Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

# Variationelle Verfeinerung zur Schätzung von Szenenfluss

Sandra García Bescós

| | |
|---|---|
| **Studiengang:** | Informatik |
| **Prüfer/in:** | Prof. Dr.-Ing. Andrés Bruhn |
| **Betreuer/in:** | Prof. Dr.-Ing. Andrés Bruhn,<br>Lukas Mehl, M.Sc. |
| **Beginn am:** | 24. Mai 2022 |
| **Beendet am:** | 8. Dezember 2022 |

## Kurzfassung

Die Schätzung der durch den Szenenfluss beschriebenen dreidimensionalen Bewegung aus Bildsequenzen ist eine der anspruchsvollsten Aufgaben im Bereich des Maschinensehens. Bei der Berechnung des Szenenflusses liefern Neuronale Netze aktuell die besten Ergebnisse. Eine weitere Verbesserung versprechen variationelle Ansätze zur Verfeinerung einer initialen Szenenflussschätzung. Dies motiviert das Ziel der vorliegenden Arbeit: Das Entwickeln variationeller Modelle zur Verfeinerung einer initialen Szenenflussschätzung. Dafür werden sich die enge Verwandtschaft von Szenenfluss und optischem Fluss sowie die Errungenschaften variationeller Verfahren zur Schätzung des optischen Flusses zu Nutze gemacht. Deshalb wird der Szenenfluss in dieser Arbeit als Kombination von optischer Fluss- und Stereo-Schätzung auf Basis eines Stereobildpaares in der Bildebene berechnet. Zur Formulierung des Verfeinerungsschrittes wird die von Brox *et al.* [9] eingeführte und auf den optischen Fluss bezogene differentielle Formulierung auf den Szenenfluss übertragen. Nach einer Einführung in das Thema und der Darstellung benötigter Konzepte wird ein verallgemeinertes, variationelles Modell zur Szenenflussverfeinerung entwickelt, diskretisiert und ein Iterationsschema zur Berechnung der Verfeinerung aufgestellt. Außerdem werden weiterführende Konstanz- und Glattheitsannahmen des optischen Flusses auf den Szenenfluss angewandt und differentiell formuliert. Daraus ergibt sich eine Vielzahl von Modellvarianten, von denen eine Vorauswahl, mit dem Ziel, die beste zu ermitteln, vergleichend evaluiert wird. Anhand einer aktuellen Benchmark wird überprüft, inwiefern der in dieser Arbeit entwickelte variationelle Verfeinerungsschritt eine Verbesserung der initialen Szenenflussschätzung erreicht. Dabei basiert die initiale Schätzung des Szenenflusses auf einem *state-of-the-art* Neuronalen Netzwerk. Die Experimente zeigen, dass sich mit keinem der evaluierten Modellvarianten eine Verbesserung der initialen Szenenflussschätzung erreichen lässt – wird der gesamte Datensatz der Evaluation betrachtet. Allerdings kann in einzelnen Bildsequenzen tatsächlich eine deutliche Verbesserung nachgewiesen werden, die das Potential der vorgestellten Methode erkennen lässt.

# Inhaltsverzeichnis

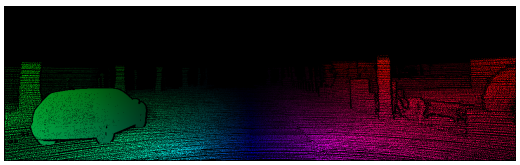
# 1 Einleitung

## 1.1 Motivation und Zielsetzung

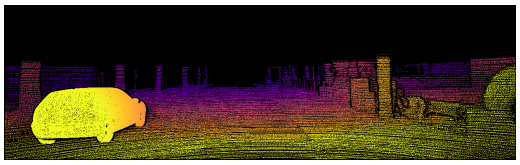
Der Szenenfluss hat als Bereich des Maschinensehens (*Computer Vision*) seit der ersten wissenschaftlichen Veröffentlichung über seine Schätzung im Jahre 1999 von Vedula *et al.* [84] stark an Bedeutung gewonnen. Schließlich stellt er die dreidimensionale Verschiebung von Oberflächenpunkten einer dreidimensionalen Szenerie dar und ist damit in der Lage die Bewegung von Objekten unserer Realität zu erfassen. Deshalb wird er in den verschiedensten Feldern, die im Folgenden skizziert werden, eingesetzt und an seiner Schätzung kontinuierlich geforscht.

Eines der Hauptfelder, in dem der Szenenfluss erfolgreich eingesetzt wird, ist der Straßenverkehr. Fahrerassistenzsysteme wie Kollisionsvermeidung, Spurhalteassistent und Kreuzungsassistent, benötigen eine zuverlässige 3D-Rekonstruktion der Bewegung anderer Straßenteilnehmer*innen, wie Autos, Fahrräder und Fußgänger*innen, oder anderer Arten von Hindernissen. Sich dem Ziel des autonomen Fahrens nähernd, wird intensiv geforscht: Beispielsweise entwickelten Lenz *et al.* [36], Menze *et al.* [51] und Franke *et al.* [18] neue Methoden, die den Szenenfluss zur Objekterkennung im Kontext des Straßenverkehrs verwenden. Geiger *et al.* veröffentlichten 2012 [20] und 2015 [52] einen neuen Datensatz, die sogenannte *KITTI Vision Benchmark Suite*, welche speziell Szenerien im Straßenverkehr beinhaltet, sodass aktuelle Methoden zur Stereo- und Szenenflussschätzung im Rahmen des Straßenverkehrs miteinander verglichen werden können. Eine Beispielaufnahme mit Werten zur Grundwahrheit (*ground truth*) aus diesem Datensatz ist in Abbildung 1.1 zu sehen.

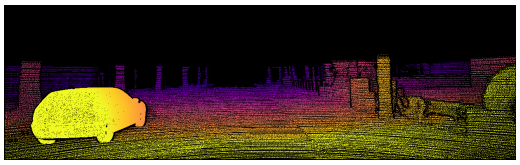
**(a)** Referenzframe zum Zeitpunkt $t$.



**(b)** Grundwahrheit des optischen Flussanteils der Szenenflussschätzung.



**(c)** Grundwahrheit des Startdisparitätanteils der Szenenflussschätzung.



**(d)** Grundwahrheit des Zieldisparitätanteils der Szenenflussschätzung.

**Abbildung 1.1:** Sequenz Nummer 173 mit Grundwahrheitswerten zum dazugehörigen Szenenfluss aus dem KITTI-Trainingsdatensatz [52]. (b) bis (d) in Farbwertkodierung.

Ein weiterer Bereich des Maschinensehens, der sich der Szenenflussschätzung bedient, befasst sich mit menschlicher Mimik und Gestik. Gesichtsausdrücke und Körperbewegungen haben eine große Bedeutung in unserem Alltag, da sie maßgeblich die zwischenmenschliche Kommunikation prägen. Mithilfe des Szenenflusses kann Mimik und Gestik auch zur Kommunikation zwischen Mensch und Computer beitragen, wie zum Beispiel Liu *et al.* [40] anhand von Kopfbewegungn oder Wang *et al.* [87] an Ganzkörperbewegungen gezeigt haben. Im Allgemeinen ist bereits die Erkennung von Menschen in Filmen und Videos ein schwieriges Problem, da sich sowohl die Personen, die Kamera und der Hintergrund bewegen können als auch die Körperhaltung, das Aussehen, die Kleidung, die Beleuchtung und die Hintergrundschärfe variieren [16]. Auch die Elastizität der Haut, welche sich bei Gesichtsausdrücken dehnt und krümmt, stellt eine Herausforderung dar, die Furukawa und Ponce [19] sowie Valgaerts *et al.* [82] adressierten und jeweils ein Modell für menschliche Gesichter aufstellten. Um dem zusätzlichen Problem der fehlenden Testdaten entgegenzuwirken, erforschten Laptev *et al.* [35] die Möglichkeit zur automatischen Erfassung von Trainingsdaten für menschliche Handlungen und zeigten, dass diese Daten zum Training eines Klassifikators für die Handlungserkennung verwendet werden können.

Auch in der Medizinwelt findet der Szenenfluss Verwendung, zum Beispiel bei der Realisierung von minimalinvasiven chirurgischen Eingriffen durch Roboterassistenz. In diesem Kontext befasst sich Stoyanov [72] mit der Elastizität von menschlichem Gewebe.

Die Erfolge, welche der Szenenfluss verzeichnet, sind mitunter auf seine Verwandtschaft mit dem optischen Fluss, welcher nach Vedula *et al.* [84] als die Projektion vom Szenenfluss auf die Bildebene gesehen werden kann, zurückzuführen. Bereits 1998 veröffentlichten Horn und Schunck [27] die erste wissenschaftliche Arbeit zur Berechnung des optischen Flusses mit Variationsansätzen, die sowohl für den optischen Fluss als auch für den Szenenfluss wegweisend war. Seither wird der Szenenfluss von vielen Verfahren variationell im zweidimensionalen Bildbereich berechnet [73], die eine ähnliche Formulierung zu der von Horn und Schunck aufweisen. Variationsansätze minimieren ein Energiefunktional, in welchem Abweichungen von Konstanz- und Glattheitsannahmen bestraft werden. Dieses Verfahren hat für die Berechnung des Szenenflusses den Vorteil, dass bereits erforschte Grundlagen des optischen Flusses angewandt werden können. Allerdings entfernen sich neuere Methoden in beiden Flussbereichen von dieser variationellen Formulierung und orientieren sich in Richtung des Machinellen Lernens, indem Neuronale Netze genutzt werden. Dosovitskiy *et al.* [17] entwickelten im Jahre 2015 das erste erfolgreiche faltende Neuronale Netzwerk (*Convolutional Neural Network*) FlowNet zur Berechnung des optischen Flusses. Aktuelle Beispiele für den Szenenfluss berechnende Neuronale Netze, welche mitunter die besten Ergebnisse im KITTI-Datensatz [52] zum Szenenfluss erzielen, sind die Netzwerke von Ma *et al.* [42] und Yang und Ramanan [95, 96] sowie RAFT-3D [77], ACOSF [37], M-Fuse [50] und CamLiFlow [39].

Um den Szenenfluss berechnen zu können, werden zwei aufeinanderfolgende Bildpaare aus einer Stereosequenz benötigt, sodass die Dreidimensionalität der Szenerie rekonstruiert werden kann. Alternativ kann auch ein Bildpaar mit zusätzlichen Tiefen- bzw. Disparitätsinformationen (RGB-D-Bilder) genutzt werden, wodurch der Stereoaufbau überflüssig wird. In einer binokularen Umgebung, welche für den Szenenfluss basierend auf Stereobildpaaren gegeben ist, kann dieser nach Wedel *et al.* [90] einfach durch die Kopplung von Stereo und optischem Fluss erfasst werden, wodurch die Disparitätsschätzung, die aus einem Stereobildpaar berechnet werden kann, zu einem wesentlichen Bestandteil der Schätzung des Szenenflusses wird. Manche Methoden berechnen die Disparität implizit, das heißt gleichzeitig zum restlichen Szenenfluss, beispielsweise die von Guizilini *et al.* [23] und Ma *et al.* [42], andere jedoch benötigen eine initiale Disparitätsschätzung zur Berechnung

des Szenenflusses, exemplarisch RAFT-3D [77] sowie die Verfahren von Herbst *et al.* [26] und Zanfir und Sminchisescu [98]. Diese initiale Disparitätsschätzung kann durch *Stereo Matching* Methoden berechnet werden, welche in variationeller Form aufstellbar sind, exemplarisch die von Slesareva *et al.* [68]. Allerdings haben sich Neuronale Netze in diesem Gebiet ebenfalls als geeignet herausgestellt. *State-of-the-art* Netzwerke zur Disparitätsschätzung sind in den Arbeiten von Cheng *et al.* [14], Li *et al.* [38], Zhang *et al.* [99] und Xu *et al.* [94], welche sich unter den aktuell Bestplazierten im KITTI-Datensatz [52] zur Stereoschätzung befinden.

Trotz häufiger Anwendung ist die Berechnung des Szenenflusses immer noch eine Herausforderung, die nach Menze und Geiger [52] bisher nicht vollständig zufriedenstellend gelöst werden konnte. Große Bewegungen, die sich über mehrere Pixel erstrecken und texturlose Oberflächen, die zum Beispiel spiegeln, bringen viele Methoden an ihre Grenzen, weshalb weiter geforscht wird und neue Ansätze entwickelt werden.

Die in dieser Arbeit vorgestellte Methode greift auf die Vorteile der Variationsrechnung zu, bedient sich jedoch zusätzlich einer initialen Schätzung der Disparität und des Szenenflusses, sodass eine Verfeinerung des Szenenflusses berechnet werden kann (*refinement*). Eine solche variationelle Verfeinerung wird bereits bei Maurer *et al.* [45] auf den optischen Fluss und in nicht variationeller Form bei Mehl *et al.* [50] sowie Park *et al.* [56] auf den Szenenfluss angewandt. Die Ausgaben von RAFT-3D [77] und GA-Net [99] werden hier als initiale Schätzung des Szenenflusses und der Disparität verwendet.

Ziel dieser Arbeit ist das Entwickeln mehrerer variationeller Modelle zur Verfeinerung des Szenenflusses basierend auf Brox *et al.* [9] sowie die Evaluierung der Verfeinerungsmodelle am KITTI-Datensatz [52].

## 1.2 Verwandte Arbeiten

Wie in Abschnitt 1.1 bereits ausgeführt, findet der Szenenfluss in vielen Bereichen Verwendung und ist von großer Bedeutung. Es überrascht daher nicht, dass sich zahlreiche Arbeiten schon seit 23 Jahren mit dem Szenenfluss beschäftigen und neue Ansätze finden, wie auch bestehende Verfahren erweitern. An dieser Stelle soll ein Überblick über die verwandten Arbeiten und den aktuellen Stand der Forschung präsentiert werden. Dabei können Methoden zur Berechnung vom Szenenfluss unter anderem anhand des Berechnungsschemas (Verfahren zur Energieminimierung, lernbasierte Verfahren und *Sparse-to-Dense*-Verfahren) oder des Eingabeformats (Monokularbilder, Stereobildpaar, RGB-D-Bilder und LiDAR-Punktwolke) kategorisiert werden, wobei beispielsweise Yang und Ramanan [96] und Badki *et al.* [3] Monokularbilder, Ma *et al.* [42], Li *et al.* [37] und Yang und Ramanan [95] Stereobildpaare, Herbst *et al.* [26], Quiroga *et al.* [59], Golyanik *et al.* [22] sowie Teed und Deng [77] RGB-D-Bilder und Liu *et al.* [39] die LiDAR-Punktwolke verwenden.

Dieser Abschnitt ist inhaltlich nach den Berechnungsschemata strukturiert und bietet insbesondere bei den variationellen Ansätzen, als Verfahren zur Energieminimierung, und für lernbasierten Methoden einen detaillierten Überblick verwandter Abhandlungen, da diese eine hohe Relevanz für die vorliegende Arbeit haben. Zur Vervollständigung des Überblicks wird anschließend noch auf Arbeiten zu *Sparse-to-Dense*-Verfahren und Methoden zur Verfeinerung eingegangen.

### 1.2.1 Verfahren zur Energieminimierung

Die Formulierung des Szenenflussproblems ist in Form eines Energiefunktionals möglich, das zur Lösungsfindung minimiert werden muss. Über diese Energie können auf verständliche Art und Weise allgemeine Annahmen getroffen werden, die über den gesamten Bildbereich gelten, weswegen Verfahren zur Energieminimierung in einer Vielzahl von Arbeiten erfolgreich eingesetzt wurden. An dieser Stelle soll eine exemplarische Auswahl vorgestellt werden, die einen Überblick über die wichtigsten Eckpunkte gibt.

Einige Verfahren zur Energieminimierung, wie beispielsweise von Vogel *et al.*[85, 86] und Zanfir und Sminchisescu [98], gehen von einer lokalen Starrheit aus, bei der die Pixel in einer kleinen Region dieselbe Bewegung haben. Die von Vogel *et al.* [86] vorgeschlagene Energie kombiniert einen okklusionssensitiven Datenterm mit geeigneter Form-, Bewegungs- und Segmentierungsregularisierung. Zanfir und Sminchisescu [98] schlugen eine grob-zu-fein, dichte und korrespondenzbasierte Szenenflussformulierung vor, die sich auf geometrische Annahmen stützt, um die Auswirkungen von großen Verschiebungen zu berücksichtigen und Okklusion zu modellieren. Das ansichtskonsistente Mehrbildverfahren von Vogel *et al.* [85] ist ebenfalls in der Lage, Okklusionen zu handhaben. Anstatt davon auszugehen, dass die Szene aus einer Reihe unabhängiger, sich starr bewegender Teile besteht, verwendeten Jaimez *et al.* [30] nicht-binäre Markierungen, um nicht-starre Verformungen an den Übergängen zwischen den starren Teilen der Szene zu erfassen und so die Schätzung zu verbessern. Die aktuell beste, auf Energieminimierung basierende Schätzungsmethode stellt im KITTI-Datensatz [52] das Verfahren von Sommer *et al.* [69] dar. Es arbeitet mit zwei aufeinanderfolgenden Stereo- oder RGB-D-Bildern und nutzt bestehende Algorithmen zur Schätzung des optischen Flusses und der Disparität. Es geht ebenfalls von einer Starrheit benachbarter Pixel aus und berechnet neben dem Szenenfluss zusätzlich Objektsegmentierung und visuelle Odometrie.

Zu den verbreitetsten Verfahren zur Energieminimierung gehören Variationsansätze. Sie kommen schon seit über 40 Jahren in zahlreichen Arbeiten im Bereich der Bewegungsschätzung zum Einsatz und sind daher von historischer Bedeutung. Variationsansätze zur Berechnung des Szenenflusses beruhen oftmals auf der optischen Flussmethode von Brox *et al.* [9], welche auf der grundlegenden Arbeit von Horn und Schunck [27] basiert, die als erste Variationsansätze zur Berechnung des optischen Flusses nutzten. Die folgenden Arbeiten lassen sich zu den variationellen Verfahren zur Szenenflussschätzung zuordnen.

Die erste erfolgreiche Berechnung des Szenenflusses mit Variationsansätzen stammt aus dem Jahre 2007 von Huguet und Devernay [28]. Ihre Methode setzt eine kalibrierte Kameraeinstellung voraus und stützt sich auf die Bedingungen der epipolaren Geometrie. Allerdings hat ihre Herangehensweise einen großen Nachteil: Sie ist langsam in der Berechnung. Wedel *et al.* [90] beschleunigten diesen Prozess, indem Stereo- und Bewegungsschätzung getrennt berechnet werden. Diese Spaltung erlaubt die Berechnung in Echtzeit und zusätzlich die Kalkulation auf Basis lückenhafter wie auch dichter Stereodaten. Ebenfalls eine schnellere Berechnung erzielten Wedel *et al.* [89], indem sie eine sehr effiziente Methode entwickelten, welche die Disparitätskarte auf einem FPGA und den Szenenfluss auf einem Grafikprozessor berechnet. Im Gegensatz zu den bisher aufgeführten variationellen Verfahren integrierten Valgaerts *et al.* [81] die Schätzung der Kameraparameter, wodurch ihr Verfahren auf nicht-kalibrierte Systeme angewandt werden kann.

Mit der Einführung der RGB-D-Kamera eröffnete sich die Möglichkeit Bilder aufzunehmen, die neben der Farb- auch Disparitätsinformation enthalten. Dies machten sich Herbst *et al.* [26] zu Nutze und zeigten, dass Szenenfluss aus einem Bildpaar mit RGB-D-Daten geschätzt werden kann, womit

der Stereoaufbau überflüssig und die Berechnung des Szenenflusses mit nur einer Kamera möglich wird. Ihre Methode zeichnet sich jedoch durch eine fehlende Handhabung von Okklusion aus. Das Verfahren von Basha *et al.* [4] nutzt ein kalibriertes System mehrerer Kameras und zum ersten Mal eine 3D-Punktwolken-Parametrisierung des Szenenflusses. Auch hier stellt besonders die Okklusion ein Problem dar, da es mit zunehmender Anzahl an Kameras zu mehr Verdeckungen kommt. Nüssle [55] erweiterte die Idee von Basha *et al.* um die Gradientenkonstanzannahme, welche Ergebnisse bei Helligkeitsunterschieden zwischen den Bildern verbessern kann.

Wie bisher geschildert, werden Variationsansätze häufig in Verbindung mit Szenenfluss verwendet. Allerdings stoßen sie bei großen Verschiebungen oder lokalen Mehrdeutigkeiten, beispielsweise bei texturlosen oder reflektierenden Oberflächen, an ihre Grenzen. Ein alternatives Berechnungsschema, welches diese Probleme überwinden kann, stellen lernbasierte Methoden dar, auf die im Folgenden näher eingegangen wird.

### 1.2.2 Lernbasierte Methoden

Neuronale Netze gehören zu lernbasierten Methoden und berechnen aktuell im Kontext der Szenenflussschätzung im Straßenverkehr bessere Ergebnisse als alternative Berechnungsarten. Das zeigt sich dadurch, dass sie aktuell die Spitzenreiter im KITTI-Datensatz [52] sind. Es dauerte jedoch vier Jahre seit der ersten Szenenflussschätzung durch ein Neuronales Netz von Mayer *et al.* [47], bis sie sich tatsächlich gegenüber den anderen Verfahren durchsetzen konnten. Dies ist darauf zurückzuführen, dass das Szenenflussproblem mehr Freiheitsgrade hat als Probleme, die sich nur im Ein- oder Zweidimensionalen bewegen.

Dass Neuronale Netze die zuvor beschriebenen Limitationen von Variationsansätzen überwinden können, zeigten Behl *et al.* [5], deren Methode ein faltendes Neuronales Netz (*Convolutional Neural Network*, CNN) zur Objekterkennung im Straßenverkehr nutzt. Auch Ren *et al.* [61] und Yang und Ramanan [95] segmentierten Objekte und gingen davon aus, dass Szenen aus Vordergrundobjekten bestehen, die sich starr vor einem statischen Hintergrund bewegen. Ma *et al.* [42], deren Modell ebenfalls starre Objekte annimmt, formulierten das Problem als Energieminimierung in einem *Deep Learning* Modell, das auf dem Grafikprozessor effizient gelöst werden kann und so alle bis zum Jahre 2019 veröffentlichten Methoden in Bezug auf Rechenzeit um einen Faktor von 800 schlägt.

Ein ebenfalls zeitlich effizientes Netz wurde von Saxena *et al.* [64] vorgeschlagen. Es wird unter einer neuartigen selbstüberwachten Strategie zur Vorhersage von dichten Okklusionskarten aus Bildern trainiert, wodurch die Schätzung des Szenenflusses verbessert wird. Jiang *et al.* [31] führten das kompakte Netz SENSE für die ganzheitliche Schätzung von Szenenflüssen (optischer Fluss, Disparität aus Stereo, Okklusion und semantische Segmentierung) ein. Die gemeinsame Nutzung von Merkmalen macht das Netz kompakt und führt zu einer besseren Darstellung der Merkmale. Eine zusätzliche Erweiterung bietet das Modell von Badki *et al.* [3], welches den Szenenfluss um den binären *Time-to-contact* (TTC) erweitert und mit geringer Latenz voraussagt, ob der Beobachter innerhalb einer bestimmten Zeit mit einem Hindernis kollidieren wird, was oft wichtiger ist, als die genaue Kenntnis des TTC pro Pixel.

Im Gegensatz zu den meisten bisherigen Methoden stützt das Verfahren von Schuster *et al.* [65] sich nicht auf ein konstantes Bewegungsmodell, sondern lernt eine generische zeitliche Beziehung der Bewegung aus den Daten und nutzt mehrere aufeinanderfolgende Bilder einer Bildsequenz. Der Ansatz von Yang und Ramanan [96] unterscheidet sich ebenfalls grundlegend von den obigen – durch

die Untersuchung der optischen Ausdehnung. Beim Integrieren der intrinsischen Kameraparameter kann die optische Ausdehnung in normalisierte 3D-Szenenflussvektoren umgewandelt werden, die aussagekräftige Richtungen der 3D-Bewegung, aber nicht deren Größe, liefern. Der normalisierte Szenenfluss kann zu einem echten 3D-Szenenfluss mit Kenntnis der Disparität in einem Bild ‚aufgewertet' werden.

Im Rahmen lernbasierter Methoden besteht ein Mangel an realistischen Weltszenarien mit *ground truth* Werten, die zum Trainieren vieler Netze notwendig sind. Li *et al.* [37] bewältigten dieses Problem mit einer zweistufigen adaptiven Methode zur Schätzung des Objektszenenflusses unter Verwendung eines hybriden CNN-CRF-Modells (*Convolutional Neural Network, Conditional Random Fields* Modell). CNNs werden hier eingesetzt, um Disparität und optischen Fluss mit CRF-Nachbearbeitung zu erhalten.

Die schon genannten Netze von Behl *et al.* [5] und Ren *et al.* [61] haben gegenüber RAFT-3D [77] den Nachteil, dass sie das Trainieren von Instanzsegmentierungen erfordern, sodass die Bewegung neuer unbekannter Objekte nicht erkannt werden kann. Eine wichtige Neuerung von RAFT-3D [77], das sich auf der Stereoschätzung von GA-Net [99] stützt, ist die Einbettung starrer Bewegungen, die jedoch eine weiche Gruppierung von Pixeln in starre Objekte darstellt. Durch diese weiche Gruppierung können auch unbekannte Objekte erkannt werden. Eine Einschränkung von RAFT-3D allerdings ist, dass das vom Netz geschätzte SE3-Bewegungsfeld auf ein Achtel der Auflösung begrenzt ist, weshalb eine nachträgliche Verfeinerung die Ergebnisse potentiell verbessern kann. Trotz ausgefeilter Starrheitsannahmen und Parametrisierungen sind neuronale Netze wie RAFT-3D in der Regel auf nur zwei Bildpaare beschränkt, was es ihnen nicht erlaubt, zeitliche Informationen zu nutzen. Das Modell M-FUSE von Mehl *et al.* [50] geht diesen Mangel mit einem neuartigen Multi-Frame-Ansatz an, der ein zusätzliches vorangehendes Stereopaar berücksichtigt. Es konnte eine Verfeinerung der Ergebnisse von RAFT-3D um mehr als 16 % erreichen, indem zunächst der vorangehende Stereoschätzer GA-Net [99] mit dem neueren LEAStereo [14] ersetzt wurde und die U-Netz-ähnliche Architektur eine Fusion von Vorwärts- und Rückwärtsflussschätzungen durchführt, sodass die Integration zeitlicher Informationen über die Nachfrage ermöglicht wird.

Die besten Ergebnisse im KITTI-Datensatz [52] erzielt aktuell das Netz CamLiFlow [39]. Auf Basis eines synchronisierten Kamera- und LiDAR-Bilderpaars schätzt es gemeinsam den dichten optischen Fluss für Kamerabilder und den lückenhaften Szenenfluss für LiDAR-Bilder. CamLiFlow besteht aus zwei symmetrischen Zweigen, dem Bildzweig (für 2D-Daten) und dem Punktzweig (für 3D-Daten), mit mehreren bidirektionalen Verbindungen zwischen ihnen. Es benötigt im Gegensatz zu M-FUSE [50], welches zweitplatziert ist, keine initialen Schätzungen, enthält insgesamt weniger Parameter als das Netz von Yang und Ramanan [95], welches drittplatziert ist und arbeitet schneller als RAFT-3D [77], welches unter den Ergebnissen veröffentlichter Paper zu lernbasierten Methoden viertplatziert ist.

### 1.2.3 *Sparse-to-Dense*-Verfahren

Einen grundlegend anderen Ansatz verfolgen *Sparse-to-Dense*-Verfahren. Sie beschleunigen die Berechnung energiebasierter Methoden, indem zunächst einzelne Korrespondenzen gefunden und anschließend die Zwischenräume mithilfe von robuster Interpolation geschlossen werden. Diesen Ansatz verfolgt die von Čech *et al.* [12] vorgeschlagene Methode, die von Korrespondenzen ausgeht und diese auf ihre Nachbarschaft überträgt. Sie ist für komplexe Szenen mit großen Bewegungen

genau und liefert zeitlich kohärente Disparitäts- und optische Flussergebnisse. *Sparse-to-Dense*-Verfahren können auch in gemischter Form mit Variationsansätzen auftauchen, wie Stoll *et al.* [70] zeigten. Sie verwendeten diese Kombination, um das Defizit der Variationsansätze in Bezug auf große Pixelsprünge auszugleichen. Dass der Szenenfluss auch ohne Variationsansätze oder Starrheitsannahmen mithilfe von *Sparse-to-Dense*-Methoden berechnet werden kann, bewiesen Schuster *et al.* [66]. Ihr Verfahren findet ohne vorherige Regularisierung einzelne Übereinstimmungen zwischen zwei Stereobildpaaren und führt eine dichte Interpolation durch, bei der geometrische Kanten und Bewegungsgrenzen durch die Verwendung von Kanteninformationen erhalten bleiben. Später erweiterten Schuster *et al.* [67] das Modell auf Bilder mehrerer Zeitpunkte und optimierten die Laufzeit.

### 1.2.4 Methoden zur Verfeinerung

Im Bereich des Szenenflusses, wie auch beim optischen Fluss, findet die Idee der Verfeinerung immer stärkeren Anklang, da zuvor berechnete Ergebnisse durch einen zusätzlichen Schritt verbessert werden können. Dabei kann der Verfeinerungsschritt prinzipiell mit den Berechnungsschemata, die in den Abschnitten 1.2.1 bis 1.2.3 genannt werden, kombiniert werden. Die Verfeinerung selbst kann ebenfalls in einem der drei Berechnungsschemata durchgeführt werden.

Verfeinerungsschritte können, wie beim Verfahren von Liu *et al.* [41], das eine lernfähige neue Schicht zur Verfeinerung des Szenenflusses enthält, in einem Modell integriert und somit Teil dessen sein. So auch bei Schuster *et al.* [65], die sich eines Neuronalen Netzes als Verfeinerungsschritt bedienten, welches bidirektionale Schätzungen des Szenenflusses aus einem gemeinsamen Referenzrahmen kombiniert, was zu einer verfeinerten Schätzung und einem natürlichen Nebenprodukt von Verdeckungsmasken führt. Auf diese Weise bietet dieser Ansatz eine schnelle *Multi-Frame*-Erweiterung für eine Vielzahl von Szenenflussschätzern. Verfeinerungsschritte können aber auch unabhängige Methoden sein, die sich auf andere Modelle stützen und ihre Ergebnisse verbessern, wie das Verfahren von Mehl *et al.* [50], welches sich ebenfalls weitere Stereobildpaare zu Nutze macht und Vorwärts- sowie Rückwärtsflussschätzungen, basierend auf dem RAFT-3D Modell [77], kombiniert, womit die Integration zeitlicher Informationen möglich ist. Eine andere Art der Verfeinerung, der *Tensor-Voting*-Ansatz für die Schätzung und Verfeinerung von 3D-Szenenflüssen, wurde von Park *et al.* [56] vorgeschlagen. Ein einzigartiger, zweistufiger Verfeinerungsprozess reguliert nacheinander die Richtung und Größe des Szenenflusses. Die Richtung des Szenenflusses wird durch die Nutzung der 3D-Nachbarschaftsglätte verfeinert, die durch Tensorabstimmung definiert wird. Die Größe des Szenenflusses wird durch die Verbindung der impliziten *Patches* über die aufeinanderfolgenden 3D-Punktwolken verfeinert.

Variationsansätze werden heutzutage in Bezug auf Geschwindigkeit und Genauigkeit von anderen Ansätzen übertroffen und werden daher nur noch als Verfeinerungsschritt verwendet [67]. Das Verfahren von Richardt *et al.* [62] verfeinert die berechneten Korrespondenzfelder für den Szenenfluss in einer Variationsformel und zeigt dichte Szenenflussergebnisse, die aus anspruchsvollen Datensätzen mit sich unabhängig voneinander bewegenden, handgeführten Kameras mit unterschiedlichen Kameraeinstellungen berechnet wurden. Laut Schuster *et al.* [66] und Wang *et al.* [88], die ebenfalls einen abschließenden variationellen Verfeinerungsschritt nutzten, soll die variationelle Verfeinerung nur bis zu dreißig Iterationen beinhalten, da sonst die Gefahr besteht, dass eine zu starke Glättung eingeführt wird und so die Ergebnisse verschlechtert statt verbessert werden.

Auf diesen Untersuchungen basiert die Motivation für die vorliegende Arbeit: das Entwickeln variationeller Modelle zur Verfeinerung einer initialen Szenenflussschätzung. So sollen die Nachteile von Neuronalen Netzen, wie beispielsweise die reduzierte Auflösung von RAFT-3D [77], mithilfe der Glattheitsterme korrigiert werden, sodass eine genauere Schätzung möglich wird.

## 1.3 Gliederung und Vorgehensweise

Die vorliegende Arbeit wird in acht Kapitel untergliedert. Nach der Einleitung, zu welcher dieser Abschnitt zu zählen ist, werden in Kapitel 2 die Grundlagen, die für das Verständnis der zu entwickelnden Verfeinerungsmethode benötigt werden, erklärt. Hierfür werden die verwendeten Notationen aufgelistet, Termini definiert und wichtige Konzepte eingeführt. Das anschließende Kapitel 3 soll durch den Prozess der Berechnung der Szenenflussschätzung mithilfe von Variationsansätzen führen. Dabei werden die Annahmen als Energiefunktional modelliert und letztgenanntes mittels der Euler-Lagrange-Gleichungen minimiert. Diese werden diskretisiert und durch ein iteratives Lösungsverfahren gelöst. Wie das erarbeitete variationelle Szenenflussverfahren in differentieller Formulierung als Verfeinerung eingesetzt werden kann, wird daraufhin in Kapitel 4 gezeigt. Das so entstandene, grundlegende Modell zur Verfeinerung des Szenenflusses wird in Kapitel 5 um weiterführende Datenterme und in Kapitel 6 um weiterführende Glattheitsterme ergänzt. Die vorgestellten Datenterm- und Glattheitstermvarianten werden in Kapitel 7 vergleichend evaluiert, wodurch ein optimales Gesamtmodell gefunden werden soll, welches selbst ebenfalls evaluiert wird. Abschließend werden in Kapitel 8 die wichtigsten Punkte der Arbeit zusammengefasst und ein Ausblick auf daran anknüpfende Analysen, Experimente sowie Verbesserungsmöglichkeiten der Modellierung gegeben.

# 2 Grundlagen

Im Rahmen der vorliegenden Arbeit soll die Schätzung des Szenenflusses durch variationelle Verfeinerung verbessert werden. Bevor das konkrete Vorgehen zur Erreichung des genannten Ziels im Detail geschildert wird, werden zunächst in Abschnitt 2.1 die relevanten Notationen, die in dieser Arbeit Verwendung finden, aufgelistet. Anschließend werden in Abschnitt 2.2 wichtige Definitionen und Konzepte eingeführt. Hierzu zählen vorrangig die Grundlagen des Szenenflusses mit benötigten Eingabegrößen, dem Stereo-Kamera-Setup, Grundlagen zu Disparität und optischem Fluss sowie die Parametrisierung des Szenenflusses selbst.

## 2.1 Notation

Folgende Notationen werden in dieser Arbeit verwendet:

- Vektoren $\mathbf{v}$ werden mit fettgedruckten Kleinbuchstaben bezeichnet.
- Tensoren $\mathbf{T}$ werden mit fettgedruckten Großbuchstaben dargestellt.
- Die Inverse einer Matrix $A$ wird wie folgt notiert: $A^{-1}$.
- Die Spur einer Matrix $A$ wird als Summe der Einträge ihrer Diagonalen definiert und als $tr(A)$ bezeichnet.
- Die einzelnen Tensoreinträge werden mit tiefgestellten Zahlen indiziert, zum Beispiel $\mathbf{T}_{12}$.
- Die Diskretisierung einer Größe $f$ wird anhand eckiger Klammern mit tiefgestellten Indizes gekennzeichnet, wie $[f]_{ij}$.
- Der transponiert Operator wird durch ein hochgestelltes $\mathbf{v}^\top$ gekennzeichnet.
- Der zum Vektor $\mathbf{v}$ orthogonale Vektor wird als $\mathbf{v}^\perp$ notiert.
- Die erste Ableitung einer eindimensionalen Funktion $\Psi$ wird als $\Psi'$ dargestellt.
- Partielle Ableitungen erster Ordnung mehrdimensionaler Funktionen werden durch einen tiefgestellten Index abgekürzt, wie beispielsweise $f_x = \frac{\partial f}{\partial x}$.
- Analog dazu werden partielle Ableitungen zweiter Ordnung durch zwei tiefgestellte Indizes abgekürzt, wie beispielsweise $f_{xx} = \frac{\partial^2 f}{\partial x^2}$.
- Der Gradientenoperator $\nabla$, der als Vektor aller partieller Ableitungen definiert ist, fasst diese Ableitungen zusammen. In der vorliegenden Arbeit wird er konkret als zweidimensionaler räumlicher Gradient $\nabla f = (f_x, f_y)^\top$ mit den partiellen $x$- und $y$-Ableitungen definiert, da die Berechnungen zum Großteil auf einer zweidimensionalen Domäne durchgeführt werden.
- Der Laplace-Operator $\Delta$ wird im zweidimensionalen Raum als $\Delta f = f_{xx} + f_{yy}$ festgelegt.

- Die Divergenz eines Vektorfeldes $(u(x,y),v(x,y))^\top$ ist definiert als $\mathrm{div}((u(x,y),v(x,y))^\top) = u_x(x,y) + v_y(x,y)$.
- Die Faltung zweier Funktionen $f$ und $g$ ist definiert als $(f * g) = \int_{\mathbb{R}^n} f(\tau) \cdot g(t-\tau)\,\mathrm{d}\tau$.
- Die boolsche Operation $\mathbf{1}$ wird als $\mathbf{1}_{(X)} = \begin{cases} 1 & \text{wenn } X \\ 0 & \text{sonst} \end{cases}$ mit Bedingung $X$ definiert.

## 2.2 Szenenfluss

Der Szenenfluss beschreibt die dreidimensionale Bewegung von Oberflächenpunkten einer dreidimensionalen Szenerie zwischen Zeitpunkt $t$ und Zeitpunkt $t+1$. Dies wird in Abbildung 2.1 dargestellt.

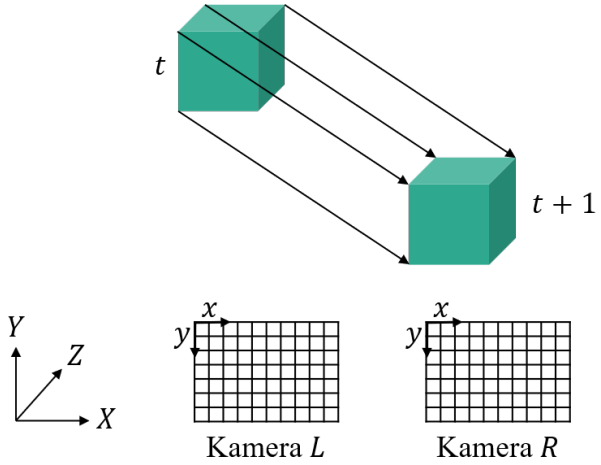



**Abbildung 2.1:** Szenenfluss beispielhaft an einem sich bewegenden Quader in einem Stereo-Kamera-Setup visualisiert (gleichzeitige Darstellung von den Zeitpunkten $t$ und $t+1$). Die jeweiligen Bildebenen der Kameras, welche hier durch Raster dargestellt werden, befinden sich ebenfalls im dreidimensionalen Raum (hier beispielsweise auf der $X$-$Y$-Ebene der Weltkoordinaten). Die Pfeile am Quader stehen für die dreidimensionalen Bewegungsvektoren, die hier nur exemplarisch an den Ecken gezeigt werden, jedoch für jeden Oberflächenpunkt des Quaders existieren.

Um die, im Vergleich zum optischen Fluss (siehe Abschnitt 2.2.4), höhere Dimensionalität einzufangen, genügt es nicht die Szenerie mit einer RGB-Kamera zu zwei Zeitpunkten aufzunehmen. Es wird entweder eine zweite Kamera benötigt, mithilfe derer die Tiefe über die Disparität berechnet werden kann (mehr dazu in den Abschnitten 2.2.2 und 2.2.3), oder die Kamera muss in der Lage sein zusätzlich die Tiefe zu messen, wie es beispielsweise durch die LiDAR-Technologie ermöglicht wird [39]. Diese beiden Optionen ermöglichen das Erzeugen der jeweils benötigten Eingabegrößen. Das konkrete Format dieser Eingaben wird in Abschnitt 2.2.1 erläutert.

Neben den Eingabegrößen muss auch der Szenenfluss selbst im Rahmen einer dreidimensionalen Szenerie beschrieben werden. Nach Wedel *et al.* [90] kann er als Kombination von Disparität und optischem Fluss gesehen werden, weswegen die Szenenflussschätzung in vielen Arbeiten in die Schätzung von Disparität und optischem Fluss aufgeteilt wird, wie beispielsweise bei Liu *et al.* [39], Sommer *et al.* [69], Teed und Deng [77], Badki *et al.* [3] und Yang und Ramanan [95, 96]. Diese zwei Konzepte werden in den Abschnitten 2.2.3 und 2.2.4 vorgestellt, aus denen in Abschnitt 2.2.5 die Parametrisierung des Szenenflusses abgeleitet wird.

Insgesamt führt dieses Kapitel die benötigten Eingabegrößen, das Kamera-Setup zur Eingabeerzeugung, Grundlagen der Disparität sowie des optischen Flusses und die Parametrisierung des Szenenflusses ein. Das konkrete Verfahren zur Szenenflussschätzung wird in Kapitel 3 aufgezeigt und basiert auf den hier präsentierten Grundlagen.

### 2.2.1 Eingabegrößen

Verfahren, die den Szenenfluss schätzen, benötigen Informationen über die Szenerie, genauer über die dreidimensionale Position der Oberflächenpunkte, deren Bewegung rekonstruiert werden soll. Die Position kann dabei entweder als dreidimensionale Punktwolke, wie mithilfe der LiDAR-Technologie [39] erfasst, oder als zweidimensionale Pixelposition mit zusätzlicher Tiefeninformation als Disparität beschrieben werden. Das häufigste Vorgehen bei der Datenakquisition für die Schätzung des Szenenflusses nutzt die letztgenannte Beschreibung der Szenerie und beschränkt sich auf die Aufnahme eines Stereobildpaares mithilfe von zwei RGB-Kameras, die zunächst keine Informationen über die Tiefe ermitteln (wie in Abschnitt 2.2.2 geschildert). Erst im nächsten Schritt wird die Tiefe als Disparität, beispielsweise durch einen Stereoschätzer wie GA-Net [99], aus dem Bildpaar berechnet. Manche Szenenflussverfahren integrieren die Schätzung der Disparität und benötigen nur das Bildpaar zu jedem Zeitpunkt als Eingabe. Dazu gehören die Arbeiten von Wedel *et al.* [89], Huguet und Devernay [28], Behl *et al.* [5] und Li *et al.* [37]. Bessere Ergebnisse liefern allerdings Methoden, die den Output von Stereoschätzern nutzen, wie die von Yang und Ramanan [95, 96], Ma *et al.* [42], Liu *et al.* [39], Sommer *et al.* [69], Teed und Deng [77], Badki *et al.* [3], die auf Grundlage der Bildfolgen mit zusätzlicher Disparitätsinformation arbeiten – dies ist auch in der vorliegenden Arbeit der Fall.

Die Eingabegrößen sind aufgrund dessen wie folgt definiert: Sei $f(x, y, t)$ eine Bildfolge aus Grauwertbildern und $\zeta(x, y, t)$ die jeweilige Disparitätskarte, bei denen $(x, y)^\top \in \Omega$ die räumliche Lage auf dem Bildbereich $\Omega \subset \mathbb{R}^2$ und $t \in \mathbb{R}_0^+$ den Zeitpunkt bezeichnen. Die Bildfolge $f$ bezieht sich dabei ohne Beschränkung der Allgemeinheit auf die linke Kamera des Kamera-Setups (siehe Abschnitt 2.2.2), da das linke Bild zum Zeitpunkt $t$ als Referenz für die gesamte Parametrisierung gilt und daher als Referenzframe bezeichnet wird. Die Disparitätskarte $\zeta(x, y, t)$ ist auf das linke Bild registriert und sagt für jeden Bildpunkt $(x, y)$ der linken Kamera aus, welche Disparität sich zum korrespondierenden Bildpunkt der rechten Kamera zu einem Zeitpunkt messen lässt (siehe Abschnitt 2.2.3). Mithilfe dieser Information kann die rechte Bildfolge im Rahmen dieser Arbeit ausgeblendet werden, da die linke Bildfolge und die jeweilige Disparitätskarte ausreichend Informationen über die dreidimensionale Szenerie bereitstellen. Anzumerken ist an dieser Stelle, dass $f$ und $\zeta$ aus den vorgeglätteten Ursprungsbildern $f_0$ und $\zeta_0$ entstehen,

$$f(x, y, t) = K_\sigma * f_0(x, y, t), \quad \zeta(x, y, t) = K_\sigma * \zeta_0(x, y, t), \tag{2.1}$$

indem die Ursprungsbilder mit einer zweidimensionalen Gaußkurve $K_\sigma$ mit Mittelwert $\mu = 0$ und Standardabweichung $\sigma$,

$$K_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp(\frac{x^2 + y^2}{2\sigma^2}), \tag{2.2}$$

gefaltet werden (*Gaussian Presmoothing*). Somit wird der Einfluss von Rauschen und Ausreißern reduziert, jedoch der Mittelwert der Bilder bewahrt. Außerdem sind $f$ und $\zeta$ dadurch unendlich oft differenzierbar, d.h. $f \in \mathcal{C}^\infty$ und $\zeta \in \mathcal{C}^\infty$ [101]. Dies ist für die spätere Diskretisierung (siehe Abschnitt 3.3) wichtig. Des Weiteren soll die Bildfolge $f_0$ aus Grauwertbildern bestehen, weshalb ein zusätzlicher Vorverarbeitungsschritt – eine Umwandlung von RGB-Bildern zu Grauwertbildern – stattfindet. Die Ursprungsbilder werden mittels der folgenden Berechnung umgewandelt:

$$f_0(x, y, t) = 0.2125 \cdot R(x, y, t) + 0.7154 \cdot G(x, y, t) + 0.0721 \cdot B(x, y, t), \tag{2.3}$$

wobei $R$ den Rot-, $G$ den Grün- und $B$ den Blauwertkanal des RGB-Bildes zum Zeitpunkt $t$ beschreiben.

Wie diese Bilder und die dazugehörigen Disparitätskarten mithilfe eines Stereoaufbaus ermittelt werden, wird im nachstehenden Abschnitt geschildert.

### 2.2.2 Stereo-Kamera-Setup

Zur Berechnung des Szenenflusses benötigt das schätzende Verfahren 3D-Informationen über die Szenerie. Diese können entweder durch eine RGB-D-Kamera, welche zusätzlich zum Farbwert ebenfalls die Tiefe messen kann, oder mithilfe von zwei RGB-Kameras in einem Stereo-Aufbau ermittelt werden. In der vorliegenden Arbeit wird die letztgenannte Möglichkeit verwendet. Dabei arbeiten Szenenflussverfahren entweder direkt auf Grundlage der Bildpaare oder sie nutzen – als Vorverarbeitungsschritt zur Schätzung der Disparität – Stereoschätzer, die auf Basis der aufgenommenen Stereobildpaare die Tiefe der Szene rekonstruieren können. Im Rahmen dieser Arbeit wird die zweite Variante genutzt. Da sie ebenfalls die Aufnahme von Stereobildpaaren benötigt, soll das Stereo-Kamera-Setup in diesem Abschnitt vorgestellt werden.

Kameras mit lichtempfindlichen Sensoren werden im Bereich des Maschinensehens gebräuchlich mit dem Lochkameramodell approximiert [25]. Es beschreibt die Projektion eines Punktes $P \in \mathbb{R}^3$ mit Weltkoordinaten $(X_W, Y_W, Z_W)$ auf die Bildebene $\Omega \subset \mathbb{R}^2$ einer Kamera. Das Ergebnis der Projektion ist der Punkt $p \in \Omega$ mit Bildkoordinaten $(x_B, y_B)$. Dieses Prinzip wird in Abbildung 2.2 mit der Erweiterung vorgestellt, dass der Punkt $P$ auf die Bildebenen von zwei Kameras projiziert wird. $C_L$ und $C_R$ sind die Kamerazentren in Weltkoordinaten und $(x_0, y_0)$ die Mittelpunkte der jeweiligen Kamerabilder relativ zur oberen linken Ecke der Bildebene in Bildkoordinaten. Die Projektion des Punktes $P$ auf der linken Kamera befindet sich auf den Bildkoordinaten $(x_L, y_L)$, auf der rechten Kamera auf $(x_R, y_R)$. Die Kameras werden dabei als identisch angenommen. Dieser Aufbau erzeugt ein Stereobildpaar, das eine Szenerie aus zwei verschiedenen Blickwinkeln zum gleichen Zeitpunkt $t$ mithilfe von zwei Kameras erfasst. Dabei werden beide Kameras in einem gemeinsamen Weltkoordinatensystem beschrieben, wodurch die Rekonstruktion der Tiefe möglich ist [44]. Abbildung 2.2 zeigt die Geometrie der sogenannten ortho-parallelen Kameraeinstellung, in der die Kameras ausschließlich eine Verschiebung in $x$-Richtung, jedoch die gleiche Orientierung, aufweisen. Im Gegensatz dazu sind sie im allgemeinen Fall zusätzlich etwas zueinander gedreht, wodurch die Berechnungen im späteren Verlauf jedoch um einiges komplexer werden. Eine quantitative

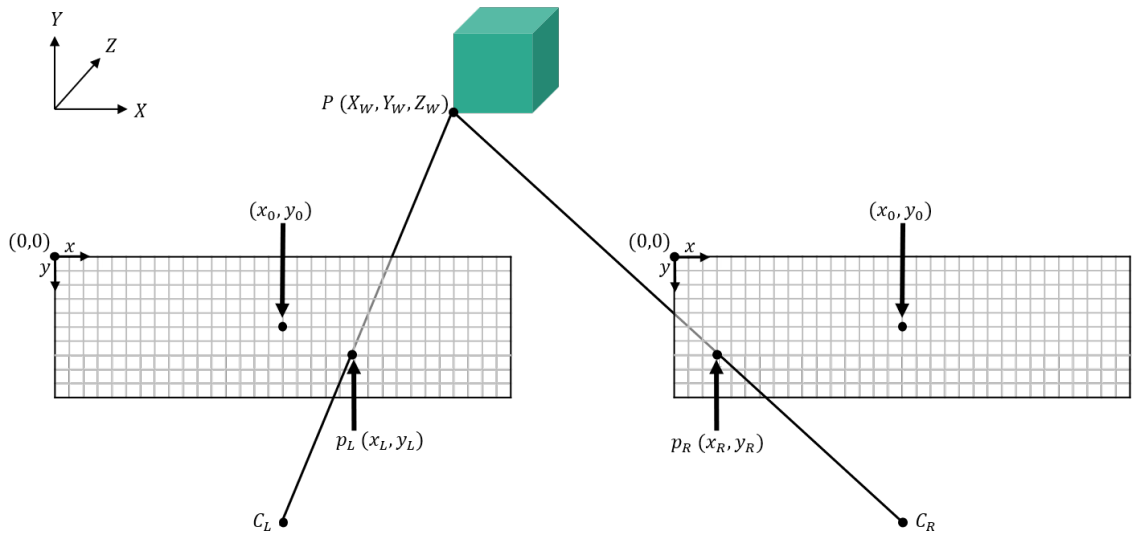



**Abbildung 2.2:** Die Geometrie eines Stereobildpaares (ortho-parallele Kameraeinstellung) mit Kamerazentren $C_L$ und $C_R$, Mittelpunkten der Bildebenen $(x_0, y_0)$ relativ zur jeweiligen oberen linken Ecke sowie den projizierten Punkten $p_L$ und $p_R$ des Oberflächenpunktes $P$.

