

Institut für Visualisierung und Interaktive Systeme  
Universität Stuttgart  
Universitätsstraße 38  
D - 70569 Stuttgart

Bachelorarbeit Nr. 173

# Variationsansätze zur Berechnung des optischen Flusses für Mehrkanalbilder

Niklas Kaulitz

**Studiengang:** Informatik  
**Prüfer/in:** Prof. Dr.-Ing. Andrés Bruhn  
**Betreuer/in:** Prof. Dr.-Ing. Andrés Bruhn

**Beginn am:** 01. Juni 2014  
**Beendet am:** 28. Oktober 2014  
**CR-Nummer:** G 1.3, G 1.6, G 1.8, I 2.10, I 4.8



## ZUSAMMENFASSUNG

---

Optischer Fluss ist für videobasiertes Bildverstehen eine wichtige Grundlage. Es gibt viele Methoden und Varianten um ihn zu berechnen, jedoch gibt es noch mehr ungetestete Möglichkeiten. In dieser Arbeit wird ein variationales Verfahren für Mehrkanalbilder entwickelt, welches unterschiedliche Konstanzannahmen unterstützt. Diese werden gegeneinander evaluiert und es wird getestet, ob eine Kombination mehrerer Konstanzannahmen möglich ist und gute Ergebnisse liefert.



# INHALTSVERZEICHNIS

---

1	EINLEITUNG	1
1.1	Motivation . . . . .	1
1.2	Aufgabenstellung . . . . .	2
1.3	Verwandte Arbeiten . . . . .	2
1.4	Gliederung . . . . .	3
2	GRUNDLAGEN VON HORN UND SCHUNCK	5
2.1	Optischer Fluss . . . . .	5
2.1.1	Die Grauwertkonstantsannahme . . . . .	5
2.1.2	Die Glattheitsannahme . . . . .	6
2.2	Das Energiefunktional . . . . .	7
2.3	Die Diskretisierung . . . . .	8
2.4	Lösung des Gleichungssystemes . . . . .	10
3	ERWEITERUNGEN	15
3.1	Robustifizierter Datenterm . . . . .	15
3.2	Flussgesteuerter Glattheitsterm . . . . .	15
3.3	Warping . . . . .	17
4	MEHRKANALBILDER	19
4.1	Energiefunktional . . . . .	19
4.2	Normalisierung . . . . .	19
4.3	Gruppierung . . . . .	20
4.4	Kanäle . . . . .	20
4.4.1	RGB . . . . .	20
4.4.2	Gradient . . . . .	21
4.4.3	Hessematrix . . . . .	22
4.4.4	Gradientenbetrag . . . . .	22
4.4.5	Spur der Hessematrix . . . . .	22
4.4.6	Determinante der Hessematrix . . . . .	23
4.4.7	Log-Gradient . . . . .	23
4.4.8	HSV-Farbraum . . . . .	23
5	EVALUATION	25
5.1	Benchmarks . . . . .	25
5.2	Fehlerwerte . . . . .	30
5.3	Normalisierung . . . . .	30
5.4	Konstanzannahmen für KITTI . . . . .	31
5.5	Kombination von Konstanzannahmen . . . . .	32
6	ZUSAMMENFASSUNG UND AUSBLICK	35
6.1	Zusammenfassung . . . . .	35

6.2 Ausblick . . . . .	35
LITERATUR	37
ERKLÄRUNG	39

## 1. EINLEITUNG

---

Der technologische Fortschritt der letzten Jahrzehnte ist vor Allem im Bereich der digitalen Medien enorm. Überall auf der Welt entstehen zu jeder Zeit neue Bilder, Texte und Videos. Diese Arbeit befasst sich ausschließlich mit Videos, jedoch sind auch diese heute so allgegenwärtig und massenhaft, dass ihre Analyse nicht mehr manuell vorgenommen werden kann. Die Menge der Produzierenden und Konsumierenden überragt die Anzahl an Verwaltenden so deutlich, dass automatisierte Methoden dringend benötigt werden.

Die Videoanalyse mit Hilfe des Optischen Flusses wird dabei oft als Vorverarbeitung genutzt und ist deshalb Teil von verschiedensten Algorithmen. Egal, ob Videos analysiert werden sollen um sie in einer Mediendatenbank für eine Suchfunktion vorzubereiten zu können, oder ob es sich um Daten von Überwachungskameras handelt, in denen nach Verhaltensmustern gesucht werden soll, der Optische Fluss liefert Unmengen nützlicher Informationen. Und genau wegen dieser vielseitigen Anwendbarkeit und der Notwendigkeit solcher Verfahren ist es wichtig sie zu optimieren. Je präziser der Optische Fluss, desto besser funktionieren die Algorithmen, die darauf aufbauen.

### 1.1 MOTIVATION

Wie gerade erwähnt, hat der Optische Fluss Unmengen an Anwendungsmöglichkeiten. Woher kommt nun diese Vielseitigkeit? Der Optische Fluss ist ein Vektorfeld, das sich aus zwei Bildern einer Videosequenz berechnet. Dieses Vektorfeld gibt für jedes Pixel an, wie es sich zwischen den zwei Bildern bewegt hat. Wenn man die Visualisierung dieses Flussfeldes betrachtet [1.1](#), erkennt man schnell, wie sich ein Gesamtbild ergibt, das alle Bewegungen innerhalb der Sequenz aufschlüsselt. Im schon erwähnten Beispiel einer Überwachungskamera ist es mit Hilfe des Optischen Flusses ein leichtes sich bewegende Objekte vor einem starren Hintergrund zu lokalisieren.

Die Möglichkeiten, die sich bieten, um die aktuellen Algorithmen, zur Berechnung des Optischen Flusses, zu verändern und hoffentlich zu verbessern, sind dabei enorm. Diese Arbeit fokussiert sich darauf unterschiedliche Konstanzannahmen zur Berechnung zu evaluieren und kombinieren. Konstanzannahmen sind Mutmaßungen, die es uns ermöglichen Pixel im nächsten Bild der Sequenz wiederzufinden und somit die Bewegung zu berechnen. Besonders interessant ist dabei herauszufinden, welche Konstanzannahmen in welchen Situationen die besten Ergebnisse liefern und wie man die durchaus unterschiedlichen Aussagen der Varianten kompatibel macht.

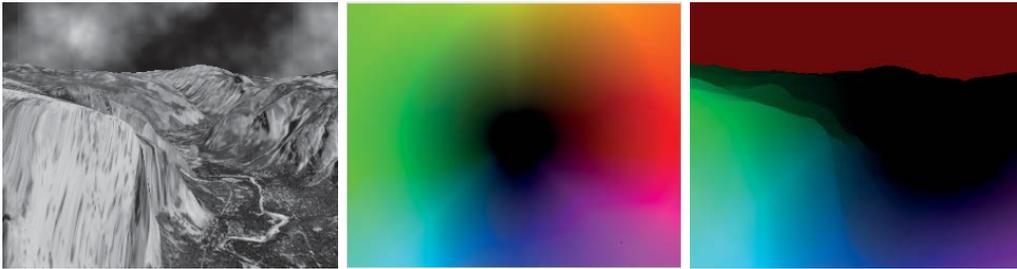


Abbildung 1.1: **Links:**Eines der beide Originalbilder für die Berechnung. **Mitte:**Die Farb-Richtungs-Codierung. **Rechts:**Das Optische Flussfeld für die Sequenz.

## 1.2 AUFGABENSTELLUNG

Ziel der Arbeit ist es deshalb zunächst, ein solch allgemeines Verfahren für Mehrkanalbilder zu implementieren. Dies soll auf Basis des Verfahrens von Brox et al.[1] geschehen. Anschließend sollen verschiedene Konstanzannahmen, als Kanäle, implementiert und bezüglich ihrer Nützlichkeit für Standardbenchmarks (z.B. Middlebury, KITTI) untersucht werden. Hierbei spielt insbesondere auch die Normalisierung der jeweiligen Kanäle (in Form von Datentermen) eine wichtige Rolle. Die Gewichtung soll hierbei a priori und global erfolgen.

## 1.3 VERWANDTE ARBEITEN

Ganz abgesehen davon, dass es noch deutlich mehr Versuche gibt, die Berechnung des Optischen Flusses zu verbessern, baut auch die Version, die in dieser Arbeit besprochen wird, auf den Ergebnissen anderer auf.

Als erstes zu nennen ist die Forschung von Horn und Schunck[2]. Sie waren mit die ersten, die sich der Berechnung und Anwendung des Optischen Flusses in der Informatik beschäftigt haben und die Erfinder der variationalen Berechnungsmethode, die als Grundlage dieser Arbeit dient.

Neuere Forschung wurde gerade im Bereich der photometrischen Invarianz in Betracht gezogen. Die Arbeit von Mileva et al.[3] hat einen Ansatz präsentiert, einen photometrisch invarianten variationalen Algorithmus zu erzeugen, der auf Farbinformationen oder auch Normalisierungsstrategien aufbaut.

Ein Vergleich zwischen unterschiedlichen Ansätzen zur Verbesserung der photometrischen Invarianz wird von Vogel et al.[4] geliefert. Hierbei werden unterschiedliche pixel- und blockbasierte Annahmen für den Datenterm verglichen.

Die Arbeit von Brox et al.[1] präsentiert die Ergebnisse eines Algorithmus, der mehrere Konstanzannahmen für die Berechnung des optischen Flusses kombiniert. Zusätzlich wird mit Vermeidung von linearisierten Konstanzannahmen die Berechnung von großen Verschiebungen ermöglicht und das bis dahin eher experimentelle Warping-

modell fundiert.

Es gibt noch weitere Arbeiten zum Thema „Verwendung mehrerer Konstanzannahmen“. Zum Beispiel verwendet die Arbeit von Kim et al.[5] einen Datenterm, der lokal variiert und aus verschiedenen Datenmodellen die passenden herausucht.

Eine andere Arbeit zum Thema große Verschiebung ist von Sevilla-Lara et al.[6]. Um große Verschiebungen zu ermöglichen, wird die Bildsequenz in verschiedene Levelsets als Kanäle zerlegt. Auf diese Art und Weise bleiben kleine Strukturen im Warping besser erhalten.

#### 1.4 GLIEDERUNG

In Kapitel 2 werden die Grundlagen des Optischen Flusses und des Algorithmus von Horn und Schunck erläutert. Des Weiteren wird gezeigt wie dieser Algorithmus auf diskrete Daten angewandt wird und wie die resultierenden Gleichungssysteme mit iterativen Lösungsverfahren approximiert werden.

Kapitel 3 befasst sich mit der Erweiterung des Algorithmus' durch robuste Datenterme, flussgesteuerte Glattheitsterme und Warping. Damit ist der Algorithmus für Graustufenbilder erklärt, der im Verlauf der Arbeit als Basis dient.

Der eigentliche Kern der Arbeit wird in Kapitel 4 dargestellt. Es wird erklärt, wie man den bisherigen Algorithmus für Mehrkanalbilder erweitert und wie man unterschiedliche Kanäle normalisieren und gruppieren kann, um sie kompatibel zu machen. Zusätzlich werden die verwendeten Kanäle in ihrer Implementierung und Funktion erklärt.

