

**Technische Universität
München**

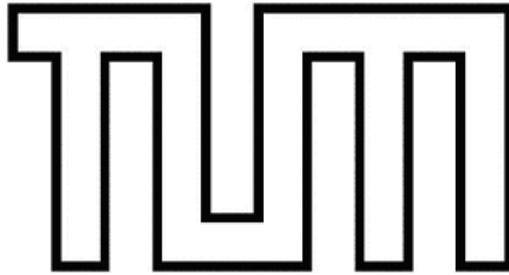
Fakultät für Informatik

Forschungs- und Lehrereinheit Informatik IX

**Fusion von Infrarotvideo- und Laserscanner-Daten
zur Fußgängererkennung**

Diplomarbeit

Stephan Thoma



**Technische Universität
München**

Fakultät für Informatik

Forschungs- und Lehrereinheit Informatik IX

**Fusion von Infrarotvideo- und Laserscanner-Daten
zur Fußgängererkennung**

Diplomarbeit

Stephan Thoma

Themensteller: Prof. Michael Beetz, PhD

Betreuer: Günter Bauer, Dipl.-Inf.
BMW Forschung und Technik GmbH

Abgabetermin: 15.12.2005

SELBSTSTÄNDIGKEITSERKLÄRUNG & GEHEIMHALTUNGSVERMERK

Erklärung

Hiermit versichere ich, dass ich diese Diplomarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Stephan Thoma
München, 15. Dezember 2005

Geheimhaltungsvermerk

Die vorliegende Diplomarbeit beinhaltet interne vertrauliche Informationen der BMW Group. Es dürfen keinerlei Kopien oder Abschriften - auch in digitaler Form - gefertigt werden. Die Diplomarbeit ist nur Korrektoren sowie Mitgliedern des Prüfungsausschusses zugänglich zu machen.

DANKSAGUNG

Dieser Abschnitt ist den Personen gewidmet, die zum Erfolg dieser Arbeit beigetragen haben.

Herrn Prof. Michael Beetz möchte ich für die Betreuung der Arbeit an der Fakultät für Informatik der TU München danken. Sein Erfahrungsschatz und seine Anregungen waren eine große Hilfe.

Besonderer Dank geht auch an Herrn Günter Bauer für die ausdauernde Betreuung der Arbeit bei der BMW Forschung und Technik GmbH. Seine Kenntnisse im Bereich der Bildverarbeitung waren von großem Nutzen und sein Optimismus sorgte immer für Aufmunterung.

Dank geht auch an Herrn Dr. Dirk Wisselmann für die Möglichkeit diese Diplomarbeit in seiner Abteilung zu schreiben und an Herrn Dr. Klaus Gresser für die Unterstützung des Themas innerhalb seiner Gruppe.

Bei Antje Neve möchte ich mich für die äußerst gute Zusammenarbeit vor allem bei der Konstruktion der Kalibriervorrichtung bedanken. In diesem Zusammenhang auch ein großes Lob und ein herzliches Dankeschön an die überaus kompetenten Werkstatt-Mitarbeiter bei der BMW Forschung und Technik GmbH. Erst durch sie war eine erfolgreiche Umsetzung der Ideen überhaupt möglich.

Daniel Schwarz und Alexandra Vogt danke ich für die uneingeschränkte Unterstützung bei allen (technischen) Herausforderungen, die während der Versuchsfahrten zu meistern waren.

Kein (Ferninfrarot-)Film kommt ohne Statisten aus. Bei Alexander Wulff, Alexandra Vogt, Daniel Schwarz und Robin Donner möchte ich mich deshalb für die bedingungslose Befolgung meiner - nicht immer nachvollziehbaren - Regieanweisungen bedanken. Die Ergebnisse aus den Versuchen brachten wichtige Erkenntnisse für den Einsatz unter realen Bedingungen.

Für das intensive Korrekturlesen und für die Übersetzung meiner Informatiker-Formulierungen ins Deutsche danke ich Sandra und Claudia Schiml sowie Wolfgang Mühlbauer.

Leonhard Walchshaeusl, Rudi Lindl und Daniel Niehues danke ich für die bereichernden Diskussionen zu fachlichen Themen und die Hilfe bei Problemen.

KURZFASSUNG

Diese Arbeit beschäftigt sich mit dem Erkennen und dem Verfolgen von Fußgängern im Fahrzeugumfeld. Um für aktive Fahrerassistenzsysteme genaue Aussagen über die Position und die Bewegungsrichtung von Fußgängern treffen zu können, wird eine Kombination aus einer Ferninfrarotkamera und einem Rotations-Laserscanner verwendet.

In der Arbeit werden eigene Verfahren zur effizienten Segmentierung der Punktwolken des Laserscanners entwickelt. Es wird das bekannte Multi-Hypothesen-Tracking-Verfahren eingesetzt, um auch bei Fußgängergruppen und Störeinflüssen eine gute Ermittlung der Objekt-Trajektorien zu gewährleisten. Die schnelle Fouriertransformation wird für die Untersuchung der Bein-Pendelbewegungen in den Messergebnissen des Laserscanners verwendet. Für die Analyse der Ferninfrarotbilder nutzt diese Arbeit vorhandene Erkennungsmethoden. Nach einer Zusammenführung der Sensordaten auf Merkmalsebene (Fusion) findet die Klassifikation mittels Bayes'scher Netze sowie Entscheidungsbäumen statt.

Es wird eine Kalibriervorrichtung entwickelt, welche im Infrarotbild ein scharfkantiges Schachbrettmuster erzeugt. Damit lassen sich mittels bekannter Kalibrierverfahren für normale Videokameras die intrinsischen Kameraparameter der FIR-Kamera genau ermitteln.

Die implementierten Verfahren werden anhand von Referenzszenen auf ihre Tauglichkeit im Straßenverkehr untersucht. Insbesondere ist von Interesse, in welchem Abstand zum Fahrzeug die verwendeten Verfahren noch zuverlässig arbeiten.

ABSTRACT

This work deals with the detection and tracking of pedestrians from a moving vehicle, especially cars. To estimate the position and moving direction of objects as precise as possible, a laser scanner device is used in combination with a far infrared camera.

We develop methods for efficient segmentation of the 3-dimensional point cloud and optimize them for the given hardware. The well-known multiple hypothesis tracking algorithm is used to ensure optimal object trajectory estimation even in complex scenarios like groups of pedestrians. With Fast Fourier Transform we analyze and detect the pendular movement of pedestrians' legs. Existing methods for pedestrian detection within far infrared images are integrated. After a feature level fusion of the sensor information, classification is performed using Bayesian networks and decision trees.

This paper describes the construction and use of a calibration device which produces a sharp-edged chess board pattern in the infrared image. Thus well-known methods for standard video cameras can be applied in order to determine the intrinsic parameters of the far infrared camera.

The implemented techniques are then evaluated with regard to their practical use in road traffic scenarios, by analyzing recorded data from an experimental vehicle. Emphasis is placed on investigating at which distance from the vehicle the methods still work properly.

INHALTSVERZEICHNIS

1	Einleitung	3
1.1	Motivation	3
1.2	Beiträge dieser Arbeit	7
1.3	Gliederung der Arbeit	9
2	Grundlagen	11
2.1	Verarbeitungskette	12
2.2	Aufbau des Versuchsfahrzeugs	14
2.3	Funktionsweise des Laserscanners	14
2.4	Vorhandene Erkennungsmethoden im FIR-Video	17
2.4.1	Ansatz mit Support Vector Machines	17
2.4.2	Ansatz mit Adaboost und Haarfeatures	18
2.5	Kalibrierung des Systems	18
2.5.1	Intrinsische Parameter der FIR-Kamera	19
2.5.2	Extrinsische Parameter des Versuchsträgers	21
2.6	Koordinatensysteme und Umsetzung als Szenen-Graph	22
2.7	Eigenbewegung des Fahrzeugs	24
3	Verarbeitung der LIDAR-Daten	27
3.1	Segmentierung	28
3.1.1	Segmentierung durch Tiefensuche	28
3.1.2	Erkennen von Schattenwurf	31
3.2	Tracking	31
3.2.1	Parameter der Kalman-Filterung	33
3.2.2	Multi-Hypothesen-Ansatz	37
3.3	Analyse der Bein-Pendelbewegungen	41
3.3.1	Extraktion der relevanten Merkmale	41
3.3.2	Fourier-Analyse zur Frequenzbestimmung	43

4	Klassifikation und Fusion	47
4.1	Grundlegende Schritte	47
4.2	Zusammenspiel mit der Bilderkennung	48
4.2.1	Ermittlung der interessanten Bildbereiche	48
4.2.2	Anpassung des Adaboost-Klassifikators	53
4.3	Klassifikation	54
4.3.1	Extraktion der Features	55
4.3.2	Klassifikation mit Bayes'schem Netz	55
4.3.3	Klassifikation durch Entscheidungsbäume	57
5	Implementierung	61
5.1	Architektur des Systems	61
5.1.1	Visualisierung	61
5.1.2	Fusions-Modul und Controller	64
5.1.3	Datenverwaltung und -aufzeichnung	66
5.2	Vorhandene Bibliotheken	66
6	Bewertung und Analyse der Ergebnisse	69
6.1	Beurteilung der Verfahren bezüglich Abstand zum Fahrzeug	70
6.2	Betrachtung der Rechenzeiten	72
6.2.1	Multi-Hypothesen-Tracker	73
6.2.2	FIR-Bildverarbeitung	74
6.2.3	Gesamte Verarbeitungsdauer	77
6.3	Beurteilung der Erkennungsleistung	77
6.3.1	Untersuchung ohne Bildverarbeitung	78
6.3.2	Untersuchung mit Bildverarbeitung	79
7	Zusammenfassung und Ausblick	81
	Glossar	83
	Abbildungsverzeichnis	85
	Tabellenverzeichnis	87
	Literaturverzeichnis	89
A	Unfallstatistiken	93
B	UML-Diagramm	95
C	Klassifikationsergebnisse	97
D	Matlab Code	101

EINLEITUNG

1.1 Motivation

Automobilhersteller unternehmen große Anstrengungen, um die Gefahren für Menschen im Straßenverkehr zu minimieren. Beginnend mit der Einführung des Airbags, wurde der Insassenschutz stetig verbessert. Darüber hinaus sind mittlerweile auch Systeme verfügbar, welche kurz vor einem unvermeidbaren Zusammenstoß sicherheitsrelevante Maßnahmen ergreifen um das Verletzungsrisiko weiter zu senken (Stichwort „Precrash“). Beispielsweise lassen sich die Rückhaltesysteme im Fahrzeuginnenraum bereits vor einem Zusammenstoß konditionieren. Doch auch dem Schutz von Verkehrsteilnehmern außerhalb des Fahrzeugs wird immer größere Bedeutung beigemessen. Fußgänger sind besonders gefährdet, da ihnen die lebensrettende Knautschzone der Fahrzeugkarosserie fehlt. Der Frontbereich des Fahrzeugs wird deshalb umgestaltet, damit eine Kollision mit einem Fußgänger weniger schwere Verletzungen hervorruft. Dies findet mittlerweile auch Einzug in die europäische Gesetzgebung [2].

All diese Maßnahmen sind jedoch überwiegend reaktiver Natur und sollen in erster Linie die Unfallfolgen mindern. Zur Unfall*vermeidung* tragen sie nicht bei. Die verwendete Sensorik ist deshalb meist auf sehr kurze Distanzen ausgelegt und für die Reaktion auf unvermeidbare und kurz bevorstehende Ereignisse optimiert. Beispielsweise werden empfindliche Drucksensoren in den Stoßfängern eingesetzt, welche erst beim Kontakt mit einem Fußgänger eine Aktion auslösen.

Viel sinnvoller wäre jedoch eine vorausschauende Vorgehensweise, um Unfälle, wann immer möglich, im Ansatz zu vermeiden. Dazu ist eine Sensorik mit höherer Reichweite notwendig, welche für das Verkehrsgeschehen relevante Informationen aus der Fahrzeugumgebung aufnimmt. Die Interpretation der so gewonnenen Messdaten kann in manchen Fällen dem Fahrer zugemutet werden, wie dies bereits heute bei der Einblendung eines Ferninfrarot-

(FIR) oder Nahinfrarot- (NIR) Videos während der Fahrt geschieht¹. Dadurch verbessert sich die Reaktion des Fahrers in Situationen, in denen die eigene Wahrnehmung zu schlecht ist (Dunkelheit, Regen, etc.). Viele Unfälle entstehen jedoch durch eine Überforderung bei der Verarbeitung der auf den Fahrer zukommenden Reize. Deshalb sollte die Fahrzeugelektronik möglichst große Teile der Informationsverarbeitung übernehmen. Das Ergebnis dieser Verarbeitung wird dann in geeigneter Weise präsentiert.

Die BMW Forschung und Technik GmbH, eine Tochtergesellschaft der BMW Group, untersucht aus diesem Grund eine Vielzahl von Fahrerassistenzsystemen, die teils reine Warnsysteme darstellen, aber auch aktiv in das Fahrverhalten des Fahrzeugs eingreifen. Vor allem bei einem aktiven Eingriff ist eine absolut sichere Entscheidungsgrundlage unbedingt erforderlich. Ein Fahrerassistenz-System, welches die Vermeidung von Unfällen mit Fußgängern zum Ziel hat, muss also auf eine zuverlässige Sensorik und auf eine robuste Klassifikation der Objekte im Fahrzeugumfeld zurückgreifen können. Eine schnelle Reaktion ist natürlich ebenfalls wichtig. Dennoch kann, falls eine Vermeidung des Unfalls nicht mehr möglich ist, die Schwere der Verletzungen deutlich gesenkt werden, zum Beispiel, indem die Aufprallgeschwindigkeit reduziert wird. Aus Abbildung A.1 geht hervor, dass, bei einer Halbierung der Aufprallgeschwindigkeit von 50 auf 25 km/h, die Anzahl der schweren Verletzungen bereits um 85 Prozent sinkt.

Da FIR-Kameras Wärmeunterschiede sichtbar machen, sind sie geradezu prädestiniert um Fußgänger in der Umgebung - auch bei völliger Dunkelheit - zu erkennen. In vielen Situationen unterscheiden sich Fußgänger durch eine höhere Wärmeabstrahlung von der restlichen Szene und können so gut segmentiert werden [30]. Eine Herausforderung stellt jedoch eine komplexe Umgebung dar, wie sie im innerstädtischen Verkehr oft vorkommt. Unter Umständen ist dann keine sichere Unterscheidung von anderen wärmeabstrahlenden Objekten mehr möglich (z. B. warme Teile an Fahrzeugen). Ein möglicher Ausweg ist die kombinierte Verarbeitung mit einer normalen Videokamera, wie in [19] gezeigt. Bei Tageslicht ist dann eine bessere Unterscheidung anhand von Farbmerkmalen möglich.

Das eigentliche Ziel geht jedoch über das reine Erkennen der Fußgänger im Infrarotbild hinaus. Letztendlich ist eine möglichst präzise Abschätzung der Position und der Bewegungsrichtung eines Fußgängers notwendig, um zu entscheiden, ob die Gefahr einer Kollision besteht. Bei Ein-Kamera Systemen ist eine ausreichend genaue Entfernungsbestimmung kaum möglich und wird in diesem Fall durch den niedrigen Verbauort der Kamera im Fahrzeug zusätzlich erschwert [30]. Stereo-Systeme stellen nicht in jeder Situation eine Alternative dar, da eine genaue Positionsbestimmung nur bei gut texturierten Szenen möglich ist.

Aufgrund der vorherigen Ausführungen liegt es nahe, für die Ermittlung der Position eines Fußgängers eine spezielle Sensorik zu verwenden. In dieser Arbeit kommt ein für den Einsatz in Automobilen konzipierter Laserscanner zum Einsatz. Die Kombination aus einem Laserscanner und einer FIR-Kamera, welche beide den Gefahrenbereich vor dem Fahrzeug abdecken, verspricht gute Ergebnisse für die Fußgängererkennung, da die Vorteile bei-

¹http://www.bmw.de/bmw_aktuell/innovation/nightvision2005.html

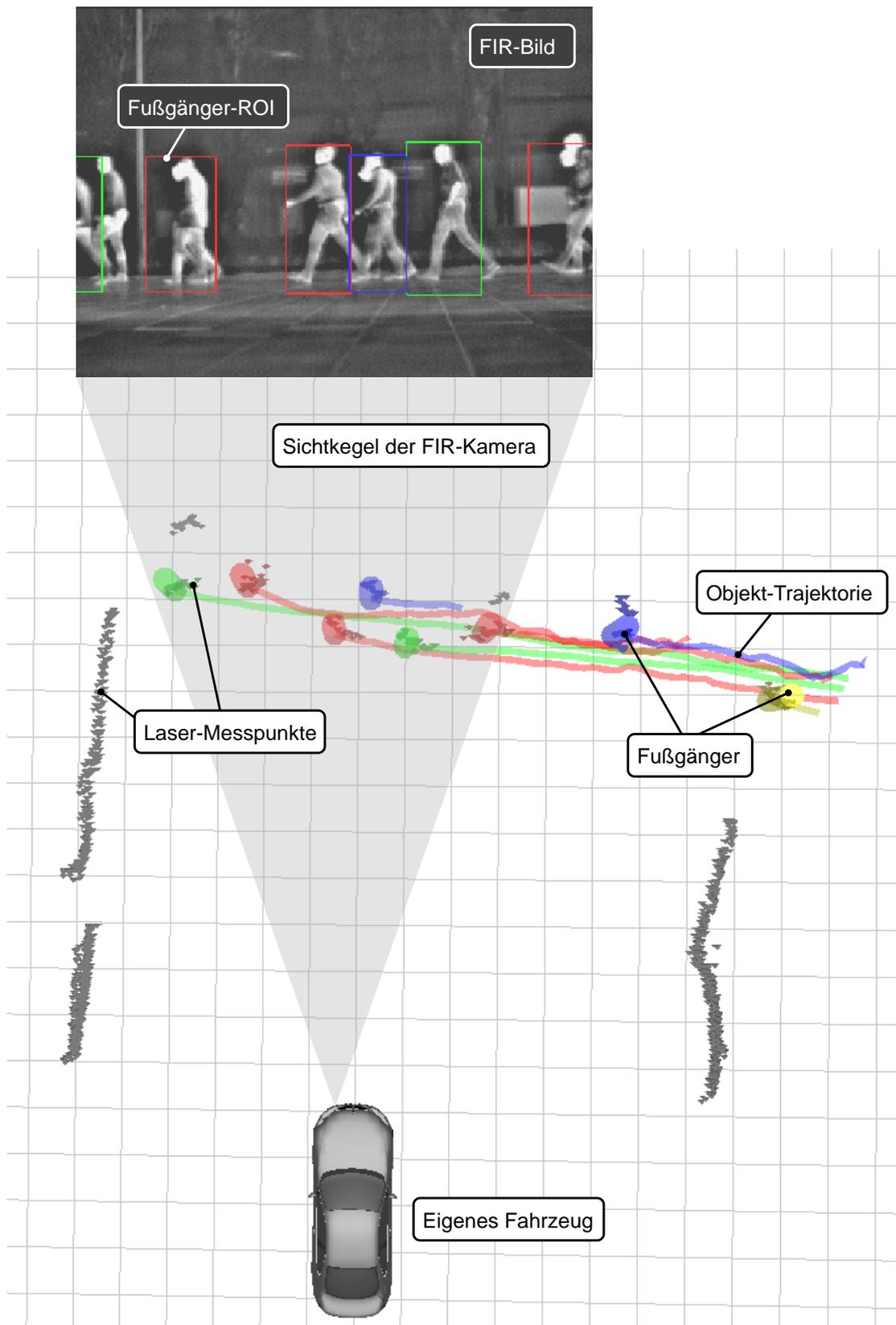


Abbildung 1.1: Beispielszene mit mehreren gleichzeitig verfolgten Fußgängern vor dem Fahrzeug

der Messprinzipien vereint werden. Diese Arbeit untersucht deshalb das Synergiepotential bei der Kombination dieser beiden Sensoren (Fusion). Dabei liegt der Schwerpunkt auf innerstädtischen Verkehrssituationen. Dort ist die Umgebungskomplexität meist hoch und die Eigengeschwindigkeit des Fahrzeugs liegt im Normalfall bei etwa 50-60 km/h.

Abbildung 1.1 zeigt exemplarisch eine typische Szene. Im Infrarotbild ist eine Fußgängergruppe zu sehen, welche die Fahrbahn von rechts nach links quert. Die Fußgänger heben sich durch eine höhere Wärmeabstrahlung von der Umgebung ab und erscheinen deshalb im FIR-Bild heller. Im unteren Teil der Abbildung sind die Messpunkte des Laserscanners, der Sichtkegel der FIR-Kamera und das Fahrzeug zu sehen. Ein Gitternetz mit einem Meter Kantenlänge soll den Maßstab verdeutlichen.

Zunächst ist es nötig, die Messpunkte des Laserscanners geeignet zu gruppieren. Dies ist in der Abbildung durch eine unterschiedliche Farbgebung dargestellt. Im Idealfall entsprechen diese Punktgruppen (Segmente) bereits den realen Objekten, die sich im Bereich des Laserscanners befinden. Die Schwierigkeit besteht darin, geeignete Verfahren zu entwickeln, welche die gesamte Menge an Messpunkten in sinnvolle Segmente zerlegen und dabei auch mit veräuschten Sensorinformationen umgehen können. Verfolgt man die Position dieser Segmente über einen längeren Zeitraum hinweg („Tracking“), so lassen sich Bewegungsrichtung und -geschwindigkeit von Objekten ermitteln. Die Abbildung zeigt die Trajektorien für die vor dem Fahrzeug querenden Fußgänger. Gerade bei Fußgängergruppen, wie im gezeigten Fall, besteht die Herausforderung darin, eine geeignete Zuordnung von Segmenten zu Objekten vorzunehmen. Es ist sicherzustellen, dass die ermittelten Segmente den bereits erkannten Objekten korrekt zugeordnet werden und falls nötig, neue Objekte erzeugt werden. Dazu ist auch eine Vorhersage der zukünftigen Objektposition erforderlich. So lässt sich eingrenzen, an welcher Stelle im Laserscan das bereits erkannte Objekt im nächsten Zeitschritt erwartet wird.

Um die Vorteile der FIR-Kamera in die Auswertung zu integrieren, wird jedem Objekt im Laserscan ein Bildbereich im FIR-Bild zugeordnet. Dies ist in der Abbildung durch farbige Rechtecke im FIR-Bild dargestellt („Regions of Interest“, ROIs). Diese Bereiche werden von der FIR-Bildverarbeitung dahingehend untersucht, ob es sich um einen Fußgänger handelt. Aufgrund des eingeschränkten Sichtbereichs der Kamera sind nicht alle verfolgten Fußgänger im FIR-Bild sichtbar. Der Laserscanner hat einen Sichtbereich von nahezu 180° vor dem Fahrzeug und die Kamera lediglich 36° .

Die Literatur [9, 15] unterscheidet die folgenden grundsätzlichen Möglichkeiten für eine Sensorfusion, welche auch in Abbildung 1.2 veranschaulicht sind:

- **Sensorfusion auf niedriger Ebene (Raw Level)**
Messungen mehrerer Sensoren werden ohne Vorverarbeitung direkt kombiniert.
- **Sensorfusion auf mittlerer Ebene (Feature Level)**
Jeder Sensor liefert Objekteigenschaften, z. B. Länge, Breite oder Geschwindigkeit. Die Klassifikation erfolgt auf der Kombination aller Merkmale.