Bewertung der in dieser Arbeit entwickelten Modelle ist auch mit den erstgenannten, idealisierten Annahmen möglich. Daher wird bei der mathematischen Beschreibung auf den allgemeineren Fall von zusätzlich zueinander geneigten Kameras verzichtet.

Anhand des hier vorgestellten und in Abbildung 2.2 illustrierten Stereo-Kamera-Setups soll im nachstehenden Abschnitt die Disparität erläutert werden.

### 2.2.3 Disparität

In einem ortho-parallelen Stereo-Kamera-Setup wird die Disparität eines Oberflächenpunktes nach dem Stereo-Prinzip bestimmt, wodurch anschließend seine Tiefe rekonstruiert werden kann. Im Folgenden wird dies anhand von Abbildung 2.3 erklärt.

Das Stereo-Kamera-Setup in Abbildung 2.3 entspricht dem in Abbildung 2.2 vorgestellten, bei dem ein Oberflächenpunkt $P$ auf die Bildebenen der zwei Kameras projiziert wird. Zusätzlich wurde die Grundlinie (*baseline*), welche die Kamerazentren verbindet, eingezeichnet. Außerdem wurde die Epipolarebene (*epipolar plane*), welche die Kamerazentren und den Punkt $P$ enthält, orange eingefärbt. Ebenfalls markiert wurden die beiden Schnittgeraden, in der sich Epipolarebene und jeweilige Bildebene schneiden – die sogenannten Epipolarlinien (*epipolar lines*).

Ein Oberflächenpunkt $P$ mit unbekannten Weltkoordinaten $(X_W, Y_W, Z_W)$, welcher auf die bekannte Position $p_L = (x_L, y_L)$ im Bildbereich der linken Kamera projiziert wird, befindet sich im Bildbereich der rechten Kamera an der noch unbekannten Position $p_R = (x_R, y_R)$. Um die Disparität zu bestimmen, muss der zu $p_L$ korrespondierende Punkt $p_R$ ermittelt werden. Hierfür wird die epipolare Bedingung (*epipolar constraint*) genutzt, die vorgibt, dass der Punkt $p_R$ auf der – durch $p_L$ und der Grundlinie festgelegten – Epipolarlinie liegen muss. Diese Bedingung

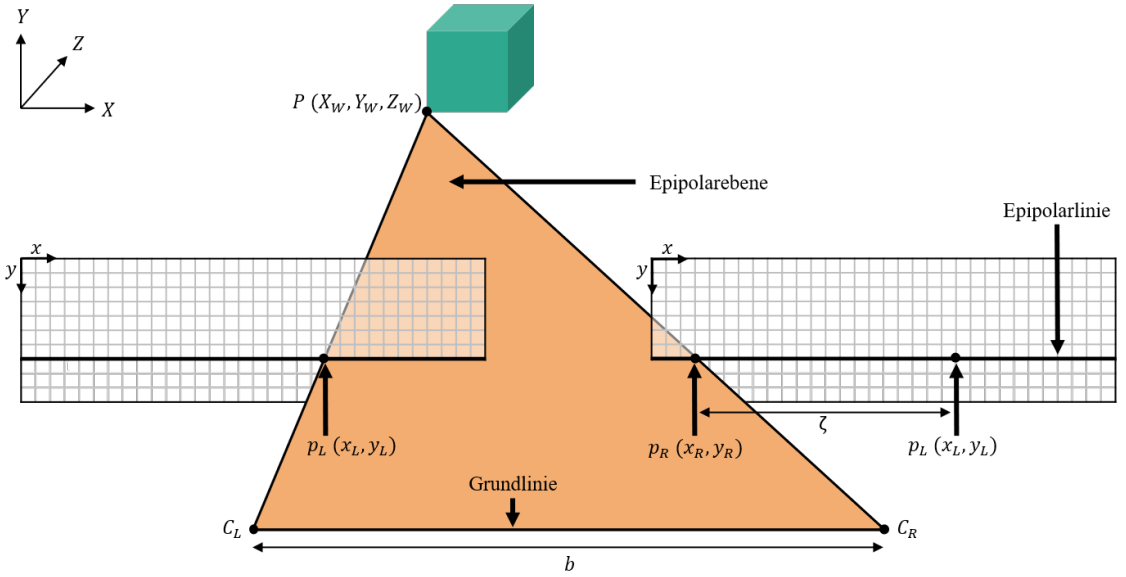



**Abbildung 2.3:** Nutzung der Geometrie eines Stereobildpaares (ortho-parallele Kameraeinstellung) mit Kamerazentren $C_L$ und $C_R$ sowie der Grundlinienlänge $b$ zur Disparitätsbestimmung. Der Punkt $P$ wird auf die Bildpunkte $p_L$ und $p_R$ projiziert. Die Disparität lässt sich aus der Differenz der $x$-Koordinaten der Bildpunkte bestimmen.

schränkt den Suchraum ein, in dem sich $p_R$ befindet. Der ortho-parallele Aufbau hat hier den Vorteil, dass die Epipolarlinien horizontal verlaufen und die selbe $y-$Koordinate besitzen. Aufgrund der gewählten Kameraanordnung erfolgt daher die Verschiebung zwischen der Projektion des Punktes $P$ auf der linken Kamera $p_L$ zum korrespondierenden Punkt $p_R$ auf der rechten Kamera nur entlang der $x$-Achse. Diese Verschiebung wird allgemein als Disparität $\zeta$ bezeichnet. Die $x$-Koordinate des Punktes $p_R$ ergibt sich aus der Summe der $x$-Koordinate von $p_L$ und der gesuchten Disparität $\zeta$. Die Bildkoordinaten des Punktes $p_R$ lauten daher:

$$p_R = (x_L + \zeta,\ y_L). \tag{2.4}$$

Somit lässt sich die Disparität aus den korrespondierenden Punkten $p_L$ und $p_R$ berechnen:

$$\zeta(x_L, y_L) = x_R - x_L. \tag{2.5}$$

Seien nun $w_x$ und $w_y$ die Brennweite der Kameras, $b$ die Länge der Grundlinie und $(x_0, y_0)$ die Mittelpunkte der Kamerabilder relativ zur oberen linken Ecke der jeweiligen Bildebene in Bildkoordinaten. Die Projektion des Oberflächenpunktes $P = (X_W, Y_W, Z_W)$ auf den Punkt $p_L = (x_L, y_L)$ im Bildbereich der linken Kamera ist gemäß Wedel *et al.* [90]:

$$\begin{pmatrix} x_L \\ y_L \\ \zeta \end{pmatrix} = \frac{1}{Z_W} \begin{pmatrix} X_W \cdot w_x \\ Y_W \cdot w_y \\ -b \cdot w_x \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \\ 0 \end{pmatrix}. \tag{2.6}$$

Wurde die Disparität $\zeta$ mithilfe von Gleichung (2.5) berechnet, so lässt sich die Tiefe $Z_W$ des Oberflächenpunktes $P$ wie folgt bestimmen:

$$Z_W = \frac{-b \cdot w_x}{\zeta}. \tag{2.7}$$

Das Ziel eines jeden Stereo-Algorithmus ist es daher, die Disparität $\zeta$ der Oberflächenpunkte aller Objekte im Kamerabild zu bestimmen, um die 3D-Szene zu rekonstruieren [44]. Die Gesamtheit dieser Disparitäten für einen Zeitpunkt $t$ wird auch Disparitätskarte $\zeta(x, y, t)$ genannt.

### 2.2.4 Optischer Fluss

Im Gegensatz zur Disparität beschreibt der optische Fluss nicht die Verschiebung der Projektion eines Punktes zwischen zwei Kameras zu einem Zeitpunkt $t$, sondern die Verschiebung der Projektion eines Punktes auf der gleichen Kamera von Zeitpunkt $t$ zu $t + 1$. Der optische Fluss kann selbst als Projektion des dreidimensionalen Szenenflusses auf die zweidimensionale Bildebene gesehen werden, da die dreidimensionale Bewegung von Oberflächenpunkten auf den zweidimensionalen Bildraum projiziert wird. Um den optischen Fluss zu erfassen, wird nur eine Kamera benötigt, die aber zu zwei Zeitpunkten ein Bild aufnehmen muss. Dies wird in Abbildung 2.4 dargestellt.

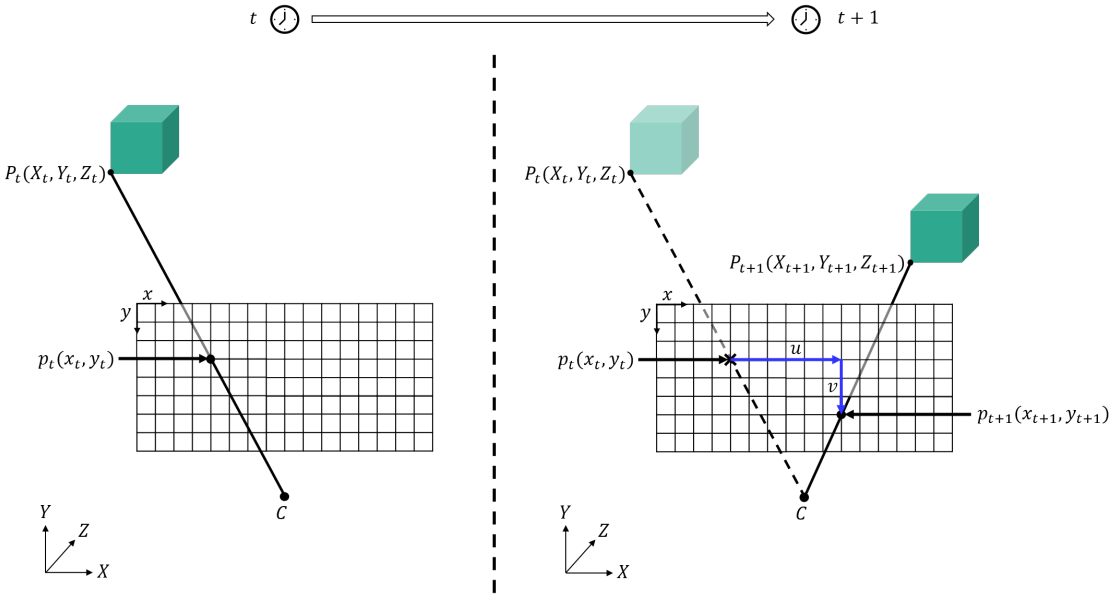



**Abbildung 2.4:** Schematische Darstellung zum optischen Fluss mit Kamerazentrum $C$: Zu Zeitpunkt $t$ wird ein Oberflächenpunkt auf die Bildebene auf den Bildpunkt $p_t$ projiziert. Von Zeitpunkt $t$ zu $t+1$ bewegt sich dieser Oberflächenpunkt und wird nun auf den Bildbereich der selben Kamera auf den Bildpunkt $p_{t+1}$ projiziert. Die Differenz $u$ der $x$-Koordinaten und $v$ der $y$-Koordinaten wird als optischer Fluss bezeichnet.

Zum Zeitpunkt $t$ wird ein Oberflächenpunkt mit Weltkoordinaten $(X_t, Y_t, Z_t)$ auf den Punkte $p_t$ mit Bildkoordinaten $(x_t, y_t)$ projiziert. Von Zeitpunkt $t$ zu $t + 1$ bewegt sich der Oberflächenpunkt zu den Weltkoordinaten $(X_{t+1}, Y_{t+1}, Z_{t+1})$ und wird nun auf den Punkte $p_{t+1}$ mit Bildkoordinaten $(x_{t+1}, y_{t+1})$ projiziert. Der optische Fluss beschreibt die Verschiebung als Vektor und kann für die korrespondierenden Punkte $p_t$ und $p_{t+1}$ folgendermaßen berechnet werden:

$$\begin{pmatrix} u(x_t, y_t) \\ v(x_t, y_t) \end{pmatrix} = \begin{pmatrix} x_{t+1} - x_t \\ y_{t+1} - y_t \end{pmatrix} \in \mathbb{R}^2, \tag{2.8}$$

wobei $u$ die $x$- und $v$ die $y$-Verschiebung beschreiben. Mit bekannten Werten von $u$ und $v$ lässt sich die neue Bildposition zum Zeitpunkt $t + 1$ wie folgt aus der vorherigen zum Zeitpunkt $t$ berechnen:

$$x_{t+1} = x_t + u(x_t, y_t), \quad y_{t+1} = y_t + v(x_t, y_t). \tag{2.9}$$

Wie gezeigt worden ist, beschreibt der optische Fluss nur die zweidimensionale Projektion der dreidimensionalen Bewegung von Oberflächenpunkten. Wird jedoch mithilfe einer zweiten Kamera zusätzlich die Disparität erfasst, kann die dreidimensionale Bewegung von Oberflächenpunkten – als Szenenfluss – rekonstruiert werden.

Die Parametrisierung des Szenenflusses aus optischen Fluss und Disparität wird im folgenden Abschnitt geschildert.

### 2.2.5 Parametrisierung des Szenenflusses

In Abschnitt 2.2.3 und 2.2.4 wurde gezeigt, wie die Disparität mithilfe von zwei Kameras zu einem Zeitpunkt und der optische Fluss mithilfe einer Kamera zu zwei Zeitpunkten definiert wird. Eine Kombination dieser beiden Prinzipien, der Szenenfluss, erlaubt die Rekonstruktion der dreidimensionalen Bewegung der Oberflächenpunkte und soll in diesem Abschnitt parametrisiert werden.

Allgemein kann der Szenenfluss als zweidimensionaler Array von dreidimensionalen Vektoren gesehen werden, der für jeden Oberflächenpunkt im zweidimensionalen Bildbereich die dreidimensionale Bewegung angibt. Eine erste intuitive Parametrisierung eines solchen Vektors wäre die Positionsänderung in $x$-, $y$ -und $z$-Richtung. Da jedoch die Tiefeninformation hier nicht direkt als Tiefe, sondern in Form der Disparität (siehe Abschnitt 2.2.3) gemessen wird, ist eine alternative Parametrisierung sinnvoll. Im Rahmen dieser Arbeit soll der Szenenfluss wie folgt definiert werden:

$$\boldsymbol{w}(x, y) := \begin{pmatrix} u(x, y) \\ v(x, y) \\ d(x, y) \end{pmatrix} \in \mathbb{R}^3. \tag{2.10}$$

Der Szenenfluss setzt sich aus optischem Fluss $(u(x, y), v(x, y))^\top \in \mathbb{R}^2$ und Zieldisparität $d(x, y) \in \mathbb{R}$ zusammen. Der optische Fluss beschreibt die zweidimensionale Bewegung der Projektion eines Oberflächenpunktes auf dem Bildbereich. Dabei beschreibt $u$ die Bewegung in $x$- und $v$ in $y$-Richtung vom Referenzzeitschritt $t$ zu $t + 1$, wie in Abschnitt 2.2.4 eingeführt. Die dritte Komponente des Szenenflusses ist die Zieldisparität $d$ und stellt für jede, auf das Referenzframe registrierte, Bildposition $(x, y)$ die Disparität zum Zeitpunkt $t+1$ dar. Der komplexe Zusammenhang zwischen Disparitäten und optischem Fluss wird in Abbildung 2.5 visualisiert und im Folgenden kurz erläutert.

Die Startdisparität $\zeta(x, y, t)$ beschreibt die $x$-Verschiebung der Bildposition vom linken zum rechten Bild zum Zeitpunkt $t$, also von Position $(x, y)$ zu $(x + \zeta(x, y, t), y)$, wie in Abschnitt 2.2.3 näher ausgeführt. Analog dazu beschreibt die Zieldisparität $d(x, y)$ die $x$-Verschiebung der Bildposition vom linken zum rechten Bild zum späteren Zeitpunkt $t+1$, also von Position $(x+u(x, y), y+v(x, y))$ zu $(x + u(x, y) + d(x, y), y + v(x, y))$, mit dem Unterschied, dass die Werte von $u, v$ und $d$ auf die Position $(x, y)$ des Referenzframes – also des linken Bildes zum Zeitpunkt $t$ – registriert sind. Demnach lässt sich anhand von $\zeta(x, y, t)$ und $d(x, y)$ sowie mithilfe von $u$ und $v$ direkt ablesen,

welche Disparität ein Punkt, der sich in der linken Kamera von $(x, y)$ zur Position $(x + u(x, y), y + v(x, y))$ bewegt, zu den Zeitpunkten $t$ und $t + 1$ besitzt und seine Bewegung ist eindeutig definiert und lokalisiert.

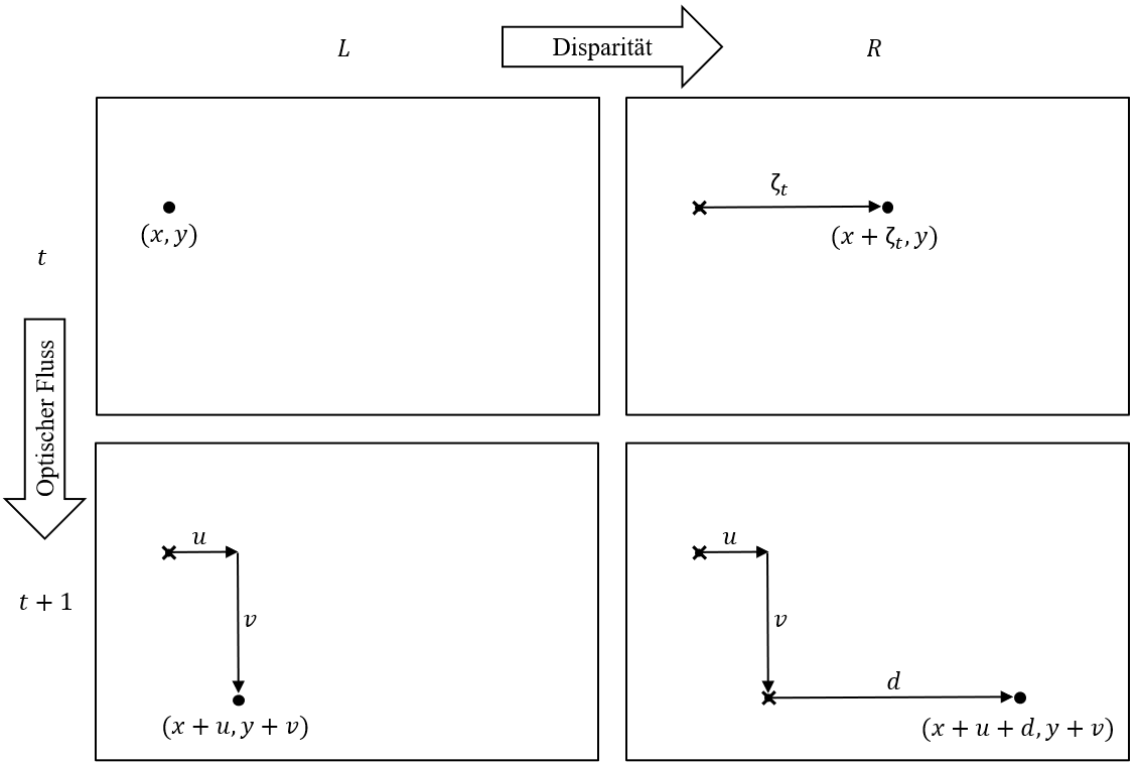



**Abbildung 2.5:** Zusammenhang von linken und rechten Bildern ($L$ und $R$) zu den Zeitpunkten $t$ und $t + 1$: Das linke Bild zum Zeitpunkt $t$ ist das Referenzframe. Ein Oberflächenpunkt eines Objekts, welches auf die Pixelposition $(x, y)$ des Referenzframes projiziert wird, findet sich im rechten Bild auf der Position $(x + \zeta_t, y)$. Dabei beschreibt $\zeta_t = \zeta(x, y, t)$ die horizontale Disparität, die aufgrund des orthoparallelen Stereo-Kamera-Setups entsteht. Der selbe Oberflächenpunkt bewegt sich vom Zeitpunkt $t$ zu $t + 1$ und wird zum späteren Zeitpunkt auf die neue Bildposition $(x + u(x, y), y + v(x, y))$ des linken Bildes projiziert. Dabei beschreiben $u$ und $v$ die Verschiebung in $x$- bzw. $y$-Richtung im linken Bild. Die horizontale Disparität zwischen dieser Pixelposition und der im rechten Bild zum Zeitpunkt $t + 1$ ist die Zieldisparität $d$, womit sich der Oberflächenpunkt im rechten Bild an der Position $(x + u(x, y) + d(x, y), y + v(x, y))$ befindet. Wie $u$ und $v$ ist auch die Zieldisparität $d$ auf das Referenzframe zum Zeitpunkt $t$ registriert.

Zur Rekonstruktion der dreidimensionalen Bewegung eines Oberflächenpunktes werden neben dem Bild $f$ der linken Kamera zum Zeitpunkt $t$ die Werte $u(x, y), v(x, y), \zeta(x, y)$ und $d(x, y)$ benötigt, aus denen die dreidimensionalen Positionen zum Zeitpunkt $t$ und $t + 1$ berechnet werden können. Im KITTI-Datensatz [52] wird die Startdisparität $\zeta$ zum Zeitpunkt $t$ als $disp_0$ und die auf das linke Bild zum Zeitpunkt $t + 1$ registrierte Zieldisparität (hier $d$) als $disp_1$ bezeichnet. Sie gehören mit dem berechneten optischen Fluss $u$ und $v$ zum standardmäßigen Output aktueller Szenenflussverfahren.

Manche Szenenflussverfahren schätzen zusätzlich zu den drei Werten $u$, $v$ und $d$ die ursprüngliche Startdisparität zum Zeitpunkt $t$ (also neben $disp_1$ auch $disp_0$), zum Beispiel die von Huguet und Devernay [28], Ma *et al.* [42], Behl *et al.* [5] und Li *et al.* [37]. Dies ist notwendig, wenn keine Disparitätskarte $\zeta$ eines Stereoschätzers als Eingabegröße verwendet wird, kann aber auch bei einer ungenauen oder verrauschten Disparitätskarte sinnvoll sein, da dadurch die Stereoschätzung und damit die Anfangs- und Endpositionen der Oberflächenpunkte genauer bestimmt werden können. Allerdings beschränken sich die aktuell besten Verfahren auf die Schätzung der drei Komponenten $u$, $v$ und $d$ – wie bei Liu *et al.* [39], Sommer *et al.* [69], Teed und Deng [77], Badki *et al.* [3] und Yang und Ramanan [95, 96] –, so auch diese Arbeit. Dabei wird die Startdisparität $disp_0$, welche von einem vorherigen Stereoschätzer als $\zeta(x, y, t)$ berechnet wurde, unverändert ausgegeben.

Anzumerken ist an dieser Stelle, dass einigen Verfahren, wie beispielsweise das von Wedel *et al.* [90], Rabe *et al.* [60] oder Hung *et al.* [29], eine andere Definition der dritten Komponente des Szenenflusses $d$ zugrunde liegt. Statt der – auf das linke Bild zu Zeitpunkt $t + 1$ registrierten – Zieldisparität $disp_1$ wird hier die Disparitätsänderung vom Zeitpunkt $t$ zu $t + 1$ geschätzt. Diese Änderung kann mit der hier gewählten Parametrisierung durch $d(x, y) - \zeta(x, y, t)$ ebenfalls berechnet werden.

Im Folgenden werden für die Ausdrücke $u(x, y), v(x, y)$ und $d(x, y)$ die abgekürzte Schreibweise $u, v$ und $d$ mit impliziten Argumenten $(x, y)$ verwendet. Außerdem soll an dieser Stelle die erweiterte Notation des Szenenflusses,

$$\boldsymbol{w} := \begin{pmatrix} u \\ v \\ d \\ 1 \end{pmatrix}, \tag{2.11}$$

vorgestellt werden. Sie ermöglicht im Rahmen der Modellierung eine kompakte Schreibweise des Datenterms, die in Abschnitt 3.1 eingeführt wird. Konvention in dieser Arbeit ist, dass in den Gleichungen die erweiterte Form des Szenenflusses nach Gleichung (2.11) verwendet wird, während ansonsten die reduzierte Form nach Gleichung (2.10) genutzt wird.

# 3 Szenenflussschätzung mithilfe von Variationsansätzen

Obschon die Variationsrechnung historisch gesehen aus konkreten Fragestellungen der Geometrie und Physik entstanden ist [34], hat sie sich auch im Bereich der Bildverarbeitung und des Maschinensehens etabliert und findet in zahlreichen Arbeiten Verwendung. Der erste Variationsansatz zur Berechnung des optischen Flusses stammt von Horn und Schunck [27] aus dem Jahre 1981. Viele nachfolgende Abhandlungen, die das gleiche Problem lösen, basieren auf dieser Methode, wie zum Beispiel die von Black und Anandan [7], Brox *et al.* [9], Nagel und Enkelmann [54], Werlberger *et al.* [93] und Zimmer *et al.* [101].

Wie in Abschnitt 2.2.4 dargelegt, kann der optische Fluss als Projektion des Szenenflusses auf den zweidimensionalen Bildbereich gesehen werden. Daher ergeben sich für den Szenenfluss ähnliche Herausforderungen, wie beispielsweise Okklusionen, große Bewegungen, Objekte, die sich über den Bildrand hinausbewegen oder variierende Beleuchtungen. Gleichzeitig eröffnet die Ähnlichkeit dieser beiden Konzepte die Möglichkeit, die beim optischen Fluss verwendeten Methoden der Variationsrechnung auf den Szenenfluss zu übertragen. Die von Horn und Schunck [27] entwickelte Modellierung der Variationsrechnung zur Bestimmung des optischen Flusses kann daher auch als Basis für die Variationsrechnung des Szenenflusses verwendet werden, wie die Erweiterungen des von Horn und Schunck aufgestellten Grundmodells bei Basha *et al.* [4], Huguet und Devernay [28] sowie Wedel *et al.* [90] zeigen.

In der Variationsrechnung wird das zu lösende Problem als Minimierungsproblem eines Energiefunktionals formuliert. Im Gegensatz zu einer Funktion, welche eine Abbildung vom $\mathbb{R}^n$ zum $\mathbb{R}^m$ darstellt, ist ein Funktional keine Abbildung endlich vieler reeller Variablen. Der Definitionsbereich ist allgemeiner zu fassen und wird vom linearen Raum $\mathbb{R}^n$ zu einem unendlich dimensionalen Funktionsraum ausgeweitet [34]. Das bedeutet, ein Funktional nimmt eine Funktion als Eingabe und berechnet daraus einen Skalar- oder Vektorwert. Um nun das gesuchte Minimum des Energiefunktionals zu finden, werden aus seinen partiellen Ableitungen die Euler-Lagrange-Gleichungen gebildet, welche bei Gleichsetzung mit Null die notwendigen Bedingungen für besagtes Extremum darstellen.

Dieses Kapitel leitet durch den gesamten Prozess der Berechnung des Szenenflusses mithilfe von Variationsansätzen – von der Modellierung der Annahmen als Energiefunktional über die Minimierung mithilfe der Euler-Lagrange-Gleichungen, der Diskretisierung bis hin zum iterativen Lösungsverfahren – und bildet die theoretische Grundlage für diese Arbeit.

## 3.1 Modellierung

Zur Schätzung des Szenenflusses werden entweder zwei aufeinanderfolgende Bildpaare einer Stereosequenz oder ein Bildpaar mit zusätzlichen Tiefeninformationen benötigt. Das in dieser Arbeit verwendete Verfahren nutzt jeweils ein Bild zum Zeitpunkt $t$ und $t + 1$, wie auch die jeweilige Tiefeninformation als Disparitätskarte (siehe Abschnitt 2.2.1).

Nun soll das Szenenflussproblem als Minimierungsproblem eines Energiefunktionals, mithilfe der kontinuierlichen Formulierung der Bildfolge $f$ und der Disparitätskarte $\zeta$, beschrieben werden. Dafür werden globale Annahmen, welche die gegebenen Eingabegrößen $f$ und $\zeta$ mit dem gesuchten optischen Fluss $u$, $v$ und der Zieldisparität $d$ verknüpfen (siehe Abschnitt 3.1.1 und 3.1.2), in ein Energiefunktional $E(\boldsymbol{w})$ eingebettet. Dieses weist jedem möglichen Szenenfluss $\boldsymbol{w}$ einen skalaren Energiewert $E$ zu. Dieser Wert kann als Güte der Szenenflussschätzung in Bezug auf die gegebenen globalen Annahmen gesehen werden, welche über die gesamte Bildebene gelten müssen. Je weniger Abweichungen es von diesen Annahmen gibt, desto niedriger ist $E$ und desto besser die Schätzung.

Diese Formulierung als Energiefunktional erlaubt es, das Szenenflussproblem klar zu modellieren und die Güte anhand des Energiewertes $E$ abzuschätzen. In allgemeiner Form ist das Energiefunktional für einen Modellierungsterm $F$ gegeben durch:

$$E(\boldsymbol{w}) = \int_{\Omega} F(\boldsymbol{w}) \, \mathrm{d}x \, \mathrm{d}y. \tag{3.1}$$

Die Integration erfolgt dabei über den gesamten Bildbereich $\Omega \subset \mathbb{R}^2$. Dem Beispiel von Horn und Schunck [27] folgend, setzte sich der Modellierungsterm $F$ üblicherweise aus einem Daten- und einem Glattheitsterm zusammen:

$$F(\boldsymbol{w}) = D(\boldsymbol{w}) + \alpha \cdot R(\boldsymbol{w}). \tag{3.2}$$

Der Datenterm $D$ bestraft Abweichungen von den Annahmen, die häufig als Konstanzannahmen formuliert werden. Der mit dem Parameter $\alpha$ gewichtete Glattheitsterm $R$ bestraft die Abweichung von der Glattheit der Lösung. Durch die angenommene Glattheit können Lücken, die durch fehlende Informationen, die z.B. durch Überlappungen oder Verschwinden von Objekten entstehen, geschlossen werden, indem an diesen Stellen Informationen aus den benachbarten Pixeln herangezogen werden [6]. Dies ist der sogenannte *filling-in-effect*. Der Parameter $\alpha$ stellt eine Balance dieser beiden Terme dar: Ein größerer Wert von $\alpha$ entspricht einer stärkeren Betonung des Glattheitsterms, ein kleinerer Wert einer stärkeren Betonung des Datenterms. Im Extremfall wird bei einem Wert von unendlich der Datenterm ignoriert, sodass die Minimierung auf einen konstanten Fluss hinausläuft.

In den folgenden zwei Abschnitten wird detaillierter auf den Aufbau des Daten- und Glattheitsterms eingegangen.

### 3.1.1 Datenterm

Der Datenterm $D(\boldsymbol{w})$ ist ein essentieller Bestandteil der Modellierung eines Minimierungsproblems mithilfe der Variationsrechnung und beschreibt Konstanzannahmen, die basierend auf der Bildfolge $f$ und der Disparitätskarte $\zeta$ formuliert werden.

In einem ersten Ansatz werden zunächst zwei Annahmen im Datenterm berücksichtigt (in Kapitel 5 werden Erweiterungen um zusätzliche Annahmen hinzugefügt). Die erste Annahme betrifft den optischen Fluss und entspricht der von Horn und Schunck [27] eingeführten Grauwertkonstanz (*brightness constancy assumption* kurz *bca*):

$$f(x+u, y+v, t+1) - f(x, y, t) = 0. \tag{3.3}$$

Es wird angenommen, dass es keine Veränderungen der Beleuchtung gibt, sodass die Helligkeit $f(x, y, t)$ eines Oberflächenpunktes zum Zeitpunkt $t$ der Helligkeit $f(x+u, y+v, t+1)$ dieses Punktes nach der Bewegung $u$ und $v$ zum Zeitpunkt $t+1$ entspricht.

Die zweite Konstanzannahme befasst sich mit der Zieldisparität, die auf das linke Bild zum Zeitpunkt $t$ registriert ist. Es soll gelten, dass $d$ an der Stelle $(x, y)$ der Disparität zum Zeitpunkt $t+1$ an der Stelle $(x+u, y+v)$ entspricht. Dies wird auch als Grauwertkonstanz der Zieldisparität bezeichnet:

$$\zeta(x+u, y+v, t+1) - d = 0. \tag{3.4}$$

Die gewichtete Summe dieser beiden Annahmen mit quadratischer Bestrafung von Abweichungen ergibt den Datenterm

$$\begin{aligned} D(u, v, d) = &\Big(f(x+u, y+v, t+1) - f(x, y, t)\Big)^2 \\ &+ \mu \cdot \Big(\zeta(x+u, y+v, t+1) - d\Big)^2, \end{aligned} \tag{3.5}$$

mit $\mu \geq 0$ als Parameter, welcher die Gewichtung der Konstanzannahme der Disparität gegenüber der Grauwertkonstanz des Bildes festlegt. Der Datenterm $D$ ist in dieser Form nichtlinear in $u$ und $v$, denn sie kommen in den Argumenten von $f$ und $\zeta$ vor. Das erschwert die Minimierung des Energiefunktionals $E(\boldsymbol{w})$. Werden $f(x+u, y+v, t+1)$ und $\zeta(x+u, y+v, t+1)$ anhand der Taylorformel um den Punkt $(x, y)$ zum Zeitpunkt $t$ linearisiert, so erhält man folgende Approximationen:

$$f(x+u, y+v, t+1) \approx f(x, y, t) + f(x, y, t)_x u + f(x, y, t)_y v + f(x, y, t)_t, \tag{3.6}$$

$$\zeta(x+u, y+v, t+1) \approx \zeta(x, y, t) + \zeta(x, y, t)_x u + \zeta(x, y, t)_y v + \zeta(x, y, t)_t. \tag{3.7}$$

Diese Approximationen in Gleichung (3.5) eingesetzt, ergeben den linearisierten Datenterm, wobei die Argumente $(x, y, t)$ von $f$ und $\zeta$ für eine verkürzte Schreibweise hier, wie im Weiteren, nicht mehr explizit aufgeführt werden:

$$\begin{aligned} D(u, v, d) = &(f_x u + f_y v + f_t)^2 \\ &+ \mu \cdot (\zeta_x u + \zeta_y v + \zeta_t + \zeta - d)^2. \end{aligned} \tag{3.8}$$

Diese Linearisierung gilt für kleine Werte von $u, v, d$ und falls $f$ und $\zeta$ die notwendige Glattheit aufweisen. In dieser Arbeit wird davon ausgegangen, dass diese Linearisierungen eine hinreichend gute Näherung darstellen.

Um Gleichung (3.8) noch kompakter zu notieren, können die einzelnen Annahmen in Tensorschreibweise formuliert werden. Diese wurde von Bruhn und Weickert [10] eingeführt und findet in zahlreichen Arbeiten, wie beispielsweise bei Bruhn [11], Nüssle [55] und Mehl *et al.* [49], wie auch in der vorliegenden Arbeit Verwendung.

Mit der erweiterten Form des Szenenflusses aus Gleichung (2.11) kann eine Konstanzannahme kompakt durch $\boldsymbol{w}^\top \mathbf{J} \boldsymbol{w}$ ausgedrückt werden. Mehrere Einzeltensoren verschiedener linearer Annahmen lassen sich zu einem Gesamttensor zusammenfassen. Im konkreten Fall wird aus den Einzeltensoren der linearisierten Konstanzannahmen zur Bildfolge $\mathbf{J}_{Bild}$ und zur Disparitätskarte $\mathbf{J}_{Disp}$, die sich wie folgt zusammensetzen,

$$\mathbf{J}_{Bild} = \begin{pmatrix} f_x \\ f_y \\ 0 \\ f_t \end{pmatrix} \cdot \begin{pmatrix} f_x \\ f_y \\ 0 \\ f_t \end{pmatrix}^\top, \quad \mathbf{J}_{Disp} = \begin{pmatrix} \zeta_x \\ \zeta_y \\ -1 \\ \zeta_t + \zeta \end{pmatrix} \cdot \begin{pmatrix} \zeta_x \\ \zeta_y \\ -1 \\ \zeta_t + \zeta \end{pmatrix}^\top, \tag{3.9}$$

der Gesamttensor $\mathbf{J}$ als gewichtete Summe gebildet:

$$\mathbf{J} = \mathbf{J}_{Bild} + \mu \cdot \mathbf{J}_{Disp}. \tag{3.10}$$

Damit erhält Gleichung (3.8) die folgende kompakte Form:

$$D(\boldsymbol{w}) = \boldsymbol{w}^\top \mathbf{J} \boldsymbol{w}. \tag{3.11}$$

Die Tensornotation hat den großen Vorteil, dass weitere Konstanzannahmen, wie sie in Kapitel 5 eingeführt werden, auf einfache Art und Weise eingebunden werden können, ohne die weiteren Schritte der Minimierung gravierend zu verändern, da die zusätzlichen Annahmen in den Gesamttensor $\mathbf{J}$ eingebettet werden.

Im Allgemeinen reichen die im Datenterm enthaltenen Annahmen nicht aus, um eine eindeutige Lösung des Szenenflusses zu berechnen. Daher wird im Folgenden ein zusätzlicher Glattheitsterm eingeführt.

### 3.1.2 Glattheitsterm

Überlappungen, das Verschwinden von Objekten und das sogenannte Aperturproblem[1] führen zu lückenhaften Informationen im optischen Fluss und der Zieldisparität. Daher kann der Szenenfluss nicht global über das komplette Bild berechnet werden, sondern nur an den Stellen, an denen die nötigen Informationen verfügbar sind. Um diesem Problem entgegenzuwirken, wird neben dem Datenterm $D(\boldsymbol{w})$ zusätzlich ein Glattheitsterm $R(\boldsymbol{w})$ eingesetzt, der auch als Regularisierungsterm bezeichnet wird und den sogenannten *filling-in-effect* ermöglicht: An Positionen mit fehlenden Informationen werden diese von den benachbarten Pixeln bezogen. Diese *propagation* stellt einen Diffusionsprozess dar, der die Lücken füllt und eine dichte Berechnung des Szenenflusses erlaubt.

Konkret trifft der Glattheitsterm im Gegensatz zum Datenterm keine Annahmen über die Bildfolge $f$ oder Disparitätskarte $\zeta$, sondern über den Szenenfluss $\boldsymbol{w} = (u, v, d)^\top$ selbst. Die grundlegende Annahme für diesen Term bildet die Glattheit von $u, v$ und $d$, die besagt, dass sich nicht alle Pixel unabhängig voneinander bewegen, wenn ausgedehnte Objekte abgebildet werden. Wie bei Horn und Schunck [27] eingeführt, wird zur Modellierung der Glattheit von $u$ und $v$ deren Gradientenbetrag

[1] Das Aperturproblem besagt, dass lediglich die Flusskomponente orthogonal zu den Bildkanten berechnet werden kann, wie zum Beispiel bei Zimmer *et al.* [102] ausgeführt wird.

quadratisch bestraft. Wie auch bei Rabe *et al.* [60] wird in dieser Arbeit zusätzlich $d$ mit dem gleichen Vorgehen regularisiert. Damit ergibt sich folgender Glattheitsterm, der große Gradientenbeträge von $u, v$ und $d$ bestraft:

$$R(u, v, d) = |\nabla u|^2 + |\nabla v|^2 + \beta \cdot |\nabla d|^2. \tag{3.12}$$

Mit $\beta \geq 0$ ist eine zusätzliche Gewichtung der Glattheit der Zieldisparität möglich.

Im Gesamtfunktional wird nun mithilfe des Gewichts $\alpha$ ein Kompromiss zwischen dem kompakten, linearisierten Datenterm und dem soeben beschriebenen Glattheitsterm gefunden:

$$E(\boldsymbol{w}) = \int_\Omega \left( \boldsymbol{w}^\top \mathbf{J} \boldsymbol{w} + \alpha \left( |\nabla u|^2 + |\nabla v|^2 + \beta \cdot |\nabla d|^2 \right) \right) \mathrm{d}x \, \mathrm{d}y. \tag{3.13}$$

Mit dieser Beschreibung der Annahmen über $f$ und $\zeta$ im Datenterm sowie über $u, v$ und $d$ im Glattheitsterm ist eine erste Modellierung des Szenenflusses $\boldsymbol{w}$ abgeschlossen. Für eine bestmögliche Schätzung von $\boldsymbol{w}$ muss das Minimum des Funktionals $E(\boldsymbol{w})$ gefunden werden. Wie dabei vorgegangen wird, soll im nächsten Abschnitt beschrieben werden.

## 3.2 Minimierung

Extremwerte eines Variationsproblems, wie beispielsweise in Gleichung (3.13) modelliert, sind durch das Gleichungssystem der Euler-Lagrange-Gleichungen gegeben [15]. Analog zu einer kontinuierlichen Funktion $g(x)$, bei der die notwendige Bedingung für ein Minimum durch $g'(x) \stackrel{!}{=} 0$ gegeben ist, bilden die Euler-Lagrange-Gleichungen die notwendige Bedingung zum Finden des Minimierers für $E(\boldsymbol{w})$. Im Allgemeinen ist eine Lösung, die ein solches Funktional minimiert, selbst eine Funktion, in diesem Fall $\boldsymbol{w}(x, y) = (u(x, y), v(x, y), d(x, y))^\top$.

Die Euler-Lagrange-Gleichungen sind ein System von Differentialgleichungen, welche um 1755 unabhängig voneinander von Euler und Lagrange entwickelt wurden [34]. Um das konkrete System zum Modell aus Gleichung (3.13) herzuleiten, werden zunächst die allgemeinen Euler-Lagrange-Gleichungen für den ein- und mehrdimensionalen Fall dargestellt.

Für ein eindimensionales Energiefunktional,

$$E(u) = \int_a^b F(x, u, u') \, \mathrm{d}x, \tag{3.14}$$

das die unbekannte Funktion $u$ wie auch deren Ableitung $u'$ enthält, reduziert sich das System der Euler-Lagrange-Glechungen auf folgenden Ausdruck:

$$0 = F_u - \frac{\mathrm{d}}{\mathrm{d}x} F_{u'}. \tag{3.15}$$

Zusätzlich muss für die Randwerte $x = a$ und $x = b$ die Neumann-Randbedingung,

$$F_{u'} = 0, \tag{3.16}$$

erfüllt werden. Die Herleitung ist z.B. bei Kielhöfer [33] beschrieben.

In dieser Arbeit begrenzt sich die Minimierung auf zweidimensionale Funktionale der Form

$$E(\boldsymbol{w}) = \int_{\Omega} F(x, y, u, v, d, u_x, u_y, v_x, v_y, d_x, d_y)\,\mathrm{d}x\,\mathrm{d}y, \tag{3.17}$$

mit zweidimensionaler Domäne $(x, y) \in \Omega \subset \mathbb{R}^2$ und dreidimensionaler Co-Domäne $\boldsymbol{w} = (u, v, d)^\top \in \mathbb{R}^3$. Die notwendigen Bedingungen zur Bestimmung des Minimierers $\boldsymbol{w}$ sind über die folgenden drei Euler-Lagrange-Gleichungen gegeben:

$$0 = F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y}, \tag{3.18}$$

$$0 = F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y}, \tag{3.19}$$

$$0 = F_d - \frac{\partial}{\partial x} F_{d_x} - \frac{\partial}{\partial y} F_{d_y}. \tag{3.20}$$

Eine allgemeine Herleitung ist z.B. bei Courant und Hilbert [15] zu finden. Mit dem Normalenvektor $\boldsymbol{n}$, der orthogonal zur Grenze des Bildbereichs steht, lauten die Neumann-Randbedingungen im mehrdimensionalen Fall laut Maurer [44]:

$$\boldsymbol{n}^\top \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} = 0, \quad \boldsymbol{n}^\top \begin{pmatrix} F_{v_x} \\ F_{v_y} \end{pmatrix} = 0, \quad \boldsymbol{n}^\top \begin{pmatrix} F_{d_x} \\ F_{d_y} \end{pmatrix} = 0. \tag{3.21}$$

Konkret ergibt sich für das Energiefunktional zum Szenenfluss aus Gleichung (3.13), unter Benutzung der Tensornotation, folgendes Euler-Lagrange-Gleichungssystem:

$$0 = \mathbf{J}_{11} \cdot u + \mathbf{J}_{12} \cdot v + \mathbf{J}_{13} \cdot d + \mathbf{J}_{14} - \alpha \Delta u, \tag{3.22}$$

$$0 = \mathbf{J}_{12} \cdot u + \mathbf{J}_{22} \cdot v + \mathbf{J}_{23} \cdot d + \mathbf{J}_{24} - \alpha \Delta v, \tag{3.23}$$

$$0 = \mathbf{J}_{13} \cdot u + \mathbf{J}_{23} \cdot v + \mathbf{J}_{33} \cdot d + \mathbf{J}_{34} - \alpha\beta \Delta d, \tag{3.24}$$

mit den Neumann-Randbedingungen:

$$\boldsymbol{n}^\top \nabla u = 0, \quad \boldsymbol{n}^\top \nabla v = 0, \quad \boldsymbol{n}^\top \nabla d = 0. \tag{3.25}$$

## 3.3 Diskretisierung

Für ein iteratives Lösungsverfahren der Euler-Lagrange-Gleichungen (3.22) bis (3.24) erfolgt in einem ersten Schritt eine Diskretisierung der enthaltenen Funktionen. Dazu wird die Bildebene $\Omega$ in ein äquidistantes Gitter von $M \times N$ Punkten diskretisiert, das einen Gitterabstand von $h_x$ in $x$- und $h_y$ in $y$-Richtung hat, wie in Abbildung 3.1 dargestellt. Die Gitterpunkte befinden sich auf den Koordinaten $(i \cdot h_x, j \cdot h_y)$ mit $(i, j) \in [1, M] \times [1, N]$ und können als untere rechte Ecke eines korrespondierenden Pixels betrachtet werden. Auf diesen Gitterpunkten ist der gesuchte Szenenfluss $\boldsymbol{w} = (u, v, d)^\top$ gegeben durch:

$$u_{ij} := u(i \cdot h_x, j \cdot h_y), \tag{3.26}$$

$$v_{ij} := v(i \cdot h_x, j \cdot h_y), \tag{3.27}$$

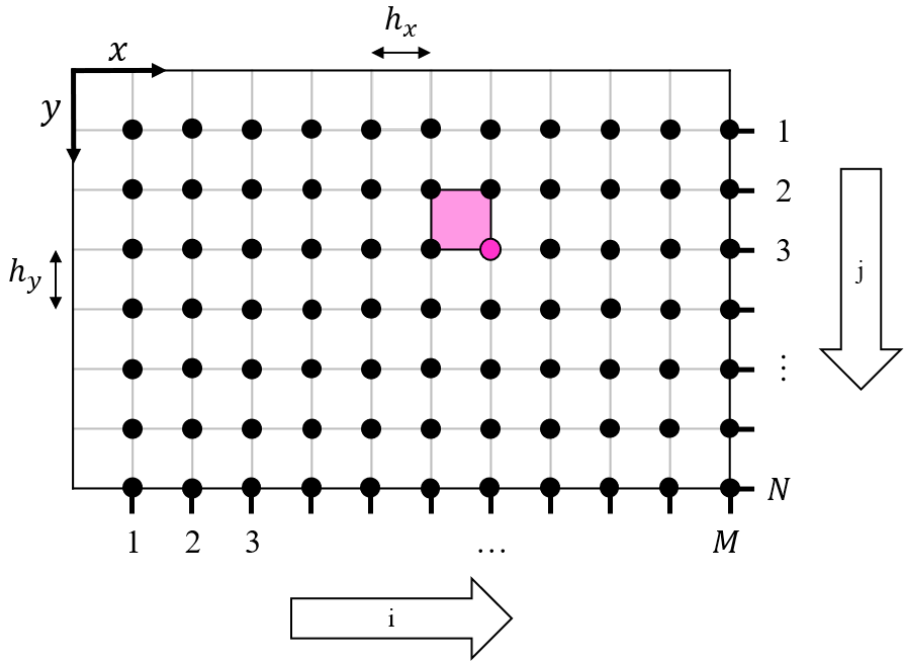



**Abbildung 3.1:** Diskrete Gitterpositionen der Bildebene auf einem äquidistanten Gitter, hier mit $h_x = h_y$. Die Gitterpunkte befinden sich auf den Koordinaten $(i \cdot h_x, j \cdot h_y)$ mit $(i, j) \in [1, M] \times [1, N]$ und können als untere rechte Ecke (hier: rosa Punkt) eines Pixels (hier: rosa Quadrat) betrachtet werden.

$$d_{ij} := d(i \cdot h_x, j \cdot h_y). \tag{3.28}$$

Analog dazu ergibt sich für die diskrete Form von Bild $f$ und Disparitätskarte $\zeta$:

$$f_{ij,t} := f(i \cdot h_x, j \cdot h_y, t), \tag{3.29}$$

$$\zeta_{ij,t} := \zeta(i \cdot h_x, j \cdot h_y, t). \tag{3.30}$$

Die räumlichen Ableitungen von $f$ und $\zeta$ können durch zentrale Differenzen approximiert werden, die zeitliche Ableitung durch eine Vorwärtsdifferenz, bei der ohne Beschränkung der Allgemeinheit die Zeitschrittweite zu $h_t = 1$ festgelegt wird. Da die Berechnung der Approximationen der Ableitungen von $f$ analog zu der von $\zeta$ ist, werden nur die diskretisierten Ableitungen für $f$ aufgezeigt:

$$[f_x]_{ij} := \frac{f_{i+1,j,t} - f_{i-1,j,t}}{2h_x}, \tag{3.31}$$

$$[f_y]_{ij} := \frac{f_{i,j+1,t} - f_{i,j-1,t}}{2h_y}, \tag{3.32}$$

$$[f_t]_{ij} := f_{i,j,t+1} - f_{i,j,t}. \tag{3.33}$$

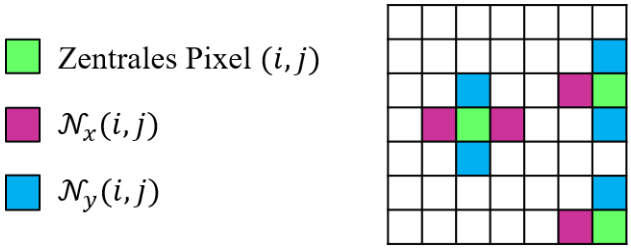



**Abbildung 3.2:** Nachbarschaften $\mathcal{N}_x(i,j)$ und $\mathcal{N}_y(i,j)$ beispielhaft an einer zentralen, Rand- und Eckposition gezeigt.

