

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3696

Bestimmung der Fundamentalmatrix aus dichten Korrespondenzfeldern

Simon Geßner

Studiengang:	Informatik
Prüfer/in:	Prof. Dr.-Ing. Andrés Bruhn
Betreuer/in:	Prof. Dr.-Ing. Andrés Bruhn
Beginn am:	19. November 2014
Beendet am:	21. Mai 2015
CR-Nummer:	I.5.4, I.4.8, I.4.6, G.1.3, G.1.6

Kurzfassung

Im Maschinensehen stellt die Extraktion von Bewegungsinformation aus Bildsequenzen ein zentrales Problem dar. Typischerweise möchte man dabei das Verschiebungsvektorfeld (Korrespondenzfeld) zwischen zwei aufeinanderfolgenden Bildern einer Bildfolge bestimmen, welches in der Literatur auch als *optischer Fluss* bezeichnet wird. Dieser kann heute bereits sehr genau mithilfe sogenannter Variationsansätze berechnet werden, welche trotzdem noch Ungenauigkeiten in informationsarmen, homogenen Bildbereichen aufweisen. Entsteht der optische Fluss ausschließlich aufgrund der Eigenbewegung der Kamera (Stereofall), so gibt es eine weitere geometrische Bedingung, die bei dessen Berechnung verwendet werden sollte: die Epipolarbedingung. Diese erleichtert die Korrespondenzfindung durch Beschränkung des Suchraums und wird mathematisch durch die sogenannte Fundamentalmatrix beschrieben. Ziel der Arbeit ist die Untersuchung verschiedener Ansätze zur Schätzung dieser Fundamentalmatrix aus initialen Punktkorrespondenzen. Zum einen, indem viele Punktkorrespondenzen aus einem zuvor berechneten optischen Flussfeld verwendet werden und zum anderen durch das Heranziehen weniger, aber dafür sehr genauer, Punktpeare. Schließlich wird die Qualität der geschätzten Fundamentalmatrix mit einem geeigneten Fehlermaß evaluiert.

Abstract

The extraction of motion information from image sequences is one of the central problems in computer vision. In this context one is typically concerned about the computation of the displacement vector field (correspondence field) between two consecutive frames of an image sequence, which is also referred to in literature as *Optical Flow* (or *Optic Flow*). Today the Optical Flow can already be computed quite accurately with the help of so-called variational methods, which are nevertheless showing inexactness in homogeneous image areas with low information content. If movement of the camera is the sole cause for the Optical Flow (stereo case), another geometric constraint should be used for its computation: the epipolar constraint. It simplifies finding the correspondences by reducing the search area and is mathematically described by the so-called fundamental matrix. The goal of this thesis is the investigation of different approaches for estimating this fundamental matrix with the help of initial point correspondences. On the one hand by using many point correspondences from a previously estimated optical flow field and on the other hand by only taking few, but precise ones. Finally the quality of the estimated fundamental matrix will be evaluated by an appropriate error measurement.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Ziel	8
1.2	Gliederung der Arbeit	8
2	Grundlagen	9
2.1	Optischer Fluss	9
2.1.1	Grundlegendes	10
2.1.2	Verfahren zur Berechnung des optischen Flusses	12
2.1.3	Verfahren von Demetz <i>et al.</i>	17
2.2	Feature-Matches	22
2.2.1	Detektion der Skalenraum-Extrema	23
2.2.2	Akkurate Key-Point-Lokalisierung	24
2.2.3	Richtungszuweisung	28
2.2.4	SIFT-Key-Point-Deskriptor	29
2.2.5	SIFT-Matching	29
2.2.6	Andere Feature-Deskriptoren	30
2.3	Stereogeometrie	31
2.3.1	Lochkameramodell	31
2.3.2	Epipolargeometrie	33
3	Verwandte Arbeiten	37
4	Bestimmung der Fundamentalmatrix	39
4.1	7-Punkt-Algorithmus	39
4.2	8-Punkt-Algorithmus	40
4.3	Dichte Korrespondenzen	41
4.3.1	M-Schätzer	41
4.3.2	Geometrische Ansätze	42
4.3.3	Normalisierung und Rang-2-Bedingung	44
4.4	RANSAC	46
4.4.1	Nachteil traditioneller Verfahren	46
4.4.2	Algorithmus	47
4.4.3	Erfolgskriterien	48
4.4.4	Anwendung auf Fundamentalmatrix	49
4.5	Variationen mit RANSAC	50
4.5.1	Least Median of Squares	50
4.5.2	Ausdünnung der Korrespondenzen	51

4.5.3	SIFT und RANSAC	52
5	Evaluation	55
5.1	Distanz zweier Fundamentalmatrizen	55
5.2	Berechnung der Fundamentalmatrix aus Projektionsmatrizen	58
5.3	Dichte Verfahren	59
5.3.1	8-Punkt-Algorithmus	60
5.3.2	Normalisierung	60
5.3.3	M-Schätzer	61
5.4	RANSAC-basierte Verfahren	64
5.4.1	RANSAC	64
5.4.2	RANSAC ausgedünnt	71
5.4.3	SIFT und RANSAC	78
5.5	Dichter Ansatz vs. RANSAC	81
5.6	Qualität des Ausgangsflussfeldes	82
6	Zusammenfassung und Ausblick	85
6.1	Zusammenfassung	85
6.2	Ausblick	86
	Literaturverzeichnis	89

1 Einleitung

Das wichtigste Sinnesorgan des Menschen sind seine Augen. Man geht davon aus, dass sie für etwa 80 Prozent unserer Sinneseindrücke verantwortlich sind und dem Menschen somit am meisten Information über dessen Umwelt liefern. Nimmt man sich dies zum Vorbild, stellt für Maschinen die Analyse von Bildsensordaten ebenfalls ein enormes Potenzial für deren Interaktion mit der Umwelt dar. Der wissenschaftliche Bereich, der sich mit der Analyse von Bildern beschäftigt, ist das *Maschinensehen* (engl. *Computer Vision*). Ein großes Ziel darin ist es, aus aufeinanderfolgenden Bildern einer Bildfolge möglichst genaue Bewegungsinformation, den *Optischen Fluss*, zu extrahieren.

Diese Extraktion erfolgt bereits relativ gut mithilfe sogenannter *Variationsansätze*. Deren Genauigkeit ist jedoch nicht in allen Bereichen des Bildes gleich. Handelt es sich um einen Bildbereich mit viel Farbänderungen (Textur), dann gibt es bei der Schätzung des optischen Flusses mit diesen Ansätzen kaum Probleme. In homogenen Bildbereichen mit wenig Farbänderungen jedoch weichen deren Ergebnisse oft von der tatsächlichen Bewegung (der sogenannten *Ground Truth*) ab, da die Zuordnung nicht eindeutig ist.

Nun stellt sich die Frage: Gibt es eine Möglichkeit, die Berechnung des optischen Flusses gerade in diesen problematischen Bereichen zu verbessern?

Falls der optische Fluss ausschließlich durch die Eigenbewegung der Kamera zustande kommt, gibt es eine zusätzliche geometrische Bedingung, die bei der Schätzung berücksichtigt werden kann: die Epipolarbedingung. Diese schränkt den Suchraum bei der Korrespondenzfindung auf Linien ein und ist mathematisch in Form der sogenannten Fundamentalmatrix gegeben. Da die Fundamentalmatrix von der Kamerabewegung abhängig und somit im Allgemeinen nicht bekannt ist, muss sie in einem ersten Schritt zunächst aus initialen Korrespondenzen geschätzt werden. Hierzu können erneut Verfahren zur Bewegungsberechnung verwendet werden, diesmal jedoch ohne Vorkenntnis der Epipolarbedingung. Da die Fundamentalmatrix nur wenige Freiheitsgrade besitzt, bieten sich hierfür ebenfalls Variationsansätze an, die tausende Korrespondenzen (dichte Verschiebungsvektorfelder) liefern und somit eine robuste Schätzung der Fundamentalmatrix ermöglichen sollten.

Zur Fundamentalmatrix haben bereits Valgaerts *et al.* [VBMW12] gezeigt, dass diese mithilfe dichter Korrespondenzfelder recht gut bestimmt werden kann. Allerdings können die damals verwendeten Modelle zur Bewegungsberechnung aus heutiger Sicht nicht mehr als Stand der Kunst bezeichnet werden. Zudem gab es zu dieser Zeit keine geeigneten Benchmarks im Bereich der Bewegungsschätzung, die eine Verwendung der Epipolarbedingung gerechtfertigt hätten.

1.1 Ziel

Ziel dieser Arbeit ist es deshalb, die Qualität der Bestimmung der Fundamentalmatrix basierend auf einem aktuellen Ansatz zur Bewegungsschätzung neu zu untersuchen. Als Basisverfahren soll hierbei der Ansatz von Demetz *et al.* [DSV⁺14] dienen. Dazu sollen verschiedene Verfahren zur Schätzung der Fundamentalmatrix implementiert und untersucht werden, die es mit geeigneten Testbildpaaren und Fehlermaßen zu evaluieren gilt. Dabei soll der Fokus vor allem auf Ansätzen liegen, die eine robuste Bestimmung der Fundamentalmatrix erlauben (z.B. M-Glätter, RANSAC). Auch ein Verfahren, das auf einem merkmalsbasierten Ansatz beruht (z.B. SIFT), soll als Vergleichsmethode implementiert und evaluiert werden.

1.2 Gliederung der Arbeit

Im Folgenden soll nun ein kurzer Überblick über die Kapitel dieser Arbeit gegeben werden, um deren inhaltliche Schwerpunkte zu erläutern:

Im zweiten Kapitel werden zunächst Grundlagen dieser Arbeit beschrieben. Der Begriff des optischen Flusses wird eingeführt sowie verschiedene Verfahren zu dessen Berechnung erläutert. In diesem Zusammenhang wird auch das Verfahren von Demetz *et al.* erläutert, das später zur Bestimmung von dichten Punktkorrespondenzfeldern herangezogen wird. Dem folgt die Beschreibung der Scale Invariant Feature Transform (SIFT), welche die Detektion von besonders charakteristischen Merkmalen ermöglicht und somit die Bestimmung von einzelnen, sehr genauen Punktkorrespondenzen erlaubt. Im Anschluss werden Lochkamera- und Stereogeometriemodelle erklärt, um die Bedeutung der Fundamentalmatrix in der Epipolargeometrie herauszustellen.

In Kapitel 3 werden verwandte Arbeiten vorgestellt, die sich ebenfalls mit der Bestimmung der Fundamentalmatrix befassen.

Kapitel 4 widmet sich dann verschiedenen Ansätzen zur Bestimmung der Fundamentalmatrix. Hierbei werden zunächst der 7-Punkt-Algorithmus sowie der grundlegende 8-Punkt-Algorithmus erklärt. Für letzteren werden verbesserte Verfahren zur Bestimmung der Fundamentalmatrix unter der Verwendung dichter sowie spärlicher Korrespondenzen erklärt. Zudem wird mit dem RANSAC-Algorithmus ein robustes Verfahren zur Modellbestimmung basierend auf einer großen Menge fehlerbehafteter Datenpunkte vorgestellt. Der wesentliche Unterschied dichter und spärlicher Verfahren besteht dabei in der Anzahl und Auswahl der verwendeten Punktkorrespondenzen.

Nachdem die verschiedenen Verfahren vorgestellt wurden, werden deren Ergebnisse in Kapitel 5 ausgewertet und verglichen. Hierfür wird die Berechnung einer geometrisch repräsentativen Distanz zwischen zwei Fundamentalmatrizen von Faugeras [FLP01] als zu verwendendes Fehlermaß behandelt und gezeigt, wie man die Fundamentalmatrix aus zwei Kamera-Projektionsmatrizen berechnen kann. Die eigentliche Evaluation findet anhand von vier repräsentativen Stereobildpaaren statt.

Zum Abschluss werden die wichtigsten Ergebnisse der Arbeit in Kapitel 6 zusammengefasst und es wird ein Ausblick auf weitere Möglichkeiten in diesem Gebiet gegeben.

2 Grundlagen

Nachdem zuvor die Motivation und Zielstellung für diese Arbeit herausgearbeitet wurde, sollen im folgenden Kapitel grundlegende Begriffe und Verfahren erklärt werden, die für das Verständnis dieser Diplomarbeit notwendig sind. Beginnend mit dem optischen Fluss soll dieser in Abschnitt 2.1 vorgestellt und verschiedene Ansätze zu dessen Berechnung erklärt werden, da diese Korrespondenzfelder bei dichten Verfahren zur Schätzung der Fundamentalmatrix verwendet werden sollen. Im Abschnitt 2.2 wird die Scale Invariant Feature Transform als Alternative zur Bestimmung einzelner Punktkorrespondenzen erklärt. Schließlich werden im Abschnitt 2.3 ausgehend vom Lochkamera-modell die Geometrie bei einer und zwei Kameras erläutert. Dies führt dann zur Definition der Fundamentalmatrix, deren Schätzung in Kapitel 4 diskutiert wird.

2.1 Optischer Fluss

Im *Maschinensehen* (engl. *Computer Vision*) stellt sich neben der zentralen Frage, was auf einem Bild zu sehen ist, auch die weiterführende Frage: Wohin bewegt es sich? Geht man von einem einzelnen Bild aus, so lässt sich ohne Zusatzinformation kaum eine Aussage treffen, welche Bewegungen zum Zeitpunkt des Bildes gegeben waren. Möchte man dies herausfinden, muss man also mehrere Bilder zurate ziehen. Wählt man zwei Bilder aus, die die gleiche Szene beschreiben, aber zwischen denen sich etwas bewegt hat, kann man nun den *optischen Fluss* (engl. *optic flow / optical flow*) berechnen.

Der optische Fluss beschreibt eben diese Bewegung zwischen zwei Bildern, wobei er für jedes einzelne Pixel im ersten Bild eine Bewegungsinformation beinhaltet, die für dieses Pixel angibt, wohin es sich bewegt, wenn man ausgehend von der gleichen Position im zweiten Bild sucht. Diese Information wird als Vektor dargestellt und im zweidimensionalen Bildraum erhält man somit eine Richtung (einen Winkel) und eine Länge des Vektors. Alternativ können die Vektoren auch durch eine x - und eine y -Komponente beschrieben werden:

$$\vec{v} = \begin{pmatrix} u \\ v \end{pmatrix}. \quad (2.1)$$

Diese Beschreibung wird auch weiterhin in dieser Arbeit verwendet. Man kann sich den optischen Fluss also als ein Vektorfeld im zweidimensionalen Raum vorstellen, welches durch die Größe der Bilder begrenzt wird, aus denen der optische Fluss berechnet wird. Daher wird der optische Fluss im Folgenden auch als *Verschiebungsvektorfeld* bezeichnet.

2.1.1 Grundlegendes

Um zunächst einige grundlegende Aspekte des optischen Flusses zu erklären, soll hier mit der visuellen Darstellung eines solchen berechneten Vektorfeldes begonnen werden. Hierfür wird in dieser Arbeit anstatt einer Vektorfeld-Visualisierung eine Darstellung mittels Farbkodierung bevorzugt (siehe Abbildung 2.1 unten). Die Vorteile liegen darin, dass jedes Pixel des Ursprungsbildes auch in der Flussvisualisierung wieder genau ein Pixel zugewiesen bekommt, wodurch das komplette Flussfeld bei gleicher Bildgröße angezeigt werden kann. Die Farbe der Pixel wird entsprechend einer Farbtabelle, welche jedem Winkel von $0 - 360^\circ$ eine bestimmte Farbe zuweist, bestimmt. Weiterhin kann die Größe des Verschiebungsvektors über die Helligkeit des entsprechenden Pixels deutlich gemacht werden. Diese Darstellungsart erlaubt gegenüber der Vektordarstellung eine dichtere Repräsentation des optischen Flusses, die die Größe der Originalbilder nicht überschreitet. Zudem können Richtungs- und Größenänderungen im Feld durch das farbsensible menschliche Auge, welches über 2 Millionen verschiedene Farben unterscheiden kann, schnell erkannt werden. Als Negativpunkt bleibt hier jedoch zu erwähnen, dass man zunächst lernen muss, welche Farben für welche Richtung stehen, aber genau hierin bietet sich auch die Möglichkeit zur flexiblen Verwendung individueller, selbst bevorzugter Farbdarstellungen des optischen Flusses.

Grauwertkonstanz

Doch wie kann man überhaupt einen speziellen Punkt in einem anderen Bild wiederfinden? Der wichtigste Ansatzpunkt hierzu ist wohl die Farbe der Pixel. Werden von einem Punkt mehrere Bilder aufgenommen, sollte er in allen Bildern die gleiche Farbe aufweisen. Natürlich können verschiedene Kameras und Beleuchtungsunterschiede dies komplizierter machen, doch im Grunde genommen ist das der zentrale Ansatzpunkt bei der Berechnung des optischen Flusses. Man spricht auch von der *Grauwertkonstanz*:

$$f(x + u, y + v, t + 1) = f(x, y, t). \quad (2.2)$$

In obiger Formel stehen die Funktionswerte f für Farbwerte (oder Grauwerte, je nachdem, ob das Bild auf mehrere Farbkanäle oder nur einen abbildet) und diese sind abhängig von der räumlichen x - und y -Komponente sowie der zeitlichen t -Komponente. Das auf der rechten Seite beschriebene Bild wurde hier zum Zeitpunkt t aufgenommen, das linke zum Zeitpunkt $t + 1$, also einen Zeitschritt später. Während dieses Zeitschrittes hat sich der Punkt mit Bildkoordinaten $(x, y)^T$ um den Verschiebungsvektor $(u, v)^T$ zum Punkt $(x + u, y + v)^T$ verschoben. Für jeden Punkt des Bildes soll also genau dieser Verschiebungsvektor berechnet werden.

An dieser Stelle sei noch erwähnt, dass man nicht immer Folgebilder einer Bildsequenz als Ausgangsdaten haben muss. Beispielsweise können hier auch Bilder, die zum gleichen Zeitpunkt, aber aus verschiedenen Positionen aufgenommen wurden, verwendet werden. In diesem Fall nehme einfach eines der Bilder den fiktiven Zeitpunkt $t + 1$ an. Ein bekanntes Beispiel dafür sind *Stereobildpaare*.

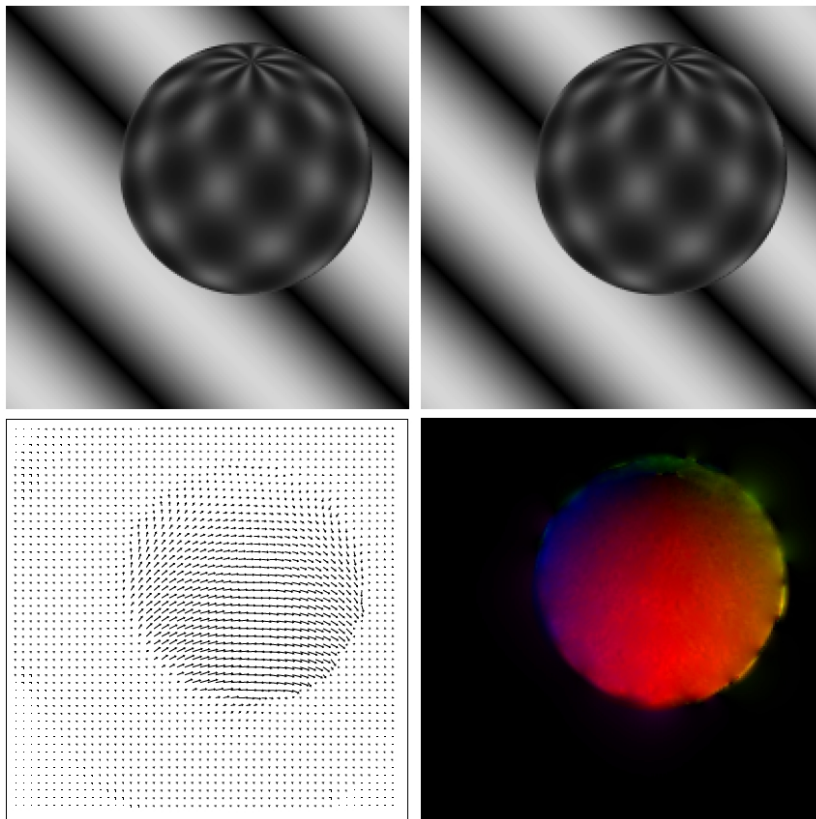


Abbildung 2.1: Visualisierung des optischen Flusses [Bru12a]. **Oben:** die beiden Ausgangsbilder. **Unten links:** Visualisierung durch Vektorfeld. **Unten rechts:** Visualisierung durch Farbtabelle.

Führt man auf der linken Seite von Gleichung (2.2) eine Taylorreihenentwicklung durch, so erhält man:

$$f(x + u, y + v, t + 1) = f(x, y, t) + f_x u + f_y v + f_t (+\text{Rest höherer Ordnung}). \quad (2.3)$$

Durch Kombination von Gleichung (2.2) mit Gleichung (2.3) ergibt sich dann der sogenannte (linearierte) *Optic Flow Constraint* (kurz: *OFC*):

$$f_x u + f_y v + f_t = 0. \quad (2.4)$$

Aperturproblem

Betrachtet man die Gleichung (2.4) des Optic Flow Constraint mit den Unbekannten u und v , so erkennt man eine Geradengleichung, was somit bedeutet, dass diese Gleichung für sich noch keine eindeutige Berechnung des optischen Flusses ermöglicht. Vielmehr gibt es eine Richtung entlang

der Gerade, auf die man einen beliebig großen Fehler zum Verschiebungsvektor hinzuaddieren kann, ohne den Optic Flow Constraint zu verletzen. Umgangssprachlich lässt sich das so erklären, dass Bewegung entlang einer Kante nicht erkannt werden kann, da entlang der Kante immer der gleiche Grauwert vorhanden ist.

Dieses Problem ist auch bekannt als das *Aperturproblem*. Es kann also nur der Flussanteil parallel zum räumlichen Gradienten des Grauwertes berechnet werden. Der Gradient stellt die Richtung dar, in die sich der Grauwert am stärksten verändert. Die zuvor genannte Gerade, die vom OFC beschrieben wird und entlang derer beliebige Flüsse unbestraft dazu addiert werden können, verläuft senkrecht zu diesem Gradienten.

Man kann den optischen Fluss also aufteilen in zwei Komponenten: die Komponente entlang des Gradienten, die man berechnen kann (den *Normalenfluss*) und die Komponente senkrecht dazu, deren Größe unbestimmt bleibt (den tangentialen Fluss).

Die Richtung des Normalenflusses ist daher die Richtung des normierten Gradienten:

$$\vec{n} = \frac{\nabla f}{|\nabla f|}. \quad (2.5)$$

Verbindet man jetzt den umgeformten OFC

$$(u, v)\nabla f = -f_t \quad (2.6)$$

mit der Definition des Normalenflusses

$$\begin{pmatrix} u_n \\ v_n \end{pmatrix} = (u, v) \frac{\nabla f}{|\nabla f|} \frac{\nabla f}{|\nabla f|}, \quad (2.7)$$

erhält man durch Einsetzen die Formel zur Berechnung des Normalenflusses:

$$\begin{pmatrix} u_n \\ v_n \end{pmatrix} = -\frac{f_t}{|\nabla f|} \cdot \frac{\nabla f}{|\nabla f|} = -\frac{1}{f_x^2 + f_y^2} \begin{pmatrix} f_x f_t \\ f_y f_t \end{pmatrix}. \quad (2.8)$$

Natürlich kann diese Berechnung des optischen Flusses aufgrund der komplett unbestimmten tangentialen Komponente in der Praxis keine guten Ergebnisse erzielen. Daher werden im folgenden Abschnitt bessere Ansätze zu dessen Bestimmung vorgestellt.

2.1.2 Verfahren zur Berechnung des optischen Flusses

Das Problem bei der Berechnung des Normalenflusses ist die fehlende Anzahl an Bedingungen pro Pixel. Es ist nur eine Gleichung (OFC) mit zwei Unbekannten vorhanden, was deren eindeutige Bestimmung unmöglich macht. Man muss also zusätzliche Annahmen treffen, um die Berechnung des optischen Flusses zu verbessern. In diesem Abschnitt sollen zwei bekannte Verfahren zur Berechnung des optischen Flusses vorgestellt werden: zum einen das lokale Verfahren von Lucas & Kanade [LK81] und zum anderen das globale Verfahren von Horn & Schunck [HS81]. Diese beiden Verfahren

unterscheiden sich vor allem in der Anzahl der zur Berechnung des optischen Flusses hinzugezogenen Pixel. Während bei lokalen Verfahren nur die Pixel in der unmittelbaren Nachbarschaft einen Einfluss auf den berechneten optischen Fluss haben, werden bei globalen Optimierungsverfahren alle Pixel verwendet.

Die Methode von Lucas und Kanade

In der Methode von Lucas & Kanade [LK81] werden demnach ausgewählte Pixel aus der Umgebung des zu berechnenden Pixels zu Hilfe gezogen, um seinen optischen Fluss zu bestimmen. Diese Nachbarschaft ist im Allgemeinen kreisförmig, mit einem festgelegten Radius ρ . Es wird versucht, für alle Pixel innerhalb der Nachbarschaft die lokale Energie $E(u, v)$, welche über die Summe der quadrierten OFC in diesem Bereich definiert ist, zu minimieren:

$$E(u, v) = \frac{1}{2} \int_{B_\rho(x_0, y_0)} (f_x u + f_y v + f_t)^2 dx dy. \quad (2.9)$$

Wie für alle Extrema einer Funktion muss für ein Minimum dieser Gleichung gelten, dass deren Ableitungen nach u und v gleich 0 sind:

$$\frac{\partial E}{\partial u} = \int_{B_\rho} f_x (f_x u + f_y v + f_t) dx dy = 0, \quad (2.10)$$

$$\frac{\partial E}{\partial v} = \int_{B_\rho} f_y (f_x u + f_y v + f_t) dx dy = 0.$$

Durch Umformung erhält man folgendes Gleichungssystem, dessen Lösung den optischen Fluss nach Lucas und Kanade liefert:

$$\begin{pmatrix} \int_{B_\rho} f_x^2 dx dy & \int_{B_\rho} f_x f_y dx dy \\ \int_{B_\rho} f_x f_y dx dy & \int_{B_\rho} f_y^2 dx dy \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} - \int_{B_\rho} f_x f_t dx dy \\ - \int_{B_\rho} f_y f_t dx dy \end{pmatrix}. \quad (2.11)$$

Auf der linken Seite von (2.11) erhält man in der resultierten Matrix den über die Region aufsummierten Strukturtenor [FG87]

$$J = \begin{pmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{pmatrix} = \begin{pmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{pmatrix}. \quad (2.12)$$

Dieser enthält Informationen über deren Grauwertverlauf. Wenn man ihn auf Eigenwerte und Eigenvektoren untersucht, kann man Informationen über die Richtungen erhalten, in denen sich der Grauwert am meisten verändert. Weiterhin gibt es nun verschiedene Möglichkeiten der Gewichtung der Umgebungspixel in diesem Verfahren. Ein Box-Filter würde alle Pixel gleich stark bewerten. Besser eignet sich hier aber ein Gauß-Filter, der die weiter vom Zentrum der Region entfernten Pixel mit einem niedrigeren Gewicht versieht. Dies macht von daher Sinn, dass man dem Pixel im Zentrum,

von dem man den optischen Fluss berechnen möchte, auch den stärksten Einfluss auf das Ergebnis zugestehen sollte.

Für die Methode von Lucas & Kanade muss man drei Fälle unterscheiden, die bei der Untersuchung einer Pixelregion auftreten können. Hierfür untersucht man den Strukturtenor J wie oben erwähnt auf seine Eigenwerte λ_1, λ_2 und die dazugehörigen Eigenvektoren:

1. Beide Eigenwerte gehen gegen 0, bzw. $\text{tr}(J) = j_{11} + j_{22} = \lambda_1 + \lambda_2 < \epsilon$:

Man kann in dieser Pixelregion keine Aussage über den optischen Fluss treffen, da die Umgebung einen sehr ähnlichen Grauwert aufweist (homogene Region).

2. Einer der Eigenwerte geht gegen 0, bzw. $\det(J) = j_{11}j_{22} - j_{12}^2 = \lambda_1\lambda_2 < \epsilon$:

Die Region hat das Aperturproblem, man befindet sich also in einer Kantenregion. In diesem Fall kann nur der Normalenfluss angegeben werden.

3. In allen anderen Fällen liefert die Pixelumgebung genügend Informationen, um mit der Lucas-Kanade-Methode einen sinnvollen optischen Flussvektor zu berechnen, was auch daran liegt, dass die Diagonalmatrix der Eigenwerte jetzt invertierbar ist. Da hier keiner der Eigenwerte gegen 0 geht, verändert sich der Grauwert in zwei Richtungen. Es handelt sich also um eine Ecke.

Offensichtlich ist diese Methode gegenüber der Normalenfluss-Berechnung im Vorteil, da sie diese mit beinhaltet. Es wird entsprechend dem Informationsgehalt der Region das Verfahren zur Berechnung des optischen Flusses angewandt, das den größten Erfolg verspricht. Ist genügend Information vorhanden, ist dieses besser als die Normalenfluss-Berechnung. Basierend auf dieser Methode gibt es noch weiterführende Ansätze zur verbesserten lokalen Berechnung des optischen Flusses. Diese Methoden können teilweise sehr akkurat den optischen Fluss berechnen, allerdings geben sie keine dichten Flussfelder aus, da für homogene Regionen keine Aussage getroffen werden kann.

Die Methode von Horn und Schunck

Hierin liegt ein großer Vorteil von globalen gegenüber lokalen Verfahren zur Berechnung des optischen Flusses. Im Speziellen soll nun die *Variationsmethode* von Horn und Schunck [HS81] vorgestellt werden, welche nicht nur eine kleine Nachbarschaft, sondern alle vorhandenen Pixel des Bildraumes Ω in die Berechnung der optischen Flussvektoren mit einfließen lässt. Der Begriff Variationsmethode beschreibt globale Verfahren, bei denen ein Energiefunktional minimiert wird.

Auch hier wird wieder vom linearisierten Optic Flow Constraint ausgegangen. Als weitere Annahme geht man von glatten Verschiebungsvektoren aus. Das heißt, es sollen abrupte Änderungen zwischen benachbarten Flussvektoren vermieden werden. Mathematisch lässt sich das zum Ausdruck bringen, indem der Wert

$$\int_{\Omega} (|\nabla u|^2 + |\nabla v|^2) dx dy \quad (2.13)$$

klein ist. Die Veränderungen der Flussvektorkomponenten u und v , und somit ihrer Gradienten, sollen also über den gesamten Bildraum hinweg quadriert aufsummiert werden. Das Ergebnis muss

dann minimiert werden. Ein Vorteil dieser zweiten Annahme liegt darin, dass selbst über homogene Regionen hinweg, in welchen keine Aussage über den optischen Fluss getroffen werden kann, eine Art Interpolation mithilfe der nahen aussagekräftigen Werte stattfindet, welche einen Wert für den optischen Fluss bestimmt.

Doch kommen wir nun zur Kombination der beiden getroffenen Annahmen im Energiefunktional nach Horn und Schunck:

$$\begin{aligned} E(u, v) &= \int_{\Omega} (D(u, v) + \alpha \cdot S(u, v)) \, dx dy = \int_{\Omega} ((f_x u + f_y v + f_t)^2 + \alpha \cdot (|\nabla u|^2 + |\nabla v|^2)) \, dx dy \\ &= \int_{\Omega} F(x, y, u, v, u_x, u_y, v_x, v_y) \, dx dy. \end{aligned} \quad (2.14)$$

Dieses Energiefunktional besteht nun also aus zwei Termen, einem Datenterm $D(u, v)$ und einem Glattheitsterm $S(u, v)$. Während der Datenterm für die eigentliche Bestimmung des optischen Flusses gedacht ist, welche eben nur in gewissen Regionen möglich ist, kann mithilfe des Glattheitsterms ein dichtes Flussfeld garantiert werden, indem in homogenen Regionen mit kleinem Datenterm, dieser Glattheitsterm überwiegt und einen Ausgleich zwischen den Regionen mit berechenbarem optischen Fluss schafft. Der Parameter α reguliert hier, wie glatt das berechnete Verschiebungsvektorfeld wird und muss > 0 gewählt werden. Je größer α gewählt wird, umso glatter wird das Bild.

