



Mathematisch-Naturwissenschaftliche Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Computer Vision and Pattern Recognition Group

Diplomarbeit

Vorgelegt an der

Rheinischen Friedrich-Wilhelms-Universität Bonn
im Fachbereich **Informatik**
zum Abschluss eines Studiums
im Studiengang **Informatik**

Detektion stationärer Hindernisse in monokularen Bildsequenzen

Andreas Wedel

Angefertigt bei: DaimlerChrysler AG
Abteilung REI/AI

Betreuer: Dr. Uwe Franke

Prüfer: Prof. Dr. Daniel Cremers
PD Dr. Volker Steinhage

Abgabetermin: 15. April 2006

Zusammenfassung

In der vorliegenden Arbeit wird die Detektion von Hindernissen vor Fahrzeugen untersucht. Auf Hindernisdetektion aufbauende Fahrerassistenzsysteme reichen von einer Warnung vor potentiellen Gefahren bis hin zum aktiven Eingreifen in das Fahrverhalten.

Derzeit werden hauptsächlich Stereokamerasysteme oder aktive Systeme wie Radar und Lidar zur Hindernisdetektion genutzt. Solche Systeme müssen kostenspielig kalibriert und synchronisiert werden und generieren ihre Ergebnisse entweder aus den Daten zweier passiver visueller Sensoren oder je einer passiven Empfangs- und aktiven Sendeeinheit. Bisher ist die Untersuchung der Fahrzeugumgebung und das robuste Wahrnehmen von Hindernissen mit einem Sensor nicht möglich. Trotz der Nutzung zweier Sensoren generieren die genannten Systeme sensorspezifische Fehlhypothesen, welche es zu verwerfen gilt.

In dieser Arbeit wird ein neuer Ansatz für die Hindernisdetektion in monokularen Bildsequenzen vorgestellt, der es ermöglicht Hindernisse im Bild über längere Zeiträume zu verfolgen und daraus Rückschlüsse auf Position und Größe zu erhalten. Zur Verfolgung der Hindernisse wird der von Hager entworfene Region Tracker Ansatz genutzt [35]. Es wird gezeigt, dass in stationären Umgebungen potentielle Hindernisse im Fahrkorridor durch Ausnutzung des Skalierungsfaktors detektiert werden können. Das entwickelte System erlaubt nicht alleine eine alternative Hindernisdetektion, sondern ermöglicht die Verifizierung von Hindernishypothesen, indem die vorhandene Hindernishypothese mit der Gegenhypothese *kein Hindernis* verglichen wird.

Das vorgestellte System wird mathematisch untersucht und mit den Daten anderer Systeme verglichen. Die Funktionalität des Systems wird mit realen Bildsequenzen bestätigt. Experimentelle Ergebnisse zeigen, dass Hindernisse bereits in 50m Entfernung robust detektiert werden. Die frühzeitige Wahrnehmung und sichere Verifizierung von Hindernissen ermöglicht ein rechtzeitiges Einschreiten des Fahrerassistenzsystems.

Abstract

This work deals with obstacle detection in traffic scenes. Applications based on the detection of obstacles in the road course ahead range from potential risk warnings to active vehicle control intervention.

Most obstacle detection systems use stereo camera systems or active sensors such as radar or lidar. The results are generated by two passive visual sensors or by an active send unit and a passive receive unit. Such systems have to be calibrated and synchronized. Up to now environment recognition and robust obstacle detection using one sensor is not fully understood and hence not yet possible. Even though using two sensors the mentioned systems lack due to false obstacle hypotheses, which have to be eliminated.

In this work a novel approach for obstacle detection in monocular image sequences is presented. Obstacles in image space are tracked throughout the sequence and conclusions about object size and location are drawn. The tracking process is based on a region tracking approach proposed by Hager et al. in [35]. Potential obstacles in the road course ahead are recognized by analyzing the scaling factor of image regions. The proposed approach not only allows an alternative obstacle detection but also motivates the verification of obstacle hypothesis by comparing them with the counter hypothesis of a free driveway.

The described basic approach is mathematically analyzed and compared to data from alternative systems. Results on real image sequences show the functionality of the proposed system. Experiments verify the robust detection of obstacles in distances up to 50m. Early detection and fail-safe verification of hazards creates a chance for driver assistance intervention in good time.

Danksagung

An dieser Stelle möchte ich mich herzlichst bei allen Mitarbeitern der Abteilung REI/AI für die tolle Zusammenarbeit, den intensiven Erfahrungsaustausch und die fachlichen Diskussionen bedanken, welche eine große Hilfe und sehr positive Erfahrung für mich waren. Besonders hervorheben möchte ich an dieser Stelle Heidi Loose, Clemens Rabe, Christoph Volmer, Dr. Stefan Gehrig, Dr. Hernan Badino, Kai Dornberger und Heiko Folkerts.

Prof. Dr. Daniel Cremers danke ich herzlich für Übernahme der Begutachtung und für die stetige Förderung der Arbeit. Für die Übernahme der Mitbegutachtung bedanke ich mich sehr herzlich bei PD Dr. Volker Steinhage. Seine Unterstützung waren mir und der Arbeit sehr hilfreich.

Besonders danken möchte ich meinen Betreuern Dr. Uwe Franke und Jens Klappstein für das in mich gesetzte Vertrauen. Sie ermöglichten mir den Einstieg in dieses interessante Themengebiet und standen mir stets mit Rat zur Seite. Auch Dr. Thomas Brox danke ich für die Anregungen zu dieser Arbeit.

Der herzlichste Dank gilt meiner ganzen Familie, die mir immer hilfsbereit beistand und mich im Gebet unterstützt hat. Darüberhinaus bin ich meiner Schwester Angelika Wedel sowie Klaudija Beutel, Frank Klughard, Dennis Wegener, Albert Rempel und Martin Janzen für die Durchsicht des Manuskripts dankbar.

Andreas Wedel

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	3
1.3	Aufbau der Arbeit	3
2	Stand der Forschung	4
2.1	Autonomes Fahren weltweit	4
2.2	Sensorik zur Objekterkennung im Fahrzeugumfeld	5
2.2.1	Diskussion der Erkenntnisse	6
2.3	Die zwei Schritte der Objekterkennung	7
2.3.1	Hypothesengenerierung	8
2.3.2	Hypothesenverifizierung	17
2.4	Forschungsbereich	19
2.4.1	Herausforderungen	20
3	Grundlagen	21
3.1	Kameramodell	21
3.2	Koordinatensysteme	22
3.2.1	Beziehung zwischen Welt- und Kamerakoordinatensystem	23
3.2.2	Beziehung zwischen Kamera- und Sensorkoordinatensystem	23
3.2.3	Beziehung zwischen Sensor- und Bildkoordinatensystem	24
3.2.4	Projektionsgleichung	24
3.2.5	Rektifizierung	25
3.3	Szenenmodell und Äußere Orientierung der Kamera	25
3.4	Structure from Motion	26
3.4.1	Herleitung des Bewegungsfeldes	26
3.4.2	Bewegungsfeld und optisches Flussfeld	29
3.5	Region Tracking	30
3.5.1	Überblick über vorhandene Verfahren zur Merkmalsverfolgung	30
3.5.2	Mathematische Herleitung des Region Tracking	36

3.5.3	Konvergenz und Implementation	38
3.5.4	Modellierung der Helligkeitsunterschiede	39
3.5.5	Modellierung von Verdeckungen	41
3.5.6	Trackergebnis	44
3.5.7	Neuinitialisieren eines Tracks	45
3.6	Orthophoto und inverse Perspektive	46
3.6.1	Sicht auf die Straße	47
3.6.2	Sicht auf stehendes Hindernis	47
4	Hypothesengenerierung	49
4.1	Konzept	49
4.2	Mathematische Herleitung	50
4.2.1	Bewegungsfeld als Anfangswertproblem	50
4.2.2	Bewegungsfeld aus reiner Translation	53
4.2.3	Kompensierung der Rotationsparameter	53
4.2.4	Übertragung der Bewegung des Versuchsträgers auf die Kamera	55
4.3	Implementierung	56
5	Hypothesenverifizierung	59
5.1	Konzept	59
5.2	Mathematische Herleitung	60
5.2.1	Planar motion parallax	62
5.2.2	Interpretation und Ergebnis	67
5.3	Implementierung	68
5.3.1	Stabilisierung der Tracks	69
5.4	Multisensorfusion	71
6	Alternativer Tracking Ansatz	72
6.1	Konzept	73
6.2	Mathematische Betrachtung und Implementierung	75
6.2.1	Ausnutzung des Fußpunktes	76
6.2.2	Schätzung der Normale	77
6.3	Diskussion	79
7	Experimentelle Untersuchungen und Ergebnisse	84
7.1	Synthetische Daten	84
7.2	Vergleich mit anderen Sensoren	86
7.2.1	Radar-Lidar	86
7.2.2	Stereo	87
7.3	Vergleich mit anderen Verfahren	88

7.3.1	Vergleich mit optischem Fluss	88
7.3.2	Vergleich mit Ego-Motion	90
7.4	Ergebnisse der Hindernisdetektion	91
7.5	Zusammenfassung der Ergebnisse	99
8	Zusammenfassung und Ausblick	100
A	Radargrundlagen	102
A.1	Funktionsweise des Radar	102
A.2	Winkelbestimmung über die sequentielle Mehrkeulenbildung	103
	Literaturverzeichnis	105
	Erklärung zur Diplomarbeit	113

Abbildungsverzeichnis

1.1	klassischer Auffahrunfall	2
2.1	Hinderniserkennung in zwei Schritten	8
2.2	Symmetrieoperator	9
2.3	Schatten und Kanten	10
2.4	Rektifizierung und Triangulation	12
2.5	Beispielbild einer Stereokamera.	12
2.6	Prinzip der Inverse Perspektive	13
2.7	Prinzip des Motion Stereo	15
2.8	Hinderniserkennung im Fahrzeugnahbereich	16
2.9	Verfahren der Computer Vision in einem Objekt-Szene Koordinatensystem	19
3.1	Lochkamera	22
3.2	Beziehung zwischen Sensor- und Bildkoordinatensystem	24
3.3	Aufbau des Versuchsträgers	26
3.4	Mehrdeutigkeit des Bewegungsfeldes	28
3.5	Bewegungsfeld mit Expansionspunkt	29
3.6	Verschiedenheit von optischem Fluss und Bewegungsfeld	30
3.7	Berechnung von Korrelationskoeffizienten	32
3.8	Bildpyramide	32
3.9	Vergleich der Hager und CONDENSATION Tracker	34
3.10	Beleuchtungsmodell für Lambertsche Oberflächen	39
3.11	Fehler durch Verdeckungen	42
3.12	Beispiel einer Inversen Perspektive	48
4.1	Skalierung eines Hindernisses	54
4.2	Kompensierung der Kamerarotation	56
4.3	Ablauf der Hypothesengenerierung	58
5.1	Planar motion parallax ($d = 10$ Meter)	65
5.2	Planar motion parallax ($d = 30$ m)	65

5.3	Planar motion parallax - Schnitt durch YZ Ebene	66
5.4	Abbildung einzelner Unterregionen bei Kameratranslation	70
6.1	Frame 141 und 144 aus der Malsheim Sequenz	74
6.2	Fehlergebirge	74
6.3	Verbessertes Trackergebnis durch Gewichtung der Beobachtungen	79
6.4	Fehlergebirge für abgebildete Straße	80
6.5	Frame 43 aus der LKW Beladestation	80
6.6	Ebenenschätzung in synthetischen Sequenzen.	81
6.7	Gütegebirge für die Ebenenschätzung	82
6.8	Ebenenschätzung in realen Sequenzen.	83
7.1	Entfernungsschätzung für synthetische Daten	84
7.2	Hypothesenverifizierung für synthetische Daten	85
7.3	Vergleich mit Radar	86
7.4	Vergleich mit Stereo Entfernungsmessung	87
7.5	Optische Flussverfahren	89
7.6	Schätzung des relativen Nickwinkels	91
7.7	Malsheim Sequenz	93
7.8	Böblinger Parkplatzsequenz	95
7.9	LKW Beladestation	97
A.1	Richtcharakteristik eines Radarsignals	103
A.2	Abdeckung eines Radarsensors im Automobilbereich	104

Tabellenverzeichnis

2.1	Überblick über Winkelauflösung verschiedener Sensoren zur Objekterkennung . . .	6
2.2	Vor- und Nachteile verschiedener Sensoren zur Objekterkennung	7
2.3	Vergleich verschiedener Ansätze zur Hypothesengenerierung.	17
3.1	Berechnung von Korrelationskoeffizienten	32
5.1	Notwendige Geschwindigkeit zur Hindernisdetektion	66
7.1	Ergebnisse der Entfernungsmessung aus optischem Fluss	89

Kapitel 1

Einleitung

1.1 Motivation

Menschen halten die Art, wie sie Ihre Umgebung wahrnehmen, oftmals für selbstverständlich. Wir sehen ein Hindernis im Straßenverkehr und umfahren es automatisch, ohne uns etwas dabei zu denken. Tatsächlich aber stellt eine so einfache Routine wie das Umfahren von Hindernissen ein komplexes Zusammenspiel von Wahrnehmung, Interpretation, Erfahrung und Koordination dar, das wir nur sehr begrenzt verstehen können.

Ein Roboter, und somit auch ein autonomes Fahrzeug, ist für die Wahrnehmung der Umgebung auf Messdaten angewiesen. Ein breites Spektrum von Sensoren wird benutzt, um die Umwelt abzutasten. Die Verarbeitung von Umgebungsinformationen in der Robotik und im Automobilbau hat in den letzten Jahren immer mehr an Bedeutung gewonnen. Die Detektion von stationären Hindernissen im Fahrkorridor kann zum Beispiel helfen, Unfälle zu vermeiden. Viele gefährliche Auffahrunfälle auf Stauenden, wie in Abbildung 1.1 aufgezeigt, könnten mit einem entsprechenden Warnsystem verhindert werden. Neueste Untersuchungen haben gezeigt, dass sechs von zehn Auffahrunfällen hätten vermieden werden können, wenn der Fahrer auch nur Sekundenbruchteile früher reagiert hätte [2]. Während ursprünglich vor allem die Entfernungsbestimmung im Vordergrund stand, gewinnt zunehmend die automatische Wahrnehmung und Verifizierung von Hindernissen - eine Domäne der Computer Vision - an Bedeutung. Bisher können sich visuelle Verfahren jedoch nicht gegen aktive Sensoren (z.B. Radar, Lidar) durchsetzen, was vor allem an der komplexeren und damit langsameren Informationsverarbeitung liegt. Mit immer höherer Prozessorleistung und modernen Algorithmen wächst jedoch das Potenzial, mit Bildverarbeitung in Echtzeit eine Umgebungserfassung durchzuführen. Dies hat dazu geführt, dass das Aufgabenfeld der Computer Vision sich am Anfang eines breiteren Einsatzes in zahlreichen industriellen Produkten befindet. Gleichzeitig ist es extrem schwierig, einem autonomen Roboter menschliche Fähigkeiten beizubringen. Es gibt beispielsweise kein Programm, welches automatisch beliebige Arten von Hindernissen erkennt um umfährt.



Abbildung 1.1: klassischer Auffahrunfall [80]

Die automatische Detektion von Hindernissen ist ein bekanntes Problem im Bereich Computer Vision. Die Abteilung REI/AI der DaimlerChrysler AG beschäftigt sich mit der Umfeldinterpretation basierend auf visuellen Sensoren [26]. Diese bieten vor allem den Vorteil einer hohen räumlichen Auflösung. Zur Detektion von Hindernissen im Fahrkorridor wurde beispielsweise der *Vision Bumper*¹ entwickelt. Über ein Stereokamerasystem wird die Fahrbahn vor dem Fahrzeug beobachtet und das Fahrzeug im Falle einer Hindernisdetektion gezielt abgebremst, so dass es vor dem Hindernis zum Stehen kommt. Die Lösung des Problems der Bewegungsschätzung und Hindernisdetektion in Stereosequenzen ist seit langem bekannt, allerdings gibt es bisher noch keine allgemeine Lösung für monokulare Videosequenzen. Monokulare Systeme sind jedoch, vor allem wegen der aufwendigen Kalibrierarbeit von Stereosystemen und den niedrigeren Kosten, attraktiver als ein Stereosystem. Steht lediglich eine Kamera zur Verfügung, so kann die 3D Information nicht unmittelbar abgeleitet werden. Allerdings kann bei bewegter Kamera unter Nutzung von Bewegungsinformation diese Information aus einer Bildsequenz bestimmt werden. Das Problem liegt jedoch gerade darin, Hindernisse in größerer Entfernung im Fahrkorridor vor dem Fahrzeug wahrzunehmen, da die abgebildeten relativen Bewegungen weiter entfernter Objekte nur sehr gering sind. An diesem Punkt scheitern alle bisherigen Lösungen für monokulare Systeme.

Die Wahrnehmung und Verifizierung von Hindernissen ist einer der ersten Schritte, diese zu umfahren. Außerdem kann eine Hindernisdetektion im Zusammenhang mit einem Spurhaltesystem zum Einsatz kommen, um ein semiautonomes Fahren zu ermöglichen, welches nur bei Hindernissen im Fahrkorridor manuelle Unterstützung benötigt. Daneben ist die monokulare Hindernisdetektion ein möglicher Ansatz, eine komplette 3D Rekonstruktion mit nur einer Kamera zu verwirklichen.

¹engl. *vision* - das Sehen, *bumper* - die Stoßstange

1.2 Zielsetzung

Ziel der Arbeit ist es, die Detektion und Verifikation von stationären (sich nicht bewegenden) Hindernissen in Bildsequenzen einer auf einem bewegten Fahrzeug fest installierten monokularen Kamera zu untersuchen. Der Algorithmus soll in Echtzeit ausführbar sein, damit Hindernisse als solche rechtzeitig erkannt werden. Dabei ist es wichtig, Aussagen über Position und Größe des Hindernisses zu formulieren. Ein Schwerpunkt liegt hier in der Bestimmung von horizontalen Abmessungen. Die für die Nutzung von Bewegungsinformation notwendige Geschwindigkeit des Fahrzeuges ist bekannt.

Für den Fall, dass eine Hindernishypothese mit Position des Hindernisses gegeben ist (beispielsweise durch Radar), soll das monokulare System als Verifikation der Hypothese eingesetzt werden. Es soll erforscht werden, wann eine zuverlässige Aussage über die vom aktiven oder passiven Sensor gelieferte Objekthypothese gemacht werden kann. Hierbei spielt sowohl die absolute Entfernung des Hindernisses als auch der zurückgelegte Weg eine Rolle, ab dem eine sichere Aussage möglich ist. Die vorliegende Forschungsarbeit soll mit einem experimentellen Teil auf Videosequenzen aus einem Fahrzeug abgeschlossen werden.

Ziel der Arbeit ist es, die Schwachstelle eines monokularen Systems, also den Fahrkorridor vor dem Fahrzeug, zu untersuchen. Dabei ist nicht die Klassifizierung von Hindernissen Ziel der Arbeit. Vielmehr soll ein alternativer Erkennungskanal realisiert werden, der in der Lage ist, ohne Nutzung von Vorwissen auf sich vor dem Fahrzeug befindende stationäre Hindernisse jeglicher Art zu reagieren. In der vorliegenden Ausarbeitung wird daher der Begriff *Hinderniserkennung* im Sinne einer Hinderniswahrnehmung und -detektion verstanden.

1.3 Aufbau der Arbeit

Im Anschluss an diese Einleitung wird in Kapitel 2 ein Überblick über die in der Literatur beschriebenen Ansätze zur Hinderniserkennung gegeben. Der Schwerpunkt liegt dabei auf den optischen Verfahren, insbesondere bei der Videobildverarbeitung. In Kapitel 3 werden die für diese Arbeit notwendigen Grundlagen der Computer Vision erläutert.

Die visuelle Hinderniserkennung wird in Kapitel 4 vorgestellt. Kapitel 5 beschäftigt sich mit der Verifikation von Hindernissen. In diesem Zusammenhang wird auch eine Multisensorfusion angesprochen. Motiviert durch die in Kapitel 4 und Kapitel 5 dargestellten Ansätze wird in Kapitel 6 ein alternatives Tracking Verfahren zur Tiefenrekonstruktion beschrieben.

Die erzielten Ergebnisse der Untersuchungen werden in Kapitel 7 dargestellt. Die Zusammenfassung der Arbeit und ein Ausblick auf folgende Untersuchungen finden sich in Kapitel 8. Hierbei werden auch mögliche Anwendungen diskutiert.

In Anhang erfolgt eine kurze Einführung in die Radartechnik. Sie dient zum besseren Verständnis der meist genutzten Methode der Hinderniserkennung und der damit verbundenen Probleme.

Kapitel 2

Überblick über den aktuellen Stand der Forschung

Zur autonomen und kollisionsfreien Navigation ist eine Positionsbestimmung und Hindernisdetektion in Echtzeit durch das jeweilige autonome System nötig. Dazu muss die Umgebung mittels Sensorik wahrgenommen werden, um (analog dem menschlichen Auge) *sehen* zu können. Es gibt verschiedene Ansätze dieses Problem zu lösen. Im folgenden Abschnitt wird zuerst ein kurzer historischer Überblick über autonome Fahrzeuge weltweit gegeben. Danach folgt ein Abschnitt zur Sensorik im Umfeld eines Fahrzeuges. Hierbei wird der Ansatz motiviert, Hinderniserkennung auf monokularen Videosequenzen durchzuführen. Anschließend werden die einzelnen Schritte der Hindernisdetektion und -verifikation erläutert. Der Schwerpunkt liegt dabei, motiviert durch den vorherigen Abschnitt, auf Verfahren mit monokularen Bildsequenzen. Abschließend wird noch einmal zusammenfassend die Einordnung der Hinderniserkennung im Automobilbereich in den Stand der Technik dargestellt.

2.1 Autonomes Fahren weltweit

Mit dem ambitionierten Ziel, autonome Fahrzeuge zu bauen, haben viele öffentliche aber auch private Institutionen verschiedene Projekte weltweit gestartet, an denen eine große Anzahl an Forschungsgruppen aktiv beteiligt sind. Diese Bemühungen brachten einige Prototypen und Lösungen basierend auf verschiedenen Ansätzen hervor [4].

In Europa geht hierbei ein großer Anteil der Aktivitäten auf das PROMETHEUS¹ Projekt zurück, welches zu einer großen Anzahl an Veröffentlichungen von 1990 bis 1996 beitrug. Mehr als 13 Autohersteller und Forschungsinstitute aus 19 europäischen Ländern waren an dem Projekt beteiligt. Mehrere Prototypen (unter anderem VaMoRs, VaMP, VITA, MOB-LAB) sind während der Projektzeit entstanden. 1996 entstand die *Advanced Cruise-Assist Highway*

¹PROgram for a European Traffic with Highest Efficiency and Unprecedented Safety

System Research Association (AHSRA) zwischen der Automobilindustrie und einer großen Anzahl an Forschungseinrichtungen.

Auch in den USA wurden viele Initiativen gestartet, autonome Fahrzeuge zu entwickeln. 1995 richtete die US Regierung das *National Automated Highway System Consortium* ein und startete die *Intelligent Vehicle Initiative* (IVI) im Jahr 1997. Einige viel versprechende Protosysteme wurden in den letzten 15 Jahren entwickelt [81]. Im März 2004 stieß die "grand challenge", veranstaltet von der DARPA, auf großes Medieninteresse [85]. Spätestens seit diesem Zeitpunkt war die Entwicklung nicht mehr aufzuhalten. In diesem Wettkampf müssen 250 Meilen (400 km) Wüstenparcour innerhalb einer festbestimmten Zeit absolviert werden, um ein Preisgeld von einer Million Dollar zu gewinnen. Der komplette Kurs muss ohne jegliche menschliche Interaktion absolviert werden. Obwohl das beste Team gerade einmal sieben Meilen weit fuhr, war dies ein großer Schritt in Richtung autonomes Fahren. Ein Jahr später, 2005, sollte es zwei Teams gelingen, innerhalb der vorgegebenen Zeit die komplette Strecke ohne menschliche Interaktion zurückzulegen.

2.2 Sensorik zur Objekterkennung im Fahrzeugumfeld

Die meisten in Serie befindlichen Objekterkennungssysteme basieren auf aktiven Sensoren wie Radar oder LIDAR, beispielsweise die DISTRONIC PLUS [16] bei DaimlerChrysler. Sie heißen aktiv, weil sie selbst Strahlung aussenden. Lediglich im *Active Driving Assist* (ADA) von Subaru wird ein Stereokamerasystem eingesetzt. Aktive Sensoren haben den Vorteil, dass sie gewisse Daten (zum Beispiel Abstand) direkt messen können und dadurch weniger Rechenaufwand benötigt wird. Prototypen mit aktiven Sensoren haben viel versprechende Ergebnisse erzielt. Trotzdem haben aktive Sensoren einige Nachteile. Vor allem die geringe räumliche (*spatiale*) Auflösung und die langsame Abtastgeschwindigkeit sind hier zu nennen. Wenn sich zusätzlich viele Fahrzeuge simultan in die gleiche Richtung bewegen, ist Interferenz ein weiteres Problem aktiver Sensoren.

Optische Sensoren, welche zur Gruppe der passiven Sensoren gehören, reagieren lediglich auf in der Umgebung vorhandene Strahlung. Offensichtlich liefert ein passiver Sensor keine Daten, wenn keine Strahlung in der Umgebung vorhanden ist, auf welche der Sensor reagieren kann. Bei handelsüblichen Kameras ist dies der Fall, wenn kein Umgebungslicht vorhanden ist. Ein Vorteil eines optischen Sensors sind die Kosten. Vor allem in Serienproduktion zahlen sich geringere Stückkosten aus. Optische Sensoren können bewegte Objekte effizienter verfolgen (*tracke*) als aktive Sensoren automatisch verfolgen und auch in anderen Applikationen, wie zum Beispiel der Spur- oder Verkehrszeichenerkennung, angewandt werden. Somit ergeben sich Synergievorteile. Ein weiterer wesentlicher Vorteil optischer Sensoren gegenüber aktiven Sensoren ist die höhere räumliche Auflösung. Einen Überblick über die Auflösungen verschiedener Sensoren liefert Tabelle 2.1. Bei einem Hindernis in 50 Metern Entfernung kann man

errechnen, dass mit einer Standardkamera je nach Öffnungswinkel bei einer horizontalen Auflösung von 640 Pixeln alle 3,5cm bis 5,5cm eine Beobachtung möglich ist. Gängige Laserscanner mit 1° Winkelauflösung liefern lediglich eine Beobachtung alle 87cm. Der Unterschied in der räumlichen Auflösung ist hier enorm.

Sensor	Reichweite	Winkelauflösung
Ultraschall	< 10m	sehr gut
Hella Fix Beam (16 Beams)	100m	1°
SICK LMS 200	80m	$1^\circ - 0.25^\circ$ (abh. von Ansprechzeit)
Radar	-	Punktziele
Kamera (40° , 640px)	-	0.0625°
Kamera (25° , 640px)	-	0.04°

Tabelle 2.1: Überblick über Winkelauflösung verschiedener Sensoren. Bei der Berechnung für die Standardkameras sind zugrunde liegender Öffnungswinkel und Auflösung mit angegeben. Daten nach [41][76]

In Tabelle 2.2 sind, anlehnend an [31][41], die Vor- und Nachteile der einzelnen Sensoren dargestellt. Die Funktionsweise der einzelnen Sensoren werden in dieser Arbeit nicht erläutert. Zum besseren Verständnis ist jedoch in Anhang A ein Überblick zu finden. Gerade im Bezug auf sicherheitsrelevante Systeme bietet sich eine Sensorfusion an, um die Vorteile verschiedener Sensoren zu kombinieren. Mehr dazu ist in Abschnitt 5.4 zu finden.

2.2.1 Diskussion der Erkenntnisse

Objekterkennungssysteme mit aktiven Sensoren sind bereits weitgehend erforscht und vielfach im Einsatz. Trotzdem haben diese Systeme spezifische Probleme, die gerade im Bezug auf die Sicherheitsrelevanz an Bedeutung gewinnen. Die Winkelauflösung von LIDAR motiviert eine alternative Objekterkennung um Objektgrenzen besser zu lokalisieren. Das Radar liefert als potentielle Objekte lediglich Punktziele und somit keine Objektgrenzen. Aufgrund von Reflexion kann es außerdem vorkommen, dass fälschlicherweise Hindernisse erkannt werden. Gibt es hier noch schnelle Methoden, die Reflexionen zu kompensieren, so ist es weiterhin nur mit sehr hohem Aufwand - falls überhaupt - möglich, Objekte wie zum Beispiel Straßenbahnschienen oder Dehnungsfugen von Autobahnbrücken bei positiven Hindernishypothesen zu falsifizieren. Hier bieten sich jedoch Verfahren der Bildverarbeitung an.

In der Vergangenheit wurde vor allem Stereobildverarbeitung betrieben, um solche Probleme zu lösen. Allerdings ist eine genaue Kalibrierung eines solchen Kamerasystems notwendig, um akzeptable Ergebnisse zu erhalten. Daher liegen die Kosten eines Stereosystems weit über den Kosten eines einfachen monokularen Videosensors. Bisher ist aber noch keine allgemeine Lösung für die Umgebungswahrnehmung aus monokularen Videosequenzen bekannt. Ein bes-

Sensor	Vorteile	Nachteile
Ultraschall	geringe Kosten, unproblematisch bei Dunkelheit	sehr kurze Reichweite, sehr empfindlich gegenüber Wetterbedingungen
LIDAR	unproblematisch bei Dunkelheit, geringer Rechenaufwand, konstante Entfernungsauflösung, ausreichende Winkelauflösung da gute Fokussierung möglich	empfindlich gegenüber Wasser, Schmutz, Umgebungslicht, Geschwindigkeitssignal wird aus Entfernungsänderung errechnet
Radar	unproblematisch bei nahezu allen Licht- und Wetterverhältnissen, geringer Rechenaufwand, konstante Entfernungsauflösung, direkte Messung der Relativgeschwindigkeit unter Nutzung des Dopplereffekts, Einbau hinter Plastik möglich	teuer, schlechte Winkelauflösung, Punktziele, problematisch bei querbewegten Objekten und Reflexionen, reagiert auf Metall in der Umwelt, beispielsweise Dehnungsfugen oder Eisenbahnschienen
Kamera	Detektion mehrerer Objekte und Klassifikation möglich, gute Auflösung	Wetterabhängig, problematisch bei Dunkelheit, keine direkte Abstandsbestimmung möglich, rechenintensive Algorithmen

Tabelle 2.2: Überblick über die Vor- und Nachteile verschiedener Sensoren zur Objekterkennung.

seres Verständnis der Umgebungswahrnehmung mit einem monokularen System ist für viele Bereiche der Forschung und Industrie interessant. In der Miniaturisierung beispielsweise ist man auf die Verarbeitung von monokularen Bildsequenzen angewiesen, da aufgrund verschwindend kleiner Basislänge keine Stereoinformation abgeleitet werden kann. Aus einer einzelnen Kamera lässt sich keine direkte Tiefeninformation ableiten. Betrachtet man allerdings in einem monokularen System bei bewegter Kamera dieselbe Szene zu zwei verschiedenen Zeitpunkten, so lässt sich das Prinzip der Stereobildverarbeitung übertragen. Der Oberbegriff für ein solches Vorgehen heißt *motion stereo*. Eine genauere Erläuterung ist in Abschnitt 2.3.1.3 zu finden. Dass solche Verfahren gut funktionieren können lehrt uns die Natur. Bienen beispielsweise nutzen diesen Effekt zur Hinderniserkennung und zur Navigation [23].

2.3 Die zwei Schritte der Objekterkennung

Die robuste Objekterkennung lässt sich in zwei Schritte unterteilen [79]. Die gesamte Umwelt nach potentiellen Hindernissen zu untersuchen ist aufgrund der Echtzeitbedingung wenig sinnvoll. Die meisten Verfahren in der Literatur folgen daher den beiden Schritten der Hypothesengenerierung (HG) und der Hypothesenverifizierung (HV) (vgl. Abbildung 2.1). Während der Hypothesengenerierung werden möglichst schnell eine oder mehrere Hypothesen möglicher

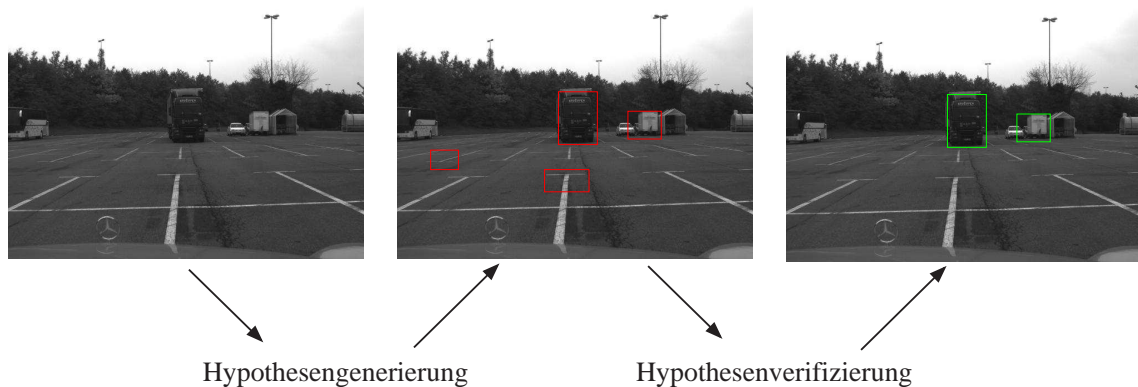


Abbildung 2.1: Hinderniserkennung in zwei Schritten

Positionen von Hindernissen aufgestellt. In der Hypothesenverifizierung werden Tests durchgeführt, um das Vorhandensein eines Hindernisses zu bestätigen. Die Hypothesengenerierung kann beispielsweise sehr schnell mittels eines aktiven Sensors erfolgen. Ein kurzer Überblick wird in Anhang A gegeben. Im Folgenden wird die Hypothesengenerierung mit optischem Sensor näher betrachtet. Bei der Hypothesenverifizierung ist ein aktiver Sensor einem optischen Sensor unterlegen, da ein optischer Sensor eine größere Auswahl an Tests generieren kann. Die nächsten beiden Abschnitte zeigen Möglichkeiten auf, beide Schritte mit optischen Sensoren zu realisieren.

2.3.1 Hypothesengenerierung

Das Ziel der Hypothesengenerierung ist es, schnell Kandidaten für mögliche Hindernisse im Bild zu lokalisieren. Sun [79] teilt die Ansätze grob in drei Unterkategorien auf:

1. Vorwissenbasierte Ansätze.
2. Stereobasierte Ansätze.
3. Bewegungsbasierte Ansätze.

Enkelmann [22] teilt die Verfahren zur Objekterkennung ein einem Fahrzeug in vier Kategorien ein, indem er die Auswertung von Differenzbildern nicht der Kategorie der stereobasierten Verfahren zuordnet, sondern als eigene Kategorie ansieht. Im Folgenden werden die einzelnen Ansätze kurz dargestellt und anschließend tabellarisch zusammengefasst.

2.3.1.1 Vorwissenbasierter Ansatz

Auf Vorwissen basierende Ansätze benutzen a-priori Wissen, um Kandidaten potentieller Hindernisse im Bild zu finden. Einige Herangehensweisen sind zum Beispiel die Benutzung von

Farbe, Symmetrie, Schatten, Ecken, Kanten, Texturen und Fahrzeuglichtern. Zum besseren Verständnis der Problemstellung folgt ein kurzer Überblick über diese Verfahren und deren Problematik. Diese Ansätze werden nicht detailliert diskutiert, da das Nutzen von Vorwissen nicht Gegenstand dieser Arbeit ist. Vielmehr soll in dieser Arbeit ein alternativer Erkennungskanal entwickelt werden.

- **Farbe:** Obwohl nur wenige Systeme Farbinformationen für eine Hypothesengenerierung nutzen, beinhaltet Farbe wertvolle Informationen für die Hinderniserkennung, Spurerkennung und Verkehrszeichenerkennung [40][8]. Farbinformationen können auch dazu genutzt werden, Objekte aus dem Hintergrund heraus zu segmentieren [37]. Ähnliche Ansätze können für die Hypothesengenerierung genutzt werden, da alle Regionen im Bild, die eine andere Farbe als die typische der Straße aufweisen, potentielle Hindernisse darstellen. Das Problem der Verwendung von Farbinformationen liegt darin begründet, dass zum einen mehr Information verarbeitet werden muss und zum anderen Farbwerte für verschiedene Beleuchtungen und Betrachtungsrichtungen unterschiedlich sein können. Insbesondere im Außenbereich kann dieses Problem nicht umgangen werden.
- **Symmetrie:** Hindernisse, speziell Autos, welche von vorne oder hinten abgelichtet werden, sind in der Regel sowohl in horizontaler als auch in vertikaler Richtung symmetrisch. Abbildung 2.2 zeigt ein Beispielbild, in dem symmetrische Bereiche automatisch detektiert wurden. Bereits in den frühen 90er Jahren wurde dies zur Fahrzeugerkennung genutzt [57]. Problematisch wird die Erkennung symmetrischer Strukturen in homogenen Regionen, da in diesen Regionen Symmetrieberechnung anfällig für Rauschen ist. In [6] werden deshalb Ecken und Kanten zur Berechnung der Symmetrie mitgenutzt. Allerdings müssen zwei Dinge beachtet werden, wenn man lokale Symmetrien sucht: Erstens ist es zu aufwendig, das komplette Bild nach Symmetrien zu untersuchen und zweitens ist der Symmetrieoperator anfällig für Falschdetektionen.

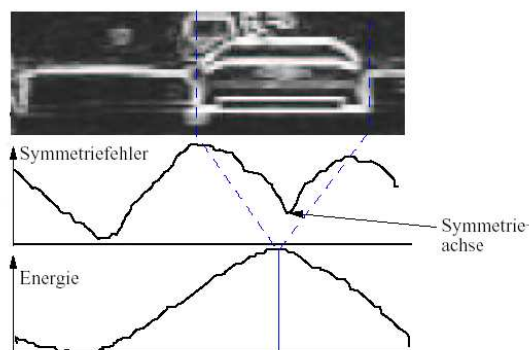


Abbildung 2.2: Kombination von Symmetrie und Energie zur Lokalisation von Fahrzeugen [12]



Abbildung 2.3: (a) ROI. (b) In weiß die Bildpunkte, die einen vordefinierten Grauwert unterschreiten. (c) Vertikale, horizontale and kombinierte Bildkanten. [48]

- **Schatten:** In [67] und [19] wurde festgestellt, dass die Region unter potentiellen Hindernissen dunkler ist, als andere Regionen der Straße. Wie in Abbildung 2.3a gezeigt wird, sind Grauwerte der Reifen und einige Bildzeilen darüber sowie darunter sehr dunkel. Um nach solchen Mustern zu suchen, wird ein binäres Bild der ROI (*region of interest*, Zielregion) berechnet. In Abbildung 2.3b werden alle Bildpunkte in weiß dargestellt, die einen von der Beleuchtung abhängigen Grauwert unterschreiten. Alle Bildpunkte die über dem Schwellwert liegen sind schwarz. Allerdings kann in allgemeinen Situationen kein vernünftiger Schwellwert für eine Hypothesengenerierung gefunden werden. Die Intensität der Zielregion hängt ab von der Helligkeit des Bildes, welche wiederum von den Wetterbedingungen abhängt. Daher ist solch ein Schwellwert keinesfalls fix.
- **Ecken:** Obleich Fahrzeuge als meistauftretende Hindernisse rechteckig sind, schlägt Bertozzi [3] vor, Ecken zur Hypothesengenerierung zu benutzen. Vier Templates (Schablonen) werden benutzt um alle Ecken im Bild zu finden. Die so gefundenen Ecken werden auf Zusammengehörigkeit untersucht.
- **Kanten:** Die meisten Hindernisse beinhalten viele horizontale und auch vertikale Kanten. Fahrzeuge zum Beispiel weisen solche Strukturen vor allem entlang der Außenkanten auf. Abbildung 2.3c zeigt horizontale, vertikale und kombinierte Kanten in der ROI des Objekts. Viele Verfahren benutzen eine bestimmte Anordnung von Kanten, um Hypothesen für Fahrzeuge zu erstellen [74][17]. Betke et al. [7] benutzen Kanteninformation, um weit entfernte Fahrzeuge zu detektieren. Es wird eine horizontale und vertikale Kantendetektion ausgeführt und die Werte für die Spalten und Zeilen werden addiert. Mit einem Filter wird das Profil geglättet und über Maximumssuche Hypothesen für Fahrzeuge erstellt. Der Algorithmus benutzt verschiedene Auflösungsstufen des Bildes, um Rechtecke mit horizontalen und vertikalen Kanten zu finden und propagiert das Ergebnis von der größten Auflösungsstufe auf die höchste (*Coarse-to-fine* Suche). In [77] werden vertikale und horizontale Kanten separat mit dem Sobelfilter gesucht. Danach werden Filter, welche Kanteninformation nutzen, verwendet um die Ergebnisse nach Fahrzeugen zu durchsuchen.

Die Nutzung von Kanten und Ecken als Schlüssel für eine Objekterkennung kann zu sehr effektiven Ergebnissen führen. Allerdings ist noch nicht genauer untersucht, wie verschiedene Parametereinstellungen die Robustheit solcher Systeme beeinträchtigt. Zum Beispiel müssen Schwellwerte für Kantendetektoren, Schwellwerte für wichtigste Kanten und Schwellwerte für Maxima und Minima eingegeben werden. Besonders in Außenbereichen mit sich ändernden Sichtverhältnissen ist die richtige Einstellung dieser Parameter problematisch.

- **Textur:** Objekte im Bild führen zu lokalen Veränderungen der Intensitäten. Für Fahrzeuge beispielsweise ist die Intensitätsänderung, bis auf einige Ausnahmen, ähnlich. Dieses Muster der Intensitätsänderung wird in [34] auch Textur genannt. Haberäcker [34] nennt als Maße für die Textur Streuung (Richtigkeit einer Struktur), Gerichtetheit einer Textur, den Kontrast und die Fleckigkeit. Gängige Operatoren basieren auf den Grauwerten des Bildes, dem Gradientenbild, der Autokorrelation oder dem zweidimensionalen Fourierspektrum. In [49] wird als Maß für die lokale Struktur die Entropie berechnet. Eine solche Texturinformation kann die Suche nach Hindernissen einschränken. Allerdings ist a-priori Wissen bei einem solchen Ansatz unumgänglich.
- **Frequenz:** Unter der Annahme, dass Hindernisse texturiert sind, gehen frequenzbasierte Ansätze davon aus, dass bei perspektivischen Abbildungen mit zunehmender Entfernung auch die Bildfrequenz zunimmt. In psychologischen Studien wurde gezeigt, dass auch das menschliche Auge ein solches Prinzip nutzt [89]. In [72] wird für Regionen im Bild eine so genannte *average peak frequency* (APF) berechnet, um das räumliche Frequenzspektrum zu charakterisieren. Daraus werden Tiefenwerte abgeleitet. Für viele künstlich generierte Bilder sind hervorragende Ergebnisse erzielt worden, allerdings ist der Nachteil dieses Ansatzes, dass Regionen mit einer großen Anzahl an Intensitätsunterschieden und damit hohen Frequenzen generell zu weit entfernten Objekten korrespondieren, was nicht notwendigerweise der Fall ist.
- **Fahrzeuglichter:** Die meisten der oben genannten Verfahren sind nachts nicht hilfreich. Der Versuch, Ecken, Kanten oder Schatten zu erkennen wäre sinnlos. Fahrzeugfront- und -rückleuchten sind jedoch nachts sehr gut erkennbar. In ihrer Ausarbeitung haben Cucchiara et al. [15] einen morphologischen Operator benutzt, um solche Lichtpaare zu detektieren.