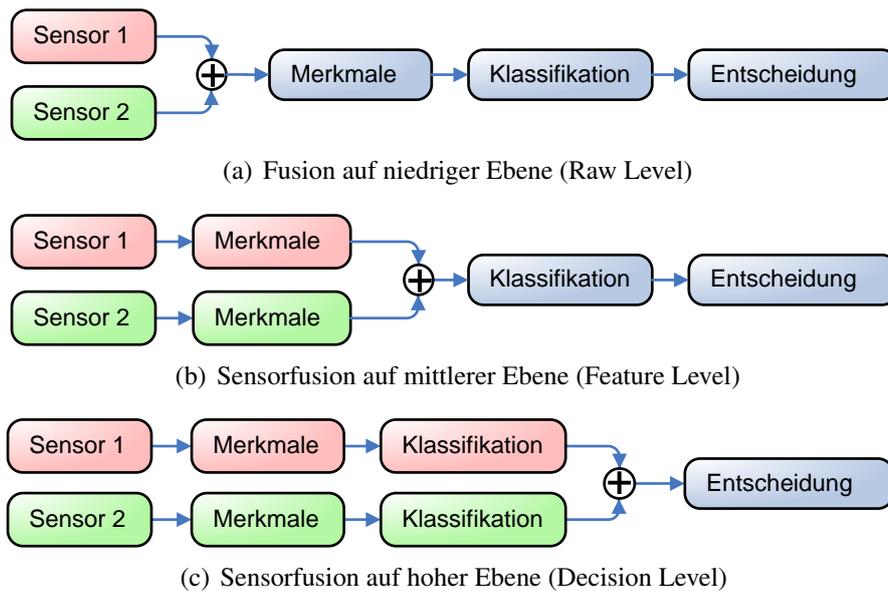


Abbildung 1.2: Verschiedene Ansätze für die Sensorfusion

- **Sensorfusion auf hoher Ebene (Decision Level)**

Alle Sensoren führen eine vollständige Klassifikation durch. Die Kombination der Ergebnisse erfolgt beispielsweise durch Voting-Verfahren.

Da in dieser Arbeit zwei Sensoren mit unterschiedlichem Messprinzip (Kamera und Laserscanner) verwendet werden, ist eine Fusion auf niedrigerer Ebene weniger sinnvoll und wird nicht untersucht. Realisiert wird eine Kombination aus der zweiten und dritten Variante. Die Bilderkennung führt zwar die Klassifikation auf bestimmten Bildausschnitten vollständig durch, das Ergebnis wird jedoch bei der endgültigen Klassifikation nur als weiteres Kriterium, zusammen mit anderen Eigenschaften aus dem Laserscan, verarbeitet.

1.2 Beiträge dieser Arbeit

Abbildung 1.3 zeigt eine schematische Darstellung des Datenflusses in dieser Arbeit. Alle blau hinterlegten Rechtecke stehen für Komponenten, die im Rahmen dieser Arbeit implementiert werden. Eingangsgrößen sind zunächst die Daten der beiden Sensoren: Zum einen sind das die Videobilder einer FIR-Kamera und zum anderen die einzelnen Messpunkte eines Laserscanners. Weitere Eingangsdaten sind Messgrößen, wie z. B. Gierrate und Geschwindigkeit, welche unter dem Sammelbegriff „Fahr-dynamikdaten“ zusammengefasst sind. Diese Informationen sind nötig, um die Eigenbewegung des Fahrzeugs zu kompensieren. Nur so lassen sich absolute Geschwindigkeiten der Objekte in der Umgebung ermitteln. Dies ist wichtig, da gerade die Geschwindigkeit, mit der sich ein Objekt bewegt, Rückschlüsse auf die Art des Objekts zulässt (z. B. Fahrradfahrer, Fußgänger, Autos, ...).

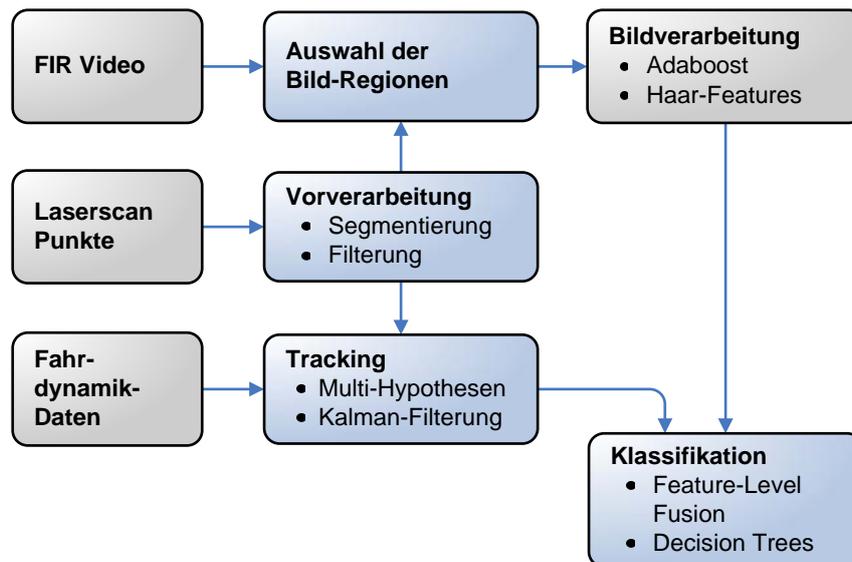


Abbildung 1.3: Grundsätzlicher Aufbau des Systems und Datenfluss

Ziel ist es nun, die bereits vorhandene Algorithmik [19, 30] zur Erkennung von Fußgängern in FIR-Bildern zu verwenden, und gleichzeitig die Vorteile eines zusätzlichen Laserscanners mit in die endgültige Klassifikation einfließen zu lassen.

Diese Arbeit beschäftigt sich ausführlich mit der Verarbeitung des Laserscans. Es werden Segmentierungsverfahren beschrieben, welche die Eigenschaften des Sensors und der Objekte berücksichtigen. Zudem ist eine besonders effiziente Implementierung notwendig, um eine Echtzeitanwendung zu ermöglichen. Das Tracking der Objekte findet aufgrund der gemessenen Laserdaten statt. Die Eigenbewegung des Fahrzeugs wird vorher kompensiert. Das eigentliche Tracking übernimmt das ursprünglich in [24] beschriebene und in vielen Anwendungen bewährte (z. B. [26, 29, 10]) Multi-Hypothesen-Verfahren. Als Basis dient das in [6] entwickelte Framework, welches im Rahmen dieser Arbeit an die Eigenschaften der zu verfolgenden Objekte angepasst wird. Um zusätzliche Evidenz für das Vorhandensein eines Fußgängers zu erhalten, wird in Anlehnung an [8] ein Verfahren implementiert, welches die Beinbewegungen mittels einer Fouriertransformation auf Periodizität überprüft.

Bei der Bilderkennung ist insbesondere eine Vorkonditionierung auf die bereits ermittelte Position eines potentiellen Fußgängers von Interesse. Die Bildverarbeitung beschränkt sich auf jene Bereiche, welche vom Laserscanner bereits als möglicher Fußgänger erkannt wurden. Unter Verwendung aller Merkmale entscheidet dann eine Klassifikationsstufe, ob es sich bei einem gefundenen Objekt um einen Fußgänger handelt. Zwei Ansätze werden verfolgt: Bayes'sche Netze und Entscheidungsbäume. Die Entscheidungsbäume werden nach dem in [22] beschriebenen ID3-Algorithmus trainiert und vereinfacht.

1.3 Gliederung der Arbeit

Kapitel 2 geht zu Beginn auf die Abläufe während der Fusion ein und beschäftigt sich dann mit dem Aufbau des Versuchsfahrzeugs und der Anordnung der Sensoren. Es werden technische Details, wie beispielsweise die Funktionsweise des verwendeten Laserscanners, erklärt. Ein weiterer Bereich ist die Kalibrierung des Systems. Es wird eine Kalibriervorrichtung für FIR-Kameras entwickelt. Mit ihr lassen sich die intrinsischen Kameraparameter (z. B. Brennweite) bestimmen. Die extrinsischen Parameter (Position der Sensoren am Fahrzeug) werden durch eine hochgenaue Vermessung auf einer Fahrzeug-Messplatte ermittelt. Es wird ein Überblick über die bereits bei der BMW Forschung und Technik GmbH verwendeten Bilderkennungsverfahren gegeben. Ein Modell des Fahrzeugs, der Sensoren und der erkannten Objekte wird in Form eines Szenenbaums realisiert. Abschließend erfolgt die Darlegung der Methoden zur Bestimmung der Eigenbewegung des Fahrzeugs aus den Fahrdynamikdaten.

Die Verarbeitung des Laserscans wird in Kapitel 3 umfassend behandelt. Es wird detailliert auf das verwendete Segmentierungsverfahren sowie auf das verwendete Multi-Hypothesen-Tracking eingegangen. Es erfolgt die Parametrisierung der eingesetzten Kalman-Filter. Ein weiterer Teil ist die Analyse der für Fußgänger typischen Pendelbewegung der Beine mittels Fourier-Transformation über mehrere aufeinander folgende Laserscans hinweg.

Kapitel 4 geht dann auf die eigentliche Erkennung der Fußgänger ein. Nach der Kombination mit der Bilderkennung werden alle relevanten Merkmale (Features) extrahiert und dann mittels Entscheidungsbaum und Bayes'schem Netz einer Klasse zugeordnet. Ein weiterer Aspekt ist die Beschränkung der Bilderkennung auf die interessanten Bildbereiche, welche zuvor aus den Laserscanner-Daten ermittelt wurden.

Einige Details zur Architektur und Implementierung der entwickelten Software spricht Kapitel 5 an. Es wird auch auf die Lösungsansätze bei der synchronen Datenaufzeichnung und Wiedergabe eingegangen. Einen wichtigen Teil, bei immer komplexer werdenden Sensoren, stellt auch die Umsetzung einer geeigneten Visualisierung dar.

In Kapitel 6 findet eine Analyse der Ergebnisse und der Erkennungsleistung statt. Es werden auch Untersuchungen zur Tauglichkeit der einzelnen Verfahren bezüglich Fahrzeug-Fußgänger-Abstand durchgeführt.

Abschließend gibt das 7. Kapitel eine Zusammenfassung der erreichten Erkennung und zeigt mögliche Schwerpunkte künftiger Arbeiten auf diesem Gebiet auf.

GRUNDLAGEN

Dieses Kapitel behandelt alle grundlegenden Aspekte, die für das Verständnis der Arbeit von Bedeutung sind. Folgende Bereiche werden abgedeckt:

- **Der Ablauf der Fusion**

Es werden die grundlegenden Schritte des Verarbeitungsprozesses in einer schematischen Übersicht verdeutlicht. Alle folgenden Kapitel gehen dann näher auf die dargestellten Komponenten und Vorgänge während der Verarbeitung ein.

- **Der Aufbau des Versuchsfahrzeugs**

Hier wird kurz auf die Positionen der Kameras und des Laserscanners eingegangen. Ebenfalls dargestellt ist, welche Sensoren welchen Bereich abdecken.

- **Die Eigenschaften des Laserscanners**

Die Verfahren in Kapitel 3 setzen die Kenntnis der genauen Eigenschaften des Sensors voraus. Deshalb gibt Abschnitt 2.3 einen Einblick in die genauen Daten (z. B. die möglichen Winkelauflösungen und deren Vor- und Nachteile).

- **Vorhandene FIR-Bildverarbeitung**

Da bereits einige Arbeiten zum Thema Fußgängererkennung im FIR-Video existieren, werden zusammenfassend die wichtigsten Merkmale der verwendeten Methoden dargestellt.

- **Kalibrierung des Fahrzeugs**

Da die Verwendung mehrerer Sensoren auch eine gemeinsame Abstimmung erfordert, geht Abschnitt 2.5 genauer auf die durchgeführte Kalibrierung ein.

2.1 Verarbeitungskette

Abbildung 2.1 zeigt den schematischen Ablauf des gesamten Verarbeitungsprozesses, wie er in dieser Arbeit implementiert wird. Alle weiteren Kapitel beziehen sich auf diese Darstellung und gehen näher auf die einzelnen Komponenten und Vorgänge ein. Dabei beziehen sich die Nummern ❶ bis ❹ immer auf diese Abbildung.

Die folgende Liste stellt alle Schritte kurz dar und verweist auf die entsprechenden Kapitel in der Arbeit:

- ❶ Um für Objekte, die sich im Bereich des Fahrzeugs aufhalten, absolute Geschwindigkeiten ermitteln zu können, ist es notwendig die Bewegung des eigenen Fahrzeugs zu kompensieren. Abschnitt 2.7 stellt die Vorgehensweise zur Schätzung der Eigenbewegung dar.
- ❷ Der Laserscanner liefert zunächst eine Menge von Messpunkten im Raum, die durch die in Abschnitt 3.1 gezeigten Verfahren in kleinere Punktgruppen (Segmente) zerlegt werden. Es findet auch eine Berücksichtigung von Schattenwurf statt.
- ❸ Das Tracking übernimmt in dieser Arbeit ein Multi-Hypothesen-Tracker. Abschnitt 3.2 geht näher auf die dafür notwendigen Kalman-Filter und das eigentliche Tracking-Verfahren ein.
- ❹❺ Fußgänger lassen sich in vielen Fällen durch typische Bein-Pendelbewegungen von anderen Objekten unterscheiden. Abschnitt 3.3 widmet sich deshalb der Analyse der periodischen Bewegungen im Laserscan mittels Fourier-Transformation (FFT).
- ❻❼ Damit die eigentliche Fusion mit der Bildverarbeitung stattfinden kann, werden die im Laserscan erkannten Objekte in das Bild projiziert und somit interessante Bildbereiche herausgestellt. Da die Kamera und der Laserscanner nicht an der selben Position am Fahrzeug montiert sind, haben die beiden Sensoren einen leicht unterschiedlichen Sichtbereich. Deshalb ist es notwendig, die vorher ermittelten interessanten Bildbereiche nach der Entfernung zur FIR-Kamera zu sortieren. Die Art und Weise dieser Projektion und die Sortierung erklärt Abschnitt 4.2.1.
- ❽ Nachdem die interessanten Bildbereiche festgelegt sind, untersucht die Bildverarbeitung diese Bereiche auf das Vorhandensein eines Fußgängers. Die verwendeten Bildverarbeitungs-Verfahren sind in Abschnitt 2.4 dargestellt. Da eine direkte Anwendung auf die interessanten Bereiche nicht immer möglich ist, beschreibt Abschnitt 4.2.2 einige Optimierungen für die verwendeten Klassifikationsverfahren.
- ❾ Die endgültige Klassifikation stützt sich auf alle vorher ermittelten Merkmale der Objekte und entscheidet, ob es sich um einen Fußgänger handelt. Abschnitt 4.3 geht auf die verwendeten Klassifikatoren ein.

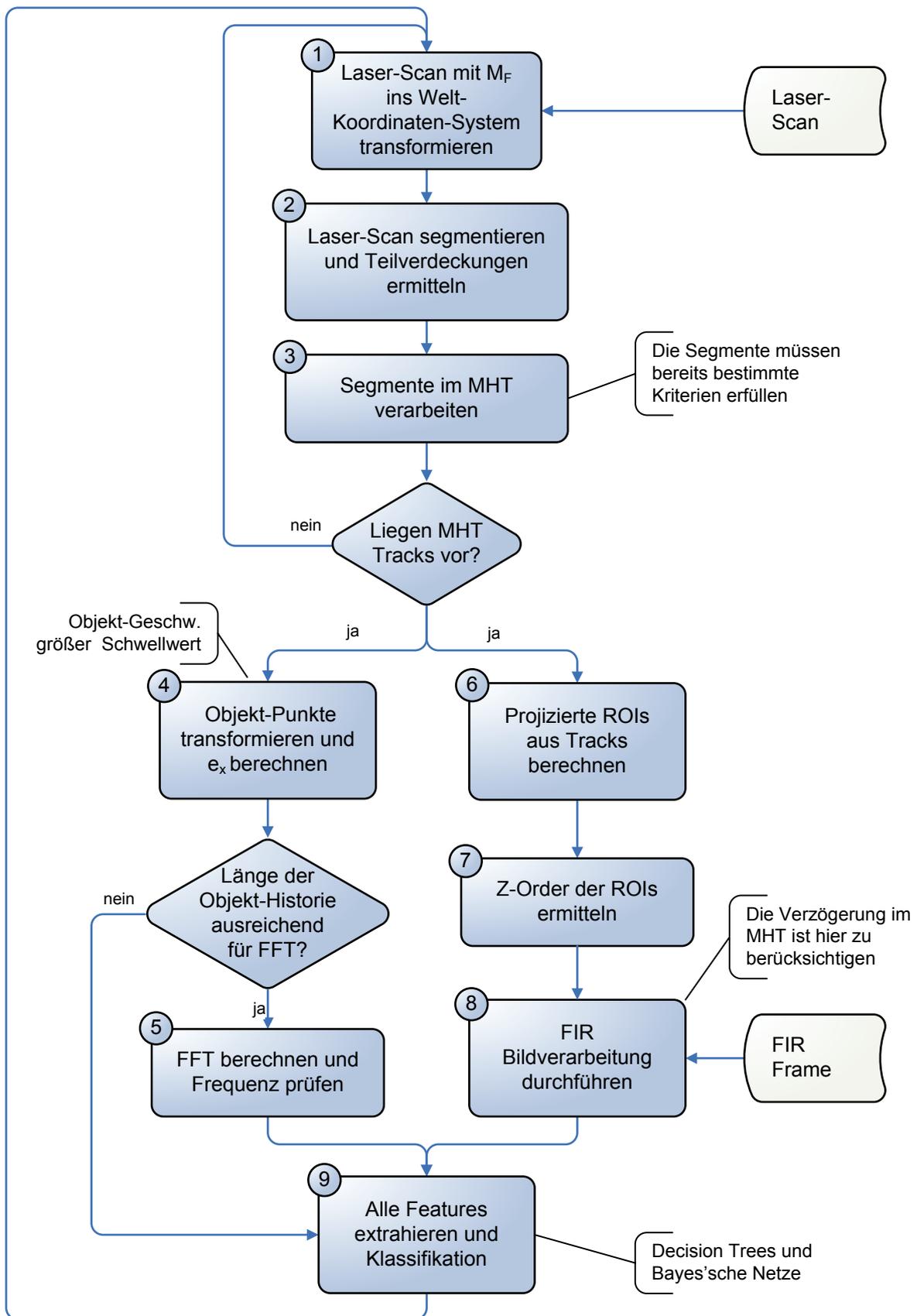


Abbildung 2.1: Der Ablauf des Fusions-Prozesses

2.2 Aufbau des Versuchsfahrzeugs

Während der gesamten Dauer der Arbeit stand ein BMW Versuchsfahrzeug, welches für verschiedene Fahrerassistenz- und Bilderkennungsprojekte eingesetzt wird, zur Verfügung. Abbildung 2.2 zeigt eine Vorderansicht des Fahrzeugs mit montiertem Laserscanner. Der Laserscanner wurde nicht ins Fahrzeuginnere integriert, um eine flexiblere Anpassung der Ausrichtung des Scanners zu ermöglichen.

FIR-Kamera und Laserscanner befinden sich etwa auf gleicher Höhe über dem Boden. Eine normale Farb-Videokamera hinter der Windschutzscheibe vereinfacht die spätere Interpretation der aufgenommenen Szenen am Rechner durch einen menschlichen Benutzer. Die Bildverarbeitung sollte diese Kamera laut Themenstellung jedoch nicht nutzen. Abbildung 2.3 auf Seite 16 zeigt eine schematische Ansicht des Versuchsträgers in einer Draufsicht und einer Seitenansicht. Dargestellt sind die Öffnungswinkel der Kameras und des Laserscanners.

2.3 Funktionsweise des Laserscanners

Für die Arbeit kam ein speziell für das automobiler Umfeld entwickelter Rotationslaserscanner der Firma IBEO Automobile Sensor zum Einsatz [34]. Der Scanner vom Typ „Alasca“ hat eine Reichweite von bis zu 80 Metern (auf Fahrzeughecks) und tastet die Umgebung in vier Ebenen ab. Abbildung 2.4 zeigt schematisch das Koordinatensystem des Scanners und die Scan-Ebenen. Die horizontale Winkelauflösung liegt zwischen $0,25^\circ$ und $1,0^\circ$. Die Messung des Abstands erfolgt über die Auswertung der Laufzeit des ausgesendeten Laserimpulses vom Scanner zum Objekt und zurück. Es werden dabei durch die Elektronik bis zu zwei Echos eines ausgesendeten Laser-Impulses erfasst. Die erreichbare horizontale Winkelauflösung hängt indirekt von der gewählten Scanfrequenz ab.

Tabelle 2.1 stellt den Zusammenhang zwischen Scanfrequenz und Winkelauflösung dar. Die dritte Spalte zeigt, in welchem Abstand ein Fußgänger (Annahme-Breite 0,5 m) bei optimalen Bedingungen noch mindestens 2 Messpunkte in einer Ebene zurückwirft. In der Realität ist dieser Wert noch um einiges nach unten zu korrigieren, da ein Fußgänger durch dunkle Kleidung oder einen ungünstigen Auftreffwinkel bereits bei weitaus kleineren Entfernungen wesentlich weniger Messpunkte liefert. Jedoch gibt bereits diese Gegenüberstellung einen Anhaltspunkt dafür, dass eine Scanfrequenz von 10 Hz für die Ziele in dieser Arbeit am günstigsten ist. Im Normalfall geht man bei innerstädtischem Verkehr von einer maximalen Fahrzeuggeschwindigkeit von 15 Metern pro Sekunde aus. Damit der Fahrer überhaupt noch reagieren kann, sollte eine Warnung mindestens eine Sekunde vor einem möglichen Zusammenstoß erfolgen. Rechnet man noch eine weitere Sekunde für die Erkennung und Klassifikation des Fußgängers mit ein, so müssen Objekte spätestens in einem Abstand von 30 Metern erkannt werden. Nur bei einer Scanfrequenz von 10 Hz lässt sich dieses Ziel erreichen.