Für die Minimierung des Energiefunktionals (2.14) müssen die folgenden Euler-Lagrange-Gleichungen (entsprechend der ersten Ableitung bei Lucas und Kanade) gleich 0 sein

$$\begin{aligned} \frac{\partial}{\partial x} F_{u_x} + \frac{\partial}{\partial y} F_{u_y} - F_u &= 0, \\ \frac{\partial}{\partial x} F_{v_x} + \frac{\partial}{\partial y} F_{v_y} - F_v &= 0. \end{aligned} \quad (2.15)$$

Dabei gelten die Neumann-Bedingungen an den Bildrändern

$$\begin{aligned} n^T \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} &= 0, \\ n^T \begin{pmatrix} F_{v_x} \\ F_{v_y} \end{pmatrix} &= 0, \end{aligned} \quad (2.16)$$

wobei n für die Normalvektoren auf den Bildrändern der Länge 1 steht. Die Euler-Lagrange Gleichungen (2.15), mit den entsprechenden Werten des Funktionals (2.14) eingesetzt, liefern die Gleichungen

$$\begin{aligned} \Delta u - \frac{1}{\alpha} f_x (f_x u + f_y v + f_t) &= 0, \\ \Delta v - \frac{1}{\alpha} f_y (f_x u + f_y v + f_t) &= 0. \end{aligned} \quad (2.17)$$

Hierbei wird der Laplace-Operator $\Delta f = \text{div}(\text{grad}(f))$ verwendet.

Diskretisiert man diese Gleichungen mit finiten Differenzen und schreibt sie abhängig vom Pixel i , ergibt sich:

$$\begin{aligned} \sum_{j \in N(i)} (u_j - u_i) - \frac{1}{\alpha} f_{xi} (f_{xi} u_i + f_{yi} v_i + f_{ti}) &= 0, \\ \sum_{j \in N(i)} (v_j - v_i) - \frac{1}{\alpha} f_{yi} (f_{xi} u_i + f_{yi} v_i + f_{ti}) &= 0. \end{aligned} \quad (2.18)$$

Die Pixel $j \in N(i)$ umfassen hier alle direkten Nachbarn des Pixels i , also maximal vier Stück, an den Rändern drei und an den Ecken zwei. Dies stellt ein sehr großes Gleichungssystem dar, da es für jedes Pixel i zwei Gleichungen und somit insgesamt $2N$ Unbekannte gibt. Schreibt man die Koeffizienten aber in Matrixform, so sind diese dafür recht spärlich besetzt.

Gelöst werden kann dieses Gleichungssystem beispielsweise mit dem iterativen Jacobi-Verfahren. Ausgehend von einem linearen Gleichungssystem der Form $Ax = b$ kann man die Matrix A aufteilen in einen Diagonalmatrix-Teil D und einen Restteil R , sodass $A = D - R$. Beginnend mit $k = 0$ wird dann folgende Iteration

$$x^{k+1} = D^{-1}(Rx^k + b) \quad (2.19)$$

solange durchgeführt, bis das Residuum $r^k = Ax^k - b$ klein genug wird. Als Abbruchbedingung kann das relative Residuum $\frac{|r^k|}{|r^0|} < \epsilon$ berechnet werden.

Angewandt für das Funktional von Horn und Schunck ergeben sich aus Gleichung (2.19) die Iterationsgleichungen:

$$\begin{aligned} u_i^{k+1} &= \frac{\sum_{j \in N(i)} u_j^k - \frac{1}{\alpha} f_{xi} (f_{yi} v_i^k + f_{ti})}{|N(i)| + \frac{1}{\alpha} f_{xi}^2}, \\ v_i^{k+1} &= \frac{\sum_{j \in N(i)} v_j^k - \frac{1}{\alpha} f_{yi} (f_{xi} u_i^k + f_{ti})}{|N(i)| + \frac{1}{\alpha} f_{yi}^2}. \end{aligned} \quad (2.20)$$

Hierbei steht $|N(i)|$ für die Anzahl der direkten Nachbarn des Pixel i . Die Initialisierung kann beispielsweise mit dem Nullvektor erfolgen, ist aber beliebig. Für eine effektivere Lösung des Gleichungssystems kann auch das Gauß-Seidel Verfahren angewendet werden, auf das aber hier nicht weiter eingegangen werden soll.

Verglichen mit dem zuvor vorgestellten Verfahren von Lucas und Kanade hat das Horn und Schunck Verfahren ganz klar den Vorteil, dass es ein dichtes Flussfeld berechnen kann, das keine Lücken mehr enthält. Weiterhin besteht hier noch Verbesserungspotenzial, da statt dem linearisierten OFC auch andere Datenterme verwendet werden können oder auch mehrere Datenterme in einer Summe zugleich, die dann zum Beispiel robust gegen Beleuchtungsänderung sein können. Andererseits gestaltet sich die Berechnung des optischen Flusses komplexer, da ein sehr großes Gleichungssystem gelöst werden muss.

Robuste Datenterme

Eine Möglichkeit, den Datenterm zu verbessern, ohne ihn komplett auszutauschen, besteht darin, ihn robuster gegen Ausreißer zu machen [BA91]. Im Horn und Schunck Verfahren geht der linearisierte OFC im Quadrat ins Energiefunktional ein. Dies lässt Ausreißer die Gesamtenergie recht stark anheben und macht das Verfahren dementsprechend anfällig für Fehler. Die Idee bei den robusten Datentermen besteht im Grunde darin, den quadrierten Datenterm subquadratisch zu machen und somit die Auswirkung von Ausreißern einzudämmen. Somit werden auch stärkere Abweichungen von den Daten toleriert.

Eine gute Funktion zur Robustifizierung wäre beispielsweise diese:

$$\Psi(s^2) = \sqrt{s^2 + \epsilon^2}. \quad (2.21)$$

Von dem quadratischen Datenterm wird hier einfach die Wurzel gezogen, wobei ein kleines $\epsilon > 0$ quadratisch hinzuaddiert wird, um die Differenzierbarkeit der Funktion zu gewährleisten.

Bei der Robustifizierung mehrerer Datenterme gibt es nun allerdings zwei Möglichkeiten. Die gemeinsame Robustifizierung oder die separate Robustifizierung. Man betrachte folgende vereinfachte Schreibweise eines Energiefunktionsals nach Horn und Schunck mit Datentermen $D_i(u, v)$ und Glattheitsterm $S(u, v)$:

$$E(u, v) = \int_{\Omega} \sum_{i=1}^n D_i(u, v) + \alpha \cdot S(u, v) \, dx dy. \quad (2.22)$$

Eine gemeinsame Robustifizierung macht Sinn bei zueinander in Beziehung stehenden Datentermen, wie z.B. den Datentermen für die RGB Farbkanäle und sieht folgendermaßen aus:

$$E(u, v) = \int_{\Omega} \Psi\left(\sum_{i=1}^n \lambda_i D_i(u, v)\right) + \alpha \cdot S(u, v) \, dx dy. \quad (2.23)$$

Dagegen läuft die separate Robustifizierung bei unabhängigen Datentermen so ab:

$$E(u, v) = \int_{\Omega} \sum_{i=1}^n \lambda_i \Psi(D_i(u, v)) + \alpha \cdot S(u, v) \, dx dy. \quad (2.24)$$

Die verschiedenen λ_i sind dabei die Gewichte der entsprechenden Datenterme $D_i(u, v)$.

Eine Robustifizierung kann übrigens durchaus nicht nur bei Datentermen, sondern auch bei Glattheitstermen Sinn machen. Sie erlaubt dann ebenfalls stärkere Abweichungen in der Glattheit, was vor allem für die Erhaltung von Kanten sinnvoll ist.

2.1.3 Verfahren von Demetz *et al.*

Nach der Vorstellung des Verfahrens von Horn und Schunck wurden noch viele weitere Variationsansätze zur Berechnung des optischen Flusses erarbeitet. Um aber den Umfang dieser Arbeit nicht zu sprengen, sei an dieser Stelle nur noch ein letztes Verfahren von Demetz *et al.* [DSV⁺14] vorgestellt.

Dieses Verfahren ist es auch, mit welchem die Ausgangsdaten des optischen Flusses erstellt wurden, mit deren Hilfe in dieser Diplomarbeit die Fundamentalmatrix bestimmt werden soll.

Es basiert grundsätzlich auf folgendem Modell zur Minimierung eines Energiefunktional:

$$E(w, c) = \int_{\Omega} (D(w, c) + R_{flow}(w) + R_{illum}(c)) dx. \quad (2.25)$$

Hier ist zu beachten, dass das Energiefunktional zusätzlich zum optischen Flussvektor $w = (u, v)^T$ auch von einem Koeffizientenvektor $c \in \mathbb{R}^n$ abhängig ist, welcher Beleuchtungsänderungen beschreibt. Dieser wird im nächsten Abschnitt noch genauer beschrieben. Inspiriert vom klassischen Ansatz von Cornelius und Kanade [CK83] werden w und c hier gleichzeitig durch Minimierung des Energiefunktional bestimmt. Im Groben kann man dabei sagen, dass das Energiefunktional aus einem Datenterm $D(w, c)$ und zwei Glattheitstermen (auch *Regularisierungsterme* genannt) besteht: einem Term $R_{flow}(w)$, der wie bei Horn und Schunck für einen glatten Verlauf der Flussvektoren verantwortlich ist und einem weiteren Term $R_{illum}(c)$, für einen gewünschten Verlauf der Beleuchtungsänderungen.

Datenterm

Da bereits der Datenterm unter anderem vom Koeffizientenvektor c abhängt, wird nun erklärt, was dieser mit Beleuchtungsänderungen zwischen den Bildern zu tun hat. Hierbei werfen wir einen Blick auf die sogenannte *Brightness Transfer Function* (kurz: *BTF*), welche von Grossberg und Nayar [GN04] vorgestellt wurde:

$$\Phi(c, f) = \bar{\phi}(f) + \sum_{j=1}^n c_j \cdot \phi_j(f). \quad (2.26)$$

Diese Funktion weist jedem Pixel der Helligkeit f die veränderte Helligkeit $\Phi(c, f)$ zu, die dessen korrespondierendes Pixel im anderen Bild aufgrund von Beleuchtungsveränderungen aufweist. Die BTF besteht aus n Basisfunktionen ϕ_j und einer mittleren Brightness Transfer Function $\bar{\phi}$, welche zunächst mithilfe von Trainingsbildern mit bekanntem optischen Fluss gelernt werden müssen. Diese Trainingsbilder wurden hier vom KITTI Benchmark aus dem Automobilbereich [GLU12] gewählt, da für diese einerseits ein genaues optisches Flussfeld bekannt ist und sie andererseits naturgetreue, herausfordernde Beleuchtungsänderungen beinhalten.

Damit diese Diplomarbeit an dieser Stelle auf das Wesentlichste beschränkt bleibt, sei der interessierte Leser für genauere Informationen zur Wahl und Evaluation der zu lernenden Basisfunktionen der Brightness Transfer Function auf das Paper zum Verfahren von Demetz *et al.* [DSV⁺14] verwiesen.

Durch die Wahl der Koeffizienten c_j wird aufgrund dieser Basis nun also für jedes Pixel eine konkrete BTF ausgewählt. Die Basisfunktionen ϕ_j sind dabei für alle Pixel gleich, während sich die Koeffizientenvektoren c von Pixel zu Pixel unterscheiden. Deshalb sollen diese zusammen mit dem ebenfalls für jedes Pixel zu bestimmenden optischen Flussvektor w durch Minimierung des Energiefunktional aus Gleichung (2.25) berechnet werden. Dadurch werden im Modell von Demetz *et al.* mögliche

Beleuchtungsänderungen berücksichtigt, was diesen Ansatz sehr robust gegenüber unkontrollierten Beleuchtungsänderungen macht.

Der Datenterm sieht nun folgendermaßen aus:

$$D(w, c) = D_{hell}(w, c) + \nu D_{grad}(w, c). \quad (2.27)$$

Das bedeutet, es werden zwei Terme verwendet, welche mit dem positiven Gewicht ν gegeneinander gewichtet werden. Einer steht für die Helligkeitskonstanz und der andere für Gradientenkonstanz. Ersterer wird nun wie folgt definiert:

$$D_{hell}(w, c) = \Psi_d \left((f_2(x+w) - \Phi(c(x), f_1(x)))^2 \right). \quad (2.28)$$

Im Grunde besteht der Helligkeits-Datenterm also aus dem Farbunterschied des Korrespondenzpixels zu seinem Originalpixel, von welchem allerdings der angepasste Farbwert entsprechend der BTF berechnet wird. Diese Differenz wird quadriert, damit im Energiefunktional keine gegenseitige Auslöschung von negativen und positiven Differenzen entsteht. Schließlich wird der Datenterm mit der subquadratischen Bestrafungsfunktion

$$\Psi_d(s^2) = 2\lambda_d^2 \sqrt{1 + \frac{s^2}{\lambda_d^2}} \quad (2.29)$$

robustifiziert, um dem quadratischen Einfluss von Ausreißern entgegen zu wirken, wie bereits im vorhergehenden Abschnitt beschrieben wurde. Dabei stellt λ_d den zu wählenden Kontrastparameter dar.

Der Datenterm für die Gradientenkonstanz

$$D_{grad}(w, c) = \Psi_d \left(\|\nabla f_2(x+w) - \nabla \Phi(c(x), f_1(x))\|_2^2 \right) \quad (2.30)$$

vergleicht entsprechend die Gradienten der oben erklärten Helligkeitswerte, wobei die gleiche Funktion $\Psi_d(s^2)$ zur Robustifizierung verwendet wird.

Fluss-Regularisierungsterm zweiter Ordnung

Doch der Datenterm allein kann, wie bei Horn und Schunck bereits gezeigt, kein glattes, aussagekräftiges Flussfeld garantieren. Daher gibt es auch hier einen Regularisierungsterm, der große Änderungen des optischen Flusses vermeiden soll, indem er diese mit hohen Werten im Energiefunktional bestraft. Er sieht folgendermaßen aus:

$$R_{flow}(w) = \alpha \cdot \Psi_s(\|Hu\|_F^2 + \|Hv\|_F^2). \quad (2.31)$$

In diesem Fall soll nicht, wie etwa bei Horn und Schunck, der Betrag der ersten Ableitung (also des Gradienten) der Flusskomponenten u und v klein bleiben, sondern es wird stattdessen die zweite

Ableitung verwendet. Somit stellt die Frobenius-Norm der Hessematrix über den Flusskomponenten $\|Hu\|_F$ und $\|Hv\|_F$ das zu minimierende Maß dar.

Der Vorteil, der sich durch die Verwendung der zweiten Ableitung ergibt, liegt darin, dass diese sich besser für affine Lösungen eignet. Dies ist gerade im Hinblick auf die Bilder des KITTI-Benchmarks [GLU12] interessant, weil dort durch die Fortbewegung der Automobile eine approximativ affine Transformation der Umgebung zwischen den Bildern stattfindet, welche hier bei der Bestimmung der Flussvektoren besser berücksichtigt wird.

Ansonsten stellt α wieder ein positives Gewicht dar, welches bestimmt, wie stark die Glattheit der Flussvektoren ins Gewicht fallen soll. Die Terme der zweiten Ableitung werden auch hier quadriert, aufsummiert und entsprechend der Gleichung

$$\Psi_s(s^2) = 2\lambda_s^2 \sqrt{1 + \frac{s^2}{\lambda_s^2}} \quad (2.32)$$

robustifiziert, wie es zuvor schon beim Datenterm geschehen ist.

Koeffizienten-Regularisierungsterm

Da das Energiefunktional dieses Mal nicht nur vom optischen Flussvektor w , sondern auch vom Koeffizientenvektor c abhängt, welcher die BTF bestimmt, macht es Sinn, für dessen Verlauf über die Pixel hinweg auch eine Glattheits-Annahme zu treffen. Dies erfolgt, indem der zweite Regularisierungsterm

$$R_{illum}(c) = \beta \cdot \sum_{i=1}^2 \Psi_{illum}^i \left(\sum_{j=1}^n \gamma_j (r_i^T \nabla c_j)^2 \right) \quad (2.33)$$

zum Energiefunktional hinzugefügt wird. β stellt wiederum das positive Gewicht dar, mit dem der Term ins Energiefunktional einfließt. Auffällig ist allerdings, dass es zwei verschiedene Bestrafungsfunktionen Ψ_{illum}^1 und Ψ_{illum}^2 gibt, deren Summe den Wert des Regularisierungsterms ausmacht. Das kommt daher, dass in der Umgebung eines jeden Pixels ein Regularisierungstensor nach Zimmer [ZBW11] berechnet wird, dessen Eigenvektoren die Richtungen senkrecht zu Kanten r_1 und entlang zu ihnen $r_2 = r_1^\perp$ darstellen. Der Verlauf der Gradienten des Koeffizientenvektors ∇c_j wird also unterschiedlich bewertet, je nachdem, ob er parallel zu einer Kante oder senkrecht dazu verläuft. Daher nennt man dies auch einen anisotropen Regularisierungsterm.

Das bedeutet, dass über Kanten hinweg (in r_1 -Richtung) ein kantenverstärkender Perona-Malik-Regularisierer [PM90]

$$\Psi_{illum}^1(s^2) = \lambda_c^2 \log\left(1 + \frac{s^2}{\lambda_c^2}\right) \quad (2.34)$$

verwendet wird, während entlang von Kanten (in r_2 -Richtung) der kantenerhaltende Charbonnier-Regularisierer [CBFAB94]

$$\Psi_{illum}^2(s^2) = 2\lambda_c^2 \sqrt{1 + \frac{s^2}{\lambda_c^2}} \quad (2.35)$$

zum Einsatz kommt, der bereits für den Datenterm und die Fluss-Regularisierung verwendet wurde.

Die Bestrafungsfunktionen Ψ_{illum}^i robustifizieren dabei die Koeffizienten gemeinsam. Deren Gradienten werden zwar einzeln im Skalarprodukt mit den Richtungen r_1 und r_2 verglichen, jedoch werden diese Skalarprodukte quadriert und über alle Koeffizienten gewichtet aufsummiert. Zur Wahl der entsprechenden Gewichte γ_i sei wiederum auf das Paper von Demetz *et al.* [DSV⁺14] verwiesen.

Umgangssprachlich könnte man obige Vorgehensweise so erklären: Kanten trennen in einem Bild oft unterschiedlich beleuchtete Bereiche voneinander ab. Dies ist z.B. an der Grenze von Schatten zu Nicht-Schatten der Fall. Diese Bereiche sollten dann auch bei deren Beleuchtung unabhängig voneinander angesehen werden. Deshalb macht es Sinn, entlang von Kanten, wenn also kein Übertreten dieser Grenze stattfindet, wie von Horn und Schunck gewohnt, eine Glattheitsannahme zu treffen, sodass die Beleuchtungs-Koeffizienten sich nicht groß verändern sollen. Bewegt man sich aber senkrecht zur Kante und übertritt diese Grenze hinüber zum Bereich mit neuer, unabhängiger Beleuchtung, so macht es Sinn, die Unabhängigkeit in der Beleuchtung zu wahren, indem eine kantenverstärkende Funktion eingesetzt wird, welche die Bereiche voneinander abgrenzt.

Fluss-Regularisierungsterm erster Ordnung

Alternativ zum Fluss-Regularisierungsterm zweiter Ordnung kann für Bildpaare mit hauptsächlich fronto-parallelen Verschiebungen ein Term der ersten Ordnung sinnvoller sein. Dieser sieht folgendermaßen aus:

$$R_{flow}(w) = \alpha \cdot \sum_{i=1}^2 \Psi_s^i \left(|r_i^T \nabla u|^2 + |r_i^T \nabla v|^2 \right). \quad (2.36)$$

Anstatt der Hessematrix wird also der Gradient verwendet, wobei hier wie beim Koeffizientenregularisierungsterm eine anisotrope Variante mit den Regularisierern Ψ_s^1 und Ψ_s^2 gewählt wird. Diese entsprechen den Funktionen Ψ_{illum}^1 und Ψ_{illum}^2 und erreichen somit, dass über Bildkanten hinweg Sprünge im Gradienten des Flusses toleriert werden, während entlang von Kanten dessen Änderungen möglichst klein bleiben sollen.

Nachdem nun alle Terme des Energiefunktionals nach Demetz *et al.* erläutert wurden, stellt sich noch die Frage, wie dieses gelöst wird, also wie dessen Minimierung vonstatten geht.

An dieser Stelle soll dies allerdings nicht weiter vertieft werden. Es soll nur erwähnt bleiben, dass die gesuchten Minima mithilfe einer sogenannten *Warping-Strategie* berechnet werden können, die schon von Brox *et al.* [BBPW04] verwendet wurde und sich gerade für größere Verschiebungen eignet. Für mehr Informationen sei auch hier wieder auf die Ausführungen von Demetz *et al.* [DSV⁺14] verwiesen.

Abschließend lässt sich über diesen Variationsansatz sagen, dass er ein aktuelles, robustes Verfahren zur Berechnung des optischen Flusses darstellt. Durch die Berücksichtigung von Beleuchtungsänderungen im variationalen Modell wird sogar bei anspruchsvollen, unkontrollierten Beleuchtungen realer Bildpaare ein recht genaues optisches Flussfeld berechnet, welches auch größere Pixelverschiebungen berücksichtigt.

2.2 Feature-Matches

Nun folgt zunächst ein Abschnitt über ein Verfahren, das es uns erlaubt, markante Punkte (sog. *Feature-Punkte*) in Bildern zu finden und diese so zu beschreiben, dass sie in anderen Bildern trotz veränderter Größe und weiteren Transformationen wiedergefunden werden können. Dieses Verfahren stammt von Lowe und nennt sich *Scale Invariant Feature Transform* [Low04]. Es soll bei der Bestimmung der Fundamentalmatrix in Kapitel 4 als Alternative zur Verwendung von dichten Verschiebungsvektorfeldern (dem optischen Fluss) dienen.

Im Einzelnen ist das Besondere dieses Verfahrens, dass die erkannten Feature-Punkte invariant einerseits unter Skalierung, Verschiebung und Rotation und andererseits unter additiven und multiplikativen Beleuchtungsänderungen sind. Die Beschreibung eines Punktes bleibt also unter verschiedenen Beleuchtungen und unter affinen Transformationen des Bildes immer gleich. Dies führt dazu, dass man in Bildpaaren mit diesem Verfahren sehr genaue Punktkorrespondenzen finden kann, weshalb es auch in dieser Diplomarbeit seine Anwendung findet.

Doch wie genau findet und beschreibt man die besagten Feature-Punkte? Hierfür müssen folgende Schritte durchlaufen werden [Low04]:

Detektion der Skalenraum-Extrema

Zunächst muss über allen Bildpositionen und Skalierungen im *Skalenraum* (engl. *Scale Space*) eine Suche stattfinden. In dieser wird eine Difference-of-Gaussian Funktion verwendet, um sogenannte *Interest-Points* zu finden, welche invariant gegenüber Änderungen der Größe und Orientierung sind.

Akkurate Key-Point-Lokalisierung

An jedem Interest-Point wird ein detailliertes Modell eingesetzt, um deren genaue Position und Skalierung zu bestimmen. Basierend auf verschiedenen Stabilitätsmaßen werden aus den Interest-Points die *Key-Points* ausgewählt.

Richtungszuweisung

Jedem Key-Point werden eine oder mehrere Orientierungen zugewiesen. Diese basieren auf den Richtungen des lokalen Bildgradienten.

SIFT-Key-Point-Deskriptor

In der Umgebung jedes Key-Points werden die lokalen Bildgradienten entsprechend ihrer Skalierung gemessen und in eine Repräsentation transformiert, die signifikante Verzerrungen und Beleuchtungsänderungen zulässt.

Anschließend wird noch gezeigt, wie das Matching der SIFT-Key-Point-Deskriptoren funktionieren kann, um tatsächliche Punktkorrespondenzen zu erhalten. Schließlich werden noch kurz alternative Feature-Extraktoren genannt.

2.2.1 Detektion der Skalenraum-Extrema

Um die Interest-Points, eine erste Auswahl potentiell markanter Feature-Punkte, zu finden, bedienen wir uns des Konzeptes des *Skalenraums* (engl. *Scale Space*) [Iij59, Iij62, WII97, Lin94b].

Der Skalenraum

Diesen kann man sich beispielsweise vorstellen als eine Pyramide bestehend aus mehreren Instanzen desselben Bildes, nur sind diese nach oben hin immer stärker mit einem Gauß-Filter geglättet. Man kann diesen Gauß-Skalenraum also als eine Funktion $L(x, y, \sigma)$ beschreiben, die durch Gauß-Faltung eines Bildes $I(x, y)$ mit der von der Standardabweichung σ abhängigen Gauß-Funktion $G(x, y, \sigma)$ entsteht:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y). \quad (2.37)$$

Hierbei stellt $*$ den Faltungsoperator dar, mit

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}. \quad (2.38)$$

Difference-of-Gaussians

Ausgehend vom Gauß-Skalenraum eines Bildes kann man nun als weiteren Schritt zwischen je zwei benachbarten Bildern verschiedener Skalierungen eine *Difference-of-Gaussians* erstellen. Hierfür wird einfach die Differenz der entsprechenden Werte der beiden Bilder gebildet (siehe Abb. 2.2):

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \quad (2.39)$$

Stellt man den Skalenraum und somit die Difference-of-Gaussians regelmäßig auf, indem sich der σ -Wert je um einen Faktor k unterscheidet und wählt k als n -te Wurzel von 2, sodass sich alle n Schritte σ verdoppelt, so kann man den Skalenraum, wie in Abb. 2.2 zu sehen, in entsprechende Oktaven aufteilen (mit jeder Oktave verdoppelt sich σ). Da mit immer stärkerer Gauß-Faltung immer mehr hohe Frequenzen im Bild eliminiert werden, macht es hier Sinn, zu jeder Oktave dessen Dimensionen zu halbieren, also sowohl in x - als auch in y -Richtung nur noch jedes zweite Pixel zu verwenden. Dadurch kann man viel Rechenaufwand sparen.

Die Verwendung der Difference-of-Gaussians bietet einige Vorteile. Sie ist sehr effizient und zu ihrer Berechnung ist nur eine einfache Bildsubtraktion notwendig. Weiterhin produziert sie stabile Bildfeatures, welche skaleninvariant sind. Durch die Eigenschaft des Gauß-Filters, hohe Frequenzen zu eliminieren, verhält sie sich auch wie ein Bandpass-Filter. Das bedeutet, dass die Differences-of-Gaussians unterschiedlicher Skalierungen jeweils Informationen zu unterschiedlichen Frequenzbereichen enthalten.

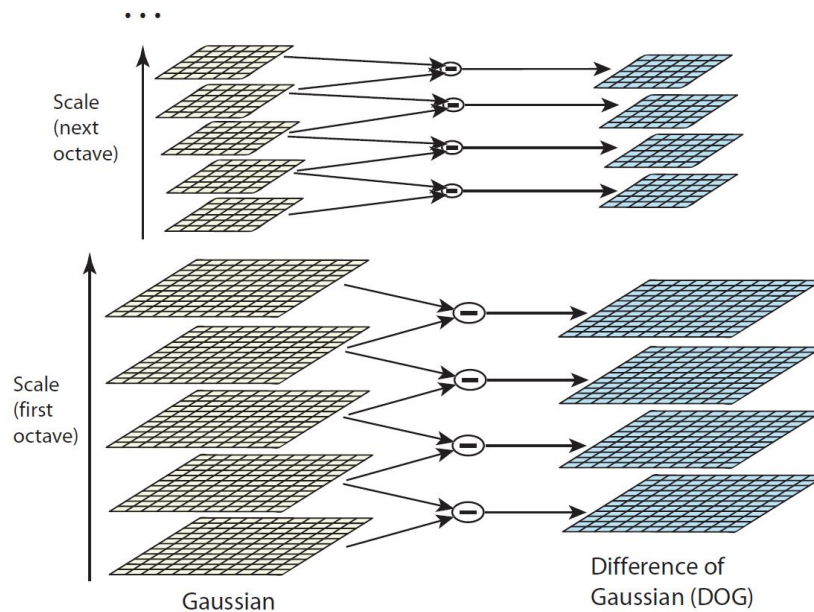


Abbildung 2.2: Mit jeder Oktave verkleinern sich die Bilder des Gauß-Skalenraums. Die Differences-of-Gaussians entstehen durch Subtraktion benachbarter Bilder im Skalenraum [Low04].

Desweiteren kann gezeigt werden, dass die Difference-of-Gaussians eine Approximation der *normalisierten Laplacian-of-Gaussians* ist:

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \Delta G \quad (2.40)$$

$$\sim \sigma^2 \Delta G.$$

Lindeberg zeigte, dass die Normalisierung des Laplacian mit dem Faktor σ^2 wahre Skaleninvarianz mit sich bringt [Lin94a]. Am Ort der größten Farbänderung einer charakteristischen Skalierung hat die Difference-of-Gaussians also extremale Werte. Daher müssen nun Extrema der Difference-of-Gaussians gefunden werden, deren Position und Skalierung schließlich die Interest-Points bilden. Um diese Extrema zu finden, betrachte man Abb.2.3. Durch einen einfachen Vergleich jedes Wertes mit seinen 26 Nachbarn im Skalenraum (8 direkte Nachbarn, jeweils 9 in der Skalierung darüber und darunter) können diese identifiziert werden.

2.2.2 Akkurate Key-Point-Lokalisierung

Nachdem die Interest-Points gefunden sind, gilt es zunächst, deren Lage mithilfe der umliegenden Punkte im Skalenraum genauer zu bestimmen. Dies macht von daher Sinn, dass der Skalenraum durch konkrete Skalierungen und Pixellokationen diskretisiert ist und die wahren Extrema in den Zwischenräumen der Skalierungen und Pixel liegen können.

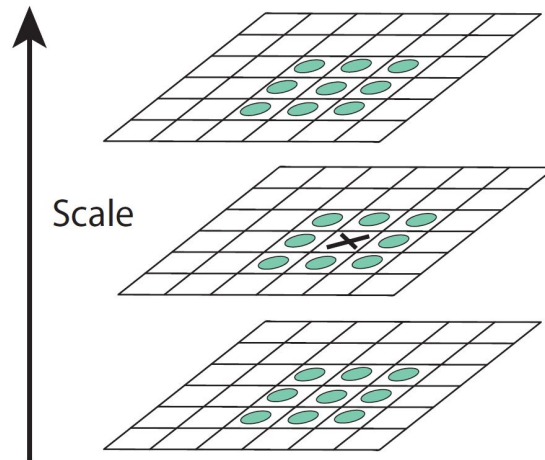


Abbildung 2.3: Im Skalenraum findet man Extrema durch Vergleich eines Elements mit seinen Nachbarn in der gleichen Skalierung, aber auch mit den Nachbarn in der nächsthöheren und -tieferen Skalierung [Low04].

Mithilfe einer Taylor-Expansion kann somit um die Interest-Points $\mathbf{x}_i = (x_i, y_i, \sigma_i)$ eine quadratische Funktion zur Interpolation dieser Extrema verwendet werden, deren Lösung eine deutliche Verbesserung für das spätere Matching und die Stabilität mit sich bringt:

$$D(\mathbf{x}) \approx D(\mathbf{x}_i) + \nabla D(\mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_i)^T H(D(\mathbf{x}_i)) (\mathbf{x} - \mathbf{x}_i). \quad (2.41)$$

Der Gradient ∇D sowie die Hesse-Matrix $H(D)$ können dabei mithilfe der benachbarten Pixel durch finite Differenzen approximiert werden. Setzt man nun den Offset $h = \mathbf{x} - \mathbf{x}_i$, so erhält man den Extrempunkt der quadratischen Funktion durch Nullsetzen der Ableitung:

$$h = -H^{-1}(D(\mathbf{x}_i)) \nabla D(\mathbf{x}_i). \quad (2.42)$$

Die neue Position und Skalierung des Feature-Punktes ergibt sich durch $\mathbf{x}_{neu} = \mathbf{x}_i + h$. Durch Einsetzen in die quadratische Funktion kann man schließlich den neuen Wert des Extremums berechnen:

$$D(\mathbf{x}_{neu}) = D(\mathbf{x}_i) + \frac{1}{2} \nabla D(\mathbf{x}_i)^T h. \quad (2.43)$$

Ergibt der Betrag des Offsets h mehr als 0.5 in einer seiner Dimensionen, so liegt das Extremum näher an einem benachbarten Pixel, welches somit nochmals als Zentrum der Interpolation betrachtet werden muss.