2.3.1.2 Stereobasierter Ansatz

Es gibt (nach Sun [79]) zwei Arten von Methoden, welche Stereoinformation zur Hindernisdetektion nutzen. Der erste Ansatz benutzt Disparitäten während der zweite auf die so genannte Inverse Perspektive aufbaut.

- Disparität:** Der Unterschied korrespondierender Pixel zwischen linkem und rechtem Bild eines rektifizierten Stereobildpaares heißt Disparität. Bei bekannter Orientierung beider Kameras kann mit Hilfe der Disparität der in die beiden korrespondierenden Pixel abgebildete 3D Weltpunkt bestimmt werden. In Abbildung 2.5 ist ein Bild eines Stereokamerasystems zu sehen, in dem der Farbverlauf die Entfernung kodiert. Ein solches Verfahren nennt sich Triangulation [24] und ist in Abbildung 2.4 verdeutlicht. Nach Franke et al. [27] können die Verfahren in zwei Kategorien unterteilt werden: die Flächen basierenden, wie zum Beispiel [54][73][27], und die auf Merkmalen basierenden Ansätze, beispielsweise [25][30]. Erstere berechnen zunächst eine dichte Tiefenkarte der in beiden Kameras aufgenommenen Szene, um darauf ein Clustering zu Objekten durchzuführen. Die Dichte der Tiefenkarte sowie die angewandten globalen und lokalen Randbedingungen minimieren das Auftreten von Fehlkorrespondenzen. In [39] wird das von Franke et al. beschriebene Stereosystem mit optischem Fluss fusioniert, um sich bewegende Objekte frühzeitig zu erkennen.

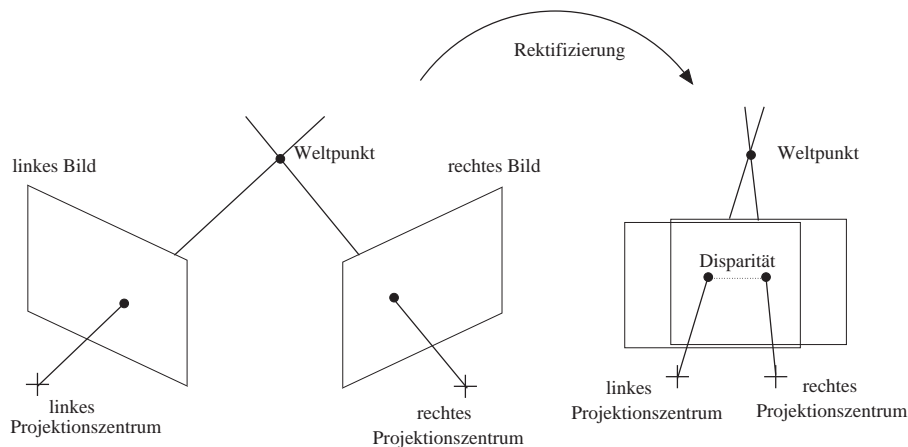


Abbildung 2.4: Rektifizierung und Triangulation

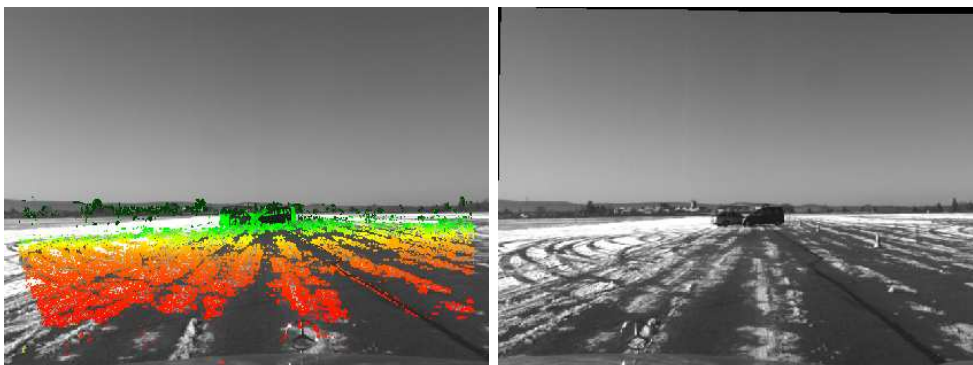


Abbildung 2.5: Beispielbild einer Stereokamera; der rot-grün Farbverlauf kodiert die Entfernung.

Die auf Merkmalsextraktion basierenden Ansätze suchen nach markanten Features wie Ecken oder Kanten, welche im Bild der zweiten Kamera korrespondierenden Pixeln zugeordnet werden können. Es ergeben sich dünn besetzte Tiefenkarten. Der Vorteil gegenüber den dicht besetzten Tiefenkarten ist die eingesparte Rechenleistung. Andererseits sind die Feature basierenden Ansätze nicht dicht genug für eine robuste Hindernisdetektion [25]. Der klassische Triangulationsansatz lässt sich auf bewegte monokulare Kameras übertragen, was den Begriff *motion stereo* erklärt. Hierauf wird weiter in Abschnitt 2.3.1.3 näher eingegangen.

- **Inverse Perspektive:** Der Ausdruck *Inverse Perspektive* trifft nicht auf die tatsächliche Inversion des Abbildungsvorgangs zu. Dies ist mathematisch nicht möglich. Vielmehr ist es eine Inversion unter der zusätzlichen Annahme, dass sich alle Punkte auf einer Ebene befinden. Somit handelt es sich um eine Homographie. Unter der Annahme einer ebenen Straße wird das linke Bild eines Stereokamerasystems unter Benutzung der Inversen Perspektive in das rechte Bild transformiert [93][91]. Genauer gesagt wird dabei jeder Punkt des linken Bildes unter der Ebenenannahme auf die Straßenebene projiziert und dann in das rechte Bild zurückprojiziert, wo er mit dem Bildpunkt im tatsächlich aufgenommenen rechten Bild verglichen wird. Alternativ können auch die Bilder beider Kameras in die Straßenebene projiziert werden. So ist es möglich, Regionen zu finden, die nicht in der Straßenebene liegen. Das Prinzip ist in Abbildung 2.6 verdeutlicht. Obwohl zwei

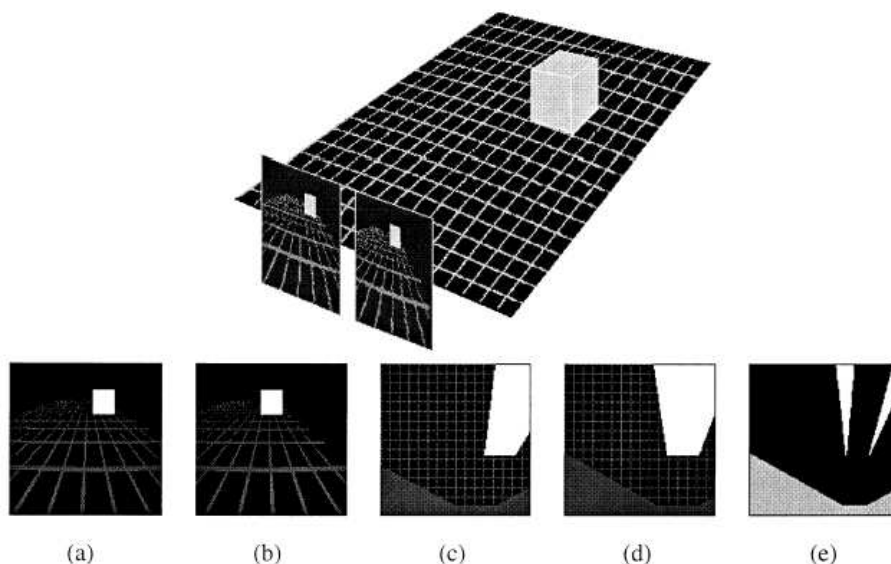


Abbildung 2.6: Das Prinzip der Inversen Perspektive: (a) linkes Bild; (b) rechtes Bild; (c) linkes Bild in die Straßenebene projiziert; (d) rechtes Bild auf die Straßenebene projiziert; (e) Differenzbild in dem der graue Bereich die Region zeigt, welche nicht von beiden Kameras eingesehen werden kann [5] .

Kameras für ein Verfahren mit Inverser Perspektive ausreichen, liefern weitere Kameras viele Vorteile. Vor allem können Fehlzuordnungen aufgrund repetitiver Muster herausgefiltert werden [90]. Anstatt mehrere Kameras zu benutzen, kann man auch Bildfolgen aufeinander warpen (entsprechend transformieren). Somit kann, wie auch bei der Nutzung der Disparität, der ursprünglich stereobasierte Ansatz auf eine monokulare Kamera übertragen werden [21].

Generell sind stereobasierte Ansätze nur dann exakt und robust, wenn die Parameter der Stereoanordnung genau bekannt sind. Dies ist bei Anwendungen im Außenbereich, insbesondere im bewegten Fahrzeug, nur schwer zu garantieren. Vibrationen können die Parameter der Kameras so verändern, dass eine zuverlässige Aussage nicht mehr ohne Nachkalibrierung möglich ist. Ein weiterer Nachteil der stereobasierten Ansätze gegenüber monokularen Systemen sind die höheren Kosten.

2.3.1.3 Bewegungsbasierter Ansatz

Alle bisher betrachteten Ansätze benutzen abgebildete Merkmale, um Hindernisse vom Hintergrund zu trennen. Eine andere Möglichkeit ist es, relative Bewegungen zu betrachten. Die relative Bewegung von Bildpunkten wird auch optischer Fluss genannt. Das Vektorfeld der beobachtbaren Verschiebungsvektoren im Bild wird optisches Flussfeld genannt. Bei bewegter Kamera und einer stationären Welt ergibt sich ein divergentes Flussfeld. Stationäre Hindernisse heben sich von dem Straßenfluss hervor. Bewegt sich frontal nähernde Hindernisse erzeugen einen divergierenden Fluss, welcher auch quantitativ von dem Fluss, welcher durch die Straßenebene generiert wird, abweicht. Bewegt, sich entfernende Hindernisse (beispielsweise überholende Fahrzeuge) erzeugen ein konvergentes Flussfeld. Betrachtet man bei bewegter Kamera dieselbe Szene zu zwei verschiedenen Zeitpunkten, lässt sich das Prinzip der Stereobildverarbeitung direkt übertragen. Die Kenntnis der Kameratrajektorie liefert die Orientierung der beiden Kameras und der Verschiebungsvektor eines Bildpunktes entspricht der Disparität (vgl. Abbildung 2.7). Der Oberbegriff einer solchen Analogie heißt *structure from motion, depth from motion* oder *Hindernisdetektion*. Ein Vergleich [32] zwischen der Nutzung unterschiedlicher optischer Verfahren wie Differentialmethoden erster und zweiter Ordnung für die Hinderniserkennung liefern enttäuschende Ergebnisse aus hauptsächlich drei Gründen:

- die relative Verschiebung zwischen aufeinanderfolgenden Frames,
- wenig Textur und
- Vibration und Eigenrotation des Fahrzeugs.

Gegeben das Szenario einer eigenbewegten Kamera, ist es nicht leicht, ein dichtes Flussfeld zu berechnen. Als Lösung schlägt Giachetti korrelationsbasierte Methoden vor, allerdings wird dabei verlangt, dass sich die zu detektierenden Objekte bewegen und sich so deutlich vom

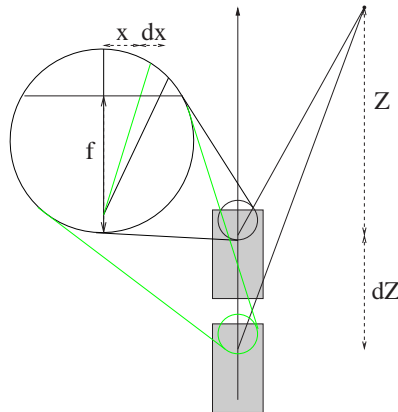


Abbildung 2.7: Prinzip des Motion Stereo: $\frac{x}{x+dx} = \frac{Z}{Z+dZ}$.

Hintergrund abheben. Dadurch ist eine direkte 3D Objektlokalisierung aufgrund der Fremdbewegung jedoch nicht möglich.

Im Gegensatz zu sehr zeitaufwendigen, dichten Flussverfahren, nutzen featurebasierte Verfahren Merkmale wie Ecken oder Kanten und sind aufgrund der markanten Features weniger rauschanfällig. Krüger [55] und Tomasi [82][75] nutzen den räumlich-zeitlichen Bildgradienten eines Grauwertbildes für einen lokalen Ansatz. Ein anschließendes Clustering [55] eliminiert Ausreißer. Andere Ansätze sind zu finden in [78][1].

Die Anwendung des *structure from motion* Ansatzes zur Bestimmung der 3D Position ausgewählter Bildpunkte ist allerdings einigen Einschränkungen unterworfen. Ausführliche Beschreibungen und mathematische Nachweise sind in den jeweiligen Quellangaben und in Kapitel 3.4 zu finden:

- Die beobachtete Szene muss statisch sein, da ansonsten außer der Kamerabewegung auch die Bewegung der Objekte das optische Flussfeld beeinflussen (rigidity constraint). Dies entspricht der Annahme, dass im Stereofall beide Bilder zum gleichen Zeitpunkt aufgenommen werden, also die gleiche Szene abbilden.
- Belichtungs- und Beleuchtungsänderungen können die gleichen Effekte im Bild verursachen wie der optische Fluss, welcher durch Bewegung verursacht wird [42]. Diese Einflüsse müssen als vernachlässigbar angenommen werden.
- Repetitive Muster und homogene Regionen können eventuell nicht eindeutig zugeordnet werden (Aperturproblem, Korrespondenzproblem) [38][66].
- Die Qualität der Szenenrekonstruktion hängt von der Länge der Flussvektoren ab. Die Varianz, also Unsicherheit in der Berechnung des 3D Weltpunktes, steigt überproportional mit geringen Verschiebungsvektoren [70].

Bewegungsbasierte Verfahren für Hindernisdetektion funktionieren sehr gut in Regionen, in denen Objekte große Verschiebungsvektoren gegenüber dem Hintergrund und dem Straßenfluss aufweisen. Dies ist vor allem der Nahbereich um das Fahrzeug. Hier konnten vielversprechende Verfahren entwickelt werden [84][70]. Auch bei überholenden Fahrzeugen ist diese Bedingung gegeben. Im Bereich des Fluchtpunktes (in Bewegungsrichtung der Kamera), dem so genannten *Focus of Expansion*, geht der Fluss und damit der Verschiebungsvektor allerdings gegen Null. Betke [7] und Handmann [36] benutzen daher für Objekte mit kleinen Winkeln in Richtung der Fahrzeugachse Methoden der Einzelbildauswertung mit a-priori Wissen, wie sie in Abschnitt 2.3.1.1 beschrieben sind. Bisherige Ansätze haben für Objekte in großer Entfernung direkt vor dem Fahrzeug (also im Bereich um den Focus of Expansion) keine erfolgreiche Nutzung von Bewegungsinformation erzielen können. Insbesondere die Erkennung stationärer Objekte, welche sich nicht deutlich vom Straßenfluss hervorheben, ist ein bisher nicht gelöstes Problem. Vor dem Hintergrund der sehr guten Ergebnisse, welche in [70] für den Fahrzeugnahbereich erzielt wurden (vgl. Abbildung 2.8), ist allerdings ein Ansatz, der Hinderniserkennung durch Verarbeitung von Einzelbildern und ohne Nutzung von Bewegungsinformation durchführt, als nicht optimal anzusehen.

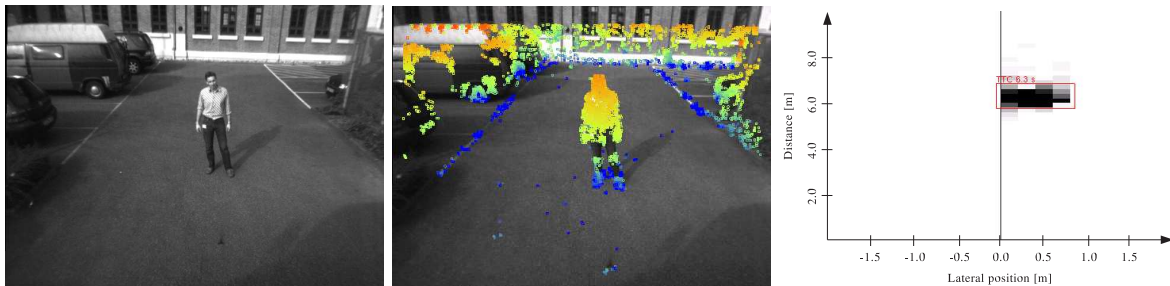


Abbildung 2.8: Hinderniserkennung im Fahrzeugnahbereich. Links Originalbild; mitte farbcodierte Höenschätzung nach 10 Frames; rechts Tiefenkarte mit detektiertem Hindernis nach 10 Frames.

2.3.1.4 Gegenüberstellung

Die einzelnen Ansätze der Hypothesengenerierung sind, anlehnend an die Klassifizierung in Abschnitt 2.3.1, noch einmal in Tabelle 2.3 zusammengefasst. Aufgrund der in Abschnitt 2.2 getroffenen Vorüberlegungen ist eine Hinderniserkennung mit monokularen Bildsequenzen das Ziel dieser Arbeit. Damit verbleiben Vorwissen-basierte Verfahren der Einzelbildauswertung und bewegungsbasierte Ansätze als mögliche Ansätze zur Hypothesengenerierung. In der in Abschnitt 1.2 formulierten Zielsetzung ist eine Erkennung stationärer Hindernisse ohne Klassifikation formuliert. Eine solche Klassifikation ist aber implizit Voraussetzung für auf

	Pro	Kontra
auf Vorwissen basierend	Einfacher Algorithmus, schnelle Ausführung, kein Einfluss von Fahrzeugeigenbewegung.	Keine Tiefeninformation ohne Zusatzannahmen, kein Erfolg falls Objekt nicht dem Model entspricht.
Stereo Vision	Erkennung allgemeiner Objekte, direkte 3D Rekonstruktion.	Komplexe Berechnung, genaue Kalibrierung notwendig
Nutzung von Bewegung	Erkennung allgemeiner Objekte, erlaubt Berechnung der Relativgeschwindigkeit.	Komplexe Berechnung, Anfällig bei nicht erfasster Eigenbewegung, keine Ergebnisse falls keine Eigenbewegung vorhanden.

Tabelle 2.3: Vergleich verschiedener Ansätze zur Hypothesengenerierung.

Vorwissen basierenden Verfahren. Somit ist ein Ansatz mit Nutzung der Bewegungsinformationen wünschenswert. Aufgrund der in Abschnitt 1.2 formulierten Problemstellung ist vor allem der Bereich in größerer Entfernung vor dem Fahrzeug zu untersuchen. Gerade hier liegen die Probleme der bewegungsbasierten Ansätze. Folglich müssen vernünftige Annahmen über die Szene getroffen werden, um dennoch eine Lösung zu finden. Eine ausführliche Diskussion und mathematische Betrachtungen sind in Abschnitt 4 zu finden. Alternativ ist eine Hypothesengenerierung mit aktivem Sensor denkbar.

2.3.2 Hypothesenverifizierung

Die Eingabe einer Hypothesenverifizierung ist eine Menge von Hypothesen, welche aus der Hypothesengenerierung stammt. Während der Verifizierungsphase werden Tests durchgeführt, um die Richtigkeit der Hypothese zu prüfen. Beim gegenwärtigen Stand der Technik kann nicht davon ausgegangen werden, dass eine vollständige semantische Interpretation aller denkbaren Situationen, also ein echtes *visuelles Verstehen* von natürlichen Bildsequenzen mit vertretbarem Aufwand realisierbar ist. Dies stellt jedoch die Verwendbarkeit der Computer Vision in klar umrissenen Szenarien nicht in Frage. Zwei Forderungen sind nach [65] Voraussetzung dafür, dass ein Vision-System als praxistauglich angesehen werden kann: Zuverlässigkeit und Effizienz. Praxistauglichkeit wird von Mester et al. [65] dabei verstanden als eine Vertrauenswürdigkeit und Verlässlichkeit des gelieferten Ergebnisses, wobei auch die Aussage *diese Daten können nicht ausgewertet werden* ein mögliches Ergebnis ist. Wünschenswert ist also ein Verfahren, welches seine eigenen Resultate bewerten kann. Förstner hat für Ansätze dieser Art den Begriff *traffic light modules* geprägt [28]: Ein Modul ist in der Lage zu bewerten, ob sein Ergebnis korrekt, zweifelhaft oder definitiv fehlerbehaftet (grün, gelb oder rot) ist. Gerade die Hypothesenverifizierung verlangt zwingend eine solche Implementierung.

Hypothesenverifizierungsansätze können nach [79] grob in zwei Hauptkategorien eingeteilt werden: templatebasierte Methoden und erscheinungsbasierte Methoden. Beide Methoden werden im Anschluss kurz vorgestellt. Nachteil dieser Ansätze ist die Notwendigkeit von a-priori Wissen. Durch Nutzung eines sehr lockeren Templates kann die Nutzung von Vorwissen minimiert werden, allerdings geschieht dies auf Kosten der Aussagengenauigkeit. Hier gilt es, vernünftige einschränkende Annahmen zu treffen, um allgemeine Hindernisse als solche verifizieren zu können. Die genauen Einschränkungen und mathematischen Betrachtungen sind in Abschnitt 5 zu finden.

2.3.2.1 Templatebasierend

Templatebasierte Methoden nutzen vordefinierte Schablonen oder Muster einer Objektklasse, so genannte Templates, und führen eine Korrelation zwischen Template und Bildregion durch. Einige Templates sind sehr locker formuliert, während andere sehr eingeschränkt sind. Parodi et al [68] schlägt eine Hypothesenverifizierung basierend auf Fahrzeugkennzeichen und Rückfenstern sowie der Fahrzeuggeometrie vor, um eine Hypothese für Fahrzeuge zu verifizieren. Handmann [36] schlägt eine templatebasierte Methode vor, in der auf das Vorhandensein einer *U-Form* geprüft wird (eine horizontale, zwei vertikale Kanten). Die meisten Templates beinhalten a priori Wissen und führen eine Klassifikation durch. Auf stationäre Hindernisse jeglicher Art zu reagieren ist somit nicht Ziel eines traditionellen templatebasierten Ansatzes.

2.3.2.2 Erscheinungsbasierend

Erscheinungsbasierte Ansätze lernen die Charakteristiken von Objektklassen aus einer großen Anzahl von Trainingsbildern, welche unterschiedliche Objektansichten beinhalten. Üblicherweise wird auch das Spektrum aller Negativaussagen für eine bestimmte Objektklasse mitbehandelt, um Performancevorteile zu erhalten. Zuerst wird jedes Trainingsbild durch eine Menge lokaler oder globaler Features repräsentiert, dann wird eine Entscheidungsgrenze zwischen Objekt- und Nichtobjektklassen generiert, indem ein Klassifikator trainiert wird (zum Beispiel neuronale Netze) oder indem die Wahrscheinlichkeitsverteilungen der Features einer jeden Klasse modelliert werden (zum Beispiel unter Benutzung der Bayesschen Regel).

In [64] wird die *Principal Component Analysis* (PCA) zur Featureextraktion genutzt und ein neuronales Netz zur Klassifikation eingesetzt. Alle Fahrzeuge werden in ein 20×20 Pixel Fenster skaliert, dieses wird in 25 einzelne 4×4 Fenster unterteilt und für ein jedes solches Fenster ein lokaler PCA Wert errechnet. Diese Werte dienen als Eingabe für ein neuronales Netz, um die Hypothese zu verifizieren. Andere Methoden nutzen Local Orientation Coding (LOC) [33] oder die Hausdorff Distanz [50], Expectation Maximization [86], Haar Wavelets, verschiedene statistische Modelle, Gabor Filter, Kanten, Linien, usw. Unzählige Papers sind im Bereich der erscheinungsbasierten Hypothesenverifizierung veröffentlicht worden. Hier steht dem Algorithmus im Vergleich zum templatebasierten Ansatz noch mehr a priori Wissen zur

Verfügung. Eine Klassifikation von Fahrzeugen ist heutzutage in kontrollierten Umgebungen auf Autobahnen sehr gut möglich. Problematisch wird es, wenn Fahrzeuge seitlich auftauchen und verschiedene Eigenbauten, Traktoren oder andere stationäre Hindernisse, welche der Klassifikator nicht als Hindernisse kennt, verifiziert werden sollen. In solchen *unkontrollierten* Umgebungen ist ein erscheinungsbasierter Ansatz nicht ausreichend.

2.4 Forschungsbereich

Hinderniserkennung im Straßenbereich unterscheidet sich von anderen Ansätzen im Bereich des Computer Vision in Bezug auf die Bekanntheit der Szene und der zu erkennenden Objekte. In Abbildung 2.9 wird dies besonders deutlich. In der Abbildung ist der Bekanntheitsgrad der Objekte und der Szene in einem Objekt-Szene Koordinatensystem, anlehnend an die Einordnung in [43], eingetragen. Die vier extremen Gebiete sind nach [43] Fertigung (zum Beispiel Qualitätskontrolle), Verkehrsschilderkennung, Innenraumüberwachung und Hinderniserkennung im Automobilbereich.

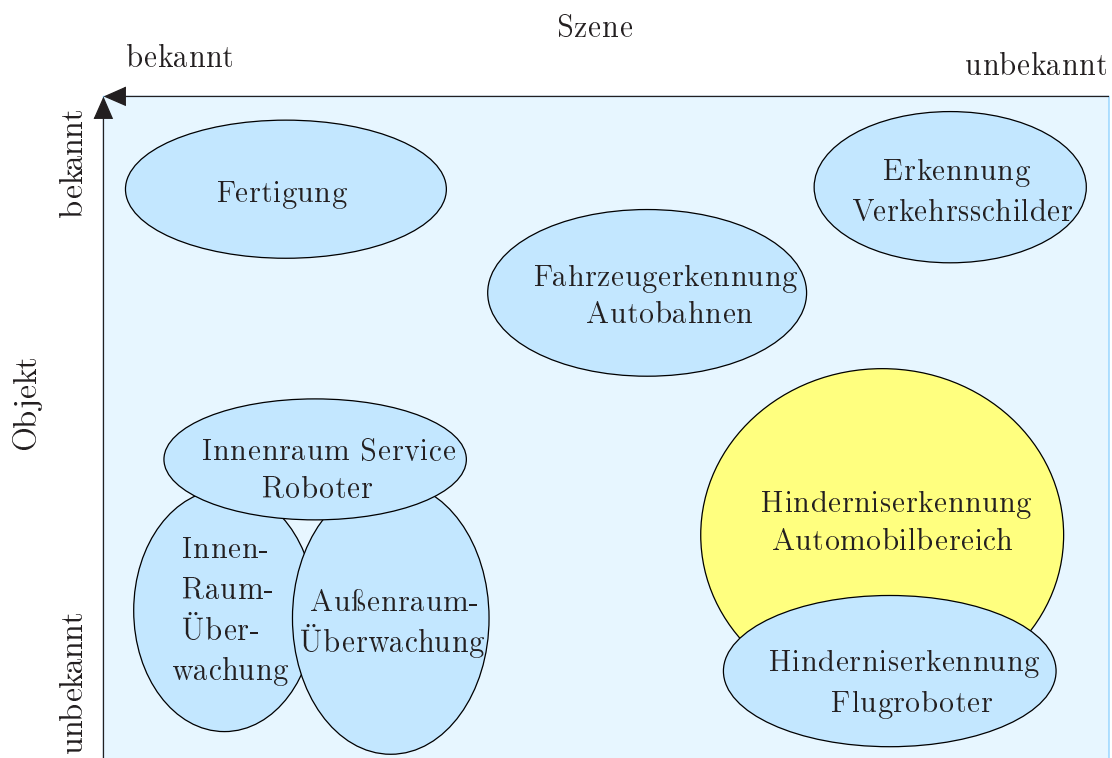


Abbildung 2.9: Verfahren der Computer Vision in einem Objekt-Szene Koordinatensystem

2.4.1 Herausforderungen

Obgleich sehr viel Mühe in das Gebiet der Hinderniserkennung gesteckt wurde, eine große Anzahl an Papers veröffentlicht wurde, viele neue Algorithmen und etliche Prototypen entwickelt wurden, ist es immer noch nicht gelungen, ein robustes und zuverlässiges autonomes System zu bauen. Hinderniserkennung ist so herausfordernd, dass keine der in diesem Kapitel genannten Methoden alleine eine absolut zuverlässige Aussage treffen kann. Für verschiedene Situationen müssen individuelle Lösungen realisiert werden. Im folgenden werden verschiedene Lösungsansätze genannt:

Sensorfusion Die Informationen eines einzelnen Sensors reichen beim aktuellen Stand der Technik nicht aus, ein zuverlässiges autonomes System zu bauen. Hier ist es notwendig, Systeme zu entwickeln, welche Informationen mehrerer Sensoren (sowohl aktiver als auch passiver) vereinen und auswerten. Ein Möglicher Ansatz wird in Kapitel 5.4 diskutiert.

Fehldetektion Verschiedene Sensoren haben jeweils ein gewisses Umfeld, in dem sie gute bzw. schlechte Ergebnisse liefern. Ein visueller Sensor beispielsweise liefert bei Regen und nachts schlechte Ergebnisse. Ein zuverlässiges System sollte in der Lage sein, selbst zu erkennen, wann es nicht mehr in der Lage ist, gewisse Anforderungen an Aussagengenauigkeit einzuhalten. Solch eine Information sollte von der Applikation abgefangen werden.

Hardwarefähigkeit Fahrzeugelektronik soll schnell und zuverlässig funktionieren. Unter anderem ist die Echtzeitfähigkeit ein wichtiger Schritt. Hier ist die Herausforderung, Algorithmen zu entwickeln, die auf automotive-tauglichen Steuergeräten in Echtzeit ausführbar sind. Des Weiteren sind die Kosten eines Systems zu beachten. Es ist kostentechnisch nicht realistisch, alle vorhandenen Sensoren in möglichst großer Stückzahl zu verbauen.

Kapitel 3

Grundlagen

In diesem Kapitel werden die nötigen Grundlagen erklärt, um die in dieser Ausarbeitung beschriebene bewegungsbasierte Hindernisdetektion durchzuführen. Zuerst wird auf das Kameramodell eingegangen. Dann werden das Szenenmodell und die Orientierung der Kamera zur Welt dargestellt. Anschließend folgt eine mathematische Einleitung zum optischen Fluss und zu Trackingverfahren, welche in dieser Arbeit eingesetzt werden. Auf alternative Trackingverfahren wird in dem Zusammenhang kurz verwiesen.

3.1 Kameramodell

Das Kameramodell beschreibt die Abbildung der dreidimensionalen Welt in ein zweidimensionales Bild. Als Kameramodell hat sich das Lochkamera-Modell bewährt. Dieses Modell der Zentralprojektion kann die Funktionsweise realer Kameras mit Linsensystem nicht exakt beschreiben, da radiale und andere Verzerrungen der Linse nicht berücksichtigt werden. Sie eignet sich aber trotzdem für die hier gezeigten mathematischen Betrachtungen.

Das Modell der Lochkamera (vgl. Abbildung 3.1) basiert auf der Funktionsweise der *camera obscura*¹, einer Kamera bestehend aus einer dunklen Kammer mit einem kleinen Loch an der Vorderseite. Befindet sich vor der Kamera ein Objekt, so dringen von diesem Objekt reflektierte oder emittierte Lichtstrahlen durch dieses Loch und erzeugen auf der Bildebene, der dem Loch gegenüberliegenden Seite, eine umgekehrte Abbildung dieses Objektes. Im Modell wird das Loch als punktförmig angenommen und als Projektionszentrum X_0 oder 0_K bezeichnet, da alle in die Kamera einfallenden Strahlen sich hier schneiden. Der Abstand der Bildebene vom Projektionszentrum f wird Brennweite genannt. Die Abbildung eines Raumstrahls ergibt sich aus dem Schnittpunkt des Sehstrahls, definiert als Gerade durch Raumpunkt und Projektionszentrum, und der Bildebene. Der Schnittpunkt der Gerade, welche senkrecht auf die Bildebene steht (optische Achse genannt) mit der Bildebene definiert den Hauptpunkt.

¹lat. camera - die Kammer, obscura - dunkel

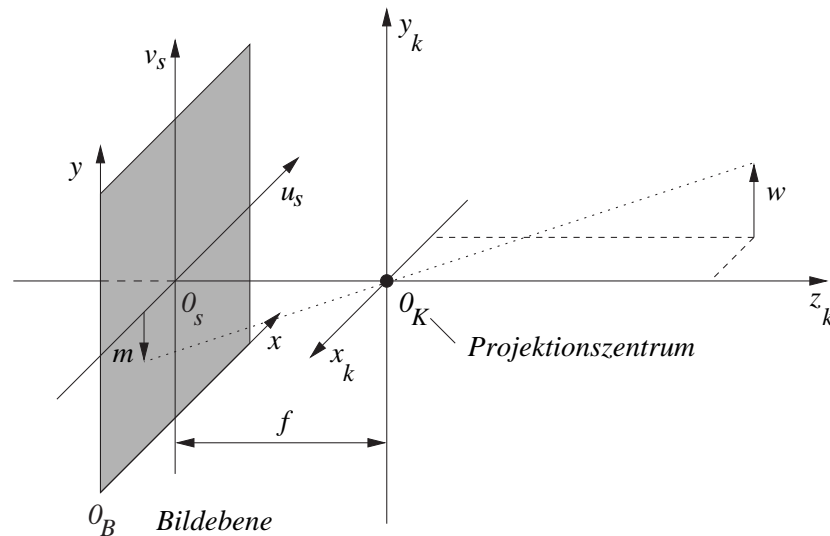


Abbildung 3.1: Lochkamera

3.2 Koordinatensysteme

Die folgenden Koordinatensysteme dienen in Anlehnung an [62, S. 19 f.] der mathematischen Beschreibung des Abbildungsvorgangs:

- Weltkoordinatensystem K_W
Das Weltkoordinatensystem $K_W(\mathbf{0}, X, Y, Z)$ ist ein dreidimensionales linkshändiges orthogonales Koordinatensystem, welches alle Raumpunkte mit Weltkoordinaten beinhaltet. Meistens ist es jedoch ausreichend, Weltpunkte im Kamerakoordinatensystem zu betrachten.
- Kamerakoordinatensystem K_K
Das Kamerakoordinatensystem $K_K(\mathbf{O}_K, X_K, Y_K, Z_K)$ ist auch ein dreidimensionales linkshändiges orthogonales Koordinatensystem welches seinen Ursprung \mathbf{O}_K im Projektionszentrum hat. Außerdem ist es gegenüber dem Weltkoordinatensystem gedreht. Seine Z -Achse steht senkrecht auf die Bildebene und wird auch optische Achse genannt.
- Sensorkoordinatensystem K_S
Das zweidimensionale orthogonale Koordinatensystem $K_S(\mathbf{O}_S, u_S, v_S)$ beschreibt die Sensor- oder Projektionsebene. Der Ursprung \mathbf{O}_S liegt im Hauptpunkt. Die Koordinatenachse u_S ist parallel zur X_K -Achse, zeigt jedoch in die entgegengesetzte Richtung. Die v_S -Achse ist parallel und gleichgerichtet zur Y_K -Achse.
- Bildkoordinatensystem K_B
Das Bildkoordinatensystem $K_B(\mathbf{O}_B, x, y)$ ist ein zweidimensionales diskretes orthogonales Koordinatensystem und liegt in der gleichen Ebene wie das Sensorkoordinatensystem.

Seine Koordinatenachsen sind parallel zu denen des Sensorkoordinatensystems. Es ist aber diesem gegenüber verschoben.

3.2.1 Beziehung zwischen Welt- und Kamerakoordinatensystem

Das Weltkoordinatensystem wird mittels Translation in den Kameraursprung oder das Projektionszentrum $\mathbf{X}_0 = (X_0, Y_0, Z_0)^T$ und Rotation in das Kamerakoordinatensystem transformiert. Für einen Punkt $\mathbf{P}_W = (X_W, Y_W, Z_W)^T$ im Weltkoordinatensystem ergibt sich der Punkt $\mathbf{P}_K = (X_K, Y_K, Z_K)^T$ im Kamerakoordinatensystem mittels

$$\begin{pmatrix} X_K \\ Y_K \\ Z_K \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X_W - X_0 \\ Y_W - Y_0 \\ Z_W - Z_0 \end{pmatrix}. \quad (3.1)$$

Nach Einführung homogener Koordinaten kann diese Beziehung auch geschrieben werden als

$$\begin{bmatrix} X_K \\ Y_K \\ Z_K \\ W_K \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (3.2)$$

3.2.2 Beziehung zwischen Kamera- und Sensorkoordinatensystem

Die Transformation eines Weltpunktes im Kamerakoordinatensystem in das Sensorkoordinatensystem beschreibt den Abbildungsvorgang der Kamera. Durch Anwendung des Strahlensatzes ergeben sich für einen Punkt $\mathbf{P}_K = (X_K, Y_K, Z_K)$ im Kamerakoordinatensystem und seiner Projektion $\mathbf{p}_S = (u_S, v_S)^T$ im Sensorkoordinatensystem folgende Beziehung:

$$\frac{X_K}{Z_K} = \frac{u_S}{f} \quad (3.3)$$

$$\frac{Y_K}{Z_K} = -\frac{v_S}{f}. \quad (3.4)$$

In homogenen Koordinaten geschrieben ergibt sich folgende Transformation:

$$\begin{bmatrix} u_S \\ v_S \\ w_S \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_K \\ Y_K \\ Z_K \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \begin{bmatrix} X_K \\ Y_K \\ Z_K \\ 1 \end{bmatrix}. \quad (3.5)$$

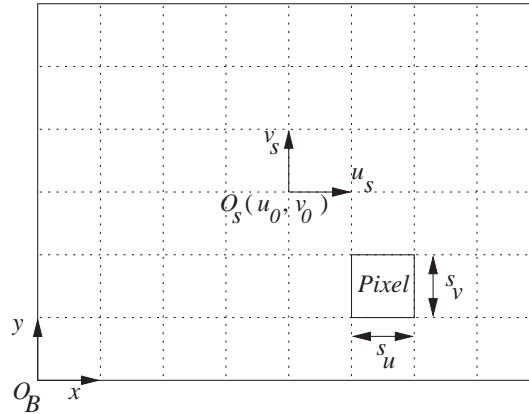


Abbildung 3.2: Beziehung zwischen Sensor- und Bildkoordinatensystem

3.2.3 Beziehung zwischen Sensor- und Bildkoordinatensystem

Der Zusammenhang zwischen diskretem Bildkoordinatensystem und Sensorkoordinatensystem ist in Abbildung 3.2 dargestellt. Ein Pixel ist ein Sensorelement auf dem Bildwandler, welches das auf die Fläche treffende Licht in ein digitales Signal umwandelt. Die Fläche eines Pixel wird durch die Längen s_u und s_v bestimmt. Der Mittelpunkt der gesamten Sensorfläche besitzt die Bildkoordinate u_0 und v_0 und ist Ursprung des Sensorkoordinatensystems. Die Beziehung zwischen Bild- und Sensorkoordinaten lautet somit:

$$x = \frac{u_S}{s_u} + u_0 \quad (3.6)$$

$$y = \frac{v_S}{s_v} + v_0 \quad (3.7)$$

3.2.4 Projektionsgleichung

Setzt man Gleichung 3.7 und 3.5 in Gleichung 3.2 ein, so erhält man die Beziehung zwischen einem Punkt \mathbf{X} im Weltkoordinatensystem und dem korrespondierenden Pixel \mathbf{x} im Bildkoordinatensystem:

$$\begin{aligned} \begin{bmatrix} x \\ y \\ h \end{bmatrix} &= \begin{bmatrix} \frac{1}{s_u} & 0 & u_0 \\ 0 & \frac{1}{s_v} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{f}{s_u} & 0 & u_0 \\ 0 & -\frac{f}{s_v} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \\ &= KR[I| -\mathbf{X}_0] \mathbf{X} \end{aligned} \quad (3.8)$$

Die Matrix $P = KR[I| - X_0]$ heißt Projektionsmatrix. Die Rotation R und Translation X_0 beschreiben die äußeren Parameter der Kamera (Transformation des Weltkoordinatensystems in das Kamerakoordinatensystem), die Matrix K beinhaltet die Parameter der inneren Orientierung. Damit ergeben sich die so genannten Kollinearitätsgleichungen für die Bildpunkte:

$$x = \frac{f}{s_u} \frac{r_{11}(X_W - X_0) + r_{12}(Y_W - Y_0) + r_{13}(Z_W - Z_0)}{r_{31}(X_W - X_0) + r_{32}(Y_W - Y_0) + r_{33}(Z_W - Z_0)} + u_0 \quad \text{und} \quad (3.9)$$

$$y = -\frac{f}{s_v} \frac{r_{21}(X_W - X_0) + r_{22}(Y_W - Y_0) + r_{23}(Z_W - Z_0)}{r_{31}(X_W - X_0) + r_{32}(Y_W - Y_0) + r_{33}(Z_W - Z_0)} + v_0. \quad (3.10)$$

3.2.5 Rektifizierung

Das Modell der Lochkamera ist lediglich eine Annäherung an die tatsächliche Kamera mit Objektiv. Die beim Abbildungsvorgang entstehenden Verzerrungen und Fehlerquellen sind kameraspezifisch und können im Labor hinreichend genau bestimmt werden. Das Herausrechnen dieser Anteile wird Rektifizierung genannt. Die Rektifizierung kann als Vorverarbeitungsschritt auf dem ganzen Bild durchgeführt werden. Alternativ können die Verzerrungen direkt als Funktion der Bildkoordinaten auf die rechte Seite von Gleichungen 3.9 und 3.10 addiert werden. Damit ergeben sich die allgemeinen Kollinearitätsgleichungen. In dieser Arbeit wird die Rektifizierung als Vorverarbeitungsschritt auf dem ganzen Bild berechnet.

3.3 Szenenmodell und Äußere Orientierung der Kamera

Wie in Kapitel 2.3.1.3 bereits erwähnt, ist eine Bestimmung der absoluten 3D-Koordinaten und der Geschwindigkeiten für bewegte Objekte aus reinen Flussdaten mathematisch nicht möglich. Um den bewegungsbasierten Ansatz für die Objekterkennung zu nutzen, müssen deshalb einschränkende Annahmen über die abgebildete Szene getroffen werden. In dieser Arbeit wird von einer stationären Welt ausgegangen. Des Weiteren wird die Annahme getroffen, alle Hindernisse seien durch einen umgebenden Quader (engl. *bounding box*) approximierbar und die Vorderseite dieser bounding box sei orthogonal sowohl zur X - Y -Ebene (Straßenebene) als auch zur Z - Y -Ebene des Weltkoordinatensystems. Damit ist die aufgrund von Bewegung in das Bild projizierte Transformation von Hindernissen mathematisch eindeutig interpretierbar. Für Hindernisse in großen Entfernungen sind solche Annahmen durchaus berechtigt, da die relativen Tiefenunterschiede zwischen Punkten auf der Hindernisoberfläche untereinander im Vergleich zu deren Abstand zur Kamera nur gering sind. Ein solches Modell wird auch *Bauklotzmodell* genannt.