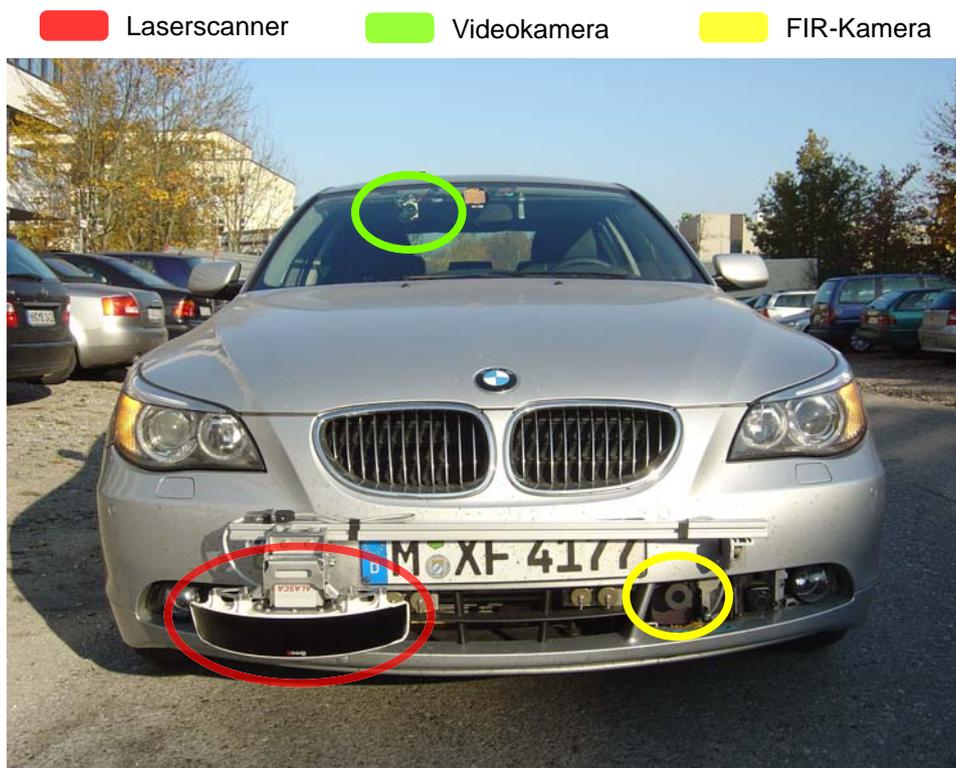


Abbildung 2.2: Frontansicht des Versuchsträgers

Fußgänger bewegen sich im Vergleich zu anderen Verkehrsteilnehmern mit einer relativ geringen absoluten Geschwindigkeit. Die Wegstrecke, die sie in 100 Millisekunden zurückzulegen im Stande sind, kann relativ gut eingegrenzt und beim Tracking berücksichtigt werden. Die hohe Winkelauflösung ist jedoch unabdingbar, da bei zu grobem Raster Objekte erst sehr spät vor dem Fahrzeug zu erkennen sind. Die Verfahren zur Analyse der Beinbewegungen in späteren Kapiteln sind ebenfalls auf ein gutes Auflösungsvermögen des Scanners angewiesen.

Wie aus Abbildung 2.4 hervorgeht, besitzt der Laserscanner einen vertikalen Öffnungswinkel von $3,2^\circ$, welcher jedoch zur Seite hin ($\pm 90^\circ$) auf Null zusammenschrumpft. Dies ist bedingt durch die Verwendung eines rotierenden Ablenkspiegels und einer fest verbauten Sende- und Empfangselektronik. Dieses „Kippen“ der Scan-Ebenen führt dazu, dass rechts und links vom Fahrzeug effektiv nur noch in einer Höhe über dem Boden gemessen wird. Damit besteht in diesem Bereich keine Möglichkeit einer Höhenabschätzung. Aber selbst in der Mitte vor dem Fahrzeug ergibt der Öffnungswinkel von $3,2^\circ$ bei 20 Metern Abstand lediglich eine vertikale Abdeckung von $h = \tan(3,2^\circ) \cdot 20 \text{ m} \approx 1,1 \text{ m}$. Die Höhe eines Segments¹ lässt folglich nur bedingt Rückschlüsse auf die Art des gemessenen Objekts zu.

Der Laserscanner verfügt über einen externen Vorverarbeitungsrechner, welcher die Auf-

¹Im Folgenden bezeichnet „Segment“ eine Menge von Messpunkten, die aufgrund ihrer Anordnung ein und demselben Objekt zugeordnet werden.

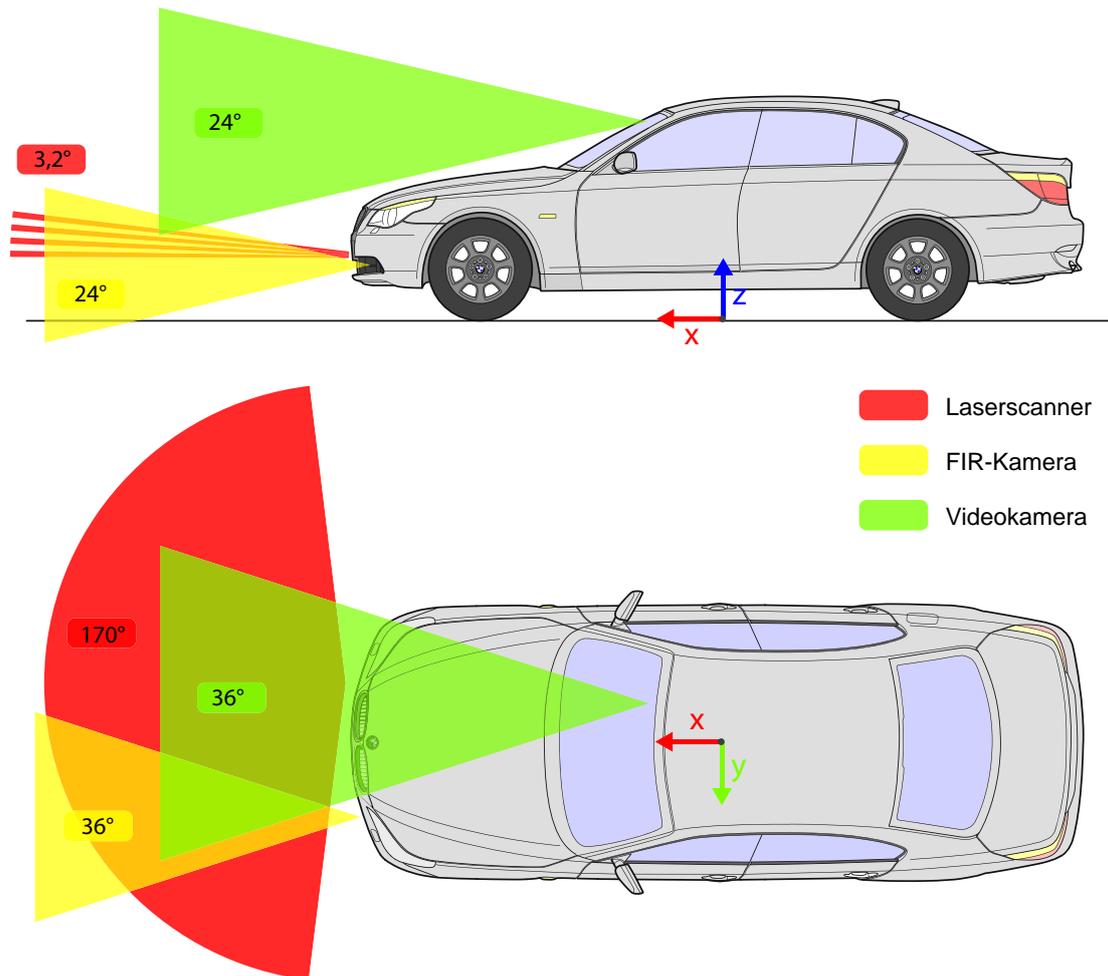


Abbildung 2.3: Schematische Darstellung des Versuchsträgers und der Sensorik

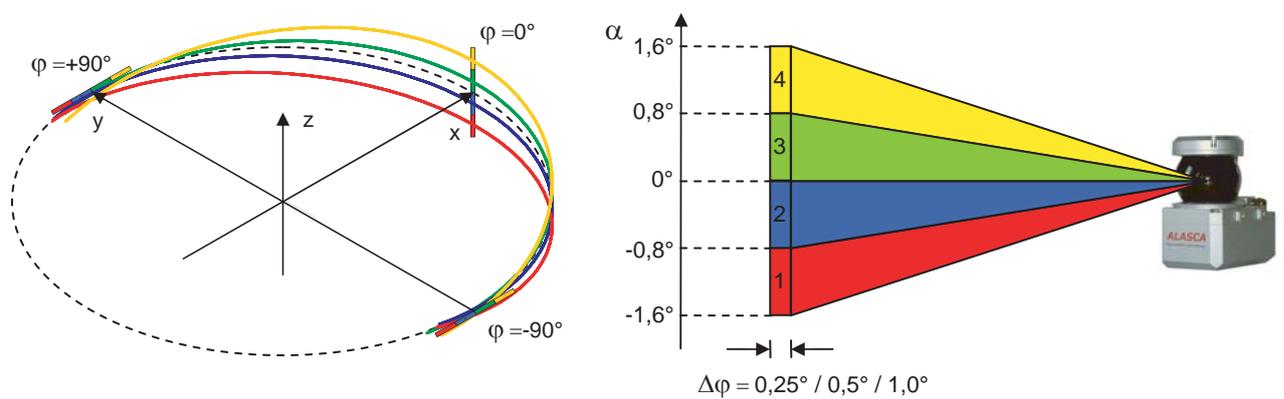


Abbildung 2.4: Darstellung der Scan-Ebenen und des Koordinatensystems, Quelle: IBEO AS

Rotations-Frequenz	Winkelauflösung	FG sichtbar bis Abstand
10 Hz	0,25°	$\frac{0,5 \text{ m}}{\tan(2 \cdot 0,25^\circ)} \approx 57 \text{ m}$
20 Hz	0,5°	$\frac{0,5 \text{ m}}{\tan(2 \cdot 0,5^\circ)} \approx 28 \text{ m}$
40 Hz	1,0°	$\frac{0,5 \text{ m}}{\tan(2 \cdot 1,0^\circ)} \approx 14 \text{ m}$

Tabelle 2.1: Zusammenhang zwischen Scanfrequenz und Winkelauflösung des Scanners

bereitung der Roh-Daten vornimmt und die gemessenen Entfernungen in ein kartesisches Koordinatensystem umrechnet. Die Ausgabe der Messdaten erfolgt dann über eine Ethernet-Schnittstelle.

2.4 Vorhandene Erkennungsmethoden im FIR-Video

Diese Arbeit stützt sich auf zwei vorhandene Methoden der Fußgängererkennung im FIR-Bild. Um in den folgenden Ausführungen einfacher Bezug nehmen zu können, werden durchgehend zwei Akronyme verwendet:

RubVision steht für den Ansatz von Aalzen Wiegersma in [33].

FirVision steht für den Ansatz von Antje Neve in [19].

Zunächst werden die Ansätze kurz dargestellt.

2.4.1 Ansatz mit Support Vector Machines

Die Arbeit [33] verwendet für die Segmentierung zwei verschiedene Ansätze:

- Das Intensitäts-Histogramm der Bilder wird in einen warmen und einen kalten Bereich aufgeteilt. Dadurch erhält man ein Binärbild, in dem die Fußgänger als warme Regionen erhalten bleiben. Die letzten 5% des Histogramms werden beispielsweise als „warm“ interpretiert. Diese Methode ist vor allem bei niedrigen Außentemperaturen gut geeignet.
- Auf dem Graustufen-Bild werden vertikale Gradienten mit einer Filtergröße von 5x5 Pixeln berechnet. Nach einer weiteren Filterung mittels Schwellwert liegen nur noch die vertikalen Elemente des Bildes als Vordergrund-Pixel vor, die dann zu größeren Regions of Interest (ROIs) gruppiert werden.

Die so gewonnenen ROIs durchlaufen dann einen Support-Vector-Machine-Klassifikator mit RBF-Kernel. Dieser wurde mit mehreren hundert Positiv- sowie Negativ-Beispielen trainiert. Der Klassifikator operiert auf vertikalen und horizontalen Haar-Features. Insgesamt liegt der

Schwerpunkt bei dem vorgestellten Verfahren auf Fußgängern, welche sich in einem relativ großen Abstand zum Fahrzeug befinden. Damit soll eine möglichst frühe Reaktion in wenig komplexen Szenen und bei hohen Eigengeschwindigkeiten erreicht werden (z. B. auf der Landstraße).

2.4.2 Ansatz mit Adaboost und Haarfeatures

Die Erkennungsmethode aus [19] basiert auf der Bildverarbeitungs-Bibliothek OpenCV. Der Klassifikator verwendet eine Kaskade aus verstärkten einfachen Klassifikatoren, die wiederum auf Haar-ähnlichen Merkmalen operieren (siehe [3]). Die Verstärkung der einfachen Klassifikatoren erfolgt über den in [7] vorgestellten Adaptive-Boosting-Algorithmus (kurz Adaboost).

Der Klassifikator wird zunächst mit Positiv- und Negativ-Beispielen trainiert. Dazu werden Bildregionen manuell ausgewählt („Labeling“). All diese Beispiele sind vor dem Training auf die gleiche Größe zu skalieren. Der Klassifikator kann jetzt verwendet werden, um für unbekannte Bildregionen eine Aussage zu liefern, ob diese ein Objekt ähnlich den trainierten Beispielen enthält. Um Objekte im ganzen Bild finden zu können, muss das zu untersuchende Fenster in Schritten über das gesamte Bild hinweg bewegt werden. Der Klassifikator kann auch skaliert werden, um Objekte zu finden, die größer oder kleiner als die ursprünglich trainierte Größe sind.

In [19] werden Klassifikatoren für ganze Körper, Beine und Oberkörper trainiert. Im vorliegenden Fall kommt nur der Klassifikator für ganze Körper im FIR-Bild zum Einsatz.

2.5 Kalibrierung des Systems

Bei einem System mit mehreren Sensoren ist es unerlässlich, diese genau aufeinander abzustimmen und die Eigenschaften der einzelnen Sensoren zu kennen. Im vorliegenden Fall arbeitet der Laserscanner in einem dreidimensionalen Koordinatensystem, wohingegen die Kamera Pixel-Bilder liefert. Sollen die Ergebnisse dieser beiden Sensoren fusioniert werden, ergibt sich die Notwendigkeit, die Parameter beider Sensoren zu bestimmen. Denn letztendlich sollen z. B. Messpunkte, welche der Laserscanner liefert, in das Bild der FIR-Kamera projiziert werden. Der folgende Abschnitt gibt einen Einblick in die zur Kalibrierung des Versuchsfahrzeugs verwendeten Verfahren. Zum einen werden die intrinsischen Parameter der verwendeten FIR-Kamera bestimmt und zum anderen wird die Lage des Laserscanners und der Kamera in Bezug auf das Fahrzeug ermittelt. Ersteres geschah mittels einer eigens dafür entwickelten Kalibriervorrichtung für FIR-Kameras, letzteres durch hochgenaues Vermessen der Positionen der Sensoren auf einer Messplatte in der Werkstatt.

2.5.1 Intrinsische Parameter der FIR-Kamera

Ein Verfahren zur Kalibrierung von Videokameras ist in [35] beschrieben. Dabei werden Punkte, welche auf einer Ebene liegen, aus verschiedenen Perspektiven mit der Kamera aufgenommen. Daraus errechnen sich dann die relevanten Parameter wie Brennweite, Bildmittelpunkt oder Verzerrung. Eine Umsetzung dieses Verfahrens - in leicht abgewandelter Form - stellt die „Calib Toolbox“² für Matlab dar. Dabei dienen als Referenzpunkte die Schnittpunkte der Kanten auf einem Schachbrett. Diese Punkte können „subpixel-genau“ extrahiert werden. Das beschriebene Vorgehen setzt allerdings voraus, dass ein entsprechendes Schachbrettmuster auf eine Oberfläche aufgebracht wird und die Kamera dann einzelne Aufnahmen aus unterschiedlichen Richtungen liefert. Für die hier verwendete FIR-Kamera bedeutet dies jedoch, dass das Schachbrett im Ferninfrarot-Spektrum und nicht nur im sichtbaren Spektrum sichtbar sein muss. Die Schwierigkeit bestand nun darin, ein Schachbrett aus warmen (im FIR-Bild weiß) und kalten Flächen (im FIR-Bild schwarz) herzustellen³.

Dabei ist eine scharfe Abgrenzung der warmen von den kalten Flächen für einen guten Kontrast im FIR-Bild wichtig. Ist der Temperaturunterschied zu gering, sind im FIR-Bild nur noch Abstufungen von Grau zu sehen, aber kein „scharfes“ Schachbrettmuster.

Zunächst werden auf einer geheizten Metallplatte isolierende Materialien (Styropor, Karton, ...) aufgebracht. Dies führt dazu, dass Bereiche *ohne* Isolierung im FIR-Bild heller erscheinen als Bereiche *mit* Isolierung. Abbildung 2.5 zeigt einen Vergleich verschiedener Isolations-Materialien. In Abbildung 2.5(b) ist ein Farbbild der Versuchsanordnung zu sehen. Auf einer schwarzen Aluplatte sind verschiedene Isolationsmaterialien aufgebracht. Die Aluplatte wird dann erhitzt. In Abbildung 2.5(a) ist das resultierende Bild in der FIR-Kamera abgedruckt.

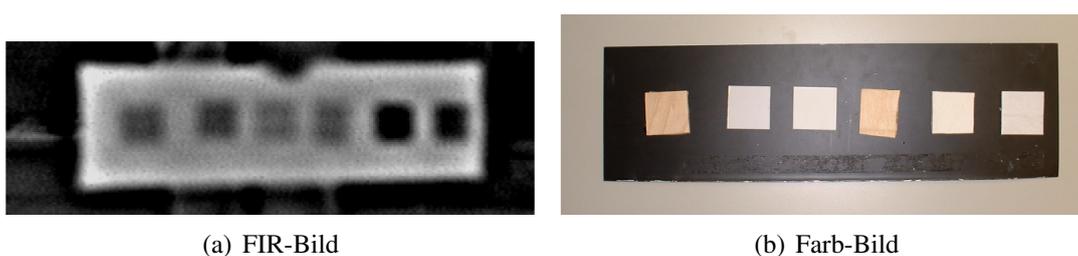


Abbildung 2.5: Versuchsreihe mit verschiedenen Isolations-Materialien auf einer Metallplatte

Es wird deutlich, dass der Temperaturunterschied zwischen warmen und kalten Flächen zum Teil nicht groß genug ist. In allen Fällen sind die Kanten zu unscharf um automatisiert Schnittpunkte subpixel-genau extrahieren zu können.

²http://www.vision.caltech.edu/bouguetj/calib_doc/

³Die Aufgabe wurde gemeinsam mit Antje Neve gelöst.

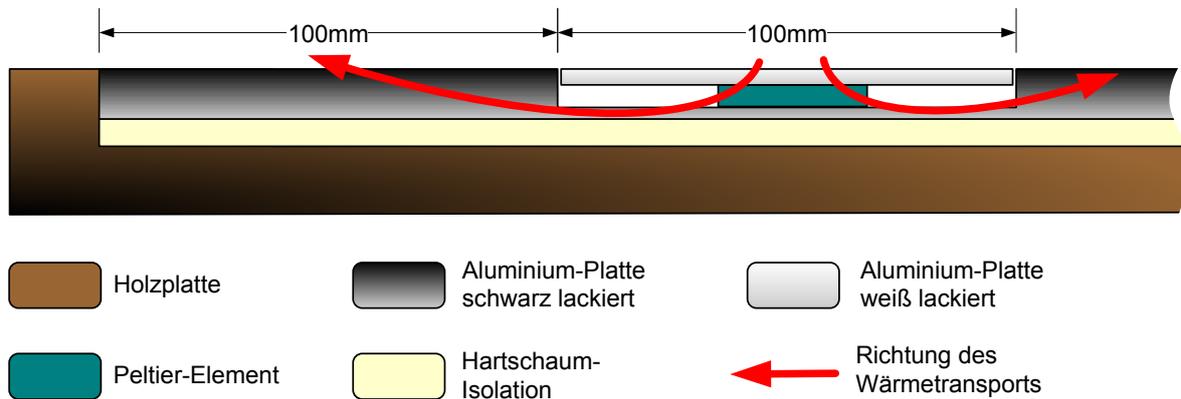
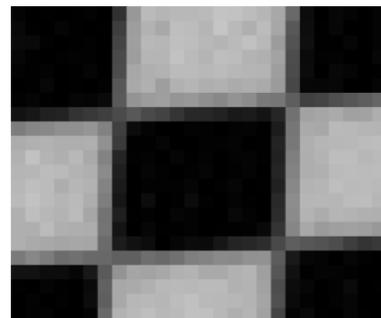


Abbildung 2.6: Zusammenbauskizze des Schachbretts



(a) Ansicht im FIR-Bild



(b) Detailansicht der Kanten

Abbildung 2.7: Ansicht des Schachbretts im FIR-Bild

Daraus entstand die Idee, die kalten Flächen nicht nur zu isolieren, sondern diese vielmehr aktiv zu kühlen. Dies wurde letztendlich durch den Einsatz von Peltier-Elementen bewerkstelligt, welche 32 der 64 Flächen des Schachbretts kühlen. Wird ein Peltier-Element von Gleichstrom durchflossen, so führt dies zu einem Wärmetransport von der einen zur anderen Seite des Elements. Für physikalische Details sei beispielsweise auf [12] und [27] verwiesen. Das heißt, Peltier-Elemente stellen lediglich einen Temperaturunterschied her. Doch genau dieser Effekt kann dazu genutzt werden, um von den zu kühlenden Flächen des Schachbretts Wärme zu den zu heizenden Flächen zu transportieren. So ist ein getrenntes Aufheizen der warmen Flächen nicht nötig, sondern ein glücklicher Nebeneffekt des Abkühlens der 32 anderen Flächen. Im Rahmen dieser Arbeit wurde ein Prototyp eines solchen Schachbretts gefertigt. In

Abbildung 2.6 ist der prinzipielle Aufbau dargestellt. Abbildung 2.7 zeigt eine Aufnahme des Schachbretts im FIR-Bild sowie eine Detailansicht der Kanten im Bild. Es sind jeweils zwei Peltier-Elemente in Reihe geschaltet und dann in einer Parallelschaltung zusammengefasst. Experimentell wurde ein Strom von ca. 20 A durch die gesamte Schaltung für gut befunden. Ein einzelnes Peltier-Element durchfließt demnach ein Strom von ca. 1,25 A, was deutlich unter der Grenze von 4,6 A für die verwendeten Elemente liegt.

Ein weiterer Vorteil der erstellten Konstruktion liegt auch darin, dass das Schachbrettmuster nun sowohl im sichtbaren Spektralbereich als auch im FIR-Spektrum erkennbar ist. Diese Tatsache nutzt die Arbeit [19] aus, um ein Stereo-System aus einer normalen Videokamera und einer FIR-Kamera zu kalibrieren.

2.5.2 Extrinsische Parameter des Versuchsträgers

Die extrinsische Kalibrierung hat das Ziel, die einzelnen Koordinatensysteme der Sensoren zueinander in Bezug zu setzen und die Positionen relativ zum Fahrzeug zu ermitteln. Letztendlich fließen die hier ermittelten Daten in die Konstruktion des Szenen-Baums (siehe 2.6) und dessen Transformations-Matrizen ein.

Zunächst wurde das Fahrzeug auf eine dafür vorgesehene Messplatte gefahren. Mit einem Messarm können dann 3D-Koordinaten von beliebigen Punkten am Fahrzeug auf wenige hundertstel Millimeter genau bestimmt werden. Tabelle 2.2 zeigt einige Messwerte. Die relevanten Sensoren sind fett gedruckt. Die so gewonnenen Messpunkte stellen jeweils den Ursprung des Sensor-Koordinatensystems dar.

Messpunkt am Fahrzeug	x [mm]	y [mm]	z [mm]
Nullpunkt BMW Emblem vorne	0	0	0
Boden ohne Beladung im Fahrzeug			-755
Zentrum Abschleppöse rechts	134	-426	-314
FIR rechts (Raytheon)	62	334	-417
FIR Mitte (Indigo 24°)	58	421	-434
FIR links außen (Indigo 36°)	6	511	-447
ALASCA Gehäuse Oberkante Mitte	256	-350	-313
MACOM Radar links	65	465	-239
MACOM Radar rechts	47	-492	-217
Radnabe vorne links	-704	850	-441
Radnabe vorne rechts	-730	-850	-443
Farbkamera Innenraum	-1982	163	460

Tabelle 2.2: Messpunkte am Fahrzeug für die extrinsische Kalibrierung

Für die „Blickrichtung“ der Sensoren wird kein spezielles Kalibrierverfahren angewendet. Ausgehend von der Ideal-Blickrichtung des Sensors (parallel zur positiven x-Achse) wurden

der Roll-, der Neigungs- und der Gierwinkel⁴ manuell ermittelt. Dies geschah mittels einer Einblendung der Lasermesspunkte ins FIR-Bild (wie in Abbildung 2.8) und gleichzeitiger Anpassung der Winkel.