Mit diesen Definitionen ergibt sich für die Tensoreinträge von **J**:

$$\begin{aligned}[\mathbf{J}]_{ij} &= [\mathbf{J}_{Bild}]_{ij} + \mu \cdot [\mathbf{J}_{Disp}]_{ij} \\ &= \begin{pmatrix} [f_x]_{ij} \\ [f_y]_{ij} \\ 0 \\ [f_t]_{ij} \end{pmatrix} \cdot \begin{pmatrix} [f_x]_{ij} \\ [f_y]_{ij} \\ 0 \\ [f_t]_{ij} \end{pmatrix}^\top + \mu \cdot \begin{pmatrix} [\zeta_x]_{ij} \\ [\zeta_y]_{ij} \\ -1 \\ [\zeta_t]_{ij} + \zeta_{ij} \end{pmatrix} \cdot \begin{pmatrix} [\zeta_x]_{ij} \\ [\zeta_y]_{ij} \\ -1 \\ [\zeta_t]_{ij} + \zeta_{ij} \end{pmatrix}^\top .\end{aligned} \tag{3.34}$$

Zuletzt wird der Laplace-Operator für $u, v$ und $d$ durch zwei verschachtelte zentrale Differenzen diskretisiert, wie hier beispielhaft für $\Delta u$ gezeigt:

$$[\Delta u]_{ij} = \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i}\tilde{j}} - u_{ij}}{h_l^2}. \tag{3.35}$$

Dabei bezeichnet $\mathcal{N}_l(i,j)$ die Nachbarschaft von $i,j$ in Richtung der jeweiligen $x-$ bzw. $y$-Achse, wie in Abbildung 3.2 illustriert. Somit ergeben sich für $\forall (i,j) \in [1,M] \times [1,N]$ die vollständig diskretisierten Euler-Lagrange-Gleichungen:

$$0 = [\mathbf{J}_{11}]_{ij} \cdot u_{ij} + [\mathbf{J}_{12}]_{ij} \cdot v_{ij} + [\mathbf{J}_{13}]_{ij} \cdot d_{ij} + [\mathbf{J}_{14}]_{ij} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i}\tilde{j}} - u_{ij}}{h_l^2}, \tag{3.36}$$

$$0 = [\mathbf{J}_{12}]_{ij} \cdot u_{ij} + [\mathbf{J}_{22}]_{ij} \cdot v_{ij} + [\mathbf{J}_{23}]_{ij} \cdot d_{ij} + [\mathbf{J}_{24}]_{ij} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i}\tilde{j}} - v_{ij}}{h_l^2}, \tag{3.37}$$

$$0 = [\mathbf{J}_{13}]_{ij} \cdot u_{ij} + [\mathbf{J}_{23}]_{ij} \cdot v_{ij} + [\mathbf{J}_{33}]_{ij} \cdot d_{ij} + [\mathbf{J}_{34}]_{ij} - \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{d_{\tilde{i}\tilde{j}} - d_{ij}}{h_l^2}. \tag{3.38}$$

Dieses Gleichungssystem ist in Bezug auf alle $u_{ij}, v_{ij}$ und $d_{ij}$ linear und besteht aus $3 \cdot M \cdot N$ Gleichungen für die gleiche Anzahl an Unbekannten.

Der Laplace-Operator kann alternativ auch durch einen Stencil $S$ definiert werden, welcher die Gewichte einer lokalen Nachbarschaft beschreibt, die für die gewichtete Summe benötigt werden. Im Fall des Laplace-Operators ergibt sich für eine Pixelposition $(i,j)$, für die alle Nachbarn existieren, folgender Stencil:

$$S_{ij} = \begin{array}{|c|c|c|} \hline 0 & \frac{1}{h_y^2} & 0 \\ \hline \frac{1}{h_x^2} & -\frac{2}{x_y^2} - \frac{2}{h_y^2} & \frac{1}{h_x^2} \\ \hline 0 & \frac{1}{h_y^2} & 0 \\ \hline \end{array}. \tag{3.39}$$

Zur Beschreibung der Neumann-Randbedingung, kann eine Indikatorfunktion, wie

$$\chi_{ij} = \begin{cases} 1 & \text{wenn} \quad (i,j) \in [1,M] \times [1,N] \\ 0 & \text{sonst} \end{cases}, \tag{3.40}$$

genutzt werden. Sie ist eine boolesche Funktion und beschreibt, ob sich eine Position $(i,j)$ innerhalb oder außerhalb des Bildbereichs $\Omega$ befindet. Der Stencil $S$ für den Laplace-Operator mit integrierter Indikatorfunktion $\chi$ hat die Form

$$S_{ij} = \begin{array}{|c|c|c|} \hline 0 & \frac{\chi_{i,j+1}}{h_y^2} & 0 \\ \hline \frac{\chi_{i-1,j}}{h_x^2} & -\frac{\chi_{i+1,j}}{h_x^2} - \frac{\chi_{i-1,j}}{h_x^2} - \frac{\chi_{i,j+1}}{h_y^2} - \frac{\chi_{i-1,j}}{h_y^2} & \frac{\chi_{i,j-1}}{h_x^2} \\ \hline 0 & \frac{\chi_{i,j-1}}{h_y^2} & 0 \\ \hline \end{array}. \tag{3.41}$$

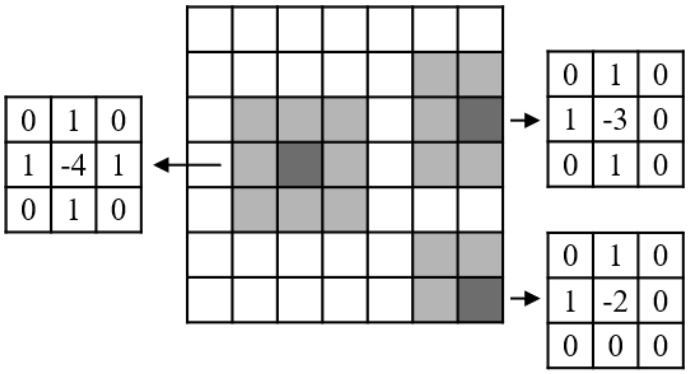



**Abbildung 3.3:** Räumlich abhängiger Stencil für den diskretisierten Laplace-Operator. Beispielwerte für eine zentrale, Rand- und Eckpunktposition für $h_x = h_y = 1$.

In Abbildung 3.3 ist der Stencil $S$ für einen zentralen, Rand- und Eckpunkt visualisiert. Für die Rand- und Eckpunkte werden die fehlenden Nachbarn im Stencil auf Null gesetzt.

Die finalen Euler-Lagrange Gleichungen lauten in Stencil-Schreibweise :

$$0 = [\mathbf{J}_{11}]_{ij} \cdot u_{ij} + [\mathbf{J}_{12}]_{ij} \cdot v_{ij} + [\mathbf{J}_{13}]_{ij} \cdot d_{ij} + [\mathbf{J}_{14}]_{ij} - \alpha \sum_{(\tilde{i},\tilde{j}) \in} \sum_{\{-1,0,1\}} S_{ij,\tilde{i}\tilde{j}} \cdot u_{i+\tilde{i},j+\tilde{j}}, \tag{3.42}$$

$$0 = [\mathbf{J}_{12}]_{ij} \cdot u_{ij} + [\mathbf{J}_{22}]_{ij} \cdot v_{ij} + [\mathbf{J}_{23}]_{ij} \cdot d_{ij} + [\mathbf{J}_{24}]_{ij} - \alpha \sum_{(\tilde{i},\tilde{j}) \in} \sum_{\{-1,0,1\}} S_{ij,\tilde{i}\tilde{j}} \cdot v_{i+\tilde{i},j+\tilde{j}}, \tag{3.43}$$

$$0 = [\mathbf{J}_{13}]_{ij} \cdot u_{ij} + [\mathbf{J}_{23}]_{ij} \cdot v_{ij} + [\mathbf{J}_{33}]_{ij} \cdot d_{ij} + [\mathbf{J}_{34}]_{ij} - \alpha\beta \sum_{(\tilde{i},\tilde{j}) \in} \sum_{\{-1,0,1\}} S_{ij,\tilde{i}\tilde{j}} \cdot d_{i+\tilde{i},j+\tilde{j}}. \tag{3.44}$$

Diese Notation hat den Vorteil, dass Formulierungen komplexerer Glattheitsterme, wie sie später beispielsweise in Kapitel 6 vorkommen, mit komplexeren Stencils in dieser kompakten Schreibweise zusammengefasst werden können. Die Stencil-Schreibweise wird in Abschnitt 6.4 bei der Diskretisierung der weiterführenden Glattheitsterme genutzt.

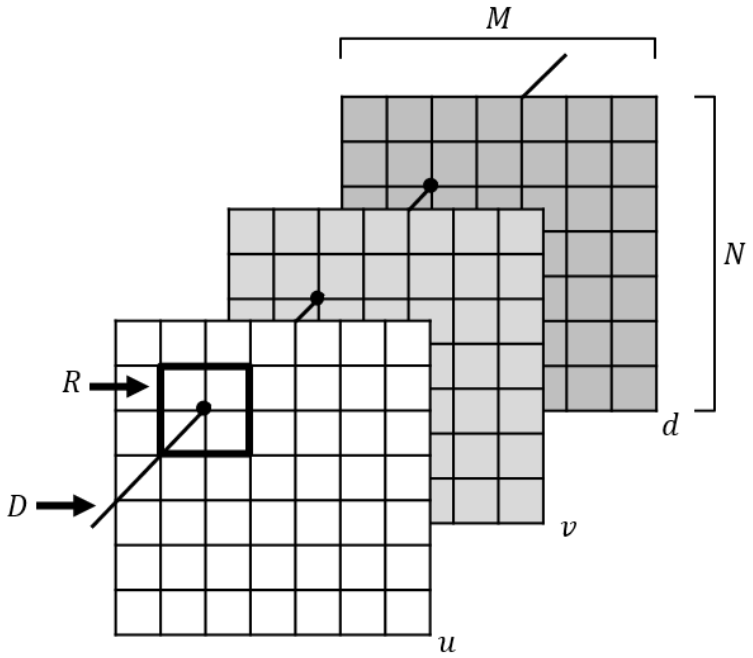



**Abbildung 3.4:** Schematische Visualisierung der Berechnungsbereiche von Datenterm $D$ und Glattheitsterm $R$ im Unbekanntenvolumen von $u_{ij}$, $v_{ij}$ und $d_{ij}$ für die erste Euler-Lagrange-Gleichung (3.36): Der Datentermanteil beschränkt sich auf dieselbe Pixelposition $(i, j)$, die anhand eines Strahls durch das Volumen visualisiert ist. Der Glattheitstermanteil berechnet sich aus der Nachbarschaft derselben Unbekannten, die anhand des Quadrats um das zentrale Pixel gezeigt ist.

Werden die diskretisierten Unbekannten $u_{ij}$, $v_{ij}$ und $d_{ij}$ als Unbekanntenvolumen betrachtet, wie in Abbildung 3.4 dargestellt, so ist der Berechnungsbereich von Daten- und Glattheitsterm in den Euler-Lagrange-Gleichungen erkennbar: Die Berechnung der Datentermanteile geschieht punktweise, das heißt, es werden nur Werte der Unbekannten an derselben Pixelposition $(i, j)$ benötigt. Für die Berechnung der Glattheitstermanteile werden nur die benachbarten Werte derselben Unbekannten benötigt. Die beiden Terme gelten in diesem Unbekanntenvolumen in verschiedenen Dimensionen: Der Datentermanteil beschränkt sich auf dieselbe Pixelposition, der Glattheitstermanteil auf die Nachbarschaft derselben Unbekannten.

## 3.4 Struktur des Gleichungssystems

Das Gleichungssystem, welches sich aus den Euler-Lagrange-Gleichungen (3.36) bis (3.38) ableitet, ist linear und kann daher in Matrixform $A \cdot \mathbf{x} = \mathbf{b}$ notiert werden. Dabei setzt sich der Vektor $\mathbf{x}$ aus den übereinander gereihten Einträgen aller Unbekannten $u_{ij}$, $v_{ij}$ und $d_{ij}$ und der Vektor $\mathbf{b}$ aus

allen Teilen, die nicht mit den Unbekannten multipliziert werden und daher zur rechten Seite der Gleichung gehören, also aus den übereinander gereihten $[J_{14}]_{ij}$, $[J_{24}]_{ij}$ und $[J_{34}]_{ij}$, zusammen:

$$\mathbf{x} = \begin{pmatrix} u_{00} \\ \vdots \\ u_{MN} \\ \hline v_{00} \\ \vdots \\ v_{MN} \\ \hline d_{00} \\ \vdots \\ d_{MN} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} -[J_{14}]_{00} \\ \vdots \\ -[J_{14}]_{MN} \\ \hline -[J_{24}]_{00} \\ \vdots \\ -[J_{24}]_{MN} \\ \hline -[J_{34}]_{00} \\ \vdots \\ -[J_{34}]_{MN} \end{pmatrix}. \tag{3.45}$$

Die Matrix $A$ beinhaltet die Faktoren für alle $u_{ij}$, $v_{ij}$ und $d_{ij}$ aus den Gleichungen (3.36) bis (3.38) und besteht aus neun Blöcken à $(M \cdot N)^2$ Einträgen. Diese Systemmatrix is positiv und semidefinit, denn für Bilder, die natürlich aufgenommen wurden, sind die diagonalen Blöcke positiv definit und $A$ blockdiagonaldominant [76]:

$$A = \left(\begin{array}{c|c|c} \text{diag}([J_{11}]) & \text{diag}([J_{12}]) & \text{diag}([J_{13}]) \\ \hline \text{diag}([J_{12}]) & \text{diag}([J_{22}]) & \text{diag}([J_{23}]) \\ \hline \text{diag}([J_{13}]) & \text{diag}([J_{23}]) & \text{diag}([J_{33}]) \end{array}\right) - \alpha \cdot \left(\begin{array}{c|c|c} \langle S \rangle & 0 & 0 \\ \hline 0 & \langle S \rangle & 0 \\ \hline 0 & 0 & \beta \cdot \langle S \rangle \end{array}\right). \tag{3.46}$$

Der obere linke Block von $A$ verknüpft die Unbekannten $u_{ij}$ zu sich selbst und enthält alle diskretisierten Tensoren $[J_{11}]_{ij}$ auf der Diagonalen:

$$\text{diag}([J_{11}]) = \begin{pmatrix} [J_{11}]_{00} & & 0 \\ & \ddots & \\ 0 & & [J_{11}]_{MN} \end{pmatrix}. \tag{3.47}$$

Der Block $\text{diag}([J_{12}])$ verknüpft einmal alle Unbekannten $u_{ij}$ mit allen $v_{ij}$ und ein weiteres Mal in umgekehrter Richtung. In analoger Art und Weise verknüpfen die restlichen Blöcke die Unbekannten $u_{ij}$, $v_{ij}$ und $d_{ij}$ mit sich selbst sowie miteinander.

Zusätzlich zu allen diskretisierten Tensoren beinhaltet $A$ die entpackte und transformierte Version des Stencils des Glattheitsterms. Der Stencil $S_{ij}$ muss, um in Matrixnotation gebracht zu werden, in den Bildraum transformiert und entpackt werden, hier wie bei Mehl [48], als $\langle S \rangle$ gekennzeichnet. Zur Berechnung wird zunächst ein Gitter der Größe $M \cdot N$ mit Nullen initialisiert und anschließend der Stencil an die Stelle $(i, j)$ positioniert. Daraufhin wird diese zweidimensionale Matrix in einen eindimensionalen Zeilenvektor umgewandelt. Als letzter Schritt werden die Vektoren für alle $(i, j)$ gestapelt und ergeben so den vollständig entpackten Stencil $\langle S \rangle$ der Form

$$\langle S \rangle = \begin{pmatrix} \langle S_{00} \rangle \\ \vdots \\ \langle S_{MN} \rangle \end{pmatrix}. \tag{3.48}$$

Wird die Struktur der Systemmatrix genauer analysiert, kann festgestellt werden, dass $A$ zum Großteil mit Nullen besetzt ist. Dies wird in Abbildung 3.5 an einem Beispielbild der Größe $10 \times 10$ visualisiert, bei der die Nicht-Null-Einträge als weiße und die Null-Einträge als schwarze Pixel

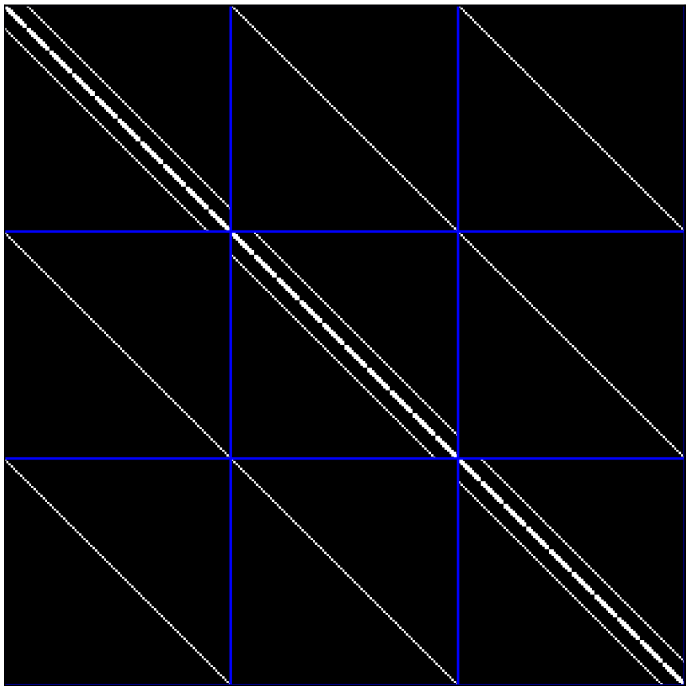


**Abbildung 3.5:** Visualisierung der Systemmatrix $A$ für eine Bildgröße von $10 \times 10$ Pixeln. Potentielle Nicht-Null-Einträge sind weiß, während Null-Einträge schwarz dargestellt sind. Um die Blockstruktur besser zu erkennen, wurden blaue Hilfslinien eingefügt.

dargestellt sind. Zusätzlich wurden blaue Hilfslinien hinzugefügt, um die $3 \times 3$-Blockstruktur der Systemmatrix zu verdeutlichen. Da die Unbekannten, wie zuvor beschrieben, übereinander gereiht sind, ist es möglich anhand der verschiedenen Abschnitte zu erkennen, welche Unbekannten gekoppelt sind. Die Tatsache, dass die Systemmatrix $A$ größtenteils mit Nullen besetzt ist, wird für das weitere Vorgehen, das Lösen des Gleichungssystems mit iterativen Lösungsverfahren, von besonderer Bedeutung sein.

## 3.5 Iterative Lösungsverfahren

Der letzte Schritt in der Kette von Modellierung, Minimierung und Diskretisierung beschäftigt sich mit dem Lösen des diskreten Gleichungssystems. Da dieses mit $3 \cdot M \cdot N$ Gleichungen und ebenso vielen Unbekannten ein sehr großes System darstellt, ist die Gaußsche Eliminierung mit ihrer Zeitkomplexität von $O(n^3)$ nicht praktikabel. Abhilfe verschaffen in solchen Fällen, in denen Systemmatrizen sehr groß und zur Mehrheit mit Nullen besetzt sind, iterative Lösungsverfahren, welche auf der Zerlegung der Systemmatrix $A$ basieren [63]. Die zugrundeliegende Idee ist die Unterteilung von $A$ in die Summe zweier Matrizen $A_1$ und $A_2$, wobei $A_1$ eine angemessene Approximation von $A$ darstellen sollte, deren Inverse einfach und schnell zu berechnen ist [11]. Mithilfe dieser Zerlegung lässt sich durch Umstellen des Gleichungssystems,

$$(A_1 + A_2)\mathbf{x} = \mathbf{b} \iff A_1\mathbf{x} = \mathbf{b} - A_2\mathbf{x}, \tag{3.49}$$

eine Fixpunktiteration der folgenden Form einführen:

$$A_1\mathbf{x}^{k+1} = \mathbf{b} - A_2\mathbf{x}^k \Longleftrightarrow \mathbf{x}^{k+1} = A_1^{-1}(\mathbf{b} - A_2\mathbf{x}^k). \tag{3.50}$$

Häufig wird $A$ in drei Matrizen,

$$A = D - U - O, \tag{3.51}$$

unterteilt, die je nach iterativem Verfahren zu $A_1$ oder $A_2$ zugeordnet werden. Dabei enthält $D$ nur Einträge auf der Diagonalen, $U$ nur Einträge unterhalb und $O$ nur oberhalb der Diagonalen, wie in Abbildung 3.6 illustriert. Der Anteil $D$ wird hier als Diagonalbereich, $U$ als streng unterer und $O$ als streng oberer Dreiecksbereich bezeichnet. Wichtig für die Konvergenz der Verfahren ist, dass $A_1$ nichtsingulär, $A_1^{-1}$ nichtnegativ und $A_2$ nichtnegativ ist [63].

Im Folgenden werden drei häufig genutzte iterative Lösungsverfahren vorgestellt, welche die genannte Zerlegung von $A$ nutzen.

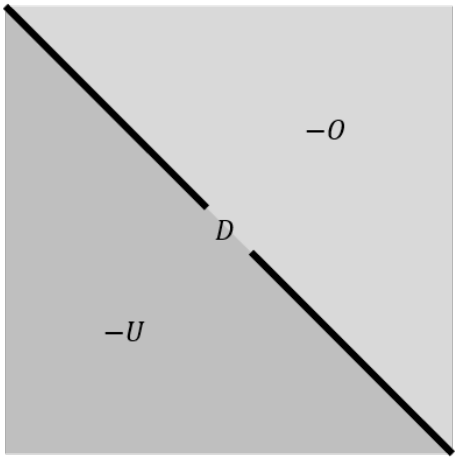



**Abbildung 3.6:** Aufteilung der Systemmatrix $A$ für iterative Lösungsverfahren in $A = D - U - O$, wobei $D$ nur Einträge auf der Diagonalen, $U$ nur Einträge unterhalb und $O$ nur oberhalb der Diagonalen enthalten.

### 3.5.1 Jacobi-Verfahren

Das Jacobi-Verfahren nutzt eine Zerlegung der Matrix $A$ zum einen in Diagonalbereich und zum anderen in streng oberen und unteren Dreiecksbereich:

$$A_1 = D, \quad A_2 = -U - O. \tag{3.52}$$

Nach Saad [63] ergibt sich mit dieser Zerlegung für alle Elemente $x_i$ des Unbekanntenvektors **x** ein Iterationsschritt von $k$ nach $k+1$ der Form:

$$x_i^{k+1} = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x_j^k \right), \tag{3.53}$$

mit der Annahme, die Einträge der Diagonalen $a_{ii}$ seien ungleich Null. Wie von Saad [63] bewiesen, konvergiert der Jacobi-Schritt für einen beliebigen Startwert $\mathbf{x}^0$ zur gewünschten Lösung, falls $A$ eine diagonaldominante Matrix oder eine irreduzibel diagonaldominante Matrix ist. Dies ist bei dem in diesem Kapitel vorgestellten Modell der Fall, wie von Sundaram *et al.* [76] bewiesen.

Nutzt man nun diese Methode zum Lösen der Gleichungen (3.42) bis (3.44), ergeben sich die folgenden Iterationsschritte für $u$, $v$ und $d$:

$$u_{ij}^{k+1} = \frac{-[\mathbf{J}_{14}]_{ij} - \left( [\mathbf{J}_{12}]_{ij} \cdot v_{ij}^k + [\mathbf{J}_{13}]_{ij} \cdot d_{ij}^k - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i}\tilde{j}}^k}{h_l^2} \right)}{[\mathbf{J}_{11}]_{ij} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}, \tag{3.54}$$

$$v_{ij}^{k+1} = \frac{-[\mathbf{J}_{24}]_{ij} - \left( [\mathbf{J}_{12}]_{ij} \cdot u_{ij}^k + [\mathbf{J}_{23}]_{ij} \cdot d_{ij}^k - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i}\tilde{j}}^k}{h_l^2} \right)}{[\mathbf{J}_{22}]_{ij} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}, \tag{3.55}$$

$$d_{ij}^{k+1} = \frac{-[\mathbf{J}_{34}]_{ij} - \left( [\mathbf{J}_{13}]_{ij} \cdot u_{ij}^k + [\mathbf{J}_{23}]_{ij} \cdot v_{ij}^k - \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{d_{\tilde{i}\tilde{j}}^k}{h_l^2} \right)}{[\mathbf{J}_{33}]_{ij} + \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}. \tag{3.56}$$

### 3.5.2 Gauß-Seidel-Verfahren

Das Gauß-Seidel-Verfahren nutzt, im Gegensatz zum Jacobi-Verfahren, eine andere Zerlegung von $A$, wobei sich $A_1$ aus dem Diagonalbereich sowie dem streng unteren Dreiecksbereich und $A_2$ aus dem oberen Dreiecksbereich zusammensetzen:

$$A_1 = D - U, \quad A_2 = -O. \tag{3.57}$$

Die sich ergebende Fixpunktiteration für alle Elemente $x_i$ des Unbekanntenvektors $\mathbf{x}$ lautet dann nach Saad [63]

$$x_i^{k+1} = \frac{1}{a_{ii}} \left( b_i - \sum_{j<i} a_{ij} x_j^{k+1} - \sum_{j>i} a_{ij} x_j^k \right), \tag{3.58}$$

mit der Annahme, die Einträge der Diagonalen $a_{ii}$ seien ungleich Null. Auch für den Gauß-Seidel-Schritt bewies Saad [63] die Konvergenz für einen beliebigen Startwert $\mathbf{x}^0$ zur gewünschten Lösung, falls $A$ eine diagonaldominante oder eine irreduzibel diagonaldominante Matrix ist. Dies ist bei dem in diesem Kapitel vorgestellten Modell der Fall, wie Saad [63] zeigt.

Dieser Iterationsschritt hat den implementierungstechnischen Vorteil, dass die Näherungslösung unmittelbar nach der Bestimmung der neuen Komponente aktualisiert wird [63]. Das bedeutet, dass beim Aktualisieren der Werte zum Schritt $k+1$ die vorherigen Werte des Schritts $k$ nicht

zusätzlich gespeichert werden müssen, wie es beim Jacobi-Schritt der Fall ist. Einerseits wird die Implementierung dadurch einfacher, andererseits ist die asymptotische Konvergenzrate im Vergleich zum Jacobi-Verfahren in etwa doppelt so hoch [63].

Nutzt man nun das Gauß-Seidel-Verfahren zum Lösen der Gleichungen (3.42) bis (3.44), ergeben sich die folgenden Iterationsschritte für $u$, $v$ und $d$:

$$\begin{aligned} u_{ij}^{k+1} =& \frac{-[\mathbf{J}_{14}]_{ij} - \left([\mathbf{J}_{12}]_{ij} \cdot v_{ij}^{k} + [\mathbf{J}_{13}]_{ij} \cdot d_{ij}^{k}\right)}{[\mathbf{J}_{11}]_{ij} + \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}} \\ & - \frac{\left( -\alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^-(i,j)} \frac{u_{\tilde{i}\tilde{j}}^{k+1}}{h_l^2} - \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^+(i,j)} \frac{u_{\tilde{i}\tilde{j}}^{k}}{h_l^2} \right)}{[\mathbf{J}_{11}]_{ij} + \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}, \end{aligned} \tag{3.59}$$

$$\begin{aligned} v_{ij}^{k+1} =& \frac{-[\mathbf{J}_{24}]_{ij} - \left([\mathbf{J}_{12}]_{ij} \cdot u_{ij}^{k+1} + [\mathbf{J}_{23}]_{ij} \cdot d_{ij}^{k}\right)}{[\mathbf{J}_{22}]_{ij} + \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}} \\ & - \frac{\left( -\alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^-(i,j)} \frac{v_{\tilde{i}\tilde{j}}^{k+1}}{h_l^2} - \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^+(i,j)} \frac{v_{\tilde{i}\tilde{j}}^{k}}{h_l^2} \right)}{[\mathbf{J}_{22}]_{ij} + \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}, \end{aligned} \tag{3.60}$$

$$\begin{aligned} d_{ij}^{k+1} =& \frac{-[\mathbf{J}_{24}]_{ij} - \left([\mathbf{J}_{13}]_{ij} \cdot u_{ij}^{k+1} + [\mathbf{J}_{23}]_{ij} \cdot v_{ij}^{k+1}\right)}{[\mathbf{J}_{33}]_{ij} + \alpha\beta \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}} \\ & - \frac{\left( -\alpha\beta \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^-(i,j)} \frac{d_{\tilde{i}\tilde{j}}^{k+1}}{h_l^2} - \alpha\beta \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^+(i,j)} \frac{d_{\tilde{i}\tilde{j}}^{k}}{h_l^2} \right)}{[\mathbf{J}_{33}]_{ij} + \alpha\beta \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}, \end{aligned} \tag{3.61}$$

wobei $\mathcal{N}_l^-(i,j)$ die Menge der Nachbarn von Pixel $i,j$ in Richtung der $x$- bzw. $y$-Achse, die in der aktuellen Iteration bereits berechnet wurden, und $\mathcal{N}_l^+(i,j)$ die Menge der noch nicht berechnet Nachbarpixel ist.

### 3.5.3 *Successive Over-Relaxation* (SOR)

Eine Erweiterung des Gauß-Seidel-Verfahrens stellt die *Successive Over-Relaxation* (SOR) Methode dar, welche 1954 von Young [97] entwickelt wurde. Das Ziel war die Reduzierung der benötigten Iterationsschritte bis zur Konvergenz und so das Beschleunigen des Lösungsverfahrens. Der SOR-Schritt ist eine Linearkombination aus zuletzt berechnetem Unbekanntenvektor $\mathbf{x}^k$ und dem Ergebnis des Gauß-Seidel-Schritts $\mathbf{x}^{GS}$,

$$x_i^{k+1} = (1-\omega) \cdot x_i^k + \omega \cdot x_i^{GS}, \tag{3.62}$$

mit von Saad [63] bewiesener Konvergenz für den Relaxationsparameter $\omega \in (0, 2)$ genau dann, wenn A positiv definit ist. Für einen Wert von $\omega = 1$ reduziert sich der SOR-Schritt zum Gauß-Seidel-Schritt. Von einer Unterrelaxation (*underrelaxation*) wird für einen Wert von $0 < \omega < 1$ gesprochen. Da in diesem Fall die Schrittweite verkleinert wird, kann dies die Stabilität erhöhen und die Glattheitseigenschaften verbessern [11]. Für $1 < \omega < 2$ findet eine punktweise Extrapolation der Ergebnisse des Gauß-Seidel-Verfahrens statt. Diese sogenannte Überrelaxation (*overrelaxation*) kann nach Saad [63] zu einer Beschleunigung der Konvergenz von bis zu einem Faktor zwei führen.

Abbildung 3.7 visualisiert die Iterationsschritte vom Gauß-Seidel-Verfahren und der SOR-Methode für den Fall der Über- und Unterrelaxation. Da es im Allgemeinen nicht möglich ist, den optimalen Wert für $\omega$ im Voraus zu berechnen, muss er entweder empirisch für ein bestimmtes Modellproblem oder heuristisch auf der Grundlage einer Schätzungsmethode bestimmt werden [11].

Da das SOR-Verfahren keine zusätzlichen Stabilitätsbedingungen stellt und eine schnellere Konvergenz bzw. höhere Stabilität als die Verfahren von Jacobi und Gauß-Seidel aufweist, ist es die geeignetere Wahl zum Lösen von Szenenflussproblemen und wird in der vorliegenden Arbeit mit $0 < \omega < 1$ verwendet.

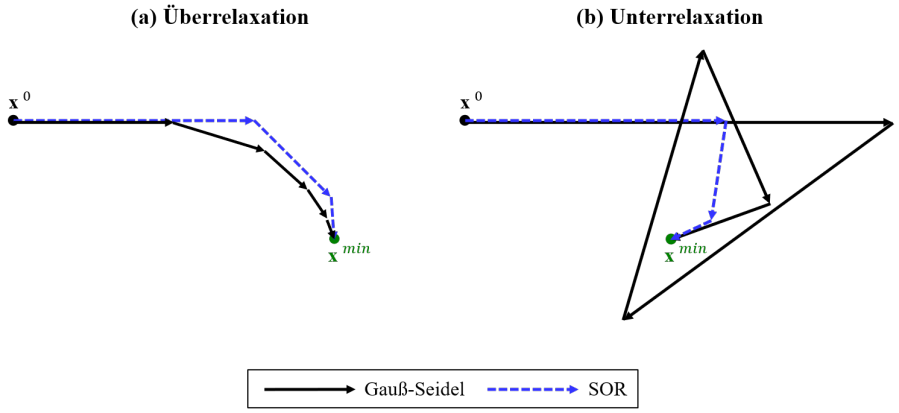



**Abbildung 3.7:** Schematische Darstellung der benötigten Iterationsschritte des Gauß-Seidel-Verfahrens (schwarz) und der SOR-Methode (blau) im zweidimensionalen Raum von einem Ausgangspunkt $\mathbf{x}^0$ bis zur Konvergenz in der Umgebung des Minimums $\mathbf{x}^{min}$ für den Fall (a) einer Überrelaxation, also $1 < \omega < 2$ und (b) einer Unterrelaxation, also $0 < \omega < 1$.

# 4 Verfeinerung der Szenenflussschätzung

Im Rahmen der vorliegenden Arbeit soll mithilfe einer initialen Schätzung des Szenenflusses eine Verfeinerung berechnet werden, welche diese Schätzung verbessert. Dafür werden Initialwerte für $u$, $v$, und $d$ an allen Gitterpunkten des Bildbereichs benötigt, die von einem Stereo- und einem Szenenflussschätzer in einem vorherigen Schritt berechnet werden. Dieser Zusammenhang ist in Abbildung 4.1 aufgezeigt: Zunächst kalkuliert ein Stereoschätzer aus dem linken und rechten Bild für die Zeitpunkte $t$ und $t + 1$ unabhängig voneinander die Disparitätskarten $\zeta$. Diese und die linken Bilder werden dem Szenenflussschätzer weitergereicht, der die initialen Schätzungen für $u$, $v$, und $d$ berechnet. Diese Werte sowie die linken Bilder und die Disparitätskarten werden vom Verfeinerungsverfahren benötigt, um eine Verbesserung von $u$, $v$ und $d$ – nämlich $du$, $dv$ und $dd$ – zu berechnen. Für die finale Evaluation mithilfe des KITTI-Datensatzes [52] (siehe Abschnitt 7.1) wird zusätzlich die Startdisparität $disp_0$ des Referenzframes benötigt. Sie ist über die Disparitätskarte $\zeta$ mit $disp_0(x, y) = \zeta(x, y, t)$ gegeben, da diese Schätzung der Startdisparität weder vom initialen Szenenflussschätzer noch vom Verfeinerungsschritt verändert wird.

Das in Kapitel 3 eingeführte Vorgehen zur Schätzung des Szenenflusses mithilfe von Variationsansätzen – bestehend aus Modellierung der Annahmen als Energiefunktional, Minimierung mithilfe der Euler-Lagrange-Gleichungen, Diskretisierung und iterativen Lösungsverfahren – kann in gleicher Form auch auf die Verfeinerung angewandt werden. Das wird in diesem Kapitel beschrieben. Dabei werden nur die Schritte des Verfahrens erläutert oder erweitert, die sich durch die differentielle Formulierung verändern. Die übrigen Schritte finden sich im vorigen Kapitel und sollen an dieser Stelle nicht erneut aufgeführt werden.

Im Folgenden wird die differentielle Parametrisierung und Modellierung kurz beschrieben. Im Anschluss daran werden die veränderten Euler-Lagrange-Gleichungen sowie die finale Formulierung der Iterationsschritte aufgezeigt.

## 4.1 Differentielle Parametrisierung

Die in dieser Arbeit vorgestellte Verfeinerung soll aus der initialen Schätzung $\boldsymbol{w}_{initial}$ einen verbesserten Szenenfluss $\boldsymbol{w}_{verfeinert}$ berechnen. Gesucht ist also der differentielle Szenenfluss $\boldsymbol{dw}$, der wie folgt definiert wird:

$$\boldsymbol{w}_{verfeinert} = \boldsymbol{w}_{initial} + \boldsymbol{dw}. \tag{4.1}$$

Der initiale Fluss, der über die gesamte Berechnung der Verfeinerung hinweg konstant gehalten wird, ist wie in Abschnitt 2.2.5 mit $(x, y) \in \Omega$ gegeben durch:

$$\boldsymbol{w}_{initial} = \begin{pmatrix} u(x, y) \\ v(x, y) \\ d(x, y) \end{pmatrix} \in \mathbb{R}^3. \tag{4.2}$$

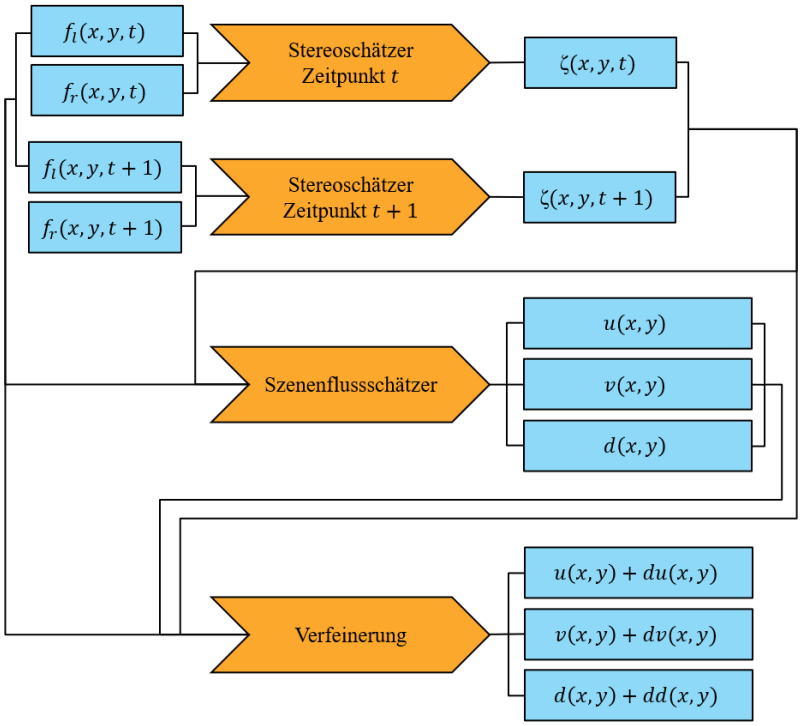



**Abbildung 4.1:** Einordnung der Szenenflussmethode dieser Arbeit in den Gesamtkontext. Zunächst nutzt ein Stereoschätzer die linken und rechten Bilder $f_l$ und $f_r$ zu den Zeitpunkten $t$ und $t+1$ zur Schätzung der Disparitätskarten $\zeta$. Diese dienen mit den linken Bildern einem Szenenflussschätzer als Eingabegrößen, welcher die initialen Szenenflusswerte $u$, $v$ und $d$ berechnet. Auf Basis der linken Bilder, der Disparitätskarten und der initialen Schätzungen berechnet das in dieser Arbeit vorgestellte Verfahren eine Verfeinerung des Szenenflusses $du$, $dv$, $dd$.

Der differentielle Fluss $\boldsymbol{dw}$, der nun die gesuchte Größe darstellt, lautet:

$$\boldsymbol{dw}(x,y) = \begin{pmatrix} du(x,y) \\ dv(x,y) \\ dd(x,y) \end{pmatrix} \in \mathbb{R}^3. \tag{4.3}$$

Die verfeinerten Komponenten des optischen Flusses werden an bestimmten Stellen zur übersichtlicheren Darstellung in der folgenden Notation abgekürzt:

$$\overline{u} := u + du, \quad \overline{v} := v + dv, \quad \overline{d} := d + dd. \tag{4.4}$$

## 4.2 Differentielle Modellierung

Die Modellierung des Szenenflussproblems wird im Verfeinerungsschritt ebenfalls über ein Energiefunktional vorgenommen, das den gleichen Aufbau wie in Gleichung (3.2) hat und aus einem Datenterm $D$ und einen Glattheitsterm $R$ besteht:

$$E(\boldsymbol{dw}) = \int_\Omega \Big( D(\boldsymbol{dw}) + \alpha \cdot R(\boldsymbol{dw}) \Big) \,\mathrm{d}x\,\mathrm{d}y. \tag{4.5}$$

Im Folgenden wird der Aufbau dieser beiden Terme detaillierter aufgezeigt.

### 4.2.1 Datenterm

Der Datenterm beschreibt Konstanzannahmen, die in Bezug auf die Bildfolge $f$ und die Disparitätskarte $\zeta$ getroffen werden. Es gelten auch weiterhin die Grauwertkonstanzannahmen aus den Gleichungen (3.3) und (3.4), die in differentieller Formulierung lauten:

$$f(x+u+du, y+v+dv, t+1) - f(x,y,t) = 0, \tag{4.6}$$

$$\zeta(x+u+du, y+v+dv, t+1) - (d+dd) = 0. \tag{4.7}$$

Die erste Annahme entspricht der Grauwertkonstanz (*brightness constancy assumption* kurz *bca*) von Horn und Schunck [27]. Es wird angenommen, dass die Helligkeit $f(x,y,t)$ eines Oberflächenpunktes zum Zeitpunkt $t$ der Helligkeit $f(x+u+du, y+v+dv, t+1)$ dieses Punktes nach der Bewegung $u+du$ und $v+dv$ zum Zeitpunkt $t+1$ entspricht, es also keine Veränderungen der Beleuchtung gibt. Die zweite Konstanzannahme befasst sich mit der Zieldisparität $d$, die auf das erste Bild zum Zeitpunkt $t$ registriert ist. Es soll gelten, dass $d$ an der Stelle $(x,y)$ der Disparität $\zeta$ zum Zeitpunkt $t+1$ an der Stelle $(x+u+du, y+v+dv)$ entspricht. Die gewichtete Summe dieser beiden Annahmen mit quadratischer Bestrafung von Abweichungen ergibt den Datenterm

$$\begin{aligned} D(du,dv,dd) = &\Big(f(x+u+du, y+v+dv, t+1) - f(x,y,t)\Big)^2 \\ &+ \mu \cdot \Big(\zeta(x+u+du, y+v+dv, t+1) - (d+dd)\Big)^2, \end{aligned} \tag{4.8}$$

mit $\mu \geq 0$ als Parameter, welcher die Gewichtung der Konstanzannahme der Disparität gegenüber der Grauwertkonstanz der Bilder festlegt.

An dieser Stelle wird nun eine Linearisierung von $f$ und $\zeta$ um den Punkt $(x+u, y+v)$ zum Zeitpunkt $t+1$ vorgenommen:

$$\begin{aligned} f(x+u+du, y+v+dv, t+1) \approx &f(x+u, y+v, t+1) \\ &+ f_x(x+u, y+v, t+1) \cdot du \\ &+ f_y(x+u, y+v, t+1) \cdot dv, \end{aligned} \tag{4.9}$$

$$\begin{aligned} \zeta(x+u+du, y+v+dv, t+1) \approx &\zeta(x+u, y+v, t+1) \\ &+ \zeta_x(x+u, y+v, t+1) \cdot du \\ &+ \zeta_y(x+u, y+v, t+1) \cdot dv. \end{aligned} \tag{4.10}$$

Werden diese Linearisierungen in den Datenterm eingesetzt, lautet er in linearisierter Form:

$$\begin{aligned} D(du,dv,dd) = &\Big(f(x+u, y+v, t+1) - f(x,y,t) \\ &+ f_x(x+u, y+v, t+1) \cdot du + f_y(x+u, y+v, t+1) \cdot dv\Big)^2 \\ &+ \mu \cdot \Big(\zeta(x+u, y+v, t+1) - (d+dd) \\ &\quad + \zeta_x(x+u, y+v, t+1) \cdot du \\ &\quad + \zeta_y(x+u, y+v, t+1) \cdot dv\Big)^2. \end{aligned} \tag{4.11}$$

Es kann festgestellt werden, dass $u$, $v$ und $d$ nach der Linearisierung des Datenterms noch immer Teil der Argumente von $f$ und $\zeta$ sind. Um die kompakte Notation in Tensorschreibweise zu ermöglichen, muss ein Berechnungsschritt für $f$ und $\zeta$ an den Positionen $(x+u, y+v, t+1)$ durchgeführt werden, der als Bewegungskompensation um $u$ und $v$ interpretiert werden kann. Dieser Schritt wird *Warping* genannt und soll im Folgenden beschrieben werden.

*Warping* wurde zum ersten Mal von Brox *et al.* [9] in Bezug auf den optischen Fluss angewandt. Es wird im Rahmen der Schätzung von optischem Fluss und Szenenfluss genutzt, wenn differentielle Flüsse berechnet werden müssen. Dies ist beispielsweise bei Grob-zu-fein-Methoden oder Verfeinerungen notwendig und findet daher in vielen Arbeiten Verwendung, wie zum Beispiel bei Slesareva *et al.* [68], Bruhn [11], Sun *et al.* [75], Stone *et al.* [71], Mehl *et al.* [49], und Mehl *et al.* [50]. Das Ergebnis des *Warping*-Schritts ist ein rückwärtsregistriertes Bild $\tilde{f}$ und eine rückwärtsregistrierte Disparitätskarte $\tilde{\zeta}$, für die gelten:

$$\tilde{f}(x,y) = f(x+u, v+y, t+1), \quad \tilde{\zeta}(x,y) = \zeta(x+u, v+y, t+1). \tag{4.12}$$

Das Prinzip dieser Rückwärtsregistrierung zur Bewegungskompensation wird in Abbildung 4.2 visualisiert. Für jede Position $(x, y, t+1) \in \Omega$ wird der Wert von $f$ an der Stelle $(x+u, y+v, t+1)$ durch bilineare Interpolation der benachbarten Pixel berechnet und in $\tilde{f}$ zurück an die Stelle $(x, y, t+1)$ geschrieben. Analog dazu geschieht dies auch für $\zeta$.

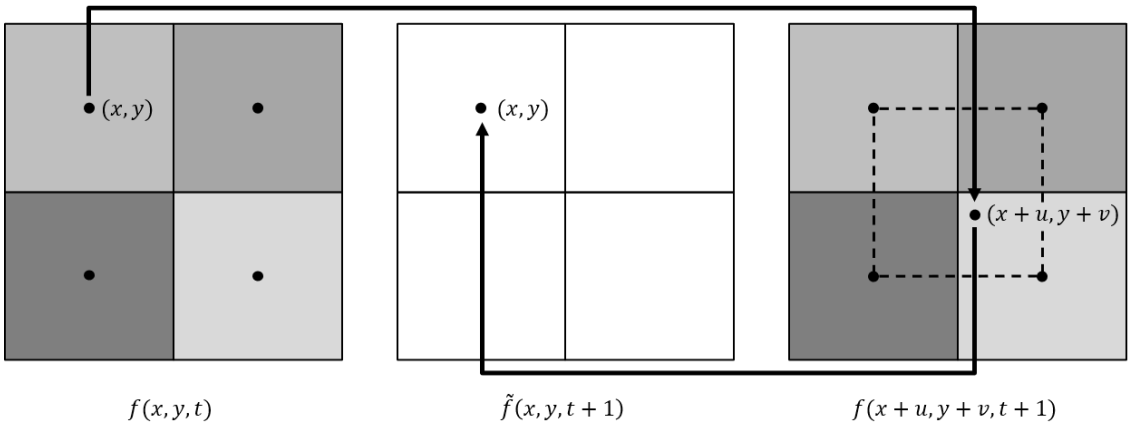



**Abbildung 4.2:** Schematische Darstellung des *Warping*-Mechanismus anhand von $f$: Für jede Position $(x, y)$ wird mithilfe des initialen Szenenflusses die neue Position $(x+u, y+v)$ berechnet, an welcher der Grauwert des Bildes zum Zeitpunkt $t+1$ durch bilineare Interpolation berechnet und in $\tilde{f}$ an die Position $(x, y)$ geschrieben wird.

Mithilfe der rückwärtsregistrierten $\tilde{f}$ und $\tilde{\zeta}$ können die linearen Annahmen zu Tensoren zusammengefasst werden. Aus dem Einzeltensor zur Bildfolge und zur Disparitätskarte,

$$\mathbf{J}_{Bild} = \begin{pmatrix} \tilde{f}_x \\ \tilde{f}_y \\ 0 \\ \tilde{f} - f \end{pmatrix} \cdot \begin{pmatrix} \tilde{f}_x \\ \tilde{f}_y \\ 0 \\ \tilde{f} - f \end{pmatrix}^\top, \quad \mathbf{J}_{Disp} = \begin{pmatrix} \tilde{\zeta}_x \\ \tilde{\zeta}_y \\ -1 \\ \tilde{\zeta} - d \end{pmatrix} \cdot \begin{pmatrix} \tilde{\zeta}_x \\ \tilde{\zeta}_y \\ -1 \\ \tilde{\zeta} - d \end{pmatrix}^\top, \tag{4.13}$$

ergibt sich analog zu Gleichung (3.10) der Gesamttensor $\mathbf{J} = \mathbf{J}_{Bild} + \mu \cdot \mathbf{J}_{Disp}$. Damit lässt sich der linearisierte Datenterm aus Gleichung (4.11) in kompakter Notation formulieren:

$$D(\boldsymbol{dw}) = \boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw}. \tag{4.14}$$

### 4.2.2 Glattheitsterm

Der Glattheitsterm trifft im Gegensatz zum Datenterm keine Annahmen über die Bildfolge $f$ oder Disparitätskarte $\zeta$, sondern über den Szenenfluss selbst. Da eine Glattheit des resultierenden, verfeinerten Flusses, der aus initialem und differentiellem Fluss besteht, erwünscht ist, darf der Regularisierungsterm nicht nur die Gradienten der Komponenten des differentiellen Szenenflusses $\boldsymbol{dw}$ enthalten. Stattdessen müssen sie auf die initialen Schätzungen aufaddiert werden,

$$R(du, dv, dd) = |\nabla(u + du)|^2 + |\nabla(v + dv)|^2 + \beta \cdot |\nabla(d + dd)|^2, \tag{4.15}$$

sodass der gesamte verfeinerte Fluss regularisiert wird.

