF}(f)$ und die der Abbildung f^* .

Sei nun $p \in X$ beliebig und $p_n \in X$ eine Folge mit $p_n \rightarrow p$. Zeige $\lim_{n \rightarrow \infty} f^*(p_n) \stackrel{!}{\geq} f^*(p)$
 $[\Leftrightarrow \forall \alpha \in \mathbb{R} \text{ gilt } (\alpha \geq \lim_{n \rightarrow \infty} f^*(p_n) \Rightarrow \alpha \geq f^*(p))]$:

Sei also $\alpha \geq \lim_{n \rightarrow \infty} f^*(p_n)$ beliebig. Dann folgt:

$$\begin{aligned} \alpha &\geq \lim_{n \rightarrow \infty} f^*(p_n) = \lim_{n \rightarrow \infty} \sup_{z \in X} z \cdot p_n - f(z) \\ &\geq \lim_{n \rightarrow \infty} z \cdot p_n - f(z) \quad \forall z \in X \\ &= z \cdot p - f(z) \quad \forall z \in X \\ \Rightarrow \quad \alpha &\geq \sup_{z \in X} z \cdot p - f(z) = f^*(p) \end{aligned}$$

(2) Mit Satz 2.1.6 erhält man:

$$\begin{aligned} f(x) &= \sup\{\alpha(x) \mid \alpha \text{ affin}, \alpha(z) \leq f(z), \forall z \in X\} \\ &= \sup\{r + x \cdot y \mid r \in \mathbb{R}, y \in \mathbb{R}^n, r + z \cdot y \leq f(z), \forall z \in X\} \\ &= \sup\{r + x \cdot y \mid r \in \mathbb{R}, y \in \mathbb{R}^n, z \cdot y - f(z) \leq -r, \forall z \in X\} \\ &= \sup\{r + x \cdot y \mid r \in \mathbb{R}, y \in \mathbb{R}^n, \sup_{z \in X} (z \cdot y - f(z)) \leq -r\} \\ &= \sup\{r + x \cdot y \mid r \in \mathbb{R}, y \in U_{LF}(f), f^*(y) \leq -r\} \\ &= \sup\{x \cdot y - f^*(y) \mid y \in U_{LF}(f)\} \end{aligned}$$

(3) Man kann die LF-Transformation auch schreiben als $f^*(p) = \sup_{(x,\mu) \in \text{epi } f} x \cdot p - \mu$. Damit folgt:

$$f^*(p) = \sup_{(x,\mu) \in \text{epi } f} x \cdot p - \mu = \sup_{(x,\mu) \in \overline{\text{epi } f}} x \cdot p - \mu = \sup_{(x,\mu) \in \text{epi } \bar{f}} x \cdot p - \mu = (\bar{f})^*$$

□

Bemerkung. Der Beweis kann analog mit dem alternativen Definitionsbereich von f^* durchgeführt werden.

Definition 2.1.11. Eine Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ heißt positiv homogen (vom Grad 1),

falls für alle $x \in \mathbb{R}^n$ und alle $\lambda > 0$ gilt:

$$f(\lambda x) = \lambda f(x)$$

Definition 2.1.12. Sei $C \subset \mathbb{R}^n$ eine nicht-leere, konvexe Menge. Dann ist die Stützfunktion χ_C^* von C :

$$\chi_C^* : \mathbb{R}^n \rightarrow \mathbb{R}, \quad v \mapsto \sup \{x \cdot v \mid x \in C\}$$

Satz 2.1.13. 1. Sei C eine nicht-leere, abgeschlossene, konvexe Menge. Dann sind die Indikatorfunktion χ_C^∞ (mit den Werten 0 und $+\infty$) und die Stützfunktion χ_C^* jeweils die LF-Transformation des anderen.

2. Die Stützfunktionen der nicht-leeren konvexen Mengen sind die unterhalbstetigen, positiv homogenen, konvexen Funktionen.

Beweis. 1. Sei $\chi_C^\infty(x) = \begin{cases} 0 & \text{falls } x \in C \\ +\infty & \text{sonst} \end{cases}$ die Indikatorfunktion einer abgeschlossenen konvexen Menge C (mit den Werten 0 und $+\infty$) und χ_C^* die dazugehörige Stützfunktion. Dann gilt:

$$\begin{aligned} (\chi_C^\infty)^*(y) &= \sup_x x \cdot y - \chi_C^\infty(x) = \sup_{x \in C} x \cdot y = \chi_C^*(y) \\ (\chi_C^*)^*(x) &= \sup_y x \cdot y - \chi_C^*(y) = \sup_y x \cdot y - (\sup_z z \cdot y - \chi_C^\infty(z)) \\ &= \sup_y \inf_z \langle x - z, y \rangle + \chi_C^\infty(z) \stackrel{C \text{ abg.}}{=} \begin{cases} 0 & x \in C \\ +\infty & x \notin C \end{cases} = \chi_C^\infty(x) \end{aligned}$$

2.(a) Sei f^* unterhalbstetig, konvex und positiv homogen. Dann gilt

$$f^*(\lambda y) = \lambda f^*(y) \quad \forall y \wedge \forall \lambda > 0 \quad .$$

Definiere nun $f := (f^*)^*$. Dann gilt für alle $\lambda > 0$ und alle $x \in \mathbb{R}$:

$$\lambda f(x) = \lambda (\sup_y x \cdot y - f^*(y)) = \sup_y x \cdot (\lambda y) - \lambda f^*(y) = \sup_{\tilde{y}} x \cdot \tilde{y} - f^*(\tilde{y}) = f(x)$$

Also gilt $f(x) \in \{-\infty, 0, +\infty\}$ für alle $x \in \mathbb{R}$. Wegen der positiven Homogenität muss $f^*(0) = 0$ und damit $f > -\infty$ gelten. Also ist f eine Indikatorfunktion für die Menge $C = \{x \mid f(x) \leq 0\} = \{x \mid x \cdot y \leq f^*(y), \forall y\}$ und f^* wegen 1. und Satz 2.1.10 die dazugehörige

Stützfunktion.

(b) Sei nun C eine nicht-leere konvexe Menge und f die dazugehörige Indikatorfunktion. Dann gilt für alle $x \in \mathbb{R}^n$ und alle $\lambda > 0$: $f(x) = \lambda f(x)$. Die eindeutige Stützfunktion von C ist nach 1. die LF-Transformierte von f . Es folgt für $\lambda > 0$ und jedes $y \in \mathbb{R}^n$:

$$\lambda f^*(y) = \lambda(\sup_x x \cdot y - f(x)) = \sup_x x \cdot (\lambda y) - f(x) = f^*(\lambda y) \quad .$$

Mit Satz 2.1.10 folgt die eindeutige Stützfunktion f^* ist unterhalbstetig, konvex und positiv homogen. \square

2.2 Differenzierbarkeit konvexer Funktionen

Definition 2.2.1. Sei $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine Funktion. Dann ist die Fortsetzung \hat{f} von f definiert durch:

$$\hat{f} : \mathbb{R}^n \rightarrow \mathbb{R}, x \mapsto \begin{cases} f(x) & \text{falls } x \in U \\ \infty & \text{sonst} \end{cases}$$

Für die Fortsetzung gilt: f ist genau dann konvex, wenn \hat{f} konvex ist.

Definition 2.2.2. (1) Sei $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine Funktion. Ein Vektor $x^* \in \mathbb{R}^n$ heißt Subgradient (von f an der Stelle x), falls gilt:

$$\hat{f}(z) \geq f(x) + \langle x^*, z - x \rangle \quad , \forall z \in \mathbb{R}^n$$

(2) Die Menge aller Subgradienten (von f an der Stelle x) wird als Subdifferential bezeichnet und mit $\partial f(x)$ gekennzeichnet. Ebenso wird die Abbildung $\partial f : U \rightarrow \mathbb{P}(\mathbb{R}^n)$, $x \mapsto \partial f(x)$ als Subdifferential bezeichnet.

(3) Falls die Menge $\partial f(x)$ nicht leer ist, heißt die Funktion subdifferenzierbar an der Stelle x .

Korollar 2.2.3. 1. Die Ungleichung bedeutet, dass der Graph von $h(z) = f(x) + \langle x^*, z - x \rangle$ eine nicht vertikale Stützebene an f bildet.

2. Außerdem folgt direkt aus der Definition, dass das Subdifferential eine konvexe Menge ist:

$$x^*, y^* \in \partial f(x) \Rightarrow z^* := \lambda x^* + (1 - \lambda)y^* \in \partial f(x) \quad \forall \lambda \in [0, 1] \quad .$$

3. Für eine konvexe Funktion f ist eine notwendige Bedingung für ein Minimum an der Stelle x somit $0 \in \partial f(x)$.

Beweis. 2. Sei $\lambda \in [0, 1]$ beliebig. Dann gilt:

$$\begin{aligned} f(x) + \langle z^*, z - x \rangle &= f(x) + \langle \lambda x^* + (1 - \lambda)y^*, z - x \rangle \\ &= \lambda(f(x) + \langle x^*, z - x \rangle) + (1 - \lambda)(f(x) + \langle y^*, z - x \rangle) \\ &\leq \hat{f}(z) \end{aligned}$$

3. Jedes lokale Minimum einer konvexen Funktion f ist nach Satz 2.1.5 auch ein globales Minimum. Mit dem Satz folgt also:

$$\begin{aligned} f \text{ hat lokales Minimum an der Stelle } x &\Leftrightarrow f \text{ hat globales Minimum an der Stelle } x \\ &\Leftrightarrow f(x) \leq f(z) \quad \forall z \\ &\Leftrightarrow 0 \in \partial f(x) \end{aligned}$$

□

Definition 2.2.4. Sei $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine Funktion. Dann ist die einseitige Richtungsableitung in Richtung v :

$$\partial_v^+ f : U \rightarrow \mathbb{R} \cup \{+\infty\}, \quad \partial_v^+ f(x) = \lim_{t \downarrow 0} \frac{\hat{f}(x + tv) - f(x)}{t}$$

Die Fortsetzung nimmt nur reelle Werte oder $+\infty$ an. Da der Wertebereich vergrößert wurde existiert die Richtungsableitung somit immer und ist niemals $-\infty$.

Bemerkung. Falls $\partial_v^+ f(x) = \partial_{-v}^+ f(x)$ gilt, so entspricht die einseitige Richtungsableitung der gewöhnlichen Richtungsableitung.

Satz 2.2.5. Sei $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine konvexe Funktion (und endlich an der Stelle x). Dann ist der Differenzenquotient $\frac{\hat{f}(x+tv)-f(x)}{t}$ als von t abhängige Funktion für alle $v \in \mathbb{R}^n$ nicht fallend, sodass $\partial_v^+ f(x)$ existiert und $\partial_v^+ f(x) = \inf_{t>0} \frac{\hat{f}(x+tv)-f(x)}{t}$ erfüllt. Als Abbildung in v ist die einseitige Richtungsableitung konvex und positiv homogen ist.

Beweis. Analog zu Satz 2.1.6 kann man, weil x endlich ist eine affine Stützfunktion definieren. Diese ist definiert durch die Gleichung (2.1). Für die Funktion $\alpha(z) = f(x) +$

$\langle x^*, x - z \rangle$ gilt: α ist affin und $\alpha(z) \leq f(z) \forall z \in X$. Also folgt:

$$\begin{aligned} & \alpha(x + tv) \leq \hat{f}(x + tv) \\ \Leftrightarrow & \frac{\alpha(x + tv) - f(x)}{t} \leq \frac{\hat{f}(x + tv) - f(x)}{t} \\ \Leftrightarrow & -x^* \cdot v \leq \frac{\hat{f}(x + tv) - f(x)}{t} \end{aligned}$$

Also ist der Differenzenquotient von $\partial_v^+ f(x)$ nach unten beschränkt. Sei nun $v \in \mathbb{R}^n$ beliebig. Definiere $g_v : \mathbb{R}^+ \rightarrow \mathbb{R} \cup \{\infty\}$, $t \mapsto \frac{\hat{f}(x+tv) - f(x)}{t}$ und wähle $s > 0$ beliebig. Dann folgt für alle $t \in (0, 1)$:

$$\begin{aligned} & \hat{f}(x + tsv) \leq t\hat{f}(x + sv) + (1 - t)\hat{f}(x) \\ \Leftrightarrow & \hat{f}(x + tsv) - f(x) \leq t\hat{f}(x + sv) - tf(x) \\ \Leftrightarrow & \frac{\hat{f}(x + tsv) - f(x)}{ts} \leq \frac{\hat{f}(x + sv) - f(x)}{s} \\ \Leftrightarrow & g_v(ts) \stackrel{!}{\leq} g_v(s) \end{aligned}$$

Also ist die Funktion g_v nicht fallend und $\partial_v^+ f(x) = \inf_{t>0} \frac{\hat{f}(x+tv) - f(x)}{t}$. Man sieht in dieser Schreibweise, die Funktion $\partial_v^+ f(x)(v)$ ist wegen der Konvexität von f selbst konvex und außerdem positiv homogen:

$$\partial_{\lambda v}^+ f(x) = \inf_{t>0} \frac{\hat{f}(x + t\lambda v) - f(x)}{t} = \inf_{\tilde{t}>0} \lambda \frac{\hat{f}(x + \tilde{t}v) - f(x)}{\tilde{t}} = \lambda \partial_v^+ f(x)$$

□

Korollar 2.2.6. Sei $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine konvexe Funktion (und endlich an der Stelle x). Dann ist x^* genau dann ein Subgradient von f an der Stelle x , wenn

$$\partial_v^+ f(x) \geq x^* \cdot v \quad \forall v \in \mathbb{R}^n$$

Mit dieser Beschreibung des Subdifferentials sieht man außerdem, dass der Abschluss von $\partial_v^+ f(x)(v)$ die Stützfunktion des Subdifferentials ist.

Beweis. Aus Satz 2.2.5 folgt die Gleichheit $\partial_v^+ f(x) = \inf_{t>0} \frac{\hat{f}(x+tv) - f(x)}{t}$. Definiert man nun $z := x + tv$ folgt die erste Behauptung direkt aus der Definition des Subdifferentials. Aus Satz 2.2.5 folgt außerdem, dass $\partial_v^+ f(x)(v)$ eine konvexe, positiv homogene Funktion ist. Damit erfüllt der Abschluss die Voraussetzungen von Satz 2.1.13 und die Behauptung

ist bewiesen wenn die Menge C dem Subdifferential entspricht. Nach dem 1. Teil dieses Korollars und Satz 2.1.10 ist $\partial f(x) = \{x^* | x^* \cdot v \leq \partial_v^+ f(x), \forall v\} = \{x^* | (\partial^+ f(x))^*(x^*) \leq 0\} = \{x^* | \overline{(\partial^+ f(x))^*(x^*)} \leq 0\}$. Das ist nach dem Beweis von Satz 2.1.13 genau die Menge, für die der Abschluss der einseitigen Richtungsableitung die Stützfunktion ist. \square

Satz 2.2.7. *Sei $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\pm\infty\}$ eine konvexe Funktion und endlich an der Stelle x (Beachte dabei den Wertebereich von $f!$). Dann ist f genau dann subdifferenzierbar an der Stelle x , wenn $f(y) > -\infty, \forall y \in \mathbb{R}^n$.*

Beweis. I. Sei f subdifferenzierbar an der Stelle x . Dann gilt die Ungleichung $f(z) \geq f(x) + \langle x^*, z - x \rangle, \forall z \in \mathbb{R}^n$. Also ist $f(z) > -\infty$.

II. Sei f nicht subdifferenzierbar an der Stelle x . Dann ist die Menge $\partial f(x)$ leer. Also gilt

$$\forall x^* \in \mathbb{R}^n \exists z \in \mathbb{R}^n, \text{ sodass gilt } f(z) < f(x) + \langle x^*, z - x \rangle$$

Mit anderen Worten, es existiert keine Stützebene am Punkt $(x, f(x)) \in \text{epi } f$. Da f aber konvex ist, existiert eine Stützebene an den Graphen $\text{epi } f|_{\{x | f(x) \in \mathbb{R}\}}$. Der Rand dieser Stützebene kann durch eine affine Funktion α beschrieben werden (analog zum Beweis von Satz 2.1.6). Also gilt:

$$\alpha(z) = f(x) + \langle x^*, z - x \rangle \leq f(z), \forall z \in \{x | f(x) \in \mathbb{R}\}$$

Würde $f(z) = +\infty \forall z \in \{x | f(x) \notin \mathbb{R}\}$ gelten, wäre α auch eine Stützfunktion von f ohne Einschränkung am Punkt $(x, f(x))$. Also muss es mindestens ein $z \in \{x | f(x) \notin \mathbb{R}\}$ geben, für das gilt $f(z) = -\infty$. \square

Satz 2.2.8 (Relation von Gradient und Subgradient). *Sei $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine konvexe Funktion. Dann ist f genau dann differenzierbar an der Stelle x , wenn der Subgradient von f an der Stelle x aus genau einem Vektor besteht.*

Beweis. I. Sei f differenzierbar an der Stelle x . Dann wissen wir aus den Grundlagen der Analysis, dass $\partial_v^+ f(x) = \nabla f(x) \cdot v$ gilt. Aus Korollar 2.2.6 folgt, dass x^* genau dann ein Subgradient von f an der Stelle x ist, wenn:

$$\nabla f(x) \cdot v = \partial_v^+ f(x) \geq x^* \cdot v, \forall v \in \mathbb{R}^n$$

Da für $v = x^* - \nabla f(x)$ dann aber gilt $0 \geq |x^* - \nabla f(x)|^2$, folgt $\nabla f(x)$ ist der eindeutige Subgradient von f an der Stelle x .

II. f habe einen eindeutigen Subgradienten an der Stelle x . Definiere $g(v) := f(x + v) - f(x) - x^* \cdot v$. Dann ist g eine konvexe Funktion. Es ist zu zeigen, dass $\lim_{v \rightarrow 0} \frac{g(v)}{|v|} = 0$ gilt.

$$\begin{aligned} \partial_v^+ g(0) &= \lim_{t \downarrow 0} \frac{g(tv) - g(0)}{t} \\ &= \lim_{t \downarrow 0} \frac{f(x + tv) - f(x) - x^* \cdot tv}{t} \\ &= \partial_v^+ f(x) - x^* \cdot v \end{aligned}$$

Da f konvex ist und nach Voraussetzung das eindeutige Subdifferential $\{x^*\} = \partial f(x)$ hat, gilt für das Subdifferential der konvexen Funktion g :

$$\begin{aligned} y^* \in \partial g(0) &\Leftrightarrow \partial_v^+ g(0) \geq y^* \cdot v, \forall v \in \mathbb{R}^n \\ &\Leftrightarrow \partial_v^+ f(x) \geq \langle y^* + x^*, v \rangle, \forall v \in \mathbb{R}^n \\ &\stackrel{n.V.}{\Leftrightarrow} y^* = 0 \end{aligned}$$

Der Abschluss der einseitigen Richtungsableitung von g ist nach Satz 2.2.5 die Stützfunktion des Subdifferentials. Die Stützfunktion von g ist konstant 0, weil $\partial g(0) = \{0\}$ das eindeutige Subdifferential an der Stelle 0 ist. Die einseitige Richtungsableitung kann sich nur auf dem Rand der Menge $\{v \in \mathbb{R}^n | \partial_v^+ g(0) < \infty\} = \mathbb{R}^n$ von seinem Abschluss unterscheiden. Es folgt:

$$0 = \partial_v^+ g(0) = \lim_{t \downarrow 0} \frac{g(0 + tv) - g(0)}{t} = \lim_{t \downarrow 0} \frac{g(tv)}{t} \quad \forall v$$

Jetzt ist bekannt, dass für jedes v und jede Folge $t_n \rightarrow 0$ der Quotient gegen 0 geht. Allerdings ist noch nicht klar, ob auch für jede Folge v_n mit $|v_n| \rightarrow 0$ der Quotient gegen 0 geht. Sei B die abgeschlossene Einheitskugel im \mathbb{R}^n und $\{e_1, \dots, e_m\}$ eine Menge an Punkten, dessen konvexe Hülle ($= \{\sum_{i=1}^m \alpha_i e_i | \sum_{i=1}^m \alpha_i = 1, \alpha_i > 0\}$) B enthält. (Wähle dafür zum Beispiele alle Ecken des Einheitswürfels. Für weitere Hinweise zur konvexen Hülle vgl. [Roc70] Part 1.2).

Sei nun $v \in B$ beliebig. Definiere $u = \frac{v}{|v|} \in B$. Dann existiert eine Konvexkombination mit der oben gewählten Menge: $u = \sum_{i=1}^m u_i e_i$ mit $\sum_{i=1}^m u_i = 1$. Nach Satz 2.2.5 ist der Differenzenquotient einer konvexen Funktion eine nicht fallende Funktion in t . Mit der Jensenschen Ungleichung folgt:

$$0 \leq \inf_{t > 0} \frac{g(tu)}{t} \leq \frac{g(tu)}{t} \leq \sum_{i=1}^m u_i \frac{g(te_i)}{t} \leq \max_i \frac{g(te_i)}{t} \quad \forall t > 0$$

Für jedes i ist bekannt, dass $\frac{g(te_i)}{t} \rightarrow 0$ für $t \rightarrow 0$. Also gilt das selbe auch für das Maximum $\max_i \frac{g(te_i)}{t}$. Das Maximum ist somit für jedes u eine Majorante, die gegen 0 strebt wenn t gegen 0 strebt. Daraus ergibt sich für $|v| \rightarrow 0$ mit $u = \frac{v}{|v|} \in B$:

$$0 \leq \frac{g(v)}{|v|} = \frac{g(|v|u)}{|v|} \leq \max_i \frac{g(|v|e_i)}{|v|} \rightarrow 0 \quad .$$

□

Lemma 2.2.9. *Seien $\lambda > 0$, $\beta, \tilde{u} \in \mathbb{R}^n$ und $g \in \mathbb{R}$ gegeben. Definiere $f(u) = \beta u - g$. Dann ist die Lösung für das folgende Minimierungsproblem:*

$$\operatorname{argmin}_u \frac{|u - \tilde{u}|^2}{2} + \lambda |\beta u - g|$$

gegeben durch:

$$u = \tilde{u} + \begin{cases} \lambda\beta & \text{falls } f(\tilde{u}) < -\lambda\beta^2 \\ -\lambda\beta & \text{falls } f(\tilde{u}) > \lambda\beta^2 \\ -\frac{f(\tilde{u})\beta}{|\beta|^2} & \text{falls } |f(\tilde{u})| < \lambda\beta^2 \end{cases}$$

Beweis. Das Lemma inklusive Beweis wurde sinngemäß aus der Dissertation von H. Dirks übernommen (vgl. [Dir15]).

Nach Korollar 2.2.3 muss für das konvexe Minimierungsproblem die notwendige Bedingung gelten:

$$0 \in (u - \tilde{u}) + \lambda \partial(|\beta u - g|)$$

1. Fall: $\beta u - g \neq 0$.

Dann ist das Subdifferential eindeutig bestimmt, weil auch die Betragsfunktion differenzierbar ist. Es folgt:

$$\begin{aligned} \partial(|\beta u - g|) &= \operatorname{sgn}(\beta u - g)\beta \\ \Rightarrow u &= \tilde{u} - \lambda \operatorname{sgn}(\beta u - g)\beta \end{aligned}$$

Die Bedingung für den 1. Fall kann man damit äquivalent umformen zu:

$$\begin{aligned} 0 &\geq \beta u - g = \beta(\tilde{u} - \lambda(\mp 1)\beta) - g \\ &= \beta\tilde{u} - g + \lambda(\pm 1)\beta^2 \\ \Leftrightarrow \mp \lambda\beta^2 &\geq f(\tilde{u}) \end{aligned}$$

2. Fall: $\beta u - g = 0$.

In diesem Fall ist das Subdifferential mit der Kettenregel und dem Subdifferential für den Betrag gegeben durch $\partial(|\beta u - g|) = [-\beta, \beta]$. Die notwendige Bedingung lässt sich mit $\partial \in \partial(|\beta u - g|)$ schreiben als:

$$0 = u - \tilde{u} + \lambda \partial \Leftrightarrow u = \tilde{u} - \lambda \partial \stackrel{\beta u = g}{\Leftrightarrow} g = \beta(\tilde{u} - \lambda \partial) \stackrel{\partial \in [-\beta, \beta]}{\Leftrightarrow} \lambda \partial = \frac{\beta(\beta \tilde{u}) - \beta g}{|\beta|^2}$$

Durch das Einfügen dieser Gleichung in die notwendige Bedingung lässt sich der Term passend darstellen:

$$u = \tilde{u} - \left(\frac{\beta(\beta \tilde{u} - g)}{|\beta|^2} \right) = \tilde{u} - \frac{f(\tilde{u})\beta}{|\beta|^2}$$

Für die Bedingung muss eine letzte Umformung gemacht werden. Da $\partial \in [-\beta, \beta]$ gilt:

$$\left| \frac{\beta(\beta \tilde{u})}{|\beta|^2 \lambda} - \frac{\beta g}{|\beta|^2 \lambda} \right| = |\partial| \leq |\beta| \Leftrightarrow |\beta \tilde{u} - g| \leq \lambda \beta^2 \Leftrightarrow |f(\tilde{u})| \leq \lambda \beta^2$$

□

2.3 Umformung in die Primal duale Formulierung

Definition 2.3.1 (Huber-Norm und Huber-Betrag). *Der Huber-Betrag $|\cdot|_\epsilon : \mathbb{R}^n \rightarrow \mathbb{R}$ ist wie folgt definiert:*

$$x \mapsto \begin{cases} \frac{|x|^2}{2\epsilon} & \text{falls } |x| \leq \epsilon \\ |x| - \frac{\epsilon}{2} & \text{falls } |x| > \epsilon \end{cases}$$

Darauf aufbauend wird die diskrete Huber-Norm $\|\cdot\|_\epsilon : \mathbb{R}^{M \times n} \rightarrow \mathbb{R}$ definiert als:

$$v \mapsto \sum_{i=1}^M |v_i|_\epsilon$$

Bemerkung. 1. Man kann analog zu dieser diskreten Huber-Norm auch eine kontinuierliche Version auf dem Raum der Funktionen definieren.