Weiter wird von einer fest installierten Kamera mit bekannter äußerer Orientierung auf dem Versuchsträger ausgegangen. Die übliche Installation einer Kamera ist in 3.3 abgebildet. Der Versuchsträger - als einziges nicht statisches Objekt - bewegt sich annähernd parallel zur Z -Achse des Weltkoordinatensystems. Dabei ändern sich die Rotationsparameter der äußeren

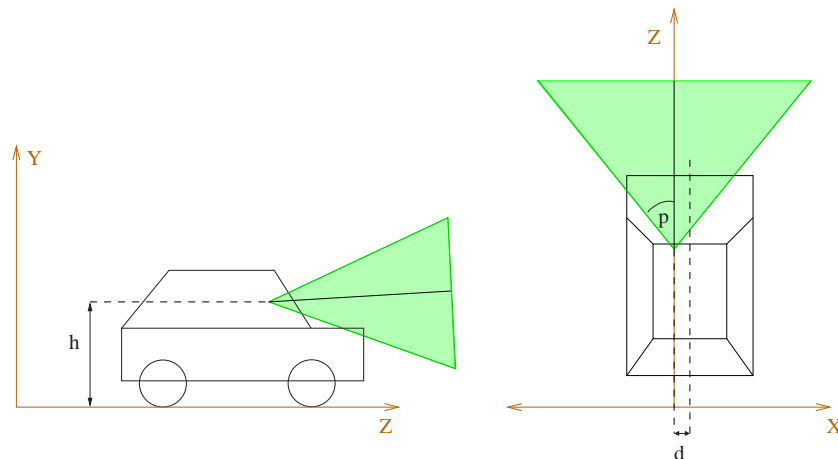


Abbildung 3.3: Aufbau des Versuchsträgers im Weltkoordinatensystem mit Kamerahöhe h , lateralem Versatz zur Fahrzeugmitte d und Öffnungswinkel der Kamera $2p$.

Orientierung nicht. In der Praxis ist solch eine Restriktion der Bewegungsparameter nicht haltbar. Rotationen um die X , Z und Y -Achse (Nick- (Ω_X), Wank- (Ω_Z) sowie Gierbewegungen (Ω_Y)) und Translationsparameter in X und Y treten auf und müssen bei der Modellierung mit berücksichtigt werden, um zuverlässige Ergebnisse zu erhalten.

3.4 Structure from Motion

Um den bewegungsbasierten Ansatz zur Hinderniserkennung, wie er in Kapitel 2.3.1.3 eingeführt wurde, besser zu verstehen, ist eine mathematische Betrachtung notwendig. Im folgenden Kapitel wird zunächst die zweidimensionale Projektion des dreidimensionalen Feldes von Geschwindigkeiten (das Bewegungsfeld) untersucht. Anhand dieser rein geometrischen Herangehensweise werden wesentliche Eigenschaften erläutert, welche für das Verständnis des optischen Flusses unerlässlich sind. Anschließend erfolgt eine Gegenüberstellung mit dem tatsächlich beobachtetem optischem Flussfeld. Das Verständnis des Flussfeldes ist Voraussetzung für alle bewegungsbasierten Objekterkennungsverfahren.

3.4.1 Herleitung des Bewegungsfeldes

Anlehnend an die Ausarbeitung von Longuet-Higgins und Prazdny [59] soll in diesem Kapitel das Bewegungsfeld eines Beobachters in einer starren Welt vorgestellt werden. Das Bewegungsfeld der starren Welt wird im Kamerakoordinatensystem, das abgebildete Bewegungsfeld im Sensorkoordinatensystem angegeben. Das dreidimensionale Feld $\omega(\mathbf{X}) \in \mathbb{R}^3$ beschreibt die Relativbewegung zwischen dem Beobachter und einem Punkt $\mathbf{X} = (X, Y, Z)^t$ im Raum zu einem bestimmten Zeitpunkt. Durch die Projektion dieses Feldes für alle abgebildeten Punkte auf

die Bildebene entsteht das so genannte Bewegungsfeld $\mathbf{v}(\mathbf{x}) \in \mathbb{R}^2$.

Die Bewegung eines Beobachters im Raum kann zu jedem Zeitpunkt zerlegt werden in eine Translation und eine Rotation. Hierbei ist auf die Reihenfolge der Zerlegung zu achten, da solch eine Zerlegung im Allgemeinen nicht eindeutig ist. Mit der Translationsgeschwindigkeit $\mathbf{T} = (T_X, T_Y, T_Z)^t$ und der Rotationsgeschwindigkeit $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)^t$ ergibt sich somit für das dreidimensionale Bewegungsfeld für alle stationären Punkte:

$$\boldsymbol{\omega}(\mathbf{X}) = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = -\mathbf{T} - \boldsymbol{\Omega} \times \mathbf{X}. \quad (3.11)$$

Der Translationsanteil ist über das gesamte Feld konstant, während der Rotationsanteil durch das Kreuzprodukt aus Rotationsgeschwindigkeit und Raumpunkt ortsabhängig ist. Betrachtet man einen Raumpunkt im Kamerakoordinatensystem, so ist die Projektion dieses Raumpunktes \mathbf{X} auf die Bildebene gegeben durch eine Zentralprojektion. Nach Gleichung 3.5 ergibt sich mit $f = \frac{f}{s_u} = \frac{f}{s_v}$:

$$\mathbf{x} = \begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X \\ -Y \end{pmatrix}, \quad (3.12)$$

also für die Geschwindigkeit

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \frac{d}{dt} \left(\frac{f}{Z} \begin{pmatrix} X \\ -Y \end{pmatrix} \right) = \frac{f}{Z^2} \begin{pmatrix} \dot{X}Z - X\dot{Z} \\ Y\dot{Z} - \dot{Y}Z \end{pmatrix}. \quad (3.13)$$

Setzt man Gleichung 3.11 in Gleichung 3.13 ein und teilt die Translations- und Rotationskomponenten, so ergibt sich das Bewegungsfeld als Überlagerungsfeld zweier Vektorfelder

$$\mathbf{v}(\mathbf{x}) = \mathbf{v}_t(\mathbf{x}) + \mathbf{v}_r(\mathbf{x}) \quad (3.14)$$

mit

$$\mathbf{v}_t(\mathbf{x}) = \frac{1}{Z} \begin{pmatrix} uT_Z - fT_X \\ vT_Z + fT_Y \end{pmatrix} \quad \text{und} \quad (3.15)$$

$$\mathbf{v}_r(\mathbf{x}) = \begin{pmatrix} -\Omega_X \frac{uv}{f} - \Omega_Y \left(f + \frac{u^2}{f} \right) - \Omega_Z v \\ -\Omega_X \left(f + \frac{v^2}{f} \right) - \Omega_Y \frac{uv}{f} + \Omega_Z u \end{pmatrix}. \quad (3.16)$$

Anhand dieser Gleichungen können wesentliche Eigenschaften des Bewegungsfeldes, welche zum Teil schon in 2.3.1.3 erwähnt wurden, nachgewiesen werden. Bei genauerer Betrachtung von Gleichung 3.15 fällt auf, dass der Translationsanteil des Bewegungsfeldes wegen der Division durch Z tiefenabhängig ist. Eine Translation im Raum führt für einen festen Bildpunkt zu

einer umso kleineren Bildbewegung, je größer die Entfernung zwischen dem Beobachter dem in den Bildpunkt projizierten Objekt ist. Dadurch lässt sich direkt die Tiefeninformation von Objekten auf dem Sehstrahl ableiten. Das Bewegungsfeld enthält also, falls eine Translation vorliegt, Informationen über die Geometrie der sichtbaren Objekte. Der Rotationsanteil 3.16 hingegen ist unabhängig von der Position im Raum, folglich lässt sich aus einer Rotation keine Tiefeninformation ableiten.

Ist die Translationsgeschwindigkeit allerdings nicht bekannt, zum Beispiel falls keine reine Eigenbewegung vorliegt (d.h. auch die betrachteten Objekte bewegen sich), so lässt sich die Geometrie der betrachteten Objekte nur bis auf einen konstanten Faktor bestimmen. Wenn man in Gleichung 3.15 sowohl Z als auch die Translation T_Z , T_Y und T_X mit dem gleichen Faktor multipliziert, so bleibt der Verschiebungsvektor gleich. Abbildung 3.4 verdeutlicht dies: Die gemessenen Geschwindigkeiten können durch unterschiedliche Raugeschwindigkeiten in der jeweiligen Entfernung verursacht werden. Dies bedeutet, dass zusätzliche Annahmen über Entfernung oder Geschwindigkeit der betrachteten Objekte getroffen werden müssen, um eindeutige Lösungen zu erhalten. In dieser Arbeit wird gefordert, dass sich der Beobachter in einer stationären Welt bewegt und die Eigenbewegung des Beobachters bekannt ist.

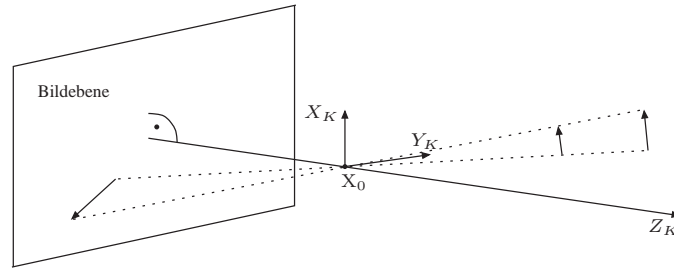


Abbildung 3.4: Mehrdeutigkeit des Bewegungsfeldes

Eine weitere Eigenschaft, die bereits in Kapitel 2.3.1.3 angesprochen wurde, lässt sich nun ableiten. Betrachtet man das Translationsfeld für $T_Z > 0$ und setzt Gleichung 3.15 gleich Null, so erhält man als Lösung den Bildpunkt:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{x}_{\text{FOE}} = \frac{f}{T_Z} \begin{pmatrix} T_X \\ -T_Y \end{pmatrix}. \quad (3.17)$$

Durch Umformen erhält man nun das Translationsfeld in Abhängigkeit vom Bildpunkt x :

$$\mathbf{v}_t(\mathbf{x}) = \frac{1}{Z} \begin{pmatrix} uT_Z - fT_X \\ vT_Z + fT_Y \end{pmatrix} = \frac{T_Z}{Z} \left(\begin{pmatrix} u \\ v \end{pmatrix} - \frac{f}{T_Z} \begin{pmatrix} T_X \\ -T_Y \end{pmatrix} \right) = \frac{T_Z}{Z} (\mathbf{x} - \mathbf{x}_{\text{FOE}}). \quad (3.18)$$

Es handelt sich also um ein lineares Vektorfeld. Alle Vektoren zeigen bei einer positiven Geschwindigkeit T_Z von dem Punkt \mathbf{x}_{FOE} weg, deshalb wird er auch *focus of expansion*, zu

Deutsch Expansionspunkt, genannt. Weiter erkennt man, dass die Länge der Vektoren neben der Tiefe des Objektes auf dem Sehstrahl auch von der Entfernung des Bildpunktes vom Expansionspunkt abhängt. Insbesondere in der Nähe des Expansionspunktes ist der Fluss nahezu Null. Ein exemplarisches, lediglich durch Eigentranslation erzeugtes Bewegungsfeld zeigt Abbildung 3.5. Es gibt verschiedene Verfahren zur Berechnung des optischen Flussfeldes. Darauf wird in dieser Arbeit nicht näher eingegangen, da lediglich die mathematische Betrachtung in dieser Arbeit notwendig ist. Näheres zur Berechnung ist zu finden in [1][55][10].

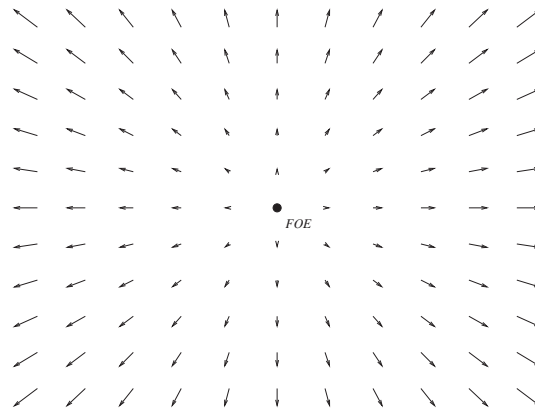


Abbildung 3.5: Bewegungsfeld mit Expansionspunkt

3.4.2 Bewegungsfeld und optisches Flussfeld

Das im Kapitel 3.4.1 hergeleitete Bewegungsfeld beschreibt ein Feld von Verschiebungsvektoren auf der Retina bzw. dem Bildsensor. Die Informationen für die Schätzung der Eigenbewegung und der Raumstruktur können allerdings nicht direkt gemessen werden, sondern nur die Änderung der Bildfunktion $I(\mathbf{x}, t)$, die zu jedem Zeitpunkt t jedem Bildpunkt $\mathbf{x} = (u, v)^t$ einen Helligkeitswert zuweist. Die Veränderung dieser Helligkeit bzw. die Änderung des Helligkeitsmusters kann als Vektorfeld angegeben werden und wird optischer Fluss genannt [42].

Das Bewegungsfeld und der optische Fluss müssen aber keineswegs notwendig gleich sein. Dies ist ein großes Problem bei der Ausnutzung des optischen Flusses zur Rekonstruktion der Geometrie. Horn gibt dazu ein anschauliches Beispiel [42]: Man betrachte eine Kugel mit einheitlicher Lambertscher Oberfläche, welche sich um die eigene Achse dreht. Durch die einheitliche Oberfläche ist trotz vorhandener Bewegung kein Fluss messbar. Bei der ruhenden Kugel und einer sich bewegenden Lichtquelle wird hingegen ein Fluss wahrgenommen (vgl. Abbildung 3.5). Schatten und Reflexionen sind weitere Beispiele für Probleme bei der Ausnutzung von optischem Fluss. Allerdings kann man davon ausgehen, dass insbesondere für Bilder mit viel Textur das Bewegungsfeld und das optische Flussfeld nahezu identisch sind, wie Verri und Poggio gezeigt haben [83].

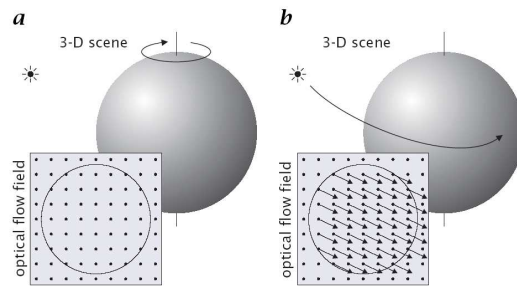


Abbildung 3.6: Verschiedenheit von optischem Fluss und Bewegungsfeld

3.5 Region Tracking

Die in dieser Ausarbeitung verwendeten Algorithmen nutzen die Verfolgung (engl. *Tracking*) von Bildmerkmalen. Dabei beschreibt eine vorgegebene Region im Ausgangsbild die für das Tracken erforderlichen Merkmale. Daraus ergibt sich der Fachausdruck *Region Tracking* (zu Deutsch: Verfolgen von Regionen). Es folgt ein Überblick über bestehende Techniken zum Tracking von Bildregionen. Aus den Anforderungen für einen geeigneten Algorithmus ergibt sich, welcher Ansatz für das in dieser Arbeit gestellte Problem der sinnvollste ist. Für diesen Ansatz erfolgt eine mathematische Herleitung.

3.5.1 Überblick über vorhandene Verfahren zur Merkmalsverfolgung

Das Verfolgen von Merkmalen im Bild bedingt das Wiederfinden eines Merkmales aus dem Bild $F(\mathbf{x}) = I(\mathbf{x}, t)$ im Bild $G(\mathbf{x}) = I(\mathbf{x}, t + \tau)$. Man spricht auch von Bildzuordnung (*image registration*). Dabei kann eine beliebige Transformation das Bild F in das Bild G überführen. Die Identifikation einer Zuordnung erfolgt über den Vergleich von Bildinhalten. Das Problem der Bildzuordnung ist ein Spezialfall des so genannten Zuordnungsproblems oder Korrespondenzproblems (engl. *matching*). Das Zuordnungs- oder Korrespondenzproblem besteht in folgender Aufgabe [29]:

Gegeben: Beschreibung zweier Objekte

Gesucht:

- Bestimmung einer möglichst guten Abbildung zwischen den Beschreibungen oder
- Identifizierung zuzuordnender Teile der Beschreibung

Bei den Objekten kann es sich etwa um Maschinen und Arbeiter handeln, die möglichst gute Produktivität erzielen sollen oder um Gruppen von Männern und Frauen, welche möglichst zur Zufriedenheit aller im Sinne einer Partnervermittlung zueinander vermittelt werden sollen. In der Graphentheorie ist dieses Problem daher auch als *Heiratsproblem* bekannt. Mit einer

geeigneten Beschreibung der Objekte und einer sinnvollen Kostenfunktion für die Zuordnung der Beschreibungen kann man dieses Problem mit Verfahren aus der Graphentheorie lösen. Im Spezialfall der Bildzuordnung haben sich vor allem vier verschiedene Ansätze etabliert, welche kurz vorgestellt werden: die Zuordnung basierend auf Merkmalsvektoren, Korrelation, Template Matching und Histogramm Matching.

3.5.1.1 Merkmalsvektoren

Die Zuordnung und die Auswahl von Merkmalen kann auf sehr verschiedene Art und Weise erfolgen. Merkmalsvektoren versuchen, lokale Bildinhalte möglichst gut zu kodieren, so dass sie mit entsprechend kodierten Merkmalsvektoren im zweiten Bild verglichen werden können. Komplexe Merkmalsvektoren wie SIFT (Scale Invariant Feature Transform) [60] oder PCA-SIFT [52] nutzen dabei Bildpyramiden der räumlichen Ableitungen (Gradienten) und sind unabhängig gegenüber kleinen Skalierungsänderungen. Beide Verfahren liefern subpixelgenaue Korrespondenzen, sind aber für die Echtzeitanwendung nicht geeignet und generieren in der Regel nur für deskriptive Bildpunkte Merkmale [87]. Unter Umständen sind nur wenige solcher Bildpunkte in der Referenzregion enthalten. Echtzeitfähige Verfahren liefern keine Subpixelgenauigkeit und können in die Klasse des Histogramm Matching eingeordnet werden.

3.5.1.2 Korrelation

Ein anderer Ansatz nutzt Korrelation [29], um Bildregionen in einem anderen Bild wieder zu finden. Das Prinzip der Korrelation basiert auf den folgenden zwei Annahmen (*brightness constancy assumption*):

- Die beiden Bilder unterscheiden sich geometrisch lediglich durch eine Translation.
- Die beiden Bilder unterscheiden sich radiometrisch lediglich in Helligkeit und Kontrast.

Es wird für jede in Frage kommende Verschiebung der Referenzregion im Zielbild ein so genannter Korrelationskoeffizient berechnet. Dieser liegt zwischen -1 (die verschobene Region ist genau entgegengesetzt korreliert) bis $+1$ (beiden Regionen stimmen überein). Exemplarisch sind in Tabelle 3.1 die Korrelationskoeffizienten für die in Abbildung 3.7 gezeigten Bildregionen angegeben. Typischerweise wird der Prozess des Findens einer korrespondierenden Zielregion in einem spiralförmigen Suchalgorithmus um einen Startwert ausgeführt und angehalten, sobald ein relatives Maximum des Korrelationskoeffizienten gefunden ist. Ist kein Startwert angegeben, muss das ganze Bild durchsucht werden, was sehr zeitaufwendig ist. Besser ist ein Verfahren, welches Bildpyramiden nutzt, also Bildsätze mit verschiedenen Auflösungsstufen, um schneller das Ergebnis zu finden (vgl. Abb. 3.8). Unterscheiden sich die Bilder geometrisch nicht nur durch eine Translation, so muss die Dimension des Suchraums um jeden zusätzlichen Parameter erhöht werden. Dadurch steigt der Aufwand exponentiell und die Verarbeitung in Echtzeit kann nicht garantiert werden.



Abbildung 3.7: Oben die zwei Bilder mit markierten Regionen, welche unterhalb der Bilder vergrößert dargestellt sind. Die Berechnung der Korrelationskoeffizienten für die 5×5 möglichen Matches sind in Tabelle 3.1 abgegeben [29].

	1	2	3	4	5
1	0.9639	-0.3994	-0.1627	-0.3868	0.1914
2	-0.0533	0.7503	-0.4677	0.5115	0.7193
3	-0.1826	-0.3905	0.7730	0.1475	-0.7457
4	-0.2724	0.4878	0.1640	0.7862	0.2077
5	0.0835	0.5044	-0.4541	0.2802	0.9876

Tabelle 3.1: Korrelationskoeffizienten für die Fensterausschnitte in Abbildung 3.7 nach [29].

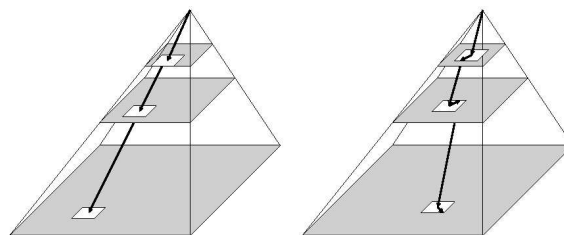


Abbildung 3.8: Rechts Bildpyramide des Ausgangsbildes; die verschiedenen Levels entsprechen unterschiedlichen Auflösungsstufen; weiß die Referenzregion. Links Bildpyramide des Zielbildes; die Suche einer zur Referenzregion korrespondierenden Region beginnt auf der höchsten Stufe (niedrigste Auflösung). Das Ergebnis dient als Anfangswert der Suche auf der nächsten Pyramidenstufe [29].

3.5.1.3 Histogramm Matching

Zwei weitere Klassen von Verfahren werden in [18] untersucht. Die erste Klasse, das Histogramm Matching, identifiziert das Ziel als Bildregion $\mathbf{r}(\boldsymbol{\mu}(t))$, wobei $\boldsymbol{\mu}(t)$, auch Status genannt, die variablen Parameter der Region beschreibt. Ein Beispiel für eine Region $\mathbf{r}(\boldsymbol{\mu}(t))$ ist ein Rechteck fester Größe. Der Status der Region ist der Mittelpunkt des Rechtecks in Bildkoordinaten, $\boldsymbol{\mu}(t) = (m_x(t), m_y(t))^t$, für jeden Zeitpunkt t . Mit diesem simplen Modell kann jede Translation beschrieben werden, indem der Schwerpunkt des Rechtecks, $\boldsymbol{\mu}(t)$, geschätzt wird. Wenn auch noch weitere Transformationen, wie zum Beispiel die Größe des Rechtecks, geschätzt werden sollen, wird der Statusvektor entsprechend erweitert.

Die in der Region enthaltene Information (beispielsweise Farbinformation, Intensitäten der Grauwerte oder Gradienten) wird benutzt, um das (bewegte) Objekt zu modellieren. Allerdings werden nicht einzelne Pixel und deren Grauwerten, sondern die Verteilung der Features, welche jedem Pixel zugeordnet sind, genutzt. Zu jedem Zeitpunkt t und für jeden Status $\boldsymbol{\mu}(t)$ besteht die Repräsentation des bewegten Objektes aus einer Wahrscheinlichkeitsverteilungsfunktion $p(\boldsymbol{\mu}(t))$, welche von den ausgewählten Features in der Region $\mathbf{r}(\boldsymbol{\mu}(t))$ abhängt. In der Praxis muss solch eine Verteilungsfunktion aus den Bilddaten geschätzt werden. Für Performancezwecke wird ein gewichtetes Histogramm $\mathbf{q}(\boldsymbol{\mu}(t)) = (q_1(\boldsymbol{\mu}(t)), q_2(\boldsymbol{\mu}(t)), \dots, q_N(\boldsymbol{\mu}(t)))^t$ mit N Kästen als Schätzung der wahren Verteilung genutzt. Jeder einzelne Kasten $q_i(\boldsymbol{\mu}(t))$ des Histogramms wird berechnet als

$$q_i(\boldsymbol{\mu}(t)) = C_{\boldsymbol{\mu}(t)} \sum_{u \in \mathbf{r}(\boldsymbol{\mu}(t))} L_{\boldsymbol{\mu}(t)}(u) \delta(b_t(u) - i), \quad i = 1, \dots, N \quad (3.19)$$

mit einer Gewichtsfunktion $L_{\boldsymbol{\mu}(t)}(u)$, der Funktion $b_t(u)$, welche das Pixel u dem korrespondierenden Kasten $j \in \{1, \dots, N\}$ zuordnet und der Kronecker Funktion δ . Der Wert $C_{\boldsymbol{\mu}(t)} = \frac{1}{\sum_{u \in \mathbf{r}(\boldsymbol{\mu}(t))} L_{\boldsymbol{\mu}(t)}(u)}$ ist eine Konstante zur Normalisierung. Mit anderen Worten werden in 3.19 gewichtet alle Vorkommen der Pixel gezählt, welche in den Kasten i fallen. Das Tracken des Objekts kann nun als ein Optimierungsproblem definiert werden. Mit einer vordefinierten Startregion zum Zeitpunkt $t = t_0$ wird ein initiales Histogramm $\mathbf{q}(\boldsymbol{\mu}(t_0))$ berechnet. Für jedes $t > t_0$ ist die korrespondierende Region definiert durch

$$\boldsymbol{\mu}(t) = \arg \min_{\boldsymbol{\mu}} D(\mathbf{q}(\boldsymbol{\mu}(t_0)), \mathbf{q}(\boldsymbol{\mu}(t))), \quad (3.20)$$

wobei $D(\cdot, \cdot)$ eine passende Distanzfunktion für Histogramme ist. Verschiedene Arbeiten im Bereich des Histogramm Matching wurden angefertigt. Dabei gibt es sowohl Ansätze, welche ein lokales Optimum berechnen und somit den Suchraum eingrenzen [13][58], als auch Ansätze, welche ein globales Optimum berechnen. Einer der letztgenannten Ansätze basierend auf Partikel Filter wurde von Perez et al. vorgeschlagen [69] (vgl. auch Abbildung 3.9). Der Vorteil eines Histogramm basierten Ansatzes ist es, *verloren gegangene* Regionen (zum Beispiel we-

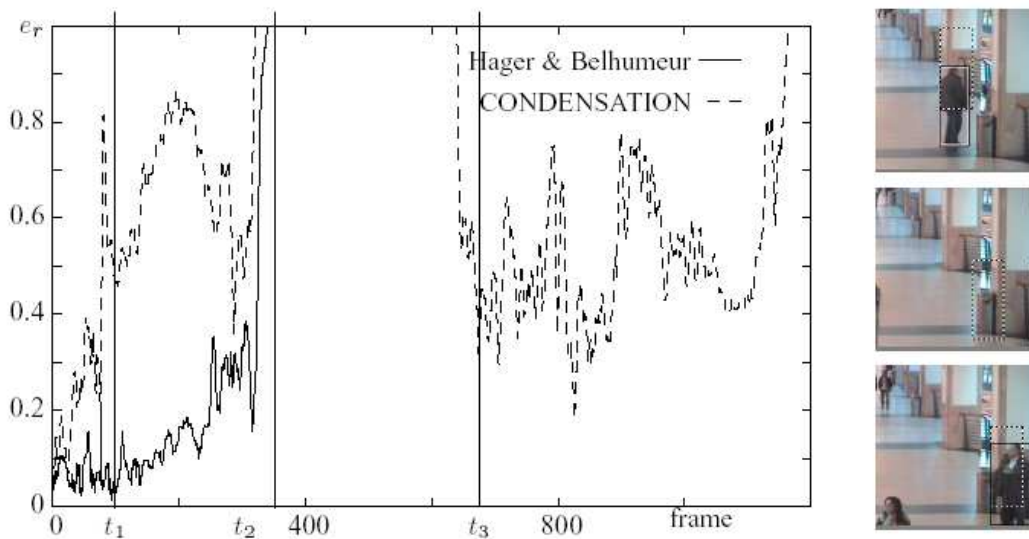


Abbildung 3.9: Vergleich der Hager [35] und CONDENSATION [69] Tracker; das schwarze Rechteck repräsentiert die wahren Werte, das weiße Rechteck zeigt den Hager Tracker, die gestrichelte Linie den CONDENSATION Tracker. Das obere, mittlere und untere Bild zeigt die Frames zum Zeitpunkt t_1 , t_2 und t_3 . Die zu trackende Person verlässt das Kamerablickfeld und erscheint erneut am linken Bildrand. Der Hager Tracker ist genauer, verliert die Person aber, während der CONDENSATION Tracker es schafft, die Person erneut zu tracken. Das aufgetragene Fehlermaß ist definiert als

$$e_r(A, B) = \frac{|A \setminus B| + |B \setminus A|}{|A| + |B|} \in [0, 1],$$

wobei A und B für die Menge der Bildpunkte der getrackten und wahren Bildregion stehen und $|\cdot|$ die Kardinalzahl einer Menge ist [18].

gen Verdeckung) wieder finden zu können. Der Nachteil ist, dass die Ergebnisse aufgrund des diskretisierten Histogrammansatzes nicht so exakt wie möglich berechnet werden können [18]. Histogrammbasierte Methoden können ihre vollen Vorteil ausspielen, wenn Objekte getrackt werden, welche sich mit der Zeit deformieren, wie zum Beispiel Personen, die sich bewegen. In statischen Welten kommt der Nachteil der geringen Exaktheit des Ergebnisses zu tragen.

3.5.1.4 Template Matching

Die andere Klasse der in [18] untersuchten Verfahren nutzt das Verfolgen basierend auf Bildtemplates. Während der Initialisierung werden die Intensitätswerte der Bildfunktion zum Zeitpunkt t_0 für eine bestimmte Referenzregion $\mathbf{r} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^t$ im Referenzbild ausgewählt.

Diese Werte bilden das so genannte Template. Ziel ist es, diese Referenzregion geometrisch und radiometrisch so zu transformieren, dass sie mit der durch die Transformation vorgegebenen Region im Zielbild möglichst gut übereinstimmt. Als zu minimierende Kostenfunktion wird meistens die Summe der quadratischen Grauwertunterschiede zwischen den Werten des Referenztemplates und der deformierten Referenzregion im aktuellen Bild zum Zeitpunkt t gewählt. Daher nennt man solche Verfahren auch *Least Squares Matching*. Eine vollständige Suche im Raum aller möglichen Transformationen ist zu zeitintensiv. Effizienter sind Ansätze, welche die Veränderung der Transformationsparameter mittels eines linearen Systems approximieren. Dies kann entweder durch trainieren mit einer großen Zahl von Transformationen geschehen (Hyperplane Ansatz [47]) oder auf eine analytischere Art und Weise unter Ausnutzung von Gradienteninformation (Linearisierung mit Jacobimatrix), wie beispielsweise in [35]. In ihrem Vergleich haben Deutsch et al. [18] beide Verfahren untersucht. Dabei hat der analytische Ansatz von Hager et al. sich als stabiler und wesentlich schneller erwiesen. In dem von Hager et al. vorgestellten Ansatz wird die Deformation vom Zeitpunkt t_0 zum jeweils aktuellen Zeitpunkt t geschätzt. Dadurch muss das Referenztemplate nur einmal initialisiert werden. Andere Ansätze, welche Gradienteninformation nutzen, schätzen nur die Transformation zweier aufeinanderfolgender Bilder (Zeitpunkt t zu $t + 1$). Dies hat den Nachteil, dass Gradienten für jedes Bild berechnet werden müssen und Transformationsfehler sich akkumulieren. Um dieses Problem zu umgehen werden mehrere Referenztemplates zu vergangenen Zeitpunkten für die Schätzung genutzt. Ein solcher Ansatz ist zu zeitaufwendig für Echtzeitanwendungen. Templatebasierte Verfahren liefern sehr exakte Ergebnisse, tendieren aber dazu, die richtige Lösung schneller zu verlieren als Histogramm basierte Methoden (vgl. Abbildung 3.9). Letztendlich ist aber die Genauigkeit entscheidend. Ein weiterer Vorteil ist, dass Ungenauigkeiten für die geschätzten Transformationsparameter angegeben werden können (vgl. Abschnitt 3.5.6.2).

3.5.1.5 Auswahl des Algorithmus

Aus den Anforderungen Echtzeit, Genauigkeit, Freiheit in der Wahl der Position und Größe der Referenzregion sowie basierend auf dem in [18] durchgeführten Vergleich ist unter den gängigen Ansätzen in der Bildverarbeitung der Template basierte Ansatz allen anderen vorzuziehen. Aufgrund geringer Rechenzeit und Vermeidung der Aufintegration von Schätzfehlern bietet sich der Ansatz nach Hager et al. [35] an. Bezahlen muss man diese Vorteile dadurch, dass ein solches Trackingverfahren nur dann funktioniert, wenn die Veränderung der Referenzregion hinreichend gut modelliert wird. Veränderungen und Deformationen des zu trackenden Objektes sowie Verdeckungen und Helligkeitsunterschiede, welche nicht modelliert werden (zum Beispiel Schatten), führen dazu, dass das zu trackende Objekt verloren geht. Wünschenswert wäre es, wenn der Algorithmus selbständig einen solchen Fall meldet. Im folgenden Kapitel soll der ausgewählte Algorithmus näher beschrieben werden.

3.5.2 Mathematische Herleitung des Region Tracking

Der in diesem Kapitel vorgestellte Tracking Algorithmus basiert auf Minimierung der Summe der quadratischen Grauwertdifferenzen (engl. sum of squared differences, SSD) zwischen zwei Bildregionen. Dabei ist die geometrische Transformation durch ein beliebiges parametrisiertes Bewegungsmodell $f(\mathbf{x}, \mu)$ gegeben. Im nächsten Kapitel wird gezeigt, wie auch die radiometrischen Größen (Kontrast und Helligkeit) mitberücksichtigt werden können. Das Problem wird in ein linearisiertes Optimierungsproblem umgewandelt und erlaubt damit Rechenzeiten, die eine Echtzeitanwendung erlauben. Der Algorithmus wurde von Hager et al. in [35] detailliert hergeleitet.

Sei $I(\mathbf{x}, t)$ der Grauwert eines Bildes zum Zeitpunkt t an der Position $\mathbf{x} = (x, y)^t$ und $\nabla_x I(\mathbf{x}, t)$ die räumliche Ableitung der Bildfunktion. Das Bild zum Zeitpunkt t_0 sei das Referenzbild (Initialisierung des Algorithmus). Die Menge $R = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ sei die Menge der N Bildpunkte, welche die Zielregion bilden. Die Grauwerte dieser Zielregion zum Zeitpunkt t_0 bilden das Referenz Template.

Eine relative Bewegung zwischen Objekt und Kamera resultiert in einer Veränderung der Zielregion. Die Transformation der Zielregion wird beschrieben durch ein parametrisiertes Bewegungsmodell als $f(\mathbf{x}, \mu)$ mit $\mu = (\mu_1, \mu_2, \dots, \mu_n)^t$, $n < N$ und $f(\mathbf{x}, 0) = \mathbf{x}$. Die Funktion f sei in \mathbf{x} und μ ableitbar. Das Ziel ist eine Schätzung des Parametervektors μ zu jedem Zeitpunkt t . Der tatsächliche Wert des Parametervektors zum Zeitpunkt t ist $\mu^*(t)$, die Schätzung $\mu(t)$. Falls alle Veränderungen der Zielregion durch f beschrieben werden (speziell keine Helligkeits- und Kontraständerung), so gilt:

$$\forall t > t_0 \quad \exists \mu^*(t) \quad \forall \mathbf{x} \in R: \quad I(\mathbf{x}, t_0) = I(f(\mathbf{x}, \mu^*(t)), t). \quad (3.21)$$

Dies ist eine Generalisierung der so genannten *brightness constancy assumption* [42]. Der Bewegungsparametervektor μ der Zielregion kann zum Zeitpunkt t geschätzt werden, indem man folgendes Least Squares Problem minimiert:

$$O(\mu) = \sum_{\mathbf{x} \in R} (I(f(\mathbf{x}, \mu), t) - I(\mathbf{x}, t_0))^2. \quad (3.22)$$

In Vektorschreibweise mit

$$I(\mu, t) := \begin{pmatrix} I(f(\mathbf{x}_1, \mu), t) \\ I(f(\mathbf{x}_2, \mu), t) \\ \vdots \\ I(f(\mathbf{x}_N, \mu), t) \end{pmatrix} \quad (3.23)$$

erhält man vereinfacht für die Gleichungen 3.21 und 3.22:

$$\mathbf{I}(\boldsymbol{\mu}^*(t), t) = \mathbf{I}(0, t_0) \quad \text{und} \quad (3.24)$$

$$O(\boldsymbol{\mu}) = \|\mathbf{I}(\boldsymbol{\mu}, t) - \mathbf{I}(0, t_0)\|^2. \quad (3.25)$$

Allgemein ist Gleichung 3.22 eine nichtkonvexe Zielfunktion und somit (falls kein genauer Startpunkt bekannt ist) sehr schwierig und damit zeitaufwendig zu lösen. Im Spezialfall des visuellen Trackings kann man allerdings von einer kontinuierlichen Bewegung ausgehen und somit einen guten Startpunkt in der Nähe der erwarteten Parameter erhalten. Unter der Annahme, dass die Veränderung der Transformationsparameter kontinuierlich ist, insbesondere $\boldsymbol{\mu}(t + \tau) = \boldsymbol{\mu}(t) + \delta\boldsymbol{\mu}$, kann die Zielfunktion auch umformuliert werden zu

$$O(\delta\boldsymbol{\mu}) = \|\mathbf{I}(\boldsymbol{\mu}(t) + \delta\boldsymbol{\mu}, t + \tau) - \mathbf{I}(0, t_0)\|^2 \quad (3.26)$$

$$\boldsymbol{\mu}(t + \tau) = \boldsymbol{\mu}(t) + \delta\boldsymbol{\mu}$$

und bekannte Methoden der kontinuierlichen Optimierungstheorie können angewandt werden. Mit den partiellen Ableitungen nach der Zeit $\mathbf{I}_t(\boldsymbol{\mu}, t)$ und der Jacobimatrix $M(\boldsymbol{\mu}, t) = (I_{\mu_j}(f(\mathbf{x}_i, \boldsymbol{\mu}), t))_{ij}$ ergibt sich folgende Taylorreihenentwicklung:

$$\mathbf{I}(\boldsymbol{\mu} + \delta\boldsymbol{\mu}, t + \tau) = \mathbf{I}(\boldsymbol{\mu}, t) + M(\boldsymbol{\mu}, t)\delta\boldsymbol{\mu} + \tau\mathbf{I}_t(\boldsymbol{\mu}, t) + \dots \quad (3.27)$$

Mit der zusätzlichen Approximierung $\tau\mathbf{I}_t(\boldsymbol{\mu}, t) \approx \mathbf{I}(\boldsymbol{\mu}, t + \tau) - \mathbf{I}(\boldsymbol{\mu}, t)$ lässt sich 3.26 nun umformen zu

$$O(\delta\boldsymbol{\mu}) \approx \|M\delta\boldsymbol{\mu} + \underbrace{\mathbf{I}(\boldsymbol{\mu}, t + \tau) - \mathbf{I}(0, t_0)}_{=: \mathbf{e}(t+\tau)}\|^2. \quad (3.28)$$

Die Lösung des Minimierungsproblems kann unter der Voraussetzung, dass $M^t M$ invertierbar ist berechnet werden:

$$\begin{aligned} \min_{\delta\boldsymbol{\mu}} O(\delta\boldsymbol{\mu}) &\Rightarrow \nabla O(\delta\boldsymbol{\mu}) = 0 = \|M\delta\boldsymbol{\mu} + \mathbf{e}(t + \tau)\|^2. \\ &\Leftrightarrow 0 = M\delta\boldsymbol{\mu} + \mathbf{e}(t + \tau) \\ &\Leftrightarrow -M^t \mathbf{e}(t + \tau) = M^t M \delta\boldsymbol{\mu} \\ &\Leftrightarrow -(M^t M)^{-1} M^t \mathbf{e}(t + \tau) = \delta\boldsymbol{\mu}. \end{aligned} \quad (3.29)$$

Um die Zielregion zu tracken muss die Jacobimatrix

$$M(\boldsymbol{\mu}, t) = (I_{\mu_j}(f(\mathbf{x}_i, \boldsymbol{\mu}), t))_{ij} = (\nabla_f I(f(\mathbf{x}_i, \boldsymbol{\mu}), t)^t f_{\mu_j}(\mathbf{x}_i, \boldsymbol{\mu}))_{ij} \quad (3.30)$$

berechnet werden. Mit den Jacobimatrizen der Transformation f ,

$$f_\mu(\mathbf{x}, \mu) = \left[\frac{\partial f(\mathbf{x}, \mu)}{\partial \mu_1} \mid \frac{\partial f(\mathbf{x}, \mu)}{\partial \mu_2} \mid \dots \mid \frac{\partial f(\mathbf{x}, \mu)}{\partial \mu_n} \right] \text{ und } f_x(\mathbf{x}, \mu) = \left[\frac{\partial f(\mathbf{x}, \mu)}{\partial x} \mid \frac{\partial f(\mathbf{x}, \mu)}{\partial y} \right] \quad (3.31)$$

und der Ableitung beider Seiten von 3.21 unter der Annahme $\mu(t) = \mu^*(t)$,

$$\nabla_x I(\mathbf{x}, t_0) = f_x(\mathbf{x}, \mu)^t \nabla_f I(f(\mathbf{x}, \mu), t), \quad (3.32)$$

kann die Matrix M auch geschrieben werden als:

$$M(\mu) = \begin{bmatrix} \nabla_x I(\mathbf{x}_1, t_0)^t f_x(\mathbf{x}_1, \mu)^{-1} f_\mu(\mathbf{x}_1, \mu) \\ \nabla_x I(\mathbf{x}_2, t_0)^t f_x(\mathbf{x}_2, \mu)^{-1} f_\mu(\mathbf{x}_2, \mu) \\ \vdots \\ \nabla_x I(\mathbf{x}_N, t_0)^t f_x(\mathbf{x}_N, \mu)^{-1} f_\mu(\mathbf{x}_N, \mu) \end{bmatrix}. \quad (3.33)$$

Da für jede Art von Transformation des Bildes der Gradient lediglich zum Zeitpunkt t_0 berechnet werden muss, ist M nicht mehr abhängig von t , sondern nur noch von den Bewegungsparametern μ . Dies ist nicht verwunderlich, da jegliche Art von Transformation des Bildes zum Zeitpunkt $t > t_0$ eine Deformation der Zielregion zum Zeitpunkt t_0 ist und somit auch die Gradienten eine Deformation der Gradienten zum Zeitpunkt t_0 sind. Gelingt es nun weiterhin, die Matrix $f_x(\mathbf{x}_i, \mu)^{-1} f_\mu(\mathbf{x}_i, \mu)$ in eine $2 \times k$ Matrix $\Gamma(\mathbf{x})$ und eine $k \times n$ Matrix $\Sigma(\mu)$ zu faktorisieren (die Beschreibung der Zerlegung erfolgt in den jeweiligen Anwendungen), so kann die Matrix M größtenteils bereits zur Initialisierung, also *Offline* vorberechnet werden:

$$M(\mu) = \begin{bmatrix} \nabla_x I(\mathbf{x}_1, t_0)^t \Gamma(\mathbf{x}_1) \\ \nabla_x I(\mathbf{x}_2, t_0)^t \Gamma(\mathbf{x}_2) \\ \vdots \\ \nabla_x I(\mathbf{x}_N, t_0)^t \Gamma(\mathbf{x}_N) \end{bmatrix} \Sigma(\mu) = M_0 \Sigma(\mu). \quad (3.34)$$

Damit ergibt sich eingesetzt in Gleichung 3.29 ein sehr effizienter Tracking Algorithmus. Aufgrund der durchgeführten Linearisierung wird mit den verbesserten Parametern $\mu + \delta\mu$ als Eingabe eine erneute Ausgleiche gerechnet bis die Norm des Fehlervektors unter einen definierten Schwellwert fällt oder die Anzahl der Iterationen ihr Maximum erreicht hat. Man spricht daher von einem iterativen Verfahren.