Das finale Energiefunktional mit Datenterm aus Gleichung (4.14) und Glattheitsterm aus Gleichung (4.15) lautet dann:

$$\begin{aligned} E(\boldsymbol{dw}) = \int_\Omega \Big( & \boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw} \\ & + \alpha \left( |\nabla(u + du)|^2 + |\nabla(v + dv)|^2 + \beta \cdot |\nabla(d + dd)|^2 \right) \Big) \, \mathrm{d}x \, \mathrm{d}y. \end{aligned} \tag{4.16}$$

## 4.3 Euler-Lagrange-Gleichungen

Zur Minimierung des Energiefunktionals aus Gleichung (4.16) werden analog zu den Gleichungen (3.22) bis (3.24) die dazugehörigen Euler-Lagrange-Gleichungen formuliert:

$$0 = \mathbf{J}_{11} \cdot du + \mathbf{J}_{12} \cdot dv + \mathbf{J}_{13} \cdot dd + \mathbf{J}_{14} - \alpha \Delta(u + du), \tag{4.17}$$

$$0 = \mathbf{J}_{12} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{23} \cdot dd + \mathbf{J}_{24} - \alpha \Delta(v + dv), \tag{4.18}$$

$$0 = \mathbf{J}_{13} \cdot du + \mathbf{J}_{23} \cdot dv + \mathbf{J}_{33} \cdot dd + \mathbf{J}_{34} - \alpha\beta \cdot \Delta(d + dd). \tag{4.19}$$

Die Unbekannten des entstandenen Gleichungssystems sind nun nicht mehr $u$, $v$ und $d$, sondern $du$, $dv$, $dd$. Der initiale Fluss mit $u$, $v$ und $d$ kommt nur noch im Laplace-Operator vor.

Werden nun alle Komponenten dieser Gleichungen, wie in Abschnitt 3.3 diskretisiert, wobei die Diskretisierung von $du$, $dv$ und $dd$ analog zu der von $u$, $v$ und $d$ erfolgt, so resultiert folgendes Gleichungssystem $\forall (i, j) \in [1, M] \times [1, N]$:

$$\begin{aligned} 0 = & [\mathbf{J}_{11}]_{ij} \cdot du_{ij} + [\mathbf{J}_{12}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{13}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{14}]_{ij} \\ & - \alpha \sum_{l \in x, y} \sum_{(\tilde{i}, \tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du_{\tilde{i}\tilde{j}} - du_{ij}}{h_l^2} \end{aligned}, \tag{4.20}$$

$$
\begin{aligned}
0 =& [\mathbf{J}_{12}]_{ij} \cdot du_{ij} + [\mathbf{J}_{22}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{24}]_{ij} \\
& - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv_{\tilde{i}\tilde{j}} - dv_{ij}}{h_l^2} \quad , \qquad (4.21) \\
0 =& [\mathbf{J}_{13}]_{ij} \cdot du_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{33}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{34}]_{ij} \\
& - \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd_{\tilde{i}\tilde{j}} - dd_{ij}}{h_l^2} \quad . \qquad (4.22)
\end{aligned}
$$

## 4.4 Iterationsschritte

Abschließend werden die Iterationsschritte von $du$, $dv$ und $dd$, für die in Abschnitt 3.5 vorgestellten iterativen Lösungsverfahren Jacobi, Gauß-Seidel und SOR, aufgestellt.

Für das Jacobi-Verfahren lautet das Iterationsschema:

$$
du_{ij}^{k+1} = \frac{-[\mathbf{J}_{14}]_{ij} - \left( [\mathbf{J}_{12}]_{ij} \cdot dv_{ij}^{k} + [\mathbf{J}_{13}]_{ij} \cdot dd_{ij}^{k} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du_{\tilde{i}\tilde{j}}^{k}}{h_l^2} \right)}{[\mathbf{J}_{11}]_{ij} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}, \qquad (4.23)
$$

$$
dv_{ij}^{k+1} = \frac{-[\mathbf{J}_{24}]_{ij} - \left( [\mathbf{J}_{12}]_{ij} \cdot du_{ij}^{k+1} + [\mathbf{J}_{23}]_{ij} \cdot dd_{ij}^{k} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv_{\tilde{i}\tilde{j}}^{k}}{h_l^2} \right)}{[\mathbf{J}_{22}]_{ij} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}, \qquad (4.24)
$$

$$
dd_{ij}^{k+1} = \frac{-[\mathbf{J}_{34}]_{ij} - \left( [\mathbf{J}_{13}]_{ij} \cdot du_{ij}^{k+1} + [\mathbf{J}_{23}]_{ij} \cdot dv_{ij}^{k+1} - \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd_{\tilde{i}\tilde{j}}^{k}}{h_l^2} \right)}{[\mathbf{J}_{33}]_{ij} + \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}. \qquad (4.25)
$$

Für das Gauß-Seidel-Verfahren gilt:

$$du_{ij}^{k+1} = \frac{-[\mathbf{J}_{14}]_{ij} - \left([\mathbf{J}_{12}]_{ij} \cdot dv_{ij}^{k} + [\mathbf{J}_{13}]_{ij} \cdot dd_{ij}^{k}\right)}{[\mathbf{J}_{11}]_{ij} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}$$

$$- \frac{\left(-\alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^-(i,j)} \frac{du_{\tilde{i}\tilde{j}}^{k+1} + u_{\tilde{i}\tilde{j}} - u_{ij}}{h_l^2} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^+(i,j)} \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du_{\tilde{i}\tilde{j}}^{k}}{h_l^2}\right)}{[\mathbf{J}_{11}]_{ij} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}, \tag{4.26}$$

$$dv_{ij}^{k+1} = \frac{-[\mathbf{J}_{24}]_{ij} - \left([\mathbf{J}_{12}]_{ij} \cdot du_{ij}^{k+1} + [\mathbf{J}_{23}]_{ij} \cdot dd_{ij}^{k}\right)}{[\mathbf{J}_{22}]_{ij} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}$$

$$- \frac{\left(-\alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^-(i,j)} \frac{dv_{\tilde{i}\tilde{j}}^{k+1} + v_{\tilde{i}\tilde{j}} - v_{ij}}{h_l^2} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^+(i,j)} \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv_{\tilde{i}\tilde{j}}^{k}}{h_l^2}\right)}{[\mathbf{J}_{22}]_{ij} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}, \tag{4.27}$$

$$dd_{ij}^{k+1} = \frac{-[\mathbf{J}_{34}]_{ij} - \left([\mathbf{J}_{13}]_{ij} \cdot du_{ij}^{k+1} + [\mathbf{J}_{23}]_{ij} \cdot dv_{ij}^{k+1}\right)}{[\mathbf{J}_{33}]_{ij} + \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}$$

$$- \frac{\left(-\alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^-(i,j)} \frac{dd_{\tilde{i}\tilde{j}}^{k+1} + d_{\tilde{i}\tilde{j}} - d_{ij}}{h_l^2} - \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^+(i,j)} \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd_{\tilde{i}\tilde{j}}^{k}}{h_l^2}\right)}{[\mathbf{J}_{33}]_{ij} + \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}. \tag{4.28}$$

Wie in Abschnitt 3.5.3 beschrieben, lässt sich der SOR-Schritt aus der Linearkombination des zuletzt berechneten Unbekanntenvektors $\mathbf{x}^k$ und dem Ergebnis des Gauß-Seidel-Schritts $\mathbf{x}^{GS}$ ermitteln:

$$x_i^{k+1} = (1 - \omega) \cdot x_i^k + \omega \cdot x_i^{GS}. \tag{3.62 Wdh.}$$

Dieses Kapitel hat gezeigt, wie das variationelle Szenenflussverfahren aus Kapitel 3 als Verfeinerung eingesetzt werden kann und bildet mit der Modellierung aus Gleichung (4.16) eine erste Vorlage für die Verfeinerung, die mit weiterführenden Daten- und Glattheitstermen erweitert werden kann.

# 5 Weiterführende Datenterme

Seit der Veröffentlichung des ersten variationellen Verfahrens zur Berechnung des optischen Flusses von Horn und Schunck [27] im Jahre 1981 hat es viele Neuerungen im Bereich der Variationsansätze, zunächst bezüglich des optischen Flusses und später des Szenenflusses, gegeben. Die in diesem Kapitel präsentierten Ideen stammen zum Großteil aus dem Bereich der variationellen Schätzung des optischen Flusses und werden hier auf die variationelle Verfeinerung des Szenenflusses übertragen.

Das in Kapitel 4 in Gleichung (4.16) eingeführte Modell zur Verfeinerung des Szenenflusses enthält bisher nur grundlegende Datenterme. Bereits im ersten variationellen Verfahren zur Schätzung des Szenenflusses von Huguet und Devernay [28] werden komplexere Konzepte, wie beispielsweise subquadratische Bestrafung und Gradientenkonstanz umgesetzt. Neben diesen Termen werden in dieser Arbeit weitere Datenterme genutzt, die sich als erfolgreich herausgestellt haben und die in den folgenden Abschnitten in differentieller Formulierung eingeführt werden.

## 5.1 Subquadratische Datenterme

Das bisher gezeigte Modell zur variationellen Berechnung vom Szenenfluss kann hinsichtlich der Robustheit gegenüber Ausreißern, verursacht durch Rauschen und Verdeckungen, verbessert werden. Aufgrund der quadratischen Bestrafung weist es einem Ausreißer eine hohe Energie zu (siehe Abbildung 5.1) und führt daher zu einem großen Einfluss des Ausreißers auf den Wert des Energiefunktionals $E(\boldsymbol{dw})$. Deshalb schlugen Black und Anandan [8] vor, auf eine quadratische Bestrafung zu verzichten und stattdessen eine subquadratische Bestrafungsfunktion $\Psi(s^2)$ zu verwenden (siehe Abschnitt 5.1.3), bei der $s^2$ den quadratischen Datenterm bezeichnet. Durch das Überführen des quadratischen Datenterms in einen subquadratischen Datenterm wird der Einfluss von Ausreißern abgeschwächt. Dies wird als Robustifizierung bezeichnet. Besteht der Datenterm aus mehreren Annahmen können diese entweder gemeinsam oder separat robustifiziert werden.

In den folgenden Abschnitten werden nun die verschiedenen Varianten der Robustifizierung des Datenterms vorgestellt, bevor anschließend ein Überblick zu den am häufigsten genutzten subquadratischen Bestrafungsfunktionen gegeben wird. Alle vorgestellten Varianten werden in Kapitel 7 auf die untersuchten Konstanzannahmen angewandt und verglichen.

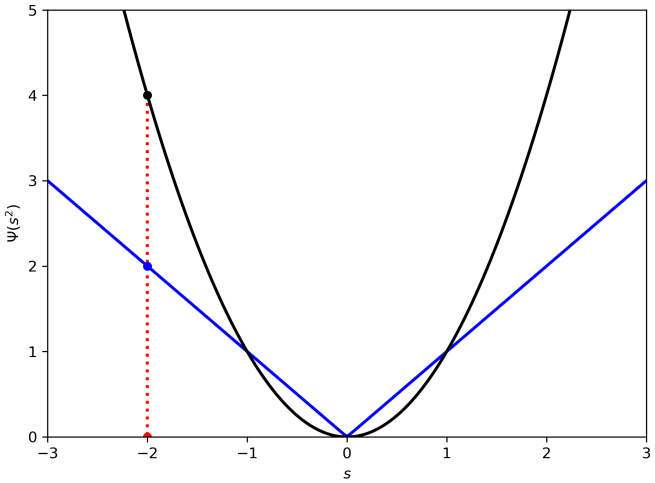



**Abbildung 5.1:** Auswirkung von quadratischer (schwarz) und subquadratischer Bestrafung (blau) auf einen Ausreißer (rot).

### 5.1.1 Gemeinsame Robustifizierung

Eine gemeinsame Robustifizierung, wie bei Brox *et al.* [9], ist bei Annahmen sinnvoll, die korreliert sind, wie beispielsweise RGB-Farbkanäle (siehe Abschnitt 5.3). Ein Datenterm mit $n$ gemeinsam robustifizierten Annahmen und den zugehörigen Gewichten $\lambda_i$ hat die Form:

$$D(\boldsymbol{dw}) = \Psi\left(\sum_{i=1}^{n} \lambda_i \boldsymbol{dw}^\top \mathbf{J}_i \boldsymbol{dw}\right). \tag{5.1}$$

Da die Funktion $\Psi$ auf die Summe der $n$ gewichteten Annahmen angewandt wird, hängen diese linear zusammen, sodass deren Einzeltensoren $\mathbf{J}_i$ zu einem Gesamttensor $\mathbf{J} = \sum_{i=1}^{n} \lambda_i \mathbf{J}_i$ zusammengefasst werden können. Das Energiefunktional aus Gleichung (4.16) lautet demzufolge bei gemeinsam robustifizierten Annahmen wie folgt:

$$\begin{aligned} E(\boldsymbol{dw}) = \int_\Omega \Big( & \Psi\left(\boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw}\right) \\ & + \alpha\left(|\nabla(u+du)|^2 + |\nabla(v+dv)|^2 + \beta \cdot |\nabla(d+dd)|^2\right)\Big) \,\mathrm{d}x\,\mathrm{d}y. \end{aligned} \tag{5.2}$$

Die dazugehörigen Euler-Lagrange-Gleichungen sind durch die gemeinsame Robustifizierung nun nicht mehr linear in $du$, $dv$ und $dd$:

$$0 = \Psi'\left(\boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw}\right)\left(\mathbf{J}_{11} \cdot du + \mathbf{J}_{12} \cdot dv + \mathbf{J}_{13} \cdot dd + \mathbf{J}_{14}\right) - \alpha(\Delta u + \Delta du), \tag{5.3}$$

$$0 = \Psi'\left(\boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw}\right)\left(\mathbf{J}_{12} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{23} \cdot dd + \mathbf{J}_{24}\right) - \alpha(\Delta v + \Delta dv), \tag{5.4}$$

$$0 = \Psi'\left(\boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw}\right)\left(\mathbf{J}_{13} \cdot du + \mathbf{J}_{23} \cdot dv + \mathbf{J}_{33} \cdot dd + \mathbf{J}_{34}\right) - \alpha\beta(\Delta d + \Delta dd), \tag{5.5}$$

mit $\Psi'(s^2) = \frac{\mathrm{d}\Psi}{\mathrm{d}(s^2)}$. Der einzige Unterschied zu den Euler-Lagrange-Gleichungen (4.17) bis (4.19) des nicht robustifizierten Modells ist die Multiplikation mit $\Psi'$. Nach der Diskretisierung ergibt sich das folgende nichtlineare Gleichungssystem:

$$\begin{aligned} 0 =& [\Psi']_{ij} \left( [\mathbf{J}_{11}]_{ij} \cdot du_{ij} + [\mathbf{J}_{12}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{13}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{14}]_{ij} \right) \\ & - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du_{\tilde{i}\tilde{j}} - du_{ij}}{h_l^2}, \end{aligned} \tag{5.6}$$

$$\begin{aligned} 0 =& [\Psi']_{ij} \left( [\mathbf{J}_{12}]_{ij} \cdot du_{ij} + [\mathbf{J}_{22}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{24}]_{ij} \right) \\ & - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv_{\tilde{i}\tilde{j}} - dv_{ij}}{h_l^2}, \end{aligned} \tag{5.7}$$

$$\begin{aligned} 0 =& [\Psi']_{ij} \left( [\mathbf{J}_{13}]_{ij} \cdot du_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{33}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{34}]_{ij} \right) \\ & - \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd_{\tilde{i}\tilde{j}} - dd_{ij}}{h_l^2}, \end{aligned} \tag{5.8}$$

mit der verkürzten Notation $[\Psi']_{ij} = \Psi' \left( [\boldsymbol{dw}]_{ij}^\top [\mathbf{J}]_{ij} [\boldsymbol{dw}]_{ij} \right)$. Um ein solches, nichtlineares Gleichungssystem zu lösen, kann die verzögerte Nichtlinearitätsmethode von Kačur *et al.* [32] angewandt werden. Dabei wird das nichtlineare Gleichungssystem in eine Reihe von linearen Gleichungssystemen umgewandelt, indem eine zweite Fixpunktiteration eingeführt wird, in der die nichtlinearen Anteile $[\Psi']_{ij}$ mit $du$, $dv$ und $dd$ aus dem alten Iterationsschritt $l$ evaluiert werden. Der Rest wird aus dem neuen Iterationsschritt $l+1$ berechnet:

$$\begin{aligned} 0 =& [\Psi']_{ij}^l \left( [\mathbf{J}_{11}]_{ij} \cdot du_{ij}^{l+1} + [\mathbf{J}_{12}]_{ij} \cdot dv_{ij}^{l+1} + [\mathbf{J}_{13}]_{ij} \cdot dd_{ij}^{l+1} + [\mathbf{J}_{14}]_{ij} \right) \\ & - \alpha \sum_{l' \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_{l'}(i,j)} \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du_{\tilde{i}\tilde{j}}^{l+1} - du_{ij}^{l+1}}{h_{l'}^2}, \end{aligned} \tag{5.9}$$

$$\begin{aligned} 0 =& [\Psi']_{ij}^l \left( [\mathbf{J}_{12}]_{ij} \cdot du_{ij}^{l+1} + [\mathbf{J}_{22}]_{ij} \cdot dv_{ij}^{l+1} + [\mathbf{J}_{23}]_{ij} \cdot dd_{ij}^{l+1} + [\mathbf{J}_{24}]_{ij} \right) \\ & - \alpha \sum_{l' \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_{l'}(i,j)} \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv_{\tilde{i}\tilde{j}}^{l+1} - dv_{ij}^{l+1}}{h_{l'}^2}, \end{aligned} \tag{5.10}$$

$$\begin{aligned} 0 =& [\Psi']_{ij}^l \left( [\mathbf{J}_{13}]_{ij} \cdot du_{ij}^{l+1} + [\mathbf{J}_{23}]_{ij} \cdot dv_{ij}^{l+1} + [\mathbf{J}_{33}]_{ij} \cdot dd_{ij}^{l+1} + [\mathbf{J}_{34}]_{ij} \right) \\ & - \alpha\beta \sum_{l' \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_{l'}(i,j)} \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd_{\tilde{i}\tilde{j}}^{l+1} - dd_{ij}^{l+1}}{h_{l'}^2}. \end{aligned} \tag{5.11}$$

### 5.1.2 Separate Robustifizierung

Eine separate Robustifizierung, wie bei Bruhn und Weickert [10] und Zimmer *et al.* [101, 102], ist bei Annahmen sinnvoll, die unabhängig voneinander erfüllt werden können, wie beispielsweise die Gradientenkonstanz (siehe Abschnitt 5.2). Ein Datenterm mit $n$ separat robustifizierten Annahmen hat die Form:

$$D(\boldsymbol{dw}) = \sum_{i=1}^{n} \lambda_i \Psi \left( \boldsymbol{dw}^\top \mathbf{J}_i \boldsymbol{dw} \right). \tag{5.12}$$

Da die subquadratische Bestrafungsfunktion zuerst auf die Konstanzannahmen angewandt wird und diese erst danach aufsummiert werden, hängen sie nicht linear zusammen und können nicht zu einem Gesamttensor zusammengefasst werden. Demzufolge lautet das gesamte Energiefunktional mit separat robustifiziertem Datenterm:

$$\begin{aligned} E(\boldsymbol{dw}) = \int_\Omega \Big( \sum_{i=1}^{n} \lambda_i \Psi \left( \boldsymbol{dw}^\top \mathbf{J}_i \boldsymbol{dw} \right) \\ + \alpha \left( |\nabla(u+du)|^2 + |\nabla(v+dv)|^2 + \beta \cdot |\nabla(d+dd)|^2 \right) \Big) \, \mathrm{d}x \, \mathrm{d}y. \end{aligned} \tag{5.13}$$

Die dazugehörigen Euler-Lagrange-Gleichungen sind durch die separate Robustifizierung ebenfalls nichtlinear in $du$, $dv$ und $dd$,

$$\begin{aligned} 0 = &\sum_{i=1}^{n} \lambda_i \Psi \left( \boldsymbol{dw}^\top \mathbf{J}_i \boldsymbol{dw} \right) \left( \mathbf{J}_{i,11} \cdot du + \mathbf{J}_{i,12} \cdot dv + \mathbf{J}_{i,13} \cdot dd + \mathbf{J}_{i,14} \right) \\ &- \alpha(\Delta u + \Delta du), \end{aligned} \tag{5.14}$$

$$\begin{aligned} 0 = &\sum_{i=1}^{n} \lambda_i \Psi \left( \boldsymbol{dw}^\top \mathbf{J}_i \boldsymbol{dw} \right) \left( \mathbf{J}_{i,12} \cdot du + \mathbf{J}_{i,22} \cdot dv + \mathbf{J}_{i,23} \cdot dd + \mathbf{J}_{i,24} \right) \\ &- \alpha(\Delta v + \Delta dv), \end{aligned} \tag{5.15}$$

$$\begin{aligned} 0 = &\sum_{i=1}^{n} \lambda_i \Psi \left( \boldsymbol{dw}^\top \mathbf{J}_i \boldsymbol{dw} \right) \left( \mathbf{J}_{i,13} \cdot du + \mathbf{J}_{i,23} \cdot dv + \mathbf{J}_{i,33} \cdot dd + \mathbf{J}_{i,34} \right) \\ &- \alpha\beta(\Delta d + \Delta dd), \end{aligned} \tag{5.16}$$

und ergeben nach Diskretisierung das Gleichungssystem mit der verkürzten Notation $[\Psi'_{i'}]_{ij} = \lambda_{i'} \Psi \left( [\boldsymbol{dw}]_{ij}^\top [\mathbf{J}_{i'}]_{ij} [\boldsymbol{dw}]_{ij} \right)$:

$$\begin{aligned} 0 = &\sum_{i'=1}^{n} [\Psi'_{i'}]_{ij} \left( [\mathbf{J}_{i',11}]_{ij} \cdot du_{ij} + [\mathbf{J}_{i',12}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{i',13}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{i',14}]_{ij} \right) \\ &- \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du_{\tilde{i}\tilde{j}} - du_{ij}}{h_l^2}, \end{aligned} \tag{5.17}$$

$$\begin{aligned} 0 = &\sum_{i'=1}^{n} [\Psi'_{i'}]_{ij} \left( [\mathbf{J}_{i',12}]_{ij} \cdot du_{ij} + [\mathbf{J}_{i',22}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{i',23}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{i',24}]_{ij} \right) \\ &- \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv_{\tilde{i}\tilde{j}} - dv_{ij}}{h_l^2}, \end{aligned} \tag{5.18}$$

$$\begin{aligned} 0 = &\sum_{i'=1}^{n} [\Psi'_{i'}]_{ij} \left( [\mathbf{J}_{i',13}]_{ij} \cdot du_{ij} + [\mathbf{J}_{i',23}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{i',33}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{i',34}]_{ij} \right) \\ &- \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd_{\tilde{i}\tilde{j}} - dd_{ij}}{h_l^2}. \end{aligned} \tag{5.19}$$

Da dieses Gleichungssystem nichtlinear ist, wird auch hier die verzögerte Nichtlinearitätsmethode von Kačur *et al.* [32] angewandt. Analog zu Abschnitt 5.1.1 ergibt sich:

$$\begin{aligned} 0 = &\sum_{i'=1}^{n} [\Psi'_{i'}]^l_{ij} \left( [\mathbf{J}_{i',11}]_{ij} \cdot du^{l+1}_{ij} + [\mathbf{J}_{i',12}]_{ij} \cdot dv^{l+1}_{ij} + [\mathbf{J}_{i',13}]_{ij} \cdot dd^{l+1}_{ij} + [\mathbf{J}_{i',14}]_{ij} \right) \\ &- \alpha \sum_{l' \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_{l'}(i,j)} \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du^{l+1}_{\tilde{i}\tilde{j}} - du^{l+1}_{ij}}{h^2_{l'}}, \end{aligned} \tag{5.20}$$

$$\begin{aligned} 0 = &\sum_{i'=1}^{n} [\Psi'_{i'}]^l_{ij} \left( [\mathbf{J}_{i',12}]_{ij} \cdot du^{l+1}_{ij} + [\mathbf{J}_{i',22}]_{ij} \cdot dv^{l+1}_{ij} + [\mathbf{J}_{i',23}]_{ij} \cdot dd^{l+1}_{ij} + [\mathbf{J}_{i',24}]_{ij} \right) \\ &- \alpha \sum_{l' \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_{l'}(i,j)} \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv^{l+1}_{\tilde{i}\tilde{j}} - dv^{l+1}_{ij}}{h^2_{l'}}, \end{aligned} \tag{5.21}$$

$$\begin{aligned} 0 = &\sum_{i'=1}^{n} [\Psi'_{i'}]^l_{ij} \left( [\mathbf{J}_{i',13}]_{ij} \cdot du^{l+1}_{ij} + [\mathbf{J}_{i',23}]_{ij} \cdot dv^{l+1}_{ij} + [\mathbf{J}_{i',33}]_{ij} \cdot dd^{l+1}_{ij} + [\mathbf{J}_{i',34}]_{ij} \right) \\ &- \alpha \sum_{l' \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_{l'}(i,j)} \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd^{l+1}_{\tilde{i}\tilde{j}} - dd^{l+1}_{ij}}{h^2_{l'}}. \end{aligned} \tag{5.22}$$

### 5.1.3 Bestrafungsfunktionen

Um einen quadratisch bestraften Term linear zu machen, wird eine subquadratische Bestrafungsfunktion $\Psi(s^2)$ benötigt. Im Allgemeinen sollte $\Psi(s^2)$ laut Zimmer *et al.* [101] positiv, stetig wachsend, subquadratisch und streng konvex sein.

Dieser Abschnitt gibt einen Überblick über die drei am häufigsten verwendeten Bestrafungsfunktionen $\Psi(s^2)$ und deren Ableitungen $\Psi'(s^2)$, da diese Bestandteil der Euler-Lagrange-Gleichungen sind. Sie werden in Abbildung 5.2 dargestellt.

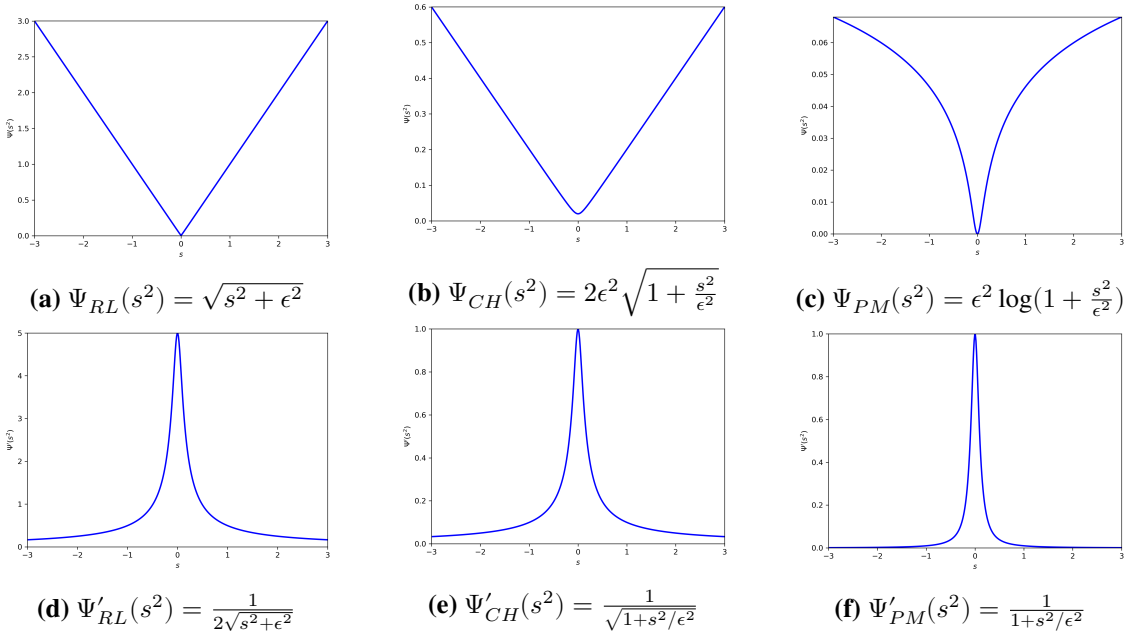
#### Regularisierte Lineare Bestrafung

Eine einfache mathematische Operation, um eine quadratische Funktion linear zu machen, ist die Anwendung der Quadratwurzel. Da die Funktion differenzierbar bleiben soll, wird eine regularisierte Version mit einem kleinen $\epsilon > 0$ genutzt (siehe Brox *et al.* [9]):

$$\Psi_{RL}(s^2) = \sqrt{s^2 + \epsilon^2}. \tag{5.23}$$

Die Ableitung der regularisiert linearen Bestrafung lautet:

$$\Psi'_{RL}(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon^2}}. \tag{5.24}$$



**(a)** $\Psi_{RL}(s^2) = \sqrt{s^2 + \epsilon^2}$ **(b)** $\Psi_{CH}(s^2) = 2\epsilon^2\sqrt{1 + \frac{s^2}{\epsilon^2}}$ **(c)** $\Psi_{PM}(s^2) = \epsilon^2 \log(1 + \frac{s^2}{\epsilon^2})$

**(d)** $\Psi'_{RL}(s^2) = \frac{1}{2\sqrt{s^2+\epsilon^2}}$ **(e)** $\Psi'_{CH}(s^2) = \frac{1}{\sqrt{1+s^2/\epsilon^2}}$ **(f)** $\Psi'_{PM}(s^2) = \frac{1}{1+s^2/\epsilon^2}$

**Abbildung 5.2:** Übersicht über die subquadratischen Bestrafungsfunktionen. Obere Reihe: Bestrafungsfunktionen; untere Reihe: die jeweilige Ableitung. Von links nach rechts: regularisiert linear, Charbonnier, Perona-Malik mit $\epsilon = 0.1$.

### Charbonnier-Bestrafung

Die von Charbonnier *et al.* [13] vorgeschlagene Bestrafung hat die Form:

$$\Psi_{CH}(s^2) = 2\epsilon^2\sqrt{1 + \frac{s^2}{\epsilon^2}}, \tag{5.25}$$

mit

$$\Psi'_{CH}(s^2) = \frac{1}{\sqrt{1 + s^2/\epsilon^2}}. \tag{5.26}$$

Sie hat gegenüber der regularisierten linearen Bestrafung den Vorteil, dass ihre Ableitung auf eins begrenzt ist. Dies ist bei der Festlegung eines Stabilitätskriteriums hilfreich. Daher wird die Charbonnier-Bestrafung im Allgemeinen gegenüber der regularisierten linearen Bestrafung bevorzugt eingesetzt. Daher wird sie in der vorliegenden Arbeit verwendet.

### Perona-Malik-Bestrafung

Abschließend wird als Beispiel für eine sublineare Bestrafungsfunktion die von Perona und Malik [57] vorgeschlagene Bestrafung und ihre Ableitung,

$$\Psi_{PM}(s^2) = \epsilon^2 \log(1 + \frac{s^2}{\epsilon^2}), \tag{5.27}$$

$$\Psi'_{PM}(s^2) = \frac{1}{1 + s^2/\epsilon^2}, \tag{5.28}$$

vorgestellt. Wie Maurer [43] zeigt ist diese sublineare Betrafung nicht-konvex.

## 5.2 Gradientenkonstanzannahme

Die bisher angenommene Grauwertkonstanz im Datenterm, siehe Gleichung (4.6), hat einen großen Nachteil: Sie ist anfällig für leichte Helligkeitsschwankungen, die in natürlichen Szenen häufig auftreten. Daher ist es sinnvoll, den optischen Flussanteil des Szenenflusses mithilfe eines Kriteriums zu bestimmen, das kleine Veränderungen des Grauwerts zulässt [9]. Ein solches Kriterium ist der Gradient des Grauwerts des Bildes, welcher invariant bei globalen additiven Beleuchtungsänderungen ist und von dem angenommen werden kann, dass er sich durch die Verschiebung $u$ und $v$ nicht verändert:

$$\nabla f(x, y, t) = \nabla f(x + u + du, y + v + dv, t + 1). \tag{5.29}$$

Die Idee einer Gradientenkonstanz (*gradient constancy assumption* kurz *gca*) wurde von Uras *et al.* [80] und Tistarelli [79] vorgeschlagen und im Rahmen von Variationsansätzen das erste Mal von Brox *et al.* [9] eingesetzt. Seither findet es in vielen weiteren Arbeiten Verwendung, wie beispielsweise bei Bruhn und Weickert [10], Sundaram *et al.* [76], Maurer [44], Nüssle [55] und Mehl *et al.* [49].

Die beiden Annahmen zur Gradientenkonstanz der Bilder (in $x$- und $y$-Richtung) können analog zu Gleichung (4.9) linearisiert werden und ergeben mit dem rückwärtsregistrierten Bild $\tilde{f}$ aus Gleichung (4.12) analog zu Brox *et al.* [9] die beiden Gleichungen:

$$\tilde{f}_x - f_x + \tilde{f}_{xx} du + \tilde{f}_{xy} dv = 0, \tag{5.30}$$

$$\tilde{f}_y - f_y + \tilde{f}_{yx} du + \tilde{f}_{yy} dv = 0. \tag{5.31}$$

Da diese Annahmen linear in $du$, $dv$ und $dd$ sind, können sie mithilfe der nachstehenden Tensornotation zusammengefasst werden:

$$\mathbf{J}_{gca,Bild} = \begin{pmatrix} \tilde{f}_{xx} \\ \tilde{f}_{xy} \\ 0 \\ \tilde{f}_x - f_x \end{pmatrix} \cdot \begin{pmatrix} \tilde{f}_{xx} \\ \tilde{f}_{xy} \\ 0 \\ \tilde{f}_x - f_x \end{pmatrix}^\top + \begin{pmatrix} \tilde{f}_{yx} \\ \tilde{f}_{yy} \\ 0 \\ \tilde{f}_y - f_y \end{pmatrix} \cdot \begin{pmatrix} \tilde{f}_{yx} \\ \tilde{f}_{yy} \\ 0 \\ \tilde{f}_y - f_y \end{pmatrix}^\top . \tag{5.32}$$

Für natürliche Bilder ist die Gradientenkonstanz robuster als die Grauwertkonstanz. Deshalb wird im Folgenden untersucht, ob diese Grundidee auch auf die Zieldisparität angewandt werden kann. Die Gradientenkonstanz der Zieldisparität lautet:

$$\nabla(d + dd) = \nabla\zeta(x + u + du, y + v + dv, t + 1). \tag{5.33}$$

Dafür werden die beiden Annahmen zur Gradientenkonstanz der Disparität (in $x$- und $y$-Richtung), analog zu Gleichung (4.10), ebenfalls linearisiert und ergeben mit dem rückwärtsregistrierten $\tilde{\zeta}$ aus Gleichung (4.12):

$$\tilde{\zeta}_x - (d_x + dd_x) + \tilde{\zeta}_{xx} du + \tilde{\zeta}_{xy} dv = 0, \tag{5.34}$$

$$\tilde{\zeta}_y - (d_y + dd_y) + \tilde{\zeta}_{yx} du + \tilde{\zeta}_{yy} dv = 0. \tag{5.35}$$

Diese Annahmen für die Zieldisparität enthalten, im Gegensatz zu denen vom Bild aus Gleichungen (5.30) und (5.31), zusätzlich zu den Variablen $du$ und $dv$ auch die partiellen Ableitungen von $dd$. Dies macht die Berechnung vom Datenterm nachbarschaftsbezogen, da aufgrund der Diskretisierung der partiellen Ableitungen $dd_x$ und $dd_y$ mithilfe finiter Differenzen die Nachbarschaft,

ähnlich wie beim Glattheitsterm, betrachtet werden muss. Dies führt dazu, dass die dazugehörigen Euler-Lagrange-Gleichungen nicht mehr mit dem in Abschnitt 3.3 aufgestellten Berechnungsschema lösbar sind. Die Gradientenkonstanz der Zieldisparität übersteigt damit den für diese Arbeit vorgesehenen Rahmen. Deshalb wird sich hier auf die Gradientenkonstanz vom Bild beschränkt.

Obwohl die Gradientenkonstanz der Bilder den Vorteil der Beleuchtungsinvarianz hat, gibt es gegenüber der Grauwertkonstanz auch Nachteile: Durch die Ableitungsbildung anhand von finiter Differenzen wird die Empfindlichkeit bei Rauschen erhöht und der Preis für die Beleuchtungsinvarianz ist Informationsverlust. Außerdem ist die Gradientenkonstanzannahme nicht für alle Fälle geeignet, beispielsweise bei räumlich variierenden additiven Veränderungen, die nicht konstant über die gesamte Bildebene sind, oder wenn Bewegungen Drehungen enthalten, da eine implizite Konstanthaltung der Richtung aufgrund der Gradienten stattfindet. Da beide Annahmen Vor- und Nachteile haben, kann es sinnvoll sein, einen Datenterm aufzustellen, welcher sowohl die Grauwert- als auch die Gradientenkonstanz umfasst. Dafür wird der Gesamttensor,

$$\mathbf{J} = \mathbf{J}_{bca,Bild} + \mu \cdot \mathbf{J}_{bca,Disp} + \gamma \cdot \mathbf{J}_{gca,Bild}, \tag{5.36}$$

aus den einzelnen Tensoren gebildet, wobei $\mathbf{J}_{bca,Bild}$ und $\mathbf{J}_{bca,Disp}$ den Tensoren aus Gleichung (4.13) entsprechen und $\mu \geq 0$ die Grauwertkonstanz der Zieldisparität und $\gamma \geq 0$ die Gradientenkonstanz der Bilder gewichtet. Mithilfe dieser Gewichte kann eine Balance zwischen den Annahmen gefunden werden. Der Datenterm kann mit $\gamma = 0$ aber auch zur reinen Grauwertkonstanz zurückgeführt werden. Die Tensornotation hat hier den Vorteil, dass die Euler-Lagrange-Gleichungen sowie die Fixpunktiterationen aus Kapitel 4 unverändert bleiben. Häufig werden Gradienten- und Grauwertkonstanz separat robustifiziert, wie beispielsweise bei Bruhn und Weickert [10]. In diesem Fall kann der Gesamttensor nicht im Vorhinein gebildet werden und es gilt analog zu Gleichung (5.13):

$$\begin{aligned} D(du, dv, dd) =& \Psi(\boldsymbol{dw}^\top \mathbf{J}_{bca,Bild} \boldsymbol{dw}) \\ &+ \mu \cdot \Psi(\boldsymbol{dw}^\top \mathbf{J}_{bca,Disp} \boldsymbol{dw}) \\ &+ \gamma \cdot \Psi(\boldsymbol{dw}^\top \mathbf{J}_{gca,Bild} \boldsymbol{dw}). \end{aligned} \tag{5.37}$$

Die Nutzung der hier vorgestellten Gradientenkonstanz wird in Kapitel 7 mit und ohne Hinzunahme der Grauwertkonstanz vergleichend evaluiert.

## 5.3 Farbkanäle

Wie bei Golland und Bruckstein [21] gezeigt, kann es von Vorteil sein, die Grauwertkonstanz auf die einzelnen Farbkanäle eines Bildes zu übertragen. Dafür wird das Grauwertbild ersetzt durch:

$$\Big(f^1(x,y,t), f^2(x,y,t), f^3(x,y,t)\Big) = \Big(f^R(x,y,t), f^G(x,y,t), f^B(x,y,t)\Big). \tag{5.38}$$

Die Farbkanäle $f^R$, $f^G$ und $f^B$ stehen dabei für den Rot-, Grün- und Blauanteil der jeweiligen Pixel an der Position $(x, y)$ zum Zeitpunkt $t$. Nach Golland und Bruckstein [21] gilt die Grauwertkonstanz nun bezüglich jedes einzelnen Farbkanals $f^i$:

$$f^i(x + u + du, y + v + dv, t + 1) - f^i(x, y, t) = 0, \quad i \in \{1, 2, 3\}. \tag{5.39}$$

Dies ist die verallgemeinerte Form der Grauwertkonstanz.

Der dazugehörige Datenterm wird als Summe der Annahmen geschrieben und vereint damit alle drei Farbkanäle:

$$\begin{aligned} D(du,dv,dd) = &\sum_{i=1}^{3} \Big( f^i(x+u+du, y+v+dv, t+1) - f^i(x,y,t) \Big)^2 \\ &+ \mu \cdot \Big( \zeta(x+u+du, y+v+dv, t+1) - (d+dd) \Big)^2 . \end{aligned} \tag{5.40}$$

Die Linearisierung der Annahmen erfolgt analog zu Gleichung (4.9). Dies ermöglicht eine kompakte Tensornotation der Form

$$\mathbf{J} = \sum_{i=1}^{3} \mathbf{J}^i_{bca,Bild} \quad , \tag{5.41}$$

mit

$$\mathbf{J}^i_{bca,Bild} = \begin{pmatrix} \tilde{f}^i_x \\ \tilde{f}^i_y \\ 0 \\ \tilde{f}^i - f^i \end{pmatrix} \cdot \begin{pmatrix} \tilde{f}^i_x \\ \tilde{f}^i_y \\ 0 \\ \tilde{f}^i - f^i \end{pmatrix}^\top . \tag{5.42}$$

Gemäß Gleichung (4.12) werden je Farbkanal $i$ das rückwärtsregistrierte Bild $\tilde{f}^i = \tilde{f}^i(x,y) = f^i(x+u, y+v, t+1)$ berechnet. Die partiellen Ableitungen der Farbkanäle werden analog zu denen des Grauwertbildes in Gleichungen (3.31) und (3.32) gebildet.

Nach van de Weijer und Gevers [83] sollten realistische Beleuchtungsmodelle den Einfluss multiplikativer Beleuchtungsänderungen umfassen, die allerdings weder von der Grauwert- noch von der Gradientenkonstanz erfasst werden. Golland und Bruckstein [21] schlagen zur Lösung dieses Problems den Wechsel vom RGB-Farbraum in den HSV-Farbraum vor:

$$(f^1(x,y,t), f^2(x,y,t), f^3(x,y,t)) = (f^H(x,y,t), f^S(x,y,t), f^V(x,y,t)). \tag{5.43}$$

Der Farbtonkanal H (*hue*) ist invariant bei multiplikativen Beleuchtungsänderungen, insbesondere Schatten, Schattierungen, Highlights oder spekularen Reflexen. Der Sättigungskanal S (*saturation*) ist nur in Bezug auf Schatten und Schattierungen invariant. Der Wertkanal V (*value*) weist keine dieser Invarianten auf [102]. Mileva *et al.* [53] nutzten nur den Farbtonkanal für die Berechnung des optischen Flusses. Zimmer *et al.* [101, 102] verwendeten zusätzlich den Sättigungs- und Wertkanal. Ihr Datenterm ist außerdem normalisiert und enthält Grauwert- und Gradientenkonstanz der Farbkanäle unter Anwendung einer separaten Robustifizierung zur Gewichtung der Konstanzannahmen während der Schätzung.

Werden diese Ideen auf den differentiellen Szenenfluss angewandt, ergibt sich folgender Datenterm:

$$\begin{aligned} D(du,dv,dd) = &\sum_{i=1}^{3} \Psi \Big( \theta^i \Big( f^i(x+u+du, y+v+dv, t+1) - f^i(x,y,t) \Big)^2 \Big) \\ &+ \mu \cdot \Psi \Big( \Big( \zeta(x+u+du, y+v+dv, t+1) - (d+dd) \Big)^2 \Big) \\ &+ \gamma \sum_{i=1}^{3} \Psi \left( \begin{pmatrix} \theta^i_x \\ \theta^i_y \end{pmatrix}^\top \begin{pmatrix} (f^i_x(x+u+du, y+v+dv, t+1) - f^i_x(x,y,t))^2 \\ (f^i_y(x+u+du, y+v+dv, t+1) - f^i_y(x,y,t))^2 \end{pmatrix} \right) , \end{aligned} \tag{5.44}$$

wobei $\mu$ die Grauwertkonstanz der Disparität und $\gamma$ die Gradientenkonstanz der Bildfolge gewichtet. Die Normalisierung nach Zimmer *et al.* [102],

$$\theta^i = \frac{1}{|\Delta f^i|^2 + \epsilon^2}, \quad \theta^i_x = \frac{1}{|\Delta f^i_x|^2 + \epsilon^2}, \quad \theta^i_y = \frac{1}{|\Delta f^i_y|^2 + \epsilon^2}, \tag{5.45}$$

ist an dieser Stelle nötig, da eine implizite Gewichtung mit dem quadrierten räumlichen Bildgradienten stattfindet. Dies führt zu einer stärkeren Durchsetzung des Datenterms an Orten mit hohem Gradienten. Eine solche Übergewichtung kann unerwünscht sein, da große Gradienten durch unzuverlässige Strukturen, wie Rauschen oder Verdeckungen, verursacht werden können. Die Normalisierung wirkt dem entgegen.

Werden die einzelnen Annahmen in Gleichung (5.44) analog zu 4.9 und 4.10 linearisiert, können sie zu Tensoren zusammengefasst werden, in denen die Normalisierung eingebettet wird:

$$\overline{\mathbf{J}}^i_{bca,Bild} = \theta^i \mathbf{J}^i_{bca,Bild}, \quad \overline{\mathbf{J}}^i_{gca,Bild} = \theta^i \mathbf{J}^i_{gca,Bild}. \tag{5.46}$$

Eingesetzt in die Formulierung des separat robustifizierten Datenterms aus Gleichung (5.12) ergeben sie mit $J_{bca,Disp}$ aus Gleichung (4.13):

$$\begin{aligned} D(du, dv, dd) = &\sum_{i=1}^{3} \Psi \left( \boldsymbol{dw}^\top \overline{\mathbf{J}}^i_{bca,Bild} \boldsymbol{dw} \right) \\ &+ \mu \cdot \Psi \left( \boldsymbol{dw}^\top \mathbf{J}_{bca,Disp} \boldsymbol{dw} \right) \\ &+ \gamma \cdot \sum_{i=1}^{3} \Psi \left( \boldsymbol{dw}^\top \overline{\mathbf{J}}^i_{gca,Bild} \boldsymbol{dw} \right). \end{aligned} \tag{5.47}$$

In Kapitel 7 wird der Nutzen von Grauwert- und Gradientenkonstanz sowohl von HSV-Farbbildern als auch von Grauwertbildern bewertet.

# 6 Weiterführende Glattheitsterme

Nachdem in Kapitel 5 Erweiterungen des Datenterms $D(\boldsymbol{dw})$ vorgestellt wurden, die eine Berücksichtigung komplexer Annahmen ermöglichen, werden nun Erweiterungen für den Glattheitsterm $R(\boldsymbol{dw})$ vorgestellt.

Wie bei Zimmer *et al.* [102] ausgeführt, besteht eine enge Beziehung zwischen Glattheitsterm und Diffusionsprozessen: Während der Glattheitsterm ein glattes Bewegungsfeld als gewünschten Zustand modelliert, beschreiben die entsprechenden Euler-Lagrange-Gleichungen den dazugehörigen Glättungsprozess, der als Diffusionsprozess betrachtet werden kann. Damit können die etablierten Konzepte bei der Beschreibung von Diffusionsprozessen auf den Szenenfluss übertragen werden. Die bisherige Beschreibung des Glattheitsterms nach Gleichung (4.15) beinhaltet nur eine homogene Glattheit, da der Laplace-Operator der daraus resultierenden Euler-Lagrange-Gleichungen, siehe Gleichungen (4.17) bis (4.17), einen homogenen Diffusionsprozess darstellt, der in alle Richtungen gleichermaßen glättet. Dies kann dazu führen, dass beispielsweise Kanten im Ergebnisfluss nicht ausreichend scharf abgebildet werden. Um diese Kanten zu bewahren, können nach Huguet und Devernay [28] sowohl bildgetriebene als auch flussgetriebene Glattheitsterme (siehe Abschnitte 6.2 und 6.3) verwendet werden.

Die zwei Glattheitskategorien – bildgetrieben und flussgetrieben – lassen sich jeweils in isotrop und anisotrop unterteilen. Isotrope Glattheit orientiert sich an der betragsmäßigen Größe der Kanten und erlaubt eine schwächere Glättung an Positionen mit bedeutenden Kanten. Anisotrope Glattheit untersucht im Gegensatz dazu die Richtung der Kanten und macht eine schwächere Glättung orthogonal zu ihnen möglich. Die Arbeit von Weickert und Schnörr [92] stellt diese vier Varianten des Glattheitsterms für den optischen Fluss dar und beweist die *well-posedness* der jeweiligen mathematischen Beschreibung. Da der Szenenfluss aufgrund der zusätzlichen Zieldisparität einen komplexeren Fall darstellt, können diese vier Glattheitstypen weiter untergliedert werden.

In Abschnitt 6.1 wird diese Untergliederung erläutert und in diesem Zusammenhang der Begriff der treibenden Domäne eingeführt. Abschnitt 6.2 befasst sich dann mit der isotropen und anisotropen bildgetriebenen Glattheit, während Abschnitt 6.3 die isotrope und anisotrope flussgetriebene Glattheit beschreibt. Abschließend wird in Abschnitt 6.4 eine allgemeine Formulierung der Diskretisierung der weiterführenden Glattheitsterme vorgestellt.

## 6.1 Treibende Domäne

Wie bereits erwähnt, führt der Glattheitsterm über die Euler-Lagrange-Gleichungen zu einem Glättungsprozess. Dies hat in der bisherigen Modellierung den negativen Effekt, dass Kanten im Szenenfluss homogen geglättet werden. Ziel ist es daher, diese Kantenglättung zu reduzieren.

Hierfür müssen über geeignete mathematische Operationen zunächst die Stärken von Kanten bestimmt werden, um dann den Diffusionsprozess des Szenenflusses an den entsprechenden Stellen abzuschwächen.