2. Die Abbildung $\|\cdot\|_\epsilon$ wird hier zwar als Norm bezeichnet, ist aber nicht für alle ϵ auch tatsächlich eine Norm. Der Begriff wurde gewählt, weil sie später analog zur L^1 - und L^2 -Norm in den Funktionalen verwendet wird.

3. Für $\epsilon = 0$ entspricht die Huber-Norm damit der L^1 -Norm (mit vektoriellem Betrag für

Vektorfelder).

Beispiel. Die Legendre-Fenchel-Transformierte des Huber-Betrags wird wie folgt bestimmt:

$$\lambda|p_d|_\epsilon^* = \sup_{q_d} \{q_d \cdot p_d - \lambda|q_d|_\epsilon\} = \hat{q}_d \cdot p_d - \lambda|\hat{q}_d|_\epsilon \quad (\text{falls das Maximum } \hat{q}_d \text{ existiert})$$

1. Fall $|p_d| \leq \lambda$

(a) $|\hat{q}_d| \leq \epsilon$:

$$\begin{aligned} \lambda|p_d|_\epsilon^* &= \sup_{|q_d| \leq \epsilon} \left\{ q_d \cdot p_d - \lambda \frac{|q_d|^2}{2\epsilon} \right\} \\ \Rightarrow 0 &= \frac{\partial}{\partial q_d} \left\{ q_d \cdot p_d - \lambda \frac{|q_d|^2}{2\epsilon} \right\} \\ &= p_d - \lambda \frac{q_d}{\epsilon} \quad \Rightarrow \quad q_d = \frac{\epsilon}{\lambda} p_d \\ \Rightarrow \lambda|p_d|_\epsilon^* &= \sup_{|q_d| \leq \epsilon} \left\{ \frac{\epsilon}{\lambda} |p_d|^2 - \frac{\epsilon}{2\lambda} |p_d|^2 \right\} \\ &= \frac{\epsilon}{2\lambda} |p_d|^2 \end{aligned}$$

(b) $|\hat{q}_d| > \epsilon$:

$$\begin{aligned} \lambda|p_d|_\epsilon^* &= \sup_{|q_d| > \epsilon} \left\{ q_d \cdot p_d - \lambda \left(|q_d| - \frac{\epsilon}{2} \right) \right\} \\ &= \sup_{|q_d| > \epsilon} \left\{ |q_d| |p_d| - \lambda \left(|q_d| - \frac{\epsilon}{2} \right) \right\} \\ B(|q_d|) &:= |q_d| |p_d| - \lambda \left(|q_d| - \frac{\epsilon}{2} \right) \\ B'(|q_d|) &= |p_d| - \lambda \leq 0 \quad \Rightarrow \quad |q_d| = \epsilon \\ \lambda|p_d|_\epsilon^* &= \epsilon |p_d| - \frac{\lambda\epsilon}{2} \end{aligned}$$

Durch den Vergleich beider Ergebnisse wird ersichtlich, dass man im Fall (b) kein Maximum (Supremum) erhält (Ausnahme $\epsilon = 0$):

$$\begin{aligned} \Leftrightarrow \quad \epsilon |p_d| - \frac{\lambda\epsilon}{2} &\leq \frac{\epsilon}{2\lambda} |p_d|^2 \\ \Leftrightarrow \quad \lambda\epsilon |p_d| &\leq \frac{\epsilon}{2} (|p_d|^2 + \lambda^2) \\ \Leftrightarrow \quad 2\lambda |p_d| &\leq |p_d|^2 + \lambda \\ \Leftrightarrow \quad 0 &\leq (|p_d| - \lambda)^2 \end{aligned}$$

2. Fall $|p_d| > \lambda$

Wähle nun $q_d = r p_d$ und $r = |q_d| > \epsilon$

$$\begin{aligned} \Rightarrow \quad \lambda|p_d|_\epsilon^* &\geq q_d \cdot p_d - \lambda \left(|q_d| - \frac{\epsilon}{2} \right) = r |p_d|^2 - \lambda \left(r |p_d| - \frac{\epsilon}{2} \right) \\ &= r \underbrace{(|p_d|^2 - \lambda |p_d|)}_{>0} + \frac{\lambda\epsilon}{2} \quad \xrightarrow{r=|q_d| \rightarrow \infty} \infty \end{aligned}$$

Mit $\chi_K^\infty = \begin{cases} 0 & \text{falls } x \in K \\ \infty & \text{falls } x \notin K \end{cases}$ erhält man insgesamt $\lambda|p_d|_\epsilon^* = \chi_{\{|p_d| \leq \lambda\}}^\infty + \frac{\epsilon|p_d|^2}{2\lambda}$.

Weiterhin erhält man damit direkt durch die Definition:

$$\lambda|q|_\epsilon^{**} = \sup_p \left\{ p \cdot q - \lambda|p|_\epsilon^* \right\} = \sup_{\{|p| \leq \lambda\}} \left\{ q \cdot p - \epsilon \frac{|p|^2}{2\lambda} \right\}$$

Nun ist der Huber-Betrag konvex und stetig. Also gilt $|\cdot|_\epsilon = |\cdot|_\epsilon^{**} = \sup_{\{|p| \leq \lambda\}} \left\{ q \cdot p - \epsilon \frac{|p|^2}{2\lambda} \right\}$.

Bemerkung. Analog zur Berechnung der LF-Transformation des Huber-Betrags kann man auch für den gewöhnlichen Betrag und damit für die L^1 -Norm mit den Fällen 1.(b) und 2. die LF-Transformation berechnen. Für $\epsilon = 0$ entspricht damit die Huber-Norm der L^1 -Norm. Es gilt:

$$|\cdot|^* = \chi_{\{|p| \leq 1\}}^\infty(\cdot)$$



(a) Mequon mit L^1 -Norm auf Gesamtvektor (b) Mequon mit vektoriellem Betrag
 Abbildung 2.2: Veranschaulichung des Unterschiedes der Optimierung mit L1-Norm als Gesamtvektor und L1-Norm mit vektoriellem Betrag im Optical Flow Problem des Bildes Mequon aus der Middlebury Database.

Man beachte, dass die L^1 -Norm hier nicht die Vektorfelder auf den Bildern als einen großen Vektor aus dem $\mathbb{R}^{M \times N \times 2}$ betrachtet, sondern für jedes Pixel der vektorielle Betrag aus dem \mathbb{R}^2 genutzt wird. Das wird deshalb gemacht, da man sonst die Hauptachsenrichtungen bevorzugen würde. Das würde in diesem Anwendungsfall aber keinen Sinn ergeben. Den Unterschied kann man in den farbcodierten Flussbildern aus Abbildung 2.2 gut erkennen.

2.4 Primal-Dual Algorithmus

In der Arbeit [CP10] von Chambolle und Pock wird ein sehr allgemeiner Ansatz vorgestellt um bestimmte Sattelpunktprobleme numerisch zu lösen. Die wichtigsten Erkenntnisse für die Anwendung werden hier zusammengefasst.

Definition 2.4.1. Sei $K : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ein stetiger linearer Operator. Dann ist die induzierte Operatornorm von K :

$$\|K\| = \max\{\|K(x)\| \mid x \in \mathbb{R}^n, \|x\| \leq 1\} < \infty$$

Definition 2.4.2. Sei $K : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ein stetiger, linearer Operator und $G : \mathbb{R}^n \rightarrow [0, \infty)$ und $F^* : \mathbb{R}^m \rightarrow [0, \infty)$ zwei konvexe, unterhalbstetige Funktionen. Dann ist das primal-duale Sattelpunktproblem gegeben durch:

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} K(x) \cdot y + G(x) - F^*(y)$$

Definition 2.4.3. Sei $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine subdifferenzierbare Funktion. Dann ist der Resolventenoperator $(\mathbf{1} + \tau \partial f)^{-1}$ von f definiert durch:

$$x = (\mathbf{1} + \tau \partial f)^{-1}(y) = \operatorname{argmin}_x \left\{ \frac{\|x - y\|^2}{2\tau} + f(x) \right\}$$

Bemerkung. Die wesentliche Eigenschaft die aus dieser Definition für den Beweis des Algorithmus folgt, ist die notwendige Eigenschaft für das Minimum $\frac{y-x}{\tau} \in \partial f(x)$.

Beispiel. Sei $F^*((p, q_1, \dots, q_n)) = F_p^*(p) + \sum_{i=1}^n F_{q_i}^*(q_i)$ mit $F_p^*(p) = \frac{\epsilon}{2\mu} \|p\|^2 + \chi_{\{|p| \leq \mu\}}^\infty$ und $F_{q_i}^*(q_i) = q_i \cdot f_i + \frac{\delta}{2} \|q_i\|^2 + \chi_{\{|q_i| \leq 1\}}^\infty$ und den konstanten Werten $\epsilon, \delta, \mu > 0$ und $f_i \in C(\Omega)$. Dann bestimmt man den Resolventenoperator von F^* wie folgt:

$$\begin{aligned} (\mathbf{1} + \sigma \partial F^*)^{-1}(z) &= \operatorname{argmin}_{pq} \left(\frac{\|pq - z\|^2}{2\sigma} + F^*(pq) \right) \\ &= \operatorname{argmin}_{pq} \left(\frac{\|p - z_0\|^2 + \sum_{i=1}^n \|q_i - z_i\|^2}{2\sigma} + F_p^*(p) + \sum_{i=1}^n F_{q_i}^*(q_i) \right) \\ &= \operatorname{argmin}_p \left(\frac{\|p - z_0\|^2}{2\sigma} + F_p^*(p) \right) + \sum_{i=1}^n \operatorname{argmin}_{q_i} \left(\frac{\|q_i - z_i\|^2}{2\sigma} + F_{q_i}^*(q_i) \right) \\ &= (\mathbf{1} + \sigma \partial F_p^*)^{-1}(z_0) + \sum_{i=1}^n (\mathbf{1} + \sigma \partial F_{q_i}^*)^{-1}(z_i) \end{aligned}$$

Also müssen die Abbildungen $(\mathbf{1} + \sigma \partial F_p^*)^{-1}$ und $(\mathbf{1} + \sigma \partial F_{q_i}^*)^{-1}$ berechnet werden. Da die Funktionen bis auf Konstanten gleich sind, reicht es eine Funktion beispielhaft zu berechnen. Das gilt auch für die Funktion F_p^* , mit der Konstante $f_0 \equiv 0$:

$$\begin{aligned}
 (\mathbf{1} + \sigma \partial F_{q_i}^*)^{-1}(z_i) &= \operatorname{argmin}_{q_i} \frac{\|q_i - z_i\|^2}{2\sigma} + q_i \cdot f_i + \frac{\delta}{2} \|q_i\|^2 + \chi_{\{|q_i| \leq 1\}}^\infty \\
 &= \operatorname{argmin}_{q_i} \frac{\|q_i - z_i\|^2}{2\sigma} + \left\| \sqrt{\frac{\delta}{2}} q_i + \sqrt{\frac{1}{2\delta}} f_i \right\|^2 - \frac{1}{2\delta} \|f_i\|^2 + \chi_{\{|q_i| \leq 1\}}^\infty \\
 &\stackrel{\text{s.u.}}{=} \prod_{\{|q_i| \leq 1\}} \left(\operatorname{argmin}_{q_i} \frac{\|q_i - z_i\|^2}{2\sigma} + \left\| \sqrt{\frac{\delta}{2}} q_i + \sqrt{\frac{1}{2\delta}} f_i \right\|^2 - \frac{1}{2\delta} \|f_i\|^2 \right) \\
 &= \prod_{\{|q_i| \leq 1\}} \left(\operatorname{argmin}_{q_i} \frac{\|q_i - z_i\|^2}{2\sigma} + q_i \cdot f_i + \frac{\delta}{2} \|q_i\|^2 \right)
 \end{aligned}$$

Wie in der oberen Gleichung zu sehen, ist die zu minimierende Funktion quadratisch. Die Indikatorfunktion $\chi_{\{|q_i| \leq 1\}}^\infty$ fordert, dass das Minimum in der abgeschlossenen Menge $C = \{|q_i| \leq 1\}$ angenommen werden muss und ist somit eine konvexe Nebenbedingung für die Optimierung. Diese wirken sich bei quadratischen Funktionen durch Projektionen aus. Für das Minimierungsproblem innerhalb der Projektion folgt die notwendige Bedingung:

$$\begin{aligned}
 0 &= \frac{(q_i - z_i)}{\sigma} + f_i + \delta q_i \\
 \Leftrightarrow & \quad q_i = \frac{z_i - \sigma f_i}{1 + \sigma \delta} \\
 \Rightarrow & \quad (\mathbf{1} + \sigma \partial F_{q_i}^*)^{-1}(z_i) = \prod_{\{|q_i| \leq 1\}} \left(\frac{z_i - \sigma f_i}{1 + \sigma \delta} \right)
 \end{aligned}$$

Definition 2.4.4. Seien $\tau, \sigma > 0$, $\theta \in [0, 1]$, $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m$ und setze $\bar{x}^0 = x^0$. Dann wird der Primal-Dual Algorithmus durch folgende Iterationsschritte definiert:

$$\begin{aligned}
 y^{n+1} &= (\mathbf{1} + \sigma \partial F^*)^{-1}(y^n + \sigma K \bar{x}^n) \\
 x^{n+1} &= (\mathbf{1} + \tau \partial G)^{-1}(x^n - \tau K^T y^{n+1}) \\
 \bar{x}^{n+1} &= x^{n+1} + \theta(x^{n+1} - x^n)
 \end{aligned}$$

Satz 2.4.5. Sei $L = \|K\|$ und nehme an, dass ein Sattelpunkt $(\hat{x}, \hat{y}) \in \mathbb{R}^n \times \mathbb{R}^m$ existiert. Wählt man $\theta = 1$ und $\tau \sigma L^2 < 1$, so konvergiert der primal-dual Algorithmus gegen einen Sattelpunkt $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^m$.

Beweis. Für einen genauen Beweis siehe [CP10] Kapitel 3.1.1 \square

Definition 2.4.6. Eine Funktion $G : \mathbb{R}^n \rightarrow [0, \infty)$ heißt gleichmäßig konvex, falls ein $\gamma > 0$ existiert, sodass für alle x mit $G(x) < \infty$ gilt:

$$G(y) \geq G(x) + \langle p, y - x \rangle + \frac{\gamma}{2} |y - x|^2, \quad \forall p \in \partial G(x), y \in \mathbb{R}^n$$

Definition 2.4.7. Seien $\tau_0, \sigma_0 > 0$, $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m$ und $\bar{x}^0 = x^0$ gegeben. Dann wird der adaptive Primal-Dual Algorithmus durch folgende Iterationsschritte definiert:

$$\begin{aligned} y^{n+1} &= (\mathbf{1} + \sigma_n \partial F^*)^{-1}(y^n + \sigma_n K \bar{x}^n) \\ x^{n+1} &= (\mathbf{1} + \tau_n \partial G)^{-1}(x^n - \tau_n K^T y^{n+1}) \\ \theta_n &= 1/\sqrt{1 + 2\gamma\tau_n} && (\text{oder } = 1/\sqrt{1 + 2\gamma\sigma_n}) \\ \tau_{n+1} &= \theta_n \tau_n && (\text{oder } = \tau_n/\theta_n) \\ \sigma_{n+1} &= \sigma_n/\theta_n && (\text{oder } = \theta_n \sigma_n) \\ \bar{x}^{n+1} &= x^{n+1} + \theta_n(x^{n+1} - x^n) \end{aligned}$$

Satz 2.4.8. Sei $L = \|K\|$ und G (oder F^*) gleichmäßig konvex mit Konstante γ . Wählt man $\tau_0 > 0$, $\sigma_0 = \frac{1}{\tau_0 L^2}$ und (x^n, y^n) wie im adaptiven Primal-Dual Algorithmus, so existiert für jedes $\delta > 0$ ein N_0 , sodass für alle $N \geq N_0$ und einen Sattelpunkt (x^*, y^*) des Problems gilt:

$$|x^* - x^N|^2 \leq \frac{1 + \delta}{N^2} \left(\frac{|x^* - x_0|^2}{\gamma^2 \tau_0^2} + \frac{L^2}{\gamma^2} |y^* - y_0|^2 \right)$$

Beweis. Für einen genauen Beweis siehe [CP10] Kapitel 5.1 \square

Bemerkung. Das interessante an dem adaptiven Primal-Dual Verfahren ist, dass die Konvergenz gegen das gesuchte Minimum in der primalen Variable eine Konvergenzrate von $\mathcal{O}(\frac{1}{N^2})$ hat während das nicht adaptive Verfahren nur in $\mathcal{O}(\frac{1}{N})$ konvergiert (vgl. [CP10] Kapitel 5.1).

Bemerkung. Um die Anzahl der Iterationsschritte nicht bestimmen zu müssen wäre ein geeignetes Abbruchkriterium für den Primal-Dual Algorithmus wünschenswert. Es gibt die Möglichkeit entweder über das primale und duale Residuum (vgl. [GLY⁺13]) oder über die primal-duale Lücke ein Abbruchkriterium zu definieren. Beide Umsetzungen haben Vor- und Nachteile. Das Abbruchkriterium mit den Residuen hat den Nachteil, dass zwar eine Nähe der Argumente zum Sattelpunkt sichergestellt werden kann, aber die Energie nicht dadurch beschränkt wird. Die Möglichkeit mit der primal-dualen Lücke hat den Nachteil,

dass in vielen Fällen die Indikatorfunktionen nur die Aussage zulassen ob das aktuelle Argument der Sattelpunkt ist oder nicht. Deshalb wurden die Abbruchkriterien zwar kurz getestet, für die Evaluation der Algorithmen habe ich mich jedoch gegen die Nutzung eines Abbruchkriteriums entschieden.

3 Das Optical Flow Problem

3.1 Motivation

Ein wichtiges Teilproblem bei der Berechnung der Superresolution Bilder ist die Berechnung der Überlagerung von Eingabebildern. Dabei überlagert der Warping-Operator ein ausgewähltes Eingabebild, Referenzbild genannt, jeweils mit allen anderen Eingabebildern. Es wird also pro Eingabebild ein Warping-Operator berechnet. Dabei können einzelne Bereiche, vor allem Vordergrund und Hintergrund, unabhängig voneinander bewegt werden. So versucht man eine Bewegung, also eine Art Fluss (Optical Flow) von einem Bild zum nächsten zu erhalten. Ziel ist es, für jedes einzelne Pixel im Referenzbild zusätzliche Informationen zu gewinnen. Das Problem muss sehr präzise berechnet, da der Informationsgewinn im Vergleich zu den Eingabebildern auf Subpixelebene stattfindet. Deshalb untersuche ich in dieser Arbeit, welche Ansätze zur Berechnung des Optical Flow Problems für die Lösung des Superresolution Problems die besten Ergebnisse erzielen. Bei dieser Berechnung gibt es generell leicht unterschiedliche Vorgehensweisen. Es besteht beispielsweise die Möglichkeit, entweder den Fluss von einem Referenzbild zu den übrigen Eingabebildern zu berechnen, oder den Fluss von einem Bild zum nächsten zu ermitteln und dann die Ergebnisse auf das Referenzbild zu übertragen. Erwähnenswert ist dabei vor allem, dass aufgrund der präzisen Überlagerung der Bilder, besonders gute Ergebnisse für das Superresolution Problem erzielt werden, wenn die Kamera leichten Bewegungen ausgesetzt ist. Die leichten Bewegungen erhöhen den Informationsgehalt auf Subpixelebene. Handyvideos eignen sich deshalb gut für die Optimierung mit dem Superresolution Ansatz.

3.2 Mathematische Umsetzung

Ein (schwarz-weiß) Bild ist mathematisch betrachtet eine Abbildung von einem Gebiet $\Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$. In den meisten Fällen geht man von einem rechteckigen Bild aus, und kann, aufgrund der Projektionen zwischen einem beliebigen Rechteck und $[0, 1]^2$, Ω als $[0, 1]^2$ annehmen. In der Realität ist das gespeicherte Bild jedoch nicht kontinuierlich, sondern diskret. Für ein diskretes Bild kann die Ableitung immer definiert werden. Deshalb nehme ich für das kontinuierliche Bild an, dass es differenzierbar ist. Mit anderen Worten jedes Bild I ist eine Abbildung aus dem Raum $C(\Omega)$.

Beim Optical Flow Problem wird ein Vektorfeld $v : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^2 (v \in C(\Omega)^2)$ gesucht, sodass für zwei Bilder mit einer kleinen Bewegung des Motivs folgende Beziehung gilt: $I_1(x + v(x)) = I_2(x) \forall x \in \Omega$. Als Minimierungsproblem wird es wie folgt aufgestellt (vgl. [FW05]):

Definition 3.2.1 (Allgemeines Optical Flow Problem). *Das allgemeine Optical Flow Problem wird durch folgendes Minimierungsproblem definiert:*

$$\min_v \left\{ \int_{\Omega} |I_R(x+v(x)) - I_T(x)| dx \right\} \quad (3.1)$$

$$\text{mit} \quad v(x) = \begin{pmatrix} v_1(x) \\ v_2(x) \end{pmatrix} : \text{Verschiebungsvektorfeld} \quad (3.2)$$

$$I_R, I_T : \text{Bild}$$

Da der Term in diesem Funktional nicht konvex ist, wird er durch eine Taylorentwicklung erster Ordnung vereinfacht. Zusätzlich nimmt man an, dass ein passendes Vektorfeld v existiert, sodass $I_R(x+v(x)) = I_T(x)$ gilt $\forall x \in \Omega$. Daraus ergibt sich:

$$\begin{aligned} I_T(x) &= I_R(x + v(x)) = I_R(x) + v(x) \cdot \nabla I_R(x) + \mathcal{O}(|v(x)|^2) \\ \Rightarrow \quad 0 &\approx v(x) \cdot \nabla I_R(x) + I_R(x) - I_T(x) \end{aligned} \quad (3.3)$$

Die Gleichung (3.3) wird in der Literatur als Optical Flow Constraint Equation bezeichnet. Alternativ dazu kann man auch die Gleichung

$$0 = v(x) \cdot \nabla I_T(x) + I_R(x) - I_T(x)$$

lösen. Das zu lösende ursprüngliche Problem wäre dann $I_R(x) = I_T(x - v(x))$. Der Unterschied in dieser Formulierung liegt darin, dass man nicht das Referenzbild auf das

Zielbild verschiebt, sondern umgekehrt. Im Bezug auf den Warping-Operator würde man dann nicht $I_R(x + v(x)) = \mathbf{W}(I_R, v)$, sondern $I_R \circ (x - v(x))^{-1} = \mathbf{W}(I_R, v)$ bilden. Da $(x - v(x))$ im Allgemeinen (zum Beispiel bei Überdeckungen in I_t) nicht invertierbar ist, kann man die Verschiebung von I_R nach I_T damit theoretisch nicht realisieren. Bedenkt man jedoch, dass das umgekehrte Problem auch nicht immer lösbar ist (zum Beispiel bei Überdeckungen in I_R), kann man diesen Umstand auch ignorieren und für die Inverse $(x - v(x))^{-1} = x + v(x)$ annehmen. Dadurch wird der Warping-Operator bei beiden Ansätzen gleich berechnet. Vergleicht man die beiden Ansätze, stellt man fest, dass keiner der beiden Ansätze dem anderen generell überlegen ist. Vielmehr hängt die Qualität des Ergebnisses davon ab, in welchem Bild die wichtigeren Details (zum Beispiel durch Überdeckungen) fehlen. Da der erste Ansatz, den Fluss vom Referenzbild zum Zielbild berechnet und das Referenzbild auf das Zielbild verschoben werden soll, wird dieser direkte Weg für unser Problem ausgewählt.

Das zu lösende Problem mit ersetzttem Datenterm lautet demnach wie folgt:

Definition 3.2.2 (Allgemeines Optical Flow Problem mit Flow Constraint Equation).
Das allgemeine Optical Flow Problem mit Flow Constraint Equation wird durch folgendes konvexes Minimierungsproblem definiert:

$$\min_v \left\{ \int_{\Omega} |\rho(v)| dx \right\} \tag{3.4}$$

mit $v = \begin{pmatrix} v_1(x) \\ v_2(x) \end{pmatrix} : \text{Verschiebungsvektorfeld}$

$$\rho(v) = v(x) \cdot \nabla I_R(x) + I_R(x) - I_T(x) : \text{Flow Constraint Equation}$$

Da man bei diesem Problem für jedes Pixel des Bildes nur eine Gleichung erhält, ist die Lösung für v nicht eindeutig. Dieses Problem wird in der Literatur als aperture problem (dt. Blendenproblem) bezeichnet. Der Grund hierfür ist, dass der Mensch beim Blick durch eine Blende, genau wie das modellierte Problem auch, nur die Bewegungen tangential zum Bildgradienten wahrnehmen kann.