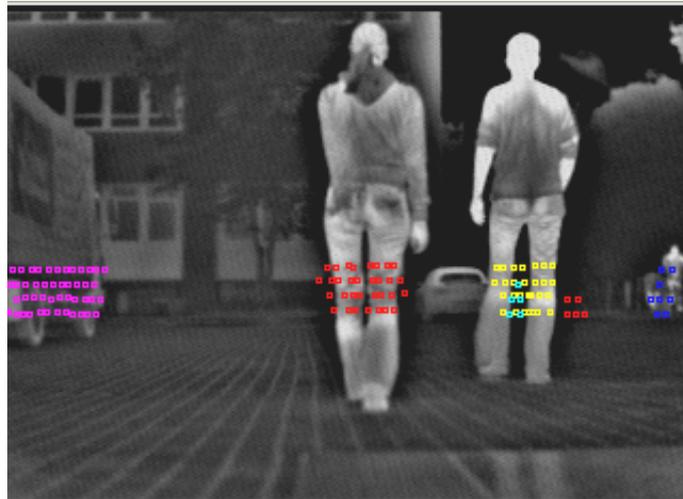


Abbildung 2.8: FIR-Bild mit überlagerten Laser-Messpunkten

2.6 Koordinatensysteme und Umsetzung als Szenen-Graph

Im Umfeld der 3D-Visualisierung sind Szenenbäume gängige Praxis [28, 20]. Sie ermöglichen es, eine Hierarchie aus Objekten, Matrix-Transformationen und anderen Eigenschaften (z.B. Texturen, Farben, ...) zu bilden. Modifikationen an Eltern-Knoten wirken sich dann immer auf alle Kinder-Knoten aus. Somit lässt sich die Beziehung „besteht aus“, wie sie in der Realität oft vorkommt, leicht modellieren. Für die Sensor-Fusion bieten Szenen-Bäume den Vorteil, dass alle Sensoren, das Fahrzeug und die erkannten Objekte in einer Baumstruktur zusammengefasst werden können. Eine Umrechnung der Koordinaten erfolgt dann durch „Wandern“ entlang der Verzweigungen und gleichzeitigem Aufmultiplizieren der Transformations-Matrizen. Alle Berechnungen werden in homogenen Koordinaten ausgeführt. Abbildung 2.9 zeigt den für diese Arbeit verwendeten Szenenbaum. Im Gegensatz zu klassischen Darstellungen wurden die Koordinatensysteme explizit angegeben. Die Richtung für die Multiplikationen wird folgendermaßen festgelegt:

- Transformation eines Punktes vom Eltern-Koordinatensystem ins Kind-Koordinatensystem: Multiplikation mit M
- Transformation eines Punktes vom Kind-Koordinatensystem ins Eltern-Koordinatensystem: Multiplikation mit M^{-1}

⁴engl.: roll, pitch and yaw

Alle Matrizen M sind orthogonale 4x4 Transformations-Matrizen. Die Matrizen P_I und P_F sind Projektions-Matrizen. Folgende Tabelle gibt Aufschluss über die Herkunft der einzelnen Matrizen:

Matrix	Beschreibung
M_F	gibt die Transformation von einem stationären Welt-Koordinatensystem ins bewegte Fahrzeug-Koordinatensystem an. M_F wird nach jedem Zeitschritt durch die Verarbeitung der Fahrdynamikdaten neu berechnet und spiegelt die Bewegung des Fahrzeugs im Welt-Koordinatensystem wieder (siehe dazu auch 2.7).
M_S	Alle Sensoren sind unter einem einzelnen Knoten zusammengefasst, um den Bezugspunkt im Fahrzeug frei wählen zu können. M_S ist konstant und wurde während der Kalibrierung bestimmt (siehe 2.5.2).
M_I	gibt die Position der Infrarot-Kamera in Bezug auf das Sensor-Koordinatensystem an.
M_F	gibt die Position der Farb-Kamera in Bezug auf das Sensor-Koordinatensystem an.
M_L	gibt die Position des Laserscanners in Bezug auf das Sensor-Koordinatensystem an.
P_I	Bei P_I handelt es sich um eine Projektionsmatrix, welche 3D-Punkte in das 2D-Koordinatensystem der FIR-Kamera abbildet. Die Elemente dieser Matrix bestimmen sich durch die Ergebnisse der intrinsischen Kalibrierung in Abschnitt 2.5.1.
P_F	Projektionsmatrix für die Farb-Kamera. Diese spielt jedoch nur eine untergeordnete Rolle, da für die Arbeit lediglich die FIR-Kamera relevant ist. Dennoch wurden z. T. zur Kontrolle die Objekte auch ins Farbbild eingeblendet.
$M_1 \dots M_n$	Diese Matrizen transformieren Punkte vom Welt-Koordinatensystem in ein objekt-lokales Koordinatensystem. Jedem Objekt wird ein System zugeordnet, bei dem der Nullpunkt im Schwerpunkt des Objekts und die X-Achse immer in Bewegungsrichtung zeigt. Dies wird später für die Analyse der Pendelbewegungen wichtig werden.

Für die perspektivischen Projektionsmatrizen P_F und P_I werden nur die Werte für die Brennweite und den optischen Mittelpunkt verwendet. Eine Berücksichtigung der Verzerrung findet nicht statt. Es zeigt sich jedoch, dass diese Annahme genau genug für den vorliegenden Anwendungsfall ist.

Nach [13] lässt sich für die perspektivische Projektion mit Hilfe des Lochkamera-Modells folgende Abbildung definieren:

$$(X, Y, Z)^T \mapsto (fX/Z + u_c, fY/Z + v_c)^T \quad (2.1)$$

Dabei stellt f die Brennweite der Kamera in Pixel dar. u_c und v_c sind die Pixelkoordinaten des optischen Bildmittelpunkts der Kamera (siehe auch Abbildung 2.10).

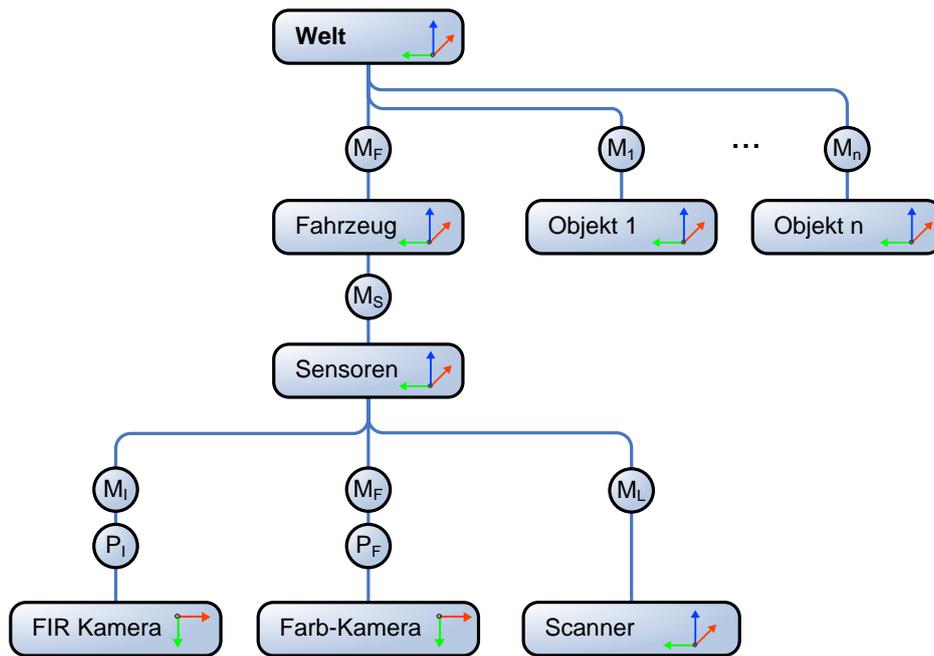


Abbildung 2.9: Der Szenenbaum mit allen Transformations-Matrizen

Nach dem allgemein gültigen Standard im automobilen Umfeld sind alle Koordinatensysteme „Rechte-Hand-Systeme“ mit X nach vorne, Y nach links und Z nach oben. Deshalb ist eine Vertauschung der Koordinaten nötig. Zusätzlich liegt der Ursprung unseres Pixelkoordinatensystems der Kamera am linken oberen Bildrand. Deshalb ist die Gleichung 2.1 umzuschreiben:

$$(X, Y, Z)^T \mapsto (-fY/X + u_c, -fZ/X + v_c)^T \quad (2.2)$$

Diese Abbildung lässt sich mit Hilfe der homogenen Koordinaten als Matrix-Produkt schreiben:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} -fY + Xu_c \\ -fZ + Xv_c \\ X \end{pmatrix} = \begin{bmatrix} u_c & -f & 0 & 0 \\ v_c & 0 & -f & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.3)$$

2.7 Eigenbewegung des Fahrzeugs

Befindet sich das Fahrzeug in Bewegung, so muss sich dieser Umstand im Modell entsprechend widerspiegeln. Dazu dient die bereits eingeführte Transformations-Matrix M_F , welche

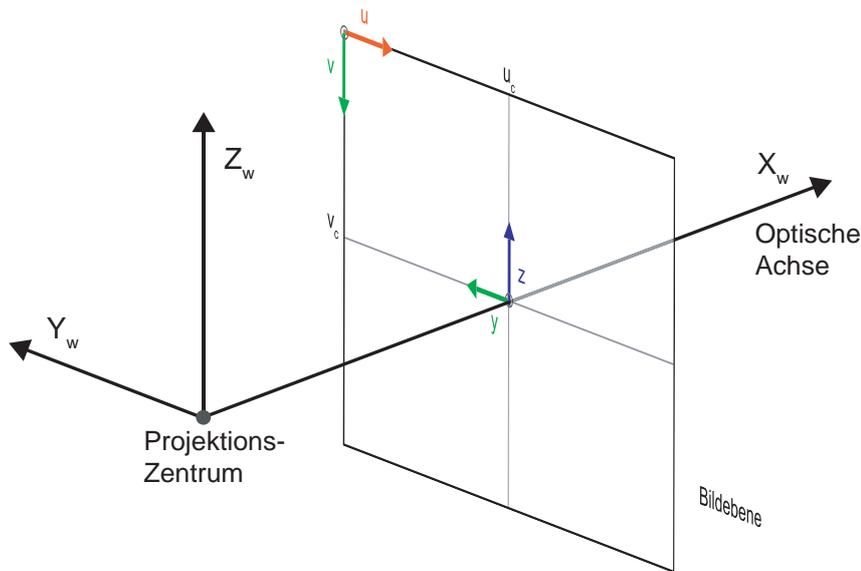


Abbildung 2.10: Herleitung der Projektionsmatrix für die Kamera

Punkte vom stationären Koordinatensystem in ein relativ zum Fahrzeug gelegenes Koordinatensystem überführt. Da alle Sensoren unterhalb dieses Knotens im Szenenbaum gelegen sind, wirkt sich eine Änderung der Position des Fahrzeugs automatisch auf die Position aller Sensoren aus.

Es ist nun das Ziel, aus den zur Verfügung stehenden Messdaten während der Fahrt, möglichst genau die realen Bewegungen des Fahrzeugs zu rekonstruieren. Dazu bildet man im Normalfall ein Modell des Fahrzeugs (z. B. Einspur- oder Zweispurmodell in [31]), parametrisiert dieses auf den jeweiligen Fahrzeugtyp und verwendet dann die realen Daten als Eingabe. Eingabegrößen können z. B. die Eigengeschwindigkeit, der Gierwinkel, die einzelnen Radgeschwindigkeiten oder die Querbeschleunigung sein. Da die Rekonstruktion der genauen Fahrzeug-Trajektorie kein Schwerpunkt in dieser Arbeit ist, wird auf eine detaillierte Entwicklung eines Fahrzeugmodells verzichtet.

Auf trockener Fahrbahn und bei geringen Geschwindigkeiten werden zur ersten Näherung die Eigengeschwindigkeit v [m/s] und die Gierrate ψ [rad/s] des Fahrzeugs herangezogen. Die Transformation lässt sich dann in zwei Teile zerlegen, nämlich in eine Drehung M_{F_r} und in eine Verschiebung M_{F_t} . Das Fahrzeugmodell wird in jedem Zeitschritt um einen Winkelbetrag von $\psi\Delta t$ um die Z-Achse gedreht und dann um einen Betrag von $v\Delta t$ in Fahrtrichtung verschoben.

Daraus ergibt sich die Innovations-Gleichung für die Drehung des Fahrzeugs:

$$M_{F_r}[t + \Delta t] = \begin{pmatrix} \cos(\psi\Delta t) & \sin(\psi\Delta t) & 0 & 0 \\ -\sin(\psi\Delta t) & \cos(\psi\Delta t) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} M_{F_r}[t] \quad (2.4)$$

Für die Verschiebung des Fahrzeugs lässt sich folgende Gleichung angeben:

$$M_{F_i}[t + \Delta t] = M_{F_r}[t + \Delta t] \begin{pmatrix} 0 & 0 & 0 & v\Delta t \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + M_{F_i}[t] \quad (2.5)$$

Aus 2.4 und 2.5 folgt damit für die vollständige Transformation⁵:

$$M_F[t + \Delta t]^{-1} = M_{F_i}[t + \Delta t] M_{F_r}[t + \Delta t]$$

⁵Die Inversenbildung in dieser Gleichung ist notwendig, um nicht mit der Szenenbaum-Definition weiter vorne in Konflikt zu geraten.

VERARBEITUNG DER LIDAR-DATEN

Wie bereits in der Einleitung erwähnt, liegt einer der Schwerpunkte der Arbeit auf der Verarbeitung und Aufbereitung der Laserscan-Daten. Drei Teilaspekte, welche die folgenden Abschnitte genauer beleuchten, sind von besonderem Interesse:

- **Die Segmentierung der Messpunkte**

Das heißt, solche Punkte zusammenzufassen, welche auch in der Realität zu ein und demselben Objekt gehören. Dazu zählt auch das Auffinden von teilverdeckten Segmenten. Hierzu werden eigene Verfahren entwickelt.

- **Das Tracking der gefundenen Objekte**

So ist eine Schätzung der Geschwindigkeit und der Bewegungsrichtung möglich. Dies ist zum einen wichtig, um für spätere Anwendungen entscheiden zu können, ob sich ein Fußgänger im Gefahrenbereich befindet. Zum anderen gibt die Geschwindigkeit auch Aufschluss über den Typ eines Objekts. Für das Tracking wird ein vorhandenes Multi-Hypothesen-Framework angepasst und die Kalman-Filter werden parametrisiert.

- **Analyse der Pendelbewegungen der Beine**

Da ein Laserscanner, anders als eine Kamera, keine Texturen wahrnimmt, ist dies ein wichtiges Kriterium, um Fußgänger von anderen, ähnlich „aussehenden“ Objekten zu unterscheiden. Dazu wird in Anlehnung an [8] ein Verfahren implementiert, welches die Pendelbewegungen mittels Fourier-Analyse untersucht.

Die genannten Schritte sind auch in Abbildung 2.1 auf Seite 13 mit den Kennziffern ②, ③, ④ und ⑤ dargestellt.

3.1 Segmentierung

Nach dem Empfang eines neuen Laserscans wird dieser zunächst in einzelne Teile, die Segmente, zerlegt. Später dienen dann die Segmente als kleinste Einheit, um neue Objekte für den Tracker zu kreieren. Da der Tracker nur jeweils einen Punkt pro Objekt verfolgt, muss ein geeigneter Referenz-Punkt für jedes Segment bestimmt werden. Hier wird der Schwerpunkt aus allen zu einem Segment gehörenden Punkten gewählt. Es zeigt sich, dass dieser Ansatz für relativ kleine Objekte, zu denen auch Fußgänger gehören, gute Ergebnisse liefert. Der folgende Abschnitt soll einen Einblick in das verwendete Segmentierungs-Verfahren geben.

3.1.1 Segmentierung durch Tiefensuche

Grundlegender Ansatz ist die Vermutung, dass zwei Messpunkte mit geringem räumlichen Abstand zum gleichen Objekt gehören. Man startet also mit einem Messpunkt und sucht weitere Punkte in der Umgebung, die innerhalb eines bestimmten Abstands liegen. Dies führt zu einem aus der Bildverarbeitung bekannten Verfahren, dem „Region Growing“. Es wird versucht, ausgehend von einem Startpunkt, immer neue Nachbarn zu suchen, welche das Abstandskriterium erfüllen. Der Startpunkt wird dann für das nächste Segment, außerhalb aller bisher gefundenen Segmente, neu gewählt und das Verfahren wird wiederholt. Formal lässt sich das Verfahren als Graphenproblem darstellen:

Sei $G = (V, E)$ ein ungerichteter Graph mit $V = \{ \text{„Menge aller Messpunkte“}, \mathbf{v} \in \mathbb{R}^3 \}$ und $E = \{ (\mathbf{v}_1; \mathbf{v}_2) \mid \mathbf{v}_1, \mathbf{v}_2 \in V \wedge \|\mathbf{v}_1 - \mathbf{v}_2\| < d_{max} \}$. Ziel ist es nun, alle Zusammenhangskomponenten des Graphen zu finden. Diese entsprechen dann den gesuchten Segmenten. Dies lässt sich mit Hilfe einer Tiefensuche realisieren. Allerdings ist es nicht erstrebenswert, die Menge E a priori vollständig zu ermitteln. Zum einen müsste so jeder Punkt mit jedem anderen in V verglichen werden und zum anderen ist der in der Definition angegebene Schwellwert d_{max} keine Konstante, sondern abhängig vom Abstand der gemessenen Punkte. Um dem radialen Messprinzip des Laserscanners gerecht zu werden, ist der Schwellwert für weiter entfernte Punkte größer. In größerem Abstand zum Scanner resultiert die gleiche Winkelauflösung nämlich in einem größeren euklidischen Abstand. Damit scheitern auch Verfahren, die den transponierten Graphen nutzen, um die Zusammenhangskomponenten zu finden [5].

Der Graph bzw. die Menge E wird also erst während der Tiefensuche implizit gebildet. D. h. nur, wenn man den Abstand zwischen zwei Punkten benötigt, wird dieser berechnet und mit dem Schwellwert verglichen. Um das Verfahren effizient ausführen zu können, bedarf es einer Heuristik, die dabei hilft, die Menge der potentiellen Nachbarn einzuschränken.

An dieser Stelle ist es sinnvoll, die Organisation der Daten, welche der Laserscanner liefert, zu kennen. Abbildung 3.1 zeigt die schematische Darstellung eines Laserscans mit mehreren Segmenten und der dazugehörigen Datenstruktur, die der Scanner sendet. Die Werte sind nach aufsteigendem Winkel φ sortiert. Allerdings kann diese Topologie nicht ohne weiteres ausgenutzt werden. Punkte, die im Array nebeneinander liegen, müssen in der Realität nicht

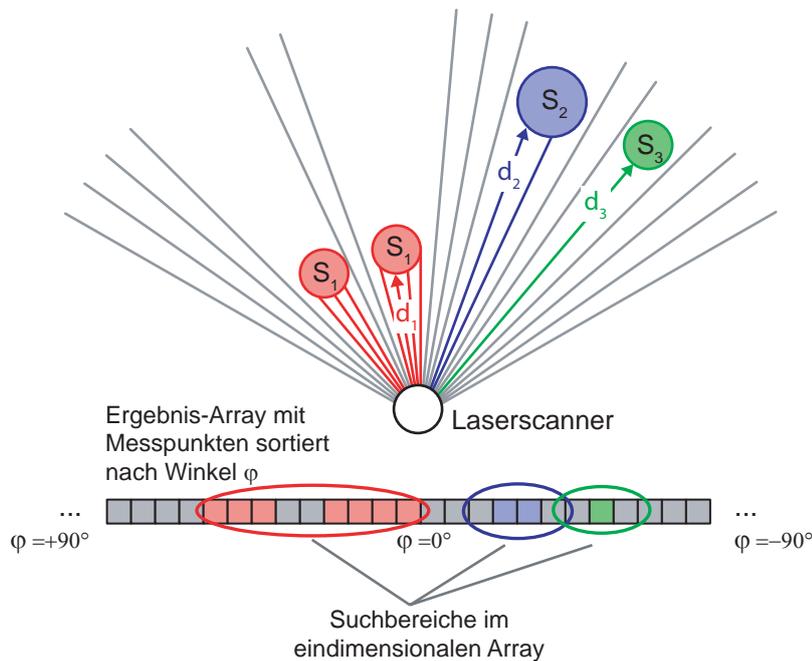


Abbildung 3.1: Ein vereinfachter Laserscan und die Organisation der Daten

unbedingt das Abstandskriterium erfüllen. Andererseits können auch Punkte, die im Array nicht nebeneinander liegen, zum gleichen Segment gehören. Die Wahrscheinlichkeit jedoch, dass zwei Punkte, die im Array nebeneinander liegen, auch im dreidimensionalen Raum dicht beisammen liegen, ist sehr hoch. Das Verfahren nutzt diesen Umstand aus, indem es bevorzugt Nachbarpunkte mit kleinem Winkelabstand $\Delta\varphi$ untersucht. Das Segment S_1 in Abbildung 3.1 zerfällt in zwei Teilstücke (z. B. die beiden Beine eines Fußgängers). Da der Abstand d_1 klein ist, wird ein größerer Bereich im Array nach potentiellen Nachbarn abgesucht. Zu allen Punkten im Suchabstand wird der euklidische Abstand gebildet und dieser dann mit dem Schwellwert verglichen. Die Segmente S_2 und S_3 hingegen haben einen größeren Abstand zum Scanner (d_2 und d_3), deshalb ist die Suchumgebung hier wesentlich kleiner.

So passt sich also die Tiefensuche im Graphen an die gemessenen Abstände an, um unnötige Abstandsmessungen zwischen Punkten zu vermeiden. Die bisherige Betrachtung hat das Problem auf eine Scanebene beschränkt. Wenn die Punkte aller Ebenen in das Array eingefügt werden, sind die Suchbereiche lediglich mit einem konstanten Vergrößerungsfaktor zu belegen. So können zwischen zwei Messpunkten ein und derselben Ebene auch einige Punkte anderer Ebenen liegen. Das restliche Verfahren bleibt davon unberührt.