Dazu wird nun der Begriff der ‚treibenden Domäne' eingeführt: Die treibende Domäne gibt die Größe an, in der die Stärke von Kanten bestimmt wird. Vom bildgetriebenen Fall wird gesprochen, wenn die Bestimmung der Kantenstärke im Bild oder der Disparitätskarte erfolgt, während im flussgetriebenen Fall die Bestimmung in den Komponenten des Szenenflusses selbst stattfindet. Folglich sind im bildgetriebenen Fall die treibenden Domänen $f$, $\zeta$ oder die Kombination von $f$ und $\zeta$, im flussgetriebenen Fall $(\overline{u}, \overline{v})$, $\overline{d}$ oder die Kombination von $(\overline{u}, \overline{v})$ und $\overline{d}$. Wird beispielsweise $f$ als treibende Domäne gewählt, so wird an Positionen mit starken Kanten von $f$ abgeschwächt geglättet.

Auf Basis der treibenden Domänen können nun die in den Euler-Lagrange-Gleichungen formulierten Diffusionsprozesse für die einzelnen Komponenten des Szenenflusses modifiziert werden, um die Diffusion an den Kantenpositionen der treibenden Domänen abzuschwächen. Die Abschwächung der Diffusion kann dabei für jede Komponente des Szenenflusses $(\overline{u}, \overline{v}, \overline{d})^\top$ unabhängig modelliert werden. Für $\overline{u}$ und $\overline{v}$ wird in dieser Arbeit eine gemeinsame Modellierung gewählt, da ihre Kanten im Allgemeinen korreliert sind.

Werden für die Diffusionsprozesse von optischem Fluss und Zieldisparität unterschiedliche treibende Domänen gewählt, so wird von separaten treibenden Domänen gesprochen. Werden die gleichen verwendet, so werden sie als kombinierte treibende Domänen bezeichnet. Beide Fälle werden in der vorliegenden Arbeit behandelt.

## 6.2 Bildgetriebene Glattheit

Da sich Objekte meist starr im Ganzen bewegen und so ihre Kanten mit den Bewegungskanten übereinstimmen, sind die Szenenflusskanten im Allgemeinen laut Nagel und Enkelmann [54] eine Teilmenge der Bildkanten. Dies macht sich die bildgetriebene Glattheit zu Nutze und schwächt die Glättung von $u+du$, $v+dv$ und $d+dd$ im isotropen Fall bei betragsmäßig großen Kanten des Bildes $f$ und im anisotropen Fall orthogonal zu diesen, sodass scharfe Szenenflusskanten an Positionen solcher Kanten entstehen können. Beim Glattheitstyp ‚bildgetrieben' kann im Szenenfluss sowohl $f$ als auch $\zeta$ als treibende Domäne auftreten.

### 6.2.1 Bildgetriebene isotrope Glattheit

Die bildgetriebene isotrope Glattheit nach Nagel und Enkelmann [54] bzw. Álvarez León *et al.* [2] verringert die Glättung an den Bildkanten, die durch große Werte von $|\nabla f^2|$ erkennbar sind. Das gleiche Prinzip kann auf die Disparitätskarte angewandt werden, deren Kanten durch große Werte von $|\nabla\zeta|^2$ gefunden werden. Nach Zimmer *et al.* [100] hat die Disparitätskarte in der Regel weniger Kanten als das Bild $f$. Diese Disparitätskanten entsprechen überwiegend den tatsächlichen Szenenflusskanten. Daher kann es von Vorteil sein $\zeta$ als treibende Domäne zu nutzen. Allerdings kann eine verrauschte Disparität zu schlechten Ergebnissen führen, weswegen es sinnvoll sein kann, Disparitätskarte und Bild als treibende Domänen zu kombinieren, wie bei Rabe *et al.* [60] ausgeführt wird.

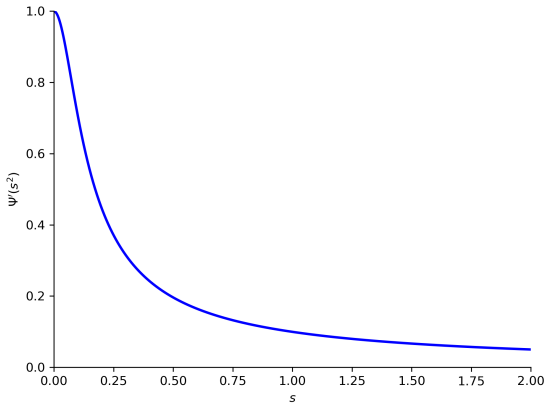



**Abbildung 6.1:** Darstellung der in dieser Arbeit verwendeten Übertragungsfunktion für die bildgetriebene isotrope Glattheit $g(s^2) = \frac{1}{\sqrt{1+s^2/\epsilon^2}}$.

Ganz allgemein soll an Kanten von Bild oder Disparitätskarte, also an Stellen, an denen der Gradient $|\nabla f|^2$ bzw. $|\nabla \zeta|^2$ groß ist, abgeschwächt geglättet werden. Dafür wird eine Übertragunsfunktion $g(s^2)$ benötigt, welche großen Eingabewerten kleine Gewichte zuordnet und umgekehrt. Daher ist $g(s^2)$ positiv und monoton fallend. Die in dieser Arbeit verwendete Übertragungsfunktion ist in Abbildung 6.1 gezeigt. Sie entspricht der Ableitung der Charbonnier-Bestrafung aus Gleichung (5.26).

Der Glattheitsterm für die bildgetriebene isotrope Glattheit mit separaten treibenden Domänen hat somit die nachstehende Form:

$$\begin{aligned} R(du, dv, dd) =& g(|\nabla f|^2) \left(|\nabla(u + du)|^2 + |\nabla(v + dv)|^2\right) \\ &+ \beta \cdot g(|\nabla \zeta|^2) |\nabla(d + dd)|^2. \end{aligned} \tag{6.1}$$

Dabei wird $g(|\nabla f|^2)$ als gemeinsamer Faktor für die Glattheit von $(u+du)$ und $(v+dv)$ genutzt, da sie in den meisten Fällen identische Kanten enthalten. Für $g(|\nabla f|^2) = 1$ und $g(|\nabla \zeta|^2) = 1$ lässt sich Gleichung (6.1) auf den bekannten homogenen Glattheitsterm aus Gleichung (4.15) zurückführen. In einer allgemeineren Formulierung des Glattheitsterms wird $g(|\nabla f|^2)$ durch $g^{uv}$ und $g(|\nabla \zeta|^2)$ durch $g^d$ersetzt:

$$\begin{aligned} R(du, dv, dd) =& g^{uv} \cdot \left(|\nabla(u + du)|^2 + |\nabla(v + dv)|^2\right) \\ &+ \beta \cdot g^d \cdot |\nabla(d + dd)|^2. \end{aligned} \tag{6.2}$$

Der Ausdruck $g^{uv}$ wird auch als Diffusivität des optischen Flusses bezeichnet und $g^d$ als Diffusivität der Zieldisparität. Als treibende Domäne für $g^{uv}$ und $g^d$ können dabei jeweils das Bild $f$, die Disparitätskarte $\zeta$ oder ihre Kombination gewählt werden. Die möglichen Kombinationen sind in Tabelle 6.1 aufgelistet. Für $g^{uv}$ und $g^d$ kann dabei jeweils unabhängig voneinander einer der Ausdrücke für die Diffusivität $g$ verwendet werden. Die hier vorgestellen Varianten der bildgetriebenen isotropen Glattheit werden in Abschnitt 7.6 verglichen und evaluiert.

| treibende Domäne | keine | $f$ | $\zeta$ | $f$ und $\zeta$ |
|---|---|---|---|---|
| Diffusivität $g$ | 1 | $g(\|\nabla f\|^2)$ | $g(\|\nabla \zeta\|^2)$ | $g(\|\nabla f\|^2 + \|\nabla \zeta\|^2)$ |

**Tabelle 6.1:** Diffusivität $g$ für die verschiedenen treibenden Domänen bei bildgetriebener isotroper Glattheit. Die zweite Spalte ohne treibende Domäne beschreibt den homogenen Fall.

Mit dem Glattheitsterm aus Gleichung (6.2) ergibt sich das Energiefunktional mit bildgetriebener isotroper Glattheit:

$$\begin{aligned} E(du, dv, dd) = \int_\Omega \Big( & \boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw} \\ & + \alpha \cdot g^{uv} \left( |\nabla(u + du)|^2 + |\nabla(v + dv)|^2 \right) \\ & + \alpha\beta \cdot g^d |\nabla(d + dd)|^2 \Big) \, \mathrm{d}x \, \mathrm{d}y. \end{aligned} \tag{6.3}$$

Daraus leiten sich die folgenden Euler-Lagrange-Gleichungen ab:

$$0 = \mathbf{J}_{11} \cdot du + \mathbf{J}_{12} \cdot dv + \mathbf{J}_{13} \cdot dd + \mathbf{J}_{14} - \alpha \cdot \operatorname{div}\left( g^{uv} \cdot \nabla(u + du) \right), \tag{6.4}$$

$$0 = \mathbf{J}_{12} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{23} \cdot dd + \mathbf{J}_{24} - \alpha \cdot \operatorname{div}\left( g^{uv} \cdot \nabla(v + dv) \right), \tag{6.5}$$

$$0 = \mathbf{J}_{13} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{33} \cdot dd + \mathbf{J}_{34} - \alpha\beta \cdot \operatorname{div}\left( g^d \cdot \nabla(d + dd) \right). \tag{6.6}$$

Für eine Diskretisierung der Euler-Lagrange-Gleichungen ist nun zusätzlich die Diskretisierung der Diffusivitäten $g^{uv}$ und $g^d$ notwendig – hier beispielhaft für die Diffusivitäten aus Gleichung (6.1) gezeigt:

$$g^{uv}_{ij} = g(|\nabla f|^2_{ij}) = g([f_x]^2_{ij} + [f_y]^2_{ij}), \tag{6.7}$$

$$g^{d}_{ij} = g(|\nabla \zeta|^2_{ij}) = g([\zeta_x]^2_{ij} + [\zeta_y]^2_{ij}). \tag{6.8}$$

Demzufolge lauten die diskretisierten Euler-Lagrange-Gleichungen:

$$\begin{aligned} 0 = & [\mathbf{J}_{11}]_{ij} \cdot du_{ij} + [\mathbf{J}_{12}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{13}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{14}]_{ij} \\ & - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{g^{uv}_{\tilde{i}\tilde{j}} + g^{uv}_{ij}}{2} \left( \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du_{\tilde{i}\tilde{j}} - du_{ij}}{h_l^2} \right), \end{aligned} \tag{6.9}$$

$$\begin{aligned} 0 = & [\mathbf{J}_{12}]_{ij} \cdot du_{ij} + [\mathbf{J}_{22}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{24}]_{ij} \\ & - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{g^{uv}_{\tilde{i}\tilde{j}} + g^{uv}_{ij}}{2} \left( \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv_{\tilde{i}\tilde{j}} - dv_{ij}}{h_l^2} \right), \end{aligned} \tag{6.10}$$

$$\begin{aligned} 0 = & [\mathbf{J}_{13}]_{ij} \cdot du_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{33}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{34}]_{ij} \\ & - \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{g^{d}_{\tilde{i}\tilde{j}} + g^{d}_{ij}}{2} \left( \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd_{\tilde{i}\tilde{j}} - dd_{ij}}{h_l^2} \right), \end{aligned} \tag{6.11}$$

mit den Neumann-Randbedingungen

$$\boldsymbol{n}^\top g^{uv} I \nabla(u + du) = 0, \quad \boldsymbol{n}^\top g^{uv} I \nabla(v + dv) = 0, \quad \boldsymbol{n}^\top g^{d} I \nabla(d + dd) = 0, \tag{6.12}$$

und der Identitätsmatrix $I = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. Da $g^{uv}$ und $g^d$ unabhängig von $du$, $dv$ und $dd$ sind, werden sie als Konstanten behandelt und können für jede Pixelposition $(i, j) \in [1, M] \times [1, N]$ vor der Schleife zur iterativen Berechnung von $du$, $dv$ und $dd$ berechnet werden.

| treibende Domäne | keine | $f$ | $\zeta$ | $f$ und $\zeta$ |
|---|---|---|---|---|
| Diffusionstensor $\mathbf{D}$ | $I$ | $D_{img}(\nabla f)$ | $D_{img}(\nabla \zeta)$ | $D_{img}(\nabla f + \nabla \zeta)$ |

**Tabelle 6.2:** Diffusionstensor $\mathbf{D}$ für die verschiedenen treibenden Domänen der bildgetriebenen anisotropen Glattheit. Die zweite Spalte ohne treibende Domäne beschreibt den homogenen Fall.

### 6.2.2 Bildgetriebene anisotrope Glattheit

Die bildgetriebene anisotrope Glattheit wurde von Nagel und Enkelmann [54] eingeführt. Die Glättung wird dabei orthogonal zu den Bildkanten reduziert, während die Glättung entlang der Bildkanten gefördert wird. Im Gegensatz zum isotropen Fall spielt bei der anisotropen Glattheit nicht nur die Lage sondern auch die Richtung der Kante eine entscheidende Rolle. Ein Glattheitsterm, der diese Richtungsabhängigkeit implementiert und analog zum Modell von Nagel und Enkelmann [54] aufgebaut ist, hat die Form:

$$\begin{aligned} R(du, dv, dd) =& \nabla(u + du)^\top \mathbf{D}^{uv} \nabla(u + du) + \nabla(v + dv)^\top \mathbf{D}^{uv} \nabla(v + dv) \\ &+ \beta \cdot \nabla(d + dd)^\top \mathbf{D}^{d} \nabla(d + dd). \end{aligned} \tag{6.13}$$

Als treibende Domänen für die Diffusionstensoren $\mathbf{D}^{uv}$ und $\mathbf{D}^{d}$ können dabei jeweils das Bild $f$, die Disparitätskarte $\zeta$ oder ihre Kombination gewählt werden. Die möglichen Kombinationen sind in Tabelle 6.2 aufgelistet. Für $\mathbf{D}^{uv}$ und $\mathbf{D}^{d}$ kann dabei jeweils unabhängig voneinander einer der Ausdrücke für den Diffusionstensor $\mathbf{D}$ verwendet werden. Dabei wird $\mathbf{D}^{uv}$ wieder als gemeinsamer Faktor für die Glattheit von $(u + du)$ und $(v + dv)$ genutzt, da sie in den meisten Fällen identische Kanten enthalten. Die Verwendung eines Diffusionstensors $\mathbf{D}$ anstelle einer skalarwertigen Diffusivität ermöglicht ein richtungsabhängiges Glättungsverhalten.

Die in dieser Arbeit verwendete Berechnungsvorschrift $D_{img}$ für die bildgetriebenen anisotropen Diffusionstensoren $\mathbf{D}^{uv}$ und $\mathbf{D}^{d}$ ist hier beispielhaft für die treibende Domäne $f$ gezeigt:

$$\mathbf{D} = D_{img}(\nabla f) = \frac{1}{|\nabla f|^2 + 2\epsilon^2} \begin{pmatrix} f_y^2 & -f_x f_y \\ -f_x f_y & f_x^2 \end{pmatrix} + \epsilon^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \epsilon > 0. \tag{6.14}$$

Der Diffusionstensor $\mathbf{D}$ stellt eine Projektionsmatrix entlang der Kante, also orthogonal zum Gradienten, dar. Nach Weickert und Schnörr [92] sind die Eigenvektoren von $\mathbf{D}$ durch $\mathbf{v}_1 = \nabla f$ und $\mathbf{v}_2 = \nabla f^\perp$ gegeben. Die dazugehörigen Eigenwerte $\lambda$ lauten:

$$\lambda_1(|\nabla f|) = \frac{\epsilon^2}{|\nabla f| + 2\epsilon^2}, \tag{6.15}$$

$$\lambda_2(|\nabla f|) = \frac{|\nabla f| + \epsilon^2}{|\nabla f| + 2\epsilon^2}. \tag{6.16}$$

Innerhalb von Objekten mit $|\nabla f| \to 0$, also in Regionen ohne Kanten, gilt $\lambda_1 \to \frac{1}{2}$ und $\lambda_2 \to \frac{1}{2}$. Dies entspricht isotroper Glättung, die gleichmäßig in beide Richtungen vorgenommen wird. An idealen Kanten mit $|\nabla f| \to \infty$ gilt $\lambda_1 \to 0$ und $\lambda_2 \to 1$, sodass Abweichungen von der Glattheit entlang der Kanten nicht bestraft, sondern erhalten werden. Das gleiche Prinzip gilt für die Disparitätskarte $\zeta$ als treibende Domäne.

| Isotropie | Treibende Domänen | | |
|---|---|---|---|
| | $f$ | $\zeta$ | $f$ und $\zeta$ |
| homogen | $I$ | $I$ | $I$ |
| isotrop | $g(\lvert\nabla f\rvert^2)I$ | $g(\lvert\nabla\zeta\rvert^2)I$ | $g(\lvert\nabla f\rvert^2+\lvert\nabla\zeta\rvert^2)I$ |
| anisotrop | $D_{img}(\nabla f)$ | $D_{img}(\nabla\zeta)$ | $D_{img}(\nabla f+\nabla\zeta)$ |

**Tabelle 6.3:** Diffusionstensor $\mathbf{D}$ für die möglichen Kombinationen aus treibender Domäne ($f$ und/oder $\zeta$) und Isotropie (homogen, isotrop oder anisotrop) der bildgetriebenen Glattheit.

Mit dem Glattheitsterm aus Gleichung (6.13) lautet das Energiefunktional mit bildgetriebener anisotroper Glattheit:

$$\begin{aligned} E(du,dv,dd) = \int_\Omega \Big( & \boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw} \\ & + \alpha \cdot \Big( \nabla(u+du)^\top \mathbf{D}^{uv} \nabla(u+du) + \nabla(v+dv)^\top \mathbf{D}^{uv} \nabla(v+dv) \Big) \\ & + \alpha\beta \cdot \nabla(d+dd)^\top \mathbf{D}^{d} \nabla(d+dd) \Big) \,\mathrm{d}x\,\mathrm{d}y. \end{aligned} \tag{6.17}$$

Daraus leiten sich die folgenden Euler-Lagrange-Gleichungen ab:

$$0 = \mathbf{J}_{11}\cdot du + \mathbf{J}_{12}\cdot dv + \mathbf{J}_{13}\cdot dd + \mathbf{J}_{14} - \alpha\cdot \operatorname{div}\Big(\mathbf{D}^{uv}\cdot\nabla(u+du)\Big), \tag{6.18}$$

$$0 = \mathbf{J}_{12}\cdot du + \mathbf{J}_{22}\cdot dv + \mathbf{J}_{23}\cdot dd + \mathbf{J}_{24} - \alpha\cdot \operatorname{div}\Big(\mathbf{D}^{uv}\cdot\nabla(v+dv)\Big), \tag{6.19}$$

$$0 = \mathbf{J}_{13}\cdot du + \mathbf{J}_{22}\cdot dv + \mathbf{J}_{33}\cdot dd + \mathbf{J}_{34} - \alpha\beta\cdot \operatorname{div}\Big(\mathbf{D}^{d}\cdot\nabla(d+dd)\Big), \tag{6.20}$$

mit den Neumann-Randbedingungen

$$\boldsymbol{n}^\top \mathbf{D}^{uv}\nabla(u+du) = 0, \quad \boldsymbol{n}^\top \mathbf{D}^{uv}\nabla(v+dv) = 0, \quad \boldsymbol{n}^\top \mathbf{D}^{d}\nabla(d+dd) = 0. \tag{6.21}$$

Der Glattheitsterm aus Gleichung (6.13) sowie die Euler-Lagrange-Gleichungen aus den Gleichungen (6.18) bis (6.20) können für $\mathbf{D}^{uv} = \mathbf{D}^{d} = I$ auf den homogenen und für $\mathbf{D}^{uv} = g^{uv}I$ und $\mathbf{D}^{d} = g^{d}I$ auf die bildgetriebene isotrope Glattheit zurückgeführt werden und stellen damit die verallgemeinerte Form der Euler-Lagrange-Gleichungen aller bildgetriebenen Glattheitsterme dar.

In Tabelle 6.3 sind die Diffusionstensoren für die möglichen Kombinationen aus treibender Domäne ($f$ und/oder $\zeta$) und Isotropie (homogen, isotrop oder anisotrop) gelistet. Die hier vorgestellen Varianten der bildgetriebenen Glattheit werden in Kapitel 7 vergleichend evaluiert.

Auf eine Darstellung der Diskretisierung der hier aufgeführten Euler-Lagrange-Gleichungen wird an dieser Stelle verzichtet, da in Abschnitt 6.4 eine allgemeine Formulierung vorgestellt wird, die für die anisotrope bild- und flussgetriebene Glattheit gilt.

## 6.3 Flussgetriebene Glattheit

Es wurde festgestellt, dass Szenenflusskanten nach Nagel und Enkelmann [54] eine Teilmenge der Bildkanten sind und es daher sinnvoll ist, an solchen Kanten bzw. orthogonal zu diesen eine reduzierte Glättung durchzuführen. Bildgetriebene Strategien haben laut Zimmer *et al.* [102] allerdings auch einen großen Nachteil: Strukturierte Bildregionen, in denen die Bildkanten nicht unbedingt mit den Flusskanten übereinstimmen, sind anfällig für Übersegmentierungsartefakte. Dieses Problem trifft nicht auf flussgetriebene Strategien zu, die anstatt auf die Kanten des Bildes oder der Disparitätskarte auf die des Szenenflusses selbst achten. Das bedeutet, die treibenden Domänen sind in diesem Fall der optische Fluss $(u + du, v + dv)^\top$ und die Zieldisparität $d + dd$. Flussgetriebene Methoden haben im Gegensatz zu bildgetriebenen Verfahren den Nachteil, dass die Flusskanten nicht so gut lokalisiert sind, so Weickert und Schnörr [92], da sich die Kanten des Szenenflusses erst während der Berechnung herausbilden. Aus diesem Grund werden fluss- und bildgetriebene Verfahren auch in Kombination genutzt, wie beispielsweise von Sun *et al.* [74] und Zimmer *et al.* [102]. In dieser Arbeit soll sich auf die reine fluss- und bildgetriebene Glattheit beschränkt werden. Im Folgenden wird die flussgetriebene isotrope und anisotrope Glattheit vorgestellt.

### 6.3.1 Flussgetriebene isotrope Glattheit

Um die Glättung nur an den Bewegungskanten zu reduzieren, führten Black und Anandan [8] den flussgetriebenen isotropen Glattheitsterm ein. Bei ihrem Modell handelt es sich um die subquadratische oder robuste Glattheit. Eine direkte Erweiterung auf den Szenenfluss ergibt einen Glattheitsterm der Form:

$$R(du, dv, dd) = \Psi\left(|\nabla(u + du)|^2 + |\nabla(v + dv)|^2\right) + \beta \cdot \Psi\left(|\nabla(d + dd)|^2\right). \tag{6.22}$$

Im Allgemeinen sollte $\Psi(s^2)$, wie die in Abschnitt 5.1.3 eingeführten subquadratischen Bestrafungsfunktionen, positiv, steigend, subquadratisch und streng konvex sein. Dabei werden $(u + du)$ und $(v + dv)$ gemeinsam robustifiziert, da ihre Kanten korrelieren.

Mit dem Glattheitsterm aus Gleichung (6.22) ergibt sich das Energiefunktional mit flussgetriebener isotroper Glattheit:

$$\begin{aligned} E(du, dv, dd) = \int_\Omega \Big( & \boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw} \\ & + \alpha \cdot \Psi\left(|\nabla(u + du)|^2 + |\nabla(v + dv)|^2\right) \\ & + \alpha\beta \cdot \Psi\left(|\nabla(d + dd)|^2\right) \Big) \, \mathrm{d}x \, \mathrm{d}y. \end{aligned} \tag{6.23}$$

Daraus leiten sich die folgenden Euler-Lagrange-Gleichungen ab:

$$\begin{aligned} 0 = & \mathbf{J}_{11} \cdot du + \mathbf{J}_{12} \cdot dv + \mathbf{J}_{13} \cdot dd + \mathbf{J}_{14} \\ & - \alpha \cdot \operatorname{div}\left(\Psi'(|\nabla(u + du)|^2 + |\nabla(v + dv)|^2) \cdot \nabla(u + du)\right), \end{aligned} \tag{6.24}$$

$$\begin{aligned} 0 = & \mathbf{J}_{12} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{23} \cdot dd + \mathbf{J}_{24} \\ & - \alpha \cdot \operatorname{div}\left(\Psi'(|\nabla(u + du)|^2 + |\nabla(v + dv)|^2) \cdot \nabla(v + dv)\right), \end{aligned} \tag{6.25}$$

| treibende Domäne | keine | $\overline{u}$, $\overline{v}$ | $\overline{d}$ | $\overline{u}$, $\overline{v}$ und $\overline{d}$ |
|---|---|---|---|---|
| Diffusivität $g$ | 1 | $g(\lvert\nabla\overline{u}\rvert^2 + \lvert\nabla\overline{v}\rvert^2)$ | $g(\lvert\nabla\overline{d}\rvert^2)$ | $g(\lvert\nabla\overline{u}\rvert^2 + \lvert\nabla\overline{v}\rvert^2 + \lvert\nabla\overline{d}\rvert^2)$ |

**Tabelle 6.4:** Alle Kombinationen der treibenden Domänen von flussgetriebener isotroper Glattheit und die jeweilige Zusammensetzung der Diffusivität $g$. Die Spalte ohne treibende Domäne beschreibt den homogenen Fall.

$$\begin{aligned} 0 = &\mathbf{J}_{13} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{33} \cdot dd + \mathbf{J}_{34} \\ &- \alpha\beta \cdot \operatorname{div}\left(\Psi'(\lvert\nabla(d+dd)\rvert^2) \cdot \nabla(d+dd)\right). \end{aligned} \tag{6.26}$$

Im Vergleich zur bildgetriebenen isotropen Glattheit aus Gleichungen (6.4) bis (6.6) kann festgestellt werden, dass die Glattheit der Unbekannten ebenfalls über die Ableitung einer subquadratischen Funktion $\Psi'$ gewichtet wird. Der Unterschied liegt jedoch darin, dass an dieser Stelle der Gradient des optischen Flusses und der Zieldisparität selbst, anstelle des Bildes und der Disparitätskarte, als treibende Domäne genutzt werden. Daraus kann induziert werden, dass hier gleichermaßen eine allgemeine Formulierung existiert, die eine beliebige Kombination der flussgetriebenen isotropen Domänen mit $g = \Psi'$ erlaubt:

$$0 = \mathbf{J}_{11} \cdot du + \mathbf{J}_{12} \cdot dv + \mathbf{J}_{13} \cdot dd + \mathbf{J}_{14} - \alpha \cdot \operatorname{div}\left(g^{uv} \cdot \nabla(u+du)\right), \tag{6.27}$$

$$0 = \mathbf{J}_{12} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{23} \cdot dd + \mathbf{J}_{24} - \alpha \cdot \operatorname{div}\left(g^{uv} \cdot \nabla(v+dv)\right), \tag{6.28}$$

$$0 = \mathbf{J}_{13} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{33} \cdot dd + \mathbf{J}_{34} - \alpha\beta \cdot \operatorname{div}\left(g^{d} \cdot \nabla(d+dd)\right). \tag{6.29}$$

Wie in Abschnitt 6.2.1 wird $g^{uv}$ als Diffusivität des optischen Flusses und $g^d$ als Diffusivität der Zieldisparität bezeichnet. Hier können jedoch die treibenden Domänen optischer Fluss $(\overline{u}, \overline{v})$, Zieldisparität $\overline{d}$ oder eine Kombination aus beiden gewählt werden. Die möglichen Kombinationen finden sich in Tabelle 6.4. Für $g^{uv}$ und $g^d$ kann dabei jeweils unabhängig voneinander einer der Ausdrücke für die Diffusivität $g$ verwendet werden. Die hier vorgestellten Varianten der flussgetriebenen isotropen Glattheit werden in Abschnitt 7.6 verglichen und evaluiert.

Diese Ausdrücke $g^{uv}$ und $g^d$ müssen nun für eine Diskretisierung der Euler-Lagrange-Gleichungen zusätzlich diskretisiert werden – hier beispielhaft für separate treibende Domänen gezeigt:

$$\begin{aligned} g^{uv}_{ij} &= \Psi'(\lvert\nabla(u+du)\rvert^2_{ij} + \lvert\nabla(v+dv)\rvert^2_{ij}) \\ &= \Psi'([u_x + du_x]^2_{ij} + [u_y + du_y]^2_{ij} + [v_x + dv_x]^2_{ij} + [v_y + dv_y]^2_{ij}), \end{aligned} \tag{6.30}$$

$$g^{d}_{ij} = \Psi'(\lvert\nabla(d+dd)\rvert^2_{ij}) = \Psi'([d_x + dd_x]^2_{ij} + [d_y + dd_y]^2_{ij}). \tag{6.31}$$

Demzufolge lauten die diskretisierten Euler-Lagrange-Gleichungen:

$$\begin{aligned} 0 = &[\mathbf{J}_{11}]_{ij} \cdot du_{ij} + [\mathbf{J}_{12}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{13}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{14}]_{ij} \\ &- \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{g^{uv}_{\tilde{i}\tilde{j}} + g^{uv}_{ij}}{2} \left( \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du_{\tilde{i}\tilde{j}} - du_{ij}}{h_l^2} \right), \end{aligned} \tag{6.32}$$

$$\begin{aligned} 0 = &[\mathbf{J}_{12}]_{ij} \cdot du_{ij} + [\mathbf{J}_{22}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{24}]_{ij} \\ &- \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{g^{uv}_{\tilde{i}\tilde{j}} + g^{uv}_{ij}}{2} \left( \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv_{\tilde{i}\tilde{j}} - dv_{ij}}{h_l^2} \right), \end{aligned} \tag{6.33}$$

$$
\begin{aligned}
0 =& [\mathbf{J}_{13}]_{ij} \cdot du_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{33}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{34}]_{ij} \\
& - \alpha\beta \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{g^d_{\tilde{i}\tilde{j}} + g^d_{ij}}{2} \left( \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd_{\tilde{i}\tilde{j}} - dd_{ij}}{h_l^2} \right),
\end{aligned} \tag{6.34}
$$

mit den Neumann-Randbedingungen

$$
\boldsymbol{n}^\top g^{uv} I \nabla (u + du) = 0, \quad \boldsymbol{n}^\top g^{uv} I \nabla (v + dv) = 0, \quad \boldsymbol{n}^\top g^d I \nabla (d + dd) = 0. \tag{6.35}
$$

Im Gegensatz zum bildgetriebenen isotropen Fall sind die hier gezeigten Gleichungen aufgrund der subquadratischen Bestrafung nun nicht mehr linear. Um ein solches, nichtlineares Gleichungssystem zu lösen, kann die verzögerte Nichtlinearitätsmethode von Kačur *et al.* [32] angewandt werden. Dabei wird das nichtlineare Gleichungssystem in eine Reihe von linearen Gleichungssystemen umgewandelt, indem eine zweite Fixpunktiteration eingeführt wird, welche die nichtlinearen Teile der Divergenz, $g^{uv}_{ij} = \Psi'(|\nabla(u + du)|^2_{ij} + |\nabla(v + dv)|^2_{ij})$ und $g^d_{ij} = \Psi'(|\nabla(d + dd)|^2_{ij})$, aus dem alten Zeitschritt $l$ berechnet, die in der inneren Iteration konstant gehalten werden:

$$
\begin{aligned}
0 =& [\mathbf{J}_{11}]_{ij} \cdot du^{l+1}_{ij} + [\mathbf{J}_{12}]_{ij} \cdot dv^{l+1}_{ij} + [\mathbf{J}_{13}]_{ij} \cdot dd^{l+1}_{ij} + [\mathbf{J}_{14}]_{ij} \\
& - \alpha \sum_{l' \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_{l'}(i,j)} \frac{g^{f,l}_{\tilde{i}\tilde{j}} + g^{f,l}_{ij}}{2} \left( \frac{u_{\tilde{i}\tilde{j}} - u_{ij} + du^{l+1}_{\tilde{i}\tilde{j}} - du^{l+1}_{ij}}{h_{l'}^2} \right),
\end{aligned} \tag{6.36}
$$

$$
\begin{aligned}
0 =& [\mathbf{J}_{12}]_{ij} \cdot du^{l+1}_{ij} + [\mathbf{J}_{22}]_{ij} \cdot dv^{l+1}_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dd^{l+1}_{ij} + [\mathbf{J}_{24}]_{ij} \\
& - \alpha \sum_{l' \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_{l'}(i,j)} \frac{g^{f,l}_{\tilde{i}\tilde{j}} + g^{f,l}_{ij}}{2} \left( \frac{v_{\tilde{i}\tilde{j}} - v_{ij} + dv^{l+1}_{\tilde{i}\tilde{j}} - dv^{l+1}_{ij}}{h_{l'}^2} \right),
\end{aligned} \tag{6.37}
$$

$$
\begin{aligned}
0 =& [\mathbf{J}_{13}]_{ij} \cdot du^{l+1}_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dv^{l+1}_{ij} + [\mathbf{J}_{33}]_{ij} \cdot dd^{l+1}_{ij} + [\mathbf{J}_{34}]_{ij} \\
& - \alpha\beta \sum_{l' \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_{l'}(i,j)} \frac{g^{\zeta,l}_{\tilde{i}\tilde{j}} + g^{\zeta,l}_{ij}}{2} \left( \frac{d_{\tilde{i}\tilde{j}} - d_{ij} + dd^{l+1}_{\tilde{i}\tilde{j}} - dd^{l+1}_{ij}}{h_{l'}^2} \right).
\end{aligned} \tag{6.38}
$$

Werden die Euler-Lagrange-Gleichungen der isotropen bild- und flussgetriebenen Glattheit – Gleichungen (6.9) bis (6.11) und Gleichungen (6.36) bis (6.38) – miteinander verglichen, fällt auf, dass sie fast identisch sind. Der einzige Unterschied besteht darin, dass bei der flussgetriebenen Glattheit $g^{uv}$ und $g^d$ nichtlinear von $du$, $dv$ und $dd$ abhängen, sodass diese in einer zusätzlichen äußeren Schleife berechnet werden müssen, wobei bei der bildgetriebenen Glattheit $g^{uv}$ und $g^d$ nicht von $du$, $dv$ und $dd$ abhängen, sodass diese schon vor der Fixpunktiteration berechnet werden können. Die Ähnlichkeit der Gleichungen wird von der späteren Diskretisierung mithilfe von Stencil in Abschnitt 6.4 für eine allgemeine Formulierung genutzt.

### 6.3.2 Flussgetriebene anisotrope Glattheit

Die flussgetriebene anisotrope Glattheit wurde von Weickert und Schnörr [92] für den optischen Fluss eingeführt und unterscheidet sich von der flussgetriebenen isotropen Glattheit darin, dass nicht mehr nur der Betrag der Gradienten des Flusses, sondern auch die Orientierung untersucht wird,

| treibende Domäne | keine | $\overline{u}, \overline{v}$ | $\overline{d}$ | $\overline{u}$, $\overline{v}$ und $\overline{d}$ |
|---|---|---|---|---|
| Diffusionstensor $\mathbf{D}$ | $I$ | $D_{flow}(\mathbf{T}_u + \mathbf{T}_v)$ | $D_{flow}(\mathbf{T}_d)$ | $D_{flow}(\mathbf{T}_u + \mathbf{T}_v + \mathbf{T}_d)$ |

**Tabelle 6.5:** Diffusionstensor $\mathbf{D}$ für die verschiedenen treibenden Domänen der flussgetriebenen anisotropen Glattheit. Die zweite Spalte ohne treibende Domäne beschreibt den homogenen Fall.

wodurch die Glättung orthogonal zu Flusskanten reduziert und entlang dieser verstärkt wird. Die Richtung des Flusses wird mithilfe eines Strukturtensors $\mathbf{T}$, hier am Beispiel für $u$, berechnet:

$$\mathbf{T}_u = \nabla(u+du)\nabla(u+du)^\top = \begin{pmatrix} (u+du)_x^2 & (u+du)_x(u+du)_y \\ (u+du)_x(u+du)_y & (u+du)_y^2 \end{pmatrix}. \tag{6.39}$$

Daraus ergibt sich der flussgetriebene anisotrope Glattheitsterm:

$$R(du, dv, dd) = tr\Psi\left(\mathbf{T}_u + \mathbf{T}_v\right) + \beta \cdot tr\Psi\left(\mathbf{T}_d\right). \tag{6.40}$$

Wie bei der isotropen flussgetriebenen Glattheit werden auch hier die Komponenten des optischen Flusses gemeinsam robustifiziert, da die Positionen ihrer Kanten korrelieren. Zu erwähnen ist an dieser Stelle, dass die Funktion $\Psi$ nicht mehr auf ein Skalar, sondern auf eine Matrix angewandt wird:

$$\Psi(A) = (\mathbf{e}_1, \mathbf{e}_2) \begin{pmatrix} \Psi_s(\lambda_1) & 0 \\ 0 & \Psi_s(\lambda_2) \end{pmatrix} (\mathbf{e}_1, \mathbf{e}_2)^\top. \tag{6.41}$$

Dabei werden die Eigenvektoren der Eingabematrix $A$ beibehalten und die skalare Funktion $\Psi_s$ (siehe Abschnitt 5.1.3) nur auf die Eigenwerte angewandt.

Mit dem Glattheitsterm aus Gleichung (6.40) lautet das Energiefunktional mit flussgetriebener anisotroper Glattheit:

$$\begin{aligned} E(du, dv, dd) = \int_\Omega \Big( & \boldsymbol{dw}^\top \mathbf{J} \boldsymbol{dw} \\ & + \alpha \cdot tr\Psi\left(\mathbf{T}_u + \mathbf{T}_v\right) \\ & + \alpha\beta \cdot tr\Psi\left(\mathbf{T}_d\right) \Big) \,\mathrm{d}x\,\mathrm{d}y. \end{aligned} \tag{6.42}$$

Daraus leiten sich analog zu Weickert und Schnörr [92] die folgenden Euler-Lagrange-Gleichungen ab:

$$0 = \mathbf{J}_{11} \cdot du + \mathbf{J}_{12} \cdot dv + \mathbf{J}_{13} \cdot dd + \mathbf{J}_{14} - \alpha \cdot \mathrm{div}\left(\Psi'\left(\mathbf{T}_u + \mathbf{T}_v\right) \cdot \nabla(u+du)\right), \tag{6.43}$$

$$0 = \mathbf{J}_{12} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{23} \cdot dd + \mathbf{J}_{24} - \alpha \cdot \mathrm{div}\left(\Psi'\left(\mathbf{T}_u + \mathbf{T}_v\right) \cdot \nabla(v+dv)\right), \tag{6.44}$$

$$0 = \mathbf{J}_{13} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{33} \cdot dd + \mathbf{J}_{34} - \alpha\beta \cdot \mathrm{div}\left(\Psi'\left(\mathbf{T}_d\right) \cdot \nabla(d+dd)\right), \tag{6.45}$$

Wie auch beim flussgetriebenen isotropen Fall findet eine Gewichtung der Glattheit statt, die eine allgemeine Formulierung mithilfe der Diffusionstensoren $\mathbf{D}^{uv}$ und $\mathbf{D}^d$ erlaubt:

$$0 = \mathbf{J}_{11} \cdot du + \mathbf{J}_{12} \cdot dv + \mathbf{J}_{13} \cdot dd + \mathbf{J}_{14} - \alpha \cdot \mathrm{div}\left(\mathbf{D}^{uv} \cdot \nabla(u+du)\right), \tag{6.46}$$

$$0 = \mathbf{J}_{12} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{23} \cdot dd + \mathbf{J}_{24} - \alpha \cdot \operatorname{div}\left(\mathbf{D}^{uv} \cdot \nabla(v + dv)\right), \tag{6.47}$$

$$0 = \mathbf{J}_{13} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{33} \cdot dd + \mathbf{J}_{34} - \alpha\beta \cdot \operatorname{div}\left(\mathbf{D}^{d} \cdot \nabla(d + dd)\right), \tag{6.48}$$

mit den Neumann-Randbedingungen

$$\boldsymbol{n}^\top \mathbf{D}^{uv} \nabla(u + du) = 0, \quad \boldsymbol{n}^\top \mathbf{D}^{uv} \nabla(v + dv) = 0, \quad \boldsymbol{n}^\top \mathbf{D}^{d} \nabla(d + dd) = 0. \tag{6.49}$$

Als treibende Domänen für die Diffusionstensoren $\mathbf{D}^{uv}$ und $\mathbf{D}^{d}$ können dabei jeweils der optische Fluss $(\overline{u}, \overline{v})$, die Zieldisparität $\overline{d}$ oder eine Kombination aus beiden gewählt werden. Die möglichen Kombinationen finden sich in Tabelle 6.5. Für $\mathbf{D}^{uv}$ und $\mathbf{D}^{d}$ kann dabei jeweils unabhängig voneinander einer der Ausdrücke für den Diffusionstensor $\mathbf{D}$ verwendet werden.

Die in dieser Arbeit verwendete Berechnungsvorschrift $D_{flow}$ für die flussgetriebenen anisotropen Diffusionstensoren $\mathbf{D}^{uv}$ und $\mathbf{D}^{d}$ entspricht der Ableitung der Funktion $\Psi$ aus Gleichung (6.41) und ist hier beispielhaft für die treibende Domäne $(\overline{u}, \overline{v})$ gezeigt:

$$\begin{aligned} \mathbf{D} &= D_{flow}(\mathbf{T}_u + \mathbf{T}_v) = \Psi'(\mathbf{T}_u + \mathbf{T}_v) \\ &= (\mathbf{e}_1, \mathbf{e}_2) \begin{pmatrix} \Psi_s'(\lambda_1) & 0 \\ 0 & \Psi_s'(\lambda_2) \end{pmatrix} (\mathbf{e}_1, \mathbf{e}_2)^\top. \end{aligned} \tag{6.50}$$

Der Glattheitsterm aus Gleichung (6.22) sowie die Euler-Lagrange-Gleichungen aus den Gleichungen (6.27) bis (6.29) können für $\mathbf{D}^{uv} = \mathbf{D}^{d} = I$ auf den homogenen und für $\mathbf{D}^{uv} = g^{uv} I$ und $\mathbf{D}^{d} = g^{d} I$ auf die flussgetriebene isotrope Glattheit zurückgeführt werden und stellen damit die verallgemeinerte Form der Euler-Lagrange-Gleichungen aller flussgetriebenen Glattheitsterme dar.

In Tabelle 6.6 sind die Diffusionstensoren für die möglichen Kombinationen aus treibender Domäne ($(\overline{u}, \overline{v})$ und/oder $\overline{d}$) und Isotropie (homogen, isotrop oder anisotrop) gelistet. Die hier vorgestellen Varianten der flussgetriebenen Glattheit werden in Abschnitt 7.6 evaluiert.

Beim Vergleich der Euler-Lagrange-Gleichungen der anisotropen fluss- und bildgetriebenen Glattheit – Gleichungen (6.46) bis (6.48) und (6.18) bis (6.20) – kann, wie zuvor schon beim Vergleich der Euler-Lagrange-Gleichungen der isotropen fluss- und bildgetriebenen Glattheit, festgestellt werden, dass sie für ein allgemeines $\mathbf{D}^{uv}$ und $\mathbf{D}^{d}$ identisch sind. Daraus lässt sich schließen, dass die hier vorgestellten Euler-Lagrange-Gleichungen (6.46) bis (6.48) eine Verallgemeinerung aller Glattheitstypen – bildgetrieben und flussgetrieben, jeweils isotrop und anisotrop sowie aller Kombinationen treibender Domänen– darstellt. Zur konkreten Bestimmung der Diffusionstensoren $\mathbf{D}^{uv}$ und $\mathbf{D}^{d}$ können für den bildgetriebenen Fall Tabelle 6.3 und für den flussgetriebenen Tabelle 6.6 herangezogen werden.

Diese Beobachtung motiviert eine allgemeine Diskretisierung, die auf alle Fälle zutrifft. Sie wird im folgenden Abschnitt hergeleitet.

| | $\overline{u}, \overline{v}$ | $\overline{d}$ | $\overline{u}, \overline{v}$ und $\overline{d}$ |
|---|---|---|---|
| homogen | $I$ | $I$ | $I$ |
| isotrop | $g(\lvert\nabla\overline{u}\rvert^2 + \lvert\nabla\overline{v}\rvert^2)I$ | $g(\lvert\nabla\overline{d}\rvert^2)I$ | $g(\lvert\nabla\overline{u}\rvert^2 + \lvert\nabla\overline{v}\rvert^2 + \lvert\nabla\overline{d}\rvert^2)I$ |
| anisotrop | $D_{flow}(\mathbf{T}_u + \mathbf{T}_v)$ | $D_{flow}(\mathbf{T}_d)$ | $D_{flow}(\mathbf{T}_u + \mathbf{T}_v + \mathbf{T}_d)$ |

**Tabelle 6.6:** Übersicht über alle möglichen Diffusionstensoren aus der Kombination von Isotropie (homogen, isotrop oder anisotrop) und treibender Domänen (optischer Fluss $\overline{u}, \overline{v}$ und/oder Zieldisparitäat $\overline{d}$) für die flussgetriebene Glattheit (mit Identitätsmatrix $I$).

## 6.4 Diskretisierung

Im vorherigen Abschnitt wurde festgestellt, dass die Euler-Lagrange-Gleichungen,

$$0 = \mathbf{J}_{11} \cdot du + \mathbf{J}_{12} \cdot dv + \mathbf{J}_{13} \cdot dd + \mathbf{J}_{14} - \alpha \cdot \text{div}\left(\mathbf{D}^{uv} \cdot \nabla(u + du)\right), \tag{6.46 Wdh.}$$

$$0 = \mathbf{J}_{12} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{23} \cdot dd + \mathbf{J}_{24} - \alpha \cdot \text{div}\left(\mathbf{D}^{uv} \cdot \nabla(v + dv)\right), \tag{6.47 Wdh.}$$

$$0 = \mathbf{J}_{13} \cdot du + \mathbf{J}_{22} \cdot dv + \mathbf{J}_{33} \cdot dd + \mathbf{J}_{34} - \alpha\beta \cdot \text{div}\left(\mathbf{D}^{d} \cdot \nabla(d + dd)\right), \tag{6.48 Wdh.}$$

für die in den Tabellen 6.3 und 6.6 aufgeführten Werte von $\mathbf{D}^{uv}$ und $\mathbf{D}^{d}$ alle Kombinationen von fluss- wie auch bildgetriebener isotroper und anisotroper Glattheit verschiedener treibender Domänen darstellen können und somit alle Glattheitstypen zusammenfassen. An dieser Stelle soll eine Diskretisierung dieser Gleichungen hergeleitet werden, die für alle Varianten der Glattheit gilt.