3.5.3 Konvergenz und Implementation

In [61] wird für eine reine Translation einer eindimensionalen Bildfunktion $F(x) = \sin(x)$ gezeigt, dass obiges Verfahren für Anfangsverschiebungen bis zu einer halben Wellenlänge konvergiert. Dies legt den Schluss nahe, dass der Bereich, in dem das Verfahren korrekt kon-

vergiert, erweitert werden kann, indem man hohe Frequenzen herausfiltert, zum Beispiel durch Glätten des Bildes. Hager et al. [35] schlägt vor, eine Auflösungspyramide zu verwenden (vgl. Abbildung 3.8). In der vorliegenden Ausarbeitung wird der Algorithmus auf einer Gauß'schen Auflösungspyramide berechnet. Aus Konditionsgründen wird dabei der Ursprung der Bildkoordinaten in den Mittelpunkt der zu trackenden Region verschoben. Die Gradienten werden mit dem Sobel-Operator berechnet. Die Filterkerne sehen dabei wie folgt aus:

$$\text{Gauß: } \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad S_x : \frac{1}{8} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \text{und } S_y : \frac{1}{8} \begin{bmatrix} 1 & 2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (3.35)$$

3.5.4 Modellierung der Helligkeitsunterschiede

Die *brightness constancy equation* (Gleichung 3.21) geht davon aus, dass keine Helligkeits- und Kontrastunterschiede beim Tracken auftreten. In der Realität ist solch eine Annahme, vor allem über längere Zeitabstände, nicht haltbar. Um trotzdem eine möglichst lange und möglichst genaue Trackdauer zu gewährleisten ist es deshalb notwendig, Helligkeitsunterschiede mit in die Modellierung einzubeziehen. Hager et al. [35] betrachtet das Templatebild unter verschiedenen Beleuchtungen und generiert eine Beleuchtungsbasis um die *Bidirectional Reflectance Distribution Function* (BRDF) zu approximieren. Andere Ansätze versuchen lediglich Lambertsche Strahlung zu modellieren und erhalten ein vereinfachtes Beleuchtungsmodell, welches eine Spezialform des Ansatzes von Hager ist für den Fall, dass keine Aufnahmen des Objekt unter verschiedenen Beleuchtungen vorhanden sind.

Die folgende Herleitung des Beleuchtungsmodells ist auch in [46] zu finden. Gegeben sei eine Lichtquelle L im 3-D Raum und eine glatte Oberfläche S auf dem betrachteten Objekt. Wie in Abbildung 3.10 dargestellt, hängt die Helligkeit der Grauwerte eines jeden Bildpunktes

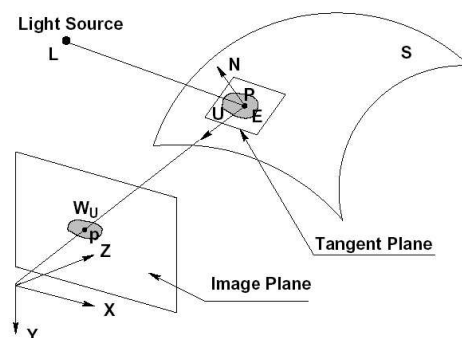


Abbildung 3.10: Beleuchtungsmodell für Lambertsche Oberflächen

vom einfallenden Licht, welches von der Lichtquelle L ausgestrahlt und von der Oberfläche S reflektiert wird, ab. Für den Fall, dass die Lichtquelle von der Oberfläche weit entfernt ist, sind

die darauf einfallenden Lichtstrahlen nahezu parallel. Unter der zusätzlichen Annahme, dass der Blickwinkel und die Normale für jeden Punkt der Oberfläche innerhalb einer Umgebung U eines Punktes $\mathbf{P} \in S$ konstant sind (insbesondere ist die Oberfläche planar in U), ergibt sich für Lambertsche Oberflächen die vereinfachte BRDF

$$I(\mathbf{x}) = \lambda_E E(\mathbf{X}) \quad \forall \mathbf{X} \in U, \mathbf{x} \in W(U) \quad (3.36)$$

mit der Albedofunktion $E(\mathbf{X})$, der Projektion $\mathbf{x} = W(\mathbf{X})$ eines Weltpunktes \mathbf{X} in den Bildpunkt \mathbf{x} und der konstanten λ_E abhängig vom Einfallswinkel der Lichtstrahlen auf die Oberfläche. Um zusätzliche Effekte, wie zum Beispiel die sich ändernden Reflektionen benachbarter Objekte auf die Oberfläche zu behandeln, ist es notwendig, einen additiven Term in die Gleichung 3.36 einzubauen. Das Beleuchtungsmodell lautet mit der konstanten γ_E :

$$I(\mathbf{x}) = \lambda_E E(\mathbf{X}) + \gamma_E \quad \forall \mathbf{X} \in U, \mathbf{x} \in W(U). \quad (3.37)$$

λ_E und γ_E können als Parameter betrachtet werden, welche den Kontrast und die Helligkeit der Bildregion modellieren. Wenn entweder die Szene oder die Kamera sich bewegen, verändern sich diese Parameter. Damit lässt sich das Modell für die Beleuchtungsänderung aufstellen:

$$I(\mathbf{x}, t_0) = \lambda(t)I(\mathbf{x}, t) + \gamma(t) \quad \forall \mathbf{x} \in W(U) \quad (3.38)$$

Kombiniert man sowohl die geometrischen als auch die radiometrischen Transformationen einer zu trackenden Bildregion, so erhält man für Gleichungen 3.21 und 3.22:

$$\forall t > t_0 \quad \exists \mu^*(t), \lambda^*(t), \gamma^*(t) \quad \forall \mathbf{x} \in R: \quad I(\mathbf{x}, t_0) = \lambda^*(t)I(f(\mathbf{x}, \mu^*(t)), t) + \gamma^*(t) \quad (3.39)$$

und

$$O(\mu, \lambda, \gamma) = \sum_{\mathbf{x} \in R} (\lambda(t)I(f(\mathbf{x}, \mu), t) + \gamma(t) - I(\mathbf{x}, t_0))^2. \quad (3.40)$$

Analog zu der in Kapitel 3.5.2 definierten Vektorschreibweise lässt sich die zu minimierende Funktion unter der Annahme, dass die Transformationsparameter kontinuierlich sind, schreiben als:

$$\begin{aligned} O(\delta\mu, \delta\lambda, \delta\gamma) &= \|(\lambda + \delta\lambda)I(\mu(t) + \delta\mu, t + \tau) + (\gamma + \delta\gamma) - I(0, t_0)\|^2 & (3.41) \\ \mu(t + \tau) &= \mu(t) + \delta\mu \\ \lambda(t + \tau) &= \lambda(t) + \delta\lambda \\ \gamma(t + \tau) &= \gamma(t) + \delta\gamma \end{aligned}$$

Mit der Taylorreihenentwicklung lässt sich Gleichung 3.41 nun umformen zu (Terme höherer

Ordnung werden ignoriert):

$$O(\delta\mu, \delta\lambda, \delta\gamma) = \|\lambda\mathbf{I}(\mu, t) + \delta + \lambda M\delta\mu + \mathbf{I}(\mu, t)\delta\lambda + \delta\gamma + \lambda\tau\mathbf{I}_t(\mu, t) - \mathbf{I}(0, t_0)\| \quad (3.42)$$

und unter Verwendung der zusätzlichen Approximierung

$$\lambda\tau\mathbf{I}_t(\mu, t) \approx \lambda\mathbf{I}(\mu, t + \tau) - \lambda\mathbf{I}(\mu, t)$$

lässt sich der Algorithmus analog zu Gleichung 3.29 angeben:

$$\begin{aligned} \min O(\dots) &\Rightarrow 0 = \|\lambda M\delta\mu + \mathbf{I}(\mu, t)\delta\lambda + \delta\gamma + \underbrace{\lambda\mathbf{I}(\mu, t) + \gamma - \mathbf{I}(0, t_0)}_{=: \mathbf{e}(t+\tau)}\| \\ &\Leftrightarrow \mathbf{e}(t + \tau) = \lambda M\delta\mu + \mathbf{I}(\mu, t)\delta\lambda + \delta\gamma \\ &\Leftrightarrow \mathbf{e}(t + \tau) = \begin{bmatrix} \lambda M & | & \mathbf{I}(\mu, t) & | & \mathbf{1} \end{bmatrix} \begin{bmatrix} \delta\mu \\ \delta\lambda \\ \delta\gamma \end{bmatrix} \\ &\Leftrightarrow \mathbf{e}(t + \tau) = \begin{bmatrix} M & | & \mathbf{I}(\mu, t) & | & \mathbf{1} \end{bmatrix} \begin{bmatrix} \lambda\delta\mu \\ \delta\lambda \\ \delta\gamma \end{bmatrix} \\ &\Leftrightarrow \begin{bmatrix} M^t \\ \mathbf{I}(\mu, t)^t \\ \mathbf{1}^t \end{bmatrix} \mathbf{e}(t + \tau) = \begin{bmatrix} M^t M & M^t \mathbf{I}(\mu, t) & M^t \mathbf{1} \\ \mathbf{I}(\mu, t)^t M & \mathbf{I}(\mu, t)^t \mathbf{I}(\mu, t) & \mathbf{I}(\mu, t)^t \mathbf{1} \\ \mathbf{1}^t M & \mathbf{1}^t \mathbf{I}(\mu, t) & \mathbf{1}^t \mathbf{1} \end{bmatrix} \begin{bmatrix} \lambda\delta\mu \\ \delta\lambda \\ \delta\gamma \end{bmatrix} \end{aligned} \quad (3.43)$$

Wie in Kapitel 3.5.2 beschrieben lässt sich auch hier die Matrix M vorberechnen, um so Rechenzeit einzusparen. Die zwei neu eingeführten Parameter führen zu zwei zusätzlichen Gleichungen, was zu erwarten war. Mit mehr freien Parametern steigt jedoch das Risiko, eine Region falsch zu tracken. Insbesondere wenn Verdeckungen auftreten, werden Grauwerte der Zielregion verändert. Die Grauwertunterschiede im Vektor \mathbf{e} können als Beobachtungen angesehen werden, die in die Schätzung der Parameter eingehen. Verdeckungen führen also dazu, dass falsche Beobachtungen in die Ausgleichsrechnung eingehen und dazu beitragen, dass die Region nicht richtig getrackt wird. Ein möglicher Ansatz, Verdeckungen im Modell mit zu berücksichtigen, wird im nächsten Kapitel geliefert.

3.5.5 Modellierung von Verdeckungen

Wird ein Objekt über einen längeren Zeitraum getrackt, so kann es vorkommen, dass andere Objekte in das Bild eindringen und das zu trackende Objekt verdecken. Zum Beispiel können fahrende Autos Teile einer zu trackenden Region verdecken. Auch aufgrund der Kamerabewegung kann es zu Verdeckungen kommen. Außerdem ist der Hintergrund um ein zu trackendes Objekt oft in der ausgewählten Region mit enthalten. Die Projektion des Hintergrundes in

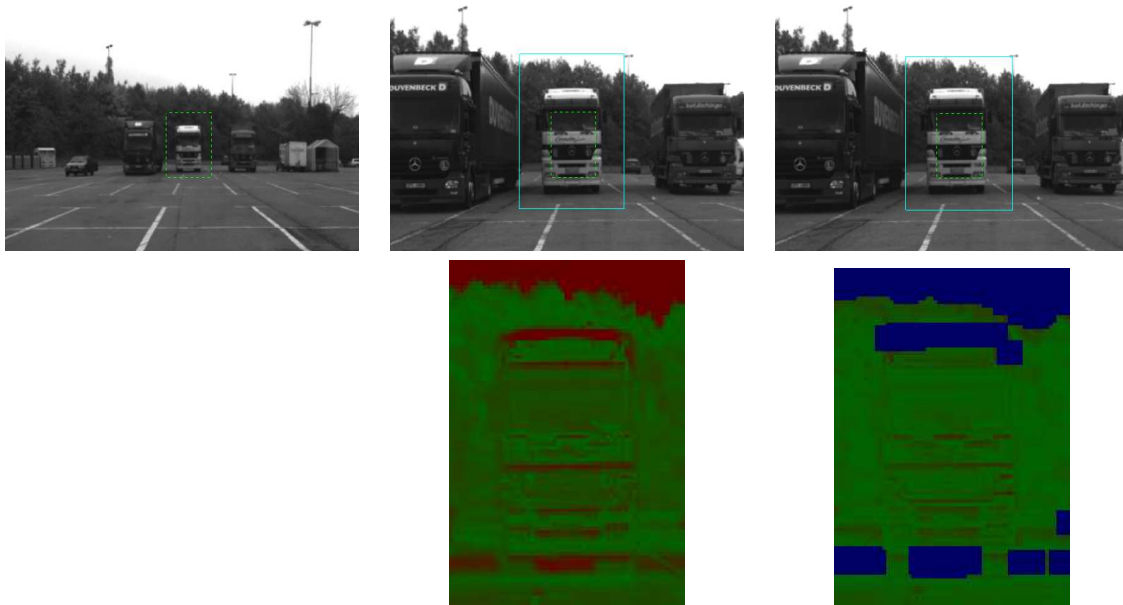


Abbildung 3.11: Fehler durch Verdeckungen. Oben links Zielregion einer Sequenz in Bild 0; oben mittig die getrackte Zielregion in Bild 150 ohne Verdeckungsrechnung, oben rechts mit Verdeckungsrechnung. Unterhalb der Bilder sind die Residuen der Zielregion zu sehen. Große Residuen sind rot eingefärbt. Der Verdeckungsbereich ist blau hinterlegt. Das verbesserte Trackergebnis ist am Residuenbild deutlich erkennbar.

das Bild transformiert sich dabei meistens anders als das abgebildete Objekt (vgl. Abbildung 3.11). Alle Effekte, welcher Art sie auch sein mögen, die nicht im Modell berücksichtigt sind, beeinflussen die Schätzung der Bewegungsparameter und können langfristig gesehen zu einem falschen Tracking führen. In diesem Abschnitt wird ein Verfahren kurz beschrieben, welches auch in dem Ansatz von Hager et al. [35] verwendet wird, um solche Probleme zu vermeiden.

Der Ansatz geht von der Annahme aus, dass Verdeckungen große Grauwertunterschiede zwischen der nicht-transformierten Zielregion zum Zeitpunkt t_0 und der transformierten Zielregion zum Zeitpunkt t aufweisen. Damit können solche Bildpunkte als Ausreißer behandelt werden [9]. Ein Ansatz eine robuste Least Squares Schätzung mit Berücksichtigung von Ausreißern durchzuführen, ist das so genannte *iteratively reweighted least squares* Verfahren (IRLS). Hager et al. nutzen in ihrer Implementation eine IRLS Methode von Dutter und Huber [20]. Neben der bereits bestehenden *äußeren* Iteration, in der jedes Mal die Zielregion auf das aktuelle Bild gewarpt werden muss und der Fehlervektor \mathbf{e} berechnet wird, wird eine zusätzliche *innere* Iteration eingeführt, welche die Ausreißer bestimmt und damit eine robuste Schätzung

durchführt. Der Fehlervektor $\mathbf{e}(t + \tau)$ wird wie dabei wie folgt modifiziert:

$$\mathbf{e}(t + \tau)^i = W^i \underbrace{(\mathbf{e}(t + \tau) - M(\mu)\delta\mu^i)}_{=\mathbf{r}}. \quad (3.44)$$

Der Vektor \mathbf{r} wird auch Residuenvektor genannt. Der Index i gibt die i -te innere Iteration an und W ist eine Diagonalmatrix mit Gewichten w_{kk}^i ,

$$w_{kk}^i = \begin{cases} 1 & \text{falls } |r_i| \leq \alpha \\ \frac{c}{|r_i|} & \text{falls } |r_i| > \alpha \end{cases} \quad (3.45)$$

mit den Konstanten α und c , welche angeben wann ein Residuum als Ausreißer angesehen wird und wie Ausreißer gewichtet werden. Wie Dutter und Huber in [20] gezeigt haben, konvergiert dieser Ansatz für lineare Probleme zu einer eindeutigen globalen Lösung.

Der Algorithmus berechnet für jede äußere Iteration die Gewichtsmatrix komplett neu. Vor dem Hintergrund, dass Tracken ein kontinuierlicher Prozess ist, wäre es sinnvoller, jede äußere Iteration mit einer Gewichtsmatrix zu starten, welche ähnlich ist zu der im vorherigen Iterationsschritt berechneten. Dabei sollten zwei Dinge beachtet werden. Erstens der Fakt, dass das zu lösende lineare System lediglich eine Approximation eines nichtlinearen Vorgangs ist und daher für größere Verschiebungen die Terme höherer Ordnung in der Taylorreihe dazu führen können, dass einzelne Pixel fälschlicherweise als Ausreißer deklariert werden. Zweitens kann davon ausgegangen werden, dass zusammenhängende Regionen von Pixeln mit niedrigen Gewichten zu Ausreißern, also Verdeckungen, gehören. Verdeckungen können weiter in die Region eindringen und im nächsten Bild eine größere Fläche einnehmen. Hier macht es Sinn, eine *Pufferzone* um diese Regionen einzurichten.

Beide Effekte können berücksichtigt werden, wenn die Gewichte der Diagonalmatrix W als Bild angesehen werden, in dem Pixel mit Werten kleiner Eins zu Verdeckungen gehören und Pixel gleich Eins das erwartete Objekt repräsentieren. Hager [35] rechnet deshalb auf dem so definierten Bild der Gewichte die morphologischen Operationen Erosion und Dilatation

$$\text{Erosion: } I(\mathbf{x}) = \max_{\mathbf{v} \in Q(\mathbf{x})} I(\mathbf{v}) \quad (3.46)$$

$$\text{Dilatation: } \mathbf{x} = \min_{\mathbf{v} \in Q(\mathbf{x})} I(\mathbf{v}). \quad (3.47)$$

mit der Achter-Nachbarschaft von \mathbf{x} , $Q(\mathbf{x})$. Erosion hat den Effekt, dass kleine Regionen von Ausreißern entfernt werden, wobei Dilatation Regionen vergrößert, also eine Pufferzone um Regionen von Ausreißern schafft. In der Regel wird ein Mal Erosion durchgeführt gefolgt von mehreren Dilatationsschritten. Durch Nutzung dieser morphologischen Operatoren zwischen aufeinanderfolgenden Frames wird eine gute Approximation der neuen Gewichtsmatrix geschaffen. Zusätzlich können andere problemspezifische Bedingungen formuliert werden. Zum Beispiel kann erzwungen werden, dass Verdeckungen generell nur vom Rand her geschehen dürfen.

3.5.6 Trackergebnis

In den meisten Anwendungen ist es wünschenswert, als Ergebnis eines Trackingvorganges neben den Bewegungsparametern der Zielregion eine Aussage zu erhalten, wie *gut* die Grauwerte der transformierten Zielregion zu den Grauwerten des aktuellen Bildes passen. Zur Beurteilung des Ergebnisses haben sich in der Literatur drei Verfahren durchgesetzt: Korrelation und Summe der quadratischen sowie absoluten Grauwertunterschiede. Daneben kann man Aussagen treffen, die die Bewegungsparameter nutzen. So kann man zum Beispiel die Seitenlängen der getrackten Zielregion betrachten. Falls eine Region in einer oder beiden Dimensionen zu stark schrumpft, kann es vorkommen, dass die enthaltene Information wegen der Minimierung verloren geht. Außerdem kann man die Varianzen der Bewegungsparameter nutzen, um Aussagen über die Qualität eines Tracks zu machen. Im Folgenden sollen die einzelnen Möglichkeiten beschrieben werden.

3.5.6.1 Korrelation, SSD und SAD

Korrelation, wie sie in Kapitel 3.5.1.2 beschrieben ist, berücksichtigt implizit Intensitätsvariationen. Regionen mit großer Varianz führen aufgrund der Interpolation zur Berechnung der Grauwerte zu großen Residuen. Hier ist die Nutzung von Korrelation zur Bewertung des Matching Ergebnisses in der Regel besser als die Summierung über Grauwertunterschiede [46]. Fällt der Korrelationskoeffizient NC unter einen fest gewählten Schwellwert, typische Werte liegen zwischen 0,7 und 0,95, so wird das Ergebnis als nicht vertrauenswürdig angesehen. Die Summe der quadratischen Grauwertunterschiede (*sum of squared differences, SSD*) und die Summe der absoluten Grauwertunterschiede (*sum of absolute differences, SAD*) ist schneller zu berechnen, da im allgemeinen auf den Normalisierungsschritt verzichtet wird. Allerdings sind solche Ansätze anfällig gegenüber Ausreißern. Insbesondere bei der Berechnung der SSD führen Ausreißer zu sehr großen Werten, da der Grauwertunterschied hier quadratisch eingeht. Schwellwerte für die Beurteilung des Ergebnisses sind empirisch zu bestimmen. Die Berechnung der Ähnlichkeitsmessungen für eine Referenz Region $g_1(i) = \{I(\mathbf{x}_i, t_0)\}_{i=0}^N$ und deren transformierte Bildregion $g_2(i) = \{I(\mathbf{x}_i, t)\}_{i=0}^N$ geschieht nach [92] wie folgt:

$$SSD = \sum_i (g_1(i) - g_2(i))^2, \quad (3.48)$$

$$SAD = \sum_i |g_1(i) - g_2(i)|, \quad (3.49)$$

$$NC = \sum_i \tilde{g}_1(i) \tilde{g}_2(i), \quad (3.50)$$

wobei \tilde{g}_k die normalisierten Grauwerte der Region g_k sind, so dass

$$\sum_i \tilde{g}_k(i) = 0 \text{ und } \sum_i \tilde{g}_k(i)^2 = 1.$$

3.5.6.2 Berechnung der Varianzen

Fliessen die geschätzten Parameter der geometrischen und radiometrischen Transformation in die Bewertung des Tracks mit ein oder werden diese Parameter für weitere mathematische Genauigkeitsbetrachtungen benötigt, so ist die Varianz dieser Parameter nötig. Wird ein Gaußsches Rauschen angenommen, so kann man nach [29] die Standardabweichungen σ_k der einzelnen Transformationsparameter μ_k aus der Inverse der so genannten *Normalgleichungsmatrix* (vergleiche Gleichung 3.29) ableiten, indem man mit dem Bildrauschen σ_n multipliziert:

$$\sigma_k = \sigma_n \left((M^t M)^{-1} \right)_{kk}. \quad (3.51)$$

Das Bildrauschen muss entsprechend geschätzt werden. Eine gute Approximation ist es, den mittleren SSD Wert pro Pixel für das Bildrauschen einzusetzen. Eine detailliertere mathematische Betrachtung ist in [63] zu finden.

3.5.7 Neuinitialisieren eines Tracks

Wenn die Beurteilung des Ergebnisses eines Tracks als nicht mehr hinreichend vertrauenswürdig angesehen wird, macht es keinen Sinn, die Zielregion weiter zu tracken. In solch einem Fall stimmen entweder die Modellannahmen nicht mit den tatsächlichen Beobachtungen überein oder der Algorithmus ist in ein lokales Minimum gelaufen und findet die Zielregion nicht. Geht man von ersterem und der zusätzlichen Annahme, dass die Diskrepanz zwischen Modell und tatsächlichen Beobachtungen mit der Zeit zunimmt aus, so macht es Sinn, die Zielregion neu zu formulieren. Dabei ist darauf zu achten, die Transformationsparameter unter Berücksichtigung der bisherigen Transformation anzugeben und auch die Fehler entsprechend zu propagieren. Intuitiv wird damit klar, dass bei jeder Neuinitialisierung die Varianz der Parameter zunimmt. Der Vorteil eines Verfahrens, welches es erlaubt über viele Frames hinweg zu tracken, wird somit mit jeder Neuinitialisierung gemindert. Auf der anderen Seite kann es durchaus sinnvoller sein, eine Region neu aufzusetzen, als mit falschen Annahmen weiter zu verfolgen. Hierbei ist besonders auf Schwellwerte zu achten.

3.5.7.1 Fehlerfortpflanzung

Wird die zu trackende Region zum Zeitpunkt t_s neu aufgesetzt und werden die Transformationsparameter und insbesondere die Fehler vom Zeitpunkt t_s zum aktuellen Zeitpunkt t berechnet, so muss mit Fehlerfortpflanzung dafür gesorgt werden, dass die bereits vorhandenen Varianzen von t_0 bis t_s mitberücksichtigt werden. Seien die geschätzten Parametervektoren μ_a und μ_b gegeben und sei der Parametervektor vom Zeitpunkt t_0 zum aktuellen Zeitpunkt t

berechenbar als $\mu = f(\mu_a, \mu_b)$. Für die Varianzen ergibt sich dann mit Fehlerfortpflanzung:

$$Cov[\mu] = J_{\mu_a} Cov[\mu_a] J_{\mu_a}^t + J_{\mu_b} Cov[\mu_b] J_{\mu_b}^t, \quad J_{\mu_i} = \begin{pmatrix} \frac{\partial f_1}{\partial \mu_{i1}} & \frac{\partial f_1}{\partial \mu_{i2}} & \cdots \\ \frac{\partial f_2}{\partial \mu_{i1}} & \ddots & \vdots \\ \vdots & \dots & . \end{pmatrix}. \quad (3.52)$$

Für den Spezialfall einer affinen Abbildung mit $\mu_a = (a_1, \dots, a_6)^t$,

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_4 & a_5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_3 \\ a_6 \end{pmatrix},$$

gilt nach Neuinitialisierung mit dem neu geschätzten Parametervektor $\mu_b = (b_1, \dots, b_6)^t$:

$$\begin{aligned} \begin{pmatrix} x'' \\ y'' \end{pmatrix} &= \begin{pmatrix} b_1 & b_2 \\ b_4 & b_5 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} b_3 \\ b_6 \end{pmatrix} \\ &= \begin{pmatrix} b_1 & b_2 \\ b_4 & b_5 \end{pmatrix} \left(\begin{pmatrix} a_1 & a_2 \\ a_4 & a_5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_3 \\ a_6 \end{pmatrix} \right) + \begin{pmatrix} b_3 \\ b_6 \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} b_1 & b_2 \\ b_4 & b_5 \end{pmatrix} \begin{pmatrix} a_1 & a_2 \\ a_4 & a_5 \end{pmatrix}}_{\begin{pmatrix} \mu_1 & \mu_2 \\ \mu_4 & \mu_5 \end{pmatrix}} \begin{pmatrix} x \\ y \end{pmatrix} + \underbrace{\begin{pmatrix} b_1 & b_2 \\ b_4 & b_5 \end{pmatrix} \begin{pmatrix} a_3 \\ a_6 \end{pmatrix}}_{\begin{pmatrix} \mu_3 \\ \mu_6 \end{pmatrix}} + \begin{pmatrix} b_3 \\ b_6 \end{pmatrix} \end{aligned} \quad (3.53)$$

und somit gilt für die Fehlerfortpflanzung mit gegebenen Kovarianzmatrizen C_a und C_b :

$$Cov[\mu] = \begin{pmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 \\ 0 & b_1 & 0 & 0 & b_2 & 0 \\ 0 & 0 & b_1 & 0 & 0 & b_2 \\ b_4 & 0 & 0 & b_5 & 0 & 0 \\ 0 & b_4 & 0 & 0 & b_5 & 0 \\ 0 & 0 & b_4 & 0 & 0 & b_5 \end{pmatrix} C_a \begin{pmatrix} \dots \\ \dots \\ \dots \end{pmatrix}^t + \begin{pmatrix} a_1 & a_4 & 0 & 0 & 0 & 0 \\ a_2 & a_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & a_4 & 0 \\ 0 & 0 & 0 & a_2 & a_5 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} C_b \begin{pmatrix} \dots \\ \dots \\ \dots \end{pmatrix}^t \quad (3.54)$$

3.6 Orthophoto und inverse Perspektive

Der Begriff Orthophoto stammt aus der Photogrammetrie [44]. Gemeint ist ein durch Differentialverzerrung verändertes, photographisches Bild, das geometrisch weitgehend einer orthogonalen Projektion des Objekts auf eine abwickelbare Bezugsfläche entspricht. Dieses Bild ist bereits entzerrt und auf ein orthogonales Koordinatensystem transformiert. Der große Vorteil einer solchen Entzerrung ist es, dass direkt im Bild gemessen werden kann. Dieses Prinzip

des Orthophotos lässt sich auf Videobildverarbeitung übertragen, indem man als Objekt die Straße oder aber eine beliebige andere Fläche definiert. Ist die Fläche eben, so hat man den Spezialfall einer Homographie, wie bereits in Kapitel 2.3.1.3 beschrieben. Da ein Orthophoto in einem solchen Fall der Invertierung der Homographie gleichkommt, nennt man solch eine Abbildung auch *Inverse Perspektive*. So lässt sich zum Beispiel ein Orthophoto der Straßenebene erstellen um so Distanzen messen zu können. Problematisch ist dabei der absolute Nickwinkel zur Straße, welcher bekannt sein muss. Selbst kleine Änderungen führen hier zu großen Verzerrungen. Obwohl das Bild dann nur für die definierte Fläche gilt, bietet dieses Verfahren trotzdem Vorteile. Unter der Annahme, alle stehenden Hindernisse stehen senkrecht zur Straße, wie es in Bauklotzszenen der Fall ist, unterscheiden sich die Orthophotos auf stehende Hindernisse lediglich um einen Skalierungsfaktor und eine Translation. Alle anderen perspektiven Effekte werden somit herausgerechnet. Dies vereinfacht mathematische Betrachtungen und beschleunigt das Tracken, da die verbleibenden Transformationen einfacher zu rechnen sind. Im Folgenden werden sowohl die Sicht auf die Straße als auch auf stehende Hindernisse für Kameras mit gegebener äußerer Orientierung mathematisch hergeleitet.

3.6.1 Sicht auf die Straße

Bei der Sicht auf die Straße haben alle Weltkoordinaten die Y -Koordinate 0. Die Achsen des Orthophotos entsprechen somit der X -Achse sowie der Z -Achse des Weltkoordinatensystems. Folgende Abbildung gilt es zu invertieren:

$$\begin{aligned}
 \begin{bmatrix} x \\ y \\ h \end{bmatrix} &= \begin{bmatrix} \frac{f}{s_u} & 0 & u_0 \\ 0 & -\frac{f}{s_v} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \end{bmatrix} \begin{bmatrix} X_W \\ 0 \\ Z_W \\ 1 \end{bmatrix} \\
 &= \underbrace{\begin{bmatrix} \frac{f}{s_u} & 0 & u_0 \\ 0 & -\frac{f}{s_v} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & -X_0 \\ 0 & 0 & -Y_0 \\ 0 & 1 & -Z_0 \end{bmatrix}}_{{}^B T_S} \begin{bmatrix} X_W \\ Z_W \\ 1 \end{bmatrix} \quad (3.55)
 \end{aligned}$$

Im Allgemeinen ist die Matrix ${}^B T_S$ invertierbar und damit die orthogonale Sicht auf die Straßenebene gegeben. In Abbildung 3.12 ist zur Veranschaulichung ein Beispiel abgebildet.

3.6.2 Sicht auf stehendes Hindernis

Bei Sicht auf stehende Hindernisse ist es notwendig eine Z -Koordinate anzugeben (welche vor der Kamera liegen sollte). Die Z -Koordinate ist proportional zum Skalierungsfaktor mit dem

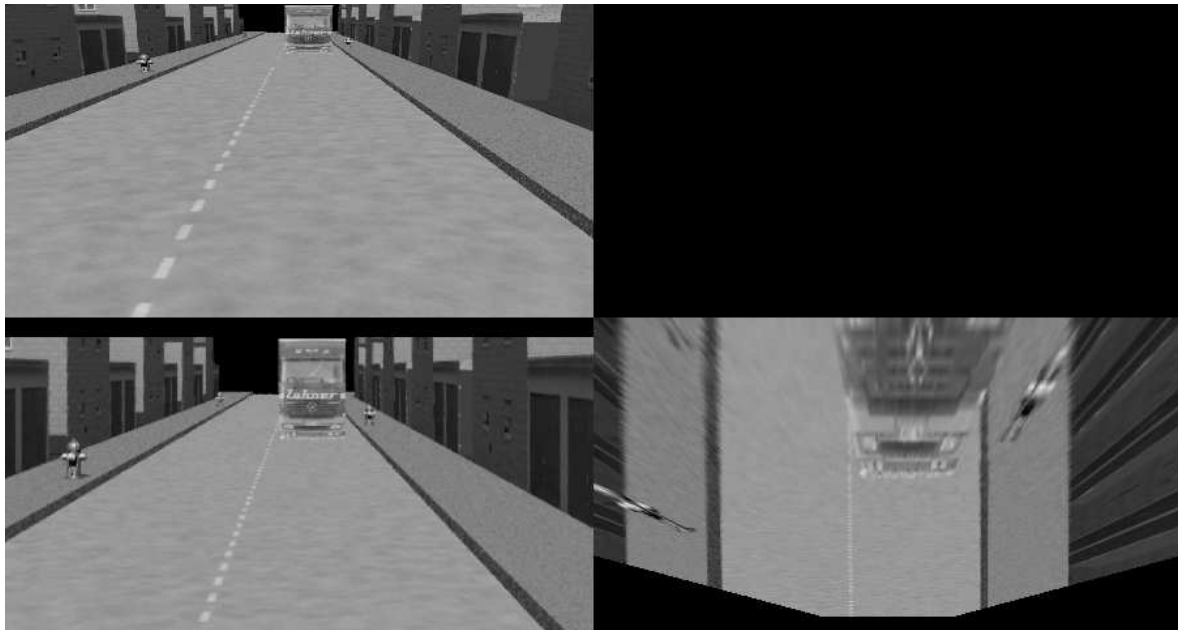


Abbildung 3.12: Beispiel einer Inversen Perspektive. Oben links ist das Originalbild auf welches die Inverse Perspektive gerechnet wurde; der LKW ist ca. 50m entfernt. Das untere linke Bild zeigt die Sicht auf die Ebene $Z = 7m$; das untere rechte Bild zeigt die Sicht von oben auf die Straßenebene. Die Bilder sind entsprechend skaliert, so dass der LKW abgebildet ist.

FOE als Fixpunkt. Die zu invertierende Abbildung ist für die Ebene $Z = d$:

$$\begin{aligned}
 \begin{bmatrix} x \\ y \\ h \end{bmatrix} &= \begin{bmatrix} \frac{f}{s_u} & 0 & u_0 \\ 0 & -\frac{f}{s_v} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ d \\ 1 \end{bmatrix} \\
 &= \underbrace{\begin{bmatrix} \frac{f}{s_u} & 0 & u_0 \\ 0 & -\frac{f}{s_v} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & -X_0 \\ 0 & 1 & -Y_0 \\ 0 & 0 & d - Z_0 \end{bmatrix}}_{{}^B T_O} \begin{bmatrix} X_W \\ Z_W \\ 1 \end{bmatrix} \quad (3.56)
 \end{aligned}$$

Im Allgemeinen ist die Matrix ${}^B T_O$ invertierbar und damit die orthogonale Sicht auf Objekte mit Z -Koordinate d gegeben. Abbildung 3.12 zeigt ein Beispiel dieser Abbildung.

Kapitel 4

Hypothesengenerierung

In diesem Kapitel wird die Hypothesengenerierung vorgestellt. Deren Ziel ist es, bei bewegter Kamera in einer stationären Welt möglichst alle Hindernisse zu detektieren. Hindernisse sind stehenden Objekte, welche sich von der Straßenebene abheben.

Der Input zur Hypothesengenerierung ist eine Videosequenz einer im Versuchsträger eingebauten Kamera mit bekannter innerer Orientierung, die sechs Parameter der äußeren Orientierung (Translation und Rotation) der Kamera und die gefahrene Distanz des Versuchsträgers zwischen den Bildern der Videosequenz. Das Ergebnis der Hypothesengenerierung ist die absolute Position potentieller Hindernisse im Weltkoordinatensystem.

Im Folgenden Abschnitt wird das Konzept erläutert. In Abschnitt 4.2 wird die mathematische Umsetzung des Konzeptes beschrieben. Die Implementierung des Konzeptes wird in Abschnitt 4.3 vorgestellt.

4.1 Konzept

Bei der projektiven Abbildung einer Kamera werden Punkte aus dreidimensionalen Weltkoordinaten in zweidimensionale Bildkoordinaten transformiert, wobei eine Dimension verloren geht. Um jedem Bildpunkt einen Weltpunkt zuzuordnen zu können, muss eine der drei Weltkoordinaten bekannt sein. Ist die Z -Koordinate bekannt, so kann jeder abgebildete Punkt lokalisiert werden.

Im Fall von stationären Hindernissen ist für jedes Hindernis aufgrund des Bauklotzmodells die Z -Koordinate der Frontfläche identisch. Es bietet sich also an, diese Entfernung aus möglichst vielen Punkten auf dem Hindernis zu schätzen.

Im nächsten Abschnitt wird die Entfernungsschätzung für Hindernisse aus dem Bewegungsfeld hergeleitet. Dabei gilt die Annahme, dass das Bild in Regionen aufgeteilt werden kann und alle Bildpunkte einer Region zu einem potentiellen Hindernis gehören. Beim Verfolgen eines Hindernisses und Schätzen der Bewegungsparameter wird das in Abschnitt 3.5.2 hergeleitete Region Tracking Verfahren angewendet.

4.2 Mathematische Herleitung

Die in diesem Abschnitt untersuchten Zusammenhänge beziehen sich auf das Kamera- und Sensorkoordinatensystem. Ausgehend von einer Bewegung parallel zur Z -Achse des Kamerakoordinatensystems und unter Vernachlässigung der Rotationsparameter der Bewegung, also

$$T_X = T_Y = 0 \text{ und } \Omega_X = \Omega_Y = \Omega_Z = 0, \quad (4.1)$$

ergibt sich nach Gleichung 3.15 folgendes Bewegungsfeld auf dem Sensor:

$$\mathbf{v}(\mathbf{x}, Z, T_Z) = \frac{T_Z}{Z} \begin{pmatrix} u \\ v \end{pmatrix} = \frac{T_Z}{Z} \mathbf{x}. \quad (4.2)$$

Bei bekannter Translationsgeschwindigkeit T_Z und bekannter Geschwindigkeit eines Bildpunktes auf dem Sensor kann damit die Entfernung Z des in den Bildpunkt $\mathbf{x} = (u, v)^t$ abgebildeten Raumpunktes berechnet werden. Dabei treten drei Probleme auf:

- Die Bewegung einer im Versuchsträger angebrachten Kamera ist bei Vorwärtsbewegung des Versuchsträgers in der Regel keine reine Translation in Z -Richtung.
- Im Versuchsträger treten Rotationen und Verschiebungen auf, die von der reinen Vorwärtsbewegung abweichen. Insbesondere Nickbewegungen (Ω_X) und Gierbewegungen (Ω_Y, T_X) können auftreten und müssen mitmodelliert werden.
- Liegt ein optisches Flussfeld vor, so kann man in der Regel davon ausgehen, dass dies ein Feld von Verschiebungsvektoren zwischen zwei Bildern ist. Diese Verschiebungsvektoren können aber, insbesondere bei größeren Zeitabständen zwischen den Bildern, nicht mit den Geschwindigkeiten gleichgesetzt werden.

In den nächsten Abschnitten wird eine Lösung zur Berechnung der Entfernung Z erarbeitet, die diese Probleme berücksichtigt.

4.2.1 Bewegungsfeld als Anfangswertproblem

In der Herleitung des Bewegungsfeldes in Abschnitt 3.4.1 war die Fragestellung:

Gegeben: ein 3D-Weltpunkt \mathbf{X} zum Zeitpunkt t und die linearisierte Eigenbewegung in Form von Translations- sowie Rotationsgeschwindigkeit T und Ω .

Gesucht: das in das Bild projizierte 2D Bewegungsfeld zum Zeitpunkt t .

Bei Gleichung 3.15 handelt es sich somit also um eine Linearisierung des Bewegungsfeldes, wodurch dieses nur für den Zeitpunkt t gültig ist. Insbesondere ändert sich der Skalierungsfaktor $s = \frac{T_Z}{Z}$, wenn sich Translationsgeschwindigkeit oder Entfernung des Weltpunktes ändern.

Eine Bildfolge besteht aus diskreten Beobachtungen (Bildern) zu festen Zeitpunkten. In der Regel sind lediglich zu diesen diskreten Zeitpunkten Messungen der Translationsgeschwindigkeit und der Entfernung vorhanden. Messungen im Bild sind zwischen einzelnen Aufnahmen gar nicht möglich. Um also die Bewegung eines Pixels $\mathbf{x}(t)$ bei bekannter Entfernung des Weltpunktes und bekannter Translationsgeschwindigkeit zu modellieren, muss folgendes Anfangswertproblem (AWP) gelöst werden:

$$\text{AWP : } \quad \dot{\mathbf{x}} = f(\mathbf{x}(t), t), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (4.3)$$

wobei $f(\mathbf{x}, t)$ das projizierte Bewegungsfeld des Raumpunktes \mathbf{X} ist (siehe Gleichung 3.13) und \mathbf{x}_0 der beobachtete Bildpunkt zum Zeitpunkt $t = t_0$. Damit taucht man in die Mathematik gewöhnlicher Differentialgleichungen ein. Bisherige Arbeiten haben die Bewegung von Bildpunkten numerisch mit inkrementalen Bewegungsschritten modelliert (z.B. [51], [70]). Dies entspricht der Lösung von 4.3 mit numerischen Integrationsverfahren und hat den Nachteil, dass sich aufgrund der Linearisierung Fehler akkumulieren. Um die Fehler gering zu halten, muss der Zeitabstand zwischen den Beobachtungen (die *Schrittweite*) entsprechend klein sein. Verschiedene Ansätze, dieses Problem zu lösen, liefern zudem unterschiedliche Möglichkeiten, den Skalierungsfaktor s zu interpretieren. Im Folgenden soll das explizite Eulerverfahren, was für viele Anwendungen benutzt wird, näher untersucht und dessen Probleme aufgezeigt werden.

Jede Lösung $\mathbf{x}(t)$ von $\dot{\mathbf{x}} = f(\mathbf{x}(t), t)$ zeichnet sich gerade dadurch aus, dass jeder Punkt $\mathbf{x}(t) = (u, v)^t$ die Ableitung $f(\mathbf{x}(t), t)$ hat. Es liegt daher nahe, die Lösung in einer Umgebung von t dadurch zu approximieren, dass man dem Gradienten in $\mathbf{x}(t)$ von t bis $t + h$ folgt [56]. Bei bekanntem $\mathbf{x}(t_k)$ erhält man für die diskreten Zeitpunkte $t_k = t_0 + hk$ die Lösung

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + hf(\mathbf{x}(t_k), t_k) = \mathbf{x}(t_k) + h(\mathbf{v}_t(\mathbf{x}(t_k), t_k) + \mathbf{v}_r(\mathbf{x}(t_k), t_k)), \quad (4.4)$$

wobei \mathbf{v}_t das Translations- und \mathbf{v}_r das Rotationsbewegungsfeld sind (siehe Abschnitt 3.4.1). Dieses Verfahren wird auch Euler'sche Polygonzugmethode oder explizites Eulerverfahren genannt. Die Euler'sche Polygonzugmethode besteht nun darin, dieses Verfahren mit dem neuen Startwert $\mathbf{x}(t_{k+1})$ zu wiederholen.