Abbildung 3.1: Veranschaulichung Aperture Problem

Die orthogonale Komponente der Bewegung kann hingegen nicht wahrgenommen werden. In Abbildung 3.1 wird veranschaulicht, wie sich ein Bild zu einem anderen Bild verändert. Die eingezeichneten Rechtecke verdeutlichen welche Bewegung das Bild bei diesem Ergebnis gemacht haben könnte. Würde man die Bilder ohne Rahmen betrachten, so könnte man die beiden rechten Bilder nicht unterscheiden, obwohl unterschiedliche Bewegungen stattgefunden haben.

Um das Blendenproblem zu lösen, wird ein Regularisierungsterm eingefügt. Damit sieht das Problem dann wie folgt aus:

Definition 3.2.3 (Optical Flow Problem). *Das Optical Flow Problem (mit Huber - Regularisierung) wird durch folgendes konvexes Minimierungsproblem definiert:*

$$\min_v \left\{ \int_{\Omega} \sum_{d=1}^2 |\nabla v_d|_{\epsilon} + \lambda |\rho(v)| dx \right\} \quad (3.5)$$

mit

$$v = \begin{pmatrix} v_1(x) \\ v_2(x) \end{pmatrix} : \text{Verschiebungsvektorfeld}$$

$$\rho(v) : \text{Flow Constraint Equation}$$

$$|\cdot|_{\epsilon} : \text{Huber-Betrag}$$

Nachdem das Modell für das Optical Flow Problem soweit aufgestellt wurde, muss der Datenterm $|\rho(v)|$ genauer untersucht werden. Dabei fällt auf, dass die Umsetzung durch die Flow Constraint Equation einen Nachteil mit sich bringt. Das minimierende Vektorfeld muss in Richtung des Bildgradienten nah an dem Vektorfeld

$$v(x) = \frac{\nabla I_T}{|\nabla I_T|^2}(x)(-\partial_t I_T(x))$$

sein. Im konkreten Fall bei Bildern $I_T \subset [0, 1]^{M \times N}$ ist die Lösung an stärkeren Kanten $|\nabla I_T(x)| \gg 0$ dadurch beschränkt. Es können also keine größeren Bewegungen $|v(x)| \gg 1$ die minimale Lösung sein. Um diese größeren Bewegungen dennoch umsetzen zu können, und das Problem der Lokalität zu umgehen, muss das Problem für eine gute Lösung bei der Berechnung wiederholt angepasst werden. Deshalb wird das Referenzbild iterativ verschoben, wodurch das Optical Flow Problem wiederholt von sich stückweise annähernden Bildern gelöst wird. Mathematisch ausgedrückt bedeutet das in der allgemeinen Formulierung:

$$\min_v \left\{ \int_{\Omega} |I_R(x + \sum_k v^k(x) + v(x)) - I_T(x)| dx \right\}$$

Motiviert wird die Idee auch durch den Fehler bei der Taylor-Approximation. Sei $v = v_1 + v_2$. Dann gilt:

$$\begin{aligned}
 I_R(x + v_1 + v_2) &= I_R(x + v_1) + v_2 \cdot \nabla I_R(x + v_1) + \mathcal{O}(|v_2|^2) \\
 &= I_R(x) + v_1 \cdot \nabla I_R(x) + v_2 \cdot \nabla I_R(x + v_1) + \mathcal{O}(|v_1|^2) + \mathcal{O}(|v_2|^2) \\
 I_R(x + v_1 + v_2) &= I_R(x + v) = I_R(x) + v \cdot \nabla I_R(x) + \mathcal{O}(|v|^2) \\
 &= I_R(x) + v \cdot \nabla I_R(x) + \mathcal{O}(|v_1|^2 + |v_2|^2 + v_1 \cdot v_2)
 \end{aligned}$$

Geht man davon aus, dass v_1 und v_2 ungefähr in die gleiche Richtung gehen (also zumindest nicht in die entgegengesetzte), so ist der Approximationsfehler im zweiten Fall größer, als im ersten Fall.

Die Numerische Umsetzung dieser Idee wird Coarse-to-Fine Pyramide genannt und im Kapitel 3.4 genauer erläutert.

3.3 Lösungsverfahren

3.3.1 Lösungsverfahren mit dem Primal-Dual Algorithmus

Die Nutzung des Primal-Dual Algorithmus erfordert zunächst eine primal duale Umformung des Problems. Dies geschieht mit der Legendre-Fenchel-Transformation, die für die Huber-Norm schon in Kapitel 2.3 berechnet wurde. Damit kann das Problem umgestellt werden zu:

$$\min_v \sup_{|p_1|, |p_2| \leq 1} \left\{ \int_{\Omega} \sum_{d=1}^2 \nabla v_d \cdot p_d - \epsilon \frac{|p_d|^2}{2} + \lambda |\rho(v)| dx \right\} \quad (3.6)$$

Das Problem kann in dieser Form mit dem Primal-Dual Algorithmus gelöst werden. Dazu müssen die Operatoren aufgestellt und eingesetzt werden. Die Notation orientiert sich dabei an der Notation aus Kapitel 2.4, in dem das Verfahren eingeführt wur-

de.

$$\begin{aligned} \mathbf{K} : C(\Omega)^2 &\rightarrow C(\Omega)^2 \times C(\Omega)^2, & v &\mapsto \begin{pmatrix} \nabla v_1 \\ \nabla v_2 \end{pmatrix} \\ \mathbf{K}^{\mathbf{T}} : C(\Omega)^2 \times C(\Omega)^2 &\rightarrow C(\Omega)^2, & p &\mapsto \begin{pmatrix} -\operatorname{div}(p_1) \\ -\operatorname{div}(p_2) \end{pmatrix} \\ \mathbf{F}^* : C(\Omega)^2 \times C(\Omega)^2 &\rightarrow \mathbb{R}_{>0}, & p &\mapsto \epsilon \frac{\|p_1\|^2 + \|p_2\|^2}{2} + \chi_{\{|p_1|, |p_2| \leq 1\}}^\infty(p) \\ \mathbf{G} : C(\Omega)^2 &\rightarrow \mathbb{R}_{>0}, & v &\mapsto \lambda |v \cdot \nabla I_R + I_R - I_T| \end{aligned}$$

Damit ergeben sich folgende Resolventenoperatoren für \mathbf{F}^* und \mathbf{G} :

$$\begin{aligned} (\mathbf{1} + \sigma \partial \mathbf{F}^*)^{-1} : C(\Omega)^2 \times C(\Omega)^2 &\rightarrow C(\Omega)^2 \times C(\Omega)^2, & p &\mapsto \prod_{|p_1|, |p_2| \leq 1} \left(\frac{p}{1 + \sigma \epsilon} \right) \\ (\mathbf{1} + \tau \partial \mathbf{G})^{-1} : C(\Omega)^2 &\rightarrow C(\Omega)^2, & v &\mapsto v + \begin{cases} \lambda \tau \nabla I_R & \text{falls } \rho(v) < -\lambda \tau |\nabla I_R|^2 \\ -\lambda \tau \nabla I_R & \text{falls } \rho(v) > \lambda \tau |\nabla I_R|^2 \\ -\frac{\rho(v) \nabla I_R}{|\nabla I_R|^2} & \text{sonst} \end{cases} \end{aligned}$$

Für das Minimierungsproblem, das zum Operator von \mathbf{G} gehört, wird das Lemma 2.2.9 genutzt. Das ist möglich, weil man das diskrete Problem pro Pixel löst. Eingesetzt in das Iterationsverfahren ergeben sich folgende Iterationsschritte:

$$\begin{aligned} p_1^{n+1} &= \frac{p_1^n + \sigma \nabla \bar{v}_1^n}{\max\{1 + \sigma \epsilon, |p_1^n + \sigma \nabla \bar{v}_1^n|\}} \\ p_2^{n+1} &= \frac{p_2^n + \sigma \nabla \bar{v}_2^n}{\max\{1 + \sigma \epsilon, |p_2^n + \sigma \nabla \bar{v}_2^n|\}} \\ v^{n+1} &= (\mathbf{1} + \tau \partial \mathbf{G})^{-1}(w) \\ &= v^n + \tau \begin{pmatrix} \operatorname{div}(p_1^{n+1}) \\ \operatorname{div}(p_2^{n+1}) \end{pmatrix} + \begin{cases} \lambda \tau \nabla I_R & \text{falls } \rho(w) < -\lambda \tau |\nabla I_R|^2 \\ -\lambda \tau \nabla I_R & \text{falls } \rho(w) > \lambda \tau |\nabla I_R|^2 \\ -\frac{\rho(w) \nabla I_R}{|\nabla I_R|^2} & \text{sonst} \end{cases} \\ \bar{v}^{n+1} &= 2v^{n+1} - v^n \end{aligned}$$

Um die Schrittweite passend wählen zu können, muss noch die Operatornorm von \mathbf{K} bestimmt werden. Dazu ist es wichtig, wie der Operator diskretisiert wird. Die Berechnung folgt also in Kapitel 3.5.

3.3.2 Lösungsverfahren mit dem Gradientenabstiegsverfahren

Zur Vereinfachung des Problems werden eine Hilfsvariable w und ein Kopplungsterm eingefügt, wodurch der Datenterm und der Regularisierungsterm im Wechsel approximiert werden können. Die Anwendung des von Chambolle vorgestellten Algorithmus für ROF-Probleme von 2004 (vgl. [Cha04]) erfordert eine primal-duale Umformung des Problems. Mit der LF-Transformation des Huber-Betrags ergibt sich:

$$\min_{v,w} \sup_{|p_1|,|p_2|\leq 1} \left\{ \int_{\Omega} \sum_{d=1}^2 \left[\nabla v_d \cdot p_d - \epsilon \frac{|p_d|^2}{2} + \frac{1}{2\theta} (v_d - w_d)^2 \right] + \lambda |\rho(w)| dx \right\}$$

Danach wird das Problem für die numerische Approximation in zwei Teile aufgeteilt. Für den ersten Teil wird mit fixiertem w folgendes Problem gelöst:

$$\min_v \sup_{|p_1|,|p_2|\leq 1} \left\{ \int_{\Omega} \sum_{d=1}^2 \left[\nabla v_d \cdot p_d - \epsilon \frac{|p_d|^2}{2} + \frac{1}{2\theta} (v_d - w_d)^2 \right] dx \right\}$$

An dieser Stelle wird die Herleitung eines Iterationsverfahrens leider etwas ungenau. Ich habe mich an das Vorgehen von Werlberger et al. gehalten (vgl. [WTP⁺09]), die in ihrer Herleitung auf die Arbeit von Chambolle verweisen (vgl. [Cha04]). Dieser löst das Problem für Funktionale mit L^1 -Regularisierung und benötigt, dass die Norm positiv homogen ist. Dies ist für die Huber-Norm nicht gegeben. Bei meinen Versuchen das Verfahren, sowie einige leichte Abwandlungen, mit dem präsentierten Vorgehen zu beweisen, bin ich leider jedes Mal am Beweis der Monotonie gescheitert, die im Paper zum Erfolg führt. Da ich während des Verfahrens aber für jeden Iterationsschritt die Energie berechnen kann, kann ich empirisch feststellen, dass bei meinen Testbeispielen das Verfahren gegen einen Sattelpunkt konvergiert. Deshalb habe ich es für den späteren Vergleich aufgenommen. Auffällig ist, dass neuere Verfahren wie der Primal-Dual Algorithmus sehr ähnliche Iterationsschritte machen. Da der Beweis nicht funktioniert hat, gebe ich die Iterationsschritte für das Teilproblem ohne weitere Herleitung an:

$$\begin{aligned} v_d^{n+1} &= w_d^n + \theta \operatorname{div}(p_d^n) \\ p_d^{n+1} &= \frac{p_d^n + \tau(\nabla v_d^{n+1} - \epsilon p_d^n)}{\max\left\{1, |p_d^n + \tau(\nabla v_d^{n+1} - \epsilon p_d^n)|\right\}} \end{aligned}$$

Im zweiten Teil löst man mit fixiertem v das Problem:

$$\min_w \int_{\Omega} \frac{|v - w|^2}{2\theta} + \lambda |\rho(w)| dx$$

Das Problem entspricht genau dem Problem der Bestimmung des Resolventenoperators von G beim Primal-Dual Algorithmus. Es kann also mit Lemma 2.2.9 direkt der Iterationsschritt angegeben werden:

$$w^{n+1} = v^{n+1} + \begin{cases} \lambda\theta\nabla I_R & \text{falls } \rho(v^{n+1}) < -\lambda\theta|\nabla I_R|^2 \\ -\lambda\theta\nabla I_R & \text{falls } \rho(v^{n+1}) > \lambda\theta|\nabla I_R|^2 \\ -\frac{\rho(v^{n+1})\nabla I_R}{|\nabla I_R|^2} & \text{sonst} \end{cases}$$

3.4 Diskretisierung

Die Bilder $I \in C(\Omega)$ werden mit einem kartesischen Gitter diskretisiert:

$$\Omega = \{(r, c) | 1 \leq r \leq M, 1 \leq c \leq N \text{ mit } i, j, M, N \in \mathbb{N}\}$$

Für gewöhnlich wird Ω mit einer Gitterweite diskretisiert, sodass das diskrete $\Omega \subset [0, 1]^2$ ist. Dann wird für die Skalierungsinvarianz der Bilder auch das dazugehörige Skalarprodukt mit Hilfe der Gitterweite angepasst. Beim Optical Flow Problem wirkt sich der Skalierungsfaktor nur auf die genaue Wahl der Parameter aus. Alle weiteren Veränderungen sind nur in der Darstellung vorhanden. Um die Darstellung deshalb so einfach wie möglich zu halten, wurde auf diese Verbindung zum Kontinuierlichen verzichtet und das Standardskalarprodukt vom isomorphen $\mathbb{R}^{M \times N}$ gewählt.

Bei den Gradienten werden die Bildgradienten und die Gradienten vom Fluss unterschieden. Beim Fluss wird mit Vorwärtsdifferenzen und Neumann-0-Randwerten diskretisiert:

$$\begin{aligned} \partial_x u(r, c) &= \begin{cases} u(r, c+1) - u(r, c) & \text{falls } 1 \leq c \leq N-1 \\ 0 & \text{falls } c = N \end{cases} \\ \partial_y u(r, c) &= \begin{cases} u(r+1, c) - u(r, c) & \text{falls } 1 \leq r \leq M-1 \\ 0 & \text{falls } r = M \end{cases} \\ \nabla u(r, c) &= \begin{pmatrix} \partial_x u(r, c) \\ \partial_y u(r, c) \end{pmatrix} \end{aligned}$$

Damit auch im diskreten Fall $\langle \nabla u, v \rangle = \langle u, -\operatorname{div}(v) \rangle$ gilt, muss der passende Divergenzoperator gewählt werden (für $v : \Omega \rightarrow \mathbb{R}^2$):

$$\operatorname{div}(v)(r, c) = \begin{cases} v_1(r, c) - v_1(r, c-1) & 1 < c < N \\ v_1(r, c) & c = 1 \\ -v_1(r, c-1) & c = N \end{cases} \\ + \begin{cases} v_2(r, c) - v_2(r-1, c) & 1 < r < M \\ v_2(r, c) & r = 1 \\ -v_2(r-1, c) & r = M \end{cases}$$

Da der Gradient eine lineare Abbildung ist, kann auch eine Matrix aufgestellt werden, die durch Multiplikation die Abbildung darstellt. In diesem Fall muss das Bild als Vektor dargestellt werden. Die Matrix für den negativen Divergenzoperator erhält man dann einfach durch Transponieren. Die Bildgradienten werden mit zentralen Differenzen approximiert, da sich bei kleinen empirischen Versuchen bessere Ergebnisse gezeigt haben:

$$\partial_x^{ZD} I(r, c) = \begin{cases} \frac{I(r, c+1) - I(r, c-1)}{2} & \text{falls } 1 < c < N \\ 0 & \text{falls } c = 1 \text{ oder } c = N \end{cases} \\ \partial_y^{ZD} I(r, c) = \begin{cases} \frac{I(r+1, c) - I(r-1, c)}{2} & \text{falls } 1 < r < M \\ 0 & \text{falls } r = 1 \text{ oder } r = M \end{cases} \\ \nabla I(r, c) = \begin{pmatrix} \partial_x^{ZD} I(r, c) \\ \partial_y^{ZD} I(r, c) \end{pmatrix}$$

Der adjungierte Operator des Bildgradienten wird im weiteren Verlauf nicht benötigt. Deshalb wird er an dieser Stelle, um Verwirrungen zu vermeiden, nicht eingeführt. Allerdings wird der Bildgradient auch nicht extra anders dargestellt. Es geht dann aus dem Argument des Gradienten hervor, welche Approximation verwendet wird.

Die bereits angekündigte Coarse-to-Fine Pyramide wurde schon in Kapitel 3.2 motiviert. Nun gibt es verschiedene Möglichkeiten der Umsetzung. Alle Umsetzung haben gemein,

dass folgendes Optical Flow Problem wiederholt gelöst wird:

$$\min_v \left\{ \int_{\Omega} \sum_{d=1}^2 |\nabla v_d| + \lambda |\rho(v)| dx \right\}$$

mit $\rho(v) = v \cdot \nabla I_R(x + w_k(x)) + I_R(x + w_k(x)) - I_T(x)$

$$\Rightarrow w_{k+1}(x) = w_k(x) + v$$

Im Wesentlichen werden zwei Ideen unterschiedlich umgesetzt. Die erste Idee ist, die Bilder schrittweise zu verkleinern. Dadurch wird, im Vergleich zu einer wiederholten Optimierung ohne Verkleinerung, die Laufzeit für das Verfahren insgesamt verbessert und große und kleine Bewegungen durch die schrittweise Verfeinerung trotzdem zuverlässig erkannt. Begonnen wird dabei mit dem kleinsten Bild, um vor allem die groben Bewegungen zu erkennen. Geendet wird mit dem Optical Flow Problem für die Bilder in Originalgröße. Dadurch werden auch die feineren Strukturen erkannt. Die zweite Idee versucht, das Bildmaterial durch verschiedene Operationen (z.B. Glättung) zu optimieren, um dadurch robuster auf Störungen zu reagieren. Die genaue Umsetzung variiert dabei sehr stark. In der Arbeit von Werlberger et al. (vgl. [WTP⁺09]) wird zum Beispiel ein ROF-TV denoising Algorithmus genutzt. Ich verwende einen Gaußfilter mit anpassbarer Standardabweichungen. Der Pseudocode für das vorgehen lautet wie folgt:

Code 3.1: Pseudocode Coarse-to-Fine Pyramide

```

1 create(Groessenpyramide);           %von klein nach gross
2 create(Glaettungswertepyramide);
3 referenceImage = originalReferenzBild;
4 targetImage = originalZielBild;
5 for i=1:pyramidlevels
6     downscale(referenceImage , Groessenpyramide[i]);
7     blur(referenceImage , Glaettungspyramide[i]);
8     downscale(targetImage , Groessenpyramide[i]);
9     blur(targetImage , Glaettungspyramide[i]);
10    downscale(flow , Groessenpyramide[i]);
11    flow = flow + solveOFP(referenceImage , targetImage);
12    upscale(flow , OriginalBildGroesse);
13    referenceImage = warp(originalReferenzBild , flow);
14    targetImage = originalZielBild;
15 end

```

Für die genauen Implementierungsdetails sind hier noch einige Hinweise:

1. Die Größenpyramide kann auch mehrere gleiche Werte nacheinander enthalten (z.B. $[3 \times 3, 3 \times 3, 3 \times 3, 5 \times 5, 5 \times 5, 5 \times 5, \dots, 1000 \times 1000]$).
2. Die Skalierungen können auf unterschiedliche Weise umgesetzt werden (z.B. bilinear, bikubisch, ...). In der Coarse-to-Fine Pyramide wird in meiner Umsetzung bikubisch skaliert.
3. Die Anzahl der pyramidlevels wird für gewöhnlich, anhand eines Skalierungsfaktors für die Verkleinerung und einer Minimalgröße, automatisch bestimmt.

3.5 Implementierung

Die Implementierung erfolgte mit Matlab. Sowohl bei der Umsetzung des Primal-Dual Algorithmus, als auch beim Gradientenverfahren wurden eine Coarse-to-Fine Pyramide verwendet. Die Pyramide bekommt als Parameter einen Skalierungsfaktor (*pyramidfactor*) und berechnet damit, wie oft das Problem gelöst werden muss. Die Anzahl an Wiederholungen (*pyramidlevels*) wird mit folgender Formel aus der Höhe (*height*) und Breite (*width*) des Bildes berechnet:

$$pyramidlevels = \text{ceil} \left(\frac{\log(16) - \log(\min(width, height))}{\log(pyramidfactor)} \right)$$

Das entspricht der Anzahl an Wiederholungen, mit der die kleinste Seitenlänge kleiner gleich 16 ist. Auf jedem Level, mit Ausnahme des feinsten, werden das Referenz- und das Zielbild mit einem Gaußfilter (s. Kapitel 4.2.1) der Standardabweichung $s = 2$ geglättet. Die feinste Stufe wird nicht geglättet, damit keine Details verloren gehen und das Ergebnis subpixelgenau ist.

Der Primal-Dual Algorithmus wird auf jeder Stufe mit den Parametern $\lambda = 15$, $\epsilon = 0.1$, $\sigma = \frac{1}{\sqrt{8}}$ und $\tau = \frac{1}{\sqrt{9}}$ gelöst. λ und ϵ wurden dabei in einem kleinen Parametertest bestimmt und im Vergleich zur theoretischen Überlegung als plausibel akzeptiert. Die Anzahl der Iterationen für den Primal-Dual Algorithmus wurde auf 250 festgesetzt und durch eine Betrachtung der Energiekurve während des Iterationsverfahrens validiert. Eine Reduktion aus Performance-Gründen könnte mit geringen Auswirkungen auf das Ergebnis vorgenommen werden. τ und σ wurden so gewählt, dass $\tau\sigma\|\mathbf{K}\|^2 < 1$ erfüllt ist. Man kann durch eine kleine Rechnung zeigen, dass für den Gradient mit Vorwärtsdifferenzen

$\|\mathbf{K}\| \leq \sqrt{8}$ gilt (vgl. [Cha04]):

$$\begin{aligned}
 \|\mathbf{K}(p)\|^2 &= \|\nabla(p)\|^2 = \sum_{ij} |\partial_x(p)_{ij}|^2 + |\partial_y(p)_{ij}|^2 \\
 &= \sum_{i=1}^M \sum_{j=1}^{N-1} (|p_{ij+1} - p_{ij}|^2) + \sum_{i=1}^{M-1} \sum_{j=1}^N (|p_{i+1j} - p_{ij}|^2) \\
 &\leq \sum_{i=1}^M \sum_{j=1}^N (2|p_{ij}|^2 + 2|p_{ij}|^2) - 2 \sum_{i=1}^M \sum_{j=1}^{N-1} (p_{ij+1}p_{ij}) - 2 \sum_{i=1}^{M-1} \sum_{j=1}^N (p_{i+1j}p_{ij}) \\
 &\stackrel{Young}{\leq} 4 \sum_{i=1}^M \sum_{j=1}^N (|p_{ij}|^2) + 2 \sum_{i=1}^M \sum_{j=1}^N (|p_{ij}|^2 + |p_{ij}|^2) \\
 &\leq 8\|p\|^2 \\
 \Rightarrow \quad \|\mathbf{K}\| &\leq \sqrt{8}
 \end{aligned}$$

3.6 Evaluation

3.6.1 Fehlermaße

Um gute Verfahren zu entwickeln, und den Vergleich untereinander zu ermöglichen, ist es sehr wichtig, das Ergebnis objektiv bewerten zu können. Baker et al. haben dazu eine Datenbank für Bilderpaare und Groundtruth-Flüsse entwickelt und in ihrem Paper Fehlermaße für den Performance Vergleich angegeben (vgl. [BSL⁺11], vision.middlebury.edu/flow/).

Für die Fehlermaße können generell zwei mögliche Ansätze unterscheiden werden. Der erste Ansatz vergleicht den berechneten Fluss v mit dem Groundtruth-Fluss v_{GT} . Der einfachste Weg dafür ist der Absolute Endpoint Error. Dieser misst vektorweise die Distanz der beiden Flüsse:

$$AEE = \frac{1}{|\Omega|} \sum_{x \in \Omega} |v(x) - v_{GT}(x)|$$

Eine zweite Umsetzung des ersten Ansatzes berechnet den Angular Error. Der Angular Error ist vor allem sehr populär, weil er in der ausführlichen Arbeit von Barron et al. ([BFB94]) und daraufhin in einigen späteren Arbeiten genutzt wurde. Dabei werden die beiden Verschiebungsvektorfelder v und v_{GT} in den \mathbb{R}^3 abgebildet und dort der Mittelwert

der Winkel zwischen ihren Einträgen gebildet:

$$\hat{v} = \frac{(v_1, v_2, 1)}{\sqrt{|v|^2 + 1}}, \quad \hat{v}_{GT} = \frac{(v_{GT,1}, v_{GT,2}, 1)}{\sqrt{|v_{GT}|^2 + 1}}$$

$$AE = \frac{1}{|\Omega|} \sum_{x \in \Omega} \arccos(\hat{v}(x) \cdot \hat{v}_{GT}(x))$$

Der Nachteil dieser Vorgehensweise ist, dass der Fehler bei kleinen Flüssen stärker gewichtet wird als bei großen, da sich der Winkel bei gleichem absoluten Fehler stärker ändert. Aus diesem Grund hat Baker zusätzlich den Absolut Endpoint Error vorgeschlagen.