Der Term, der in den notwendigen Bedingungen mehrfach vorkommt, ist die Divergenz eines Tensors, multipliziert mit einem Gradienten und hat, hier beispielhaft für $(u + du)$ aufgeführt, die Form:

$$\text{div}\left(\mathbf{D} \cdot \nabla(u + du)\right). \tag{6.51}$$

Dies stellt den Diffusionsprozess des Feldes $(u + du)$ mit Diffusionstensor $\mathbf{D}$ dar. Wie von Weickert [91] beschrieben, basiert dieser Diffusionsprozess auf dem Ausgleich von Konzentrationen, wie er durch das Ficksche Gesetz und die Kontinuitätsgleichung beschrieben wird. Der Diffusionstensor $\mathbf{D}$, der den Diffusionsprozess charakterisiert, ist eine positiv definite, symmetrische Matrix, die aufgrund der Symmetrie die folgende Struktur aufweist:

$$\mathbf{D} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}. \tag{6.52}$$

Der Divergenz-Ausdruck setzt sich aus vier Termen zusammen, von denen zwei gemischte Ableitungen enthalten:

$$\begin{aligned} \text{div}(\mathbf{D} \cdot \nabla(u + du)) &= \text{div} \begin{pmatrix} a\partial_x(u + du) & b\partial_y(u + du) \\ b\partial_x(u + du) & c\partial_y(u + du) \end{pmatrix} \\ &= \partial_x(a\partial_x(u + du)) + \partial_x(b\partial_y(u + du)) \\ &\quad + \partial_y(b\partial_x(u + du)) + \partial_y(c\partial_y(u + du)). \end{aligned} \tag{6.53}$$

Die gemischten Terme können durch die zentralen Differenzen,

$$[\partial_x\Big(b\partial_y(u+du)\Big)]_{ij} \approx \frac{1}{2h_x}\Big(b_{i+1,j}\frac{[u+du]_{i+1,j+1}-[u+du]_{i+1,j-1}}{2h_y} - b_{i-1,j}\frac{[u+du]_{i-1,j+1}-[u+du]_{i-1,j-1}}{2h_y}\Big), \tag{6.54}$$

$$[\partial_y\Big(b\partial_x(u+du)\Big)]_{ij} \approx \frac{1}{2h_y}\Big(b_{i,j+1}\frac{[u+du]_{i+1,j+1}-[u+du]_{i-1,j+1}}{2h_x} - b_{i,j-1}\frac{[u+du]_{i+1,j-1}-[u+du]_{i-1,j-1}}{2h_x}\Big), \tag{6.55}$$

mit $[u+du]_{ij} = u_{ij} + du_{ij}$ diskretisiert werden. Die restlichen Terme werden, wie in Abschnitt 3.3 eingeführt, diskretisiert. Daraus ergibt sich der Stencil zur Diskretisierung der Divergenz:

$$S_{ij} = \begin{array}{|c|c|c|}\hline \frac{-b_{i-1,j}-b_{i,j+1}}{4h_xh_y} & \frac{c_{i,j+1}+c_{i,j}}{2h_y^2} & \frac{b_{i+1,j}-b_{i,j+1}}{4h_xh_y} \\ \hline \frac{a_{i-1,j}+a_{ij}}{2h_x^2} & \begin{array}{c}-\frac{a_{i-1,j}+2a_{ij}+a_{i+1,j}}{2h_x^2}\\ -\frac{c_{i,j-1}+2c_{i,j}+c_{i,j+1}}{2h_y^2}\end{array} & \frac{a_{i+1,j}+a_{ij}}{2h_x^2} \\ \hline \frac{b_{i-1,j}+b_{i,j-1}}{4h_xh_y} & \frac{c_{i,j-1}+c_{i,j}}{2h_y^2} & \frac{-b_{i+1,j}-b_{i,j-1}}{4h_xh_y} \\ \hline \end{array}. \tag{6.56}$$

Unter Berücksichtigung der Ränder der Bildebene verändert sich der Stencil mit der Indikatorfunktion aus Gleichung (3.40) zu:

$$S_{ij} = \begin{array}{|c|c|c|}\hline \frac{\chi_{i-1,j+1}(-b_{i-1,j}-b_{i,j+1})}{4h_xh_y} & \frac{\chi_{i,j+1}(c_{i,j+1}+c_{i,j})}{2h_y^2} & \frac{\chi_{i+1,j+1}(b_{i+1,j}-b_{i,j+1})}{4h_xh_y} \\ \hline \frac{\chi_{i-1,j}(a_{i-1,j}+a_{ij})}{2h_x^2} & \begin{array}{c}-\frac{\chi_{i-1,j}(a_{i-1,j}+a_{ij})}{2h_x^2}\\ -\frac{\chi_{i+1,j}(a_{i+1,j}+a_{ij})}{2h_x^2}\\ -\frac{\chi_{i,j+1}(c_{i,j+1}+c_{i,j})}{2h_y^2}\\ -\frac{\chi_{i,j-1}(c_{i,j-1}+c_{i,j})}{2h_y^2}\end{array} & \frac{\chi_{i+1,j}(a_{i+1,j}+a_{ij})}{2h_x^2} \\ \hline \frac{\chi_{i-1,j-1}(b_{i-1,j}+b_{i,j-1})}{4h_xh_y} & \frac{\chi_{i,j-1}(c_{i,j-1}+c_{i,j})}{2h_y^2} & \frac{\chi_{i+1,j-1}(-b_{i+1,j}-b_{i,j-1})}{4h_xh_y} \\ \hline \end{array}. \tag{6.57}$$

Mithilfe dieser Stencil-Notation lauten die diskretisierten Euler-Lagrange-Gleichungen wie folgt:

$$0 = [\mathbf{J}_{11}]_{ij}\cdot du_{ij} + [\mathbf{J}_{12}]_{ij}\cdot dv_{ij} + [\mathbf{J}_{13}]_{ij}\cdot dd_{ij} + [\mathbf{J}_{14}]_{ij} - \alpha \sum_{(\tilde{i},\tilde{j})\in\{-1,0,1\}} \sum S^f_{ij,\tilde{i}\tilde{j}}\cdot du_{i+\tilde{i},j+\tilde{j}}, \tag{6.58}$$

$$0 = [\mathbf{J}_{12}]_{ij} \cdot du_{ij} + [\mathbf{J}_{22}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{24}]_{ij} - \alpha \sum_{(\tilde{i},\tilde{j}) \in \{-1,0,1\}} \sum S^{f}_{ij,\tilde{i}\tilde{j}} \cdot dv_{i+\tilde{i},j+\tilde{j}}, \quad (6.59)$$

$$0 = [\mathbf{J}_{13}]_{ij} \cdot du_{ij} + [\mathbf{J}_{23}]_{ij} \cdot dv_{ij} + [\mathbf{J}_{33}]_{ij} \cdot dd_{ij} + [\mathbf{J}_{34}]_{ij} - \alpha\beta \sum_{(\tilde{i},\tilde{j}) \in \{-1,0,1\}} \sum S^{\zeta}_{ij,\tilde{i}\tilde{j}} \cdot dd_{i+\tilde{i},j+\tilde{j}}, \quad (6.60)$$

mit $S^f$ und $S^\zeta$ als Stencil, die zu den jeweiligen Diffusionstensoren $\mathbf{D}^{uv}$ und $\mathbf{D}^d$ gehören und den Neumann-Randbedingungen

$$\boldsymbol{n}^\top \mathbf{D}^{uv} \nabla (u + du) = 0, \quad (6.61)$$

$$\boldsymbol{n}^\top \mathbf{D}^{uv} \nabla (v + dv) = 0, \quad (6.62)$$

$$\boldsymbol{n}^\top \mathbf{D}^{d} \nabla (d + dd) = 0. \quad (6.63)$$

Je nach Glattheitstyp muss eine Unterscheidung in der Berechnung des Diffusionstensors getroffen werden. Für eine bildgetriebene Glattheit müssen die Werte von $a$, $b$ und $c$ nur einmal vor der Fixpunktiteration berechnet werden, da sie nicht von $du$, $dv$ oder $dd$ abhängen und sich somit im Verlauf der Schätzung nicht verändern. Für eine flussgetriebene Glattheit müssen sie, aufgrund der Abhängigkeit von $du$, $dv$ und $dd$, in einer äußeren Schleife (verzögerte Nichtlinearitätsmethode von Kačur *et al.* [32] siehe Abschnitt 6.3) aktualisiert werden.

Im Allgemeinen können die Diffusionstensoren $\mathbf{D}^{uv}$ für den optischen Fluss wie auch $\mathbf{D}^d$ für die Zieldisparität unabhängig voneinander aus einer Kombination der treibenden Domänen (oder aus einer einzelnen) gewählt werden. Diese sind im bildgetriebenen Fall das Bild $f$ und/oder die Disparitätskarte $\zeta$ und im flussgetriebenen Fall der optische Fluss selbst $\overline{u}, \overline{v}$ und/oder die Zieldisparität $\overline{d}$ selbst.

Die vorliegende Arbeit hat die erfolgreiche Herleitung einer Vielzahl von Daten- und Glattheitstermmodellierungen zur Szenenflussverfeinerung gezeigt. Diese werden im folgenden Kapitel vergleichend evaluiert.

# 7 Evaluation

Vorrangiges Ziel der Evaluation ist es, die Frage zu beantworten, inwiefern der vorgeschlagene variationelle Ansatz in der Lage ist, eine Verbesserung der initialen Szenenflussschätzung zu erreichen. Dafür wurde eine Implementierung gewählt, die es prinzipiell erlaubt, alle Kombinationen der in Kapitel 4 bis 6 vorgestellten Modellierungen für den Daten- und Glattheitsterm zu evaluieren.

In diesem Kapitel wird zunächst der verwendete Datensatz präsentiert, an dem die Evaluation durchgeführt wurde. Anschließend werden die Fehlermaße für die Evaluation definiert und die Verfahren vorgestellt, die zur initialen Schätzung des Szenenflusses verwendet wurden. Danach werden die durchgeführten Experimente zur Festlegung der Modellparameter und zur Auswahl von geeigneten Daten- und Glattheitstermen beschrieben. Im Anschluss wird für das ausgewählte Modell eine detaillierte Analyse der Ergebnisse durchgeführt.

## 7.1 Verwendeter Datensatz

Zum besseren Vergleich von Methoden zur Szenenflussschätzung entwickelten Menze und Geiger [52] im Jahr 2015 den KITTI[2]-Datensatz. Er besteht aus jeweils 200 Test- und Trainings-Farbbildpaaren von typischen hochdynamischen Straßenverkehrsszenen, die vom Dach eines fahrenden Fahrzeugs aus aufgenommen wurden. Zusätzlich wurden die zugehörigen Grundwahrheitswerte für den Szenenfluss auf Basis einer 3D-CAD-Modellierung angefertigt.

Die Evaluation der in dieser Arbeit entwickelten Verfeinerungsmodelle erfolgt am KITTI-Trainingsdatensatz, da die benötigten Grundwahrheitswerte für den Szenenfluss für diesen Datensatz veröffentlicht wurden. Beispielhaft sind in Abbildung 7.1 vier Sequenzen gezeigt.

Hingewiesen sei an dieser Stelle auf die für den KITTI-Datensatz verwendeten Begrifflichkeiten: Der geschätzte Szenenfluss wird über die Größen $u, v, disp_0, disp_1$ definiert, die dazugehörigen Grundwahrheiten als $u^{gt}, v^{gt}, disp_0^{gt}, disp_1^{gt}$ bezeichnet. Die Startdisparität $disp_0$ entspricht in der vorliegenden Arbeit der Disparitätskarte $\zeta$ zum Zeitpunkt $t$ und $disp_1$ der Zieldisparität $\overline{d}$.

## 7.2 Fehlermaße

Um die Qualität verschiedener Szenenflussschätzungen vergleichen zu können, muss der Fehler dieser Schätzungen zur Grundwahrheit quantifiziert werden. Ein häufig genutztes Maß zur Evaluation einzelner Pixel ist der sogenannte Endpunktfehler EE (*endpoint error*), der als Abstand zwischen geschätzter und tatsächlicher Endpunktposition definiert ist. Für eine Pixelposition $(i, j)$ eines

[2] KITTI steht für ‚Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago'.


**Abbildung 7.1:** Beispielsequenzen des KITTI-Trainingsdatensatzes. Die linke Spalte zeigt jeweils das Bild der linken Kamera des Stereo-Aufbaus zum Zeitpunkt $t$, die rechte Spalte das entsprechende Bild zum Zeitpunkt $t+1$. Von oben nach unten sind die Sequenzen 0, 23, 93 und 181 dargestellt.

Bildes der Größe $M \times N$ lautet der Endpunktfehler des geschätzten optischen Flusses $u, v$ mit dazugehörigen Grundwahrheitswerten $u^{gt}, v^{gt}$:

$$EE_{ij}(u, v, u^{gt}, v^{gt}) = \sqrt{(u_{ij} - u_{ij}^{gt})^2 + (v_{ij} - v_{ij}^{gt})^2}. \tag{7.1}$$

Der Endpunktfehler der geschätzten Startdisparität $disp_0$ mit dazugehörigen Grundwahrheitswerten $disp_0^{gt}$ ist:

$$EE_{ij}(disp_0, disp_0^{gt}) = |disp_{0,ij} - disp_{0,ij}^{gt}|. \tag{7.2}$$

Für die geschätzte Zieldisparität $disp_1$ mit dazugehörigen Grundwahrheitswerten $disp_1^{gt}$ ist der Endpunktfehler definiert als:

$$EE_{ij}(disp_1, disp_1^{gt}) = |disp_{1,ij} - disp_{1,ij}^{gt}|. \tag{7.3}$$

Aus diesen Pixelfehlern werden nun Fehlermaße für den gesamten Bildbereich eingeführt.

Für Schätzungen des optischen Flusses werden typischerweise zwei Fehlermaße verwendet: Der durchschnittliche Endpunktfehler AEE (*average endpoint error*) gibt den über alle Pixel $(i, j)$ gemittelten Endpunktfehler $EE_{ij}$ an, der *bad pixel error* FL wird als Prozentsatz der Pixel $(i, j)$ definiert, bei denen der Endpunktfehler $EE_{ij}$ über einem bestimmten Schwellenwert $T$ liegt. Der durchschnittliche Endpunktfehler und der *bad pixel* Fehler lassen sich für eine Schätzung $u, v$

und die dazugehörigen Grundwahrheitswerte $u^{gt}, v^{gt}$ eines Bildes der Größe $M \times N$ wie folgt berechnen:

$$AEE(u, v, u^{gt}, v^{gt}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} EE_{ij}(u, v, u^{gt}, v^{gt}), \tag{7.4}$$

$$FL(u, v, u^{gt}, v^{gt}) = \frac{100}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{1}_{(EE_{ij}(u,v,u^{gt},v^{gt})>T)}. \tag{7.5}$$

Zusätzlich werden für die quantitative Bewertung der gesamten Szenenflussschätzung zwei weitere Fehlermaße $D1$ und $D2$ definiert, die den prozentualen Anteil der *bad pixel* von Start- und Zieldisparität angeben und analog zu Gleichung (7.5) berechnet werden:

$$D1(disp_0, disp_0^{gt}) = \frac{100}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{1}_{EE_{ij}(disp_0, disp_0^{gt})>T)}, \tag{7.6}$$

$$D2(disp_1, disp_1^{gt}) = \frac{100}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{1}_{EE_{ij}(disp_1, disp_1^{gt})>T)}. \tag{7.7}$$

Die Fehler $D1$, $D2$ und $FL$ werden noch in der Größe $SF$ zusammengefasst, die ein Fehlermaß für den gesamten Szenenfluss definiert und den Prozentsatz der Pixel angibt, die in $D1$, $D2$ oder $FL$ als Ausreißer gelten:

$$\begin{aligned} &SF(u, v, disp_0, disp_1, u^{gt}, v^{gt}, disp_0^{gt}, disp_1^{gt}) \\ &= 100 - \frac{100}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{1}_{(EE_{ij}(u,v,u^{gt},v^{gt})\leq T)} \cdot \mathbf{1}_{(EE_{ij}(disp_0, disp_0^{gt})\leq T)} \cdot \mathbf{1}_{(EE_{ij}(disp_1, disp_1^{gt})\leq T)}. \end{aligned} \tag{7.8}$$

Der durchschnittliche Endpunktfehler $AEE$ wird im Rahmen dieser Arbeit nicht verwendet, da zur besseren Vergleichbarkeit ausschließlich die *bad pixel* Fehler von $SF$, $FL$ und $D2$ genutzt werden. Außerdem wird das Fehlermaß $D1$ nicht in den Vergleichstabellen von Abschnitt 7.5 und 7.6 aufgeführt, da die Verfeinerung nur auf den optischen Fluss $u, v$ und die Zieldisparität $disp_1$ angewandt wird und sich damit $disp_0$ und der dazugehörige $D1$-Fehler nicht verändern.

Die Ergebnistabellen in Abschnitt 7.5 und 7.6 geben zusätzlich auch die relative Verbesserung in diesen Fehlerwerten als Prozentsatz, um den sich der jeweilige Fehler der initialen Schätzung durch den Verfeinerungsschritt verbessert hat, an:

$$\hat{SF} = 100 \cdot \frac{SF^0 - SF^1}{SF^0}, \hat{D2} = 100 \cdot \frac{D2^0 - D2^1}{D2^0}, \hat{FL} = 100 \cdot \frac{FL^0 - FL^1}{FL^0}. \tag{7.9}$$

Der Wert $SF^0$ bezeichnet dabei den $SF$-Fehler der initialen Schätzung und $SF^1$ den der verfeinerten Schätzung. Analoges gilt für $D2$ und $FL$. Ist die relative Verbesserung positiv, so wurde der initiale Wert durch den Verfeinerungsschritt verbessert. Umgekehrt bedeutet ein negativer Wert eine Verschlechterung.

Die eingeführten Fehlermaße wurden zunächst für eine einzelne Sequenz definiert. Für die Evaluation werden im Allgemeinen mehrere Sequenzen herangezogen. Daher werden für eine gesamtheitliche Bewertung die in Gleichungen (7.5) bis (7.9) definierten Fehlermaße implizit als Mittelwert über die Sequenzen berechnet und interpretiert.

Die Evaluation mithilfe des KITTI-Trainingsdatensatzes erfolgt standardmäßig in drei Regionen: Hintergrund (bg), Vordergrund (fg) und gesamtes Bild (all). Unter Vordergrund werden fahrende Autos verstanden, die mit einem 3D-CAD-Modell modelliert wurden, während der Hintergrund den Rest beschreibt. Für jeden dieser Bereiche werden die hier eingeführten Fehlerwerte $D1$, $D2$, $FL$ und $SF$ berechnet. Der Schwellenwert $T$ liegt typischerweise bei 5% der Gesamtvektorlänge mindestens aber 3 Pixel, siehe Menze und Geiger [52].

## 7.3 Initiale Szenenflussschätzung

Zur Generierung der initialen Szenenflussschätzung, werden eine Stereo- und ein Szenenflussschätzer benötigt, wie im Flussdiagramm in Abbildung 4.1 zu erkennen ist. In dieser Arbeit wird das Modell GA-Net von Zhang *et al.* [99] als Stereoschätzer für die Disparitätskarte $\zeta$ und das Modell RAFT-3D als Szenenflusschätzer für $u$,$v$ und $d$ verwendet. Daher werden die beiden Modelle im Folgenden kurz vorgestellt.

### 7.3.1 GA-Net

Das von Zhang *et al.* [99] vorgeschlagene End-to-End-Stereo-Rekonstruktionsmodell GA-Net (*Guided Aggregation Net*) aus dem Jahr 2019 enthält zwei neuartige neuronale Netzschichten, die darauf abzielen, lokale und globale Abhängigkeiten zu erfassen, und erreichte zum Zeitpunkt der Veröffentlichung *state-of-the-art* Genauigkeit im KITTI-Datensatz [52] zur Stereoschätzung. Die erste dieser Schichten ist eine semiglobale Aggregationsschicht (SGA), die eine differenzierbare Annäherung an das semiglobale Matching darstellt und die Matchingkosten in verschiedenen Richtungen über das gesamte Bild aggregiert. Dies ermöglicht genaue Schätzungen in verdeckten, großflächig texturlosen oder reflektierenden Regionen. Die zweite Schicht ist eine lokal geführte Aggregationsschicht (LGA), die eine traditionelle Kostenfilterungsstrategie zur Verfeinerung dünner Strukturen und Objektkanten verfolgt, um den durch *Down*- und *Upsampling*-Schichten verursachten Detailverlust auszugleichen. Diese beiden Schichten können die weit verbreitete 3D-Faltungsschicht (*3D convolutional layer*) ersetzen, die aufgrund ihrer kubischen Rechen- bzw. Speicherkomplexität sehr rechen- und speicheraufwändig ist. Auf diese Weise stellt GA-Net ein Echtzeitmodell dar, das Geschwindigkeit von 15 bis 20 *fps* ereicht und im Vergleich zu anderen bestehenden Echtzeit-Algorithmen zum Zeitpunkt der Veröffentlichung eine bessere Genauigkeit erzielt.

GA-Net besteht aus vier Teilen, die in Abbildung 7.2 aufgezeigt sind: dem Block für die Merkmalsextraktion, der Kostenaggregation für das 4D-Kostenvolumen, dem Guidance-Subnetz zur Erzeugung der Gewichte für die Kostenaggregation und der Disparitätsregression. Die Merkmalsextraktion, die auf das linke und rechte Bild angewandt wird, besteht aus einenem gestapelten Sanduhrnetz (*stacked hourglass network*), das durch Verkettungen zwischen verschiedenen Schichten dicht verbunden ist. Die extrahierten Merkmale für das linke und rechte Bild werden daraufhin verwendet, um ein 4D-Kostenvolumen zu konstruieren. Für die Kostenaggregation werden mehrere SGA-Schichten und eine LGA-Schicht genutzt. Die LGA-Schicht verfeinert das 4D-Kostenvolumen mehrmals lokal. Die von den SGA- und LGA-Schichten benötigten Gewichtsmatrizen werden durch ein zusätzliches Guidance-Subnetz generiert, das aus mehreren 2D-Faltungsschichten besteht, die schneller als

3D-Faltungsschichten sind und die Referenzansicht (beispielsweise das linke Bild) als Eingabe verwendet. Der letzte Schritt ist die Disparitätsregression, welche die Disparität basierend auf dem 4D-Kostenvolumen schätzt.

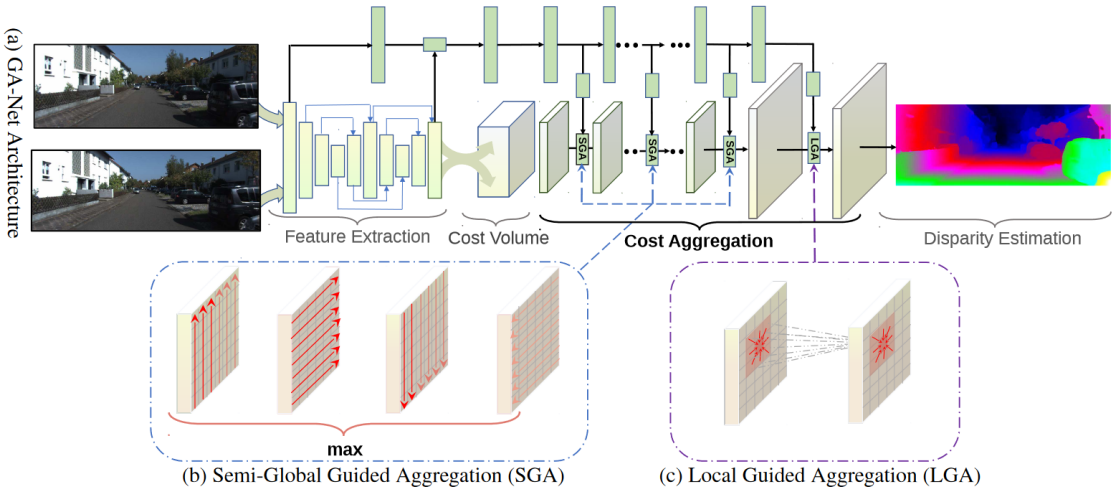



**Abbildung 7.2:** (a) Überblick über die GA-Net Architektur [99]. Beide Einzelbilder einer Stereosequenz werden in eine Pipeline zur Merkmalsextraktion gegeben, die aus einem gestapelten Sanduhrnetz (*stacked hourglass network*) besteht und durch Verkettungen verbunden ist. Die extrahierten linken und rechten Bildmerkmale werden daraufhin verwendet, um ein 4D-Kostenvolumen zu bilden, das in einen Kostenaggregationsblock zur Regularisierung und Verfeinerung und Disparitätsregression eingespeist wird. Das Guidance-Subnetz (grün) erzeugt die Gewichtsmatrizen für die geführten Kostenaggregationen (SGA und LGA). (b) SGA-Schichten aggregieren das Kostenvolumen semiglobal in vier Richtungen. (c) Die LGA-Schicht wird vor der Disparitätsregression verwendet und verfeinert das 4D-Kostenvolumen mehrmals lokal.

### 7.3.2 RAFT-3D

Das lernbasierte Verfahren RAFT-3D von Teed und Deng [77] aus dem Jahr 2021 ist ein *state-of-the-art* Netzwerk zur Szenenflussschätzung, welches zum Zeitpunkt der Veröffentlichung die besten Ergebnisse im KITTI-Datensatz [52] erzielte. Es basiert auf dem RAFT (*Recurrent All-Pairs Field Transforms*) Modell [78] , das zur Schätzung des optischen Flusses entwickelt wurde. RAFT-3D aktualisiert iterativ das dichte Feld der pixelweisen SE3-Bewegung anstelle der 2D-Bewegung. Die Schlüsselinnovation ist dabei die Kodierung starrer Bewegung, die eine weiche Segmentierung von Pixeln in starre Objekte darstellt. Dafür wird die sogenannte Dense-SE3-Schicht verwendet, eine differenzierbare Schicht, die geometrische Konsistenz erzwingt und das dichte Feld der SE3-Bewegung pro Pixel iterativ aktualisiert.

Das RAFT-3D Modell ist wie in Abbildung 7.3 skizziert aufgebaut: Aus dem RGB-D-Bildpaar werden Merkmale extrahiert ein 4D-Korrelationsvolumen durch Berechnung der visuellen Ähnlichkeit zwischen allen Pixelpaaren erstellt. Dabei speichert und aktualisiert es ein dichtes Feld der pixelwei-

| Training | Evaluation | D2-all | Fl-all | SF-all |
|---|---|---|---|---|
| Fly + Train | Train | 0.678 | 1.186 | 1.324 |
| Fly + Train | Test | 3.67 | 4.29 | 5.77 |
| Fly | Train | 4.101 | 9.721 | 9.951 |

**Tabelle 7.1:** Evaluierung verschiedener RAFT-3D Modelle. Die erste Spalte gibt an, auf welchem Datensatz das Modell trainiert wurde, die zweite Spalte beschreibt den zur Evaluation genutzten Datensatz. ‚Fly' bzeichnet den FlyingThings3D-Datensatz [47], ‚Train' den KITTI-Trainingsdatensatz und ‚Test' den KITTI-Testdatensatz. Die restlichen Spalten geben den über dem jeweiligen Evaluationsdatensatz gemittelten Szenenflussfehler.

sen SE3-Bewegung. Bei jeder Iteration wird die aktuelle Schätzung der SE3-Bewegung verwendet, um aus dem Korrelationsvolumen zu indizieren. Ein rekurrenter GRU-basierter Aktualisierungsoperator nimmt die Korrelationsmerkmale und erzeugt eine Schätzung der Pixelkorrespondenz. Diese wrid dann von der Dense-SE3-Schicht verwendet, um Aktualisierungen des SE3-Bewegungsfeldes zu erzeugen.

Sind die Eingabebilder nicht im RGB-D-Format sondern als Stereobildpaare gegeben, muss ein Stereoschätzer, wie beispielsweise GA-Net [99], verwendet werden, um die Disparität zu berechnen und somit das benötigte Format zu generieren.

Anzumerken ist an dieser Stelle, dass die Merkmalsextraktion, welche aus einem Merkmalscodierer und einem Kontextcodierer besteht, auf einem Achtel der Bildauflösung arbeitet. Dies wird später durch ein *Upsampling* ausgeglichen, lässt jedoch vermuten, dass eine nachfolgende Verfeinerung mit höherer Auflösung die Ergebnisschätzung verbessern kann. Die Fehler des von RAFT-3D berechneten Szenenflusses mit vorheriger Disparitätsberechnung durch GA-Net sind in Tabelle 7.1 für den KITTI-Datensatz [52] aufgezeigt. Dafür wurde das Modell auf dem FlyingThings3D-Datensatz [47] trainiert und mit den Trainingsbildern des KITTI-Datensatzes [52] *gefinetuned*.

Für die Evaluation des in dieser Arbeit entwickelten Verfeinerungsschrittes wird der KITTI-Trainigsdatensatz verwendet, wie in Abschnitt 7.1 motiviert. Da RAFT-3D auf diesem Datensatz trainiert wurde, besteht die Gefahr des *Overfitting* für die initiale Szenenflussschätzung. Dies kann dazu führen, dass das Verbesserungspotential für den nachfolgenden Verfeinerungsschritt im Vergleich zu einem unbekannten Datensatz stark eingeschränkt wird. Daher wurde für diese Arbeit das RAFT-3D Modell ausschließlich auf dem unabhängigen FlyingThings3D-Datensatz [47] trainiert und damit die initiale Szenenflussschätzung sowie ihre Fehler für den KITTI-Trainingsdatensatz berechnet.

Tabelle 7.1 vergleicht die Fehlerwerte des orginalen und des neuen RAFT-3D Modells für den KITTI-Test- und Trainingsdatensatz. Ein Vergleich der Fehlerwerte des originalen Modells für den Test- und Trainingsdatensatz zeigt, dass ein *Overfitting* vorliegt.

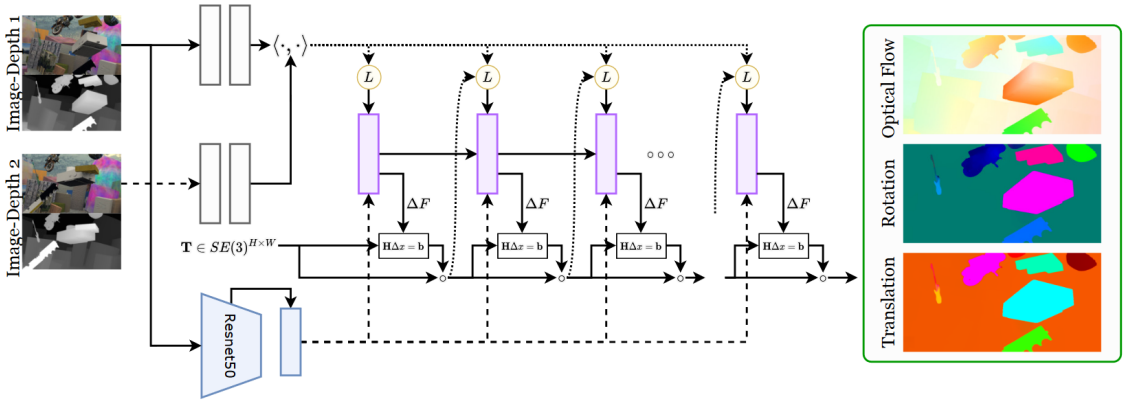



**Abbildung 7.3:** Überblick über die RAFT-3D Architektur [77]. Die aus den Eingabedaten extrahierten Merkmale werden verwendet, um ein 4D-Korrelationsvolumen zu konstruieren. Während jeder Iteration verwendet der Aktualisierungsoperator die aktuelle SE3-Bewegungsschätzung, um aus dem Korrelationsvolumen zu indizieren. Diese Schätzungen werden in die Dense-SE3-Schicht eingefügt, eine Optimierungsschicht, die geometrische Beschränkungen verwendet, um eine Aktualisierung des SE3-Feldes zu erzeugen. Nach der iterativen Berechnung wird ein dichtes SE3-Feld, das in eine Rotations- und eine Translationskomponente zerlegt werden kann, ausgegeben. Das SE3-Feld kann auf das Bild projiziert werden, um den optischen Fluss zu ermitteln.

## 7.4 Parameteroptimierung

Die in Kapitel 4 bis 6 eingeführten Daten- und Glattheitsterme zur Modellierung der Szenenflussverfeinerung enthalten eine Vielzahl von Parametern, welche in Tabelle 7.2 aufgelistet sind und einen großen Einfluss auf die Schätzung haben können. Um ein aussagekräftiges Modell aufstellen zu können, müssen Daten- und Glattheitsterm jeweils optimiert und zueinander ausbalanciert werden. Dazu ist es eigentlich notwendig die Parameteroptimierung für jede mögliche Kombination von Daten- und Glattheitsterm durchzuführen. Wegen der Vielzahl an möglichen Kombinationen ist dies aus praktischer Sicht nicht möglich. Daher wurde zunächst die Parameteroptimierung für ein

| Symbol | Wert | Erklärung |
|---|---|---|
| $\mu_{Bild}$ | 5.35* | Gewichtung der Grauwertkonstanz der Bildfolge im Datenterm |
| $\mu_{Disp}$ | 2.37* | Gewichtung der Grauwertkonstanz der Disparität im Datenterm |
| $\gamma$ | 0.02* | Gewichtung der Gradientenkonstanz der Bildfolge im Datenterm |
| $\alpha$ | 25.44* | Gewichtung des gesamten Glattheitsterms |
| $\beta$ | 2.06* | Gewichtung der Disparitätsglätte im Glattheitsterm |
| $\sigma$ | 2.20* | Standardabweichung der Gaußkurve zur Vorglättung der Eingabegrößen |
| $\omega$ | 0.8 | Relaxationsparameter des SOR-Verfahrens |
| $\epsilon$ | 0.05 | kleine Konstante von Funktionen |

**Tabelle 7.2:** Übersicht über die festgelegten Modellparameter. Die mit Stern markierten Werte wurden gemäß dem Framework von Akiba *et al.* [1] ermittelt. Die nicht markierten basieren auf Erfahrungswerten und Veröffentlichungen.

ausgewähltes Modell durchgeführt. Die Auswahl wurde basierend auf Erfahrungen der Arbeitsgruppe und Veröffentlichungen getroffen. Das ausgewählte Modell setzt sich aus einem Datenterm mit separat robustifizierter Grauwertkonstanz von HSV-Bild und Disparität sowie Gradientenkonstanz von HSV-Bild und einem flussgetriebenen anisotropen Glattheitsterm, bei dem $(\overline{u}, \overline{v})$ als treibende Domäne des optischen Flusses und $\overline{d}$ als die der Zieldisparität festgelegt wurden, zusammen. Dies entspricht dem Datenterm *bcaGcaColSep* aus Tabelle 7.3 und dem Glattheitsterm *flowAnisoSep* aus Tabelle 7.5. Damit ergibt sich folgendes Energiefunktional:

$$\begin{aligned} E(\boldsymbol{dw}) = \int_{\Omega} \Big( & \mu_{Bild} \cdot \sum_{i=1}^{3} \Psi \left( \boldsymbol{dw}^{\top} \overline{\mathbf{J}}^{i}_{bca,Bild} \boldsymbol{dw} \right) \\ & + \mu_{Disp} \cdot \Psi \left( \boldsymbol{dw}^{\top} \mathbf{J}_{bca,Disp} \boldsymbol{dw} \right) \\ & + \gamma \cdot \sum_{i=1}^{3} \Psi \left( \boldsymbol{dw}^{\top} \overline{\mathbf{J}}^{i}_{gca,Bild} \boldsymbol{dw} \right) \\ & + \alpha \cdot tr\Psi \left( \mathbf{T}_u + \mathbf{T}_v \right) \\ & + \alpha\beta \cdot tr\Psi \left( \mathbf{T}_d \right) \Big) \, \mathrm{d}x \, \mathrm{d}y. \end{aligned} \tag{7.10}$$

Dabei sind die Tensoren $\overline{\mathbf{J}}^{i}_{bca,Bild}$ und $\overline{\mathbf{J}}^{i}_{gca,Bild}$ aus Gleichung (5.46), $\mathbf{J}_{bca,Disp}$ aus Gleichung (4.13) und die Strukturtensoren $\mathbf{T}$ aus Gleichung (6.39) entnommen. Dieses Modell enthält alle Parameter, die für die Modellierung des Verfeinerungsschrittes im Rahmen dieser Arbeit eingeführt wurden. Sie werden in Tabelle 7.2 aufgelistet und erklärt.

Zur Optimierung der Parameter wurde das Framework von Akiba *et al.* [1] mit der Optimierungsstrategie *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) von Hansen und Ostermeier [24] genutzt. Die Anzahl der äußeren und inneren Iterationen wurde auf drei gesetzt. Außerdem wurde der Relaxationsparameter des SOR-Verfahrens zu $\omega = 0.8$ festgelegt, um feinschrittig ein Minimum des aufgestellten Energiefunktionals durch Unterrelaxation zu finden. Als Bestrafungsfunktion der Robustifizierung von Daten- und Glattheitsterm wurde die Charbonnier-Bestrafung aus Gleichung (5.25) mit $\epsilon = 0.05$ verwendet. Aus Laufzeitgründen wurde die Optimierung auf 16 zufällig ausgewählten Bildern des KITTI-Trainingsdatensatzes durchgeführt. Das Ergebnis der Parameteroptimierung für das ausgewählte Modell findet sich in Tabelle 7.2.

Die so festgelegten Modellparameter werden im weiteren Verlauf der Arbeit für alle aufgestellten Modelle (siehe Tabellen 7.3 und 7.5) verwendet, um deren Güte zu evaluieren. Die Parameter sind damit zwar nicht für jedes Modell optimiert, ermöglichen aber dennoch zumindest einen qualitativen Vergleich der verschiedenen Modelle.

Im Folgenden werden diese Parameter genutzt, um zunächst das optimale Modell für den Datenterm, unter Annahme homogener Glattheit, zu bestimmen. Danach werden mit diesem so erhaltenen Datenterm die verschiedenen Modelle für den Glattheitsterm evaluiert. Dies führt dann zum optimierten Gesamtmodell für die Verfeinerung der Szenenflussschätzung.

## 7.5 Auswahl des Datenterms

Zur Erstellung des optimalen Modells zur Szenenflussverfeinerung wird nun in einem ersten Schritt der bestmögliche Datenterm ermittelt. Hierfür wird eine Auswahl an Datentermen, bestehend aus Annahmen von Abschnitt 4.2.1 und 5, anhand des KITTI-Trainingsdatensatzes evaluiert. Für eine bessere Vergleichbarkeit, die eine Unabhängigkeit vom Glattheitsterm gewährleisten soll, werden alle mit einem homogenen Glattheitsterm kombiniert.

Die erarbeitete Implementierung ermöglicht eine beliebige Zusammensetzung des Datenterms aus Grauwertkonstanz von Grauwert- und Farbbildern sowie Zieldisparität (siehe Abschnitt 3.1.1), Gradientenkonstanz von Grauwert- und Farbbildern (siehe Abschnitt 5.2) und zusätzlich vollständig separater oder gemeinsamer Robustifizierung (siehe Abschnitt 5.1). Die ausgewählten Modellkombinationen, für die im Rahmen dieser Arbeit eine Bewertung durchgeführt wurde, sind nach Annahmen sortiert in Tabelle 7.3 aufgelistet.

Für die Modellnamen wurde folgende Konvention gewählt, anhand derer die Zusammensetzung des jeweiligen Datenterms erkennbar ist:

- *bca*: Grauwertkonstanz der Bildfolge
- *gca*: Gradientenkonstanz der Bildfolge
- *gray*: Grauwertbilder
- *col*: HSV-Bilder
- *join*: gemeinsame Robustifizierung
- *sep*: separate Robustifizierung.

Kein Bestandteil der Nomenklatur ist die Grauwertkonstanz der Zieldisparität, da sie in jedem der aufgelisteten Modelle enthalten ist. Die Zieldisparität umfasst nur eine Konstanzannahme – die Grauwertkonstanz –, da weder mehrere Kanäle zur Verfügung stehen, noch die Gradientenkonstanz implementiert wurde (siehe Abschnitt 5.2).

Als Bestrafungsfunktion der Robustifizierung wird die Charbonnier-Bestrafung aus Gleichung (5.25) eingesetzt. Modelle, die weder separat noch gemeinsam robustifiziert sind, enthalten den quadratisch bestraften Datenterm.

| | BCA | | | GCA | | | Robustifizierung | |
|---|---|---|---|---|---|---|---|---|
| **Datentermmodell** | **Grau** | **Farbe** | **Zieldisparität** | **Grau** | **Farbe** | **Zieldisparität** | **Separat** | **Gemeinsam** |
| bcaGray | ✓ | - | ✓ | - | - | - | - | - |
| bcaGraySep | ✓ | - | ✓ | - | - | - | ✓ | - |
| bcaGrayJoin | ✓ | - | ✓ | - | - | - | - | ✓ |
| bcaColSep | - | ✓ | ✓ | - | - | - | ✓ | - |
| bcaColJoin | - | ✓ | ✓ | - | - | - | - | ✓ |
| gcaGray | - | - | ✓ | ✓ | - | - | - | - |
| gcaGraySep | - | - | ✓ | ✓ | - | - | ✓ | - |
| gcaGrayJoin | - | - | ✓ | ✓ | - | - | - | ✓ |
| gcaColSep | - | - | ✓ | - | ✓ | - | ✓ | - |
| gcaColJoin | - | - | ✓ | - | ✓ | - | - | ✓ |
| bcaGcaGraySep | ✓ | - | ✓ | ✓ | - | - | ✓ | - |
| bcaGcaGrayJoin | ✓ | - | ✓ | ✓ | - | - | - | ✓ |
| bcaGcaColSep* | - | ✓ | ✓ | - | ✓ | - | ✓ | - |
| bcaGcaColJoin | - | ✓ | ✓ | - | ✓ | - | - | ✓ |
| bcaGrayGcaColSep | ✓ | - | ✓ | - | ✓ | - | ✓ | - |
| bcaGrayGcaColJoin | ✓ | - | ✓ | - | ✓ | - | - | ✓ |
| bcaColGcaGraySep | - | ✓ | ✓ | ✓ | - | - | ✓ | - |
| bcaColGcaGrayJoin | - | ✓ | ✓ | ✓ | - | - | - | ✓ |

**Tabelle 7.3:** Überblick über die Nomenklatur und die Zusammensetzung für die im Rahmen dieser Arbeit analysierten Modellvarianten für den Datenterm. Die Spalte ‚BCA' enthält die Optionen für die Grauwertkonstanz und ist in ‚Grau' für Grauwertbilder, ‚Farbe' für Farbbilder und ‚Zieldisparität' unterteilt. Diese Aufteilung gilt auch für die Spalte ‚GCA', bezieht sich jedoch auf die Gradientenkonstanz. Die Spalte ‚Robustifizierung' ist in ‚Separat' und ‚Gemeinsam' unterteilt und bezieht sich auf vollständig separate oder gemeinsame Robustifizierung des Datenterms. Das mit Stern gekennzeichnete Modell wurde für die Parameteroptimierung herangezogen.

| Datentermmodell | D2-bg | D2-fg | D2-all | FL-bg | FL-fg | FL-all | SF-bg | SF-fg | SF-all |
|---|---|---|---|---|---|---|---|---|---|
| RAFT-3D [77] | **3.682** | **0.420** | **4.101** | **7.741** | **1.980** | **9.721** | **9.344** | **13.307** | **9.951** |
| bcaGray | 76.237 | 11.353 | 87.589 | 8.292 | 2.580 | 10.872 | 90.451 | 77.697 | 88.497 |
| bcaGraySep | 4.258 | 0.611 | 4.870 | 8.269 | 2.506 | 10.776 | 10.218 | 16.817 | 11.229 |
| bcaGrayJoin | 62.411 | 10.292 | 72.703 | 8.275 | 2.579 | 10.854 | 76.995 | 72.927 | 76.371 |
| bcaColSep | 4.258 | 0.611 | 4.868 | 8.925 | 2.666 | 11.591 | 10.965 | 17.853 | 12.020 |
| bcaColJoin | 4.900 | 0.778 | 5.679 | 69.730 | 12.094 | 81.825 | 82.630 | 79.149 | 82.097 |
| gcaGray | 76.237 | 11.353 | 87.589 | 8.292 | 2.580 | 10.872 | 90.451 | 77.697 | 88.497 |
| gcaGraySep | 4.258 | 0.611 | 4.870 | 8.269 | 2.506 | 10.776 | 10.218 | 16.817 | 11.229 |
| gcaGrayJoin | 62.428 | 10.315 | 72.744 | 8.275 | 2.579 | 10.854 | 77.011 | 73.025 | 76.400 |
| gcaColSep | 4.258 | 0.611 | 4.870 | 8.268 | 2.504 | 10.773 | 10.217 | 16.804 | 11.226 |
| gcaColJoin | 58.246 | 9.293 | 67.539 | 8.264 | 2.563 | 10.827 | 72.279 | 67.364 | 71.526 |
| bcaGcaGraySep | 4.258 | 0.611 | 4.870 | 8.269 | 2.506 | 10.776 | 10.218 | 16.817 | 11.229 |
| bcaGcaGrayJoin | 62.411 | 10.292 | 72.703 | 8.275 | 2.579 | 10.854 | 76.995 | 72.927 | 76.371 |
| bcaGcaColSep* | 4.258 | 0.611 | 4.868 | 8.913 | 2.663 | 11.576 | 10.952 | 17.833 | 12.006 |
| bcaGcaColJoin | 4.841 | 0.767 | 5.608 | 69.615 | 12.072 | 81.687 | 82.485 | 79.001 | 81.951 |
| bcaGrayGcaColSep | 4.258 | 0.611 | 4.869 | 8.873 | 2.583 | 11.456 | 10.921 | 17.316 | 11.901 |
| bcaGrayGcaColJoin | 20.544 | 2.093 | 22.636 | 16.497 | 3.503 | 20.001 | 36.505 | 30.894 | 35.646 |
| bcaColGcaGraySep | 4.258 | 0.611 | 4.868 | 8.923 | 2.666 | 11.589 | 10.963 | 17.851 | 12.018 |
| bcaColGcaGrayJoin | 4.883 | 0.772 | 5.655 | 69.537 | 12.085 | 81.622 | 82.401 | 79.089 | 81.894 |
| Relative Verbesserung in % | -15.65 | -45.57 | -18.71 | -6.76 | -26.47 | -10.82 | -9.34 | -26.28 | -12.81 |

**Tabelle 7.4:** Ergebnisse der Evaluation der aufgestellten Modellvarianten für den Datenterm mit homogener Glattheit am KITTI-Trainingsdatensatz. Die erste Spalte benennt die Modellvarianten, die restlichen die Fehlerwerte aus Abschnitt 7.2. In der letzten Zeile ist die relative Verbesserung des für den jeweiligen Fehlertyp besten Modells zum Fehler der initialen Schätzung von RAFT-3D berechnet. Das mit Stern gekennzeichnete Modell wurde für die Parameteroptimierung herangezogen. Die niedrigsten Fehler jeder Spalte sind fettgedruckt, die zweitniedrigsten unterstrichen. Vermeintlich gleiche, zweitbeste Werte, von denen nicht alle unterstrichen sind, unterscheiden sich ungerundet nach der dritten Nachkommastelle.

Die Ergebnisse der Evaluation der aufgestellten Modellvarianten für den Datenterm des Verfeinerungsschritts mit homogener Glattheit sind in Tabelle 7.4 gelistet. Zusätzlich sind die Fehler der initialen Szenenflussschätzung aufgezeigt. Im Folgenden werden die Ergebnisse interpretiert.

Auffallend ist zunächst, dass sich mit keinem der ausgewählten Modelle eine tatsächliche Verbesserung der initialen Szenenflussschätzung erreichen lässt. Mögliche Ursache dafür könnte im Glattheitsterm begründet sein, der nur eine homogene Glattheit enthält, wodurch es zu einer zu starken Glättung von Szenenflusskanten kommt. Ob die Nutzung weiterführender Glattheitsterme eine Verbesserung erzielen kann, wird in Abschnitt 7.6 untersucht. Die erfolgreiche Verfeinerung war allerdings auch nicht primäres Ziel dieser Experimente. An dieser Stelle soll der beste Datenterm herausgearbeitet werden.

Werden die Modelle untereinander und nicht mit der initialen Schätzung von RAFT-3D verglichen, so sind die meisten Fehler des Modells *gcaColSep*, welches aus separat robustifizierter Grauwertkonstanz der Disparität und Gradientenkonstanz der HSV-Bilder besteht, minimal. Das Modell erreicht die niedrigsten Fehlerwerte aller Fehlermaßen außer in den D2-Fehlern und dem FL-Fehler des Hintergrundes. In diesen unterscheidet es sich allerdings nur geringfügig (um weniger als 0.01 Prozentpunkte) vom minimalen Wert. Der Fokus dieser Arbeit liegt auf dem Szenenfluss, weswegen das SF-Fehlermaß, welches ein Qualitätsmaß für den gesamten Szenenfluss darstellt, von besonderer Bedeutung ist. Da das Modell *gcaColSep* geringere SF-Fehler aufweist als die anderen Modelle und in den restlichen Fehlermaßen minimal oder sehr nah am Minimum ist, wird dieser Datenterm aus Grauwertkonstanz der Disparität und Gradientenkonstanz der HSV-Bilder als bester Datenterm ausgewählt. Er verschlechtert den SF-all Fehler der initialen Schätzung um weniger als 2 Prozentpunkte und damit relativ gesehen um 12.81%. Ähnliche SF-all-Fehlerwerte erzielen die Modelle *bcaGraySep*, *gcaGraySep* und *bcaGcaGraySep* mit 11.229%. Dennoch erreicht *gcaColSep* in den restlichen Fehlern leicht geringere Werte und schneidet in fünf der neun Fehlermaße am besten ab, weswegen es als beste Modellvariante des Datenterms gewählt wird.

Aus Tabelle 7.4 lassen sich außerdem weitere Ergebnisse hinsichtlich der Modellannahmen des Datenterms ableiten. Es zeigt sich deutlich, dass eine separate Robustifizierung der gemeinsamen in allen Fällen überlegen ist. Dahingegen erzielt eine gemeinsame Robustifizierung nur eine leichte Verbesserung im Vergleich zu den nicht-robustifizierten Termen. Dies liegt daran, dass die Annahmen des Datenterms nicht korreliert sind und es somit nicht gewinnbringend ist, sie gemeinsam zu bestrafen.

Eine weitere Erkenntnis ist, dass die Nutzung von Farb- statt Grauwertbildern unter Gradientenkonstanz keine relevante Verbesserung bewirkt (Vergleich Modelle $gcaGraySep$ mit $gcaColSep$ und $gcaGrayJoin$ mit $gcaColJoin$). Unter Grauwertkonstanz erzielen Grauwertbilder bessere Ergebnisse als Farbbilder mit Außnahme der D2-Fehler (Vergleich Modelle $bcaGraySep$ mit $bcaColSep$ und $bcaGrayJoin$ mit $bcaColJoin$). Auch verschiedene Kombinationen von Grauwert- und Farbbildern in Verbindung mit Grauwert- und Gradientenkonstanz erzielen keine besseren Ergebnisse als das beste Datentermmodell $gcaColSep$, obwohl eine solche Kombination mit $bcaGcaColSep$ dem Datenterm der Parameteroptimierung entspricht. Dies lässt darauf schließen, dass die gewählten Parameter nicht nur für das optimierte Modell, sondern auch für andere Modelle (hier $gcaColSep$) eine gute Wahl sind und bestätigt das gewählte Vorgehen bei der Parameteroptimierung (siehe Abschnitt 7.4) . Somit sind die hier gezeigten Ergebnisse vergleichbar, obwohl die Optimierung der Parameter nur für ein bestimmtes Modell durchgeführt wurde.

Wird die relative Verbesserung betrachtet, so erfährt die Zieldisparität eine stärkere Verschlechterung als der optische Fluss (Vergleich D2-bg, D2-fg, D2-all mit FL-bg, FL-fg, FL-all). Ebenfalls stärker wird die Schätzung des Vordergrunds im Gegensatz zu der des Hintergrunds verschlechtert (Vergleich D2-fg, FL-fg, SF-fg mit D2-bg,FL-bg, SF-bg). Eine denkbare Erklärung hierfür ist die potentiell größere Bewegung der Vordergrundobjekte, die große Pixelverschiebung verursachen. An solchen Vordergrundpositionen fällt die grundlegende homogene Glattheit besonders negativ ins Gewicht. Um das beste Gesamtmodell zu erhalten, muss also ein Glattheitsterm mit weiterführenden Annahmen eingesetzt werden.

Im nächsten Abschnitt erfolgt die Evaluation verschiedener Glattheitstermmodelle unter Einsatz des Datentermmodells *gcaColSep* aus Tabelle 7.3, das sich aus separat robustifizierter Grauwertkonstanz der Disparität und Gradientenkonstanz der HSV-Bilder zusammensetzt.

## 7.6 Auswahl des Glattheitsterms

Um nun das beste Gesamtmodell zu finden, wird für den zuvor ausgewählten Datenterm, bestehend aus separat robustifizierter Grauwertkonstanz der Disparität und Gradientenkonstanz der HSV-Bilder (*gcaColSep*), ein geeigneter Glattheitsterm selektiert. Dabei wird analog zum Vorgehen des vorherigen Abschnitts, basierend auf Erfahrungen der Arbeitsgruppe und Veröffentlichungen, eine Vorauswahl von vielversprechenden Glattheitstermmodellen getroffen, die in Kombination mit diesem Datenterm anhand des KITTI-Trainingsdatensatzes evaluiert wird.

Die erarbeitete Implementierung ermöglicht eine beliebige Zusammensetzung des Glattheitsterms aus bild- und flussgetriebener sowie jeweils isotroper und anisotroper Glattheit mit einer freien Kombination aus treibenden Domänen für den optischen Fluss und für die Zieldisparität (siehe Kapitel 6). Die ausgewählten Modellkombinationen, für die im Rahmen dieser Arbeit eine Bewertung durchgeführt wurde, sind nach Annahmen sortiert in Tabelle 7.5 aufgelistet. Zusätzlich befindet sich im Anhang in Tabelle A.1 eine erweiterte Darstellung mit Berechnung der Diffusionstensoren für die Glattheit vom optischen Fluss und der Zieldisparität jeder Modellvariante.