Zunächst einmal ist nicht klar, ob solch ein Verfahren überhaupt für kleine Schrittweiten h gegen die wahre Lösung $\mathbf{x}^*(t)$ konvergiert. Zu zeigen ist, dass f mit $\mathbf{v} = \mathbf{v}_t + \mathbf{v}_r$ der Lipschitzbedingung $\|f(\mathbf{x}_1, t) - f(\mathbf{x}_2, t)\| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|$ mit $L \in \mathbb{R}$ genügt:

$$\|f(\mathbf{x}_1, t) - f(\mathbf{x}_2, t)\| = \|\mathbf{v}(\mathbf{x}_1, t) - \mathbf{v}(\mathbf{x}_2, t)\| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|. \quad (4.5)$$

Für den Fall einer stationären Welt ist die Translationsgeschwindigkeit und Rotationsgeschwindigkeit aller abgebildeten Weltpunkte für einen bestimmten Zeitpunkt identisch. Somit unterscheiden sie sich lediglich in den absoluten Koordinaten. Außerdem gilt nach Annahme, dass

die Z -Koordinaten auf Hindernissen identisch sind, also $Z_1 = Z_2$. Es folgt mit Gleichung 3.13:

$$\begin{aligned}
\|f(\mathbf{x}_1, t) - f(\mathbf{x}_2, t)\| &= \left\| \frac{f}{Z_1^2} \begin{pmatrix} \dot{X}Z_1 - X_1\dot{Z} \\ Y_1\dot{Z} - \dot{Y}Z_1 \end{pmatrix} - \frac{f}{Z_2^2} \begin{pmatrix} \dot{X}Z_2 - X_2\dot{Z} \\ Y_2\dot{Z} - \dot{Y}Z_2 \end{pmatrix} \right\| \\
&= \left\| \underbrace{\left(\frac{f}{Z_1} - \frac{f}{Z_2} \right)}_0 \begin{pmatrix} \dot{X} \\ -\dot{Y} \end{pmatrix} + \frac{f\dot{Z}}{Z_1^2} \begin{pmatrix} -X_1 \\ Y_1 \end{pmatrix} - \frac{f\dot{Z}}{Z_2^2} \begin{pmatrix} -X_2 \\ Y_2 \end{pmatrix} \right\| \\
&\stackrel{3.12}{=} \left\| -\frac{\dot{Z}}{Z}(\mathbf{x}_1 - \mathbf{x}_2) \right\| \\
&\leq \left| \frac{\dot{Z}}{Z} \right| \|\mathbf{x}_1 - \mathbf{x}_2\|. \tag{4.6}
\end{aligned}$$

Man kann davon ausgehen, dass die Geschwindigkeit eines Fahrzeuges begrenzt ist, also $\dot{Z} \leq G$ und dass man sich nur bis auf eine gewisse Mindestentfernung dem Hindernis nähert, bevor das Fahrzeug zum Stillstand kommt, also $Z > Z_{min}$. Es folgt die Lipschitzstetigkeit mit $\frac{\dot{Z}}{Z} \leq \frac{G}{Z_{min}}$. Für $h \rightarrow 0$ konvergiert das Verfahren also gegen die wahre Lösung. Umgekehrt ist bei Kenntnis der Verschiebungsvektoren im Bild für $h \rightarrow 0$ die errechnete Z -Koordinate eines Hindernisses gleich der wahren Z -Koordinate. Es ist aber, vor allem im *Focus of Expansion*, nahezu unmöglich, für kleine h (also kleine Zeitabstände zwischen den einzelnen Aufnahmen) Flussvektoren exakt zu vermessen. Auch in einem Trackingansatz, in dem die Transformationsparameter für ganze Regionen berechnet werden, nimmt die relative Ungenauigkeit der geschätzten Parameter mit größeren Bewegungen und damit größeren Zeitschritten ab.

Für größere Zeitschritte aber ist die Euler'sche Polygonzugmethode nicht geeignet, da lediglich eine Linearisierung um den Startwert vorgenommen wird. Sei $Z(t)$ die Entfernung eines Bildpunktes $\mathbf{x}(t)$ und $T_{Z,t,t+h}$ die zurückgelegte Entfernung zwischen den beiden Zeitpunkten $t = t_k$ und $t + h = t_{k+1}$. Damit gilt für den Fall wie er in Gleichung 4.2 beschrieben ist:

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + hf(\mathbf{x}(t_k), t_k) = \mathbf{x}(t_k) + h \frac{T_{Z(t_k)}}{Z(t_k)} \mathbf{x}(t_k) = \left(1 + \frac{T_{Z,t,t+h}}{Z(t_k)} \right) \mathbf{x}(t_k). \tag{4.7}$$

Wegen $T_{Z,t,t+h} \leq Z(t_k)$, da man sonst bereits mit dem Hindernis kollidiert wäre, gilt weiter

$$\mathbf{x}(t_{k+1}) \leq 2\mathbf{x}(t_k).$$

Die Abbildung eines weit entfernten Objektes skaliert sich also bei Auffahrt auf dieses Objekt um maximal das doppelte. Liegt umgekehrt die Abbildung eines Objektes zum Zeitpunkt t_{k+1} vor, muss die Abbildung desselben Objektes zum Zeitpunkt t_k mindestens halb so groß gewesen sein. Bei positivem $T_{Z,t,t+h}$ gilt $Z(t_k) > Z(t_{k+1})$. In perspektivischen Abbildungen werden weit entfernte Objekte aber verschwindend klein. Offensichtlich ist diese Schlussfolgerung falsch und

das explizite Eulerverfahren zumindest für große Zeitschritte nicht geeignet.

Das Eulerverfahren bietet sich also an, Bewegungen zwischen zwei aufeinanderfolgenden Bildern zu interpretieren, wenn die zurückgelegte Distanz $T_{Z,t,t+h}$ nicht zu groß ist. Allerdings muss man beachten, dass bei kurzen Flussvektoren die relative Messungenauigkeit des optischen Flussfeldes groß ist und sich Fehler, welche aufgrund der Linearisierung gemacht werden, akkumulieren. In [70] werden daher Kalmanfilter verwendet, um die Daten zeitlich zu filtern.

4.2.2 Bewegungsfeld aus reiner Translation

Für den Fall, dass eine reine Translation vorliegt, lässt sich das Bewegungsfeld auch anders interpretieren. Gegeben seien die Koordinaten eines Weltpunktes zum Zeitpunkt t :

$$X(t), Y(t) \text{ und } Z(t)$$

sowie die Translation der Kamera vom Zeitpunkt t zum Zeitpunkt $t + h$

$$T_{X,t,t+h}, T_{Y,t,t+h} \text{ und } T_{Z,t,t+h}.$$

Damit gilt für die Abbildung in den Bildpunkt $\mathbf{x}(t)$:

$$\begin{aligned} \mathbf{x}(t+h) &\stackrel{3.12}{=} \frac{f}{Z(t+h)} \begin{pmatrix} X(t+h) \\ -Y(t+h) \end{pmatrix} \\ &= \frac{f}{Z(t+h)} \frac{Z(t)}{Z(t)} \begin{pmatrix} X(t) + T_{X,t,t+h} \\ -Y(t) - T_{Y,t,t+h} \end{pmatrix} \\ &= \frac{Z(t)}{Z(t+h)} \frac{f}{Z(t)} \begin{pmatrix} X(t) \\ -Y(t) \end{pmatrix} + \frac{f}{Z(t+h)} \begin{pmatrix} T_{X,t,t+h} \\ -T_{Y,t,t+h} \end{pmatrix} \\ &= \frac{Z(t)}{Z(t) + T_{Z,t,t+h}} \mathbf{x}(t) + \mathbf{m}(t). \end{aligned} \quad (4.8)$$

Das Bewegungsfeld einer Region mit gleicher Z -Koordinate ist demnach eine Skalierung mit dem Faktor $\frac{Z(t)}{Z(t+h)}$ und eine Verschiebung um $\mathbf{m}(t)$. Schätzt man für diese Region die Skalierung, so lässt sich die Entfernung des Hindernisses bei Kenntnis der Translation der Kamera in Z -Richtung exakt angeben. In Abbildung 4.1 ist für den Fall einer reinen Translation in Z -Richtung das Prinzip der Skalierung verdeutlicht. Mit dem Strahlensatz gilt

$$\frac{d'}{d} = \frac{Z(t)}{Z(t+h)}.$$

4.2.3 Kompensierung der Rotationsparameter

Um eine reine Translation zu modellieren, muss eine Skalierung und Translation einer Bildregion geschätzt werden. In diesem Abschnitt wird erarbeitet, wie die Rotationsparameter

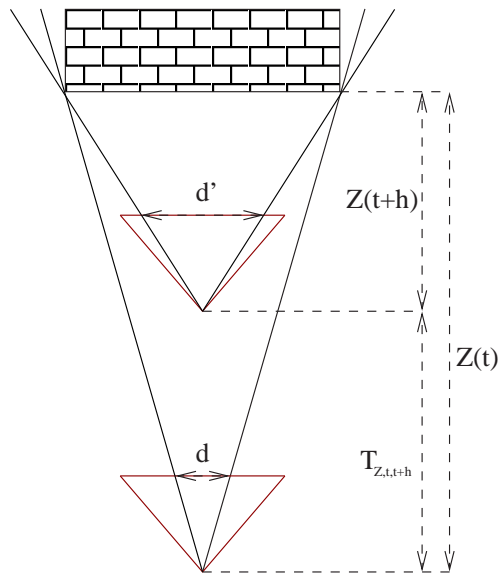


Abbildung 4.1: Skalierung eines Hindernisses

ebenfalls in die Translation einfließen können. Damit lässt sich die Entfernung von Hindernissen wie in Gleichung 4.8 berechnen.

Ω_X - Der Einfluss des Nickwinkels ist nach Gleichung 3.16:

$$-\Omega_X \begin{pmatrix} \frac{uv}{f} \\ f + \frac{v^2}{f} \end{pmatrix} = -\Omega_X \begin{pmatrix} 0 \\ f \end{pmatrix} - \frac{\Omega_X}{f} \begin{pmatrix} uv \\ v^2 \end{pmatrix}.$$

Damit besteht das vom Nickwinkel erzeugte Flussfeld aus einer konstanten Verschiebung in v und einer quadratischen Funktion in den Bildkoordinaten. Der Nickwinkel kann mit weniger als 5° angenommen werden.

Ω_Y - Der Gierwinkel Ω_Y beeinflusst das Flussfeld ebenfalls. Die Anteile des Gierwinkels sind:

$$-\Omega_Y \begin{pmatrix} f + \frac{u^2}{f} \\ \frac{uv}{f} \end{pmatrix} = -\Omega_Y \begin{pmatrix} f \\ 0 \end{pmatrix} - \frac{\Omega_Y}{f} \begin{pmatrix} u^2 \\ uv \end{pmatrix}.$$

Der Gierwinkel geht sowohl konstant (Verschiebung in u) als auch quadratisch in das Flussfeld ein. Größere Gierwinkel sind zwar im Szenenmodell nicht vorgesehen, eine Untersuchung über den entstehenden Fehler ist aber dennoch nötig.

Ω_Z - Die einzelnen Anteile des Wankwinkels auf die Bildkoordinaten sind

$$\Omega_Z \begin{pmatrix} -v \\ u \end{pmatrix}.$$

Der Wankwinkel ist in der Regel klein und daher vernachlässigbar.

Die konstanten Anteile der Rotationsparameter werden direkt in der Schätzung der Translation mit berücksichtigt. Geht man weiter davon aus, dass der Versuchsträger sich nicht in Y -Richtung bewegt (also nicht *hüpft*), so kann die Translation in v direkt zur Berechnung des (relativen) Nickwinkels über den Ansatz der Euler'schen Polygonzugmethode genutzt werden. Die Translation in u ist die Summe aus der Bewegung in X -Richtung und der Translation im Bild aufgrund des Gierwinkels.

Der Einfluss des Wankwinkels auf die Transformation ist in der Regel vernachlässigbar, da von einer ebenen Straße ausgegangen wird. Durch die entstehende Scherung des Bildes bei größeren Wankwinkeln wird die Berechnung einer Skalierungsschätzung geringfügig verfälscht. Der Offset im Skalierungsfaktor ist allerdings bei einem Wankwinkel von weniger als 5° Abweichung vom initialen Zustand beschränkt und kleiner als 0.01. Bei längerer Zufahrt auf das Hindernis und einem Skalierungsfaktor $\gg 1$ ist der Effekt, welchen der Wankwinkel hat, somit vernachlässigbar.

Die quadratischen Terme aus Nick- und Gierwinkel werden mit der Inversen der fokalen Länge f skaliert. Da in allen üblichen Kameras für Hinderniserkennung $f > u$ und $f > v$ gilt, folgt damit:

$$u > \frac{u^2}{f}, \quad u > \frac{uv}{f}, \quad v > \frac{uv}{f} \quad \text{und} \quad v > \frac{v^2}{f}.$$

Die Effekte der quadratischen Anteile sind also geringer als eine Skalierung mit dem jeweiligen Nick- oder Gierwinkel. Ebenso wie für den Wankwinkel kann bei Skalierungsfaktoren $\gg 1$ auch der Effekt von Nick- und Gierwinkel auf das Bewegungsfeld vernachlässigt werden. Vorsicht ist allerdings geboten, wenn größere Nick- oder Gierbewegungen stattfinden. Hier bietet es sich an, das Bild um diese Bewegungen zu bereinigen. Da die Effekte der Rotationsparameter nicht tiefenabhängig sind, können die Rotationen aus dem Bild heraus gerechnet werden [51].

Wie die Untersuchungen in diesem Abschnitt zeigen, ist eine robuste Schätzung der Entfernung erst bei einem Skalierungsfaktor möglich, der signifikant von 1.0 abweicht. Je größer der Skalierungsfaktor ist, desto besser wird die relative Genauigkeit der Entfernungsschätzung. Hinzu kommt, dass der relative Nickwinkel und der Wankwinkel im Mittel 0 sind und daher die Effekte dieser Rotationsparameter auf die Skalierungsschätzung im Mittel 0 sind. Es bleibt zu zeigen, wie die Bewegung des Versuchsträgers auf eine Kamerabewegung in Z -Richtung übertragen werden kann.

4.2.4 Übertragung der Bewegung des Versuchsträgers auf die Kamera

In den letzten Abschnitten wurde eine Bewegung der Kamera parallel zur Z -Achse des Kamerakoordinatensystems betrachtet. In der Realität ist die Kamera allerdings auf einem Versuchsträger angebracht. Aufgrund der relativen Orientierung von Kamera und Versuchsträger muss die Translation eines Versuchsträgers auf die Kameratranslation übertragen werden. Eine Translation des Versuchsträgers ist auch eine Translation der Kamera. Da die Translation in

realisieren:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = f(\mathbf{x}, \mu) = s \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} u \\ v \end{pmatrix}. \quad (4.9)$$

Die Parameter der Schätzung sind also $\mu = (s, u, v)^t$. Der Trackingalgorithmus benötigt zur effizienten Berechnung eine Zerlegung der Matrix M in $M = M_0 \Sigma(\mu)$ (vergleiche Gleichung 3.34). Genauer muss folgende Zerlegung durchgeführt werden:

$$f_{\mathbf{x}}(\mathbf{x}, \mu)^{-1} f_{\mu}(\mathbf{x}, \mu) = \Gamma(\mathbf{x}) \Sigma(\mu). \quad (4.10)$$

Mit den Ableitungen nach dem Bildpunkt und den Bewegungsparametern

$$f_{\mathbf{x}}(\mathbf{x}, \mu) = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \quad \text{und} \quad f_{\mu}(\mathbf{x}, \mu) = \begin{pmatrix} x & 1 & 0 \\ y & 0 & 1 \end{pmatrix} \quad (4.11)$$

führt dies zu

$$\Gamma(\mathbf{x}) = \begin{pmatrix} x & 1 & 0 \\ y & 0 & 1 \end{pmatrix} \quad \text{und} \quad \Sigma(\mu) = \begin{pmatrix} \frac{1}{s} & 0 \\ 0 & \frac{1}{s} \end{pmatrix}. \quad (4.12)$$

Unter der Annahme, dass in der ausgesuchten Bildregion ein Hindernis abgebildet ist, wird aus der geschätzten Skalierung direkt die Entfernung berechnet. Diese wird mit einem Tiefpassfilter unter Berücksichtigung der Standardabweichungen für die geschätzten Größen gefiltert. Außerdem erfolgt eine Schätzung des relativen Nickwinkels zwischen den einzelnen Frames, wie in diesem Kapitel beschrieben.

Um Hypothesen für potentielle Hindernisse aufzustellen, wird der Bereich vor dem Fahrzeug untersucht. Ein vorher definierter Bildabschnitt wird in mehrere Regionen unterteilt, die unabhängig voneinander getrackt werden. Fällt der Korrelationskoeffizient einer Region unter einen vorher definierten Schwellwert, so wird diese Region erneut initialisiert. Die einzelnen Entfernungen der Tracker werden unter Berücksichtigung einer Gewichtsfunktion $G(x)$ auf die x -Achse des Bildes projiziert. Für die Regionen $\{R_i\}_i$ sei $D(R_i)$ die Distanz aus der Skalierungsschätzung, $M(R_i)$ der Mittelpunkt und $S(R_i)$ die Skalierung der Region R_i . Mit der charakteristischen Funktion

$$\chi_{R_i}(x) = \begin{cases} 1 & x \in R_i \\ 0 & \text{sonst} \end{cases}$$

und $G(x)$ ergibt sich damit für die x -Koordinaten die Entfernung:

$$D(x) = \frac{1}{\sum_i \chi_{R_i}(x) G_{R_i}(x)} \sum_i \chi_{R_i}(x) G_{R_i}(x) D(R_i) \quad (4.13)$$

mit $G_{R_i}(x) = G\left(\frac{x - M(R_i)}{S(R_i)}\right)$.

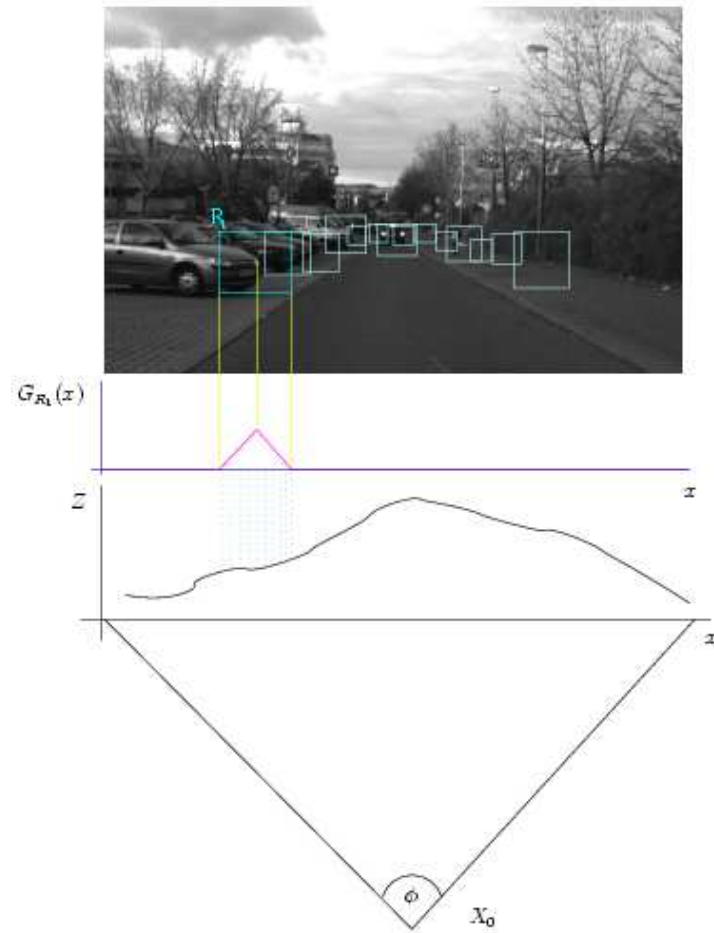


Abbildung 4.3: Schematischer Ablauf der Hypothesengenerierung. Die Entfernung aus Region R_1 wird mit der Gewichtungsfunktion $G_{R_1}(x)$ multipliziert und gewichtet auf die x Achse projiziert.

Schematisch ist der Algorithmus in Abbildung 4.3 dargestellt. Als Gewichtungsfunktion wurde die Hutfunktion gewählt. In die Gewichtungsfunktion können auch die geschätzten Standardabweichungen der Parameter mitberücksichtigt werden (vgl. Abschnitt 3.5.6.2). Über Fehlerfortpflanzung kann aus der Standardabweichung des Skalierungsfaktors die Ungenauigkeit der Entfernung berechnet werden. Die einzelnen Koordinaten der x -Achse lassen sich als diskrete Winkelauflösungen des Kameraöffnungswinkels ϕ interpretieren. Befindet sich ein Hindernis innerhalb eines vordefinierten Abstandes zum Versuchsträger, wird eine Hindernishypothese generiert.

Kapitel 5

Hypothesenverifizierung

Das in Kapitel 4 vorgestellte Verfahren erzeugt Hindernishypothesen. Der zweite Schritt einer Hinderniserkennung, die Hypothesenverifizierung, ist die Bestätigung oder Verwerfung einer solchen Hypothese, die aus Sichtwinkel (Winkel zur Z -Achse des Kamerakoordinatensystems) und Entfernung eines potentiellen Hindernisses besteht. Die Hindernishypothese beinhaltet keine Aussagen über Breite und Höhe eines Hindernisses. Ziel der Hypothesenverifizierung ist es, bei Bestätigung der Hypothese die Breite und absolute Position des Hindernisses zu bestimmen. Ist kein eindeutiges Ergebnis möglich, soll der Algorithmus dies als Aussage formulieren.

Die nachfolgenden Abschnitte erläutern das im Rahmen dieser Arbeit entwickelte Konzept zur Hypothesenverifizierung. Die mathematische Herleitung in 5.2 beschäftigt sich mit der Umsetzung des Konzepts zu einem Algorithmus. Es folgt in Abschnitt 5.3 eine Übersicht über die Implementierung des Algorithmus. Als Eingabe ist jeglicher Sensor geeignet, der Hypothesen für Hindernisse erstellen kann. Abschnitt 5.4 beschäftigt sich daher mit den Möglichkeiten einer Multisensorfusion.

5.1 Konzept

Mit Kenntnis des Sichtwinkels und der Entfernung kann unter der Annahme einer ebenen Straße der Fußpunkt eines potentiellen Hindernisses im Weltkoordinatensystem berechnet werden. Unter Benutzung der projektiven Abbildung und Annahmen über Breite und Höhe eines Hindernisses lässt sich die Bildregion, in die das potentielle Hindernis abgebildet wird, bestimmen.

Aufgrund des Bauklotzmodells ist in der berechneten Bildregion entweder ein stehendes Hindernis, die Straße (Grundebene) oder beides abgebildet. Die Bildregion wird so unterteilt, dass in jeder Teilregion entweder die Frontfläche eines Hindernisses mit bekannter Entfernung $Z = d$ oder die Grundebene abgebildet ist. Für beide Ebenen kann die Transformation der abgebildeten Region unter bekannter Translation des Versuchsträgers berechnet werden. Ziel ist nun ein Hypothesenvergleich zwischen der Hindernishypothese und der Straßenhypothese.

Die Bildregion wird entsprechend den beiden möglichen Transformationen verzerrt. Die Grauwerte der transformierten Bildregion im aktuellen Bild werden mit den Grauwerten der initialen Bildregion, der Zielregion, verglichen. Daraus soll eine Aussage darüber gemacht werden, welche Transformation für die Bildregion wahrscheinlicher ist. Problematisch sind in einem solchen Ansatz Nick-, Wank-, Gier- und Translationsbewegungen, welche nicht erfasst und nicht modelliert werden. Außerdem ist die Entfernung meistens mit einer Ungenauigkeit behaftet. Für kleine Eigenbewegungen können solche Einflüsse auf die Transformation durch Translation der Region ausgeglichen werden. Trotzdem führen sie für größere Eigenbewegungen, bei konstanter Bewegung für größere Zeitabstände, dazu, dass die Modellannahmen nicht mehr stimmen. Die Grauwertdifferenzen der jeweils aktuellen Bildregion zur Zielregion steigen dann aufgrund falscher Annahmen für beide Transformationsannahmen. Eine Aussage ist in diesem Fall nicht mehr möglich. Hier müssen vernünftige Annahmen getroffen werden, um einen solchen Fall abzufangen.

5.2 Mathematische Herleitung

Seien d , ϕ die Entfernung und der Sichtwinkel für den Schwerpunkt eines zu verifizierenden Hindernisses. Mit den Kollinearitätsgleichungen 3.9 und 3.10 ergeben sich daraus die Bildkoordinaten $(x, y)^t$ für den Fußpunkt $Y = 0$:

$$x = \frac{f}{s_u} \frac{r_{11}(d \sin \phi - X_0) - r_{12}Y_0 + r_{13}(d - Z_0)}{r_{31}(d \sin \phi - X_0) - r_{32}Y_0 + r_{33}(d - Z_0)} + u_0 \quad \text{und} \quad (5.1)$$

$$y = -\frac{f}{s_v} \frac{r_{21}(d \sin \phi - X_0) - r_{22}Y_0 + r_{23}(d - Z_0)}{r_{31}(d \sin \phi - X_0) - r_{32}Y_0 + r_{33}(d - Z_0)} + v_0. \quad (5.2)$$

Damit ergibt sich die zu untersuchende Bildregion $R = [x - w, x + w] \times [y, y + h]$ mit einer von der Entfernung abhängigen Breite $2w$ und einer Höhe h in Bildkoordinaten.

Die Translation des Versuchsträgers relativ zur statischen Szene sei $T_Z < d$. Die Verschiebung des Fahrzeuges kann mit einer Verschiebung des Weltkoordinatensystems in Z -Richtung um T_Z modelliert werden. Damit gilt für jeden Weltpunkt nach Verschiebung des Koordinatensystems:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = B \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z - T_Z \end{pmatrix}. \quad (5.3)$$

Um andere Bewegung zu modellieren, ist Gleichung 5.3 entsprechend anzupassen. Die Transformation einer Bildregion ($\mathbf{x} \rightarrow f(\mathbf{x}) = \mathbf{x}'$) bei bekannter Translation des Versuchsträgers kann wie folgt modelliert werden:

$$\mathbf{x}' = f(\mathbf{x}, T_Z) = P(B(P_{\text{INV}}(\mathbf{x}))) \quad \text{mit} \quad \mathbf{X} = P_{\text{INV}}(\mathbf{x}) \Rightarrow P(\mathbf{X}) = \mathbf{x}, \quad (5.4)$$

wobei $P(\mathbf{X}) = \mathbf{x}$ die projektive Abbildung eines Weltpunktes $\mathbf{X} \in \mathbb{R}^3$ in den Bildpunkt $\mathbf{x} \in \mathbb{R}^2$ ist. Da beim Abbildungsvorgang eine Dimension verloren geht, ist dieser nicht invertierbar. Unter der zusätzlichen Annahme, dass eine Koordinate des Vektors X bekannt ist, lässt sich die Inverse der Abbildung angeben. Für die beiden zu untersuchenden Fälle ist entweder die Entfernung eines Hindernisses $Z = d$ bekannt oder die Höhe der Grundfläche $Y = 0$. Damit ergeben sich die beiden in Abschnitt 3.6 hergeleiteten Matrizen ${}^B T_O$ für Hindernisse und ${}^B T_S$ für die Grundfläche. Diese Matrizen sind in der Regel invertierbar. Aufgrund der perspektiven Abbildung ist die Inverse der Straßenebene jedoch für Bildpunkte, die über dem Horizont liegen, nicht direkt interpretierbar. Soll die Objekthypothese verworfen werden, so bedeutet das, dass kein Objekt in der Zielregion abgebildet ist und daher die Region über dem Horizont sich nicht verändert (insbesondere gilt hier die Annahme, dass über dem Horizont die Objekte unendlich weit entfernt sind). Die in diesem Kapitel durchgeführten Betrachtungen für die Straßenebene beziehen sich, wenn nicht anders vermerkt, immer auf die Bildregion unter dem Horizont. Im Folgenden sei

$${}^O T_B = ({}^B T_O)^{-1} \quad \text{und} \quad {}^S T_B = ({}^B T_S)^{-1}. \quad (5.5)$$

Damit lässt sich die Transformation für ein Objekt in gegebener Entfernung d ($T_{O,d,B}(\mathbf{x})$) und für die Grundebene ($T_{S,B}(\mathbf{x})$) in allgemeiner Form angeben:

$$T_{O,d,B}(\mathbf{x}) = P(B({}^O T_B(\mathbf{x}))) \quad (5.6)$$

$$T_{S,B}(\mathbf{x}) = P(B({}^S T_B(\mathbf{x}))). \quad (5.7)$$

Mit Gleichung 5.3 ergibt sich für homogene Bildkoordinaten $\tilde{\mathbf{x}}$:

$$T_{O,d,B}(\mathbf{x}) = KR[I | -\mathbf{X}_0] \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d - T_Z \\ 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} 1 & 0 & -X_0 \\ 0 & 1 & -Y_0 \\ 0 & 0 & d - Z_0 \end{bmatrix}^{-1} R^{-1} K^{-1} \tilde{\mathbf{x}} \quad (5.8)$$

und

$$T_{S,B}(\mathbf{x}) = KR[I | -\mathbf{X}_0] \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & -T_Z \\ 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} 1 & 0 & -X_0 \\ 0 & 0 & -Y_0 \\ 0 & 1 & -Z_0 \end{bmatrix}^{-1} R^{-1} K^{-1} \tilde{\mathbf{x}}. \quad (5.9)$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{T_Z}{Y_0} & 1 \end{bmatrix}$$

Lässt man die Rotation der Kamera und die innere Orientierung (Skalierung und Translation des Bildkoordinatensystems) außer Acht, so ergibt sich im Fall von stehenden Hindernissen eine Skalierung, während die Transformation der Straße perspektivische Anteile (Einträge in der dritten Zeile der Matrix) aufweist. Diese perspektivischen Anteile sind abhängig von der Kamerahöhe und der gefahrenen Distanz, während die Skalierung in Gleichung 5.8 neben der gefahrenen Distanz abhängig ist von der Hindernisentfernung. Stimmt entweder die Kamerahöhe oder die gefahrene Distanz nicht, können keine zuverlässigen Aussagen getroffen werden. Außerdem stimmt die Fußpunktberechnung bei falsch angenommener Kamerahöhe nicht. Ist ein Hindernis mit anderer Entfernung als angegeben in der Region abgebildet, ist auch keine zuverlässige Aussage möglich, da keine der Transformationen dem Modell entsprechen. Dabei ist der relative Fehler in der Kamerahöhe und der Entfernungsschätzung wichtig. Kleine Fehler können durch eine Translation ausgeglichen werden.

Zudem beeinflussen nicht modellierte Bewegungen des Versuchsträgers das Ergebnis. Je größer diese sind, desto größer werden die Fehler in den Transformationsmatrizen. Folglich nehmen die Grauwertdifferenzen für beide Transformationen aufgrund von sich akkumulierenden Fehlern in der Eigenbewegung mit der Zeit zu und das Ergebnis wird unzuverlässig.

Auf der anderen Seite sind erst nach einer gewissen Fahrzeugbewegung die Transformationen für potentielle Hindernisse und für die Grundebene signifikant unterschiedlich. Liegt keine Fahrzeugbewegung vor, ist die Transformation die Identität und keine Aussage ist möglich. Je größer die zurückgelegte Bewegung des Versuchsträgers ist, desto besser lässt sich die tatsächliche Transformation einer der beiden Transformationsannahmen zuordnen. Hier muss ein Kompromiss zwischen Informationsverlust aufgrund nicht modellierter Fehler und Informationsgewinn aufgrund besserer Unterscheidung der Transformationen gefunden werden. Die nicht modellierten Fehler können minimiert werden, indem eine robuste Schätzung der Eigenbewegung (Rotationsbewegung) durchgeführt wird und das Bild der Rotation des Versuchsträgers entsprechend angepasst wird. Ein solcher Ansatz wird in [45] untersucht.

Der folgende Abschnitt untersucht die Differenz der beiden Transformationen. Es soll bestimmt werden, wie die modellierten Verschiebungen von Bildpunkten in der Zielregion unter den beiden Transformationsannahmen voneinander abweichen. Zu erwarten ist eine Abhängigkeit von der zurückgelegten Strecke des Versuchsträgers, T_Z . Abschnitt 5.2.2 befasst sich mit dem Vergleich und der Interpretation beider Transformationen, um eine Aussage zu treffen, welche Annahme die wahrscheinlichere ist.

5.2.1 Planar motion parallax

Ziel der Hypothesenverifizierung ist es, stehende Hindernisse von der Straße zu unterscheiden. Dies ist möglich, da stehende Hindernisse ein anderes Flussfeld im Bild generieren. Die Differenz der beiden Flussfelder wird auch *planar motion parallax* genannt. Für die mathematische

Modellierung beider Flussfelder lassen sich homogene Transformationsmatrizen angeben (siehe Abschnitt 5.2). Die zugrunde liegende Abbildung ist eine Homographie. In diesem Abschnitt werden diese Matrizen untersucht und eine Aussage über die Differenz der Flussfelder abgeleitet. Für Algorithmen, die auf Bewegungsanalyse einzelner Bildpunkte basieren, ist diese Differenz ein Maß für die notwendige Genauigkeit der Zuordnung. Für den Vergleich basierend auf Bildregionen ist sie ein Maß dafür, wie signifikant zwei Transformationen voneinander abweichen.

Gegeben sei eine Kamera mit äußerer Orientierung

$$R = Id, \quad X_0 = Z_0 = 0, \quad Y_0 = H \quad (5.10)$$

und ein potentiell Hindernis in der Entfernung d . Die Bewegung der Kamera sei eine Translation in Z -Richtung um T . Außerdem ist die fokale Länge f bekannt. Die folgenden Betrachtungen beziehen sich auf Raumpunkte X im Weltkoordinatensystem und Bildpunkte x im Sensorkoordinatensystem. Die Kalibriermatrix K lautet:

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5.11)$$

Die Berechnung der Differenz beider Transformationen lässt sich aufteilen in eine Betrachtung für Bildpunkte oberhalb und unterhalb des Horizonts.

- oberhalb des Horizonts:

Da oberhalb des Horizonts im Bild keine Straße abgebildet ist, wird statt dessen die Abbildung eines unendlich weit entfernten Hindernisses angenommen. Die Transformation entspricht also der Identität. Zusammen mit der in Abschnitt 5.2 hergeleiteten Transformation für stehende Objekte folgt für die Differenz:

$$\|T_{S,B}(x) - T_{O,d,B}(x)\| = \left\| x - \frac{d}{d-T}x \right\| = \left| \frac{T}{d-T} \right| \|x\|. \quad (5.12)$$

- unterhalb des Horizonts:

Die Transformation eines Bildpunktes unter der Annahme einer ebenen Straße geschieht nach Abschnitt 5.2 wie folgt:

$$T_{S,B} = KR \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{T}{H} & 1 \end{bmatrix} R^{-1}K^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{T}{Hf} & 1 \end{bmatrix}. \quad (5.13)$$

Ein stehendes Hindernis in der Entfernung $Z = d$ erzeugt bei Translation eine Skalierung

im Bild:

$$T_{O,d,B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{d-T}{d} \end{bmatrix}. \quad (5.14)$$

Umgeformt ergibt sich für die euklidischen Bildkoordinaten:

$$T_{S,B}(\mathbf{x}) = \frac{Hf}{Hf - yT} \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{und} \quad T_{O,d,B}(\mathbf{x}) = \frac{d}{d-T} \begin{pmatrix} x \\ y \end{pmatrix}. \quad (5.15)$$

Daraus lässt sich nun die Differenz berechnen:

$$\|T_{S,B}(\mathbf{x}) - T_{O,d,B}(\mathbf{x})\| = \left| \frac{Hf}{Hf - yT} - \frac{d}{d-T} \right| \left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\| \quad (5.16)$$

$$= \left| \frac{T}{d-T} \right| \left| \frac{yd - fH}{fH - yT} \right| \left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|. \quad (5.17)$$

Wie man erkennt, ist die Differenz abhängig von der Bildzeile. Dies ist darauf zurückzuführen, dass die Transformation der Straße perspektivische Effekte aufweist. Weiter ist zu erkennen, dass für

$$\frac{fH}{y} = d$$

die Differenz gleich Null ist. Die Bildzeile $y = \frac{fH}{d}$ entspricht dem Fußpunkt eines Hindernisses in gegebener Entfernung d . Außerdem ist die Differenz im Punkt

$$\mathbf{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

gleich Null. Dies entspricht dem *focus of expansion*. Weiter kann vorausgesetzt werden, dass $d > T$ und $fH > yT$, da ansonsten der Bildpunkt im zweiten Bild nicht abgebildet ist. Mit diesen zwei Voraussetzungen ist die Differenz annähernd proportional zur Translation der Kamera in Z -Richtung.

Die Differenz lässt sich nun für bekannte Kamerahöhe, fokale Länge und Entfernung eines potentiellen Hindernisses berechnen. Abbildung 5.1 zeigt karbkodierte Differenzbilder für verschiedene Translationen des Versuchsträgers für ein potentielles Hindernis in 10m Entfernung. Man erkennt den FOE und den Fußpunkt des potentiellen Hindernisses. Hier lassen sich die zwei Flussfelder der Straße und des potentiellen Hindernisses nicht voneinander unterscheiden. Der Bildbereich unterhalb des Fußpunktes ist zur Hindernisverifikation irrelevant, da ein Hindernis sich oberhalb der Straße befindet. Über dem Horizont ist das erwartete radiale Flussfeld eines stehenden Hindernisses erkennbar. Abbildung 5.1 zeigt deutlich, wo die Schwierigkeiten

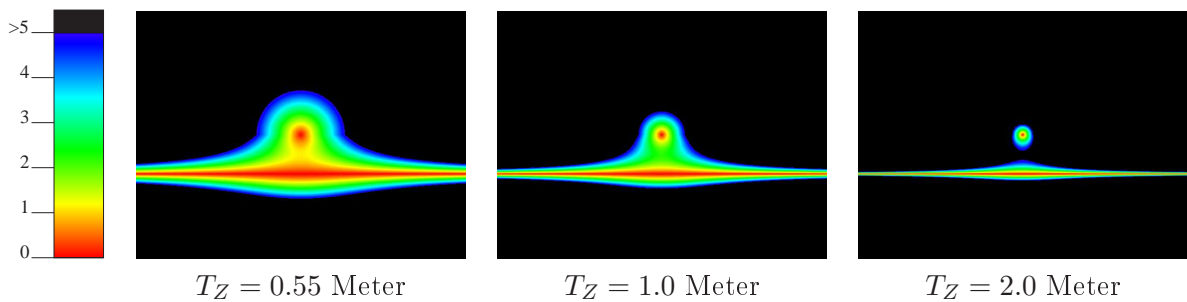


Abbildung 5.1: Planar motion parallax in Pixel. Der Farbverlauf zeigt die Differenz des Flussfeldes eines stehenden Hindernisses in 10m Entfernung und der Stra enebene unterhalb des Horizonts respektive der Identit t oberhalb des Horizonts. Die Kamerah he ist 1.10m, die fokale L nge 695 Pixel. Die zur ckgelegte Strecke von 0.55m entspricht bei eine Geschwindigkeit des Versuchstr gers von 50km/h in etwa 40ms. Schwarz im Bild (640×480 Pixel) eingezeichnet sind Verschiebungsunterschiede gr o er als 5 Pixel.

des Vergleichs der beiden Transformationsannahmen liegen. Der Bereich zwischen Fu punkt eines potentiellen Hindernisses und FOE weist in N he des FOE nur sehr kleine Differenzen auf. F r ein potentielles Hindernis in 30m Entfernung ist in Abbildung 5.2 die Differenz in

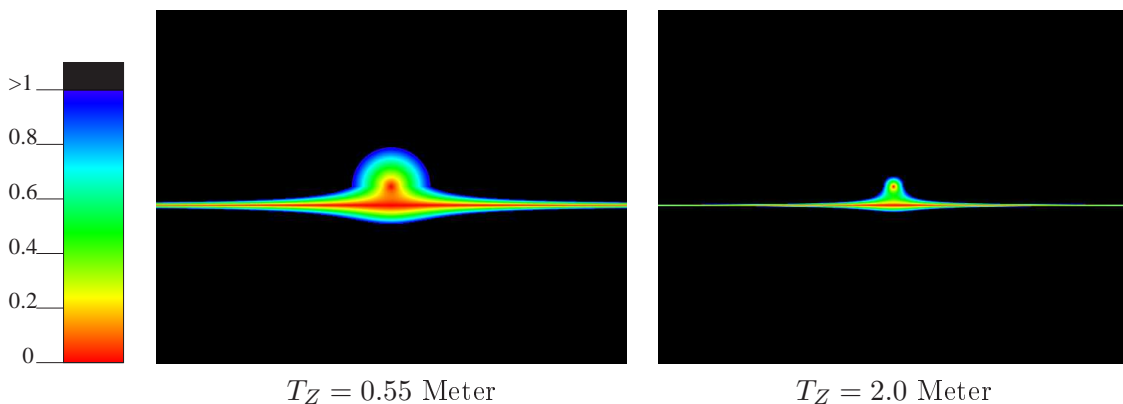


Abbildung 5.2: Planar motion parallax in Pixel f r ein Hindernis in 30m Entfernung. Der Farbverlauf kodiert die Flussfelddifferenz des Hindernisses und der Stra enebene. Die Kamerah he ist 1.10m, die fokale L nge 695 Pixel. Schwarz im Bild (640×480 Pixel) eingezeichnet sind Verschiebungsunterschiede gr o er als 1 Pixel.

Pixel dargestellt. Man erkennt, dass bei einer gefahrenen Distanz von 2m die Differenz im Bereich zwischen FOE und Fu punkt immer noch unter einem Pixel ist. In diesem Bereich m ssen Verfahren, die mit einzelnen Pixeln rechnen, eine sehr hohe Genauigkeit vorweisen. Um eine robuste Unterscheidung der beiden Transformationsannahmen durchzuf hren ist es besser, Regionen zu betrachten und nicht einzelne Pixel.

In [53] wurde der eindimensionale Fall der *planar motion parallax* untersucht. Dies entspricht dem Spezialfall der hier gezeigten Herleitung für den Schnitt der Bildebene mit der Y - Z Ebene. Abbildung 5.3 zeigt den Schnitt für die in Abbildung 5.1 und 5.2 dargestellten

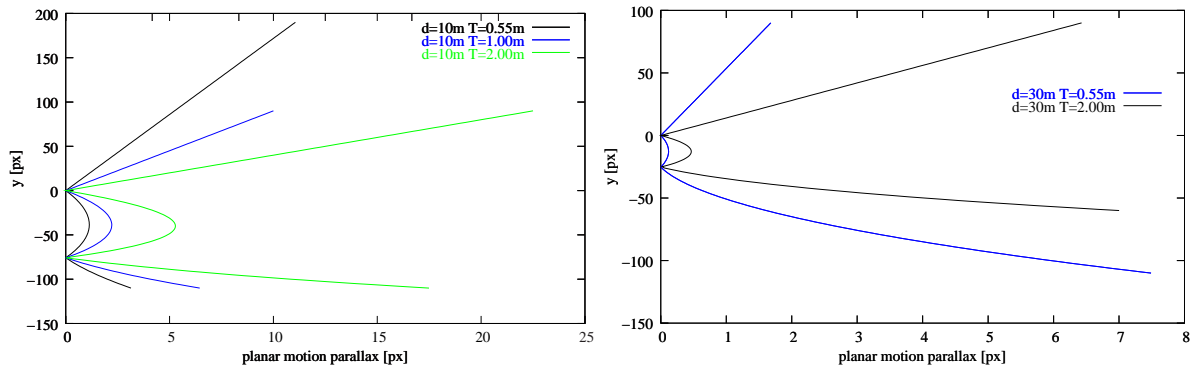


Abbildung 5.3: Planar motion parallax - Schnitt durch YZ Ebene (entspricht im Bild $x=0$) für Abbildungen 5.1 und 5.2.

Transformationsdifferenzen. Untersucht wurde die notwendige Geschwindigkeit des Versuchsträgers, um mit pixelgenauen optischen Flussverfahren in zwei aufeinander folgenden Bildern in Echtzeit eine Unterscheidung zwischen Straßenfluss (Verschiebung im Bild von abgebildeten Punkten auf der Straßenebene) und Fluss eines Objektes in gegebener Entfernung durchzuführen. Tabelle 5.1 zeigt einen Überblick über die Ergebnisse.

Objekthöhe [m]	Fahrzeuggeschwindigkeit [km/h]
0,1	112,5
0.2	72
0.3	58,5

Tabelle 5.1: Notwendige Geschwindigkeit zur Hindernisdetektion für pixelgenaue Flussverfahren. ($f = 695\text{px}$, $H = 1\text{m}$, $d = 10\text{m}$, Framerate = 25Hz).

Offensichtlich sind 40 Millisekunden nicht genug, um ein Hindernis zu erkennen. Eine Notbremsung aus $58,5\text{ km/h}$ würde $13,2\text{m}$ benötigen, aber das Hindernis ist lediglich 10m entfernt. Folglich ist eine frühere Erkennung notwendig. Betrachtet man das Objekt über einen längeren Zeitpunkt, so verringert sich die Mindestgeschwindigkeit drastisch. Allerdings macht es nicht viel Sinn, Bilder einer Sequenz zu überspringen, nur um Objekte besser zu detektieren. Idealerweise sollten alle Bilder genutzt werden.