Kapitel 5 ist den Ergebnissen der Experimente gewidmet. Es wird dargestellt mit welchen Parameterkonstruktionen man die besten Ergebnisse erzielen kann und in welchen Situationen welche Kanäle gut funktionieren.

Zum Abschluss liefert Kapitel 6 noch eine Zusammenfassung und spricht einige Möglichkeiten an, wie man dieses Thema noch erweitern kann.



## 2. GRUNDLAGEN VON HORN UND SCHUNCK

---

Um sich damit befassen zu können, was in dieser Arbeit erforscht wird, muss man natürlich erst mal einen Überblick über den verwendeten Stand der Forschung erhalten. In diesem Fall handelt es sich um den Forschungsbereich des Optischen Flusses. Präziser, es betrifft den Algorithmus von Horn und Schunck[2] zur Berechnung des Optischen Flusses.

In dieser Arbeit geht es zwar darum, den Optischen Fluss für Bildsequenzen mit mehreren Kanälen zu berechnen, allerdings sind die mathematischen Grundlagen für alle Kanäle die gleichen. Ein Kanal ist eine Funktion, die jedem Pixel einen Wert zuweist. Es ist quasi ein Graustufenbild, dessen Definitionsbereich sich nicht auf 0 bis 255 limitieren lässt. Neben den sehr üblichen Kanälen R, G und B, aus welchen ein Farbbild besteht, werden noch andere verwendet, die allerdings noch im Detail erklärt werden. Die folgenden Erklärungen gehen von zwei Graustufenbildern aus. Diese werden durch die mathematische Funktion  $f(x, y, t)$  gegeben. Die Parameter  $x$  und  $y$  geben den Bildpunkt an, während  $t$  angibt um welches Bild es sich handelt. Unsere beiden Bilder sind ein Ausschnitt aus einer Bildsequenz. Deshalb ist  $t$  das erste und  $t + 1$  das zweite Bild.

### 2.1 OPTISCHER FLUSS

Der Optische Fluss ist ein Bewegungsfeld, das zwischen zwei Bildern einer Bildsequenz berechnet wird. Er gibt dabei für jeden Bildpunkt eine Richtung und Distanz an, wie er sich zwischen den Bildern bewegt hat. Visualisiert wird er üblicherweise über eine Art Heatmap. Die Richtung wird dabei über die Farbe kodiert und die Distanz befindet sich in der Helligkeit.

Der Optische Fluss wird üblicherweise mit  $u$  und  $v$  bezeichnet. Diese zwei Funktionen geben für jeden Punkt des Bildes den Optischen Fluss, die Verschiebung in  $x$ - und  $y$ -Richtung an.

#### 2.1.1 Die Grauwertkonstantensannahme

Die einfachste Annahme zur Berechnung des Optischen Flusses, ist die Grauwertkonstantensannahme. Diese kann wie folgt mathematisch formuliert werden.

$$f(x, y, t) = f(x + u, y + v, t + 1)$$

Diese Annahme besagt, dass ein Punkt in Bild 1 den gleichen Grauwert haben soll wie der ihm über den Optischen Fluss zugeordneten Punkt in Bild 2. Man kann diese

Gleichung mit Hilfe einer Taylorentwicklung der rechten Seite umformen. Diese Version ermöglicht es  $u$  und  $v$  analytisch zu berechnen, ohne den ganzen Bildbereich nach einem passenden Partner in Bild zwei abzusuchen.

$$f_x u + f_y v + f_t = 0$$

Diese Gleichung bestimmt aus Änderung des Bildes in  $x$ - und  $y$ -Richtung sowie der zeitlichen Änderung des zu betrachtenden Punktes den Optischen Fluss. Die Ergebnisse der Linearisierung sind zwar nur verlässlich, wenn die Bewegungen im Verhältnis zur Glattheit des Bildes klein sind, dies ist jedoch kein Problem, da der Berechnung eine Weichzeichnung vorangestellt wird. Es wird hier ein Tiefpassfilter mit einem Gauß Kern angewendet.

Die Ergebnisse, die man erzielt, wenn man den Optischen Flow nur mit der Grauwertkonstanzannahme berechnet haben jedoch eine eindeutige Schwäche. Sie ziehen nur die Informationen eines einzelnen Punktes zur Berechnung des Flusses an einer Stelle in Betracht. Bei Kanten kann die Bewegung orthogonal zu ihrem Verlauf zwar, nicht jedoch die Bewegung entlang ihres Verlaufes, durch die Beobachtung eines einzelnen Punktes bestimmt werden. In konstanten Bereichen im Bild fehlen sämtlichen Informationen, die zu einer Berechnung des Flusses nötig sind. Diese Uneindeutigkeit, die daraus entsteht, dass die bisherige Annahme aus einer Gleichung mit zwei Unbekannten besteht, nennt man Aperturproblem.

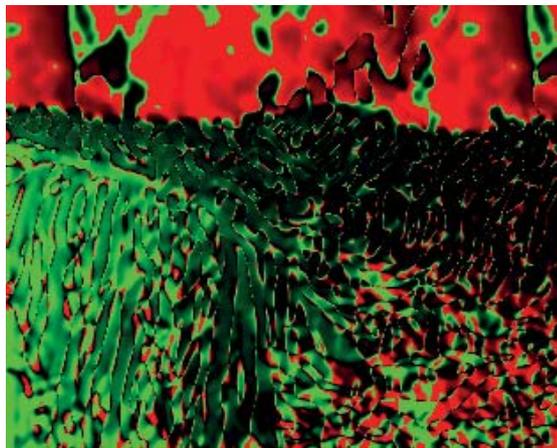


Abbildung 2.1: Optischer Fluss auf ausschließlicher Basis der Grauwertkonstanzannahme.

### 2.1.2 Die Glattheitsannahme

Um die benötigten Informationen einzufüllen, trifft man eine weitere Annahme.

$$|\nabla u|^2 + |\nabla v|^2 = 0$$

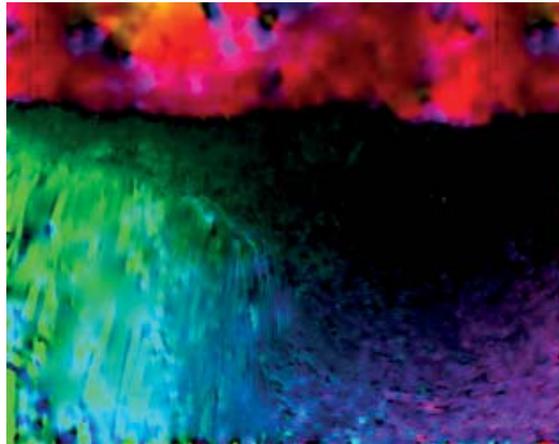


Abbildung 2.2: Optischer Fluss berechnet mit der Kombination aus Grauwertkonstanz- und Glattheitsannahme.

Diese Annahme verlangt, dass sich das Optische Flussfeld lokal möglichst nicht ändert. Die Überlegung dahinter ist, dass sich Objekte, die ganze Flächen im Bild einnehmen, gleich bewegen und dementsprechend der Fluss in der ganzen Fläche einheitlich ist.

Mit Hilfe dieser Annahme ist es möglich, die Informationen einzufüllen, die aus der Grauwertkonstanzannahme nicht hervorgehen. Innerhalb einer konstanten Fläche kann man somit die Bewegung der nächstgelegenen Kante übernehmen. Allerdings versucht die Glattheitsannahme auch die Bewegungskanten zwischen Objekten zu glätten, weshalb adaptive Glattheitsfunktionen benötigen werden um brauchbare Ergebnisse zu erzielen.

## 2.2 DAS ENERGIEFUNKTIONAL

Wenn man nun versucht beide Gleichungen zu lösen, fällt einem recht schnell auf, dass dies so nicht möglich ist. Die 0, die auf der rechten Seite der Gleichung steht, wird in der Realität praktisch nie erreicht. Bei der Grauwertkonstanzannahme liegt es meist an Änderungen an der Beleuchtung, der Reflexion oder daran, dass sich Objekte gegenseitig verdecken. Die Glattheitsannahme ergibt nur dann tatsächlich 0, wenn sich die gesamte Fläche um die gleiche Verschiebung in die selbe Richtung bewegt hat. Auch dies ist in einer klassischen dreidimensionalen Szene mit mehreren Objekten eher unwahrscheinlich.

Deshalb fasst man die beiden Gleichungen in ein Energiefunktional zusammen, in dem man sie gemeinsam minimiert.

$$E(u, v) = \int \int_{x, y} \underbrace{(f_x u + f_y v + f_t)^2}_{\text{Datenterm}} + \alpha \underbrace{(|\nabla u|^2 + |\nabla v|^2)}_{\text{Glattheitsterm}} dx dy$$

Den Datenterm genannte Teil dieser Gleichung bildet die Grauwertkonstantenannahme. Das Quadrat sorgt dafür, dass der Term positiv ist und somit positive und negative Abweichungen gleich behandelt werden. Der Parameter  $\alpha$  ist ein Gewichtungsfaktor, der es einem ermöglicht die Relevanz des Daten- und des Glattheitsterms einzustellen. Die Glattheitsannahme kann direkt übernommen werden, da sie durch die Quadrierung der Einzelterme schon positiv ist.

Mit Hilfe des Bewegungstensors, der eine Form des Strukturtenors ist, kann die spätere Diskretisierung einfacher gemacht werden.

$$J = \nabla f \nabla f^T = \begin{pmatrix} f_x^2 & f_x f_y & f_x f_t \\ f_x f_y & f_y^2 & f_y f_t \\ f_x f_t & f_y f_t & f_t^2 \end{pmatrix}$$

Mit  $\mathbf{w} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$  kann der Datenterm nun wie folgt geschrieben werden.

$$(f_x u + f_y v + f_t)^2 = \mathbf{w}^T J \mathbf{w}$$

Um das Energiefunktional zu minimieren, müssen die Euler-Lagrange-Gleichungen als notwendige Bedingungen gelöst werden.

$$0 \stackrel{!}{=} F_u - \frac{\delta}{\delta x} F_{u_x} - \frac{\delta}{\delta y} F_{u_y}$$

$$0 \stackrel{!}{=} F_v - \frac{\delta}{\delta x} F_{v_x} - \frac{\delta}{\delta y} F_{v_y}$$

mit den Neumann-Randbedingungen  $\mathbf{n}^T \nabla u = 0$  und  $\mathbf{n}^T \nabla v = 0$ , wobei der Richtungsvektor  $\mathbf{n}$  über den Bildrand hinweg zeigt. In diesem Fall ergeben sich somit die folgenden Gleichungen.

$$0 = J_{11}u + J_{12}v + J_{13} - \alpha(u_{xx} + u_{yy})$$

$$0 = J_{12}u + J_{22}v + J_{23} - \alpha(v_{xx} + v_{yy})$$

### 2.3 DIE DISKRETISIERUNG

Die Diskretisierung der Gleichungen ist nötig um sie später numerisch lösen zu können. Der Großteil passiert hier über die Diskretisierung der Funktionen, beispielhaft hier an der Bildfunktion dargestellt.