Für die Modellnamen, die sich jeweils aus drei Kürzeln zusammensetzen, wurde folgende Konvention gewählt, anhand derer die Zusammensetzung des jeweiligen Glattheitsterms erkennbar ist:

- Erstes Kürzel: Glattheitstyp
  - *img*: bildgetriebene Glattheit
  - *flow*: flussgetriebene Glattheit
- Zweites Kürzel: Isotropie
  - *iso*: isotrope Glattheit
  - *aniso*: anisotrope Glattheit
- Drittes Kürzel: treibende Domänen
  - *sep*: unterschiedliche treibende Domänen von optischem Fluss und Zieldisparität
    - bildgetrieben: $f$ für den optischen Fluss und $\zeta$ für die Zieldisparität

    - flussgetrieben: der optische Fluss $\overline{u}, \overline{v}$ für sich selbst und die Zieldisparität $\overline{d}$ ebenfalls für sich selbst

- *join*: kombinierte treibende Domänen für den optischen Fluss und die Zieldisparität
    - bildgetrieben: $f$ und $\zeta$
    - flussgetrieben: $\overline{u}, \overline{v}$ und $\overline{d}$
- *img*: bildgetrieben durch $f$
- *disp*:
    - bildgetrieben: $\zeta$
    - flussgetrieben: $\overline{d}$
- *flow*: flussgetrieben durch $\overline{u}, \overline{v}$.

Außerdem wird als Bestrafungsfunktion der Robustifizierung die Charbonnier-Bestrafung aus Gleichung (5.25) eingesetzt. Modelle, die weder bild- noch flussgetrieben sind, entsprechen der homogenen Glattheit.

| | **Treibende Domänen von** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Optischer Fluss** | | | | **Zieldisparität** | | | | **Isotropie** | |
| Modelle | $f$ | $\zeta$ | $\overline{u}, \overline{v}$ | $\overline{d}$ | $f$ | $\zeta$ | $\overline{u}, \overline{v}$ | $\overline{d}$ | isotrop | anisotrop |
| homogen | - | - | - | - | - | - | - | - | - | - |
| imgIsoSep | ✓ | - | - | - | - | ✓ | - | - | ✓ | - |
| imgIsoJoin | ✓ | ✓ | - | - | ✓ | ✓ | - | - | ✓ | - |
| imgIsoImg | ✓ | - | - | - | ✓ | - | - | - | ✓ | - |
| imgIsoDisp | - | ✓ | - | - | - | ✓ | - | - | ✓ | - |
| imgAnisoSep | ✓ | - | - | - | - | ✓ | - | - | - | ✓ |
| imgAnisoJoin | ✓ | ✓ | - | - | ✓ | ✓ | - | - | - | ✓ |
| imgAnisoImg | ✓ | - | - | - | ✓ | - | - | - | - | ✓ |
| imgAnisoDisp | - | ✓ | - | - | - | ✓ | - | - | - | ✓ |
| flowIsoSep | - | - | ✓ | - | - | - | - | ✓ | ✓ | - |
| flowIsoJoin | - | - | ✓ | ✓ | - | - | ✓ | ✓ | ✓ | - |
| flowIsoFlow | - | - | ✓ | - | - | - | ✓ | - | ✓ | - |
| flowIsoDisp | - | - | - | ✓ | - | - | - | ✓ | ✓ | - |
| flowAnisoSep* | - | - | ✓ | - | - | - | - | ✓ | - | ✓ |
| flowAnisoJoin | - | - | ✓ | ✓ | - | - | ✓ | ✓ | - | ✓ |
| flowAnisoFlow | - | - | ✓ | - | - | - | ✓ | - | - | ✓ |
| flowAnisoDisp | - | - | - | ✓ | - | - | - | ✓ | - | ✓ |

**Tabelle 7.5:** Übersicht über die verschiedenen Modelle des Glattheitsterms. Die Spalten ‚Optischer Fluss' und ‚Zieldisparität' beschreiben jeweils die möglichen treibenden Domänen: $f$ steht für die Bildfolge, $\zeta$ für die Disparitätskarte, $\overline{u}, \overline{v}$ für den optischen Fluss und $\overline{d}$ für die Zieldisparität. Das mit Stern gekennzeichnete Modell wurde für die Parameteroptimierung herangezogen.

| Modell | D2-bg | D2-fg | D2-all | FL-bg | FL-fg | FL-all | SF-bg | SF-fg | SF-all |
|---|---|---|---|---|---|---|---|---|---|
| RAFT-3D [77] | **3.682** | **0.420** | **4.101** | **7.741** | **1.980** | **9.721** | **9.344** | **13.307** | **9.951** |
| homogen | 4.258 | 0.611 | 4.870 | 8.268 | 2.504 | 10.773 | 10.217 | 16.804 | 11.226 |
| imgIsoSep | 4.084 | 0.550 | 4.634 | 8.223 | 2.487 | 10.710 | 10.087 | 16.650 | 11.092 |
| imgIsoJoin | 4.757 | 1.013 | 5.770 | 8.182 | 2.474 | 10.655 | 10.591 | 18.157 | 11.750 |
| imgIsoImg | 4.779 | 1.014 | 5.793 | 8.206 | 2.472 | 10.678 | 10.620 | 18.133 | 11.771 |
| imgIsoDisp | 4.085 | 0.551 | 4.636 | 8.147 | 2.406 | 10.554 | 9.994 | 16.118 | 10.932 |
| imgAnisoSep | 3.962 | 0.471 | 4.432 | 8.014 | 2.220 | 10.234 | 9.761 | 14.827 | 10.537 |
| imgAnisoJoin | 4.021 | 0.481 | 4.502 | 7.970 | 2.205 | 10.175 | 9.732 | 14.773 | 10.504 |
| imgAnisoImg | 4.076 | 0.492 | 4.568 | 8.017 | 2.219 | 10.236 | 9.814 | 14.888 | 10.591 |
| imgAnisoDisp | 3.962 | 0.471 | 4.432 | 7.912 | 2.161 | 10.073 | 9.617 | 14.435 | 10.355 |
| flowIsoSep | 3.907 | 0.444 | 4.351 | 7.929 | 2.157 | 10.086 | 9.653 | 14.411 | 10.382 |
| flowIsoJoin | 3.921 | 0.479 | 4.400 | 7.921 | 2.188 | 10.109 | 9.641 | 14.618 | 10.404 |
| flowIsoFlow | 4.024 | 0.525 | 4.549 | 7.929 | 2.153 | 10.082 | 9.762 | 14.512 | 10.490 |
| flowIsoDisp | 3.908 | 0.444 | 4.352 | 8.043 | 2.384 | 10.427 | 9.782 | 15.871 | 10.715 |
| flowAnisoSep* | 3.860 | 0.428 | 4.287 | 7.884 | 2.101 | 9.985 | 9.580 | 14.043 | 10.264 |
| flowAnisoJoin | <u>3.849</u> | 0.429 | <u>4.278</u> | <u>7.869</u> | <u>2.096</u> | <u>9.965</u> | <u>9.558</u> | <u>14.008</u> | <u>10.240</u> |
| flowAnisoFlow | 3.959 | 0.467 | 4.426 | 7.884 | 2.101 | 9.985 | 9.689 | 14.158 | 10.374 |
| flowAnisoDisp | 3.860 | <u>0.427</u> | 4.287 | 7.996 | 2.334 | 10.330 | 9.709 | 15.533 | 10.601 |
| Relative Verbesserung in % | -4.54 | -1.76 | -4.31 | -1.66 | -5.83 | -2.51 | -2.29 | -5.27 | -2.90 |

**Tabelle 7.6:** Ergebnisse aller ausgewählten Glattheitsterme mit Datenterm aus separat robustifizierte Gradientenkonstanz der Farbbilder und Grauwertkonstanz der Disparität (*gcaColSep*) am KITTI-Trainingsdatensatz evaluiert. In der letzten Zeile ist die relative Verbesserung des für den jeweiligen Fehlertyp besten Modells zum Fehler der initialen Schätzung von RAFT-3D berechnet. Das mit Stern gekennzeichnete Modell wurde für die Parameteroptimierung herangezogen. Die niedrigsten Fehler jeder Spalte sind fettgedruckt, die zweitniedrigsten unterstrichen. Vermeintlich gleiche, zweitbeste Werte, von denen nicht alle unterstrichen sind, unterscheiden sich ungerundet nach der dritten Nachkommastelle.

Die Ergebnisse der Evaluation der aufgestellten Modellvarianten für den Glattheitsterm des Verfeinerungsschritts sind in Tabelle 7.6 gelistet. Zusätzlich sind die Fehler der initialen Szenenflussschätzung aufgezeigt. Im Folgenden werden die Ergebnisse interpretiert.

Auffallend ist wieder, dass sich mit keinem der ausgewählten Modelle eine tatsächliche Verbesserung der initialen Schätzung erreichen lässt. Die meisten Modelle erzielen jedoch einen geringeren Fehler als der homogene Fall, welcher exakt dem besten Modell aus dem vorherigen Abschnitt entspricht.

Beim Vergleich der Modelle untereinander, erzielt das Modell *flowAnisoJoin*, das aus flussgetriebener anisotroper Glattheit mit kombinierten treibenden Domänen besteht, bis auf im D2-Fehler des Vordergrundes die geringsten Fehlerwerte. In diesen unterscheidet es sich allerdings nur geringfügig (weniger als 0.01 Prozentpunkte) vom minimalen Wert. Besagtes Modell weist geringere SF-Fehler als die anderen auf und ist in den restlichen Fehlermaßen minimal oder sehr nah am Minimum. Insgesamt betrachtet lässt sich daraus das beste Modell ableiten, welches aus dem zuvor gewählten Datenterm mit Grauwertkonstanz der Disparität und Gradientenkonstanz der HSV-Bilder (*gcaColSep*) und einem Glattheitsterm aus flussgetriebener anisotroper Glattheit mit kombinierten treibenden Domänen (*flowAnisoJoin*) besteht. Es verschlechtert den SF-all Fehler der initialen Schätzung nur noch um weniger als 0.5 Prozentpunkte, relativ gesehen um 2.9%, und stellt damit eine eindeutige Verbesserung zum homogenen Modell des selben Datenterms dar. Ähnliche SF-all-Fehlerwerte erzielt das Modell *flowAnisoSep* mit 10.264%. Dennoch erreicht *flowAnisoJoin* in acht der neun Fehlermaße die niedrigsten Fehler, weswegen es als beste Modellvariante des Glattheitsterms gewählt wird.

Aus Tabelle 7.6 lassen sich außerdem weitere Ergebnisse hinsichtlich der Modellannahmen des Glattheitsterms ableiten. Zunächst soll der Einfluss von Isotropie untersucht werden: Anisotrope bildgetriebene Modelle (*imgAnisoSep* bis *imgAnisoDisp*) erreichen immer geringere Fehler als die isotropen bildgetriebenen (*imgIsoSep* bis *imgIsoDisp*). Anisotrope flussgetriebene Modelle (*flowAnisoSep* bis *flowAnisoDisp*) erzielen größtenteils bessere Ergebnisse als isotrope flussgetriebene (*flowIsoSep* bis *flowIsoDisp*). Damit stellt sich anisotrope Glattheit gegenüber der isotropen als überlegen heraus. Dies entspricht den Ergebnissen von Maurer *et al.* [46] und ist nachvollziehbar, da im anisotropen Fall im Gegensatz zum isotropen auch parallel zu den Kanten geglättet wird, wodurch die vorausgesetzte Glattheit nur orthogonal zu den Kanten gelockert wird.

Als Nächstes soll die Auswirkung des Glattheitstyps analysiert werden. Flussgetriebene Modelle (*flowIsoSep* bis *flowAnisoDisp*) erreichen im Vergleich zu bildgetriebenen (*imgIsoSep* bis *imgAnisoDisp*) mehrheitlich bessere Ergebnisse. Dies stimmt mit dem aktuellen Stand der Forschung bezüglich der Glattheitstypen überein, denn flussgetriebene Methoden haben sich gegenüber bildgetriebenen durchgesetzt. Letztgenannte haben den Nachteil, dass nicht alle Bildkanten gleichzeitig Szenenflusskanten sind, sodass es zu Übersegmentierung kommt. Flussgetriebene Methoden haben dieses Problem nicht, die Kanten sind jedoch nicht so gut lokalisiert, da sie erst während der Schätzung entstehen. Dieser Nachteil ist für eine Verfeinerung mit guter initialer Schätzung, wie es hier der Fall ist, stark gemildert, da die Szenenflusskanten schon gegeben sind.

Zuletzt soll der Effekt der treibenden Domänen untersucht werden. Wie bereits festgestellt wurde, erzielen flussgetriebene Modelle mehrheitlich bessere Ergebnisse als bildgetriebene. Da flussgetriebenen Varianten die treibenden Domänen optischer Fluss $(\overline{u}, \overline{v})$ und Zieldisparität $\overline{d}$ zugrunde liegen, sind diese zwei besser als die treibenden Domänen Bild $f$ und Disparitätskarte $\zeta$ der bildgetriebenen Modelle. Innerhalb der bildgetriebenen Modelle stellt sich die Disparität $\zeta$ alleine (*imgIsoDisp*, *imgAnisoDisp*) als ein besserer Indikator für Flusskanten heraus, als in Kombination

mit dem Bild $f$ (*imgIsoJoin*, *imgAnisoJoin*) oder dieses alleine (*imgIsoImg*, *imgAnisoImg*). Dies gilt mehrheitlich auch im Vergleich zu separaten treibenden Domänen (*imgIsoSep*, *imgAnisoSep*). Das lässt sich damit erklären, dass $\zeta$ viel weniger Kanten als $f$ hat, die bei einer guten Schätzung von $\zeta$ mit den Szenenflusskanten übereinstimmen. Innerhalb der flussgetriebenen Modelle erzielen die Varianten mit der Zieldisparität $\overline{d}$ als alleinige treibende Domäne die schlechtesten Ergebnisse (*flowIsoDisp* und *flowAnisoDisp*). Während die Disparitätskarte $\zeta$ gegenüber dem Bild $f$ den Vorteil einer geringeren Kantenanzahl aufweist, ist dies für Zieldisparität $\overline{d}$ gegenüber dem optischen Fluss $(\overline{u}, \overline{v})$ nicht der Fall. Für die flussgetriebene isotrope Glattheit kann keine beste Kombination von treibenden Domänen gefunden werden (*flowIsoSep* bis *flowIsoDisp*). Dahingegen stellt sich der optische Fluss in Verbindung mit der Zieldisparität (*flowAnisoJoin*) als bester Indikator der flussgetriebenen anisotropen Glattheit (*flowAnisoSep* bis *flowAnisoDisp*) heraus.

Wird die relative Verbesserung betrachtet, so erfährt die Zieldisparität im Hintergrund und im gesamten Bild, wie in Abschnitt 7.5 , eine stärkere Verschlechterung als der optische Fluss (Vergleich D2-bg, D2-all mit FL-bg, FL-all). Bezüglich des Szenenflusses und des optischen Flusses wird die Schätzung des Vordergrunds weiterhin stärker als die des Hintergrunds verschlechtert (Vergleich SF-fg, FL-fg mit SF-bg, FL-bg). Im Gegensatz zu den Ergebnissen der Datentermevaluation in Abschnitt 7.5 wird die Schätzung des optischen Flusses im Vordergrund stärker verschlechtert als die der Zieldisparität (Vergleich FL-fg mit D2-fg). Zur Klärung dieser Feststellung können im Rahmen zukünftiger Arbeiten weitere Analysen durchgeführt werden.

Zusammenfassend kann festgestellt werden, dass keines der aufgestellten Modelle für den Verfeinerungsschritt zu einer Verbesserung der initiale Szenenflussschätzung von RAFT-3D führt, wenn der gesamte KITTI-Trainingsdatensatz betrachtet wird. Jedoch könnten eine umfassendere Parameteroptimierung und die Hinzunahme weiterer Konstanzannahmen bezüglich der Zieldisparität zu einer Verbesserung führen. Darauf weist die Betrachtung von einzelnen Sequenzen hin, deren initiale Szenenflussschätzung erfolgreich verbessert werden konnte. Daher werden im Folgenden für das ermittelte Gesamtmodell, dessen Datenterm *gcaColSep* sich aus separat robustifizierter Grauwertkonstanz der Disparität und Gradientenkonstanz der HSV-Bilder und dessen Glattheitsterm *flowAnisoJoin* sich aus flussgetriebener anisotroper Glattheit mit kombinierten treibenden Domänen zusammensetzen, weitere Analysen durchgeführt, um das Potential der in dieser Arbeit entwickelten Verfeinerungsmethode zu zeigen.

## 7.7 Iterationsabhängige Fehleranalyse

Das im vorherigen Abschnitt ermittelte Gesamtmodell (*gcaColSep* mit *flowAnisoJoin*) konnte die über den gesamten KITTI-Traininigsdatensatz evaluierten, durchschnittlichen Fehlerwerte der initialen Szenenflussschätzung nicht verbessern. Werden jedoch einzelne Sequenzen betrachtet, können tatsächliche Verbesserungen gefunden werden. Die Anzahl der verbesserten Sequenzen soll in diesem Abschnitt im Rahmen einer iterationsabhängigen Fehleranalyse untersucht werden.

Zunächst soll für die Fehlermaße SF-all, FL-all und D2-all untersucht werden, wie viele Sequenzen nach jeder Iteration eine Verbesserung gegenüber der initialen Schätzung aufweisen. Dies ist in Abbildung 7.4 gezeigt. Es kann beobachtet werden, dass nach der ersten Iteration jeweils über 40 Sequenzen verbessert wurden. Über alle Iterationen hinweg liegt die Anzahl der verbesserten Sequenzen beim D2-all-Fehler deutlich unter der Anzahl beim FL-all-Fehler, weswegen die Anzahl beim SF-all Fehler ebenfalls niedriger ist. Insgesamt betrachtet sinkt die Anzahl der verbesserten

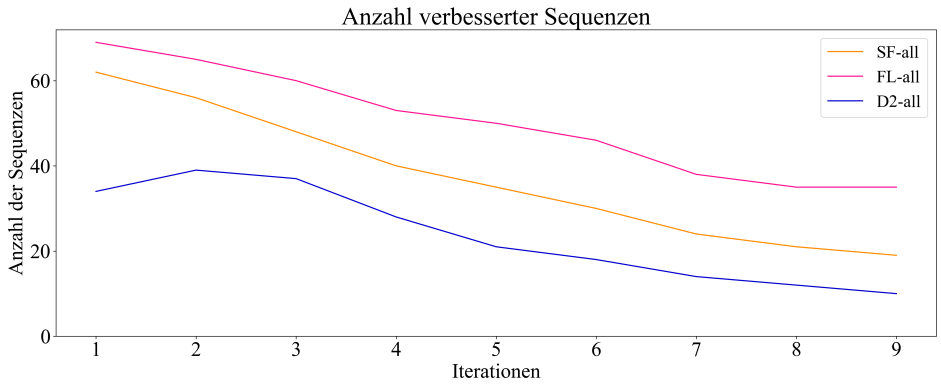



**Abbildung 7.4:** Verlauf der Anzahl der in den Gesamtbildfehlern SF-all, FL-all und D2-all verbesserten Bilder über die Iterationen.

Sequenzen mit der Anzahl der Iterationen. Dennoch sind nach neun Iterationen weiterhin Sequenzen mit einer verbesserten Szenenflussschätzung vorhanden. Das Maximum der Anzahl verbesserter Sequenzen liegt dabei nach der ersten Iteration vor, außer beim Disparitätsfehler D2-all, für den erst nach der zweiten Iteration das Maximum erreicht wird. Vorerst scheint es deshalb so, als sei die gewählte Iterationsanzahl zu hoch, da die Anzahl der Sequenzen, die nach der ersten bzw. zweiten Iteration tatsächlich verbessert wurden, mit weiteren Iterationen abnimmt.

An dieser Stelle ist nicht nur die Anzahl der verbesserten Sequenzen von Interesse, sondern auch die Verteilung der Fehler, wie in Abbildung 7.5 anhand der relativen Verbesserung aus Gleichung (7.9) von SF-all, FL-all und D2-all dargestellt wird. Sequenzen, die verbessert wurden, weisen eine positive relative Verbesserung auf (in Abbildung 7.5 grün), wohingegen eine Verschlechterung durch eine negative relative Verbesserung gekennzeichnet wird (in Abbildung 7.5 orange, rot, dunkelrot). Im Detail beschreibt Abbildung 7.5 wie sich die relativen Verbesserungen über die Iterationen verändern, wobei die genannten Farben den folgenden Bereichen zugeordnet werden: grün steht für die Anzahl an Sequenzen, die verbessert wurden (relative Verbesserung größer 0), orange für die Anzahl an Sequenzen, die sich leicht verschlechtert haben (relative Verbesserung zwischen 0 und -10), rot für die Anzahl an Sequenzen, die sich stark verschlechtert haben (relative Verbesserung zwischen -10 und -30) und dunkelrot für die Anzahl an Sequenzen, die sich gravierend verschlechtert haben (relative Verbesserung kleiner -30). Auch hier ist erkennbar, dass die Anzahl der verbesserten Sequenzen in allen Fehlermaßen mit zunehmender Iterationszahl sinkt. Zusätzlich ist ersichtlich, dass die Anzahl der Sequenzen, die sich gravierend verschlechtern (relative Verbesserung kleiner -30), steigt. In besonderem Maße kann dies bei der Zieldisparität beobachtet werden. Diese Erkenntnis deutet darauf hin, dass eine Modellierung der Zieldisparität mit weiterführenden Annahmen für eine Verbesserung der Szenenflussverfeinerung notwendig ist. Außerdem scheint die gewählte Anzahl der Iterationen für das aktuell beste Modell, angewandt auf den gesamten KITTI-Trainingsdatensatz, erneut zu hoch zu sein, womit ekrklärt werden kann, warum sich die Durchschnittsfehler über die Iterationen hinweg stetig verschlechtern. Letztgenanntes ist in Abbildung 7.6, die den Verlauf der Gesamtbildfehler SF-all, FL-all und D2-all über die Iterationen hinweg zeigt, zu sehen. Das Minimum der durchschnittlichen Fehler von SF-all, FL-all und D2-all liegt nach der ersten Iteration vor und verschlechtert bereits die initialen Fehler.

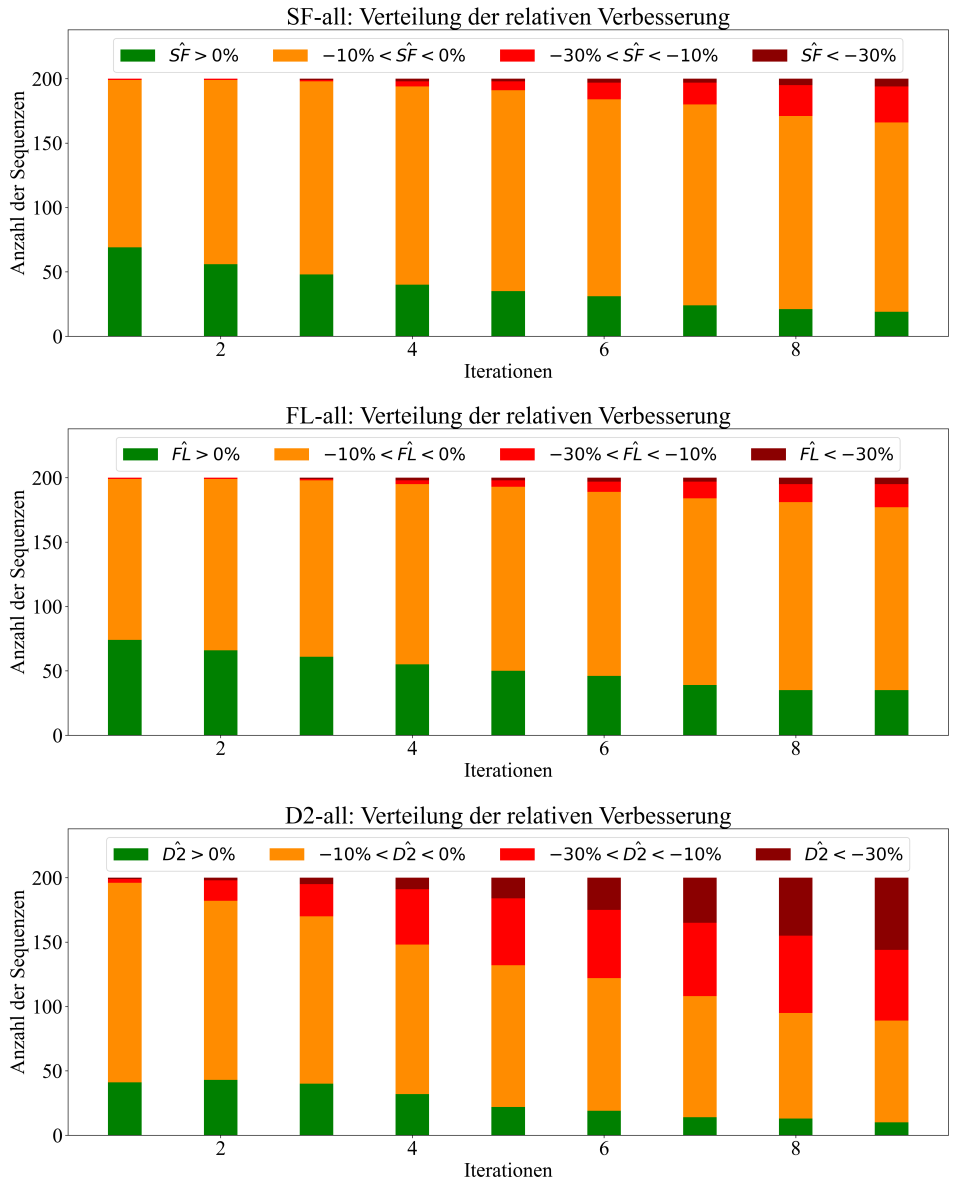



**Abbildung 7.5:** Aufteilung der relativen Verbesserung der Gesamtbildfehler SF-all, FL-all und D2-all über die Iterationen.

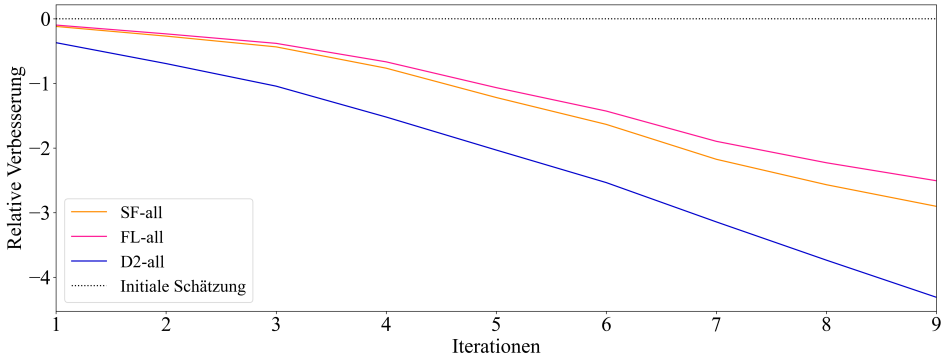



**Abbildung 7.6:** Verlauf der Gesamtbildfehler SF-all, FL-all und D2-all über die Iterationen.

Bisher wurden die Gesamtbildfehler SF-all, FL-all und D2-all unabhängig voneinander betrachtet. Werden nun pro Iteration die Anzahl der Sequenzen summiert, die gleichzeitig in allen Gesamtbildfehlern eine Verbesserung erfahren, so ergibt sich der in Abbildung 7.7 dargestellte Verlauf. Nach der ersten Iteration gibt es neun Sequenzen, die sich in allen Fehlern verbessert haben. Nach der siebten Iteration bleibt immer noch eine Sequenz – Nummer 171 – übrig, welche auch nach vollständigem Durchlaufen des Verfahrens mit neun Iterationen eine gesamtheitliche Verfeinerung aufweist. In Kombination mit Abbildung 7.4, bei der ersichtlich ist, dass nach neun Iterationen in jedem Fehler jeweils über zehn verbesserte Sequenzen verbleiben, kann Folgendes geschlossen werden: Die Szenenflussverfeinerung verbessert entweder den optischen Fluss und verschlechtert dabei die Zieldisparität oder umgekehrt. Beides gleichzeitig gelingt nur in Sequenz 171. Die Abbildung 7.7 legt aufgrund der stetig sinkenden Anzahl der in allen Gesamtbildfehlern verbesserten Sequenzen nahe, dass die Liste der nach dem ersten Iterationsschritt verbesserten Sequenzen pro weiterer Iteration schrumpft. Das ist nur zum Teil der Fall. Es gibt allerdings Sequenzen, die erst nach der zweiten bzw. vierten Iteration (Sequenzen 0, 87, 92, 144) zum ersten Mal eine gleichzeitige Verbesserung in allen Gesamtbildfehlern erzielen. Die pro Iteration konkret verbesserten Sequenzen sind in Tabelle 7.7 aufgeführt.

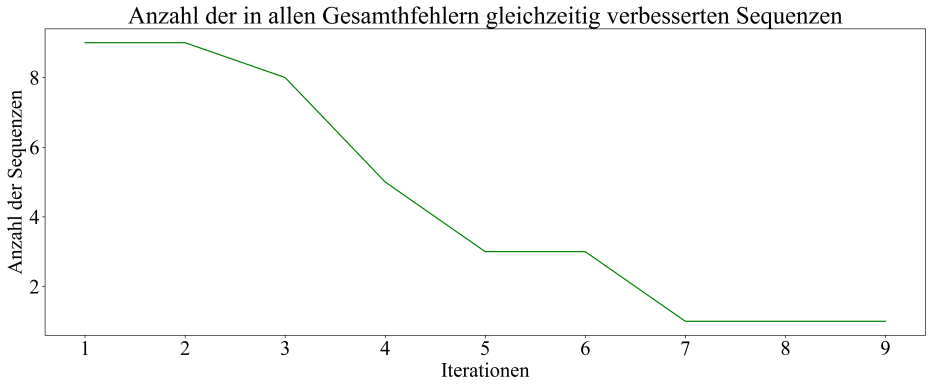



**Abbildung 7.7:** Verlauf der Anzahl der in allen Gesamtbildfehlern (SF-all, FL-all und D2-all) gleichzeitig verbesserten Bilder über die Iterationen.

| Sequenz | Iteration | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | | ✓ | ✓ | | | | | | |
| 38 | ✓ | | | | | | | | |
| 51 | ✓ | ✓ | ✓ | ✓ | | | | | |
| 87 | | ✓ | ✓ | ✓ | | | | | |
| 92 | | | | ✓ | ✓ | ✓ | | | |
| 144 | | ✓ | | | | | | | |
| 150 | ✓ | ✓ | ✓ | | | | | | |
| 152 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| 155 | ✓ | ✓ | ✓ | | | | | | |
| 171 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 189 | ✓ | | | | | | | | |
| 191 | ✓ | | | | | | | | |
| 193 | ✓ | ✓ | ✓ | | | | | | |

**Tabelle 7.7:** Auflistung der konkret in allen Gesamtbildfehlermaßen (SF-all, C2-all, FL-all) gleichzeitig verbesserten Sequenzen pro Iterationsschritt.

Wie der obige Abschnitt gezeigt hat, wirkt der Verfeinerungsschritt nicht für alle Sequenzen und alle Fehlermaße gleichermaßen. Deshalb sollen im Folgenden einzelne Sequenzen, die in besonderem Maße herausstechen, analysiert werden.

## 7.8 Analyse von Einzelsequenzen

Für die Analyse von Einzelsequenzen wurden die nachstehenden vier Sequenzen ausgewählt: 171 als Sequenz, die über alle Iterationsschritte hinweg in allen Fehlermaßen verbessert wurde; 51 als Sequenz die zunächst verbessert, später jedoch verschlechtert wurde; 0 als Sequenz, die nach einer anfänglichen Verschlechterung erst nach der zweiten Iteration eine Verbesserung erfährt und 133 als Sequenz mit höchstem finalen SF-all-Fehler. Zu jeder dieser Sequenzen sollen zwei Abbildungen präsentiert werden. Zum einen wird eine Visualisierung der Szenenflussverfeinerung – bestehend aus Eingabegrößen (Eingabebild und Disparitätskarte) zum Zeitpunkt $t$, initialem Szenenfluss, verfeinertem Szenenfluss, deren Differenz (zur besseren Sichtbarkeit mit einem Faktor von zehn multipliziert), den Grundwahrheitswerten sowie der Endpunkt-Verbesserung, jeweils in optischer Fluss und Disparität aufgeteilt – gezeigt. Zum anderen wird der Verlauf der Fehler SF-all, FL-all und D2-all jeweils für jede ausgewählte Sequenz über die Iterationen dargestellt.

Für erstgenannte Abbildungen findet die Farbkodierung der optischen Flussvisualisierungen im HSV-Raum statt, entsprechend der bei Bruhn [11] vorgestellten Strategie. Dabei repräsentiert der V-Kanal die optische Flussvektorlänge und der H-Kanal die Flussvektorrichtung während der S-Kanal konstant auf eins gehalten wird. Die Visualisierungen der Endpunkt-Verbesserungen stellen die Veränderung der Endpunktfehler pro Pixel zwischen initialer Schätzung und Verfeinerung dar. Für den optischen Fall entstammt der Endpunktfehler Gleichung (7.1), für die Zieldisparität aus

Gleichung (7.3). Die Farbkodierung ist in Abbildung 7.8 gezeigt. Pixel mit einem verbesserten Endpunktfehler werden in blau, Pixel mit einem verschlechterten Endpunktfehler in rot dargestellt.

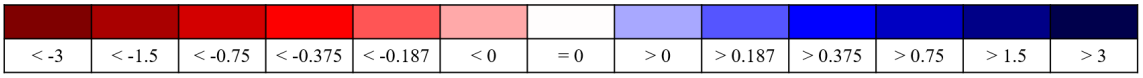



**Abbildung 7.8:** Farbkodierung der Visualisierung der Endpunkt-Verbesserung. Pixel mit einem verbesserten Endpunktfehler werden in blau, Pixel mit einem verschlechterten Endpunktfehler in rot dargestellt.

### Sequenz 171

Als erste Sequenz wird Nummer 171 untersucht. Abbildung 7.13 zeigt die Visualisierung der Szenenflussverfeinerung dieser Sequenz. Zunächst fällt auf, dass die Verfeinerung hauptsächlich an den Kanten der Objekte stattfindet (siehe optischer Differenzfluss und Differenzdisparität). Beim optischen Fluss wirkt diese an den Kanten hauptsächlich verschlechternd (siehe Endpunkt-Verbesserung des optischen Flusses), während die Kanten der Zieldisparität zum Teil verbessert werden (siehe Endpunkt-Verbesserung der Zieldisparität). Innerhalb des abgebildeten Fahrzeugs verschlechtern sich die Endpunktpositionen des optischen Flusses leicht, dahingegen verbessern sich die der Zieldisparität großflächig. Im Hintergrundbereich gibt es in beiden Szenenflusskomponenten sowohl leichte Verbesserungen als auch leichte Verschlechterungen.

In Abbildung 7.9 wird die relative Verbesserung der Fehlermaße SF-all, FL-all und D2-all von Sequenz 171 über die Iterationen dargestellt. Wie in Tabelle 7.7 ist ersichtlich, dass alle Fehlermaße über alle Iterationen hinweg eine Verbesserung gegenüber den initialen Fehlern erzielen. In besonderem Maße fällt auf, dass der D2-all-Fehler sich deutlich stärker verbessert, als der FL-all-Fehler und bei Iteration vier und fünf eine relative Verbesserung von über 30% erreicht. Dies spiegelt die Interpretation aus Abschnitt 7.7 wider, laut der die Iterationsanzahl neun zu hoch ist.

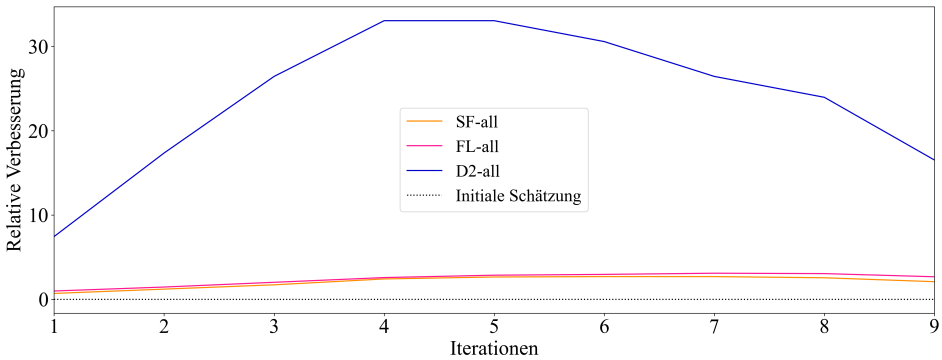



**Abbildung 7.9:** Verlauf der relativen Verbesserung von SF-all, FL-all und D2-all von Sequenz 171 über die Iterationen.

### Sequenz 51

Als zweite Sequenz wird Nummer 51 analysiert. Sie wird zunächst in allen Gesamtbildfehlermaßen gleichzeitig verbessert, jedoch ab Iteration fünf nicht mehr (siehe Tabelle 7.7). Damit unterscheidet sie sich von Sequenz 171. Die Visualisierung der Szenenflussverfeinerung von Sequenz 51 wird in Abbildung 7.14 dargestellt. Zunächst kann wieder beobachtet werden, dass die Verfeinerung hauptsächlich an den Kanten der Objekte stattfindet (siehe optischer Differenzfluss und Differenzdisparität). Innerhalb der drei abgebildeten Fahrzeuge verschlechtern sich die Endpunktpositionen des optischen Flusses wieder leicht, wobei die Verschlechterung mit dem Abstand zur Kamera auffallend zunimmt. Die Endpunktpositionen der Zieldisparität verschlechtern sich im vorderen Fahrzeug stärker als beim optischen Fluss – dies war bei Sequenz 171 umgekehrt. Für die beiden hinteren Fahrzeuge von Sequenz 51 verbessern sich die Endpunktpositionen der Zieldisparität deutlich. Im Hintergrundbereich gibt es in der Zieldisparität sowohl leichte Verbesserungen als auch leichte Verschlechterungen, im optischen Fluss verändert sich der Hintergrund kaum.

In Abbildung 7.10 wird die relative Verbesserung der Fehlermaße SF-all, FL-all und D2-all von Sequenz 51 über die Iterationen dargestellt. Auffallend ist, dass bis einschließlich Iteration vier alle Gesamtbildfehlermaße gegenüber den initialen Fehlern verbessert werden. Ab Iteration fünf verschlechtert sich der D2-all-Fehler jedoch drastisch um ca. 30%. Dies spiegelt, wie schon in Abschnitt 7.7 und bei Sequenz 171 festgestellt, die Interpretation wider, dass die Iterationsanzahl neun zu hoch ist.

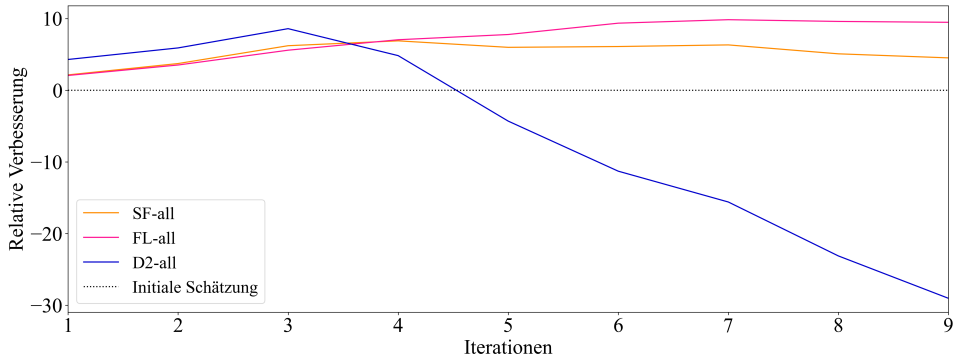



**Abbildung 7.10:** Verlauf der relativen Verbesserung von SF-all, FL-all und D2-all von Sequenz 51 über die Iterationen.

### Sequenz 0

Als dritte Sequenz wird Nummer 0 untersucht. Sie wird nicht wie Sequenz 171 und 51 bereits nach der ersten Iteration in allen Gesamtbildfehlermaßen gleichzeitig verbessert, sondern erst ab Iteration zwei (siehe Tabelle 7.7). Damit stellt sie mit drei weiteren Sequenzen (Nummer 87, 92 und 144) eine Ausnahme dar und ist von besonderem Interesse. Die Visualisierung der Szenenflussverfeinerung von Sequenz 0 wird in Abbildung 7.15 dargestellt. Es ist deutlich zu erkennen, dass sich die Schätzung des optischen Flusses über das gesamte Bild leicht verschlechtert, besonders im Inneren

des abgebildeten Fahrzeugs (siehe Endpunkt-Verbesserung des optischen Flusses). Dahingegen verbessert sich die Schätzung der Zieldisparität stark, vor allem im Inneren des Fahrzeugs (siehe Endpunkt-Verbesserung der Zieldisparität).

Wird Abbildung 7.11 betrachtet, in der die relative Verbesserung der Fehlermaße SF-all, FL-all und D2-all von Sequenz 0 über die Iterationen dargestellt ist, lässt sich vorherige Beobachtung bestätigen. Zusätzlich kann abgelesen werden, dass sich der D2-all-Fehler kontinuierlich verbessert. Der FL-all-Fehler ist nach der ersten Iteration zunächst verschlechtert, nach Iteration zwei bis vier verbessert und ab Iteration fünf wieder verschlechtert. Zur Ursachenklärung kann im Rahmen einer zukünftigen Arbeit eine weiterführende Analyse durchgeführt werden.

Beim Vergleich der Sequenzen 0 und 51 fällt auf, dass nach neun Iterationen entweder der D2-all-Fehler oder der FL-all-Fehler verbessert wurde. Dies bestärkt die Vermutung aus Abschnitt 7.7, laut der – mit Ausnahme von Sequenz 171 – nur eines der beiden Fehlermaße verbessert werden kann.

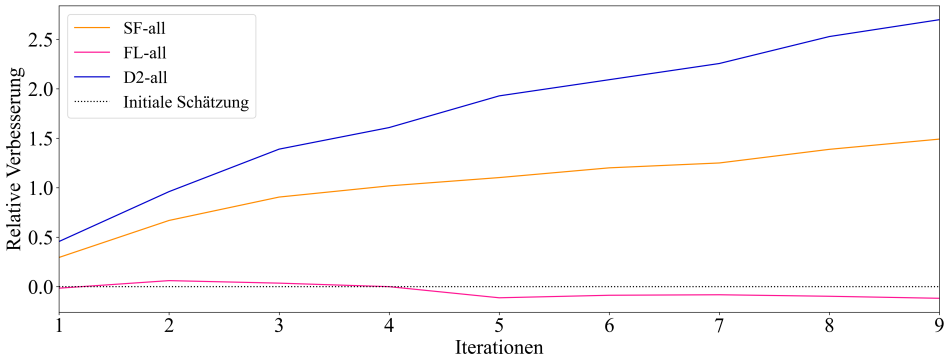



**Abbildung 7.11:** Verlauf der relativen Verbesserung von SF-all, FL-all und D2-all von Sequenz 0 über die Iterationen.

### Sequenz 133

Zuletzt wird Sequenz Nummer 133 analysiert, da es nach neun Iterationen die größte relative Verschlechterung im SF-all-Fehler erfährt. Abbildung 7.16 zeigt die Visualisierung der Szenenflussverfeinerung dieser Sequenz. In besonderem Maße fällt auf, dass der optische Fluss sehr stark und über das gesamte Bild gleichermaßen verschlechtert wurde (siehe Endpunkt-Verbesserung des optischen Flusses). Die Ursache dafür könnte die Grundbewegung der Szenerie in dieser Sequenz sein, die für große Pixelsprünge sorgt und damit die Schätzung erschwert. Dies kann anhand des optischen Differenzflusses gesehen werden, welcher eine konstante Verschiebung über alle Pixel, mit Ausnahme der Fahrzeugränder, auf den initialen optischen Fluss aufaddiert. Die Zieldisparität verschlechtert sich vor allem im Hintergrund, während im Inneren des Fahrzeugs verbesserte Endpunktpositionen berechnet wurden (siehe Endpunkt-Verbesserung der Zieldisparität).

In Abbildung 7.12 wird die relative Verbesserung der Fehlermaße SF-all, FL-all und D2-all von Sequenz 133 über die Iterationen dargestellt. Die relative Verschlechterung der Schätzung des optischen Flusses ist hier ebenfalls klar erkennbar und beträgt nach Iteration neun mehr als 350%.

Im Vergleich dazu verschlechtert sich die Schätzung der Zieldisparität nur leicht, weswegen davon auszugehen ist, dass der FL-all-Fehler für die Verschlechterung im SF-all-Fehler verantwortlich ist. Wie bei Sequenz 0 und 51 bereits beobachtet, korrelieren auch hier der FL-all- und D2-all-Fehler nicht.

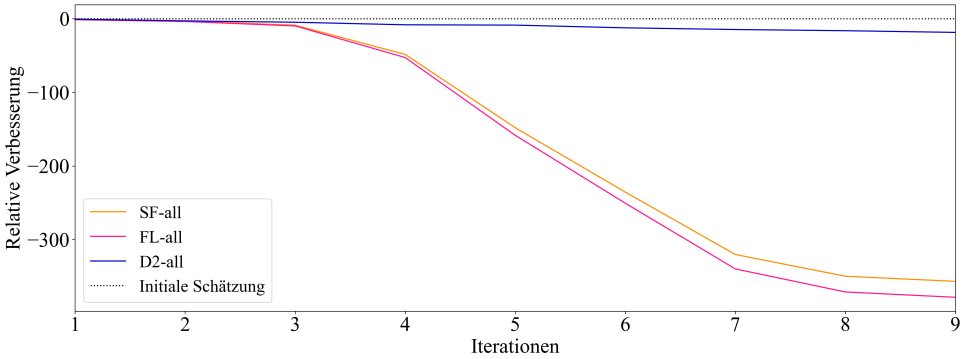



**Abbildung 7.12:** Verlauf der relativen Verbesserung von SF-all, FL-all und D2-all von Sequenz 133 über die Iterationen.

Abschließend sollen die wichtigsten Beobachtungen, die in der Evaluation angestellt worden sind, kurz zusammengefasst werden: Aus den aufgestellten Datentermvarianten hat sich der Datenterm *gcaColSep* als bester herausgestellt. Von den aufgestellten Glattheitstermvarianten erreichte der Glattheitsterm *flowAnisoJoin* die geringsten Fehler. Deshalb wurde das Gesamtmodell aus *gcaColSep* und *flowAnisoJoin* am KITTI-Trainingsdatensatz evaluiert. Insgesamt erzielte dieses Modell keine Verbesserung des gesamten Datensatzes. Die Betrachtung der einzelnen Sequenzen, ohne Durchschnittsbildung, hat gezeigt, dass es durchaus Sequenzen gibt, die in den verschiedenen Fehlermaßen verbessert wurden. Die beste Sequenz stellt dabei Nummer 171 dar, die auch nach neun Iterationen über alle Gesamtbildfehler verbessert wurde. Weitere Sequenzen wurden zwar nach der ersten Iteration ebenfalls verbessert, die Anzahl sinkt jedoch mit steigender Iterationsanzahl. Zusätzlich wurde festgestellt, dass in Sequenzen mit einer Verbesserung entweder der optische Fluss oder die Zieldisparität eine Verbesserung erfährt, allerdings nicht beide gleichzeitig – außer in Sequenz 171. Des Weiteren wurde qualitativ ermittelt, dass die Verfeinerung hauptsächlich an den Objektkanten, allerdings mit negativer Wirkung, durchgeführt wird.

Insgesamt hat die Evaluation das Potential der in dieser Arbeit vorgestellten variationellen Verfeinerungsmethode vom Szenenfluss gezeigt. Sie bietet Grundlage für weitere Forschung, worauf im folgenden Kapitel, nach der Zusammenfassung aller gewonnenen Erkenntnisse, näher eingegangen wird.

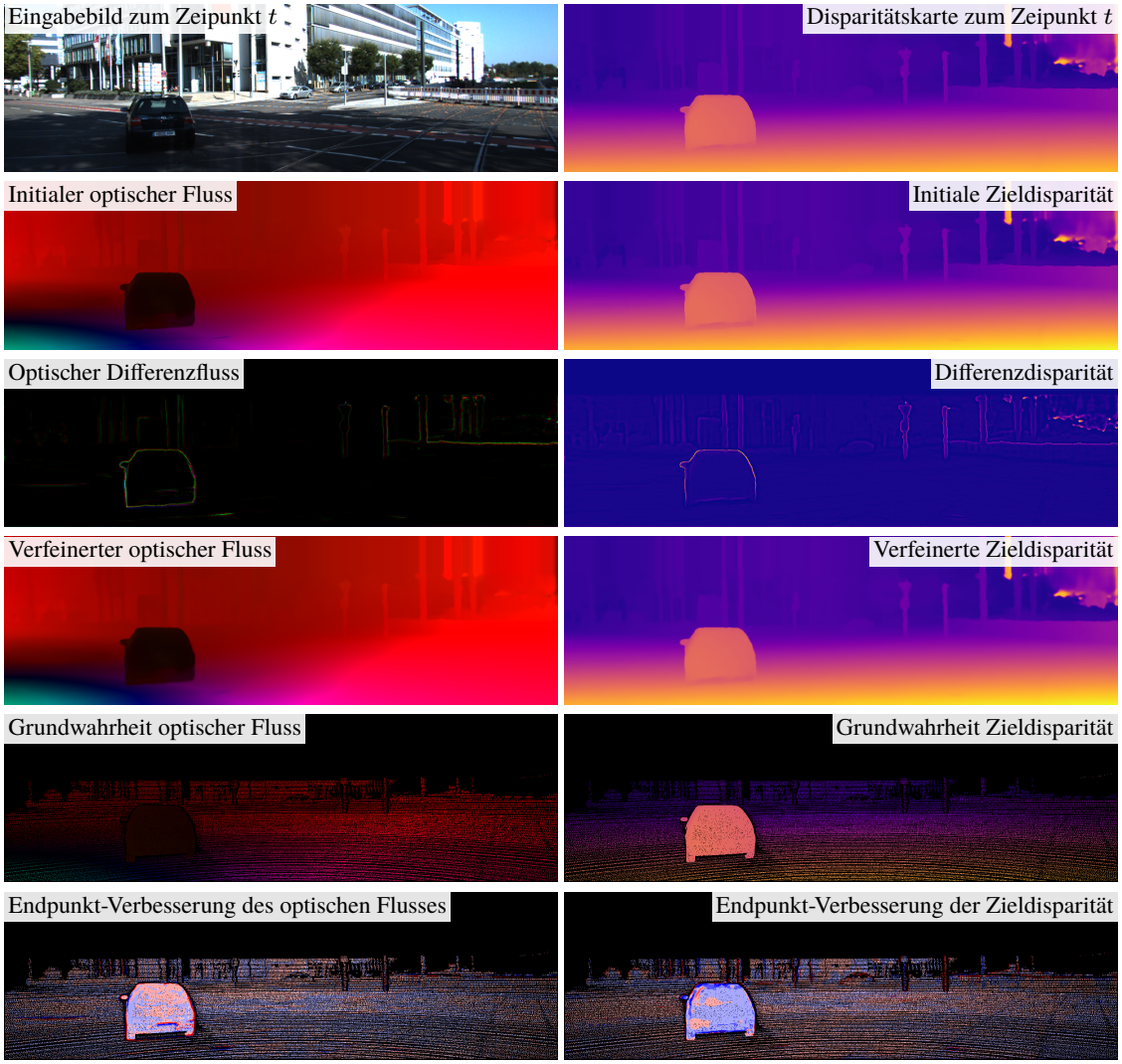



**Abbildung 7.13:** Ergebnisse von Sequenz Nummer 171 aus dem KITTI-Datensatz [52]. Die linke Seite bezieht sich auf den optischen Fluss, die rechte auf die Zieldisparität. Von oben nach unten beschreiben die Bilder: Eingabegröße, initiale Schätzung, Differenz zwischen verfeinerter und initialer Schätzung (zur besseren Sichtbarkeit mit einem Faktor von zehn multipliziert), verfeinerte Schätzung, Grundwahrheit und Endpunkt-Verbesserung.