Der Vorteil von regionenbasierten Verfahren liegt darin, dass die Informationen aller Pixel in der Region genutzt werden, also eine räumliche Integration geschieht. Integriert man das Ergebnis zusätzlich zeitlich, ist eine robuste Unterscheidung der beiden möglichen Transfor-

mationen erreicht. Um Robustheit zu garantieren, soll der Algorithmus melden, wenn keine Zuordnung aufgrund zu geringer Information (z. B. das Fahrzeug bewegt sich nicht) getroffen werden kann. In diesem Abschnitt wurde ein Ansatz vorgestellt, mit dem die Differenz der Flussfelder beider Hypothesen berechnet werden kann. Es wurde gezeigt, wie diese Information dazu verwendet werden kann, die Mindesttranslation der Kamera (abhängig von der Genauigkeit des verwendeten Trackers sowie der Hindernisentfernung und -höhe) bis zu einer sicheren Aussage der Hypothesenverifizierung zu bestimmen. Der nächste Abschnitt befasst sich mit der Interpretation der Grauwertdifferenzen beider Annahmen.

5.2.2 Interpretation und Ergebnis

Um die Hypothese *stehendes Hindernis* von der Hypothese *Straße* zu unterscheiden, ist es notwendig, Wahrscheinlichkeiten für beide Hypothesen aufzustellen. Betrachtet man, unabhängig von der Transformation T , die Grauwerte der Zielregion (I_0) mit den Grauwerten der transformierten Regionen (I_t), so kann die Wahrscheinlichkeit, dass ein Pixel \mathbf{x} der Bewegungsvorschrift $T(\mathbf{x})$ folgt, wie folgt modelliert werden:

$$p(\mathbf{x} \rightarrow T(\mathbf{x})) \propto e^{-|I_0(\mathbf{x})-I_t(\mathbf{x})|} \quad \text{oder} \quad (5.18)$$

$$p(\mathbf{x} \rightarrow T(\mathbf{x})) \propto e^{-(I_0(\mathbf{x})-I_t(\mathbf{x}))^2} \quad (5.19)$$

Beide Wahrscheinlichkeiten haben die folgenden Eigenschaften:

- p ist maximal, wenn die Grauwertdifferenz Null ist und
- p sinkt mit wachsender Grauwertdifferenz.

Für die Wahrscheinlichkeit, dass eine Region R der Transformation T folgt, sind die einzelnen Wahrscheinlichkeiten für die Pixel dieser Region zu multiplizieren. Damit ergibt sich

$$p(R \rightarrow T(R)) = \prod_{\mathbf{x} \in R} p(\mathbf{x} \rightarrow T(\mathbf{x})) \quad (5.20)$$

für die gesamte Region und im konkreten Fall für die beiden Wahrscheinlichkeitsmodelle:

$$P_{SAD}(R, T) \propto e^{-\sum_{\mathbf{x} \in R} |I_0(\mathbf{x})-I_t(\mathbf{x})|} \quad (5.21)$$

$$P_{SSD}(R, T) \propto e^{-\sum_{\mathbf{x} \in R} (I_0(\mathbf{x})-I_t(\mathbf{x}))^2}. \quad (5.22)$$

Für die zwei Transformationen $T_{O,d,B}$ und $T_{S,B}$, welche zu den Hypothesen *stehendes Hindernis* und *Straße* gehören, können nun Wahrscheinlichkeiten angegeben werden und der Quotient beider Wahrscheinlichkeiten kann berechnet werden (vgl. Gleichung 5.23). Der Vergleich mit dem Wert 1.0 zeigt dann, welche der beiden Hypothesen wahrscheinlicher ist. Wegen Interpolation der Grauwerte und Rechenungenauigkeiten lässt man einen Bereich $[1 - \epsilon, 1 + \epsilon]$ zu, in

dem keine Aussage möglich ist. Für das SAD Differenzmaß (siehe Abschnitt 3.5.6) ergibt dies:

$$\frac{P_{SAD}(R, T_{O,d,B})}{P_{SAD}(R, T_{S,B})} = \begin{cases} > 1 - \epsilon & \Rightarrow \text{stehendes Hindernis} \\ < 1 + \epsilon & \Rightarrow \text{Straße} \\ \text{sonst keine Aussage} & \end{cases} \quad (5.23)$$

Anstatt die größere Wahrscheinlichkeit zu bestimmen, kann auch der kleinere negative Logarithmus der Wahrscheinlichkeit bestimmt werden. Der negative (2er-)Logarithmus einer Wahrscheinlichkeit wird auch *bit* genannt, eine Einheit für Information. Je kleiner die Wahrscheinlichkeit ist, desto größer ist damit der Informationsgehalt. Für den hier betrachteten Fall heißt das: größere Grauwertdifferenzen bedeuten einen größeren Informationsgehalt. Erst wenn die Grauwertdifferenzen für eine Hypothese groß genug sind, kann der Algorithmus eine Entscheidung treffen. Daher ist der Informationsbegriff in diesem Zusammenhang auch direkt übertragbar. Für Gleichung 5.23 bedeutet eine Logarithmierung:

$$\begin{aligned} -\log \left(\frac{P_{SAD}(R, T_{O,d,B})}{P_{SAD}(R, T_{S,B})} \right) &= -\log e^{-\sum_{\mathbf{x} \in R} |I_0(\mathbf{x}) - I_{O,d,B}(\mathbf{x})|} + \log e^{-\sum_{\mathbf{x} \in R} |I_0(\mathbf{x}) - I_{S,B}(\mathbf{x})|} \\ &= \sum_{\mathbf{x} \in R} |I_0(\mathbf{x}) - I_{O,d,B}(\mathbf{x})| - \sum_{\mathbf{x} \in R} |I_0(\mathbf{x}) - I_{S,B}(\mathbf{x})| \\ &= \begin{cases} < 0 - \delta & \Rightarrow \text{stehendes Hindernis} \\ > 0 + \delta & \Rightarrow \text{Straße} \\ \text{sonst keine Aussage} & \end{cases} \quad (5.24) \end{aligned}$$

wobei $[0 - \delta, 0 + \delta]$ der Bereich ist, in dem keine Aussage möglich ist. Damit erhält man eine Interpretation, in der direkt das Ergebnis eines Trackingvorgangs in Form von SAD oder SSD benutzt wird. Gleichung 5.24 gibt an, welche der beiden Hypothesen einen höheren Informationsgehalt hat.

5.3 Implementierung

Die Bildregion der Breite $2w$ und Höhe h wird vertikal in n Regionen fester Breite eingeteilt. Diese können sich überlappen. Seien M_1, \dots, M_{n-1} die Mittelpunkte dieser Regionen. Durch die Aufteilung wird für große n gewährleistet, dass bis auf Ausnahmen jede Unterregion entweder nur ein stehendes Hindernis oder nur Straße abbildet. Potentielle Hindernisse bestimmen die Transformation der Unterregion, auch wenn nicht alle Bildpunkte der Region zu Punkten auf dem Hindernis gehören. Dass diese Annahme gerechtfertigt ist, zeigen die Untersuchungen in Kapitel 6.

Die einzelnen Unterregionen werden entsprechend den Transformationsvorschriften $T_{S,B}$ und $T_{O,d,B}$ transformiert. Um Ungenauigkeiten in der Eigenbewegung zu kompensieren, wird anschließend ein Region Tracker benutzt. Das Least Squares Verfahren rechnet eine Ausgleichs-

chung unter Annahme einer Translation im Bild. Für kleine Ungenauigkeiten ist dies ausreichend. Das Region Tracking Verfahren liefert als Ergebnis den SAD oder SSD Wert der Unterregion für beide Transformationsannahmen. Diese Werte können direkt interpretiert und so für jede Unterregion die Objekthypothese bestätigt oder verworfen werden.

Sei S_i die (quadratische) Summe der Grauwertdifferenzen der Unterregion n_i . Die einzelnen $\{S_i\}_i$ werden die Entfernungen wie in Gleichung 4.13 auf die x-Achse projiziert. Auf die projizierten Werte wird ein zeitlicher Filtervorgang (moving average) angewendet. Anschließend werden zusammenhängende Regionen negativer Werte als Hindernis bestätigt. Für zusammenhängende Regionen positiver Werte wird die Hindernishypothese abgelehnt.

5.3.1 Stabilisierung der Tracks

Das Region Tracking ist insbesondere für kleine Regionen nicht immer erfolgreich. Es kann vorkommen, dass falsche Ergebnisse sich in der Iteration aufsummieren und einzelne Unterregionen falsch getrackt werden. Für den vorliegenden Fall der Hypothesenverifizierung durch Vergleich zweier Hypothesen bietet es sich an, die Unterregionen zu stabilisieren. Dabei ist nützlich, dass Homographien geradentreu sind (vgl. Abb. 5.4). Die zu überprüfenden Bildregionen werden so in Unterregionen aufgeteilt, dass die Mittelpunkte der Unterregionen auf einer Geraden liegen. Allgemein lautet die Gleichung für die Gerade mit den Mittelpunkten

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} : \exists a, b \in \mathbb{R} : y_i = ax_i + b, \quad x_1 < x_2 < \dots \quad (5.25)$$

Im betrachteten Fall ist die y -Koordinate der Mittelpunkte identisch, also $y_i = b_i$. Angenommen, in allen Teilregionen ist bis auf einzelne Ausnahmen entweder das potentielle Hindernis oder aber die Straße abgebildet. Betrachtet man die Linie durch diese Mittelpunkte für eine der beiden Transformationsannahmen, so bedeutet dies: durch die Mittelpunkte lässt sich auch nach der Transformation eine ausgleichende Gerade legen. Sei

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} \quad (5.26)$$

der Mittelpunkt nach der Transformation. Damit gilt:

$$\exists a_t, b_t \in \mathbb{R} : y'_i = a_t x'_i + b_t. \quad (5.27)$$

Durch Schätzen einer Ausgleichsgeraden kann das Tracking stabilisiert werden. Unter der zusätzlichen Annahme, dass die Tiefe der in die Mittelpunkte abgebildeten Raumpunkte identisch ist, werden die Verhältnisse $\left(\frac{x_i - x_j}{x'_i - x'_j}\right)_{i,j}$ alle mit demselben Faktor skaliert. Für gleiche y_i ist diese Annahme erfüllt, wenn ein einzelnes Hindernis oder die Straße in der gesamten betrachtete Region abgebildet ist. In diesem Fall lässt sich eine ausgleichende Gerade durch

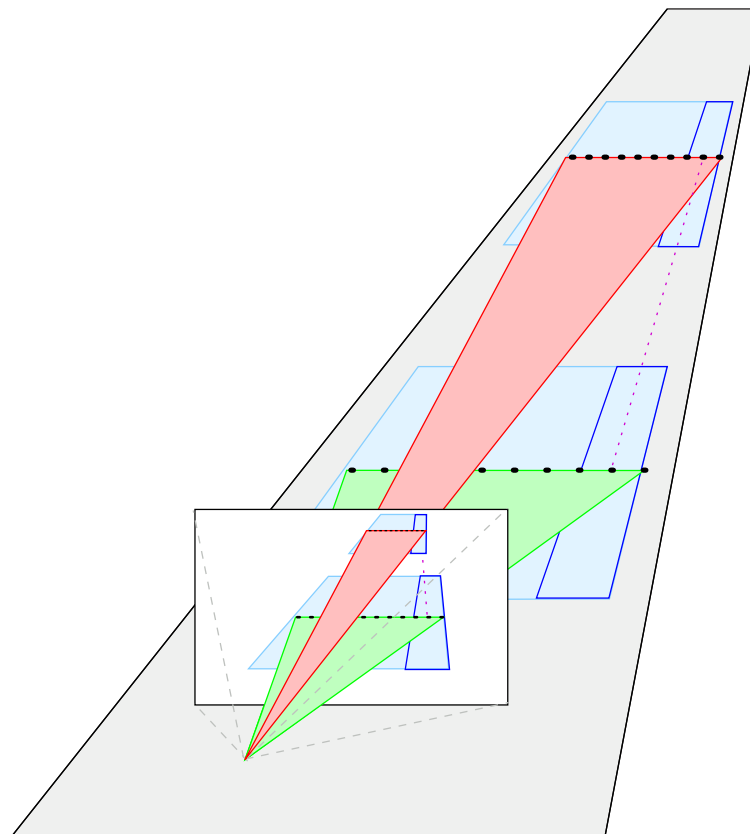


Abbildung 5.4: Abbildung einzelner Unterregionen bei Kameratranslation für die Straßenebene. Die gesamte zu beobachtende Region ist blau hinterlegt. Eine einzelne Unterregion ist dunkelblau umrandet. Die Mittelpunkte sind als Punkte dargestellt. Da die Mittelpunkte auf der Straße alle kollinear sind und in einer Ebene liegen, werden sie im Bild auf eine Gerade abgebildet. Gleiches lässt sich für stehende Hindernisse zeigen.

die x -Koordinaten legen. Da die x -Koordinaten zur Initialisierung äquidistant sind, also

$$x_1 < x_2 < \dots < x_n \text{ mit } x_{i+1} = x_i + \delta,$$

folgt daraus

$$\exists a_x, b_x \in \mathbb{R} : i = a_x x'_i + b_x. \quad (5.28)$$

5.4 Multisensorfusion

Als Quelle für die Eingabedaten zur Hypothesenverifizierung dient ein Sensor, der Entfernung und Sichtwinkel zu potentiellen Hindernissen generiert. Die Verifizierung kann mit falschen Hindernishypothesen (*false positive* Hypothesen) umgehen. Solche sollen im Verifikationsschritt ausgeschlossen werden. Hindernisse von denen keine Hypothesen generiert werden (*false negative Hypothesen*) können auch im Verifikationsschritt nicht bestätigt werden. Eine Sensorfusion sollte vor allem die *false negative* Rate verringern. Eine Sensorfusion kann auch dazu genutzt werden, Hindernisse genauer zu lokalisieren. Dadurch ist eine bessere Verifikation möglich und die *false negative* Rate der Verifikation kann gesenkt werden. Für Algorithmen basierend auf visuellen Sensoren bietet sich eine Fusion mit einem Radarsystem an. Die einzelnen Vor- und Nachteile (siehe Tabelle 2.2) ergänzen sich gegenseitig. Es gibt Ansätze, solch eine Fusion durchzuführen, beispielsweise in der Spurerkennung [31]. Der Ansatz beschreibt eine Zielfunktion aus dem Raum aller möglichen Messungen in den Raum der relevanten Informationen. Eine Möglichkeit ist, die Radarinformationen in Gleichung 4.13 mit einfließen zu lassen. Alternativ kann das Radar auch als alleinige Eingabe für die Hindernisverifizierung genutzt werden. Damit dient die Bildverarbeitung lediglich als Test, um die Daten des Radars zu verifizieren. Diese Möglichkeit wird hier nicht näher untersucht und wird Gegenstand folgender Arbeiten sein. Im Kapitel 7.2.1 wird jedoch die in dieser Ausarbeitung vorgestellte Hypothesengenerierung basierend auf Bildverarbeitung mit Daten des Radars verglichen.

Kapitel 6

Alternativer Tracking Ansatz

Inspiziert von den in dieser Arbeit gewonnenen Erkenntnissen bezüglich Tracking von Bildregionen wird in diesem Kapitel ein neuer Ansatz vorgestellt, Hindernishypothesen robust zu generieren. Die Hypothesengenerierung, wie sie in Kapitel 4 vorgestellt wurde, ist in den folgenden zwei Punkten problematisch:

- Wird in der zu trackenden Region Straße abgebildet, so liefert das verwendete Least Squares Verfahren ein Ergebnis, welches im Minimum der Fehlerfunktion (Gleichung 3.26) liegt. Das zugrunde liegende Modell in dem Verfahren geht von einem stehenden Hindernis aus. Es kann passieren, dass eine Hypothese für Hindernisse in einer Entfernung geliefert wird, die am besten zum Straßenfluss passt, obwohl kein Hindernis in der Region abgebildet ist. Der Anteil an falschen Hindernishypothesen wird *false positive rate* genannt.
- Weil die Modellannahmen für Straße nicht stimmen, ist zu erwarten, dass die Residuen groß sind, falls die Straßenebene in der Zielregion abgebildet ist. Die Residuen sinken aber mit steigender Anzahl an freien Parametern. Deshalb ist der Residuenunterschied zwischen Hindernis und Straßenebene kleiner, wenn neben der geometrischen Transformation auch eine radiometrischen Transformation geschätzt wird. Verzichtet man auf das Schätzen der radiometrischen Parameter, so können stehende Hindernisse nicht mehr über Helligkeitsunterschiede getrackt und eventuell nicht mehr erkannt werden. Der Anteil an nicht erkannten Hindernissen im Vergleich zur Anzahl an Hindernishypothesen wird *false negative rate* genannt.

Ziel eines weiterführenden Ansatzes muss es sein, die *false negative rate* und *false positive rate* zu verringern. Im nächsten Abschnitt werden zwei Konzepte vorgestellt, mit denen eine bessere Modellapproximation erfolgen soll. Abschnitt 6.2 beschreibt die mathematische Herleitung und die Implementierung. In Abschnitt 6.3 werden die Ergebnisse vorgestellt und diskutiert.

6.1 Konzept

Das Modell in Kapitel 4 geht von einem stehenden Hindernis aus und schätzt für die gesamte Zielregion einen Skalierungsfaktor und eine Translation. Ist tatsächlich ein Hindernis in der Region abgebildet, ist in der Fehlerfunktion aus Gleichung 3.22

$$O(\mu) = \sum_{\mathbf{x} \in R} I(f(\mathbf{x}, \mu), t) - I(\mathbf{x}, t_0))^2$$

ein ausgeprägtes Minimum erkennbar (vgl. Abbildung 6.2 oben). Für das Beispiel aus Abbildung 6.1 entspricht die geschätzte Skalierung von 1.03565 bei einer gefahrenen Distanz von 1.275m einer Entfernung des Hindernisses von 37.04m. Ein Vergleich mit der Entfernung aus dem Stereokamerasystem (ca. 40.5m) zeigt, dass die Entfernung relativ genau bestimmt worden ist.

Oftmals ist ein Hindernis aber nur in einem Teilbereich der Zielregion abgebildet oder die gesamte Region beinhaltet kein Hindernis, sondern abgebildete Punkte der Straßenebene. In Abbildung 6.2 unten ist beispielhaft die Fehlerfunktion für abgebildete Straßenebene dargestellt. Deutlich ist zu erkennen, dass sowohl die Skalierung als auch die Translation schlechter lokalisierbar sind. Für die Generierung von Hindernishypothesen ist die Skalierung entscheidend. Die geschätzte Skalierung von 1.08459 entspricht bei 1.275m gefahrener Distanz einem Hindernis in 16.35m. Allerdings ist in der betrachteten Bildregion gar kein Hindernis, sondern Straße abgebildet. Es wird eine *false positive* Hindernishypothese generiert.

Um solche *false positive* Hypothesen zu vermeiden, sollen Pixel der Zielregion gewichtet werden; anstatt für die gesamte Region im Bild einen Skalierungsfaktor zu berechnen, soll nur der Bereich über dem Fußpunkt eines Hindernisses zur Berechnung genutzt werden. Das ist sinnvoll, da Hindernisse auf der Grundebene aufliegen und deshalb unterhalb des Fußpunktes im Bild die Grundebene abgebildet ist. Der Fußpunkt kann direkt aus der Entfernung eines Hindernisses errechnet werden. Für die gewichtete Entfernungsschätzung muss der Fußpunkt allerdings bereits bekannt sein. Die beiden Beziehungen lassen sich nicht topologisch sortieren, bilden also keine Halbordnung. In der Umgangssprache ist dieses Problem als *Henne-Ei Problem* bekannt.

In dieser Arbeit wird zuerst eine Skalierungsschätzung mit fester Gewichtsverteilung durchgeführt. Die Schätzung der Skalierung wird für verschiedene y -Koordinaten im Bild verschieden gewichtet. Die Gewichte in der Zielregion nehmen von oben nach unten mit steigender y -Koordinate ab. Dies ist sinnvoll, da auch die Entfernungen der Bildpunkte unter der Annahme, dass die Grundfläche abgebildet ist, von oben nach unten im Bild abnehmen. Dadurch soll verhindert werden, dass Beobachtungen, die unter dem Fußpunkt liegen und eine geringe Entfernung zur Kamera haben, hohe Gewichte bekommen. Die Werte werden in jeder Iteration des Trackers so angepasst, dass Beobachtungen auf dem Objekt mit geschätzter Entfernung maximale Gewichte erhalten und Beobachtungen unterhalb des Fußpunktes abnehmende Gewichte.



Abbildung 6.1: Frame 141 und 144 aus der Malmshiem Sequenz mit Hindernissen im Focus of Expansion. Die getrackten Regionen sind im linken Bild grün markiert. Im rechten Bild sind die Hindernishypothesen oberhalb des Fußpunktes rot markiert. Die gelbe Linie ist der Horizont.

	C-Klasse	Straße
Entfernung (Stereo)	40.5m	-
Entfernung (aus Skalierung)	37.04m	16.35m
Minimaler SSD-Wert	32.46	85.66
Isolinie (in Abbildung 6.2, 150% vom Minimum)	48.69	128.8

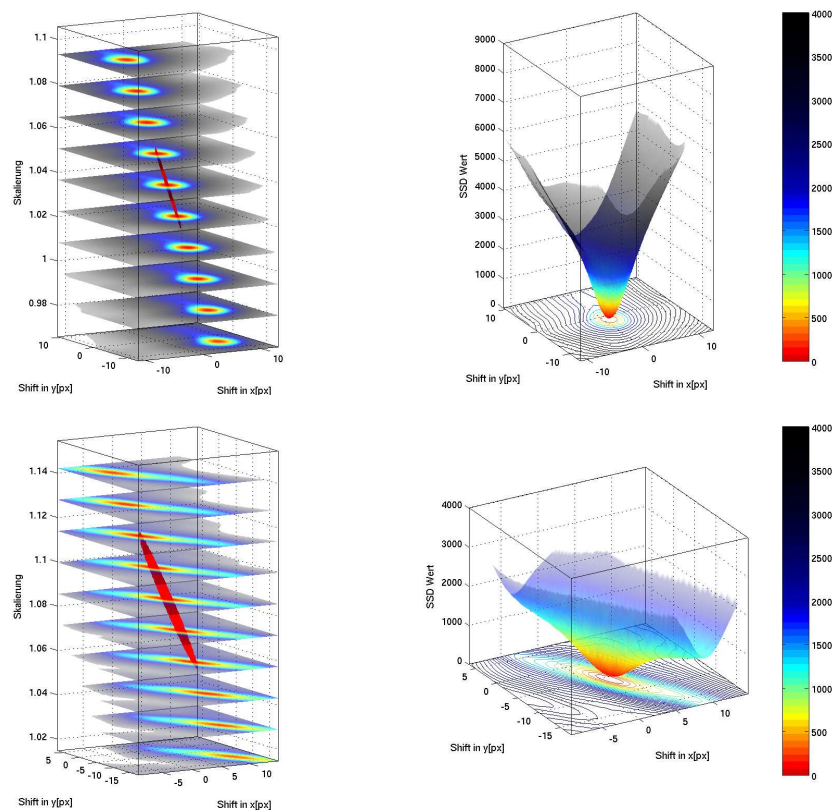


Abbildung 6.2: Fehlergebirge für die getrackten Bildregionen aus Abbildung 6.1. Die Farbskala kodiert den durchschnittlichen SSD Fehler. Links alle drei Parameter; rechts die Translation im gefundenen Minimum der Fehlerfunktion; oben die C-Klasse; unten die Straße. Die gefahrene Distanz und durchschnittlichen SSD Werte sind in der Tabelle über dem Plot zu finden. Es ist zu erkennen, dass das zugrunde liegende Modell besser auf die Region mit der C-Klasse passt.

Im nächsten Abschnitt wird beschrieben, wie sich der Tracking Ansatz aus Kapitel 3.5.2 so modifizieren lässt, dass verschiedene y -Koordinaten individuell gewichtet werden können.

Mit der Skalierungsschätzung liegt eine Schätzung für die Entfernung vor. Über die Entfernung lässt sich bei bekannter Kameraorientierung der Fußpunkt eines Hindernisses berechnen und die Zielregion horizontal aufteilen. Die Region oberhalb des Fußpunktes transformiert sich über die Annahme eines stehenden Objektes. Die Region unterhalb des Fußpunktes wird mit der Straßenannahme transformiert. Durch die iterative Anpassung der Gewichte an die tatsächlichen Beobachtungen und die Aufteilung der Zielregion wird eine bessere Konvergenz an die korrekte Lösung erwartet, als bei dem in Kapitel 4 vorgestellten Ansatz.

Neben der Aufteilung der Bildregion und dem gleichzeitigen Tracken sowohl der Hindernishypothese als auch der Straßenhypothese kann es vorkommen, dass nicht planare Straßenebenen zu falschen Beobachtungen führen. Insbesondere ist der Winkel zwischen Versuchsträger und Straße in der Praxis nicht konstant und damit der angenommene Straßenfluss falsch. Mit der Annahme eines konstanten Wankwinkels und vernachlässigbaren Gierwinkels (Versuchsträger *fährt geradeaus*) soll die Normale auf die in die Zielregion abgebildete Ebene berechnet werden. Im betrachteten Bauklotzmodell sind die Frontflächen aller Hindernisse und die Grundfläche senkrecht zur Y - Z -Ebene, daher ist die Normale eine Linearkombination aus den Einheitsvektoren der Z und Y -Achse. Anhand der Normale kann zwischen Hindernis (*Normale zeigt in Richtung Z-Achse*) und Straße (*Normale zeigt in Richtung Y-Achse*) unterschieden werden. In [45] und [51] werden Ansätze beschrieben, wie solch eine Normale für beliebige Ebenen errechnet werden kann. Ziel ist es, diese Ansätze auf den hier beschriebenen Tracking Ansatz zu übertragen. Dies ist ein erster Schritt, das Bauklotzmodell durch andere Modelle zu ersetzen.

6.2 Mathematische Betrachtung und Implementierung

In diesem Abschnitt werden die vom Tracker benötigten Matrizen für die dargestellten Konzepte hergeleitet. Die Integration der Matrizen in den Trackingvorgang wird beschrieben und die Konzepte mathematisch untersucht. Das zugrunde liegende Modell für die Kameraszene ist das in Kapitel 3.3 erläuterte Bauklotzmodell. Die inneren und äußeren Parameter der Kameraorientierung sind bekannt und der Versuchsträger bewegt sich in Z -Richtung des Weltkoordinatensystems. Um die Mathematik so einfach wie möglich zu halten, wird von einem Kamerakoordinatensystem ausgegangen, das parallel zum Weltkoordinatensystem liegt. Dies kann durch Herausrechnen der Rotationsanteile aus dem Bild (vergleiche Abbildung 4.2) erreicht werden.

6.2.1 Ausnutzung des Fußpunktes

Bei bekannter Entfernung eines Hindernisses d kann über Strahlensatz die Bildzeile des Fußpunktes berechnet werden:

$$\frac{y}{f} = \frac{d}{-Y_0} \Rightarrow y = \frac{fd}{-Y_0}. \quad (6.1)$$

Mit Kenntnis des Fußpunktes kann die zu trackende Region horizontal unterteilt werden. Oberhalb des Fußpunktes ist das Hindernis abgebildet, unterhalb die Straßenebene. Abhängig von der y -Koordinate können dann den Beobachtungen im Bild zeilenweise verschiedene Gewichte zugeordnet werden. Im Tracking Ansatz, wie er in Abschnitt 3.5.2 beschrieben ist, ist es nicht möglich iterativ den einzelnen Pixeln Gewichte zu verteilen ohne die komplette Matrix M in

$$-(M^t M)^{-1} M^t \mathbf{e}(t + \tau) = \delta \boldsymbol{\mu} \quad (\text{vergleiche Gleichung 3.29})$$

neu aufzustellen. Im Folgenden wird eine Möglichkeit ausgearbeitet, die einzelnen Zeilen in jedem Iterationsschritt verschieden zu gewichten. Prinzipiell geschieht eine blockweise Gewichtung der Pixel in linearer Zeit abhängig von der Anzahl der Zeilen in der Region. Die zu minimierende Fehlerfunktion 3.28 lässt sich mit einer Gewichtsmatrix $G = \text{diag}(g_1, g_2, \dots, g_n)$ umschreiben zu

$$O(\delta \boldsymbol{\mu}) = \|G(M\delta \boldsymbol{\mu} + \mathbf{e}(t + \tau))\|^2. \quad (6.2)$$

Daraus ergibt sich für das Minimum (vgl. Gleichung 3.29):

$$\delta \boldsymbol{\mu} = -(M^t G M)^{-1} M^t G \mathbf{e}(t + \tau). \quad (6.3)$$

Der Fehlervektor \mathbf{e} wird mit der Gewichtsmatrix multipliziert und die Matrix $M^t G M$ muss abhängig von der Gewichtsmatrix berechnet werden. Ändert sich die Gewichtsmatrix, so bedeutet dies, dass jedes mal die Matrix neu berechnet werden muss. Für den Fall, dass die Gewichte zeilenweise gleich sind, lässt sich die Berechnung blockweise durchführen. Es gilt:

$$M^t G M = \sum_i \mathbf{m}_i g_i \mathbf{m}_i^t = \sum_i g_i \mathbf{m}_i \mathbf{m}_i^t = \sum_y g_y \sum_{y_i} \mathbf{m}_{y_i} \mathbf{m}_{y_i}^t \quad (6.4)$$

mit

$$M = \begin{pmatrix} \mathbf{m}_1^t \\ \mathbf{m}_2^t \\ \vdots \\ \mathbf{m}_n^t \end{pmatrix}$$

und Gewichten g_y für die einzelnen Zeilen sowie dem Index y_i , welcher die Zeilen $\mathbf{m}_{y_i}^t$ der Matrix M eindeutig den Zeilen \mathbf{m}_i^t zuordnet. Die einzelnen Summen $\sum_{y_i} \mathbf{m}_{y_i} \mathbf{m}_{y_i}^t$ können offline vorberechnet werden, so dass lediglich eine Summierung über die verschiedenen Gewichte für die Zeilen vorgenommen werden muss. Das Tracking geschieht wie in Abschnitt 3.5.2 hergeleitet in zwei Schritten. Im ersten Schritt wird für die gesamte Region eine Skalierung und Translation mit Gewichtung der einzelnen Zeilen gerechnet. Im zweiten Schritt wird die Region in Fußpunkthöhe aufgeteilt und die erwartete Transformation für ein Hindernis und die Straße gerechnet mit anschließender Schätzung einer Translation, um nicht modellierte Eigenbewegung zu kompensieren. In beiden Schätzungen werden neben den geometrischen Parametern auch die radiometrischen Parameter bestimmt.

Zur Bestimmung der Gewichte für die einzelnen Zeilen im Bild wird (bei zunehmender y -Koordinate von *oben nach unten* im Bild) eine Gewichtsfunktion $g(y)$ benutzt mit

$$g : [y_{min}, y_{max}] \rightarrow [0, 1] : g(y_{min}) = 1, g(y_{max}) = 0 \text{ und } g'(y) \leq 0. \quad (6.5)$$

Eine entsprechende Gewichtsfunktion ist

$$g(y) = \left(\frac{y_{max} - y}{y_{max} - y_{min}} \right)^n.$$

Die Region wird entsprechend der Gewichtsfunktion aufgeteilt. Nach der Schätzung des Fußpunktes werden Pixel oberhalb des Fußpunktes maximal gewichtet. Pixel unterhalb erhalten mit größerem Abstand zum Fußpunkt aufgrund der Monotonie der Gewichtsfunktion abnehmende Gewichte. Dies soll bewirken, dass Pixel auf der Straße, welche näher zur Kamera sind, den Skalierungsfaktor weniger beeinflussen als weiter entfernte Pixel und solche auf dem Hindernis.

6.2.2 Schätzung der Normale

Liegen alle Bildpunkte einer Region auf einer Ebene

$$Z = A + BX + CY \quad (6.6)$$

im Kamerakoordinatensystem, so lässt sich bei bekannter innerer Orientierung der Kamera für jeden abgebildeten Raumpunkt die Z -Koordinate aus den Bildkoordinaten berechnen ([45]) als

$$\frac{1}{Z} = \alpha + \beta x + \gamma y = \frac{1}{A} - \frac{B}{fA}x - \frac{C}{fA}y. \quad (6.7)$$

Geht man von einer reinen Vorwärtsbewegung (T_Z) der Kamera aus und benutzt die zusätzliche Annahme, dass die betrachtete Ebene unabhängig von X ist, also senkrecht auf

der Y - Z -Ebene steht (speziell $Z = A + CY$), so lässt sich das Geschwindigkeitsfeld im Bild (Gleichung 3.13) umschreiben zu

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} \alpha T_Z u + \gamma T_Z u v \\ \alpha T_Z v + \gamma T_Z y^2 \end{pmatrix} \quad (6.8)$$

Die zu schätzenden Parameter sind α und γ . Aus der Ebenengleichung lassen sich die Z -Koordinate des Schnittpunktes der Ebene mit der Z -Achse (A) und Normale (\mathbf{n}) auf die betrachtete Ebene ableiten:

$$A = \frac{1}{\alpha} \text{ und } \mathbf{n} = \left\| \begin{pmatrix} 0 \\ C \\ -1 \end{pmatrix} \right\|. \quad (6.9)$$

Bei einer Kamera parallel zum Weltkoordinatensystem ergeben sich für ein stehendes Hindernis in Entfernung d die theoretischen Werte $A = d$ und $\mathbf{n} = (0, 0, -1)^t$. Die Straßenebene führt in der Theorie zu den Werten $A = \infty$ und $\mathbf{n} = (0, 1, 0)^t$. Letzteres entspricht einem degenerierten Fall, $C = A = \infty$. Für solche Fälle ist Gleichung 6.8 schlecht konditioniert. Die Kondition eines solchen Ansatzes wurde in [51] für den Fall einer allgemeinen Bewegung untersucht. Die Autoren schlagen vor, für bessere Kondition bei Betrachtung der Straßenebene die Kamera so zu drehen, dass sie virtuell in Y -Richtung des Weltkoordinatensystems schaut. Die Autoren modellieren eine Translation in Z - und Y -Richtung sowie Rotation um die X -Achse des Weltkoordinatensystems. Der Winkel einer Ebene zur Z -Achse ist somit in [51] fest vorgegeben.

In dieser Arbeit wird auf eine virtuelle Drehung der Kamera verzichtet, da in der Praxis das Kamerakoordinatensystem nicht parallel zum Weltkoordinatensystem liegt. Damit treten keine Nebeneffekte aufgrund der notwendigen Interpolation der Grauwerte bei virtueller Kameradrehung auf, aber die äußeren Parameter der Kamera müssen in Gleichung 6.8 mit berücksichtigt werden. Der Einbauwinkel der Kamera um die X -Achse des Versuchsträgers, Ω_X , ist vorgegeben. Mit einer Translation T_Z des Versuchsträgers führt dies zu:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} \alpha \cos(\Omega_X) T_Z u + \gamma \sin(\Omega_X) T_Z u v \\ -\alpha f \sin(\Omega_X) T_Z + (\alpha \cos(\Omega_X) - \gamma f \sin(\Omega_X)) T_Z v + \gamma \sin(\Omega_X) T_Z y^2 \end{pmatrix}. \quad (6.10)$$

Um die Matrizen im Region Tracking offline berechnen zu können, muss die Transformationsmatrix M in Abschnitt 3.5.2 aufgeteilt werden. Dies ist bei affinen Abbildungen möglich. In Gleichung 6.10 treten aber quadratische Terme auf und die Matrix M muss für jede Iteration erneut berechnet werden. Für die Untersuchung dieses Ansatzes ist ein solcher Ansatz durchaus vertretbar, auch wenn dies den Trackingalgorithmus verlangsamt. Der nächste Abschnitt befasst sich mit den Ergebnissen der beiden vorgestellten Ansätze zum Tracken von Bildregionen.



Abbildung 6.3: Verbessertes Trackergebnis durch Gewichtung der Beobachtungen. Das Hindernis wird in 41.8m lokalisiert (links) und liegt näher an der Messung des Stereokamerasystems von 40.5m; die Skalierung der abgebildeten Straße entspricht einer Entfernung von 22.31m und damit dem oberen Bereich der abgebildeten Region (rechts).

6.3 Diskussion

Die in diesem Kapitel vorgestellten Ansätze zur Stabilisierung des Trackings werden in diesem Abschnitt untersucht und die Ergebnisse diskutiert. Zur Untersuchung werden sowohl künstlich generierte Sequenzen als auch reale Sequenzen verwendet. Die verwendeten realen Sequenzen sind mit einem Stereokamerasystem aufgenommen worden, dessen linkes Bild Eingabe für den monokularen Algorithmus ist. Die Entfernungen sind für Vergleichszwecke über Triangulation aus den Disparitäten der Stereokamera direkt messbar.

Für die in Abbildung 6.1 dargestellte Situation liefert die Gewichtung der Beobachtungen und Ausnutzung des Fußpunktes eine Verbesserung des Ergebnisses. Abbildung 6.3 zeigt das verbesserte Trackergebnis. Die Entfernung des stehenden Hindernisses liegt näher an der Messung des Stereokamerasystems, was daran liegt, dass die Beobachtungen auf der Straße vor dem Hindernis eine geringere Gewichtung erhalten haben.

Die abgebildete Region auf der Straße generiert ein weiter entferntes Hindernis als ohne Gewichtung. Vergleicht man in Abbildung 6.4 das Fehlergebirge für die Straßenregion mit Abbildung 6.2, so fällt auf, dass das Minimum der Skalierung sich verschoben hat. Allerdings sind in der Fehlerfunktion nun lokale Minima in der Nähe des globalen Minimums erkennbar, in die der Algorithmus bei falscher Initialisierung läuft. Dies ist ein noch ungelöstes Problem des vorgeschlagenen Ansatzes.

Das Tracking mit Gewichtung der Beobachtungen liefert bei richtiger Fußpunktschätzung bessere Ergebnisse als ein Tracking ohne Gewichtung. Allerdings ist die Schätzung des Fußpunktes in realen Szenen oftmals ungenau. Das Hindernis in Abbildung 6.5 befindet sich in 95m Entfernung. Die Entfernung wird richtig geschätzt, allerdings stimmt der berechnete Fußpunkt nicht mit den Beobachtungen überein. In folgenden Frames beeinflussen die Bäume im Hintergrund den Skalierungsfaktor stärker als der LKW, obwohl sie nur einen sehr kleinen

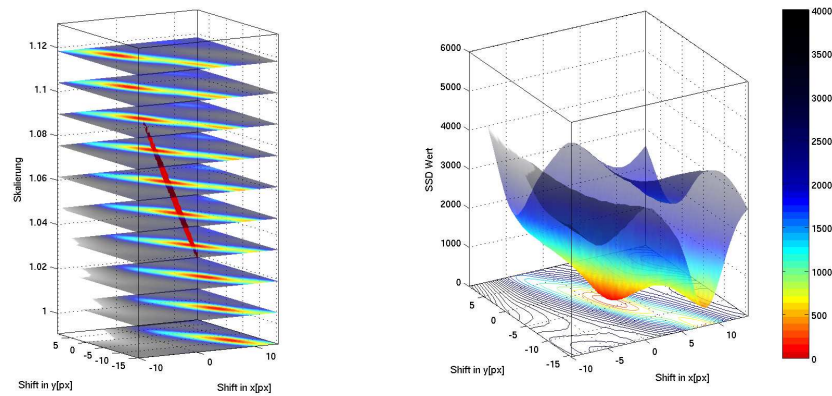


Abbildung 6.4: Fehlergebirge für die Region in Abbildung 6.3 rechts. Links alle drei Parameter; rechts die Translation im gefundenen Minimum der Fehlerfunktion; der minimale SSD Wert ist 40.99, die eingezeichnete Isofläche entspricht einem Wert von 61.49 (150% des Minimums).

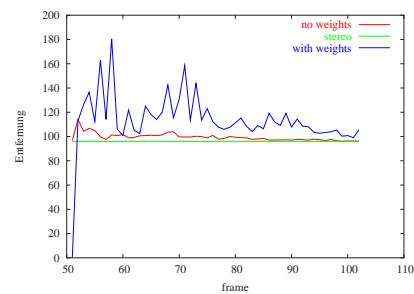
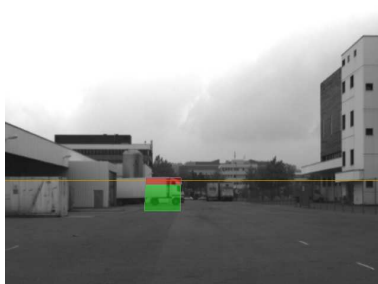


Abbildung 6.5: Frame 43 aus der LKW Beladestation. Aufgrund unebener Straße schaut die Kamera nach oben. Die gelb eingezeichnete Linie (links) ist der angenommene Horizont. Die Schätzung der Entfernung (rechts) wird wegen der falschen Fußpunktschätzung für den Ansatz mit Gewichtung der Beobachtungen ungenauer.

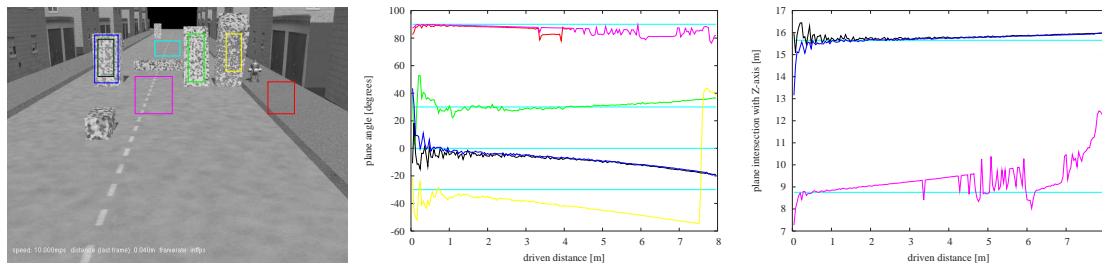


Abbildung 6.6: Ebenenschätzung in synthetischen Sequenzen. Für die im Bild eingezeichneten Regionen wurde eine Ebenengleichung geschätzt; zyanblau die erwarteten Werte. Der Nickwinkel ist für alle Regionen bis auf die zyanblaue Region (keine Ergebnisse möglich) bestimmt worden, die Entfernung ist exemplarisch für das stehende Hindernis und die Straßenebene eingezeichnet. Kamerahöhe 2.52m, fokale Länge 720px, Kameranickwinkel 16.3°

Teil der Region ausmachen und die im Bild falsche Annahme des Fußpunktes führt zu hohen Residuen für das Tracken der Straßenebene. Dies führt dazu, dass das Hindernis nicht richtig getrackt wird und Schätzung der Skalierung sowie Translation der Zielregion fehlerhaft sind. Der Effekt ist bei niedrigeren Hindernissen verstärkt zu beobachten. Die einfache Schätzung einer Ähnlichkeit ohne Gewichtung der Beobachtungen liefert in solchen Fällen bessere Ergebnisse, da die Hindernisse mit voller Gewichtung in die Schätzung eingehen.

Die Fußpunktsuche kann mit besserer Modellierung der Straßenebene, beispielsweise bei Kenntnis der Normalen der Straßenebene im Kamerakoordinatensystem, effizienter angegangen werden. Eine andere Möglichkeit ist eine Flussfeldsegmentierung ähnlich wie in [14], um die beiden Flussfelder eines stehenden Hindernisses und der Straßenebene zu trennen.

In der vorliegenden Arbeit ist die Schätzung einer Normalen untersucht worden. Anhand einer künstlich generierten Sequenz, in der alle Daten bekannt sind, wurde für verschiedene Regionen die Normale sowie der Abstand der Ebene im Raum geschätzt. Die Ergebnisse sind in Abbildung 6.6 zu sehen. Für den Nahbereich vor dem Fahrzeug ist die Schätzung der Normale sehr gut. Wegen der Linearisierung des Bewegungsfeldes in Gleichung 6.10 funktioniert die Schätzung allerdings nur, wenn die Bewegung des Versuchsträgers nicht allzu groß ist. Die Umformulierung des Problems in einen Ansatz, der unabhängig von der gefahrenen Distanz korrekt ist (vgl. Kapitel 4), lässt sich nicht anwenden, da die Z -Koordinate der abgebildeten Punkte in der Zielregion nicht identisch ist. Die Parameter in Abbildung 6.6 sind bezogen auf das erste Bild geschätzt worden und weichen daher bei größerer gefahrener Distanz immer mehr von der tatsächlichen Lösung ab. Betrachtet man die Gütegebirge (Abbildung 6.7) für die Schätzung der Parameter A und C für ausgewählte Regionen aus Bild 6.6 nach 0.4m zurückgelegter Distanz, stellt man fest, dass es mehrere Minima gibt, weil die Orientierung

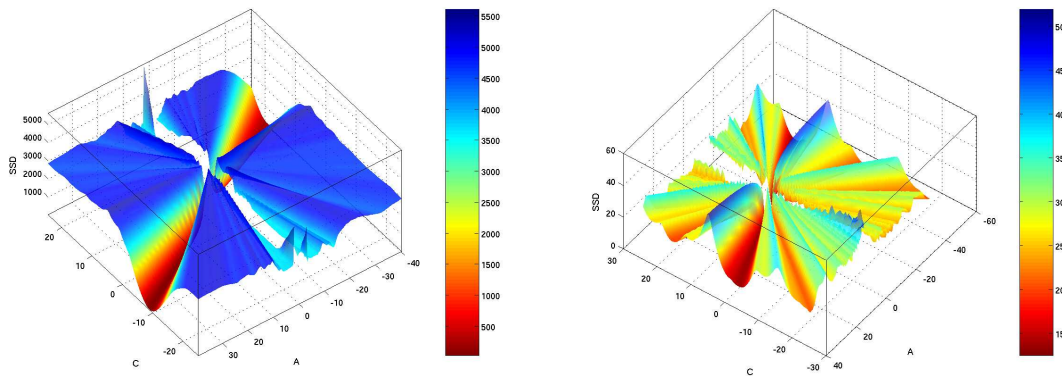


Abbildung 6.7: Gütegebirge für die Ebenenschätzung mit den Parametern aus Gleichung 6.6 mit $B = 0$ für ein Patch der Größe 37×37 Pixel im Bild aus Abbildung 6.6. Links Hindernis in 15m Entfernung; rechts Straße in ca. 25m Entfernung. Der Bereich um $A = 0$ wird nicht angezeigt (Singularität, $A = 0 \Leftrightarrow$ Ebene geht durch das Projektionszentrum).

der Normalen zur Ebene nicht eindeutig festgelegt ist. Außerdem ist an den langgezogenen *Tälern* im Gütegebirge zu erkennen, dass die beiden Parameter A und C korreliert sind. Dies ist dadurch zu erklären, dass für eine Region im Bild eine Schätzung der Z -Koordinaten der abgebildeten Raumpunkte erfolgt. Der Algorithmus findet eine ausgleichende Ebene, welche zu den Beobachtungen (Grauwertdifferenzen) am besten passt. Für verschiedene Schnittpunkte $(0, 0, A)^t$ mit der Z -Achse des Weltkoordinatensystems lässt sich für die gleiche Punktwolke eine Ebene und damit eine Normale abhängig vom Schnittpunkt finden, die die Raumpunkte am besten approximiert.

Das Gütegebirge ist im Minimum umso flacher, je weniger Struktur in der Region enthalten ist und je größer die Z -Koordinaten der abgebildeten Raumpunkte einer Zielregion sind, da die relativen Bewegungen von der Z -Koordinate der Raumpunkte abhängen. Deshalb ist das Gebirge für die Straßenebene in Abbildung 6.7 insgesamt flacher als das Gebirge für die Region mit dem abgebildeten Hindernis. Empirische Test haben gezeigt, dass für die künstlich generierte Szene aus Abbildung 6.6 ab einer Entfernung der abgebildeten Punkte von 25m keine robuste Schätzung der Ebene mehr möglich ist.

Ein Ergebnis auf realen Daten ist in Abbildung 6.8 ersichtlich. Die Ergebnisse sind schlechter als die Ergebnisse für synthetische Bildfolgen. Der Einbauwinkel der Kamera ist mit -0.027 rad kleiner als in den künstlichen Sequenzen und das Kamerakoordinatensystem ist nahezu parallel zum Straßenkoordinatensystem gerichtet. Daher ist das Problem schlechter konditioniert (vgl. Ausarbeitung in 6.2.2). Zudem erschwert Rauschen die Schätzung der Parameter. Für weiter entfernte Hindernisse, wo die Fußpunktschätzung Probleme bereitet, ist

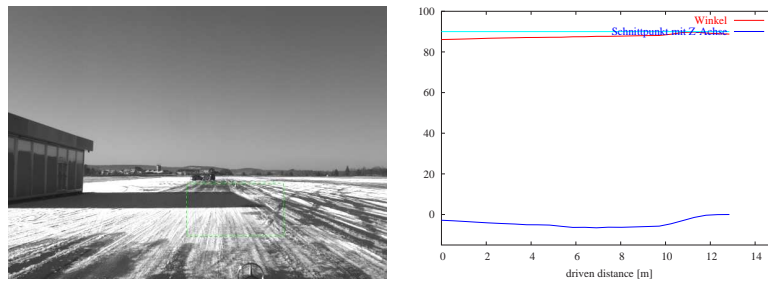


Abbildung 6.8: Ebenenschätzung in realen Sequenzen. Für große Regionen mit viel Struktur ist eine Schätzung der Normalen möglich (in zyanblau die Annahme einer planaren Straße). Die Schätzung des Schnittpunktes mit der Z -Achse ist ungenau. Bei Annahme einer planaren Straße liegt der Schnittpunkt bei -40m , unter Berücksichtigung des geschätzten Nickwinkels der Straße bei $-11,5\text{m}$. Kamerahöhe 1.1m , fokale Länge 842px , Kameranickwinkel -0.027rad

dieses Verfahren der Nickwinkelschätzung nicht direkt anwendbar, um die Straßenebene zu modellieren. Der Fehler im Nickwinkel ist mit bis zu 5 Grad zu groß für praktische Anwendungen. Weitere Untersuchungen sind hier nötig, um robuste Ergebnisse zu erhalten.

Die in diesem Kapitel dargestellten Ansätze haben das Tracking in kontrollierter Umgebung verbessern und stabilisieren können. Allerdings sind die Ansätze nicht robust genug, um in allgemeinen Situationen verlässliche Ergebnisse zu liefern. Für die Nutzung im Straßenverkehr ist die Hypothesengenerierung aus Kapitel 4 daher besser geeignet. Im nächsten Kapitel werden die Ansätze zum Verfolgen von Hindernissen und Schätzen der Entfernung mit anderen Sensoren verglichen. Anschließend wird die Hypothesengenerierung und -verifizierung als Gesamtkonzept näher untersucht und diskutiert.

Kapitel 7

Experimentelle Untersuchungen und Ergebnisse

Das entwickelte System zur Hindernisdetektion wurde zunächst mit synthetischen Daten untersucht. Anschließend wurden Untersuchungen auf Videosequenzen aus zwei unterschiedlichen Versuchsträgern durchgeführt. Zunächst wurden die Komponenten der Hindernisdetektion mit Daten anderer Sensoren und mit alternativen Algorithmen verglichen. Dies erlaubt eine qualitative Beurteilung des vorgeschlagenen Algorithmus. In Abschnitt 7.4 werden die Ergebnisse des entwickelten Systems beispielhaft für drei reale Szenen dargestellt.

7.1 Synthetische Daten

Bei der Untersuchung des Systems mit synthetischen Sequenzen ist die genaue Position der Hindernisse bekannt. Für die in Abbildung 7.1 dargestellte Szene wurde für das stehende Hindernis und die texturierte Strasse die Entfernung aus der Skalierung der ausgewählten Bildbereiche bestimmt.

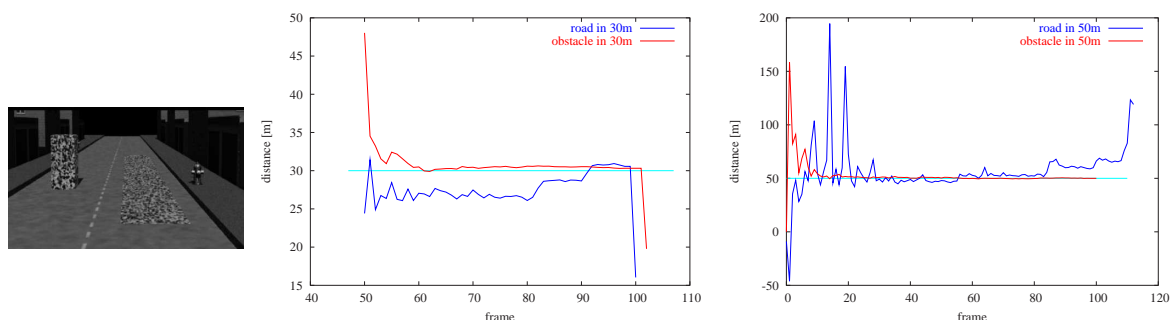


Abbildung 7.1: Entfernungsschätzung für synthetische Daten

Die Entfernung wird bei guter Initialisierung auf dem Objekt bereits nach kurzer gefahrener Distanz korrekt geschätzt. Ist in der Region nur Straße abgebildet, so schwankt die Entfernungsschätzung stark. An der Abweichung vom Mittelwert ließe sich somit bereits eine Aussage darüber treffen, ob ein Hindernis oder die Straßenebene in der Region abgebildet ist. Die Untersuchung dieses Effektes ist Gegenstand weiterer Arbeiten.

In der Regel ist die Strasse nicht so stark strukturiert, wie in Abbildung 7.1. In solchen Fällen schwankt die Entfernungsschätzung noch stärker, weil eine Bestimmung der Transformation im Bild in homogenen Regionen nur sehr schwer möglich ist. Die Verfolgung der Bildregionen nutzt die räumlichen Ableitungen (Gradienten) der beinhalteten Pixel, welche in homogenen Regionen Null sind. Es ist nicht vorherzusehen, zu welchen Ergebnissen das falsche Modell eines stehenden Hindernisses dann führt. Sind Hindernisse in der Region abgebildet, so weisen sie in der Regel deutlich mehr Struktur und somit betragsmäßig größere Gradienten auf, als die Straßenebene. Die Entfernungsbestimmung ist in solch einem Fall deutlich genauer.

Neben der Bewertung einer einzelnen Skalierungsschätzung wurde auch der in dieser Arbeit vorgeschlagene Ansatz einer Hypothesenverifizierung als zweiter Schritt der Hindernisdetektion untersucht. Die Hypothesenverifizierung soll die falsche Hindernishypothese in 30m aus Abbildung 7.1 für die Straßenebene verwerfen und die korrekte Hypothese für das stehende Hindernis bestätigen. In Abbildung 7.2 ist das Ergebnis der Hypothesenverifizierung zu erkennen. Die korrekte Hypothese wurde erfolgreich bestätigt, die falsche Hypothese herausgefiltert.

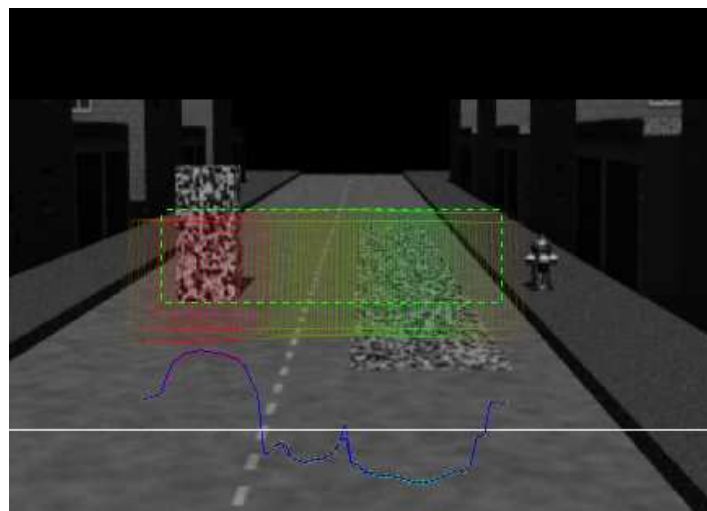


Abbildung 7.2: Hypothesenverifizierung für synthetische Daten. Der Plot im unteren Teil des Bildes zeigt die Differenz der SSD-Werte für die Annahmen *Hindernis* und *Straßenebene*. Die Differenz ist außerdem farblich kodiert im Bild für die einzelnen betrachteten Unterregionen dargestellt.

7.2 Vergleich mit anderen Sensoren

In diesem Abschnitt wird die Entfernung aus der Skalierungsschätzung mit den Entfernungangaben anderer Sensoren verglichen. Dabei wird sowohl ein Vergleich mit einem aktiven Sensor (Radar-Lidar Fusion) sowie mit einem Stereokamerasystem durchgeführt.

7.2.1 Radar-Lidar

Für den Vergleich der Entfernungs- und Breitenschätzung eines Hindernisses mit einer Sensorfusion aus Radar und Lidar wurde eine texturierte Wand aufgestellt. Diese ist zwecks besserer Reflexionseigenschaften mit einer Spezialfolie beklebt und so optimal auf das Radar-Lidar System abgestimmt. Die Sensorfusion bestimmt eine Entfernung und eine Breite des Hindernisses. Diese Daten werden mit den in dieser Arbeit entwickelten Messmethoden verglichen. Abbildung 7.3 zeigt den Versuchsaufbau und die Ergebnisse. Die Entfernungsmessung ist mit

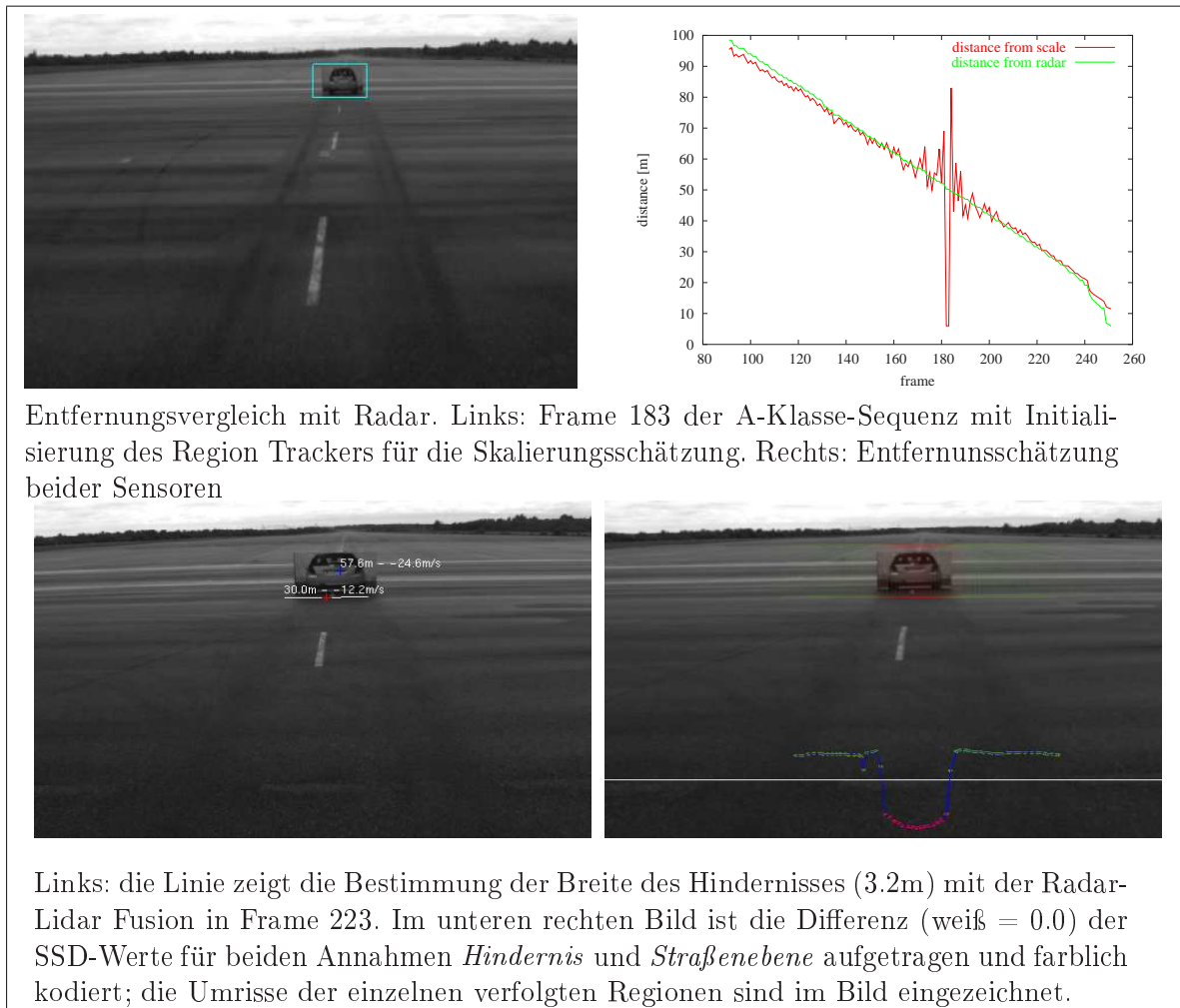


Abbildung 7.3: Vergleich mit Radar.

der Radar-Lidar-Entfernungsmessung vergleichbar, obwohl nur ein einziger visueller Sensor verwendet wird. Die Messung der Skalierung ist um den Initialisierungsbereich (frame 183, ca. 50m) ungenau, da der Skalierungsfaktor nahe bei 1.0 liegt. Daher ist auch die Entfernungsmessung in diesem Bereich stark verrauscht. Die Breite des Hindernisses wird mit dem Hypothesenvergleich für das 30m entfernte Objekt genau bestimmt. Der Schwerpunkt des Objektes aus der Radar-Lidar Fusion stimmt mit dem visuellen Schwerpunkt nicht exakt überein, das visuelle Verfahren bestimmt den Schwerpunkt genauer.

7.2.2 Stereo

Der Vergleich mit Stereo dient dazu, das vorgeschlagene System einem anderen optischen System gegenüberzustellen. Die durchschnittlichen Entfernungsmessungen aus den Disparitäten eines kalibrierten Stereokamerasystems für die ausgewählte Bildregion werden mit der monokularen Entfernungsmessung verglichen. Wie in Abbildung 7.4 zu sehen ist, stimmen die

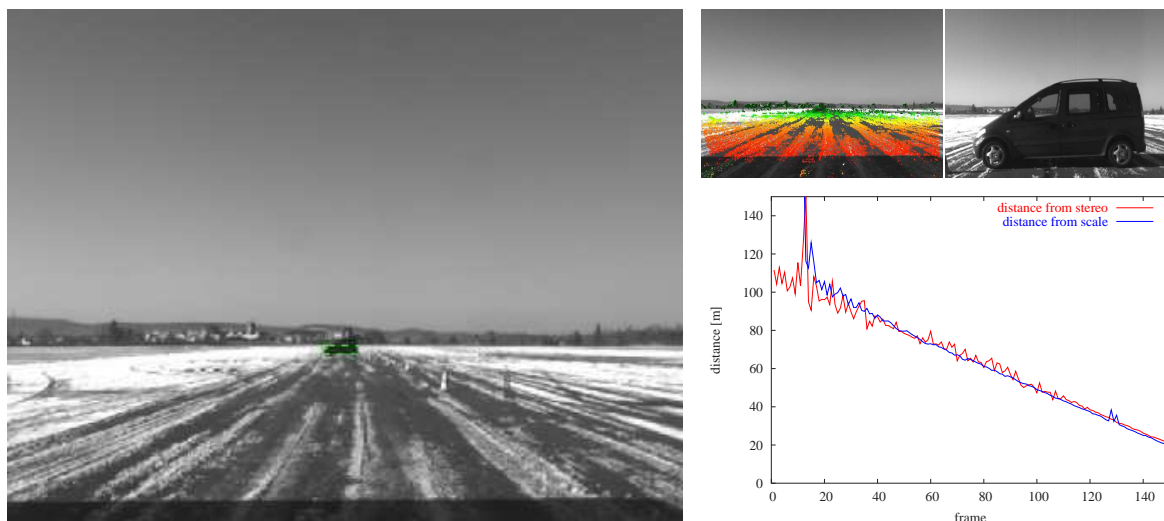


Abbildung 7.4: Vergleich mit Stereo Entfernungsmessung. Das linke Bild zeigt das erste Bild der Sequenz mit ausgewählter Region für das Tracken. Das obere linke Bild zeigt die Disparitäten in diesem Bild; oben rechts das letzte Bild der Sequenz. Der Plot zeigt die ermittelten Entfernungen (die Disparitäten der Region wurden zur Stereoentfernungsbestimmung gemittelt).

Entfernungsmessungen für ein weit entferntes Objekt im *Focus of Expansion* bereits nach kurzer gefahrener Distanz überein. Die monokulare Entfernungsmessung scheint deutlich weniger rauschanfällig zu sein. Dies ist dadurch zu erklären, dass das jeweils aktuelle Frame für die Skalierungsschätzung mit Frame 1 verglichen wird und die relative Ungenauigkeit im Skalierungsfaktor abnimmt, während für die Stereoentfernungsmessung nur das jeweils aktuelle Stereobildpaar benutzt wird.

7.3 Vergleich mit anderen Verfahren

Die Erkennung von Hindernissen vor Fahrzeugen und Schätzung ihrer absoluten Position durch Auswertung einer monokularen Bildsequenz wurde bisher im Computer Vision Bereich vor allem mit optischen Flussverfahren durchgeführt. Im nächsten Abschnitt werden die Ergebnisse des entwickelten Systems mit denen optischer Flussverfahren verglichen. In Abschnitt 7.3.2 wird die vorgeschlagene Nickwinkelschätzung mit dem Ergebnis einer Ego-Motion Schätzung verglichen.

7.3.1 Vergleich mit optischem Fluss

Die Entfernungsschätzung aus der Skalierung wird in diesem Abschnitt mit optischen Flussverfahren verglichen. Optische Flussverfahren rechnen die Verschiebung von Bildpunkten zwischen zwei Bildern und über mehrere Frames akkumulieren sich Fehler auf. Das in dieser Arbeit vorgestellte Verfahren schätzt die Parameter jeweils zum ersten Bild, weshalb die erwarteten absoluten Ungenauigkeiten der Schätzung gleich bleiben und daher die relativen Ungenauigkeiten (bei zunehmendem Skalierungsfaktor) abnehmen. Über längere Zeiträume über mehrere Frames ist das Tracking Verfahren deshalb genauer als optische Flussverfahren. Über Filterung lässt sich die Schätzung aus Flussdaten stabilisieren, allerdings ist ein direkter Vergleich dann nur schwer möglich. Ein Vergleich ist für die Entfernungsschätzung in zwei aufeinanderfolgenden Bildern mit kurzem Zeitabstand sinnvoll. Die hier benutzten Flussverfahren sind

- das von DaimlerChrysler entwickelte Power Flow, welches in Echtzeit ein relativ dichtes Feld von pixelgenauen Flussvektoren rechnet [78],
- der in vielen Verfahren genutzte KLT-Tracker [70][84], welcher markante Punkte sucht und in Echtzeit verfolgt und
- der in [10] beschriebene Variationsansatz, welcher für jedes Pixel einen Flussvektor bestimmt. Letzteres Verfahren ist in der aktuellen Implementierung auf VGA Bildern (640×480 Pixel) nicht echtzeitfähig.

Alle Verfahren rechnen auf der höchsten Auflösungsstufe Flussvektoren für die markierten Regionen in Abbildung 7.5. Anschließend wird eine Skalierung und eine Translation geschätzt, wobei alle Flussvektoren mit gleichem Gewicht in die Ausgleichung eingehen. Der Tracker trackt auf der höchsten Auflösungsstufe die markierten Regionen und liefert einen Skalierungsfaktor und eine Verschiebung, um die Ergebnisse vergleichen zu können. Die Messung der Entfernung mit Stereo beträgt 32m bis 33m. Der Versuchsträger hat 0,423m zwischen beiden Aufnahmen zurückgelegt. Für den Vergleich der verschiedenen Verfahren sind in Tabelle 7.1 die geschätzte Entfernung für die ausgewählte Bildregion sowie der Fixpunkt der Translation und Skalierung eingetragen.

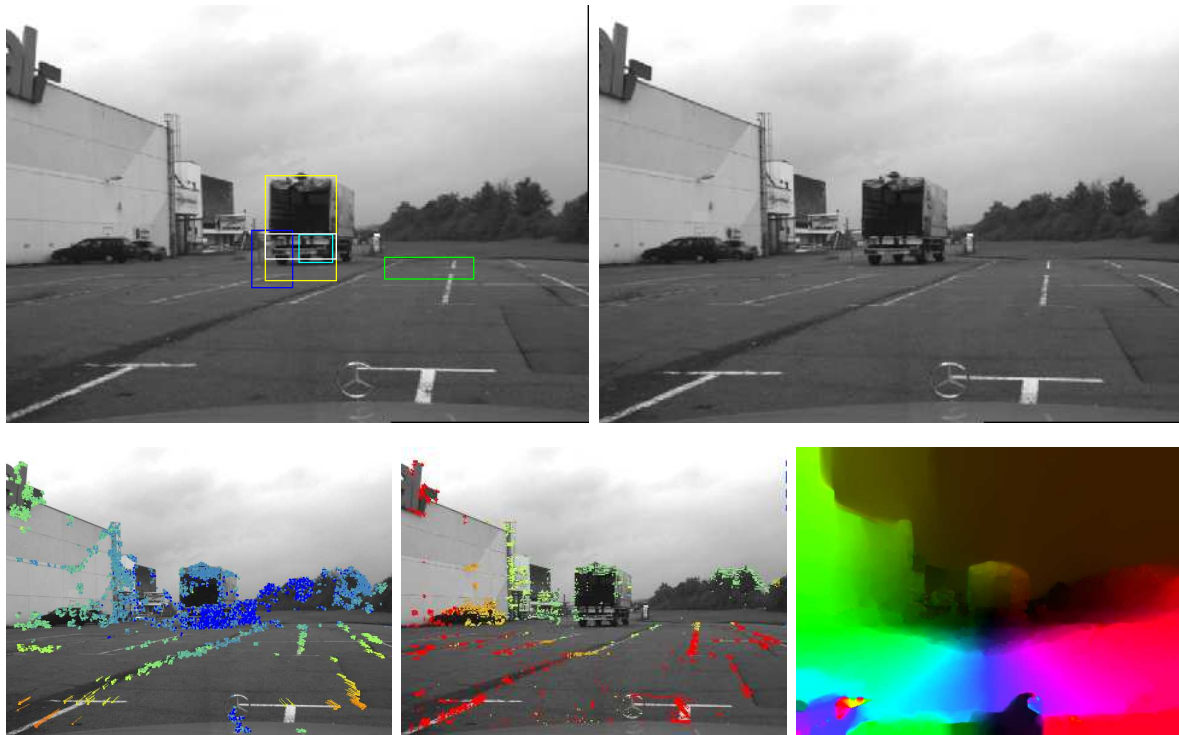


Abbildung 7.5: Optische Flussverfahren. Oben: Frame 100 und 102 der LKW-Rückfront-Sequenz, im Frame 100 sind die betrachteten Bildregionen eingezeichnet. Die unteren drei Bilder zeigen die Ergebnisse der verschiedenen Flussverfahren: PowerFlow, KLT (jeweils Flusslänge farbkodiert) und Variationsansatz (farbkodiert die Flussrichtung, intensitätskodiert die Flusslänge).

	PowerFlow	KLT	denseFlow	Tracking
●	1,00076 \Rightarrow 556,6 m	1,0117 \Rightarrow 36,2 m	1,0136 \Rightarrow 31,1 m	1,01339 \Rightarrow 31,6 m
●	1,0118 \Rightarrow 35,9 m	1,0132 \Rightarrow 32,1 m	1,0127 \Rightarrow 33,3 m	1,01232 \Rightarrow 34,3 m
○	1,0095 \Rightarrow 44,5 m	1,0112 \Rightarrow 37,8 m	1,0129 \Rightarrow 32,8 m	1,01332 \Rightarrow 31,76 m
●	1,0249 \Rightarrow 17,0 m	1,0259 \Rightarrow 16,3 m	1,0174 \Rightarrow 24,3 m	1,02413 \Rightarrow 17,6 m
●	1,0159 \Rightarrow 26,6 m	1,0185 \Rightarrow 22,8 m	1,0182 \Rightarrow 23,2 m	1,01511 \Rightarrow 28,0 m

Tabelle 7.1: Ergebnisse der Entfernungsmessung aus optischem Fluss. In der Tabelle sind die Skalierungsschätzung und die daraus abgeleitete Entfernung für die Regionen in Abbildung 7.5 eingetragen.

Ist in der untersuchten Bildregion ein Hindernis abgebildet, so liefert der vorgestellte Ansatz mit Region Tracking und der Variationsansatz zur Flussfeldberechnung die genauesten Ergebnisse. Dies liegt daran, dass die meisten Beobachtungen in die Rechnung einfließen und die Ungenauigkeit zur Bestimmung der Parameter am geringsten ist.

Der Variationsansatz liefert für jeden Bildpunkt einen Flussvektor. Rechnet man auf dem Flussfeld ohne Zusatzinformation, so geht jeder Vektor mit gleichem Gewicht in die Rechnung ein. Dies wirkt sich negativ aus, wenn in Teilen der betrachteten Region Straße abgebildet ist. Die anderen Flussverfahren rechnen nur für markante Bildpunkte Flussvektoren aus. Solche markanten Flusspunkte sind hauptsächlich auf Hindernissen zu finden. Das Tracking arbeitet direkt am Bild. Die Berechnung der inkrementellen Bewegungsparameter nutzt hauptsächlich Pixel, die hohe räumliche Ableitung haben und einen hohen Grauwertunterschied zwischen den Zeitpunkten aufweisen. Dies ist dann der Fall, wenn Regionen viel Struktur aufweisen, daher werden Hindernisse gegenüber homogener Straße implizit höher gewichtet und die Schätzung der Parameter für das Hindernis ist robust.

Im Bereich um den Focus of Expansion auf dem Hindernis schneidet das PowerFlow Verfahren am schlechtesten ab. Weil nur ein pixelgenaues Flussfeld errechnet wird, war dies auch zu erwarten. Das Tracking und der Variationsansatz liefern in diesem Fall das genaueste Ergebnis, wobei das Tracking geringfügig besser abschneidet, da direkt iterativ auf dem Bild gearbeitet wird und nicht in zwei Schritten zuerst die Flussvektoren und dann die Skalierung geschätzt werden muss.

Ist das angenommene Modell in der Bildregion nicht vorhanden (z.B. wenn die Straßenebene in der Region abgebildet ist), ist das Ergebnis aller Verfahren eine falsche Schätzung der Skalierung. Es wird jeweils eine Hypothese für ein Hindernis aufgestellt, welches zu sich näher als 25m befindet. Ein solcher Fall muss schnellstmöglich abgefangen werden, um eine falsche Notbremsung zu vermeiden. Hier bietet sich die vorgeschlagene Hypothesenverifikation an.

7.3.2 Vergleich mit Ego-Motion

In Abschnitt 4.2 wurde gezeigt, wie eine Schätzung des relativen Nickwinkels aus der Verschiebung von Regionen im Bild berechnet werden kann. Diese Nickwinkelschätzung wurde mit einem aktuell bei DaimlerChrysler in der Entwicklungsphase stehenden monokularen Ego-Motion Verfahren verglichen. Dabei wurde ein *unendlich* weit entferntes Objekt (Wolke) getrackt. Bei unendlich weit entfernten Objekten ist der Translationsanteil in der Verschiebung im Bild gleich Null (wegen Division durch die Entfernung). Bei einem zweiten Test wurde die Nickwinkelschätzung aus der Translation der Hypothesenverifizierung für die Auffahrt auf mehrere stehende LKW verglichen.

Die Nickwinkelschätzung sowohl aus der Verschiebung als auch aus dem Ego-Motion Verfahren liefern ähnliche Werte (vgl. Abbildung 7.6). Für die LKW-Auffahrt entspricht die aufsummierte quadratische Abweichung beider Verfahren voneinander über 55 Frames lediglich

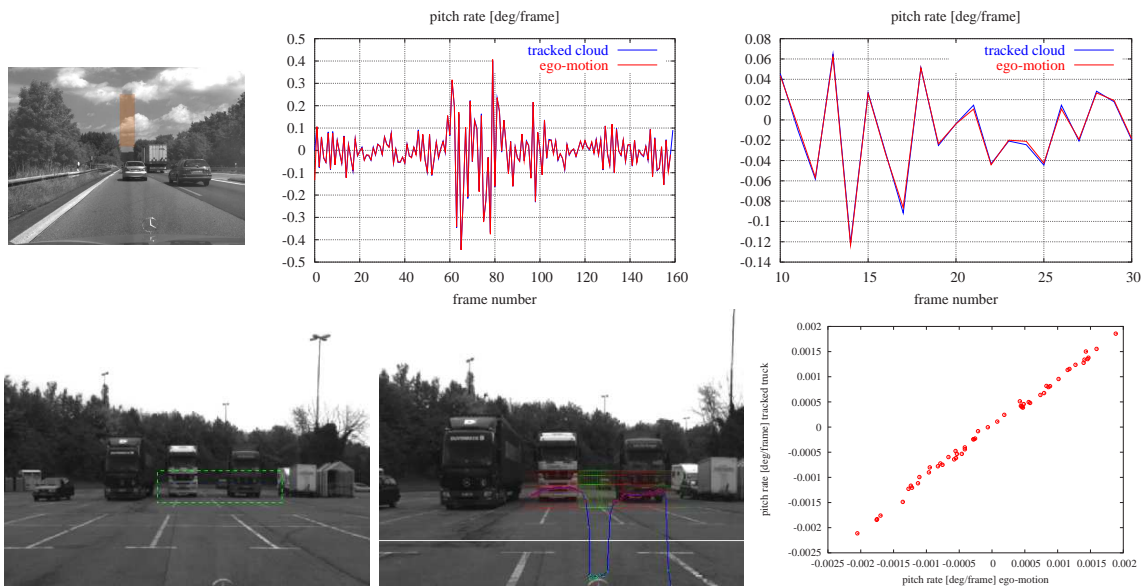


Abbildung 7.6: Schätzung des relativen Nickwinkels. Oben: die getrackte Wolke und die Schätzung des relativen Nickwinkels im Vergleich zu Daten aus der Ego Motion; unten: erstes und letztes Frame aus der Hypothesenverifizierung sowie geschätzter Nickwinkel im Vergleich zur Ego Motion. Initialisierung bei 40m, gefahrene Distanz 10,50m.

0.00051 Grad. Dies spricht entweder dafür, dass die Ergebnisse robust geschätzt werden oder beide Verfahren falsch sind. Da beide Verfahren auf völlig anderen Ansätzen aufbauen, spricht dies für die Robustheit beider Verfahren bei der Berechnung des relativen Nickwinkels.

7.4 Ergebnisse der Hindernisdetektion

In diesem Abschnitt werden die Ergebnisse der zweistufigen Hindernisdetektion vorgestellt. Drei Beispielszenarien demonstrieren die Funktionalität des in dieser Arbeit erarbeiteten Systems. Dabei werden Hindernisse im Fahrzeugkorridor (im Bereich des Focus of Expansion) betrachtet. Im Folgenden werden die Daten der einzelnen Abbildungen kurz erläutert.

Die Ansicht aus der Hypothesengenerierung (jeweils linkes Bild) zeigt den Fahrkorridor (gelb) sowie die Umrisse der einzelnen Unterregionen, für welche die Skalierung und Translation geschätzt wird. Der Focus of Expansion liegt genau in der Spitze des Dreiecks, welches den Fahrkorridor darstellt. Auf Hindernisse untersucht wird die Bildregion $[x1, x2] \times [y1, y2]$. Aufgestellte Hindernishypothesen sind in orange eingezeichnet. Es werden nur Hypothesen für Hindernisse im Fahrkorridor aufgestellt, da andere Hindernisse keine potentielle Gefahr in stationären Szenen darstellen. Direkt über der Hindernishypothese ist die Entfernung und die angenommene Zeit bis zum Aufprall (bei konstanter Geschwindigkeit des Versuchsträgers) im

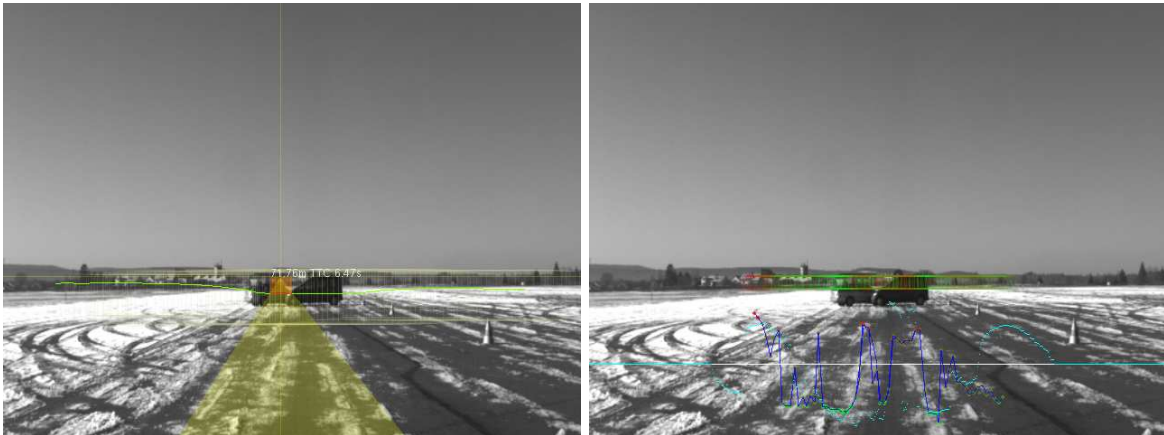
Bild eingezeichnet. Die ermittelten Entfernungen für die gesamte betrachtete stationäre Szene sind durch die grüne Kurve in das Bild projiziert.

Die Ansicht der Hypothesenverifizierung (jeweils rechtes Bild) zeigt ebenfalls die Umrisse der einzelnen Unterregionen, für welche eine Hypothesenverifizierung gerechnet wird. Der Quotient der Wahrscheinlichkeiten für die beiden Annahmen eines Hindernisses und der Straßenebene (Differenz der SSD-Werte) bestimmt die Farbkodierung. Zusätzlich zeigt der Plot am unteren Bildrand die aktuellen Werte für die einzelnen Regionen farblich kodiert auf der blauen Kurve und die zeitlich gefilterten Werte in cyanblau. Die weiße Linie bedeutet Differenz Null. In diesem Fall sind beide Annahmen gleich wahrscheinlich, beispielsweise bei der Initialisierung der Hypothesenverifizierung oder außerhalb des untersuchten Bildbereiches.

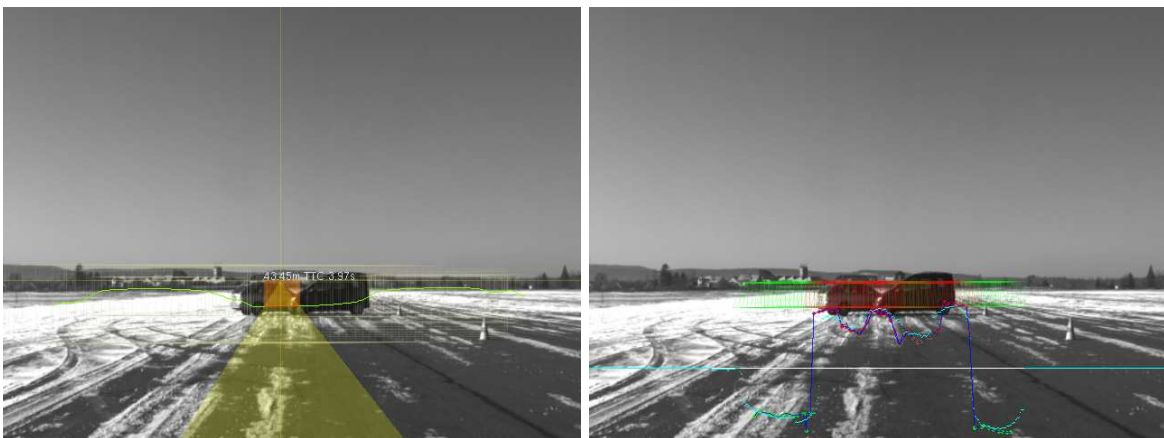
Die monokularen Videosequenzen wurden mit einer Kamera mit fokaler Länge von 6mm und 40° Öffnungswinkel aufgenommen. Die Geschwindigkeit des Versuchsträgers kann für die angegebenen Aufnahmen aus Entfernung des potentiellen Hindernisses und Zeit bis zum Aufprall errechnet werden. Sie beträgt zwischen 5m/s und 15m/s. Die Hypothesengenerierung trackt 80 Hypothesen mit jeweils 120×40 Pixel auf einem Pentium IV mit 3.0GHz in Echtzeit (23fps). Sobald die Hypothesenverifizierung dazugeschaltet wird, sinkt die Framerate auf 13fps. Die Hypothesenverifizierung trackt zwei mal 80 Hypothesen je 25×50 Pixel. Damit ist in dieser Arbeit ein System zur Hindernisdetektion aus monokularen Bildsequenzen in Echtzeit hergeleitet, untersucht und realisiert worden.



Abbildung 7.7: Malmshiem Sequenz - das Hindernis ist im ersten Frame mehr als 100m vom Versuchsträger entfernt.

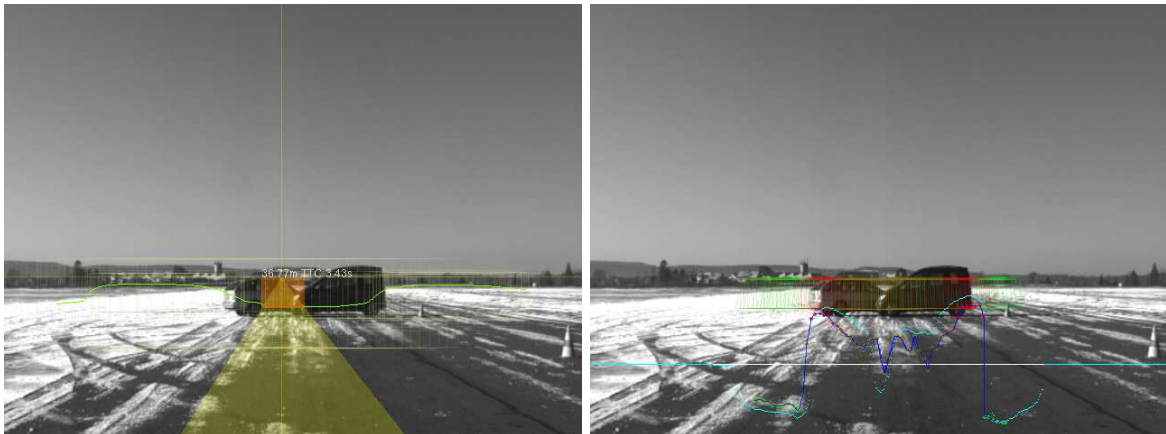


Frame 90: das Hindernis wurde bei 72m detektiert. Es bleiben ca. 6 Sekunden bis zum Aufprall, in dieser Entfernung ist ein Verifizieren noch nicht möglich. Auch der Nickwinkel ist nicht korrekt, weshalb die Verifizierung zu hoch aufsetzt.