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

Mit  $i = 1, \dots, N$ ,  $j = 1, \dots, M$  und  $t = 1, 2$ . Hierbei geben  $h_x$  und  $h_y$  den Gitterabstand der Pixel an und resultieren aus der Abtastrate. Auf die gleiche Art und Weise verfährt man mit  $u$ ,  $v$  und dem Bewegungstensor.

$$u_{i,j} = u(i \cdot h_x, j \cdot h_y)$$

$$v_{i,j} = v(i \cdot h_x, j \cdot h_y)$$

$$J_{i,j} = J(i \cdot h_x, j \cdot h_y)$$

Für die Berechnung des Bewegungstensors werden die Ableitungen mit finiten Differenzen diskretisiert. Bei  $f_x$  und  $f_y$  werden gemittelte zentrale Differenzen benutzt.

$$[f_x]_{i,j} = \frac{1}{2} \left( \frac{f_{i+1,j,t+1} - f_{i-1,j,t+1}}{2h_x} + \frac{f_{i+1,j,t} - f_{i-1,j,t}}{2h_x} \right)$$

$$[f_y]_{i,j} = \frac{1}{2} \left( \frac{f_{i,j+1,t+1} - f_{i,j-1,t+1}}{2h_y} + \frac{f_{i,j+1,t} - f_{i,j-1,t}}{2h_y} \right)$$

während für  $f_t$  wird eine normale vorwärts Differenz benutzt:

$$[f_t]_{i,j} = \frac{f_{i,j,t+1} - f_{i,j,t}}{h_t}.$$

Für  $h_t$  wird üblicherweise der Wert 1 angenommen.

Schließlich müssen auch noch  $\Delta u = u_{xx} + u_{yy}$  und  $\Delta v = v_{xx} + v_{yy}$  diskretisiert werden. Auch dies geschieht wieder über finite Differenzen:

$$\begin{aligned} \Delta u &= (u_x)_x + (u_y)_y \\ &\approx \frac{(u_x)_{i+\frac{1}{2},j} - (u_x)_{i-\frac{1}{2},j}}{2(\frac{1}{2}h_x)} + \frac{(u_y)_{i,j+\frac{1}{2}} - (u_y)_{i,j-\frac{1}{2}}}{2(\frac{1}{2}h_y)} \\ &\approx \frac{\frac{u_{i+1,j} - u_{i,j}}{2(\frac{1}{2}h_x)} - \frac{u_{i,j} - u_{i-1,j}}{2(\frac{1}{2}h_x)}}{2(\frac{1}{2}h_x)} + \frac{\frac{u_{i,j+1} - u_{i,j}}{2(\frac{1}{2}h_y)} - \frac{u_{i,j} - u_{i,j-1}}{2(\frac{1}{2}h_y)}}{2(\frac{1}{2}h_y)} \\ &= \frac{u_{i+1,j} - u_{i,j}}{h_x^2} - \frac{u_{i,j} - u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - u_{i,j}}{h_y^2} - \frac{u_{i,j} - u_{i,j-1}}{h_y^2}. \end{aligned}$$

die Diskretisierung von  $\Delta v$  formt man auf die gleiche Art und Weise um:

$$\Delta v = \frac{v_{i+1,j} - v_{i,j}}{h_x^2} - \frac{v_{i,j} - v_{i-1,j}}{h_x^2} + \frac{v_{i,j+1} - v_{i,j}}{h_y^2} - \frac{v_{i,j} - v_{i,j-1}}{h_y^2}.$$

Man kann an den zwei finalen Gleichungen erkennen, wie jedes Pixel separat, mit allen vier Nachbarn verrechnet wird. Die Neumann-Randbedingung, sagt uns, dass Ableitungen über den Bildrand verschwinden. An solchen Stellen muss man ein Pixel

nur mit den zwei, oder drei übrigen Nachbarn verrechnen.  
Die fertigen diskretisierten Euler-Lagrange-Gleichungen sind:

$$0 = [J_{11}]_{i,j} u_{i,j} + [J_{12}]_{i,j} v_{i,j} + [J_{13}]_{i,j} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l(i,j)} \frac{u_{\tilde{i}, \tilde{j}} - u_{i,j}}{h_l^2}$$

$$0 = [J_{12}]_{i,j} u_{i,j} + [J_{22}]_{i,j} v_{i,j} + [J_{23}]_{i,j} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l(i,j)} \frac{v_{\tilde{i}, \tilde{j}} - v_{i,j}}{h_l^2}$$

für  $i = 1, \dots, N$  und  $j = 1, \dots, M$ .

Hierbei steht  $N_l(i, j)$  für alle Nachbarn des Pixels  $(i, j)$  in Richtung  $l$ . im 2-D gibt es in jeder Dimension minimal ein und maximal zwei Nachbarn.

#### 2.4 LÖSUNG DES GLEICHUNGSSYSTEMES

Insgesamt entsteht ein lineares Gleichungssystem mit den  $2 \cdot (N \cdot M)$  unbekanntem  $u(i, j)$  und  $v(i, j)$ . Auf Grund der Bildgröße handelt es sich um ein sehr großes Gleichungssystem, das jedoch recht dünn besetzt ist. Somit bieten sich iterative Lösungsverfahren deutlich eher an als eine direkte Lösung über Beispielsweise eine Gauß-Elimination. Ein iteratives Lösungsverfahren betrachtet pro Berechnungsschritt nicht alle Einträge der Matrix. Deshalb sind die einzelnen Schritte extrem schnell, die Lösung ist jedoch nur eine grobe Approximation. Um diese zu verbessern, wiederholt man die Berechnung mit den Ergebnissen des vorhergehenden Durchganges mehrere Male. Je öfters man die Berechnung wiederholt, desto näher kommt die Approximation dem tatsächlichen Ergebnis.

Lineare Gleichungssysteme lassen sich als Matrix-Vektor-Multiplikation darstellen.

$$Ax = b$$

Wobei  $A$  die Werte der Gleichungen enthält,  $x$  aus den zugehörigen Parametern besteht und in  $b$  jeweils die Ergebnisse stehen. Die Idee ist nun die Matrix  $A$  geschickt aufzuspalten um das Invertieren der Originalmatrix zu umgehen.

$$A = A_1 + A_2 \Rightarrow (A_1 + A_2)x = b \Leftrightarrow A_1x = b - A_2x$$

Nach dem Einführen einer Fixpunktiteration erhält man:

$$A_1x^{k+1} = b - A_2x^k \Leftrightarrow x^{k+1} = A_1^{-1}(b - A_2x^k)$$

$A_1$  sollte einfach zu invertieren sein um hierdurch einen Vorteil erzielen zu können. Das Verfahren, das hier benutzt wird, zerlegt  $A$  in die drei Teile  $D$ ,  $L$  und  $R$ .

- $D$  enthält nur die Diagonale von  $A$

- L enthält sämtliche Einträge links unterhalb der Diagonale
- R enthält sämtliche Einträge rechts oberhalb der Diagonale

Das einfachste Verfahren, das auf dieser Zerlegung basiert, ist das Jacobi Verfahren. Hierbei ist  $A_1 = D$  und  $A_2 = -L - U$ .

$$x^{k+1} = D^{-1}(b + (L + U)x^k)$$

Da D eine Diagonalmatrix ist, ist sie leicht zu invertieren. Auf das zu lösende Gleichungssystem angewendet erhält man diese Berechnungsvorschriften.

$$u_{i,j}^{k+1} = \frac{-[J_{13}]_{i,j} - [J_{12}]_{i,j}v_{i,j}^k + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^-(i,j)} \frac{u_{\tilde{i}, \tilde{j}}^k}{h_l^2}}{[J_{11}]_{i,j} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^-(i,j)} \frac{1}{h_l^2}}$$

$$v_{i,j}^{k+1} = \frac{-[J_{23}]_{i,j} - [J_{22}]_{i,j}u_{i,j}^k + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^+(i,j)} \frac{v_{\tilde{i}, \tilde{j}}^k}{h_l^2}}{[J_{22}]_{i,j} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^+(i,j)} \frac{1}{h_l^2}}$$

Die Berechnung von  $u_{i,j}^{k+1}$  und  $v_{i,j}^{k+1}$  basiert nun nur auf Werten der vorhergehenden Iteration k.

Das nächst einfache Verfahren, ist die Gauß-Seidel-Methode. Hierbei verwendet man

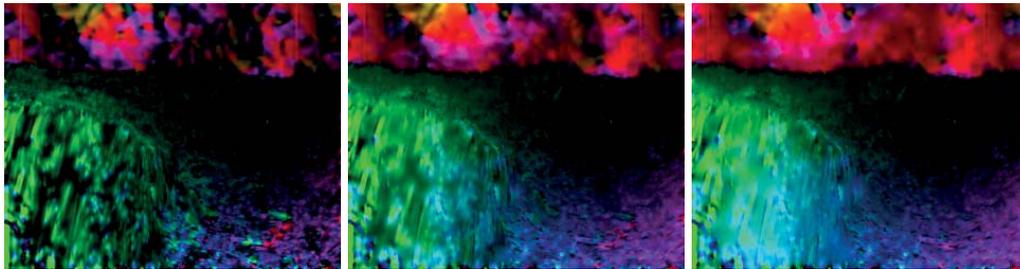


Abbildung 2.3: Berechnet mit der Gauß-Seidel-Methode und unterschiedlichen Anzahlen an Iterationen. **V.l.n.R.:** 10, 50 und 200 Iterationen

$A_1 = D - L$  und  $A_2 = -R$ . Die Dreiecksmatrix  $D - L$  ist eine bessere Darstellung von A, wodurch die Ergebnisse an Qualität gewinnen. Jedoch sind Dreiecksmatrizen immer noch leicht zu invertieren.

$$x^{k+1} = (D - L)^{-1}(b + Ux^k)$$

Die Berechnungsvorschriften ändern sich hierdurch wie folgt.

$$u_{i,j}^{k+1} = \frac{-[J_{13}]_{i,j} - [J_{12}]_{i,j}v_{i,j}^k + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^-(i,j)} \frac{u_{\tilde{i}, \tilde{j}}^{k+1}}{h_l^2} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^+(i,j)} \frac{u_{\tilde{i}, \tilde{j}}^k}{h_l^2}}{[J_{11}]_{i,j} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^-(i,j)} \frac{1}{h_l^2}}$$

$$v_{i,j}^{k+1} = \frac{-[J_{23}]_{i,j} - [J_{12}]_{i,j} u_{i,j}^{k+1} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^-(i,j)} \frac{v_{\tilde{i}, \tilde{j}}^{k+1}}{h_l^2} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^+(i,j)} \frac{v_{\tilde{i}, \tilde{j}}^k}{h_l^2}}{[J_{22}]_{i,j} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l(i,j)} \frac{1}{h_l^2}}$$

Hierbei bezeichnet  $N_l^-(i, j)$  die Nachbarschaft in Richtung  $l$ , die schon berechnet worden ist und  $N_l^+(i, j)$  die Nachbarschaft in Richtung  $l$ , die noch nicht berechnet worden sind.