Algorithmus 3.1 beschreibt informell die Umsetzung des Verfahrens. Die Funktion „FindeNächstenStartpunkt“ sucht einen Punkt, dem bisher noch kein Segment zugewiesen wurde. Dieser dient dann als Startpunkt und wird auf den Stack für die Tiefensuche gelegt. Die innere While-Schleife läuft dann solange, bis keine weiteren Punkte mehr gefunden werden, die zu dem eben erzeugten Segment gehören. Dazu werden jeweils die Nachbarn des obersten Elements des Stacks betrachtet. Der Suchbereich im Array wird anhand der vorher beschriebenen

Algorithmus 3.1 : Segmentierung des Laserscans mit adaptiver Tiefensuche

Input : Array mit Messpunkten sortiert nach φ

Result : Punkte zu Segmenten zusammengefasst

Data : IndexS : Startindex; IndexA: Aktueller Index; IndexN: Nachbar-Index

begin

```
    while IndexS =FindeNächstenStartpunkt () do
        Stack .Push (IndexS);
        Segment =ErzeugeNeuesSegment (IndexS);
        while not Stack .IstLeer () do
            IndexA = Stack .Peek ();
            Suchbereich = ErmittleSuchbereich (IndexA);
            NachbarGefunden = false ;
            for IndexN = IndexA- Suchbereich to IndexA + Suchbereich do
                if not SegmentZugewiesen (IndexN) and IndexN!=IndexA then
                    if Abstand (IndexA, IndexN) < ErmittleMaxAbstand (IndexA) then
                        NachbarGefunden = true ;
                        Segment .FügeHinzu (IndexA);
                        Stack .Push (IndexA);
                    end
                end
            end
            if not NachbarGefunden then
                Stack .Pop (IndexA)
            end
        end
        if Segment .AnzahlPunkte () < MinAnzahlPunkte then
            VerwerfeSegment ( Segment);
        end
    end
end
```

Methode dynamisch bestimmt. Ist das Abstandskriterium erfüllt, so landet der eben gefundene Nachbar auf dem Stack und wird Teil des Segments. Der zulässige Maximalabstand ist ebenfalls variabel und, wie bereits beschrieben, vom Punkt-Sensor-Abstand abhängig. Segmente mit weniger als 5 Messpunkten werden am Ende verworfen. In den meisten Fällen stammen diese Messpunkte von eher uninteressanten Objekten, wie beispielsweise Blättern auf der Straße.

3.1.2 Erkennen von Schattenwurf

Ein weiterer Aspekt bei der Verarbeitung der LIDAR-Messdaten ist die Betrachtung des Schattenwurfs des Scanners. Auch ein Laserscanner kann nicht durch massive Objekte hindurchsehen. Es gibt deshalb Bereiche im Scan, die nicht erfasst werden können, da diese durch andere Objekte verdeckt sind. Abbildung 3.2 zeigt exemplarisch zwei Segmente S_1 , S_2 und deren Schattenwurf. Die Dreiecke sollen stark vereinfacht einige Messpunkte des Scanners darstellen. Gezeigt ist hier ein Fall, der die Notwendigkeit einer getrennten Behandlung von Schattenbereichen verdeutlicht. S_1 und S_2 seien die zu zwei Fußgängern gehörenden Segmente, die sich von rechts nach links am Scanner vorbei bewegen. Durch die „Lücke“ zwischen den Objekten entsteht auf einem größeren Hindernis im Hintergrund ein weiteres Segment S_3 . Dieses Segment wird sich im Laserscan ebenfalls nach links bewegen und eine ähnliche Größe wie ein Fußgänger aufweisen. Sogar Bein-Pendelbewegungen ließen sich unter Umständen nachweisen, da das Segment gewissermaßen ein „Negativ“ der beiden Segmente im Vordergrund ist.

Verallgemeinernd lässt sich sagen, dass Segmente, die an Schattenbereiche angrenzen, für ein weiteres Tracking nur bedingt tauglich sind. Deshalb schließt sich der Suche nach den eigentlichen Segmenten eine Markierung solcher teilweise verdeckter Segmente an. Betrachtet man Punkte von zwei aneinander grenzenden Segmenten S_1 und S_3 , so lässt sich sagen, dass S_3 dann teilverdeckt ist, falls der radiale Abstand d_r sehr viel größer als der tangentielle Abstand d_t ist (siehe Abbildung 3.2). Algorithmus 3.2 nutzt diese Eigenschaft aus, indem er jeweils benachbarte Punkte im Array vergleicht. Immer, wenn der tangentielle Abstand eine bestimmte Grenze unterschreitet, wird geprüft, ob gleichzeitig der radiale Abstand eine Grenze übersteigt. Je nach Vorzeichen des radialen Abstandes d_r ist entweder das eine oder andere Segment als teilverdeckt zu markieren. Segmente, welche der vorige Segmentierungs-Schritt (Algorithmus 3.1) als zu klein eingestuft hat, werden bei der Suche nach teilverdeckten Segmenten übersprungen.

3.2 Tracking

Um Objekte über einen längeren Zeitraum zu verfolgen, kommen Trackingverfahren zum Einsatz, welche aktuelle Messungen mit bereits ermittelten Objekten in Beziehung setzen. So ist beispielsweise eine Schätzung der Geschwindigkeit und der Bewegungsrichtung von

Algorithmus 3.2 : Auffinden von teilverdeckten Segmenten

Input : Array mit Messpunkten sortiert nach φ und zugewiesener Segment Nummer

Result : Markierung aller teilweise verdeckten Segmente

Data : IndexA: Aktueller Index; IndexN: Nachbar-Index

begin

```

    IndexA = 0;
    while IndexA < AnzahlScanpunkte do
        /* Prüfen ob der Punkt zu einem kleinen Segment gehört (siehe
           Alg. 3.1) */
        if GehörtZuVerworfenemSegment (IndexA) then
            IndexA = IndexA +1;
            continue ;
        end
        /* Nach einem passenden Nachbar-Punkt im Array Suchen */
        IndexN = IndexA ;
        repeat
            IndexN =IndexN +1;
        until (IndexN ≥ AnzahlScanpunkte or
              (SegmentVon (IndexN) ≠SegmentVon (IndexA)
              and not GehörtZuVerworfenemSegment (IndexN) ) ) ;

        if IndexN ≥AnzahlScanpunkte then
            break ;
        end
         $d_r$  =RadialerAbstand (IndexN, IndexA);
         $d_t$  =TangentialerAbstand (IndexN, IndexA);
        if  $d_t$  < MaxTangentialerAbstand then
            if  $d_r$  > MaxRadialerAbstand then
                MarkiereSegmentAlsVerdeckt (SegmentVon (IndexA));
            else if  $d_r$  < -MaxRadialerAbstand then
                MarkiereSegmentAlsVerdeckt (SegmentVon (IndexN));
            end
        end
        IndexA = IndexN +1;
    end
end

```

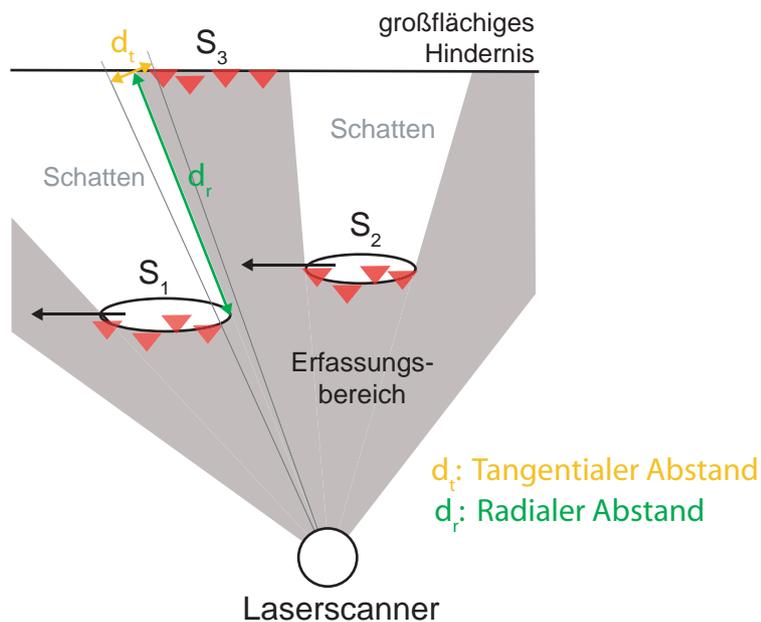


Abbildung 3.2: Zwei Segmente und deren Schattenwurf in der Draufsicht

Objekten über die Zeit hinweg möglich. Die Geschwindigkeit wird später ein wichtiges Kriterium für die Entscheidung, ob es sich bei einem Objekt um einen Fußgänger handelt. Die Richtung der Bewegung ist vor allem für Risikoabschätzungen oder „Time-To-Collision“ Berechnungen interessant.

Bei einem Trackingverfahren stellen sich meist zwei Aufgaben:

- **Die Modellierung der Bewegungsmuster** der zu trackenden Objekte (z. B. Autos, Fußgänger, etc.). Hier fließt Wissen über die Objekte bezüglich maximaler Geschwindigkeit oder Beschleunigung ein, aber auch die Art und Weise, wie Richtungswechsel durchgeführt werden können. Dazu kommen Kalman-Filter zum Einsatz.
- Das Erzeugen von neuen Objekten aus Messungen und das Wiederfinden vorhandener Objekte in einer neuen Messung; also kurz die „Messwert-Objekt-Zuordnung“. Diese Aufgabe übernimmt in unserem Fall ein **Multi-Hypothesen-Framework**, welches mehrere mögliche Zuordnungen gleichzeitig verfolgen kann.

Die folgenden Abschnitte gehen genauer auf die verwendeten Kalman-Filter und das implementierte Multi-Hypothesen-Tracking ein.

3.2.1 Parameter der Kalman-Filterung

Kalman-Filter sind eine bewährte Technik, um aus verrauschten Sensorinformationen die relevanten Größen zu schätzen. Sie erlauben eine wahrscheinlichkeitstheoretische Modellie-

rung von stochastischen Prozessen und deren Eingangsgrößen. Eine zeitliche Vorhersage des Zustands des Systems ist ebenfalls möglich. Sie bilden deshalb die Grundlage der meisten Tracking-Verfahren. In der Arbeit wird die Nomenklatur aus [32] verwendet.

Der Kalman-Filter ist ein Konzept zur Abschätzung des Zustandes x eines zeitabhängigen Prozesses. Die Innovationsgleichung setzt den Zustand x_k zum Zeitpunkt k mit dem Zustand x_{k-1} einen Zeitschritt davor in Bezug:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (3.1)$$

Dies geschieht über die Matrix A , welche den Prozess modelliert, also die Art und Weise, wie sich der Zustand x über die Zeit hinweg entwickelt. Zusätzlich sieht die ursprüngliche Form der Kalman-Gleichung auch noch einen kontrollierenden Faktor Bu_{k-1} vor, der eine Steuerung des Prozesses von außen ermöglicht. Dies ist im vorliegenden Fall jedoch nicht relevant, da kein Einfluss auf die Bewegung eines Fußgängers möglich ist. w_{k-1} schließlich ist eine normalverteilte Zufallsvariable, die das Prozessrauschen darstellt, also gewissermaßen die Unsicherheit, die im Prozess selbst vorhanden ist. Dies wird später relevant werden, da selbst bei genauester Positionsbestimmung eines Fußgängers die Bewegung desselben nicht mit absoluter Sicherheit vorhergesagt werden kann. Der Prozess „Gehender Fußgänger“ ist mit einer gewissen zufälligen Unsicherheit behaftet. w heißt auch Vorhersagefehler.

Ist der Prozess beschrieben, setzt man als nächstes eine Messung z mit dem Zustand x in Beziehung:

$$z_k = Hx_k + v_k \quad (3.2)$$

Die Matrix H bildet einen Zustand in den Messraum ab. Eine Messung wird später zum Beispiel die Position des Fußgängers sein. Da auch die Messung mit einem Fehler behaftet ist, führt man den ebenfalls normalverteilten Faktor v , den Messfehler, ein. Die beiden Fehler w und v sind folgendermaßen verteilt:

$$w \sim N(0, Q) \quad (3.3)$$

$$v \sim N(0, R) \quad (3.4)$$

Q ist hierbei die „Vorhersagefehler-Kovarianzmatrix“ und R die „Messfehler-Kovarianzmatrix“. Diese beiden Matrizen müssen zusammen mit den Matrizen A und H sinnvoll belegt werden.

Der Kalman-Algorithmus löst das Problem rekursiv. In jedem Schritt findet zunächst eine Projektion des Zustands x in die Zukunft (Vorhersage) statt. Darauf folgt eine Korrektur durch eine reale Messung. Für Details zur Herleitung sei auf [32] verwiesen. Der Vollständigkeit halber werden hier die Gleichungen für die Vorhersage genannt:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (3.5)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (3.6)$$

Dabei wird aus der a-posteriori-Schätzung¹ \hat{x} zum Zeitpunkt $k-1$, eine a-priori-Schätzung² \hat{x}^- zum Zeitpunkt k ermittelt. Gleichzeitig überführt man die a-posteriori-Schätzfehlerkovarianz P zum Zeitpunkt $k-1$ in eine a-priori-Schätzung P_k^- zum Zeitpunkt k .

Die Gleichungen für die Korrektur der Schätzung verwenden die aktuelle Messung, um eine a-posteriori-Schätzung zu ermitteln:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (3.7)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \quad (3.8)$$

$$P_k = (I - K_k H) P_k^- \quad (3.9)$$

K wird in dieser Gleichung als Kalman-Gain bezeichnet.

Für das Fußgänger-Tracking wird der Zustandsvektor x folgendermaßen definiert:

$$x = \begin{pmatrix} p_x \\ p_y \\ v_x \\ v_y \\ e_x \\ e_y \\ e_z \end{pmatrix} \quad (3.10)$$

Wobei p die Position im stationären Weltkoordinatensystem, v die absolute Geschwindigkeit und e die räumliche Ausdehnung des Objekts in den drei Dimensionen ist. e wird später für die Erzeugung einer geeigneten „Region of Interest“ (ROI) im FIR-Bild relevant werden. Für die Matrix A , welche den Zustand in den nächsten Zeitschritt überführt, wird in dieser Arbeit Folgendes verwendet:

$$A = \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.11)$$

Die Operation Ax_k führt dazu, dass in jedem Zeitschritt für beide Dimensionen x und y die Position folgendermaßen berechnet wird:

$$p_k = p_{k-1} + \Delta t v_{k-1} \quad (3.12)$$

¹a posteriori: im Sinne von „nach der Messung im letzten Zeitschritt“

²a priori: im Sinne von „vor der Messung in diesem Zeitschritt“

Eine Messung z besteht aus folgenden Komponenten:

$$z = \begin{pmatrix} p_x \\ p_y \\ e_x \\ e_y \\ e_z \end{pmatrix} \quad (3.13)$$

Anders als beispielsweise Radar-Sensoren liefern LIDAR-Sensoren keine Information über die (Relativ-)Geschwindigkeit eines gemessenen Objekts. Da die Größen p und e direkt übernommen werden, ergibt sich für H Folgendes:

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.14)$$

Die Matrix Q wird in Anlehnung an [14] belegt. Ausgangsbasis ist die Annahme, dass ein Fußgänger eine bestimmte normalverteilte Geschwindigkeits-Änderung in einem Zeitschritt Δt vollziehen kann. Dies kann in beiden Richtungen unabhängig voneinander erfolgen. Zunächst sei dieser Unterschied Δv mit Eins gewählt. Dann ergibt sich für die maximale Positionsabweichung in einem Schritt $\Delta v \Delta t = \Delta t$. Dies lässt sich nun für die Berechnung der Kovarianz-Matrix nutzen:

$$Q_1 = \sigma_v^2 \begin{pmatrix} \Delta t \\ 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} \Delta t \\ 0 \\ 1 \\ 0 \end{pmatrix}^T + \sigma_v^2 \begin{pmatrix} 0 \\ \Delta t \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ \Delta t \\ 0 \\ 1 \end{pmatrix}^T \quad (3.15)$$

Da die Annahme getroffen wird, dass die Fehler in x - und y -Richtung unabhängig voneinander sind, sind die beiden Teilterme zu addieren. Durch Veränderung der Standardabweichung σ_v der Geschwindigkeit lassen sich verschiedene Annahmen über die maximale Richtungs- und Geschwindigkeitsänderung eines Fußgängers realisieren. Q_1 ist der linke obere Teil der Kovarianz-Matrix Q :

$$Q = \begin{pmatrix} \sigma_v^2 \Delta t^2 & 0 & \sigma_v^2 \Delta t & 0 & 0 & 0 & 0 \\ 0 & \sigma_v^2 \Delta t^2 & 0 & \sigma_v^2 \Delta t & 0 & 0 & 0 \\ \sigma_v^2 \Delta t & 0 & \sigma_v^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_v^2 \Delta t & 0 & \sigma_v^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_e^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_e^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_e^2 \end{pmatrix} \quad (3.16)$$

Der rechte untere Teil ist für die Filterung der räumlichen Ausdehnung in den drei Dimensionen des Objekts zuständig. Hierfür wird eine gesonderte Standardabweichung festgelegt.

3.2.2 Multi-Hypothesen-Ansatz

Bei jedem Tracking-Verfahren müssen die von den Sensoren erzeugten Messungen den bereits getrackten Objekten zugeordnet werden. Darüber hinaus muss entschieden werden, ob bestimmte Messungen zu neuen Objekten führen oder alte Tracks verworfen werden sollen. Ein normaler Tracker führt diese Messwert-Zuordnung in jedem Zeitschritt unabhängig von allen vorhergehenden Zuordnungen aus. Dies kann dazu führen, dass sich falsche Zuordnungen in der Vergangenheit auch in allen folgenden Zeitschritten negativ auswirken. Eine Korrektur ist mangels Wissens über die vorigen Zuordnungen nicht mehr möglich.

Ein Multi-Hypothesen-Tracker (MHT) hingegen schiebt die Entscheidung darüber, welche Messwerte welchen Objekten zugeordnet werden, einige Zeitschritte hinaus. Er betrachtet immer die letzten n Zuordnungen und verfolgt auch mehrere Varianten (Hypothesen). Dadurch ist es möglich, weitaus zuverlässigere Entscheidungen zu treffen, denn nur ausreichend wahrscheinliche Hypothesen werden an die Anwendung weitergegeben.

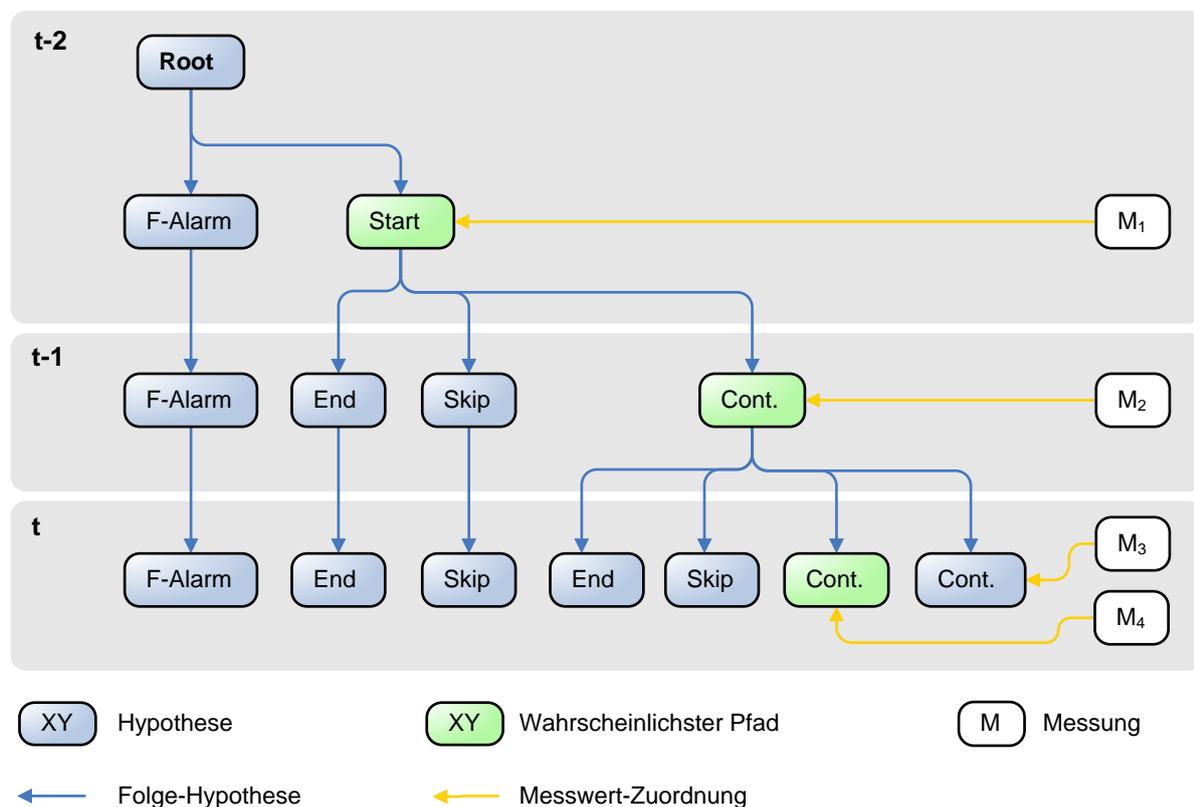


Abbildung 3.3: Entwicklung eines einzelnen Hypothesen-Baums

Eine Objekt-Hypothese ist eine Annahme über den Zustand eines Objekts in der Wirklichkeit. Dazu gehören zum einen die Position und die Geschwindigkeit des Objekts, aber auch die Wahrscheinlichkeit, dass die getroffene Annahme der Wirklichkeit entspricht. Für jedes Objekt verwaltet der MHT-Algorithmus einen Baum, dessen Elemente Hypothesen sind. Jede

Hypothese schätzt den zukünftigen Zustand eines Objekts mittels des in 3.2.1 eingeführten Kalman-Filters. Mit jedem Zeitschritt wächst der Baum um eine Ebene, indem an den Blättern neue Hypothesen generiert werden. Da sich die Anzahl der Objekte über die Zeit hinweg ändert, ändert sich die Anzahl der verwalteten Bäume ebenfalls. Abbildung 3.3 zeigt exemplarisch den Aufbau eines Hypothesen-Baums. Der Baum wird mit einer Wurzel („Root“) erzeugt und erhält im Zeitschritt $t - 2$ zwei Kinder. Einerseits ergibt sich die Hypothese, dass es sich bei M_1 um eine Fehlinformation („False-Alarm“) handelt, und andererseits, dass die Messung M_1 zu einem neuen Objekt gehört („Start“). Diese zwei Hypothesen erzeugen dann im nächsten Zeitschritt $t - 1$ weitere Folge-Hypothesen. Das Kind „Skip“ spiegelt die Annahme wider, dass im aktuellen Zeitschritt vom Objekt keine Messung vorliegt. Dieser Fall tritt häufig bei kurzzeitigen Verdeckungen auf, wie sie Abbildung 3.4 zeigt. Im Zeitschritt 4 liegt für den rot markierten Fußgänger keine Messung vor. Der Track wird dennoch fortgesetzt. Das Kind „End“ zieht die Möglichkeit in Betracht, dass das Objekt nicht mehr existiert. Wiederum gibt es auch die Annahme, dass die Messung M_2 vom gleichen Objekt stammt („continue“). Das Verfahren setzt sich für jeden Zeitschritt in gleicher Weise fort. Natürlich würden die so erzeugten Bäume beliebig wachsen. Deshalb wird nach einer festgelegten Anzahl von Zeitschritten nur noch der Teilbaum beibehalten, der den wahrscheinlichsten Pfad beinhaltet („pruning“). Dies ist durch die grün hinterlegten Felder gekennzeichnet.

Die vorgegebene Maximalhöhe der Bäume legt implizit auch die Verzögerung fest, mit der der MHT-Algorithmus auf Änderungen in der Wirklichkeit reagiert. Allerdings zeigt sich bereits bei relativ kleinen Höhen ein deutlicher Vorteil gegenüber klassischen Verfahren. In der Arbeit wird der Wert Drei für die maximale Höhe der Hypothesenbäume festgelegt. Mit der Scanfrequenz von 10 Hz ergibt sich eine Verzögerung von 0,3 s. Der MHT Tracker liefert nach dieser Zeitspanne jedoch wesentlich sicherere Annahmen. Durch die Kombination mit schnelleren Trackingverfahren ließe sich dieser Nachteil ausgleichen. Auch könnte die Höhe der Bäume, und damit die Verzögerung, den Gegebenheiten dynamisch angepasst werden. Bei niedriger Geschwindigkeit und hoher Szenen-Komplexität beispielsweise bietet sich eine größere Baum-Tiefe an. Theoretisch würde eine Reduzierung der Baumhöhen auf Eins einem vollständigen Verzicht auf die Auswertung mehrerer Hypothesen gleichkommen.

Algorithmus 3.3 beschreibt den groben Ablauf des MHT-Verfahrens. Kernstück des MHT-Algorithmus ist die Berechnung der Wahrscheinlichkeit, dass eine Messung z tatsächlich zu einer neuen Hypothese \hat{h}^- führt. Dazu nutzt man den Kalman-Filter in der Ausgangshypothese \hat{h} , um einen geschätzten Zustand des Objekts zum aktuellen Zeitpunkt zu erhalten (\hat{h}^-). Je näher die Messung an der geschätzten Position liegt, desto höher ist die Wahrscheinlichkeit, dass die Messung tatsächlich von diesem Objekt stammt.

Um diese Wahrscheinlichkeit zu quantifizieren, zieht man die Unsicherheit aus der Kalman-Filterung heran. Der Kalman-Filter aktualisiert nach jedem Vorhersageschritt die a priori Schätzfehler-Kovarianzmatrix P^- . Die ersten beiden Diagonalelemente geben im vorliegenden Fall die Varianzen für die x- und die y-Position an. Wenn man vereinfachend davon ausgeht, dass der Fehler in x- und y-Richtung unabhängig ist, lässt sich die zweidimensionale

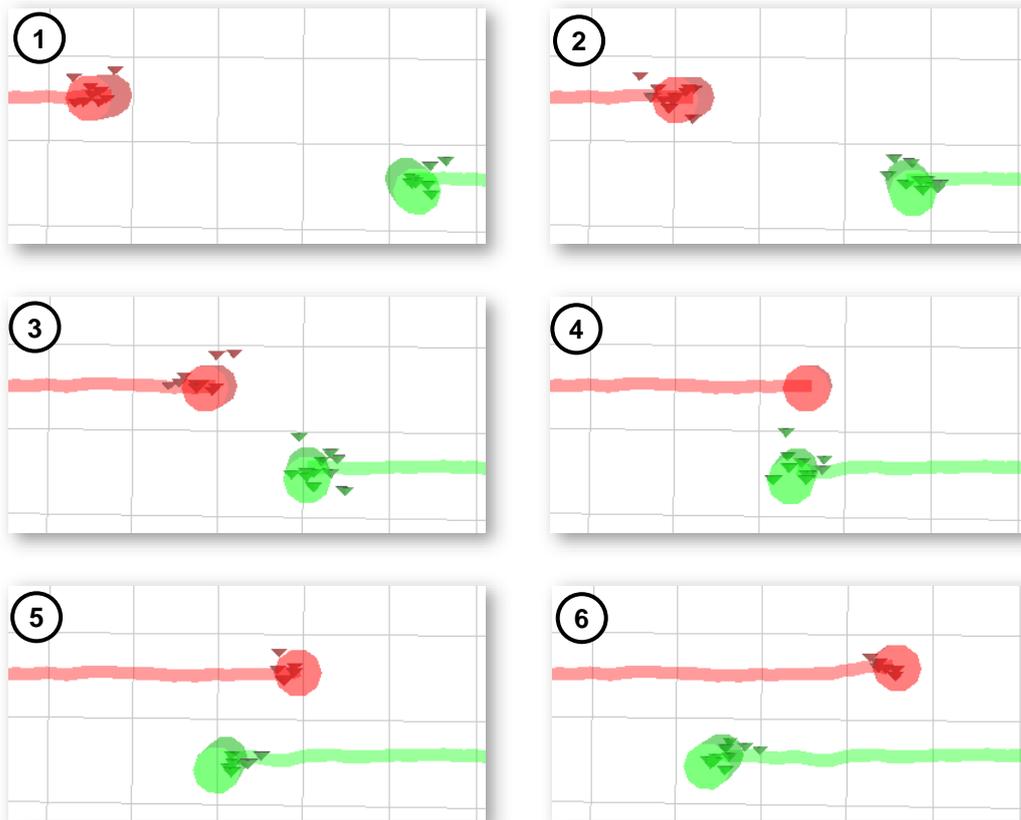


Abbildung 3.4: Tracking eines kurzzeitig verdeckten Fußgängers

Das Bild zeigt die Messpunkte von zwei Fußgängern, deren Weg sich kreuzt. In Schritt 4 liefert der Laserscanner vom rot markierten Fußgänger keine Messwerte, da dieser durch den anderen Fußgänger verdeckt wird. Die Trajektorie wird aber dennoch fortgesetzt.

Normalverteilung in folgender Form nutzen:

$$P(\hat{h}^- | z) \sim \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2} \left(\left(\frac{x_z - x_{\hat{h}^-}}{\sigma_x} \right)^2 + \left(\frac{y_z - y_{\hat{h}^-}}{\sigma_y} \right)^2 \right)} \quad (3.17)$$

Hierbei werden als Erwartungswerte die prädizierte Position des Kalman-Filters, $x_{\hat{h}^-}$ und $y_{\hat{h}^-}$, verwendet. x_z und y_z sind die Werte der aktuellen Messung z . Die Standardabweichungen ergeben sich aus der Quadratwurzel der beiden oberen Diagonalelemente von P^- .

Damit nicht unnötig viele Hypothesen generiert werden, gibt man für $P(\hat{h}^- | z)$ eine untere Schranke vor. Wird diese unterschritten, so geht man davon aus, dass die prädizierte Position zu weit von der gemessenen Position entfernt ist, um zum selben Objekt zu gehören. Es wird in diesem Fall keine neue Hypothese über die Messung z erzeugt.

Algorithmus 3.3 : Multi-Hypothesen Tracking

```

begin
  Interpretiere neue Messungen;
  /* Aktive Hypothesen (Blätter der Bäume) durch neue
     Folge-Hypothesen ergänzen */
  foreach Aktive Hypothese  $\hat{h}$  do
    Update Bewegungs-Modell von  $\hat{h}$  durch Kalman Vorhersage;
    foreach Messung  $z$  do
      Erzeuge a-priori-Hypothese  $\hat{h}^-$  aus  $\hat{h}$ ;
      Berechne Wahrscheinlichkeit  $P(\hat{h}^- | z)$ , dass  $\hat{h}^-$  aus  $\hat{h}$  hervorgeht unter
      Berücksichtigung von  $z$ ;
      if  $P(\hat{h}^- | z) > Schwelle$  then
        | Korrigiere Bewegungsmodell von  $\hat{h}^-$  durch Kalman Korrektur mittels  $z$ ;
      else
        | Verwerfe  $\hat{h}^-$  sofort;
      end
    end
  end
  /* Neue Hypothesen-Bäume für jede Messung erzeugen */
  foreach Messung  $z$  do
    | Erzeuge neue Hypothese  $\hat{h}$  aus  $z$ ;
  end
  Reduziere Hypothesen-Bäume auf die maximale Höhe;
  Interpretiere neue Wurzeln der Bäume als aktuellen Zustand;
end

```

3.3 Analyse der Bein-Pendelbewegungen

Sobald sich Fußgänger fortbewegen, zeigt sich eine typische Pendelbewegung der Beine. In [30] wird dieser Umstand bereits für die Analyse der FIR-Bilder verwendet. Doch auch beim Einsatz eines Laserscanners können diese Pendelbewegungen genutzt werden, um Fußgänger von anderen Objekten deutlich abzugrenzen. Siehe dazu auch [8]. Der folgende Abschnitt erläutert detailliert die angewandten Vorgehensweisen.

3.3.1 Extraktion der relevanten Merkmale

Zunächst werden einige Versuchsreihen gestartet um herauszufinden, ob eine Analyse der Pendelbewegungen im Laserscan überhaupt sinnvoll ist.

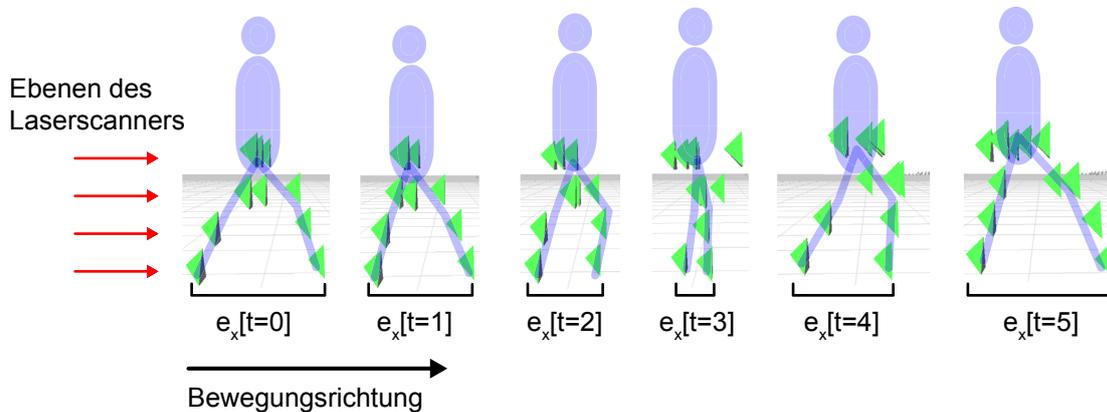


Abbildung 3.5: Messpunkte eines Fußgängers über mehrere Zeitschritte hinweg

Abbildung 3.5 zeigt eine zeitliche Abfolge realer Laserscan-Daten eines Fußgängers in der Seitenansicht. Jeder Messpunkt ist durch ein grünes Dreieck gekennzeichnet. In diesem Fall befindet sich der Fußgänger zu Beginn der Scan-Serie etwa 15 Meter vor dem Fahrzeug und entfernt sich dann in Fahrtrichtung vom Scanner. Zusätzlich sind, zum besseren Verständnis, in der Abbildung die ungefähren Positionen der Beine und des Torso manuell eingezeichnet³. Die Pfeile in der Darstellung verdeutlichen stark vereinfacht die Lage und Richtung der vier Scanebenen. Es fällt auf, dass die Pendelbewegungen durchaus noch erkennbar sind. Vorteilhaft ist in diesem Fall die niedrige Anordnung des Scanners am Fahrzeug. Die unterste Ebene des Scanners verläuft etwa parallel zum Boden. Die oberste Ebene streift den Fußgänger bei diesem Abstand etwa auf Höhe der Taille. Jedoch ist die Gesamtanzahl der Messpunkte, die einen Fußgänger in einem solchen Abstand ausmachen, bereits relativ gering. Verfahren wie beispielsweise in [29] beschrieben, gehen meist davon aus, die Beine im Laserscan trennen zu können. Dies erscheint mit der hier zur Verfügung stehenden Hardware nicht realistisch.

³Auf die restlichen anatomischen Details wie Arme wird hier verzichtet.

Zum einen ist die Auflösung des Scanners nicht hoch genug und zum anderen soll die Erkennung von Fußgängern - und auch deren Pendelbewegungen - noch in großem Abstand zum Fahrzeug möglich sein.

Der dargestellte Ansatz geht davon aus, dass die Segmentierung immer versucht „komplette“ Fußgänger zu trennen und nicht nur einzelne Beine. Dann ist es möglich, die Pendelbewegungen allein über die räumliche Ausdehnung eines Segments zu verfolgen. In Abbildung 3.5 ist deshalb am unteren Rand der Bildfolgen bereits die räumliche Ausdehnung des Segments e_x (für „extent“ in x-Richtung) angetragen. e_x ist die Ausdehnung des Segments bezüglich der untersten Scan-Ebene in x-Richtung. Die unterste Ebene zeichnet die Ausschläge naturgemäß am deutlichsten auf. Höhere Ebenen können unter Umständen auch verfälschte Werte liefern, falls z. B. schwingende Arme ebenfalls Reflexionen und dadurch Messpunkte verursachen.

Damit das Verfahren nicht auf Fußgänger, die sich genau vom Fahrzeug weg bewegen beschränkt bleibt, wird eine Verallgemeinerung vorgenommen. Die Schwingungs-Amplitude eines Segments ist in Bewegungsrichtung des Fußgängers am größten. Jedem getrackten Objekt wird deshalb ein lokales Koordinatensystem zugeordnet. Der Ursprung befindet sich im Schwerpunkt des Segments und berechnet sich durch die Mittelung aller zum Segment gehörenden Messpunkte. Im Falle von Fußgängern ist das etwa in der Mitte zwischen den beiden Beinen. Die x-Achse zeigt immer in Bewegungsrichtung des Objekts. In Abbildung 3.6 ist eine schematische Darstellung des lokalen Koordinatensystems zu sehen. x_L und y_L sind die lokalen Koordinaten. e_x ist die Ausdehnung des Segments in lokaler x-Richtung. x_w und y_w sind die stationären Weltkoordinaten. Es ist klar, dass ein Objekt zunächst getrackt werden muss, bevor die Parameter dieses Koordinatensystems bestimmbar sind. Dies ist jedoch kein Problem, da Fußgänger, die sich nicht bewegen, auch keine Pendelbewegungen aufweisen. Es ist auch nicht notwendig zu warten, bis der Kalman-Filter die korrekte absolute Geschwindigkeit ermittelt hat. Wichtiger ist die Bewegungsrichtung, welche wesentlich schneller nach dem Erzeugen eines neuen Tracks der Realität entspricht.

Sei der Geschwindigkeitsvektor eines Objekts n mit v_n und die Position im stationären Koordinatensystem mit p_n durch das Tracking gegeben. Dann lässt sich für die Transformation vom Welt-Koordinatensystem ins objekt-lokale System Folgendes schreiben:

$$\alpha = \arctan2(v_{ny}[t], v_{nx}[t]) \quad (3.18)$$

$$M_n[t] = \begin{pmatrix} \cos(-\alpha) & \sin(-\alpha) & 0 & 0 \\ -\sin(-\alpha) & \cos(-\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -p_{nx}[t] \\ 0 & 1 & 0 & -p_{ny}[t] \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.19)$$

Die Matrix M_n ist eine der in Abbildung 2.9 gezeigten Matrizen $M_1 \dots M_n$. Die Funktion „arctan2“ berechne den Arcustangens von v_{ny}/v_{nx} unter Berücksichtigung der Quadranten.

Vor dem Fahrzeug querende Fußgänger sind bedingt durch das Messprinzip des Scanners etwas schlechter zu analysieren. Die Winkelposition ist deutlich ungenauer als der gemessene Abstand zu einem Objekt. Dies gleicht die Fusion mit der FIR-Kamera jedoch aus. Pendelbewegungen von querenden Fußgängern sind im FIR-Bild relativ gut zu sehen, wohingegen

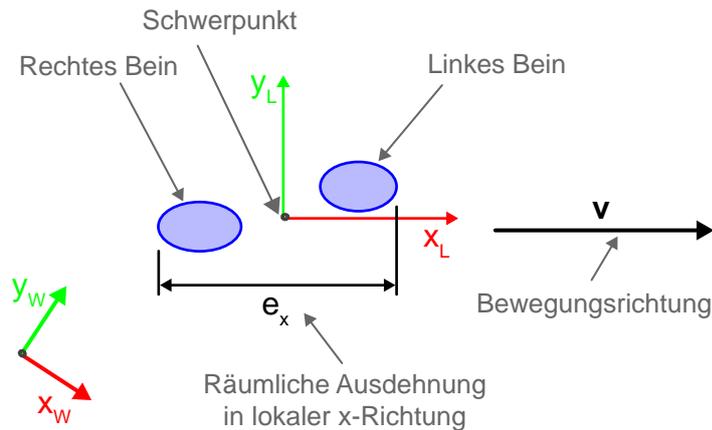


Abbildung 3.6: Ein Fußgänger und sein lokales Koordinatensystem in der Draufsicht

Pendelbewegungen von Fußgängern, die sich auf die Kamera zu oder von ihr weg bewegen (in Fahrtrichtung des Fahrzeugs), so gut wie überhaupt nicht sichtbar sind.

Eine Einschränkung ergibt sich, wenn das eine Bein im Schatten des anderen liegt. Grundsätzlich gilt aber, dass dieser Fall das Ergebnis nur dann negativ beeinflusst, wenn der Fußgänger gerade die Beine zum Schritt geöffnet hat. Denn bei geschlossenen Beinen ist die räumliche Ausdehnung eines einzelnen Beins (in Fortbewegungsrichtung) unwesentlich anders als die Ausdehnung beider Beine. Die Praxis zeigt, dass selbst bei geöffneten Beinen und ungünstigem Winkel zum Scanner eine komplette Verdeckung nur sehr selten auftritt.

3.3.2 Fourier-Analyse zur Frequenzbestimmung

Nachdem verdeutlicht wurde, welche Messgröße für eine Analyse der Pendelbewegungen relevant ist, geht es darum, zu bestimmen, wie sich aus dem zeitlichen Verlauf Rückschlüsse ziehen lassen, ob es sich bei einem bestimmten Objekt um einen Fußgänger handelt. Zunächst werden einige exemplarische Szenen in Matlab ausgewertet. Abbildung 3.7 auf Seite 45 zeigt den Verlauf von e_x über ca. 120 Scans hinweg. Dies entspricht bei einer Scanfrequenz von 10 Hz etwa 12 Sekunden Beobachtungsdauer.

Da die Pendelbewegungen einen periodischen Ablauf über die Zeit darstellen, bietet es sich an, eine diskrete Fourier-Transformation (DFT) zur Analyse zu verwenden. Dabei wird sowohl das Spektrum als auch das Signal diskretisiert betrachtet. Die DFT berechnet sich nach folgender Formel:

$$\tilde{X}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-jk(2\pi/N)n}, \quad k = 0, 1, \dots, N-1 \quad (3.20)$$

Die $x[n]$ sind dabei die N Werte des zeitdiskreten Signals. In diesem Fall wird das $e_x[t]$ sein.

In Abbildung 3.8 ist dann die Fourier-Transformation des gesamten Signals in den Frequenzbereich dargestellt. Da mit 10 Hz abgetastet wird, reicht die Frequenz-Skala der Transformation von 0 bis 5 Hz. Es ist ein deutlicher Peak bei etwa 1,7 Hz zu sehen. Diese Frequenz liegt in einem für Fußgänger typischen Bereich. Der Peak bei 0 Hz ist der Gleichanteil des Signals und wird nicht weiter berücksichtigt. Die Bedingung $f_{abstast} > 2 \cdot f_{max}$ des Abtast-Theorems ist ebenfalls eingehalten, da Frequenzen zwischen 1 und 3 Hz erwartet werden.

Natürlich ist es nicht erstrebenswert, einen Fußgänger über einen derart langen Zeitraum zu beobachten. Die Frequenz kann sich über die Zeit hinweg ändern und es ist eine möglichst schnelle Reaktion auf Fußgänger wünschenswert. Deshalb wird untersucht, ob auch die Betrachtung eines wesentlich kleineren Zeitfensters noch eine Aussage liefern kann. Dabei werden jeweils nur die letzten Werte eines Signals transformiert und neue Werte nach dem FIFO-Prinzip hinzugefügt. Für die kleinste auflösbare Frequenz f_0 gilt: $f_0 = \frac{1}{T}$. Nimmt man also für einen Fußgänger eine Minimal-Frequenz von einem Herz an, so muss das Zeitsignal mindestens eine Sekunde lang beobachtet werden, um Rückschlüsse auf das Spektrum ziehen zu können.

Nach einigen Versuchsreihen hat sich gezeigt, dass eine Transformation von 16 aufeinander folgenden Laserscans gute Ergebnisse liefert. Dies entspricht einer Beobachtungsdauer von 1,6 Sekunden. Eine Zweierpotenz wird gewählt, um den Einsatz einer Fast Fourier Transformation (kurz FFT) zu ermöglichen.

Abbildung 3.9 auf Seite 46 zeigt die Kurzzeitspektren einer FFT über jeweils 16 Werte des Zeitsignals aus Abbildung 3.7 auf der nächsten Seite. Da die FFT ein symmetrisches Spektrum liefert, sind lediglich 9 Frequenz-Werte aussagekräftig. Der Gleichanteil bei $f = 0$ wird in diesem Fall nicht mit eingezeichnet. Es ist nach wie vor ein deutlicher Ausschlag bei der typischen Frequenz zu sehen. Da der Ausschlag offenbar zu jedem Zeitpunkt signifikant hoch ist, wird auf die Anwendung einer Fensterfunktion (z. B. Hamming Window) verzichtet.

In Abbildung 3.10 wird das gleiche Verfahren auf ein Objekt, welches kein gehender Fußgänger ist, angewendet. Um die Vergleichbarkeit sicherzustellen, ist der gleiche Maßstab für die Amplitude und die Färbung gewählt. Außer einem normal verteilten Rauschen im Eingangssignal sind keine weiteren Ausschläge zu sehen.

Um entscheiden zu können, ob ein Spektrum auf einen Fußgänger hindeutet, wird ein Beobachtungsfenster zwischen einem und drei Herz definiert. Überschreitet in diesem Fenster ein lokales Maximum einen bestimmten Schwellwert, so wird dies als Indiz für einen Fußgänger gewertet.

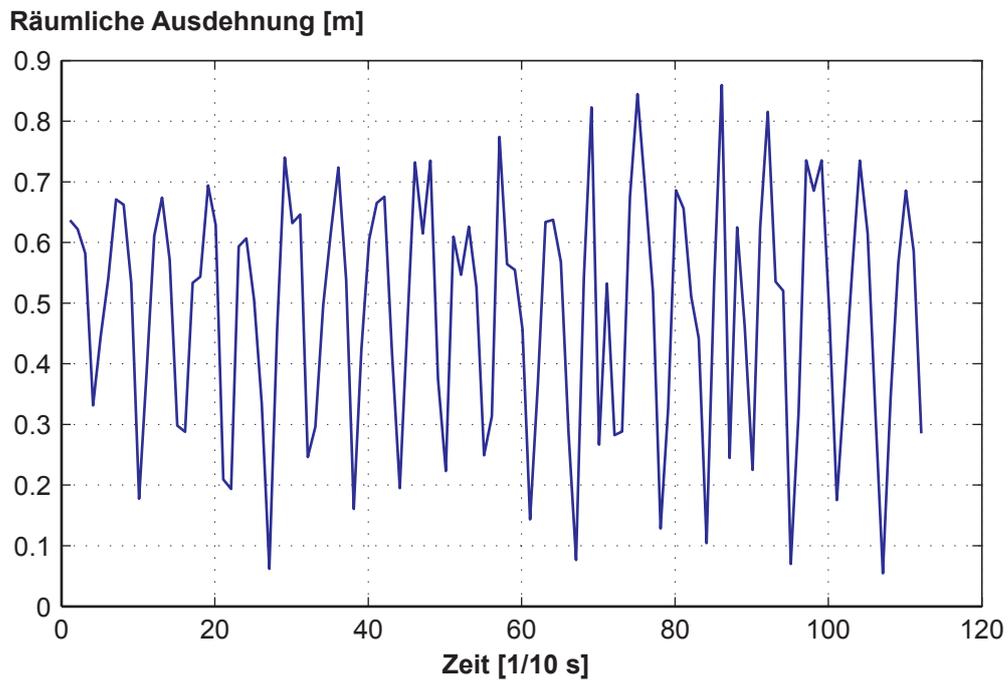


Abbildung 3.7: Die räumliche Ausdehnung eines Fußgänger-Segments

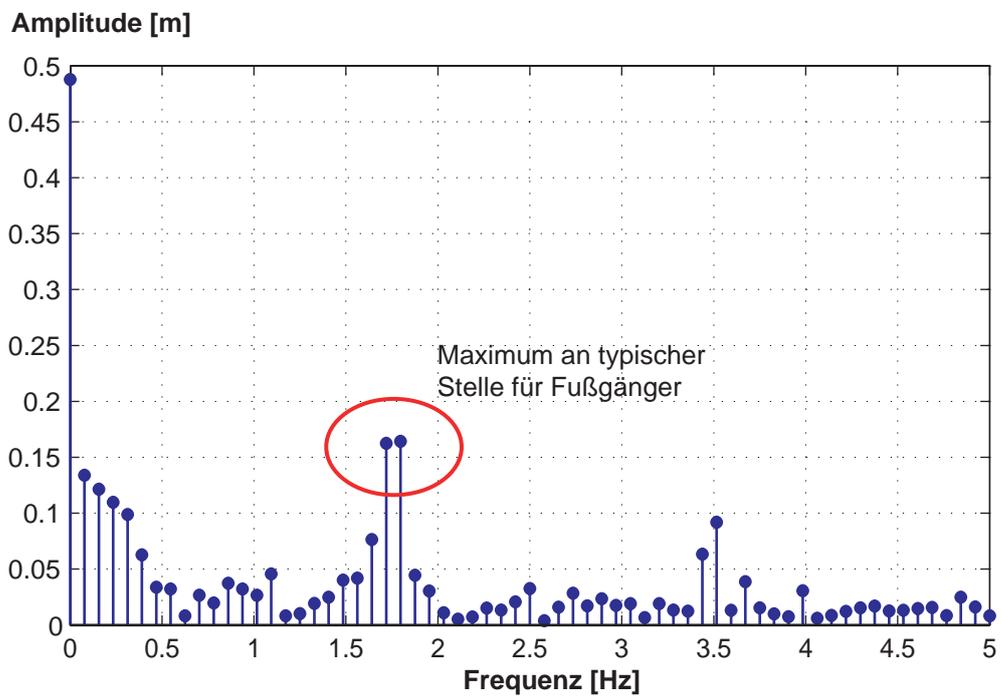


Abbildung 3.8: Diskrete Fourier-Transformation der räumlichen Ausdehnung eines Fußgänger-Segments

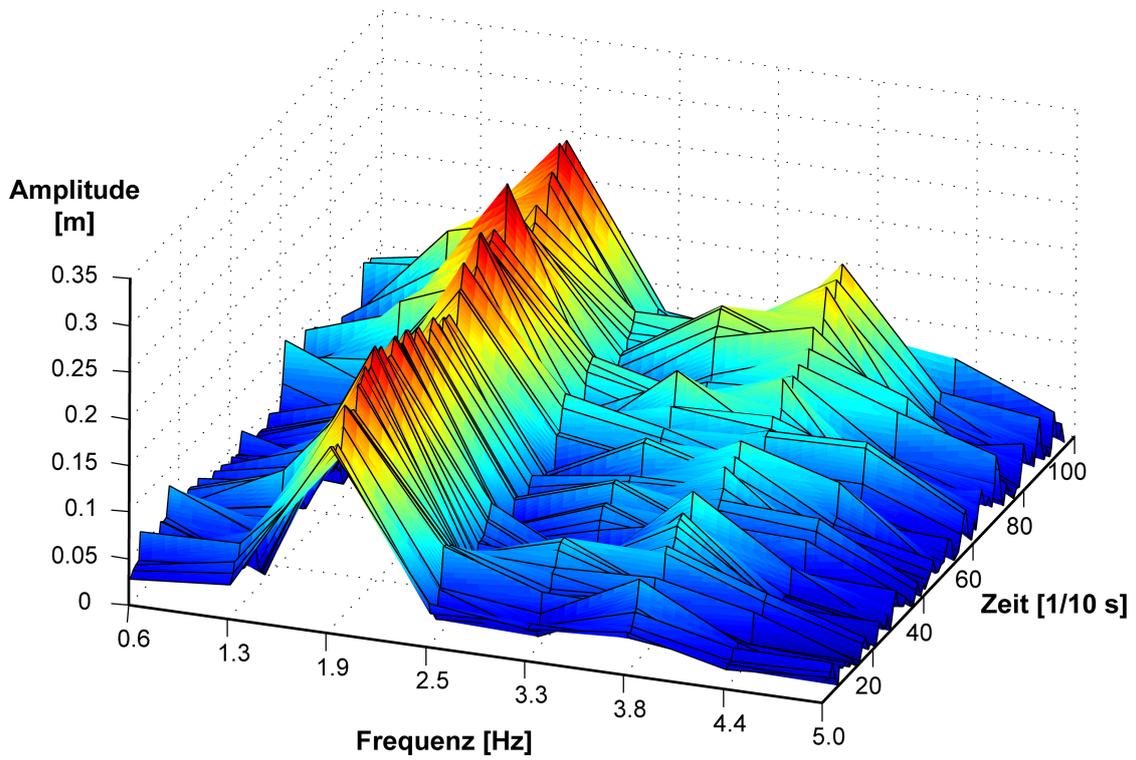


Abbildung 3.9: DFT Kurzzeitspektren der räumlichen Ausdehnung eines gehenden Fußgängers

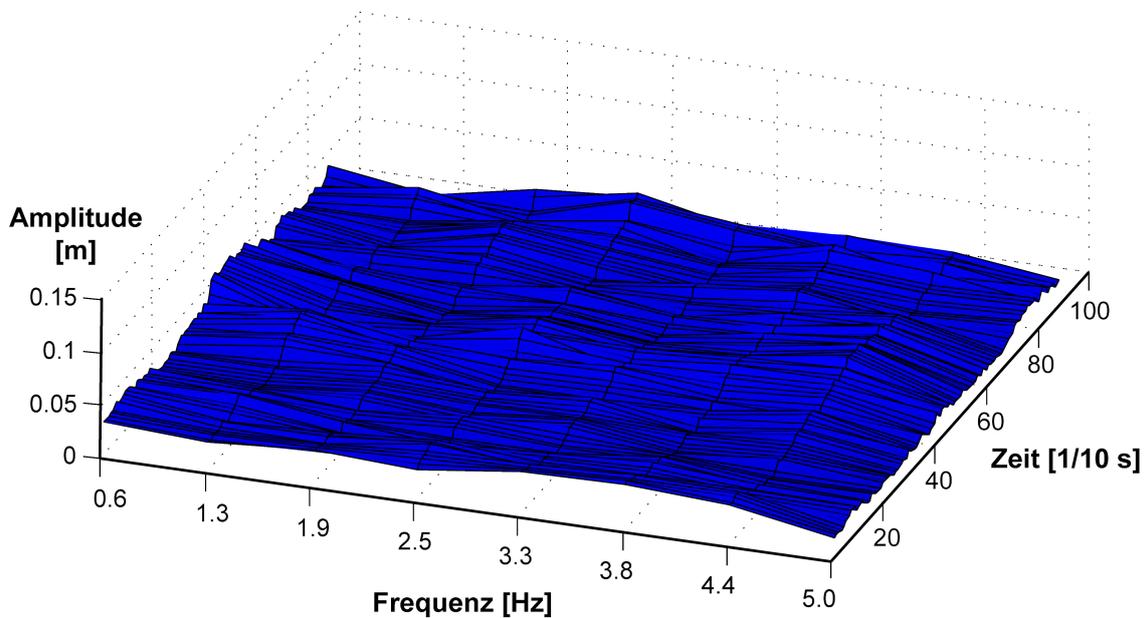


Abbildung 3.10: DFT Kurzzeitspektren der räumlichen Ausdehnung eines Objekts, welches kein gehender Fußgänger ist

KLASSIFIKATION UND FUSION

Das folgende Kapitel geht genauer auf den Prozess der Fusion und der Klassifikation der Objekte ein. Die in Kapitel 3 entwickelten Verfahren sind ein Teilaspekt in der gesamten Verarbeitungskette. Zusätzlich zu den Merkmalen aus der Verarbeitung des Laserscans kommen nun die Ergebnisse der Bildverarbeitung hinzu.

Zunächst gibt das Kapitel einen Überblick über die grundlegenden Abläufe während der eigentlichen Erkennung und Fusion. Anschließend wird detailliert erklärt, welche Merkmale letztendlich für die Klassifikation herangezogen werden. Den verwendeten Klassifikatoren und den Lernverfahren ist ebenfalls ein Abschnitt gewidmet.

Die Nummern ❶ bis ❹ beziehen sich in diesem Kapitel immer auf Abbildung 2.1 auf Seite 13.

4.1 Grundlegende Schritte

Zunächst wird mittels der in Abschnitt 2.7 erklärten Gleichungen die *aktuelle Position und Orientierung des Fahrzeugs* ❶ im stationären Welt-Koordinatensystem berechnet. Der *Laserscan* wird dann Punkt für Punkt über den Szenen-Baum vom lokalen Sensor-Koordinatensystem in das Welt-Koordinatensystem *transformiert*. Dadurch ermittelt das Multi-Hypothesen-Tracking bereits absolute Geschwindigkeiten.

Darauf folgt die *Segmentierung des Laserscans* ❷ und das Auffinden von teilverdeckten Segmenten. Die Referenz-Punkte der Segmente werden schon während des Segmentierungsvorgangs berechnet. Dies ist in diesem Fall das Mittel aller zum Segment gehörenden Punkte. Die räumliche Ausdehnung jedes Segments wird ebenfalls bereits während der Segmentierung berechnet. Die so erhaltenen Segmente durchlaufen eine erste *Filterung bezüglich ihrer Größe*, wobei sich „Größe“ in diesem Fall auf die räumliche Ausdehnung der Punktwolke in den drei Dimensionen bezieht. Segmente, die zu klein oder zu groß sind, um von einem

Fußgänger zu stammen, werden für das Tracking nicht weiter berücksichtigt. Dies ist auch notwendig, da dem Tracker ausschließlich ein auf Fußgänger passendes Bewegungsmodell zur Verfügung steht. Sollten auch andere Objekte (wie z. B. Fahrzeuge) vom Tracking berücksichtigt werden, so müsste an dieser Stelle eine Unterscheidung stattfinden. Dann könnten Multi-Modell-Verfahren, wie die in [14], Verwendung finden.

Die Segmente, welche die erste Filterung erfolgreich bestehen, dienen dem MHT als Eingabe, um *neue Objekt-Hypothesen* ③ zu generieren.

Der MHT liefert, mit der vorher festgelegten Verzögerung, *verifizierte Objekt-Hypothesen* (Tracks¹). Die Verarbeitung der LIDAR-Daten wird dann erneut auf den gefundenen Tracks gestartet. Für alle Objekte mit einer ausreichenden absoluten Geschwindigkeit kommt das in Abschnitt 3.3 beschriebene Verfahren zur Analyse der Bein-Pendelbewegungen zum Einsatz. Zunächst sind alle Punkte des Objekts in ein lokales Koordinatensystem zu transformieren ④. Da die Fourier-Analyse auf den Daten von 16 aufeinander folgenden Zeitschritten operiert, muss für jedes Objekt eine *Objekt-Historie*, also eine Liste mit vergangenen Zuständen, mitgeführt werden. In jedem Zeitschritt kommt ein neuer Zustand nach dem FIFO-Prinzip dazu. Sind ausreichend Werte vorhanden, ermittelt die *FFT* ⑤ das Spektrum des zeitlichen Verlaufs von e_x . Wird ein Maximum im Frequenz-Spektrum gefunden, so erhält das Objekt eine Markierung für den erfolgreichen *Nachweis einer Pendelbewegung*. Diese Markierung ist dann für die Klassifikation wichtig.

4.2 Zusammenspiel mit der Bilderkennung

4.2.1 Ermittlung der interessanten Bildbereiche

Liegen neue MHT-Objekte vor, so werden diese auch mit der Bildverarbeitung kombiniert. Dies geschieht, indem jedem Objekt eine ROI im FIR-Bild zugewiesen wird ⑥. Diese 2D-ROI kommt nach dem in Abbildung 4.1 auf der nächsten Seite gezeigten Schema zustande. Auch wenn in der Abbildung ein Fußgänger als Beispiel für die ROI-Berechnung dient, so sei darauf hingewiesen, dass bis jetzt natürlich noch keine Unterscheidung zwischen Fußgängern und anderen Objekten stattfindet. Dies leistet dann erst der Klassifikator am Ende des Prozesses. Die ROI im Bild ergibt sich durch die Projektion eines Quaders in das Kamera-Koordinatensystem. Dazu sind die acht Eckpunkte des Quaders getrennt zu projizieren. Dann kann um die so gewonnenen 2D-Punkte wieder ein umschließendes Rechteck als ROI gelegt werden. Zwei Annahmen über die Dimensionierung des Quaders, im Folgenden „Bounding Box“ genannt, werden getroffen:

Die Tiefe, Breite und xy-Position der Bounding Box wird durch die gefilterte räumliche Ausdehnung des Objekts und die gefilterte Position bestimmt. Jedem Objekt wird ein Lascan-Segment zugeordnet. Die räumliche Ausdehnung der zugeordneten Segmente

¹Die Begriffe „Track“ und „Objekt“ werden gleichbedeutend verwendet.

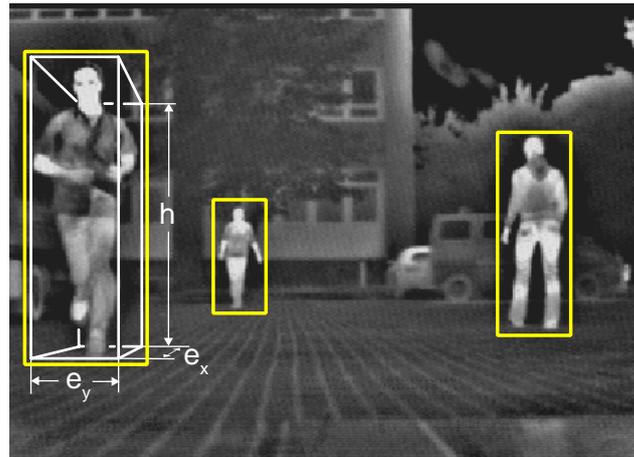


Abbildung 4.1: Berechnung der ROI (gelb) aus der räumlichen Ausdehnung eines Segments (weiß)

wird über die Zeit hinweg, wie in Abschnitt 3.2.1 beschrieben, gefiltert. Die Standardabweichung σ_e wird so gewählt, dass kurzzeitige Schwankungen in der Größe der Segmente keine Auswirkung auf die angenommene Größe des Objekts haben. Abbildung 4.2 zeigt den Verlauf der räumlichen Ausdehnung des jeweils zugeordneten Segments (ungefiltert). Die starken Schwankungen kommen durch Bein-Pendelbewegungen des Fußgängers zustande. Die etwas kleineren Ausschläge ergeben sich durch Ungenauigkeiten in der Messung des Laserscanners. Die Objektgröße, die für die Position der Bounding Box verwendet wird, ändert sich hingegen wesentlich träger. Gezeigt ist hier nur eine Dimension. Das Vorgehen ist in x- und y-Richtung analog.

Die Unterkante und die Höhe der Bounding Box kommt durch eine einfache Schätzung zustande, da die Ausdehnung in z-Richtung aus folgendem Grund nicht weiter verwendet wird: Weil der vertikale Öffnungswinkel des Laserscanners relativ klein ist (siehe Abschnitt 2.2), kann die Höhe eines Objekts nicht über den Laserscan geschätzt werden. Deshalb sei für die Höhe h ein Standardwert von 1,8 m angenommen.

Es ist auch keine Aussage über den Auftreffpunkt der Laserstrahlen auf einem Fußgänger möglich. Bedingt durch Nickbewegungen des Fahrzeugs kann der Laserscanner den Fußgänger höher oder tiefer über dem Boden „streifen“. Den Sachverhalt verdeutlicht auch Abbildung 2.8 auf Seite 22, welche die Auftreffpunkte der Laserstrahlen für eine Szene zeigt. Der Fußpunkt des Fußgängers wird deshalb auf der Höhe $z = 0$ im stationären Weltkoordinatensystem vermutet.

Da der Laserscanner und die FIR-Kamera am verwendeten Versuchsfahrzeug etwa einen Meter auseinander liegen, können die ermittelten ROIs im Bild Überlappungen aufweisen, da der Laserscanner unter Umständen Objekte „sieht“, die aus Sicht der FIR-Kamera verdeckt sind. Verdeckungen können auch auftreten, wenn der Tracker die Objekt-Position prädiziert, ohne eine aktuelle Lidar-Messung als Verifikation zu haben.

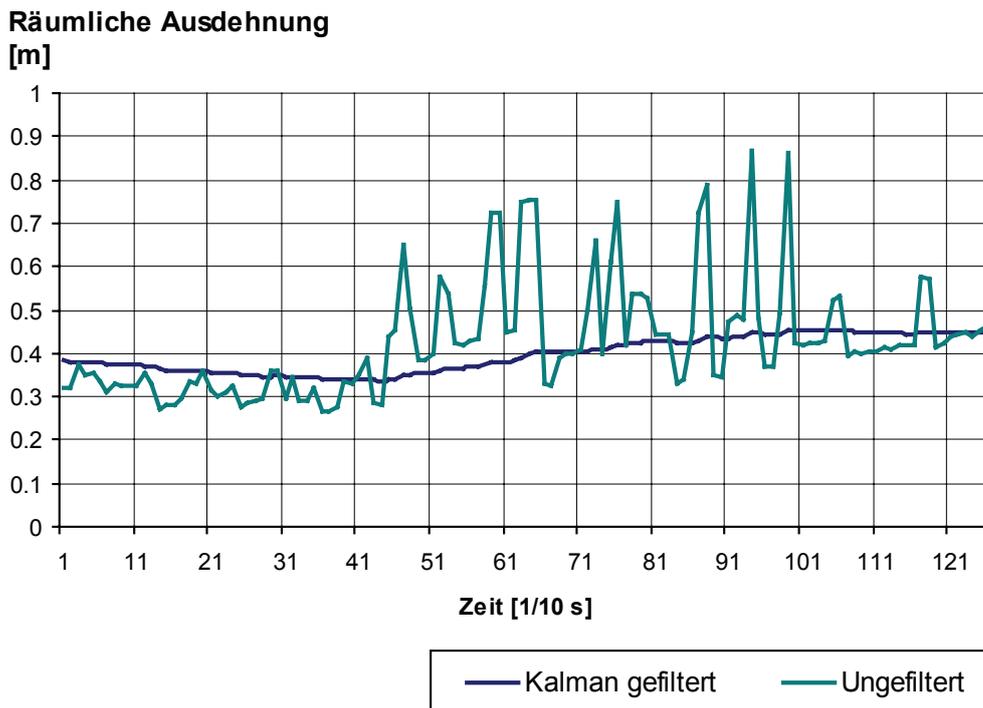


Abbildung 4.2: Vergleich der realen Ausdehnung eines Segments und der gefilterten Objektgröße

Deshalb findet anschließend eine *Sortierung der ROIs* ⑦ nach Tiefenwert statt. Die bezüglich dieser Sortierung weiter hinten liegenden ROIs sind zu verwerfen, sobald eine Verdeckung mit einer weiter vorne gelegenen ROI auftritt.

Fehlt dieser Schritt im Verarbeitungsprozess, so ist es möglich, dass Objekte im Hintergrund fälschlicherweise eine Klassifikation als Fußgänger über die FIR-Bildverarbeitung erhalten. Abbildung 4.3 auf der nächsten Seite zeigt eine Beispielszene, in der zwei Fußgänger aneinander vorbeigehen. Das Tracking verfolgt zunächst beide Objekte (gelbes und grünes Rechteck). Aufgrund der räumlichen Position zur Kamera tritt die ROI eines Fußgängers in den Hintergrund (graues Rechteck). Tritt keine Verdeckung mehr auf, können wieder beide ROIs zur FIR-Bildanalyse herangezogen werden.

Für die Kombination mit der Bilderkennung ⑧ werden zwei Varianten implementiert:

- Die Bildverarbeitung arbeitet zunächst unabhängig und liefert Rechtecke mit erkannten Fußgängern im FIR-Bild. Diese Rechtecke werden dann den vorher ermittelten Objekt-ROIs gegenübergestellt. Findet sich für eine Objekt-ROI ein Rechteck mit ausreichend hohem Überdeckungsgrad, erhält das Objekt eine Markierung für eine erfolgreiche Verifikation im FIR-Bild. Diese Situation ist in Abbildung 4.4 auf Seite 52 dargestellt. Der Vorteil dieser Methode liegt darin, dass die Bildverarbeitung nicht angepasst werden muss. Unabhängig vom eingesetzten Verfahren für die Fußgängererkennung im FIR-Bild ist eine gute Kombination der gefundenen Objekte möglich.

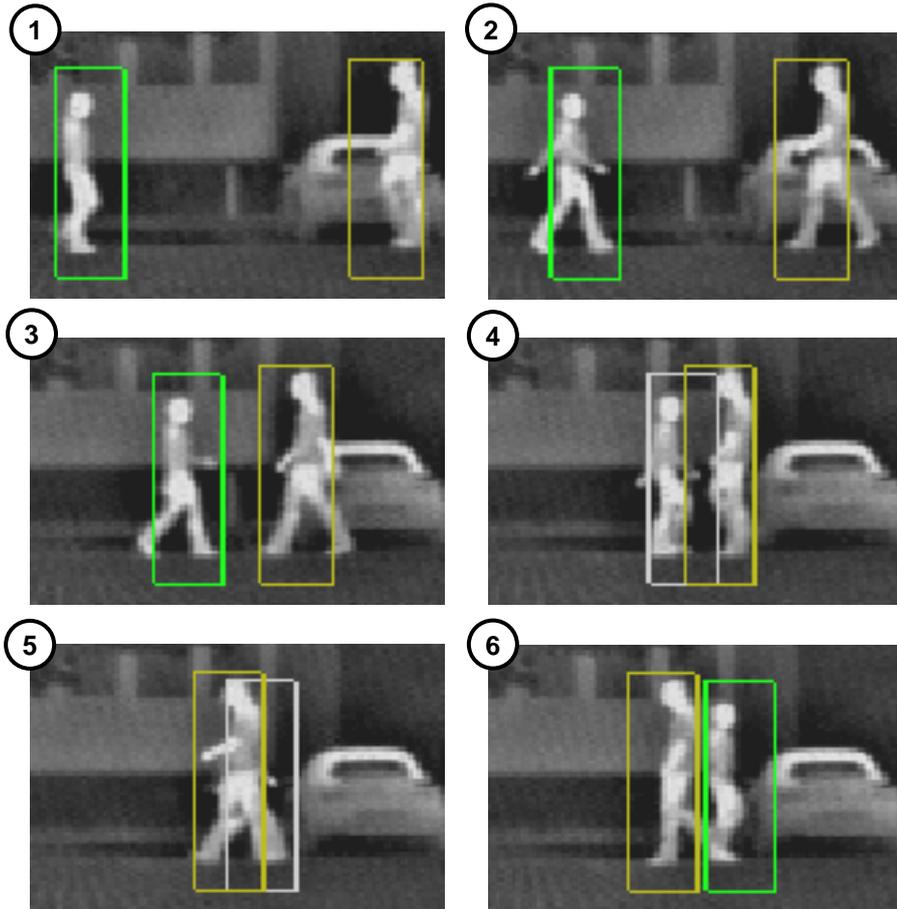


Abbildung 4.3: Beispielszene zur Ermittlung der Tiefenreihenfolge der ROIs
Gezeigt sind zwei Fußgänger, deren Weg sich vor dem Fahrzeug kreuzt. In den Bildern 4 und 5 tritt die ROI des einen Fußgängers kurzzeitig in den Hintergrund. Dies wird durch den grauen Rahmen verdeutlicht.