Der zweite Ansatz geht davon aus, dass der Fluss ein Verschiebungsvektorfeld ist, welches das Referenzbild auf das Zielbild verschiebt, also dass $I_R(x + v(x)) = I_T(x)$ gilt. Der einfachste Fehler, der aus dieser Annahme abgeleitet werden kann, ist die root-mean-squared-Distanz vom verschobenen Referenzbild und vom Zielbild (der Mittelwert wird nicht über den Gesamtfehler gebildet, da so der Wertebereich des Fehlers für Bilder auf $[0, 1]$ festgelegt ist). Der Fehler wird als Interpolation Error bezeichnet:

$$IE = \sqrt{\frac{1}{|\Omega|} \sum_{x \in \Omega} |I_R(x + v(x)) - I_T(x)|^2}$$

Der diskrete Gradient ist relativ anfällig für Rauschen, ein Effekt der besonders stark an Kanten auftritt. Deshalb gibt es noch eine Erweiterung des Interpolation Errors der den Fehler mit der Länge der Gradienten des Zielbildes gewichtet. Dieser Fehler wird normalized Interpolation Error genannt und wie folgt definiert:

$$NIE = \sqrt{\frac{1}{|\Omega|} \sum_{x \in \Omega} \frac{|I_R(x + v(x)) - I_T(x)|^2}{|\nabla I_T(x)|^2 + 1}}$$

Da die Fehler auch pixelweise berechnet werden können, kann man auch weiterreichende statistische Untersuchungen machen. Ich beschränke mich an dieser Stelle auf die Mittelwerte der Fehler und verweise deshalb für weitere statistische Analysen auf [BSL⁺11] Kapitel 4.2. Statistics. Die Bezeichnung der Fehlermaße wird in Anlehnung an die Dissertation von H. Dirks vorgenommen (vgl. [Dir15]).

Ergänzend zu dieser objektiven Strategie die Flüsse zu bewerten, gibt es noch einen subjektiven Ansatz. Dazu wird der Fluss in ein farbcodiertes Bild umgerechnet. Die Farbe entspricht dabei der Richtung (vgl. Abbildung 3.2) und die Helligkeit der Länge des

Flussvektors. Standardmäßig wird die Helligkeit mit dem maximalen Fluss normalisiert. Bei Vergleichen von Flüssen wurde deshalb darauf geachtet, die Skalierung einander anzupassen.

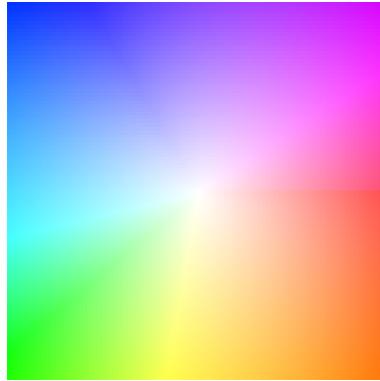


Abbildung 3.2: Farbskala der Flussbilder. Darstellung eines nach außen gerichteten Vektorfeldes.

3.6.2 Mögliche Herausforderungen

Um feststellen zu können, welcher Algorithmus im Umgang mit bestimmten Herausforderungen am besten geeignet ist, muss zuerst einmal festgestellt werden, welche Herausforderungen bestehen. Solange keine zusätzlichen Hürden die Problemlösung erschweren, sollten die drei Standardoperationen Translationen, Rotationen und Skalierungen generell kein Problem für die Algorithmen darstellen. Bei Bilderserien bewegt sich meistens jedoch nicht das Bild als ganzes, sondern einige Gegenstände, wie z.B. Autos oder Tiere, weisen eine andere Bewegung auf als der restliche Teil des Bildes. Dadurch entstehen dann unter anderem Überdeckungen und Unstetigkeiten im Fluss, die deutlich größere Herausforderungen darstellen. Nachfolgend werde ich deshalb eine kleine Liste mit Herausforderungen aufführen, die auf Probleme aufmerksam machen und diese knapp erläutern soll (orientiert hat sich die Liste an den Problemen die in [BSL⁺11] aufgeführt werden).

1. Das aperture problem (Blendenproblem). Für eine Erklärung siehe die Herleitung des Optical Flow Problems in Kapitel 3.2

2. Texturlose Regionen. Geht man davon aus, dass die Verschiebung eines Objekts erkannt werden soll, stellen texturlose Bereiche ein Problem dar, weil dort der Bildgradient konstant 0 ist und somit der Datenterm nicht von dem Verschiebungsvektorfeld v abhängt. Resultat sind beliebig verformte, aber nicht das gewünschte, konstante, Vek-

torfelder innerhalb dieser Bereiche.

3. Unstetigkeiten. Durch die Bewegung aneinandergrenzender Objekte in verschiedene Richtungen oder Verschiebungen im Vorder- und Hintergrund, entstehen Unstetigkeiten im Fluss. Für den Datenterm stellen diese Unstetigkeiten an sich kein Problem dar, da dieser nur lokal, also für jedes Pixel einzeln berechnet wird. Aufgrund des Regularisierungsterms, der versucht den Fluss glatt also stetig zu halten, werden diese unstetigen Lösungen jedoch bestraft.

4. Überdeckungen. Überdeckungen in Bildern bringen zwei Probleme mit sich. Erstens treten bei Überdeckungen auch immer Unstetigkeiten in der Bewegung auf. Und zweitens fehlen in dem Bild, in dem die Überdeckung auftritt, gegenüber dem anderen Bild Informationen, die dann geschätzt werden müssen. Dies führt zum Beispiel zu konstanten Fortsetzungen, also Schlieren.

5. Große Bewegungen. Für eine kurze Erläuterung siehe die Motivation der Coarse-to-Fine Pyramide in Kapitel 3.2.

6. Kleine Objekte Die Herausforderung in kleinen Objekten liegt im Verhältnis zum gesamten Bild begründet. Insbesondere bei Verfahren die ein Funktional minimieren, besteht die Gefahr, dass die Bewegung kleiner Objekte missachtet wird, da der Fehler durch den Regularisierungsterm den Fehler durch den Datenterm ausgleicht.

7. Verformungen (non-rigid motion). Verformungen sind vor allem im Zusammenspiel mit Bewegungen starrer Objekte eine Herausforderung. Sie zeichnen sich dadurch aus, dass innerhalb eines Objektes unterschiedliche Bewegungen stattfinden, während starre Objekte sich als Ganzes bewegen. Die Herausforderung besteht nun darin, richtig zu erkennen ob ein starres Objekt dargestellt wird, dass sich als Ganzes bewegt, oder man innerhalb des Objektes leichte Veränderungen vorfindet. Außerdem können bei Verformungen lokale Charakteristika, die für einige Verfahren als wichtige Anhaltspunkte dienen, verändert werden oder sogar verloren gehen.

8. Gemischte Pixel. Mit gemischten Pixeln (engl. mixed pixels) sind Pixel gemeint, die Informationen mehrerer eigentlich unterschiedlicher Bereiche enthalten. Das kann zum Beispiel ein Pixel an einer Kante sein, oder ein Pixel in einem Gebiet mit starker Struktur, wie z.B. ein Nadelbaum. Das Problem liegt darin, das Pixel korrekt zuzuordnen und bei zwei möglicherweise unterschiedlichen Bewegungen eine gute Lösung zu finden. Wann eine Lösung in diesem Fall gut ist, ist dabei natürlich ein Problem für sich, da ein Pixel nur eine Bewegung vollziehen kann.

Im Zusammenhang mit dem Superresolution Problem sind diese Pixel von besonderem Interesse, da eine Verfeinerung des Bildes gerade dort gewünscht ist.

9. Belichtungsänderungen. Das Optical Flow Problem schätzt nicht die tatsächli-

che Bewegung der dargestellten Objekte, sondern die optischen Veränderungen im Bild. Mit Änderungen in der Belichtung gehen Änderungen in der Objektstruktur, Helligkeitsunterschiede und veränderte Schattierungen einher. Diese optischen Veränderungen erschweren die Berechnung des korrekten Flusses erheblich, da sie vom Algorithmus kompensiert werden müssen.

10. Nicht-Lambertsche Reflexionen. Bei idealen Lambertschen Oberflächen ändert sich die Strahlungsstärke auftreffender Strahlen nach dem Abstrahlwinkel. Für das menschliche Auge oder eine Kamera erscheint die Oberfläche aus jedem Blickwinkel gleich hell. Diese Regel trifft normalerweise zwar nicht einhundertprozentig zu, aber es gibt einige Materialien (wie z.B. Papier oder Holz), die es fast perfekt erfüllen, sowie andere Materialien (wie z.B. Glas, Lackierte Autos), deren Helligkeit stark vom Blickwinkel abhängen. Für das Optical Flow Problem erschweren dabei zusätzliche Reflexionsspitzen und abstrakte Spiegelungen die Lösungsfindung.

11. Bewegungsunschärfe. Durch schnelle Bewegungen entstehen sowohl für die Kamera als auch für das menschliche Auge Unschärfen. Diese können sich bei Videos mit der Zeit leicht verändern und sind deshalb ein Problem für den Optical Flow. Außerdem entstehen bei diesem Problem meist auch Überdeckungen und teilweise große Bewegungen, die zusammen dann eine besondere Hürde darstellen.

12. Kamera Rauschen. Die Sensoren in einer Kamera sind ständigen Ungenauigkeiten ausgesetzt. Diese werden Rauschen genannt und können in unterschiedlichen Formen auftreten (z.B. gaußsches Rauschen, Salt-and-Pepper Noise). Als Bildfehler behindern sie die Lösungsfindung durch Fehlinformationen teilweise stark und müssen behoben werden.

3.6.3 Experimentelle Erweiterungen

Während der Recherche- und Implementierungsphase habe ich einige Erweiterungsmöglichkeiten kennengelernt und erprobt. Um darunter potentielle Verbesserungen zu erkennen, habe ich mit einer kleinen Auswahl von 7 Bilderpaaren, jeweils den Fluss und die Fehlermaße berechnet. Im Folgenden werden die Erweiterungen zunächst vorgestellt. Im Anschluss daran werden Vor- und Nachteile diskutiert und festgehalten, welche Erweiterungen einen Mehrwert haben.

Für den Vergleich wurden die Bilder Dimetrodon, Hydrangea, Rubberwhale und Urban3 aus der Middlebury Database genutzt (vision.middlebury.edu/flow/). Zusätzlich wurden die Bilder cameramotion, fish und hand aus der Datenbank von Liu et al. verwendet, da sie unter realeren Bedingungen aufgenommen wurden (vgl. [LFAW08],

Database: <http://people.csail.mit.edu/celiu/motionAnnotation/>). In der dazugehörigen Arbeit haben Liu et al. außerdem ein überwachtes Verfahren umgesetzt, welches einen Groundtruth-Fluss liefert. Ich möchte an dieser Stelle darauf hinweisen, dass dieser Groundtruth-Fluss möglicherweise nicht ganz optimal ist. Da er aber generell sehr präzise ist, wird er als Vergleich genutzt.

Die erste Erweiterung des gewöhnlichen Primal-Dual Algorithmus ist die Umsetzung einer adaptiven Schrittweite (vgl. Satz 2.4.4 und Satz 2.4.7). Die hier verwendete Umsetzung macht sich die Problem-Struktur zu nutze, und passt iterativ die Schrittweiten an. Zudem gibt es auch die Möglichkeit, die Adaption an die Residuen der primalen und dualen Variable anzupassen. Für mehr Details dazu verweise ich auf Goldstein et al. (vgl. [GLY⁺13]).

Für den Vergleich wurde in beiden Fällen die Coarse-to-Fine Pyramide mit Größenskalierung verwendet. Der Reduzierungsfaktor wurde auf 0.8 und die Standardabweichung für den gaußschen Glättungsoperator auf $s = 2$ gesetzt. Für die Verfahren wurden die Parameter $\lambda = 15$, $\epsilon = 0.01$ und 250 Iterationsschritte verwendet. Die Schrittweiten wurden so gewählt, dass die jeweiligen Voraussetzungen $\sigma\tau\|\mathbf{K}\|^2 < 1$ bzw. $\sigma\tau\|\mathbf{K}\|^2 = 1$ gelten. Für die Schrittweiten ohne Adaption bedeutet das $\sigma = \frac{1}{\sqrt{8}}$, $\tau = \frac{1}{\sqrt{9}}$. Die Anfangswerte der Schrittweiten mit Adaption wurden auf $\sigma_0 = \tau_0 = \frac{1}{\sqrt{8}}$ festgelegt (auch ein Test mit dem Wert $\tau_0 = \frac{1}{\sqrt{9}}$ ergab fast identische Ergebnisse und Fehlerwerte!).

Vergleicht man die Fehlerwerte beider Verfahren fällt auf, dass trotz der besseren Konvergenzordnung des adaptiven Verfahrens in der primalen Variable, alle Fehlerwerte, außer die Groundtruth-Vergleichsfehler vom Bild Dimetrodon, minimal schlechter sind (s. Tabelle 3.1 und 3.2).

	AE	AEE	IE	NIE	time	Energie
cameramotion	0.1410	0.5615	0.0302	0.0298	125.7434	312747.73
Dimetrodon	0.1289	0.3340	0.0093	0.0093	89.3778	71871.58
Rubberwhale	0.2108	0.3722	0.0106	0.0106	90.1834	71814.83

Tabelle 3.1: Fehlermaße des Algorithmus ohne adaptive Schrittweite

	AE	AEE	IE	NIE	time	Energie
cameramotion	0.1519	0.6536	0.0343	0.0338	126.6512	301840.14
Dimetrodon	0.1275	0.3294	0.0105	0.0105	89.3754	70557.22
Rubberwhale	0.2463	0.4293	0.0126	0.0125	89.5841	70952.37

Tabelle 3.2: Fehlermaße des Algorithmus mit adaptive Schrittweite

Auf den ersten Blick scheint also der normale Primal-Dual Algorithmus besser zu sein. Vergleicht man jedoch die Energien der beiden Verfahren, so stellt man fest, dass (mit Ausnahme des Bildes hand, bei dem das Ergebnis insgesamt nicht besonders gut ausfällt) die Energie des adaptiven Verfahrens immer geringer ist. Der Fehler scheint also nicht durch das Verfahren, sondern durch das Modell zu entstehen. Nimmt man zusätzlich noch die Bilder der farbcodierten Flüsse als Bewertungsmaßstab hinzu, so fallen beim normalen Algorithmus deutlich mehr mögliche Störungen auf (s. Abbildung 3.3).



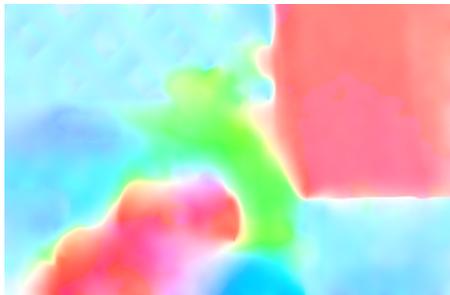
(a) adaptive Schrittweite



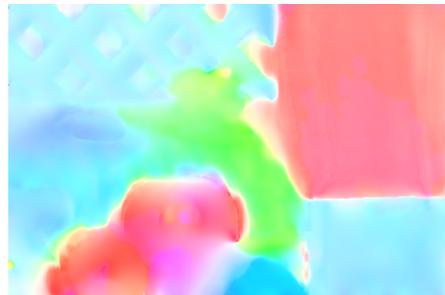
(b) konstante Schrittweite

Abbildung 3.3: Vergleich der Flussbilder von cameramotion

Andererseits könnten diese möglichen Störungen auch Details sein (s. Abbildung 3.4).



(a) adaptive Schrittweite



(b) konstante Schrittweite

Abbildung 3.4: Vergleich der Flussbilder von Rubberwhale

Es lässt sich also vorerst nicht eindeutig klären, ob der Algorithmus mit der adaptiven Schrittweite besser oder schlechter ist, als der Algorithmus ohne adaptive Schrittweite. Deshalb sehe ich ihn erstmal als Alternative an, und verschiebe die endgültige Bewertung auf den Zeitpunkt, an dem die Auswirkungen auf das Superresolution Problem bekannt sind. Davon unabhängig deuten die schlechteren Fehlerwerte des Verfahrens mit der geringeren Energie jedoch auf mögliche Defizite des Modells hin. Die dritte, vierte und

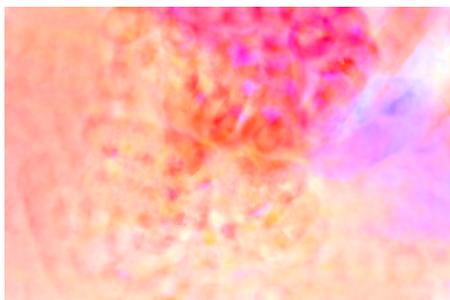
fünfte Erweiterung werden deshalb Veränderungen am Modell sein.

Die zweite Erweiterung ist die alternative Auslegung der Coarse-to-Fine Pyramide. Untersucht wird die Umsetzung mit einer reinen Glättungspyramide, die mit einer großen Standardabweichung ($s = 8$) für den gaußschen Glättungsoperator beginnt und auf dem feinsten Level ohne Glättung endet. Die Standardabweichung wird dabei schrittweise um den Faktor 0.6 reduziert. Die Größe bleibt für jede Ebene der Pyramide gleich. Diese Auslegung betrachte ich dabei als alternative Möglichkeit für die Umsetzung der Pyramide.

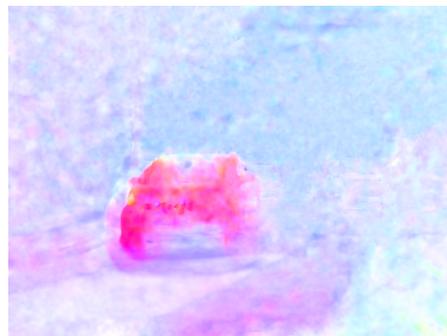
Als Ergebnis der Untersuchung lassen sich zwei wesentliche Punkte feststellen. Zum einen ist der Algorithmus trotz der deutlich geringeren Anzahl der Wiederholungen langsamer. Im Durchschnitt betrug die Berechnungszeit auf den sieben Testbildern fast das Dreifache desselben Algorithmus mit der im Kapitel 3.5 beschriebenen Umsetzung mit zusätzlicher Skalierung (113s - 334s). Die zweite und deutlich wichtigere Feststellung ist allerdings die wesentlich schlechtere Qualität der Ergebnisse. Sowohl die Fehlermaße (s. Tabelle 3.3 und 3.1) als auch die subjektive Bewertung durch die Betrachtung der Flussbilder (s. Abbildung 3.5), lassen nur den Schluss zu, dass diese Umsetzung der anderen unterlegen ist.

	AE	AEE	IE	NIE	time
cameramotion	0.2163	0.7715	0.0328	0.0325	343.4046
Dimetrodon	0.2026	0.5520	0.0123	0.0122	253.3528
Rubberwhale	0.3138	0.5736	0.0129	0.0128	253.6317

Tabelle 3.3: Fehlermaße des Algorithmus mit Glättungspyramide



(a) Dimetrodon



(b) cameramotion

Abbildung 3.5: Beispielhafte Flussbilder mit Glättungspyramide

In der Arbeit von Werlberger et al. (vgl. [WTP⁺09]) wird eine Erweiterung des Modells

vorgeschlagen, die die lokalen Bildgradienten nutzt, um damit die Norm der Regularisierung anzupassen. Im Speziellen wird eine positiv definite symmetrische Matrix D aufgestellt, um das Standardskalarprodukt zur Berechnung der Länge der Gradientenvektoren, welches auch bei der Huber-Norm verwendet wird, abzuändern. Der entscheidende Punkt dabei ist die Anisotropie des Tensors. Als Formel ausgedrückt bedeutet das:

$$\|\nabla u\|_\epsilon \rightarrow \|D\nabla u\|_\epsilon \quad \Leftrightarrow \quad |\nabla u|^2 = (\nabla u)^T(\nabla u) \rightarrow |D\nabla u|^2 = (\nabla u)^T D^2(\nabla u)$$

In meiner Umsetzung nutze ich nun die von Werlberger vorgeschlagene Definition von D :

$$D := \exp(-\alpha|\nabla I_R|^\beta)\vec{n}\vec{n}^T + \vec{n}^\perp\vec{n}^{\perp T} \quad \text{mit } \vec{n} = \frac{\nabla I}{|\nabla I|}, \quad \vec{n} \perp \vec{n}^\perp$$

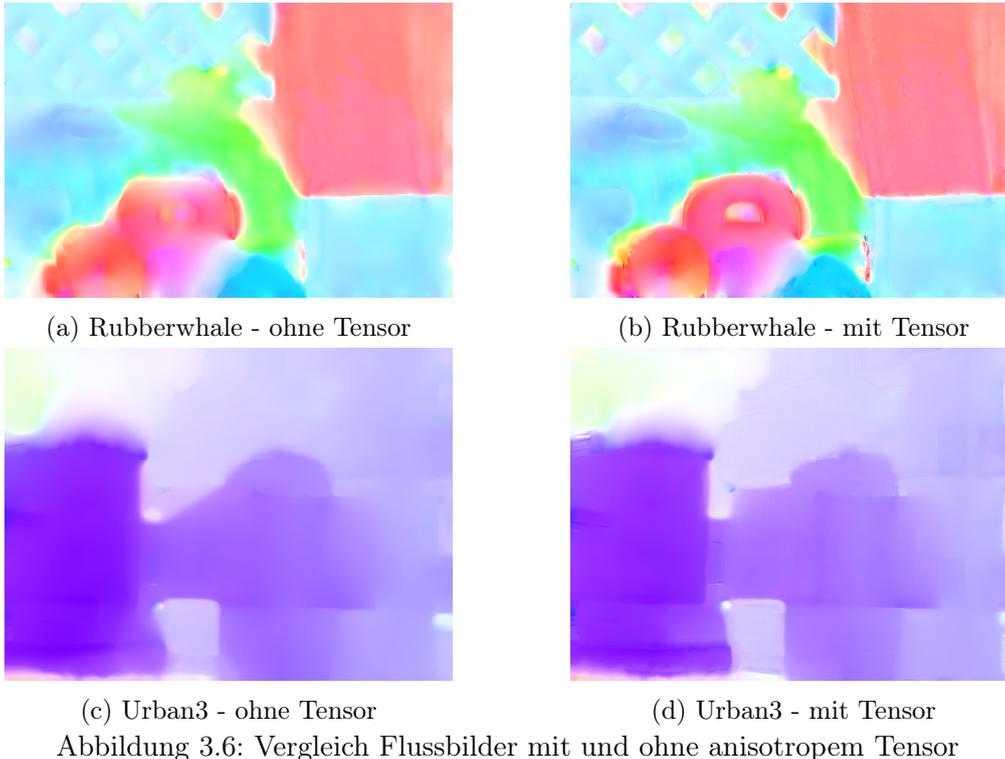
Ich wähle analog zu Werlberger $\alpha = 5$ und $\beta = 0.5$. Dadurch wird die Länge der Vektoren in Richtung des Bildgradienten reduziert, also durch den Regularisierungsterm weniger stark bestraft. Oder anders ausgedrückt, die zum Bildgradienten tangentialen Vektoren werden gegenüber den dazu orthogonalen Vektoren bevorzugt:

$$\begin{aligned} \vec{n}^T D^2 \vec{n} &= \exp(-2\alpha|\nabla I_R|^\beta) \leq 1 \\ \vec{n}^{\perp T} D^2 \vec{n}^\perp &= 1 \end{aligned}$$

Außerdem wird eine leichte Skalierung vorgenommen, da dort, wo der Bildgradient groß ist, $\exp(-2\alpha|\nabla I_R|^\beta)$ klein wird und umgekehrt. Anisotrop heißt hier, dass nicht alle Richtungen gleich behandelt werden, sondern manche Richtungen anderen gegenüber bevorzugt werden. Zu der globalen Optimierung der Energie kommt also eine lokaler Einfluss hinzu.

Im Vergleich zur Umsetzung ohne anisotropem Tensor, deuten die Interpolationsfehler auf eine leichte Überlegenheit des Verfahrens mit Tensor hin. Bei den Groundtruth-Fehlern der Bilder Dimetrodon und Hydrangea ist das Verfahren ohne Tensor besser, bei den Bildern Rubberwhale und Urban3 das Verfahren mit Tensor. Bei den Verfahren fish und cameramotion ist der AEE beim Verfahren ohne Tensor geringer und der AE beim Verfahren mit. (Das Bild hand bleibt außen vor, da dort alle getesteten Verfahren sehr schlecht abschneiden.) Um Stärken und Schwächen zu identifizieren, müssen also die Gemeinsamkeiten der Bilder untersucht werden. Der wesentliche Unterschied zwischen den Paaren Rubberwhale, Urban3 und Dimetrodon, Hydrangea scheint die deutlich höhere Anzahl an sich unterschiedlich bewegenden Objekten zu sein. Dadurch treten vor allem die beiden Probleme Unstetigkeiten im Fluss und gemischte Pixel an den Objektkanten

in den Bildern Rubberwhale und Urban 3 sehr viel häufiger auf. Die Vermutung, dass das Verfahren mit anisotropem Tensor an den Kanten, also den Stellen an denen der Tensor besonders stark wirkt, besser abschneidet, liegt somit nahe. Im Bezug auf das Superresolution Problem erhöht dies die Hoffnung auf eine Verbesserung des Ergebnisses. Unterstützt wird diese Vermutung durch die subjektive Bewertung der Flussbilder (s. Abbildung 3.6).