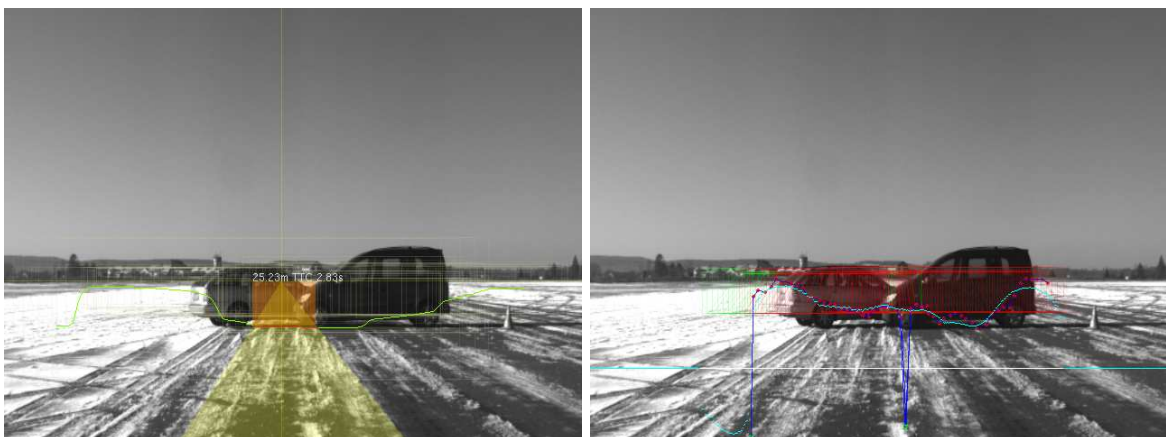


Frame 133: bei 43m ist das Hindernis verifiziert. Es bleiben 4 Sekunden zum Abbremsen. Die Zeit für das Verifizieren ist abhängig von der Distanz des Hindernisses.

Abbildung 7.7 Fortsetzung:



Frame 149: die Hindernisverifizierung ist in der Bildmitte ungenau. Zum einen sind die Verschiebungen im FOE sehr klein, zum anderen ist im Hintergrund über dem Horizont viel Struktur. Dies hat den Effekt, dass die Straßenhypothese (mit Annahme der Identität oberhalb des Horizonts) bevorzugt wird.



Frame 185: die projizierten Entfernungen der Hypothesengenerierung (linkes Bild, grüne Linie) zeigen, dass links vom Hindernis die Fahrbahn frei ist. Die Breite des Hindernisses ist durch die Verifizierung relativ genau bestimmt worden, so dass das Hindernis umfahren werden kann.



Abbildung 7.8: Böblinger Parkplatzsequenz - erstes Frame der Sequenz.

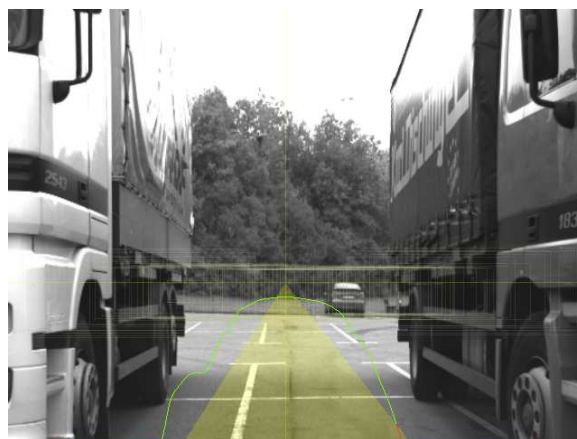
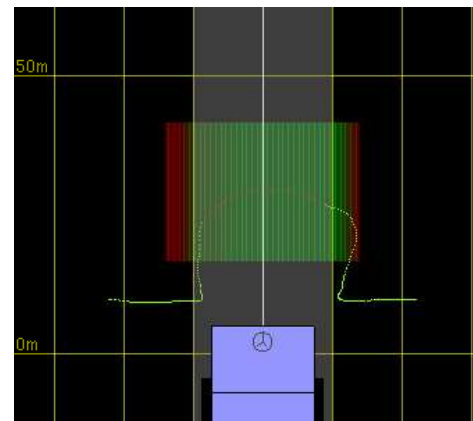


Frame 35 - im Bereich des FOE wird eine falsche Hindernishypothese aufgestellt. Die beiden LKW beeinflussen die Skalierungsschätzung. Die Hypothesenverifizierung lehnt die Hindernishypothese im FOE ab und verifiziert die stehenden LKW als Hindernisse.

Abbildung 7.8 Fortsetzung:



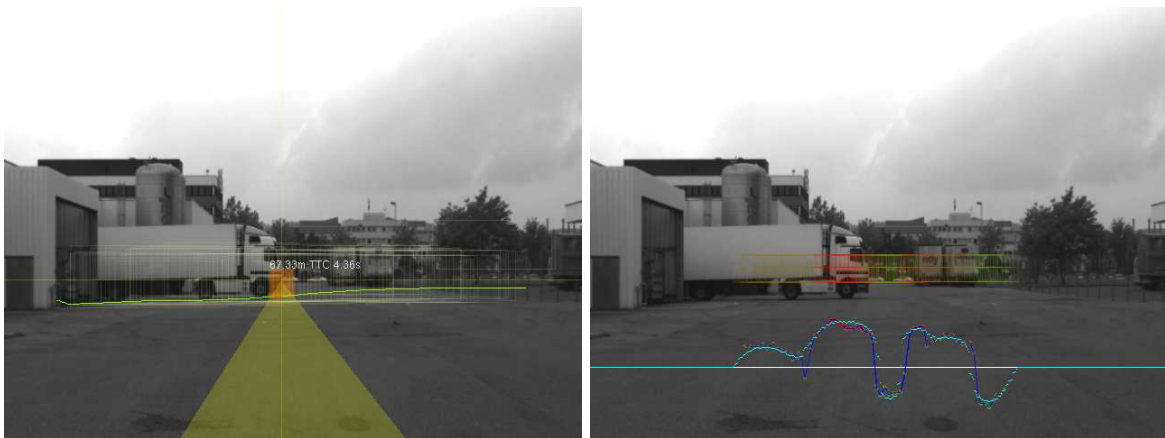
Frame 80 - die einzelnen Regionen der Hypothesengenerierung lösen sich langsam von den beiden LKW und die Hindernishypothese wird für ein weiter entferntes Hindernis aufgestellt. Der Verifizierungsschritt hat die Objektgrenzen der LKW lokalisiert und den freien Fahrkorridor bestätigt (Bild rechts, Abstand der vertikalen Linien 1m). Der freien Fahrt steht im wahrsten Sinne des Wortes *nichts mehr im Wege*.



Frame 139 - aus der Skalierungsschätzung werden nun Entfernungen von über 75m errechnet. Es wird keine Hindernishypothese aufgestellt, da die Bäume im Hintergrund noch zu weit entfernt sind.

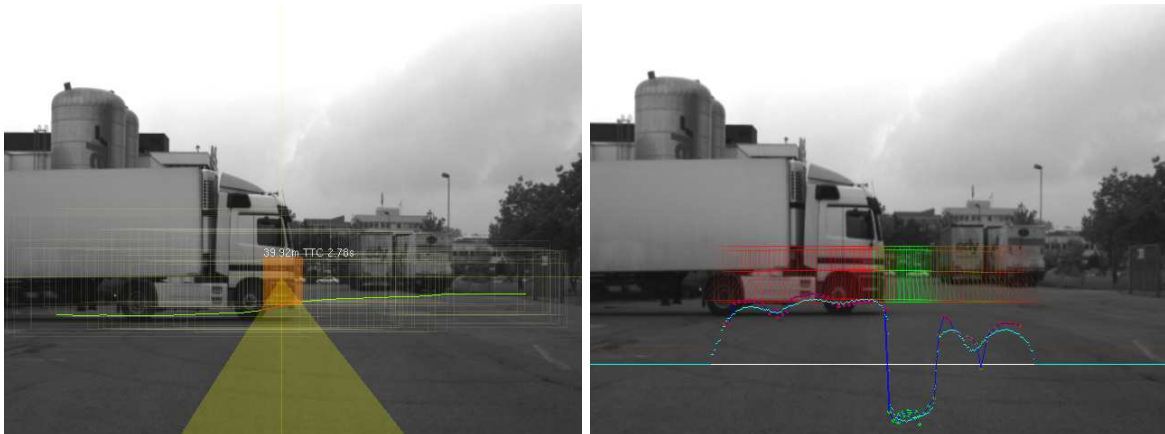


Abbildung 7.9: LKW Beladestation - erstes Frame der Sequenz.



Frame 149 - der querstehende LKW wird in 69m Entfernung erkannt, es verbleiben 4,4sek bis zum Aufprall. Die Entfernungsmessung vom Stereo sowie die in das Bild projizierten Entfernungen (grüne Linie links) bestätigen die Entfernungsschätzung. Die Hypothesenverifizierung erkennt sowohl den querstehenden LKW als stehendes Hindernis in 67m Entfernung als auch die zwei LKW im Hintergrund. Der relative Entfernungsunterschied beider Hindernisse ist klein, so dass die Hindernishypothese wahrscheinlicher ist, als die Straßenhypothese.

Abbildung 7.9 Fortsetzung:



Frame 215 - die Hindernishypothese deutet ein Hindernis in 40m an, noch ca. 3sek verbleiben bis das Hindernis erreicht ist. In der Hypothesenverifizierung wird immer klarer deutlich, dass das Hindernis umfahren werden kann. Die Objektgrenze im Bereich des *Focus of Expansion* wird hinreichend genau lokalisiert.



Frame 292 - das Hindernis ist noch 10m entfernt. Die Objektgrenze ist trotz der Schwierigkeit, Hypothesen im *Focus of Expansion* zu bestätigen oder zu verwerfen erkannt worden und das Hindernis kann nun sicher umfahren werden.

7.5 Zusammenfassung der Ergebnisse

Die Ergebnisse aus Kapitel 6 und 7 sollen an dieser Stelle kurz zusammengefasst werden.

- Die Ergebnisse auf synthetischen Daten sind auf reale Daten übertragbar, wenn die Bewegung des Versuchsträgers bekannt ist und das Bauklotzmodell angenommen werden kann.
- Der Vergleich mit Multi-Sensor-Systemen wie Radar/Lidar und Stereo zeigt die genaue Entfernungsschätzung des vorgeschlagenen Ansatzes mit einem visuellen Sensor.
- Die motivierte Schätzung des relativen Nickwinkels ist mit der Schätzung aus der vollen Ego-Motion Schätzung vergleichbar.
- Die Gegenüberstellung des entwickelten Systems und optischer Flussverfahren motiviert den Einsatz des Region Tracking zur Hindernisdetektion im Focus of Expansion sowie die Nutzung von Bewegungsinformation zur Hinderniserkennung.
- Die Aufteilung der Hindernisdetektion in die zwei Schritte der Hypothesengenerierung und Hypothesenverifizierung ermöglicht ein robustes Detektieren von Hindernissen in monokularen Videosequenzen.
- Eine Detektion von Hindernissen in der Größe eines Kleinwagens ist mit einer Kamera mit fokaler Länge von 6mm robust bis zu einer Entfernung von 50m in Echtzeit möglich.
- Die Hindernisdetektion als Gesamtsystem liefert die Breite des Hindernisses exakt genug für ein Fahrerassistenzsystem, um in das Geschehen eingreifen zu können.
- Die Initialisierung der betrachteten Regionen spiegelt sich in der Genauigkeit der Daten wieder.
- Falsche Hindernishypothesen werden in der Hypothesenverifizierung zum größten Teil robust herausgefiltert. Problematisch sind falsche Nickwinkelannahmen.
- Das Tracken der Regionen nimmt mit Abstand den rechenintensivsten Teil ein. Die Auswertung der Daten benötigt vergleichsweise wenig Rechenzeit.

Kapitel 8

Zusammenfassung und Ausblick

In dieser Arbeit wurde ein System zur Detektion von Hindernissen im Fahrkorridor durch Auswertung monokularer Bildsequenzen entwickelt. Die Hindernisdetektion wird zweistufig mit einer Hypothesengenerierung und einer Hypothesenverifizierung durchgeführt. Die Information wird dabei aus der Bewegung der Hindernisse im Bild gewonnen. Durch das Betrachten und Verfolgen von Bildregionen werden die Position und die Breite von Hindernissen zuverlässig ermittelt. Das vorgestellte System erreicht eine Framerate von 13fps bis zu 28fps und ist damit echtzeitfähig. Der erste Schritt, die Hypothesengenerierung, überwacht eine vorher festgelegte Bildregion. Für Teilbereiche dieser Region wird über eine Skalierungsschätzung die Entfernung von potentiellen Hindernissen geschätzt. Über Fehlerfortpflanzung kann neben der Entfernungsschätzung auch deren Standardabweichung angegeben werden. Die Position des Hindernisses in der Welt wird aus der Position im Bild und der Entfernungsschätzung bestimmt.

Der zweite Schritt, die Hypothesenverifizierung, vergleicht die angenommene Transformation eines hypothetischen Hindernisses in gegebener Entfernung mit der Transformation der Straßenebene. Dabei wird die Breite des Hindernisses bestimmt oder die Hindernishypothese wird verworfen. Als sogenanntes *traffic light modul* ist die Hypothesenverifizierung in der Lage, ihre Aussage selbständig zu bewerten. Auch das Ergebnis *keine Aussage möglich* ist somit zulässig. Dadurch wird die Hindernisdetektion robust und ein Eingreifen eines Fahrerassistenzsystems ermöglicht.

Die experimentellen Untersuchungen zeigen die Funktionalität des Verfahrens. Hindernisse werden als solche erkannt und die horizontalen Hinderniskanten als Begrenzungen detektiert. Untersuchungen in realen Bildsequenzen zeigen, dass Hindernisse bereits in 50m robust detektiert werden. Eine detailliertere Untersuchung über die Anzahl der nicht detektierten und fälschlich detektierten Hindernisse ist Gegenstand weiterer Arbeiten.

Vor einem Einsatz in einem Fahrzeug sind jedoch weitere Einflüsse zu untersuchen. In dieser Arbeit wurde ein nahezu fehlerfreies Inertialsystem vorausgesetzt. Fehler in der Geschwindigkeit führen zwangsweise dazu, dass die Entfernung falsch geschätzt wird. Nickwinkelschwan-

kungen führen außerdem dazu, dass die Hindernisverifikation zwangsweise zwei falsche Modelle miteinander vergleicht. Weiterführende Arbeiten sind zur besseren Modellierung dieser Fehler notwendig.

Untersuchungen wurden angestellt, den Fußpunkt eines Hindernisses oder den Nickwinkel des Fahrzeuges in die Schätzung der Entfernung mit einfließen zu lassen. Der vorgeschlagene Ansatz hat sich als nicht zuverlässig erwiesen und muss verbessert werden. Entsprechende Vorschläge wurden im Abschnitt 6.3 gemacht.

Die Verarbeitungsgeschwindigkeit des gesamten Systems wird hauptsächlich vom Trackingverfahren bestimmt. In bisherigen Ansätzen werden alle Beobachtungen einer Region genutzt. In [11] wurde gezeigt, dass die geschätzte Transformation hauptsächlich von solchen Pixeln abhängt, deren Gradient *groß* ist gegenüber anderen Pixeln der betrachteten Region. Über Pixelselektion kann das Tracken beschleunigt werden.

Neben der Nutzung zweier Bilder zum Schätzen der Transformation können auch mehrere Bilder der zu trackenden Bildregion genutzt werden, um die Bewegungsparameter zu schätzen (vgl. [71]). Ziel eines solchen Ansatzes ist es, das Verfahren robuster zu gestalten. Ein Nutzen aller Bilder aus der Historie ist zu zeitaufwendig. Hier sind weiterführende Untersuchungen notwendig, um das Tracken effizient und robust zu gestalten.

Bisher wurden sowohl zur Hypothesengenerierung als auch zur Hypothesenverifizierung Teilregionen gleichmäßig in der zu untersuchenden Region verteilt. Durch geschickte Verteilung und Konzentration in Bereichen potentieller Hindernisse ist eine bessere Generierung von Hindernishypothesen denkbar. Ein möglicher Ansatz ist die Nutzung von Partikelfiltern. Die Fusion des in dieser Arbeit vorgeschlagenen Algorithmus mit einem Partikelfilteransatz ist Gegenstand weiterer Arbeiten.

Der ausgearbeitete Ansatz zur Hypothesenverifizierung kann als Klassifizierung der Bildregion in die zwei Klassen *Hindernis* und *Straßenebene* verstanden werden. Eine solche Herangehensweise motiviert die Nutzung von Trainingsmethoden der Klassifizierung, um die Hypothesen voneinander zu unterscheiden. Man könnte testen, ob dadurch ein besseres Differenzmaß erreicht werden kann.

Anhang A

Radargrundlagen

Der Begriff Radar umfasst Methoden zur Lokalisierung von Objekten mit Hilfe elektromagnetischer Wellen. Im Folgenden werden die grundlegenden Funktionsweisen des Radar (emphRadio detection and ranging) erläutert. Dabei wird vor allem der Einsatz des Radar in Fahrzeugen dargestellt. Die Einführung orientiert sich an [31] und [88].

A.1 Funktionsweise des Radar

Das Radar ist ein aktives Verfahren zur Objekterkennung. Es besteht aus einer Sende- und einer Empfangseinheit. Die Sendeeinheit strahlt hochfrequente elektromagnetische Wellen gebündelt in den Raum ab. Treffen diese Wellen auf metallische oder dielektrische Objekte, werden sie als Echoimpuls reflektiert und von der Empfangseinheit registriert. Ein für das Radar zu erkennendes Objekt muss sich dabei bezüglich seiner elektromagnetischen Eigenschaften von denen des Ausbreitungsmediums unterscheiden. Allgemein gilt, dass sich Material mit guter elektrischer Leitfähigkeit auch gut als Reflektor für elektromagnetische Wellen eignet. Prinzipiell können aus der Energie der reflektierten Wellen, der Feldstärke, Rückschlüsse auf Objektposition, Objektgeschwindigkeit und auch Objektgröße ermittelt werden.

Im Automobilbereich wird das Pulsradar verwendet, bei dem Impulse von der Sendeeinheit ausgesendet werden (im Gegensatz zum Dauerstrichradar, wie es beispielsweise im Polizeiradar eingesetzt wird). Zur Messung von Abständen wird die vergangene Zeit zwischen Sende- und Echosignal ausgewertet. Bei der Abstandsmessung spielt außerdem der sogenannte *Eindeutigkeitsbereich* eine Rolle. Mehrdeutigkeiten können bei einem Pulsradar zustande kommen, wenn bei einer periodischen Pulsaussendung das Empfangssignal nicht eindeutig dem Sendeimpuls zugeordnet werden kann. Der Eindeutigkeitsbereich gibt an, bis zu welchen Impulsabständen eine korrekte Distanz berechnet werden kann.

Durch Auswerten der Phaseninformation des empfangenen Signals (kohärentes Pulsradar) lassen sich Aussagen über die Bewegung eines Objektes ableiten. Bei kontinuierlicher Beobachtung eines Radarziels erzeugt dies bei konstanter Geschwindigkeit eine zeitlineare Laufzeit-

und Phasenänderung (Dopplereffekt). Die direkt messbare (Relativ-)Geschwindigkeit ist ein Vorteil gegenüber anderer Sensoren wie Lidar oder Kameras, bei welchen die Geschwindigkeiten aus Messungen geschätzt werden müssen.

A.2 Winkelbestimmung über die sequentielle Mehrkeulenbildung

Beabsichtigt man neben der Entfernung den Winkel, unter dem ein Radarziel detektiert wurde, zu bestimmen, so ergibt sich das Problem, dass die Feldstärke des reflektierten Signals zur Seite nichtlinear abnimmt. Dies bedeutet, dass ein Zielobjekt unter verschiedenen Winkeln ein anderes Reflexionsverhalten (Richtcharakteristik) aufweist. In Abbildung A.1 ist zur Verdeutlichung die Richtcharakteristik eines Radarsignals dargestellt.

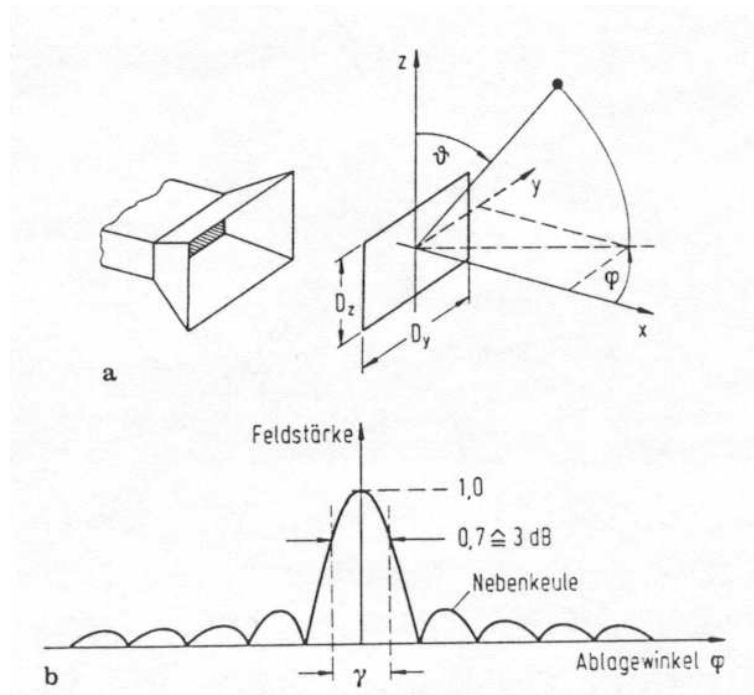


Abbildung A.1: Richtcharakteristik eines Radarsignals. Oben ist der Radarsensor schematisch dargestellt. Unten ist die Richtcharakteristik für den Ablagewinkel aufgetragen. Die nichtlineare Abnahme der Feldstärke zur Seite hin ist klar erkennbar.

Anhand der Energie des reflektierten Signals ist nicht zu erkennen, ob die Energie von einem schwachen Reflektor (schwach reflektierendes Objekt) in Hauptstrahlrichtung oder von einem guten Reflektor unter einem großen Winkel empfangen wurde. Ein Radarkegel hat im Gegensatz zu visuellen Sensoren keinen scharfen Grenzbereich sondern wird zu den Seiten hin empfangsschwach. Bei beweglichen Systemen (Drehung oder Ablenkung) kann aus der

Ausrichtung der Antenne eine Winkelinformation gewonnen werden. Bei nicht beweglichen Radarsystemen werden meistens Verfahren wie die *sequentielle Mehrkeulenbildung* eingesetzt, welche auf dem Vergleich mehrerer sequentiell geschalteter, sich überlappender Radarstrahler basiert. Eine beispielhafte Anordnung ist in Abbildung A.2 ersichtlich.

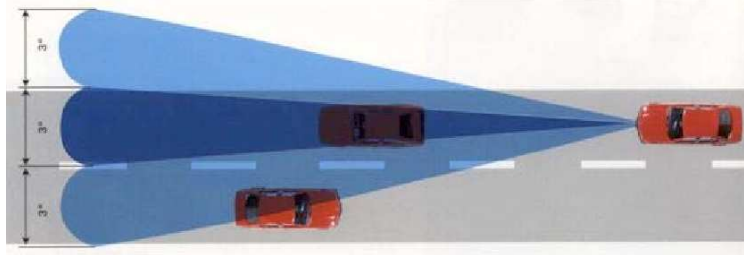


Abbildung A.2: Abdeckung eines Radarsensors im Automobilbereich [88]

Liegt ein Objekt im Erfassungsbereich der Hauptkeulen zweier Radarstrahler, so ergeben sich Empfangsimpulse mit individuellen Amplituden, über welche eine Winkelzuordnung durchgeführt wird. Für jede Keule wird auf Basis der Richtcharakteristik und anhand der Amplitude ein möglicher Winkel bestimmt. Im allgemeinen Fall liefert dies zwei Lösungen pro Keule, da nichts über die relative Lage des Radarziels zur Radareinheit bekannt ist. Da sich die Keulen aber nur in einem Bereich überschneiden, ist das Winkelvorzeichen in der Regel eindeutig. Weichen Winkel und Amplitude der beiden Radarkeulen für das potentielle Radarziel nicht signifikant voneinander ab, ist das Objekt lokalisiert. Wird ein Toleranzbereich für die Abweichung überschritten, ist keine Aussage möglich. Letztlich ist in diesem Fall unklar, ob es sich um ein- und dasselbe Zielobjekt oder um querbewegte Objekte handelt.

Bei den Zielobjekten des Radar handelt es sich um reine Punktziele, die den Reflexionsschwerpunkt des Objektes angeben (welcher in der Regel nicht dem visuellen Schwerpunkt entspricht). Aussagen über Objektgröße können ohne exaktes Wissen über die Objekteigenschaften nicht gemacht werden. Da der Reflexionsschwerpunkt für verschiedene Radarkeulen nicht identisch ist und sich außerdem bei relativer Bewegung zum Zielobjekt ständig ändert, treten häufig Unstetigkeiten bezüglich des detektierten Winkels zum Zielobjekt auf. Dies verdeutlicht die Schwierigkeit der Winkelbestimmung.

Literaturverzeichnis

- [1] BARRON, J. L. ; FLEET, D. J. ; BEAUCHEMIN, S. S.: Performance of Optical Flow Techniques. In: *International Journal of Computer Vision* 12 (1994), Nr. 1, S. 43–77
- [2] BEECHAM, M.: Global market review of driver assistance systems (collision avoidance) - forecasts to 2011. August 2005 (4307068). – Forschungsbericht. Aroq Limited
- [3] BERTOZZI, M ; BROGGI, A. ; CASTELLUCCIO, S.: A real-time oriented system for vehicle detection. In: *Journal of Systems Architecture* 12 (1997), Nr. 1-5, S. 317–325
- [4] BERTOZZI, M. ; BROGGI, A. ; CELLARIO, M. ; FASCIOLI, A. ; LOMBARDI, P. ; ; PORTA, M.: Artificial Vision in Road Vehicles. In: *Proceedings of the IEEE* Bd. 90, 2002, S. 1258–1271
- [5] BERTOZZI, M. ; BROGGI, A. ; FASCIOLI, A.: Stereo inverse perspective mapping: theory and applications. In: *Image and Vision Computing Journal* 16 (1998), S. 585–590
- [6] BERTOZZI, M. ; BROGGI, A. ; FASCIOLI, A. ; NICHELE, S.: Stereo Vision-based Vehicle Detection. In: *IEEE Intelligent Vehicle Symposium* Bd. 12, 2000, S. 39–44
- [7] BETKE, M. ; HARITAOGU, E. ; DAVIS, L.S.: Multiple Vehicle Detection and Tracking in Hard Real Time. In: *Proc. IEEE Conference on Intelligent Vehicles*, 1996, S. 351–356
- [8] BETKE, M. ; HARITAOGU, E. ; DAVIS, L.S.: Highway Scene Analysis in Hard Real-Time. In: *Proc. IEEE Conference on Intelligent Transportation Systems*, 1997
- [9] BLACK, M. ; JEPSON, A.: EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. In: *Proc. European Conference Computer Vision*, 1996, S. 329–342
- [10] BROX, T. ; BRUHN, A. ; PAPENBERG, N. ; WEICKERT, J.: High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: *Proceedings of the 8th European Conference on Computer Vision*, Springer Verlag, 2004, S. 25–36
- [11] BUENAPOSADA, J.M. ; BAUMELA, L.: Speeding up SSD planar tracking by pixel selection. In: *ICIP (1)*, 2002, S. 565–568

- [12] BÖHM, T.: *Erkennung und Verfolgung von Fahrzeugen im Videobild*, Institut für Parallele und Verteilte Systeme der Universität Stuttgart, Diplomarbeit, April 2003
- [13] COMANICIU, D. ; MEER, P. ; RAMESH, V.: Kernel-Based Object Tracking. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2004), S. 564–575
- [14] CREMERS, D. ; SOATTO, S.: Motion Competition: A Variational Approach to Piecewise Parametric Motion Segmentation. In: *International Journal of Computer Vision* 63 (2005), Nr. 3, S. 249–265
- [15] CUCCHIARA, R. ; PICCARDI, M.: Vehicle Detection under Day and Night Illumination. In: *International ICSC Symposium on Intelligent Industrial Automation*, 1999
- [16] DAIMLERCHRYSLER: Hightech Report. 2005 (2). – Forschungsbericht. DaimlerChrysler
- [17] DELLAERT, F. ; THORPE, C.: Robust Car Tracking using Kalman filtering and Bayesian templates. In: *Conference on Intelligent Transportation Systems*, 1997
- [18] DEUTSCH, B. ; GRÄSSL, Ch. ; BAJRAMOVIC, F. ; DENZLER, J.: A Comparative Evaluation of Template and Histogram Based 2D Tracking Algorithms. In: *DAGM-Symposium 2005*, 2005, S. 269–276
- [19] DICKMANN, E.D. ; BEHRINGER, R. ; DICKMANN, D. ; HILDEBRANDT, T. ; MAURER, M. ; THOMANEK, F. ; SCHIEHLEN, J.: The Seeing Passenger Car 'VaMoRs-P'. In: *International Symposium on Intelligent Vehicles*, 1994, S. 68–73
- [20] DUTTER, R. ; HUBER, P.: Numerical Methods for Nonlinear Robust Regression Problems. In: *Journal of Statistical and Computational Simulation* 13 (1981), Nr. 2, S. 79–113
- [21] ENKELMANN, W.: Obstacle Detection by Evaluation of Optical Flow Fields from Image Sequences. In: *Proc. European Conference Computer Vision*, 1990, S. 134–138
- [22] ENKELMANN, W.: *Entwicklung von Systemen zur Interpretation von Straßenverkehrsszenen durch Bildfolgenauswertung*. infix Verlag, 1997
- [23] ESCH, H.E. ; ZHANG, S. ; SRINIVASAN, M.V. ; TAUTZ, J.: Honeybee dances communicate distances measured by optic flow. In: *Nature* 411 (2001), Mai, S. 581–583
- [24] FAUGERAS, O.: *Three-Dimensional Computer Vision*. MIT-Press, 1993
- [25] FRANKE, U.: *Intelligente Automobile für den Verkehr der Zukunft*. 1996. – Forschungsbericht. Daimler-Benz AG, Abteilung F1M/IA
- [26] FRANKE, U. ; GAVRILA, D. ; GÖRZIG, S. ; LINDNER, F. ; PAETZOLD, F. ; WÖHLER, C.: Autonomous Driving approaches Downtown. In: *IEEE Intelligent Systems* Bd. 13, 1999, S. 40–48

- [27] FRANKE, U. ; JOOS, A.: Real-time Stereo Vision for Urban Traffic Scene Understanding. In: *Proc. IEEE Conference on Intelligent Vehicles*, 2000, S. 273–278
- [28] FÖRSTNER, W.: 10 Pros and Cons against performance characterization of vision algorithms. In: *Performance Characteristics of Vision Algorithms* (1996)
- [29] FÖRSTNER, W.: *Vorlesungsunterlagen zur Vorlesung Photogrammetrie*. 2002. – Rheinische Friedrich-Wilhelms-Universität Bonn
- [30] GEHRIG, S.K.: *Design, Simulation and Implementation of a Vision-based Vehicle-Following System*, Eberhard Karl-Universität Tübingen, Fakultät für Physik, Dissertation, 2000
- [31] GERN, A.: *Multisensorielle Spurerkennung für Fahrerassistenzsysteme*. logos Verlag, 2006. – ISBN 3-8325-1103-2
- [32] GIACHETTI, A. ; CAMPANI, M. ; TORRE, V.: The Use of Optical Flow for Road Navigation. In: *IEEE Transactions on Robotics and Automation* Bd. 14, 1998, S. 34–48
- [33] GOERICK, C. ; NOLL, D. ; WERNER, M.: Artificial neural networks in real-time car detection and tracking applications. In: *Pattern Recognition Letters* 17 (1996), S. 335–343
- [34] HABERÄCKER, P.: *Digitale Bildverarbeitung - Grundlagen und Anwendungen*. Hanser Verlag, 1994. – 25–30 S
- [35] HAGER, G. ; BELHUMEUR, P.: Efficient region tracking with parametric models of geometry and illumination. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1998), S. 1025–1039
- [36] HANDMANN, U. ; KALINKE, T. ; TZOMAKAS, C. ; WERNER, M. ; SEELEN, W.v.: An Image Processing System for Driver Assistance. In: *Image and Vision Computing* 18 (2000), S. 367–376
- [37] HANDMANN, U. ; LORENZ, G. ; SCHNITGER, T. ; SEELEN, W.v.: Fusion of Different Sensors and Algorithms for Segmentation. In: *Proceedings of the IEEE Conference on Intelligent Vehicles*, 1998, S. 499–504
- [38] HAUSSECKER, H. ; JAHNE, B.: Motion. In: *Computer Vision and Applications - A Guide for Students and Practitioners*. Academic Press, März 2000, S. 347–395. – ISBN 0-12-379777-2
- [39] HEINRICH, S.: Fast Obstacle Detection Using Flow/Depth Constraint. In: *Proc. IEEE Conference on Intelligent Vehicles*, 2002, S. 658–666

- [40] HEISELE, B.: *Objektdetektion in Straßenverkehrsszenen durch Auswertung von Farbbildfolgen*, Universität Stuttgart, Dissertation, 1998
- [41] HELLA: *Vortragsunterlagen, Hella-Tech Show*. Juli 2002
- [42] HORN, B.K.P.: *Robot Vision*. MIT Press, Cambridge, Massachusetts, 1986. – ISBN 0-262-08159-8
- [43] HULIN, B.: *Videobasierte Hinderniserkennung im Durchgangsraum des Stromabnehmers elektrischer Bahnen*, Technische Universität München, Dissertation, 2003
- [44] INDUSTRIENORM, Deutsche: *Orthophoto - Geoinformatik-Service*. Januar 2006. – URL www.geoinformatik.uni-rostock.de/einzel.asp?ID=1307
- [45] IRANI, M. ; ROUSSO, B. ; PELEG, S.: Robust Recovery of Ego-Motion. In: *Computer Analysis of Images and Patterns*, 1993, S. 371–378
- [46] JIN, H. ; FAVARO, P. ; SOATTO, S.: Real-Time Feature Tracking and Outlier Rejection with Changes in Illumination. In: *Proc. International Conference on Computer Vision*, 2001, S. 684–689
- [47] JURIE, F. ; DHOME, M.: Hyperplane Approximation for Template Matching. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002), S. 996–1000
- [48] KAEMPCHEN, N. ; FUERSTENBERG, K. ; DIETMAYER, K.: Ein Sensorfusionssystem für Automotive Sicherheits- und Komfortapplikationen / Aktive Sicherheit durch Fahrerassistenz. März 2004. – Forschungsbericht. TÜV-Akademie
- [49] KALINKE, T. ; SEELEN, W.v.: Entropie als Maß des lokalen Informationsgehalts in Bildern zur Realisierung einer Aufmerksamkeitssteuerung. In: *Mustererkennung 1996*, Springer Verlag, 1996, S. 627–634
- [50] KALINKE, T. ; TZOMAKAS, C. ; SEELEN, W.v.: A texture-based object detection and an adaptive model-based classification. In: *IEEE International Conference on Intelligent Vehicles*, 1998, S. 143–148
- [51] KE, Q. ; KANADE, T.: Transforming Camera Geometry to A Virtual Downward-Looking Camera: Robust Ego-Motion Estimation and Ground-Layer Detection. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003, S. 390–397
- [52] KE, Y. ; SUKTHANKAR, R.: PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* Bd. 2, 2004, S. 506–513

- [53] KLAPPSTEIN, Jens: Detection of static protruding objects. 2005. – Forschungsbericht. DaimlerChrysler
- [54] KOLLER, D. ; LUONG, Q.-T. ; MALIK, J.: Using Binocular Stereopsis for Vision-Based Vehicle Control. In: *Proc. IEEE Conference on Intelligent Vehicles*, 1994, S. 237–242
- [55] KRUGER, W. ; ENKELMANN, W. ; ROSSLE, S.: Real-time estimation and tracking of optical flow vectors for obstacle detection. In: *IEEE Intelligent vehicle symposium*, 1995, S. 304–309
- [56] KUNOTH, A. ; JÖRRES, S.: *Vorlesungsunterlagen zur Vorlesung Praktische Mathematik II*. Juli 2003. – Rheinische Friedrich-Wilhelms-Universität Bonn
- [57] KÜHNLE, A.: Symmetry-based recognition for vehicle rears. In: *Pattern Recognition Letters* 12 (1991), S. 249–258
- [58] LIU, T.L. ; CHEN, H.T.: Real-Time Tracking Using Trust-Region Methods. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2004), S. 397–402
- [59] LONGUET-HIGGINS, H.C. ; PRAZDNY, K.: The Interpretation of a Moving Retinal Image. In: *Proceedings of the Royal Society of London* Bd. 208, July 1980, S. 385–397
- [60] LOWE, D.G.: Distinctive image features from scale-invariant keypoints. In: *International Journal of Computer Vision* (2004), S. 91–110
- [61] LUCAS, B.D. ; KANADE, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981, S. 674–679
- [62] LUX, S.: *Schätzung der Position stationärer Objekte mit Hilfe eines aktiven Sehsystems*, Fachhochschule Stuttgart - Hochschule für Technik, Diplomarbeit, 2001
- [63] MATHWORLD: *Least Square Fitting*. 2006. – URL mathworld.wolfram.com/LeastSquaresFitting.html
- [64] MATTHEWS, N.D. ; AN, P.E. ; CHARNLEY, D. ; HARRIS, C.J.: Vehicle detection and recognition in greyscale imagery. In: *Control Engineering Practice* Bd. 4, 1996, S. 473–479
- [65] MESTER, R. ; HÖTTER, M. ; PÖCHMÜLLER, W.: Umwelterfassung mit bewegten Kameras. In: *Proc. in Artificial Intelligence*, 1996, S. 117–126
- [66] MEZGER, U.: *Motion Stereo and Passive Navigation for Autonomous Mobile Robots*, University of Western Australia, Perth, Australia, Diplomarbeit, Dezember 2000

- [67] MORI, H. ; CHARKAI, N.: Shadow and rhythm as sign patterns of obstacle detection. In: *International Symposium on industrial electronics*, 1993, S. 271–277
- [68] PARODI, P. ; PICCIOLI, G.: A feature-based recognition scheme for traffic scenes. In: *Proc. Intelligent Vehicles Symposium*, 1995, S. 229–234
- [69] PEREZ, P. ; HUE, C. ; VERMAAK, J. ; GANGNET, M.: Color-Based Probabilistic Tracking. In: *7th European Conference on Computer Vision* Bd. 1, 2002, S. 661–675
- [70] RABE, C.: *Detektion von Hindernissen vor Fahrzeugen durch Bewegungsanalyse*, Fachhochschule Würzburg-Schweinfurt - Informatik und Wirtschaftsinformatik, Diplomarbeit, 2005
- [71] RAHIMI, A. ; MORENCY, L.P. ; DARRELL, T.: Reducing Drift in Parametric Motion Tracking. In: *ICCV*, 2001, S. 315–322
- [72] SAKAI, K. ; FINKEL, L.H.: Spatial-frequency analysis in the perception of perspective depth. In: *Computation in Neural Systems* Bd. 8, 1997, S. 335–352
- [73] SANEYOSHI, K.: Drive Assist System Using Stereo Image Recognition. In: *Proc. IEEE Conference on Intelligent Vehicles*, 1996, S. 230–235
- [74] SCHWARZINGER, M. ; ZIELKE, T. ; NOLL, D. ; BRAUCKMANN, M.E. ; SEELEN, W.V.: Vision-Based Car-Following: Detection, Tracking and Identification. In: *Proc. IEEE Conference on Intelligent Vehicles*, 1992, S. 24–29
- [75] SHI, J. ; TOMASI, C.: Good Features to track. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, S. 592–600
- [76] SICK: *Homepage*. November 2005. – URL www.sick.com
- [77] SRINIVASA, N.: A Vision-Based Vehicle Detection and Tracking Method for Forward Collision Warning in Automobiles. In: *IEEE Intelligent vehicle symposium*, 2002, S. 626–631
- [78] STEIN, F.: Efficient Computation of Optical Flow Using the Census Transform. In: *DAGM04*, 2004, S. 79–86
- [79] SUN, Z. ; BEBIS, G. ; MILLER, R.: On-Road Vehicle Detection Using Optical Sensors: A review. In: *IEEE International Conference on Intelligent Transportation Systems* Bd. 6, 2004, S. 125 – 137
- [80] TELEKOLLEG: *Die Bochumer Medizin, Homepage*. Januar 2006. – URL <http://www.diebochumermedizin.de/bilder/projekte%20sw%2001.jpg>

- [81] THORPE, C. ; CARLSON, J. ; DUGGINS, D. ; GOWDY, J. ; MACLACHLAN, R. ; MERTZ, C. ; SUPPE, A. ; WANG, B.: Safe Robot Driving in Cluttered Environments. In: *Proceedings of the 11th International Symposium of Robotics Research*, Oktober 2003
- [82] TOMASI, C. ; KANADE, T.: Detection and Tracking of Point Features. 1991. – Forschungsbericht. Carnegie Mellon University Technical Report CMU-CS-91-132
- [83] VERRI, A. ; POGGIO, T.: Against quantitative optical flow. In: *Proceedings of the 1st International Conference on Computer*, Juni 1987, S. 171–180
- [84] VOLMER, C.: *Kalmanfilter-basierte 3D-Schätzung aus monokularen Bildfolgen*, Technische Universität Ilmenau, Fakultät für Elektrotechnik und Informationstechnik, Diplomarbeit, 2005
- [85] WALTON, M.: Robots fail to complete grand challenge. In: *CNN News* (2004), März. – URL <http://www.cnn.com/2004/TECH/ptech/03/14/darpa.race/>
- [86] WEBER, M. ; WELLING, M. ; PERONA, P.: Unsupervised learning of models for recognition. In: *European Conference on Computer Vision*, 2000, S. 18–32
- [87] WEDEL, A.: Optimierung von SIFT Features zum Wiederfinden von Hausfassaden. Institut für Photogrammetrie, 2004. – Forschungsbericht. – URL www.ipb.uni-bonn.de/FotoNav/docs/pdf/Wedel-Optimierung.pdf. Rheinische Friedrich-Wilhelms-Universität Bonn
- [88] WIESBECK, W.: *Vorlesungsunterlagen zur Vorlesung Radar System Engineering*. 2004. – Institut für Höchsthfrequenz und Elektronik, Universität Karlsruhe
- [89] WU, B. ; OOI, T.L. ; HE, Z.J.: Perceiving distance accurately by a directional process of integrating ground information. In: *Letters to Nature* 428 (2004), S. 73–77
- [90] WÜRZ-WESSEL, A. ; STEIN, F.J.: Calibration of a Free-Form Surface Mirror in a Stereo Vision System. In: *Proc. IEEE Conference on Intelligent Vehicles*, 2002, S. 471 – 476
- [91] YANG, M. ; YU, Q. ; WANG, H. ; ZHANG, B.: Vision-based Real-time Obstacles Detection and Tracking for Autonomous Vehicle Guidance. In: *Proceedings of SPIE Photonics West 2002 - Electronic Imaging 2002*, 2002, S. 65–74
- [92] YANG, Q. ; STEELE, M. ; NISTÉR, D. ; JAYNES, C.: Learning the Probability of Correspondences without Ground Truth. In: *International Conference on Computer Vision*, 2005, S. 1140–1147
- [93] ZIELKE, T. ; STORJOHANN, K. ; MALLOT, H.A. ; SEELEN, W.v.: Adapting computer vision systems to the visual environment: topographic mapping. In: *Proc. European Conference Computer Vision*, 1990, S. 613–615

Erklärung zur Diplomarbeit nach §19.7 Diplomprüfungsordnung

Hiermit erkläre ich, dass ich diese Arbeit selbständig durchgeführt und abgefasst habe.
Quellen, Literatur und Hilfsmittel, die von mir benutzt wurden, sind als solche
gekennzeichnet.

Bonn, den _____

Andreas Wedel