Weiterhin wird mit dem Neuberechneten Extremwert des Feature-Punktes ein Kontrast zu seiner Umgebung berechnet. Die Feature-Punkte mit einem Kontrast unterhalb eines Schwellwerts werden

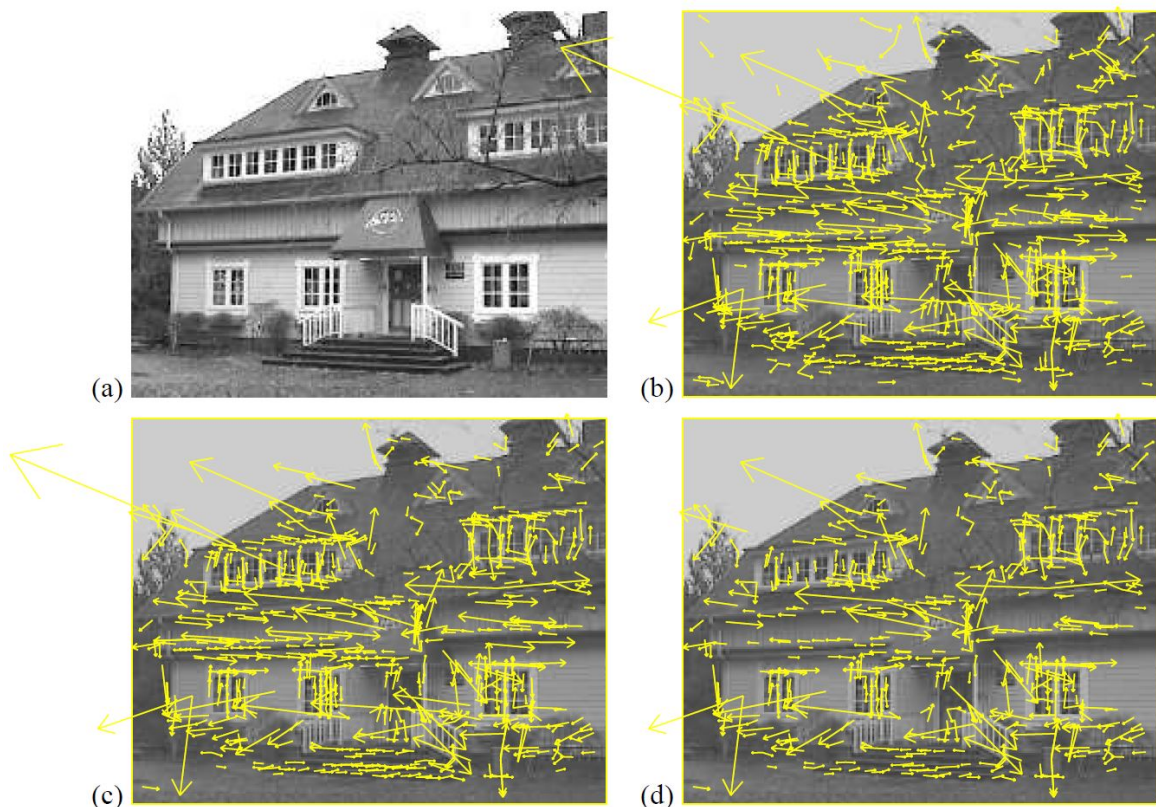


Abbildung 2.4: Die Pfeile geben die Position, Skalierung und Orientierung von Feature-Points im Bild an: **(a)** Ausgangsbild, **(b)** erste Auswahl an Interest-Points, **(c)** übrig gebliebene Interest-Points mit einem höheren Kontrast, **(d)** übrig gebliebene Interest-Points, die nicht nur Kanteninformation enthalten, bilden die finalen Key-Points [Low04].

verworfen. In Abb. 2.4 zeigt Teil (b) alle zunächst gewählten Interest-Points, dargestellt als Pfeile. Der Ansatzpunkt der Pfeile stellt deren Position dar, die Länge die entsprechende Skalierung und die Richtung steht für die Orientierung der Feature-Punkte, welche im Folgenden noch bestimmt wird. Part (c) zeigt nur noch die Punkte mit entsprechend höherem Kontrast.

Für die Stabilität der ausgewählten Feature-Punkte und anschließend für das Matching muss noch beachtet werden, dass entlang von Bildkanten viele Punkte gefunden werden, da die Difference-of-Gaussians-Funktion stark auf die Farbänderung senkrecht zu Kanten anschlägt. Diese Punkte entlang derselben Kante bergen die Gefahr, später beim Matching verwechselt zu werden. Deshalb sollen die Punkte eliminiert werden, welche nur in eine Richtung bedeutende Gradienteninformation enthalten. Anders gesagt sind die interessanten Punkte die, die in mehrere Richtungen Information enthalten, wie z.B. Eckpunkte. Um diese Differenzierung treffen zu können, nehmen wir wieder die Hesse-Matrix zuhilfe:

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}. \quad (2.44)$$

Deren Eigenwerte λ_1 und λ_2 geben Auskunft darüber, wieviel Information die Umgebung des gewählten Pixels in Richtung der zugehörigen Eigenvektoren enthält. Da bis hier nur Extrempunkte mit einem gewissen Kontrast übrig sind, kann davon ausgegangen werden, dass der größere Eigenwert λ_1 eine ausreichende Mindestgröße haben muss. Ist aber im Verhältnis dazu λ_2 sehr viel kleiner, so enthält die Pixelumgebung keine bedeutenden Informationen mehr in einer zweiten Richtung, es handelt sich also um einen Punkt entlang einer Kante.

Um keine Eigenwertanalyse der Matrix durchführen zu müssen, verwendet man hier zwei schneller zu berechnende Werte, die helfen, eine Entscheidung zu treffen: zum einen die *Spur* der Hesse-Matrix

$$tr(H) = D_{xx} + D_{yy} = \lambda_1 + \lambda_2 \quad (2.45)$$

und zum anderen ihre *Determinante*

$$det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \lambda_1 \cdot \lambda_2. \quad (2.46)$$

Wäre der Wert der Determinante negativ, so wäre die untersuchte Stelle kein Extremum. Man kann hier also von einer positiven Determinante ausgehen. Da im Moment der Größenunterschied der beiden Eigenwerte interessant ist, sei im Folgenden $\lambda_1 = r \cdot \lambda_2$ (wobei λ_1 der größere Eigenwert ist, und damit die Konditionszahl $r \geq 1$). Somit gilt:

$$\frac{tr(H)^2}{det(H)} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} = \frac{(r\lambda_2 + \lambda_2)^2}{r\lambda_2^2} = \frac{(r+1)^2}{r}. \quad (2.47)$$

Nun kann man eine Grenze für die Konditionszahl r der Eigenwerte wählen, womit sich folgendes Stabilitätskriterium ergibt:

$$\frac{tr(H)^2}{det(H)} < \frac{(r+1)^2}{r} \quad (2.48)$$

Auf diese Weise kann man sehr effizient wie gewünscht Kantenpunkte von solchen mit mehr Informationsgehalt, und somit eindeutiger beschreibbaren Punkten, unterscheiden. Im Teil (d) von Abb. 2.4 bleiben entsprechend nur noch auf diese Weise ausgewählte Key-Points übrig. Lowe wählte für dieses Bild ein Verhältnis von $r = 10$ [Low04].

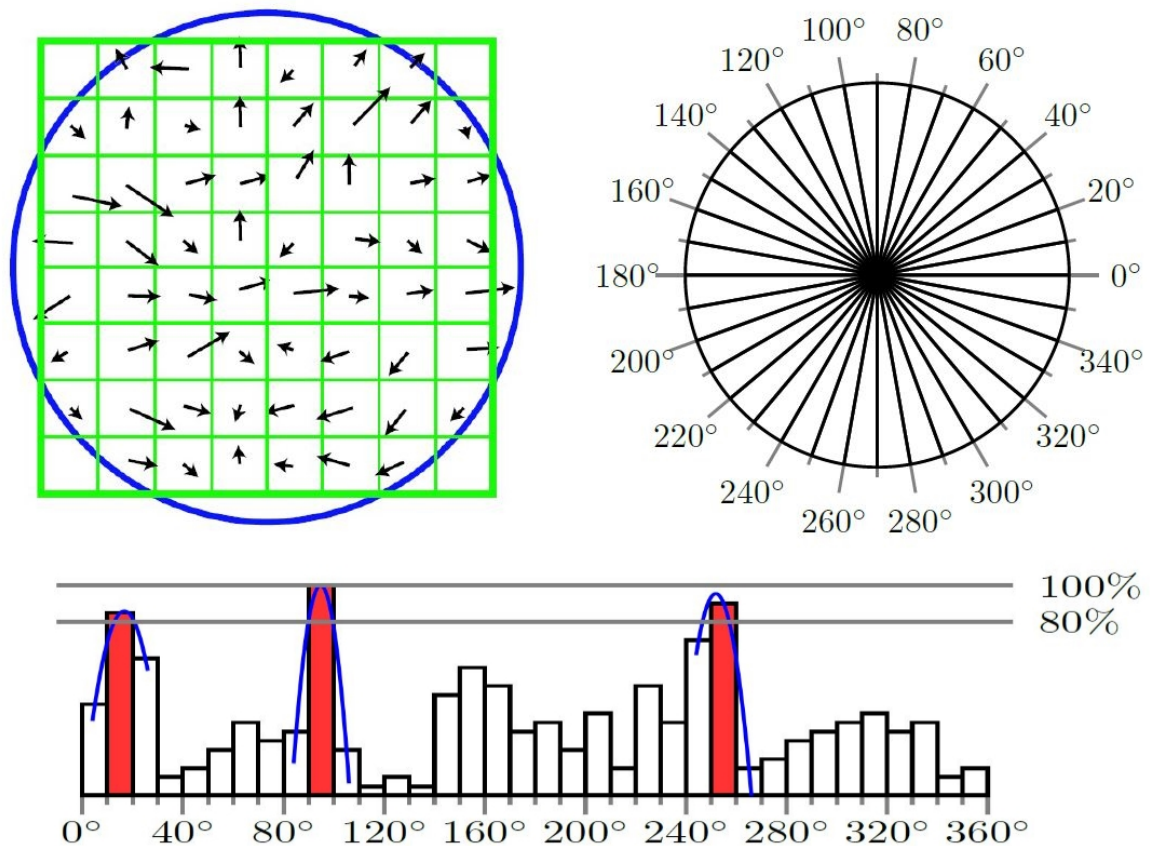


Abbildung 2.5: Links oben: Bildgradienten in der Kreisumgebung eines Key-Points, Rechts oben: Einteilung der Winkel von 0-360° in 36 gleich große Bereiche, Unten: Histogramm der Gradienten mit den aufsummierten Gradienten-Beträgen der jeweiligen Bereiche. Zweimaliges Klonen des Key-Points und genaue Bestimmung der Orientierungen durch Parabeln [Bru12a].

2.2.3 Richtungszuweisung

Als nächstes ist es nötig, jedem Key-Point eine Orientierung zuzuweisen. Somit kann später der Deskriptor dieses Punktes relativ zu dieser Richtung dargestellt werden. Das verleiht dem Deskriptor Invarianz gegenüber Rotation. Dazu betrachte man Abb. 2.5. An jeder Key-Point-Position bestimme man die Nachbarpixel in dieser Skalierung innerhalb eines Radius von 1.5σ . Zu jedem dieser Pixel der Kreisnachbarschaft wird nun der Betrag und die Richtung seines Bildgradienten mithilfe der jeweiligen Nachbarschaftspixel berechnet und mit der Gauß-Funktion gewichtet. Nun findet eine Unterteilung aller möglichen Richtungen von 0 - 360° in 36 gleich große Bereiche, die je ein 10°-Intervall abdecken, statt.

Somit lässt sich ein Histogramm erstellen (siehe Abb. 2.5 unten), in welchem alle gaußgewichteten Gradientenbeträge der gezeigten Kreismachbarschaft in ihrem zugehörigen Richtungsbereich hinzuaddiert werden. Die Orientierung des Key-Points ist nun definiert durch den Bereich des Histogramms mit dem größten Wert. Sollte es noch weitere Richtungen geben, in deren Bereich ähnlich große Werte erreicht wie der der dominanten Orientierung, so wird der Key-Point kurzerhand kopiert und als separater Key-Point mit gleicher Position und Skalierung, aber unterschiedlicher Orientierung gespeichert. Als Grenze gibt Lowe hier an, dass 80% des Bereichs mit dominanter Orientierung erreicht werden müssen, um den Key-Point zu klonen.

Um die Orientierung wiederum genauer bestimmen zu können, wird auch hier eine Interpolation durchgeführt, indem durch die drei Histogrammwerte, die jedem Maximum am nächsten stehen, eine Parabel gelegt wird. Deren Scheitelpunkt repräsentiert die verbesserte Orientierung des Key-Points. Und um diesen berechneten Winkel werden nun alle einzelnen Gradientenpositionen und Gradientenwinkel zurückgerechnet, damit im Folgenden eine rotationsinvariante Darstellung des SIFT-Deskriptors erreicht werden kann.

2.2.4 SIFT-Key-Point-Deskriptor

Der SIFT-Key-Point-Deskriptor soll nun jeden Feature-Punkt eindeutig mithilfe eines Vektors beschreiben. Dieser Vektor besteht aus 128 Gleitkommazahlen, welche ähnlich zum vorhergehenden Abschnitt der Richtungszuweisung 16 aneinandergereihte Histogramme mit je 8 gleichmäßig verteilten Richtungen darstellen. Dieser Vektor ist ein Einheitsvektor, was bedeutet er ist auf die Länge 1 normalisiert.

Doch wieso 16 Histogramme, wenn doch zuvor nur ein Histogramm zur Findung der Orientierung nötig war?

Das liegt daran, dass die kreisförmige Pixelumgebung nun in 4x4 Blöcke unterteilt wird. Abb. 2.6 stellt diese Unterteilung beispielhaft für 2x2 Blöcke dar. Jeder dieser Blöcke bekommt sein eigenes Histogramm, welches auch eine gröbere Aufteilung der Richtungsbereiche hat, nämlich 8 Stück, welche je ein Intervall von 45° abdecken. Mit dieser gröberen Aufteilung geht eine größere Robustheit des Deskriptors einher, da geringe Gradientenabweichungen so weniger Einfluss haben.

Ein Problem ergibt sich allerdings durch diese Aufteilung an den Übergängen zwischen den Blöcken. Bei kleinen Änderungen in Rotationen und Translationen kann der gleiche Gradient zu verschiedenen Blöcken und somit verschiedenen Bereichen beitragen. Dies kann dadurch abgemildert werden, indem ein Gradient seinen Betrag anteilig auf seine räumlich benachbarten Blöcke, sowie auf seine benachbarten Winkel verteilt. Dies wird mittels trilinearer Interpolation erreicht. Jeder Eintrag in einen Bereich wird also mit dem Gewicht $(1 - d)$ multipliziert, wobei d den Abstand des Samples vom Zentrum des Bereichs beschreibt und die Einheiten in den Abständen der Bereiche gemessen sind.

2.2.5 SIFT-Matching

Nachdem nun ein robuster, gegen Beleuchtungsänderungen sowie die meisten Bildtransformationen invarianter, Deskriptor für Feature-Punkte gefunden wurde, bleibt noch die Frage, wie man diese

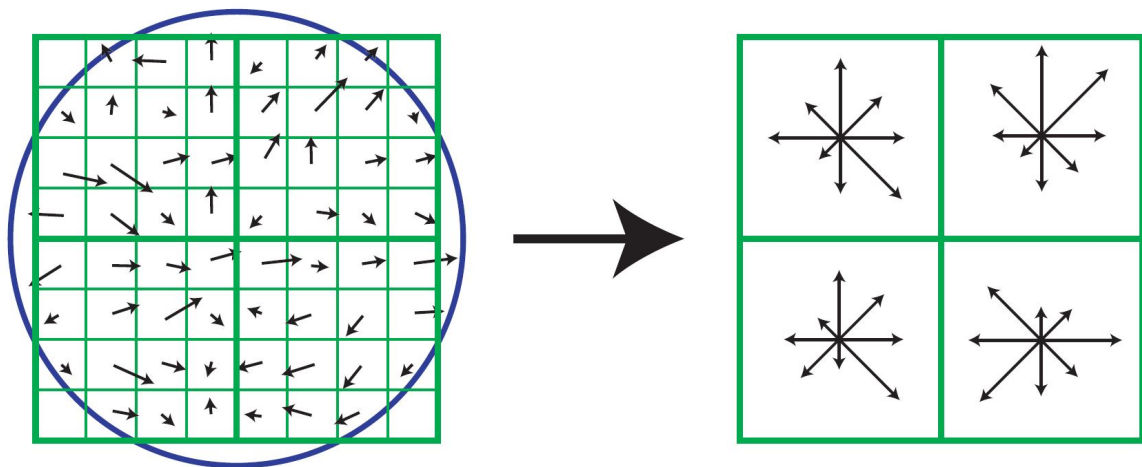


Abbildung 2.6: Links: Unterteilung der Umgebungsgradienten in vier Regionen, Rechts: für jede der vier Regionen ein eigenes Histogramm der Gradienten mit je 8 Richtungsbereichen [Low04].

Key-Points in anderen Bildern wiederfindet und die richtigen Punktkorrespondenzen zuweist. Dieser Vorgang wird auch Matching genannt und im Falle des SIFT-Key-Point-Deskriptors kann man folgendermaßen vorgehen.

Man findet für die beiden Ausgangsbilder separat ihre Key-Points und berechnet deren SIFT-Deskriptoren. Ausgehend vom ersten Bild nimmt man sich alle einzelnen Key-Points nacheinander heraus. Der beste Match zu diesem Key-Point ist sein *Nearest-Neighbor* im zweiten Bild, also der Key-Point, dessen euklidische Distanz zu ihm am kleinsten ist. Allerdings passiert es oft, dass der Korrespondenzpunkt eines Key-Points im zweiten Bild nicht erkannt wurde. In diesem Fall gibt es kein korrektes Match für den Key-Point. Um hier eine falsche Zuweisung zu einem anderen Key-Point zu vermeiden, braucht man einen etwas anderen Ansatz, als global den Nearest-Neighbor zuzuweisen.

Hierfür sucht man sich für den Key-Point das beste und das zweitbeste Match. Ist das Verhältnis der zwei euklidischen Distanzen < 0.8 , so wird das Match als tatsächliche Punktkorrespondenz gewertet. Andernfalls wäre das Match nicht wirklich eindeutig. Da bei falschen Matches meistens mehrere eine ähnliche Distanz ergeben, funktioniert dies in der Praxis relativ gut. Beim obigen Schwellwert von 0.8 werden etwa 90% der falschen Matches eliminiert, während weniger als 5% korrekte Matches ebenfalls ignoriert werden [Low04].

2.2.6 Andere Feature-Deskriptoren

Abschließend kann man sagen, dass SIFT eine sehr gute Möglichkeit darstellt, Features in mehreren Bildern zu entdecken und diese, bei nicht allzu extremen zwischenliegenden Transformationen oder Beleuchtungsänderungen, einander mit hoher Genauigkeit zuzuordnen. Die praktischen Anwendungen reichen von der Objekterkennung [Low99] über die Erstellung von Panoramabildern [BL03]

bishin zur 3D-Rekonstruktion ganzer Szenen aus Videos, bzw. einer Menge von Ausgangsbildern [GL06].

Es gibt allerdings viele weitere Ansätze, die entweder rein zur Feature-Detektion gedacht sind oder auch kombinierte Ansätze, welche wie auch SIFT einen Feature-Deskriptor bieten, welcher zur Findung von Korrespondenzen geeignet ist. Feature-Detektoren können dabei unterschiedliche Features finden, wie z.B. beim Canny-Edge-Detector [Can86], welcher Kanten erkennt. Oftmals werden als Zielfeatures aber Eckpunkte erkannt, so auch beim Harris-Operator [HS88], dem Ansatz von Shi und Tomasi [ST94] oder bei FAST [TPM06].

Sogenannte *Blobs* (deutsch: Kleckse, Tropfen) können beispielsweise mithilfe des Laplacian-of-Gaussians oder der Difference-of-Gaussians, wie zu Beginn dieses Kapitels beschrieben, erkannt werden. Hierauf setzen auch viele Feature-Deskriptoren, wie SURF [BETVG08], GLOH [MS05], ORB [RRKB11] oder die hier vorgestellte SIFT [Low04]. Manche dieser Verfahren bieten leichte Stabilitäts-, Genauigkeits- oder starke Laufzeitverbesserungen gegenüber SIFT.

Weil dies für diese Diplomarbeit aber nicht von entscheidender Bedeutung ist, wird im Folgenden der Ansatz von SIFT den anderen vorgezogen, da er sich bereits lange bewährt hat und keiner der anderen Ansätze eine deutliche Verbesserung der Genauigkeit erreicht.

2.3 Stereogeometrie

Einen Spezialfall für die Punktkorrespondenzfindung stellt eine Situation dar, in der alle Verschiebungen zwischen zwei Bildern nur durch die Eigenbewegung der Kamera verursacht werden. Mit anderen Worten: die Objekte in der Szene haben sich nicht selbst bewegt, sondern werden nur aus einer anderen Position betrachtet. Dies erinnert an den Menschen, der mit seinen beiden Augen die Umgebung aus zueinander verschobenen Positionen sieht und somit Tiefeneindrücke wahrnehmen kann. Auch im Kino macht man sich das zunutze und mit der Hilfe von 3D-Brillen werden den beiden Augen verschiedene Bilder geliefert, welche das menschliche Gehirn in dreidimensionale Information umrechnet. In diesem Abschnitt soll nun gezeigt werden, wie man auch bei einer Maschine aus zwei Bildern Tiefeninformationen extrahieren, Punktkorrespondenzen finden und die Beziehung der beiden Bilder zueinander mit einer Fundamentalmatrix beschreiben kann.

2.3.1 Lochkameramodell

Doch dafür sollte man zunächst verstehen, wie ein Bild überhaupt zustande kommt. Hierbei eignet sich sehr gut der einfache Fall des Lochkamera-Modells. Dort wird angenommen, dass das Licht aus der Umgebung durch ein einziges beliebig kleines Loch, genannt der *Brennpunkt*, in die Kamera einfällt und sich so auf eine Bildebene projiziert, die parallel zur Brennebene im Abstand f hinter dieser liegt (siehe Abb. 2.7). Ein Punkt M aus der Umgebung sendet nun einen Lichtstrahl durch den Brennpunkt, der im Bildpunkt m auf die Bildebene trifft. Hierdurch findet eine Projektion von der dreidimensionalen Realität in die zweidimensionale Bildebene statt, was somit den Informationsverlust einer ganzen Dimension, nämlich der Tiefe, bedeutet. Denn nur durch den Punkt m mit Koordinaten

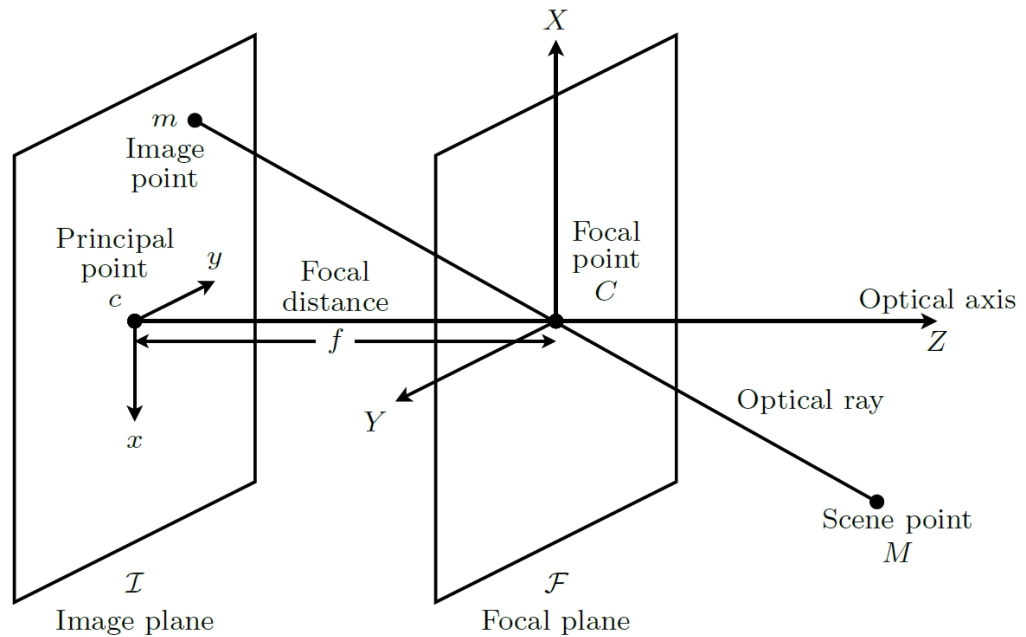


Abbildung 2.7: Das Lochkamera-Modell. Projektion eines Szenenpunktes durch den Brennpunkt C auf die Bildebene I [Bru12a].

(x, y) ist nicht entscheidbar, wie weit entfernt vom Brennpunkt entlang dem Lichtstrahl der Punkt $M(X, Y, Z)$ liegt.

Um mit dem Punkt m alle möglichen Ursprungspunkte M direkt beschreiben zu können, kann man das Konzept der *homogenen Koordinaten* verwenden, die auch mathematisch weitere Vorteile bringen. Der 2D-Punkt (x, y) wird nun als 3D-Punkt (wx, wy, w) angegeben. Das beschreibt gleich alle Punkte des Lichtstrahls von m durch den Brennpunkt und per Division durch w kann jederzeit der 2D-Punkt zurückgewonnen werden. Ebenso kann der Punkt M homogen beschrieben werden und folgende lineare Gleichung beschreibt dann die Projektion von M auf m :

$$\begin{pmatrix} wx \\ wy \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.49)$$

Die Matrix in der Mitte nennt sich dabei die *Projektionsmatrix*. Nun gibt es aber noch mehr Parameter, die in diese Projektion mit einfließen können.

Zum einen gibt es die *extrinsischen* Parameter, welche die Position von der Kamera zur Umgebung beschreiben. Der Punkt M , der in Weltkoordinaten angegeben ist, muss also zunächst in ein kamera-eigenes Koordinatensystem umgerechnet werden. Dies geschieht mittels einer Rotation

$$R = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{pmatrix} \quad (2.50)$$

und einer Translation

$$t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}. \quad (2.51)$$

Dementsprechend ergibt sich die extrinsische Matrix:

$$\begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.52)$$

Zum anderen gibt es noch die *intrinsischen* Parameter, welche von gewissen kamerainternen Geometrien beeinflusst werden, wie der Größe der Pixel in x -Richtung h_u und in y -Richtung h_v , des Winkels θ der Koordinatenachsen der Bildebene zueinander und der Verschiebung des Bildzentrums (*Principal Point* c) in x -Richtung u_0 und in y -Richtung v_0 .

Es ergibt sich diesmal die intrinsische Matrix:

$$\begin{pmatrix} h_u & -h_u \cot \theta & u_0 \\ 0 & h_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.53)$$

Damit erweitert sich die Projektionsgleichung (2.49) zu:

$$\begin{pmatrix} wx \\ wy \\ w \end{pmatrix} = \begin{pmatrix} h_u & -h_u \cot \theta & u_0 \\ 0 & h_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.54)$$

2.3.2 Epipolargeometrie

Nachdem nun gezeigt wurde, wie das Bild einer Lochkamera entsteht, lohnt sich ein Blick auf zwei Kameras, die zum gleichen Zeitpunkt ein Bild der Umgebung aus zwei verschiedenen Positionen machen. Denn damit lässt sich die Tiefeninformation von aufgenommenen Punkten, welche bei der Projektion in die Bildebene verloren ging, wieder bestimmen.

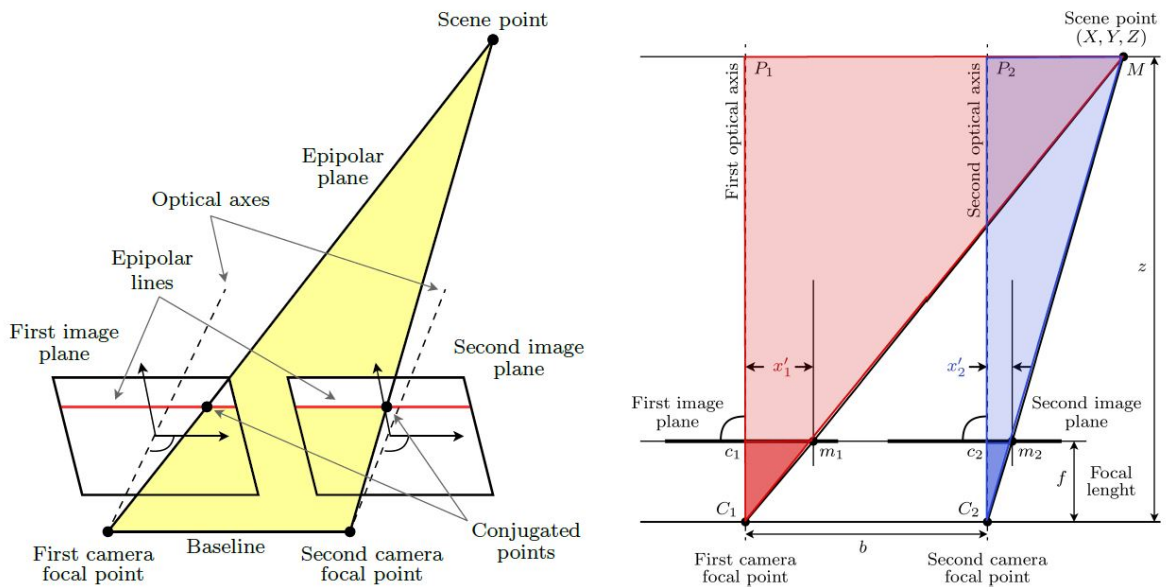


Abbildung 2.8: Links: Aufbau orthoparalleler Kameras und Projektion eines Szenenpunktes in deren Bildebenen, Rechts: vereinfachte Darstellung in 2D [Bru12a]

Orthoparallele Kameras

Ein einfacherer Fall hiervon sind *orthoparallele Kameras*. Wie in Abb. 2.8 zu sehen ist, liegen die beiden Bildebenen der Kameras in der gleichen Ebene. Lediglich eine Verschiebung b der Bildzentren entlang der *Baseline* (Verbindungsline der Principal Points) ist vorhanden.

In Abbildung 2.8 wurden die Bildebenen der Übersichtlichkeit halber vor, statt hinter die Brennpunkte gesetzt. Dies hat den angenehmen Nebeneffekt, dass die Achsen der Bildebenen nun in die gleiche Richtung zeigen, wie die des Weltkoordinatensystems. Beim Punkt (X, Y, Z) , der auf die beiden Punkte (x_1, y_1) und (x_2, y_2) abgebildet wird, ergibt sich in diesem speziellen Aufbau nur ein Unterschied zwischen den x -Koordinaten der Bildpunkte. Im Folgenden sei angenommen, dass der Ursprung des Weltkoordinatensystems im Brennpunkt der ersten Kamera liegt.

Dann liefert der Strahlensatz die Bedingungen

$$\frac{X}{Z} = \frac{x_1}{f} \quad (2.55)$$

und

$$\frac{X - b}{Z} = \frac{x_2}{f} \quad (2.56)$$

und es lässt sich damit die Tiefe Z des Punktes durch folgende einfache Formel wiedergewinnen:

$$Z = \frac{bf}{x_1 - x_2} \quad (2.57)$$

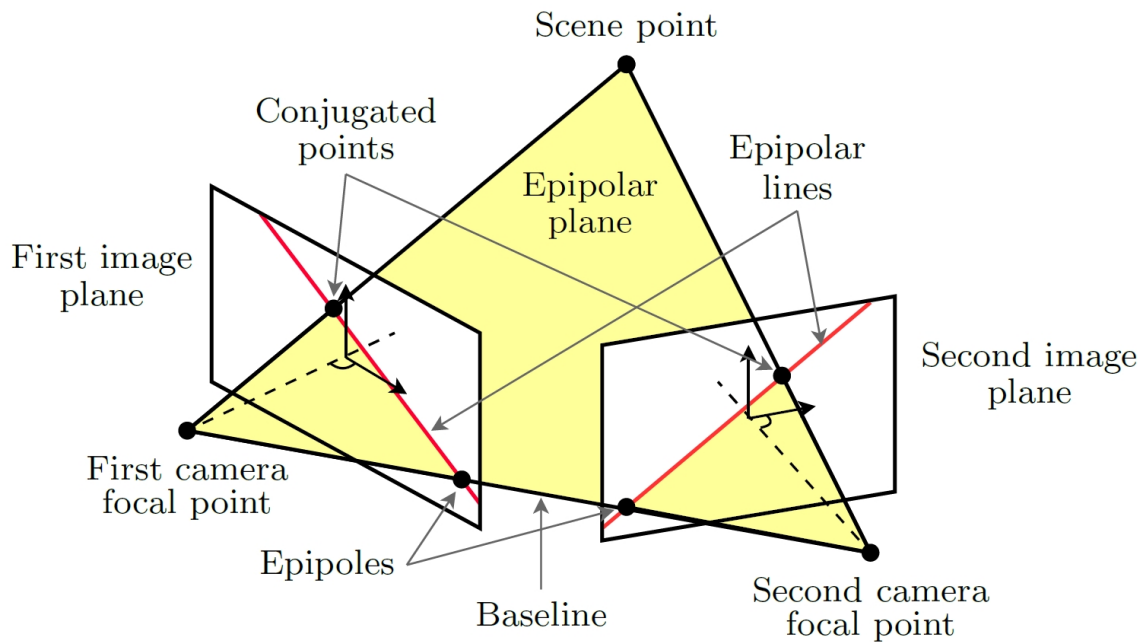


Abbildung 2.9: Epipolargeometrie bei konvergierenden Kameras [Bru12a].

Die Differenz der beiden Bildpunkte im Nenner wird auch *Disparität* genannt. Ist diese groß, so liegt der Punkt also nahe an den Kameras. Ist sie gering, liegt er weit weg.

Konvergierende Kameras

Ein allgemeinerer Aufbau als der orthoparallele sind die *konvergierenden Kameras*. Hierzu betrachte man Abb. 2.9. Die Brennpunkte der beiden Kameras sind wieder mit der Baseline verbunden, jedoch liegen diesmal die Bildebenen nicht mehr komplanar, sondern vielmehr beliebig zueinander gedreht.

Der Szenenpunkt M bildet zusammen mit der Baseline die sogenannte *Epipolarebene*. Auf ihr müssen die beiden Bildpunkte m_1 und m_2 liegen. Desweiteren bilden die Schnittgeraden zwischen der Epipolarebene und den beiden Bildebenen die sogenannten *Epipolarlinien*. Auf ihnen befindet sich jeweils einer der beiden Bildpunkte. Anschaulicher kann man die Epipolarlinien als die Projektion des Sehstrahls der jeweils anderen Kamera durch den Szenenpunkt in die eigene Bildebene beschreiben.

Die beiden korrespondierenden Bildpunkte können daher nicht beliebig in den beiden Bildebenen verteilt sein. Hat man die Position eines der beiden Punkte bestimmt, kann der zweite Punkt aufgrund der Epipolargeometrie nur noch auf der zugehörigen Epipolarlinie der anderen Bildebene liegen. Dieses Verhältnis lässt sich mathematisch festhalten durch die sogenannte *Epipolarbedingung*

$$m_2^T F m_1 = 0. \quad (2.58)$$

Die Matrix F nennt man dabei *Fundamentalmatrix*, sie hat 9 Einträge (3×3), 7 Freiheitsgrade und ist vom Rang 2. Die Punkte m_1 und m_2 sind dabei in homogenen Koordinaten gegeben. Um in der Epipolarbedingung die Epipolarlinien erkennbar zu machen, kann man die zugehörigen Normalen berechnen durch:

$$\begin{aligned}l_2 &= Fm_1, \\l_1 &= F^T m_2.\end{aligned}\tag{2.59}$$

Damit kann man die Epipolarbedingung (2.58) umschreiben in die Geradengleichungen

$$\begin{aligned}m_1^T l_1 &= 0, \\m_2^T l_2 &= 0.\end{aligned}\tag{2.60}$$

Weiterhin kann man die Epipolarbedingung auch als Vektormultiplikation schreiben. Diese Schreibweise wird auch später in dieser Arbeit wieder verwendet werden:

$$m_2^T Fm_1 = s^T f = 0\tag{2.61}$$

mit

$$s = (x_1x_2, y_1x_2, x_2, x_1y_2, y_1y_2, y_2, x_1, y_1, 1)^T\tag{2.62}$$

und

$$f = (f_{1,1}, f_{1,2}, f_{1,3}, f_{2,1}, f_{2,2}, f_{2,3}, f_{3,1}, f_{3,2}, f_{3,3}).\tag{2.63}$$

Diese Schreibweise erlaubt es, die Epipolarbedingung explizit als lineare Bedingung in den Einträgen der Fundamentalmatrix aufzufassen. Auf die Bestimmung der Fundamentalmatrix selbst wird dann in Kapitel 4 eingegangen.

3 Verwandte Arbeiten

Nachdem mit dem optischen Fluss, der Epipolargeometrie und dem Feature-basierten Detektor und Deskriptor SIFT im letzten Kapitel einige Grundlagen behandelt wurden, die wichtig für das weitere Verständnis dieser Arbeit sind, soll sich dieses Kapitel mit verwandten Arbeiten befassen, die für die Bestimmung der Fundamentalmatrix von Bedeutung sind. Eine erste grobe Einteilung kann man vornehmen durch die Art der Korrespondenzen, welche zur Berechnung der Fundamentalmatrix zurate gezogen werden.

Zum einen können dies wenige Feature-Matches sein, die etwa mit dem SIFT- [Low04] oder dem SURF-Deskriptor [BETVG08] beschrieben sind. Anwendungen dazu, wie das Panorama-Stitching [BL03], bei dem zwei oder mehr Bilder aneinandergeheftet werden sollen, haben meist nur einen kleinen Bereich zur Verfügung, in dem sich die Bilder überschneiden. Das Problem liegt also eher darin, die wenigen Korrespondenzen, die überhaupt vorhanden sind, möglichst genau zu finden.

Unterscheiden sich die Bildbereiche aber nur minimal, wie bei vielen Stereobildern oder aufeinanderfolgenden Bildern eines Videos, kann auch ein dichtes Flussfeld zur Berechnung der Fundamentalmatrix verwendet werden, welches z.B. mit dem Verfahren von Brox *et al.* [BBPW04] oder Demetz *et al.* [DSV⁺14] berechnet werden kann.

Die nächste zu stellende Frage lautet, wie viele Korrespondenzen genau zur Berechnung der Fundamentalmatrix verwendet werden. Dies kann entweder mit 7 [HN94] oder 8 und mehr Punktepaaren geschehen. Die Grundidee von letzterem für kalibrierte Kameras bietet Longuet-Higgins [Lon81]. Für den unkalibrierten Fall bietet Zhang [Zha98] eine gute Übersicht mit den besten Methoden. Eine detaillierte Studie des 8-Punkt-Algorithmus ist auch bei Hartley [Har97] zu finden.

Bei der Minimierung des Energiefunktionalen im 8-Punkt-Algorithmus spielt vor allem eine Rolle, was bestraft wird. Die Epipolarbedingung selbst oder eine geometrische Distanz wie der Abstand zur Epipolarlinie? Und wie robust sind die Verfahren im Hinblick auf Ausreißer? Hier unterscheiden sie sich zumeist durch quadratische oder subquadratische Bestrafungsfunktionen. Ein weiterer Ansatz zur robusten Schätzung der Fundamentalmatrix mithilfe des RANSAC-Algorithmus ist bei Zhang [ZDFL95] zu finden.

Valgaerts *et al.* [VBMW12] führten bereits einen Vergleich von dichten zu spärlichen Verfahren zur Bestimmung der Fundamentalmatrix durch. Auch Mainberger [MBW08] untersuchte die Qualität von Fundamentalmatrizen, die durch dichte optische Flussfelder berechnet wurden. Bei beiden Arbeiten konnte mithilfe der dichten Flussfelder ein besseres Ergebnis erzielt werden. Jedoch ist jeweils das Basisverfahren zur Bestimmung des optischen Flusses heutzutage nicht mehr ganz Stand der Kunst.

Ein aussagekräftiges Fehlermaß einer geschätzten Fundamentalmatrix zur Ground-Truth wurde von Faugeras [FLP01] vorgestellt. Dies ermöglicht eine Evaluation der Schätzqualität.

4 Bestimmung der Fundamentalmatrix

Angelehnt an die eben vorgestellte Einteilung der Verfahren zur Bestimmung der Fundamentalmatrix, sollen deren Funktionsweisen im folgenden Kapitel näher erklärt werden. Beginnend mit den grundlegenden Verfahren des 7-Punkt- sowie des 8-Punkt-Algorithmus wird gezeigt, wie viele Punktpaare zur Bestimmung der Fundamentalmatrix wirklich nötig sind und welche Bedingungen dafür herangezogen werden.

Da der 7-Punkt-Algorithmus allerdings keine eindeutige Lösung ausgibt, folgen daraufhin weitere Verbesserungen zum 8-Punkt-Algorithmus. Bei dichten Korrespondenzen kann z.B. durch den Einsatz von M-Schätzern eine höhere Robustheit gegen Ausreißer erreicht werden. Ebenso kann den Termen in der Berechnung eine geometrische Bedeutung verliehen werden, während sich durch Normalisierung der Korrespondenzpunkte die Genauigkeit noch weiter verbessern lässt.

Einen etwas anderen Ansatz verfolgt die Kombination des 8-Punkt-Algorithmus mit RANSAC, wo Robustheit durch die oftmalige zufällige Auswahl der Mindestanzahl an nötigen Punktkorrespondenzen erreicht wird. Hierfür wird dann versucht, die lange Laufzeit des Algorithmus durch das Treffen einer Vorauswahl an Punktkorrespondenzen zu verringern. Schließlich kommt auch SIFT zum Einsatz, um für den RANSAC-Algorithmus eine kleine Menge an Feature-Punktkorrespondenzen bereitzustellen.

4.1 7-Punkt-Algorithmus

Doch beginnen wir zunächst mit dem Algorithmus, der am wenigsten Daten zur Berechnung einer Fundamentalmatrix benötigt. Dies ist der *7-Punkt-Algorithmus* [HN94, Zha98]. Wie der Name schon impliziert, benötigt er 7 bekannte Punktkorrespondenzen, welche je eine lineare Gleichung durch die Epipolarbedingung liefern. Als zusätzliche Bedingung setzt man $\det(F) = 0$, da der Rang von F ja nur 2 ist. Diese 8 Bedingungen reichen schon aus, um die 9 Einträge von F bis auf einen Faktor genau zu bestimmen.

Wie bereits in Gleichung (2.61) gezeigt wurde, lässt sich die Epipolarbedingung auch als Vektorprodukt $s^T f = 0$ schreiben. Mit dieser Schreibweise kann man aus den 7 Punktkorrespondenzen das Gleichungssystem $Sf = 0$ aufstellen. Als Ergebnis erhält man zwei linear unabhängige Lösungsvektoren f_1 und f_2 . Man kann die Fundamentalmatrix F dann als Linearkombination der beiden Lösungen darstellen:

$$F = \alpha \cdot F_1 + (1 - \alpha) \cdot F_2, \quad \alpha \in \mathbb{R}. \quad (4.1)$$

Zusammen mit der Bedingung, dass die Determinante 0 ist, ergibt sich die Gleichung

$$\det(F) = \det(\alpha \cdot F_1 + (1 - \alpha) \cdot F_2) = 0. \quad (4.2)$$

Dies stellt eine kubische Gleichung dar, die somit mindestens ein, aber bis zu drei verschiedene α als Lösung ermöglicht. Dies entspricht ebenso vielen Fundamentalmatrizen.

Da diese Arbeit allerdings an einer eindeutigen Lösung zur Fundamentalmatrix interessiert ist, wurde dieser Algorithmus hier nicht verwendet. Stattdessen kommt ein Algorithmus zum Einsatz, der mehr als 7 Punktkorrespondenzen zur Berechnung der Fundamentalmatrix verwendet und die gewünschte eindeutige Lösung liefert. Dieser wird im nächsten Abschnitt beschrieben.

4.2 8-Punkt-Algorithmus

Die andere Möglichkeit zur Berechnung der Fundamentalmatrix ist der *8-Punkt-Algorithmus*. Dieser arbeitet mit mindestens 8, aber sonst beliebig vielen Punktkorrespondenzen. Dadurch ist außer den Epipolarbedingungen keine zusätzliche Bedingung mehr nötig, wie etwa beim 7-Punkt-Algorithmus, allerdings kann auch Rang 2 nicht direkt sichergestellt werden.

Prinzipiell würde wieder das Gleichungssystem $Sf = 0$ betrachtet werden, doch aufgrund der vielen Punktpaare und Messungenauigkeiten würde hier der Lösungsvektor f nicht bestimmbar sein. Stattdessen versucht man nun, $|Sf|$ zu minimieren. Dies geschieht mithilfe der Methode der kleinsten Quadrate nach Longuet-Higgins [Lon81]:

$$E(f) = \sum_{i=1}^N (s_i^T f)^2 = f^T \sum_{i=1}^N (s_i s_i^T) f. \quad (4.3)$$

Dabei muss gelten $|f| = 1$, um die triviale Lösung des Nullvektors zu vermeiden. Diese Bedingung lässt sich als Lagrange-Multiplikator $C(f)$ folgendermaßen in das Energiefunktional mit einbinden:

$$\begin{aligned} E^*(f, \lambda) &= E(f) + \lambda C(f) & (4.4) \\ &= \sum_{i=1}^N (s_i^T f)^2 + \lambda(1 - f^T f) \\ &= f^T \sum_{i=1}^N (s_i s_i^T) f + \lambda(1 - f^T f) \\ &= f^T S^T S f + \lambda(1 - f^T f). \end{aligned}$$

Um das Energiefunktional $E(f)$ zu minimieren, muss nun der Sattelpunkt von $E^*(f)$ gefunden werden. Dafür muss $E^*(f)$ einerseits nach f abgeleitet und $= 0$ gesetzt werden

$$(S^T S - \lambda I) f = 0 \quad (4.5)$$

und die Ableitung nach λ

$$1 - f^T f = 0 \quad (4.6)$$

stellt dabei sicher, dass $|f| = 1$.

Dies stellt ein Eigenwertproblem der Matrix

$$S^T S = \sum_{i=1}^N (s_i s_i^T) \quad (4.7)$$

dar. Sie hat 9×9 Einträge und der Eigenvektor zum kleinsten Eigenwert liefert den Vektor, der das Minimum für die Energie der aufsummierten Epipolarbedingungen erreicht und somit unsere gesuchte Fundamentalmatrix darstellt, welche nur noch von der Vektorform in eine Matrix umzuschreiben ist.

Bemerkenswert ist hier, dass es sich um ein lineares Problem handelt, dessen Lösung aus diesem Grund mit relativ geringem Aufwand zu berechnen ist. Mit diesem Verfahren hat man ebenfalls die Freiheit, beliebig viele Punktkorrespondenzen (solange es mindestens 8 linear unabhängige sind) zur Berechnung der Fundamentalmatrix zu verwenden. Es kann daher je nach Auswahl der Punktepaare sinnvoll sein, die minimale Anzahl zu verwenden oder sogar alle Korrespondenzen, die einem zur Verfügung stehen.

4.3 Dichte Korrespondenzen

Im folgenden Abschnitt soll zunächst untersucht werden, wie man viele, bzw. alle zur Verfügung stehenden Korrespondenzen verwenden kann, um auf Basis des 8-Punkt-Algorithmus die Fundamentalmatrix zu schätzen. Dabei bieten sich einige Verbesserungen für den Algorithmus an.

4.3.1 M-Schätzer

Beispielsweise stellen fehlerbehaftete Punktkorrespondenzen ein Problem in der Berechnung des 8-Punkt-Algorithmus dar. Diese Ausreißer, bzw. deren Epipolarbedingungen, fließen auch hier quadratisch in das Energiefunktional ein. Das kann einen starken negativen Einfluss auf die Genauigkeit der berechneten Fundamentalmatrix haben. Um dem entgegen zu wirken, ist die Idee von M-Schätzern, die Funktion im Energiefunktional durch eine andere zu ersetzen, welche weniger empfindlich auf Ausreißer reagiert. Dies kann man, wie bereits in Kapitel 2.1.2 (bei den robusten Datentermen in der Berechnung des optischen Flusses) beschrieben, durch eine subquadratische Funktion erreichen, die dem quadratischen Einfluss der Epipolarbedingung entgegenwirkt. Sie kann beispielsweise wieder durch die Funktion

$$\Psi(s^2) = \sqrt{s^2 + \epsilon^2} \quad (4.8)$$

realisiert werden. Die neue Energie ergibt sich somit durch:

$$E^*(f, \lambda) = \sum_{i=1}^N \Psi \left((s_i^T f)^2 \right) + \lambda(1 - f^T f) \quad (4.9)$$

Dieses Energiefunktional stellt nun jedoch kein lineares Problem mehr dar, weshalb ein neuer Ansatz zu dessen Lösung gefunden werden muss. Deshalb versucht man, dieses nicht-lineare Problem durch eine sogenannte *Lagged-Nonlinearity-Strategie* in ein iteratives, lineares Problem umzuwandeln.

Hierfür wiederholt man entsprechend dem normalen 8-Punkt-Algorithmus den Schritt zur Lösung des Eigenwertproblems für die Matrix und erhält diesmal ein Eigenwertproblem der Form

$$S^T W S = \sum_{i=1}^N \Psi' \left((s_i^T f_{old})^2 \right) s_i s_i^T \quad (4.10)$$

mit

$$\Psi'(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon^2}} \quad (4.11)$$

und der Diagonalmatrix W , in deren diagonalen Einträgen die Gewichte Ψ' der einzelnen Korrespondenzen stehen. Dabei ist f_{old} die Lösung der vorangegangenen Iteration. Für jedes Punktepaar wird sozusagen mithilfe der zuvor berechneten Fundamentalmatrix die Epipolarbedingung berechnet, mit der Funktion Ψ' gewichtet und schließlich zur Matrix hinzuaddiert. So verbessert sich im Idealfall die Fundamentalmatrix mit jeder Iteration, bis sie gegen die Lösung konvergiert. Diese Vorgehensweise wird auch *Iteratively Reweighted Least Squares* genannt. Für die erste Iteration ist jedoch noch kein f_{old} bekannt, weshalb das Gewicht zum Anfang auf 1 gesetzt wird, was einer Funktion

$$\Psi(s^2) = s^2 \quad (4.12)$$

entspricht. Nachdem die Fundamentalmatrix zum ersten Mal berechnet ist, wird das Verfahren dann wie zuvor erläutert fortgeführt.

Somit kann festgehalten werden, dass durch den Einsatz von M-Schätzern der Einfluss von Ausreißern eingedämmt werden kann, wofür aufgrund der Nichtlinearität allerdings eine iterative Lösung vonnöten ist, welche ein Vielfaches an Berechnungen durchführen muss.

4.3.2 Geometrische Ansätze

Bis zu diesem Punkt wurde im Energiefunktional des 8-Punkt-Algorithmus immer die Epipolarbedingung verwendet und aufsummiert, um für jede Punktkorrespondenz Abweichungen davon zu bestrafen. Das ist auch von Vorteil, da sie linear ist und die Lösung des Energiefunktionals somit gut zu berechnen ist.

Doch was sagt dieser Bestrafungswert für ein Punktepaar überhaupt aus? Kann man diesen Wert als eine Distanz des zweiten Punktes zu der Epipolarlinie im zweiten Bild, auf der er liegen sollte, sehen? Um das herauszufinden, betrachte man die Epipolarbedingung des Punktepaares m_1 und m_2 mit dem Index i :

$$d_L(m_2^i, Fm_1^i) = m_2^{iT} F m_1^i = m_2^{iT} l_2^i. \quad (4.13)$$

d_L steht hier für die (lineare) Distanz des Punktes m_2^i zur zugehörigen Epipolarlinie Fm_1^i . Liegt der Punkt genau auf der Epipolarlinie, so ergibt sich die Distanz d_L zu 0. Bei Punkten jedoch, die nicht auf ihr liegen, ergibt sich genau dann der Abstand vom Punkt zur Geraden, wenn die ersten beiden Einträge des Geradenvektors normalisiert sind, also die Hessesche Normalform der Gerade beschreiben.

Weil die ersten beiden Einträge von l_2^i in unserem Fall aber beliebig groß sein können, ist diese geometrische Interpretation so noch nicht unterstützt. Vielmehr muss die Distanz noch mithilfe der ersten beiden Einträge des Geradenvektors normalisiert werden, um folgende nicht-lineare Distanz zu erhalten, welche nun aber tatsächlich für den geometrischen Abstand vom Bildpunkt m_2^i zur Epipolarlinie steht [Bru12b]:

$$d_{NL}(m_2^i, Fm_1^i) = \frac{1}{\sqrt{(Fm_1^i)_1^2 + (Fm_1^i)_2^2}} m_2^{iT} F m_1^i = \frac{m_2^{iT} l_2^i}{\sqrt{(l_2^i)_1^2 + (l_2^i)_2^2}}. \quad (4.14)$$

Folgende Gleichung verhilft dabei zum besseren Verständnis der verwendeten Indizes:

$$l_2^i = \begin{pmatrix} (l_2^i)_1 \\ (l_2^i)_2 \\ (l_2^i)_3 \end{pmatrix} = \begin{pmatrix} (Fm_1^i)_1 \\ (Fm_1^i)_2 \\ (Fm_1^i)_3 \end{pmatrix} = Fm_1^i. \quad (4.15)$$

Da die geometrische Distanz leider wieder nicht linear ist, muss auch hier wieder ein iteratives Verfahren der lagged nonlinearity zur Minimierung des Energiefunktional angewendet werden.

Dies funktioniert entsprechend zum vorhergehenden Abschnitt über M-Schätzer. Die Funktion Ψ' wird einfach so angepasst, dass das resultierende Gewicht im Energiefunktional der Normalisierung des Epipolarlinienvektors für die nicht-lineare Distanz entspricht. Auch hier muss natürlich für die erste Iteration das Gewicht auf 1 gesetzt werden, damit eine erste Fundamentalmatrix für die weiteren Iterationen berechnet werden kann.

Varianten des geometrischen Ansatzes

Zur Verwendung von geometrischen Distanzwerten können nun verschiedene Varianten verwendet werden. Davon sollen an dieser Stelle drei gezeigt werden [Bru12b]:

Der einfache Ansatz minimiert die gerade vorgestellte nicht-lineare Distanz der Punkte aus dem zweiten Bild zu ihren Epipolarlinien und ergibt das folgende Energiefunktional:

$$\begin{aligned} E(F) &= \sum_{i=1}^N d_{NL}^2(m_2^i, Fm_1^i) \\ &= \sum_{i=1}^N \frac{1}{(Fm_1^i)_1^2 + (Fm_1^i)_2^2} (m_2^{iT} F m_1^i)^2. \end{aligned} \quad (4.16)$$

4 Bestimmung der Fundamentalmatrix

Die zweite Variante ist der symmetrische Ansatz, welcher gleichzeitig die Distanz sowohl der Punkte im ersten Bild, als auch der Punkte im zweiten Bild zu ihren Epipolarlinien minimiert. Das dazugehörige Energiefunktional lautet:

$$\begin{aligned} E(F) &= \sum_{i=1}^N \left(d_{NL}^2(m_2^i, F m_1^i) + d_{NL}^2(m_1^i, F^T m_2^i) \right) \\ &= \sum_{i=1}^N \left(\frac{1}{(F m_1^i)_1^2 + (F m_1^i)_2^2} + \frac{1}{(F^T m_2^i)_1^2 + (F^T m_2^i)_2^2} \right) (m_2^{iT} F m_1^i)^2. \end{aligned} \quad (4.17)$$

Zuletzt sei hier noch der Gradientenansatz gezeigt, bei dem die Fundamentalmatrix als Fläche im 4D-Raum gesehen wird, welche durch die Punkte $(x_1^i, y_1^i, x_2^i, y_2^i)$ gelegt wird. Das Energiefunktional dazu lautet:

$$E(F) = \sum_{i=1}^N \left(\frac{1}{(F m_1^i)_1^2 + (F m_1^i)_2^2 + (F^T m_2^i)_1^2 + (F^T m_2^i)_2^2} \right) (m_2^{iT} F m_1^i)^2. \quad (4.18)$$

Welchen geometrischen Ansatz man auch wählen mag, es macht auf jeden Fall Sinn, die allgemeine Epipolarbedingung mit einer geometrischen Bedeutung zu versehen, da man sich sonst nicht im Klaren ist, welche Bedeutung die aufsummierte Energie denn wirklich hat. Auf diese Art der Gewichtung weiß man, dass Punkte, welche weiter entfernt von ihrer Epipolarlinie liegen, auch weniger stark in die Berechnung der Fundamentalmatrix mit einbezogen werden, während man ohne diese Gewichtung nicht sicher sein kann, wie weit die Punkte wirklich entfernt sind. In dieser Arbeit wird im Folgenden jedoch nur der einfache geometrische Ansatz verwendet.

4.3.3 Normalisierung und Rang-2-Bedingung

Bei der Verwendung von vielen Punktkorrespondenzen gibt es noch eine weitere Möglichkeit, die Schätzung der Fundamentalmatrix zu verbessern. Hierfür vergleiche man die Größe der Koordinateneinträge der Korrespondenzpunkte m_1 und m_2 . Für Bilder der Größe 1024×1024 existieren beispielsweise Punkte mit Koordinaten $(0, 0, 1)^T$, aber auch $(1023, 1023, 1)^T$ (abhängig von der Indexierung der Pixel). Diese großen Unterschiede der Einträge einerseits zwischen verschiedenen Punkten, aber auch zwischen erster, zweiter und dritter Koordinate desselben Punktes bedingen verschieden starke Einflüsse auf die Schätzung der Fundamentalmatrix.

Um diesen Effekt zu vermindern, kann man die Punkte des ersten und zweiten Bildes als separate Punktwolken M_1 und M_2 auffassen und über diesen eine *Normalisierung* durchführen [Har97]. Dies funktioniert so, dass die Punkte mit je einer Transformationsmatrix T_1 für die Punktwolke M_1 und einer Matrix T_2 für die Punkte M_2 multipliziert werden.

$$\bar{m}_j^i = T_j m_j^i, \quad (4.19)$$

mit $i = 1, \dots, N$ und $j = 1, 2$. Dabei sind die Transformationsmatrizen:

$$T_j = \begin{pmatrix} s_j & 0 & -s_j \bar{x}_j \\ 0 & s_j & -s_j \bar{y}_j \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.20)$$

Die Parameter in den Matrizen stellen zum einen eine Skalierung s_j dar und zum anderen die Verschiebungen der Punkte in x - und y -Richtung \bar{x}_j sowie \bar{y}_j .

Diese Parameter werden nun so gewählt, dass die normalisierten Punktwolken \bar{M}_1 und \bar{M}_2 den Mittelwert 0 haben und die durchschnittliche Größe der Vektoren $\bar{m}_j^i \sqrt{3}$ ist.

Dies wird durch die Auswertung der folgenden Formeln erreicht:

$$\bar{x}_j = \frac{1}{N} \sum_{i=1}^N x_j^i, \quad (4.21)$$

$$\bar{y}_j = \frac{1}{N} \sum_{i=1}^N y_j^i, \quad (4.22)$$

$$s_j = \sqrt{3} \cdot \frac{1}{\frac{1}{N} \sum_{i=1}^N \sqrt{(x_j^i - \bar{x}_j)^2 + (y_j^i - \bar{y}_j)^2 + 1^2}}. \quad (4.23)$$

Das elegante an dieser Normalisierung ist der Umstand, dass die Berechnung der Fundamentalmatrix aus den normalisierten Punkten komplett unverändert funktioniert:

$$m_2^{iT} F m_1^i = \bar{m}_2^{iT} T_2^{-T} F T_1^{-1} \bar{m}_1^i = \bar{m}_2^{iT} \bar{F} \bar{m}_1^i \quad (4.24)$$

Es muss nur noch nach der Berechnung der normalisierten Fundamentalmatrix diese wieder zurück transformiert werden, was einfach mithilfe der gleichen Transformationsmatrizen T_1 und T_2 wie folgt funktioniert:

$$F = T_2^T \bar{F} T_1 \quad (4.25)$$

Mithilfe einer solchen Normalisierung können teilweise sehr deutliche Verbesserungen bei der Schätzung der Fundamentalmatrix erreicht werden. Da die Transformation und Rücktransformation die Laufzeit auch in der Komplexität nicht erhöhen, ist es bei der Verwendung von vielen Punktkorrespondenzen immer ratsam, diese Methode zu verwenden.

Sicherstellung von Rang 2

Im Gegensatz zum 7-Punkt-Algorithmus, bei dem die Zusatzbedingung, dass der Rang der Fundamentalmatrix gleich 2 sein muss, eben diesen Rang garantiert, verwendet der 8-Punkt-Algorithmus nur Epipolarbedingungen zur deren Berechnung. Daher ist die geschätzte Fundamentalmatrix im Normalfall nicht vom Rang 2.

Möchte man diesen aber nachträglich noch erzwingen, so kann man für F eine Singulärwertzerlegung durchführen. Diese sieht folgendermaßen aus:

$$F = UDV^T = U \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} V^T \quad (4.26)$$

U und V sind dabei Matrizen mit den linken und rechten Eigenvektoren von F . Um den Rang auf 2 zu ändern, muss man jetzt einfach den kleinsten Singulärwert σ_3 auf 0 setzen und F nach Gleichung (4.26) wieder ausmultiplizieren. Dadurch erhält man die Fundamentalmatrix vom Rang 2, die F im Sinne der Frobeniusnorm am nächsten ist.

Das bedeutet aber nicht automatisch, dass sich dadurch auch die Qualität der geschätzten Fundamentalmatrix verbessert. Die Änderungen, die sich ergeben, sind auch meist eher gering. Deshalb ist eine Normalisierung der Punkte deutlich stärker zu empfehlen als die Singulärwertzerlegung mit Sicherstellung vom Rang 2 der Fundamentalmatrix.

4.4 RANSAC

Nachdem mit den letzten vorgestellten Verfahren ausführlich gezeigt wurde, wie die Schätzung der Fundamentalmatrix mit dem 8-Punkt-Algorithmus unter der Verwendung von vielen Punktkorrespondenzen durchgeführt und mit einigen Modifikationen noch verbessert werden kann, soll sich dieser Abschnitt damit beschäftigen, zwar auch den 8-Punkt-Algorithmus zu verwenden, aber hierfür nur die gerade notwendigen 8 Punktepaare zu benutzen. Im folgenden Teil steht jetzt nicht mehr die eigentliche Berechnung der Fundamentalmatrix im Vordergrund, sondern vielmehr die richtige Auswahl der 8 Korrespondenzen, aus denen F berechnet wird.

Hinter der Abkürzung *RANSAC* verbirgt sich der Name *Random Sampling Consensus* [FB81]. Allgemein geht es bei diesem Verfahren darum, aus einer Datenmenge per zufälliger Stichprobenauswahl ein Modell zu schätzen, beispielsweise eine geometrische Kurve oder, wie für diese Arbeit interessanter, eine Fundamentalmatrix. Dabei geht man davon aus, dass unter den Datenpunkten viele Ausreißer mit teils großem Fehler vorhanden sind.

4.4.1 Nachteil traditioneller Verfahren

Wendet man auf eine solche Datenmenge mit vielen und großen Fehlern etwa ein traditionelles Verfahren wie die Methode der kleinsten Quadrate an, so ergibt sich ein Problem. Selbst wenn man davon ausgeht, dass wenige Fehlerpunkte vorhanden sind, so kann doch ein einziger Ausreißer das Ergebnis stark verfälschen. Hierzu betrachte man Abb. 4.1, bei der eine Gerade aus einer Menge von Datenpunkten gesucht wird. Trotz dass viele Punkte sehr gut zur Gerade passen, verläuft die Ausgleichsgerade aller Punkte aufgrund der starken Wirkung des Ausreißers nicht wie gewünscht durch die meisten, genaueren Punkte. Geht man nun von einer Datenmenge aus, die bekanntermaßen viele Ausreißer enthält, so ist die Wahl einer robusteren Methode sinnvoll. Während die Verwendung von M -Glättern im vorigen Abschnitt darauf beruht, Ausreißer mit einer Robustifizierungsfunktion herunterzugewichten, wird im RANSAC-Algorithmus dagegen versucht, erst gar keine Ausreißer in die Berechnung mit einfließen zu lassen.

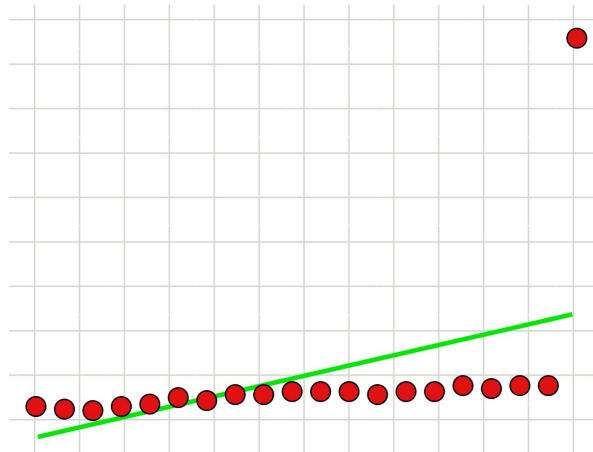


Abbildung 4.1: Ein fehlerbehafteter Wert aus einer Menge von Datenpunkten verursacht eine deutliche Veränderung der Ausgleichsgeraden durch diese Punkte [Wik]

4.4.2 Algorithmus

Unter der Voraussetzung, dass die Anzahl der Datenpunkte ausreichend ist, um die Mindestanzahl der zur Berechnung des Modells nötigen Punkte zu übersteigen, funktioniert der RANSAC-Algorithmus auf folgende Art und Weise:

1. Wähle genau so viele Punkte zufällig aus der Menge der Datenpunkte aus, wie mindestens zur Berechnung des Modells benötigt werden.
2. Berechne mittels dieser Punkte das Modell unter der Annahme, dass die gewählten Punkte keinen Ausreißer enthalten.
3. Bestimme das *Consensus-Set*. Dieses stellt die Menge aller Punkte dar, deren Abstand zur Kurve des berechneten Modells kleiner als ein gewählter Schwellwert ist. Ab einer gewissen Größe des Consensus-Sets, speichere dieses.
4. Wiederhole die Schritte 1-3 so oft wie gewünscht.

Es ist ersichtlich, dass hier nicht alle Datenpunkte zur Berechnung des Modells hergenommen werden und mit größerer Anzahl an Iterationen steigt die Wahrscheinlichkeit, unter den zufällig ausgewählten Punkten einmal eine Menge ohne Ausreißer gezogen zu haben. Für diese Menge nimmt man dann an, dass sie ein größeres Consensus-Set erreicht, als andere Mengen, die Ausreißer enthalten. Dies gewährleistet die Berechnung des Modells, das die größte Unterstützung aus den Datenpunkten erhält.

Am Beispiel der Geraden (siehe Abb. 4.1) würde man also z.B. 100 mal zwei Punkte aus den vorhandenen Datenpunkten ziehen. Hierbei ist die Anzahl 100 willkürlich gewählt, aber zwei Punkte ist genau die Mindestzahl, die notwendig ist, um eine Gerade zu bestimmen. Diese würde man dann auch berechnen und für jedes Punktepaar das Consensus-Set bestimmen. Dies passiert, indem für jeden Datenpunkt sein geometrischer Abstand zur Geraden ausgerechnet und dieser mit einem wiederum

selbst bestimmten Schwellwert verglichen wird. Das Punktepaar, das also das größte Consensus-Set erreicht, bestimmt die errechnete Gerade.

4.4.3 Erfolgskriterien

Für eine erfolgreiche Anwendung des RANSAC-Algorithmus sind vor allem zwei Dinge von großer Bedeutung. Einerseits die Anzahl der Ziehungen von Punktmengen aus den vorhandenen Daten und andererseits die Wahl des Schwellwerts, der mit dem Abstand der Datenpunkte zur Modellkurve verglichen wird.

Anzahl Samples

Ersteres ist von entscheidender Bedeutung, da die Anzahl der gezogenen Samples die Wahrscheinlichkeit bestimmt, mit der man in seinen Ziehungen mindestens eine sinnvolle Punktmenge dabei hat, die ausreißerfrei ist. Dieser Zusammenhang lässt sich auch mathematisch beschreiben. Die entscheidenden Variablen sind im Folgenden die Anzahl der Ziehungen m , die Anzahl der gezogenen Datenpunkte q , der geschätzte Anteil an Ausreißern zum Modell $\mu \in [0, 1]$ und die zu berechnende Wahrscheinlichkeit p .

Ausgehend von der Wahrscheinlichkeit, bei der zufälligen Auswahl von q Punkten keinen Ausreißer zu erwischen $(1 - \mu)^q$ berechnet sich die Wahrscheinlichkeit für mindestens eine gute Punktmenge mit:

$$p = 1 - [1 - (1 - \mu)^q]^m \quad (4.27)$$

Löst man die Formel (4.27) nach m auf, so ergibt sich die Formel zur Berechnung der Anzahl nötiger Ziehungen, um mit Wahrscheinlichkeit p ein gutes Sample zu bekommen:

$$m = \frac{\log(1 - p)}{\log[1 - (1 - \mu)^q]} \quad (4.28)$$

Eine kleine Auswahl an berechneten Werten für m zeigt Tabelle 4.1. Hier wird von 8 nötigen Datenpunkten zur Bestimmung des Modells (beispielsweise der Fundamentalmatrix) ausgegangen. Allerdings ist zu beachten, dass bei einem Ausreißeranteil größer als 0.5, je nach Verteilung der Ausreißerwerte, die Inlier von diesen überstimmt werden können und das größte Consensus-Set eine fehlerbehaftete Ziehung von Datenpunkten unterstützen kann. In diesem Fall wäre der Einsatz des RANSAC-Algorithmus nicht sinnvoll, da er dann fehlerhafte Ergebnisse liefert.

Maximaler Abstand der Datenpunkte

Hat man mit genügend großer Wahrscheinlichkeit eine passende Punktauswahl gefunden, muss hierzu auch zuverlässig ein passendes Consensus-Set berechnet werden. Der entscheidende Wert dabei ist der maximale Abstand der Datenpunkte zur Kurve, den die Punkte haben dürfen. Ist dieser zu groß gewählt, können auch Ziehungen mit Ausreißern ein großes Consensus-Set erreichen, wodurch

μ	$m(p = 0,99)$	$m(p = 0,95)$
0,1	9	6
0,2	26	17
0,3	78	51
0,4	272	177
0,5	1.177	766
0,6	7.025	4.570
0,7	70.188	45.659
0,8	1.798.894	1.170.207
0,9	460.516.224	299.572.640
0,99	41.479.694.633.140.224	26.983.155.659.440.128

Tabelle 4.1: Anzahl der nötigen Ziehungen m bei Ausreißeranteil μ , Erfolgswahrscheinlichkeit p und einer Auswahl von $q = 8$ Punkten

ein falsches Modell bestätigt werden könnte. Ist der Wert zu klein gewählt, kann es sein, dass für Ziehungen ohne Ausreißer das Consensus-Set zu klein wird, um überhaupt gespeichert zu werden.

Daher muss die Wahl dieses Schwellwertes innerhalb des richtigen Wertebereichs erfolgen. Leider ist dieser nicht allgemein mathematisch berechenbar, sondern muss meist empirisch gefunden werden.

Adaptive Bestimmung der Parameter

Bei der Wahl der Anzahl von Ziehungen ergibt sich meist das Problem, dass der Ausreißeranteil μ unbekannt ist. Möchte man diesen nicht selbst schätzen müssen, kann man auch auf folgende Art und Weise vorgehen:

Man initialisiere den Wert μ mit einer Worst-Case Annahme. Dieser sei z.B. 0,7. Mit einer dementsprechenden Iterationszahl starte man den RANSAC-Algorithmus. Nun berechnet man ganz normal für jede Ziehung die Größe des zugehörigen Consensus-Sets. Stellt dieses einen größeren Anteil dar, als der Worst-Case annehmen ließ (in diesem Beispiel $> 0,3$, so bestimme man die Anzahl der Iterationen aufgrund des neuen maximalen Ausreißeranteils ($1 - \text{Anteil Consensus-Set}$) neu.

4.4.4 Anwendung auf Fundamentalmatrix

In dieser Arbeit soll der RANSAC-Algorithmus aber nicht für die Bestimmung einer geometrischen Kurve eingesetzt werden, wie sie in obiger Erklärung als Beispiel verwendet wird, sondern das zu bestimmende Modell ist die Fundamentalmatrix.

Anstatt von Punkten besteht das Ausgangsdatenmaterial hier aus Punktkorrespondenzen, also Paaren von Punkten. Die für die Bestimmung des Modells notwendige Mindestanzahl an Daten ist daher 8, und die Berechnung des Modells wird mit dem normalen 8-Punkt-Algorithmus durchgeführt. Für jede

Ziehung von 8 Korrespondenzen muss natürlich auch das Consensus-Set bestimmt werden. Hierfür wird zu jeder Punktkorrespondenz aus den Ausgangsdaten vorzugsweise der Betrag $|d_{NL}|$ oder das Quadrat d_{NL}^2 der *geometrischen Distanz* (aus Abschnitt 4.3.2) des Punktes im zweiten Bild zu seiner Epipolarlinie

$$d_{NL}(m_2^i, Fm_1^i) = \frac{1}{\sqrt{(Fm_1^i)_1^2 + (Fm_1^i)_2^2}} m_2^{iT} F m_1^i = \frac{m_2^{iT} l_2^i}{\sqrt{(l_2^i)_1^2 + (l_2^i)_2^2}}. \quad (4.29)$$

berechnet und mit einem festgelegten Schwellwert verglichen. Die Fundamentalmatrix, für die die meisten Punkte näher als der Schwellwert zu ihrer Epipolarlinie liegen, wird schließlich als Ergebnis verwendet.

Wie zuvor beschrieben, stellt natürlich nicht nur die gute Wahl des Schwellwerts für die Distanz, sondern auch die Anzahl der Iterationen einen entscheidenden Faktor für den Erfolg des Algorithmus dar. Für die Wahl dieses Wertes passend zu einer gewünschten Erfolgswahrscheinlichkeit kann bei der Bestimmung einer Fundamentalmatrix Tabelle 4.1 für die Mindestzahl an Ziehungen herangezogen werden.

4.5 Variationen mit RANSAC

Wie gerade ausgeführt, stellt der RANSAC-Algorithmus neben beispielsweise der Robustifizierung des 8-Punkt-Algorithmus durch M-Schätzer eine weitere Möglichkeit dar, aus vielen Punktkorrespondenzen auf robuste Art und Weise eine Fundamentalmatrix zu ermitteln. Auch für dieses Verfahren sollen nun ein paar Variationen vorgestellt werden, mit welchen versucht wird, eine Verbesserung zu erreichen.

4.5.1 Least Median of Squares

Ein großes Problem des RANSAC-Algorithmus liegt darin, dass es keinen allgemein gültigen Wert für den Schwellwert der Distanz zum Modell gibt. Er lässt sich auch nicht einfach automatisch berechnen, sodass uns keine andere Möglichkeit bleibt, als durch Ausprobieren diesen Wert empirisch zu bestimmen.

Es gibt allerdings eine Alternative zur Bestimmung des Consensus-Sets mit dem Schwellwert. Diese ist der *Least Median of Squares*. Dabei wird weiterhin für jede Punktkorrespondenz ihre Distanz zur Fundamentalmatrix berechnet. Die Distanzen werden aber nicht mit einem Schwellwert verglichen, sondern sie werden alle in einem Vektor gespeichert. Dieser Vektor wird nun nach der betraglichen Größe seiner Elemente sortiert. Hierfür eignet sich z.B. der *Quicksort-Algorithmus* oder ein anderer Sortieralgorithmus mit gleicher Laufzeitkomplexität. Der Wert, der uns nun interessiert, ist der Median, also der Wert, der genau in der Mitte des Vektors steht.

Für jede Ziehung aus den Punktkorrespondenzen wird also nicht die Größe des Consensus-Sets verglichen, sondern dieser berechnete Median. Der Median mit dem kleinsten Wert setzt sich durch und die zugehörige Fundamentalmatrix wird als Lösung verwendet.

Dieses Verfahren basiert auf der Annahme, dass bei kleinerem Median auch die meisten der kleineren Distanzwerte, welche immerhin die Hälfte aller berechneten Distanzen ausmachen, besser sind als die der anderen Mediane. Entscheidend für den Erfolg ist auch, dass der Ausreißeranteil bei weniger als 0.5 liegt, weil sonst der Median selbst den Wert eines Ausreißers annimmt.

Durch den zusätzlichen Sortierschritt erhöht sich die Dauer der Berechnung mit dem RANSAC-Algorithmus, da für jede Iteration ein Sortieralgorithmus über allen Pixeln mit der Komplexität $O(n \log(n))$ ausgeführt werden muss. Dafür läuft der Algorithmus aber automatisiert ab und es muss nicht mehr manuell ein passender Distanzwert als Schwellwert ermittelt werden, wodurch sich im Großen und Ganzen doch wieder eine Zeit- und Aufwandsersparnis ergibt.

4.5.2 Ausdünnung der Korrespondenzen

Ein weiteres Problem des RANSAC-Algorithmus sind die hohen Iterationszahlen, die notwendig sind, um mit großer Wahrscheinlichkeit eine einigermaßen ausreißerfreie Ziehung aus den Punktkorrespondenzen zu bekommen. Dass der Anteil an Ausreißern im Regelfall nicht bekannt ist, erleichtert die Wahl der Iterationen hierbei ebenfalls nicht.

Im Grunde möchte man nur die am genauesten geschätzten Punktkorrespondenzen finden und mit ihrer Hilfe eine ebenfalls genaue Fundamentalmatrix berechnen. Kann man also vielleicht tendenziell schlechtere Punktkorrespondenzen im Vorhinein herausfiltern und den RANSAC-Algorithmus nur noch auf den übrig gebliebenen genaueren Korrespondenzen ausführen?

Die Bereiche, in denen die Flusschätzung wahrscheinlich ungenauer wird, dürften die sein, in denen der Datenterm im Energiefunktional zu klein wird und vom Glattheitsterm überwogen wird. Das passiert vor allem in homogenen Bildbereichen, in welchen wenig Farbänderungen und somit wenig Gradienteninformation vorhanden sind.

Diese Bereiche kann man bestimmen, indem man ein Gradientenfeld des Ausgangsbildes berechnet. Dann kann wiederum ein Schwellwert bestimmt werden, der die zu kleinen von den ausreichend großen Gradientenbeträgen trennt.

Der RANSAC-Algorithmus wird demnach nur noch über den Punktkorrespondenzen ausgeführt, deren Ausgangspunkte einen genügend großen Gradientenbetrag aufweisen. Diese Auswahl entspricht einem einfachen Kantenfilter, den man auf das Bild anwendet. Als solches Kantenbild kann die Auswahl der Punkte also visualisiert werden (siehe Abb. 4.2).

Je nach Wahl des Gradientenschwellwerts kann die Menge der Ausgangsdaten demnach drastisch verringert werden. Dies führt dazu, dass pro Iteration ebenso weniger Distanzen berechnet werden müssen, was den RANSAC-Algorithmus beschleunigt. Es können daher im Vergleich zum Algorithmus ohne Vorfilterung der Daten in gleicher Zeit mehr Iterationen ausgeführt werden.

Alternativ könnte man statt dem Gradientenbetrag auch den Strukturtenor für jeden Punkt im Ausgangsbild aufstellen und auf diesem eine Eigenwertanalyse machen. Hat der kleinere Eigenwert einen

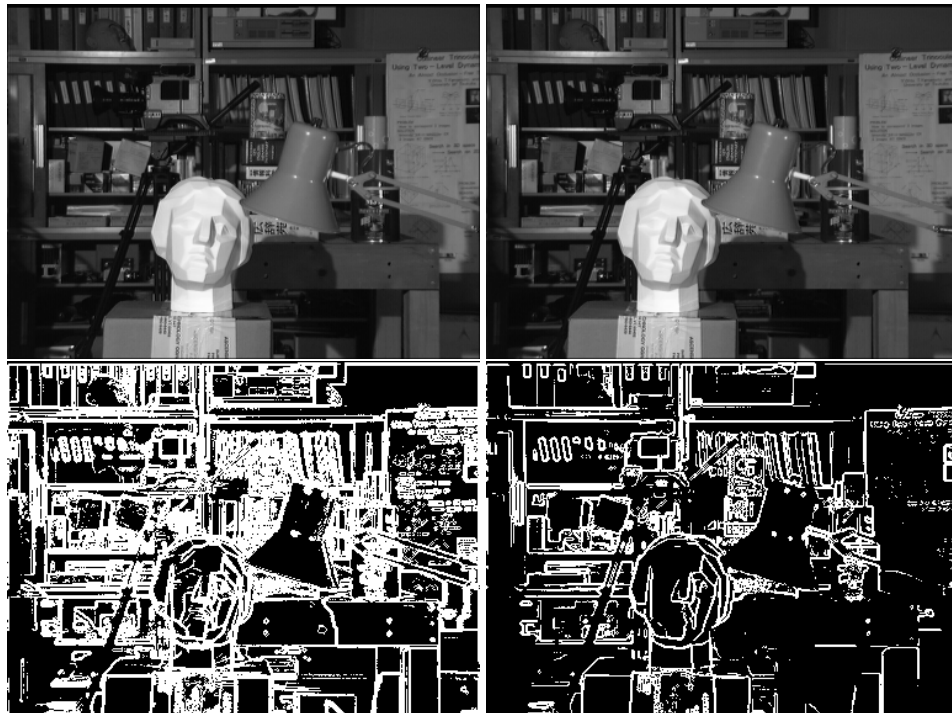


Abbildung 4.2: Gradientenfilter. **Oben:** Stereopaar *Tsukuba*. **Unten links:** Kantenbild mit dem Gradienten-Schwelldwert 5. **Unten rechts:** Kantenbild mit Gradienten-Schwelldwert 15.

Wert größer eines Schwellwerts, beinhaltet die Umgebung dieses Punktes Gradienteninformation in mehr als nur eine Richtung. Diese Filterung würde anstatt Kanten also Ecken erkennen. Gerade für diese sollte eine sehr genaue Bestimmung des optischen Flusses möglich sein. Dadurch würde auch eine größere Ausdünnung der Korrespondenzen erreicht werden.

4.5.3 SIFT und RANSAC

Angeregt vom vorherigen Abschnitt der Ausdünnung der Korrespondenzen drängt sich noch eine weitere Modifikation auf, die diese in die Spitze treibt. Mithilfe des bereits im Grundlagenkapitel vorgestellten SIFT-Deskriptors lassen sich wenige, dafür aber sehr genaue Punktkorrespondenzen (Feature-Matches) bestimmen.

Diese Matches sind exemplarisch für das *House*-Stereopaar in Abb. 4.3 zu sehen. Da sie deutlich spärlicher als bei der Kantenfilterung zuvor sind, können sie hier durch die Verbindungslinien der Start- und Zielpunkte visualisiert werden. Wie man aber durch die ein oder andere schräge Linie erkennen kann, ist die Menge der SIFT-Matches ebenfalls nicht frei von Ausreißern, weshalb die Anwendung des RANSAC-Algorithmus durchaus wieder Sinn macht. Auch die Subpixel-Präzision der gefundenen Feature-Punkte lässt hoffen, hiermit eine recht genaue Fundamentalmatrix berechnen zu

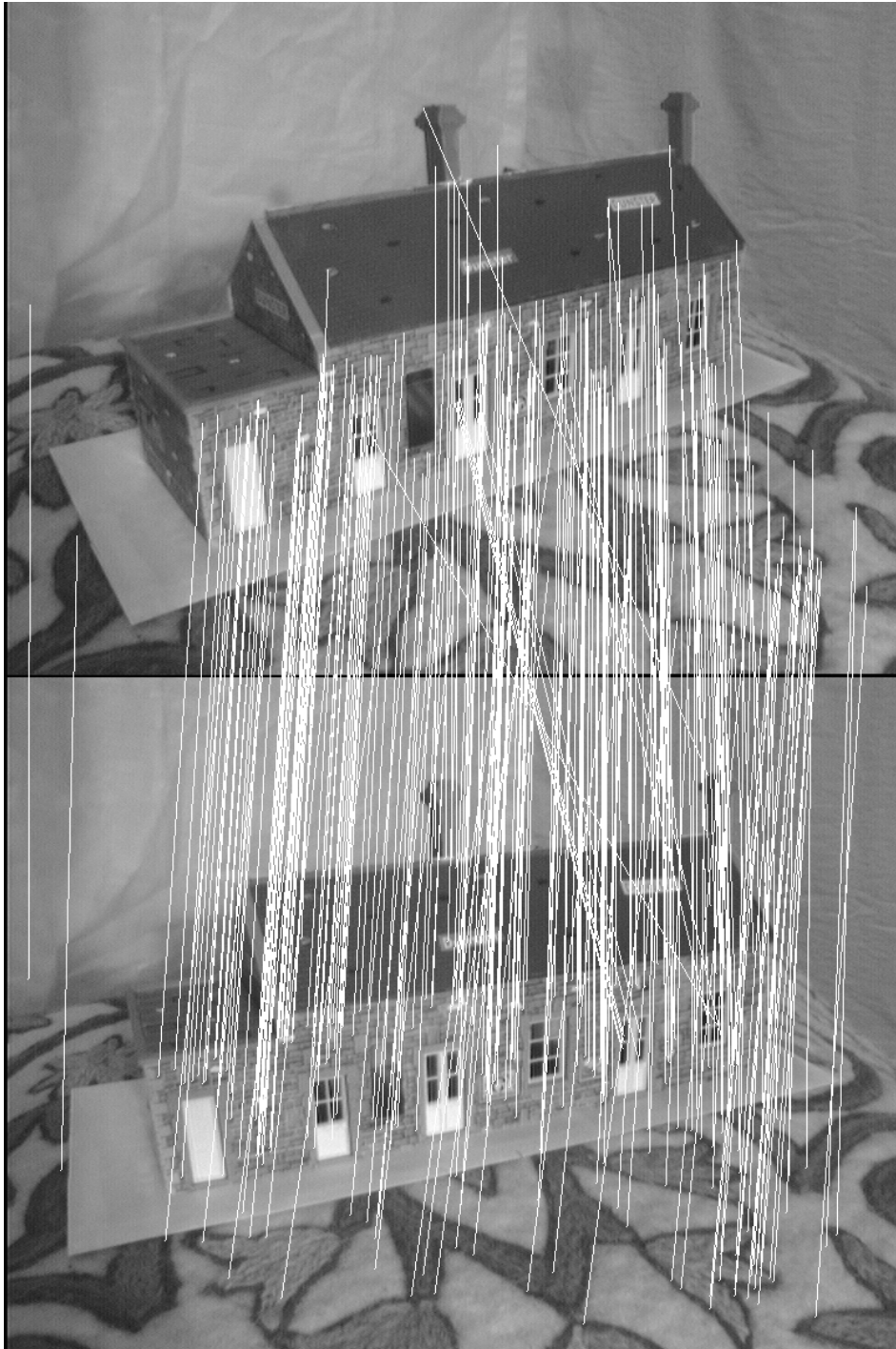


Abbildung 4.3: Gefundene SIFT-Matches für das *House*-Bildpaar. Kleine Menge von Korrespondenzen mit Ausreißern.

4 Bestimmung der Fundamentalmatrix

können. Dies bringt auch eine weitere Erhöhung der Geschwindigkeit des RANSAC-Algorithmus mit sich und ermöglicht es somit, noch mehr Iterationen in gleicher Zeit auszuführen.

Eine weitere Ausdünnung der Punktkorrespondenzen sollte dann keinen Sinn mehr machen, da mit weniger Korrespondenzen auch das Consensus-Set immer kleiner wird und an Aussagekraft verliert.

5 Evaluation

Im vorigen Kapitel wurden viele Möglichkeiten gezeigt, wie eine Fundamentalmatrix aus einer Menge gegebener Punktkorrespondenzen berechnet werden kann. Einige davon wurden im Zuge dieser Arbeit implementiert und sollen in diesem Kapitel ausgewertet und miteinander verglichen werden.

Dazu werden sie im Folgenden auf vier Testbildpaare angewandt, welche alle Stereo-Bildpaare sind, da hierfür immer eine Fundamentalmatrix für alle auftretenden Punktkorrespondenzen gültig ist. Eines davon ist das *Tsukuba*-Paar, das schon in Abb. 4.2 zu sehen war. Es hebt sich von den anderen drei Bildpaaren ab, da es mit orthoparallelen Kameras, wie in Abschnitt 2.3.2 beschrieben, aufgenommen wurde. Die anderen drei Bildpaare hingegen wurden mit beliebig konvergierenden Kameras aufgenommen. Das *House*-Bildpaar [Vis] war auch schon in Abb. 4.3 zu sehen und die Stereopaare *Javier* und *Aerial* [Vis] können in Abb. 5.1 betrachtet werden.

Im ersten Unterpunkt dieses Kapitels wird für die Bewertung der geschätzten Fundamentalmatrizen ein geeignetes Fehlermaß beschrieben, dessen Berechnung für den Vergleich der wahren Fundamentalmatrix mit den geschätzten Fundamentalmatrizen implementiert wurde. Anschließend wird im nächsten Abschnitt gezeigt, wie man aus Kamera-Projektionsmatrizen die Fundamentalmatrix gewinnen kann, um für unsere Testbildpaare deren wahre Fundamentalmatrix zu berechnen. Die restlichen Abschnitte beschäftigen sich dann mit der Auswertung der einzelnen Verfahren. Hierfür wird zunächst die Berechnung der Fundamentalmatrix aus dichten Korrespondenzen durchgeführt: erst einmal ohne, dann aber mit Normalisierung gemäß Abschnitt 4.3.3. Weiterhin wird auch ein M-Schätzer-Ansatz mit iterativer Lösung ausgewertet.

Danach werden verschiedene Verfahren untersucht, die den RANSAC-Algorithmus verwenden. Als erstes wird der normale Algorithmus angewandt, danach werden Variationen evaluiert, wie die Verwendung des Least Median, die Ausdünnung der Korrespondenzen mit einem Kantenfilter und schließlich die Kombination von SIFT-Features mit RANSAC. Zum Schluss folgt ein Vergleich zwischen der dichten Berechnung der Fundamentalmatrix und den spärlichen RANSAC-Verfahren.

5.1 Distanz zweier Fundamentalmatrizen

Zunächst wird ein Maß benötigt, um die Qualität der Ergebnisse der verschiedenen Verfahren miteinander vergleichen zu können. Da es sich bei dem in dieser Arbeit zu berechnenden Modell um eine Matrix handelt, ist die Entscheidung für das richtige Maß nicht so einfach. Würde es sich beispielsweise um zwei Vektoren handeln, deren räumliche Repräsentation man recht einfach visualisieren kann, so gäbe es eine Vielzahl an geometrisch aussagekräftigen Maßen, diese miteinander zu vergleichen. Dies wären etwa die Länge, der Winkel, das Skalarprodukt oder auch das Kreuzprodukt.

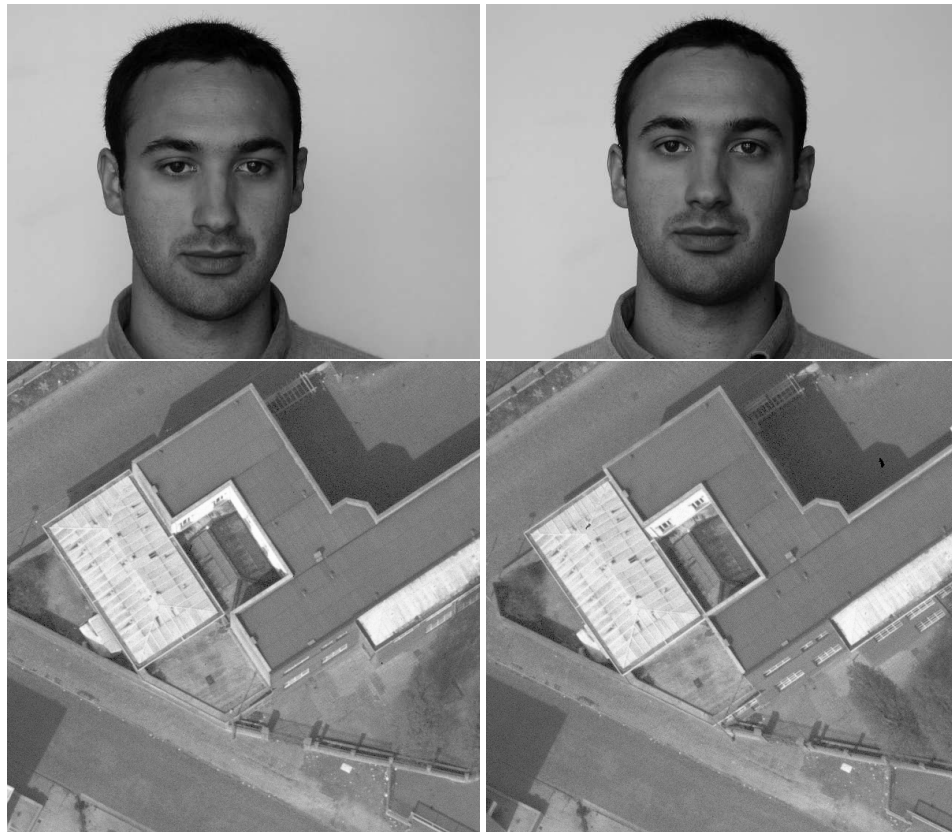


Abbildung 5.1: Stereo-Testbildpaare. **Oben:** Das *Javier*-Bildpaar mit homogenem Hintergrund.
Unten: Das *Aerial*-Bildpaar. Zwei Luftaufnahmen von Gebäuden.

Doch in unserem Fall handelt es sich um zwei Matrizen, die verglichen werden sollen. Diese lassen sich nicht einfach wie Vektoren im Raum darstellen. Ein bekanntes Vergleichsmaß, das sich für Matrizen anbietet und entsprechend dem Betrag eines Vektors gesehen werden kann, ist die Frobenius-Norm. Doch was sagt diese aus und ist deren Verwendung hier auch sinnvoll? Gerade bei Fundamentalmatrizen, welche nur von Rang 2 sind und sich um beliebige Faktoren unterscheiden können, sagt dieses Maß wenig aus, denn alle Fundamentalmatrizen können auf $\|F\| = 1$ normiert werden und verhalten sich immer noch gleich. Man könnte allerdings die beiden Matrizen normieren, sie dann voneinander subtrahieren und die Norm der resultierenden Matrix als Distanzmaß verwenden. Allerdings fehlt auch dafür jede geometrische Bedeutung und ein kleiner Wert dieser Distanz kann immer noch in komplett unterschiedlichen Epipolargeometrien münden.

Daher wird in dieser Arbeit ein anderes Maß zum Vergleich der Fundamentalmatrizen verwendet, welches auch eine aussagekräftige Bedeutung in der Epipolargeometrie hat: die Distanz nach Faugeras [FLP01]. Um diese berechnen zu können, muss, wie bereits ausgeführt, eine wahre und eine geschätzte Fundamentalmatrix für dasselbe Bildpaar vorliegen. Die Berechnung funktioniert dann folgendermaßen. Hierfür betrachte man in Abb. 5.2 zunächst das obere Bildpaar *Javier*.

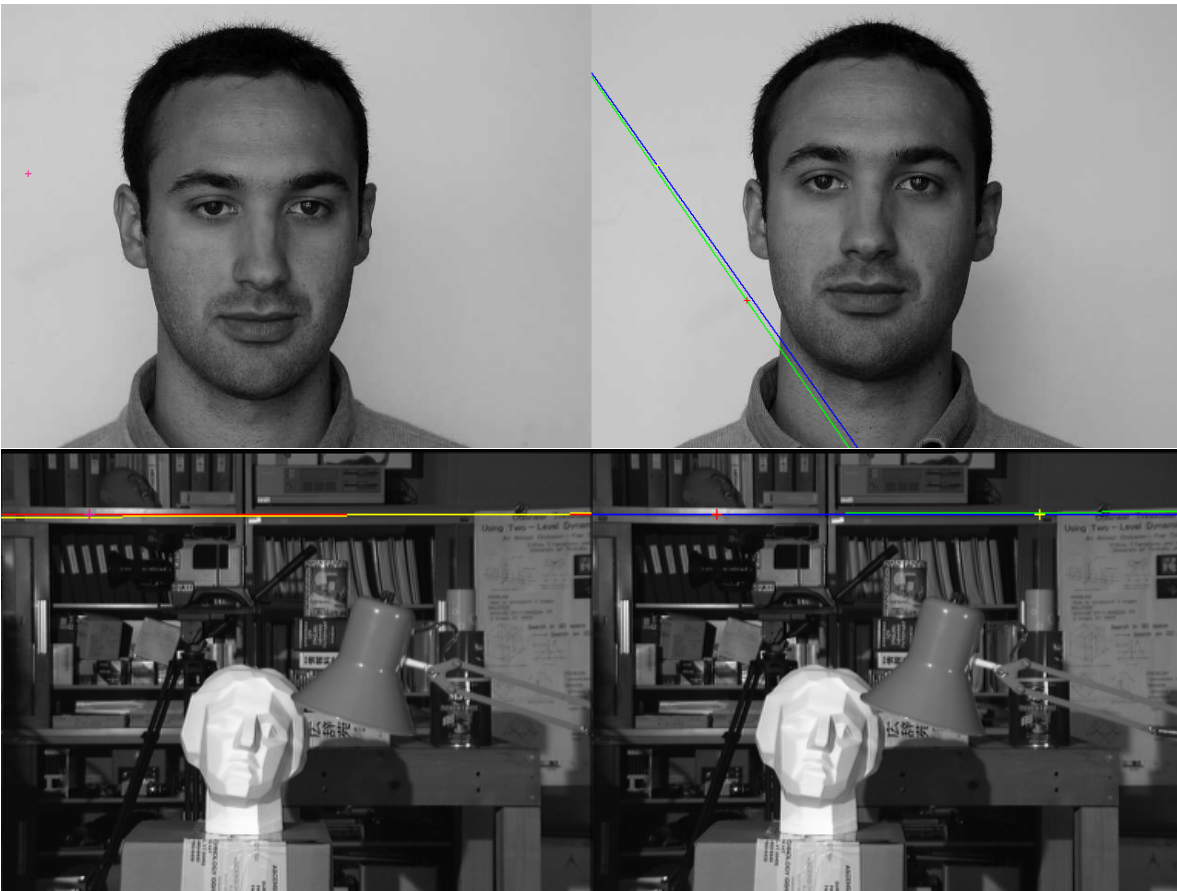


Abbildung 5.2: Teilschritte der Distanzberechnung nach Faugeras für einen zufällig platzierten Punkt. Die Größe der Fadenkreuze beträgt immer 7×7 Pixel. **Oben:** Javier-Bildpaar mit den Epipolarlinien zu einem zufällig gesetzten Punkt und den einzelnen Distanzen $d_1 = 5, 5$ und $d_2 = 3, 5$. **Unten:** Tsukuba-Bildpaar zusätzlich noch mit der Rückberechnung von den Punkten auf den Epipolarlinien im rechten Bild. Die einzelnen Distanzen sind $d_1 = 0, 2$; $d_2 = 1, 4$; $d_3 = 0, 2$ und $d_4 = 1, 5$.

Zunächst wählt man einen zufälligen Punkt m_1 im Ausgangsbild aus. Dieser ist im linken Bild durch ein rosanes Fadenkreuz markiert. Man berechnet nun mit beiden Fundamentalmatrizen dessen zugehörige Epipolarlinien im zweiten Bild

$$l_{2truth} = F_{truth} m_1, \quad (5.1)$$

welche durch die grüne Linie repräsentiert ist und

$$l_{2estimated} = F_{estimated} m_1, \quad (5.2)$$

dargestellt durch die blaue Linie. Auf diesen Linien wird jeweils wieder zufällig ein Punkt ausgewählt, der innerhalb des Bildbereiches liegt (rot und gelb). Um nun ein geometrisches Maß für die Distanz

der beiden Fundamentalmatrizen zu erhalten, kann man die geometrischen Distanzen der beiden Punkte zu jeweils der Epipolarlinie berechnen, auf der die Punkte nicht selbst liegen. Es ergeben sich also die Distanzen d_1 des roten Punktes zur blauen Linie und d_2 des gelben Punktes zur grünen Linie. Diese werden nach der Gleichung (4.14) für die nichtlineare Distanz aus Kapitel 4.3.2 berechnet, welche den Abstand der Punkte zur Epipolarlinie in Pixeln ausgibt.

Um einen ausgeglichenen Wert für die Distanz zu erhalten, werden aber noch zwei weitere Distanzen berechnet. Hierfür betrachte man nun das untere Bildpaar *Tsukuba* aus Abb. 5.2. Die beiden Punkte im rechten Bild werden nun wieder ins Ausgangsbild zurücktransformiert. Würde man dafür wieder ihre zugehörige Fundamentalmatrix verwenden, so würde man zwei Epipolarlinien erhalten, die genau durch den Ausgangspunkt (rosa) verlaufen. Deshalb wird stattdessen die jeweils andere Fundamentalmatrix verwendet und folgendermaßen die beiden Epipolarlinien im linken Bild berechnet:

$$l_{1rot} = F_{estimated}^T m_{2rot}, \quad (5.3)$$

$$l_{1gelb} = F_{truth}^T m_{2gelb}. \quad (5.4)$$

Die Linien sind nun in der gleichen Farbe gehalten, wie die Punkte, aus denen sie berechnet wurden. Eine Bezeichnung der berechneten Epipolarlinien mit *truth* oder *estimated* macht hier keinen Sinn mehr, da beide Linien mithilfe beider Fundamentalmatrizen berechnet wurden. Um nun die beiden Distanzen d_3 und d_4 zu gewinnen, berechne man den geometrischen Abstand des Ausgangspunktes m_1 zu den beiden Linien l_{1rot} und l_{1gelb} entsprechend zu den vorhergehenden Distanzen.

Die eben erklärte Vorgehensweise zur Berechnung der Distanzen d_1 , d_2 , d_3 und d_4 wiederhole man durch n -maliges zufälliges Setzen eines Punktes m_1^i , dann erhält man schließlich die gemittelte finale Distanz d zwischen den beiden Fundamentalmatrizen nach Faugeras durch

$$d = \frac{1}{4n} \sum_{i=1}^n (|d_1^i| + |d_2^i| + |d_3^i| + |d_4^i|). \quad (5.5)$$

Diese Distanz steht also für einen gemittelten Abstand der Epipolarlinien der wahren und der geschätzten Fundamentalmatrix zueinander, angegeben in Pixeln. Für die Anzahl $n = 100000$ haben sich unabhängig berechnete Distanzwerte für die gleichen Matrizen meist erst in der dritten signifikanten Stelle unterschieden. Daher wurde dieser Wert für alle Distanzberechnungen in den in dieser Arbeit implementierten Verfahren verwendet.

Zur Verifikation der richtigen Implementierung der Distanz nach Faugeras wurde eine Visualisierung für die einzelnen zufällig gewählten Punkte und zugehörigen Epipolarlinien und daraufliegenden Punkte implementiert, welche in Abb. 5.2 gezeigt wird und hier zur anschaulichen Erklärung der Berechnung der Faugeras-Distanz zuhilfe genommen wurde.

5.2 Berechnung der Fundamentalmatrix aus Projektionsmatrizen

Um die Distanz aus dem vorhergehenden Abschnitt berechnen zu können, ist es unabdinglich, die wahre Fundamentalmatrix der Testbildpaare zu kennen. Viele Stereobildpaare haben diese allerdings

nicht explizit angegeben, sondern vielmehr sind die 2 Projektionsmatrizen der Kameras gegeben, welche jeweils die extrinsischen und intrinsischen Parameter bei der Aufnahme enthalten.

So ist dies auch der Fall bei den Bildpaaren *Aerial* und *House*, welche hier für die Evaluation zum Einsatz kommen. Diese Bildpaare sind von der *Visual Geometry Group* der *University of Oxford* [Vis] bereitgestellt und haben die Projektionsmatrizen der Kameras beigefügt.

Die Berechnung funktioniert nach folgender Formel:

$$F = A_{int2}^{-T} [t]_{\times} R A_{int1}^{-1}, \quad (5.6)$$

mit den relativen extrinsischen Kamera-Parametern R und $[t]_{\times}$, wobei sich die Matrix $[t]_{\times}$ in Multiplikation mit einem Vektor wie das Kreuzprodukt des Vektors t verhält:

$$t \times m = [t]_{\times} m \quad (5.7)$$

Sie lautet daher

$$[t]_{\times} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}. \quad (5.8)$$

Bei der Implementierung dieser Berechnung wurde die OpenCV-Bibliothek [Bra00] verwendet, welche eine einfache Zerlegung einer Projektionsmatrix in ihre intrinsische Matrix, ihre Rotationsmatrix und ihren Translationsvektor ermöglicht hat. Die Berechnung der Translationsvektoren t wurde allerdings hierbei selbst mithilfe der Projektionsmatrizen und den erhaltenen intrinsischen Matrizen vorgenommen, da die zerlegten Vektoren nicht korrekt zu sein schienen. Nach der Berechnung der extrinsischen Matrizen konnten diese nach folgender Formel zu einer relativen extrinsischen Matrix

$$A_{ext} = A_{ext2} A_{ext1}^{-1} \quad (5.9)$$

multipliziert werden. Diese enthält dann die benötigte relative Rotation R sowie die relative Translation t . Nun muss nur noch obige Formel (5.6) berechnet werden und man erhält die Fundamentalmatrix.

Durch den Einsatz dieser Berechnung stehen nun, wie beispielsweise durch [Vis], viele Möglichkeiten offen, Stereobildpaare mit bekannter, wahrer Fundamentalmatrix zu finden, was die bessere Auswertung der folgenden Verfahren ermöglicht.

5.3 Dichte Verfahren

Zu Beginn dieser Auswertung müssen für alle Testbildpaare jedoch nicht nur ihre wahren Fundamentalmatrizen bereit stehen, sondern auch deren optische Flussfelder berechnet werden. Dies wurde hier mit dem Verfahren von Demetz *et al.* [DSV⁺14] getan, wobei ein Fluss-Regularisierungsterm der ersten Ordnung verwendet wurde und beim Datenterm auf die Gradientenkonstanz verzichtet wurde.

Normalisierung	Tsukuba	Aerial	Javier	House
Ohne	42,58	11,89	6,67	84,28
Mit	4,56	3,21	3,18	19,27

Tabelle 5.1: Faugeras-Distanz bei Verwendung aller Korrespondenzen. Vergleich der Ergebnisse ohne und mit Normalisierung.

5.3.1 8-Punkt-Algorithmus

Nachdem die Flussfelder berechnet wurden, können diese als Eingangsdaten für die Berechnung der Fundamentalmatrix eingesetzt werden. Hierfür wurde zunächst der 8-Punkt-Algorithmus implementiert, welcher sowohl mit genau 8, als auch beliebig viel mehr Punktkorrespondenzen verwendet werden kann. Als einfachste Möglichkeit zur Berechnung der Fundamentalmatrix können daher alle Punktkorrespondenzen berücksichtigt und für den 8-Punkt-Algorithmus verwendet werden. Die Werte für die vier Testbildpaare sind in Tabelle 5.1 in der oberen Zeile zu sehen.

5.3.2 Normalisierung

Wie in Abschnitt 4.3.3 aber schon erläutert, haben bei dieser einfachen Berechnung die verschiedenen Punktkorrespondenzen einen unterschiedlich starken Einfluss im Energiefunktional. Um dem entgegen zu wirken, wurde als nächster Schritt die Normalisierung der Start- und Zielpunktewolken vorgenommen. Die entsprechenden Werte sind ebenfalls in Tabelle 5.1 zu sehen.

Wie unschwer zu erkennen ist, ist der Fehler bei verwendeter Normalisierung durchwegs deutlich geringer als bei der Berechnung ohne Normalisierung. Es ist also in jedem Fall sinnvoll, diese bei der Verwendung von mehr als 8 Punktkorrespondenzen im 8-Punkt-Algorithmus einzusetzen. Verwendet man allerdings genau 8 Punktkorrespondenzen, so macht sie keinen Unterschied, da die Berechnung der Fundamentalmatrix in diesem Fall eindeutig ist.

Weiterhin wurde auch die Sicherstellung von Rang 2 durch eine Singulärwertzerlegung implementiert. Allerdings änderten sich die Ergebnisse damit kaum, sodass dies hier nur am Rande erwähnt sein soll.

Bei allen Verfahren wurde ferner auch die Laufzeit der Berechnungen in Sekunden genau gemessen, um diese vergleichen zu können. Da die Lösung des 8-Punkt-Algorithmus hier aber noch ein lineares Problem ist und kein iterativer Ansatz notwendig ist, benötigt diese weit unter einer Sekunde.

Ebenso wurden für jedes der Testbildpaare verschiedene Parameter α ausprobiert, welcher für den Grad der Glattheit in der Berechnung des optischen Flussfeldes verantwortlich ist. Es wurde letztendlich ein festes α ausgewählt, welches relativ gute Werte bei der normierten Berechnung der Fundamentalmatrix erzielt hat.

Zudem wurde festgestellt, dass besser geschätzte Flussfelder, deren Distanz ohne Verwendung der Normalisierung bereits niedrige Werte erreicht, auch unter Verwendung der Normalisierung bessere

Werte als die anderen Flussfelder erreichen. Deshalb wurde für alle weiteren Experimente - sofern mehr als 8 Punktkorrespondenzen verwendet wurden - die Normalisierung durchgeführt.

5.3.3 M-Schätzer

Jedoch lassen sich bei der Verwendung dichter Korrespondenzen für die Berechnung der Fundamentalmatrix noch weitere Verbesserungen erzielen. Hierfür wurde in dieser Arbeit ein M-Schätzer-Ansatz implementiert, dessen Lösung durch die iterative Neuberechnung der Fundamentalmatrix realisiert wird.

In den Tabellen 5.2 bis 5.5 werden dafür für jedes der Bildpaare verschiedene Iterationszahlen von 1 bis 1000 probiert und ebenso verschiedene Werte ϵ , welche als Parameter zur Robustifizierung verwendet werden.

Was zunächst auffällt, ist die Tatsache, dass die Werte mit gleichem ϵ und immer höher werdender Iterationszahl gegen einen festen Grenzwert zu konvergieren scheinen. Dies ist auch ein gewünschtes Verhalten, da mit fortschreitenden Iterationen die Energie des Funktionals immer weiter minimiert werden soll. Allerdings fällt weiterhin auf, dass trotz dieser Minimierung der Energie die Werte nicht für alle ϵ gegen einen Grenzwert konvergieren, der kleiner als der Anfangswert ist. Stattdessen wird hier manchmal ein Grenzwert angenommen, der größer ist. Daher ist es wichtig, die passende Wahl des ϵ -Parameters zu beachten, wenn dieses Verfahren angewendet wird.

Für die Iterationszahl 1 gilt, wie schon in Abschnitt 4.3.1 erklärt, dass noch keine berechnete Fundamentalmatrix aus einer vorhergehenden Iteration gegeben ist und somit das Gewicht für alle Korrespondenzen auf 1 gesetzt wird. Daher wird hier für jedes ϵ der gleiche Wert errechnet, welcher nur durch die zufallsbedingte Berechnung der Faugeras-Distanz kleine Abweichungen haben kann. Dieser Wert stimmt natürlich mit dem Wert aus Tabelle 5.1 bei verwendeter Normalisierung überein.

Bei den hier verwendeten Testbildern wurde der Grenzwert fast immer bereits nach 50 Iterationen erreicht, wonach sich keine merklichen Änderungen mehr einstellen. Für die Wahl des Parameters ϵ kann man leider keinen Wert festlegen, welcher für alle Bildpaare den besten Wert erzielt. Im Einzelnen ist für die Bildpaare *Tsukuba*, *Aerial* und *House* ein Wert kleiner gleich 0,0001 ideal, während beim *Javier*-Bildpaar ein Wert größer gleich 0,1 angebracht ist. Es scheint also für jedes Flussfeld hier einen individuellen Bereich zu geben, in welchem man den Parameter ϵ setzen sollte. Dieser hängt von der Qualität der geschätzten Punktkorrespondenzen ab.

Bei fast allen Testbildern lässt sich eine deutliche Verbesserung der berechneten Distanzen bei entsprechend gewähltem ϵ beobachten. Besonders bei den Bildpaaren *Tsukuba* und *Aerial* (Tabellen 5.2 und 5.3) sind nun sehr gute Distanzwerte erreicht worden, welche sich mit 0,39 bzw. 0,82 sogar im Subpixelbereich befinden. Dass *Tsukuba* hier den besten Wert erreicht, ist nicht sehr überraschend, da es das Bildpaar ist, welches mit orthoparallelen Kameras aufgenommen wurde. Leider konnte das *House*-Bildpaar keine so guten Werte erzielen, wobei sich die Distanz mit einem Bestwert von 9,23 immerhin auf etwas weniger als die Hälfte verringern ließ.

5 Evaluation

Iterationen	$\epsilon = 0,1$	$\epsilon = 0,01$	$\epsilon = 0,001$	$\epsilon = 0,0001$	$\epsilon = 0,00001$	$\epsilon = 0,000001$
1	4,58	4,56	4,56	4,58	4,55	4,56
2	4,38	2,36	1,10	1,84	2,49	2,92
5	4,38	2,18	0,55	0,54	1,07	1,48
10	4,37	2,18	0,55	0,41	0,49	0,72
20	4,37	2,18	0,55	0,40	0,39	0,41
50	4,37	2,18	0,55	0,41	0,39	0,39
100	4,37	2,18	0,55	0,40	0,39	0,39
1000	4,37	2,19	0,55	0,40	0,39	0,39

Tabelle 5.2: Tsukuba. Werte für den iterativen M-Schätzer-Ansatz bei verschiedenen Iterationszahlen und verschiedenen Robustifizierungsparametern ϵ .

Iterationen	$\epsilon = 0,1$	$\epsilon = 0,01$	$\epsilon = 0,001$	$\epsilon = 0,0001$	$\epsilon = 0,00001$	$\epsilon = 0,000001$
1	3,21	3,21	3,21	3,21	3,21	3,20
2	3,17	2,14	1,73	2,14	2,41	2,56
5	3,15	1,90	0,94	1,25	1,59	1,82
10	3,15	1,89	0,87	0,90	1,09	1,27
20	3,16	1,89	0,87	0,83	0,86	0,94
50	3,16	1,89	0,87	0,82	0,82	0,82
100	3,15	1,89	0,87	0,82	0,82	0,82
1000	3,16	1,89	0,87	0,82	0,82	0,82

Tabelle 5.3: Aerial. Werte für den iterativen M-Schätzer-Ansatz bei verschiedenen Iterationszahlen und verschiedenen Robustifizierungsparametern ϵ .

Eine Ausnahme stellt das Bildpaar *Javier* dar, bei welchem anscheinend bereits von Anfang an der bestmögliche Grenzwert von etwa 3,17 erreicht war. Es war hier durch das iterative Verfahren nicht mehr möglich, diesen Wert zu unterbieten.

Dass *Tsukuba* und *Aerial* deutlich bessere Werte erreichen als *Javier* und *House*, lässt sich dadurch begründen, dass das geschätzte Flussfeld bei den beiden (beim orthoparallelen *Tsukuba* sowieso) von besserer Qualität ist.

Das mag auch daran liegen, dass bei den hier schlechter abschneidenden Bildpaaren jeweils eine große Bildfläche mit einem homogenen Hintergrund ausgefüllt ist, welcher sich beim *House*-Bildpaar sogar, entgegen dem Rest des Bildbereiches, nicht bewegt. Dies könnte auch eine Erklärung dafür sein, dass das *House*-Bildpaar noch deutlich schlechtere Werte aufweist als die anderen drei Bildpaare.

Trotzdem kann man abschließend für die Berechnung der Fundamentalmatrix mithilfe vieler Korrespondenzen sagen, dass neben dem Einsatz der Normalisierung, welche bereits deutliche Verbesserungen der Distanzwerte für alle betrachteten Bildpaare erreichen konnte, auch der Einsatz von

Iterationen	$\epsilon = 0, 1$	$\epsilon = 0, 01$	$\epsilon = 0, 001$	$\epsilon = 0, 0001$	$\epsilon = 0, 00001$	$\epsilon = 0, 000001$
1	3,17	3,17	3,17	3,17	3,18	3,17
2	3,17	3,29	3,37	3,31	3,29	3,26
5	3,17	3,30	3,46	3,45	3,39	3,36
10	3,17	3,29	3,48	3,54	3,52	3,48
20	3,17	3,31	3,50	3,58	3,57	3,57
50	3,18	3,30	3,50	3,57	3,58	3,59
100	3,18	3,29	3,50	3,57	3,57	3,59
1000	3,18	3,29	3,50	3,58	3,59	3,59

Tabelle 5.4: Javier. Werte für den iterativen M-Schätzer-Ansatz bei verschiedenen Iterationszahlen und verschiedenen Robustifizierungsparametern ϵ .

Iterationen	$\epsilon = 0, 1$	$\epsilon = 0, 01$	$\epsilon = 0, 001$	$\epsilon = 0, 0001$	$\epsilon = 0, 00001$	$\epsilon = 0, 000001$
1	19,25	19,32	19,21	19,31	19,27	19,25
2	13,83	14,34	16,11	16,79	17,30	17,75
5	13,33	10,29	11,66	12,90	13,78	14,69
10	13,32	9,73	9,86	10,53	11,17	11,90
20	13,35	9,71	9,36	9,50	9,79	10,13
50	13,34	9,70	9,31	9,24	9,26	9,31
100	13,34	9,70	9,31	9,24	9,23	9,23
1000	13,31	9,71	9,32	9,23	9,23	9,25

Tabelle 5.5: House. Werte für den iterativen M-Schätzer-Ansatz bei verschiedenen Iterationszahlen und verschiedenen Robustifizierungsparametern ϵ .

M-Schätzern sinnvoll ist. Dieser kann zwar nicht immer garantieren, einen besseren Wert für die Distanz zu erreichen, schafft dies aber im Großteil der Fälle. Hier sind auch deutliche Verbesserungen möglich, welche sogar bis in den Subpixelbereich führen. Bei richtiger Wahl des ϵ -Parameters und der Verwendung von 50 oder mehr Iterationen sollte sich der Distanzwert zumindest einmal nicht verschlechtern.

Zur Laufzeit dieses Verfahrens lässt sich zunächst sagen, dass diese linear von der Anzahl der zu berechnenden Iterationen und der Anzahl an Korrespondenzen der Bildpaare (Größe der Ausgangsbilder) abhängt. Bei der größten gewählten Iterationszahl 1000, die bei der Berechnung dieser Tabellen gewählt wurde, reichten die Laufzeiten von 11 Sekunden (*Tsukuba*-Bildpaar mit der kleinsten Auflösung von 384×288 Pixeln) bis zu 45 Sekunden (*House*-Bildpaar mit der größten Auflösung von 768×576 Pixeln). Alle Messungen wurden dabei vom selben PC mit einem Intel Core i7-4500U Prozessor durchgeführt, wobei jede Berechnung auch von nur einem Kern durchgeführt wird, welcher in diesem Fall mit einer Frequenz von 1,8-3,0 GHz taktet. Wenn man nun die Iterationszahl auf 50 Iterationen

beschränkt, welche in diesen Messungen fast immer ausgereicht haben, um den Konvergenzwert zu erreichen, verringert sich die Laufzeit dieses Verfahrens auf unter 2,5 Sekunden.

5.4 RANSAC-basierte Verfahren

Nachdem gerade die Berechnung der Fundamentalmatrix durch viele Punktkorrespondenzen teils sehr gute Werte mit Subpixel-Genauigkeit erreicht hat, was aber leider nicht für alle Testbildpaare gilt, soll nun untersucht werden, welche Werte die Verfahren unter Verwendung des RANSAC-Algorithmus erreichen. Da diese aber aufgrund der beinhalteten zufälligen Auswahl von 8 Punktkorrespondenzen stark unterschiedliche Werte aufweisen können, soll auf folgende Weise für deren Evaluation vorgegangen werden:

Es wird für jede zu testende Parameterbelegung eines Verfahrens hundert mal die Fundamentalmatrix berechnet. Natürlich wird für jede der Matrizen auch wieder die Faugeras-Distanz zur wahren Fundamentalmatrix berechnet. Diese hundert Werte werden mithilfe eines Quicksort-Algorithmus sortiert und es werden vier Werte in den Tabellen zur Evaluation festgehalten. Das Minimum, das Maximum, der Median und der Mittelwert der berechneten Werte. Dies repräsentiert die Verteilung der Werte recht gut und kompakt.

5.4.1 RANSAC

Kommen wir also zum ersten implementierten RANSAC-basierten Verfahren, das ausgewertet werden soll. Hierfür wird zunächst der normale RANSAC-Algorithmus über allen Punktkorrespondenzen ausgeführt, es werden also mit vorzugebener Iterationszahl immer zufällig 8 Korrespondenzen ausgewählt, deren Fundamentalmatrix berechnet wird und das zugehörige Consensus-Set dazu bestimmt. Die Matrix mit dem größten Set wird schließlich verwendet und deren Faugeras-Distanz berechnet.

Bestimmung des Schwellwerts

Beim RANSAC-Algorithmus gibt es, wie in Abschnitt 4.4.3 ausgeführt, verschiedene Werte, die Einfluss auf den Erfolg des Verfahrens haben. Dies sind vor allem die Wahl des Schwellwerts t und die Anzahl an durchgeführten Iterationen n . In diesem Teil soll zunächst für jedes der Testbildpaare ein geeigneter Schwellwert gefunden werden, er soll also empirisch bestimmt werden.

Hierfür werden in den Tabellen 5.6 bis 5.9 Werte von 0,1 bis 10 ausprobiert, welche sich von einem zum nächsten in etwa verdoppeln. Dabei werden immer fest 100 Iterationen verwendet, was nicht sehr viel für das RANSAC-Verfahren ist, jedoch kann hier bereits die Eignung der verschiedenen Schwellwerte beurteilt werden. Zum Vergleich wird anstatt eines Schwellwerts auch noch der Least Median angegeben.

Bei den Testbildpaaren *Tsukuba* und *Javier* fällt die Wahl des besten Schwellwerts leicht, da alle vier Testwerte bei dem gleichen Schwellwert ihr Minimum erzielen. Hierbei seien die Least-Median-Werte vorerst nicht berücksichtigt. Es lassen sich daher bereits die Schwellwerte $t_{Tsukuba} = 0,5$ und

Schwellwert t	Minimum	Maximum	Median	Mittelwert
0,1	0,18	34,58	2,16	3,27
0,2	0,25	9,78	2,28	2,84
0,5	0,11	8,01	1,95	2,26
1	0,45	13,10	5,23	5,32
2	0,83	29,53	9,44	9,47
5	0,94	59,37	31,36	30,64
10	0,64	92,84	22,14	27,45
Least Median	0,14	15,27	1,94	2,64

Tabelle 5.6: Tsukuba. Werte bei Auswahl der Korrespondenzen mit RANSAC bei hundert Ziehungen und verschiedenen Schwellwerten t .

Schwellwert t	Minimum	Maximum	Median	Mittelwert
0,1	0,50	22,05	2,81	3,77
0,2	0,49	15,47	2,87	3,36
0,5	0,59	8,78	3,30	3,45
1	0,35	10,83	3,09	3,44
2	0,28	10,88	3,64	3,85
5	0,66	25,77	9,89	10,76
10	2,82	40,58	12,32	14,31
Least Median	0,55	15,41	3,20	3,83

Tabelle 5.7: Aerial. Werte bei Auswahl der Korrespondenzen mit RANSAC bei hundert Ziehungen und verschiedenen Schwellwerten t .

Schwellwert t	Minimum	Maximum	Median	Mittelwert
0,1	2,82	168,91	71,99	72,17
0,2	2,30	152,97	74,70	70,40
0,5	1,82	153,84	25,70	37,29
1	1,56	41,09	5,21	6,17
2	1,54	15,66	4,94	5,65
5	1,91	22,27	6,69	7,83
10	2,25	41,19	11,52	15,89
Least Median	1,72	152,32	50,23	53,23

Tabelle 5.8: Javier. Werte bei Auswahl der Korrespondenzen mit RANSAC bei hundert Ziehungen und verschiedenen Schwellwerten t .

Schwellwert t	Minimum	Maximum	Median	Mittelwert
0,1	1,37	199,24	41,31	66,06
0,2	1,71	181,95	23,40	51,15
0,5	1,11	187,49	17,53	36,86
1	2,60	67,53	11,83	15,50
2	1,69	43,04	15,38	17,03
5	1,03	62,88	13,51	17,51
10	2,35	64,67	19,89	23,02
Least Median	2,33	46,97	11,74	14,86

Tabelle 5.9: House. Werte bei Auswahl der Korrespondenzen mit RANSAC bei hundert Ziehungen und verschiedenen Schwellwerten t .

$t_{Javier} = 2,0$ festhalten. Für die anderen beiden Testbildpaare müssen aber weitere Überlegungen angestellt werden, da nicht alle Minima der vier Testwerte zum gleichen Schwellwert gehören.

Hier werden für die Entscheidung die Werte des besten Median und des besten Mittelwerts zurate gezogen, da diese robuster und aussagekräftiger sind als der maximale und minimale Wert, welcher doch sehr zufällig einmal einen sehr kleinen oder sehr großen Wert annehmen kann. Dadurch erübrigt sich die Entscheidung für das *House*-Bildpaar, denn mit dem Schwellwert $t_{House} = 1,0$ wird sowohl der minimale Median als auch der minimale Mittelwert erreicht. Damit steht noch die Entscheidung für das *Aerial*-Bildpaar aus. Diese ist ein wenig schwieriger zu treffen, da die minimalen Werte für Median und Mittelwert sich auf zwei verschiedene Schwellwerte verteilen. Letztendlich fiel die Entscheidung hier auf $t_{aerial} = 0,2$, da dessen Median den zweitbesten Wert und einen geringen Abstand zum minimalen Median aufweist. In den folgenden Berechnungen werden ab jetzt immer die gerade bestimmten Schwellwerte für die jeweiligen Bildpaare verwendet werden.

Nachdem nun die besten Schwellwerte für hundert Iterationen ermittelt wurden, stellt sich noch die Frage, wie sich die Werte unter der Verwendung des Least Median im Vergleich dazu schlagen. Für das *Tsukuba*- und *House*-Bildpaar scheint er sich sehr gut zu eignen, da hier mit die besten Werte für Median und Mittelwert erreicht werden können. Auch beim *Aerial*-Bildpaar stellen sich im Verhältnis gute Werte für den Least Median ein. Nur beim *Javier*-Bildpaar erreicht dieser sehr schlechte Werte, was wahrscheinlich darin begründet liegt, dass hier der Großteil des Bildbereiches den homogenen Hintergrund darstellt, für welchen das optische Flussfeld tendenziell größere Fehler beinhaltet. Daher fällt der Median aller Korrespondenzpaare sehr wahrscheinlich in genau diesen fehlerbehafteten Bereich, was der Grund für die hohen Distanzwerte sein kann.

Auch der RANSAC-Algorithmus an sich hat, wie schon im Abschnitt 4.4.3 beschrieben, seine Probleme mit größeren Ausreißermengen, da diese eventuell ein größeres Consensus-Set bilden können als die korrekt geschätzten Korrespondenzen. Dies mag der Grund sein, dass das *Javier*- und das *House*-Bildpaar des öfteren recht hohe Distanzwerte produzieren. Abb. 5.3 zeigt jeweils beispielhafte Consensus-Sets für das *Aerial*- und das *Javier*-Bildpaar. Es ist zu erkennen, dass bei *Aerial* der Großteil des Bildbereiches zum Consensus-Set gezählt wird, während bei *Javier* hauptsächlich der Kopfbereich dazu zählt und eine große Ausreißermenge übrig lässt.

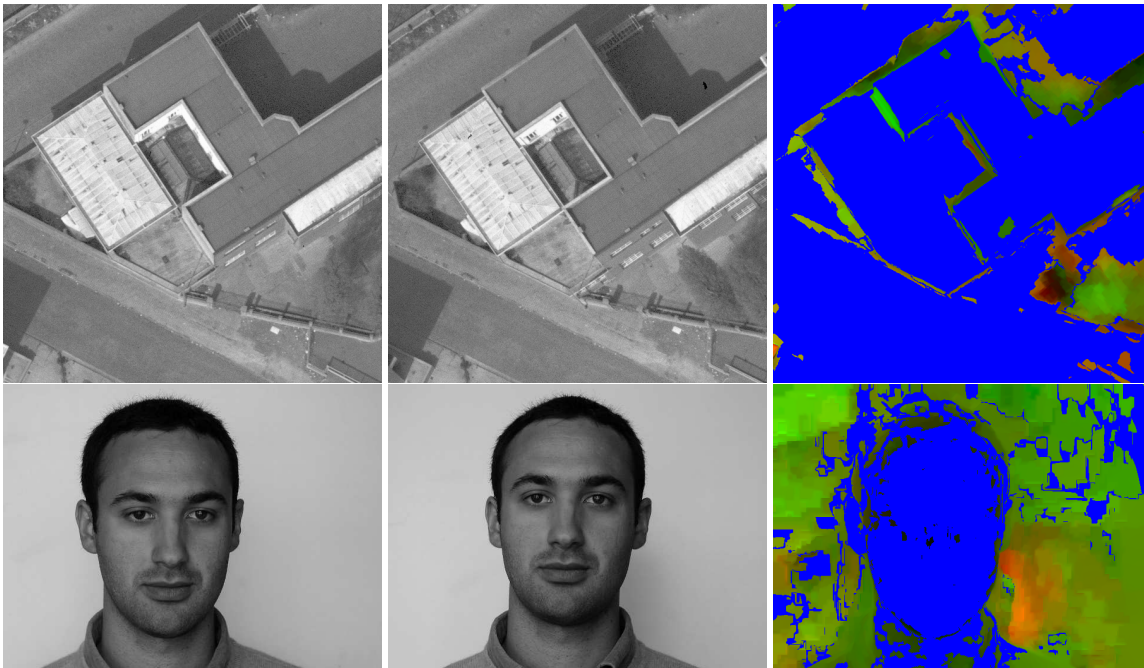


Abbildung 5.3: Consensus-Sets. **Oben:** Bildpaar *Aerial*. Auf der rechten Seite wird das größte Consensus-Set des RANSAC-Algorithmus blau markiert. Für Outlier wird das Flussfeld in einer Rot-Grün-Farbdarstellung angezeigt. **Unten:** Bildpaar *Javier*. Das größte Consensus-Set umfasst grob den Kopfbereich.

Allgemein lässt sich festhalten, dass sowohl zu kleine als auch zu große Schwellwerte große Fehler in den Distanzwerten produzieren können. Es ergibt sich daher ein kleiner Bereich für t , in dem brauchbare Werte entstehen, in denen also passende Consensus-Sets für die Korrespondenz-Ziehungen gefunden werden. Ist der Schwellwert zu groß, so werden zu viele Ausreißer mit zum Consensus-Set gezählt, was dessen Aussagekraft mildert. Ist er hingegen zu klein, so werden zu wenige korrekte Korrespondenzen dazugezählt, was die Auswahl der Fundamentalmatrix immer mehr zufällig erscheinen lässt, wodurch gerade beim *Javier*- und dem *House*-Bildpaar sehr hohe Distanzen entstehen.

Da bei diesen Berechnungen immer eine feste Iterationszahl von hundert Ziehungen verwendet wurde, unterscheiden sich deren Laufzeiten nur noch in linearer Abhängigkeit durch die unterschiedlichen Bildgrößen und somit durch die Anzahl der Korrespondenzen. Für die hier verwendeten Testbildpaare ergaben sich Laufzeiten von 31 Sekunden bis zu 2 Minuten und 11 Sekunden. Diese Zeit gilt dabei für die in der Evaluation nötige hundertmalige Auswertung. Ein einzelner Durchlauf des RANSAC-Algorithmus benötigte daher weniger als 1,5 Sekunden.

Einfluss Ziehungen

Abgesehen vom Schwellwert t hat aber, wie bereits erwähnt, auch die Anzahl der Ziehungen n beim RANSAC-Algorithmus einen großen Einfluss auf die Qualität der Ergebnisse. Daher werden in den

Ziehungen	Minimum	Maximum	Median	Mittelwert
100	0,11	8,01	1,95	2,26
250	0,13	5,29	1,98	2,18
500	0,24	6,34	1,89	2,17
1000	0,44	6,23	1,99	2,40
2000	0,07	5,57	2,17	2,29
4000	0,17	6,30	2,31	2,48

Tabelle 5.10: Tsukuba. Werte bei Auswahl der Korrespondenzen mit RANSAC bei variabler Anzahl an Ziehungen und festem Schwellwert $t = 0,5$.

Ziehungen	Minimum	Maximum	Median	Mittelwert
100	0,32	19,36	3,13	3,68
250	0,48	9,95	2,31	2,76
500	0,43	7,63	2,15	2,30
1000	0,24	5,87	1,77	1,95
2000	0,16	5,22	1,49	1,76
4000	0,29	4,33	1,37	1,59

Tabelle 5.11: Aerial. Werte bei Auswahl der Korrespondenzen mit RANSAC bei variabler Anzahl an Ziehungen und festem Schwellwert $t = 0,2$.

Tabellen 5.10 bis 5.13 diese variiert, mit festgelegten Schwellwerten, die entsprechend der Ergebnisse im vorigen Abschnitt gewählt werden. Es werden für alle Testbildpaare von 100 bis zu 4000 Iterationen im RANSAC-Algorithmus durchgeführt, wobei auch hier die Werte sich jeweils etwa verdoppeln.

Das erwartete Ergebnis ist natürlich, dass sich mit höherer Anzahl an Ziehungen auch die berechneten Werte für die Distanzen verringern und somit immer bessere Ergebnisse erreicht werden. Für das *Aerial*- und das *House*-Bildpaar sieht man diese Erwartung sehr schön bestätigt, wenn man von einzelnen kleineren Abweichungen absieht, wie etwa, dass die höchste Iterationszahl 4000 nicht auch das kleinste Minimum oder Maximum der Werte erreicht. Wie aber bereits erwähnt, lässt sich das durch die zufällige Natur der einzelnen berechneten Werte erklären und akzeptieren. Viel aussagekräftiger ist hier der stetige Rückgang der Werte des Medians und des Mittelwerts.

Beim *Javier*-Bildpaar ist die gleiche Tendenz zu erkennen, jedoch stellen gerade die Werte für 500 Ziehungen eine Ausnahme von der Regel dar. Hier scheinen etwas überdurchschnittlich gute Werte erreicht worden zu sein. Die Werte für 4000 Ziehungen stellen jedoch wieder den geringsten Mittelwert und den zweitbesten Median, weshalb im Groben die Erwartungen auch hier bestätigt werden.

Für das *Tsukuba* Bildpaar können die Erwartungen allerdings nicht bestätigt werden. Vielmehr halten sich die Werte für alle Anzahlen an Ziehungen etwa im gleichen Bereich. Bemerkenswert ist hier auch, dass die minimalen Distanzwerte durchwegs sehr gute Werte erreichen mit bis zu 0,07. Vielleicht unterstreicht dies wieder die Sonderrolle des Bildpaares, da es sich um einen orthoparallelen

Ziehungen	Minimum	Maximum	Median	Mittelwert
100	1,43	17,10	6,17	6,33
250	1,68	13,58	5,24	5,54
500	1,30	11,68	4,98	5,32
1000	1,22	13,24	5,35	5,79
2000	1,42	13,91	5,14	5,47
4000	1,38	11,64	5,05	5,23

Tabelle 5.12: Javier. Werte bei Auswahl der Korrespondenzen mit RANSAC bei variabler Anzahl an Ziehungen und festem Schwellwert $t = 2,0$.

Ziehungen	Minimum	Maximum	Median	Mittelwert
100	2,60	67,53	11,83	15,50
250	1,14	40,53	10,39	11,54
500	1,71	21,57	8,55	9,84
1000	1,57	30,18	6,53	8,19
2000	0,79	24,76	6,96	7,92
4000	1,19	22,19	5,94	6,10

Tabelle 5.13: House. Werte bei Auswahl der Korrespondenzen mit RANSAC bei variabler Anzahl an Ziehungen und festem Schwellwert $t = 1,0$.

Kameraaufbau handelt. Desweiteren wäre es eventuell möglich, dass hier eine Grenze der möglichen Genauigkeit erreicht wurde, die allein durch höhere Wahl der Ziehungen nicht mehr verbessert werden kann. Eine kleinere Wahl des Schwellwerts t könnte stattdessen weitere Verbesserungen ermöglichen, was im Folgenden allerdings nicht mehr untersucht werden soll. Stattdessen kann beim *Aerial*-Bildpaar beobachtet werden, dass dieses mit einem kleineren Schwellwert (0,2) als *Tsukuba* (0,5) auch im Median und im Mittelwert noch bessere Werte erreichen kann. Währenddessen schneiden die Bildpaare *Javier* und *House* mit einem deutlich höheren Schwellwert auch deutlich schlechter mit ihren Distanzwerten ab.

Die Laufzeiten sind auch hier wieder linear abhängig von sowohl der Anzahl der Ziehungen als auch der Anzahl der Korrespondenzen. Bei der höchsten Anzahl an Ziehungen von 4000 ergaben sie sich von 26 Minuten und 9 Sekunden bis 1 Stunde 32 Minuten und 3 Sekunden für die hundertmalige Ausführung und damit von bis zu einer Minute für die einmalige Ausführung.

Least Median

Nun stellt sich wieder die Frage, wie der Least Median im Vergleich zur Verwendung der Schwellwerte mit ebenfalls mehreren Ziehungen abschneidet. Hierfür wurden wieder verschiedene Anzahlen an Ziehungen von 100 bis 1000 ausprobiert. Es wurde also keine so hohe Iterationszahl gewählt wie beim

Ziehungen	Minimum	Maximum	Median	Mittelwert
100	0,14	15,27	1,94	2,64
250	0,10	7,91	1,54	1,84
500	0,098	6,63	1,11	1,45
1000	0,19	5,18	1,00	1,26

Tabelle 5.14: Tsukuba. Werte für das RANSAC-Verfahren unter Verwendung des Least Median und variabler Anzahl an Ziehungen.

Ziehungen	Minimum	Maximum	Median	Mittelwert
100	0,49	12,50	2,82	3,31
250	0,29	10,76	2,14	2,57
500	0,45	7,13	2,07	2,21
1000	0,35	4,55	1,66	1,80

Tabelle 5.15: Aerial. Werte für das RANSAC-Verfahren unter Verwendung des Least Median und variabler Anzahl an Ziehungen.

Schwellwert, da das Verfahren mit dem Least Median aufgrund seines jedes Mal zusätzlich benötigten Sortierschritts eine längere Laufzeit hat. Für die Ergebnisse betrachte man die Tabellen 5.14 bis 5.17.

Auch hier besteht die Erwartung darin, dass mit höherer Iterationszahl des Verfahrens bessere Werte erreicht werden. Im Fall des Least Median bestätigt sich diese Vermutung sehr schön für die drei Bildpaare *Tsukuba*, *Aerial* und *House*. Dieses Mal entstehen die unerwarteten Werte beim Bildpaar *Javier*. Wie es sich schon in Tabelle 5.8 angedeutet hat, ergeben sich auch hier durchwegs hohe Distanzwerte, welche sich auch in keinster Weise mit höheren Iterationszahlen verbessern.

Wie zuvor kann versucht werden, dies zu erklären, indem dieses Bildpaar einen größeren Ausreißerbereich hat als unterstützende Korrespondenzen (siehe Abb. 5.3) und deshalb der Median der geometrischen Distanzen der einzelnen Korrespondenzen in den Ausreißerbereich fällt. Aus diesem Grund ist er nicht aussagekräftig und kann unvorhersehbar verschiedenste Fundamentalmatrizen unterstützen. Die Verwendung des Least Median ist also nicht sinnvoll für Bildpaare mit größerem Ausreißerbereich als Inlierbereich.

Für die anderen Bildpaare allerdings lässt sich die Schätzung der Fundamentalmatrix mit höherer Anzahl an Ziehungen immer weiter verbessern. Gerade beim *Tsukuba*-Paar, bei welchem unter Verwendung des Schwellwerts keine Verbesserung mit höherer Iterationszahl mehr zu erkennen war, hilft der Least Median weiter. Da er unabhängig von Schwellwerten funktioniert, deren Wahl einen entscheidenden Einfluss auf die Genauigkeit des RANSAC-Algorithmus hat, begrenzt er diese mögliche Genauigkeit stattdessen nicht. Auch beim *House*- und dem *Aerial*-Bildpaar war hier noch keine Grenze der Verbesserung in Sicht.

Ziehungen	Minimum	Maximum	Median	Mittelwert
100	2,30	133,59	50,99	52,50
250	2,21	139,60	67,67	67,28
500	4,42	130,46	66,27	68,14
1000	39,89	134,72	65,94	73,34

Tabelle 5.16: Javier. Werte für das RANSAC-Verfahren unter Verwendung des Least Median und variabler Anzahl an Ziehungen.

Ziehungen	Minimum	Maximum	Median	Mittelwert
100	2,33	46,97	11,74	14,86
250	0,71	37,99	11,73	13,55
500	1,27	39,71	10,26	11,46
1000	0,84	22,57	7,47	8,62

Tabelle 5.17: House. Werte für das RANSAC-Verfahren unter Verwendung des Least Median und variabler Anzahl an Ziehungen.

Die Least-Median-Werte des *Tsukuba*- und *Aerial*-Stereopaars erreichten hier unter 1000 Ziehungen sogar bessere Werte als die Werte bei fest gewähltem Schwellwert. Beim *House*-Paar waren sie nicht ganz so gut und das Problem beim *Javier*-Paar wurde schon besprochen.

Die Laufzeiten lagen bei der höchsten Iterationszahl von 1000 zwischen 28 Minuten und 21 Sekunden und 2 Stunden 1 Minute und 46 Sekunden. Für die einfache Berechnung bedeutet das eine Laufzeit von unter 1,5 Minuten.

5.4.2 RANSAC ausgedünnt

Nachdem im vorigen Abschnitt der normale RANSAC-Algorithmus und eine erste Variation mit dem Least Median untersucht wurden, soll nun eine weitere Variation getestet werden. Hierfür wurde zunächst ein Gradientenfilter implementiert, welcher abhängig von einem Schwellwert $t_{Gradient}$ aus dem Ausgangsbild ein Kantenbild erstellt, welches Pixel mit einem höherem Betrag des Gradienten weiß und Pixel mit einem niedrigeren Betrag schwarz setzt. Dadurch können verschieden stark gefilterte Kantenbilder unserer Testbilder erstellt werden, wie in Abb. 5.4 zu sehen ist.

Diese Bilder sind notwendig, da diese nun vom Programm eingelesen werden und weiterhin nur noch die Kantenpixel mit ihren zugehörigen Korrespondenzen für den RANSAC-Algorithmus verwendet werden. Hierfür wurden für diese Testreihe zu jedem Bildpaar drei verschiedene Werte t_{grad} ausgewählt und deren Kantenbilder, die in Abb. 5.4 zu sehen sind, erzeugt. Für jeden Schwellwert wurde dann die Fundamentalmatrix mit dem ausgedünnten RANSAC-Algorithmus und deren Faugeras-Distanz berechnet. Dabei wurden wie im vorhergehenden Abschnitt verschiedene Iterationszahlen von 1000 bis 8000 verwendet, wobei je beim kleinsten Gradienten-Schwellwert die Testwerte für 8000

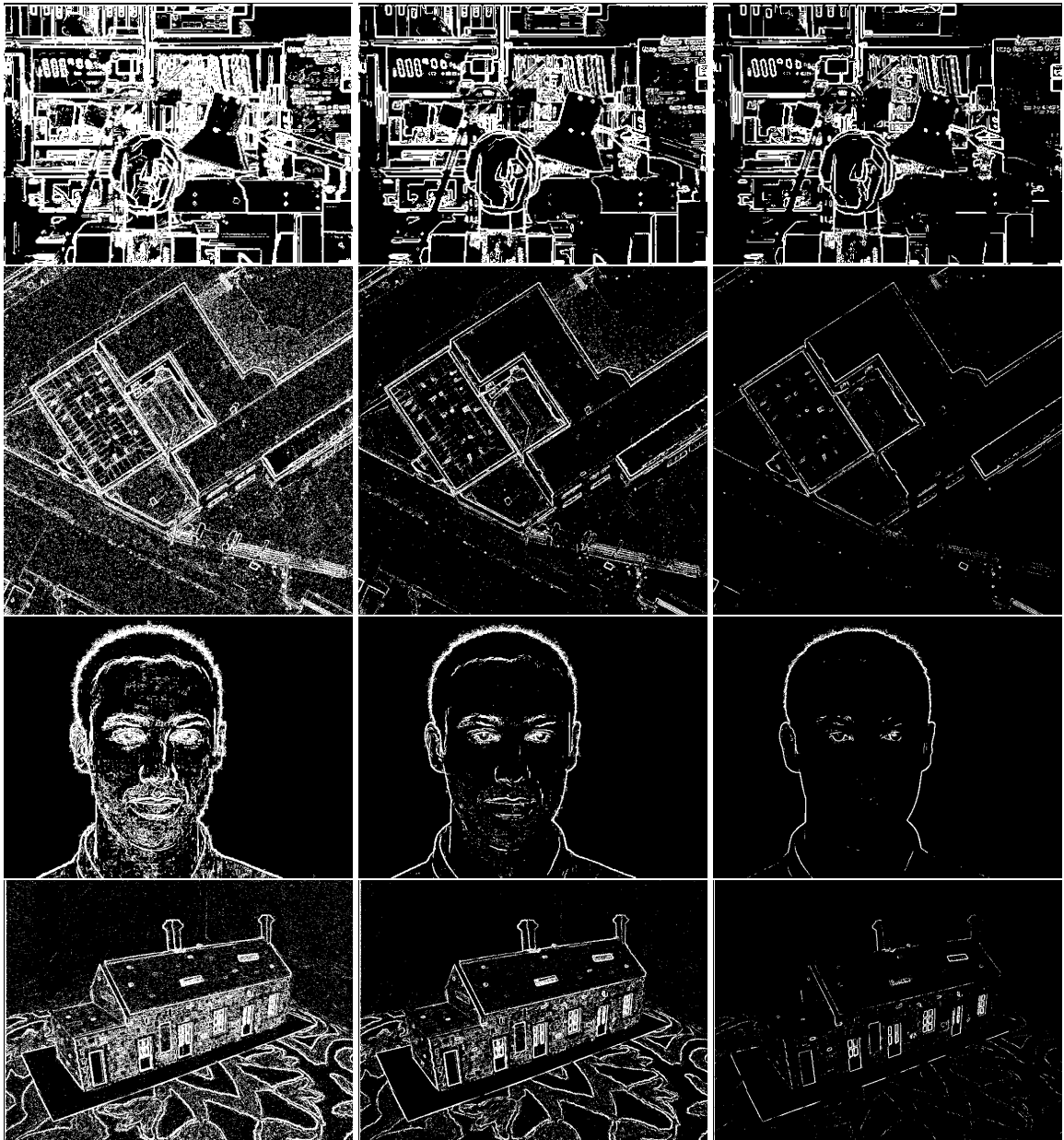


Abbildung 5.4: Gradientenbilder. Die kantengefilterten Ausgangsbilder der Testbildpaare. Weiße Pixel haben einen größeren Gradientenbetrag als t_{grad} , schwarze einen kleineren. Die verwendeten Werte für t_{grad} , in Reihenfolge der Bilder von links nach rechts: **Tsukuba:** 5, 10 und 15. **Aerial:** 10, 17 und 30. **Javier:** 5, 10 und 20. **House:** 7, 10 und 20.

Schwellwert t_{grad}	Ziehungen	Minimum	Maximum	Median	Mittelwert
5	1000	0,14	6,23	2,07	2,33
5	2000	0,47	6,94	2,36	2,62
5	4000	0,27	5,52	2,22	2,31
10	1000	0,38	6,80	2,44	2,83
10	2000	0,24	5,73	2,10	2,50
10	4000	0,15	7,20	1,96	2,37
10	8000	0,14	7,22	2,04	2,42
15	1000	0,14	7,64	2,18	2,52
15	2000	0,25	6,49	1,88	2,40
15	4000	0,36	6,85	2,25	2,41
15	8000	0,15	6,37	2,16	2,26

Tabelle 5.18: Tsukuba. Ausgedünntes RANSAC-Verfahren mit unterschiedlichen Gradienten-Schwellwerten t_{grad} und variabler Iterationszahl.

Iterationen ausgelassen wurden, da hier die Laufzeit deutlich höher liegt als bei den anderen Werten t_{grad} . Die berechneten Werte können den Tabellen 5.18 bis 5.21 entnommen werden.

Da für jedes Testbildpaar unterschiedliche Gradienten-Schwellwerte verwendet wurden, welche sich unterschiedlich stark auf die Bilder auswirken, sind die Bildpaare gegeneinander schwer zu vergleichen. Beim Bildpaar *Tsukuba* stellt sich wieder der zuvor wahrgenommene Effekt ein, dass die unterschiedlichen Iterationszahlen keinen Einfluss auf die Werte zu haben scheinen. Es scheint also auch hier, als sei der Schwellwert $t_{Tsukuba}$ für den RANSAC-Algorithmus nicht ideal gewählt und lässt daher keine höhere Genauigkeit bei der Berechnung zu.

Beim *Aerial*- und *House*-Bildpaar hingegen verbessern sich die Distanzwerte für jeden Gradienten-Schwellwert wie erwartet mit höherer Iterationszahl. Sie haben ebenfalls gemeinsam, dass bei zu hoch gewähltem t_{grad} die Distanzwerte deutlich schlechter werden. Wahrscheinlich hat das damit zu tun, dass bei zu geringer Größe des Consensus-Sets auch dessen Aussagekraft abnimmt und es nicht mehr die Mehrheit im gesamten Bildbereich repräsentiert.

Ein Spezialfall ist wiederum das *Javier*-Bildpaar. Hier scheinen sich die Werte, wie auch bei *Tsukuba*, jeweils im gleichen Bereich zu befinden, ohne merkliche Ab- oder Zunahme mit steigender Iterationszahl. Daher stellt sich die Frage, ob der Schwellwert ($t_{Javier} = 2,0$) nicht auch hier ein begrenzender Faktor für die Genauigkeit des RANSAC-Algorithmus ist. Weiterhin ergibt sich beim *Javier*-Bildpaar ebenfalls bei steigendem t_{grad} eine Verschlechterung der Werte.

Wie schlagen sich diese Werte nun im Vergleich mit dem normalen RANSAC-Algorithmus? Das *Tsukuba*-Bildpaar erreicht sehr ähnliche Werte zu vorher, doch mit der Vermutung, dass der Schwellwert $t_{Tsukuba} = 0,5$ keine höhere Genauigkeit zulässt, ließe sich diese Ähnlichkeit erklären, da beide Verfahren an dieselbe Grenze gestoßen sein könnten. Für das *Aerial*-Bildpaar sind beide Verfahren dagegen wahrscheinlich noch nicht an ihre Grenzen gestoßen, wobei mit dem normalen RANSAC-Algorithmus bei der hier aufgewendeten Zeit leicht bessere Werte erreicht werden konnten. Beim

Schwellwert t_{grad}	Ziehungen	Minimum	Maximum	Median	Mittelwert
10	1000	0,37	6,47	2,27	2,34
10	2000	0,29	3,90	2,12	2,10
10	4000	0,30	6,44	1,70	1,77
17	1000	0,57	10,78	2,86	3,29
17	2000	0,24	8,70	2,55	2,66
17	4000	0,39	7,01	2,23	2,51
17	8000	0,31	9,51	1,94	2,18
30	1000	0,73	18,14	5,46	6,43
30	2000	0,71	18,82	4,51	5,58
30	4000	0,52	11,16	3,87	4,44
30	8000	0,50	12,63	3,45	3,99

Tabelle 5.19: Aerial. Ausgedünntes RANSAC-Verfahren mit unterschiedlichen Gradienten-Schwellwerten t_{grad} und variabler Iterationszahl.

Schwellwert t_{grad}	Ziehungen	Minimum	Maximum	Median	Mittelwert
5	1000	1,72	15,88	4,26	4,90
5	2000	0,80	13,44	4,09	4,74
5	4000	0,93	9,61	4,31	4,40
10	1000	0,80	14,04	5,01	5,41
10	2000	1,03	14,10	5,83	5,82
10	4000	1,13	12,62	4,77	5,27
10	8000	1,15	13,56	5,11	5,42
20	1000	2,44	22,25	8,18	9,55
20	2000	1,22	24,72	6,77	8,79
20	4000	1,46	24,24	7,41	8,63
20	8000	1,29	20,98	8,63	8,66

Tabelle 5.20: Javier. Ausgedünntes RANSAC-Verfahren mit unterschiedlichen Gradienten-Schwellwerten t_{grad} und variabler Iterationszahl.

Schwellwert t_{grad}	Ziehungen	Minimum	Maximum	Median	Mittelwert
7	1000	0,82	26,62	7,98	9,21
7	2000	0,91	18,80	7,62	8,56
7	4000	0,87	21,58	7,09	7,81
10	1000	1,01	22,44	9,55	9,85
10	2000	0,64	21,77	7,69	8,49
10	4000	0,99	21,09	7,29	8,02
10	8000	1,20	20,86	6,96	7,30
20	1000	1,38	33,15	12,91	13,79
20	2000	0,91	28,25	10,99	11,16
20	4000	1,56	27,59	10,60	11,29
20	8000	0,58	25,77	9,18	9,53

Tabelle 5.21: House. Ausgedünntes RANSAC-Verfahren mit unterschiedlichen Gradienten-Schwellwerten t_{grad} und variabler Iterationszahl.

Javier-Bildpaar wurden hier jedoch leicht bessere Werte als zuvor erreicht. Möglicherweise liegt der Grund darin, dass der große homogene Hintergrund zuvor einen negativen Einfluss auf die Ergebnisse gehabt haben könnte, was sich durch die Elimination dieser Korrespondenzen mit dem Kantenbild gebessert haben kann. Das *House*-Bildpaar zeigt hingegen auch recht ähnliche Werte zu denen des normalen RANSAC-Algorithmus.

Insgesamt kann man für die Verwendung von Schwellwerten also festhalten, dass der Einfluss einer dem RANSAC-Algorithmus vorhergehenden Kantenfilterung sich in Grenzen hält. Bei Bildpaaren mit großem Ausreißerbereich allerdings kann dies durchaus eine Verbesserung bewirken.

Die Laufzeit ist nun etwas variabler. Sie ist zwar immer noch genauso linear abhängig von der Anzahl an Korrespondenzen und Iterationen, doch ist die Menge der Korrespondenzen unterschiedlich groß und abhängig vom Ausgangsbild sowie dem Schwellwert t_{grad} . Deshalb kommt es beispielsweise vor, dass die Laufzeit bei gleicher Iterationszahl des kleinsten Bildpaares *Tsukuba* die des größeren Bildpaares *Javier* übersteigt. Die größte Laufzeit wurde hier für das Bildpaar *House* mit einem Wert von 9 Minuten und 50 Sekunden gemessen. Die einzelnen Berechnungszeiten lagen daher alle unter 6 Sekunden.

Least Median

Natürlich sollen auch in diesem Fall wieder die gerade berechneten Werte mit denen des Least Median verglichen werden. Dafür wurden die Tabellen 5.22 bis 5.25 aufgestellt. Da wie bereits bekannt die Berechnung des RANSAC-Algorithmus mit dem Least Median länger dauert als die des gewöhnlichen RANSAC-Algorithmus, wurden auch hier wieder weniger Iterationen durchgeführt, nämlich nur bis zu halb so viele wie zuvor.

Schwellwert t_{grad}	Ziehungen	Minimum	Maximum	Median	Mittelwert
5	1000	0,07	4,50	1,25	1,41
5	2000	0,12	3,55	1,05	1,17
10	1000	0,18	4,89	1,33	1,46
10	2000	0,06	3,89	1,22	1,33
10	4000	0,09	3,93	0,95	1,07
15	1000	0,16	3,87	1,46	1,54
15	2000	0,15	3,27	1,15	1,28
15	4000	0,11	4,75	1,07	1,24

Tabelle 5.22: Tsukuba. Ausgedünntes RANSAC-Verfahren unter Verwendung des Least Median mit unterschiedlichen Gradienten-Schwellwerten t_{grad} und variabler Iterationszahl.

Wie auch zuvor schon bemerkt wurde, verhält sich das *Tsukuba*-Bildpaar durch die Verwendung des Least Median wieder wie erwartet und die Distanzwerte verringern sich mit steigender Iterationszahl für alle verschiedenen Gradienten-Schwellwerte t_{grad} . Bemerkenswert ist hier auch, dass zum ersten Mal im Verlauf dieser Arbeit ein Median der hundert Testwerte unter 1,0 liegt und somit Subpixelpräzision erreicht.

Dieses Mal wird das erwartete Verhalten, dass mit höherer Iterationszahl sich auch die Distanzwerte verringern, ebenso von allen anderen Testbildern bestätigt! Das ist bemerkenswert, da es bisher immer ein Problem gab, das einen abweichenden Verlauf der Werte zur Folge hatte. In diesem Fall gibt es aber keine großen homogenen Bildbereiche mehr, die die Menge der Ausreißer größer als das Consensus-Set werden lassen. Auch das Problem eines falsch gewählten Schwellwerts gibt es durch die Verwendung des Least Median nicht.

Daher stellt diese Variante des RANSAC-Algorithmus wohl die robusteste dar, welche zumindest für alle hier gewählten Testbildpaare mit geeigneter Wahl des Gradientenschwellwerts t_{grad} und immer höherer Iterationszahl auch immer bessere Werte für die Faugeras-Distanz zur Folge hat. Aber sind diese Werte auch besser als die der anderen bereits ausgewerteten RANSAC-Verfahren?

Leider ist das nicht für alle der verwendeten Testbilder der Fall. Allerdings werden auch keine deutlich schlechteren Werte, als bei den anderen Verfahren erreicht. Das *Aerial*- und das *House*-Bildpaar etwa erreichen Werte, welche nur wenig schlechter sind als die im normalen RANSAC-Algorithmus. Dafür ergeben sich sowohl für *Tsukuba*- als auch sehr deutlich für das *Javier*-Bildpaar Verbesserungen der Werte.

Mit einem Höchstwert von 30 Minuten und 20 Sekunden lag dafür auch die längste Laufzeit für das *House*-Bildpaar deutlich über der längsten gemessenen Laufzeit ohne Verwendung des Least Median. Die einzelnen Berechnungen dauerten somit bis zu 19 Sekunden.

Schwellwert t_{grad}	Ziehungen	Minimum	Maximum	Median	Mittelwert
10	1000	0,28	5,29	2,18	2,43
10	2000	0,27	5,76	1,85	1,99
17	1000	0,35	6,35	3,05	3,02
17	2000	0,48	7,28	2,53	2,76
17	4000	0,37	8,90	2,04	2,30
30	1000	0,40	20,28	3,94	5,06
30	2000	0,70	11,69	3,98	4,41
30	4000	0,52	12,30	3,23	3,83

Tabelle 5.23: Aerial. Ausgedünntes RANSAC-Verfahren unter Verwendung des Least Median mit unterschiedlichen Gradienten-Schwellwerten t_{grad} und variabler Iterationszahl.

Schwellwert t_{grad}	Ziehungen	Minimum	Maximum	Median	Mittelwert
5	1000	0,89	8,03	3,39	3,55
5	2000	0,90	8,85	2,82	3,29
10	1000	0,71	9,47	3,47	3,81
10	2000	0,65	7,22	3,45	3,42
10	4000	0,54	5,23	2,75	2,84
20	1000	1,66	29,03	8,81	9,72
20	2000	1,90	28,18	8,71	9,11
20	4000	2,71	16,82	8,41	8,76

Tabelle 5.24: Javier. Ausgedünntes RANSAC-Verfahren unter Verwendung des Least Median mit unterschiedlichen Gradienten-Schwellwerten t_{grad} und variabler Iterationszahl.

Schwellwert t_{grad}	Ziehungen	Minimum	Maximum	Median	Mittelwert
7	1000	0,45	26,01	9,08	10,25
7	2000	1,36	18,41	7,71	8,18
10	1000	0,56	24,44	8,47	9,66
10	2000	0,57	24,63	8,23	8,75
10	4000	0,71	23,97	7,39	7,91
20	1000	1,15	34,38	12,47	13,51
20	2000	0,55	37,88	12,09	13,76
20	4000	0,97	30,95	12,21	13,19

Tabelle 5.25: House. Ausgedünntes RANSAC-Verfahren unter Verwendung des Least Median mit unterschiedlichen Gradienten-Schwellwerten t_{grad} und variabler Iterationszahl.

5.4.3 SIFT und RANSAC

Nachdem die Ausdünnung der Korrespondenzen durch Kantenfilterung zusammen mit dem Least Median und dem RANSAC-Algorithmus bereits recht gute Ergebnisse erzielen konnte, wurde als letzte Variante in dieser Arbeit auch noch der Feature-Detektor und -Deskriptor von Lowes SIFT [Low04] eingesetzt. Diese hebt sich von den vorigen Varianten ab, da nun kein Flussfeld mehr bei der Berechnung zum Einsatz kommt, sondern ausschließlich die mit der SIFT berechneten Feature-Matches.

Zu deren Berechnung wurde eine Original-Implementierung von David Lowe verwendet, welche für Forschungszwecke bereitgestellt wird [Low]. Nachdem damit die Key-Points detektiert und beschrieben worden sind, werden deren Matches zwischen Ausgangs- und Zielbild bestimmt. Diese können dann in eine Datei geschrieben werden, welche bei der Verwendung des RANSAC-Algorithmus in der eigenen Implementierung wieder eingelesen werden kann. Die berechneten Feature-Punkte und -Matches sind dabei mit jedem Ausführen des SIFT-Programms gleich, weshalb die geschriebenen Match-Dateien für alle folgenden Berechnungen unverändert bleiben.