Man erkennt in beiden Bildern mit Tensor eine deutlich bessere Erfassung der Objekte und damit des tatsächlichen Flusses. Im Bild Rubberwhale wird dies vor allem im Zaun und dem liegenden D erkennbar, während im Bild Urban3 besonders die besseren Konturen des Objekts links und die nicht abgeschnittene Ecke in der Mitte diesen Eindruck vermitteln. Damit erhärtet sich der Verdacht, dass diese Erweiterung für das Superresolution Problem einen Mehrwert mit sich bringt und getestet werden sollte.

Die letzten beiden Erweiterungen die ich vorstellen möchte, sind Modellerweiterung um jeweils einen Term, der Störungen wie beispielsweise Rauschen oder Belichtungsunterschiede reduzieren soll. Der Algorithmus soll so insgesamt robuster werden. In der ersten

Umsetzung wird das Modell dazu wie folgt erweitert:

$$\min_{u,v} \left\{ \int_{\Omega} \sum_{d=1}^2 |\nabla v_d|_{\epsilon} + |\nabla u|_{\epsilon} + \lambda |\tilde{\rho}(u,v)| dx \right\}$$

mit $\tilde{\rho}(u,v) = v \cdot \nabla I_R + I_R - I_T + \beta u$

Hinter beiden Möglichkeiten steckt die Idee, kleine Schwankungen und Bildfehler in I_R, I_T durch u auszugleichen und so die Fehler im Fluss zu reduzieren. Diese erste Variante orientiert sich an der Erweiterung von Chambolle und Pock in [CP10] auf S. 37. Mit der Regularisierung des Gradienten sollen Belichtungsunterschiede, wie Helligkeits- und Kontraständerungen, von I_R nach I_T abgefangen werden. Die zweite Umsetzung ist der ersten sehr ähnlich. Der einzige Unterschied besteht in der unterschiedlichen Regularisierung von u :

$$\min_{u,v} \left\{ \int_{\Omega} \sum_{d=1}^2 |\nabla v_d|_{\epsilon} + |u|_{\epsilon} + \lambda |\tilde{\rho}(u,v)| dx \right\}$$

mit $\tilde{\rho}(u,v) = v \cdot \nabla I_R + I_R - I_T + \beta u$

Durch die Bestrafung des Betrags von u wird versucht Störungen durch Rauschen besser abfangen zu können. Da diese Störungen für gewöhnlich sehr vereinzelt auftreten ist es nur sinnvoll nicht den Gradienten von u , sondern die Größe von u zu bestrafen. In beiden Experimenten habe ich analog zur Parameterwahl von Chambolle und Pock für den Faktor den Wert $\beta = 0.01$ gewählt. Dies ist sinnvoll, da bei Belichtungsunterschieden von sehr glatten bzw. konstanten Lösungen, und bei Rauschen von sehr kleinen Werten ausgegangen werden kann.

Bei der Umsetzung habe ich zunächst versucht, die Bewertung des Ansatzes mit Belichtungsterm auf Grundlage der Testbilder durchzuführen. Trotz erhöhtem Rechenaufwand ist das Ergebnis jedoch minimal schlechter, als das Ergebnis ohne Belichtungsterm. In Zahlen ausgedrückt steht ein durchschnittlicher IE von 0,0202, bei einer durchschnittlichen Laufzeit von 122,8267s, einem durchschnittlichen IE von 0,0213, bei einer Laufzeit von 175.0449s, gegenüber. Das entspricht einer verlängerten Laufzeit von 42,5%. Da auch die anderen Fehler nur diesen geringen Unterschied aufwiesen, stellte sich noch die Frage, ob Störungen eventuell trotzdem besser abgefangen werden können. Deshalb habe ich die Helligkeit und den Kontrast der Bilder Hydrangea und Dimetrodon mit verschiedenen bereits implementierten Funktionen von Matlab bearbeitet. Anschließend habe ich das bearbeitete Bild als Referenzbild ausgewählt und den Fluss zum unbearbeiteten Ziel-

bild berechnet. Der Vergleich zum gegebenen Groundtruth-Fluss soll eine Aussage über das Modell mit Belichtungsterm im Vergleich zum Standardmodell ermöglichen. Die verwendeten Matlab Funktionen waren `histeq` (`histeq(refImage, 256)`) und `imadjust`. Außerdem habe ich ein konstantes Vektorfeld ($5 * \text{ones}(\text{size}(\text{refImage}))$) addiert und den roten Farbanteil des Bildes vor der Grauwert-Bild Berechnung durch den Matlab Befehl `min(refImage(:, :, 1) * 1.2, 1)` abgeändert. Das Ergebnis dieser Untersuchung ist eindeutig. Das Verfahren mit Belichtungsterm schneidet auf beiden Bildern deutlich besser ab als das Standardverfahren (s. Tabelle 3.4 und 3.5).

	AE	AEE	IE	NIE	time
histeq	1,3900	11,4851	0,0893	0,0889	126,84
imadjust	0,6673	2,3806	0,0408	0,04048	128,76
PlusConstant	1,3572	133,7232	0,2749	0,2742	126,23
increasedRed	1,0580	7,3100	0,1573	0,1570	126,25

Tabelle 3.4: Fehlermaße auf Hydrangea mit Belichtungsterm

	AE	AEE	IE	NIE	time
histeq	1,3074	22,4927	0,1175	0,1172	88,72
imadjust	0,6400	5,3850	0,0705	0,0702	89,37
PlusConstant	1,3983	133,9846	0,2876	0,2869	88,84
increasedRed	1,4377	38,4653	0,1438	0,1436	89,00

Tabelle 3.5: Fehlermaße auf Hydrangea ohne Belichtungsterm

Es ist zu erwähnen, dass es ein paar Werte gibt, bei denen das Standardverfahren etwas besser ist. Bei größeren Unterschieden ist jedoch immer das Verfahren mit Belichtungsterm besser gewesen. Insgesamt lässt sich also festhalten, dass das Modell Belichtungsunterschiede tatsächlich gut ausgleicht.

In der Umsetzung mit einem Rauschterm ergibt sich auf den 7 Testbildern zunächst der gleiche, für die Erweiterung negative ausfallende, Eindruck. Außer auf dem Bild hand sind alle Fehlermaße minimal schlechter als beim Standardverfahren. Um die Reaktion auf Störungen zu ermitteln habe ich deshalb einen zusätzlichen Test mit gaußschem Rauschen ($s = 0.01$ und $s = 0.02$) und Salt and Pepper Noise auf den Bildern `camera-motion`, `Rubberwhale` und `Urban 3` durchgeführt. Im Gegensatz zum positiven Ergebnis des Verfahrens mit Belichtungsterm bietet dieser Ansatz keinen Mehrwert. Alle Fehler des Standardverfahrens sind geringer (s. Tabelle 3.6 und 3.7).

Insgesamt ist es nicht besonders verwunderlich, dass das Verfahren mit Rauschterm keine Verbesserung darstellt, da der Fehler durch den neuen Term $|u|_\epsilon$ die Reduktion des Fehlers im Datenterm fast ohne Veränderung widerspiegelt.

	AE	AEE	IE	NIE	time
Gauß $\sigma = 0.01$	0,3364	1,0220	0,0394	0,0389	186,45
Gauß $\sigma = 0.02$	0,4296	1,2302	0,0459	0,0452	184,06
Salt and Pepper	0,4410	1,2573	0,0423	0,0416	183,85

Tabelle 3.6: Fehlermaße auf cameramotion mit Rauschterm

	AE	AEE	IE	NIE	time
Gauß $\sigma = 0.01$	0,2059	0,7685	0,0391	0,0386	123,45
Gauß $\sigma = 0.02$	0,2429	0,8471	0,0450	0,0443	124,95
Salt and Pepper	0,2311	0,8207	0,0401	0,0394	126,12

Tabelle 3.7: Fehlermaße auf cameramotion ohne Rauschterm

Fazit der Experimente ist somit, dass die Umsetzung der Coarse-to-Fine Pyramide mit Skalierung, der Variante ohne Skalierung deutlich überlegen ist. Des Weiteren haben sich für den weiteren Vergleich der anisotrope Tensor, und bei Störungen der Ansatz mit Belichtungsterm als gute Erweiterungen des Modells herausgestellt. Trotz der im Vergleich schlechteren Ergebnisse werde ich auch die adaptive Schrittweise für die weiteren Vergleiche verwenden, da der Unterschied nur sehr gering ist und die zu minimierende Energie sogar kleiner war.

3.6.4 Vergleich mit anderen Algorithmen

Für den Vergleich des Algorithmus habe ich neben den verschiedenen Erweiterungen die sich empirisch empfohlen haben, einige bekannte Verfahren zum Vergleich herangezogen. Dabei habe ich folgende Verfahren ausgewählt:

1. Lucas-Kanade Methode
2. Horn-Schunk Methode
3. Iterative Reweighted Least Squares (IRLS) von Ce Liu
4. Primal-Dual Algorithmus von Chambolle und Pock

Die Lucas-Kanade Methode ist ein sehr einfaches Verfahren, welches in der Informatik häufig genutzt wird, da es sehr schnell umzusetzen ist. Der Algorithmus beruht auf der Methode der kleinsten Quadrate für einige wenige charakteristische Punkte des Bildes. Um daraus einen dichtes Verschiebungsvektorfeld zu erhalten, müssen die Flüsse für diese Punkte auf das gesamte Bild interpoliert werden. Die hier genutzte Variante wurde von S.

Khan an der University of Central Florida umgesetzt (<http://crcv.ucf.edu/source/optical>). Es wurde dabei eine Coarse-to-Fine Pyramide verwendet. Die gewählten Parameter waren $numLevels = 3$, $windowsSize = 2$ und $iterations = 3$.

Bei der Horn-Schunck Methode wird, wie auch beim Primal-Dual Algorithmus, ein Energiefunktional minimiert, welches im Datenterm auf der Optical Flow Constraint Equation beruht. Im Unterschied zu unserer Umsetzung, wird der Gradient aber mit der L^2 -Norm regularisiert. Das Verfahren leitet sich aus der Euler-Lagrange Gleichung her und approximiert die Lösung durch ein Iterationsverfahren. Für weitere Details hierzu siehe [HS80]. Implementiert wurde das verwendete Verfahren von Mhod Khabart (<http://de.mathworks.com/matlabcentral/fileexchange/22756-horn-schunck-optical-flow-method>). Die verwendeten Parameter waren $\alpha = 0.01$ und $Iterationen = 1000$.

Das IRLS Verfahren von Ce Liu beruht auf einer Erweiterung der Idee des Verfahrens von Horn-Schunck. Die Ausgangslage ist das folgende Energiefunktional beruhend auf der Optical Flow Constraint Equation ρ :

$$E(u) = \int \alpha \phi(|\nabla u_1|^2 + |\nabla u_2|^2) + \psi(|\rho(u)|) dx$$

Liu verwendet die Funktionen $\psi = \phi = \sqrt{x^2 + \epsilon}$ mit $\epsilon = 0.001$. Ähnlich wie Horn-Schunck leitet er mit der Euler-Lagrange Gleichung ein Gleichungssystem her, das iterativ gelöst wird. Zusätzlich wird jedoch nach jeder Lösung des Gleichungssystems eine Gewichtung vorgenommen, die ein neues, leicht verändertes Gleichungssystem aufstellt. Liu fasst seinen Algorithmus in seiner Doktorarbeit schematisch zusammen (s. Abbildung 3.7).

-
- (a) Initialize $dU = 0$, $dV = 0$.
 - (b) Compute the “weight” Ψ' and Φ' based on the current estimate dU and dV .
 - (c) Solve the following linear equation

$$\begin{bmatrix} \Psi' \mathbf{I}_x^2 + \alpha \mathbf{L} & \Psi' \mathbf{I}_x \mathbf{I}_y \\ \Psi' \mathbf{I}_x \mathbf{I}_y & \Psi' \mathbf{I}_y^2 + \alpha \mathbf{L} \end{bmatrix} \begin{bmatrix} dU \\ dV \end{bmatrix} = - \begin{bmatrix} \Psi' \mathbf{I}_x I_z + \alpha \mathbf{L} U \\ \Psi' \mathbf{I}_y I_z + \alpha \mathbf{L} V \end{bmatrix} \quad (\text{A.12})$$
 - (d) If dU and dV converge, stop; otherwise, goto (b).
-

Abbildung 3.7: Pseudocode für Algorithmus von Liu. (vgl. [Liu09] S.145)

Die Implementierung des Algorithmus stellt Liu auf der Seite <https://people.csail.mit.edu/celiu/OpticalFlow/> für Forschungszwecke zur Verfügung. Das Gleichungssystem wird mit einem sukzessiven Überrelaxierungsschritt (engl. SOR) gelöst. Die Parame-

ter für das Verfahren wurden der beigefügten Demo entnommen: $\alpha = 0.012$, $ratio = 0.75$, $minWidth = 20$, $nOuterFPIterations = 7$, $nInnerIterations = 1$ und $nSOR - Iterations = 30$.

Als letzten Vergleich nutze ich den von Chambolle und Pock umgesetzten Primal-Dual Algorithmus. Das zugrundeliegende Modell entspricht dem Modell mit Belichtungsterm. Außerdem wurden sowohl der Regularisierungs- als auch der Datenterm mit der L^1 -Norm umgesetzt (vgl. [CP10]). Der Algorithmus kann zusammen mit einer Reihe von anderen Algorithmen, die andere Minimierungsprobleme lösen, auf der Seite <http://gpu4vision.icg.tugraz.at/index.php?content=downloads.php> heruntergeladen werden.

Die Evaluation wird in erster Linie mit Bildern aus der Middlebury Database (vision.middlebury.edu/flow/data/) durchgeführt. Dazu habe ich eine Reihe von Bildern ausgewählt und sie den Kategorien der möglichen Herausforderungen zugeordnet (vgl. Kapitel 3.6.2 und Tabelle 3.8). Anschließend habe ich für je zwei Bilder mit allen Optical Flow Algorithmen den Fluss und die Fehler berechnet (für einige Bilder ist kein Groundtruth-Fluss für Vergleiche verfügbar).

Bild	Kategorie	GT-Fluss
Dimetrodon	3, (4,) 7, 12	Ja
Grove2/Grove3	3, 4, 5, (6)	Ja
Hydrangea	3, 4, (6,) 8, 12	Ja
Rubberwhale	(2,) 3, 4, 7, 8, 12	Ja
Urban2/Urban3	3, 4, 5, 8	Ja
Venus	3, (4,) (6,) 8, 12	Ja
Basket Ball	2, (3,) (4,) 5, (6,) 7, 11, 12	
Beanbags	3, 4, 5, 6, 7, 11, 12	
Dumptruck	(2,) 3, 4, 5, (9,) 10, 12	
Evergreen	3, 4, 5, 8, 12	
Mequon	(2,) 3, 4, 6, 7, 12	
MiniCooper	(2,) 3, 4, 5, (6,) 7, 8, 10, 12	
Wooden	3, 4, 12	

Tabelle 3.8: Einteilung der Testbilder in die Kategorien

Da die experimentellen Erweiterungen bereits mit meinem Primal-Dual Algorithmus verglichen wurden, und die Fehlerwerte auf den Bildern wieder ein ähnliches Bild ergeben, habe ich sie in den Darstellungen zur generellen Situation der Übersichtlichkeit halber ausgeklammert. In den kleineren Vergleichen werde ich noch detaillierter auf die Verfahren eingehen.

Als Erstes werden die Interpolation Error der verschiedenen Algorithmen auf allen Testbildern verglichen. Abbildung 3.8 stellt für jeden Algorithmus die Fehler anhand einer Kurve dar. Dabei fallen zunächst die beiden schlechtesten Verfahren auf, die Horn-Schunk

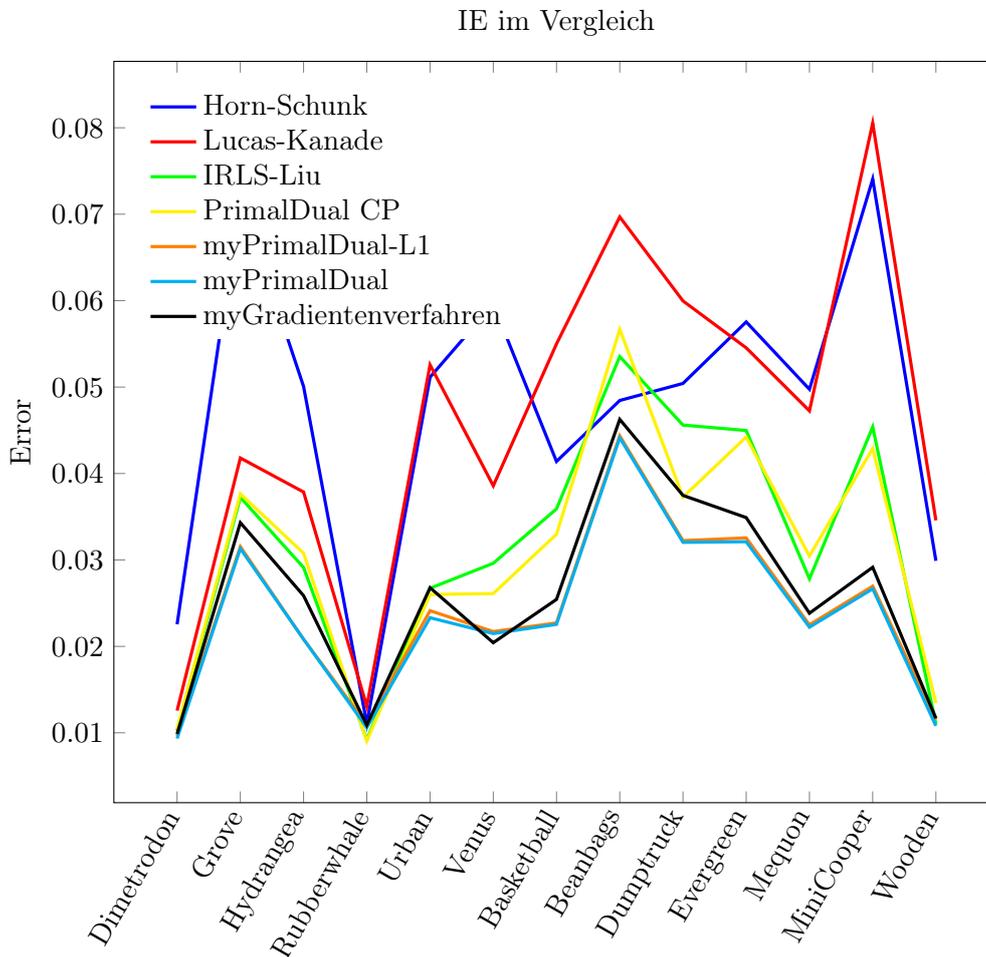


Abbildung 3.8: IE der Verschiedenen Verfahren berechnet auf allen getesteten Bildern

und die Lucas-Kanade Methode. Beide Verfahren schneiden beim IE fast immer schlechter ab als alle anderen Verfahren. Das ist im Grunde nicht verwunderlich, da die anderen umgesetzten Verfahren deutlich komplexer sind. Direkt miteinander verglichen, lässt sich feststellen, dass das Lucas-Kanade Verfahren auf den Bildern mit Groundtruth-Daten etwas besser, und auf den Bildern ohne Groundtruth-Daten etwas schlechter abschneidet. Insbesondere auf den Bildern, die in einer weniger optimalen Umgebung aufgenommen wurden (also Basketball, Beanbags, Dumptruck und Mini Cooper), erzielt die Lucas-Kanade Methode etwas schlechtere Ergebnisse. Nimmt man die Liste der Herausforde-

rungen zu Hilfe, fallen bei den Bildern einige Herausforderungen auf, die bei den anderen Bildern nicht auftauchen. Demnach hat die Lucas-Kanade Methode mit texturlosen Regionen, nicht-lambertschen Reflexionen und Bewegungsunschärfen Probleme. Das deutet auf einen Nachteil der lokalen Optimierung im Vergleich zur globalen Optimierung hin. Die genannten Effekte scheinen vor allem lokale Veränderungen hervorzurufen.

Die zweite Gruppe von Verfahren, deren Fehler sich fast identisch verhalten, umfasst den Primal-Dual Algorithmus von Chambolle und Pock und den IRLS-Algorithmus von Liu. Dass der Minimierungsalgorithmus von Liu sehr ähnliche Ergebnisse liefert wie der Primal-Dual Algorithmus, ist dabei sehr bemerkenswert, denn dadurch wird der Blick auf die Gemeinsamkeiten der beiden Algorithmen gelenkt, das gewählte Modell. Zwar wird bei Liu die L^1 -Norm durch die Funktion $\phi = \sqrt{|x|^2 + \epsilon^2}$ ersetzt, aber für das gewählte $\epsilon = 0.001$ sind diese fast identisch.

In Zahlen lässt sich diese Nähe vor allem durch den durchschnittlichen AEE unterstreichen. Dieser liegt zwischen den beiden Verfahren auf allen getesteten Bildern bei 0.0843. Dieser Wert ist deutlich unter den durchschnittlichen AEEs zum Groundtruth-Fluss, der beim Primal-Dual Algorithmus von Chambolle und Pock bei 0,3174 (niedrigster erreichter Wert) und bei meinem Primal-Dual Algorithmus bei 0,6915 lag (diese Durchschnitte natürlich nur auf den GT-Bildern). Auch der Vergleich der Flussbilder (s.Abb 3.11) unterstreicht, trotz kleiner Ausreißer (Basketball: AEE=0,2745, Beanbags AEE=0,2380), diesen Eindruck. In der Abbildung sind ein gutes Paar und ein Ausreißer zu sehen. Generell ist der Unterschied zwischen den Fehlern minimal. Nur auf den Bildern Dumptruck, Venus und Basketball scheint der Primal-Dual Algorithmus von Chambolle Pock etwas bessere Ergebnisse zu erzielen. Die Gründe dafür konnte ich nicht eindeutig klären. Beim Bild Venus scheint der Algorithmus von Liu zu stark zu regularisieren, während bei den anderen beiden Bildern vermutlich die Unstetigkeiten und Überdeckungen zu unterschiedlichen Ergebnissen geführt haben. Ich denke, man sollte diese Unterschiede aber nicht zu stark interpretieren.

Die dritte Gruppe stellt meinen Primal-Dual Algorithmus mit Huber-Norm und meinen Primal-Dual Algorithmus mit L^1 -Norm dar. Die Fehler-Kurven sind dabei fast nicht zu unterscheiden. Und auch die subjektive Betrachtung der farbcodierten Flussbilder lässt keine wesentlichen Unterschiede erkennen. Das Einzige, was sich ändert, sind die Schattierungen in Flächen mit leichtem Verlauf und die Weichzeichnung an den Kanten.