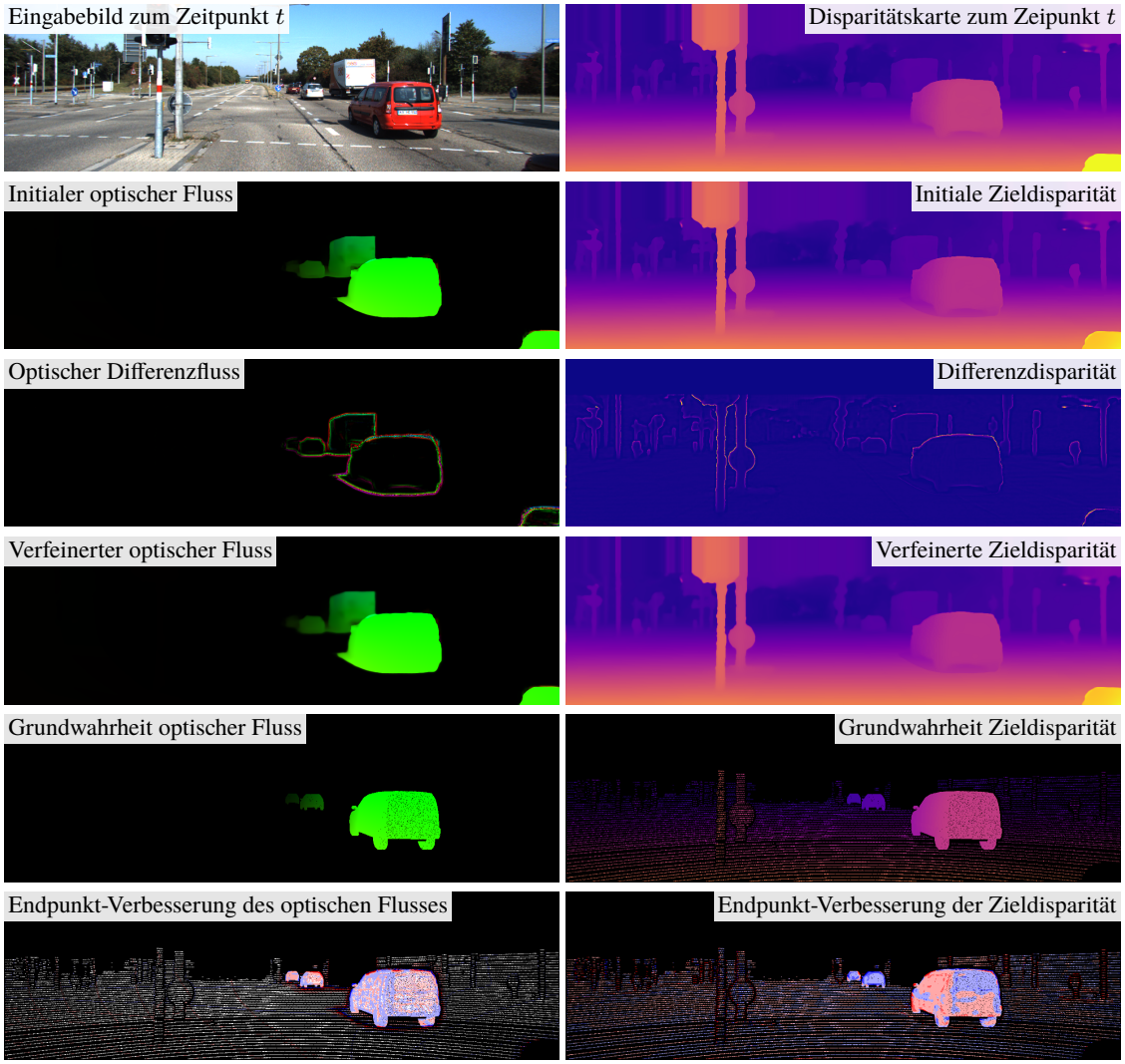



**Abbildung 7.14:** Ergebnisse von Sequenz Nummer 51 aus dem KITTI-Datensatz [52]. Die linke Seite bezieht sich auf den optischen Fluss, die rechte auf die Zieldisparität. Von oben nach unten beschreiben die Bilder: Eingabegröße, initiale Schätzung, Differenz zwischen verfeinerter und initialer Schätzung (zur besseren Sichtbarkeit mit einem Faktor von zehn multipliziert), verfeinerte Schätzung, Grundwahrheit und Endpunkt-Verbesserung.

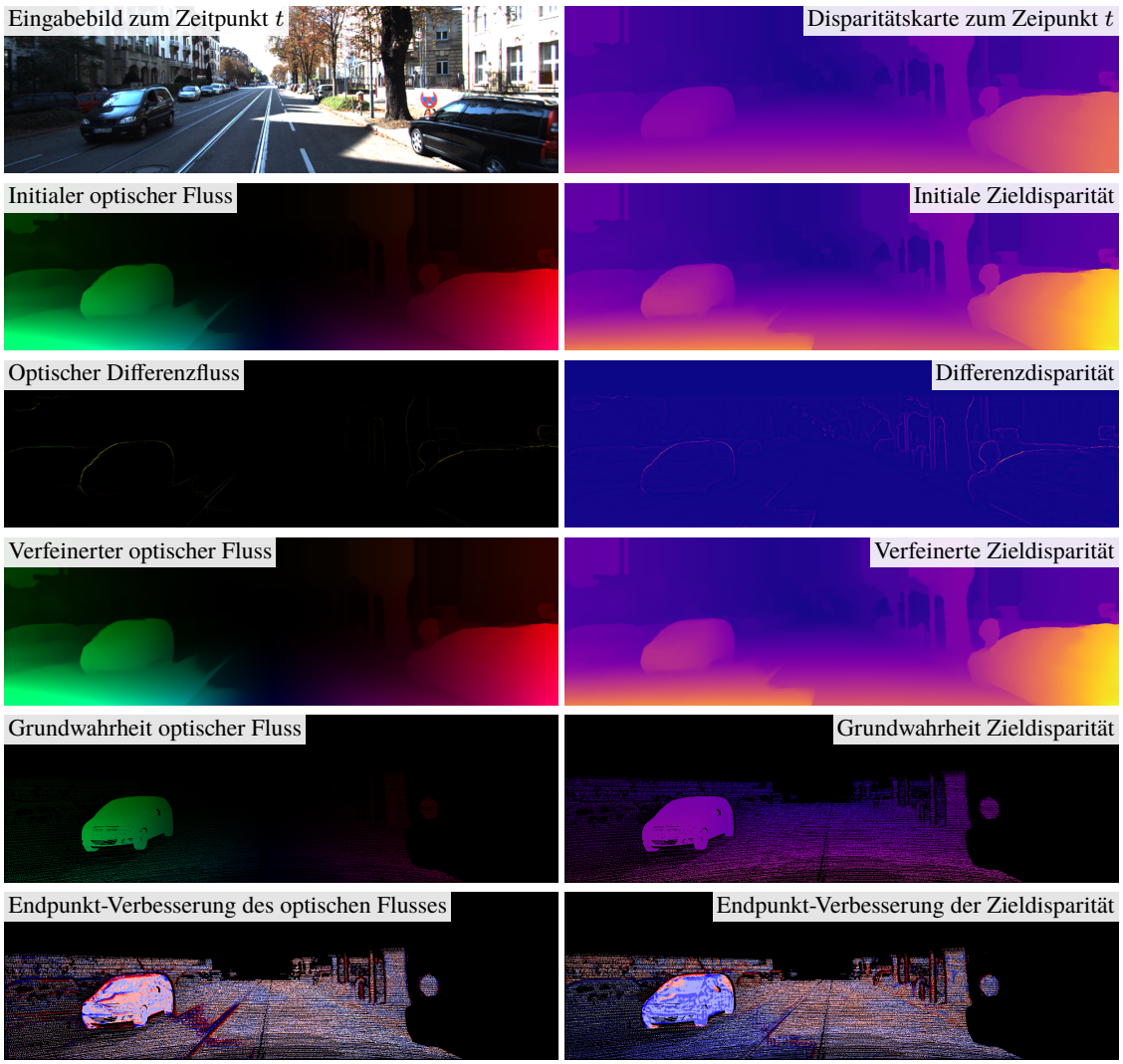



**Abbildung 7.15:** Ergebnisse von Sequenz Nummer 0 aus dem KITTI-Datensatz [52]. Die linke Seite bezieht sich auf den optischen Fluss, die rechte auf die Zieldisparität. Von oben nach unten beschreiben die Bilder: Eingabegröße, initiale Schätzung, Differenz zwischen verfeinerter und initialer Schätzung (zur besseren Sichtbarkeit mit einem Faktor von zehn multipliziert), verfeinerte Schätzung, Grundwahrheit und Endpunkt-Verbesserung.

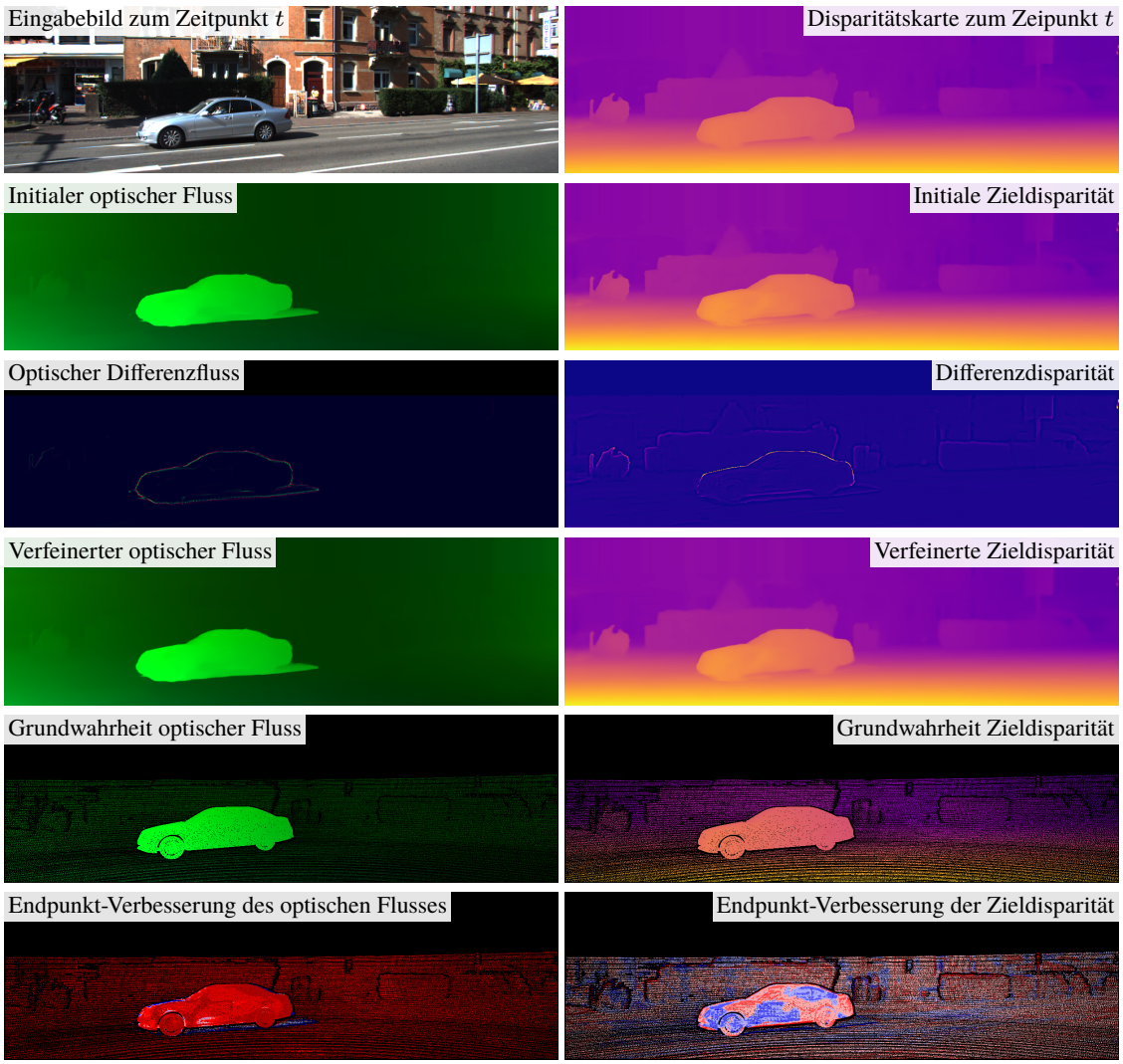



**Abbildung 7.16:** Ergebnisse von Sequenz Nummer 133 aus dem KITTI-Datensatz [52]. Die linke Seite bezieht sich auf den optischen Fluss, die rechte auf die Zieldisparität. Von oben nach unten beschreiben die Bilder: Eingabegröße, initiale Schätzung, Differenz zwischen verfeinerter und initialer Schätzung (zur besseren Sichtbarkeit mit einem Faktor von zehn multipliziert), verfeinerte Schätzung, Grundwahrheit und Endpunkt-Verbesserung.

# 8 Zusammenfassung und Ausblick

## 8.1 Zusammenfassung

Ziel dieser Arbeit war das Entwickeln variationeller Modelle zur Verfeinerung einer initialen Szenenflussschätzung. Dazu wurden in Kapitel 2 wichtige Definitionen und Konzepte eingeführt, darunter das Stereo-Kamera-Setup, Grundlagen zur Disparität und optischem Fluss sowie die Parametrisierung des Szenenflusses. Basierend auf diesen Konzepten wurde in Kapitel 3 ein Variationsansatz zur Schätzung des Szenenflusses vorgestellt. In diesem Zusammenhang wurde ein Energiefunktional, bestehend aus einem Datenterm mit Annahmen zur Grauwertkonstanz und einem Glattheitsterm mit homogener Glattheit von optischem Fluss und Zieldisparität, modelliert, welches es zu minimieren galt. Dafür wurden die dazugehörigen Euler-Lagrange-Gleichungen hergeleitet, diskretisiert und mithilfe eines iterativen Lösungsverfahrens gelöst.

In Kapitel 4 wurde das erarbeitete variationelle Szenenflussverfahren differentiell formuliert, sodass es als Verfeinerungsschritt genutzt werden kann. Der Datenterm des grundlegenden Modells wurde im anschließenden Kapitel 5 um die Annahmen zur Gradientenkonstanz und der Farbbilder erweitert. Außerdem wurde die Robustifizierung mithilfe von subquadratischer Bestrafung eingeführt. Danach wurden in Kapitel 6 Glattheitsterme aufgestellt, welche bild- und flussgetriebene sowie jeweils isotrope und anistrope Glattheit kombinieren. Hierfür wurde der Begriff der treibenden Domäne eingeführt, der das Feld oder die Unbekannte beschreibt, deren Kanten den Glattheitsprozess treiben. Die treibenden Domänen der bildgetriebenen Glattheit sind das Bild und die Disparitätskarte, die der flussgetriebenen Glattheit der optische Fluss und die Zieldisparität. Im Anschluss daran wurde eine Diskretisierung der Euler-Lagrange-Gleichungen vorgenommen, die alle Glattheitstermvarianten zusammenfasst.

Die vorgestellten Daten- und Glattheitsterme wurden für zuvor optimierte Modellparameter in Kapitel 7 am KITTI-Trainingsdatensatz [52] evaluiert. Um das beste Gesamtmodell zu ermitteln, wurde zunächst eine Auswahl von Datentermen mit homogener Glattheit miteinander verglichen. Der daraus resultierende beste Datenterm *gcaColSep* mit separat robustifizierter Grauwertkonstanz der Disparität und Gradientenkonstanz der HSV-Bilder wurde daraufhin in Kombination mit einer Auswahl von Glattheitstermen untersucht. Dabei hat sich der Glattheitsterm *flowAnisoJoin* aus flussgetriebener anisotroper Glattheit mit kombinierten treibenden Domänen als bester herausgestellt, woraus sich das Gesamtmodell *gcaColSep* mit *flowAnisoJoin* ergab. Die Experimente haben gezeigt, dass das gefundene Gesamtmodell zur Szenenflussverfeinerung keine Senkung der durchschnittlichen, initialen Fehler erreichen konnte. Eine weitere Analyse konnte jedoch demonstrieren, dass die initiale Schätzung einzelner Sequenzen tatsächlich verbessert wurde. Im Rahmen dieser wurde die Anzahl der verbesserten Sequenzen und die relative Verbesserung der *bad pixel* Fehler von optischem Fluss, Zieldisparität und Szenenfluss in Abhängigkeit der Iterationszahl untersucht. Dabei wurde festgestellt, dass die Anzahl der verbesserten Sequenzen mit steigender Iterationsanzahl abnimmt und die Verschlechterung der initialen Schätzungen – besonders der Zieldisparität – zunimmt.

Außerdem stellte sich heraus, dass in verbesserten Sequenzen entweder der optische Fluss oder die Zieldisparität eine Verbesserung erfährt, woraus geschlossen wurde, dass die entsprechenden Fehler nicht korrelieren. Zuletzt hat eine visuelle Analyse gezeigt, dass die Verfeinerung hauptsächlich an den Objektkanten stattfindet, die Schätzung an diesen Stellen allerdings verschlechtert wird.

Resümierend kann festgehalten werden, dass die erarbeitete variationelle Methode zur Szenenflussverfeinerung zwar noch nicht die gewünschte Verbesserung der initialen Szenenflussschätzung erzielen konnte, jedoch einzelne Sequenzen ihr Potential beweisen. Somit legt die in dieser Arbeit vorgestellte Methode einen Grundstein, auf dem zukünftige Forschung aufbauen kann. Diesbezüglich werden im nachstehenden Abschnitt mögliche Anknüpfungspunkte skizziert.

## 8.2 Ausblick

Wie in der vorliegenden Arbeit bereits ausführlich diskutiert wurde, zeigten die Ergebnisse der angestellten Experimente – obwohl keine Verbesserung der initialen Szenenflussschätzung aller Sequenzen erzielt werden konnte –, dass eine Weiterentwicklung der hier vorgestellten variationellen Methode zur Szenenflussverfeinerung durchaus zu einem messbaren Fortschritt führen könnte. Mögliche Themen für eine anknüpfende Forschung werden im Folgenden gelistet:

- **Parameteroptimierung:** Im Rahmen der vorliegenden Arbeit lag der Fokus nicht auf der Parameteroptimierung der Modellparameter. Deshalb wurden die Parameter, bevor das beste Gesamtmodell ermittelt wurde, an einem – basierend auf Erfahrung und Literatur – gewählten Modell optimiert und für alle Modelle übernommen. Aufgrund dessen könnte eine ausführlichere Optimierung für das beste Gesamtmodell mit Einbezug der Anzahl an inneren und äußeren Iterationen gewinnbringend sein.
- **Weiterentwicklung des Glattheitsterms:** Die Experimente haben gezeigt, dass die Verfeinerung hauptsächlich an den Objektkanten stattfindet. Die Schätzung des Szenenflusses wird an diesen Positionen jedoch verschlechtert. Das deutet darauf hin, dass eine Weiterentwicklung der aufgestellten Glattheitsterme einen Verbesserungseffekt erzielen könnte. Ein Beispiel für eine solche Erweiterung ist die Einführung der Glattheit zweiten Grades, wie bei Maurer *et al.* [46] gezeigt, oder die Benutzung gemischter bild- und flussgetriebener Glattheit, wie bei Zimmer *et al.* [102].
- **Weiterentwicklung der Annahmen zur Zieldisparität:** Bisher wurde genau eine Konstanzannahme zur Zieldisparität implementiert. In zukünftigen Arbeiten könnten weitere Annahmen, wie die Gradientenkonstanz, miteinbezogen werden.
- **Behandlung großer Bewegungen:** Die Analyse der Sequenz mit größter Verschlechterung der Szenenflussschätzung hat gezeigt, dass große Bewegungen eine besondere Herausforderung darstellen. Ursache dafür könnten die linearisierten Annahmen im Datenterm sein. Eine erfolgreiche Verfeinerung könnte durch den Verzicht der Linearisierung erreicht werden. Dies erfordert das Lösen eines nicht-konvexen, nichtlinearen Gleichungssystems.

# Literaturverzeichnis


[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama. Optuna: A next-Generation Hyperparameter Optimization Framework. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. New York, NY, USA: Association for Computing Machinery, Juli 2019, S. 2623–2631 (zitiert auf S. 73, 74).

[2] L. M. Álvarez León, J. Esclarín Monreal, M. Lefébure, J. Sánchez. A PDE Model for Computing the Optical Flow. In: *CEDYA XVI.* (1999), S. 1349–1356 (zitiert auf S. 54).

[3] A. Badki, O. Gallo, J. Kautz, P. Sen. Binary TTC: A Temporal Geofence for Autonomous Navigation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, S. 12946–12955 (zitiert auf S. 3, 5, 11, 18).

[4] T. Basha, Y. Moses, N. Kiryati. Multi-View Scene Flow Estimation: A View Centered Variational Approach. In: *International Journal of Computer Vision* 101.1 (Jan. 2013), S. 6–21 (zitiert auf S. 5, 19).

[5] A. Behl, O. Hosseini Jafari, S. Karthik Mustikovela, H. Abu Alhaija, C. Rother, A. Geiger. Bounding Boxes, Segmentations and Object Coordinates: How Important Is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios? In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, S. 2574–2583 (zitiert auf S. 5, 6, 11, 18).

[6] M. Bertero, T. Poggio, V. Torre. Ill-Posed Problems in Early Vision. In: *Proceedings of the IEEE* 76.8 (Aug. 1988), S. 869–889 (zitiert auf S. 20).

[7] M. J. Black, P. Anandan. The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields. In: *Computer Vision and Image Understanding* 63.1 (Jan. 1996), S. 75–104 (zitiert auf S. 19).

[8] M. J. Black, P. Anandan. Robust Dynamic Motion Estimation over Time. In: *CVPR*. Bd. 91. 1991, S. 296–203 (zitiert auf S. 43, 59).

[9] T. Brox, A. Bruhn, N. Papenberg, J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: *Computer Vision - ECCV 2004*. Hrsg. von T. Pajdla, J. Matas. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, S. 25–36 (zitiert auf S. I, 3, 4, 19, 38, 44, 47, 49).

[10] A. Bruhn, J. Weickert. Towards Ultimate Motion Estimation: Combining Highest Accuracy with Real-Time Performance. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Bd. 1. Okt. 2005, S. 749–755 (zitiert auf S. 21, 45, 49, 50).

[11] A. Bruhn. Variationelle Optische Flussberechnung: Präzise Modellierung Und Effiziente Numerik. Diss. Universität des Saarlandes, 2006 (zitiert auf S. 21, 30, 34, 38, 87).

[12] J. Čech, J. Sanchez-Riera, R. Horaud. Scene Flow Estimation by Growing Correspondence Seeds. In: *CVPR 2011*. Juni 2011, S. 3129–3136 (zitiert auf S. 6).

[13] P. Charbonnier, L. Blanc-Feraud, G. Aubert, M. Barlaud. Deterministic Edge-Preserving Regularization in Computed Imaging. In: *IEEE Transactions on Image Processing* 6.2 (Feb. 1997), S. 298–311 (zitiert auf S. 48).

[14] X. Cheng, Y. Zhong, M. Harandi, Y. Dai, X. Chang, T. Drummond, H. Li, Z. Ge. Hierarchical Neural Architecture Search for Deep Stereo Matching. In: *Advances in Neural Information Processing Systems* 33 (2020), S. 2218–2218. arXiv: 2010.13501 [cs] (zitiert auf S. 3, 6).

[15] R. Courant, D. Hilbert. *Methods of Mathematical Physics: Partial Differential Equations*. John Wiley & Sons, Sep. 2008 (zitiert auf S. 23, 24).

[16] N. Dalal, B. Triggs, C. Schmid. Human Detection Using Oriented Histograms of Flow and Appearance. In: *Computer Vision – ECCV 2006*. Hrsg. von A. Leonardis, H. Bischof, A. Pinz. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, S. 428–441 (zitiert auf S. 2).

[17] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, T. Brox. FlowNet: Learning Optical Flow With Convolutional Networks. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, S. 2758–2766 (zitiert auf S. 2).

[18] U. Franke, C. Rabe, H. Badino, S. Gehrig. 6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception. In: *Pattern Recognition*. Hrsg. von W. G. Kropatsch, R. Sablatnig, A. Hanbury. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, S. 216–223 (zitiert auf S. 1).

[19] Y. Furukawa, J. Ponce. Dense 3D Motion Capture for Human Faces. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Juni 2009, S. 1674–1681 (zitiert auf S. 2).

[20] A. Geiger, P. Lenz, R. Urtasun. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Juni 2012, S. 3354–3361 (zitiert auf S. 1).

[21] P. Golland, A. M. Bruckstein. Motion from Color. In: *Computer Vision and Image Understanding* 68.3 (1997), S. 346–362 (zitiert auf S. 50, 51).

[22] V. Golyanik, K. Kim, R. Maier, M. Nießner, D. Stricker, J. Kautz. Multiframe Scene Flow with Piecewise Rigid Motion. In: *2017 International Conference on 3D Vision (3DV)*. Okt. 2017, S. 273–281 (zitiert auf S. 3).

[23] V. Guizilini, K.-H. Lee, R. Ambruş, A. Gaidon. Learning Optical Flow, Depth, and Scene Flow Without Real-World Labels. In: *IEEE Robotics and Automation Letters* 7.2 (Apr. 2022), S. 3491–3498 (zitiert auf S. 2).

[24] N. Hansen, A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. In: *Evolutionary computation* 9.2 (2001), S. 159–195 (zitiert auf S. 74).

[25] R. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003 (zitiert auf S. 12).

[26] E. Herbst, X. Ren, D. Fox. RGB-D Flow: Dense 3-D Motion Estimation Using Color and Depth. In: *2013 IEEE International Conference on Robotics and Automation*. Mai 2013, S. 2276–2282 (zitiert auf S. 3, 4).

[27] B. K. P. Horn, B. G. Schunck. Determining Optical Flow. In: *Artificial Intelligence* 17.1 (Aug. 1981), S. 185–203 (zitiert auf S. 2, 4, 19–22, 37, 43).

[28] F. Huguet, F. Devernay. A Variational Method for Scene Flow Estimation from Stereo Sequences. In: *2007 IEEE 11th International Conference on Computer Vision*. Okt. 2007, S. 1–7 (zitiert auf S. 4, 11, 18, 19, 43, 53).

[29] C. H. Hung, L. Xu, J. Jia. Consistent Binocular Depth and Scene Flow with Chained Temporal Profiles. In: *International Journal of Computer Vision* 102.1 (März 2013), S. 271–292 (zitiert auf S. 18).

[30] M. Jaimez, M. Souiai, J. Stückler, J. Gonzalez-Jimenez, D. Cremers. Motion Cooperation: Smooth Piece-wise Rigid Scene Flow from RGB-D Images. In: *2015 International Conference on 3D Vision*. Okt. 2015, S. 64–72 (zitiert auf S. 4).

[31] H. Jiang, D. Sun, V. Jampani, Z. Lv, E. Learned-Miller, J. Kautz. SENSE: A Shared Encoder Network for Scene-Flow Estimation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, S. 3195–3204 (zitiert auf S. 5).

[32] J. Kačur, J. Nečas, J. Polák, J. Souček. Convergence of a Method for Solving the Magnetostatic Field in Nonlinear Media. In: *Aplikace matematiky* 13.6 (1968), S. 456–465 (zitiert auf S. 45, 47, 61, 66).

[33] H. Kielhöfer. *Calculus of Variations*. Bd. 67. Texts in Applied Mathematics. Cham: Springer International Publishing, 2018 (zitiert auf S. 23).

[34] H. Kielhöfer. *Variationsrechnung: eine Einführung in die Theorie einer unabhängigen Variablen mit Beispielen und Aufgaben*. 1. Aufl. Studium. Wiesbaden: Vieweg + Teubner, 2010 (zitiert auf S. 19, 23).

[35] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld. Learning Realistic Human Actions from Movies. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. Juni 2008, S. 1–8 (zitiert auf S. 2).

[36] P. Lenz, J. Ziegler, A. Geiger, M. Roser. Sparse Scene Flow Segmentation for Moving Object Detection in Urban Environments. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. Juni 2011, S. 926–932 (zitiert auf S. 1).

[37] C. Li, H. Ma, Q. Liao. Two-Stage Adaptive Object Scene Flow Using Hybrid CNN-CRF Model. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. Jan. 2021, S. 3876–3883 (zitiert auf S. 2, 3, 6, 11, 18).

[38] J. Li, P. Wang, P. Xiong, T. Cai, Z. Yan, L. Yang, J. Liu, H. Fan, S. Liu. Practical Stereo Matching via Cascaded Recurrent Network With Adaptive Correlation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, S. 16263–16272 (zitiert auf S. 3).

[39] H. Liu, T. Lu, Y. Xu, J. Liu, W. Li, L. Chen. CamLiFlow: Bidirectional Camera-LiDAR Fusion for Joint Optical Flow and Scene Flow Estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, S. 5791–5801 (zitiert auf S. 2, 3, 6, 10, 11, 18).

[40] P. Liu, M. Reale, L. Yin. 3D Head Pose Estimation Based on Scene Flow and Generic Head Model. In: *2012 IEEE International Conference on Multimedia and Expo*. Juli 2012, S. 794–799 (zitiert auf S. 2).

[41] X. Liu, C. R. Qi, L. J. Guibas. FlowNet3D: Learning Scene Flow in 3D Point Clouds. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, S. 529–537 (zitiert auf S. 7).

[42] W.-C. Ma, S. Wang, R. Hu, Y. Xiong, R. Urtasun. Deep Rigid Instance Scene Flow. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, S. 3614–3622 (zitiert auf S. 2, 3, 5, 11, 18).

[43] D. Maurer. Adaptive Algorithms for 3D Reconstruction and Motion Estimation. doctoralThesis. 2019 (zitiert auf S. 48).

[44] D. Maurer. Depth-Driven Variational Methods for Stereo Reconstruction. Diss. Universität Stuttgart, 2014 (zitiert auf S. 12, 15, 24, 49).

[45] D. Maurer, M. Stoll, A. Bruhn. Order-Adaptive and Illumination-Aware Variational Optical Flow Refinement. In: *British Machine Vision Conference (BMVC)*. 2017, S. 150.1–150.13 (zitiert auf S. 3).

[46] D. Maurer, M. Stoll, S. Volz, P. Gairing, A. Bruhn. A Comparison of Isotropic and Anisotropic Second Order Regularisers for Optical Flow. In: *Scale Space and Variational Methods in Computer Vision*. Hrsg. von F. Lauze, Y. Dong, A. B. Dahl. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, S. 537–549 (zitiert auf S. 82, 98).

[47] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, T. Brox. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, S. 4040–4048 (zitiert auf S. 5, 72).

[48] L. Mehl. Anisotropic Selection Schemes for Order-Adaptive Variational Optical Flow Methods. Diss. Universität Stuttgart, 2020 (zitiert auf S. 29).

[49] L. Mehl, C. Beschle, A. Barth, A. Bruhn. An Anisotropic Selection Scheme for Variational Optical Flow Methods with Order-Adaptive Regularisation. In: *Scale Space and Variational Methods in Computer Vision*. Hrsg. von A. Elmoataz, J. Fadili, Y. Quéau, J. Rabin, L. Simon. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, S. 140–152 (zitiert auf S. 21, 38, 49).

[50] L. Mehl, A. Jahedi, J. Schmalfuss, A. Bruhn. *M-FUSE: Multi-frame Fusion for Scene Flow Estimation*. Juli 2022. arXiv: 2207.05704 [cs] (zitiert auf S. 2, 3, 6, 7, 38).

[51] M. Menze, C. Heipke, A. Geiger. JOINT 3D ESTIMATION OF VEHICLES AND SCENE FLOW. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* II-3/W5 (Aug. 2015), S. 427–434 (zitiert auf S. 1).

[52] M. Menze, A. Geiger. Object Scene Flow for Autonomous Vehicles. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, S. 3061–3070 (zitiert auf S. 1–6, 17, 35, 67, 70–72, 92–95, 97).

[53] Y. Mileva, A. Bruhn, J. Weickert. Illumination-Robust Variational Optical Flow with Photometric Invariants. In: *Joint Pattern Recognition Symposium*. Springer, 2007, S. 152–162 (zitiert auf S. 51).

[54] H.-H. Nagel, W. Enkelmann. An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.5 (Sep. 1986), S. 565–593 (zitiert auf S. 19, 54, 57, 59).

[55] S. Nüssle. Berechnung des Szenenflusses mit Variationsansätzen. Diss. Universität Stuttgart, 2017 (zitiert auf S. 5, 21, 49).

[56] J. Park, T. H. Oh, J. Jung, Y.-W. Tai, I. S. Kweon. A Tensor Voting Approach for Multi-view 3D Scene Flow Estimation and Refinement. In: *Computer Vision – ECCV 2012*. Hrsg. von A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, S. 288–302 (zitiert auf S. 3, 7).

[57] P. Perona, J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7 (Juli 1990), S. 629–639 (zitiert auf S. 48).

[58] Y.-L. Qiao, L. Gao, Y. Lai, F.-L. Zhang, M.-Z. Yuan, S. Xia. SF-Net: Learning Scene Flow from RGB-D Images with CNNs. In: (2018).

[59] J. Quiroga, T. Brox, F. Devernay, J. Crowley. Dense Semi-Rigid Scene Flow Estimation from Rgbd Images. In: *European Conference on Computer Vision*. Springer, 2014, S. 567–582 (zitiert auf S. 3).

[60] C. Rabe, T. Müller, A. Wedel, U. Franke. Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time. In: *Computer Vision – ECCV 2010*. Hrsg. von K. Daniilidis, P. Maragos, N. Paragios. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, S. 582–595 (zitiert auf S. 18, 23, 54).

[61] Z. Ren, D. Sun, J. Kautz, E. Sudderth. Cascaded Scene Flow Prediction Using Semantic Segmentation. In: *2017 International Conference on 3D Vision (3DV)*. Okt. 2017, S. 225–233 (zitiert auf S. 5, 6).

[62] C. Richardt, H. Kim, L. Valgaerts, C. Theobalt. Dense Wide-Baseline Scene Flow from Two Handheld Video Cameras. In: *2016 Fourth International Conference on 3D Vision (3DV)*. Okt. 2016, S. 276–285 (zitiert auf S. 7).

[63] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 2003 (zitiert auf S. 30–34).

[64] R. Saxena, R. Schuster, O. Wasenmuller, D. Stricker. PWOC-3D: Deep Occlusion-Aware End-to-End Scene Flow Estimation. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. Juni 2019, S. 324–331 (zitiert auf S. 5).

[65] R. Schuster, C. Unger, D. Stricker. A Deep Temporal Fusion Framework for Scene Flow Using a Learnable Motion Model and Occlusions. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, S. 247–255 (zitiert auf S. 5, 7).

[66] R. Schuster, O. Wasenmuller, G. Kuschk, C. Bailer, D. Stricker. SceneFlowFields: Dense Interpolation of Sparse Scene Flow Correspondences. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. März 2018, S. 1056–1065 (zitiert auf S. 7).

[67] R. Schuster, O. Wasenmüller, C. Unger, G. Kuschk, D. Stricker. SceneFlowFields++: Multi-frame Matching, Visibility Prediction, and Robust Interpolation for Scene Flow Estimation. In: *International Journal of Computer Vision* 128.2 (Feb. 2020), S. 527–546 (zitiert auf S. 7).

[68] N. Slesareva, A. Bruhn, J. Weickert. Optic Flow Goes Stereo: A Variational Method for Estimating Discontinuity-Preserving Dense Disparity Maps. In: *Pattern Recognition*. Hrsg. von W. G. Kropatsch, R. Sablatnig, A. Hanbury. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, S. 33–40 (zitiert auf S. 3, 38).

[69] L. Sommer, P. Schröppel, T. Brox. SF2SE3: Clustering Scene Flow into SE(3)-Motions via Proposal and Selection. In: *Pattern Recognition*. Hrsg. von B. Andres, F. Bernard, D. Cremers, S. Frintrop, B. Goldlücke, I. Ihrke. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, S. 215–229 (zitiert auf S. 4, 11, 18).

[70] M. Stoll, S. Volz, A. Bruhn. Adaptive Integration of Feature Matches into Variational Optical Flow Methods. In: *Computer Vision – ACCV 2012*. Hrsg. von K. M. Lee, Y. Matsushita, J. M. Rehg, Z. Hu. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, S. 1–14 (zitiert auf S. 7).

[71] A. Stone, D. Maurer, A. Ayvaci, A. Angelova, R. Jonschkowski. SMURF: Self-teaching Multi-Frame Unsupervised RAFT with Full-Image Warping. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, S. 3887–3896 (zitiert auf S. 38).

[72] D. Stoyanov. Stereoscopic Scene Flow for Robotic Assisted Minimally Invasive Surgery. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*. Hrsg. von N. Ayache, H. Delingette, P. Golland, K. Mori. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, S. 479–486 (zitiert auf S. 2).

[73] D. Sun, S. Roth, M. J. Black. Secrets of Optical Flow Estimation and Their Principles. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Juni 2010, S. 2432–2439 (zitiert auf S. 2).

[74] D. Sun, S. Roth, J. P. Lewis, M. J. Black. Learning Optical Flow. In: *European Conference on Computer Vision*. Springer, 2008, S. 83–97 (zitiert auf S. 59).

[75] D. Sun, X. Yang, M.-Y. Liu, J. Kautz. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, S. 8934–8943 (zitiert auf S. 38).

[76] N. Sundaram, T. Brox, K. Keutzer. Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow. In: *Computer Vision – ECCV 2010*. Hrsg. von K. Daniilidis, P. Maragos, N. Paragios. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, S. 438–451 (zitiert auf S. 29, 32, 49).

[77] Z. Teed, J. Deng. RAFT-3D: Scene Flow Using Rigid-Motion Embeddings. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, Juni 2021, S. 8371–8380 (zitiert auf S. 2, 3, 6–8, 11, 18, 71, 73, 77, 81).

[78] Z. Teed, J. Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In: *Computer Vision – ECCV 2020*. Hrsg. von A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, S. 402–419 (zitiert auf S. 71).

[79] M. Tistarelli. Multiple Constraints for Optical Flow. In: *Computer Vision — ECCV '94*. Hrsg. von J.-O. Eklundh. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1994, S. 61–70 (zitiert auf S. 49).

[80] S. Uras, F. Girosi, A. Verri, V. Torre. A Computational Approach to Motion Perception. In: *Biological Cybernetics* 60.2 (Dez. 1988), S. 79–87 (zitiert auf S. 49).

[81] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, C. Theobalt. Joint Estimation of Motion, Structure and Geometry from Stereo Sequences. In: *Computer Vision – ECCV 2010*. Hrsg. von K. Daniilidis, P. Maragos, N. Paragios. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, S. 568–581 (zitiert auf S. 4).

[82] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, C. Theobalt. Lightweight Binocular Facial Performance Capture under Uncontrolled Lighting. In: *ACM Trans. Graph.* 31.6 (2012), S. 187–1 (zitiert auf S. 2).

[83] J. van de Weijer, T. Gevers. Robust Optical Flow from Photometric Invariants. In: *2004 International Conference on Image Processing, 2004. ICIP'04.* Bd. 3. IEEE, 2004, S. 1835–1838 (zitiert auf S. 51).

[84] S. Vedula, S. Baker, P. Rander, R. Collins, T. Kanade. Three-Dimensional Scene Flow. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Bd. 2. Sep. 1999, S. 722–729 (zitiert auf S. 1, 2).

[85] C. Vogel, S. Roth, K. Schindler. View-Consistent 3D Scene Flow Estimation over Multiple Frames. In: *Computer Vision – ECCV 2014*. Hrsg. von D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, S. 263–278 (zitiert auf S. 4).

[86] C. Vogel, K. Schindler, S. Roth. Piecewise Rigid Scene Flow. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, S. 1377–1384 (zitiert auf S. 4).

[87] P. Wang, W. Li, Z. Gao, Y. Zhang, C. Tang, P. Ogunbona. Scene Flow to Action Map: A New Representation for Rgb-d Based Action Recognition with Convolutional Neural Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, S. 595–604 (zitiert auf S. 2).

[88] Z. Wang, S. Li, H. Howard-Jenkins, V. Prisacariu, M. Chen. FlowNet3D++: Geometric Losses For Deep Scene Flow Estimation. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, S. 91–98 (zitiert auf S. 7).

[89] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, D. Cremers. Stereoscopic Scene Flow Computation for 3D Motion Understanding. In: *International Journal of Computer Vision* 95.1 (Okt. 2011), S. 29–51 (zitiert auf S. 4, 11).

[90] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, D. Cremers. Efficient Dense Scene Flow from Sparse or Dense Stereo Data. In: *Computer Vision – ECCV 2008*. Hrsg. von D. Forsyth, P. Torr, A. Zisserman. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, S. 739–751 (zitiert auf S. 2, 4, 11, 14, 18, 19).

[91] J. Weickert. *Anisotropic Diffusion in Image Processing*. Bd. 1. Teubner Stuttgart, 1998 (zitiert auf S. 64).

[92] J. Weickert, C. Schnörr. A Theoretical Framework for Convex Regularizers in PDE-Based Computation of Image Motion. In: *International Journal of Computer Vision* 45.3 (Dez. 2001), S. 245–264 (zitiert auf S. 53, 57, 59, 61, 62).

[93] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, H. Bischof. Anisotropic Huber-L1 Optical Flow. In: *BMVC*. Bd. 1. 2009, S. 3 (zitiert auf S. 19).

[94] G. Xu, J. Cheng, P. Guo, X. Yang. Attention Concatenation Volume for Accurate and Efficient Stereo Matching. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, S. 12981–12990 (zitiert auf S. 3).

[95] G. Yang, D. Ramanan. Learning To Segment Rigid Motions From Two Frames. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, S. 1266–1275 (zitiert auf S. 2, 3, 5, 6, 11, 18).

[96] G. Yang, D. Ramanan. Upgrading Optical Flow to 3D Scene Flow Through Optical Expansion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, S. 1334–1343 (zitiert auf S. 2, 3, 5, 11, 18).

[97] D. Young. Iterative Methods for Solving Partial Difference Equations of Elliptic Type. In: *Transactions of the American Mathematical Society* 76.1 (1954), S. 92–111 (zitiert auf S. 34).

[98] A. Zanfir, C. Sminchisescu. Large Displacement 3D Scene Flow With Occlusion Reasoning. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, S. 4417–4425 (zitiert auf S. 3, 4).

[99] F. Zhang, V. Prisacariu, R. Yang, P. H. S. Torr. GA-Net: Guided Aggregation Net for End-To-End Stereo Matching. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, S. 185–194 (zitiert auf S. 3, 6, 11, 70–72).

[100] H. Zimmer, A. Bruhn, L. Valgaerts, M. Breuß, J. Weickert, B. Rosenhahn, H.-P. Seidel. PDE-Based Anisotropic Disparity-Driven Stereo Vision. In: *VMV*. 2008, S. 263–272 (zitiert auf S. 54).

[101] H. Zimmer, A. Bruhn, J. Weickert. Optic Flow in Harmony. In: *International Journal of Computer Vision* 93.3 (Juli 2011), S. 368–388 (zitiert auf S. 12, 19, 45, 47, 51).

[102] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, H.-P. Seidel. Complementary Optic Flow. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Hrsg. von D. Cremers, Y. Boykov, A. Blake, F. R. Schmidt. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2009, S. 207–220 (zitiert auf S. 22, 45, 51–53, 59, 98).

# A Anhang

## Glattheitstermvarianten

| | Treibende Domänen von | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Optischer Fluss | | | | | Zieldisparität | | | | | Isotropie | |
| Modelle | $f$ | $\zeta$ | $\overline{u},\overline{v}$ | $\overline{d}$ | Diffusionstensor $D^{uv}$ | $f$ | $\zeta$ | $\overline{u},\overline{v}$ | $\overline{d}$ | Diffusionstensor $D^{d}$ | isotrop | anisotrop |
| homogen | - | - | - | - | $I$ | - | - | - | - | $I$ | - | - |
| imgIsoSep | ✓ | - | - | - | $g(\lvert\nabla f\rvert^2)I$ | - | ✓ | - | - | $g(\lvert\nabla \zeta\rvert^2)I$ | ✓ | - |
| imgIsoJoin | ✓ | ✓ | - | - | $g(\lvert\nabla f\rvert^2 + \lvert\nabla \zeta\rvert^2)I$ | ✓ | ✓ | - | - | $g(\lvert\nabla f\rvert^2 + \lvert\nabla \zeta\rvert^2)I$ | ✓ | - |
| imgIsoImg | ✓ | - | - | - | $g(\lvert\nabla f\rvert^2)I$ | ✓ | - | - | - | $g(\lvert\nabla f\rvert^2)I$ | ✓ | - |
| imgIsoDisp | - | ✓ | - | - | $g(\lvert\nabla \zeta\rvert^2)I$ | - | ✓ | - | - | $g(\lvert\nabla \zeta\rvert^2)I$ | ✓ | - |
| imgAnisoSep | ✓ | - | - | - | $D_{img}(\nabla f)$ | - | ✓ | - | - | $D_{img}\nabla \zeta)$ | - | ✓ |
| imgAnisoJoin | ✓ | ✓ | - | - | $D_{img}(\nabla f + \nabla \zeta)$ | ✓ | ✓ | - | - | $D_{img}\nabla f + \nabla \zeta)$ | - | ✓ |
| imgAnisoImg | ✓ | - | - | - | $D_{img}(\nabla f)$ | ✓ | - | - | - | $D_{img}\nabla f)$ | - | ✓ |
| imgAnisoDisp | - | ✓ | - | - | $D_{img}(\nabla \zeta)$ | - | ✓ | - | - | $D_{img}\nabla \zeta)$ | - | ✓ |
| flowIsoSep | - | - | ✓ | - | $g\left(\lvert\nabla \overline{u}\rvert^2 + \lvert\nabla \overline{v}\rvert^2\right) I$ | - | - | - | ✓ | $g\left(\lvert\nabla \overline{d}\rvert^2\right) I$ | ✓ | - |
| flowIsoJoin | - | - | ✓ | ✓ | $g\left(\lvert\nabla \overline{u}\rvert^2 + \lvert\nabla \overline{v}\rvert^2 + \lvert\nabla \overline{d}\rvert^2\right) I$ | - | - | ✓ | ✓ | $g\left(\lvert\nabla \overline{u}\rvert^2 + \lvert\nabla \overline{v}\rvert^2 + \lvert\nabla \overline{d}\rvert^2\right) I$ | ✓ | - |
| flowIsoFlow | - | - | ✓ | - | $g\left(\lvert\nabla \overline{u}\rvert^2 + \lvert\nabla \overline{v}\rvert^2\right) I$ | - | - | ✓ | - | $g\left(\lvert\nabla \overline{u}\rvert^2 + \lvert\nabla \overline{v}\rvert^2\right) I$ | ✓ | - |
| flowIsoDisp | - | - | - | ✓ | $g\left(\lvert\nabla \overline{d}\rvert^2\right) I$ | - | - | - | ✓ | $g\left(\lvert\nabla \overline{d}\rvert^2\right) I$ | ✓ | - |
| flowAnisoSep | - | - | ✓ | - | $D_{flow}\left(\mathbf{T}_u + \mathbf{T}_v\right)$ | - | - | - | ✓ | $D_{flow}\left(\mathbf{T}_d\right)$ | - | ✓ |
| flowAnisoJoin | - | - | ✓ | ✓ | $D_{flow}\left(\mathbf{T}_u + \mathbf{T}_v + \mathbf{T}_d\right)$ | - | - | ✓ | ✓ | $D_{flow}\left(\mathbf{T}_u + \mathbf{T}_v + \mathbf{T}_d\right)$ | - | ✓ |
| flowAnisoFlow | - | - | ✓ | - | $D_{flow}\left(\mathbf{T}_u + \mathbf{T}_v\right)$ | - | - | ✓ | - | $D_{flow}\left(\mathbf{T}_u + \mathbf{T}_v\right)$ | - | ✓ |
| flowAnisoDisp | - | - | - | ✓ | $D_{flow}\left(\mathbf{T}_d\right)$ | - | - | - | ✓ | $D_{flow}\left(\mathbf{T}_d\right)$ | - | ✓ |

**Tabelle A.1:** Übersicht über die verschiedenen Modelle des Glattheitsterms. Die Spalte ‚optischer Fluss' beschreibt die möglichen treibenden Domänen des optischen Flusses mit der Zusammensetzung des dazugehörigen Diffusionstensors. Die Spalte ‚Zieldisparität' beschreibt die möglichen treibenden Domänen der Zieldisparität mit der Zusammensetzung des dazugehörigen Diffusionstensors. Beide Spalten unterteilen sich in die treibenden Domänen, die wie folgt notiert sind: $f$ steht für die Bildfolge, $\zeta$ für die Disparitätskarte, $\overline{u},\overline{v}$ für den optischen Fluss und $\overline{d}$ für die Zieldisparität. Die ersten beiden sind die treibenden Domänen bildgetriebener Modelle, die letzten beiden die flussgetriebener Modelle. Der Diffusionstensor kommt wie in den Gleichungen (6.46) und (6.48) eingeführt zum Einsatz.

**Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift