

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit

Zwischenbildinterpolation mit optischem Fluss

Noah Berenguel Senn

Studiengang:	Informatik
Prüfer/in:	Prof. Dr.-Ing. Andrés Bruhn
Betreuer/in:	Prof. Dr.-Ing. Andrés Bruhn, Dr. Daniel Maurer, Lukas Mehl, M.Sc.
Beginn am:	2019-11-08
Beendet am:	2020-07-03

Kurzfassung

Die Erzeugung von Zwischenbildern in Bildfolgen ist eines der klassischen Probleme in der Videoverarbeitung. Viele Lösungsansätze dafür arbeiten häufig mit der Information der Bewegung im Bild, um scharfe und realistische Ergebnisse zu erzeugen. Eine Möglichkeit für die verwendete Bewegungsinformation stellt ein sogenannter optischer Fluss dar, der eine pixelweise Schätzung der Bewegung zwischen zwei aufeinanderfolgenden Bildern beschreibt. Verwendet werden kann ein optischer Fluss in sogenannten Vorwärts- und Rückwärtswarpingstrategien, bei denen entlang der Bewegungstrajektorie eine lineare Interpolation stattfindet. Bei diesen sich teils sehr ähnelnden Strategien ist aber meist nicht klar, welche Variante in der Praxis die besten Ergebnisse erzeugt. Diese Arbeit bündelt daher vieler solcher Ansätze und führt einen einheitlichen Vergleich durch.

Zuerst wird das sogenannte (Image-) Warping als Grundtechnik für die spätere Zwischenbildinterpolation vorgestellt, inklusive damit verbundenen Techniken für das Sampling und Splatting von Bildinformationen. Es wird dabei außerdem ein Verfahren für das Füllen von Lücken in Bildern (Inpainting) beschrieben, das eine zentrale Rolle für die spätere Zwischenbildinterpolation besitzt. Mithilfe der Grundtechniken werden im Anschluss eine Reihe von Bildinterpolationsverfahren schrittweise und aufbauend eingeführt, wobei sich immer an der Anzahl von verwendeten Bildern und optischen Flüssen orientiert wird (1-Bild Fall, 2-Bild Fall und bidirektionale Verfahren). In einer abschließenden Evaluation werden die Bildinterpolationsverfahren qualitativ und quantitativ miteinander verglichen. Dies geschieht anhand von Benchmarkdaten für optische Flüsse, bei denen ausgelassene Bilder als Referenzbild dienen. Als Bewertung der erzeugten Zwischenbilder dienen klassische Fehlermaße aus der Literatur und ein ergänzender visueller Eindruck. Ziel der Evaluation ist die Verifizierung der Überlegungen hinter den Verfahren und die Prüfung wie gut sie in der Praxis funktionieren.

Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation	1
1.2	Anwendung	1
1.3	Verwandte Arbeiten	3
1.4	Zielsetzung	4
1.5	Gliederung	4
2	Optischer Fluss	5
2.1	Grundlagen	5
2.2	Herausforderungen	6
2.3	ProFlow	7
3	Warping	9
3.1	Grundlagen	9
3.2	Optischer Fluss als Mappingfunktion	10
3.3	Backward Warping	11
3.3.1	Nearest Neighbour Interpolation	13
3.3.2	Uniformes Mitteln	13
3.3.3	Bilineare Interpolation	13
3.4	Forward Warping	14
3.4.1	Verteilungsstrategien	15
3.4.2	Kollisionsbehandlungen	16
3.5	Inpainting	19
3.5.1	Diffusion	19
3.5.2	Diskretisierung	20
3.5.3	Beschleunigung	22
3.6	Anwendung	23
4	Bildinterpolationsverfahren	25
4.1	Naiver Ansatz	26
4.2	Zentrale Annahme	27
4.3	1-Bild Fall	28
4.3.1	Forward Warping	28
4.3.2	Backward Warping	29
4.4	2-Bild Fall	30
4.4.1	Symmetrisches Backward Warping	31
4.4.2	Symmetrisches Forward Warping	33
4.5	Bidirektionale Verfahren	35
4.5.1	Umgedrehter 1-Bild Fall	36
4.5.2	Forward Forward Warping	37
4.5.3	Backward Backward Warping	38
4.5.4	Bidirektionaler 2-Bild Fall	39
4.6	Vergleich	42

5 Experimente	43
5.1 Testdaten	43
5.2 Fehlermaße	45
5.3 Implementierung	47
5.4 1-Bild Fall	47
5.4.1 Forward Warping	48
5.4.2 Backward Warping	50
5.4.3 Vergleich und Probleme	54
5.5 Inpainting	55
5.5.1 Geschwindigkeitsvergleich	55
5.5.2 Anwendung	57
5.5.3 Probleme	60
5.6 2-Bild Fall	61
5.6.1 Vergleich	62
5.6.2 Probleme	64
5.7 Bidirektionale Verfahren	65
5.7.1 Vorwärtsfluss vs. Rückwärtsfluss	66
5.7.2 Ersatz für symmetrische Verfahren	68
5.8 Gesamtvergleich	72
6 Zusammenfassung und Ausblick	75
6.1 Zusammenfassung	75
6.2 Ausblick	76
Literatur	79

Kapitel 1

Einführung

1.1 Motivation

Die Erzeugung von Zwischenbildern in Bildfolgen ist eines der klassischen Probleme in der Videoverarbeitung. Die Erzeugung kann durch ein Interpolationsverfahren geschehen, wofür es in der Literatur viele verschiedene Ansätze gibt. Diese arbeiten mit einer begrenzten Menge an Bildinformationen, um einen Zustand zwischen den Bildern zu schätzen. Als Grundlage für die Interpolationsverfahren kann ein sogenannter optischer Fluss dienen, bei dem es sich um eine Schätzung der Bewegung zwischen den Bildern einer Bildsequenz handelt. Mit der zusätzlichen kontextuellen Information über die Bewegung im Bild können qualitativ hochwertige Zwischenbilder erzeugt werden, die einer einfachen linearen Interpolation überlegen sind. Bei einer linearen Interpolation kommt es typischerweise zu folgendem Problem: Da Objekte sich in aufeinanderfolgenden Bildern an unterschiedlichen Positionen befinden, entstehen bei diesem Ansatz auffällige Doppelungsartefakte im Bild. Genau dem soll durch Verwendung eines optischen Flusses entgegengewirkt werden, indem eine lineare Interpolation entlang der Bewegung im Bild stattfindet. Neben der großen Anzahl an Verfahren für die Bestimmung eines optischen Flusses, gibt es auch zahlreiche Möglichkeiten für dessen Nutzung bei der Zwischenbildinterpolation. Je nach Interpolationsverfahren findet eine Nutzung von mehr oder weniger Bildern einer Bildsequenz statt, aber auch eine mehr oder weniger starke Abhängigkeit vom optischem Fluss kann vorliegen. Es kommt vor, dass sich die Interpolationsverfahren konzeptuell ähneln können, wodurch häufig nicht intuitiv klar ist, wie die qualitativen und quantitativen Unterschiede in der Praxis aussehen. In dieser Arbeit sollen daher die grundlegenden Verfahren für die Zwischenbildinterpolation mit optischem Fluss zusammengefasst und die schwer zu beurteilenden Unterschiede in einer Evaluation geklärt werden.

1.2 Anwendung

Die Zwischenbildinterpolation (mit optischem Fluss) findet in einigen unterschiedlichen Bereichen eine mögliche Anwendung, die im Folgenden kurz zusammengefasst werden.

Erhöhte Frameraten, Slow-Motion und Re-Timing Die Bildwiederholungsrate eines Videos (frame rate) spielt eine zentrale Rolle in der Videoverarbeitung, denn sie ist neben der Auflösung maßgeblich verantwortlich für die Wahrnehmung von Bildsequenzen. Während die Auflösung von Darstellungsgeräten in den letzten Jahrzehnten stetig gewachsen ist, blieb die Bildwiederholungsrate von aktuellen Filmen meist normiert bei etwa 24 Bildern pro Sekunde (Frames per second). Es ist aber

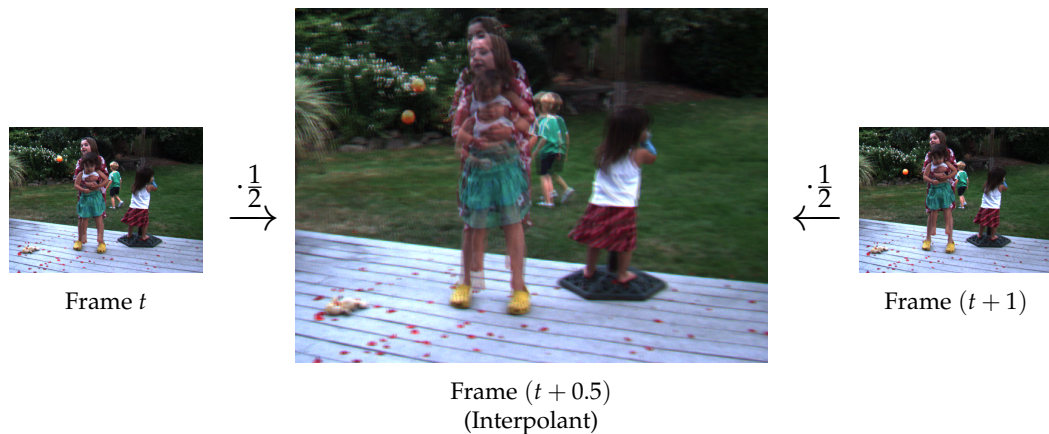


ABBILDUNG 1.1: Bei linearer Interpolation kommt es zu auffälligen Doppelungsartefakten, da unabhängig vom Inhalt der angrenzenden Frames ein einfaches *Blending* (Überlagern oder Zusammenführen) stattfindet. Die vorgestellten Verfahren sollen diesen Effekt reduzieren.

grundlegend von Interesse die Bildwiederholungsrate anzuheben, da so für den Betrachter ein flüssigerer Eindruck der Bewegungen im Bild entsteht. Dies für eine bereits existierende Bildsequenzen umzusetzen, ist meist unter dem Begriff *Frame Rate Up Conversion* (FRUC) wiederzufinden. Dazu gehört insbesondere auch die Erhöhung der Bildrate älterer Aufnahmen hin zum neuen Standard. Diese Restauration findet oft Anwendung bei beispielsweise älteren Stummfilmen aus dem Anfang des letzten Jahrhunderts. Ältere Aufnahmen besitzen neben qualitativen optischen Mängeln auch eine charakteristisch geringere Bildwiederholrate (12-14 fps). Um diese auf den heutigen Standard anzuheben, bedarf es Verfahren, die nur mit den heutzutage vorliegenden Bildinformationen zusätzliche Zwischenbilder bestimmen können. Eine direkte Möglichkeit dies umzusetzen, ist das Duplizieren einzelner Frames. Dies führt aber unter Umständen zu ruckeligen oder ungleichmäßigen Bewegungen. Eine andere Herangehensweise ist das Erzeugen von neuen Zwischenbildern durch Interpolationsverfahren. Eine einfache lineare Interpolation scheint naheliegend, aber wie bereits erwähnt reicht diese nicht aus, da es zu charakteristischen Doppelungsartefakten kommt (siehe Abbildung 1.1). Ein komplexeres Verfahren ist also notwendig, um mit Problemen wie Doppelungsartefakten klarzukommen. Die Zwischenbildinterpolation mit optischem Fluss stellt eine dieser komplexeren Herangehensweisen dar, indem sie unmittelbar auf der Bewegung im Bild aufbaut und so den Problemen entgegenwirkt.

Die berechneten Zwischenbilder können in die bisherigen Bildsequenzen integriert werden, um so die Framerate zu erhöhen, aber auch weitere Verwendungen lassen sich ableiten: Mit der erhöhten Anzahl der Frames ist auch eine nachträgliche Erzeugung eines Slow-Motion-Effekts möglich, durch Erhöhung der Frameanzahl bei gleichbleibender Framerate.

Das Zwischenspiel zwischen erhöhter Framerate und Slow-Motion lässt sich mit der Thematik vom *Re-timing* bündeln: Videosequenzen können unter Umständen in unterschiedlichen Framerates vorliegen und sollen kombiniert werden, indem sie eine angepasste Wiederholrate erhalten. Der Prozess des Angleichens oder des Veränderns der Frameraten lässt sich mithilfe von erzeugten Zwischenbildern umsetzen.

Video Kompression In manchen Umgebungen gibt es bei der Übertragung von Bildsignalen Bandbreitenbegrenzungen. In solchen Fällen kann eine Komprimierung des Bildsignals über einen *Encoder* und *Decoder* durchgeführt werden. Dabei wird ausgenutzt, dass es häufig zu Redundanzen von gleichen Bildinhalten kommt, wodurch es sich anbietet einzelne Bilder zu entfernen und später wiederherzustellen. Dies geschieht in Formaten wie beispielsweise MPEG über eine Motion Estimation, die die übrig gebliebenen Bilder dann geeignet verschiebt, um die entfernten Bilder zu rekonstruieren [18]. Es existieren auch spezielle Ansätze, die diesen Vorgang mit einem optischen Fluss umsetzen und durch eine Zwischenbildinterpolation die entfernten Frames wiederherstellen [20].

Cartoonfilme Die meisten Filme sind heutzutage für gewöhnlich an den 24fps Standard angepasst, wodurch bei handgezeichneten Cartoons die Produktion häufig in sogenannten Doubles oder Triples stattfindet. Das bedeutet, dass gezeichnete Bilder über mehrere Frames wiederholt angezeigt werden (doppelt, dreifach etc.). Der Grund dafür ist zum einen, dass bereits wenige Bilder ausreichen, um eine Bewegung erkennbar zu machen, aber auch um Kosten zu sparen, die beim Zeichnen der Bilder entstehen. Statt alle Bilder von Hand zu erstellen, bietet es sich an Zwischenbilder mithilfe eines Interpolationsverfahrens zu generieren. Damit ist es möglich Bewegungen flüssiger erscheinen zu lassen und nebenbei auch Kosten einzusparen.

Medizintechnik Zwischenbilder können aber auch in ganz anderen Bereichen von Nutzen sein. Einer davon ist zum Beispiel die Medizintechnik. In der Medizin werden häufig organische Proben oder Körper untersucht, was unter dem Mikroskop oder auf digitaler Weise geschieht. Dazu wird meist eine oder mehrere Proben oder Ausschnitte genommen und untersucht. Die Entnahme der Proben durch invasive Eingriffe oder Methoden wie eine Magnetresonanztomographie (MRT) gibt aber nur diskrete einzeln gesampelte Daten zurück. Häufig sind für Analysen aber auch räumliche Daten gewünscht, die mithilfe einer Interpolation zwischen den Samples erzeugt werden können. Es existieren Ansätze, die Volumendaten über eine Interpolation mit optischem Fluss erzeugen können [23, 37].

Bewertung für optische Flüsse Die Zwischenbildinterpolation mit optischem Fluss findet auch Gebrauch bei der Evaluation von optischen Flüssen. Dort werden optische Flüsse anhand erzeugter Zwischenbilder miteinander verglichen. Da es neben der großen Zahl an Verfahren für optische Flüsse auch eine große Anzahl verschiedener Interpolationsverfahren mit optischen Flüssen gibt, ist es nicht einfach die Ergebnisse miteinander zu vergleichen, da in verschiedenen Arbeiten häufig unterschiedliche Interpolationsverfahren verwendet werden. Aus diesem Grund werden in Benchmarks für optische Flüsse meist normierte Verfahren verwendet [2].

1.3 Verwandte Arbeiten

Verfahren für die Zwischenbildinterpolation gibt es in vielerlei Form. Gemein haben die meisten Verfahren, dass sie sich an der Bewegung im Bild orientieren, um besagte Doppelungsartefakte zu minimieren. Man spricht bei dieser Art von Verfahren von *Motion Compensated Frame Interpolation* (MCFI). Für solch eine Interpolation ist eine Information über die Bewegung (*Motion Estimation*) notwendig, die auch wieder auf verschiedene Weisen bestimmt werden kann. Nach Dinh et al. [8] gibt es zwei

wesentliche Kategorien für eine Motion Estimation: block- und pixelweise Schätzungen. Blockweise Schätzungen sind meist mit weniger Berechnungsaufwand verbunden und finden Anwendung in bekannten Encodern wie zum Beispiel MPEG [18]. Eine Interpolation mit optischem Fluss lässt sich nach obiger Definition im Allgemeinen den pixelweisen Ansätzen zuordnen. Diese Arbeit fokussiert sich also auf den pixelweisen Fall und den damit verbundenen Eigenschaften bei der Zwischenbilderzeugung. Sie baut in erster Linie auf der Arbeit von Rakêt et al. [31] auf und führt eine erweiterte Betrachtung verschiedener Interpolationsstrategien durch. Der Fokus liegt insbesondere auf der Interpolation und weniger dem verwendeten optischen Fluss. Des Weiteren bündelt diese Arbeit in gewisser Weise andere Arbeiten [2, 31, 23, 37], die ebenfalls Interpolationsverfahren vorstellen und mit diesen eine Evaluation durchführen. Dabei findet aber fast jedes Mal eine unterschiedliche Vorgehensweise bei der Interpolation statt, die in dieser Arbeit gesammelt und kategorisiert verglichen werden. Gleichbleibende Testdaten und Fehlermaße sollen in dieser Arbeit zusätzlich für einen einheitlichen Vergleich sorgen. An geeigneten Stellen wird bei der Vorstellung der Verfahren auf ähnliche Algorithmen anderer Arbeiten verwiesen.

1.4 Zielsetzung

Ziel dieser Arbeit ist es, verschiedene Interpolationsverfahren auf Basis eines optischen Flusses vorzustellen und diese in einer Evaluation qualitativ und quantitativ miteinander zu vergleichen. Bei den betrachteten Verfahren handelt es sich um Vorwärts- und Rückwärtswarpingstrategien, die auf einer linearen Interpolation entlang der Bewegungstrajektorie eines optischen Flusses basieren. Als Verfahren für den Erzeugung des optischen Flusses soll ein hausinternes Verfahren des Instituts dienen, um dessen Eignung für die Zwischenbildinterpolation zu überprüfen. Für die Evaluation werden Benchmarkdaten für optische Flüsse verwendet, bei denen ausgelassene Bilder als Referenzbild dienen. Als Bewertung der erzeugten Zwischenbilder dienen klassische Fehlermaße aus der Literatur und ein ergänzender visueller Eindruck. Ziel der Evaluation ist es die Überlegungen hinter den Verfahren zu verifizieren und ihre generelle Eignung in der Praxis zu untersuchen.

1.5 Gliederung

Diese Arbeit besitzt folgende Struktur: Zuerst wird in Kapitel 2 ein Überblick zum Thema optische Flüsse gegeben. Danach folgt in Kapitel 3 eine Vorstellung der grundlegenden Warpingverfahren, die neben dem optischen Fluss das zweite zentrale Mittel für die Zwischenbildinterpolation darstellen. Mit den beschriebenen Grundtechniken werden in Kapitel 4 die betrachteten Bildinterpolationsverfahren geordnet eingeführt und formal festgehalten. Im Anschluss findet in Kapitel 5 die Evaluation der vorher vorgestellten Verfahren statt, worauf in Kapitel 6 noch eine abschließende Zusammenfassung und ein Ausblick folgt.

Kapitel 2

Optischer Fluss

In dieser Arbeit stehen optische Flüsse namensgebend im Mittelpunkt, denn die in den späteren Kapiteln vorgestellten Interpolationsverfahren verwenden optische Flüsse als zentrales Mittel, um Zwischenbilder zu erzeugen. Aufgrund dieser Bedeutung gibt dieses Kapitel einen kurzen Überblick über das Thema optischer Fluss im Allgemeinen und einen Bezug zur Zwischenbildinterpolation.

2.1 Grundlagen

Bewegung in Folgen von Bildern zu erkennen, ist ein klassisches Problem aus dem Bereich *Computer Vision* (Maschinensehen). Die Information über die Bewegung findet in zahlreichen Themengebieten Anwendung und kann für viele verschiedene Zwecke genutzt werden, wie zum Beispiel die Detektion von Objekten in der Robotik, die Kompression von Videos oder verschiedene Analysentechniken in der Medizin [7, 10, 18, 33]. Dazu kommen auch die möglichen Anwendungen durch die Zwischenbildinterpolation mit optischem Fluss aus dem letzten Kapitel.



ABBILDUNG 2.1: Zwei mögliche Visualisierungen eines optischen Flusses.

Formal beschreibt ein optischer Fluss \vec{v} ein Vektorenfeld, das eine pixelweise Bewegungstrajektorie zwischen zwei Bildern I_0 und I_1 darstellt. Für optische Flüsse gibt es verschiedene Methoden zur Visualisierung, wovon die zwei üblichsten in Abbildung 2.1 zu sehen sind. Die Erzeugung von optischen Flüssen geschieht häufig mithilfe der Bilder I_0 und I_1 , aber es existieren auch Verfahren, die mit mehreren Bildern arbeiten [19]. Ein besonderer Teil der aktuellen Forschung sind lernbasierte Verfahren unter Nutzung von neuronalen Netzen [24, 35, 17]. Dort werden neuronale Netzwerke unter anderem verwendet, um bisherige variationsbasierte Verfahren noch zu verbessern.

2.2 Herausforderungen

Optische Flüsse versuchen eine möglicherweise komplexe und mehrdimensionale Bewegung auf ein Vektorenfeld abzubilden, weswegen es leicht zu Unterschieden gegenüber der tatsächlichen Bewegung in der Bildfolge kommen kann. Es existieren Szenarien, in denen es Probleme gibt akkurate Ergebnisse zu erzeugen, was später auffällige Auswirkungen auf die Zwischenbildinterpolation haben kann. Das Problem mit der wohl größten Auswirkung auf die Zwischenbildinterpolation ist die Detektion von weitreichenden Bewegungen im Bild. Überschreitet die Bewegung eines Objekts im Bild eine Weite von zu großem Ausmaß, wird es für die Flussverfahren schwierig, die Zusammengehörigkeit der Objekte in aufeinanderfolgenden Bildern zu detektieren. Dadurch kann es passieren, dass die Bewegung einzelner Objekte überhaupt nicht wahrgenommen wird, was problematisch für die Zwischenbildinterpolation mit optischem Fluss ist, da diese versucht sich an der beschriebenen Bewegung der Objekte zu orientieren. Ein anderes wichtiges Szenario für optische Flüsse sind Stellen im Bild, die sich durch die Bewegung von Objekten verdeckt oder offenbart werden (*Occlusion* und *Disocclusion*). Solche Fälle resultieren meist in Lücken im optischem Fluss, die in einem zusätzlichen Schritt gefüllt werden müssen. Dafür gibt es verschiedene Ansätze, wovon einige beispielsweise auf einer Diffusion basieren [10]. So eine Technik wird später auch bei der Zwischenbildinterpolation mit optischem Fluss notwendig sein.

Eine weitere zentrale Frage bei optischen Flüssen ist die Bewertung ihrer Qualität beziehungsweise ihrer Genauigkeit. Das Ziel eines jeden Flusses \vec{v} ist natürlich dem Ground-Truth \vec{v}_{GT} möglichst nahe zu kommen, aber es ist nicht einfach die Abweichungen davon zu bewerten und miteinander zu vergleichen. Es werden häufig Fehlermetriken verwendet, die einen Fehler für einen einzelnen Verschiebungsvektor berechnen. Dazu gehört beispielsweise der *Angular Error* (AE), der für einen Vektor $(u, v)^T$ des Flusses \vec{v} und einem Vektor $(u_{GT}, v_{GT})^T$ des Flusses \vec{v}_{GT} folgendermaßen definiert ist:

$$AE = \cos^{-1} \left(\frac{u \cdot u_{GT} + v \cdot v_{GT} + 1}{\sqrt{u^2 + v^2 + 1} \sqrt{u_{GT}^2 + v_{GT}^2 + 1}} \right) \quad (2.1)$$

Eine andere Möglichkeit ist die euklidische Distanz in Form vom *Endpoint Error* (EE):

$$EE = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2} \quad (2.2)$$

Wie fast jede Fehlermetrik haben auch diese Metriken ihre Schwächen und Bevorzugungen, weswegen es auch praktischere Vergleichsmethoden für optische Flüsse gibt. Eine davon ist die im letzten Kapitel bereits erwähnte Zwischenbildinterpolation, bei der die Qualität des Flusses intuitiver und nachvollziehbarer wird. Über die Zwischenbildinterpolation sind Probleme und Schwächen von optischen Flüssen direkt im erzeugten Zwischenbild erkennbar und es kann zusätzlich ein Vergleich zu einem Referenzbild über weitere Fehlermaße durchgeführt werden. Allerdings gibt es, wie bereits erwähnt, viele Möglichkeiten mithilfe eines optischen Flusses Zwischenbilder zu erzeugen, wodurch für einen repräsentativen Vergleich ein einheitliches Interpolationsverfahren notwendig ist (vgl. Benchmarks wie beispielsweise das Middlebury Benchmark [2]). Ob die vorgestellten Formen der Bewertung wirklich die tatsächliche Qualität eines Flusses widerspiegeln, ist zusammengefasst schwierig zu beurteilen. Das gleiche Problem wird es aber auch bei der kommenden Bewertung von erzeugten Zwischenbildern geben.

Es fällt auf, dass es einige Gemeinsamkeiten bei optischen Flüssen und einer darauf basierenden Zwischenbildinterpolation gibt. Im Laufe dieser Arbeit werden sie an verschiedenen Stellen zum Teil wiederauftauchen.

2.3 ProFlow

Bei dem später in den Experimenten verwendeten Verfahren für die Erzeugung eines optischen Flusses handelt es sich um die Baseline-Variante von ProFlow von Maurer und Bruhn [24]. Obwohl der Fokus dieser Arbeit nicht auf der Erzeugung von optischen Flüssen liegt, soll dieses später verwendete Verfahren in diesem Abschnitt trotzdem zur Vollständigkeit kurz vorgestellt werden.

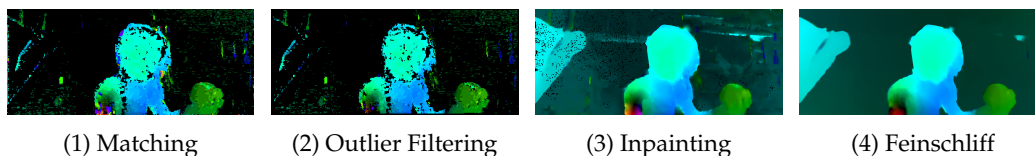


ABBILDUNG 2.2: Die vier Schritte vom ProFlow-Baseline-Verfahren (Quelle [25]).

Beim ProFlow werden mehrere Schritte in einer Pipeline durchlaufen, um einen optischen Fluss zu bestimmen. Maurer und Bruhn erwähnen in ihrer Arbeit mehrere Varianten ihres Verfahrens, wodurch das hier verwendete Baseline-Verfahren einer spezifischen Konfiguration von dort entspricht. Die verwendete Konfiguration besteht aus vier wesentlichen Schritten (siehe Abbildung 2.2), die im Folgenden kurz zusammengefasst werden: In einem ersten Schritt findet ein sogenanntes Matching statt. Dafür wird *coarse-to-fine PatchMatch* (CPM) von Hu et al. [16] verwendet. Im zweiten Schritt werden die Matches aus dem ersten Schritt verwendet und es wird ein Outlier Filtering durchgeführt. Dies geschieht durch einen Forward-Backward-Check, der bereits im CPM enthalten ist. In einem dritten Schritt findet ein sogenanntes Inpainting statt: Durch das vorherige Outlier Filtering entstehen Lücken im Fluss, die mithilfe der *robust interpolation technique* (RIC) von Hu et al. [15] aufgefüllt werden, damit der letzte variationsbasierte Schritt mit einem voll besetzten Fluss arbeiten kann. Dieser letzte Schritt, der variationsbasierte Feinschliff, wird durch *order-adaptive illumination-aware refinement* (OIR) von Maurer et al. [25] realisiert. Der lernbasierte Teil von ProFlow auf Basis von neuronalen Netzwerken ist im Baseline-Verfahren nicht enthalten. In Tabelle 2.1 sind die Ergebnisse von der Baseline-Variante bei einigen Benchmarks zu sehen. Für weitere und genauere Einzelheiten sei nochmals auf die Arbeit von Maurer und Bruhn verwiesen [24].

KITTI 2012		KITTI 2015		Sintel
AEE	BP	AEE	BP	AEE
2.61	8.98	6.82	18.70	1.95

TABELLE 2.1: Ergebnisse des Baseline-Verfahrens von Proflow [24] bei den KITTI Benchmarks [12, 27] und dem MPI Sintel Benchmark [6] für optische Flüsse. Bei AEE handelt es sich um den *Average Endpoint Error* und bei BP um den Anteil schlechter Pixel in Prozent (Quelle [24]).

Kapitel 3

Warping

3.1 Grundlagen

Mit dem Wissen über optische Flüsse kann nun mit den Interpolationsverfahren begonnen werden. Für die Verfahren ist aber noch ein zentrales Thema ziemlich wichtig: Warping. Warping ist eine grundlegende Technik aus dem Gebiet *Image Processing* und wird bei den kommenden Bildinterpolationsverfahren sehr oft als zentrales Mittel verwendet werden. Aus diesem Grund soll in diesem Kapitel Warping als grundlegende Technik kurz vorgestellt und die Verbindung zum optischen Fluss soll verdeutlicht werden.

Warping bezogen auf Bilder ist im Grunde ein einfaches Mapping: Eine Position (x, y) aus einem Bild $I(x, y)$ wird über eine beliebige Funktion $M(x, y)$ auf eine neue Position (x', y') gemappt/zugewiesen:

$$M(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \quad (3.1)$$

Die Begriffe Warping und Mapping seien ab sofort beliebig austauschbar. Anschließend wird beim eigentlichen Warpingvorgang die Bildinformation an die jeweiligen neuen Positionen im Bild verschoben. Die Funktion $M(x, y)$ kann viele verschiedene Effekte realisieren: Translationen, Rotationen, affine Transformation und vieles mehr sind möglich (mehr dazu in [13]). Abbildung 3.1 zeigt exemplarisch drei Beispiele für mögliche Funktionen $M(x, y)$ auf. Um beispielsweise alle Bildinhalte um 50 Pixel nach rechts und 30 Pixel nach oben zu verschieben, wird folgende Mappingfunktion verwendet:

$$M(x, y) = (x + 50, y - 30) \quad (3.2)$$

Neben dem Erzeugen von verschiedenen Effekten wird Warping aber noch in zahlreichen anderen Bereichen verwendet, wo sich später noch zeigen wird, dass es durchaus ähnliche Vorgehensweisen zu den in dieser Arbeit vorgestellten Verfahren gibt. Bei der Umsetzung vom Warping unterscheidet man nach einem *Forward Warping* (Vorwärtswarping) und *Backward Warping* (Rückwärtswarping). Die namensgebende Richtung ergibt sich durch die Nutzung der Mappingfunktion und der damit verbundenen Art der Beschaffung der Bildinformation. Soll mithilfe von Forward Warping ein gewarptes Bild I_{Ziel} erzeugt werden, wird die Bildinformation aus der Bildquelle I_{Quelle} über die Mappingfunktion $M(x, y)$ „nach vorne“ weitergeben:

$$I_{Ziel}(M(x, y)) = I_{Quelle}(x, y) \quad (3.3)$$

Sollte die Funktion $M(x, y)$ auch invertierbar sein, kann ein gewarptes Bild I_{Ziel} alternativ auch über Backward Warping erzeugt werden. Dort wird für jede Position im Zielbild I_{Ziel} die Bildinformation über die inverse Mappingfunktion $M^{-1}(x, y)$

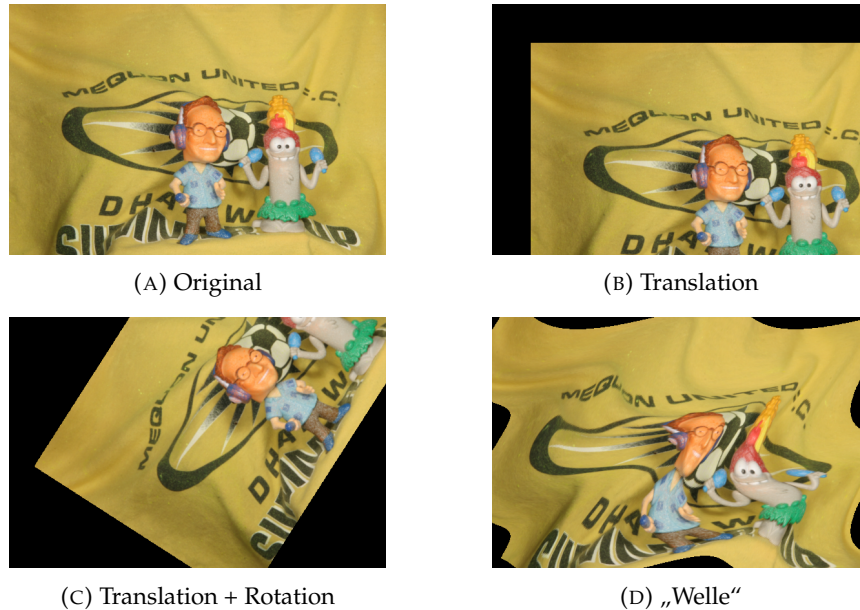


ABBILDUNG 3.1: Verschiedene Mappingfunktionen $M(x, y)$ für das Warping.

aus einer Quelle I_{Quelle} „zurück“ oder „rückwärts“ geholt:

$$I_{Ziel}(x, y) = I_{Quelle}(M^{-1}(x, y)) \quad (3.4)$$

Der wesentliche Unterschied der beiden Warping Arten ist die Domäne, die den Ausgangspunkt der Zuweisung darstellt: Beim Forward Warping sind es die Positionen der Quelle I_{Quelle} und beim Backward Warping sind es die Positionen des gewarpten Ziels I_{Ziel} . In einer technischen Umsetzung entspricht dies auch den Bestandteilen, über die iteriert wird [11]. Während beispielsweise beim Forward Warping für jeden Wert der Quelle I_{Ziel} entschieden wird, welcher Stelle im Zielbild I_{Ziel} er zugeordnet wird, wird beim Backward Warping für jede Stelle im Zielbild I_{Ziel} entschieden, welcher Wert aus der Quelle I_{Quelle} dieser Stelle zugeordnet werden muss.

In Abbildung 3.2 ist der Unterschied der zwei Warping Schemata nochmal zusätzlich visualisiert. Dort zu erkennen ist wie beim Forward Warping die Information von der Quelle nach „vorne“ zum Ziel gegeben wird und wie beim Backward Warping vom Ziel aus in der Quelle nach der Information gesucht wird. Je nach Warping Schema gibt es unterschiedliche Details zu klären (Splating und Sampling), was am kommenden Fall des optischen Flusses als Mappingfunktion nochmal genauer thematisiert wird.

3.2 Optischer Fluss als Mappingfunktion

Wie im letzten Kapitel kennengelernt, beschreiben optische Flüsse die Bewegung von Objekten zwischen zwei Bildern. Der Kerngedanke für Zwischenbildinterpolation mit optischem Fluss ist, dass der zeitlich dazwischen liegende Zustand der Szene I_t mithilfe des optischen Flusses bestimmt werden kann. Dies wird erreicht, indem die Bildinformationen zweier aufeinanderfolgenden Bilder I_0 und I_1 mittels eines optischen Flusses zum Zeitpunkt des Zwischenzustands „gewart“ werden. Das bedeutet, dass jetzt beispielsweise ein optischer Fluss \vec{v}_0 , der die Bewegung von

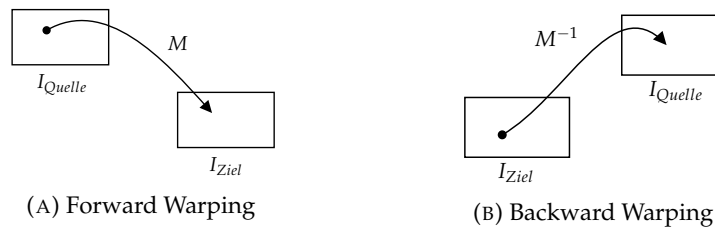


ABBILDUNG 3.2: Visualisierungen der zwei möglichen Warping Schemata.

I_0 nach I_1 beschreibt, als Mappingfunktion M verwendet wird. Natürlich könnte ohne Einschränkung das Gleiche auch mit einem Fluss \vec{v}_1 von Bild I_0 nach I_1 getan werden. Es wird sich in diesem Kapitel aber immer exemplarisch auf den Fluss \vec{v}_0 bezogen. Durch die Verwendung eines optischen Flusses \vec{v}_0 gibt es eine Einschränkung an Definition 3.1, da der optische Fluss normalerweise nur für alle diskreten Stellen im Bild definiert ist:

$$M_{flow}(x, y) : \mathbb{N}^2 \rightarrow \mathbb{R}^2 \quad (3.5)$$

Eine Stelle (x, y) wird in den kommenden Abschnitten häufig über einen Vektor \vec{x} abgekürzt. Wird ein optischer Fluss als Mappingfunktion verwendet, handelt es sich beim erzeugten Effekt im Bild um eine Translation. Im Gegensatz zur vorherigen Translation, die eine einheitliche Verschiebung für jede Stelle (x, y) im Bild vornimmt, findet jetzt eine individuelle Verschiebung für jede Stelle statt. Diese Verschiebung ist jeweils gegeben über den Verschiebungsvektor $(u, v)^T$ des optischen Flusses an dieser Stelle:

$$M_{flow}(x, y) = (x + u, y + v), \quad \begin{pmatrix} u \\ v \end{pmatrix} = \vec{v}_0(x, y) \quad (3.6)$$

Der optische Fluss als Mappingfunktion M_{flow} ist mit den beiden Arten des Warping kompatibel, was in den nächsten zwei Abschnitten für das Forward Warping und das Backward Warping nochmal einzeln vorgestellt wird. Diese grundlegenden Warpings über einen optischen Fluss werden dann später im nächsten Kapitel als Grundtechnik für die Zwischenbildinterpolation verwendet werden. Es wird mit dem Backward Warping begonnen, da dort Konzepte vorgestellt werden, die zum Teil beim Forward Warping wieder aufgegriffen werden.

3.3 Backward Warping

Für einen optischen Fluss \vec{v}_0 , der die Bewegung von einem Bild I_0 nach I_1 beschreibt, ist ein Backward Warping passend zu Definition 3.4 gegeben über:

$$I_{Ziel}(\vec{x}) = I_1(\vec{x} + \vec{v}_0(\vec{x})) \quad (3.7)$$

In Abbildung 3.3 ist gut zu erkennen, wie das Bild I_1 als Quelle für die Bildinformation I_{Quelle} dient und wie die Werte von dort zum Zielbild I_{Ziel} zurückgeholt werden. Bei diesem Backward Warping findet das namensgebende Rückwärtsgehen aber nicht nur im allgemeinen Sinne des Zielbilds I_{Ziel} und Quelle I_{Quelle} statt, sondern im Falle des Flusses \vec{v}_0 auch im zeitlichen Sinne. Es werden die Bildinhalte (des zeitlich später folgenden Bildes I_1) zu den Stellen des früheren Zeitpunkts

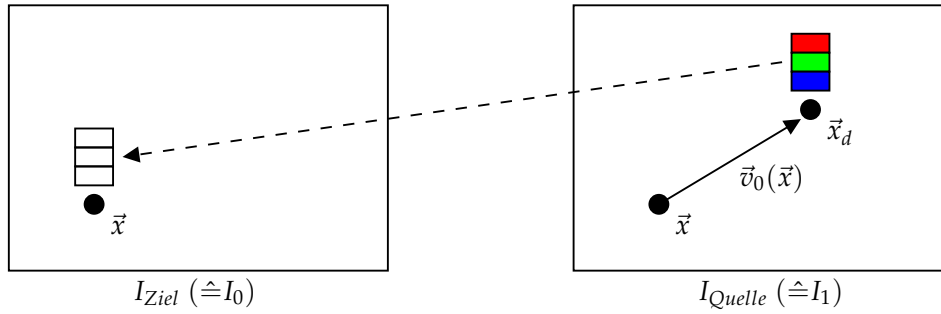


ABBILDUNG 3.3: Visualisierung des Backward Warping über einen optischen Fluss \vec{v}_0 . Es soll durch das bunte Farbtupel (beispielhaft für ein RGB-Bild) die Richtung des Transports von Bildinformation verdeutlicht werden.

in I_0 zurückgeholt. Dies geschieht entlang des optischen Flusses \vec{v}_0 , der die Verbindung zwischen den Bildern I_0 und I_1 darstellt. Dieser Gedanke wird später bei der Zwischenbildinterpolation nochmal wichtig. Für eine Stelle \vec{x} im Zielbild I_{Ziel} muss immer an einer Stelle \vec{x}_d im Quellbild I_{Quelle} ein Farbwert gesampelt werden. Die Stelle wird folgendermaßen berechnet:

$$\vec{x}_d = \vec{x} + \vec{v}_0(\vec{x}) \quad (3.8)$$

Aufgrund der Natur von optischen Flüssen kann es aber sein, dass die Position \vec{x}_d zwischen den Pixelpositionen liegt, da es sich bei optischen Flüssen für gewöhnlich um eine Sammlung von Vektoren handelt, die eine Subpixel-genaue Verschiebung beschreiben (vgl. Kapitel 3). Da es nicht möglich ist von einer Stelle, die sich zwischen den Pixeln befindet, einen Farbwert zu sampeln, müssen die Nachbarpixel für einen interpolierten Farbwert verwendet werden. Die vier Nachbarn einer Stelle zwischen den Pixelpositionen \vec{x}_d sind immer gegeben über:

$$\begin{aligned} \vec{x}_1 &= \lfloor \vec{x}_d \rfloor & \vec{x}_2 &= \lfloor \vec{x}_d \rfloor + (1, 0)^T \\ \vec{x}_3 &= \lfloor \vec{x}_d \rfloor + (0, 1)^T & \vec{x}_4 &= \lfloor \vec{x}_d \rfloor + (1, 1)^T \end{aligned} \quad (3.9)$$

Die Farbwerte an diesen vier Stellen müssen bei der Interpolation noch zusammengeführt werden, was über eine gewichtete Summe geschieht (siehe Abbildung 3.4). Für eine Stelle \vec{x} im Zielbild I_{Ziel} und eine zugehörige Sampleposition \vec{x}_d (siehe Gleichung 3.8) ergibt sich der Farbwert als gewichtete Summe der benachbarten vier Pixel:

$$I_{Ziel}(\vec{x}) = \sum_{i \in N(\vec{x}_d)} \frac{\omega_i \cdot I_1(\vec{x}_i)}{\sum_{i \in N(\vec{x}_d)} \omega_i} \quad (3.10)$$

Dabei entspricht die Nachbarschaft $N(\vec{x}_d)$ den angrenzenden Stellen auf dem Pixelraster für eine beliebige Position \vec{x}_d . In einer solchen Nachbarschaft sind nur die Indizes der vier Nachbarstellen enthalten, die sich im Bildbereich befinden. Die eigentliche Zusammenführung geschieht immer über das individuelle Gewichten der Farbwerte über den Faktor ω_i und dem anschließenden Normalisieren über die Summe aller verwendeten Gewichte. Für die individuelle Gewichtung ω_i gibt es verschiedene Möglichkeiten, wodurch sich unterschiedliche Interpolationsschemata ergeben. In dieser Arbeit werden insgesamt drei mögliche Gewichtungen herangezogen, die in den kommenden drei Abschnitten erklärt werden.

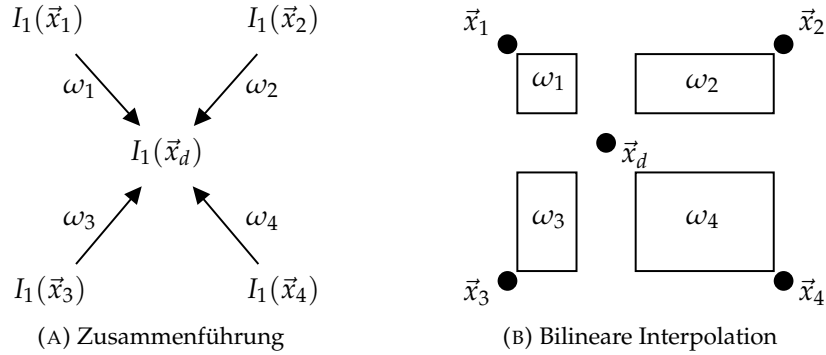


ABBILDUNG 3.4: Links ist das allgemeine Zusammenführen von Farbwerten der Nachbarn um die problematische Stelle \vec{x}_d zu sehen. Rechts sind die unterschiedlichen Gewichte der bilinearen Interpolation durch Flächen veranschaulicht.

3.3.1 Nearest Neighbour Interpolation

Bei diesem Ansatz soll statt dem Farbwert an der problematischen Stelle \vec{x}_d die Nachbarstelle \vec{x}_i verwendet werden, die die kleinste euklidische Distanz besitzt:

$$\omega_i = \begin{cases} 1, & \text{falls } |\vec{x}_i - \vec{x}_d| \text{ minimal für alle } i \in N(\vec{x}_d) \\ 0, & \text{sonst} \end{cases} \quad (3.11)$$

Dabei ist zu beachten, dass wenn der Nachbar \vec{x}_i mit dem kleinsten Abstand nicht mehr im Bild liegen sollte, immer der Nachbar mit dem nächstniedrigeren Abstand verwendet wird. Sollten alle vier Stellen x_i nicht im Bild liegen, schlägt das Sampling fehl und in I_{Ziel} entsteht an der Stelle \vec{x} eine Lücke ohne Farbwert.

3.3.2 Uniformes Mitteln

Beim zweiten Ansatz sollen bei der gewichteten Summe möglichst alle vier Nachbarn einbezogen werden. Dies soll über ein einfaches Mitteln geschehen, bei dem alle Nachbarn gleich gewichtet werden:

$$\omega_i = 1, \quad \text{für alle } i \in N(\vec{x}_d) \quad (3.12)$$

Bei dieser Interpolation, die versucht mit vier Nachbarn zu arbeiten, ist zu beachten, dass je nach tatsächlicher Nachbarschaft $N(\vec{x}_d)$ unterschiedlich viele Werte gemittelt werden. Im Falle von gar keinen Nachbarn in der Nachbarschaft $N(\vec{x}_d)$, liegt auch bei diesem Ansatz eine Lücke im Zielbild I_{Ziel} vor.

3.3.3 Bilineare Interpolation

Eine klassische Methode für das Sampling stellt die bilineare Interpolation dar. Diese Modifikation des vorherigen Ansatzes verwendet anstelle gleicher Gewichte für alle Nachbarn, eine Gewichtung abhängig vom Abstand zur Stelle \vec{x}_d . Diese ergeben sich durch eine lineare Interpolation in beide Bildrichtungen, wodurch sich folgende Gewichte für die Nachbarn ergeben:

$$\begin{aligned} \omega_1 &= (1 - \epsilon_x) \cdot (1 - \epsilon_y) & \omega_2 &= \epsilon_x \cdot (1 - \epsilon_y) & \begin{pmatrix} \epsilon_x \\ \epsilon_y \end{pmatrix} &= \vec{x}_d - \lfloor \vec{x}_d \rfloor \\ \omega_3 &= (1 - \epsilon_x) \cdot \epsilon_y & \omega_4 &= \epsilon_x \cdot \epsilon_y & & \end{aligned} \quad (3.13)$$

Diese Gewichte sind in Abbildung 3.4 nochmal als Flächen visualisiert. Je kleiner die Fläche, desto näher liegt man am Nachbar und desto größer ist die Gewichtung. Es ist zu beachten, dass wie beim vorherigen Ansatz je nach Nachbarschaft $N(\vec{x}_d)$ unterschiedlich viele gewichtete Werte summiert und normalisiert werden. Dadurch liegt ein einheitliches Vorgehen zum vorherigen Ansatz vor, entspricht daher aber nicht ganz der üblichen bilinearen Interpolation, bei der immer alle vier Nachbarn notwendig sind. Die bilineare Interpolation ist in Bildinterpolationsverfahren für das Sampling aus Bildern sehr verbreitet. Sie findet beispielsweise Anwendung im Bildinterpolationsalgorithmus des bekannten Middlebury Benchmarks [2].

3.4 Forward Warping

Für einen optischen Fluss \vec{v}_0 , der die Bewegung von einem Bild I_0 nach I_1 beschreibt, ist ein Forward Warping passend zu Definition 3.3 gegeben über:

$$I_{Ziel}(\vec{x} + \vec{v}_0(\vec{x})) = I_0(\vec{x}) \quad (3.14)$$

In Abbildung 3.5 ist gut der Unterschied zum vorangegangenen Backward Warping zu erkennen. In diesem Fall dient die Quelle der Farbwerte I_{Quelle} zur Orientierung und die Werte von dort sollen über den optischen Fluss \vec{v}_0 nach „vorne“ zur Zielposition \vec{x}_d im Zielbild I_{Ziel} weitergegeben werden. Zuvor diente das Zielbild I_{Ziel} zur Orientierung, die sich die Farbwerte „rückwärts“ zu sich geholt hat. Auch beim Forward Warping fällt bei der Verwendung des Flusses \vec{v}_0 mit der Bewegung zwischen zwei aufeinanderfolgenden Bildern I_0 und I_1 der zeitliche Aspekt auf. Es werden die Bildinhalte durch die beschriebene Verschiebung von ihrer ursprünglichen Position in I_0 zu ihrem späteren Zustand in Bild I_1 gewarpt. Dieses Verhalten wird ebenfalls bei den kommenden Bildinterpolationsverfahren nochmals aufgegriffen werden.

Genau wie beim Backward Warping muss beim Forward Warping speziell behandelt werden, dass die Zielstelle \vec{x}_d zwischen Pixelpositionen liegt. Sie wird wieder berechnet als:

$$\vec{x}_d = \vec{x} + \vec{v}_0(\vec{x}) \quad (3.15)$$

Diese Stelle soll dieses Mal aber nicht zur Bestimmung eines Farbwerts im Bild I_{Quelle} dienen, sondern selbst einen Farbwert erhalten. Einen Farbwert einer Stelle zwischen dem Pixelraster zuzuweisen ist aber nicht möglich, weshalb es erneut notwendig ist in der Nachbarschaft nach Ersatz zu suchen. Es wird dann über ein sogenanntes *Splatting* der übergebene Farbwert nicht an die berechnete Stelle \vec{x}_d gesetzt,

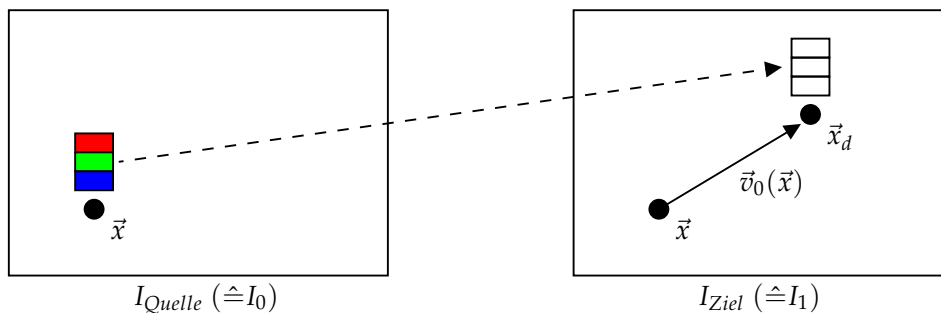


ABBILDUNG 3.5: Visualisierung vom Forward Warping über einen optischen Fluss \vec{v}_0 . Es soll durch das bunte Farbtupel (beispielhaft für ein RGB-Bild) die Richtung des Transports von Bildinformation verdeutlicht werden.

sondern auf die Nachbarn verteilt oder sinnbildlich in diesem Bereich auf das Bild „geklatst“ (englisch: *to splat*). Beim Splatting gibt es zwei wesentliche Aspekte zu beachten: In einem ersten Schritt muss entschieden werden, auf welche der möglichen Nachbarn der Farbwert verteilt werden soll und in einem zweiten Schritt, wie mit Kollisionen umzugehen ist, die bei Stellen entstehen, denen mehrere Farbwerte zugewiesen wurden.

Im Folgenden werden zwei Verteilungsstrategien und drei Kollisionsbehandlungen vorgestellt, sodass sich insgesamt sechs Kombinationen ergeben. Jede dieser Kombinationen entspricht dann einer vollständigen Splattingstrategie und stellt ein Vorgehen für die Verteilung von Farbwerten dar.

3.4.1 Verteilungsstrategien

Die erste Entscheidung die für eine Stelle \vec{x}_d getroffen werden muss, ist welchen Positionen auf dem Pixelraster im Zielbild I_{Ziel} der Farbwert zugeordnet werden soll. In dieser Arbeit wird für diese Entscheidung in erster Linie auf die Konzepte der bereits vorgestellten Samplingstrategien beim Backward Warping zurückgegriffen (vgl. Abschnitt 3.3). Dort vorgestellt wurden ein Nearest Neighbour Ansatz und zwei Ansätze, die vier Nachbarn um die Stelle \vec{x}_d gewichten und aufsummieren. Ein sehr ähnliches Vorgehen soll auf das Splatting übertragen werden: Es wird entweder eine Verteilung auf den nächsten Nachbarn oder auf vier Nachbarn vorgenommen. Dazu sind erneut die möglichen Nachbarn um die Stelle \vec{x}_d notwendig:

$$\begin{aligned} \vec{x}_1 &= \lfloor \vec{x}_d \rfloor & \vec{x}_2 &= \lfloor \vec{x}_d \rfloor + (1, 0)^T \\ \vec{x}_3 &= \lfloor \vec{x}_d \rfloor + (0, 1)^T & \vec{x}_4 &= \lfloor \vec{x}_d \rfloor + (1, 1)^T \end{aligned} \quad (3.16)$$

Formal festgehalten wird die Verteilung in einer Menge $V(\vec{x}_d)$. Darin enthalten sind für eine Stelle \vec{x}_d je nach Verteilung ein Teil der vier möglichen Nachbarn \vec{x}_i . Die Menge V wird später bei der Kollisionsbehandlung nochmal wichtig werden.

Ein Nachbar (1N) Es soll ein einziger Nachbar \vec{x}_i als Alternative ausgewählt werden, der die kleinste euklidische Distanz zur Stelle \vec{x}_d besitzt. Da Nachbarn möglicherweise außerhalb des Bildes liegen können, wird bei der Verteilung wieder mit der Nachbarschaft $N(\vec{x}_d)$ gearbeitet. Darin enthalten sind wieder die Indizes der vier Nachbarn, die sich noch im Bild I_{Ziel} befinden. Damit ergibt sich die Menge V mit:

$$V(\vec{x}_d) = \{ \vec{x}_i \mid i \in N(\vec{x}_d) \text{ und } |\vec{x}_i - \vec{x}_d| \text{ minimal} \} \quad (3.17)$$

Vier Nachbarn (4N) Durch die Verteilung auf nur einen Nachbar kann es im resultierenden Bild I_{Ziel} zu Lücken kommen, denn es ist nicht gewährleistet, dass alle Bildbereiche von einem optischem Fluss anvisiert werden. Um die Anzahl der Lücken zu reduzieren, können alternativ immer alle vier angrenzenden Nachbarn bei der Verteilung in Betracht gezogen werden. Damit ergibt sich eine Menge V mit:

$$V(\vec{x}_d) = \{ \vec{x}_i \mid i \in N(\vec{x}_d) \} \quad (3.18)$$

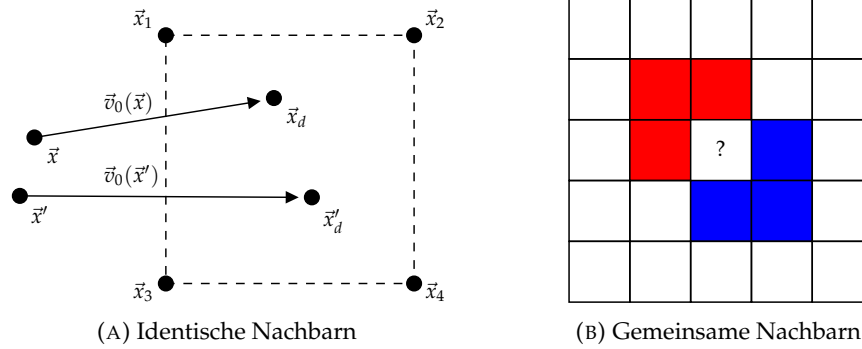


ABBILDUNG 3.6: Links ist der Fall von zwei identischen Nachbarschaften im Zielbild I_{Ziel} dargestellt. Dort besitzen die verschobenen Stellen \vec{x} und \vec{x}' die gleichen vier Nachbarn. Rechts sieht man den Fall von sich überlappenden Nachbarregionen, dargestellt für den Fall einer 4N-Verteilung. Die roten und blauen Stellen markieren jeweils die Nachbarn welche eine unterschiedliche Farbe aus I_0 erhalten.

3.4.2 Kollisionsbehandlungen

Beim Forward Warping kann es bei der Verteilung von Farbwerten durch das Splatting zu Kollisionen kommen. Diese Kollisionen entstehen zum Beispiel, wenn durch eine Verschiebung des optischen Flusses \vec{v}_0 zwei Stellen \vec{x} und \vec{x}' aus I_0 die selben vier Nachbarn im Interpolanten besitzen (siehe Abbildung 3.6 links). Eine andere Ursache sind Überlappungen der Nachbarstellen durch die gewählte Verteilungsstrategie (siehe Abbildung 3.6 rechts). Die entstehenden Kollisionen können, wie die Verteilungsstrategie beim Splatting, auf verschiedene Weisen behandelt werden. In dieser Arbeit werden insgesamt drei verschiedene Behandlungen herangezogen, wovon zwei wieder eine starke Nähe zu den Samplingstrategien beim Backward Warping besitzen (vgl. Abschnitt 3.3). Diese zwei Behandlungen (Uniform und Gewicht genannt) lösen die Kollision mehrerer Farbwerte erneut durch ein gewichtetes Aufsummieren und eine anschließende Normalisierung. Der andere Ansatz (Austausch) funktioniert etwas anders und basiert auf dem *Winner-takes-all*-Prinzip (Gewinner bekommt alles).

Uniform Um mit Mehrfachtreffern einer Stelle im Zielbild I_{Ziel} umzugehen, sollen alle übergebenen Farbwerte zunächst an jeder Stelle gewichtet, aufsummiert und nach dem abgeschlossenen Warpingvorgang normalisiert werden. Wie bereits erwähnt geschieht das Aufsummieren mit der zugewiesenen Gewichtung, wofür bei diesem Ansatz jeder verteilte Pixel $\vec{x}_i \in V(\vec{x}_d)$ das Gewicht Eins zugeordnet bekommt:

$$\omega_i = 1, \quad \text{für alle } \vec{x}_i \in V(\vec{x}_d) \quad (3.19)$$

Algorithmus 1 beschreibt zur Verdeutlichung dieses Konzepts den Ablauf beim Forward Warping mit der Kollisionsbehandlung Uniform und Gewicht. Dort wird zunächst eine Initialisierung von zwei Arrays für das Speichern der aufsummierten Farbwerte (Array I_{Ziel}) und der aufsummierten Gewichte (Array G) vorgenommen. Danach folgt in einer Schleife über alle Stellen \vec{x} in Bild I_0 die Bestimmung der Zielposition \vec{x}_d für den übergebenen Farbwert $I_0(\vec{x})$. Anschließend wird die Verteilung $V(\vec{x}_d)$ festgelegt, über welche im Anschluss iteriert wird. Dort wird für jede enthaltene Stelle \vec{x}_i das zugehörige Gewicht ω_i bestimmt, womit der übergebene Farbwert an Stelle des verteilten Pixels x_i im Array I_{Ziel} gewichtet aufaddiert wird. Für die spätere Normalisierung wird im anderen Array G das verwendete Gewicht an gleicher

Algorithmus 1 : Forward Warping mit gewichtetem Aufsummieren.

```

Input : Bild  $I_0$  und Fluss  $\vec{v}_0$ 
Output : Bild  $I_{Ziel}$ 

Initialisiere leeres Array  $I_{Ziel}$  // aufsummierte gewichtete Farbwerte
Initialisiere leeres Array  $G$  // aufsummierte Gewichte

foreach Stelle  $\vec{x}$  in  $I_0$  do
     $\vec{x}_d \leftarrow \vec{x} + \vec{v}_0$ 
    Bestimme  $V(\vec{x}_d)$  // Nearest Neighbour oder alle Nachbarn
    foreach  $\vec{x}_i \in V(\vec{x}_d)$  do
        Bestimme  $\omega_i$  // uniform oder gewichtet
         $I_{Ziel}[\vec{x}_i] \leftarrow I_{Ziel}[\vec{x}_i] + I_0[\vec{x}] \cdot \omega_i$ 
         $G[\vec{x}_i] \leftarrow G[\vec{x}_i] + \omega_i$ 
    end
end

foreach Stelle  $\vec{x}$  in  $I_{Ziel}$  do
    if  $G[\vec{x}] > 0$  then
         $I_{Ziel}[\vec{x}] \leftarrow I_{Ziel}[\vec{x}] / G[\vec{x}]$  // Normalisierung
    else
        Markiere  $\vec{x}$  als Lücke
    end
end

```

Stelle \vec{x}_i ebenfalls aufaddiert. Zuletzt wird in einer weiteren Schleife die Normalisierung aller Farbwerte im Array I_{Ziel} vorgenommen. Dies wird durch eine Teilung der aufsummierten Farbwerte aus dem Array I_{Ziel} durch die Summe der Gewichte aus dem Array G realisiert. Die Normalisierung findet allerdings nur an Stellen im Array statt, für die auch ein aufsummierter Farbwert bestimmt werden konnte, was genau an den Stellen im Array der Fall ist, wo das gleichförmige Array G einen Wert größer als Null besitzt.

Gewicht Eine andere Möglichkeit des Normalisierens besteht darin, die Farbwerte nicht mehr gleich, sondern unterschiedlich stark zu gewichten. Motiviert durch die bilineare Interpolation, soll für jeden verteilten Pixel $\vec{x}_i \in V(\vec{x}_d)$ ein unterschiedliches Gewicht verwendet werden. In diesem Ansatz erhält daher jeder verteilte Pixel $\vec{x}_i \in V(\vec{x}_d)$ das Gewicht von der bilinearen Interpolation:

$$\begin{aligned} \omega_1 &= (1 - \epsilon_x) \cdot (1 - \epsilon_y) & \omega_2 &= \epsilon_x \cdot (1 - \epsilon_y) & \begin{pmatrix} \epsilon_x \\ \epsilon_y \end{pmatrix} &= \vec{x}_d - \lfloor \vec{x}_d \rfloor & (3.20) \\ \omega_3 &= (1 - \epsilon_x) \cdot \epsilon_y & \omega_4 &= \epsilon_x \cdot \epsilon_y \end{aligned}$$

Durch die unterschiedlich starken Gewichte ist zu beachten, dass nach dem Warpingvorgang bei den aufsummierten Gewichten keine Summe von Eins entstehen muss. Dies ist auch der Grund, warum im allgemeinen Algorithmus 1 nicht einfach durch die Anzahl von Treffern einer Stelle geteilt wird. Neben den unterschiedlichen Gewichten funktioniert diese Behandlung aber analog zum uniformen Ansatz. Wegen der starken Verbindung zur bilinearen Interpolation wird diese Kollisionsbehandlung in Kombination mit einer 4N-Verteilung von Williams et al. eine „bilineare Verteilung“ genannt [37].

Algorithmus 2 : Forward Warping mit Austausch von Farbwerten.

```

Input : Bild  $I_0$  und Fluss  $\vec{v}_0$ 
Output : Bild  $I_{Ziel}$ 

Initialisiere leeres Array  $I_{Ziel}$  // gemerkter Farbwert
Initialisiere leeres Array  $G$  // gemerktes Gewicht

foreach Stelle  $\vec{x}$  in  $I_0$  do
   $\vec{x}_d \leftarrow \vec{x} + \vec{v}_0$ 
  Bestimme  $V(\vec{x}_d)$  // Nearest Neighbour oder alle Nachbarn
  foreach  $\vec{x}_i \in V(\vec{x}_d)$  do
    Bestimme  $\omega_i$  // nach dem gewichteten Verfahren
    if  $\omega_i > G[\vec{x}_i]$  then
       $I_{Ziel}[\vec{x}_i] \leftarrow I_0[\vec{x}]$ 
       $G[\vec{x}_i] \leftarrow \omega_i$ 
    end
  end
end

foreach Stelle  $\vec{x}$  in  $I_{Ziel}$  do
  if  $G[\vec{x}] == 0$  then
    Markiere  $\vec{x}$  als Lücke
  end
end

```

Austausch Eine ganz andere Art des Umgangs mit Kollisionen ist der generelle Verzicht auf eine Form von Normalisierung. Stattdessen erhält ein Pixel immer nur einen zugeordneten Wert und später bei einer Kollision wird dieser Wert gegebenenfalls ausgetauscht. Eine Möglichkeit wäre, ausgehend vom gewichteten Normalisieren des vorherigen Abschnitts, immer nur den Farbwert zu wählen, der durch das Warping dem Pixel am nächsten kam. Das bedeutet, dass am Ende des Warpingvorgangs an jeder Stelle im Interpolanten nur ein einzelner Wert aus I_0 vorliegt. Der Gedanke hinter diesem Vorgehen ist, den Einfluss der anderen Pixel aus I_0 zu unterbinden, da durch die Mischung von verschiedenen Farbwerten ein schlechteres Ergebnis entstehen kann, als bei einer Entscheidung für nur einen optimalen Farbwert. Es stellt sich heraus, dass auch solche Herangehensweisen durchaus vergleichbare Ergebnisse liefern können. Baker et al. [2] benutzen beispielsweise eine ähnliche Art der Kollisionsbehandlung bei ihrem verwendeten Forward Warping. Auch dort findet ein kompletter Austausch von Werten statt, allerdings ist er dort nicht abhängig von der Distanz zu einer gültigen Pixelposition, sondern von einer photometrischen Bewertung.

Algorithmus 2 macht den Unterschied zu den zwei vorherigen Ansätzen nochmal deutlich. Es wird zu Beginn noch analog vorgegangen, aber nach Bestimmung des Gewichts ω_i für eine Stelle \vec{x}_i (immer nach dem gewichteten Verfahren) findet kein Aufsummieren, sondern ein Vergleich statt. Es wird immer das aktuelle Gewicht ω_i für eine Stelle \vec{x}_i mit dem bisherigen höchsten Gewicht an dieser Stelle verglichen. Sollte das aktuelle Gewicht größer sein, liegt der aktuell übergebene Farbwert näher an der verteilten Stelle \vec{x}_i , woraufhin der alte Farbwert und das alte Gewicht im Array an dieser Stelle ausgetauscht werden sollen. Durch den kompletten Austausch von Farbwerten liegt nach dem Warpingvorgang im Array I_{Ziel} und im Array

G ein einzelner Farbwert und ein dazugehöriges Gewicht vor. Eine Normalisierung von mehreren aufsummierten Farbwerten ist daher nicht mehr notwendig. Was aber noch, wie bei den vorherigen Ansätzen getan werden kann, ist die Bestimmung von Lücken im Bild. Diese äußern sich wieder über leere Einträge im Array G .

3.5 Inpainting

Wie in den vorangegangenen Abschnitten gezeigt, kann es bei den verschiedenen Warpingverfahren zu Lücken im erzeugten Bild kommen, für die kein Farbwert gefunden werden konnte. Um dennoch ein vollständiges Bild zu erzeugen, bedarf es einer nachträglichen Behandlung, die allein über die gegebenen Farbwerte, die Lücken schließen kann. Dies soll über ein sogenanntes *Inpainting* geschehen, bei dem es sich um ein Verfahren handelt, das die Nachbarn einer Lücke betrachtet und daraus ableitet, welche Farbwerte in den Lücken befinden müssten (siehe Abbildung 3.7).

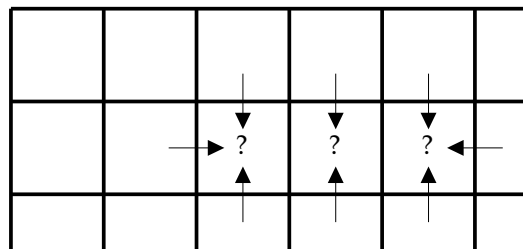


ABBILDUNG 3.7: Visualisierung vom Inpainting. Die Stellen, die durch den vorangegangenen Warpingvorgang nicht getroffen wurden (markiert mit einem Fragezeichen), werden mit der Bildinformation der Nachbarstellen aufgefüllt.

Beim Inpainting sei in diesem Abschnitt immer vom einem allgemeinen Bild u die Rede, das eine beliebige Anzahl an (Farb-) Kanälen besitzen kann. Für dieses allgemeine Bild u werden die kommenden Gleichungen und Verfahren alle nur für den Fall mit einem Kanal angegeben, da später bei der technischen Umsetzung des Inpaintings die vorgestellten Verfahren für jeden Kanal unabhängig voneinander durchgeführt werden. Die Anzahl der Kanäle spielt daher keine Rolle.

3.5.1 Diffusion

Sei f ein kontinuierliches Bild mit einer Domäne Ω . Dieses Bild besitzt eine Region Ω_K mit vorhandenen Farbwerten und eine Region mit Lücken Ω_{leer} . Diese Lücken sollen nun über eine ungerichtete Diffusion gefüllt werden, wodurch ein gefülltes Bild u entsteht. Die Diffusion im Bild basiert auf dem gleichnamigen physikalischen Prozess, bei dem in einem Medium ein Konzentrationsausgleich von beispielsweise einer Flüssigkeit oder Gas stattfindet. Die Diffusion wird über eine Diffusionsgleichung beschrieben, die folgendermaßen definiert ist:

$$\begin{aligned}
 0 &= \Delta u(x, y) \\
 &= \operatorname{div}(\nabla u(x, y)) & (x, y)^T \in \Omega_{leer} \\
 &= u_{xx}(x, y) + u_{yy}(x, y)
 \end{aligned} \tag{3.21}$$

Dabei entspricht Δ dem Laplace Operator, $\nabla u = (u_x, u_y)^T$ dem Gradienten von u und $\operatorname{div}(u) = \nabla^T u$ der Divergenz. Die zuvor genannte Gleichung gilt immer genau dann, wenn die partiellen Ableitungen des Bildes u in x - und y - Richtung (u_{xx} und

u_{yy}) gegen Null gehen. Dies ist der Fall, wenn es nur noch wenig Unterschiede zwischen den Farbwerten in den Lücken des Bildes gibt, wodurch der Ausgleich der (Farb-) Konzentrationen modelliert wird. Der Diffusionsvorgang ist an zwei zentrale Bedingungen geknüpft. Es ist wichtig, dass die Diffusion auch wirklich nur im Bereich der Lücken des Bildes stattfindet, wodurch sichergestellt wird, dass schon vorhandene Bildinhalte nicht wieder zerstört werden. Es gilt daher dauerhaft für besetzte Stellen $(x, y)^T \in \Omega_K$ folgende Eigenschaft:

$$u(x, y) = f(x, y), \quad (x, y)^T \in \Omega_K \quad (3.22)$$

Für das gesuchte Bild u gilt beim Diffusionsvorgang außerdem am Bildrand $\partial\Omega$ eine reflektierende Randbedingung:

$$\vec{n}^T \nabla u = 0, \quad (x, y)^T \in \partial\Omega \quad (3.23)$$

Dabei entspricht \vec{n} einem Normalenvektor und ∇u wieder dem Gradienten u . Diese Bedingung modelliert ein in sich geschlossenes Medium für die Diffusion, bei dem die Konzentrationen den Bildbereich nicht verlassen und damit, angelehnt an den physikalischen Prozess, nicht verloren gehen können.

3.5.2 Diskretisierung

Die Diffusionsgleichung 3.21 besitzt zwei partielle Ableitungen, was sie zu einer partiellen Differenzialgleichung macht (PDE). Diese muss numerisch gelöst werden, wofür in diesem Abschnitt eine numerische Approximation angegeben wird. Dabei ergibt sich ein Gleichungssystem, das mit einem iterativen Verfahren wie dem Jacobi-Verfahren oder Gauß-Seidel-Verfahren gelöst werden kann.

Für die Diffusionsgleichung 3.21 wird eine Approximation auf Basis von zentralen Differenzen verwendet. Damit ergibt sich eine Gleichung für jede diskrete Stelle $(i, j) \in \Omega_{leer}$ im Bild, die folgendermaßen definiert ist:

$$\begin{aligned} 0 &= u_{xx}(i, j) + u_{yy}(i, j) \\ &= \sum_{(\bar{i}, \bar{j}) \in N_4^{\Omega_{leer}}(i, j)} \frac{u_{\bar{i}, \bar{j}} - u_{i, j}}{h^2} + \sum_{(\bar{i}, \bar{j}) \in N_4^{\Omega_K}(i, j)} \frac{f_{\bar{i}, \bar{j}} - u_{i, j}}{h^2} \end{aligned} \quad (3.24)$$

Hier bezeichnet h die Schrittweite und $N_4(i, j)$ immer die Menge an Nachbarn für eine Position (i, j) , die sich noch im Bildbereich befinden. Je nach Position gibt es zwei (Ecke), drei (Kante) oder vier Nachbarn (siehe Abbildung 3.8). Die zusätzlich angegebene Domäne einer solchen Nachbarschaft schränkt noch ein, um was für eine Art von Nachbarn es sich handelt. Bei einem Zusatz Ω liegen beliebige Nachbarn vor. Bei einem Zusatz Ω_{leer} liegen Nachbarn vor, die Lücken im Bild darstellen und in Ω_{leer} enthalten sind. Beim Zusatz Ω_K handelt es sich um Nachbarn, die durch die Diffusion nicht mehr gefüllt werden müssen und in Ω_K liegen. Von der vorgestellten Gleichung gibt es insgesamt $|\Omega_{leer}|$ Stück, wodurch ein großes Gleichungssystem der Form $Ax = b$ vorliegt. Dabei ist x ein eindimensionaler Vektor, der aus den gesuchten $u_{i, j}$ besteht und $|\Omega_{leer}|$ viele Einträge besitzt (jede Lücke im Bild entspricht einem Eintrag). Bei der Systemmatrix A handelt es sich um eine Matrix der Größe $|\Omega_{leer}| \times |\Omega_{leer}|$, die in jeder Zeile die Gewichtung der Nachbarn eines Eintrags

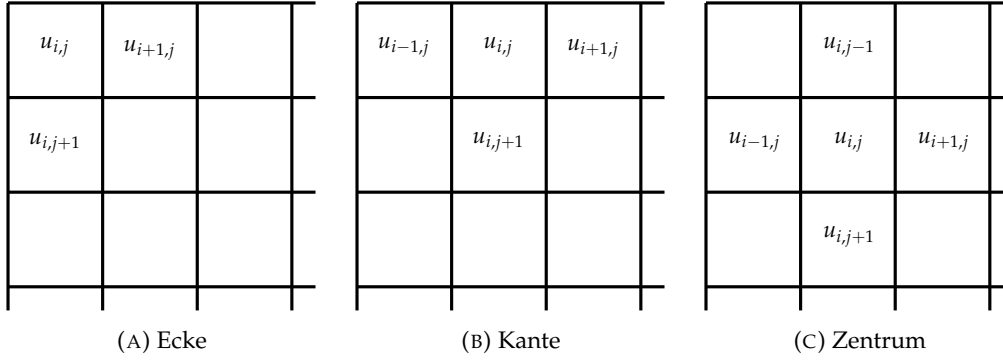


ABBILDUNG 3.8: Die verschiedenen Fälle der Gewichtung beim Inpainting. Je nach Anzahl der Nachbarn, werden unterschiedlich viele Nachbarn verwendet.

aus x besitzt. Dieses entstandene Gleichungssystem über eine Variante der Gauß-Elimination zu lösen stellt einen relativ großen Aufwand dar, da die Systemmatrix A je nach Anzahl der Lücken im Bild sehr groß sein kann.

Bei der Systemmatrix A handelt es sich außerdem um eine dünn besetzte Matrix, was folgenden Grund hat: In einer Zeile der Matrix sind immer bis zu vier Einträge vorhanden und der Rest der Matrix besteht aus Nullen. Die Anzahl an Einträge ist dabei an die Anzahl der Nachbarn geknüpft, da an einer Ecke oder Kante weniger Nachbarpositionen in einer Gleichung 3.24 involviert sind. Aus diesem Grund bietet sich ein iteratives Lösungsverfahren an, weswegen im Folgenden ein iteratives Jacobi-Verfahren als Alternative zum Gauß-Algorithmus vorgeschlagen wird. Dazu werden in einer Gleichung 3.24 einer Position $(i, j) \in \Omega_{leer}$, die Summen zunächst auseinander geschrieben. Für $h = 1$ ergibt sich:

$$\begin{aligned}
 0 &= \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_{leer}}(i, j)} (u_{\tilde{i}, \tilde{j}} - u_{i, j}) + \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_K}(i, j)} (f_{\tilde{i}, \tilde{j}} - u_{i, j}) \\
 &= \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_{leer}}(i, j)} (u_{\tilde{i}, \tilde{j}}) - \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_{leer}}(i, j)} (u_{i, j}) + \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_K}(i, j)} (f_{\tilde{i}, \tilde{j}}) - \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_K}(i, j)} (u_{i, j})
 \end{aligned} \tag{3.25}$$

Danach werden die Summen der $u_{i, j}$ auf die linke Seite gebracht und es erfolgt ein Zusammenführung, da $N_4^{\Omega_{leer}}(i, j) \cup N_4^{\Omega_K}(i, j) = N_4^{\Omega}(i, j)$:

$$\begin{aligned}
 \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_{leer}}(i, j)} (u_{i, j}) + \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_K}(i, j)} (u_{i, j}) &= \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_{leer}}(i, j)} (u_{\tilde{i}, \tilde{j}}) + \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_K}(i, j)} (f_{\tilde{i}, \tilde{j}}) \\
 \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega}(i, j)} (u_{i, j}) &= \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_{leer}}(i, j)} (u_{\tilde{i}, \tilde{j}}) + \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_K}(i, j)} (f_{\tilde{i}, \tilde{j}}) \\
 |N_4^{\Omega}(i, j)| \cdot u_{i, j} &= \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_{leer}}(i, j)} (u_{\tilde{i}, \tilde{j}}) + \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_K}(i, j)} (f_{\tilde{i}, \tilde{j}})
 \end{aligned} \tag{3.26}$$

Durch eine abschließende Division mit $|N_4^\Omega(i, j)|$ ergibt sich ein iteratives Jacobi-Schema, womit eine Lösung schrittweise approximiert werden kann:

$$u_{i,j}^{(k+1)} = \frac{\sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_{leer}}(i, j)} u_{\tilde{i}, \tilde{j}}^{(k)} + \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_K}(i, j)} f_{\tilde{i}, \tilde{j}}}{|N_4^\Omega(i, j)|} \quad (3.27)$$

Durch das iterative Annähern einer Lösung ergibt sich eine schrittweise Füllung von außen hin zum Zentrum der Lücke. Aus diesem Grund ist manchmal auch von einem *Outside-in*-Verfahren die Rede [31].

3.5.3 Beschleunigung

Eine Frage, die sich für den Anwendungsfall bei Zwischenbildinterpolation stellt, ist die notwendige Anzahl an Iterationen. Durch die pixelweise Anwendung des Löser gibt es ein Auffüllen der Lücken mit einer festen Geschwindigkeit von einem Pixel pro Iteration. Es ist also klar, dass dadurch insgesamt mindestens so viele Iterationen wie die Größe der Lücke notwendig sind. Da beim Inpainting die Größen der Lücken von Frame zu Frame variieren können, gibt es im Allgemeinen den Wunsch nach einem möglichst schnellen Konvergieren des Inpaintings, um die Iterationsanzahl möglichst niedrig zu halten. Aus diesem Grund wird statt dem Jacobi Verfahren das Gauß-Seidel-Verfahren verwendet, das zusätzlich noch mit *Successive Over Relaxation* (SOR) beschleunigt wird. Der Gauß-Seidel-Löser hat die Form:

$$u_{i,jGS}^{(k+1)} = \frac{\sum_{(\tilde{i}, \tilde{j}) \in N_{4,-}^{\Omega_{leer}}(i, j)} u_{\tilde{i}, \tilde{j}}^{(k+1)} + \sum_{(\tilde{i}, \tilde{j}) \in N_{4,+}^{\Omega_{leer}}(i, j)} u_{\tilde{i}, \tilde{j}}^{(k)} + \sum_{(\tilde{i}, \tilde{j}) \in N_4^{\Omega_K}(i, j)} f_{\tilde{i}, \tilde{j}}}{|N_4^\Omega(i, j)|} \quad (3.28)$$

Dabei wird für eine Stelle (i, j) erneut ein Mitteln der Nachbarn durchgeführt, allerdings werden jetzt Werte von Nachbarn aus unterschiedlichen Iterationsschritten verwendet. Eine weitere Erweiterung der Nachbarschaft $N_4(i, j)$ gibt daher an, aus welcher Iteration die Werte der Nachbarn stammen. $N_{4,-}$ bezeichnet die Werte der Nachbarn, die in der aktuellen Iteration schon berechnet worden sind und $N_{4,+}$ sind die Werte der Nachbarn, die in der aktuellen Iteration noch berechnet werden müssen. Durch das frühzeitige Verwenden von bereits aktualisierten Werten, ergibt sich ein insgesamt schneller konvergierendes Lösungsverfahren. Der Gauß-Seidel-Löser soll zum Schluss noch zusätzlich mit SOR beschleunigt werden. Dazu wird ein fester (nicht optimierter) Relaxationsparameter $\omega = 1.95$ gewählt. Damit ergibt sich die Berechnungsvorschrift für das endgültige Verfahren:

$$u_{i,j}^{(k+1)} = u_{i,j}^{(k)} + 1.95 \cdot (u_{i,jGS}^{(k+1)} - u_{i,j}^{(k)}) \quad (3.29)$$

Durch die Verwendung von SOR ist zu beachten, dass es zeitweise zu ungültigen Farbwerten kommen kann. Dies resultiert aus dem möglichen Überschätzen der Farbwerte, da bei SOR eine Art Extrapolation stattfindet. Aus diesem Grund ist eine ausreichend hohe Anzahl an Iterationen wichtig, um ein gültiges konvergierendes Ergebnis zu erhalten. Im späteren Kapitel 5 wird die notwendige Iterationsanzahl und die erhoffte Beschleunigung durch SOR noch genauer untersucht.

3.6 Anwendung

Image Warping wird, neben der Zwischenbildinterpolation mit optischem Fluss, in vielen unterschiedlichen Gebieten genutzt. In manchen Fällen liegt die Anwendung dort sogar konzeptuell nah an der Zwischenbildinterpolation mit optischem Fluss.

Optischer Fluss Passend zum Thema findet Warping häufig Verwendung in vielen Verfahren für optische Flüsse [5, 30]. Warping wird dort als Technik in *coarse-to-fine*-Methoden verwendet [5, 4], um mit besonders großen Verschiebungen im Bild klarzukommen. Neben dieser weit verbreiteten Methode gibt es aber auch speziell Ansätze, die Alternativen zum Warping vorschlagen [34]. In aktuell immer häufiger erscheinenden Verfahren, die auf neuronalen Netzwerken basieren, findet sich Warping auch häufiger in einer sogenannten *Warping Layer* wieder [17, 35]. Dort passiert nichts anderes als das vorgestellte Backward Warping aus Abschnitt 3.3, bei dem ein Bild um einen bisherigen optischen Fluss zu einem anderen Bild gewarpt wird. Dies wird dort häufig getan, um eine (Helligkeits-) Differenz zu berechnen, die dann in weitere Berechnungen einfließt [17].

Bildstabilisierung Warping findet auch Anwendung in der Stabilisierung von Bildsequenzen [22]. Dort werden die Bildinhalte gewarpt, um die Bewegung der Kamera auszugleichen, um einen insgesamt stabileren Bildlauf zu erhalten. Dies geschieht, ähnlich wie in dieser Arbeit, mithilfe des Wissens über die involvierte Bewegung.

Bild-Metamorphose Kernidee bei der Zwischenbildinterpolation mit optischem Fluss ist das Warping von Bildinformation zum gewünschten Interpolanten, der im Fall der Zwischenbildinterpolation zwischen zwei Nachbarbildern liegt. Bei der *Image-Metamorphosis* geschieht etwas sehr Ähnliches: Es wird versucht ein Bild zu bestimmen, das ebenfalls „zwischen“ den beiden Nachbarn liegt und den Übergang zwischen beiden Nachbarbildern darstellt. Auch dort kann die Bildinformation über ein Image Warping zum Bild in der Mitte transportiert werden [3].

Super-Resolution Die bereits angedeutete Zwischenbildinterpolation mit den vorgestellten Warpingverfahren besteht aus zwei wesentlichen Schritten: Zuerst das Warping von Bildinformation der Nachbarbilder über einen optischen Fluss, im Anschluss eine mögliche Vervollständigung in Form von Inpainting. Bei der *Super-Resolution* läuft es unter Umständen ähnlich ab: Es findet zunächst eine Zusammenführung mehrerer Bilder statt. Dies kann geschehen über einen optischen Fluss und mehrfachem Warping. Anschließend können noch Verbesserungen vorgenommen werden, wie zum Beispiel Deblurring [1].

Texture Mapping Das vorgestellte Backward Warping kommt dem Texture Mapping aus dem Bereich der Bildsynthese sehr nahe und hängt in gewisser Weise zusammen. Beim Texture Mapping wird versucht eine Textur (Bild) auf eine Oberfläche in einer dreidimensionalen Szene zu mappen [14]. Dies geschieht durch eine Form von Mapping, bei dem eine Textur als Quelle I_{Quelle} für ein Backward Warping verwendet wird.

Kapitel 4

Bildinterpolationsverfahren

Wie in Kapitel 1 kategorisiert, handelt es sich bei der Zwischenbildinterpolation mit optischem Fluss um eine Motion Compensated Frame Interpolation. Daraus resultierend bildet die Motion Estimation, hier die Erzeugung des optischen Flusses, den ersten Schritt aller in diesem Kapitel vorgestellten Verfahren. In einem nächsten Schritt wird diese Motion Estimation immer genutzt, um die Bildinformationen benachbarter Bilder über ein Warping entlang der Motion Estimation zum Zwischenbild zu transportieren. Dazu werden die grundlegenden Techniken Forward Warping und Backward Warping aus dem letzten Kapitel genutzt und zum Teil mehrfach hintereinander ausgeführt. Dieser zweite Schritt lässt sich daher als ein allgemeiner Warpingvorgang zusammenfassen und stellt gleichzeitig den Fokus dieses Kapitels dar. Wie im letzten Kapitel festgestellt, gibt es im Zusammenhang von Warpingvorgängen häufig Lücken im Bild, weswegen eine abschließende Behandlung dieser Lücken meist den finalen Berechnungsschritt der Zwischenbildinterpolation mit optischem Fluss darstellt (siehe Abbildung 4.1).

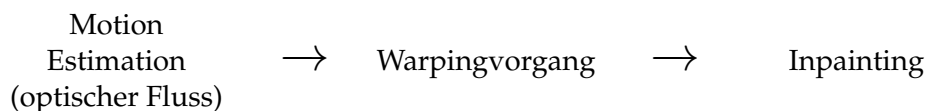


ABBILDUNG 4.1: Die drei wesentlichen Schritte bei der Zwischenbildinterpolation mit optischem Fluss.

Kategorisierung In dieser Arbeit werden die vorgestellten Bildinterpolationsverfahren kategorisiert und nach der Anzahl von verwendeter Bildinformation und der Richtung des verwendeten optischen Flusses geordnet, wodurch schrittweise verbesserte Interpolationsverfahren eingeführt werden. Es wird begonnen mit den Grundverfahren Forward Warping und Backward Warping, die den einfachen 1-Bild Fall darstellen, da nur mit einer Bildinformation und einem optischem Fluss gearbeitet wird. Sie werden direkt abgeleitet aus den Schemata des letzten Kapitels und bilden die Grundlage für die darauffolgenden Verfahren. In Abschnitt 4.4 wird der aufbauende 2-Bild Fall eingeführt, der aus Verfahren besteht, die mit zwei benachbarten Bildern arbeiten, um einen Interpolanten zu erzeugen. Bei diesen Verfahren handelt es sich um sogenannte symmetrische Warping Verfahren, die eine Erweiterung des 1-Bild Falls darstellen. Nach dem 2-Bild Fall folgt in Abschnitt 4.5 noch eine Reihe von bidirektionalen Verfahren. Diese stellen eine Erweiterung des 2-Bild Falls dar, indem sie neben zwei Bildinformationen noch eine zusätzliche Motion Estimation verwenden (zwei optische Flüsse). Die bidirektionalen Verfahren sind einerseits bidirektionale Varianten der bisherigen (direktionalen) Verfahren des 2-Bild Falls, es kommen aber auch neue Verfahren dazu, die Alternativen zu den direktionalen Verfahren darstellen.

Notation Im Folgenden ist ein Bild (Frame) als $I_t(\vec{x})$ notiert, wobei der Index t bei einem Bild I_t die relative zeitliche Position in der Bildabfolge darstellt und der Vektor \vec{x} alle möglichen Positionen innerhalb des Bildes beinhaltet. Da sich bei der Zwischenbildinterpolation einer kompletten Bildsequenz sich die Indizes immer mit verschieben und eine allgemeine Schreibweise mit beispielsweise I_t, I_{t+1}, I_{t+2} usw. etwas schwierig zu lesen ist, wird sich im Folgenden immer nur auf zwei Frames einer Folge beschränkt: I_0 und I_1 . Für optische Flüsse v_t wird immer erwähnt, auf welche Bilder sie sich beziehen. Da mit den Bildern I_0 und I_1 gearbeitet wird, gibt der Index t bei einem optischen Fluss an, auf welchem Bild der Startpunkt des Flusses liegt. Das bedeutet also, dass für die Bilder I_0 und I_1 beispielsweise der optische Fluss \vec{v}_0 von Bild I_0 nach I_1 fließt.

Problemformulierung Es wird nun folgendes Ziel verfolgt: Mithilfe von zwei benachbarten Frames I_0 und I_1 und einem optischen Fluss \vec{v}_0 soll ein Zwischenbild I_t mit $t \in [0, 1]$ erzeugt werden. Später bei den bidirektionalen Verfahren findet eine leichte Umformlierung dieses Ziels statt.

4.1 Naiver Ansatz

Bevor mit den eigentlichen Warpingverfahren begonnen wird, soll zur Vollständigkeit das naive Zwischenbildinterpolationsverfahren erwähnt werden. Dieses besteht einfach aus dem einfachen Mitteln der beiden Frames I_0 und I_1 . Der räumliche Kontext wird nicht beachtet und es findet eine einfache lineare Interpolation von Pixel zu Pixel statt (siehe Abbildung 4.2 links). Formal kann ein gemittelttes Zwischenbild I_t erzeugt werden mit:

$$I_t(\vec{x}) = (1 - t) \cdot I_0(\vec{x}) + t \cdot I_1(\vec{x}) \quad (4.1)$$

Dieses sehr einfache und intuitive Verfahren weist jedoch offensichtliche Schwächen auf und sei nur aufgeführt, weil es später als ein guter Vergleich für die Vorteile der vorgestellten Verfahren dient. Das Problem an diesem Ansatz liegt an dem vollständigen Ignorieren der Bildinhalte: Bewegen sich Objekte im Bild, erscheinen diese im Interpolanten doppelt. Diese Doppelungsartefakte sollen in den folgenden Verfahren möglichst reduziert werden, indem die Bildinhalte über den optischen Fluss mehr berücksichtigt werden.

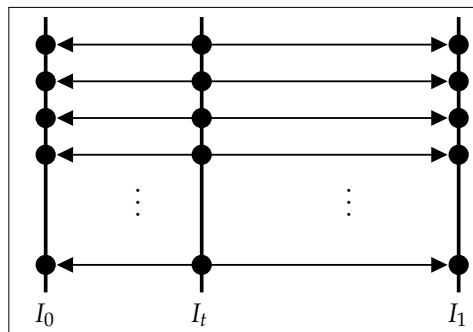


ABBILDUNG 4.2: Es ist das geradlinige lineare Interpolieren beim naiven Ansatz dargestellt. Es findet unabhängig von den Bildinhalten immer eine Zusammenführung der gleichen Pixel statt, wodurch im Falle einer Bewegung es zu Doppelungen im Interpolant kommt (Objekte an Position vor und nach der Bewegung).

4.2 Zentrale Annahme

Es soll nun mithilfe eines gegebenen optischen Flusses \vec{v} interpoliert werden. Das heißt, es liegt ein pixelweises Verschiebungsfeld vor, das für jeden Pixel die Verschiebung im Bild beschreibt. Im letzten Kapitel wurde bei der Vorstellung des optischen Flusses als Mappingfunktion bereits angedeutet, dass über die beschriebenen Verschiebungsvektoren ein Zwischenzustand erzeugt werden kann, wofür im Voraus aber noch eine Annahme getroffen werden muss. Da ein Verschiebungsvektor eines optischen Flusses eine geradlinige Verbindung zwischen den Bildern darstellt, muss als zentrale Annahme auch eine tatsächlich geradlinige Bewegungen im Bild angenommen werden. Dies ist notwendig, um linear entlang Bewegungstrajektorie zu interpolieren, um einen gedanklichen Zwischenzustand zu erhalten.

Die Einschränkung der möglichen Bewegungen im Bild ist damit zu legitimieren, dass der zeitliche Abstand zwischen den Nachbarbildern sehr gering ist. Wie bereits in Kapitel 1 erklärt, sind Videoaufnahmen meistens mit 24 Bildern pro Sekunde aufgenommen, wodurch der Zeitraum eines optischen Flusses sich in einem Bereich von einem 24-tel einer Sekunde bewegt. In so einem Zeitraum erscheinen die meisten Bewegungen für den Menschen minimal und augenscheinlich geradlinig. Die getroffene Annahme kann natürlich in gewissen Situationen verletzt werden. Es kann unter Umständen sein, dass sich ein Objekt im Bild mit einer sehr schnellen Geschwindigkeit bewegt und dadurch nicht auf der geradlinigen Verschiebung des optischen Flusses liegt (siehe Abbildung 4.3). Dieses Problem kann es auch bei der Restauration älterer Aufnahmen geben, die mit niedrigeren Bildwiederholungsraten aufgenommen wurden, da dort mehr Bewegung zwischen zwei aufeinanderfolgenden Bildern stattfindet. In diesem Fall würde der zeitliche Abstand schon bei etwa einem Zehntel einer Sekunde liegen (vgl. Kapitel 1). Bei räumlichen Daten ist man abhängig vom Abstand der Samples, zwischen denen interpoliert werden soll. Je näher die Samples räumlich beieinander liegen, desto näher ist man bei einem geradlinigen Übergang. Trotz all dieser Grenzfälle kann die zentrale Annahme für die meisten Bildsequenzen getroffen werden und die produzierten Ergebnisse wirken für den Betrachter trotzdem meist plausibel.

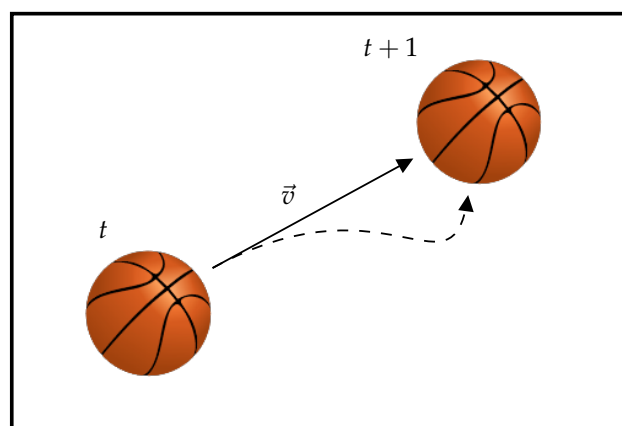


ABBILDUNG 4.3: Es wird immer eine geradlinige Bewegung im Bild angenommen, die mit dem Verschiebungsvektor des optischen Flusses \vec{v} übereinstimmt. Je größer aber der temporäre Abstand der Bilder von Zeitpunkt t nach $t + 1$, desto mehr kann die lineare Interpolation über den optischen Fluss \vec{v} von der tatsächlichen Bewegung der Objekte (gestrichelt) abweichen.

4.3 1-Bild Fall

Das Forward Warping und das Backward Warping, die in Kapitel 3 als Grundtechnik eingeführt wurden, sind nun die Basis für zwei Interpolationsverfahren, die im Folgenden vorgestellt werden sollen. Mit diesen Grundtechniken ist es beinahe direkt möglich Zwischenbilder zu erzeugen, allerdings bedarf es noch zusätzlicher Schritte, um ein korrektes Zwischenbild zu erhalten.

Gemäß der Kategorisierung als 1-Bild Fall wird bei den zwei vorgestellten Verfahren immer nur mit einer Bildinformation (entweder Bild I_0 oder I_1) und dem optischem Fluss \vec{v}_0 gearbeitet. Das bedeutet, dass der Interpolant von nur einem Nachbar die Farbwerte erhalten wird, was sich ähnlich wie bei optischen Flüssen über eine Ähnlichkeitsannahme legitimieren lässt: Neben der sowieso möglichst kleinen Bewegung zwischen den Frames (durch den zeitlichen/räumlichen Abstand) wird angenommen, dass sich aus photometrischer Sicht im Übergang zwischen den Bildern I_0 und I_1 nicht viel ändert. Aus diesem Grund ist es ausreichend, die Bildinhalte vor oder nach dem Zwischenbild nur noch an die Position zum Zeitpunkt des Zwischenbilds zu verschieben. Dies geschieht über die Grundtechniken des letzten Kapitels, welche ab sofort zur besseren Unterscheidung immer als die „elementaren“ Warpingverfahren bezeichnet werden.

4.3.1 Forward Warping

Das Forward Warping soll das erste betrachtete Verfahren sein, das einen erzeugten optischen Fluss \vec{v}_0 (von Frame I_0 zu Frame I_1) für die Erzeugung eines Zwischenbilds I_t verwendet. Der Gedanke hinter dem Forward Warping der Zwischenbildinterpolation ist das Verschieben von Bildinformation entlang des optischen Flusses \vec{v}_0 zum Zwischenbild I_t . Was also geschieht, ist eine Umpositionierung der Objekte aus dem Zustand vor dem Zwischenbild zu dem Zwischenzustand zum Zeitpunkt t .

Ein Blick auf Abbildung 4.4 macht den Grund klar, warum dies noch nicht direkt mit dem erzeugten Fluss \vec{v}_0 zu erreichen ist. Dort ist links zu erkennen, dass es durch den Fluss \vec{v}_0 eine zu weitreichende Verschiebung der Bildinformation I_0 gibt, denn der erzeugte Fluss \vec{v}_0 verschiebt die Objekte immer komplett vom Bild I_0 zum Zustand des darauffolgenden Bilds I_1 (vgl. elementares Forward Warping aus Kapitel 3.4). Um stattdessen eine Verschiebung zum Zustand des Zwischenbilds I_t zu erreichen, ist es notwendig, die durchgeführte Verschiebung zu kürzen. Dies wird umgesetzt, indem der erzeugte optische Fluss \vec{v}_0 mit dem Faktor t multipliziert wird, wodurch er eine Verschiebung auf den Zeitpunkt t beschreibt (siehe Abbildung 4.4 rechts). Dieser angepasste Fluss kann dann für elementares Forward Warping verwendet werden, um ein Zwischenbild I_t zum Zeitpunkt t zu erhalten:

$$I_t(\vec{x} + t \cdot \vec{v}_0(\vec{x})) = I_0(\vec{x}) \quad (4.2)$$

Durch die Verschiebung von einzelnen Objekten im Bild kann es zu zwei bedeutenden Situationen kommen. Es besteht beispielsweise die Möglichkeit einer Kollision von zwei Objekten, bei der ein Objekt über den optischen Fluss zu einer Stelle verschoben wird, an der sich bereits ein anderes Objekt befindet (Verdeckung). In diesem Fall bedarf es eine Kollisionsbehandlung, wofür im Kapitel 3.4.2 bereits einige Möglichkeiten vorgestellt wurden. Die andere (umgekehrte) Situation ergibt sich an der alten Stelle des verschobenen Objekts. Hier wird, sofern kein anderes Objekt

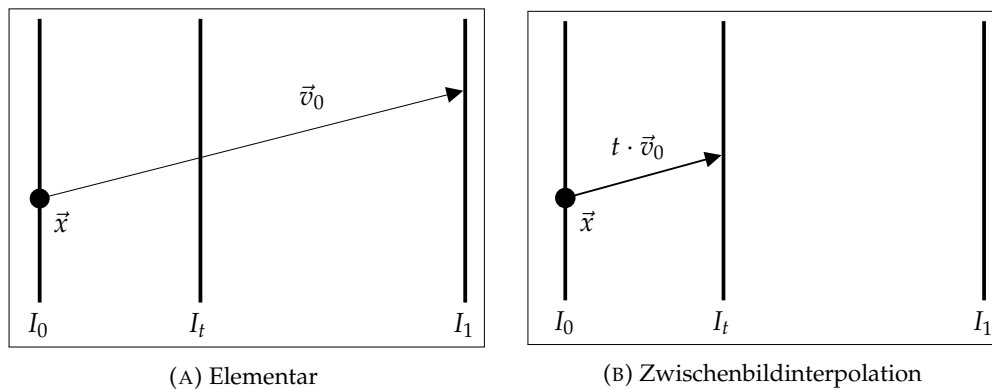


ABBILDUNG 4.4: Eine Visualisierung des Forward Warpings. Es ist rechts die Verschiebung von Bildinformation aus I_0 über \vec{v}_0 nach „vorne“ zum Interpolanten I_t zu erkennen. Auf der linken Seite ist die zu weitreichende Verschiebung durch das elementare Forward Warming zu sehen.

dorthin verschoben wird, der Hintergrund der Szene offenbart. Da für diesen keine Bildinformation bekannt ist, liegt eine Lücke im Interpolanten I_t vor, die über ein Inpainting wie in Kapitel 3.5 gefüllt werden muss. Die Lückenanzahl im Bild kann aber, abhängig von der gewählten Verteilungsstrategie beim Splatting, noch beeinflusst werden (vgl. Verteilungsstrategien in Kapitel 3.4.1). Das Erzeugen des Flusses \vec{v}_0 , das Durchführen eines elementaren Forward Warpings und das notwendige Inpainting macht das Forward Warming zu einem Interpolationsverfahren bestehend aus drei Schritten. Damit passt es genau zu der allgemeinen Beschreibung der Zwischenbildinterpolation mit optischem Fluss zu Beginn dieses Kapitels.

4.3.2 Backward Warming

Das Gegenstück zum Forward Warming ist das Backward Warming, das das zweite Interpolationsverfahren des 1-Bild Falls darstellt. Während beim Forward Warming die Bildinformation aus dem vorherigen Frame I_0 nach „vorne“ zum Zwischenbild I_t gegeben wird, wird jetzt beim Backward Warming die Bildinformation aus dem darauffolgenden Frame I_1 zum zeitlich früheren Zwischenbild I_t zurücktransportiert. Das geschieht, ähnlich wie beim vorangegangenen Forward Warming, über das Durchführen eines elementaren Backward Warpings, wofür erneut eine Modifikation am erzeugten optischen Fluss \vec{v}_0 vorgenommen wird.

Während beim vorangegangenen Forward Warming die Verschiebung durch den Fluss \vec{v}_0 zu weitreichend für die Zwischenbildinterpolation war, ist beim Backward Warming der Startpunkt des optischen Flusses \vec{v}_0 nicht korrekt. Der optische Fluss \vec{v}_0 besitzt seinen Startpunkt immer bei Frame I_0 , wodurch bei Anwendung eines elementaren Backward Warpings die Bildinformationen von I_1 immer zu den Positionen zum Zeitpunkt $t = 0$ verschoben werden (vgl. elementares Backward Warming in Kapitel 3.3). In Abbildung 4.5 ist links die Startposition des Flusses nochmal durch einen schwarzen Punkt hervorgehoben. Was für die Zwischenbildinterpolation geschehen soll, ist eine Verschiebung der Bildinformation zu den Positionen zum Zeitpunkt t des Interpolanten I_t , bei denen es sich genau um die Positionen handelt, die sich beim Forward Warming aus dem vorherigem Abschnitt ergeben. Deshalb wird noch eine Modifikation am erzeugten optischen Fluss \vec{v}_0 vorgenommen, damit ein elementares Backward Warming zur Zwischenbildinterpolation verwendet

werden kann. Dazu werden die Startpositionen des Flusses \vec{v}_0 vom Bild I_0 zum Interpolanten I_t verschoben, indem der Fluss um einen Teil von sich selbst nach vorne gewarpt wird:

$$\vec{v}_t(\vec{x} + t \cdot v_0(\vec{x})) = v_0(\vec{x}) \quad (4.3)$$

Bei diesem Warping handelt es sich genau um das gleiche Forward Warping wie aus dem vorherigem Abschnitt 4.3.1, nur das dieses Mal kein Bild sondern ein Fluss gewarpt wird. Die involvierten Konzepte des Forward Warpings können aber übertragen werden, da der Fluss \vec{v}_0 als ein Bild I_0 mit zwei Kanälen angesehen werden kann. Es ist zu beachten, dass der berechnete Fluss \vec{v}_t die Vektorenlängen vom Fluss \vec{v}_0 beibehält, weswegen der Fluss \vec{v}_t eine noch zu weitreichende Verschiebung besitzt. Aus diesem Grund ist es notwendig ihn noch zu kürzen, damit er wirklich die Verschiebung vom Zwischenbild I_t zum Frame I_1 beschreibt. Dies ist rechts in Abbildung 4.5 an der Kürzung durch den Faktor $(1 - t)$ zu erkennen. Mit dem gewarpten und gekürzten Fluss \vec{v}_t kann das elementare Backward Warping zur Zwischenbildinterpolation verwendet werden:

$$I_t(\vec{x}) = I_1(\vec{x} + (1 - t) \cdot \vec{v}_t(\vec{x})) \quad (4.4)$$

Es ist also festzustellen, dass das Backward Warping für die Zwischenbildinterpolation nur über ein enthaltenes Forward Warping möglich ist. Ähnlich wie dort kann es aber auch beim Backward Warping nach dem Schritt des elementaren Backward Warpings zu Lücken im Bild kommen. Dies kann passieren, wenn der berechnete Fluss \vec{v}_t eine Verschiebung der Objekte zu einer Position außerhalb des Bildbereichs beschreibt und die Samplingstrategien aus dem letzten Kapitel 3.3 fehlschlagen. In diesem Fall konnte kein Farbwert für den Interpolanten I_t bestimmt werden, wodurch eine Lücke im Bild vorliegt. Aus diesem Grund muss im Anschluss nach dem elementaren Backward Warping noch ein Inpainting, wie in Kapitel 3.5 beschrieben, durchgeführt werden.

Durch das enthaltene Fluss Warping wird außerdem eine neue mögliche Stellschraube eingeführt, denn es ist nicht klar, wie gut die vorgestellten Splattingstrategien mit optischen Flüssen funktionieren. Spezielle Splattingstrategien für optische Flüsse wurden nicht eingeführt, aber deren Existenz sollte erwähnt sein, denn es gibt Interpolationsverfahren, die solche Strategien verwenden. Râket et al. [31] oder Baker et al. [2] führen beispielsweise beim Splatting eine pixelweise Sortierung durch, die bei einer Kollision nur den Verschiebungsvektor behält, der die geringste photometrische Differenz an den Enden der beschriebenen Verbindung durch \vec{v}_0 besitzt. Diese Vorgehen ähnelt im Kern aber durchaus der vorgestellten Austauschstrategie aus dem letzten Kapitel 3.4.2, bei der ebenfalls ein Wert bei einer Kollision komplett ausgetauscht wird.

4.4 2-Bild Fall

Während die bisherigen Grundverfahren immer nur mit einer einzelnen Bildinformation arbeiten, findet in diesem Abschnitt die erste Erweiterung um eine zweite Bildinformation statt. Dadurch wird ab sofort neben einer Motion Estimation über einen optischen Fluss \vec{v}_0 mit zwei Bildern I_0 und I_1 gearbeitet, weswegen hier immer vom 2-Bild Fall die Rede ist. Das bisherige Forward Warping und Backward Warping verschieben bisher lediglich die Bildinformation aus entweder I_0 oder I_1 an die zeitlich korrekte Stelle in I_t , wodurch in erster Linie nur der Aspekt der linearen

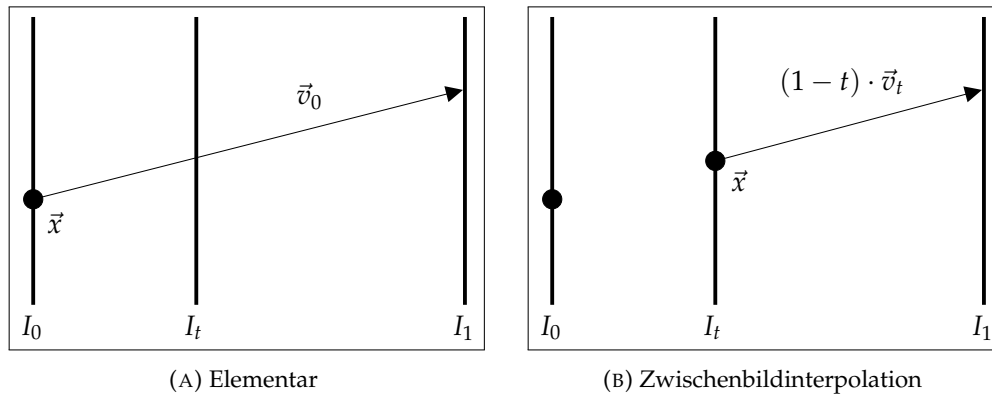


ABBILDUNG 4.5: Auswirkung des Warpings des Flusses mit sich selbst. Links ist der Fall für das elementare Backward Warping zu sehen und rechts die Anpassung davon für die Zwischenbildinterpolation.

Interpolation entlang räumlichen Verschiebung umgesetzt wurde. Was bisher aber ignoriert wird, sind die photometrischen Veränderungen in den Bildern. Farb- und Helligkeitsveränderungen sind über dem Verlauf der zwei Bilder I_0 und I_1 möglich, wodurch es eigentlich notwendig ist, dass ein Interpolant I_t auch von der Farbe und Helligkeit zwischen den Bildern liegen muss. Um dies zu erreichen, ist es notwendig die Bildinformation von beiden benachbarten Bildern zum Interpolanten zu bringen und in einer Weise zu kombinieren.

Mithilfe des kennengelernten Forward Warpings und Backward Warpings ergibt sich gedanklich eine Reihe von vier möglichen Verfahren, die Bildinformationen von I_0 und I_1 zum Interpolanten I_t verschieben können. So könnte beide Male Bildinformation von I_0 als auch von I_1 über ein Forward Warping oder über ein Backward Warping zum Interpolanten t gelangen. Die anderen zwei Möglichkeiten wären eine Art gemischtes Warping, bei dem ein Forward Warping und ein Backward Warping verwendet wird, um die Bildinformation zu transportieren. In dieser Arbeit wird sich auf den einheitlichen Fall beschränkt.

4.4.1 Symmetrisches Backward Warping

Das sogenannte symmetrische Backward Warping ist als direkte Erweiterung des Backward Warpings aus Abschnitt 4.3.2 zu sehen. Statt nur die Information aus dem gegenüberliegenden Bild I_1 zu verwenden, wird zusätzlich der berechnete Fluss \vec{v}_t gespiegelt und es wird aus beiden Bildern I_0 und I_1 gesampelt. Die Verwendung eines gespiegelten Flusses und dem dadurch verbundenen *symmetrischen* Interpolationsschemas sei der Hintergrund für die Namensgebung.

Beim symmetrischen Backward Warping ist zu Beginn wieder ein gewarpter Fluss \vec{v}_t notwendig (vgl. Backward Warping aus Abschnitt 4.3.2):

$$\vec{v}_t(\vec{x} + t \cdot \vec{v}_0(\vec{x})) = \vec{v}_0(\vec{x}) \quad (4.5)$$

Für den berechneten Fluss \vec{v}_t ergibt sich eine gespiegelte Version durch eine einfache Multiplikation der Verschiebungsvektoren mit -1 . Über den Fluss \vec{v}_t und seine Spiegelung $-\vec{v}_t$ lässt sich ein Zwischenbild I_t erzeugen über:

$$I_t(\vec{x}) = (1-t) \cdot I_0(\vec{x} - t \cdot \vec{v}_t(\vec{x})) + t \cdot I_1(\vec{x} + (1-t) \cdot \vec{v}_t(\vec{x})) \quad (4.6)$$

Algorithmus 3 : Symmetrisches Backward Warping.

Data : Zwei Bilder I_0, I_1 und optischer Fluss \vec{v}_0
Result : Interpolant I_t

```
// Bestimme den Fluss  $\vec{v}_t$ 
Forward Warping von  $\vec{v}_0$  über den Fluss  $t \cdot \vec{v}_0$ 

// Bildinformation aus  $I_0$ 
Elementares Backward Warping von  $I_0$  über den Fluss  $(-t) \cdot \vec{v}_t$ 

// Bildinformation aus  $I_1$ 
Elementares Backward Warping von  $I_1$  über den Fluss  $(1 - t) \cdot \vec{v}_t$ 

// Berechne finalen Interpolanten  $I_t$ 
Gewichtete Zusammenführung beider Bildinformationen

// Füllen von Lücken im Bild
Inpainting aus Kapitel 3.5
```

Durch die Verwendung von zwei Bildern I_0 und I_1 ist es ähnlich wie beim naiven Ansatz notwendig, die Bildinformationen unterschiedlich stark zu gewichten. Die Gewichtung hängt wie beim naiven Ansatz vom Abstand zum Zwischenbild ab (vgl. Gleichung 4.1), wodurch es sich beim symmetrischen Backward Warping insgesamt um eine Art Konvexkombination aus zwei interpolierten Bildern handelt. Durch das Vorhandensein von zwei Backward Warpings (mit den Bildern I_0 und I_1), muss eine festgelegte Behandlung für den Fall stattfinden, bei dem es in einem oder beiden Bildern nicht möglich war einen Farbwert samplen. In dieser Arbeit wird in so einem Fall ein Pixel im Interpolanten nur gesetzt, sofern das Sampling in beiden Frames erfolgreich war und zwei Farbwerte gewichtet zusammengeführt werden konnten. Verbesserte Varianten des symmetrischen Backward Warpings wählen für dieses Szenario teilweise andere Strategien. Baker et al. [2] führen beispielsweise ein selektives Zusammenführen ein, das auf mehreren Verdeckungsmasken basiert. Da in dieser Arbeit in erster Linie nur die Grundversionen der Interpolationsverfahren geprüft werden, wird auf Verdeckungsmasken oder andere Formen von sogenanntem *Depth Reasoning* verzichtet. Da es also beim Warpingvorgang zu nichtbesetzten Stellen im Interpolanten kommen kann, ist zum Schluss wieder ein Inpainting wie in Kapitel 3.5 notwendig. Weil dieses und die kommenden Verfahren zunehmend unüberschaubarer werden, wird ab sofort ein kleiner Pseudocode zum besseren Verständnis angegeben. Algorithmus 3 zeigt den Ablauf des symmetrischen Backward Warpings und macht die einzelnen Zwischenschritte nochmal deutlich. Zuerst findet ein Fluss Warping statt, danach folgen zwei elementare Backward Warpings, eine gewichtete Zusammenführung und zuletzt wird ein Inpainting durchgeführt.

Bei der Nutzung eines gespiegelten Flusses gibt es einen wichtigen Aspekt zu beachten. Während der Bestimmung des Flusses \vec{v}_t ist wie im Falle des normalen Backward Warpings ein Forward Warping involviert. Dabei kommt es durch unterschiedliche Splattingstrategien und einem Inpainting zu unterschiedlichen Flüssen \vec{v}_t . Was bei der Spiegelung eines solchen Flusses zu beachten ist, ist die vom Interpolanten I_t ausgehend getroffene Stelle in I_0 . Diese Stelle muss nicht mehr zwingenderweise mit der Startposition des ursprünglichen Flusses \vec{v}_0 übereinstimmen, was zur Folge hat, dass bei dem symmetrischen Backward Warping nicht mehr exakt entlang der ursprünglichen Verbindung \vec{v}_0 der Bilder I_0 und I_1 interpoliert wird.

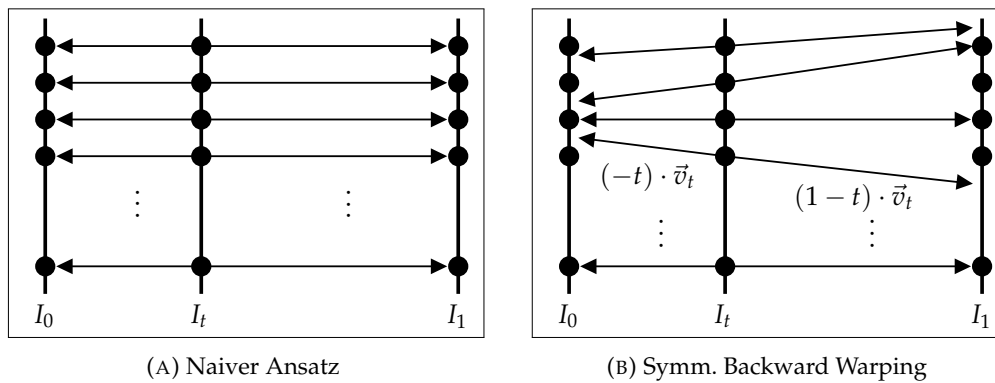


ABBILDUNG 4.6: Veranschaulichung vom symmetrischen Backward Warping gegenüber dem naiven Interpolationsansatz. Es wird beim symmetrischen Backward Warping nun ausgehend vom Zwischenbild, geradlinig entlang der Bewegung im Bild aus den anliegenden Bilder I_0 und I_1 gesampelt.

Das symmetrisches Backward Warping ist ein guter Repräsentant für die Vorteile der Zwischenbildinterpolation mit optischem Fluss gegenüber dem naiven Ansatz. Durch das symmetrische Backward Warping wird deutlich, dass der Bildinhalt nicht mehr ignoriert und jetzt entlang der Bewegungstrajektorie im Bild linear interpoliert wird (siehe Abbildung 4.6).

4.4.2 Symmetrisches Forward Warping

Wie beim 1-Bild Fall lässt sich ein weiteres Verfahren definieren, das das Gegenstück zum symmetrischen Backward Warping darstellt. Während beim symmetrischen Backward Warping die Bildinformationen I_0 und I_1 über ein Backward Warping zum Interpolanten gelangen, wird dieser Vorgang nun ersetzt durch ein Forward Warping.

Das symmetrische Forward Warping unterscheidet sich merklich vom symmetrischen Backward Warping durch die anderen Problematiken, wie Splatting von Bildinformation und der mit dem Forward Warping verbundenen Verteilung von Lücken. Für das symmetrische Forward Warping ist erneut ein neuer Fluss \vec{v}_1 notwendig, der auf eine ähnliche Weise wie der gewarppte Fluss \vec{v}_t beim (symmetrischen) Backward Warping bestimmt wird (vgl. Gleichung 4.5). Beim symmetrischen Forward Warping wird der Fluss \vec{v}_0 , der die Bewegung von Bild I_0 nach I_1 beschreibt, in einem elementaren Forward Warping um seine komplette Länge nach vorne gewarpt:

$$\vec{v}_1(\vec{x} + 1 \cdot v_0(\vec{x})) = v_0(\vec{x}) \quad (4.7)$$

Hier ist der Unterschied zum Fluss \vec{v}_t des (symmetrischen) Backward Warpings, dass das Warping des Flusses nicht bis zu einem gewissen Zeitpunkt t vorgenommen wird, sondern immer komplett bis zum anderen Nachbarbild I_1 (vgl. elementares Forward Warping in Kapitel 3.4). Wird der berechnete Fluss \vec{v}_1 noch umgedreht, so entspricht er gedanklich der Bewegungstrajektorie von Bild I_1 nach I_0 (siehe Abbildung 4.7 links). Mithilfe des erzeugten Flusses \vec{v}_0 und dem berechneten Fluss \vec{v}_1 kann im Anschluss ein Interpolant I_t über eine Zusammenführung von zwei Forward Warpings erzeugt werden.

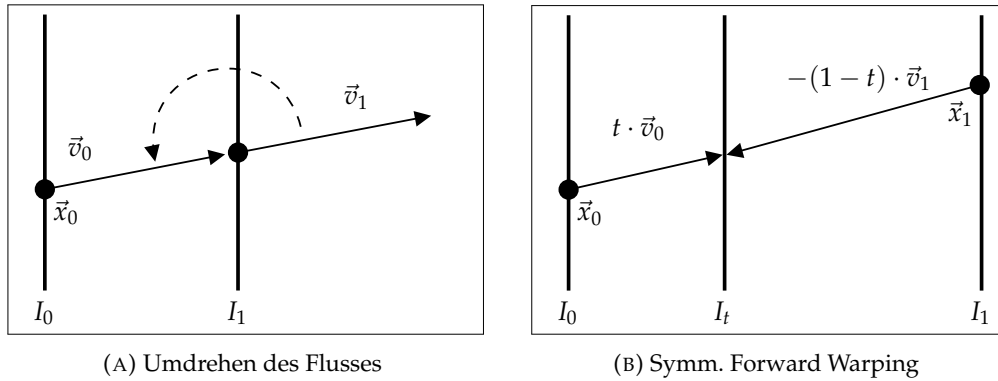


ABBILDUNG 4.7: Veranschaulichung des symmetrischen Forward Warpings. Links ist das Umdrehen des gewarppten Fluss \vec{v}_1 zu sehen. Rechts sieht man das Schema für die Interpolation.

Für die Zusammenführung sind zwei temporäre Interpolanten $I_{t,0}$ und $I_{t,1}$ nötig, die über Forward Warpings mit den Flüssen \vec{v}_0 und \vec{v}_1 bestimmt werden:

$$\begin{aligned} I_{t,0}(\vec{x}_0 + t \cdot \vec{v}_0(\vec{x}_0)) &= I_0(\vec{x}_0) \\ I_{t,1}(\vec{x}_1 + (1-t) \cdot \vec{v}_1(\vec{x}_1)) &= I_1(\vec{x}_1) \end{aligned} \quad (4.8)$$

Dabei entspricht \vec{x}_0 einer Stelle in I_0 und \vec{x}_1 einer Stelle in I_1 . Das entstandene temporäre Zwischenbild $I_{t,0}$ ergibt sich über das bisherige Forward Warping Schema aus Abschnitt 4.3.1, was bei $I_{t,1}$ auch der Fall ist, aber dort wird mit dem Frame I_1 und dem umgedrehten Fluss \vec{v}_1 gearbeitet (siehe Abbildung 4.7 rechts). Die sich beim Interpolant treffenden Bildinformationen müssen noch zusammengeführt werden. Dies geschieht wie bisher über eine Gewichtung abhängig vom Abstand zum Interpolationszeitpunkt t . Damit ergibt sich beim symmetrischen Forward Warping ein finaler Interpolant I_t über:

$$I_t(\vec{x}) = (1-t) \cdot I_{t,0}(\vec{x}) + t \cdot I_{t,1}(\vec{x}) \quad (4.9)$$

Beim symmetrischen Backward Warping aus dem letzten Abschnitt 4.4.1 wurde insbesondere erwähnt, dass eine Zusammenführung der Bildinformation in der Mitte nur stattfinden kann, wenn das Sampling aus beiden Bildern erfolgreich war. Auch beim symmetrischen Forward Warping gibt es bei der Zusammenführung wichtige Details, die für die Interpolation klar definiert sein müssen. Durch die Verwendung und der Natur des Forward Warpings ist klar, dass es bei der Erzeugung der temporären Zwischenbilder $I_{t,0}$ und $I_{t,1}$ zu unterschiedlichen Lücken kommt. Aus diesem Grund muss entschieden werden, ob vor oder nach der Zusammenführung beider Bildinformationen ein Inpainting am vorteilhaftesten ist. Würde man beispielsweise ein Inpainting vor der Zusammenführung durchführen, kann es sein, dass ein durch Inpainting gefüllter Pixel des einen temporären Zwischenbilds mit einem besetzten Pixel des anderen temporären Zwischenbilds kombiniert wird. Da anzunehmen ist, dass die Qualität von gefüllten Lücken im Allgemeinen schlechter ist, würde dies eine Verschlechterung eines eigentlich schon guten Pixels bedeuten. In dieser Arbeit wird eine Zusammenführung der temporären Zwischenbilder (welche noch Lücken besitzen) durchgeführt und erst danach folgt ein abschließendes Inpainting. Dabei werden insbesondere nur Stellen zusammengeführt, die in beiden Interpolanten besetzt sind und keine Lücke darstellen. Algorithmus 4 macht die Funktionsweise des symmetrischen Forward Warpings nochmal deutlich. Es findet zuerst ein Forward

Algorithmus 4 : Symmetrisches Forward Warping.

Data : Zwei Bilder I_0, I_1 und optischer Fluss \vec{v}_0
Result : Interpolant I_t

```
// Bestimme den Fluss  $\vec{v}_1$ 
Forward Warping von  $\vec{v}_0$  über den Fluss  $\vec{v}_0$ 

// Bildinformation aus  $I_0$ 
Elementares Forward Warping von  $I_0$  über den Fluss  $t \cdot \vec{v}_0$ 

// Bildinformation aus  $I_1$ 
Elementares Forward Warping von  $I_1$  über den Fluss  $-(1-t) \cdot \vec{v}_1$ 

// Berechne finalen Interpolanten  $I_t$ 
Gewichtete Zusammenführung beider Bildinformationen

// Füllen von Lücken im Bild
Inpainting aus Kapitel 3.5
```

Warping des Flusses \vec{v}_0 statt, um den Fluss \vec{v}_1 zu bestimmen. Danach werden die zwei temporären Interpolanten $I_{t,0}$ und $I_{t,1}$ jeweils über ein elementares Forward Warping bestimmt. Es folgt eine gewichtete Zusammenführung und ein abschließendes Inpainting wie in Kapitel 3.5.

Beim symmetrischen Forward Warping handelt es sich um ein Verfahren, bei dem es eine starke Abhängigkeit vom Forward Warping und dem involvierten Splatting gibt. Insgesamt finden drei elementare Forward Warpings statt, wovon eines ein Fluss Warping ist. Für dieses Interpolationsverfahren sei abschließend noch angemerkt, dass der berechnete und umgedrehte Fluss \vec{v}_1 von der Funktionsweise genau dem umgekehrten Fluss (Rückwärtsfluss) von Bild I_1 nach I_0 entspricht. Diese Ähnlichkeit wird später bei den bidirektionalen Verfahren nochmal aufgegriffen.

4.5 Bidirektionale Verfahren

Die bisherigen Verfahren arbeiten alle mit Bildinformationen von bis zu zwei Bildern: I_0 und I_1 . Dazu kommt der passende optische Fluss \vec{v}_0 , der die Bewegung von I_0 nach I_1 beschreibt. Bisher beachtet wurde aber nicht, warum gerade dieser Fluss \vec{v}_0 überhaupt verwendet wird. Bei zwei gegebenen Bildern gibt es zwei mögliche Richtungen für einen optischen Fluss: ein sogenannter Vorwärtsfluss $\vec{v}_{0,f}$ und ein Rückwärtsfluss $\vec{v}_{1,b}$. Die Ausrichtung sei dabei orientiert an der zeitlichen Reihenfolge der Frames I_0 und I_1 . Der Vorwärtsfluss $\vec{v}_{0,f}$ entspricht demnach dem bisherigen verwendeten Fluss \vec{v}_0 mit der Bewegung von I_0 nach I_1 und der Rückwärtsfluss $\vec{v}_{1,b}$ der Bewegung von I_1 nach I_0 .

Da prinzipiell beide Flussrichtungen gleichwertig sind, stellt sich die Frage, warum nicht der Rückwärtsfluss $\vec{v}_{1,b}$ verwendet werden sollte. Was sich bekanntlich zwischen den Flüssen unterscheiden könnte ist die Qualität, wodurch beispielsweise der bisher verwendete Vorwärtsfluss $\vec{v}_{0,f}$ eine schlechtere Qualität besitzen könnte als der Rückwärtsfluss $\vec{v}_{1,b}$. In diesem Fall wären dementsprechend beim Vorwärtsfluss auch schlechtere Interpolanten zu erwarten. Es fällt also schwer eine Richtung als die im Allgemeinen „bessere“ Richtung zu bezeichnen, was die Motivation für

die nächste Klasse an Verfahren darstellt. Die sogenannten bidirektionalen Verfahren versuchen eine ungewollte Bevorzugung einer Richtung des optischen Flusses möglichst zu minimieren oder bis zu einem gewissen Grad auszugleichen. Dazu arbeiten sie immer mit beiden Flussrichtungen und verwenden daher neben den zwei Bildern I_0 und I_1 insbesondere zwei erzeugte Motion Estimations (optischer Fluss).

In dieser Arbeit werden insgesamt zwei bidirektionale Verfahren vorgestellt, die gewissermaßen einen Ersatz für die bereits vorgestellten symmetrischen Verfahren darstellen: Forward Forward Warping und Backward Backward Warping. Dazu kommen noch zwei bidirektionale Varianten der symmetrischen Verfahren des 2-Bild Falls. Bevor diese eingeführt werden, soll im nächsten Abschnitt noch kurz die Handhabung eines Rückwärtsflusses anhand des 1-Bild falls klar gemacht werden.

4.5.1 Umgedrehter 1-Bild Fall

Beginnt man bei den Grundverfahren Forward Warping und Backward Warping, so ergeben sich durch die Verwendung des Rückwärtsflusses $\vec{v}_{1,b}$ folgende umgedrehte Varianten: Das bisherige Forward Warping aus Abschnitt 4.3.1 kann umgedreht für einen Interpolanten I_t definiert werden über:

$$I_t(\vec{x} + (1 - t) \cdot \vec{v}_{1,b}(\vec{x})) = I_1(\vec{x}) \quad (4.10)$$

Zu beachten ist, dass sich der Zeitpunkt t immer noch an der alten Richtung orientiert (Richtung des Vorwärtsflusses $\vec{v}_{0,f}$). Für das Backward Warping aus Abschnitt 4.3.2 ergibt sich in der umgedrehten Richtung folgende Definition für den gewarp-ten Fluss:

$$\vec{v}_{t,b}(\vec{x} + (1 - t) \cdot \vec{v}_{1,b}(\vec{x})) = \vec{v}_{1,b}(\vec{x}) \quad (4.11)$$

Mit dem berechneten Fluss $\vec{v}_{t,b}$ ergibt sich ein Zwischenbild I_t über:

$$I_t(\vec{x}) = I_0(\vec{x} + t \cdot \vec{v}_{t,b}(\vec{x})) \quad (4.12)$$

In Abbildung 4.8 sind beide umgedrehten Varianten zur Veranschaulichung erneut visualisiert. Es ist im Allgemeinen nicht klar, welche Richtung des Flusses für das Forward und Backward Warping die besseren Ergebnisse erzeugt, da dies stark abhängig vom verwendeten optischen Fluss ist. Weil beispielsweise der Vorwärtsfluss unter Umständen qualitativ schlechter sein könnte als der Rückwärtsfluss, ist die Überlegung der kommenden Verfahren, dass man dies durch eine Mittelung der

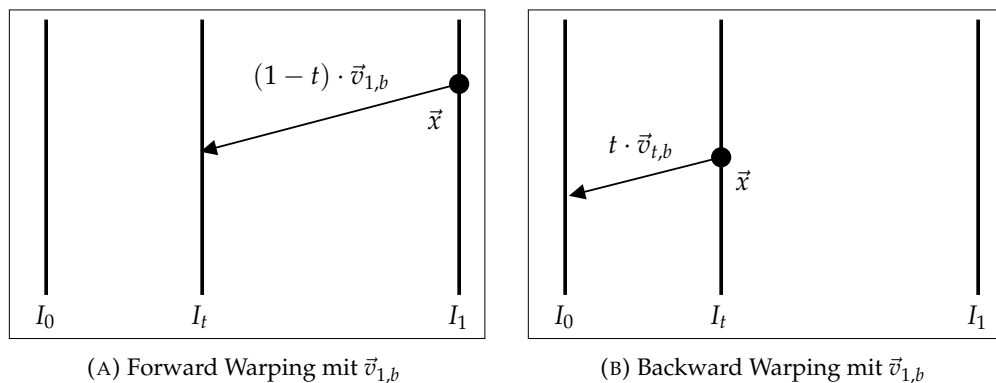


ABBILDUNG 4.8: Veranschaulichung der umgedrehten Interpolationsverfahren aus dem 1-Bild Fall. Es wird ausschließlich mit dem Rückwärtsfluss $\vec{v}_{1,b}$ gearbeitet.

Forward Warpings oder Backward Warpings beider Richtungen ausgleichen kann. Dadurch würde man zwar nicht mehr näher an den Ergebnisse der bessere Flussrichtung liegen, aber gleichzeitig sollte dies besser sein als bei der schlechteren Flussrichtung. Die im nächsten Abschnitt vorgestellten Verfahren Forward Forward Warping und dem Backward Backward Warping machen genau das und stellen dadurch gewissermaßen eine bidirektionale Variante des 1-Bild Falls dar. Gleichzeitig ähneln sie aber auch den (direktionalen) symmetrischen Verfahren aus Abschnitt 4.4, weil das Interpolationsschema sehr ähnlich ist.

4.5.2 Forward Forward Warping

Beim Forward Forward Warping wird mit zwei Bildinformationen I_0 und I_1 und den optischen Flüssen beider Richtungen $\vec{v}_{0,f}$ und $\vec{v}_{1,b}$ gearbeitet. Forward Forward Warping findet konzeptuell beispielsweise Anwendung in der Arbeit von Lobachev et al. [23] und stellt das erste Verfahren aus dem bidirektionalem 2-Bild Fall dar. Es ersetzt gewissermaßen das symmetrische Forward Warping aus Abschnitt 4.4.2, da es mit dem gleichen Interpolationsschema arbeitet, aber insgesamt mit lediglich zwei Warpingvorgängen auskommt. Insbesondere ist kein Fluss Warping mehr nötig, da direkt mit dem Rückwärtsfluss $\vec{v}_{1,b}$ gearbeitet werden kann (siehe Algorithmus 5). Mithilfe der Bilder I_0 und I_1 und den Flüssen $\vec{v}_{0,f}$ und $\vec{v}_{1,b}$ ergeben sich wieder bei einer Stelle \vec{x}_0 in I_0 und \vec{x}_1 in I_1 zwei temporäre Zwischenbilder $I_{t,0}$ und $I_{t,1}$:

$$\begin{aligned} I_{t,0}(\vec{x}_0 + t \cdot \vec{v}_{0,f}(\vec{x}_0)) &= I_0(\vec{x}_0) \\ I_{t,1}(\vec{x}_1 + (1-t) \cdot \vec{v}_{1,b}(\vec{x}_1)) &= I_1(\vec{x}_1) \end{aligned} \quad (4.13)$$

Der temporäre Interpolant $I_{t,0}$ entsteht über das bisherige Forward Warping Schema aus Abschnitt 4.3.1, was bei $I_{t,1}$ auch der Fall ist, allerdings dort mit Startpunkt des Flusses in I_1 und dem Rückwärtsfluss $\vec{v}_{1,b}$ (siehe Abbildung 4.9 rechts). Die sich beim Interpolant treffenden Bildinformationen müssen noch zusammengeführt werden. Dies geschieht wie bisher über eine Gewichtung abhängig vom Abstand zum Interpolationszeitpunkt t (Zeitpunkt t bezieht sich auf Richtung von I_0 nach I_1). Damit ergibt sich beim Forward Forward Warping ein finaler Interpolant I_t über:

$$I_t(\vec{x}) = (1-t) \cdot I_{t,0}(\vec{x}) + t \cdot I_{t,1}(\vec{x}) \quad (4.14)$$

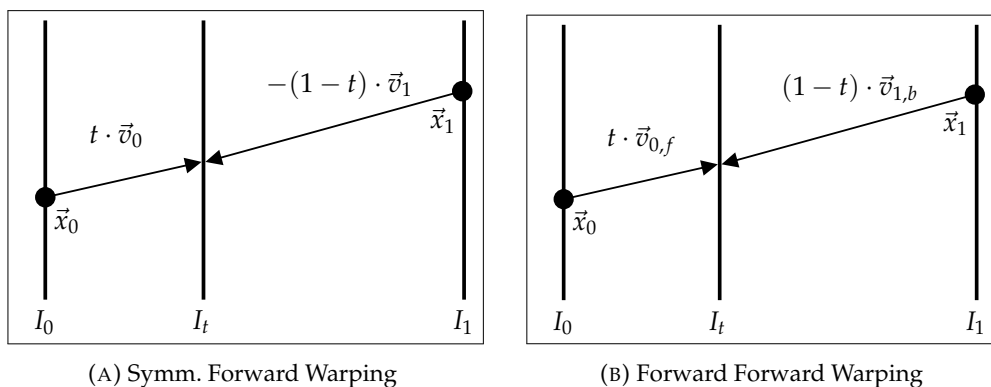


ABBILDUNG 4.9: Vergleich vom symmetrischen Forward Warping (mit Vorwärtsfluss \vec{v}_0) und dem Forward Forward Warping. Das Interpolationsschema ist sehr ähnlich allerdings wird beim Forward Forward Warping der Rückwärtsfluss $\vec{v}_{1,b}$ statt dem gewarpten und umgedrehten Fluss $-\vec{v}_1$ verwendet.

Algorithmus 5 : Forward Forward Warping.

Data : Zwei Bilder I_0, I_1 und optische Flüsse $\vec{v}_{0,f}, \vec{v}_{1,b}$ **Result** : Interpolant I_t

```

// Bildinformation aus  $I_0$ 
Elementares Forward Warping von  $I_0$  über den Fluss  $t \cdot \vec{v}_{0,f}$ 

// Bildinformation aus  $I_1$ 
Elementares Forward Warping von  $I_1$  über den Fluss  $(1 - t) \cdot \vec{v}_{1,b}$ 

// Berechne finalen Interpolanten  $I_t$ 
Gewichtete Zusammenführung beider Bildinformationen

// Füllen von Lücken im Bild
Inpainting aus Kapitel 3.5

```

Bei diesem Bildinterpolationsverfahren geschieht zusammengefasst nichts anderes als zwei unabhängige Forward Warpings, die anschließend gewichtet zusammengeführt werden (Blending). Während dieses Verfahren gedanklich dem symmetrischen Forward Warping sehr nahe kommt (Art und Weise wie Bildinformation zum Interpolanten transportiert wird), verzichtet dieses Verfahren jedoch auf das Warping eines optischen Flusses, was möglicherweise zu besseren Ergebnissen führen könnte. Ein Warpingvorgang ist immer mit Annahmen und Fehlern verbunden (Splating und Inpainting) und durch den Verzicht auf einen gewarpten Fluss ist anzunehmen, dass die Anzahl an Fehlerquellen geringer ist.

Beim Forward Forward Warping gibt es erneut eine gewichtete Zusammenführung von prinzipiell zwei vollständigen Interpolanten, was wieder mit Bedacht durchgeführt werden muss. In dieser Arbeit wird ein Vorgehen analog zu dem symmetrischen Forward Warping gewählt (vgl. Abschnitt 4.4.2). Das bedeutet, es findet wieder nur eine Zusammenführung der Interpolanten an den Stellen statt, an denen beide temporäre Interpolanten $I_{t,0}$ und $I_{t,1}$ einen gesetzten Farbwert besitzen. Dadurch kommt es auch beim Forward Forward Warping zu Lücken, was in einem abschließenden Inpainting wie in Kapitel 3.5 behoben werden muss. Algorithmus 5 fasst die Zwischenschritte dieses Interpolationsverfahrens zusammen: Nach der Erzeugung der zwei optischen Flüsse $\vec{v}_{0,f}$ und $\vec{v}_{1,b}$ finden damit zwei elementare Forward Warpings statt. Danach wird mit den Ergebnissen von dort eine gewichtete Zusammenführung durchgeführt, worauf ein abschließendes Inpainting folgt.

Zusammengefasst stellt das Forward Forward Warping eine Alternative zum symmetrischen Forward Warping dar. Wegen seiner geringen Anzahl an benötigten Warpingvorgängen (zwei) und den Verzicht auf ein Fluss Warping wirkt es für ein Verfahren des (bidirektionalen) 2-Bild Falls besonders vielversprechend.

4.5.3 Backward Backward Warping

Beim Backward Backward Warping handelt es sich um ein Verfahren, bei dem die Bildinformation I_0 und I_1 erneut über die Flüsse $\vec{v}_{0,f}$ und $\vec{v}_{1,b}$ zum Interpolanten I_t transportiert werden. Dieses Mal geschieht dies über zwei Backward Warpings, womit sich insgesamt ein Vorgehen sehr ähnlich zum symmetrischen Backward Warping ergibt (siehe Abbildung 4.10). Während beim Forward Forward Warping die geringe Anzahl an Warpings die besondere Eigenschaft war, ist es beim Backward

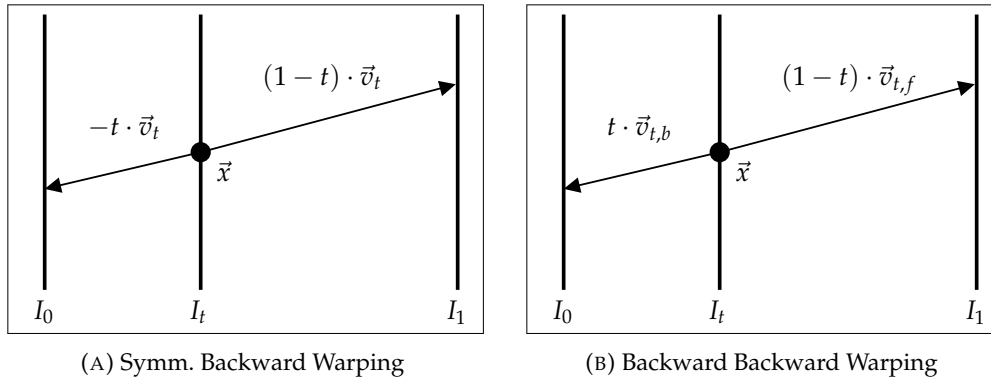


ABBILDUNG 4.10: Vergleich vom symmetrischen Backward Warping und dem Backward Backward Warping

Backward Warping die besonders hohe Anzahl an Warpingvorgängen. Im Vergleich zum symmetrischen Backward Warping steigt sie von drei auf vier an (vgl. Algorithmus 6). Das liegt an zwei notwendigen elementaren Forward Warpings der Flüsse $\vec{v}_{0,f}$ und $\vec{v}_{1,b}$ und den anschließenden zwei elementaren Backward Warpings für den Transport der Bildinformation I_0 und I_1 . Die für das Backward Warping notwendigen Flüsse $\vec{v}_{t,f}$ und $\vec{v}_{t,b}$ werden wieder über ein Fluss Warping der Flüsse $\vec{v}_{0,f}$ und $\vec{v}_{1,b}$ bestimmt:

$$\begin{aligned} \vec{v}_{t,f}(\vec{x}_0 + t \cdot v_0(\vec{x}_0)) &= \vec{v}_{0,f}(\vec{x}_0) \\ \vec{v}_{t,b}(\vec{x}_1 + (1-t) \cdot \vec{v}_{1,b}(\vec{x}_1)) &= \vec{v}_{1,b}(\vec{x}_1) \end{aligned} \quad (4.15)$$

Dabei entspricht \vec{x}_0 einer Stelle im Fluss $\vec{v}_{0,f}$ und \vec{x}_1 einer Stelle im Fluss $\vec{v}_{1,b}$. Im Anschluss wird analog zum symmetrischen Backward Warping versucht aus beiden Nachbarbildern I_0 und I_1 zu sampeln. Ein Interpolant I_t ist beim Backward Backward Warping für die Bilder I_0 und I_1 und den Flüssen $\vec{v}_{t,f}$ und $\vec{v}_{t,b}$ gegeben durch:

$$I_t(\vec{x}) = (1-t) \cdot I_0(\vec{x} + t \cdot \vec{v}_{t,b}(\vec{x})) + t \cdot I_1(\vec{x} + (1-t) \cdot \vec{v}_{t,f}(\vec{x})) \quad (4.16)$$

Beim Backward Backward Warping gibt es wie beim symmetrischen Backward Warping den Versuch aus möglichst beiden Nachbarbildern zu sampeln. Es soll, analog zum symmetrischen Backward Warping, eine Stelle im Interpolanten nur gesetzt werden, falls beide Samplingvorgänge erfolgreich sind. Dadurch entstehen wieder Lücken im Interpolanten, die mit einem abschließenden Inpainting wie in Kapitel 3.5 behoben werden müssen. Algorithmus 6 fasst die Zwischenschritte dieses Interpolationsverfahrens zusammen: Zuerst werden die erzeugten Flüsse $\vec{v}_{0,f}$ und $\vec{v}_{1,b}$ um sich selbst nach vorne gewarpt, um $\vec{v}_{t,f}$ und $\vec{v}_{t,b}$ zu erhalten. Danach können damit zwei elementare Backward Warpings durchgeführt werden, um die Bildinformationen zum Interpolanten zu transportieren. Diese werden stellenweise gewichtet zusammengeführt und im Falle einer entstandenen Lücke, findet eine Behebung durch das abschließende Inpainting wie in Kapitel 3.5 statt.

4.5.4 Bidirektionaler 2-Bild Fall

Neben den neu entstandenen bidirektionalen Verfahren (Forward Forward Warping und Backward Backward Warping) sollen zum Schluss noch bidirektionalen Varianten von bereits vorgestellten direktionalen Verfahren festgehalten werden. Diese sollen eine ungünstige Benachteiligung einer Flussrichtung abschwächen, indem

Algorithmus 6 : Backward Backward Warping.**Data :** Zwei Bilder I_0, I_1 und optische Flüsse $\vec{v}_{0,f}, \vec{v}_{1,b}$ **Result :** Interpolant I_t // Bestimme den Fluss $\vec{v}_{t,f}$ Forward Warping von $\vec{v}_{0,f}$ über den Fluss $t \cdot \vec{v}_{0,f}$ // Bestimme den Fluss $\vec{v}_{t,b}$ Forward Warping von $\vec{v}_{1,b}$ über den Fluss $(1 - t) \cdot \vec{v}_{1,b}$ // Bildinformation aus I_0 Elementares Backward Warping von I_0 über den Fluss $t \cdot \vec{v}_{t,b}$ // Bildinformation aus I_1 Elementares Backward Warping von I_1 über den Fluss $(1 - t) \cdot \vec{v}_{t,f}$ // Berechne finalen Interpolanten I_t

Gewichtete Zusammenführung beider Bildinformationen

// Füllen von Lücken im Bild

Inpainting aus Kapitel 3.5

das Mittel der Ergebnisse beider Richtungen genommen wird. Zwar kann es dadurch sein, dass ein schlechteres Ergebnis entsteht als bei einer der zwei Richtungen, aber insgesamt sollte ein konsistenteres Verfahren vorliegen. Im Folgenden werden die zwei symmetrischen Verfahren des 2-Bild Falls als bidirektionale Variante vorgestellt. Der 1-Bild Fall wurde gewissermaßen schon durch das Forward Forward Warping und durch das Backward Backward Warping behandelt.

Ein Interpolant I_t ist durch bidirektionales symmetrisches Backward Warping gegeben über die Mittelung zweier symmetrischer Backward Warpings. Es wird also das Schema aus Abschnitt 4.4.1 einmal mithilfe des Vorwärtsflusses $\vec{v}_{0,f}$ und einmal mithilfe des Rückwärtsflusses $\vec{v}_{1,b}$ angewandt. Für dieses Verfahren sind die gewarpften Flüsse wie beim vorherigem Backward Backward Warping notwendig:

$$\begin{aligned}\vec{v}_{t,f}(\vec{x}_0 + t \cdot v_0(\vec{x}_0)) &= \vec{v}_{0,f}(\vec{x}_0) \\ \vec{v}_{t,b}(\vec{x}_1 + (1 - t) \cdot \vec{v}_{1,b}(\vec{x}_1)) &= \vec{v}_{1,b}(\vec{x}_1)\end{aligned}\tag{4.17}$$

Dabei entspricht \vec{x}_0 einer Stelle im Fluss $\vec{v}_{0,f}$ und \vec{x}_1 einer Stelle im Fluss $\vec{v}_{1,b}$. Diese Flüsse werden verwendet, um mithilfe der Bilder I_0 und I_1 einen Interpolant I_t über das Mitteln zweier symmetrischer Backward Schemata zu bestimmen:

$$\begin{aligned}I_t(\vec{x}) &= \frac{1}{2} \left((1 - t) \cdot I_0(\vec{x} - t \cdot \vec{v}_{t,f}(\vec{x})) + t \cdot I_1(\vec{x} + (1 - t) \cdot \vec{v}_{t,f}(\vec{x})) \right) \\ &\quad + \frac{1}{2} \left((1 - t) \cdot I_0(\vec{x} + t \cdot \vec{v}_{t,b}(\vec{x})) + t \cdot I_1(\vec{x} - (1 - t) \cdot \vec{v}_{t,b}(\vec{x})) \right)\end{aligned}\tag{4.18}$$

Durch das Mitteln von prinzipiell zwei vollständigen Interpolanten, kommt erneut die Frage nach dem geeigneten Zeitpunkt für das Füllen der entstandenen Lücken auf. In dieser Arbeit wird sich beim Mitteln zweier Interpolanten für das gleiche Vorgehen entschieden, wie beim symmetrischen Forward Warping und dem Forward Forward Warping. Das bedeutet, es findet ein finales Inpainting erst nach dem Mitteln statt und in allen Zwischenschritten davor nicht (gewarpfte Flüsse ausgenommen). Außerdem wird das Mitteln nur an den Stellen vorgenommen, die von

Algorithmus 7 : Bidirektionales symmetrisches Backward Warping.**Data** : Zwei Bilder I_0, I_1 und optische Flüsse $\vec{v}_{0,f}, \vec{v}_{1,b}$ **Result** : Interpolant I_t

```

// Interpolant über Vorwärtsfluss  $\vec{v}_{0,f}$ 
Symmetrisches Backward Warping mit  $\vec{v}_{0,f}$  (ohne abschließendes Inpainting)

// Interpolant über Rückwärtsfluss  $\vec{v}_{1,b}$ 
Symmetrisches Backward Warping mit  $\vec{v}_{1,b}$  (ohne abschließendes Inpainting)

// Berechne finalen Interpolanten  $I_t$ 
Mitteln der Interpolanten beider Flussrichtungen

// Füllen von Lücken im Bild
Inpainting aus Kapitel 3.5

```

beiden Interpolanten der zwei Flussrichtungen besetzt sind. In Algorithmus 7 wird dieses Vorgehen nochmal verdeutlicht. Bei der bidirektionalen Variante des symmetrischen Backward Warpings finden in Summe zwei elementare Forward Warpings und vier elementare Backward Warpings statt. Dabei wird zwei Mal ein Fluss zum Interpolanten nach vorne gewarpt und pro Flussrichtung sind zwei elementare Backward Warpings für das Sampling aus den Bildern I_0 und I_1 notwendig. Ob diese hohe Anzahl an Warpingvorgängen im Verhältnis zum Endergebnis steht, wird im nächsten Kapitel geklärt. Dieses bidirektionale Verfahren findet beispielsweise Anwendung in der Arbeit von Râket et al. [31].

Auch das symmetrische Forward Warping kann als bidirektionale Variante definiert werden. Dafür sind zunächst zwei Fluss Warpings notwendig:

$$\begin{aligned}\vec{v}_{1,f}(\vec{x} + v_{0,f}(\vec{x})) &= v_{0,f}(\vec{x}) \\ \vec{v}_{0,b}(\vec{x} + v_{1,b}(\vec{x})) &= v_{1,b}(\vec{x})\end{aligned}\tag{4.19}$$

Für zwei Bilder I_0, I_1 und den Flüssen $\vec{v}_{0,f}, \vec{v}_{1,b}$ ergibt sich dieses Mal ein Zwischenbild I_t über vier temporäre Zwischenbilder. Für den Vorwärtsfluss $\vec{v}_{0,f}$, einer Stelle \vec{x}_0 in I_0 und einer Stelle \vec{x}_1 in I_1 ergeben sich die zwei Zwischenbilder $I_{t,0,f}$ und $I_{t,1,f}$:

$$\begin{aligned}I_{t,0,f}(\vec{x}_0 + t \cdot \vec{v}_{0,f}(\vec{x}_0)) &= I_0(\vec{x}_0) \\ I_{t,1,f}(\vec{x}_1 - (1-t) \cdot \vec{v}_{1,f}(\vec{x}_1)) &= I_1(\vec{x}_1)\end{aligned}\tag{4.20}$$

Für den Rückwärtsfluss $\vec{v}_{1,b}$ ergeben sich die zwei Zwischenbilder $I_{t,0,b}$ und $I_{t,1,b}$:

$$\begin{aligned}I_{t,0,b}(\vec{x}_0 - t \cdot \vec{v}_{0,b}(\vec{x}_0)) &= I_0(\vec{x}_0) \\ I_{t,1,b}(\vec{x}_1 + (1-t) \cdot \vec{v}_{1,b}(\vec{x}_1)) &= I_1(\vec{x}_1)\end{aligned}\tag{4.21}$$

Diese vier Zwischenbilder werden für den finalen Interpolanten I_t folgendermaßen zusammengeführt:

$$\begin{aligned}I_t(\vec{x}) &= \frac{1}{2} \left((1-t) \cdot I_{t,0,f}(\vec{x}) + t \cdot I_{t,1,f}(\vec{x}) \right) \\ &\quad + \frac{1}{2} \left((1-t) \cdot I_{t,0,b}(\vec{x}) + t \cdot I_{t,1,b}(\vec{x}) \right)\end{aligned}\tag{4.22}$$

Algorithmus 8 : Bidirektionales symmetrisches Forward Warping.

Data : Zwei Bilder I_0, I_1 und optische Flüsse $\vec{v}_{0,f}, \vec{v}_{1,b}$
Result : Interpolant I_t

```
// Interpolant über Vorwärtsfluss  $\vec{v}_{0,f}$ 
Symmetrisches Forward Warping mit  $\vec{v}_{0,f}$  (ohne abschließendes Inpainting)

// Interpolant über Rückwärtsfluss  $\vec{v}_{1,b}$ 
Symmetrisches Forward Warping mit  $\vec{v}_{1,b}$  (ohne abschließendes Inpainting)

// Berechne finalen Interpolanten  $I_t$ 
Mitteln der Interpolanten beider Flussrichtungen

// Füllen von Lücken im Bild
Inpainting aus Kapitel 3.5
```

Im Vergleich zur direktionalen Variante findet hier eine noch größere Zusammenführung von temporären Zwischenbildern statt, was durch das dazugekommene Mitteln verursacht wird. Analog zum bidirektionalem symmetrischen Backward Warping muss auch hier ein Umgang mit den Lücken festgelegt werden. Dies geschieht wie bisher, indem nur ein einziges Inpainting am Schluss die überlagerten Lücken aller Interpolanten füllt (vgl. Algorithmus 8). Beim bidirektionalen symmetrischen Forward Warping ist die Anzahl an Warpings ebenfalls sehr hoch. Es finden pro Richtung des Flusses analog wie in Abschnitt 4.4.2 drei Forward Warpings statt. Damit ergeben sich durch das Mitteln beider Richtungen in Summe sechs Forward Warpingvorgänge, wovon zwei Mal ein Fluss gewarpt wurde.

4.6 Vergleich

In diesem Kapitel wurden alle betrachteten Interpolationsverfahren vorgestellt. Für jedes Verfahren wurde eine explizite Definition und teilweise auch einer Berechnungsvorschrift angegeben. Um alle Verfahren komprimiert gegenüberzustellen, bündelt Tabelle 4.1 die Verfahren und gibt jeweils die enthaltene Anzahl an Warpings und Inpaintings an. An dieser Tabelle wird nochmal klar, dass sich die Verfahren durchaus unterscheiden, obwohl sie sich konzeptuell zum Teil ähneln. Im nächsten Kapitel werden diese Verfahren schließlich getestet und es wird untersucht, inwiefern die Anzahl an Warpings für die Qualität der Interpolanten von Bedeutung ist.

Verfahren	# Forward Warping	davon Flüsse	# Backward Warping	# Inpainting	davon Flüsse
Forward Warping	1	0	0	1	0
Backward Warping	1	1	1	2	1
Symm. FW	3	1	0	2	1
Symm. BW	1	1	2	2	1
FFW	2	0	0	1	0
BBW	2	2	2	3	2
Bidir. Symm. FW	6	2	0	3	2
Bidir. Symm. BW	2	2	4	3	2

TABELLE 4.1: Alle betrachteten Bildinterpolationsverfahren. Es wurde jeweils die Anzahl der involvierten elementaren Warpings und Inpaintings angegeben.

Kapitel 5

Experimente

Nachdem im letzten Kapitel zahlreiche Bildinterpolationsverfahren vorgestellt wurden, sollen diese in diesem Kapitel anhand mehrerer Testdaten getestet und verglichen werden. Es ist bereits aufgefallen, dass sich einige Verfahren in ihrer Art ähneln, aber wie sie tatsächlich gegenüber abschneiden ist nicht ganz intuitiv klar. Um die Verfahren einzustufen, wird eine Reihe von Experimenten durchgeführt, die die Verfahren auf ihre individuellen Eigenschaften testen und so mögliche Probleme aufzeigen. Die verwendeten Testdaten, Fehlermaße und Implementierungsdetails werden in den folgenden Abschnitten zunächst vorgestellt und im Anschluss folgen die einzelnen Experimente.

Begonnen wird in Abschnitt 5.4 mit dem 1-Bild Fall, wo die drei vorgestellten Samplingstrategien und die sechs verschiedenen Splattingstrategien anhand von Interpolanten, die durch ein Forward Warping und ein Backward Warping entstanden sind, untersucht werden. Danach folgt in Abschnitt 5.5 eine Betrachtung des vorgestellten Inpaintingverfahrens, das mit Hinblick auf Geschwindigkeit und Anwendung beim Forward Warping und Backward Warping untersucht wird. In Abschnitt 5.6 werden die Verfahren des 2-Bild Falls genauer betrachtet, worauf in Abschnitt 5.7 eine Überprüfung der möglichen Verbesserungen durch den bidirektionalen Aspekt folgt. Zum Schluss soll noch in Abschnitt 5.8 ein Gesamtvergleich die Interpolationsverfahren mit den besten Ergebnissen aufzeigen und damit dieses Kapitel abschließen.

5.1 Testdaten

Wie auch bei optischen Flüssen kann es bei der Zwischenbildinterpolation mit optischem Fluss je nach Szene zu stark unterschiedlich guten Ergebnissen kommen. Dies ist der maßgeblichen Abhängigkeit des verwendeten optischen Flusses zu schulden, was bereits in den vergangenen Kapiteln thematisiert wurde. Aus diesem Grund soll eine Auswahl möglichst vielfältiger Szenen verwendet werden, um einen möglichst repräsentativen Eindruck der Interpolationsverfahren zu erhalten. Mit dieser Motivation wird in den Tests auf Benchmarks für optische Flüsse zurückgegriffen, da diese Benchmarks ebenfalls versuchen eine vielfältige Auswahl an Szenen abzudecken. Es wird in dieser Arbeit primär mit den Szenen des Middlebury Benchmarks [2] gearbeitet, das gleich nochmal genauer vorgestellt wird. Für die Tests der Bildinterpolationsverfahren aus dem letzten Kapitel stellt sich die Frage, was als Referenzbild gegenüber dem Interpolanten eigentlich dienen soll (Ground Truth). Da bei einer Bildsequenz das tatsächliche Zwischenbild zwischen zwei Frames nicht existiert, werden alternativ drei aufeinander folgende Frames einer Bildsequenz für die Evaluation genutzt. Dabei ist das Ziel, einen Interpolanten zwischen den äußeren



ABBILDUNG 5.1: Visualisierung vom Auslassen von Frames, um als Ground Truth für die Tests zu dienen (Szene basketball des Middlebury Benchmarks [2]).

Frames zu erzeugen und den mittleren Frame als Referenzbild zu verwenden (siehe Abbildung 5.1). Durch den erhöhten zeitlichen Abstand gegenüber zwei direkt aufeinander folgenden Frames, kommt es aber im Allgemeinen zu größeren Verschiebungen im Bild, was womöglich Einfluss auf die Qualität der optischen Flüsse hat. Aber auch die Interpolationsverfahren leiden darunter, da es auch größere Verdeckungen gibt, die durch Inpainting gefüllt werden müssen. Aufgrund dessen kommt es womöglich zu größeren Unterschieden zum Referenzbild im Vergleich zur Zwischenbildinterpolation von zwei direkt aufeinanderfolgenden Frames. Die größeren Bewegungen und Fehler wären bei der Zwischenbildinterpolation von üblichen Bildsequenzen mit einer Wiederholungsrate von 24 fps oder 30 fps ein durchweg vorherrschendes Problem. Bei Benchmarks für optischen Flüssen werden aber passenderweise häufig High-Speed-Aufnahmen mit einer deutlich erhöhten Frame-rate verwendet, wo durch den geringeren zeitlichen Abstand der Frames wieder eine etwas geringere Bewegung im Bild stattfindet. Damit erhält man selbst bei einer Evaluation mit ausgelassenen Frames vergleichbare Ergebnisse zu einer Interpolation aufeinanderfolgender Frames einer üblichen Videoaufnahme.

Middlebury Benchmark Es werden Bildsequenzen des Middlebury Benchmarks [2] für die Evaluation verwendet, das eine Sammlung vielfältiger Szenen beinhaltet. Dazu gehören Naturszenen, Szenen aus dem Straßenverkehr, Szenen naher Objekte und teilweise auch computergenerierte Bildfolgen. Da bei der dortigen Evaluation von optischen Flüssen immer die Frames 10 und 11 verwendet werden, kommen bei den Tests dieser Arbeit immer Frame 10 bis 12 für die Evaluation der Zwischenbildinterpolation zum Einsatz. Sie werden verwendet, damit man sich in einem ähnlichen Zeitraum bewegt. Als Ground Truth wird immer Frame 11 einer Bildsequenzen verwendet. In Tabelle 5.1 befindet sich eine Auflistung aller getesteten Szenen, wovon später in den Experimenten aus Platzgründen immer nur die ersten sieben Szenen aufgeführt werden. Der in den Experimenten angegebene Durchschnitt bezieht sich aber immer auf alle zehn getesteten Szenen. Wie bereits in Kapitel 2 beschrieben, kann es bei optischen Flüssen zu Ungenauigkeiten kommen. Dies macht sich schon vor der eigentlichen Zwischenbildinterpolation bemerkbar und sollte bei

Army	Backyard	Basketball	Dumptruck	Evergreen	Grove	Mequon	Schefflera	Urban	Wooden
584x388	640x480	640x480	640x480	640x480	640x480	584x388	584x388	640x480	584x388

TABELLE 5.1: Die verwendeten Szenen des Middlebury Benchmarks [2] und die dazugehörigen Auflösungen der Bildsequenzen.

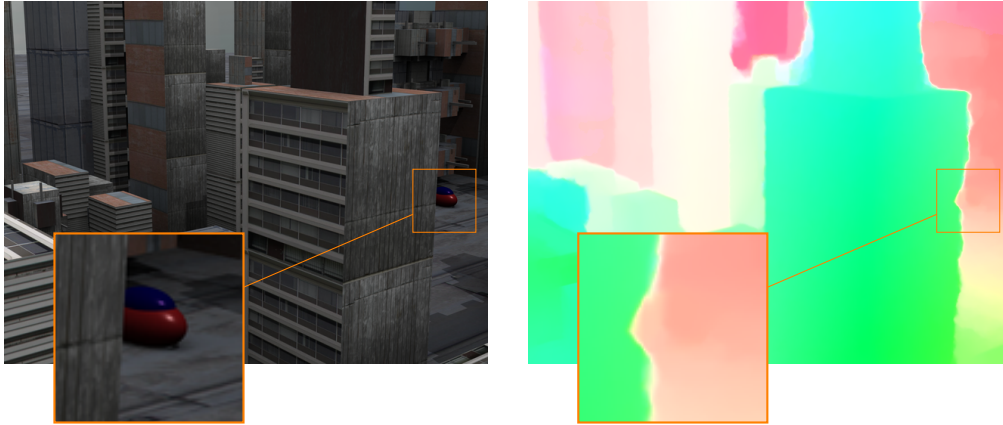


ABBILDUNG 5.2: Ungenauigkeiten bei verwendetem optischem Fluss. Bei der Szene Urban wird beispielsweise die Bewegung des Fahrzeugs nicht erkannt.

der Bewertung im Hinterkopf behalten werden. Wenn beispielsweise die Bewegung mancher Objekte nicht im optischen Fluss erkannt wird, ist klar, dass die Zwischenbildinterpolation im Zusammenhang mit diesem Objekt auch fehlerhaft sein wird. Speziell beim Middlebury Benchmark gibt es durch das verwendete Verfahren für den optischen Fluss (vgl. Kapitel 2) durchaus Fälle, wo dieses Phänomen zu beobachten ist (siehe Abbildung 5.2). Einerseits als Problem gesehen, sind diese Ungenauigkeiten typisch für Zwischenbildinterpolation mit optischem Fluss und aus diesem Grund wird ein nicht perfekter Fluss als Regelfall angenommen.

5.2 Fehlermaße

Es ist naheliegend, dass es bei den erzeugten Zwischenbildern zu Unterschieden gegenüber dem Referenzbild kommen wird, weswegen ein Maß notwendig ist, um einen Interpolanten zu bewerten. Es existieren in der Literatur zahlreiche Fehlermaße, die sich für die Bewertung von Bildern anbieten würden, wie zum Beispiel der *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE) oder die *Peak signal to noise ratio* (PSNR) [28, 26]. In dieser Arbeit wird überwiegend auf den *Root Mean Squared Error* (RMSE) zurückgegriffen, der für einen Interpolanten $I(\vec{x})$ und einem Referenzbild $I_{GT}(\vec{x})$ mit einer Pixelanzahl von N gegeben ist über:

$$RMSE(I, I_{GT}) = \left[\frac{1}{N} \sum_{\vec{x}} (\|I(\vec{x}) - I_{GT}(\vec{x})\|_2)^2 \right]^{1/2}. \quad (5.1)$$

Dieser Fehler bietet sich gut für die Zwischenbildinterpolation an, da er einen relativ anschaulichen Eindruck über die Qualität gibt, indem er den mittleren Abstand zur korrekten Intensität darstellt. Es lassen sich allerdings Fälle konstruieren, in denen Schwächen dieser klassischen Fehlermaße auffallen, die zeigen, dass es teilweise zu starken Unterschieden zur menschlichen Bewertung kommt. In Abbildung 5.3 ist so ein Fall dargestellt. Es sind dort zwei deutlich unterschiedlich aussehende Interpolanten zu sehen, was sich im RMSE als Fehlermaß aber nicht wieder spiegelt. Dieser ist für beide Interpolanten identisch. Motiviert durch solche Fälle, wird nach komplexeren Bewertungen gesucht, die der tatsächlichen Wahrnehmung des Menschen näher kommen [36, 26]. Gradientenbasierte Verfahren [2] oder ein modifizierter PSNR [9] gehören beispielsweise dazu. In dieser Arbeit wird daher

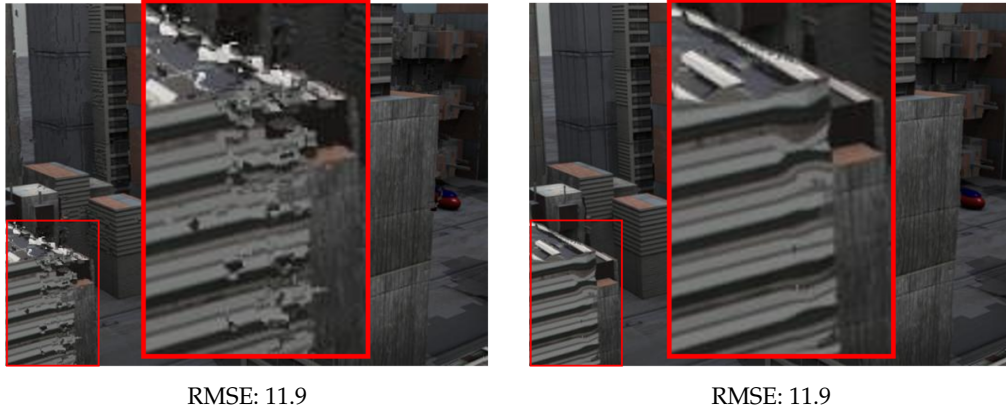


ABBILDUNG 5.3: Zwei interpolierte Frames, welche mittels verschiedener Verfahren bestimmt wurden. Der RMSE ist für beide Frames identisch, aber visuell sind deutliche Unterschiede zu erkennen (Quelle [26]).

noch zusätzlich auf ein Maß für die strukturelle Ähnlichkeit zwischen den Interpolanten I und dem Referenzbild I_{GT} zurückgegriffen. Verwendet wird die *Structural Similarity* (SSIM) von Wang et al. [36] als ein nützliches Maß neben dem klassischen RMSE. Ein sogenannter SSIM Index bewertet die strukturelle Ähnlichkeit zwischen zwei Bildern anhand einer gewichteten Kombination von einem Luminanzvergleich, Kontrastvergleich und einem strukturellem Vergleich. Diese werden nicht global bestimmt, sondern verteilt auf vielen sogenannten *Patches* im Bild. Eine solche lokale Bewertung soll abgestimmt auf die Wahrnehmung des menschlichen Auges eine differenziertere Betrachtung gegenüber Fehlermaßen wie beispielsweise dem RMSE vornehmen. Ein Index für die SSIM ist für die Patches x und y in I und I_{GT} gegeben über eine gewichtete Kombination vom Luminanzvergleich $l(x, y)$, Kontrastvergleich $c(x, y)$ und dem strukturellem Vergleich $s(x, y)$:

$$SSIM(x, y) = [l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma]. \quad (5.2)$$

Verwendet wird hier $\alpha = \beta = \gamma = 1$. Die einzelnen Vergleiche sind gegeben über:

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \quad c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \quad s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad (5.3)$$

Dabei entspricht μ den Durschnitten und σ den Varianzen der Patches x und y . Bei den Konstanten c_i handelt es sich noch um festlegbare Konstanten, die unter anderem vom Dynamikumumfang im Bild abhängen. Ein solcher SSIM Index wird in dieser Arbeit, angepasst an die Arbeit von Wang et al. [36], für ein Patch mit Fenstergröße 11 berechnet, die sich durch Verwendung eines normalisierten Gauß-Kernels mit Varianz $\sigma = 1.5$ ergibt. Dieser Kernel wird verwendet, um die durchschnittlichen Werte und Varianz eines Patches noch zusätzlich zu gewichten. Da mit Farbbildern gearbeitet wird, werden die SSIM Indizes für jeden Farbkanal separat berechnet und anschließend gemittelt. Das Mittel aller berechneten SSIM Indizes der Patches x und y ergibt schließlich die endgültige Bewertung der Bilder I und I_{GT} :

$$SSIM(I, I_{GT}) = \sum_{x \in I, y \in I_{GT}} \frac{SSIM(x, y)}{N}. \quad (5.4)$$

Dabei liegen insgesamt N Patches im Bild vor. Durch die Mittelung aller SSIM Indizes äußert sich die Bewertung des Bildes in einer Zahl zwischen -1 und 1 , wobei 1 einer sehr hohen strukturellen Ähnlichkeit und -1 einer sehr geringen Ähnlichkeit entspricht. Für die Tests in diesem Kapitel ist also ein möglichst hoher mittlerer SSIM Index gewünscht, da dies eine hohe strukturelle Ähnlichkeit des erzeugten Interpolanten I gegenüber dem Referenzbild I_{GT} bedeutet. Für noch weitere Details sei auf die Arbeit von Wang et al. [36] verwiesen.

5.3 Implementierung

Die beschriebenen Verfahren aus Kapitel 4 wurden für die Experimente mit Python 3.7 umgesetzt. Es wurde dabei bei dem Ein- und Auslesen von Bildern mit der Bibliothek Pillow¹ gearbeitet. Für die eigentlichen Bildinterpolationsverfahren kam primär die Bibliothek NumPy² zum Einsatz. Die Berechnung der optischen Flüsse (ProFlow-Baseline) stammt aus einer hauseigenen Implementierung, die netterweise von den Autoren von ProFlow [24] zur Verfügung gestellt wurde. Für das Arbeiten mit den optischen Flüssen wurde das Optical Flow Toolkit³ von Ruoteng Li verwendet. Die Berechnung des SSIM Index als Fehlermaß wurde mit der Bibliothek scikit-image⁴ durchgeführt, die eine Konfiguration entsprechend der Arbeit von Wang et al. [36] verwendet (vgl. Erklärung von SSIM im vorherigen Abschnitt).

5.4 1-Bild Fall

In Kapitel 4 wurden das Forward Warping und das Backward Warping als Interpolationsverfahren vorgestellt. In diesem Abschnitt soll nun ihre Eignung für die Zwischenbildinterpolation geprüft und die involvierten Splatting- und Samplingstrategien sollen miteinander verglichen werden. Dazu gehören beim Splatting die Verteilungskonzepte von einem oder mehreren Nachbarn (1N und 4N) und die drei verschiedenen Gewichtungsstrategien Uniform, Gewicht und Austausch. Beim Sampling wird das Nearest Neighbour Sampling (neue Kurzform: 1N-Uniform), das uniforme Mitteln (neue Kurzform: 4N-Uniform) und die bilineare Interpolation (neue Kurzform: 4N-Gewicht) betrachtet. Untersucht werden diese Konzepte anhand mehrerer Interpolanten, die für die Middlebury Szenen berechnet werden, die aber kein Inpainting zum Schluss erhalten. Mit der Notation aus dem letzten Kapitel bedeutet das, dass für eine Szene des Middlebury Benchmarks Frame 10 dem Bild I_0 , Frame 12 dem Bild I_1 und Frame 11 dem Bild I_{GT} (Ground Truth) entspricht. Berechnet wird bei den Verfahren also immer der Fall $t = 0.5$, der einem Zwischenbild $I_{0.5}$ in der Mitte entspricht und mit dem Referenzbild I_{GT} (Frame 11) verglichen wird. Beim verwendeten Fluss handelt es sich immer um den Vorwärtsfluss $\vec{v}_{0,f}$. Neben einem visuellen Eindruck sollen die Ergebnisse anhand des vorgestellten RMSE quantitativ miteinander verglichen werden. Dazu werden aber in diesem Experiment nur besetzte Stellen in $I_{0.5}$ berücksichtigt und Lücken werden aus der Fehlerberechnung ausgeschlossen. Dies hat den Hintergrund, dass zunächst nur ein reiner Fokus auf dem Splatting und Sampling liegt und gefüllte Lücken, die die Gesamtqualität des Interpolanten verschlechtern könnten, ignoriert werden sollen. Aus diesem Grund

¹<https://python-pillow.org>

²<https://www.numpy.org>

³<https://github.com/liruoteng/OpticalFlowToolkit>

⁴<https://scikit-image.org>

ist auch die Anzahl der leeren Stellen im Interpolanten interessant, denn ein Verfahren was ohne betrachtete Lücken gute Ergebnisse liefert, kann bei vielen vorhandenen Lücken nach dem Inpainting möglicherweise deutlich schlechter abschneiden. In den folgenden zwei Abschnitten werden die Beobachtungen und Interpretationen beim Forward Warping und Backward Warping festgehalten. Im Anschluss folgt in Abschnitt 5.4.3 noch ein direkter Vergleich zwischen den besten Splattingstrategien des Forward Warpings und den besten Samplingstrategien des Backward Warpings.

5.4.1 Forward Warping

Beim Forward Warping werden die sechs Splattingstrategien untersucht, die sich durch die Kreuzung der zwei Verteilungsstrategien (1N und 4N) und den Gewichtsstrategien (Uniform, Gewicht und Austausch) ergeben. Diese werden bei einem Forward Warping, wie im letzten Kapitel beschrieben, verwendet, um einen Interpolanten $I_{0,5}$ zu bestimmen (vgl. Abschnitt 4.3.1).

Bei Betrachtung der Ergebnisse lassen sich starke Unterschiede beim RMSE von Szene zu Szene beobachten (siehe Tabelle 5.2). Bei der Interpolation der Szene Army kommt es beispielsweise zu den vergleichsweise kleinsten Fehlern (ca. 6 bis 8), gleichzeitig liegt in der Szene Grove ein fast vier Mal so großer Fehler vor (ca. 35 bis 38). Bei den verschiedenen Splattingstrategien fällt auf, dass die Variante 4N-Gewicht die besten Ergebnisse erzielt. 4N-Uniform bildet im Vergleich das Schlusslicht. Das besondere Splatting mit Austausch als Kollisionsbehandlung kann mit den anderen Varianten durchaus mithalten und scheint mit der 1N-Verteilung am besten zu funktionieren. Aber selbst 1N-Austausch ist im Vergleich zu den anderen Splattingstrategien mit einem Nachbar unterlegen. Die Anzahl an leeren Stellen unterscheiden sich, wie auch der Fehler, von Szene zu Szene sehr stark. Trotzdem handelt es sich um vergleichsweise wenige Stellen (1 bis 5% des gesamten Bildes). Generell fällt die deutlich geringere Anzahl an Lücken bei den 4N-Verteilungen auf. Dort ist sogar im Fall von der Szene dumptruck nicht eine einzige Lücke vorhanden.

Visuell sind beim Splatting auf vier Nachbarn (4N) im Vergleich zum Fall mit nur einem Nachbar (1N) weichere Bilder zu beobachten (siehe Abbildung 5.4). Insbesondere das 4N-Uniform Splatting erzeugt besonders unscharfe Ergebnisse im Vergleich

Splatting	Army	Backyard	Basketball	Dumptruck	Evergreen	Grove	Mequon	\emptyset
1N-Uniform	8.169	22.110	22.859	30.404	23.247	38.882	17.138	21.978
1N-Gewicht	8.173	22.109	22.856	30.400	23.248	38.880	17.159	21.981
1N-Austausch	8.234	22.169	22.900	30.416	23.353	38.959	17.388	22.057
# leere Stellen	2293	5144	5661	707	6941	12347	7891	6142.5
4N-Uniform	9.930	24.738	22.397	35.861	24.007	36.081	16.583	22.509
4N-Gewicht	6.477	21.474	22.063	29.508	22.548	35.068	16.124	20.746
4N-Austausch	8.311	22.334	22.954	30.524	23.865	38.983	17.634	22.178
# leere Stellen	125	946	288	0	365	1341	2600	1074.4

TABELLE 5.2: Übersicht der verschiedenen Splattingstrategien beim Forward Warping. Es ist der RMSE für alle im Bild besetzten Pixel angegeben (Lücken werden ignoriert). Für die zwei Verteilungsstrategien wurde außerdem die Anzahl an Lücken angegeben.



ABBILDUNG 5.4: Optischer Vergleich beim Forward Warping. Die vorhandenen Lücken im Bild sind rot markiert. Beim naiven Ansatz sind typische Doppelungsartefakte und ein unscharfes Ergebnis zu sehen. Die uniforme Verteilung auf vier Nachbarn wirkt ebenfalls etwas unschärfer als der Rest. Im Vergleich sind bei der Verteilung auf nur einen Nachbar mehr Lücken im Bild vorhanden als bei vier.

zum Rest. Die Verteilung auf nur einen Nachbarn scheint dagegen zum Teil schärfere Ergebnisse zu bewirken (siehe Abbildung 5.4). Die Ergebnisse dieses Experiments sind insgesamt nachvollziehbar. Die starken Unterschiede beim RMSE beispielsweise sind vermutlich auf starken Unterschiede zwischen den Szenen zurückzuführen. Je nach Art von Szene (zum Beispiel Naturszene oder computergeneriert), aber auch die unterschiedliche Bewegung im Bild sind wahrscheinlich die Ursache. Bei den Splattingstrategien hat eine breitere Verteilung auf vier Pixel (4N) deutlich weniger

Lücken im Bild verursacht als der 1N-Fall. Dies war die ursprüngliche Motivation dieser Verteilung und sie hat den erwarteten Effekt erfüllt. Besonders bei dem Splatting mit einer uniformen Gewichtung kam es zu weicheren Ergebnissen, da sehr viele Farbwerte verteilt und gleichmäßig gemittelt wurden. Der Fehler war dadurch dementsprechend hoch und ein relativ unscharfes Bild ist zu beobachten. Besonders die 4N-Verteilung hat diesen Effekt gegenüber der 1N-Verteilung nochmal verstärkt. Eine angepasste Gewichtung der Farbwerte (Gewicht) konnte dieser Verwischung wie erhofft gut entgegenwirken und sowohl bei einer 1N und 4N Verteilung verbesserte Ergebnisse liefern. Die Splattingstrategien mit einer 1N-Verteilung konnten zwar insgesamt vom Fehler nicht so gut abschneiden wie das 4N-Gewicht, doch liefern sie überraschend scharfe Ergebnisse im Vergleich.

Zusammenfassend lässt sich für die getesteten Splattingstrategien sagen, dass sich das 4N-Gewicht Verfahren als besonders vielversprechend herausgestellt hat, die anderen Strategien aber nicht sehr weit davon entfernt sind.

5.4.2 Backward Warping

Beim Backward Warping gibt es im Vergleich zum Forward Warping weniger Samplingstrategien zu prüfen. Getestet werden nur die drei Samplings 1N-Uniform, 4N-Uniform und 4N-Gewicht. Dabei entspricht zur Erinnerung 4N-Gewicht einer bilinearen Interpolation und 4N-Uniform einem einfachen Mitteln der Nachbarn. Trotz der geringeren Anzahl an Samplingstrategien gibt es beim Backward Warping in Summe mehr Varianten zu Vergleichen, da beim Backward Warping für die Zwischenbildinterpolation ein Forward Warping eines optischen Flusses involviert ist (vgl. Abschnitt 4.3.2). Das Warping des Flusses wird in diesem Abschnitt ebenfalls getestet, indem die sechs verschiedenen Splattingstrategien wie im letzten Abschnitt 5.4.1 erneut für optische Flüsse getestet werden. Das bei dem Forward Warping des Flusses enthaltene Inpainting wird im nächsten Experiment in Abschnitt 5.5 nochmal genauer behandelt. Durch die drei Strategien des Samplings und den sechs Strategien des Splatting ergeben sich also in Summe 18 Varianten des Backward Warpings. Diese Varianten des Backward Warpings erzeugen nun ein Zwischenbild $I_{0.5}$ mithilfe des Vorwärtsflusses $\vec{v}_{0,f}$ und sollen miteinander verglichen werden (vgl. Abschnitt 4.3.2). Es findet auch hier kein abschließendes Inpainting statt.

Mit Blick auf Tabelle 5.3 lässt sich eine konsistente Beobachtung machen: Die besten Ergebnisse scheint ein Backward Warping mit bilinearer Interpolation (4N-Gewicht) als Samplingstrategie zu liefern. Dies ist der Fall bei allen Arten des involvierten Splatting. Eine ähnliche Gruppierung der Splattingstrategien lässt sich auch bei den anderen zwei Samplingstrategien feststellen. Zum Beispiel erzeugt das Nearest Neighbour Sampling (1N-Uniform) mit den verschiedenen Splattingstrategien als Gesamtgruppe gesehen, leicht schlechtere Ergebnisse als die bilineare Interpolation. Das Schlusslicht bildet das einfache Mitteln von vier Nachbarn (4N-Uniform) mit allen möglichen Splattingstrategien als Gesamtgruppe. Bei Betrachtung der einzelnen Splatting für das Warping des Flusses, sind bei allen Samplingstrategien nur geringfügige Unterschiede festzustellen. Allerdings ist zu erwähnen, dass 1N-Uniform und 1N-Gewicht dem Sieger 4N-Gewicht durchaus sehr nahekommen. Für die 18 getesteten Varianten des Backward Warpings ist in Tabelle 5.4 jeweils die Anzahl der Lücken angegeben. Dort auffällig zu erkennen sind drei Blöcke von Zahlen, die identisch sind. Außerdem sind innerhalb eines Blocks der verschiedenen Splattingstrategien nur sehr kleine Unterschiede zu beobachten (1-3 Lücken Unterschied).

Sampling	Army	Backyard	Basketball	Dumptruck	Evergreen	Grove	Mequon	\emptyset
1N-Uniform								
1N-Uniform	8.253	24.375	23.079	29.207	24.139	43.110	20.985	23.123
1N-Gewicht	8.242	24.382	23.082	29.207	24.137	43.109	20.990	23.123
1N-Austausch	8.228	24.422	23.088	29.211	24.139	43.109	20.996	23.128
4N-Uniform	8.222	24.342	23.074	29.208	24.143	43.128	21.0486	23.113
4N-Gewicht	8.220	24.351	23.074	29.206	24.126	43.105	21.023	23.113
4N-Austausch	8.225	24.408	23.084	29.206	24.144	43.097	21.010	23.125
4N-Uniform								
1N-Uniform	11.332	26.783	22.478	35.596	24.430	39.976	20.528	23.686
1N-Gewicht	11.332	26.787	22.477	35.596	24.432	39.976	20.524	23.685
1N-Austausch	11.341	26.833	22.484	35.596	24.428	39.973	20.513	23.688
4N-Uniform	11.400	26.796	22.473	35.683	24.452	40.003	20.590	23.718
4N-Gewicht	11.353	26.766	22.468	35.592	24.427	39.982	20.562	23.690
4N-Austausch	11.342	26.815	22.484	35.597	24.426	39.970	20.513	23.686
4N-Gewicht								
1N-Uniform	7.102	23.499	22.052	28.216	22.855	39.139	19.987	21.807
1N-Gewicht	7.101	23.503	22.054	28.216	22.855	39.139	19.984	21.807
1N-Austausch	7.104	23.565	22.059	28.218	22.860	39.142	19.993	21.816
4N-Uniform	7.135	23.502	22.056	28.217	22.853	39.178	20.058	21.818
4N-Gewicht	7.110	23.481	22.049	28.217	22.846	39.146	20.025	21.806
4N-Austausch	7.108	23.546	22.060	28.218	22.864	39.144	19.997	21.817

TABELLE 5.3: Übersicht der verschiedenen Sampling- und Splattingstrategien beim Backward Warping. Es ist der RMSE für alle im Bild besetzten Pixel angegeben (Lücken werden ignoriert). In der ersten Spalte ist die Samplingstrategie optisch hervorgehoben (**fett**). Darunter befinden sich jeweils die sechs verschiedenen Splattingstrategien für das involvierte Warping des Flusses $\vec{v}_{0,f}$.

Abbildung 5.5 zeigt erneut einen vergrößerten Ausschnitt von der Szene Army zum optischen Vergleich der Strategien. Wie dort zu erkennen, kommt es erneut beim uniformen Sampling von vier Nachbarn (4N-Uniform) zu vergleichsweise unschärferen Ergebnissen. Damit unterliegt dieser Fall mit vier Nachbarn durchgängig dem Nearest Neighbour Sampling (1N-Uniform) im optischen Vergleich. In Abbildung 5.5 ist aber auch zu erkennen, dass die Ergebnisse vom Nearest Neighbour Sampling (1N-Uniform) im Vergleich zum bilinearen Sampling teilweise schärfer wirken, obwohl der berechnete Fehler in Tabelle 5.3 höher ist.

Die Ergebnisse beim Test der Samplingstrategien bestätigen die nachvollziehbare Annahme, dass eine bilineare Interpolation einem einfachen Mitteln der Nachbarpixel überlegen ist, da es zu weniger unscharfen Ergebnissen kommt und der Fehler merklich geringer ist. Diese Beobachtung in Tabelle 5.3 zu machen, war auch den sehr ähnlichen Ergebnissen des Splattingstrategien untereinander geschuldet. Scheinbar hat die Splattingstrategie bei Forward Warping des optischen Flusses ein vergleichsweise geringen Einfluss auf das Endergebnis gehabt, wobei auch beim Fall von optischen Flüssen es erneut so scheint, als ob ein 4N-Gewicht Splatting die besten Ergebnisse liefert.

Die Lückenanzahl in Tabelle 5.4 eignet sich gut, um die Funktionsweise des Splatting und Samplings zu bestätigen. So ist beispielsweise die Anzahl an Lücken für ein Sampling mit einem oder vier Nachbarn immer gleich (1N und 4N). Dies liegt

Sampling	Army	Backyard	Basketball	Dumptruck	Evergreen	Grove	Mequon	\emptyset
1N-Uniform								
1N-Uniform	127	1232	1361	64	0	6506	526	1486.1
1N-Gewicht	127	1232	1361	64	0	6506	526	1486.2
1N-Austausch	127	1233	1361	64	0	6506	526	1486.3
4N-Uniform	127	1230	1361	64	0	6505	525	1486
4N-Gewicht	127	1231	1362	64	0	6504	524	1486.1
4N-Austausch	127	1232	1361	64	0	6506	526	1486.2
4N-Uniform								
1N-Uniform	127	1232	1361	64	0	6506	526	1486.1
1N-Gewicht	127	1232	1361	64	0	6506	526	1486.2
1N-Austausch	127	1233	1361	64	0	6506	526	1486.3
4N-Uniform	127	1230	1361	64	0	6505	525	1486
4N-Gewicht	127	1231	1362	64	0	6504	524	1486.1
4N-Austausch	127	1232	1361	64	0	6506	526	1486.2
4N-Gewicht								
1N-Uniform	127	1232	1361	64	0	6506	526	1486.1
1N-Gewicht	127	1232	1361	64	0	6506	526	1486.2
1N-Austausch	127	1233	1361	64	0	6506	526	1486.3
4N-Uniform	127	1230	1361	64	0	6505	525	1486
4N-Gewicht	127	1231	1362	64	0	6504	524	1486.1
4N-Austausch	127	1232	1361	64	0	6506	526	1486.2

TABELLE 5.4: Die zugehörige Anzahlen der Lücken für die in Tabelle 5.3 geprüften Varianten des Backward Warpings. In der ersten Spalte ist die Samplingstrategie optisch hervorgehoben (**fett**). Darunter befinden sich jeweils die sechs verschiedenen Splattingstrategien für das involvierte Warping des Flusses $\vec{v}_{0,f}$.

daran, dass bei einem 4N-Sampling immer von so vielen Stellen wie möglich versucht wird zu sampeln. Im schlechtesten Fall ist dies nur eine einzige Stelle, was wieder dem 1N-Fall entspricht. Sollte das 1N-Sampling nicht erfolgreich sein, so ist demnach auch sicher das 4N-Sampling nicht erfolgreich. Was Tabelle 5.4 ebenfalls zeigt, ist die Bedeutung des Splatting vom optischen Fluss: Nicht nur die Qualität der gesetzten Pixel ändert sich, sondern auch die Anzahl an leeren Stellen kann beeinflusst werden. Die Unterschiede waren zwar sehr gering, aber sie konnten beobachtet werden.

Zusammengefasst lässt sich für die verschiedenen Kombinationen von Sampling- und Splattingstrategien für das Backward Warping folgendes Fazit ziehen: Im Allgemeinen hat sich die bilineare Interpolation als das Sampling mit den besten Ergebnissen (4N-Gewicht) durchgesetzt, wobei das involvierte Splatting im Vergleich zum Sampling nicht ganz so stark ausschlaggebend war. Aufgrund der konsistenten und eindeutigen Ergebnisse wird in den kommenden Verfahren beim Sampling die bilineare Variante direkt verwendet.

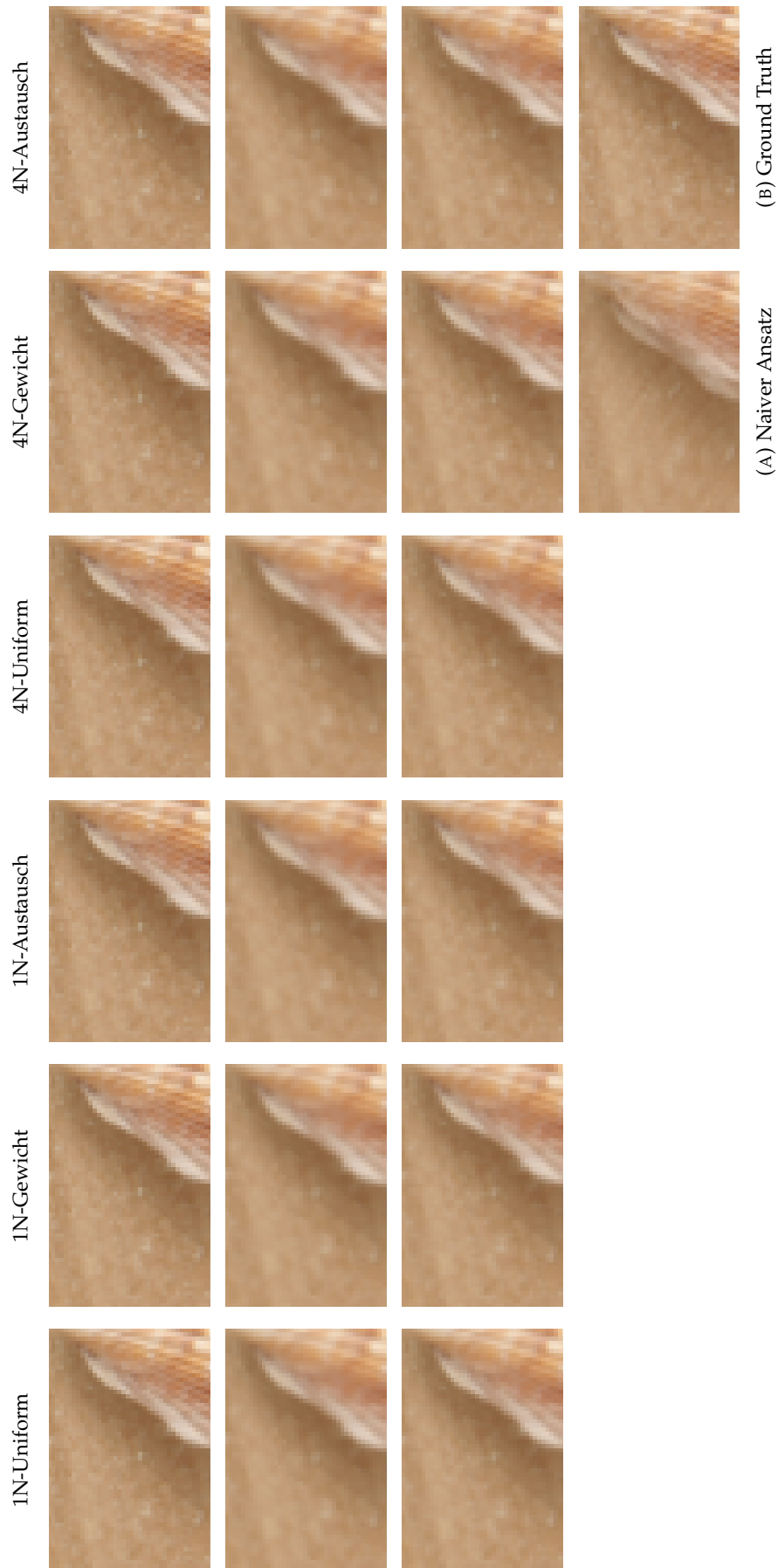


ABBILDUNG 5.5: Ein optischer Vergleich der verschiedenen Kombinationen von Splatting und Sampling beim Backward Warping. Beginnend bei der ersten Zeile sind 1N-Uniform, 4N-Uniform und 4N-Gewicht als Sampling zu sehen. Die Spalten stehen für die involvierte Splattingstrategie des optischen Flusses $\vec{v}_{0,f}$.

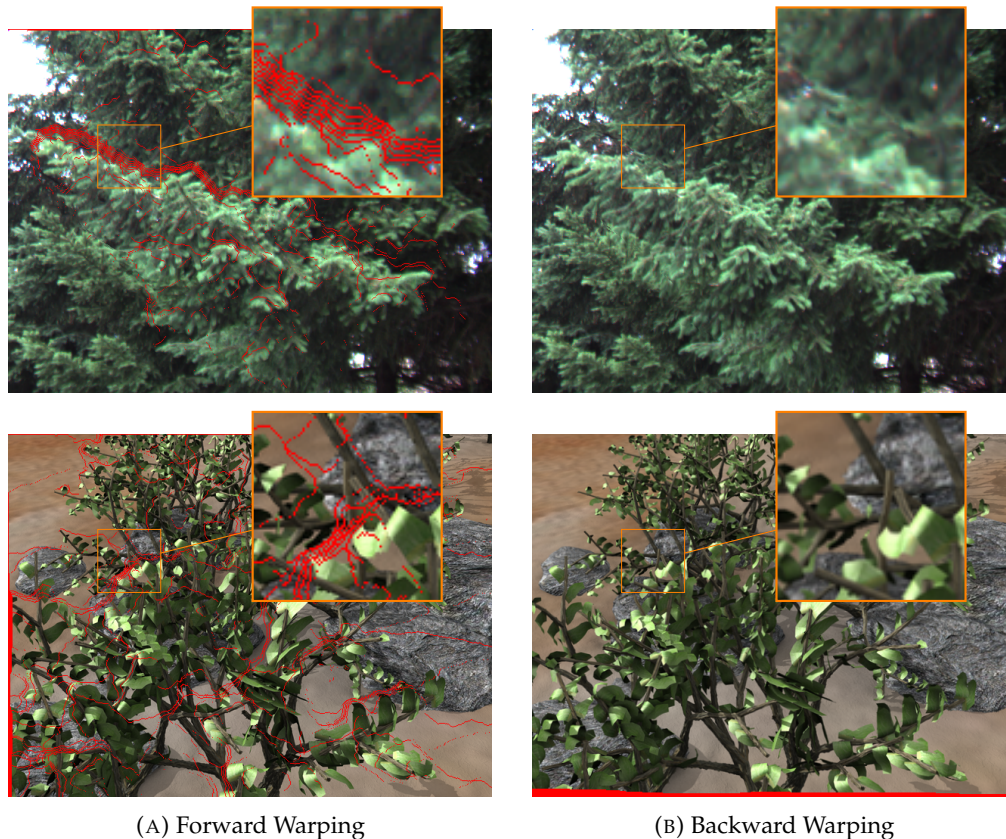


ABBILDUNG 5.6: Es sind die typischen Regionen mit Lücken (rot) zu sehen (Szenen evergreen und grove). Links beim Forward Warping (Nearest Neighbour Splatting) treten sie charakteristisch im Zentrum des Bildes auf teilweise auch an den Rändern. Rechts beim Backward Warping dagegen gibt es, wenn überhaupt, größere Lücken an den Rändern.

5.4.3 Vergleich und Probleme

In den letzten zwei Abschnitten wurden die Splatting- und Samplingstrategien vom Forward Warping und vom Backward Warping getrennt untersucht. Was abschließend noch von Interesse ist, ist ein direkter Vergleich der bisherigen Ergebnisse. Tabelle 5.5 zeigt gebündelt für die besten Varianten beider Warping Arten nochmal die Fehler und Lückenanzahl. Dort zu sehen sind nochmal die Ergebnisse vom Forward Warping mit einem 4N-Gewicht Splatting und einem Backward Warping mit einem 4N-Gewicht Splatting und Sampling. Zu beachten ist, dass ein direkter Vergleich der bisherigen Ergebnisse eigentlich etwas schwierig ist, da beim Forward Warping mit Bildinformation aus I_0 und beim Backward Warping mit Bildinformation aus I_1 gearbeitet wird. Trotzdem bietet sich ein direkter Vergleich in Bezug auf eine feste Richtung des optischen Flusses an (hier ein Vorwärtsfluss \vec{v}_0), um einen grundlegenden Eindruck zu erhalten.

Bei Betrachtung von Tabelle 5.5 ist zu erkennen, dass im Durchschnitt das Forward Warping besser abschneidet. Sowohl der durchschnittliche Fehler ist geringer als auch die mittlere Anzahl an Lücken. Was im Zusammenhang mit den Lücken neben der reinen Anzahl noch interessant ist, sind die Positionen der Lücken. Im Vergleich von Forward und Backward Warping ist zu beobachten, dass es beim Forward Warping neben Lücken an den Rändern häufiger zu Lücken im Zentrum des

Methode	Army	Backyard	Basketball	Dumptruck	Evergreen	Grove	Mequon	\emptyset
FW	6.477	21.474	22.063	29.508	22.548	35.068	16.124	20.746
# leere Stellen	125	946	288	0	365	1341	2600	1074.4
BW	7.110	23.481	22.049	28.217	22.846	39.146	20.025	21.806
# leere Stellen	127	1231	1362	64	0	6504	524	1486

TABELLE 5.5: Ein direkter Vergleich vom Forward Warping (FW) mit 4N-Gewicht Splatting aus Tabelle 5.2 und einem Backward Warping (BW) mit 4N-Gewicht Splatting und 4N-Gewicht Sampling aus Tabelle 5.3.

Bildes kommt (siehe Abbildung 5.6). Die unterschiedlichen Positionen der Lücken sind vermutlich auf die grundlegende Funktionsweise des Forward- und Backward Warpings zurückzuführen. Beim Forward Warping muss mithilfe eines guten Splatting eine möglichst hohe Ausfüllung der Pixel sichergestellt werden und es ist nicht sicher, dass jeder Pixel im Interpolanten $I_{0.5}$ einen Farbwert erhält. Dies ist für alle Stellen im Interpolanten der Fall, aber besonders auffällig ist es, wenn es Lücken im Zentrum des Bildes gibt. Beim Backward Warping ist dagegen die Wahrscheinlichkeit gerade im Zentrum des Bildes deutlich höher einen Farbwert zu besitzen. Dies liegt daran, dass die Verschiebung der Objekte meistens im Zentrum des Bildes bleibt und dadurch erfolgreich aus I_1 gesampelt werden kann.

Zusammengefasst ist aus dem direkten Vergleich zu schließen, dass unter Nutzung eines Vorwärtsflusses $\vec{v}_{0,f}$ das Forward Warping etwas besser als das Backward Warping zu sein scheint, was die Fehler und die Lückenanzahl angeht. Allerdings sind bei Forward Warping die Positionen der Lücken auch öfter im Zentrum des Bildes, was möglicherweise etwas ungünstig für das noch folgende Inpainting sein könnte.

5.5 Inpainting

Mit einem Splatting und Sampling beim Forward Warping und Backward Warping kann in Verbindung mit Inpainting ein vollständiger Interpolant erzeugt werden. In diesem Abschnitt wird daher das vorgestellte Inpainting aus dem letzten Kapitel ?? genauer untersucht werden. Zunächst aber wird im nächsten Abschnitt 5.5.1 das Verhalten der unterschiedlichen Lösungsverfahren untersucht. Dabei sollen die erhofften Beschleunigungen durch ein Gauß-Seidel-Verfahren und SOR verifiziert werden. Im Anschluss wird das Inpainting in Abschnitt 5.5.2 in Verbindung mit einem Forward Warping und Backward Warping getestet, um abschließend die Bedeutung der Lückenanzahl für das Inpainting zu klären. Dies bringt auch die Möglichkeit mit weitere Fehlermaße neben dem RMSE zu verwenden, da nach dem Inpainting alle Strukturen im Interpolanten vorliegen. Zum Schluss wird in Abschnitt 5.5.3 noch genauer auf beobachtete Probleme des verwendeten Inpaintings eingegangen.

5.5.1 Geschwindigkeitsvergleich

In diesem Abschnitt soll anhand eines künstlichen Beispiels das Konvergenzverhalten des iterativen Inpaintings untersucht werden. Dazu werden der RMSE und der SSIM Index als Fehlermaße verwendet, um die schrittweise Konvergenz zu beobachten. Bei diesem Test sind dieses Mal alle Stellen bei der Fehlerberechnung eingeschlossen. Getestet wird konkret ein Gauß-Seidel-Verfahren für das Inpainting und

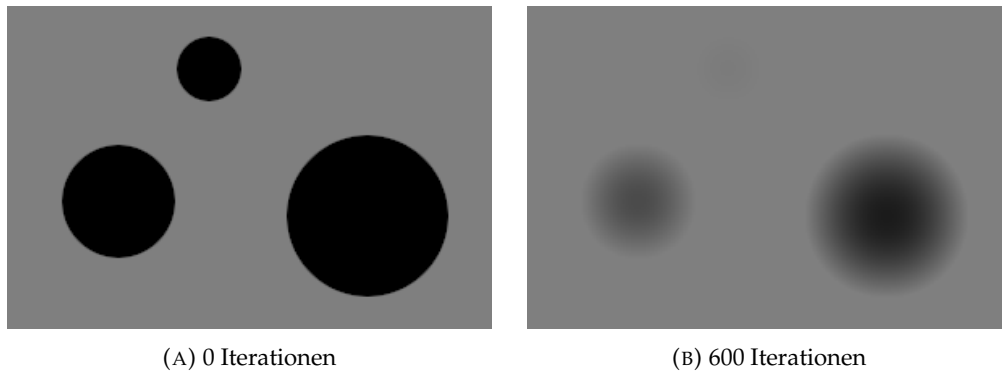


ABBILDUNG 5.7: Das künstliche Beispiel mit kreisförmigen Lücken im Bild (Durchmesser 100/70/40 Pixel). Rechts ist das Ergebnis von 600 Iterationen der Gauß-Seidel-Variante zu sehen. Kleine Lücken sind nahezu verschwunden und der RMSE ist deutlich gesunken (ca. 58.44 auf 21.08). Allerdings hat sich der SSIM gleichzeitig nur wenig verbessert (ca. 0.68 auf 0.70).

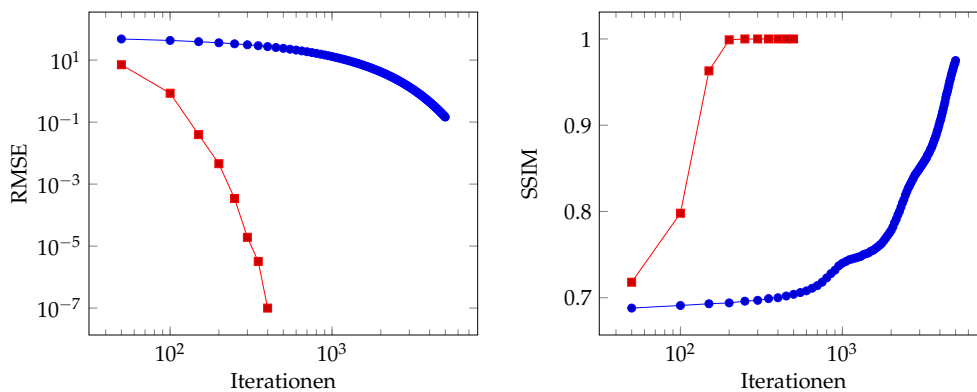


ABBILDUNG 5.8: Die abnehmenden Fehler (RSME und SSIM) beim Inpainting des künstlichen Beispiels aus Abbildung 5.7. Es sind die Ergebnisse eines Gauß-Seidel-Verfahrens (blau) und eines Gauß-Seidel-Verfahrens mit SOR (rot) dargestellt. Die SOR-Variante konvergiert deutlich schneller für beide Fehlermaße.

ein Gauß-Seidel-Verfahren, das mit SOR beschleunigt wird. Abbildung 5.7 zeigt die künstlich erzeugte Szene, mit der gearbeitet wird. Diese Szene besitzt unterschiedlich große kreisförmige Lücken, was dazu führt, dass zu dem Zeitpunkt, in dem die größte Lücke über das Inpainting gefüllt wurde, auf jeden Fall auch die kleineren Lücken nicht mehr vorhanden sind. Dieses Verhalten entspricht genau dem üblichen Szenario bei richtigen Szenen und tatsächlichen Interpolanten, bei dem auch unterschiedlich große Lücken gefüllt werden müssen. Gefüllt sein ist aber nicht gleichbedeutend mit einem konvergierten Ergebnis. Es zeigt sich, dass der Gauß-Seidel-Löser selbst nach 5000 Iterationen immer noch keine schwarzen Kreise im Bild zu erkennen gibt (ähnlich wie auf der rechten Seite in Abbildung 5.8). Auch der RMSE ist noch nicht bei Null angekommen (linke Seite in Abbildung 5.8). Die beschleunigte SOR-Variante liefert dagegen schon nach 250 Iterationen ein konvergiertes Bild ohne schwarze Kreise und mit einem SSIM von Eins (siehe rote Verläufe in Abbildung 5.8). Die Ergebnisse des Inpaintings beim künstlichen Beispiel sind auch bedeutend für die Bewertung der verwendeten Fehlermaße. Während mehrere Hundert Iterationsschritte eines Verfahrens den RMSE deutlich senken, hat sich die strukturelle Ähnlichkeit teilweise kaum verbessert (siehe Abbildung 5.7). Dort sind nach 600 Iterationen des Gauß-Seidel-Verfahrens

immer noch gut Kreise zu erkennen, wodurch es kaum eine Verbesserung der strukturellen Ähnlichkeit gibt. Der RMSE ist aber gleichzeitig mehr als halbiert worden, wodurch klar wird, dass die Bewertung durch den RMSE nur bedingt gut mit der Bewertung der menschlichen Wahrnehmung übereinstimmt. Da ein möglichst schnell konvergierendes Verfahren gesucht war und der durchgeführte Test an dem künstlichen Beispiel die Beschleunigung durch SOR bestätigt hat, wird ab sofort direkt mit dem Gauß-Seidel-Verfahren inklusive SOR weitergearbeitet.

5.5.2 Anwendung

In diesem Abschnitt soll das Inpainting bei richtigen Szenen angewendet werden (Middlebury Benchmark). Dazu wird analog wie im vorherigen Experiment vorgegangen (vgl. Abschnitt 5.4). Der Unterschied ist, dass zusätzlich die Lücken im Interpolanten $I_{0.5}$ zum Schluss noch über das Inpainting gefüllt werden (vgl. Inpainting aus Kapitel 3.5), wodurch in diesem Abschnitt das Forward Warping und das Backward Warping als vollständiges Verfahren getestet werden. Wie bereits angekündigt, wird beim Backward Warping nur noch das 4N-Gewicht Sampling betrachtet. Dadurch ergeben sich insgesamt sechs Varianten des Forward Warpings und des Backward Warpings, die hier untersucht werden (sechs Splattingstrategien). Die Fehlerberechnung für einen vollständigen Interpolanten $I_{0.5}$ erfolgt wie im letzten Abschnitt wieder unter Einschluss der Lücken und weil jetzt ein vollständiger Interpolant vorliegt, kommt auch der SSIM Index als zusätzliches Fehlermaß dazu. Für das Inpainting wird pauschal eine feste Iterationsanzahl von 500 verwendet. Tests an den Middleburyszenen haben gezeigt, dass diese Anzahl für ein Gauß-Seidel-Verfahren (inklusive SOR) ausreichend ist. Bei Betrachtung der Ergebnisse fällt in Tabelle 5.6 und 5.7 auf, dass es beim RMSE vor und nach dem Inpainting meist zu einer kleinen Verschlechterung kommt. Die Verschlechterung scheint dabei proportional zu der Anzahl an Lücken zu sein, was sich beispielsweise im Falle des

Splatting	Army		Backyard		Basketball		Dumprtruck		Evergreen		Grove		Mequon		∅	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
1N-Uniform (vorher)	8.220	0.879	22.201	0.727	22.863	0.480	30.497	0.848	23.697	0.757	38.779	0.652	17.382	0.752	22.074	0.689
	8.169	—	22.110	—	22.859	—	30.404	—	23.247	—	38.882	—	17.138	—	21.978	—
1N-Gewicht (vorher)	8.224	0.879	22.201	0.727	22.860	0.480	30.493	0.848	23.698	0.757	38.777	0.652	17.402	0.752	22.077	0.689
	8.173	—	22.109	—	22.856	—	30.400	—	23.248	—	38.880	—	17.159	—	21.981	—
1N-Austausch (vorher)	8.284	0.878	22.260	0.726	22.903	0.480	30.509	0.848	23.798	0.756	38.853	0.652	17.622	0.750	22.152	0.686
	8.234	—	22.169	—	22.900	—	30.416	—	23.353	—	38.959	—	17.388	—	22.057	—
# leere Stellen	2293		5144		5661		707		6941		12347		7891		6142.5	
4N-Uniform (vorher)	9.929	0.824	24.713	0.636	22.399	0.471	35.861	0.717	24.027	0.728	36.150	0.650	16.652	0.765	22.544	0.655
	9.930	—	24.738	—	22.397	—	35.861	—	24.007	—	36.081	—	16.583	—	22.509	—
4N-Gewicht (vorher)	6.479	0.906	21.454	0.734	22.065	0.495	29.508	0.852	22.571	0.779	35.143	0.694	16.194	0.802	20.785	0.716
	6.477	—	21.474	—	22.063	—	29.508	—	22.548	—	35.068	—	16.124	—	20.746	—
4N-Austausch (vorher)	8.311	0.878	22.313	0.726	22.954	0.479	30.524	0.848	23.886	0.755	39.032	0.651	17.680	0.749	22.208	0.688
	8.311	—	22.334	—	22.954	—	30.524	—	23.865	—	38.983	—	17.634	—	22.178	—
# leere Stellen	125		946		288		0		365		1341		2600		1074.4	

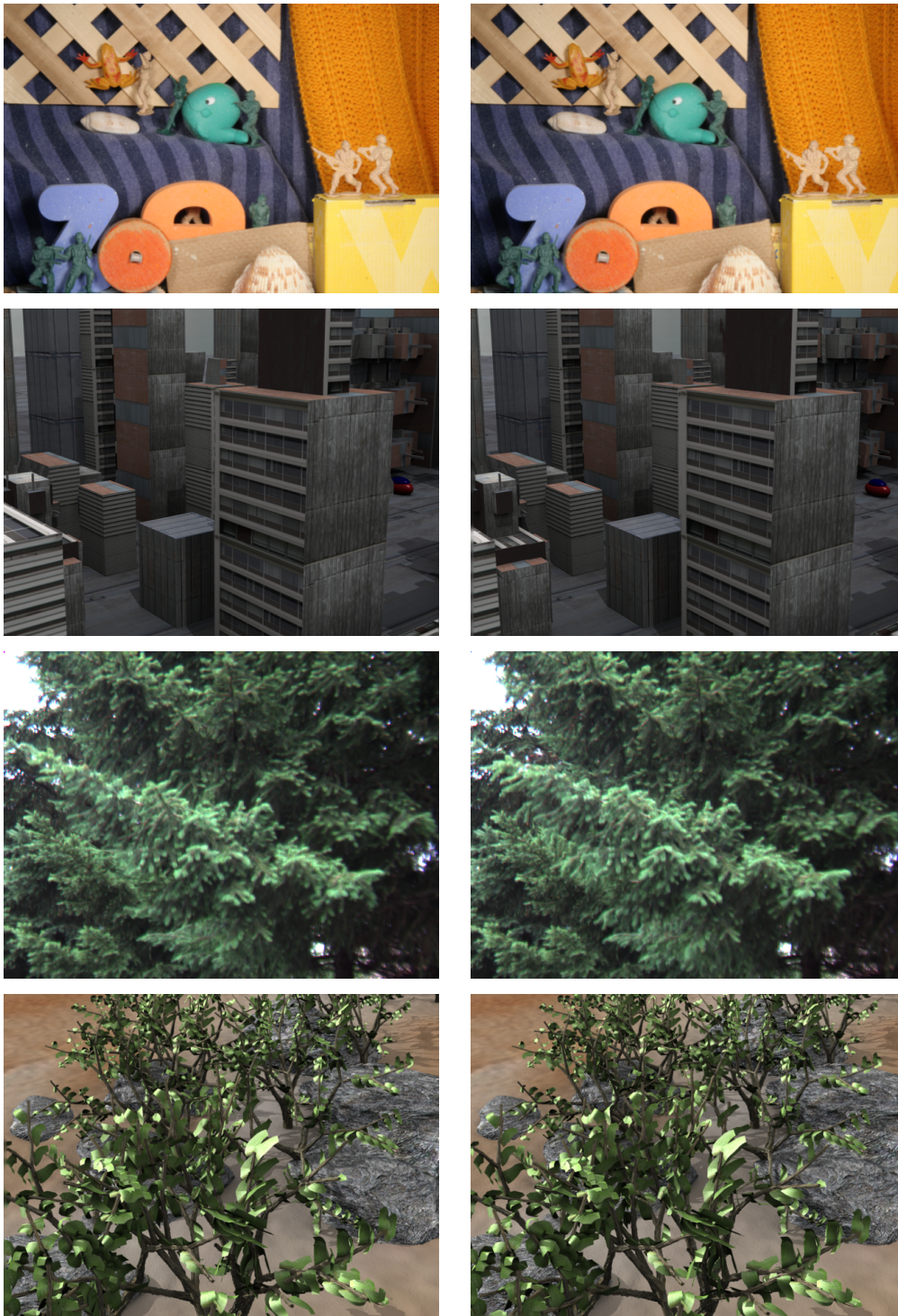
TABELLE 5.6: Übersicht der verschiedenen Splattingstrategien beim vollständigen Forward Warping mit Vorwärtsfluss $\vec{v}_{0,f}$. Es ist zusätzlich der Fehler vor dem Inpainting unter jeder Strategie angegeben (Fehler dort ignoriert Lücken). Die angegebenen Anzahlen von leeren Stellen beziehen sich auf den Stand vor dem Inpainting. Der beste Wert pro Spalte bezieht sich immer nur auf den Zustand nach dem Inpainting (vollständiges Forward Warping).

Splatting	Army		Backyard		Basketball		Dumprtruck		Evergreen		Grove		Mequon		\emptyset	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
1N-Uniform	7.103	0.904	23.563	0.733	22.051	0.489	28.228	0.854	22.855	0.780	39.529	0.658	19.965	0.789	21.929	0.707
(vorher)	7.102	—	23.499	—	22.052	—	28.216	—	22.855	—	39.139	—	19.987	—	21.807	—
# leere Stellen	127		1232		1361		64		0		6506		526		1486.1	
1N-Gewicht	7.102	0.904	23.567	0.733	22.052	0.489	28.228	0.854	22.855	0.780	39.529	0.658	19.962	0.789	21.928	0.707
(vorher)	7.101	—	23.503	—	22.054	—	28.216	—	22.855	—	39.139	—	19.984	—	21.807	—
# leere Stellen	127		1232		1361		64		0		6506		526		1486.2	
1N-Austausch	7.106	0.904	23.629	0.733	22.057	0.489	28.230	0.854	22.860	0.780	39.531	0.658	19.971	0.789	21.938	0.707
(vorher)	7.104	—	23.565	—	22.059	—	28.218	—	22.860	—	39.142	—	19.993	—	21.816	—
# leere Stellen	127		1233		1361		64		0		6506		526		1486.3	
4N-Uniform	7.136	0.904	23.566	0.733	22.054	0.489	28.229	0.854	22.853	0.780	39.566	0.657	20.036	0.789	21.939	0.707
(vorher)	7.135	—	23.502	—	22.056	—	28.217	—	22.853	—	39.178	—	20.058	—	21.818	—
# leere Stellen	127		1230		1361		64		0		6505		525		1486	
4N-Gewicht	7.112	0.904	23.545	0.733	22.047	0.490	28.229	0.854	22.846	0.780	39.535	0.658	20.003	0.789	21.928	0.707
(vorher)	7.110	—	23.481	—	22.049	—	28.217	—	22.846	—	39.146	—	20.025	—	21.806	—
# leere Stellen	127		1231		1361		64		0		6504		524		1486.1	
4N-Austausch	7.109	0.904	23.610	0.733	22.058	0.489	28.230	0.854	22.864	0.780	39.533	0.658	19.975	0.789	21.931	0.707
(vorher)	7.108	—	23.546	—	22.060	—	28.218	—	22.864	—	39.144	—	19.997	—	21.817	—
# leere Stellen	127		1232		1361		64		0		6506		526		1486.2	

TABELLE 5.7: Übersicht der verschiedenen Splattingstrategien beim vollständigen Backward Warping mit Vorwärtsfluss $\vec{v}_{0,f}$. Es wurde immer ein 4N-Gewicht Sampling verwendet (bilineare Interpolation). Es ist zusätzlich der Fehler vor dem Inpainting unter jeder Strategie angegeben (Fehler dort ignoriert Lücken). Der beste Wert pro Spalte bezieht sich immer nur auf das vollständige Backward Warping.

Forward Warpings an den stärkeren Verschlechterungen beim 1N-Splatting mit vielen Lücken und dem 4N-Splatting mit weniger Lücken gut zeigt (vgl. Tabelle 5.6). In diesem Zusammenhang ist auch zu sehen, dass die Verhältnisse von den verschiedenen Splatting- und Samplingstrategien sich zueinander nicht verändert haben und sich erneut beispielsweise ein 4N-Gewicht Splatting beim Forward Warping relativ eindeutig durchgesetzt hat. Beim SSIM Index ist zu sehen, dass es ähnlich wie beim künstlichen Beispiel zu Fällen kommt, wo der SSIM Index im Vergleich zum RMSE deutlich schlechter ist. Dies zeigt sich bei den Szenen Backyard und Basketball (vgl. Tabelle 5.6 und 5.7). Bei diesen Szenen ist zwar ein ähnlicher RMSE zu beobachten, aber es gibt starke Unterschiede beim SSIM Index. Betrachtet man den durchschnittlichen SSIM Index, liegen das Forward Warping und das Backward Warping relativ nah beieinander. Insgesamt schneidet aber die beste Variante des Forward Warping für beide Fehlermaße wieder etwas besser als das Backward Warping ab (in Bezug auf den geprüften Vorwärtsfluss $\vec{v}_{0,f}$).

Wie schon vermutet, hat die geringe Anzahl an Lücken nur einen relativ geringen Einfluss auf die Gesamtqualität der Bildqualität, was vermutlich an der einfach geringen Lückenzahl liegt. Der stärker steigende RMSE bei einer höheren Lückenzahl ist auch nachvollziehbar, da wohl eine durch Inpainting gefüllte Stelle einer durch das Warping besetzten Stelle generell unterlegen sein müsste. Dies ist insbesondere bei einer größeren Lückenzahl der Fall, wodurch dort auch die Fehler deutlicher ansteigen. Der SSIM Index liegt nach dem Inpainting im Durchschnitt bei etwa 0.7, was für den 1-Bild Fall bereits relativ nah an dem Maximalwert von Eins erscheint. Es muss sich noch zeigen, ob die kommenden Verfahren, die mit zwei Bildinformation und mehreren Flüssen arbeiten, diesen Wert noch verbessern können. Auf der nächsten Seite befindet sich noch eine kleine Auswahl an Interpolanten zur Veranschaulichung der Ergebnisse (siehe Abbildung 5.9).



(A) Forward Warping

(B) Backward Warping

ABBILDUNG 5.9: Eine Auswahl von vollständigen Zwischenbildern, die links mit einem Forward Warping (4N-Gewicht Splatting) und rechts mit einem Backward Warping (4N-Gewicht Sampling und 4N-Gewicht Splatting) erzeugt wurden. Szenen von oben nach unten: Army, Urban, Evergreen und Grove.

5.5.3 Probleme

Bei Betrachtung der Ergebnisse des Inpaintings bei den Szenen des Middlebury Benchmarks fällt ein grundlegendes Problem am Inpainting auf. Zwar kommt das vorgestellte Inpainting gut mit einzelnen leeren Stellen im Interpolanten klar, wie sie häufig beim Forward Warping entstehen, doch sind größere Lücken etwas problematisch. Größere Lücken zeigen sich besonders an den Rändern der Interpolanten und sind sowohl beim Forward Warping als auch beim Backward Warping vorhanden. Was beim Inpainting solcher Lücken geschieht, ist in Abbildung 5.10 zu sehen. Dort zu erkennen sind unscharfe Stellen, die bei aufmerksamer Betrachtung durch-

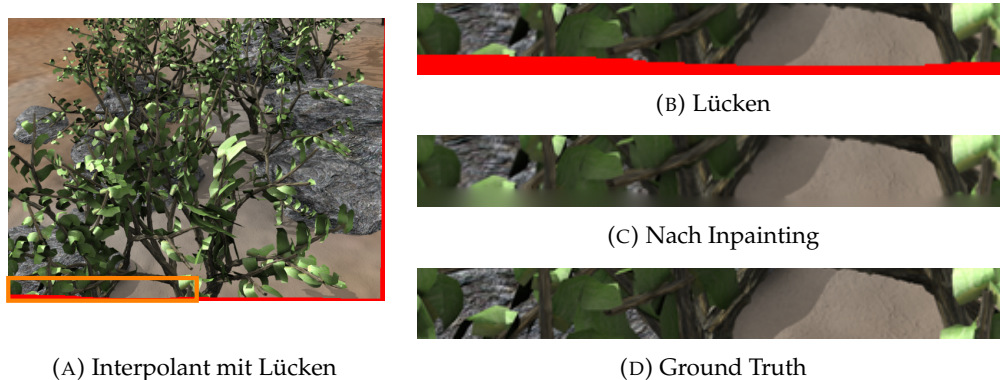


ABBILDUNG 5.10: Zu sehen ist ein unscharfer Bereich, der beim Inpainting von größeren Lücken (rot) entstehen kann (Interpolant entstanden durch Backward Warping mit bilinearem Sampling und 4N-Gewicht Splatting, Szene: Grove).

aus auffallen. Diese werden bei der ungerichteten Diffusion des Inpaintings verursacht, die lediglich eine Ähnlichkeit zu den Nachbarn der Lücke aufbauen will (vgl. Kapitel 3.5). Die Strukturen im Bild werden dabei komplett ignoriert und es scheint, dass die Farbwerte sich vom Rand aus zur Mitte der Lücke hin zunehmend vermischen. Der Prozess der ungerichteten Diffusion arbeitet unabhängig von den Übergängen und Kanten der Objekte im Bild (daher auch die Bezeichnung), weswegen eine Fortführung der Strukturen innerhalb der Lücke nicht möglich ist. Diese Schwäche des vorgestellten Inpaintings ist demnach aber nur ein Problem, wenn in der Lücke feine Strukturen vorhanden wären (vgl. Blätter in Abbildung 5.10). Bei größeren Lücken in einem gleichfarbigen Bereich gibt es dagegen ein relativ überzeugendes Angleichen an die Nachbarstruktur (vgl. Bodenbereich in Abbildung 5.10). Aus diesem Grund müsste das Inpainting theoretisch auch gut bei Interpolanten einer Bildsequenz eines Cartoons funktionieren (vgl. Kapitel 1.2), da dort üblicherweise ein Bild nur aus gröberen einfarbigen Flächen besteht. Dieser Fall wird in dieser Arbeit aber nicht weiter untersucht.

Zwar sind die unscharfen Bereiche im Bild durchaus auffällig, doch ist es fragwürdig, ob diese dem Betrachter bei einer Anzeigedauer in Bereichen von einer 30-tel Sekunde wirklich auffallen. Da das Inpainting mit einzelnen Lücken gut zurechtkommt und die Lücken trotz teilweiser vorhandener Unschärfe noch akzeptabel wirken, wird das Inpainting zusammenfassend als ausreichend angesehen.

5.6 2-Bild Fall

Mit dem Grundverfahren Forward Warping und Backward Warping lassen sich bereits richtige Zwischenbilder erzeugen. Auf diesen aufbauend wurde der 2-Bild-Fall im vorherigen Kapitel 4.4 eingeführt, der durch das Verwenden beider Nachbarbilder ein möglicherweise noch besseres photometrisches Ergebnis liefern könnte. In diesem Kapitel sollen diese aufbauenden Verfahren, symmetrisches Forward Warping und symmetrisches Backward Warping, genauer untersucht werden (vgl. Verfahren aus Kapitel 4.4.2 und 4.4.1). Dies geschieht wieder mit den Frames 10, 11 und 12 einer Middlebury Bildsequenz. Als Richtung für den Fluss wird, für einen guten Vergleich zum bisherigen Forward Warping und Backward Warping, wieder ein Vorwärtsfluss $\vec{v}_{0,f}$ von Frame 10 (I_0) nach Frame 12 (I_1) verwendet.

Wie schon bei den Experimenten zum Backward Warping angemerkt, wird bei den symmetrischen Verfahren bei den notwendigen Zwischenschritten ein 4N-Gewicht Sampling (bilineare Interpolation) verwendet (vgl. Abschnitt ??). Beim symmetrischen Backward Warping hat dies zur Folge, dass dort nur noch die verschiedenen Splattings beim Forward Warping des Flusses $\vec{v}_{0,f}$ getestet werden. Bei den vorherigen Backward Warping Experimenten aus Abschnitt 5.4.2 waren die Ergebnisse für das Fluss Warping nicht ganz eindeutig, weswegen hier eine erneute Untersuchung durchgeführt wird. Außerdem gibt es beim symmetrischen Backward Warping in gewisser Weise eine noch größere Abhängigkeit von dem Fluss Warping, da der entstandene Fluss $\vec{v}_{0,5}$ für zwei Samplings in zwei verschiedenen Richtungen verwendet wird. Insgesamt ergeben sich also sechs verschiedene Varianten des symmetrischen Backward Warpings, die in diesem Experiment getestet werden.

Beim symmetrischen Forward Warping gibt es bei sechs möglichen Splattings theoretisch 36 verschiedene Varianten des symmetrischen Forward Warpings (sechs Splattings für das Fluss Warping gekreuzt mit den sechs Splattings für das Warping der Bildinformation). In dieser Arbeit wird sich auf ein einheitliches Splatting für das Fluss Warping und das Warping von Bildinformationen beschränkt, um einen grundlegenden Eindruck vom symmetrischen Backward Warping zu vermitteln. Dadurch ergeben sich insgesamt sechs verschiedene Varianten des symmetrischen Forward Warpings, die in diesem Experiment getestet werden.

Methode	Army		Backyard		Basketball		Dumprtruck		Evergreen		Grove		Mequon		\emptyset	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
1N-Uniform	5.915	0.918	19.104	0.765	17.574	0.522	24.093	0.868	19.485	0.793	32.516	0.699	14.639	0.816	18.377	0.727
1N-Gewicht	5.918	0.918	19.099	0.766	17.575	0.522	24.093	0.868	19.487	0.793	32.515	0.699	14.742	0.816	18.387	0.727
1N-Austausch	5.918	0.918	19.172	0.765	17.586	0.521	24.097	0.868	19.498	0.794	32.526	0.699	15.074	0.816	18.441	0.727
4N-Uniform	7.369	0.865	20.902	0.691	16.943	0.505	28.174	0.769	19.678	0.775	32.179	0.670	14.562	0.790	18.928	0.691
4N-Gewicht	5.796	0.917	18.726	0.769	16.959	0.531	23.546	0.871	18.880	0.808	31.483	0.704	14.428	0.819	17.956	0.733
4N-Austausch	5.844	0.919	19.233	0.765	17.628	0.521	24.083	0.869	19.659	0.793	32.536	0.701	15.026	0.817	18.465	0.728
FW	6.479	0.906	21.454	0.734	22.065	0.495	29.508	0.852	22.571	0.779	35.143	0.694	16.194	0.802	20.785	0.716
BW	7.112	0.904	23.545	0.733	22.047	0.490	28.229	0.854	22.846	0.780	39.535	0.658	20.003	0.789	21.928	0.707
Naiver Ansatz	10.356	0.772	34.991	0.442	25.350	0.460	24.764	0.853	28.488	0.698	69.733	0.084	32.256	0.250	30.465	0.433

TABELLE 5.8: Übersicht der verschiedenen Splattingstrategien beim symmetrischen Forward Warping. Es wurde für das Bild Warping und für das involvierte Forward Warping des Flusses $\vec{v}_{0,f}$ das angegebene Splatting verwendet.

Methode	Army		Backyard		Basketball		Dumprtruck		Evergreen		Grove		Mequon		\emptyset	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
1N-Uniform	5.780	0.917	18.870	0.769	16.967	0.531	23.555	0.871	18.906	0.808	31.486	0.704	14.675	0.819	17.999	0.733
1N-Gewicht	5.780	0.917	18.869	0.769	16.968	0.531	23.555	0.871	18.905	0.808	31.485	0.704	14.687	0.819	17.997	0.733
1N-Austausch	5.783	0.917	18.882	0.769	16.970	0.531	23.557	0.871	18.915	0.808	31.487	0.704	14.772	0.819	18.009	0.733
4N-Uniform	5.787	0.917	18.861	0.769	16.966	0.531	23.553	0.871	18.893	0.808	31.477	0.705	14.663	0.819	17.990	0.733
4N-Gewicht	5.783	0.917	18.858	0.769	16.965	0.531	23.553	0.871	18.895	0.808	31.478	0.705	14.678	0.819	17.992	0.733
4N-Austausch	5.783	0.917	18.876	0.769	16.973	0.531	23.557	0.871	18.918	0.808	31.488	0.704	14.782	0.819	18.011	0.733
FW	6.479	0.906	21.454	0.734	22.065	0.495	29.508	0.852	22.571	0.779	35.143	0.694	16.194	0.802	20.785	0.716
BW	7.112	0.904	23.545	0.733	22.047	0.490	28.229	0.854	22.846	0.780	39.535	0.658	20.003	0.789	21.928	0.707
Naiver Ansatz	10.356	0.772	34.991	0.442	25.350	0.460	24.764	0.853	28.488	0.698	69.733	0.084	32.256	0.250	30.465	0.433

TABELLE 5.9: Übersicht der verschiedenen Splattingstrategien beim symmetrischen Backward Warping. Die Splattingstrategien beziehen sich auf das involvierte Forward Warping des Flusses $\vec{v}_{0,f}$. Es wurde für das Sampling aus den Bildern immer ein 4N-Gewicht Sampling verwendet (bilineare Interpolation).

5.6.1 Vergleich

In diesem Abschnitt werden die verschiedenen Varianten des symmetrischen Forward Warpings und des symmetrischen Backward Warpings hinsichtlich der bisherigen Fehlermaße (RMSE und SSIM) miteinander verglichen. Da die symmetrischen Verfahren in gewisser Weise als Erweiterung vom 1-Bild Fall gesehen werden können, werden zusätzlich die Ergebnisse von dem vollständigen Forward Warping (FW) und Backward Warping (BW) als Vergleich erneut aufgeführt (vgl. Abschnitt 5.5.2). Weil die symmetrischen Verfahren außerdem mit zwei Bildinformationen arbeiten, werden in diesem Experiment auch erstmals die Ergebnisse vom naiven Ansatz zum Vergleich herangezogen (vgl. Abschnitt 4.1).

Tabelle 5.9 und 5.8 zeigen die Ergebnisse der Interpolation der Middlebury Szenen. Dort abzulesen ist, dass die symmetrischen Verfahren eine deutliche Verbesserung zum 1-Bild Fall darstellen, denn sowohl der RMSE als auch der SSIM Index konnten verbessert werden. Ein Vergleich zum naiven Ansatz zeigt auch, dass die symmetrischen Warping Verfahren deutlich besser abschneiden. Im Schnitt scheint ein knapp halb so großer RMSE vorhanden zu sein und der SSIM Index ist in jedem Fall auch besser. Was bei Betrachtung der unterschiedlichen Splatting- und Samplingstrategien noch auffällt, sind die Verhältnisse zueinander, die sehr dem 1-Bild Fall ähneln. In Tabelle 5.8 beim symmetrischen Forward Warping setzt sich erneut relativ eindeutig das 4N-Gewicht Splatting durch. Beim symmetrischen Backward Warping in Tabelle 5.9 hat sich dagegen dieses Mal das 4N-Uniform Splatting durchgesetzt.

Wegen der deutlich besseren Fehler scheint der erhoffte Effekt eines besseren photometrischen Ergebnisses gegenüber dem 1-Bild Fall bestätigt worden zu sein. Die Ergebnisse ähneln von den Verhältnissen sehr dem 1-Bild Fall, was zu erwarten war, da die symmetrischen Verfahren eine Art Erweiterung darstellen. Überrascht hat das gute Abschneiden des 4N-Uniform Splatting beim symmetrischen Backward Warping (vgl. Tabelle 5.9), wo eigentlich wieder ein 4N-Gewicht Splatting als Sieger zu vermuten war. Bei den Tests in Abschnitt 5.3 waren diese beiden Splatting bereits nah beieinander, doch dieses Mal war scheinbar 4N-Uniform leicht vorne. Dies ist möglicherweise auf die geringe Szenenauswahl oder Flussrichtung zurückzuführen. Zusammengefasst hat sich bei einem Vorwärtsfluss $\vec{v}_{0,f}$ beim symmetrischen Forward Warping ein 4N-Gewicht Splatting durchgesetzt. Beim symmetrischen Backward Warping war es ein 4N-Gewicht Sampling und ein 4N-Uniform Splatting.



(A) Symmetrisches Forward Warping

(B) Symmetrisches Backward Warping

ABBILDUNG 5.11: Eine Auswahl von erzeugten Zwischenbildern, die links mit einem symmetrischen Forward Warping (4N-Gewicht Splatting) und rechts mit einem symmetrischen Backward Warping (4N-Gewicht Sampling und 4N-Uniform Splatting) erzeugt wurden. Es wurde immer der Vorwärtsfluss $\vec{v}_{0,f}$ verwendet. Szenen von oben nach unten: Army, Grove, Mequon und Backyard.

5.6.2 Probleme

Zwar erzeugen die symmetrischen Warpingverfahren im Vergleich zum 1-Bild Fall bessere Zwischenbilder was die Fehlermaße angeht, allerdings tun sie das mit der Einführung von Doppelungsartefakten (siehe Interpolanten in Abbildung 5.11 auf der vorherigen Seite). Im Fall der Szene Backyard in Abbildung 5.12 ist gut zu erkennen, dass es ähnlich zum naiven Ansatz zu auffälligen Doppelungsartefakten im Bild kommt (zum Beispiel die Beine des Kindes). Im Fall der gezeigten Szene sind sogar ganze Objekten doppelt im Bild zu erkennen (Ball links im Bild). Allerdings lässt sich feststellen, dass das Ausmaß an Doppelungsartefakten bei den symmetrischen Verfahren sichtbar geringer ist als beim naiven Ansatz. So fällt beispielsweise

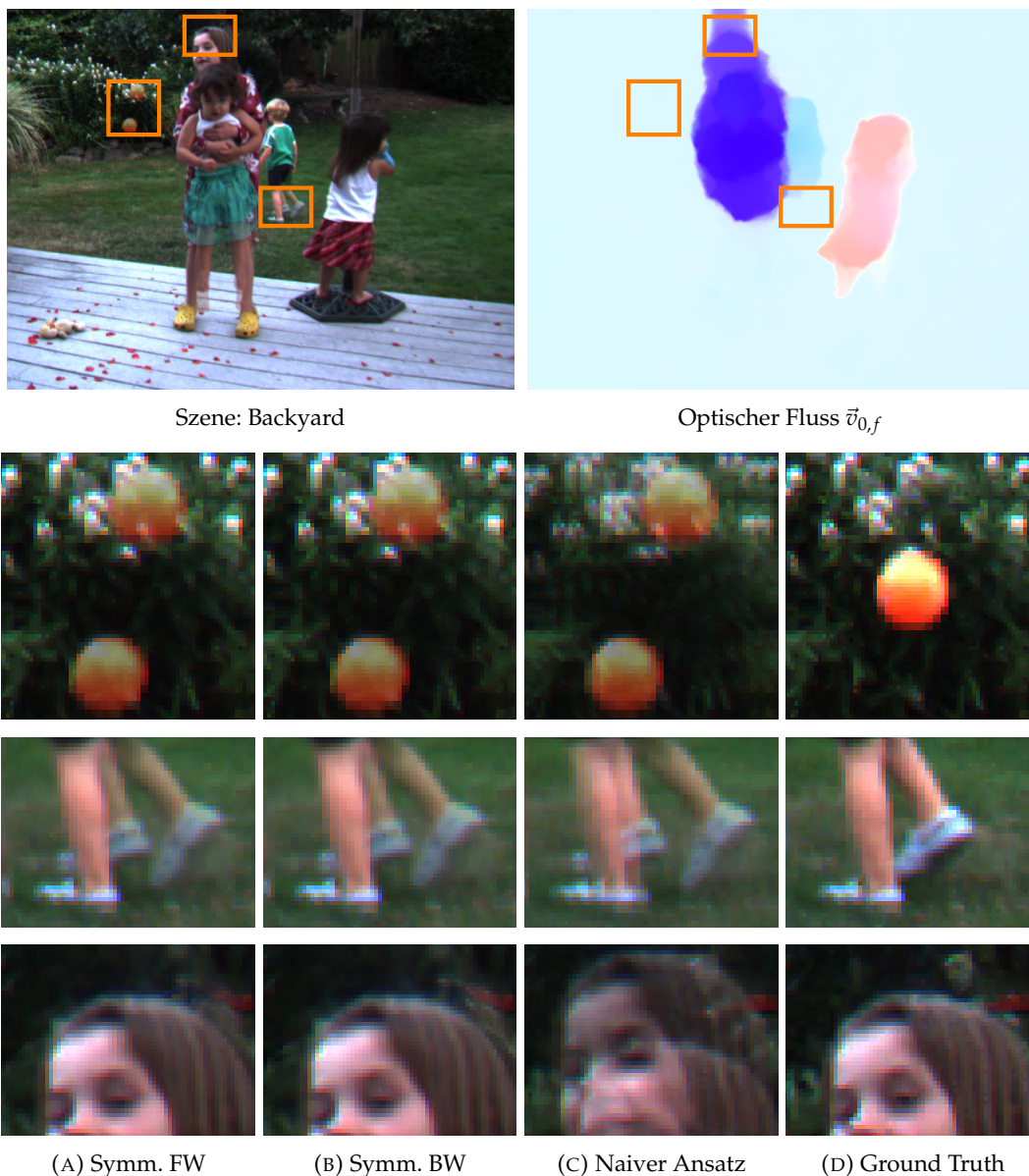


ABBILDUNG 5.12: Ein optischer Vergleich von den symmetrischen Warpingverfahren und dem naiven Ansatz. Zu erkennen ist, dass der Ball bei den Interpolanten der Szene Backyard doppelt vorhanden ist. Die Beine des Kindes wirken bei den symmetrischen Verfahren etwas besser und beim Kopf des Kindes ist im Vergleich gar keine Doppelung vorhanden.

an den Beinen des Kindes auf, dass zumindest das linke Bein korrekter wirkt im Vergleich zum naiven Ansatz. Außerdem ist beim doppelt vorhandenen Ball zumindest der Hintergrund scharf und kommt dem Ground Truth näher. An manchen Stellen kommt es sogar zu keinen Doppelungsartefakten bei den symmetrischen Verfahren (Kopf des Kindes). Dagegen sind beim naiven Ansatz an dieser Stelle erneut starke Doppelungen zu erkennen.

Die Beobachtungen dieses Beispiels zeigen gut die starke Abhängigkeit der symmetrischen Verfahren von der Qualität des optischen Flusses. An den Stellen, wo der Fluss die Bewegung gut erkannt hat, kam es im Vergleich zum naiven Ansatz zu keinen Doppelungsartefakten. Im schlechtesten Fall, wo der Fluss gar keine Bewegung einzelner Objekte erkennt (Ball), sind dagegen sehr auffällige Doppelungsartefakte zu beobachten (vgl. optischer Fluss in Abbildung 5.12). Abschließend ist die Bewertung der symmetrischen Verfahren zweigeteilt: einerseits konnte im Vergleich zum 1-Bild Fall (Forward Warping und Backward Warping) sowohl für den RMSE als auch für den SSIM Index insgesamt eine Verbesserung erlangt werden, auf der anderen Seite stören die Doppelungsartefakte das Gesamtbild der Interpolanten.

5.7 Bidirektionale Verfahren

In Abschnitt 4.5 wurde in Frage gestellt, welche Richtung des optischen Flusses überhaupt „die Richtige“ für die Zwischenbildinterpolation ist: Ist es der Vorwärtsfluss $\vec{v}_{0,f}$ oder der Rückwärtsfluss $\vec{v}_{1,b}$? Beide beschreiben die Bewegung zwischen zwei Bildern I_0 und I_1 und zumindest optisch können sie sich durchaus ähneln (siehe Abbildung 5.13). Welcher Fluss für die Interpolation besser geeignet ist, lässt sich allein über den visuellen Eindruck aber nicht ableiten. Im letzten Kapitel wurde erklärt, dass durch den zusätzlichen Aspekt der Orientierung der Bewegungstrajektorie eine weitere Reihe an Verfahren dazu kommt, deren Potential aber schwierig einzuschätzen ist. Besonders die Auswirkung des Verzichtens auf ein Fluss Warping und dem stattdessen durchgeführten Warping über einen anderen Fluss ist ohne konkrete Zahlen nur schwer zu beurteilen. Es soll daher in diesem Abschnitt auf diesen Aspekt genauer eingegangen werden. Für die Untersuchung der bidirektionalen Verfahren soll zunächst für den einfachen 1-Bild Fall geklärt werden, wie groß die Unterschiede bei Verwendung eines Vorwärtsflusses gegenüber dem Rückwärtsfluss tatsächlich sind. Danach soll in einem letzten Experiment noch geklärt werden, ob die neu dazugekommenen Verfahren (Forward Forward Warping

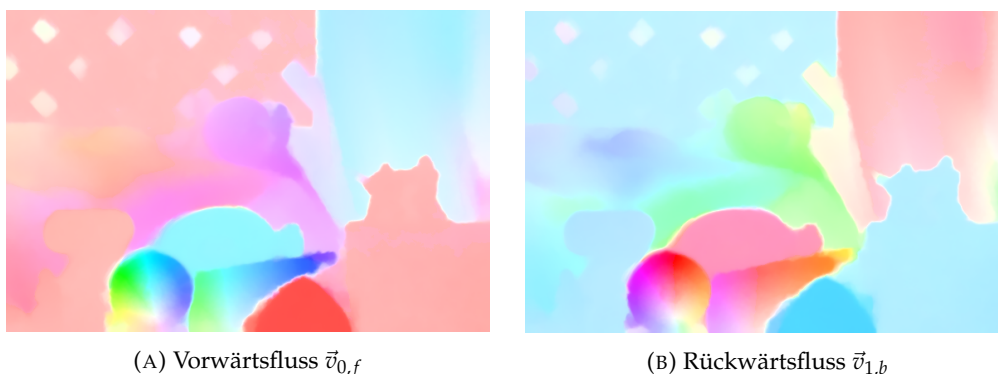


ABBILDUNG 5.13: Links ist der bisherige Vorwärtsfluss $\vec{v}_{0,f}$ (Frame 10 nach 12) zu sehen. Rechts der Rückwärtsfluss $\vec{v}_{1,b}$ (Frame 12 nach 10). Szene: Army

und Backward Backward Warping) wegen ihrer zusätzlichen Motion Estimation den symmetrischen Verfahren überlegen sind. Da die bidirektionalen Varianten der symmetrischen Verfahren ebenfalls versuchen eine Verbesserung an den symmetrischen Verfahren vorzunehmen, werden diese in den Vergleich integriert. Bei diesem abschließenden Experiment soll vor allem klar werden, ob es eine quantitative Verbesserung (Fehler) gibt, aber auch ob eine qualitative Verbesserung in Form einer Reduzierung von Doppelungsartefakten zu beobachten ist.

5.7.1 Vorwärtsfluss vs. Rückwärtsfluss

Zunächst soll anhand des 1-Bild Falls die allgemeine Auswirkung der Richtung eines optischen Flusses bei der Zwischenbildinterpolation beobachtet werden. In den bisherigen Experimenten wurde immer mit einem Vorwärtsfluss $\vec{v}_{0,f}$ gearbeitet, was bis zu diesem Zeitpunkt legitim war, da es sich bei den bisherigen Verfahren ausschließlich um direktionale Verfahren gehandelt hat. Dort wurde ein einheitlicher Vergleich anhand einer einzelnen Richtung für eine Gegenüberstellung als ausreichend angesehen. In diesem Experiment wird jetzt neben dem bisher oft verwendeten Vorwärtsfluss $\vec{v}_{0,f}$ noch der Rückwärtsfluss $\vec{v}_{1,b}$ hinzugezogen. Diese Flüsse sollen verwendet werden, um ein vollständiges Forward Warping gemäß den Definitionen aus Abschnitt 4.3.1 für beide möglichen Flussrichtungen durchzuführen und um erneut alle sechs Splattingstrategien miteinander zu vergleichen (beinhaltet Inpainting). Mit der Notation des letzten Kapitels findet jetzt also ein Forward Warping von I_0 über $\vec{v}_{0,f}$ und ein Forward Warping von I_1 über $\vec{v}_{1,b}$ statt. Dazu kommen erneut drei Frames eine Middleburyszene zum Einsatz (Frame 10, 11 und 12) und die Beurteilung findet über den RMSE und SSIM Index statt. In Tabelle 5.10 sind die Ergebnisse der zwei Forward Warpings aufgeführt.

Splatting	Army		Backyard		Basketball		Dumprtruck		Evergreen		Grove		Mequon		∅	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
1N-Uniform																
Vorwärts	8.220	0.879	22.201	0.727	22.863	0.480	30.497	0.848	23.697	0.757	38.779	0.652	17.382	0.752	22.074	0.689
Rückwärts	8.004	0.886	22.718	0.730	21.508	0.478	29.145	0.850	22.124	0.766	41.626	0.621	19.512	0.749	22.056	0.683
1N-Gewicht																
Vorwärts	8.224	0.879	22.201	0.727	22.860	0.480	30.493	0.848	23.698	0.757	38.777	0.652	17.402	0.752	22.077	0.689
Rückwärts	8.000	0.886	22.719	0.729	21.538	0.478	29.149	0.850	22.127	0.766	41.628	0.621	19.495	0.749	22.053	0.683
1N-Austausch																
Vorwärts	8.284	0.878	22.260	0.726	22.903	0.480	30.509	0.848	23.798	0.756	38.853	0.652	17.622	0.750	22.152	0.686
Rückwärts	8.032	0.885	22.776	0.729	21.690	0.477	29.180	0.850	22.219	0.766	41.824	0.620	19.512	0.748	22.112	0.682
4N-Uniform																
Vorwärts	9.929	0.824	24.713	0.636	22.399	0.471	35.861	0.717	24.027	0.728	36.150	0.650	16.652	0.765	22.544	0.655
Rückwärts	11.023	0.821	25.170	0.639	20.782	0.470	34.663	0.720	22.466	0.739	38.737	0.625	19.120	0.743	22.574	0.653
4N-Gewicht																
Vorwärts	6.479	0.906	21.454	0.734	22.065	0.495	29.508	0.852	22.571	0.779	35.143	0.694	16.194	0.802	20.785	0.716
Rückwärts	7.005	0.906	21.978	0.737	20.492	0.493	28.149	0.854	20.894	0.790	37.990	0.665	18.619	0.789	20.836	0.709
4N-Austausch																
Vorwärts	8.311	0.878	22.313	0.726	22.954	0.479	30.524	0.848	23.886	0.755	39.032	0.651	17.680	0.749	22.208	0.688
Rückwärts	8.086	0.885	22.865	0.729	21.731	0.476	29.174	0.850	22.293	0.765	41.927	0.619	19.587	0.747	22.176	0.686

TABELLE 5.10: Vergleich vom Forward Warping mittels Vorwärts- und Rückwärtsfluss. Es wurde in der ersten Spalte unter jeder Splattingstrategie (**fett**) die verwendete Flussrichtung angegeben.

Es ist zu erkennen, dass es im 1-Bild Fall tatsächlich zu unterschiedlichen Ergebnissen kommt, bei denen eine Flussrichtung besser abschneidet als die andere. Insgesamt liegen die Ergebnisse in Bezug auf die Splattingstrategien aber erneut vom Verhältnis her gleich zueinander: Das gewichtete Splatting auf vier Nachbarn (4N-Gewicht) scheint sich beim Forward Warping für beide Flussrichtungen durchzusetzen. Was außerdem auffällt, ist der durchschnittliche SSIM Index bei Verwendung Rückwärtsflusses $\vec{v}_{1,b}$. Dieser ist bei allen Splattingarten im Durchschnitt etwas schlechter als beim Forward Warping mit dem Vorwärtsfluss $\vec{v}_{0,f}$.

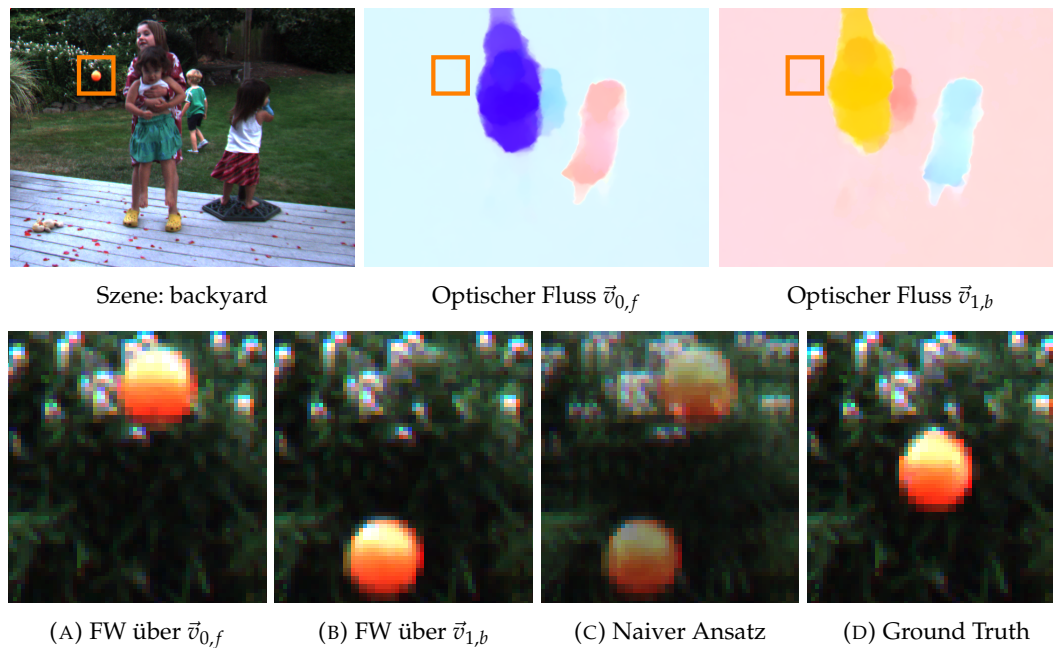


ABBILDUNG 5.14: Die Ergebnisse eines Forward Warplings von Bildinformation I_0 über $\vec{v}_{0,f}$ und einem Forward Warping von Bildinformation I_1 über $\vec{v}_{1,b}$. Es sind bei Objekten, deren Bewegung nicht vom Fluss detektiert wurde, unterschiedliche Positionen im Interpolanten zu erkennen. Szene: Backyard.

Die Ergebnisse dieses Vergleichs bestätigen für die Zwischenbildinterpolation mit optischem Fluss das Vorhandensein einer Abhängigkeit von der Flussrichtung. Je nach Flussrichtung kommt es scheinbar zu generell besseren oder schlechteren Ergebnissen, worüber auch der immer leicht niedrigere SSIM Index beim Rückwärtsfluss erklärt werden könnte.

Durch den direkten Vergleich hat sich außerdem noch ein charakteristisches Verhalten der Verfahren des 1-Bild Falls gezeigt: Für den Fall von nicht detektierter Bewegung im Bild, kommt es je nach Flussrichtung zu auffälligen Unterschieden im Interpolanten (vgl. Abbildung 5.14). Dort ist zu erkennen, dass es zwar zu keinen doppelten Objekten im Bild kommt, aber dass Objekte sich je nach Flussrichtung an unterschiedlichen Positionen befinden können. Dies liegt an der nicht vorhandenen Verschiebung dieser Objekte, wodurch sie an der alten Position, vor oder nach dem Zwischenbild, bleiben. Ob dieses Verhalten besser oder schlechter als ein Doppelungsartefakt ist, ist schwierig zu beurteilen. Um darüber eine Entscheidung zu treffen, bedarf es weiterer Tests, die zum Beispiel Zwischenbilder für längere Bildsequenzen erzeugen und anschließend die interpolierte Bewegung im Bild mit einem subjektiven Eindruck bewerten.

5.7.2 Ersatz für symmetrische Verfahren

Die symmetrischen Verfahren, die erstmals den 2-Bild Fall darstellen, wurden bereits geprüft. Der dort vorhandene charakteristische Zwischenschritt dieser Verfahren besteht aus dem Warping eines optischen Flusses mit sich selbst, dem anschließenden Umdrehen und der Kürzung des gewarpten Flusses. Speziell beim symmetrischen Forward Warping mit einem Vorwärtsfluss $\vec{v}_{0,f}$ fällt auf, dass der künstlich umgedrehte Fluss von der Bedeutung genau dem entgegengerichteten Fluss $\vec{v}_{1,b}$ entspricht, der im sehr ähnlichen Forward Forward Warping Anwendung findet (vgl. Abschnitt 4.5.2).

Was nun von Interesse ist, inwiefern sich diese beide Flüsse überhaupt voneinander unterscheiden, um so den Unterschied zwischen dem symmetrischen Forward Warping und dem Forward Forward Warping klar zu machen. In Tabelle 5.11 sind beispielhaft die Unterschiede des Vorwärtsflusses $\vec{v}_{0,f}$, der einmal komplett um sich selbst nach I_1 gewarpt und umgedreht wurde, gegenüber dem Rückwärtsfluss $\vec{v}_{1,b}$ über den durchschnittlichen Endpoint Error aus Kapitel 2 festgehalten (es wurde exemplarisch nur der Fall für den Vorwärtsfluss $\vec{v}_{0,f}$ getestet). Dies soll helfen, die Größenordnung der Unterschiede von den Flüssen einschätzen zu können.

Splattung	Army	Backyard	Basketball	Dumptruck	Evergreen	Grove	Mequon	\emptyset
1N-Uniform	0.064	0.254	0.528	0.088	0.477	0.502	0.333	0.317
1N-Gewicht	0.064	0.255	0.528	0.088	0.478	0.502	0.335	0.318
1N-Austausch	0.065	0.259	0.529	0.088	0.482	0.503	0.341	0.320
4N-Uniform	0.063	0.251	0.527	0.088	0.475	0.500	0.330	0.316
4N-Gewicht	0.063	0.252	0.527	0.088	0.475	0.500	0.330	0.315
4N-Austausch	0.065	0.260	0.530	0.088	0.483	0.505	0.342	0.321

TABELLE 5.11: Der Unterschied zwischen dem gewarpten und umgedrehten Vorwärtsfluss $\vec{v}_{0,f}$ und dem Rückwärtsfluss $\vec{v}_{1,b}$. Es wurden wieder alle möglichen Splattings für das Forward Warping des Flusses angegeben. Gemessen wurde der Unterschied über den *Average Endpoint Error* (AEE).

Bei der Betrachtung dieser Tabelle fällt auf, dass es zwar Unterschiede gibt, diese aber relativ gering sind. Ob beispielsweise die Nutzung eines Rückwärtsflusses einem gewarpten und umgedrehten Vorwärtsfluss überlegen ist, wird aus diesen Differenzen aber noch nicht klar. Aus diesem Grund soll jetzt ein ausführlicher Vergleich zwischen den teilweise sehr ähnlichen Verfahren des (bidirektionalen) 2-Bild Falls durchgeführt werden, um den Einfluss des Flusses zu klären. Außerdem sollen dabei die bidirektionalen Varianten der symmetrischen Verfahren untersucht werden, um ebenfalls herauszufinden, ob das Mitteln von zwei Flussrichtungen möglicherweise bessere Ergebnisse liefert.

Dazu werden wieder für Frame 10 (I_0) und Frame 12 (I_1) Zwischenbilder erzeugt. Dies geschieht über die symmetrischen Verfahren (symmetrisches Forward Warping und symmetrisches Backward Warping) für beide mögliche Flüsse: $\vec{v}_{0,f}$ und $\vec{v}_{1,b}$. Die Ergebnisse dieser Verfahren sollen mit ihren bidirektionalen Varianten (bidirektionales symmetrisches Forward Warping und bidirektionales symmetrisches Backward Warping) verglichen werden, die beide $\vec{v}_{0,f}$ und $\vec{v}_{1,b}$ in ihren Berechnungen verwenden. Um den eben angesprochenen Unterschied der Flüsse zu klären, werden ebenfalls das Forward Forward Warping und das Backward Backward Warping als bidirektionale Verfahren in den Vergleich integriert (beide arbeiten jeweils mit $\vec{v}_{0,f}$ und

$\vec{v}_{1,b}$). Für alle getesteten Verfahren wird, falls enthalten, ein 4N-Gewicht Sampling durchgeführt und beim Splatting werden alle sechs verschiedenen Splattingstrategien getestet. In Tabelle 5.12 auf der vorherigen Seite sind die Ergebnisse dieses Vergleichs aufgeführt.

In dieser Tabelle gibt es einige Dinge zu beobachten. Was beispielsweise gut Abzulesen ist, ist das gute Abschneiden der bidirektionalen Varianten der symmetrischen Verfahren. Es ist zu erkennen, dass die bidirektionalen Varianten mit ihren Fehlern entweder zwischen den beiden direktionalen Verfahren liegen oder sehr häufig auch bessere Ergebnisse liefern. Im Durchschnitt ist dies sowohl beim symmetrischen Forward Warping als auch beim symmetrischen Backward Warping der Fall. Das bidirektionale symmetrische Forward Warping scheint die besten Ergebnisse mit einem 4N-Gewicht Splatting zu liefern, das bidirektionale symmetrische Backward Warping scheint mit 4N-Uniform und 4N-Gewicht gleich gute Ergebnisse zu erzeugen. Neben den bidirektionalen Varianten der symmetrischen Verfahren ist zu erkennen, dass das Forward Forward Warping und das Backward Backward Warping durchaus vergleichbare Ergebnisse zu den symmetrischen Verfahren liefern. Das Forward Forward Warping und das Backward Backward Warping scheinen beide mit einem 4N-Gewicht Splatting besonders gut zu funktionieren. Besonders das Forward Forward Warping erzeugt in dieser Variante sogar die besten Ergebnisse des Vergleichs.

Zusammenfassend hat dieses Experiment die erhofften Verbesserungen der bidirektionalen Verfahren gegenüber den (direktionalen) symmetrischen Verfahren bestätigt. Die bidirektionalen Varianten konnten nicht nur konsistentere Ergebnisse liefern, sondern teilweise auch bessere. Im Falle des Forward Forward Warpings konnte gezeigt werden, dass es scheinbar tatsächlich vorteilhafter ist mit beiden Flussrichtungen zu arbeiten, statt ein Fluss Warping inklusive Umdrehen und Kürzung durchzuführen. Dies könnte an dem Wegfallen des Fluss Warpings liegen, da dieses immer mit Annahmen (Splatting und Inpainting) verbunden ist und eine mögliche Fehlerquelle darstellt.

Neben dem quantitativen Vergleich ist noch ein qualitativer Vergleich von Interesse, denn ein charakteristisches Merkmal der symmetrischen Verfahren waren die Doppelungsartefakte und es bleibt zu sehen, wie die bidirektionalen Verfahren unter diesem Aspekt abschneiden. In Abbildung 5.15 sind für die Szene Mequon die Interpolanten gegenübergestellt. Dort ist zu erkennen, dass es erneut zu Doppelungsartefakten kommt, diese bei den bidirektionalen Verfahren aber eine leicht andere Erscheinung besitzen als bei den symmetrischen Verfahren. Am Kopf in Abbildung 5.15 ist bei den symmetrischen Verfahren gut zu erkennen, dass es je nach Flussrichtung zu mehr oder weniger starken Doppelungsartefakten kommt. Die bidirektionalen Varianten dieser Verfahren wirken dagegen von der Stärke der Doppelungsartefakte in der Mitte zu liegen, was vermutlich auf die involvierte Mittelung beider Richtungen zurückzuführen ist. Diese Beobachtungen scheinen sich auch mit den Ergebnissen der symmetrischen Backward Warpings aus der Arbeit von Râket et al. [31] zu decken. Beim Forward Forward Warping, mit den besten Ergebnissen im Test, sind abschließend im Vergleich sehr schwache Doppelungen zu erkennen, daher wohl auch das gute Abschneiden bei dem RMSE und SSIM Index.

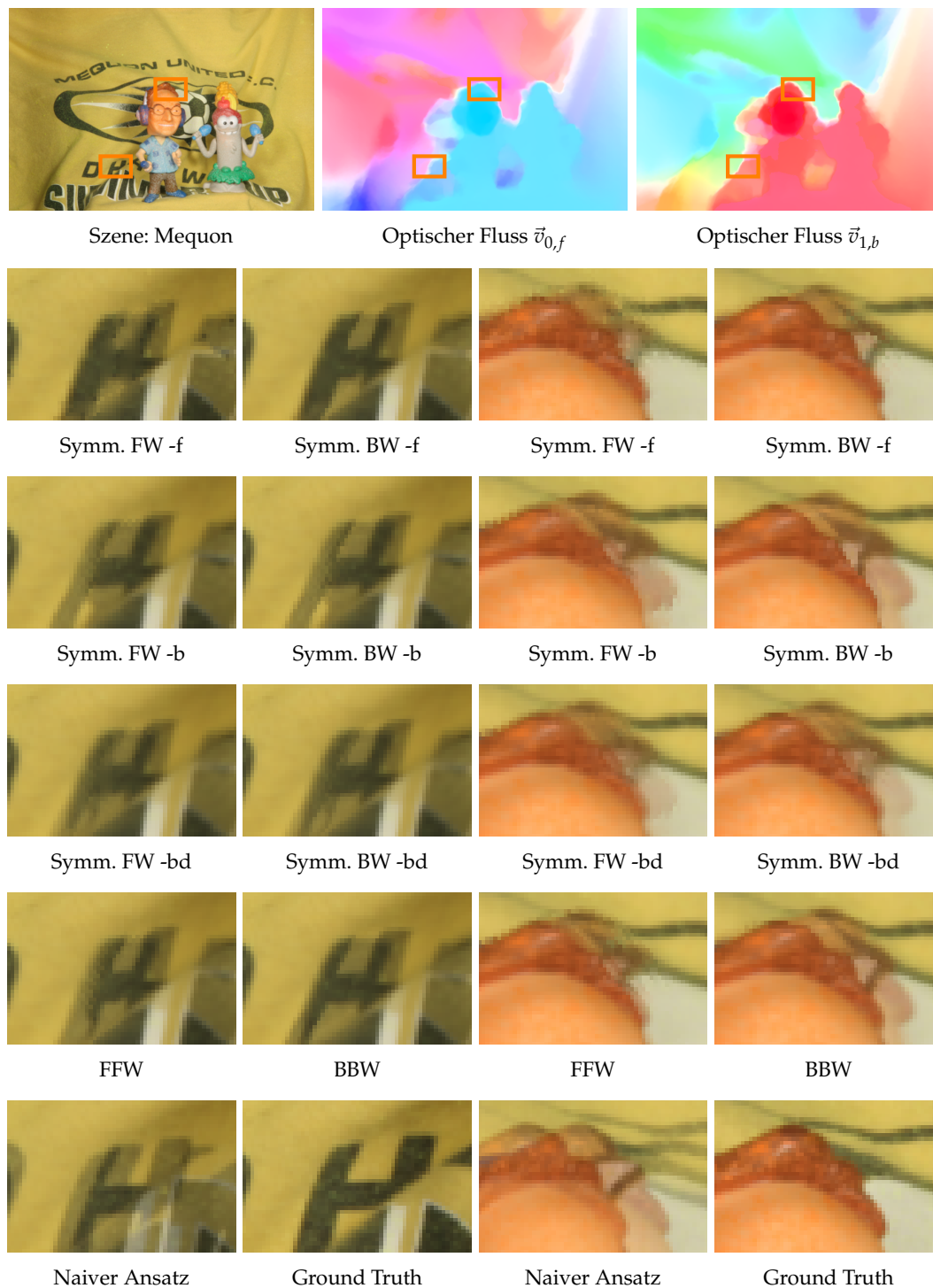


ABBILDUNG 5.15: Optischer Vergleich von erzeugten Interpolanten bei den Tests der symmetrischen und bidirektionalen Verfahren. Es sind in allen dargestellten Ergebnissen Doppelungsartefakte zu erkennen.

5.8 Gesamtvergleich

In diesem Kapitel wurden die Verfahren des letzten Kapitels in einer Reihe von Experimenten verglichen und die beinhalteten Überlegungen wurden auf ihre Richtigkeit überprüft. In diesem letzten Abschnitt soll zum Abschluss noch ein gebündelter Vergleich aller vorgestellten Verfahren vorgenommen werden, um das Bildinterpolationsverfahren mit den besten Ergebnissen zu bestimmen. Die besten Ergebnisse sollen wieder anhand der Fehlermaße des RMSE und des SSIM Index bestimmt werden, aber auch ein abschließender subjektiver Eindruck soll der Bewertung hinzugefügt werden.

Es werden in diesem Abschnitt alle bisher vorgestellten Verfahren nochmal betrachtet, was insbesondere für die direktionalen Verfahren bedeutet, dass beide möglichen Flussrichtungen separat aufgeführt werden. Für die jeweils verwendete Konfiguration an Splatting- und Samplingstrategien gilt, dass immer die Strategien mit dem im Durchschnitt besten Fehlern verwendet wird. Da die SSIM Indizes häufig identisch sind und um einen kompakten Vergleich sicherzustellen, wird sich im Falle eines gleichen SSIM Index immer für die Konfiguration mit dem besseren RMSE entschieden. In Tabelle 5.13 sind zur Übersicht alle Verfahren inklusive ihrer besten Konfiguration und Abkürzung aufgeführt. In manchen Fällen kam es vor, dass Konfigurationen im Durchschnitt für beide Fehlermaße identische Ergebnisse erzeugt haben. In diesem Fall wurden diese zwar in Tabelle 5.13 aufgeführt, es wird jedoch für den Gesamtvergleich exemplarisch die 4N-Gewicht Konfiguration als Repräsentant verwendet. Wie bereits bei der Untersuchung der Samplingstrategien in

Verfahren	Richtung	Splatting	Sampling	Abkürzung
Forward Warping	Vorwärts	4N-Gewicht	—	FW -f
Forward Warping	Rückwärts	4N-Gewicht	—	FW -b
Backward Warping	Vorwärts	1N-Gewicht/ 4N-Gewicht	4N-Gewicht	BW -f
Backward Warping	Rückwärts	4N-Uniform/ 4N-Gewicht	4N-Gewicht	BW -b
Symm. Forward Warping	Vorwärts	4N-Gewicht	—	Symm. FW -f
Symm. Forward Warping	Rückwärts	4N-Gewicht	—	Symm. FW -b
Symm. Backward Warping	Vorwärts	4N-Uniform	4N-Gewicht	Symm. BW -f
Symm. Backward Warping	Rückwärts	4N-Uniform	4N-Gewicht	Symm. BW -b
Bidir. Symm. Forward Warping	Bidirektional	4N-Gewicht	—	Symm. FW -bd
Bidir. Symm. Backward Warping	Bidirektional	4N-Uniform/ 4N-Gewicht	4N-Gewicht	Symm. BW -bd
Forward Forward Warping	Bidirektional	4N-Gewicht	—	FFW
Backward Backward Warping	Bidirektional	4N-Gewicht	4N-Gewicht	BBW
Naiver Ansatz	—	—	—	Naiver Ansatz

TABELLE 5.13: Die im Gesamtvergleich betrachteten Verfahren mit ihrer Konfiguration (Splatting- und Samplingstrategie). Die angegebene Richtung bezieht sich auf die Richtung des verwendeten optischen Flusses. Vorwärts bedeutet optischer Fluss $\vec{v}_{0,f}$ und Rückwärts bedeutet Fluss $\vec{v}_{1,b}$.

Abschnitt 5.4.2 erwähnt, waren die Ergebnisse was das Sampling angeht ziemlich eindeutig. Deswegen wurde bei allen darauffolgenden Verfahren, falls notwendig, immer nur ein 4N-Gewicht Sampling betrachtet. Dies ist der Grund, warum sich in Tabelle 5.13 ausschließlich ein 4N-Gewicht Sampling wiederfindet. Ein Blick auf Tabelle 5.14 zeigt den eigentlichen Gesamtvergleich und damit die Bündelung aller

Verfahren	Army		Backyard		Basketball		Dumprtruck		Evergreen		Grove		Mequon		∅	
	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM	RMSE	SSIM
FFW	5.771	0.917	18.367	0.771	16.541	0.534	23.504	0.871	17.834	0.815	30.892	0.708	13.974	0.820	17.590	0.735
Symm. FW -bd	5.798	0.917	18.424	0.771	16.365	0.534	23.520	0.871	17.877	0.813	30.821	0.708	13.910	0.819	17.596	0.735
Symm. BW -bd	5.788	0.917	18.476	0.771	16.366	0.534	23.525	0.871	17.954	0.814	30.821	0.709	13.996	0.819	17.612	0.735
Symm. FW -b	5.829	0.917	18.467	0.770	16.314	0.534	23.582	0.871	18.184	0.814	30.846	0.707	14.109	0.816	17.659	0.733
Symm. BW -b	5.815	0.917	18.527	0.770	16.316	0.534	23.580	0.871	18.344	0.814	30.845	0.707	14.278	0.816	17.699	0.733
BBW	5.848	0.917	18.911	0.768	16.584	0.532	23.606	0.871	18.856	0.811	31.246	0.705	14.579	0.816	17.954	0.733
Symm. FW -f	5.796	0.917	18.726	0.769	16.959	0.531	23.546	0.871	18.880	0.808	31.483	0.704	14.428	0.819	17.956	0.733
Symm. BW -f	5.787	0.917	18.861	0.769	16.966	0.531	23.553	0.871	18.893	0.808	31.477	0.705	14.663	0.819	17.990	0.733
FW -f	6.479	0.906	21.454	0.734	22.065	0.495	29.508	0.852	22.571	0.779	35.143	0.694	16.194	0.802	20.785	0.716
BW -b	6.625	0.906	21.767	0.732	21.359	0.495	29.965	0.852	23.770	0.779	34.719	0.693	17.443	0.794	21.073	0.713
FW -b	7.005	0.906	21.978	0.737	20.492	0.493	28.149	0.854	20.894	0.790	37.990	0.665	18.619	0.789	20.836	0.709
BW -f	7.112	0.904	23.545	0.733	22.047	0.490	28.229	0.854	22.846	0.780	39.535	0.658	20.003	0.789	21.928	0.707
Naiver Ansatz	10.356	0.772	34.991	0.442	25.350	0.460	24.764	0.853	28.488	0.698	69.733	0.084	32.256	0.250	30.465	0.433

TABELLE 5.14: Gebündelte Ergebnisse von den vergangenen Experimenten dieses Kapitels. Es wurde für jedes Warpingverfahren die Konfiguration aus Tabelle 5.13 verwendet. Sortiert wurde primär nach SSIM Index und sekundär nach RMSE.

Experimente dieses Kapitels. Die Verfahren wurden dort primär nach dem SSIM Index und sekundär nach RMSE geordnet und zusätzlich nach ähnlichen Ergebnissen gruppiert. Es ist zu erkennen, dass in Bezug auf den SSIM Index sich die bidirektionalen Verfahren im Gesamtvergleich durchgesetzt haben. Die Unterschiede waren was den RMSE angeht relativ gering, aber insgesamt war das Forward Forward Warping bei der Kombination beider Fehlermaße insgesamt am besten. Die nächste große Gruppe mit ähnlichen Ergebnissen bilden die restlichen (bidirektionalen) Verfahren des 2-Bild Falls. Diese weisen alle einen identischen SSIM Index auf, sind aber im Vergleich zu den drei besten Verfahren sowohl beim RMSE als auch beim SSIM Index leicht schlechter. Die letzte größere Gruppe bilden die Verfahren des 1-Bild Falls. Bei diesen Verfahren ist für beide Fehlermaße ein deutlicher Abstand zur nächstbesten Gruppe zu beobachten. Beim Schlusslicht handelt es sich um den naiven Ansatz, der nochmal mit deutlich schlechteren Werten dasteht. Generell fällt auf, dass die Verfahren, die ein Forward Warping für den Transport der Bildinformation verwenden, tendenziell bessere Ergebnisse liefern als solche die ein Backward Warping nutzen. Die Flussrichtung scheint ebenfalls einen wiederholten Einfluss zu besitzen: Zwar sind die bidirektionalen Verfahren den direktionalen Verfahren scheinbar leicht überlegen, aber es fällt auf, dass die Verwendung des Rückwärtsfluss tendenziell bessere Ergebnisse liefert als der Vorwärtsfluss.

Zusammenfassend hat sich also das Forward Forward Warping in Bezug auf beide Fehlermaße durchgesetzt. Dabei handelt es sich interessanterweise auch um ein Verfahren mit einer der geringsten Anzahl an involvierten Warpingvorgängen (zwei). Zusammen mit den bidirektionalen Varianten der symmetrischen Verfahren (bidirektionales symmetrisches Forward Warping und bidirektionales Backward Warping), die im Gegensatz die vergleichsweise meisten Warpings beinhalten (sechs), handelt es um die eindeutigen Sieger des Gesamtvergleichs.

Über einen subjektiven Eindruck ist es etwas schwieriger einen allgemeinen Sieger auszuwählen. Bei isolierter Betrachtung der Interpolanten gibt es optisch nämlich nur zwei wesentliche Klassen von Verfahren: Verfahren mit Doppelungsartefakten und Verfahren mit starker Abhängigkeit von der Flussrichtung (1-Bild Fall), wo Objekte an unterschiedlichen Positionen erscheinen. Im Allgemeinen wären vermutlich keine Doppelungsartefakte vorzuziehen, allerdings macht die Abhängigkeit der

Richtung von Szene zu Szene die Ergebnisse des 1-Bild Falls relativ uneinheitlich. Es müsste für jede individuelle Szene begründet werden, welche Richtung vorteilhafter ist. Da die Doppelungsartefakte bei den bidirektionalen Verfahren nicht so auffällig sind wie beim naiven Ansatz und durch ihre zusätzlich starke Abhängigkeit vom optischem Fluss, handelt es sich dabei vermutlich um die Verfahren mit dem meisten Potential. Denn je nach Genauigkeit des verwendeten optischen Flusses könnten noch stärker reduzierte Doppelungen erreicht werden, was in Verbindung mit den jetzt schon besten quantitativen Fehler, ein überzeugendes Verfahren darstellt. Aus diesem Grund wird sich aus subjektiver Sicht für die Verfahren mit dem meisten Potential entschieden, bei dem es sich um die ersten drei Plätze des Gesamtvergleichs handelt: Forward Forward Warping und die bidirektionalen Varianten der symmetrischen Verfahren.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

In dieser Arbeit wurden eine Vielzahl von Verfahren vorgestellt, um Zwischenbilder mit optischen Flüssen zu erzeugen. Diese wurden in schrittweiser Form vorgestellt, um die beinhalteten Überlegungen möglichst sinnvoll zu ordnen und zu kategorisieren. Bei den vorgestellten Verfahren handelt es sich in erster Linie um die Grundvarianten der einzelnen Vorgehensweisen, welche in verschiedenen Konfigurationen getestet wurden, um ihre Praxistauglichkeit zu herauszufinden.

Dafür wurde zunächst in Kapitel 2 und Kapitel 3 der optischer Fluss und das Warping als grundlegende Mittel vorgestellt. Es wurden dabei die zentralen Techniken Forward Warping und Backward Warping vorgestellt, inklusive der enthaltenen Konzepte des Samplings, Splattings und des Inpaintings. Mithilfe dieser Techniken wurden im Anschluss in Kapitel 4 die verschiedenen Bildinterpolationsverfahren einzeln vorgestellt und formal festgehalten. Begonnen beim 1-Bild Fall folgte ein erweiterter 2-Bild Fall, der mit mehr Bildern arbeitet. Danach wurde dieser Fall um eine weitere Flussrichtung erweitert (bidirektionale Verfahren). Im vorherigen Kapitel fand schließlich eine Evaluation der vorgestellten Verfahren statt, um die beinhalteten Überlegungen und möglichen Probleme mit konkreten Zahlen zu überprüfen.

Bei den durchgeführten Experimenten wurde klar, dass bereits beim 1-Bild Fall überzeugende Zwischenbilder erzeugt werden können. Durch das Verwenden von nur einer Bildinformation kam es zu keinen Doppelungsartefakten, aber in Bezug auf den quantitativen Fehler waren sie den anderen Verfahren unterlegen. Im dort durchgeführten Vergleich vom Splatting und Sampling hat sich meist die Variante, die mit vier Nachbarn arbeitet und eine gewichtete Verteilung vornimmt, durchgesetzt (4N-Gewicht bzw. bilineare Interpolation). Beim Test des Inpaintings wurde klar, dass ein schnell konvergierendes Verfahren von Vorteil ist und zusätzlich konnten die Schwächen des RMSE als Bewertung von Interpolanten beobachtet werden. Die zwei symmetrischen Verfahren des 2-Bild Falls konnten durch die Verwendung von zwei Bildern in Bezug auf beide Fehlermaße noch bessere Ergebnisse erzeugen als der 1-Bild Fall, allerdings mit einer Einführung von auffälligen Doppelungsartefakten. Der bidirektionale Ersatz über das Forward Forward Warping und das Backward Backward Warping konnte später in vielen Fällen noch zu einer kleinen Verbesserung der Fehler führen. Die vorgestellten bidirektionalen Varianten der symmetrischen Verfahren verhielten sich ähnlich zu den anderen bidirektionalen Verfahren und auch sie zeigten im Vergleich eine sichtbare Reduzierung der Fehler und der Doppelungsartefakte. Als das Verfahren mit den insgesamt besten Ergebnissen hat sich im abschließenden Gesamtvergleich das Forward Forward Warping herausgestellt.

6.2 Ausblick

Bei den vorgestellten Verfahren handelt es sich um die wesentlichen Herangehensweisen für die Zwischenbildinterpolation mit optischem Fluss bis hin zum bidirektionalen 2-Bild Fall. Dabei wurden jedoch eigentlich nur die Grundvarianten der Verfahren verwendet, die in sich noch viel Potential für Verbesserungen besitzen. Dazu gehören zusammengefasst verbessertes Splatting, Sampling, Blending oder andere photometrische Verbesserungen. Beim Splatting wurde beispielsweise nur eine kleine Auswahl an Verteilungsstrategien und Kollisionsbehandlungen getestet, wo deshalb noch ein Vergleich zu anderen Strategien durchgeführt werden könnte. Dazu gehört beispielsweise eine kreisförmige Verteilungsstrategie oder eine Kollisionsbehandlung über eine photometrische Bewertung [2]. In diesem Zusammenhang könnte insbesondere ein Splatting speziell für optische Flüsse noch genauer untersucht werden. Beim Sampling gibt es ebenfalls noch weitere Verfahren, die sich für einen erweiterten Vergleich anbieten. So könnte eine Interpolation über mehr als vier Stellen stattfinden, in Form von beispielsweise einer bikubischen Interpolation. Auch das Inpainting bestand lediglich aus einer ungerichteten Diffusion, wo teilweise unscharfe Bildteile entstanden sind. Dort könnten noch anisotropischen Anforderungen eingebracht werden, um eine möglicherweise schärfere Lückenfüllung zu beobachten. Die durchgeführte Zusammenführung (Blending) von Bildinformationen bei den Verfahren beginnend beim 2-Bild Fall sind auch erweiterbar, denn es wurde lediglich ein Blending an besetzten Stellen durchgeführt. Hier könnte noch ein selektives Blending durch eine Form von Occlusion- oder Depth-Reasoning differenziert untersucht und verglichen werden [2].

Zwar wurde der Großteil der grundlegenden Warpingstrategien getestet, doch gibt es immer noch weitere Grundverfahren, die in dieser Arbeit nicht behandelt wurden. Dazu gehören beispielsweise der Fall des gemischten Warpings, der dem 2-Bild Fall zuzuordnen ist. Dort würde mithilfe eines optischen Flusses die eine Bildinformation über ein Forward Warping und die andere Bildinformation über ein Backward Warping zum Interpolanten gelangen. Diese Variante, inklusive einer alternativen Form des Backward Warpings, findet sich beispielsweise in der Arbeit von Williams et al. [37]. Die gedanklich nächste Klasse an Verfahren nach dem bidirektionalen 2-Bild Fall wäre noch der Mehrbild-Fall gewesen, der in dieser Arbeit nicht hinzugezogen wurde. Bei dieser Art der Interpolation würde es eine leichte Abkehr von der zentralen Annahme aus Abschnitt 4.2 geben, da keine lineare Interpolation mehr stattfindet. Stattdessen wird eine Bewegung über mehrere Bilder über optische Flüsse verfolgt und es wird stellenweise eine Kurve als Bewegungstrajektorie verwendet. Diese Kurven würden eine neue Mappingfunktion für das Warping darstellen, weswegen alle vorgestellten Splatting-, Sampling- und Inpaintingstrategien für diesen Fall noch ausführlich getestet werden könnten.

Neben den Interpolationsverfahren spielte der optische Fluss auch einer sehr große Rolle in dieser Arbeit. Es wurde beobachtet, wie er maßgeblichen Einfluss auf die entstandenen Interpolanten genommen hat (Position von Objekte und Doppelungsartefakte). Verwendet wurde in dieser Arbeit jedoch nur ein einziges Verfahren, für welches die Eignung für die Zwischenbildinterpolation gezeigt werden konnte (ProFlow-Baseline). Theoretisch wäre aber noch ein Vergleich mit anderen Verfahren für optische Flüsse interessant, um deren allgemeine Kompatibilität zu den vorgestellten Techniken zu vergleichen.

Auch die Tests dieser Arbeit an sich sind erweiterbar. So bieten sich noch Experimente an, bei denen verschiedene Situationen für die Interpolation genauer betrachtet werden könnten. Dazu gehören beispielsweise Tests von Bildsequenzen mit optischen Mängeln (Blur, Rauschen oder Lücken). Auch ein Verhalten von sukzessiver Interpolation für einen angestrebten Slow-Motion Effekt könnte genauer untersucht werden. Darüber hinaus könnten die Verfahren auch noch über längere Bildfolgen miteinander verglichen werden, um so noch einen durchschnittlichen Interpolationsfehler zu untersuchen. Die Bewertung aller Zwischenbilder erfolgte in erster Linie über zwei gewählte Fehlermaße. Der tatsächlich subjektive Eindruck der menschlichen Wahrnehmung ist am Ende aber vermutlich noch aussagekräftiger als eine Bewertung über eine Zahl. Aus diesem Grund wäre eine Untersuchung der vorgestellten Verfahren in einer Studie mit mehreren Probanden möglich. So könnte eine noch aussagekräftigere Beurteilung der getesteten Verfahren erlangt werden.

Literatur

- [1] Simon Baker und Takeo Kanade. *Super-resolution optical flow*. Carnegie Mellon University, The Robotics Institute, 1999.
- [2] Simon Baker u. a. „A database and evaluation methodology for optical flow“. In: *International journal of computer vision* 92.1 (2011), S. 1–31.
- [3] Thaddeus Beier und Shawn Neely. „Feature-based image metamorphosis“. In: *ACM SIGGRAPH computer graphics* 26.2 (1992), S. 35–42.
- [4] Michael J Black und Paul Anandan. „The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields“. In: *Computer vision and image understanding* 63.1 (1996), S. 75–104.
- [5] Thomas Brox u. a. „High accuracy optical flow estimation based on a theory for warping“. In: *European conference on computer vision*. Springer. 2004, S. 25–36.
- [6] D. J. Butler u. a. „A naturalistic open source movie for optical flow evaluation“. In: *European Conf. on Computer Vision (ECCV)*. Hrsg. von A. Fitzgibbon et al. (Eds.) Part IV, LNCS 7577. Springer-Verlag, Okt. 2012, S. 611–625.
- [7] Haiyang Chao, Yu Gu und Marcello Napolitano. „A survey of optical flow techniques for robotics navigation applications“. In: *Journal of Intelligent & Robotic Systems* 73.1-4 (2014), S. 361–372.
- [8] Toan Nguyen Dinh u. a. „A novel motion compensated frame interpolation method for improving side information in distributed video coding“. In: *2007 International Symposium on Information Technology Convergence (ISITC 2007)*. IEEE. 2007, S. 179–183.
- [9] Karen Egiazarian u. a. „New full-reference quality metrics based on HVS“. In: *Proceedings of the Second International Workshop on Video Processing and Quality Metrics*. Bd. 4. 2006.
- [10] Denis Fortun, Patrick Bouthemy und Charles Kervrann. „Optical flow modeling and computation: a survey“. In: *Computer Vision and Image Understanding* 134 (2015), S. 1–21.
- [11] Thomas Funkhouser. „Image Warping“. In: *Princeton University (New Jersey)* (2000).
- [12] Andreas Geiger, Philip Lenz und Raquel Urtasun. „Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite“. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [13] Chris A Glasbey und Kantilal Vardichand Mardia. „A review of image-warping methods“. In: *Journal of applied statistics* 25.2 (1998), S. 155–171.
- [14] Paul S Heckbert. „Fundamentals of texture mapping and image warping“. In: (1989).

- [15] Yinlin Hu, Yunsong Li und Rui Song. „Robust interpolation of correspondences for large displacement optical flow“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, S. 481–489.
- [16] Yinlin Hu, Rui Song und Yunsong Li. „Efficient coarse-to-fine patchmatch for large displacement optical flow“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, S. 5704–5712.
- [17] E. Ilg u. a. „FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. URL: <http://lmb.informatik.uni-freiburg.de/Publications/2017/IMSKDB17>.
- [18] M Jakubowski und G Pastuszak. „Block-based motion estimation algorithms—a survey“. In: *Opto-Electronics Review* 21.1 (2013), S. 86–102.
- [19] Joel Janai u. a. „Unsupervised learning of multi-frame optical flow with occlusions“. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, S. 690–706.
- [20] Ravi Krishnamurthy, John W Woods und Pierre Moulin. „Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields“. In: *IEEE transactions on circuits and systems for video technology* 9.5 (1999), S. 713–726.
- [21] Marc Levoy. „Display of surfaces from volume data“. In: *IEEE Computer graphics and Applications* 8.3 (1988), S. 29–37.
- [22] Feng Liu u. a. „Content-preserving warps for 3D video stabilization“. In: *ACM Transactions on Graphics (TOG)* 28.3 (2009), S. 1–9.
- [23] Oleg Lobachev, Birte S Steiniger und Michael Guthe. „Compensating anisotropy in histological serial sections with optical flow-based interpolation.“ In: *SCCG*. 2017, S. 14–1.
- [24] Daniel Maurer und Andrés Bruhn. „Proflow: Learning to predict optical flow“. In: *Proc. British Machine Vision Conference (BMVC)*. 2018.
- [25] Daniel Maurer, Michael Stoll und Andrés Bruhn. „Order-Adaptive and Illumination-Aware Variational Optical Flow Refinement.“ In: *BMVC*. 2017.
- [26] Hui Men u. a. „Visual quality assessment for motion compensated frame interpolation“. In: *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2019, S. 1–6.
- [27] Moritz Menze und Andreas Geiger. „Object Scene Flow for Autonomous Vehicles“. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
- [28] Simon Niklaus, Long Mai und Feng Liu. „Video frame interpolation via adaptive separable convolution“. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, S. 261–270.
- [29] Michael Otte und H-H Nagel. „Optical flow estimation: advances and comparisons“. In: *European conference on computer vision*. Springer. 1994, S. 49–60.
- [30] Nils Papenberg u. a. „Highly accurate optic flow computation with theoretically justified warping“. In: *International Journal of Computer Vision* 67.2 (2006), S. 141–158.
- [31] Lars Lau Rakêt u. a. „Motion compensated frame interpolation with a symmetric optical flow constraint“. In: *International Symposium on Visual Computing*. Springer. 2012, S. 447–457.

-
- [32] Jerome Revaud u. a. „Epicflow: Edge-preserving interpolation of correspondences for optical flow“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, S. 1164–1172.
- [33] Eero P Simoncelli. „14 Bayesian Multi-Scale Differential Optical Flow“. In: (1999).
- [34] Frank Steinbrücker, Thomas Pock und Daniel Cremers. „Large displacement optical flow computation without warping“. In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, S. 1609–1614.
- [35] Deqing Sun u. a. „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.
- [36] Zhou Wang u. a. „Image quality assessment: from error visibility to structural similarity“. In: *IEEE transactions on image processing* 13.4 (2004), S. 600–612.
- [37] Winston L Williams und William A Barrett. „Optical flow interpolation of serial slice images“. In: *Medical Imaging 1993: Image Processing*. Bd. 1898. International Society for Optics und Photonics. 1993, S. 93–104.

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