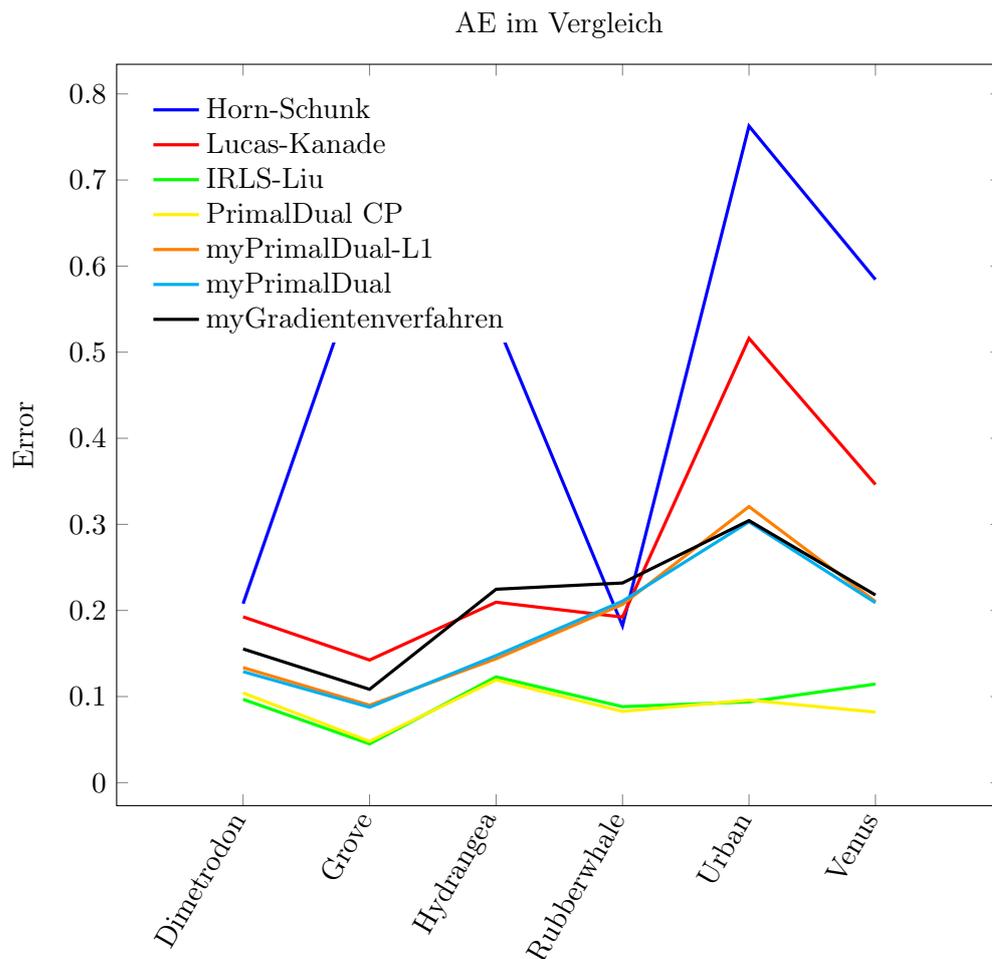


Abbildung 3.9: AE einer Auswahl von Verfahren auf allen Bildern mit Groundtruth-Flow

Eine Bewertung gegeneinander ist damit bis jetzt nicht möglich. Interessant ist aber der Vergleich dieser Gruppe mit der zweiten Gruppe. Das Diagramm der IE verdeutlicht, dass meine Implementierung des Verfahrens mit L^1 -Norm und Huber-Norm immer etwas besser abschneidet. Leider fällt der Vergleich der beiden Groundtruth-Fehler genau entgegengesetzt aus. Der Unterschied ist aufgrund der doppelten Umsetzung des Primal-Dual Verfahrens vermutlich auf Gewichtungparameter und weniger auf die genaue Umsetzung der L^1 nahen Norm (L^1 -Norm, Huber-Norm, Lius Variante) oder den Minimierungsalgorithmus zurückzuführen. Diese Erkenntnis rückt, wie schon die Analyse der zweiten Gruppe, das verwendete Modell näher in den Mittelpunkt. Betrachtet man jedoch die Umsetzung mit dem Gradientenverfahren, die wie in Kapitel 3.3.2 beschrieben, auch auf dem gleichen Modell beruht, so ist der Fortschritt der Algorithmen deutlich erkennbar.

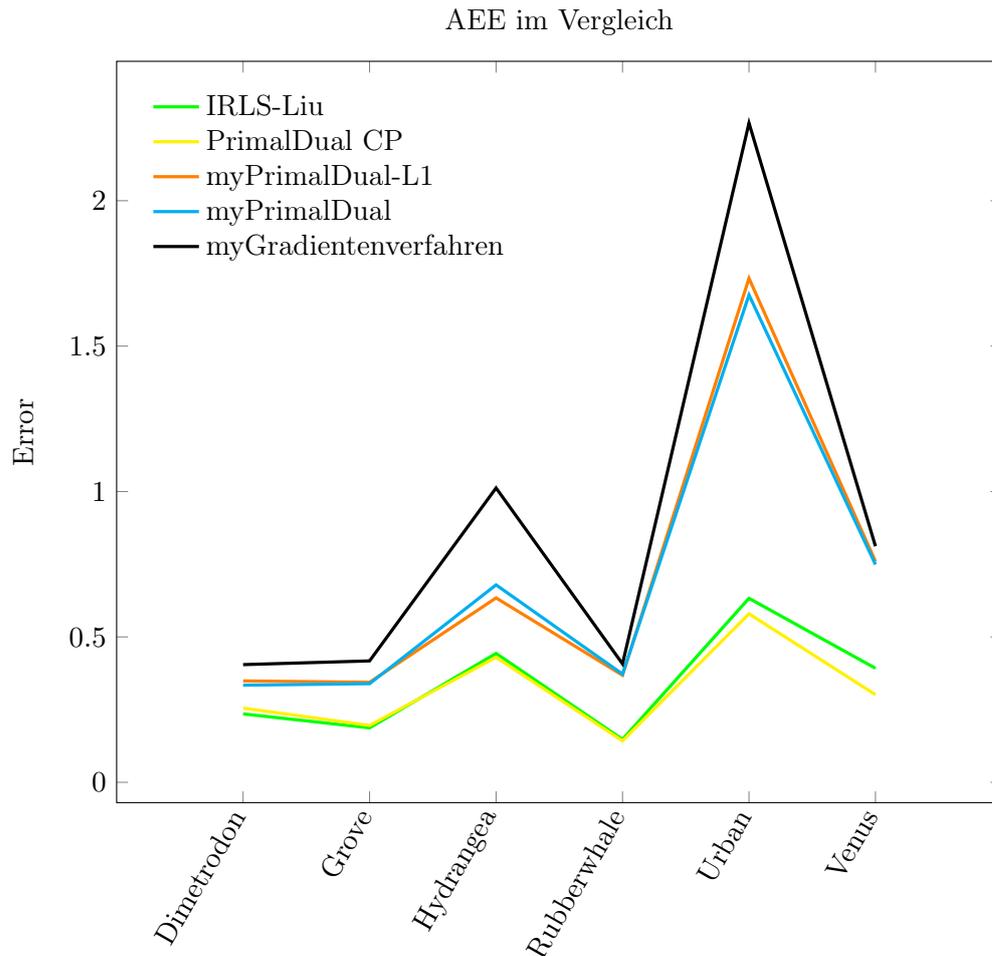


Abbildung 3.10: AEE einer Auswahl von Verfahren auf allen Bildern mit Groundtruth-Flow

Mein Primal-Dual Algorithmus erzielt in allen Bereichen bessere Ergebnisse als das Gradientenverfahren. Es ist also nicht komplett unwichtig, welcher Algorithmus verwendet wird, aber ganz sicher auch nicht am Wichtigsten.

Das Positive an dem auftretenden Unterschied ist, dass ein Vergleich der Superresolution Bilder mit den verschiedenen Optical Flow Ergebnissen möglicherweise einen Hinweis liefern kann, welches Fehlermaß für die Fluss Optimierung in Hinsicht auf das Superresolution Problem besser geeignet ist, die Groundtruth-Fehler oder die Interpolation-Fehler. Nachdem die Grenzen des Modells ausführlich besprochen wurden, folgt nun eine Einordnung der experimentellen Erweiterungen in das Gesamtbild. Wie im Diagramm mit dem IE zu sehen ist (s. Abb 3.12), bestätigt sich die Feststellung, dass der Algorithmus mit dem anisotropen Tensor bezüglich dieses Fehlers die besten Ergebnisse liefert. Außerdem zeigt



(a) Dimetrodon - IRLS-Liu



(b) Dimetrodon - PrimalDual CP



(c) Beanbags - IRLS-Liu



(d) Beanbags - PrimalDual CP

Abbildung 3.11: Vergleich Flussbilder IRLS-Liu und PrimalDualCP

sich hier, dass die adaptive Schrittweite bezüglich des IE ein ähnliches Verhalten aufweist wie der Primal-Dual Algorithmus von Chambolle und Pock bzw. wie das IRLS-Verfahren von Liu. Auch die Größenordnung des Fehlers ist die gleiche. Die anderen Erweiterungen siedeln sich im Bereich dazwischen an. Leider zeigen die Groundtruth-Fehler der Umsetzung mit adaptiver Schrittweite, dass sich das Ergebnis grundlegend vom Algorithmus von Liu und vom Algorithmus von Chambolle und Pock unterscheidet. Es lässt sich lediglich feststellen, dass das adaptive Verfahren etwas schlechtere Fehlerwerte erzielt. Auch eine endgültige Bewertung der Umsetzung mit anisotropem Tensor, ist aufgrund der Groundtruth-Fehler im Bezug auf die Verfahren von Liu und Chambolle und Pock noch nicht möglich. Dies kann hoffentlich mit der Auswertung des Superresolution-Problems nachgeholt werden.

Der Vollständigkeit halber habe ich im folgenden außerdem eine kurze Untersuchung der Standardoperationen vorgenommen. Dazu wurde zu den Operationen Translation, Rotation und Skalierung jeweils ein Vektorfeld berechnet, das diese umsetzt. Anschließend erzeugte ich damit aus einem 100×100 großen Ausschnitt des Bildes Urban2 neue künstliche Zielbilder. Die Fehler der Algorithmen bei der Berechnung des Flusses vom Originalbild zum künstlichen Zielbild, habe ich dann für den Vergleich in einer Grafik ver-

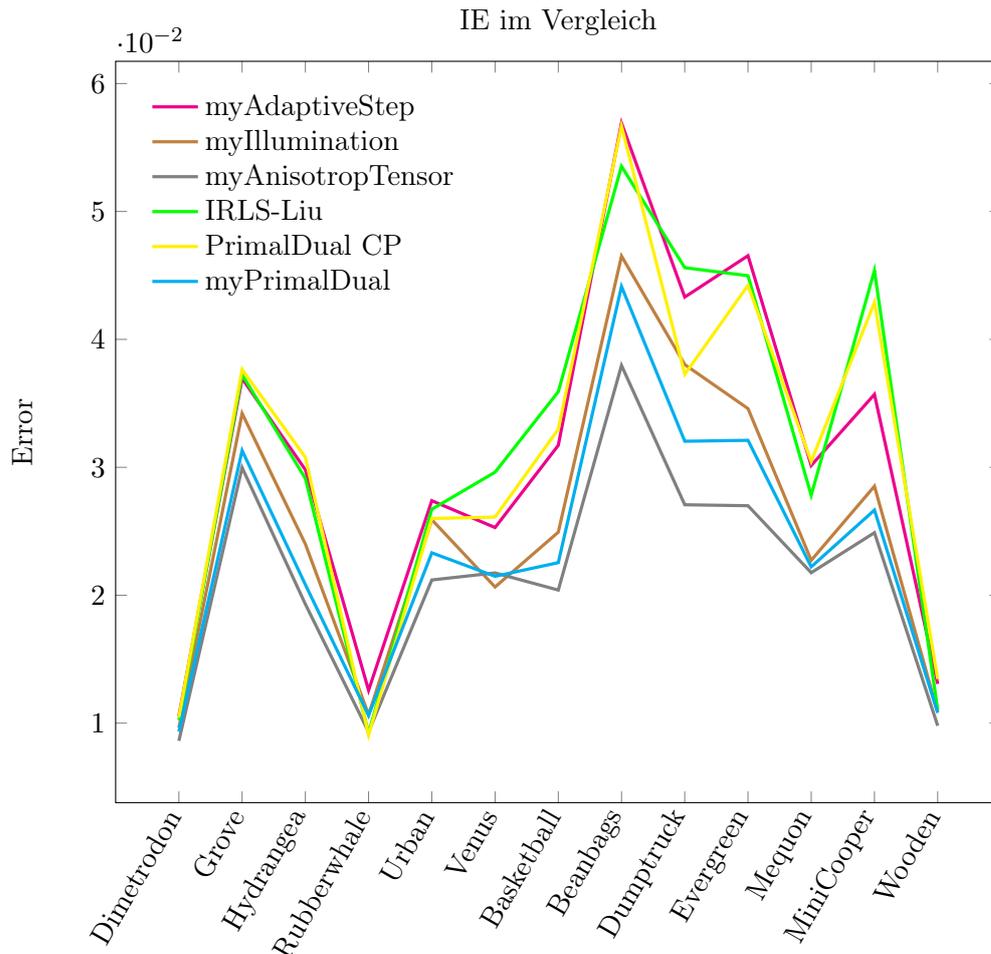


Abbildung 3.12: IE der Erweiterungen im Vergleich zu den Verfahren von Liu und von Chambolle und Pock auf allen getesteten Bildern

anschaulicht. Die Ergebnisse dieser Untersuchung decken sich dabei mit den vorherigen Ergebnissen. Interessant ist jedoch, dass alle Verfahren mit Skalierungen und Rotationen deutlich größere Schwierigkeiten haben. Ich bewerte es deshalb als sinnvoll, über eine Modellerweiterung nachzudenken, die den Winkel des Flusses bei der Optimierung beeinflusst. Ich vermute, dass dadurch auch bessere Groundtruth-Fehler erzielt werden könnten.

Fazit der Evaluation ist somit, dass bei den Verfahren der hier vorliegenden Güte, eher das Modell an seine Grenzen zu kommen scheint, als das Verfahren selbst. Obwohl die Unterschiede in den Verfahren nicht vergessen werden dürfen und auf einfachere Um-

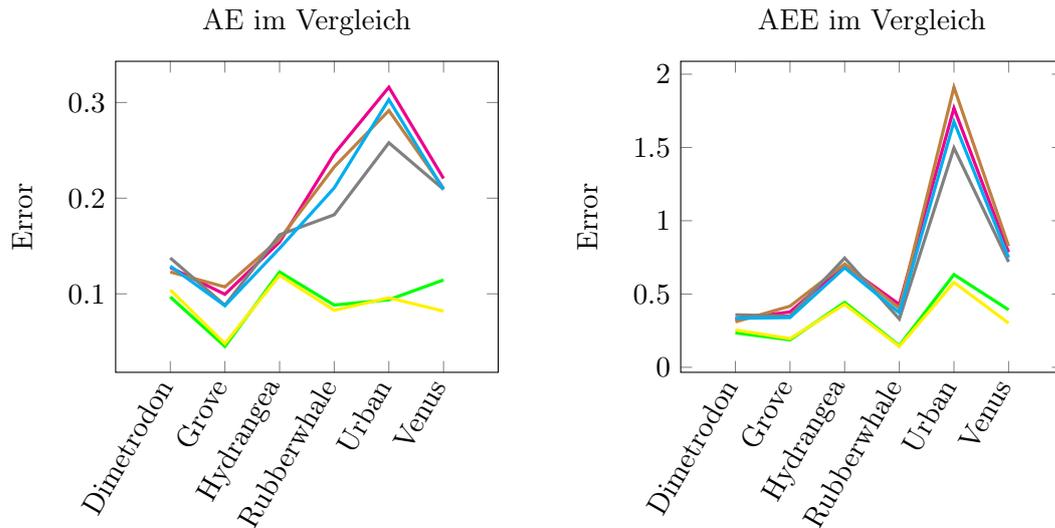


Abbildung 3.13: Groundtruth-Fehler der Erweiterungen im Vergleich zu den Verfahren von Liu und von Chambolle und Pock auf allen Groundtruth-Bildern. Die Farben entsprechen denen aus Abbildung 3.12

setzungen wie das Gradientenverfahren verzichtet werden sollte. Für eine Optimierung der Lösung geht der sinnvollere Weg über eine Anpassung des Modells. Unterstützt wird meine Aussage dabei von der verbesserten Performance mit dem anisotropen Tensor. Außerdem sollten sich weiterhin die Modellerweiterungen nicht auf die Wahl der Norm zur TV-Regularisierung beschränken, sondern beispielsweise lokale Einflussgrößen aufweisen oder ein weiteres Argument wie den Winkel oder zum Beispiel die Krümmung sinnvoll übernehmen. Dadurch könnte man die etwas schlechtere Performance auf den Standardoperationen Rotation und Skalierung möglicherweise ausgleichen. Die gute Übertragbarkeit des Primal-Dual Algorithmus könnte da eine Hilfe sein. Außerdem besteht die Hoffnung, in der Analyse des Superresolution Problems eine Aussage treffen zu können, welcher Fehler das beste Maß zur Optimierung darstellt.

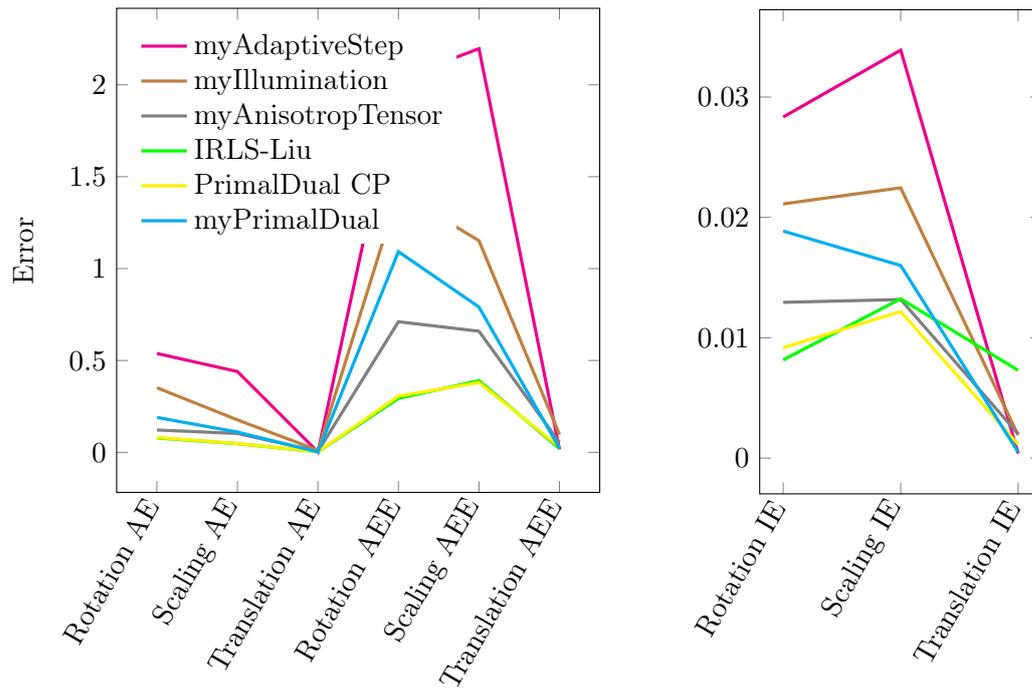


Abbildung 3.14: Standardoperationen im Vergleich

4 Das Superresolution Problem

4.1 Motivation

Es gibt viele Anwendungsgebiete, in denen hochauflösende Fotos wünschenswert sind. In der Praxis ist es jedoch häufig problematisch, die Fotos in der gewünschten Auflösung aufzunehmen. Ein Grund dafür liegt in der Sensortechnik. Soll die Auflösung eines Bildes erhöht werden, indem die Größe der Pixel reduziert wird und mehr Pixel auf einen gleichgroßen Sensor gesetzt werden, reduziert sich auch die Lichtmenge pro Pixel. Dadurch wird der Stromfluss der Sensoren aufgrund der Potentialbarriere starken Schwankungen ausgesetzt, welche wiederum in sogenanntem Schrotrauschen resultieren. Bei einer Mindestgröße der Pixel, muss also die Größe des Sensorchips erhöht werden, um hochauflösende Fotos zu gewinnen. Damit solch große Sensorchips auch genutzt werden können, muss zusätzlich auch die Kamera vergrößert werden, da sonst der Randbereich des Sensors schlecht belichtet wird und somit unbrauchbare Werte ermittelt werden (Problem des Randlichtabfalls). Der Nachteil dieser sehr großen Kameras ist gut vergleichbar mit den Nachteilen der ersten Mobiltelefone. Sie sind unhandlich und sehr kostspielig. Es gibt also aus technischer Sicht gute Gründe, andere Möglichkeiten für die Erzeugung hochauflösender Bilder zu entwickeln. Eine Idee ist zum Beispiel, mehrere nicht hochauflösende Bilder (auch niedrigauflösend genannt) aufzunehmen und durch geschickte Überlagerung der Bilder einen Informationsgewinn zu erzielen, der es ermöglicht ein hochauflösendes Bild zu berechnen. Das mathematische Problem, das sich mit dieser Aufgabenstellung befasst, nennt man Superresolution Problem. (Es sei an dieser Stelle angemerkt, dass dies nicht das einzige Aufgabenfeld ist, welches mit Superresolution bezeichnet wird.)

Das Superresolution Problem findet zahlreiche Anwendungsgebiete. Ein einfaches Beispiel ist die Verkehrsüberwachung mit Videokameras. Dort werden viele Videokameras zum Filmen des Verkehrs genutzt. Kleine Ausschnitte des Bildes sollen dabei häufig hochauflösend dargestellt werden, z.B. um das Kennzeichen eines Autos lesen zu können. Dazu wird dann eine Sequenz mit Bildern von dem gewünschten Auto ausgewählt und

ein hochauflösendes Bild berechnet, das das Kennzeichen leserlich darstellt. Ein anderes Beispiel kommt aus der medizinischen Bildgebung. Bei der Untersuchung von Schlaganfällen werden mehrere diffusionsgewichtete MRT-Bilder aufgenommen, aus denen mithilfe der Superresolution Technik ein hochauflösendes Bild berechnet werden kann. Dadurch können auch kleine Verengungen der Blutbahnen im Gehirn frühzeitig erkannt werden (vgl.[PY01]). Weitere Anwendungsgebiete sind zum Beispiel die Mikroskopie oder Satellitenbilder.

4.2 Definition des Superresolution Problems

Definition 4.2.1 (Superresolution Problem). *Das Superresolution Problem wird durch folgendes konvexes Minimierungsproblem definiert: ($\delta, \epsilon > 0$)*

$$\min_{\hat{u}} \left\{ \frac{\mu}{\xi^2} \|\nabla \hat{u}\|_{\epsilon} + \frac{1}{n} \sum_{i=1}^n \|\mathbf{DBW}_i \hat{u} - \check{f}_i\|_{\delta} \right\} \quad (4.1)$$

mit $\|\cdot\|_{\epsilon}, \|\cdot\|_{\delta} : \text{Huber-Norm}$

$\mathbf{D}, \mathbf{B}, \mathbf{W}_i : \text{Operatoren}$

$\hat{u} : \text{Superresolution Bild}$

$\mu : \text{Gewichtungsparameter}$

$\check{f}_i : \text{Low-Resolution Bild}$

$\xi : \text{Skalenverfeinerung}$

4.2.1 Die verwendeten Operatoren

Um das Superresolution Problem überhaupt sinnvoll aufstellen zu können, sind zwei Operatoren unerlässlich, der Downsampling- und der Warming-Operator. Der Downsampling-Operator ist deshalb unerlässlich, weil er die Vergleichbarkeit der verschieden großen Bilder ermöglicht. Ohne ihn kann die Auflösung des Lösungsbildes nicht erhöht werden. Auf den Warming-Operator kann man nicht verzichten, da er die im Optical Flow berechnete Registrierung des Referenzbildes mit den anderen niedrigauflösenden Bildern umsetzt. Häufig ist es am einfachsten die Operatoren auf Bildern als Matrizen anzugeben. Dazu muss das Bild aber zunächst in einen Vektor transformiert werden. Der Index des Vektors wird im Folgenden als Vektorkoordinate bezeichnet. Dazu passend wird das Indexpaar

des Bildes in Matrixform Matrixkoordinate genannt. Matlab transformiert die Matrixkoordinaten mit dieser Formel in Vektorkoordinaten: $(i, j) \mapsto \#rows(j - 1) + i$.

Wenn beide Operatoren bekannt sind, kann das hochauflösende Bild mit einem vorhandenen Verschiebungsvektorfeld verschoben und dann auf die passende Größe der niedrigauflösenden Bilder verkleinert werden. Um den adjungierten Operator von K , der für den Primal-Dual Algorithmus aus Definition 2.4.4 notwendig ist, zu bestimmen, muss für jeden definierten Operator auch der adjungierte Operator bestimmt werden. Alle weiteren Operatoren sind für das Superresolution Problem optional. Es ist dabei sehr wichtig sicherzustellen, dass die adjungierten Operatoren auch in der diskreten Umsetzung tatsächlich adjungiert sind, da der Primal-Dual Algorithmus auf Störungen in dieser Bedingung mit fehlerhaften Mustern im Bild reagiert.

Zur Verbesserung des Ergebnisses in Hinsicht auf Bildfehler wie beispielsweise Rauschen wird noch der Blurring-Operator eingesetzt. Die niedrigauflösenden Eingabebilder sind Störungen ausgesetzt. Damit diese Störungen sich nicht auf das hochauflösende Ergebnisbild übertragen, wird das verschobene Ergebnisbild mit dem Blurring-Operator leicht geglättet. Dadurch wirken sich die Störungen auf kleine Umgebungen aus und werden somit reduziert.

Die genannten Operatoren, inklusive ihrer Adjungierten, werden nun der Reihe nach genauer erläutert.

Der Downsampling-Operator

Der Downsampling-Operator entspricht einer linearen Skalierung in vertikaler und einer, nicht notwendigerweise identischen linearen Skalierung, in horizontaler Richtung. Das hochauflösende Bild der Größe $\hat{M} \times \hat{N}$ wird zu einem niedrigauflösenden Bild der Größe $\check{M} \times \check{N}$ umgeformt. Das Problem wird in zwei Schritten gelöst. Zuerst wird eine Projektionsmatrix R für die vertikale Richtung bestimmt und dann eine für die horizontale Richtung H . Der Downsampling-Operator ist dann die Multiplikation der beiden Matrizen mit dem Bild I : $\mathbf{D}(I) = RIH^T$. Die Matrix R wird durch folgende Formel bestimmt:

$$R_{ki} = \mu([1 + dy(i - 1), 1 + dy i) \cap [k, k + 1)) \text{ mit } dy = \frac{\check{M}}{\hat{M}}, 1 \leq i \leq \hat{M} \text{ und } 1 \leq k \leq \check{M}.$$

Analog wird auch die Matrix für die horizontale Richtung aufgestellt:

$$H_{lj} = \mu([1 + dx(j - 1), 1 + dx j) \cap [l, l + 1)) \text{ mit } dx = \frac{\check{N}}{\hat{N}}, 1 \leq j \leq \hat{N} \text{ und } 1 \leq l \leq \check{N}.$$

Das Maß μ entspricht hier dem Lebesguemaß auf \mathbb{R} . Die dazu adjungierte Abbildung kann damit leicht bestimmt werden:

$$\langle \mathbf{D}(I), J \rangle = \langle RIH^T, J \rangle = \langle I, R^T JH \rangle = \langle I, \mathbf{D}^T(J) \rangle$$

Der adjungierte Downsampling-Operator ist demnach $\mathbf{D}^T(J) = R^T JH$. Für die effiziente Umsetzung wurde der Downsampling-Operator aufgeteilt in die Berechnung der beiden Matrizen R und H und dessen Anwendung auf das Bild. Es ist vielleicht noch interessant anzumerken, dass der adjungierte Operator mit dieser Definition einer bilinearen Interpolation entspricht.

Der Warping-Operator

Der Warping-Operator löst, mit einem zuvor im Optical Flow Problem berechneten Vektorfeld, die Gleichung $J(x) = I(x + v(x))$. Für ein festes v ist dies eine lineare Abbildung, da die bilineare Interpolation der neuen Bildkoordinate $x + v(x)$ in dem Bild I bestimmt wird. v gibt also an, welche maximal vier Pixel aus dem Bild I mit welcher Gewichtung in das Bild J überführt werden.

Diese Abbildung kann am einfachsten durch ihre Matrix angegeben werden. Das Bild muss dafür in Vektorkoordinaten umgerechnet werden. Um die Matrix zu bestimmen, werden die vier Pixel in Matrixkoordinaten bestimmt, zwischen denen die neue reelle Koordinate $x + v(x)$ liegt. Anschließend werden die Matrixkoordinaten der vier Punkte im Bild in Vektorkoordinaten umgerechnet und in der darstellenden Matrix mit ihrem Gewicht angegeben. Die Zeile entspricht der Vektorkoordinate der neuen Position x in Bild J und die Spalte den Vektorkoordinaten der vier benachbarten Punkte. Die Gewichte werden nach der Standardformel für die bilineare Interpolation berechnet:

$$\mathbf{W} : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}^{M \times N}, \quad x \mapsto \mathbf{W}_{ik}x$$

$$\mathbf{W}_{ik} = g(x_i + v(x_i)) = \begin{cases} (1 - \Delta_x)(1 - \Delta_y) & \text{falls } k = \text{vectorcoordinate}(tl) \\ (1 - \Delta_x)\Delta_y & \text{falls } k = \text{vectorcoordinate}(bl) \\ \Delta_x(1 - \Delta_y) & \text{falls } k = \text{vectorcoordinate}(tr) \\ \Delta_x\Delta_y & \text{falls } k = \text{vectorcoordinate}(br) \\ 0 & \text{sonst} \end{cases}$$

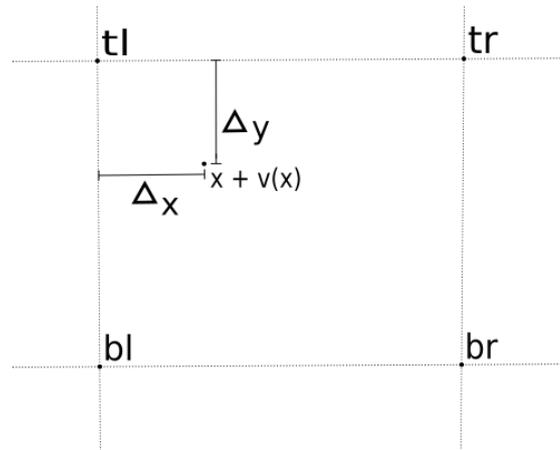


Abbildung 4.1: Bilineare Interpolatin

Der adjungierte Operator ist die Abbildung, die durch Transponieren der darstellenden Matrix berechnet wird.

Der Blurring-Operator

Der Blurring-Operator (dt. Glättungsoperator) wird durch eine Filterung mit einem Gaußfilter der Standardabweichung s modelliert. Dabei wird die Maske am Rand periodisch fortgesetzt. Zur Berechnung des Operators in Matrixform ging ich wie folgt vor. Als Erstes wird mit der Formel für die Dichtefunktion der gaußschen Glockenkurve eine maximale Filtermaske ungerader Seitenlänge mit Nullpunkt in der Mitte berechnet ($g(x, y) = \frac{1}{\sqrt{2\pi s^2}} \exp -\frac{x^2+y^2}{2s^2}$). Alle Werte unter dem Schwellwert $\Delta_s = 0.0001$ werden auf null gesetzt, sodass die Größe der Maske reduziert werden kann. Um die Helligkeit der Bilder nicht zu verändern, wird die Maske so skaliert, dass die Summe ihrer Einträge eins ergibt. Da die Filterung für jedes Pixel einer punktwweisen Multiplikation mit passend verschobener Maske entspricht, kann sie für das Bild in Vektorform als Matrixmultiplikation dargestellt werden. Diese Matrix hängt nur von der Standardabweichung s und der Bildgröße ab und kann vorberechnet werden.

Der große Vorteil an dieser Umsetzung ist, dass bei der Anwendung des Operators nur wenige Rundungsfehler auftreten und im Vergleich zur Umsetzungen mit der Fourier-Transformation oder bereits von Matlab implementierten Methoden, die Selbstadjungiertheit auch in empirischen Tests bis auf Maschinengenauigkeit besteht. Das ist wie bereits erwähnt ein wichtiges Kriterium für den Primal-Dual Algorithmus.

4.3 Lösungsverfahren mit dem Primal-Dual Algorithmus

Um den Primal-Dual Algorithmus auf das vorliegende Problem anwenden zu können, muss es zunächst umformuliert werden. Dazu wird die LF-Transformation des Huber-Betrags beim Regularisierungs- und beim Datenterm verwendet. Die Faktoren $\frac{\mu}{\xi^2}$ und $\frac{1}{n}$ werden zu $\tilde{\mu} = \frac{n\mu}{\xi^2}$ zusammengefasst. Dies führt uns zu folgendem Problem:

$$\min_{\hat{u}} \sup_{p, q} \left\{ \int_{\Omega} \langle \nabla \hat{u}, p \rangle - \frac{\epsilon}{2\tilde{\mu}} |p|^2 - \chi_{\{|p| \leq \tilde{\mu}\}}^{\infty} + \sum_{i=1}^n \left(\langle q_i, \mathbf{DBW}_i \hat{u} - f_i \rangle - \frac{\delta}{2} |q_i|^2 - \chi_{\{|q_i| \leq 1\}}^{\infty} \right) \right\}$$

Die Operatoren K, F^* ($G = 0$) werden wie folgt definiert:

$$\begin{aligned} K : \hat{X} &\rightarrow \hat{X}^2 \times \check{X}^n, & \hat{u} &\mapsto \begin{pmatrix} \nabla \hat{u} \\ \mathbf{DBW}_i \hat{u} \end{pmatrix} \\ K^T : \hat{X}^2 \times \check{X}^n &\rightarrow \hat{X}, & \begin{pmatrix} p & q_1 & \dots & q_n \end{pmatrix} &\mapsto -\operatorname{div}(p) + \mathbf{W}_i^T \mathbf{B}^T \mathbf{D}^T(q) \\ F^* : \hat{X}^2 \times \check{X}^n &\rightarrow \mathbb{R}, & \begin{pmatrix} p & q_1 & \dots & q_n \end{pmatrix} &\mapsto F_p^*(p) + \sum_{i=1}^n F_{q_i}^*(q_i) \\ F_p^* : \hat{X}^2 &\rightarrow \mathbb{R}, & p &\mapsto \frac{\epsilon}{2\tilde{\mu}} \|p\|^2 + \chi_{\{|p| \leq \tilde{\mu}\}}^{\infty} \\ F_{q_i}^* : \check{X} &\rightarrow \mathbb{R}, & q_i &\mapsto q_i \cdot f_i + \frac{\delta}{2} \|q_i\|^2 + \chi_{\{|q_i| \leq 1\}} \end{aligned}$$

Um die Iterationschritte zu bestimmen, muss der Resolventenoperator von F^* und G berechnet werden. $G \equiv 0$ hat den Resolventenoperator $(\mathbf{1} + \tau \partial G)^{-1} = \mathbf{1}$. Der Resolventenoperator von F^* wird im Beispiel nach Definition 2.4.3 berechnet. Dabei ist zu beachten, dass die $F_{q_i}^*$ nicht wie im Primal-Dual Algorithmus gefordert positiv sind, aber durch die Addition der Konstanten $\frac{1}{2\delta} \|f_i\|^2$ jeweils positiv gemacht werden können. Auf die Lösung des Problems sowie die Berechnung der Resolventenabbildung haben die Konstanten wegen der Minimierungen keinen Einfluss. Deshalb werden sie in der Problembeschreibung nicht aufgeführt. Mit dem Primal-Dual Algorithmus erhält man für das Superresolution

Problem also folgende Iterationsschritte:

$$\begin{aligned}
 p^{n+1} &= \prod_{B_0^\mu} \left(\frac{p^n + \sigma \nabla \bar{u}^n}{1 + \frac{\sigma \epsilon}{\mu}} \right) \\
 q^{n+1} &= \prod_{B_0^1} \left(\frac{q_i^n + \sigma (\mathbf{DBW}_i \bar{u}^n - f_i)}{1 + \sigma \delta} \right) \\
 \hat{u}^{n+1} &= \hat{u}^n - \tau \left(-\operatorname{div} p^{n+1} + \sum_{i=1}^n (\mathbf{W}_i^T \mathbf{B}^T \mathbf{D}^T q_i^{n+1}) \right) \\
 \bar{u}^{n+1} &= 2\hat{u}^{n+1} - \hat{u}^n
 \end{aligned}$$

4.4 Diskretisierung und Implementierung

Die Bilder werden analog zum Optical Flow Problem diskretisiert. Ω wird unabhängig von der Bildgröße durch ein kartesisches Gitter modelliert:

$$\Omega = \{(r, c) | 1 \leq r \leq M, 1 \leq c \leq N \text{ mit } i, j, M, N \in \mathbb{N}\} \quad .$$

Die unterschiedlich großen Bilder unterscheiden sich durch unterschiedliche Räume Ω . Der Raum der hochauflösenden Bilder wird mit $\hat{X} = C(\Omega_{SR})$ bezeichnet und der Raum der niedrigauflösenden Bilder entsprechend mit $\check{X} = C(\Omega_{LR})$. Das Skalarprodukt ist jeweils das Standardskalarprodukt des isomorphen reellen Vektorraums $\mathbb{R}^{\hat{M} \times \hat{N}}$ bzw. $\mathbb{R}^{\check{M} \times \check{N}}$.

Der Gradient wird wie beim Optical Flow Problem mit Vorwärtsdifferenzen und Neumann-0-Randwerten diskretisiert. Der Divergenzoperator wird so definiert, dass die Gleichung $\langle \nabla I, v \rangle = \langle I, -\operatorname{div}(v) \rangle$ gilt. Beide Operatoren können in Kapitel 3.4 nachgeschaut werden.

Die im Datenterm verwendeten Operatoren werden genauer in Kapitel 4.2.1 beschrieben. Die Huber-Norm wird wie in Definition 2.3.1 diskretisiert:

$$\|v\|_\epsilon = \sum_{i,j=1}^{M,N} |v_{(i,j)}|_\epsilon$$

$$\text{mit } |v_{(i,j)}|_\epsilon = \begin{cases} \frac{|v_{(i,j)}|^2}{2\epsilon} & \text{falls } |v_{(i,j)}| \leq \epsilon \\ |v_{(i,j)}| - \frac{\epsilon}{2} & \text{falls } |v_{(i,j)}| > \epsilon \end{cases} \quad \text{für } v_{(i,j)} \in \mathbb{R}^n \text{ und } 1 \leq i, j \leq M, N.$$

Implementiert wurde der Primal-Dual Algorithmus in Matlab mit den Parametern $\mu =$

0.1, $\epsilon = \delta = 0.01$ und $s = 0.5$ für den Blurring-Operator. Die Schrittweiten werden als $\tau = \frac{1}{L+1}$ und $\sigma = \frac{1}{L}$ gewählt. Die Norm L des Operators \mathbf{K} wird zur Laufzeit bestimmt. Dazu wird folgende Rechnung aufgestellt:

$$\|\mathbf{K}(u)\|^2 = \|\nabla u\|^2 + \sum_{i=1}^n \|\mathbf{DBW}_i(u)\|^2 \leq (8 + \sum_{i=1}^n \|\mathbf{DBW}_i\|^2) \|u\|^2$$

Die Operatoren $\mathbf{D}, \mathbf{B}, \mathbf{W}_i$ werden nach der Berechnung der Warping-Operatoren als Matrix aufgestellt und zu einer Matrix $\mathbf{H}_i = \mathbf{DBW}_i$ zusammengefasst. Die Operatornorm von \mathbf{K} wird durch die Berechnung der 2-Norm der Matrizen \mathbf{H}_i bestimmt.

4.5 Evaluation

4.5.1 Fehlermaße

Eine objektive Evaluation des Superresolution Problems ist sehr schwierig, da es sehr kompliziert ist festzustellen, wann ein Bild schärfer ist als ein anderes. In der Theorie gibt es einige Versuche geeignete Fehlermaße aufzustellen. Allerdings unterscheiden sich die Meinungen in Hinsicht auf deren Qualität häufig deutlich voneinander. Für meine Evaluation habe ich mich aufgrund der Arbeit von Nelson et al. für zwei Fehlermaße entschieden (vgl. [NBN12]). Die wesentliche Analyse habe ich dabei auf dem Mean-Square-Error (MSE) und dem Universal Image Quality Index (UQI) aufgebaut.

Der Mean-Square-Error ist für eine vorgegebene Optimallösung der Durchschnitt der quadrierten Pixelfehler:

$$MSE(I_O, I) = \sum_{i,j=1}^{M,N} \frac{(I_O(i,j) - I(i,j))^2}{MN}$$

Das Problem dieses Fehler ist, dass eine Optimallösung meist nur durch künstliche Testfälle zu generieren ist. Auch der von Wang et al. vorgestellte UQI (vgl. [WB02]) beruht darauf, das berechnete Ergebnis mit einer optimalen Lösung zu vergleichen. Nelson et al. haben in ihrer Arbeit festgestellt, dass der UQI bezüglich künstlich erzeugter Daten ein gutes Maß dafür ist, schlechte Ergebnisse entsprechend zu kennzeichnen. Die Berechnung funktioniert wie folgt:

$$UQI(I_O, I) = \frac{4\sigma_{I_O I} \bar{I}_O \bar{I}}{(\sigma_{I_O}^2 + \sigma_I^2)(\bar{I}_R^2 + \bar{I}^2)}$$

$$\begin{aligned}\bar{I}_O &= \frac{1}{MN} \sum_{i,j=1}^{M,N} I_O(i,j) & \bar{I} &= \frac{1}{MN} \sum_{i,j=1}^{M,N} I(i,j) \\ \sigma_{I_O}^2 &= \frac{1}{MN-1} \sum_{i,j=1}^{M,N} (I_O(i,j) - \bar{I}_O)^2 & \sigma_I^2 &= \frac{1}{MN-1} \sum_{i,j=1}^{M,N} (I(i,j) - \bar{I})^2 \\ \sigma_{I_O I} &= \frac{1}{MN-1} \sum_{i,j=1}^{M,N} (I_O(i,j) - \bar{I}_O)(I(i,j) - \bar{I})\end{aligned}$$

Dieser Fehler ist ein statistisches Maß zur Bemessung der Ähnlichkeit der Bilder. Der Wert dieses Fehlermaßes liegt zwischen null und eins. Der maximale Wert eins wird genau dann angenommen, wenn die beiden Bilder identisch sind. Genau wie der MSE liefern diese beiden Maße keine Auskunft darüber, welches Bild schärfer ist. Sie geben lediglich einen Hinweis darauf, wie gut sich die beiden Bilder entsprechen. Für die Evaluation bedeutet das, dass diese Fehlermaße idealerweise dann eingesetzt werden, wenn ein optimales Bild zur Verfügung steht, mit dem die berechneten Ergebnisse verglichen werden können.

Wenn kein optimales Vergleichsbild vorhanden ist werden die Bilder von mir subjektiv verglichen. Eine objektive Bewertungsmöglichkeit wäre zwar wünschenswert, es scheint jedoch bis jetzt kein geläufiges Fehlermaß zu geben, das diese Aufgabe zufriedenstellend löst.

4.5.2 Testdaten

Die Testbeispiele für die Bewertung des Superresolution Problems wurden in drei Gruppen gegliedert. In der ersten Gruppe wurde jeweils ein Bild ausgewählt, um daraus ein künstliches Bilderset zu erzeugen. Für die Erstellung wurde das Bild dann leicht verschoben, rotiert und anschließend in der Größe reduziert. Zusätzlich wurde eine Störung in Form von gaußischem Rauschen oder Salt and Pepper Rauschen hinzugefügt. Für diesen Test wurden die beiden Bilder cameraman und bird aus der Arbeit zur Performance Analyse von Nelson et al. verwendet (<http://www.deakin.edu.au/research/cisr/docs/1141868.zip>). Die Bilder wurden einmal um den Faktor zwei und einmal um den Faktor vier verkleinert. In beiden Fällen wurden 64 Bilder zur Rekonstruktion verwendet. Diese Problemstellung hat den Vorteil, dass das Originale Bild von vor der Bearbeitung somit zur Verfügung steht und als optimaler Vergleich dient.

Die zweite Gruppe von Beispielen sind die aus jeweils acht Bildern bestehenden Bildersets Mequon, Army und Evergreen aus der Middlebury Database (vision.middlebury.edu/flow/). Die idealen Aufnahmebedingungen dieser drei Bildersets haben nur wenige Störeinflüsse zur Folge. Sie eignen sich somit sehr gut als Testsets. Außerdem treten bei diesen drei Bildersets vollkommen unterschiedliche Herausforderungen für die Lösung des Optical Flow Problems und damit auch für das Superresolution Problem auf. Im Vergleich kann also festgestellt werden, wie die Performance von diesen Herausforderungen beeinflusst wird.

Die dritte Gruppe Beispielbilder sind jeweils elf Bilder umfassende Ausschnitte der Bildersets cameramotion und fish aus dem Motion Annotation Dataset von Liu (<http://people.csail.mit.edu/ce-liu/motionAnnotation/>). Wie bereits in der Evaluation vom Optical Flow erwähnt, sind diese Bildersets von Interesse, da sie unter weniger idealen Bedingungen aufgenommen wurden als die Bilder aus der Middlebury Database. Sie bieten somit eine sehr gute Ergänzung. Hinzu kommt außerdem, dass Liu in der Datenbank nicht nur für ein Bilderpaar einen Groundtruth-Fluss zur Verfügung stellt, sondern für alle aufeinanderfolgenden Bilderpaare.



Abbildung 4.2: Lösung des Superresolution Problems mit den zur Verfügung stehenden Groundtruth-Flüssen des Bildes cameramotion

Meine ursprüngliche Idee für die Evaluation des Superresolution Algorithmus lag darin, mit diesen Groundtruth-Flüssen ein Optimalbild zu berechnen, welches später für einen Vergleich genutzt werden kann. Leider sind die Groundtruth-Flüsse jedoch nicht genau genug, um daraus ein gutes Ergebnis zu berechnen. Der Umstand, dass der Fluss nicht von einem Ausgangsbild zu allen anderen Bildern berechnet wurde, verstärkt das Problem deutlich. Ein Beispiel für ein Superresolution Bild mit Groundtruth-Fluss gibt Abbildung 4.2 wieder. Die Abbildung zeigt deutlich, dass selbst ein bikubisch interpoliertes Bild deutlich hochauflösender ist. Die Evaluation auf den Bildersets cameramotion und fish wird deshalb ohne optimales Vergleichsbild durchgeführt.

Für die Auswertung ist es notwendig alle Bilder im Detail betrachten zu können. Da diese detaillierte Darstellung aller Bilder auf dem Papier nicht möglich ist, wurde darauf verzichtet die Bilder zu drucken. Für den Vergleich befinden sich auf der beigefügten CD alle Bilder der Auswertung, sowie die genutzten Testdaten.

4.5.3 Effekt der verschiedenen optischen Flüsse

Schon in der Auswertung der künstlichen Datensets bestätigt sich, dass eine objektive Bewertung mit Hilfe der Fehlermaße sehr schwierig ist. So erreicht bei den Bildern bird und cameraman mit gaußschem Rauschen die bikubische Interpolation fast immer die besten Ergebnisse der Fehlermaße UQI und MSE. Die Ergebnisse liegen jedoch auch sehr nah zusammen. Die maximale Differenz des UQI bei den Bildern mit Skalierungsfaktor zwei ist ca. 3,5%, bei den Bildern mit Skalierungsfaktor vier ca. 7,5%. Die subjektive Bewertung der Bilder spiegelt dieses Ergebnis allerdings nicht wieder. Vor allem bei den Bildern mit Skalierungsfaktor zwei, empfinde ich die bikubisch interpolierten Bilder als vergleichsweise schlecht. Die Struktur des Federkleids wird bei allen Verfahren gleich schlecht (also wahrscheinlich fast gar nicht) optimiert. Die Kanten der bikubischen Interpolation hingegen weisen starke Treppeneffekte und eine leichte Unschärfe auf. Dies ist bei den anderen Bildern nicht, oder zumindest weniger stark, zu sehen. Die Bilder mit Skalierungsfaktor vier weisen diesen glättenden Effekt auch auf, jedoch tendiert die Optimierung langsam ein wenig dazu Schlieren zu erzeugen. Besonders das Verfahren von Horn-Schunk ist davon betroffen.

Der Vergleich der Bilder untereinander ist ebenso problematisch wie der Vergleich zur bikubischen Interpolation. Der MSE und der UQI sind sich, was die Rangfolge der Verfahren angeht, bis auf in zwei kleinen zirkulären Vertauschungen, einig. Subjektiv würde ich dieser Rangfolge jedoch an einigen Stellen widersprechen. Deshalb halte ich es für

sinnvoll, nicht primär eine Rangfolge zu erstellen, sondern einige Charakteristika der verschiedenen Verfahren festzustellen und zu kritisieren. Besonders gut gefallen mir die sehr ähnlichen Ergebnisse von IRLS-Liu und Chambolle und Pock, da diese beiden Verfahren die Konturen sinnvoll glätten, nicht zu Schlieren neigen und die Details größtenteils erhalten. Fast genauso gut gefallen mir die Ergebnisse meiner Primal-Dual Umsetzungen, die sich in einigen Punkten allerdings deutlich anders verhalten. So werden die Konturen weniger stark geglättet, dafür aber die Details etwas besser ausgearbeitet. Bei dem Bild cameraman kann man dieses Verhalten vor allem in den Haaren erkennen, während es beim Bild bird am Übergang zum weißen Gefieder sichtbar wird. Auch Lucas-Kanade und Horn-Schunk erzielen passable Ergebnisse, obwohl beide zu leichten Überreaktionen neigen. Horn-Schunk zeigt wie bereits erwähnt die Tendenz zu Schlieren, während Lukas-Kanade bei stärkeren Skalierungen mit leichter Musterbildung auffällt. Bei den größeren Skalierungen fällt es mir schwer zu sagen, ob die bikubische Interpolation nicht doch etwas bessere Ergebnisse liefert als diese beiden Verfahren.

Insgesamt zeigt sich, dass die Verfahren auf gaußsches Rauschen nicht besonders gut reagieren können. Trotz nur geringer Reduktion des Informationsgehalts, sind die Ergebnisse ungefähr auf dem Level der bikubischen Interpolation.

Die Situation beim Salt and Pepper Rauschen in Bezug auf die bikubisch interpolierten Bilder ist grundlegend anders als beim gaußschen Rauschen. Der UQI der interpolierten Bilder liegt im Bereich von ca. 47% bis 61%, während auf den Bildern mit Skalierungsfaktor zwei die Ergebnisse der Verfahren im Bereich von 75% bis 99% liegen. Die Interpolation wird also deutlich schlechter bewertet, als die berechneten Lösungen. Auf den Bildern mit Skalierungsfaktor vier gibt es jedoch auch einige Ausreißer mit Werten des UQI im Bereich von 30% bis 45%. Das Bild der MSE Werte entspricht dem der UQI Werte qualitativ genau. Beide Fehlermaße deuten somit daraufhin, dass deutlich mehr Informationen die durch Salt and Pepper Rauschen gestört sind wiederhergestellt werden können. Dies wird auch von meiner subjektiven Bewertung gestützt. Beim Salt and Pepper Rauschen fallen dabei außerdem noch stärkere charakteristische Verhaltensweisen auf. Die Ergebnisse der Verfahren von Horn-Schunk, Lucas-Kanade und Liu zeigen Effekte, die gut mit den künstlerischen Filtern, wie z.B. Ölgemälde-Filtern, aus Bildbearbeitungsprogrammen vergleichbar sind. Horn-Schunk zeigt diese Effekte schon bei einem Skalierungsfaktor von zwei, während bei Lucas-Kanade und Liu in diesem Stadium nur ein sehr geringes Hintergrundmuster zu erahnen ist. Zusätzlich lassen die Verfahren von Liu und Lucas-Kanade dafür jedoch auch teils größere Elemente, wie beispielsweise die Krallen des Vogels oder das Standbein des Stativs verschwinden.

Der Primal-Dual Algorithmus von Chambolle und Pock verteilt die Störungen hinge-

gen in stark abgeschwächter Form über das gesamte Bild. Dadurch sind die wesentlichen Merkmale des Bildes wieder gut zu erkennen. Die Kralle und das Stativ gehen jedoch auch hier verloren und der Gesamteindruck bleibt stark verwaschen.

Alle von mir umgesetzten Verfahren beseitigen das Rauschen folglich nicht vollständig, sondern reduzieren lediglich die Menge und die Größe der einzelnen Störungen. Das Motiv wird dabei jedoch trotzdem wesentlich schärfer dargestellt. In Hinsicht auf den Wirkungsgrad kann man die Verfahren leicht in Gruppen einteilen. Das Verfahren mit anisotropem Tensor hat am stärksten mit der Störung zu kämpfen und beseitigt sie am wenigsten. Zwischen den Umsetzungen ohne Erweiterung, mit Belichtungsterm und mit L^1 -Norm sind kaum Unterschiede zu erkennen. Sie beseitigen die Störung vergleichbar gut. Am besten reduziert das Verfahren mit adaptiver Schrittweite die Störeinflüsse. Beim Skalierungsfaktor zwei werden sie dabei sogar vollständig entfernt.

Eine wertende Aussage fällt mir aufgrund der großen Unterschiede der Bilder sehr schwer. Sicher ist, dass ein so starkes Rauschen wie hier simuliert in den seltensten Fällen natürlich vorkommt und das Superresolution Problem somit passable Ergebnisse liefert. Die erste Wahl für die Beseitigung ist das Verfahren jedoch nicht. Davon abgesehen können dennoch einige Erkenntnisse gezogen werden. In der kleineren Skalierung vom Faktor zwei kann ich die schlechte Einstufung des bikubisch interpolierten Bildes nur unterstützen. In der größeren Skalierung hingegen, kann ich der schlechten Einschätzung meiner Verfahren, abgesehen vom Verfahren mit adaptiver Schrittweite, im Vergleich zur bikubischen Interpolation nicht bestätigen (durchschnittlicher UQI auf cameraman: 43,05% - UQI der Interpolation: 57,91%). Meiner Einschätzung nach, wird das Motiv in allen Fällen besser dargestellt und das Rauschen reduziert. Diese Fehlerwerte und meine Einschätzung weichen somit voneinander ab. Über alle Bilder hinweg, gefällt mir das Ergebnis meines Verfahrens mit adaptiver Schrittweite am besten. Von den Fehlerwerten wird dies allerdings nur auf den kleinen Skalierungen mit den besten Ergebnissen bestätigt (UQI bird: 99,17%, UQI cameraman: 97,17%). Auf den großen Skalierungen sind die Fehlerwerte von Liu und Lucas-Kanade am besten. Wegen des starken Ölgemälde-Effekts und den fehlenden Bildelementen, kann ich dieses Ergebnis jedoch nicht unterstützen. Durch die Fehlerwerte dieser Bilder werden die starken Einflüsse des Salt and Pepper Rauschens auf die Fehlermaße deutlich. Einen gemeinsamen zweiten Platz kann ich den beiden Verfahren in diesem Test trotzdem einräumen. Am schlechtesten gefällt mir das Verfahren von Horn-Schunck, da nahezu alle ungewünschten Effekte auftreten. Die restlichen Verfahren würde ich entsprechend zwischen Liu und Horn-Schunck einordnen.

Insgesamt haben die beiden Testreihen mit künstlichen Bildersets und verschiedenem Rauschen gezeigt, dass zufälliges, punktwises Rauschen auf das Superresolution Pro-

blem weniger Einfluss hat als gaußsches Rauschen. Dies deckt sich mit den Ergebnissen der Arbeit von Unger et al. (vgl. [UPWB10]). Des weiteren wurden einige Defizite der Fehlermaße UQI und MSE deutlich. Für die Bewertung des Superresolution Problems eignen sie sich nur in geringem Maße, da weder die sehr guten Ergebnisse also solche erkannt werden, noch die besonders schlechten. Interessant ist an den Ergebnissen also in erster Linie das charakteristische Verhalten der unterschiedlichen Verfahren. Es kann nun in den weiteren Testfällen untersucht werden, inwiefern die Effekte sich auch in anderen Situationen präsentieren. Im Bezug auf das Optical Flow Problem zeigt sich, dass ein geringer Groundtruth-Fehler wie bei Liu und Chambolle und Pock zu einer stärkeren Glättung der Konturen führt. Andererseits wirkt sich ein geringer Interpolation-Fehler auf die stärkere Beachtung der Details und damit auch auf die negative Beeinflussung durch eventuelle Störungen aus.

Aus der zweiten Gruppe wurde zuerst das Bild Mequon untersucht. Bei dem Bild Mequon müssen die Verfahren beweisen, wie sie mit Verformungen, Unstetigkeiten und Überdeckungen umgehen. Die Verformungen betreffen dabei das gelbe T-Shirt, welches den Hintergrund darstellt. Die Unstetigkeiten und Überdeckungen sind vor allem an den beiden Figuren im Vordergrund, teilweise aber auch in den Falten des T-Shirts zu finden. Überdeckt werden zum einen das halbe Gesicht des Musikers, und zum anderen die Schriften des T-Shirts im Hintergrund.

Mit den Überdeckungen haben ausnahmslos alle Verfahren Probleme. Die Auswirkungen sind dabei vollkommen unterschiedlich. Die Verfahren von Lucas-Kanade und Horn-Schunck reagieren mit Pixelfehlerwolken. Allerdings sind diese nicht allein auf die Unstetigkeiten beschränkt, sondern kommen auch an der nicht überdeckten Schrift vor. Die Ergebnisse dieser beiden Verfahren sind insgesamt am schlechtesten. Vermutlich reicht die Genauigkeit der Flüsse noch nicht aus. Auf die Überdeckung des Gesichts reagieren sowohl Lius Verfahren, als auch das von Chambolle und Pock, durch Aufblähen des Gesichtes. Dort wo die Figuren den Hintergrund verdecken, treten großflächig Verzerrungen und Pixelfehler auf. Die Figuren, mit Ausnahme des Kopfes des Musikers, und die Schrift werden hingegen sehr viel schärfer dargestellt. Die bikubische Interpolation wirkt im Vergleich stark geglättet. Die Verzerrungen, inklusive der dadurch entstehenden Überdeckungen, scheinen kein Problem zu bereiten. Auch mein Primal-Dual Algorithmus mit adaptiver Schrittweite neigt dazu, die überdeckten Flächen stark zu verschmieren. Pixelfehler treten dabei keine auf. Die Verfahren ohne Erweiterung, mit Belichtungsterm und mit L^1 -Norm verschmieren das Bild ebenfalls, aber in deutlich geringerem Maße. Lediglich die direkte Umgebung der Figuren und die tatsächlich überdeckten Schriften

weisen Fehler auf. Die Überdeckung am Kopf bewirkt bei ihnen eine Art Splittereffekt, der den überdeckten Teil verunstaltet. Bei der adaptiven Schrittweite werden die Figuren zudem leicht weichgezeichnet. Alle anderen von mir umgesetzten Verfahren stellen die Figuren ähnlich scharf dar wie die Algorithmen von Liu und Chambolle und Pock. Besonders gute Ergebnisse erzielt die Umsetzung mit anisotropem Tensor, der zwar den Splittereffekt vorweist, ansonsten aber keine Fehler erkennen lässt. Nur in einem Punkt sind die Verfahren von Liu und Chambolle und Pock etwas besser. In der unteren linken Ecke des Hintergrunds hat das T-Shirt noch ein wenig Struktur. Diese wird von den beiden Verfahren gut erhalten. Die Umsetzungen von mir glätten diesen Effekt teilweise raus. Die Umsetzung mit anisotropem Tensor zeigt den glättenden Effekt am geringsten. Der Belichtungsterm glättet schon etwas mehr und die anderen Verfahren innerhalb der einfarbigen Flächen sehr stark. Möglicherweise lässt sich der Effekt auf die Verformung des T-Shirts zurückführen.

Die Testergebnisse des Bildes Mequon zeigen also, dass Überdeckungen und Unstetigkeiten für alle Verfahren eine Herausforderung sind. Egal welche Umsetzung betrachtet wird, an den Stellen mit Überdeckungen und Unstetigkeiten entstehen starke Störungen. Die Verformungen scheinen hingegen kaum einen Einfluss zu haben, und wenn dann einen glättenden bei fast texturlosen Flächen.

Das zweite Bild aus Gruppe zwei ist das Bild army. Das Bild army ist das Bild Rubberwhale mit vielen zusätzlichen kleinen Spielzeugsoldaten. Die Herausforderungen dieses Bildes sind Unstetigkeiten, Überdeckungen, kleine Objekte, Verformungen und gemischte Pixel.

Die größten Probleme mit dem Bild hat das Verfahren von Horn-Schunk. Die kleinen Figuren werden zwar nicht zerstört, aber alle anderen Verfahren können die Figuren detaillierter darstellen. Einzig die bikubische Interpolation stellt die Figuren noch ungenauer dar. Außerdem verwischt das Bild dort, wo die Überdeckungen auftreten. Begleitet wird dieser Effekt von Bildfehlern, die den Gesamteindruck insgesamt bedeckt halten. Die Lucas-Kanade Methode funktioniert auf diesem Bild hingegen deutlich besser. Zwar treten an einigen Kanten Pixelfehler auf und es gibt kleine Probleme mit der Glättung der Konturen, die vielen kleinen Details werden jedoch gut erkannt und herausgestellt. Nur die Verfahren von Liu und Chambolle und Pock erreichen auf diesem Bild noch bessere Ergebnisse. Meine Verfahren haben leider alle das gleiche Problem, wobei es nicht überall gleich stark auftritt. Alle von mir umgesetzten Verfahren lassen den Bereich hinter den Soldaten rechts, die Pappe hinter der Muschel und den Soldaten hinter der Pappe verschwimmen. Wie im Bild zuvor tritt dieser Effekt bei der Umsetzung mit adaptiver Schrittweite am stärksten auf, und bei der Umsetzung mit Tensor am schwächsten. Die

drei anderen Umsetzungen liefern sehr ähnliche Ergebnisse. Auch das Rad, welches sich bei der Verschiebung leicht dreht, wird stark verwischt. Die übrigen Details werden ansonsten alle sehr gut herausgestellt. Die Verfahren von Liu und Chambolle und Pock zeigen ebenfalls leichte Schlieren an den Überdeckungen, aber eben nur sehr leichte. Die restlichen Herausforderungen meistern sie sehr gut.

Insgesamt scheint das Bild für lokale Flüsse besonders gut geeignet zu sein. Bis auf das Verfahren von Horn-Schunk machen alle Ergebnisse, teilweise mit kleinen Abstrichen, einen sehr guten Eindruck.

Das letzte Bild der zweiten Gruppe ist das Bild Evergreen. Auf dem Bild ist nichts außer einer Tanne mit einem sich bewegenden Ast zu sehen. Die Herausforderung bei diesem Bild liegt darin, neben den großen Bewegungen, Unstetigkeiten und Überdeckungen, die gemischten Pixel zu erkennen, um damit die Nadeln besser darstellen zu können.

Leider haben auf diesem Bild alle Algorithmen keine guten Ergebnisse erzielen können. Lucas-Kanade und Horn-Schunk versehen den sich bewegenden Ast mit einer Art Salt and Pepper Rauschen. Liu und Chambolle und Pock verursachen eine Weichzeichnung des Astes und der direkten Umgebung. Meine Algorithmen verschmieren den Ast ebenfalls, halten sich dabei aber etwas besser an die Grenzen. Das Verfahren mit anisotropem Tensor erzielt dabei mit einigen wenigen Pixelfehlern sogar ein halbwegs gutes Ergebnis. Das Resultat dieses Beispiels ist, dass die Verfahren mit gemischten Pixeln im Zusammenspiel mit großen Bewegungen ihre Probleme haben.

Die letzte Gruppe von Beispielbildern sind die Bilder cameramotion und fish aus der Datenbank von Liu. Zuerst wird das Beispiel fish genauer beschrieben. In dem gewählten Ausschnitt (Bild 143 bis 153) sind drei Fische und eine Schwanzflosse eines Fisches in einem Aquarium zu sehen. Das Bild ist wegen der schlechten Belichtung des Aquariums starkem Sensorrauschen ausgesetzt.

Die Verfahren von Lucas-Kanade und Horn-Schunk kommen mit diesen Anforderungen beide nicht zurecht. Weder das Rauschen wird eliminiert, noch die Fische oder die Korallen im Hintergrund besser dargestellt. Auch meine Umsetzung mit adaptiver Schrittweite verbessert das Ergebnis nicht wirklich. Zwar kann das Rauschen deutlich reduziert, und die Korallen besser erkannt werden, die Fische werden jedoch so stark geglättet, dass sogar größere Elemente wie das Auge des Fisches in der Bildmitte nicht mehr gut zu erkennen sind. Die Verfahren ohne Erweiterung, mit L^1 -Norm und mit Belichtungsterm haben dieses Problem nicht mehr. Das Rauschen wird gut entfernt, die Korallen genauer dargestellt und auch die Fische sind inklusive ihrer Details gut erkennbar. Bei genau-

rer Betrachtung fällt dennoch auf, dass neben einer leichten Glättung im Inneren, die Konturen der Fische weichgezeichnet werden. Der Ansatz mit Belichtungsterm erkennt dabei als einziges der drei Verfahren sogar das Muster des Fisches im Hintergrund. Die übrigen drei Verfahren scheinen ungefähr gleich gut zu funktionieren. Sie erkennen und erhöhen die Details, sowohl der Fische als auch der Korallen und reduzieren das Rauschen zuverlässig. Weichzeichnungen der Konturen der Fische sind dabei nicht festzustellen. Der Test zeigt also, dass vor allem die einfachen Verfahren mit schlechten Verhältnissen nicht umgehen können, weil dort der Fluss nicht exakt genug bestimmt wird. Die anderen Verfahren hingegen, fallen wie zuvor durch Überdeckungen und teilweise zu starke Glättung, negativ auf. Die drei besten Verfahren konnten mit guten Ergebnissen überzeugen.

Das zweite Beispiel aus Gruppe drei ist das Bild cameramotion aus der Datenbank zur Motion Annotation von Liu. Dieses Beispiel wurde als mögliches reales Anwendungsbeispiel getestet. Zu sehen ist eine Reihe von Fahrzeugen, die auf einer Straße durch eine Kurve fahren. Die Verfahren von Lucas-Kanade und Horn-Schunk erzielen auch in diesen beiden Bildern problematische Ergebnisse. Während das Verfahren von Horn-Schunk keine Erhöhung der Details erreichen kann und zusätzlich Fehler an den entscheidenden Stellen verursacht, treten bei Lucas-Kanade nur Bildfehler auf. Hochwertiger ist das Ergebnis deshalb trotzdem nicht. Auch die anderen Verfahren weisen ähnliche Probleme wie zuvor auf. Auch in diesem Beispiel gewinnt man den Eindruck, dass einige der von mir umgesetzten Verfahren bestimmte Bereiche (z.B. die Fahrbahnmarkierung) zu stark glätten. Die Verfahren von Liu, Chambolle und Pock und das von mir umgesetzte Primal-Dual Verfahren mit anisotropem Tensor erreichen auch bei diesem Bild gute Ergebnisse. Nicht nur die Störungen des vorausfahrenden Fahrzeugs, auch die Palme am linken Bildrand und das entgegenkommende Fahrzeug wirken etwas schärfer. Das Ziel, das Kennzeichen des entgegenkommenden Fahrzeugs lesen zu können, wurde jedoch nicht erreicht. Die drei besten Ergebnisse lassen die Schrift zumindest etwas leserlicher erscheinen.

Insgesamt schneiden die Verfahren von Horn-Schunk und Lucas-Kanade vor allem wegen ihrer fehlenden Genauigkeit nicht so gut ab, wie die anderen Verfahren. In Bezug auf die Ergebnisse aus dem Optical Flow Problem bestätigt sich, dass die Verfahren von Liu, Chambolle und Pock und mein Primal-Dual Algorithmus mit anisotropem Tensor auch hier die besten Ergebnisse erzielen. Die anderen von mir umgesetzten Erweiterungen scheinen gut zu funktionieren, übertragen jedoch ihre Ungenauigkeiten direkt auf das Superresolution Problem. Diese treten vor allem dann auf, wenn eine Unstetig-

keit im Fluss aufgrund von texturlosem Hintergrund nicht exakt umgesetzt wird. Die Groundtruth-Fehler des Optical Flow Problems scheinen dementsprechend besser für die Optimierung des Superresolution Problems geeignet zu sein. Sie würden diese Ungenauigkeit stark bestrafen, während die Interpolation-Fehler wegen des texturlosen, also gleichmäßig gefärbten, Bereichs niedriger sind. Außerdem haben alle Verfahren Probleme, mit Überdeckungen umzugehen. Es wäre sinnvoll das Optical Flow Problem oder das Superresolution Problem dahingehend zu überarbeiten. Ansonsten ist es möglich mit den Verfahren aus einer Reihe von niedrigauflösenden Bildern ein hochauflösendes Bild zu berechnen, das Details aus allen Bildern zusammenfasst. Einzelne Störungen sind für eine akkurate Lösung weniger problematisch als ein gleichmäßiger Informationsverlust wie beim gaußschen Rauschen. Stärkere lokale Einflüsse, z.B. durch den anisotropen Tensor reagieren allerdings stärker auf diese Art von Störungen.

5 Fazit

Die Arbeit zeigt insgesamt, dass die vorgestellten Minimierungsprobleme mit dem Primal-Dual Verfahren einen guten Ansatz darstellen, um aus einer Reihe von niedrigauflösenden Bildern ein hochauflösendes Bild zu errechnen. Während der Implementierungsphase wurde allerdings deutlich, dass eine gute Lösung des Optical Flow Problems ohne Coarse-to-Fine Pyramide nicht möglich ist. Nur wenn sie implementiert wird können mit dem beschriebenen Minimierungsansatz gute Lösungen erzielt werden.

Nichts desto trotz scheint das Modell jedoch an einigen Stellen an seine Grenzen zu stoßen. Unstetigkeiten bei der Registrierung stellen in einigen Situationen beispielsweise eine zu große Herausforderung dar. Bestimmte Erweiterungen des Modells können hier helfen bessere Ergebnisse zu erzielen. Durch die Erweiterung des Datenterms um einen Störfaktor mit Regularisierung, können z.B. Einflüsse durch Belichtungsunterschiede effektiv reduziert werden. In den vorliegenden Testbeispielen für das Superresolution Problem konnte hierbei zwar nur einmal eine kleine Verbesserung festgestellt werden, die Gründe hierfür liegen jedoch vermutlich in den konstanten und somit nahezu optimalen Belichtungsverhältnissen der Testbeispiele. Eine weitere sehr wirksame Verbesserung war der, an den Gradienten des Referenzbildes angepasste, Tensor zur Justierung der Regularisierung. Der Tensor hat in allen Testfällen bewiesen, dass er die Genauigkeit des Flusses an den Unstetigkeiten stark verbessert. Die Ergebnisse meines Verfahrens mit anisotropem Tensor erreichen problemlos die Qualität der Verfahren von Liu und von Chambolle und Pock. Die anderen Umsetzungen meines Verfahrens zeigen hingegen bei einigen Beispielen noch kleine Schwächen. Trotzdem erreichen sie beim Optical Flow Problem geringere Interpolation-Fehler als die Verfahren von Liu und von Chambolle und Pock. Beim Groundtruth-Fehler schneiden die beiden Verfahren im Vergleich jedoch besser ab. Das könnte ein Hinweis darauf sein, dass die Groundtruth-Fehler bessere Indikatoren für einen guten Fluss sind, als die Interpolation-Fehler. Es wäre sinnvoll dies weitergehend zu untersuchen.

Die Vergleiche zu einfacheren Verfahren, wie Lucas-Kanade und Horn-Schunk, haben gezeigt, dass die fehlende Genauigkeit, die bereits schon in der Evaluation des Optical Flow

Problems aufgefallen ist, sich auch auf das Superresolution Problem überträgt. Anstatt die Details im hochauflösenden Bild zu erhöhen, haben diese Verfahren zu Pixelwolken in diesen Bereichen geführt. Für ein gutes Superresolution Ergebnis reicht die Qualität der Verfahren somit nicht aus. Die Superresolution Fehlermaße haben in diesem Zusammenhang außerdem gezeigt, dass sie nicht besonders repräsentativ sind. In fast allen Tests zu Bildstörungen haben die Maße andere Wertungen getroffen als ich, obwohl teilweise deutliche Nachteile erkennbar waren.

Des Weiteren konnte in dieser Arbeit letztendlich ein enormer Störfaktor für alle Verfahren leider nicht beseitigt werden. Immer dann, wenn Überdeckungen in der Bildersequenz auftraten, konnte man auch Störungen in der Lösung erkennen. In der weiteren Forschung wäre es deshalb sehr sinnvoll, über mögliche Auswege für dieses Problem nachzudenken.

Ergänzend dazu, gibt es noch weitere Fragestellungen, die für die weitergehende Forschung noch tiefer gehend behandelt werden sollten:

1. Eignen sich die Groundtruth-Fehler als Optimierungsmaß des Optical Flow Problems in Bezug auf das Superresolution Problem?
2. Sind Groundtruth-Flüsse für die Lösung des Superresolution Problems optimal?
3. Gibt es ein gutes Fehlermaß für das Superresolution Problem?
4. Wie könnte das Optical Flow Problem und das Superresolution Problem erweitert werden, damit Überdeckungen weniger Störungen verursachen?

Anhang

Symbolverzeichnis

$\text{epi } f$	Epigraph von f
$B_r(x)$	Ball mit Radius r um x
$ \cdot $	Betrag in \mathbb{R} oder euklidische Länge in \mathbb{R}^n
∂C	Rand der Menge C
\overline{C}	Topologischer Abschluss der Menge C
C^C	Topologisches Komplement der Menge C
\emptyset	Leere Menge
$x \perp y$	x orthogonal zu y
$x \cdot y$	Skalarprodukt zwischen x und y
$\langle x, y \rangle$	Skalarprodukt zwischen x und y
f^*	Legendre-Fenchel-Transformierte von f
$U_{LF}(f)$	Definitionsbereich der LF-Transformierten von f
\overline{f}	Abschluss von f (größte unterhalbstetige Minorante)
$\partial_v^+ f(x), \partial_v f(x)$	(einseitige) Richtungsableitung von f an der Stelle x
χ_C^∞	Indikatorfunktion der Menge C mit den Werten 0 und $+\infty$
χ_C^*	Stützfunktion der Menge C
$\partial f(x)$	Subdifferential/Subgradient von f an der Stelle x
\hat{f}	Fortsetzung von f auf \mathbb{R}^n
$C(\Omega)$	Raum der Bilder = Raum der Funktionen von Ω nach \mathbb{R}

Literaturverzeichnis

- [BFB94] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *Int. J. Comput. Vision*, 12(1):43–77, February 1994.
- [BSL⁺11] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *Int. J. Comput. Vision*, 92(1):1–31, March 2011.
- [Cha04] Antonin Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, January 2004.
- [CP10] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. June 2010.
- [Dir15] H. Dirks. *Variational Methods for Joint Motion Estimation and Image Reconstruction*. PhD thesis, Institute for Computational and Applied Mathematics University of Muenster, June 2015.
- [FW05] David J. Fleet and Yair Weiss. Optical flow estimation, 2005.
- [GLY⁺13] T. Goldstein, M. Li, X. Yuan, E. Esser, and R. Baraniuk. Adaptive Primal-Dual Hybrid Gradient Methods for Saddle-Point Problems. *ArXiv e-prints*, May 2013.
- [HS80] Berthold K.P. Horn and Brian G. Schunk. Determining optical flow. April 1980.
- [LFAW08] Ce Liu, William T. Freeman, Edward H. Adelson, and Yair Weiss. Human-assisted motion annotation. *presented on CVPR 2008*, 2008.
- [Liu09] Ce Liu. Beyond pixels: Exploring new representations and applications for motion analysis. May 2009.
- [Mar77] Jürg T. Marti. *Konvexe Analysis*. Birkhäuser, Basel, 1977.

- [NBN12] Kyle Nelson, Asim Bhatti, and Saeid Nahavandi. Performance evaluation of multi-frame super-resolution algorithms. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, volume 2012, Fremantle, Australia, December 2012. Centre for Intelligent Systems Research, Deakin University, Geelong, Victoria, Australia.
- [PY01] S. Peled and Y. Yeshurun. Superresolution in mri: Application to human white matter fiber tract visualization by diffusion tensor imaging. 45:29–35, 2001.
- [Roc70] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.
- [UPWB10] Markus Unger, Thomas Pock, Manuel Werlberger, and Horst Bischof. A convex approach for variational super-resolution. In *Proceedings German Association for Pattern Recognition (DAGM)*, volume 6376 of *LNCS*, pages 313–322. Springer, September 2010.
- [WB02] Zhou Wang and A. C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, March 2002.
- [WTP⁺09] Manuel Werlberger, Werner Trobin, Thomas Pock, Andreas Wedel, Daniel Cremers, and Horst Bischof. Anisotropic huber-l1 optical flow. In *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK, September 2009. to appear.