Als zusätzlichen Vorteil, benötigt das Gauß-Seidel-Verfahren weniger Speicherplatz bei der Berechnung, da man nicht mehr getrennte Arrays für die Schritte  $k$  und  $k + 1$  benötigt, sondern nur ein Array braucht, in das die Ergebnisse der Berechnung gleich eingetragen werden können.

Schlussendlich wird hier jedoch das Successive Over-Relaxation Verfahren[7], kurz *SOR-Verfahren*, verwendet. Hierfür muss noch der Überrelaxationsparameter  $\omega \in (0, 2)$  eingeführt werden.

$$x^{k+1} = (D + \omega L)^{-1}(\omega b - (\omega R + (\omega - 1)D)x^k)$$

Dieser neue Parameter beschleunigt die Konvergenz des Verfahrens. Im Fall  $\omega = 1$



Abbildung 2.4: Optischer Fluss nach der Berechnung mit dem Horn und Schunck Verfahren. Lösung der Gleichungssysteme via *SOR*.

handelt es sich wiederum um das Gauß-Seidel-Verfahren. Die Berechnungsvorschriften sehen nun wie folgt aus.

$$u_{i,j}^{k+\frac{1}{2}} = \frac{-[J_{13}]_{i,j} - [J_{12}]_{i,j} v_{i,j}^k + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^-(i,j)} \frac{u_{\tilde{i}, \tilde{j}}^{k+\frac{1}{2}}}{h_l^2} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^+(i,j)} \frac{u_{\tilde{i}, \tilde{j}}^k}{h_l^2}}{[J_{11}]_{i,j} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l(i,j)} \frac{1}{h_l^2}}$$

$$u_{i,j}^{k+1} = (1 - \omega)u_{i,j}^k + \omega u_{i,j}^{k+\frac{1}{2}}$$

$$v_{i,j}^{k+\frac{1}{2}} = \frac{-[J_{23}]_{i,j} - [J_{12}]_{i,j} u_{i,j}^{k+1} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^-(i,j)} \frac{v_{\tilde{i}, \tilde{j}}^{k+1}}{h_l^2} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l^+(i,j)} \frac{v_{\tilde{i}, \tilde{j}}^k}{h_l^2}}{[J_{22}]_{i,j} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i}, \tilde{j}) \in N_l(i,j)} \frac{1}{h_l^2}}$$

$$v_{i,j}^{k+1} = (1 - \omega)v_{i,j}^k + \omega v_{i,j}^{k+\frac{1}{2}}$$



### 3. ERWEITERUNGEN

---

#### 3.1 ROBUSTIFIZIERTER DATENTERM

Im Gegensatz zum Horn und Schunck Verfahren, wird hier eine robuste Variante[8] des Datenterms benutzt.

$$E(u, v) = \int \int_{x, y} \underbrace{\psi_D(\mathbf{w}^T \mathbf{J} \mathbf{w}, \epsilon)}_{\text{Datenterm}} + \alpha \underbrace{(|\nabla u|^2 + |\nabla v|^2)}_{\text{Glattheitsterm}} dx dy,$$

wobei  $\psi_D$  durch die Funktion

$$\psi_D(x^2, \epsilon) = \sqrt{1 + \frac{x^2}{\epsilon^2}}$$

gegeben ist. Dieser Funktion reduziert den Einfluss von Ausreißern, indem Abweichungen im Datenterm linear bestraft werden. Als Euler-Lagrange-Gleichungen erhält man dieses nichtlinearen Gleichungssystem

$$0 = [\psi'_D]_{i,j} ([J_{11}]_{i,j} u_{i,j} + [J_{12}]_{i,j} v_{i,j} + [J_{13}]_{i,j}) - \alpha \sum_{l \in x, y} \sum_{(\tilde{i}, \tilde{j}) \in N_l(i, j)} \frac{u_{\tilde{i}, \tilde{j}} - u_{i,j}}{h_l^2}$$

$$0 = [\psi'_D]_{i,j} ([J_{12}]_{i,j} u_{i,j} + [J_{22}]_{i,j} v_{i,j} + [J_{23}]_{i,j}) - \alpha \sum_{l \in x, y} \sum_{(\tilde{i}, \tilde{j}) \in N_l(i, j)} \frac{v_{\tilde{i}, \tilde{j}} - v_{i,j}}{h_l^2}$$

für  $i = 1, \dots, N$  und  $j = 1, \dots, M$  mit

$$[\psi'_D]_{i,j} = \psi'_D(\omega_{i,j}^T \mathbf{J}_{i,j} \omega_{i,j}, \epsilon)$$

Da es deutlich einfacher ist, lineare Gleichungssysteme zu lösen, wird die Nichtlinearität mit Hilfe einer Fixpunktiteration behoben. Hierbei wird das  $\psi'_D$  für jeden Punkt des Bildes berechnet und mit diesem festen  $\psi'_D$  werden mehrere Iterationen des nun wieder linearen Gleichungssystems berechnet. Hiermit wird die nichtlineare Lösung approximiert. Da die Berechnung des nichtlinearen Gleichungssystems durch eine Reihe von Berechnungen von linearen Systemen umgesetzt wird, bleibt die ursprüngliche Iteration im Wesentlichen erhalten. Der gesamte Vorgang der  $\psi'_D$  Berechnung und wiederholter Lösung der Systeme mit festem  $\psi'_D$  muss mehrfach iteriert werden.

#### 3.2 FLUSSGESTEUERTER GLATTHEITSTERM

Ähnlich wie der Datenterm ist auch der Glattheitsterm anfällig für Ausreißer. Da dieser Term versucht die Glattheit des Optischen Flusses zu erzwingen, werden Kanten



Abbildung 3.1: Optisches Flussfeld berechnet mit nun zusätzlich noch  $\psi_G$  und  $\psi_D$ .

und Ecken sehr unscharf. Um dies zu verhindern werden Bewegungskanten für das Energiefunktional günstiger gemacht.

$$E(u, v) = \int \int_{x, y} \underbrace{\psi_D(\mathbf{w}^T \mathbf{J} \mathbf{w}, \epsilon)}_{\text{Datenterm}} + \alpha \underbrace{\psi_G(|\nabla u|^2 + |\nabla v|^2)}_{\text{Glattheitsterm}} dx dy$$

Auch hier wird für  $\psi_G$  die Funktion

$$\psi_G(x^2, \epsilon) = \sqrt{1 + \frac{x^2}{\epsilon^2}}$$

genutzt. Da auch der Einbau dieser Funktion ein nichtlinear Gleichungssystem ergibt, wird wie bei der robustifizierung des Datenterms auf eine Fixpunktiteration zurückgegriffen. Mittels einsetzen in die Euler-Lagrange-Gleichungen erhalten wir nun:

$$0 = [\psi'_D]_{i,j} (\mathbf{w}_{i,j}^T \mathbf{J}_{i,j} \mathbf{w}_{i,j}) - \alpha \sum_{l \in x, y} \sum_{(\tilde{i}, \tilde{j}) \in N_l(i, j)} \frac{[\psi'_G]_{\tilde{i}, \tilde{j}} + [\psi'_G]_{i, j}}{2} \left( \frac{u_{\tilde{i}, \tilde{j}} - u_{i, j}}{h_l^2} \right)$$

$$0 = [\psi'_D]_{i,j} (\mathbf{w}_{i,j}^T \mathbf{J}_{i,j} \mathbf{w}_{i,j}) - \alpha \sum_{l \in x, y} \sum_{(\tilde{i}, \tilde{j}) \in N_l(i, j)} \frac{[\psi'_G]_{\tilde{i}, \tilde{j}} + [\psi'_G]_{i, j}}{2} \left( \frac{v_{\tilde{i}, \tilde{j}} - v_{i, j}}{h_l^2} \right).$$

für  $i = 1, \dots, N$  und  $j = 1, \dots, M$  mit

$$[\psi'_G]_{i,j} = \psi'_G(|\nabla u|_{i,j}^2 + |\nabla v|_{i,j}^2, \epsilon) = \psi'_G([u_x]_{i,j}^2 + [u_y]_{i,j}^2 + [v_x]_{i,j}^2 + [v_y]_{i,j}^2).$$

Die Ableitungen von  $u$  und  $v$  werden auch hier mittels finiter Differenzen approximiert

$$\begin{aligned} [u_x]_{i,j} &= \left( \frac{u_{i+1,j} - u_{i-1,j}}{2h_x} \right) & [u_x]_{i,j} &= \left( \frac{u_{i,j+1} - u_{i,j-1}}{2h_x} \right) \\ [v_x]_{i,j} &= \left( \frac{v_{i+1,j} - v_{i-1,j}}{2h_x} \right) & [v_x]_{i,j} &= \left( \frac{v_{i,j+1} - v_{i,j-1}}{2h_x} \right). \end{aligned}$$

## 3.3 WARPING

Um die Gleichung optimal Minimieren zu können, ist das Verwenden von Details im Bild wichtig. Besonders bei sehr feinen oder repetitiven Strukturen ist die Aussage dieser Details meist nicht Eindeutig. Der bisherige Algorithmus tendiert dazu bei der Zuweisung der Pixel zwischen den Bildern möglichst kurze Bewegungen zu erhalten. Es wird das nächst gelegene lokale Minimum gesucht. Dabei ist oft der Fall, dass die Zuweisung einen ähnlichen Punkt im zweiten Bild findet, jedoch nicht das am besten passenden Pixel entdeckt. Dieses wäre das globale Minimum des Energiefunktional. Kurz gefasst: Es werden Details benötigt um die Verschiebung akkurat zu berechnen, jedoch verhindern diese, dass in detailreichen Bereichen große Verschiebungen berechnet werden können. Um dies zu beheben wird das Warping benutzt.

Die eine Hälfte des Warping ist das außer Acht lassen von Details mit unterschiedlichen Auflösungsstufen. Dafür wird eine so genannte grob-nach-fein Berechnung durchgeführt. Der Algorithmus wird auf einer unscharfen herunter-skalierten Version des Originalbildes gestartet und arbeitet sich in einigen Iterationen, immer schärfer werdend, zum Originalbild vor. Durch die Diffusion werden Details unkenntlich und die Wahrscheinlichkeit die korrekte Zuweisung zu finden erhöht. Dies liegt daran, dass lokale Minima geglättet werden und somit die Suche nach dem globalen Minimum einfacher wird. Die geringere Auflösung des Bildpaares ermöglicht es große Verschiebungen im Bildbereich zu finden, da es sich nun um kleine Verschiebungen handelt.

Um das Ergebnis dieser ersten Iteration nun als Initialisierung für die nächste Auf-

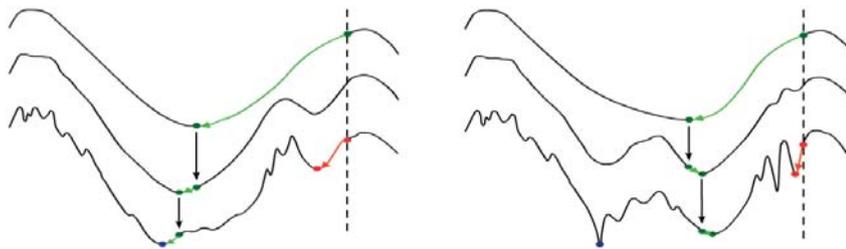


Abbildung 3.2: Visualisierung der Auflösungsstufen und wie sie die Suche des globalen Minimums ermöglichen. **Quelle:** Vorlesung COPCV[9]

lösungsstufe zu benutzen braucht man die andere Hälfte des Warping. Hierbei wird das zweite Bild der Sequenz um den bisherigen Optischen Fluss korrigiert. Da der Optische Fluss den Unterschied zwischen den beiden Bildern approximiert, ist das verzerrte zweite Bild nun fast gleich dem ersten Bild des Paares. Deshalb sind nun die Distanzen zwischen den zueinander gehörenden Strukturen kleiner, wodurch auch in detaillierten Bereichen das Finden des globalen Minimums möglich wird. Um den fertigen Optischen Fluss zu erhalten, müssen die unterschiedlichen Versionen der Auf-

lösungsstufen addiert werden.

Dieser Prozess wird mit der gewünschten Anzahl an Iterationen ausgeführt. Für den Algorithmus liefert das Warming noch zwei weitere Parameter. Der Warpstufen  $n_{\text{warp}}$  genannte Parameter gibt an wie viele Iterationen man berechnen möchte und der Parameter  $\eta$  gibt an um wie viel die Auflösung des Bildes zwischen den Stufen reduziert wird. Durch ein gegebenes  $\eta$  entsteht eine obere Grenze für  $n_{\text{warp}}$ .

## 4. MEHRKANALBILDER

---

Vom Prinzip her ändert sich kaum etwas am bisherigen Algorithmus, wenn man damit den Fluss von Mehrkanalbildern (z.B. RGB) berechnen möchte. Für jeden Kanal muss der Bewegungstensor einzeln berechnet werden. Vor der Berechnung von  $\psi_D$ ,  $u$  und  $v$  müssen diese nur noch gemittelt werden.

Dieses Verfahren reicht aus, wenn man korrelierende Kanäle miteinander verrechnet, wie es zum Beispiel bei den drei Kanälen von RGB-Bildern der Fall ist. Da in dieser Arbeit noch weitere Kanäle eingeführt werden, die andere Datenbereiche und ein unabhängiges Verhalten an den Tag legen, muss man dieses Konzept noch um zwei Punkte erweitern.

### 4.1 ENERGIEFUNKTIONAL

Um mehrere Kanäle in die Berechnung einfließen zu lassen, muss man natürlich das Energiefunktional anpassen. Im Großen und Ganzen wird dadurch der Datenterm zu einer Summe.

$$E(u, v) = \iint_{x,y} \sum_{i=0}^{|C|} \left( \gamma_i \psi_D \left( \sum_{i \in C_j} \mathbf{w}_{i,j}^T J_{i,j} \mathbf{w}_{i,j}, \epsilon \right) \right) + \alpha \psi_G (|\nabla u|^2 + |\nabla v|^2) dx dy$$

hierbei ist  $C$  die Menge an Kanalgruppierungen und  $\gamma_i$  ein zugehöriger Gewichtungsparemeter. Dazu mehr in [4.3](#).

### 4.2 NORMALISIERUNG

In dieser Arbeit wurden zwei unterschiedliche Normalisierungen benutzt und getestet. Die affine Normalisierung adressiert die unterschiedlichen Datenbereiche, indem sie jeden Kanal auf einen gewünschten Wertebereich skaliert.

$$f_{i,j} = (f_{i,j} - \min_{\text{Old}}) * \left( \frac{\max_{\text{New}} - \min_{\text{New}}}{\max_{\text{Old}} - \min_{\text{Old}}} \right) + \min_{\text{New}}$$

Bei  $\max_{\text{Old}}$  und  $\min_{\text{Old}}$  handelt es sich um den größten und kleinsten Wert der im Kanal auftritt, wohingegen  $\max_{\text{New}}$  und  $\min_{\text{New}}$  den größten und kleinsten Wert im gewünschten Bereich angeben.

Die Skalierung in dieser Normalisierung wird von Extremwerten stark dominiert, weshalb die Ableitungen der unterschiedlichen Kanäle immer noch stark unterschiedliche Werte annehmen können. Da in der Berechnung nur die Ableitungen verwendet werden, nimmt diese Normalisierung nur indirekt Einfluss auf die Kompatibilität der

Kanäle.

Die Normalisierung via Standardabweichung wirkt hier wesentlich direkter. Sie versucht das Verhalten, die Varianz in den Daten, zu normalisieren. Hierfür wird für den jeweiligen Kanal die Standardabweichung berechnet und die Datenwerte durch selbige geteilt.

$$f_{i,j} = \frac{f_{i,j}}{\sigma_f}$$

$\sigma_f$  ist hierbei der Wert der Standardabweichung des Kanals.

Wie schon erwähnt, benötigt die Berechnung des  $\psi_D$  einen Parameter  $\epsilon$ . Da dieser direkt mit den Daten verrechnet wird, ist es klug ihn mit der Skalierung der jeweiligen Normalisierung zu verrechnen. Auf diese Weise muss der Parameter für einen Kanal nicht gesondert optimiert werden.

### 4.3 GRUPPIERUNG

Der zweite Punkt, den man bei der Verwendung von so unterschiedlichen Kanälen beachten muss, ist die Gruppierung dieser, bei der Berechnung des  $\psi_D$  und der Normalisierung. In Gruppen sind jeweils ähnliche Kanäle zusammengefasst. Zum Beispiel könnte man die drei RGB-Kanäle in einer und ihre Ableitungen in einer anderen Gruppe bündeln. In einem solchen Fall kann man die Normalisierung und vor Allem die Berechnung der  $\psi_D$  in eben diesen Gruppen durchführen.

Dass die Kanäle für die Normalisierung gruppiert seien müssen ist offensichtlich, da die Normalisierung unterschiedliche Kanäle auf ein kompatibles Format bringen soll. Dass ähnliche Kanäle gemeinsam normalisiert werden liegt dementsprechend daran, dass sie schon von beginn an kompatibel sind. Für die Berechnung des  $\psi_D$  ist die Gruppierung wichtig, da auch hier das unterschiedliche Verhalten der Kanäle ins Gewicht fällt und einem durch die Trennung eine Gewichtung der unterschiedlichen Kanalkombinationen möglich ist.

### 4.4 KANÄLE

Wie schon erwähnt wurde, werden in dieser Ausarbeitung zusätzliche Kanäle verwendet um zu erforschen welchen Einfluss diese auf das Berechnungsergebnis haben. Die Berechnung eines Kanals ist quasi die Aufbereitung des Ursprungsbildes in der Hoffnung andere Informationen hervorheben und aus dem Bild ziehen zu können.

#### 4.4.1 RGB

Als Ursprung aller anderen Kanäle dienen die drei Farbkanäle Rot, Grün und Blau. Sie werden von den Bildern der Sequenz gegeben. Ihre Werte liegen im Bereich zwischen

0 und 255 und da sie von den Inhalten abhängig voneinander sind, können sie in einer Gruppe berechnet werden. Diese drei Kanäle sind auch die einzigen, die im Zuge des Warping skaliert werden. Beim Warping werden die Berechnungen über unterschiedliche Auflösungsstufen iteriert. Bei der Skalierung der Kanäle kann es passieren, dass das gewünschte Verhältnis zwischen einem Kanal und dem Originalbild nicht erhalten bleibt. Deshalb werden nur die drei Ursprungskanäle skaliert und die anderen Kanäle auf jeder Auflösungsstufe neu berechnet.

#### 4.4.2 Gradient

Wie wir schon erwähnt hatten, ist der Gradient ein Vektor, der aus der Ableitung in  $x$ - und in  $y$ -Richtung einer zweidimensionalen Funktion besteht.

$$\nabla f = \begin{pmatrix} f_x \\ f_y \end{pmatrix}$$

Die beiden Einträge des Vektors werden in zwei unterschiedliche Kanäle geschrieben. Da wir drei Ursprungskanäle haben, ergeben sich somit sechs neue Kanäle. Die Ableitungen in  $x$ - und in  $y$ -Richtung jeweils für Rot, Grün und Blau. Das Ziel dieser Kanäle ist eine Robustheit gegenüber Helligkeitsänderungen zu erzeugen.

$$f_{xx}u + f_{xy}v + f_{xt} = 0$$

$$f_{yx}u + f_{yy}v + f_{yt} = 0$$

Diese Kanäle sind nun invariant unter globalen additiven Helligkeitsänderungen. Da wir die Ableitung benutzen fällt ein Wert, der über den gesamten Bildbereich konstant ist, weg.

$$(f(x, y, t) + c)_x = f_x(x, y, t)$$

Als zusätzlichen Bonus hat man nun zwei Gleichungen für die Berechnung von zwei Unbekannten, was das Aperturproblem lindert.

Allerdings haben diese Kanäle auch ihre Nachteile. Sie sind anfälliger für Bildrauschen, da diese starken Änderungen in den Ableitungen Ausreißer ergeben; haben Schwierigkeiten mit Rotationen in der Bildsequenz, da Informationen über die Richtung von Änderungen (Gradient) zur Berechnung verwendet werden; und bezahlen für den Ausgleich der Helligkeitsänderungen mit einem reduzierten Informationsgehalt, da die absoluten Helligkeitswerte nicht mehr in die Berechnung eingehen.

#### 4.4.3 Hessematrix

Die Hessematrix folgt dem gleichen Prinzip, wie der Gradient, allerdings noch eine Ableitungsstufe weiter.

$$H(f) = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix}$$

Wie beim Gradient auch, werden die nun vier Einträge der Hessematrix in vier Kanälen gespeichert. Es entstehen somit zwölf neue Kanäle für Rot, Grün und Blau.

$$f_{xxx}u + f_{xxy}v + f_{xxt} = 0$$

$$f_{xyx}u + f_{xyy}v + f_{xyt} = 0$$

$$f_{yxx}u + f_{yyx}v + f_{yxt} = 0$$

$$f_{yyx}u + f_{yyy}v + f_{yyt} = 0$$

Für die Konstanzannahme ergeben sich nun vier neue Gleichungen. Allerdings sind zwei davon identisch, wodurch jedoch immer noch drei linear unabhängige Gleichungen für zwei Unbekannte übrig bleiben. Die Vor- und Nachteile sind, auf Grund des selben Prinzips, die gleichen wie bei den Gradientkanälen.

#### 4.4.4 Gradientenbetrag

Wie besprochen, haben die Gradientenkanäle die Problematik, dass sie Informationen über die Richtung der Änderung enthalten und deshalb nicht invariant unter Rotationen sind. Um dies zu ändern, werden die Richtungsinformationen ignoriert:

$$|\nabla f|_x u + |\nabla f|_y v + |\nabla f|_t = 0.$$

Da man nun wieder nur eine Gleichung hat, tritt das Aperturproblem wieder verstärkt auf, man erhält jedoch die Invarianz gegen globale additive Beleuchtungsänderungen. Bei dem Gradientenbetrag entsteht wiederum nur ein Kanal pro Farbe.

#### 4.4.5 Spur der Hessematrix

Die Gleiche Problematik, die die Richtungsinformationen bieten, kann man nun auch bei den Hessematrizen lösen. Hierzu ist wichtig zu wissen, dass die Eigenwerte der Matrizen unter Rotation invariant sind. Ignoriert man also die Informationen, die die Eigenvektoren einbringen, erhält man eine ähnliche Lösung jedoch eine Ableitungsstufe höher. Die Spur ist bei Matrizen als die Summe der Eigenwerte definiert.

$$\text{tr}(H(f))_x u + \text{tr}(H(f))_y v + \text{tr}(H(f))_t = 0$$

Auch hier hat nur eine Gleichung für zwei Unbekannte, wodurch das Aperturproblem auftritt. Pro Farbe hat man somit nur einen Kanal.

#### 4.4.6 Determinante der Hessematrix

Das selbe Prinzip wie bei der Spur wird auch bei der Determinante verwendet. Im Großen und Ganzen werden nur die Eigenwerte anders verrechnet. Die Determinante ist als Produkt der Eigenwerte definiert.

$$\det(H(f))_{xu} + \det(H(f))_{yv} + \det(H(f))_t = 0$$

Die Eigenschaften sind ähnlich wie bei der Spur.

#### 4.4.7 Log-Gradient

Helligkeitsänderungen sind generell ein Problem für alle bisherigen Kanäle. Höchstens globale additive Änderungen können verarbeitet werden, jedoch sinkt die Qualität der Ergebnisse schon, wenn sich die additiven Änderungen lokal unterscheiden. Um dies zu beheben wird bei diesem Kanal vor der Berechnung der Ableitungen der Logarithmus der Ursprungsdaten gezogen. Dabei ist wichtig, dass in diesem Fall die Werte der RGB-Kanäle um eins erhöht wurde. Dies liegt daran, dass der Definitionsbereich dieses Farbraums die Null einschließt und die diskrete Berechnung den Wert  $-\infty$  nicht verträgt.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} \Rightarrow \begin{pmatrix} (\ln R)_x \\ (\ln R)_y \\ (\ln G)_x \\ (\ln G)_y \\ (\ln B)_x \\ (\ln B)_y \end{pmatrix}$$

Diese sechs Kanäle sind nun invariant unter globalen multiplikativen Helligkeitsänderung und sogar unter Schattenwurf, falls die Änderungen nicht zu abrupt sind. In diesem Fall entstehen je zwei Gleichungen für R, G und B von der Form

$$(\ln f)_{xx}u + (\ln f)_{xy}v + (\ln f)_{xt} = 0$$

$$(\ln f)_{yx}u + (\ln f)_{yy}v + (\ln f)_{yt} = 0$$

wodurch das Aperturproblem abgeschwächt wird.

#### 4.4.8 HSV-Farbraum

Eine weitere Variante um zusätzliche Invarianzen zu erhalten, ist die Überführung des Originalbildes in den HSV-Farbraum. Das H steht für Hue und gibt die voll gesättigte

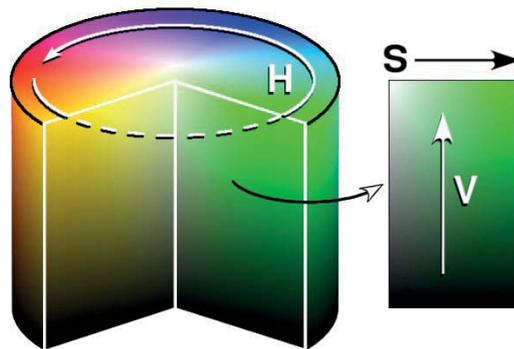


Abbildung 4.1: Eine Visualisierung des HSV-Farbraumes inklusive der Bedeutung von H, V und S. **Quelle:** [de.wikipedia.org/wiki/HSV-Farbraum](http://de.wikipedia.org/wiki/HSV-Farbraum)

Farbe an. Diese Information wird mit einem Wert zwischen  $0^\circ$  und  $360^\circ$ , der die Position auf einem Farbrad angibt, gegeben. Um diese Information in unserem Algorithmus sinnvoll verarbeiten zu können, wird diese auf zwei Kanäle aufgeteilt. Der eine enthält  $\sin H$  und der andere  $\cos H$ . Diese beiden Kanäle sollten zusammen gruppiert werden. Hinter dem S verbirgt sich die Sättigung der Farbe. Dieser Wert, der zwischen 0 und 1 liegt, gibt an wie weit die tatsächliche Farbe zwischen einem Graustufenwert und der voll gesättigten Farbe liegt. Das V steht für Value und gibt die Helligkeit des Bildpunktes an.

$$(\sin H)_x u + (\sin H)_y v + (\sin H)_t = 0$$

$$(\cos H)_x u + (\cos H)_y v + (\cos H)_t = 0$$

$$S_x u + S_y v + S_t = 0$$

$$V_x u + V_y v + V_t = 0$$

Von diesen vier Kanälen, sind die beiden Farbkanäle unter sämtlichen Helligkeitsänderungen invariant, haben dafür jedoch einen sehr geringen Informationsgehalt. Der Sättigungskanal ist immer noch unter globalen multiplikativen Helligkeitsänderungen und Schattenwurf invariant, es treten jedoch Probleme mit starken Lichtreflexionen auf, wie durch spekulare Reflexionen. Da der Valuekanal ein Graustufenbild ist, liegt es nahe, dass er keinerlei Invarianzen im Bereich Helligkeitsänderungen aufweist.

## 5. EVALUATION

---

Zur Evaluation des fertigen Algorithmus, wurde auf zwei unterschiedliche Benchmarks zurückgegriffen. Benchmarks sind vorgegebene Problemstellungen, zu denen die Lösung bekannt ist. Auf diese Art und Weise kann man sehen, wie gut die errechneten Ergebnisse sind. In diesem Fall bestanden diese zwei Benchmarks aus jeweils 4 Bildpaaren und den dazugehörigen Flussfeldern. Um nun Werte für die Evaluation zu erhalten, wurde zwischen den selbst errechneten und den gegebenen Flussfeldern der Unterschied berechnet.

### 5.1 BENCHMARKS

Ein Benchmark ist eine Testumgebung. Für den Optischen Fluss bestehen Benchmarks aus mehreren Bildsequenzen, die jeweils aus zwei aufeinanderfolgenden Bildern und dem dazugehörigen Flussfeld aufgebaut sind. Für die Tests werden die eigenen Flussfelder zwischen den Bildern berechnet und mit den vorgegebenen Verglichen. Daraus lässt sich ein Fehlerwert berechnen, der für eine Evaluation des eigenen Algorithmus verwendet werden kann.

In dieser Arbeit wurden zwei unterschiedliche Benchmarks verwendet. Der Middlebury Benchmark besteht aus Szenen, die computergeneriert und farbig sind und in denen hauptsächlich seitliche Bewegungen, sowie leichtes Kippen der Kamera für die Bildunterschiede verantwortlich sind. Zusätzlich gibt es keine Beleuchtungsänderungen. Die unterschiedlichen Bewegungen im Flussfeld resultieren aus der verschiedenen Tiefe, die die Objekte im Bildraum haben. Dadurch treten viele scharfe Kanten im Flussfeld auf. Obwohl der Middlebury Benchmark für die Evaluation verwendet wurde, tritt er in diesem Kapitel kaum in Erscheinung. Dieser Umstand ist darauf zurückzuführen, dass die meisten Konstanzannahmen das Thema photometrische Invarianz betreffen, welche mit diesem Benchmark kaum getestet werden können. Hierdurch treten auch die anderen Effekte, die getestet wurden, in den Ergebnissen des KITTI-Benchmark deutlich stärker auf.

Beim KITTI Benchmark besteht aus Szenen aus schwarzweiß Aufnahmen von Kameras, die den Blick durch die Windschutzscheibe aufgezeichnet haben. Deshalb gibt es an den gegenüberliegenden Bildrändern starke Bewegungen in entgegengesetzte Richtungen und eher weniger scharfe Kanten im Flussfeld. Zusätzlich treten in diesem Fall starke Helligkeitsänderungen auf.

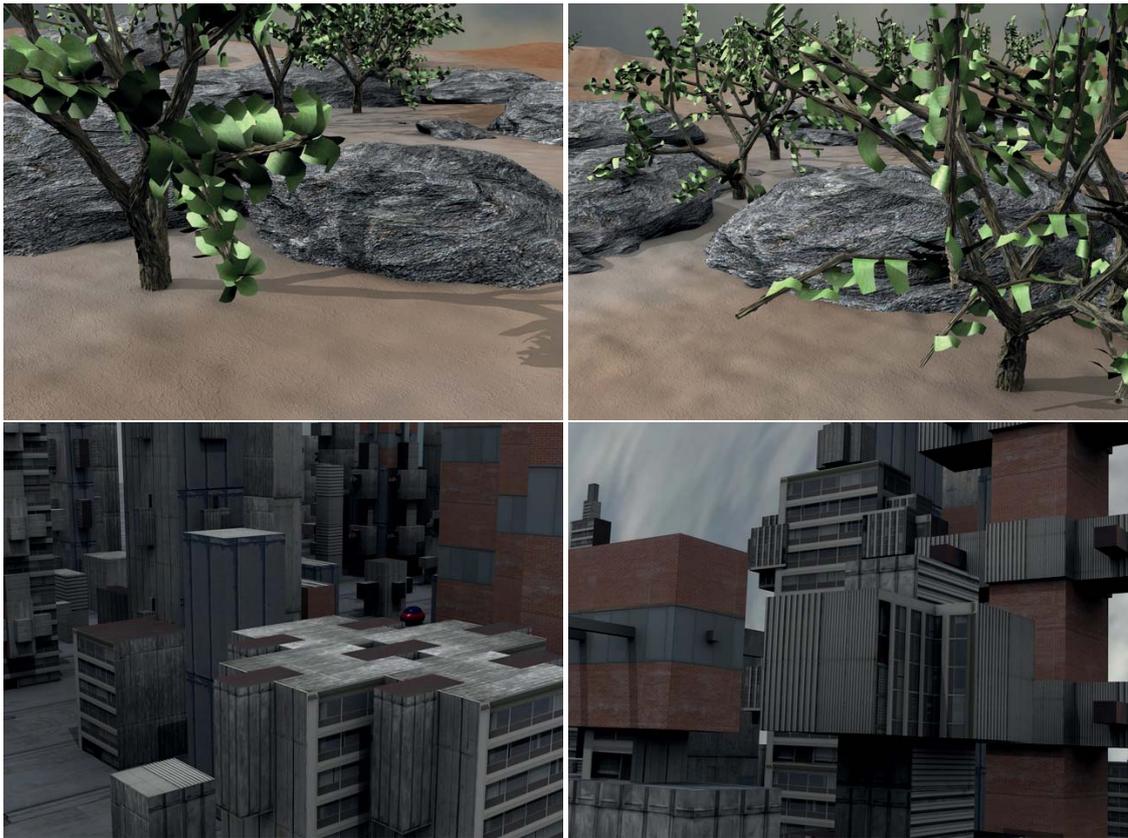


Abbildung 5.1: Die Originalbilder der Middlebury-Sequenzen.

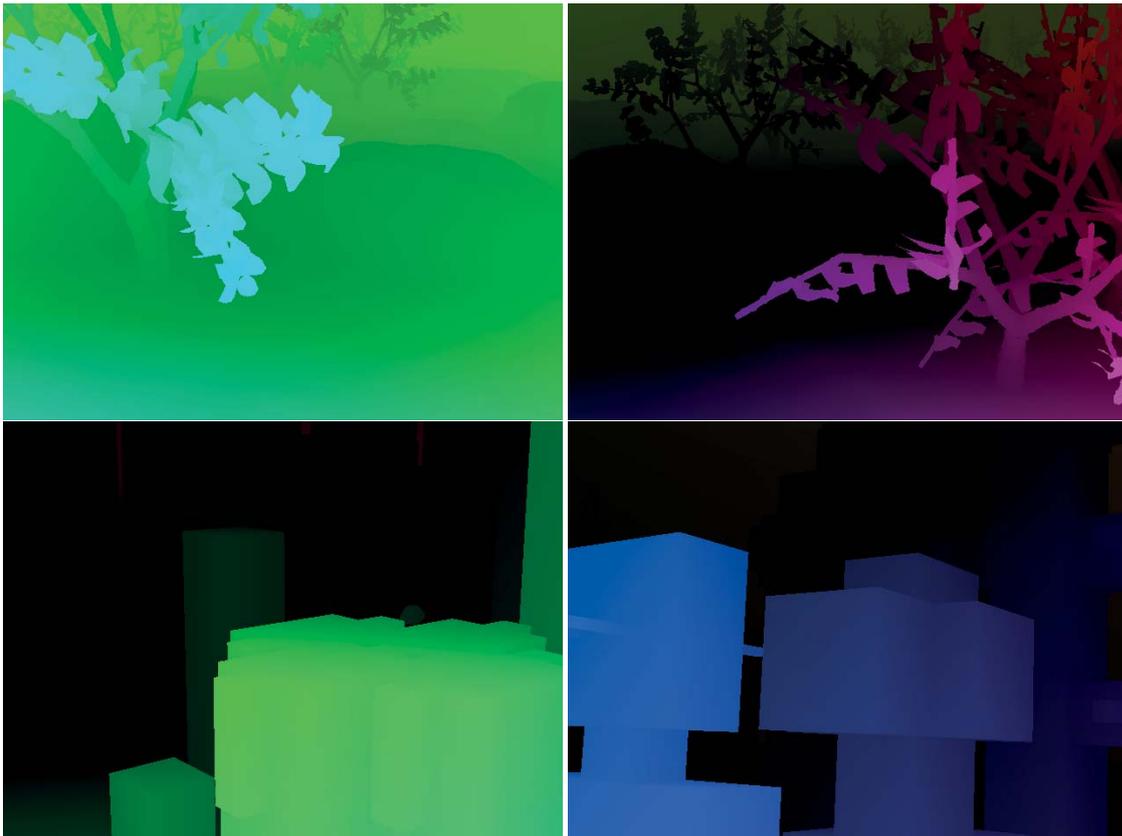


Abbildung 5.2: Die gegebenen Flussfelder der Middlebury-Sequenzen.



Abbildung 5.3: Die Originalbilder der KITTI-Sequenzen. Jeweils das erste Bild einer Sequenz.



Abbildung 5.4: Die gegebenen Flussfelder der KITTI-Sequenzen. Die schwarzen Bereiche sind nicht definiert.

## 5.2 FEHLERWERTE

Es wurde in dieser Arbeit auf den so genannten Angular Endpoint Error (AEE) zurück gegriffen.

$$AEE = \frac{\sum_{i=0}^N \sum_{j=0}^M \sqrt{\text{dif}U(i,j)^2 + \text{dif}V(i,j)^2}}{N * M}$$

Hierbei geben N und M die Bildbreite und -höhe an. Zusätzlich sind  $\text{dif}U(i,j)$  und  $\text{dif}V(i,j)$  die Differenzen zwischen den berechneten und den tatsächlichen Werten für u und v am Pixel (i,j). Im Endeffekt ist der AAE ein Wert für die durchschnittliche Differenz pro Pixel von dem berechneten Vektor (v,u)<sup>T</sup> und den tatsächlichen Vektor.

## 5.3 NORMALISIERUNG

In nahezu jedem Testfall, hat die Normalisierung über die Standardabweichung bessere Ergebnisse geliefert. Bei den Ausnahmen handelt es sich ausschließlich um die Farbkanäle des RGB-Raumes. Dies lässt sich darauf zurückführen, dass die Normalisierung wie die Standardabweichung  $\sigma_f$  deutlich weniger von Ausreißern beeinflusst werden. Im Falle der Farbkanäle hat dies kaum Einfluss, da sie von vornherein einen fixen Definitionsbereich haben. Zusätzlich wird das Verhalten des Kanals normalisiert welches in die Berechnung eingeht und es werden nicht die absoluten Grauwerte verändert, die durch die Verwendung der Ableitung nicht direkt in die Gleichung eingehen. Die Tabelle 5.1 führt für einige Beispielkanäle die Unterschiede zwischen den

Verfahren		Ergebnisse: AEE				
Konst	Norm	1	2	3	4	Avg
R	$\sigma_f$	10,594	13,575	10,430	31,580	16,545
R	Aff (-100, 100)	8,746	14,176	9,420	30,793	15,784
$\nabla R$	$\sigma_f$	7,756	7,183	4,669	21,406	10,254
$\nabla R$	Aff (-100, 100)	10,145	7,981	6,293	24,388	12,202
$\nabla(\log R)$	$\sigma_f$	6,251	8,161	4,389	20,427	9,807
$\nabla(\log R)$	Aff (-100, 100)	8,231	8,619	6,390	19,927	10,792

Tabelle 5.1: Vergleich von den verwendeten Normalisierungen bei unterschiedlichen Kanälen.

Normalisierungen auf. Für die affine Normalisierung wurde jedes Mal der Mappingbereich (-100, 100) gewählt, da er in separaten Tests, als der Beste hervor ging. Neben der bisherigen Aussage, dass nur der Farbbereich bessere Ergebnisse mit der affinen Normalisierung erzielt, kann man auch erkennen, dass sich die einzelnen Testsequenzen der Benchmark unterschiedlich verhalten. Wie man in Tabelle 5.2 erkennen kann,

1	2	3	4	Avg
1,727	0,494	1,067	0,329	0,904

Tabelle 5.2: Für alle Konstanzannahmen gemittelte Differenzen zwischen den zwei Normalisierungen.

sind die Differenzen zwischen den beiden Normalisierungen in den verschiedenen Testfällen durchaus unterschiedlich, jedoch hat die Normalisierung via Standardabweichung in jedem der vier Fälle die besseren Ergebnisse geliefert. Deshalb wurde für alles Weitere diese Form der Normalisierung gewählt.

#### 5.4 KONSTANZANNAHMEN FÜR KITTI

Im Gegensatz zu Middlebury, weist KITTI deutlich mehr Helligkeitsänderungen auf. Gerade deshalb war es sehr interessant zu testen, welche Konstanzannahmen die besten Ergebnisse für KITTI liefern. Da die KITTI-Benchmark nur Graustufenbilder beinhaltet, wurde für die Berechnung nur der Rotkanal verwendet, da die drei Farbkanäle äquivalent sind. Wie man in Tabelle 5.3 gut erkennen kann, resultiert aus der Kon-

Verfahren	Ergebnisse: AEE				
	1	2	3	4	Avg
Konst					
R	10,594	13,575	10,430	31,580	16,545
$\nabla R$	7,756	7,183	4,669	21,406	10,254
$ \nabla R $	11,493	12,874	9,043	25,179	14,647
H(R)	20,474	9,007	11,220	24,888	16,397
trH(R)	22,998	9,173	12,067	24,767	17,251
detH(R)	28,315	28,198	22,137	25,853	25,376
$\nabla(\log R)$	6,251	8,161	4,389	20,427	9,807

Tabelle 5.3: Optimierte Ergebnisse für die KITTI-Sequenzen unter Verwendung der verschiedenen Konstanzannahmen.

stanzannahme, die durch den Gradient der logarithmisierten Bildfunktion dargestellt wird,  $\nabla(\log R)$  das beste Ergebnis. Dies liegt daran, dass diese Kanäle trotz verschiedener photometrischer Invarianzen noch einen recht hohen Informationsgehalt haben. Die nächst beste Konstanzannahme ist der Gradient der Bildfunktion  $\nabla R$  selber. Um ähnlich gute Werte liefern zu können, fehlen ihr die nötigen photometrischen Invarianzen, da die KITTI-Sequenzen über einige Helligkeitsänderungen verfügen. Sämtliche anderen Versuche, photometrische Invarianzen zu erhalten, reduzieren den In-

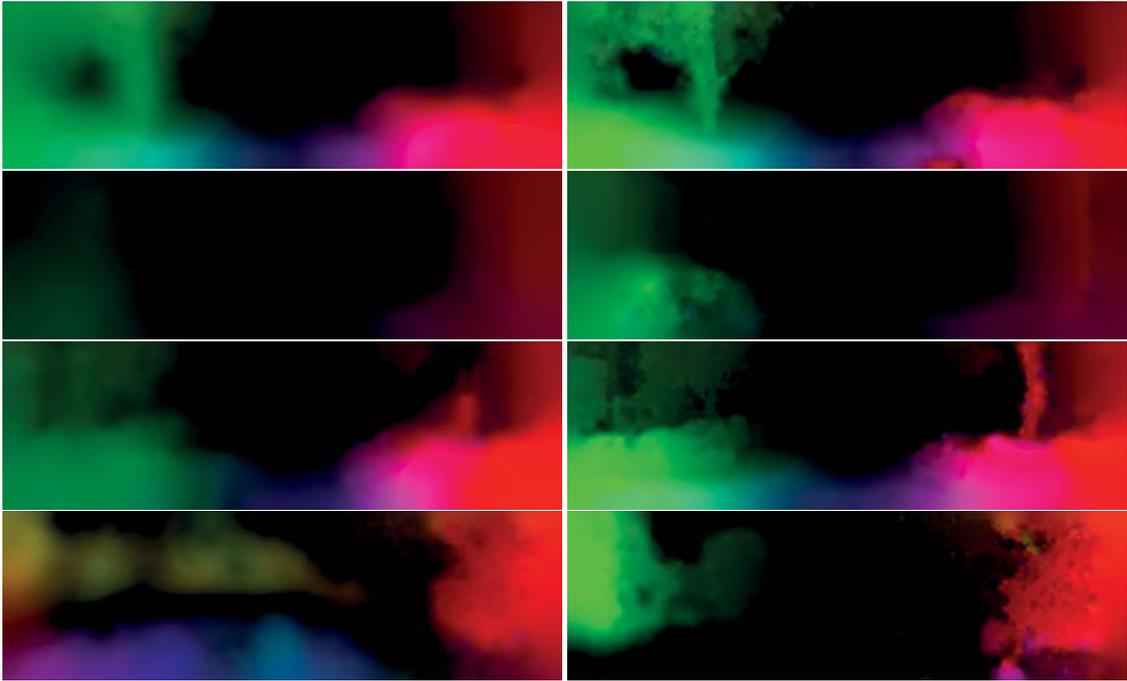


Abbildung 5.5: Die Berechneten Flussfelder für die KITT-Sequenzen und zwei unterschiedliche Konstanzannahmen. **Links:**  $R$ , **Rechts:**  $\nabla(\log R)$

formationsgehalt zu sehr um im Vergleich bestehen zu können. Eine Sonderfunktion nimmt noch der Gradientenbetrag  $|\nabla R|$  ein, da dieser wie schon erwähnt Invarianzen unter Rotation ermöglichen soll. Da weder in den KITT-Sequenzen noch in den Middlebury-Sequenzen Rotationen auftreten, kann der Erfolg dieses Versuches hier nicht vollständig evaluiert werden.

## 5.5 KOMBINATION VON KONSTANZANNAHMEN

Trotz Normalisierung der Kanäle ergeben sich für die Berechnung noch durchaus unterschiedliche Parameterwerte. Besonders auffällig ist dies beim Parameter  $\alpha$ , welcher die Gewichtung zwischen dem Daten- und dem Glattheitsterm regelt. Bei einer separaten Optimierung der Konstanzannahmen ergeben sich dafür Werte zwischen 0,05 und 15. Dies ist darauf zurück zu führen, dass die Normalisierung die absoluten Werte der Kanäle betrifft und nicht die abgeleiteten Werte, die in die Berechnung eingehen. Wenn man nun die zu kombinierenden Konstanzannahmen separat optimiert hat, ist es möglich aus den Werten für den  $\alpha$  Parameter die Gewichte der einzelnen Kanäle

Verfahren			Ergebnisse: AEE				
Konst	$\alpha$	$\gamma$	1	2	3	4	Avg
$\nabla R$	0,9	1	7,756	7,183	4,669	21,406	10,254
$\nabla(\log R)$	4	1	6,251	8,161	4,389	20,427	9,807
$(\nabla R), (\nabla(\log R))$	1,2	0,8163	6,641	7,838	4,080	20,043	9,651
		0,1837					

Tabelle 5.4: Die optimalen Ergebnisse für die Konstanzannahmen  $\nabla R$ ,  $\nabla(\log R)$  und ihre Kombination auf den KITTI-Sequenzen.

zu berechnen. Im Fall von zwei Konstanzannahmen, die aus einer Gruppe bestehen, ist dies durch

$$\gamma_1 = \frac{\alpha_2}{\alpha_1 + \alpha_2}$$

$$\gamma_2 = \frac{\alpha_1}{\alpha_1 + \alpha_2}$$

darstellbar. Die Abhängigkeit aus dem entgegengesetzten  $\alpha$  entsteht aus der Tatsache, dass  $\alpha$  ein inverser Parameter für die Größe der Daten des Kanals ist.

Im Falle von den Konstanzannahmen  $\nabla R$  und  $\nabla(\log R)$  konnte, wie aus Tabelle 5.4 hervorgeht, das Ergebnis durch die Kombination noch verbessert werden. Hier ist es sogar absolut gesehen das beste Ergebnis, das auf dem KITTI-Benchmark erzielt wurde. Man kann an den ersten beiden Sequenzen gut erkennen, dass die Konstanzannahmen unterschiedliche Stärken haben. Und gerade deshalb, hat die Kombination das Potential noch besser zu sein als die Ursprünge es einzeln sind.

Aus diesen beiden Sequenzen geht jedoch auch hervor, dass die Kombination nicht das beste Ergebnis liefert. Die Technik die hier verwendet wird um die Konstanzannahmen zu kombinieren, ist eine einfache, gleich gewichtete, Mittelung der beiden Aussagen. Um die verschiedenen Stärken der Konstanzannahmen optimal nutzen zu können, würde ein adaptives Verfahren gebraucht, bei dem die Konstanzannahmen je nach Situation unterschiedlich gewichtet werden können.

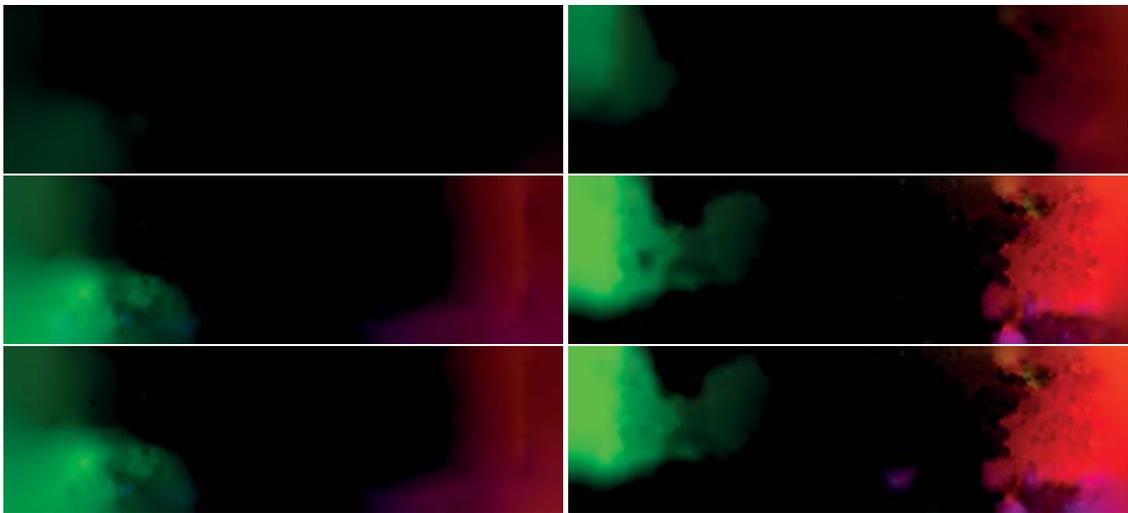


Abbildung 5.6: Gegenüberstellung von  $\nabla R$ ,  $\nabla(\log R)$  und der Kombination in zwei unterschiedlichen Sequenzen. In der Reihenfolge **V.o.n.U.**

## 6. ZUSAMMENFASSUNG UND AUSBLICK

---

### 6.1 ZUSAMMENFASSUNG

Ziel der vorliegenden Arbeit war es, einen bestehenden Algorithmus zur Berechnung des Optischen Flusses um die Möglichkeit zu erweitern, verschiedene Konstanzannahmen zur Berechnung zu verwenden und diese auch kombinieren zu können. Zusätzlich sollte der entstandene Funktionsumfang evaluiert werden. Dazu wurden, neben der Unterstützung von Mehrkanalbildern, zwei Normalisierungsmethoden und die Umwandlung von Konstanzannahmen in Bildkanäle implementiert. Diese neuen Möglichkeiten wurden nun mit Hilfe zweier Benchmarks getestet um drei Dinge herauszufinden. Es wurde geprüft welche Normalisierung die besten Ergebnisse liefert, welche Konstanzannahmen trotz der Helligkeitsänderungen des KITTI-Benchmarks bester Ergebnisse liefert und inwieweit hin die Kombination mehrerer Konstanzannahmen Verbesserungen ermöglicht.

Es wurde dabei entdeckt, dass die Normalisierung über die Standardabweichung der affinen Normalisierung überlegen ist, da sie das Verhalten des Kanals beeinflusst und weniger anfällig gegenüber Ausreißern ist.

Bei den Tests auf dem KITTI-Benchmark ist aufgefallen, dass die Konstanzannahme die den Gradient des Logarithmus des Bildes implementiert auf Grund ihrer Invarianzen die besten Ergebnisse erzeugt und dass andere ähnlich invariante Konstanzannahmen zu viele Informationen vernichten um mithalten zu können. Schlussendlich hat jedoch die Kombination des Gradienten von R und des Gradienten vom Logarithmus von R die besten Resultate hervorgebracht.

Somit wurde auch die Frage beantwortet, ob die Kombination von mehreren Konstanzannahmen erfolgversprechend ist. Obwohl das in diesem Fall zutraf, wurde auch gezeigt, dass die Informationen der beiden Konstanzannahmen - in diesem global gleichgewichteten Fall - nicht optimal verwendet wurden und dementsprechend noch Verbesserungspotential besteht.

### 6.2 AUSBLICK

Wie schon zuvor erwähnt, muss die optimale Kombination mehrerer Konstanzannahmen noch weiterentwickelt werden. Eine variable Methode wäre möglich, die sich je nach Sequenz eine Konstanzannahme heraussucht, oder es wäre vorstellbar diese Entscheidung innerhalb der Sequenz lokal zu treffen.

Sobald diese Eigenschaft verbessert ist, wäre es interessant gezielt nach Konstanzannahmen zu suchen, die einen Spezialfall besonders gut lösen können. Diese könnten

mit anderen ähnlich gestalteten Konstanzannahmen kombiniert werden um in ihrer Gesamtheit jeden Spezialfall optimal abdecken zu können.

## LITERATUR

---

- [1] T. Brox, A. Bruhn, N. Papenberg und J. Weickert, "High accuracy optical flow estimation based on a theory for warping", in *LNCS*, Bd. 3024, Springer Berlin Heidelberg, 2004, S. 25–36.
- [2] B. K. Horn und B. G. Schunck, "Determining optical flow", in *Artificial Intelligence*, Bd. 17, Elsevier B.V., 1981, S. 185–203.
- [3] Y. Mileva, A. Bruhn und J. Weickert, "Illumination-robust variational optical flow with photometric invariants", in *LNCS*, Bd. 4713, Springer Berlin Heidelberg, 2007, S. 152–162.
- [4] C. Vogel, S. Roth und K. Schindler, "An evaluation of data costs for optical flow", in *LNCS*, Bd. 8142, Springer Berlin Heidelberg, 2013, S. 343–353.
- [5] T. H. Kim, H. S. Lee und K. M. Lee, "Optical flow via locally adaptive fusion of complementary data costs", in *2013 Proc. IEEE International Conference on Computer Vision (ICCV)*, IEEE Computer Society Press, 2013, S. 3344–3351.
- [6] L. Sevilla-Lara, D. Sun, E. Learned-Miller und M. Black, "Optical flow estimation with channel constancy", in *LNCS*, Bd. 8689, Springer International Publishing, 2014, S. 423–438.
- [7] D. Young, "Iterative methods for solving partial difference equations of elliptic type", in *Transactions of the American Mathematical Society*, American Mathematical Society, 1954, S. 92–111.
- [8] M. J. Black und P. Anandan, "Robust dynamic motion estimation over time", in *1991 Proc. IEEE International Conference on Computer Vision (ICCV)*, IEEE Computer Society Press, 1991, S. 292–302.
- [9] A. Bruhn, *Correspondence problems in computer vision*, 2014.
- [10] P. Münch, *Optischer Fluss für Videosequenzen*, 2005.
- [11] I. Gelfand und S. Fomin, *Calculus of variations. Revised English edition translated and edited by Richard A. Silverman*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1963.



## 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