Zur besseren Interpretation der Ergebnisdistanzen wurde auch eine Visualisierung der ausgewählten SIFT-Feature-Matches implementiert, welche ein Feedback gibt, ob bei der Ziehung Ausreißer enthalten sind. Im Falle von Abb. 5.5 unten wurde an den Haarspitzen von *Javier* eine falsche Zuordnung von Feature-Punkten durchgeführt. Dies lässt daher eine größere Faugeras-Distanz der hier berechneten Fundamentalmatrix zur wahren Matrix erwarten. Bei den Matches des *House*-Bildpaares oben hingegen wurden die Feature-Punkte einwandfrei zugeordnet und hier ist deshalb eine geringe Distanz zu erwarten.

In den Tabellen 5.26 bis 5.29 wurden nun die Werte festgehalten, welche unter Verwendung von RANSAC und SIFT, mit Schwellwert oder Least Median und bei Iterationen von 1000, 10000 und 100000 Ziehungen berechnet wurden. Die Testbildpaare weisen hier sehr unterschiedliche Verhaltensweisen auf. Das *Tsukuba*-Bildpaar zeigt wieder einen Unterschied zwischen der Verwendung des Schwellwerts und des Least Medians. Während letzterer mit höherer Iterationszahl erwartungsgemäß kleinere Distanzwerte ausgibt, lässt sich bei Verwendung des Schwellwerts wieder keine Aussage über einen Verlauf der Werte treffen. Das *Aerial*- und das *House*-Bildpaar zeigen hingegen für alle berechneten Werte den erwarteten Verlauf. Die Distanzwerte nehmen also immer mit steigender Iterationszahl ab. Das *Javier*-Bildpaar jedoch zeigt komplett einen willkürlichen Verlauf der berechneten Werte. Noch dazu sind diese gerade unter der Verwendung des Least Median sehr groß.

Im vorigen Kapitel wurde bei zu großer Wahl des Gradienten-Schwellwerts t_{grad} und somit großer Ausdünnung der Korrespondenzen eine Verschlechterung der berechneten Distanzwerte beobachtet. Betrachtet man die Verwendung der SIFT-Matches als noch größere Ausdünnung der Korrespondenzen, so macht es Sinn, dass dieser Ansatz etwas größere Distanzen ausgibt als bei geeigneter Wahl von t_{grad} und der Verwendung des Ansatzes aus dem vorhergehenden Abschnitt. Denn die unterstützenden Consensus-Sets für die einzelnen Ziehungen von Korrespondenzen werden immer kleiner und verlieren somit an Aussagekraft.

Allerdings sind die sehr hohen Werte für das *Javier*-Bildpaar, gerade bei der Verwendung des Least Median, doch etwas überraschend. Jedoch wurde auch hier für den Least Median zumeist ein Distanzwert

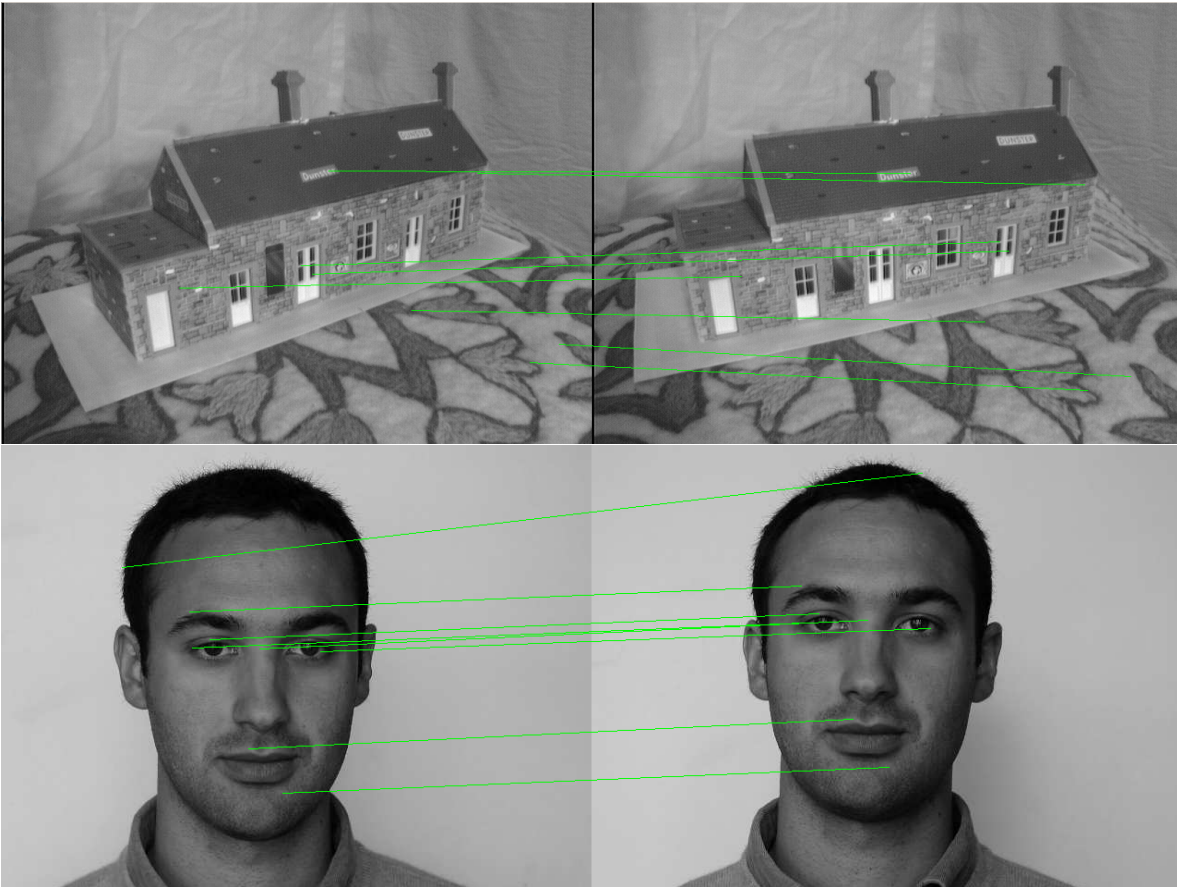


Abbildung 5.5: Mit RANSAC ausgewählte SIFT-Matches. **Oben:** Matches für das *House*-Bildpaar.
Unten: Matches für das *Javier*-Bildpaar mit enthaltenem Outlier.

deutlich kleiner als 1 erreicht. Trotzdem werden aber offensichtlich durch schlechte Feature-Matches schlechte Ergebnisse erzielt.

Die Verwendung von SIFT ist in Kombination mit dem RANSAC-Algorithmus also nicht zu empfehlen, da die gewonnenen Werte schlechter sind als die der anderen RANSAC-Verfahren.

Für die Laufzeit jedoch bringt das Verfahren eine deutliche Verbesserung gegenüber den anderen Verfahren, da der RANSAC-Algorithmus über deutlich weniger Korrespondenzen ausgeführt werden muss, was die Verwendung einer viel höheren Iterationszahl ermöglicht. Für das *Tsukuba*-Bildpaar mit 100000 Iterationen wurde ohne Verwendung des Least Median eine höchste Laufzeit von 4 Minuten und 31 Sekunden gemessen, mit Least Median erreichte sie 8 Minuten und 42 Sekunden. Die einzelnen Berechnungen unter Verwendung des Schwellwerts lagen daher unter 3 Sekunden und mit Verwendung des Least Median unter 6 Sekunden.

Least Median	Ziehungen	Minimum	Maximum	Median	Mittelwert
Nein	1000	0,43	14,44	4,32	4,65
Nein	10000	0,86	12,47	4,85	5,39
Nein	100000	1,47	11,40	5,88	5,92
Ja	1000	0,33	14,04	2,78	3,34
Ja	10000	0,30	6,37	2,01	2,13
Ja	100000	0,27	3,82	1,53	1,71

Tabelle 5.26: Tsukuba. RANSAC-Algorithmus über SIFT-Featurepaaren mit variabler Iterationszahl sowie mit und ohne Least Median.

Least Median	Ziehungen	Minimum	Maximum	Median	Mittelwert
Nein	1000	0,48	16,70	4,14	5,35
Nein	10000	0,58	7,57	3,10	3,26
Nein	100000	0,45	6,04	2,85	2,91
Ja	1000	0,59	15,89	4,42	5,38
Ja	10000	0,52	11,53	3,35	3,65
Ja	100000	0,42	12,33	2,61	2,92

Tabelle 5.27: Aerial. RANSAC-Algorithmus über SIFT-Featurepaaren mit variabler Iterationszahl sowie mit und ohne Least Median.

Least Median	Ziehungen	Minimum	Maximum	Median	Mittelwert
Nein	1000	3,75	94,46	21,01	24,78
Nein	10000	3,25	52,88	29,63	24,49
Nein	100000	6,53	34,51	30,06	26,48
Ja	1000	10,67	118,23	52,45	57,66
Ja	10000	12,40	115,66	58,87	71,50
Ja	100000	29,48	116,84	85,54	76,88

Tabelle 5.28: Javier. RANSAC-Algorithmus über SIFT-Featurepaaren mit variabler Iterationszahl sowie mit und ohne Least Median.

Least Median	Ziehungen	Minimum	Maximum	Median	Mittelwert
Nein	1000	0,97	26,27	11,21	10,90
Nein	10000	1,37	16,54	9,16	8,87
Nein	100000	1,77	17,98	8,63	8,49
Ja	1000	0,75	50,46	10,49	12,86
Ja	10000	0,89	33,93	9,73	9,72
Ja	100000	1,17	17,29	6,44	6,91

Tabelle 5.29: House. RANSAC-Algorithmus über SIFT-Featurepaaren mit variabler Iterationszahl sowie mit und ohne Least Median.

5.5 Dichter Ansatz vs. RANSAC

Nachdem nun sowohl Verfahren mit der Verwendung dichter Korrespondenzen als auch RANSAC-basierte Verfahren, welche genau acht Punktkorrespondenzen für die Berechnung der Fundamentalmatrix verwenden, analysiert wurden, stellt sich die Frage, welcher der beiden Ansätze die besseren Ergebnisse liefert.

Der beste Ansatz der dichten Verfahren ist der iterative M-Schätzer-Ansatz, welcher auch eine Normalisierung der Ausgangs- und Ziel-Punktewolken vornimmt (siehe Abschnitt 5.3.3). Dagegen steht der ausgedünnte RANSAC-Algorithmus unter Verwendung des Least Median, welcher aufgrund seiner größeren Robustheit gegenüber dem normalen RANSAC-Algorithmus zu überzeugen weiß (siehe Abschnitt 5.4.2). Die besten Werte, die mit diesen Verfahren erreicht wurden, sind in Tabelle 5.30 aufgelistet.

Der erste Unterschied der beiden Verfahren ist bereits die Tatsache, dass beim dichten Verfahren für jedes Flussfeld ein Konvergenzwert berechnet wird, welcher bei jeder wiederholten Berechnung wieder erreicht wird. Das Verfahren liefert also einen eindeutigen Wert. Der ausgedünnte RANSAC-Algorithmus hingegen berechnet mit jeder Ausführung unterschiedliche Werte, weshalb auch hier für die berechneten Distanzen Minimum, Maximum, Median und Mittelwert angegeben werden. Daher kann dieses Verfahren gegenüber dem dichten Verfahren zwar für alle Testbildpaare deutlich kleinere Distanzen berechnen, aber auch deutlich größere. Deshalb wird besser wieder der Median und Mittelwert des RANSAC-Verfahrens zum Vergleich herangezogen.

Es zeigt sich, dass der iterative Ansatz für die beiden Bildpaare *Tsukuba* und *Aerial* mit Subpixelpräzision um über den Faktor 2 bessere Werte als der ausgedünnte RANSAC-Algorithmus erreicht. Für die beiden anderen Bildpaare *Javier* und *House* jedoch erreicht das RANSAC-Verfahren die etwas besseren Werte.

Bei genauerer Überlegung, für welche Bildpaare der RANSAC-Algorithmus die besseren Werte liefert, erkennt man, dass dies genau die beiden Stereopaare sind, welche einen größeren, homogenen Hintergrund aufweisen, wodurch sich vermutlich viele Ausreißer ergeben. Hierdurch hat die dichte Berechnung des iterativen Ansatzes ihre Probleme, da sie immer alle Punktkorrespondenzen bei

Dichtes Verfahren	Tsukuba	Aerial	Javier	House
M-Schätzer	0,39	0,82	3,17	9,23
RANSAC ausgedünnt	Tsukuba	Aerial	Javier	House
Minimum	0,086	0,27	0,54	0,71
Maximum	3,93	5,76	5,23	23,97
Median	0,95	1,85	2,75	7,39
Mittelwert	1,07	1,99	2,84	7,91

Tabelle 5.30: Dichte Korrespondenzen vs. Vorgefilterter RANSAC. Gegenüberstellung der besten Werte des iterativen M-Schätzer-Ansatzes und des ausgedünnten RANSAC-Verfahrens für alle Testbildpaare.

der Berechnung verwendet. Für die anderen beiden Bildpaare hingegen ergibt sich ein sehr genaues Flussfeld mit wenigen Ausreißern, wodurch sehr gute Werte erreicht werden.

Weiterhin ist zu überlegen, ob das dichte Verfahren nicht auch bei ungünstigeren Flussfeldern dem ausgedünnten RANSAC vorgezogen werden soll, da dieser immer das Risiko birgt, eine sehr große Distanz zu erhalten. Das dichte Verfahren hingegen bietet durch seine Konvergenz Verlässlichkeit für seine gelieferten Werte. Diese sollte auch Priorität haben.

Ein weiterer Vorteil des dichten Verfahrens liegt in der sekundenschnellen Berechnung der Fundamentalmatrix, während der RANSAC-Algorithmus für die Produktion guter Werte deutlich länger braucht.

5.6 Qualität des Ausgangsflussfeldes

Wie nun des Öfteren erkennbar war, bieten die verschiedenen Bildpaare eine unterschiedliche, erreichte Qualität der geschätzten Fundamentalmatrix. Das liegt daran, dass bereits der geschätzte optische Fluss nicht immer die gleiche Qualität hat. Wie groß diese Unterschiede zwischen gut und weniger gut geschätzten Flussfeldern in der Praxis sind, soll daher in diesem Abschnitt noch einmal anhand des *Javier*-Bildpaares verdeutlicht werden.

Es werden nun 3 verschiedene Flussfelder für das Stereopaar miteinander verglichen. Deren Visualisierungen können in Abb. 5.6 betrachtet werden. Das mittlere davon ist das auch in dieser Arbeit evaluierte Flussfeld, das mit dem Demetz-Verfahren und einem α -Wert von 9000 berechnet wurde. Das linke wurde mit dem gleichen Verfahren und einem α -Wert von 3000 berechnet, während das letzte nach der Methode von Brox *et al.* [BBPW04] berechnet wurde.

Es sollen hier nur deren Werte für die dichten Verfahren ohne und mit Normalisierung sowie des iterativen M-Schätzer-Ansatzes mit Normalisierung verglichen werden. Diese sind in Tabelle 5.31 zu finden.

Sieht man sich die drei Flussfelder in Abb. 5.6 an, so erkennt man zunächst, dass das erste Feld einige Aktivitäten im Hintergrund zeigt. Deshalb erreicht dieses Flussfeld auch die höchsten Distanzwerte für



Abbildung 5.6: *Javier* Flussfelder. **Links:** Demetz *et al.*, $\alpha = 3000$. **Mitte:** Demetz *et al.*, $\alpha = 9000$. **Rechts:** Brox *et al.*

Verfahren	Demetz α 3000	Demetz α 9000	Brox
Ohne Normalisierung	96,15	6,67	11,60
Mit Normalisierung	23,87	3,18	0,88
M-Schätzer m.N.	10,85	3,17	0.86

Tabelle 5.31: Distanzen zu verschiedenen Flussfeldern des Bildpaares *Javier*.

dessen berechnete Fundamentalmatrizen. Um dies zu verbessern, kann man den Parameter α erhöhen, welcher den Einfluss des Glattheitsterms im Energiefunktional der optischen Flussberechnung steuert. Setzt man α also von 3000 auf 9000 hoch, so erhält man aus dem linken Flussfeld das mittlere Flussfeld. Hier hat der Hintergrund nun einen gleichmäßigen optischen Fluss, was sofort eine starke Verbesserung der Distanz der berechneten Fundamentalmatrix zur Folge hat. Das rechte Flussfeld schließlich zeigt ein noch besseres Flussfeld, das nach der Methode von Brox *et al.* berechnet wurde und einen viel glatteren Verlauf des Flussfeldes auch im Gesichtsbereich aufweist. Dagegen zeigt das mittlere Flussfeld noch sichtbare Treppeneffekte. Hiermit erhält man beim Einsatz der Normalisierung in der Fundamentalmatrix-Berechnung sogar wieder eine Distanz in Subpixelgenauigkeit.

Es zeigt sich also, dass schon kleine Abweichungen in der Qualität des Ausgangsflussfeldes sehr wohl einen großen Einfluss auf die Qualität der Fundamentalmatrix-Schätzung haben. Ebenso scheint es ratsam zu sein, eher etwas zu glatte Flussfelder zu verwenden.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

Im Rahmen dieser Diplomarbeit wurden im Hinblick auf das Ziel, eine Verbesserung der Schätzung des optischen Flussfeldes für den Spezialfall von Stereo-Bildpaaren zu erreichen und hierfür eine möglichst genaue Fundamentalmatrix bereitzustellen, verschiedene Verfahren zur Schätzung der Fundamentalmatrix vorgestellt, implementiert und qualitativ verglichen.

Diese kann grundlegend entweder mit dem 7-Punkt-Algorithmus berechnet werden, welcher allerdings keine eindeutige Lösung liefert oder mit dem 8-Punkt-Algorithmus. Nach der Entscheidung für den 8-Punkt-Algorithmus aufgrund seiner Eindeutigkeit wurde dann eine grobe Unterteilung der möglichen Verfahren getroffen. Einerseits in dichte Verfahren, welche auf Basis eines Ausgangsflussfeldes alle darin zur Verfügung stehenden Punktkorrespondenzen verwenden, um die Fundamentalmatrix zu berechnen und andererseits in spärliche Verfahren, welche genau 8 Punktkorrespondenzen verwenden, welche mithilfe des RANSAC-Algorithmus aus dem Flussfeld ausgesucht werden.

Für die dichten Verfahren konnten hier einige Verbesserungen implementiert werden. Durch den Einsatz von M-Schätzern wurde eine größere Robustheit des Verfahrens gegenüber Ausreißern erreicht. Dadurch entstand allerdings ein nicht-lineares Gleichungssystem, was mithilfe eines iterativen Ansatzes gelöst werden musste. Weiterhin wurde durch das Durchführen einer Normalisierung der Ausgangs- und Ziel-Punktewolken der unterschiedlich starke Einfluss der Koordinatenwerte im Energiefunktional ausgeglichen. Dies bedeutete in allen Testläufen eine stark verbesserte Qualität der geschätzten Fundamentalmatrix. Ebenso wurde eine Singulärwertzerlegung implementiert, welche nachträglich den Rang 2 der Fundamentalmatrix gewährleisten kann. Deren Einfluss auf die Ergebnisse war allerdings sehr gering.

Für die spärlichen Verfahren stellt der RANSAC-Algorithmus bereits ein robustes Verfahren dar. Auch hierfür wurden noch Variationen implementiert, wie der Einsatz des Least Median anstatt eines Schwellwerts zur Bestimmung des besten Consensus-Sets. Dieser machte die prekäre Wahl eines guten Schwellwerts, welcher die mögliche Qualität der Fundamentalmatrix stark beeinflusst, überflüssig, kann jedoch bei zu großen Ausreißermengen von über 50 % nur noch zufällige Ergebnisse erreichen, weil der Median in den Ausreißer-Bereich fällt. Ebenfalls wurde vor dem Ausführen des RANSAC-Algorithmus eine Ausdünnung der Korrespondenzen durch einen Kantenfilter ausprobiert, was gerade in Kombination mit dem Least Median gute Werte erzielen konnte. Probleme, die vorher bei zu großen Ausreißermengen oder schlecht gewählten Schwellwerten lagen, traten bei dieser Kombination nicht mehr auf. Auch die Verwendung von SIFT-Feature-Matches wurde in Kombination mit dem RANSAC-Algorithmus getestet. Allerdings konnten dessen Ergebnisse nicht überzeugen.

Für alle verwendeten Verfahren wurde mit der Faugeras-Distanz ein sinnvolles, in der Epipolargeometrie Bedeutung tragendes, Fehlermaß für den Vergleich von der geschätzten zur wahren Fundamentalmatrix gefunden. Dies ermöglichte eine Gegenüberstellung der Ansätze, wobei sich bei den dichten Verfahren der iterativ neugewichtete M-Schätzer-Ansatz mit Normalisierung durchsetzen konnte, während das beste RANSAC-Verfahren das mit vorgefilterten Korrespondenzen war.

Letztendlich konnte das dichte Verfahren überzeugen, da es für entsprechend geeignete Flussfelder verlässliche Werte produziert, welche sogar Subpixeldistanz erreichen und auch eine geringere Laufzeit bietet als der RANSAC-Algorithmus, der deutlich länger laufen muss, um einigermaßen ausreißerfreie Werte zu liefern.

Schließlich wurde noch die große Bedeutung der genauen Schätzung des Ausgangsflussfeldes dargestellt, indem hier für das gleiche Bildpaar verschiedene gute Flussfelder verglichen wurden. Für die Faugeras-Distanz ergaben sich mittels eines für ein Bildpaar optimierten dichten Verfahrens Werte, die vom zweistelligen Bereich bis in den Subpixelbereich variieren.

6.2 Ausblick

Im Folgenden sollen noch weiterführende Ideen diskutiert werden, welche in zukünftigen Arbeiten verfolgt werden können.

Beispielsweise stellt sich die Frage, ob es wirklich nötig ist, sich zwischen den dichten und spärlichen Ansätzen zur Fundamentalmatrix-Bestimmung zu entscheiden. Vielmehr wäre es doch wünschenswert, wenn man die Stärken der beiden kombinieren könnte. Hierfür zwei Anregungen:

Einerseits wurde festgestellt, dass der dichte Ansatz nicht so genaue Ergebnisse produzieren kann, wenn viele starke Ausreißer in den Korrespondenzen enthalten sind. Beim RANSAC-Algorithmus wurde bereits eine vorherige Kantenfilterung vorgenommen, um gerade diese Korrespondenzen auszuschließen. Wieso sollte dies nicht auch dem dichten Ansatz weiterhelfen und dieser nur über den entsprechend gefilterten Korrespondenzen, die durch die Kantenpixel gegeben sind, ausgeführt werden? Wie schon zuvor in dieser Arbeit erwähnt wurde, könnte sich hier statt einem Gradienten-Filter auch ein Ecken-Filter besser eignen, welcher alle Pixel auf den kleineren Eigenwert des Struktur-tensors untersucht. Für diese Eckpunkte sollte das optische Flussfeld nämlich noch genauere Werte enthalten.

Andererseits bietet sich eine Erweiterung des RANSAC-Algorithmus an, der *Locally Optimized RANSAC* (kurz: *LO-RANSAC*). Für die 8 ausgewählten Punktkorrespondenzen wird schließlich immer ein Consensus-Set gebildet, für dessen Korrespondenzen der geometrische Abstand kleiner einem Schwellwert ist. Diese Korrespondenzen sind daher auch im Sinne der berechneten Fundamentalmatrix ausreißerfrei und durch Verwendung des ganzen Consensus-Sets kann die Fundamentalmatrix neu und genauer bestimmt werden. Hierfür bieten sich natürlich die behandelten dichten Methoden an, wobei für die erste Iteration eines iterativ neugewichteten M-Schätzer-Ansatzes die zu diesem Consensus-Set führende Fundamentalmatrix verwendet werden könnte.

Leider konnte im Rahmen dieser Arbeit keine Untersuchung mehr angestellt werden, ob mithilfe der berechneten Fundamentalmatrix eine verbesserte Schätzung des optischen Flusses bei Stereobildpaaren erreicht werden kann. Da mit bekannter Fundamentalmatrix der Suchraum für Korrespondenzpunkte vom gesamten Bildbereich auf eine Linie, also eine ganze Dimension weniger, begrenzt wird, sollte dies durchaus interessant sein. Mithilfe eines Benchmarks wie KITTI, bei dem sich die Testbilder hauptsächlich durch die Fortbewegung des Autos, und somit auch der Kamera die im Auto angebracht ist, unterscheiden, wäre die Untersuchung dieses erweiterten Ansatzes sicher gut möglich. Es wäre empfehlenswert, ein bereits sehr gutes Verfahren in einem solchen Benchmark als Grundlage zu nehmen und dann den iterativen M-Schätzer-Ansatz zu verwenden, welcher dann sehr gute Werte bis in die Subpixelpräzision für die Distanz der errechneten Fundamentalmatrix erreichen sollte. Da dies auch eine Motivation für diese Arbeit dargestellt hat, wäre eine solche Erweiterung eines geeigneten Variationsansatzes zur Schätzung des optischen Flusses wünschenswert.

Literaturverzeichnis

- [BA91] M. Black, P. Anandan. Robust dynamic motion estimation over time. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, S. 296–302. 1991. (Zitiert auf Seite 17)
- [BBPW04] T. Brox, A. Bruhn, N. Papenberg, J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. European Conference on Computer Vision*, S. 25–36. 2004. (Zitiert auf den Seiten 21, 37 und 82)
- [BETVG08] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. (Zitiert auf den Seiten 31 und 37)
- [BL03] M. Brown, D. Lowe. Recognising panoramas. In *Proc. IEEE International Conference on Computer Vision*, S. 1218–1225. 2003. (Zitiert auf den Seiten 30 und 37)
- [Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. (Zitiert auf Seite 59)
- [Bru12a] A. Bruhn. Vorlesungsskript Computer Vision, Wintersemester 2012/2013. Institut für Visualisierung und Interaktive Systeme, Universität Stuttgart, Germany, 2012. (Zitiert auf den Seiten 11, 28, 32, 34 und 35)
- [Bru12b] A. Bruhn. Vorlesungsskript Correspondence Problems in Computer Vision, Sommersemester 2012. Institut für Visualisierung und Interaktive Systeme, Universität Stuttgart, Germany, 2012. (Zitiert auf Seite 43)
- [Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. (Zitiert auf Seite 31)
- [CBFAB94] P. Charbonnier, L. Blanc-Feraud, G. Aubert, M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proc. IEEE International Conference on Image Processing*, S. 168–172. 1994. (Zitiert auf Seite 20)
- [CK83] N. H. Cornelius, T. Kanade. Adapting optical-flow to measure object motion in reflectance and x-ray image sequences. In *Proc. ACM SIGGRAPH/SIGART Workshop on Motion: Representation and Perception*, S. 50–58. 1983. (Zitiert auf Seite 18)
- [DSV⁺14] O. Demetz, M. Stoll, S. Volz, J. Weickert, A. Bruhn. Learning brightness transfer functions for the joint recovery of illumination changes and optical flow. In *Proc. European Conference on Computer Vision*, S. 455–471. 2014. (Zitiert auf den Seiten 8, 17, 18, 21, 37 und 59)

- [FB81] M. A. Fischler, R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. (Zitiert auf Seite 46)
- [FG87] W. Förstner, E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*. 1987. (Zitiert auf Seite 13)
- [FLP01] O. D. Faugeras, Q.-T. Luong, T. Papadopoulo. *The geometry of multiple images*. MIT Press, 2001. (Zitiert auf den Seiten 8, 37 und 56)
- [GL06] I. Gordon, D. Lowe. What and where: 3D object recognition with accurate pose. In J. Ponce, M. Hebert, C. Schmid, A. Zisserman, Herausgeber, *Toward Category-Level Object Recognition*, Band 4170 von *Lecture Notes in Computer Science*, S. 67–82. Springer, 2006. (Zitiert auf Seite 31)
- [GLU12] A. Geiger, P. Lenz, R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2012. (Zitiert auf den Seiten 18 und 20)
- [GN04] M. Grossberg, S. Nayar. Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, 2004. (Zitiert auf Seite 18)
- [Har97] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997. (Zitiert auf den Seiten 37 und 44)
- [HN94] T. S. Huang, A. N. Netravali. Motion and structure from feature correspondences: a review. *Proceedings of the IEEE*, 82:252–268, 1994. (Zitiert auf den Seiten 37 und 39)
- [HS81] B. K. P. Horn, B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981. (Zitiert auf den Seiten 12 und 14)
- [HS88] C. Harris, M. Stephens. A combined corner and edge detector. In *Proc. Alvey Vision Conference*, S. 147–151. 1988. (Zitiert auf Seite 31)
- [Iij59] T. Iijima. Basic theory of pattern observation. In *Papers of Technical Group on Automata and Automatic Control. IECE, Japan*, 1959. (Zitiert auf Seite 23)
- [Iij62] T. Iijima. Observation theory of two-dimensional visual patterns. In *Papers of Technical Group on Automata and Automatic Control. IECE, Japan*, 1962. (Zitiert auf Seite 23)
- [Lin94a] T. Lindeberg. Scale-space theory: a basic tool for analysing structures at different scales. *Journal of Applied Statistics*, S. 224–270, 1994. (Zitiert auf Seite 24)
- [Lin94b] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA, 1994. (Zitiert auf Seite 23)
- [LK81] B. D. Lucas, T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. International Joint Conference on Artificial Intelligence*, S. 674–679. 1981. (Zitiert auf den Seiten 12 und 13)

- [Lon81] Longuet. A computer algorithm for reconstructing a scene from two projections. *Nature*, S. 133–135, 1981. (Zitiert auf den Seiten 37 und 40)
- [Low] D. G. Lowe. Demo Software: SIFT Keypoint Detector. <http://www.cs.ubc.ca/~lowe/keypoints/>. (Zitiert auf Seite 78)
- [Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE International Conference on Computer Vision*, S. 1150–1157. 1999. (Zitiert auf Seite 30)
- [Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. (Zitiert auf den Seiten 22, 24, 25, 26, 27, 30, 31, 37 und 78)
- [MBW08] M. Mainberger, A. Bruhn, J. Weickert. Is dense optic flow useful to compute the fundamental matrix? In *Proc. International Conference of Image Analysis and Recognition*, S. 630–639. 2008. (Zitiert auf Seite 37)
- [MS05] K. Mikolajczyk, C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. (Zitiert auf Seite 31)
- [PM90] P. Perona, J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990. (Zitiert auf Seite 20)
- [RRKB11] E. Rublee, V. Rabaud, K. Konolige, G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proc. IEEE International Conference on Computer Vision*. 2011. (Zitiert auf Seite 31)
- [ST94] J. Shi, C. Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, S. 593 – 600. 1994. (Zitiert auf Seite 31)
- [TPM06] O. Tuzel, F. Porikli, P. Meer. Region Covariance: A Fast Descriptor for Detection And Classification. In *Proc. European Conference on Computer Vision*, S. 589–600. 2006. (Zitiert auf Seite 31)
- [VBMW12] L. Valgaerts, A. Bruhn, M. Mainberger, J. Weickert. Dense versus sparse approaches for estimating the fundamental matrix. *International Journal of Computer Vision*, 96(2):212–234, 2012. (Zitiert auf den Seiten 7 und 37)
- [Vis] Visual Geometry Group Home Page, University of Oxford. <http://www.robots.ox.ac.uk/~vgg/data1.html>. (Zitiert auf den Seiten 55 und 59)
- [WII97] J. Weickert, S. Ishikawa, A. Imiya. *Scale-Space has been discovered in Japan*. Datalogisk Institut København: DIKU-Rapport. Datalogisk Institut, Københavns Universitet, 1997. (Zitiert auf Seite 23)
- [Wik] RANSAC-Algorithmus. <http://de.wikipedia.org/wiki/RANSAC-Algorithmus>. (Zitiert auf Seite 47)
- [ZBW11] H. Zimmer, A. Bruhn, J. Weickert. Optic flow in harmony. *International Journal of Computer Vision*, 93(3):368–388, 2011. (Zitiert auf Seite 20)

Literaturverzeichnis

- [ZDFL95] Z. Zhang, R. Deriche, O. Faugeras, Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87 – 119, 1995. (Zitiert auf Seite 37)
- [Zha98] Z. Zhang. Determining the epipolar geometry and its uncertainty: a review. *International Journal of Computer Vision*, 27(2):161–195, 1998. (Zitiert auf den Seiten 37 und 39)

Alle URLs wurden zuletzt am 19. 05. 2015 geprüft.

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