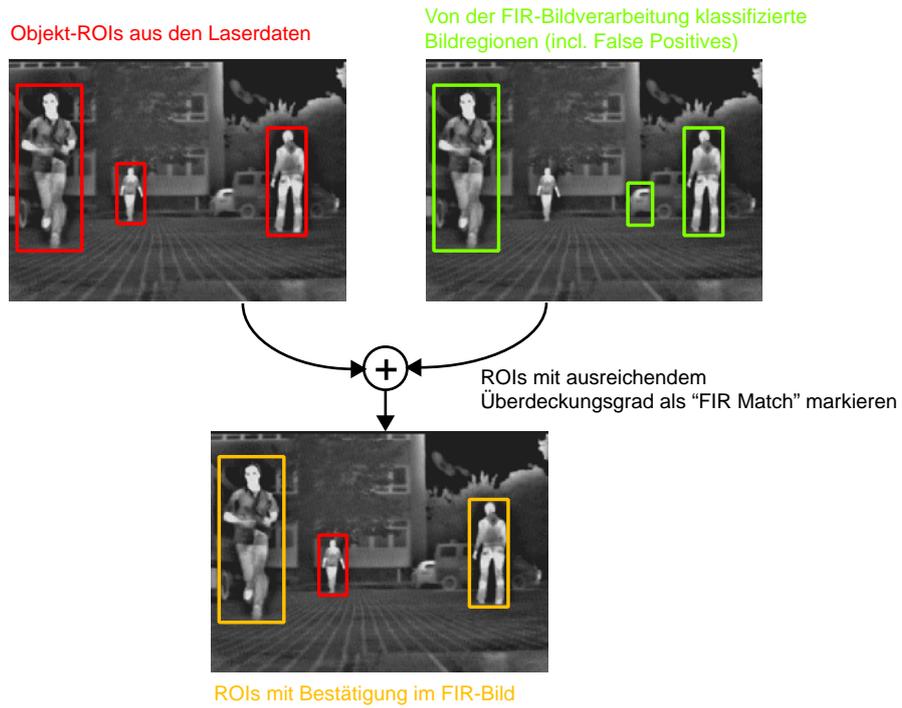


Abbildung 4.4: Kombination der ROIs aus den Laserdaten und der unabhängig gefundenen FIR-Bildregionen

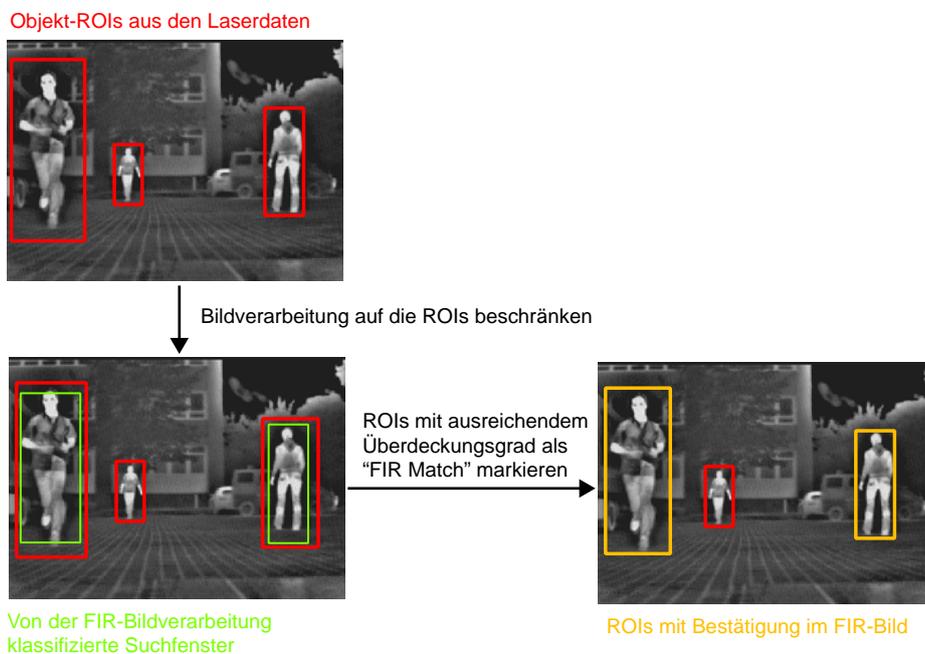


Abbildung 4.5: Beschränkung der Bildverarbeitung auf die ermittelten ROIs aus den Laserdaten

- Die Bildverarbeitung beschränkt sich von Anfang an auf die ermittelten Objekt-ROIs. Dies reduziert die zu untersuchende Datenmenge im FIR-Bild in vielen Fällen erheblich. Dieses Vorgehen ist in Abbildung 4.5 auf der vorherigen Seite veranschaulicht. Der Nachteil bei dieser Methode ist, dass die Erkennungsleistung der Bildverarbeitung unter Umständen stark von der Art und Weise abhängt, wie die ROI im Bild gewählt wird (z. B. Position oder Pixelabstand zum Objekt). Deshalb wird der in [19] beschriebene Adaboost-Klassifikator angepasst.

4.2.2 Anpassung des Adaboost-Klassifikators

Die so erzeugten ROIs werden dazu genutzt, um den in [19] verwendeten OpenCV-Algorithmus zu beschleunigen, welcher in der Original-Version das gesamte Bild durchsucht.

Der ursprüngliche Ansatz sieht einen kompletten Verzicht auf die *Suche* nach Fußgängern im FIR-Bild vor. Es sollten lediglich die vom Laserscanner gelieferten ROIs dahingehend untersucht werden, ob sie einen Fußgänger enthalten. Es zeigt sich jedoch, dass dies mit nur einem Durchlauf des Klassifikators nicht möglich ist. Das liegt vor allem daran, dass die Wahl der Region, die vom Klassifikator untersucht werden soll, starken Einfluss auf das Ergebnis der Klassifikation hat. Die optimale Position ist nicht abhängig von der tatsächlichen Pixelgröße des Objekts im Bild, sondern vielmehr ein Resultat der verwendeten Trainingsdaten, welche wiederum von der Markierung der verwendeten Szenen abhängen („Labeling“).

Dieser Umstand führt dazu, dass innerhalb der ROIs nach wie vor nach Fußgängern *gesucht* werden muss. Es ergibt sich aber immer noch ein signifikanter Vorteil gegenüber der Suche im ganzen Bild. Damit der Adaboost-Klassifikator Ergebnisse liefert, muss die ROI „künstlich“ vergrößert werden. Abbildung 4.6 auf der nächsten Seite zeigt zwei ROIs (rot). Es ist zu sehen, dass die Höhe deutlich über der Fußgängergröße im Bild liegt. Dies begründet sich durch die Unsicherheit bezüglich der Höhe und der z-Position des Objekts. Die Toleranz in x-Richtung im Bild wird experimentell auf einen sinnvollen Wert festgelegt.

In der Abbildung bezeichnet „Suchfenster“ die Größe des Ausschnitts, der ursprünglich für das Training verwendet wird. Dieser Ausschnitt wird nun in definierten Schritten über die gesamte ROI „geschoben“ und der Klassifikator ausgeführt. Um verschiedene Skalierungen zu berücksichtigen, erfolgt dies in mehreren Durchgängen mit verschiedenen Skalierungsfaktoren. Der Bereich, in dem dieser Faktor variiert wird, lässt sich mit Hilfe der vorhandenen ROI deutlich besser eingrenzen.

Zur Veranschaulichung sei ein kurzes Beispiel gegeben:

Der Klassifikator aus [19] wird mit Ausschnitten trainiert, die alle auf 15×35 Pixel skaliert werden. Ohne weitere Informationen wird die Suche erst abgebrochen, wenn die Größe des Suchfensters die Größe des Bildes erreicht hat. In diesem Fall variiert der Faktor im Bereich $s \in [1; \min(\frac{240}{35}, \frac{320}{15}) \approx 6,9]$. Nimmt man nun eine realistische ROI von 53×89 Pixeln, so kann der Faktor nach oben begrenzt werden: $s \in [1; \min(\frac{89}{35}, \frac{53}{15}) \approx 2,5]$. Doch auch nach unten kann der Faktor stärker eingegrenzt werden. Denn wenn in einer relativ großen ROI ein

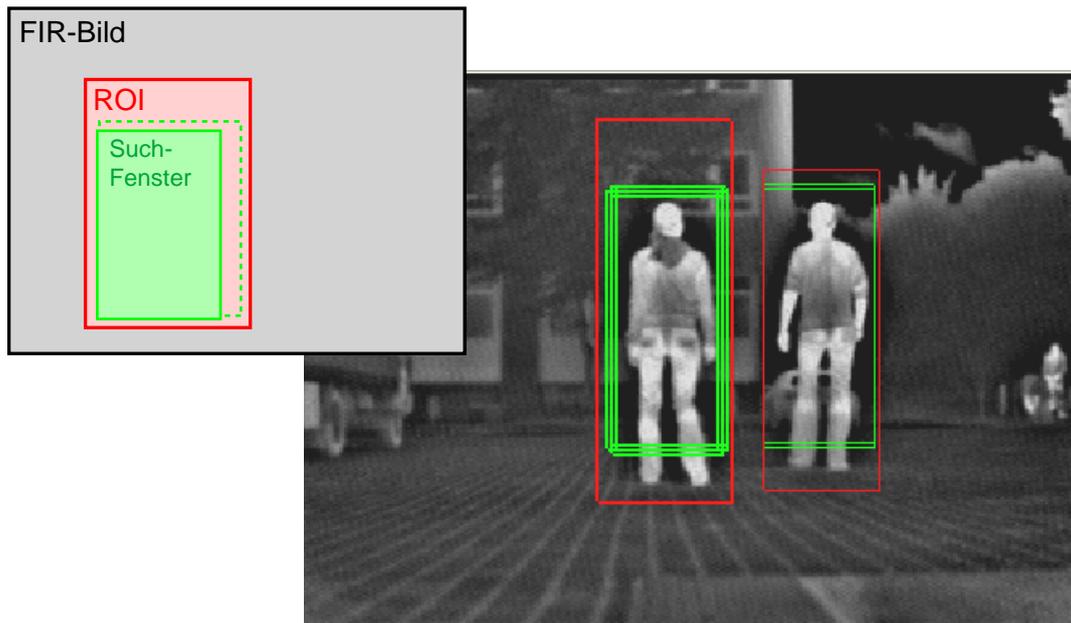


Abbildung 4.6: FIR-Bildverarbeitung mit Hilfe der ROIs aus den LIDAR-Daten

sehr „kleiner“ Fußgänger gefunden wird, so handelt es sich in den meisten Fällen um eine Falschmeldung („false positive“).

Wie in Abbildung 4.6 zu sehen ist, werden mehrere Fenster als potentielle Fußgänger erkannt (grüne Rechtecke). Die ursprüngliche Version des OpenCV-Algorithmus fasst diese dann noch zu gemittelten Rechtecken zusammen. Auf diesen Schritt wird aus Effizienzgründen verzichtet.

4.3 Klassifikation

Dieses Kapitel geht nun genauer auf die *Klassifikation* ⑨ der Objekte ein. Die in den vorhergehenden Schritten ermittelten Merkmale werden jetzt in einem Merkmalsraum kombiniert und eine Klassifikationsstufe darauf ausgeführt. Es werden zwei Verfahren implementiert:

Bayes'sches Netz: Um rasch Ergebnisse zu erhalten, wird zunächst ein passendes Bayes'sches Netz erstellt und die Wahrscheinlichkeiten durch Expertenwissen belegt.

Entscheidungsbaum: In Referenz-Szenen werden alle MHT-Tracks mit einer Markierung versehen („Labeln“). Die Features dieser Tracks dienen dann als Basis, um einen Entscheidungsbaum zu erstellen.

4.3.1 Extraktion der Features

Da für beide Verfahren die verwendeten Merkmale identisch sind, werden diese hier nochmals kurz zusammengefasst. Um die Merkmale in den Ausführungen leichter benennen zu können, werden kurze Namen vergeben:

Feature	Beschreibung	Typ
e_x	Die Kalman gefilterte räumliche Ausdehnung des Objekts in x-Richtung in [m].	double
e_y	Die Kalman gefilterte räumliche Ausdehnung des Objekts in y-Richtung in [m].	double
v	Die Kalman gefilterte Geschwindigkeit im stationären Koordinatensystem in [m/s].	double
OscillatingExtent	Ein Indikator, der das Ergebnis der FFT widerspiegelt. <i>True</i> , falls ein Maximum im Spektrum an entsprechender Stelle gefunden wird, ansonsten <i>false</i> .	boolean
FIRMatch	Ein Indikator, der angibt, ob für ein Objekt eine Fußgänger-Wärmesignatur im FIR-Bild gefunden werden konnte.	boolean

Wie bereits erwähnt, sieht der Ansatz in dieser Arbeit eine Feature-Level-Fusion vor. Somit geht das Ergebnis der Klassifikation im FIR-Bild wieder als Merkmal in den abschließenden Klassifikations-Prozess ein.

4.3.2 Klassifikation mit Bayes'schem Netz

Bayes'sche Netze stellen eine Möglichkeit dar, auch bei unsicheren Eingangsdaten Entscheidungen zu treffen. In unserem Fall ist dies also die Entscheidung darüber, ob es sich bei einem Objekt mit den beobachteten Merkmalen (Features), um einen Fußgänger handelt oder nicht. Nach [16] wird das System durch eine n-dimensionale diskrete Zufallsvariable X_1, \dots, X_n repräsentiert.

Um später eine einfachere Auswertung des Netzes zu ermöglichen, wird allen Merkmalen eine binäre Zufallsvariable zugeordnet. Die beiden Merkmale „FIRMatch“ und „OscillatingExtent“ sind bereits binär, da sie nur die Werte „wahr“ oder „falsch“ annehmen können. Jedoch sind die Merkmale e_x , e_y und v nicht diskret. Deshalb werden vorab experimentell Bereiche festgelegt, die den kontinuierlichen Wertebereich aufteilen in „für Fußgänger typisch“ und „für Fußgänger untypisch“.

Folgende Tabelle zeigt die Zuordnung von Features zu binären Zufallsvariablen:

Binäre Zufallsvariable	Wahr, wenn gilt...
HasCharacteristicExtent	$(0,2 \leq e_x \leq 1,2) \wedge (0,2 \leq e_y \leq 1,2)$
HasCharacteristicVelocity	$(0,8 \leq v \leq 1,9)$
HasOscillatingExtent	OscillatingExtent=true
HasFIRMatch	FIRMatch=true

Abbildung 4.7 zeigt das verwendete Bayes'sche Netz. Jede Zufallsvariable wird als Knoten in einem gerichteten Graph dargestellt. Die Pfeilspitzen zeigen nach [16] zu den Knoten, über deren Verteilungsdichte eine Aussage getroffen wird. Die Pfeile beginnen an dem Knoten, dessen Wert in die Verteilungsdichte des Ziels eingeht. Der oberste Knoten „IsPedestrian“ ist die Zufallsvariable, die angibt, wie wahrscheinlich es ist, dass ein Fußgänger vorliegt. Letztendlich soll, aufgrund der Werte der anderen binären Zufallsvariablen, eine Aussage über den Wert des Knotens „IsPedestrian“ getroffen werden.

Dies geschieht durch die Instantiierung des Netzes, indem die Werte der binären Zufallsvariablen entsprechend der Beobachtung in die Knoten eingetragen werden. Wird beispielsweise eine Pendelbewegung erkannt, so ergibt sich dadurch $P(HasOscillatingExtent) = 1$.

Um die Wahrscheinlichkeiten im Netz zu ermitteln, wird der ursprünglich in [21] vorgestellte Propagations-Algorithmus verwendet. Die Implementierung des Algorithmus stammt aus [18].

Folgende Wahrscheinlichkeiten werden experimentell bestimmt:

	IsPedestrian=true	IsPedestrian=false
$P(HasCharacteristicExtent)$	0,85	0,2
$P(HasCharacteristicVelocity)$	0,85	0,2
$P(HasOscillatingExtent)$	0,9	0,05
$P(HasFIRMatch)$	0,9	0,05

$P(HasOscillatingExtent|IsPedestrian = true)$ gibt beispielsweise die Wahrscheinlichkeit an, mit der sich Pendelbewegungen detektieren lassen, unter der Voraussetzung, dass es sich

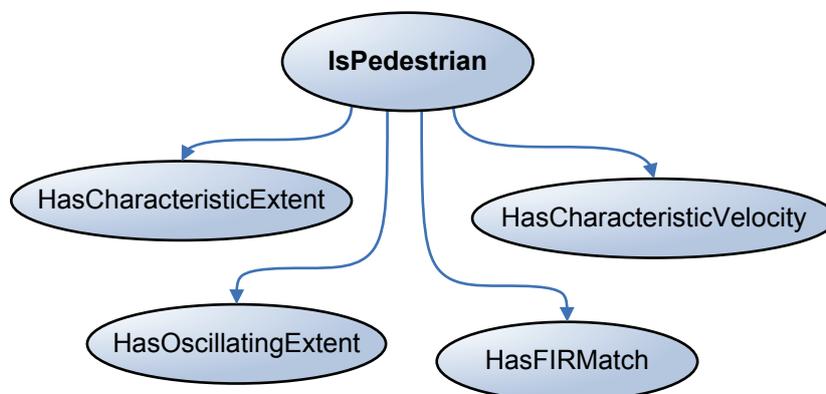


Abbildung 4.7: Bayes'sches Netz für die Klassifikation

um einen Fußgänger handelt (bedingte Wahrscheinlichkeit).

4.3.3 Klassifikation durch Entscheidungsbäume

Entscheidungsbäume² stellen eine Möglichkeit dar, um von einer Menge an Eingabeattributen auf eine Klassenzugehörigkeit zu schließen. Der Name deutet bereits an, dass die Entscheidung, um welche Klasse es sich handelt, in mehreren Schritten in einer baumartig verzweigten Struktur getroffen wird. Dabei zerlegt jeweils ein Knoten im Baum den Merkmalsraum anhand eines Attributs in disjunkte Teilbereiche. Bei diskreten Attributen ist eine Entscheidung direkt durch Aufspaltung nach den Werten des Attributs möglich (z. B. *FIRMatch=true/false*). Bei kontinuierlichen Attributen kann diese Aufspaltung durch Einführen von Vergleichswerten herbeigeführt werden (z. B. Geschwindigkeit > 0,3 m/s).

Markieren der Daten und Generierung des Trainingsmaterials

Um mit den später vorgestellten Algorithmen einen entsprechenden Entscheidungsbaum zu erstellen, sind zunächst Trainingsdaten erforderlich. Dazu werden exemplarisch Referenzszenen manuell ausgewertet und den Objekten, die der MHT-Tracker liefert, entsprechende Markierungen zugewiesen³. Da das Ziel der Arbeit lediglich in der Abgrenzung von Fußgängern zu anderen Objekten liegt, gibt es zwei Typen von Markierungen: „Ped“ (kurz für „Pedestrian“) und „NonPed“ (kurz für „Non-Pedestrian“).

Die Markierungen werden über eine grafische Benutzeroberfläche erstellt und dann in einer Datenbank zusammen mit der Szene abgespeichert. In der Datenbank ist für jeden Frame der Szene eine Liste mit Objekt-Positionen und Objekt-Klassen hinterlegt. Die Objekt-Positionen werden bezüglich des Laserscanner-Koordinatensystems abgespeichert. Dies ist notwendig, damit bei einer nachträglichen Änderung des Fahrzeugmodells, und damit einer anderen Fahrzeugbahn, die Markierungsdatenbank nicht verworfen werden muss. Die Relativ-Positionen der Objekte zum Fahrzeug bleiben von Änderungen im Fahrzeugmodell unbeeinflusst. Nach der Zuweisung der Markierungen können die Szenen erneut, diesmal ohne Benutzereingriff, abgespielt werden. Die vorher abgelegten Positionen werden mit den aktuellen Objekten verglichen und ggf. die Markierungen wieder erzeugt. Alle Objekte mit Markierung dienen dann zur Erzeugung des Trainingsmaterials. Tabelle 4.2 auf der nächsten Seite zeigt einen Ausschnitt aus den so gewonnenen Trainingsdatensätzen. v , e_x , e_y , *OscillatingExtent* und *FIRMatch* sind nicht-kategorische Attribute. Die Spalte *Klasse* gibt das einzige kategorische Attribut an.

²engl.: Decision Trees

³engl.: labeling

v [m/s]	e_x [m]	e_y [m]	Osc.Extent	FIRMatch	Klasse
0.011	0.435	0.343	false	false	NonPed
0.006	0.233	0.184	false	false	NonPed
0.005	1.184	1.067	false	false	NonPed
1.062	0.357	0.351	false	true	Ped
1.188	0.448	0.338	true	false	Ped
0.043	0.541	0.844	false	false	NonPed
0.007	0.435	0.344	false	false	NonPed
0.002	0.234	0.183	false	false	NonPed
0.004	1.184	1.068	false	false	NonPed
1.100	0.360	0.353	false	true	Ped
1.200	0.450	0.337	true	true	Ped
0.022	0.545	0.844	false	false	NonPed
0.003	0.435	0.346	false	false	NonPed
...					

Tabelle 4.2: Beispiel eines Trainingsdatensatzes für den Entscheidungsbaum

Lernverfahren

Das Trainingsmaterial dient nun dazu, einen Entscheidungsbaum zu erstellen, welcher sowohl bekannte als auch unbekannte Datensätze möglichst zuverlässig der richtigen Klasse zuordnen kann. Ross Quinlan beschreibt dazu in [23] das rekursive Verfahren ID3 (Iterative Dichotomiser 3).

Sei T die Menge aller Trainingsdatensätze. T wird in Teilmengen C_j bezüglich des kategorischen Attributs zerlegt. In jeder Teilmenge ist das kategorische Attribut aller Datensätze gleich. Man definiert die Entropie der Verteilung bezüglich des kategorischen Attributs folgendermaßen:

$$I(T) = - \sum_{j=1}^k \frac{|C_j|}{|T|} \log \frac{|C_j|}{|T|} \quad (4.1)$$

Intuitiv stellt die Entropie ein Maß für den Informationsgehalt einer Verteilung dar. Je höher der Informationsgehalt bezüglich des kategorischen Attributs ist, desto höher muss auch der Informationsgehalt der nicht-kategorischen Attribute sein, um Datensätze korrekt zuordnen zu können.

Das Vorgehen bei ID3 ist nun so, dass man die Menge aller Datensätze aufgrund des Wertes eines nicht-kategorischen Attributs X in Teilmengen T_j zerlegt. Innerhalb der Teilmenge T_j ist der Wert für dieses Attribut immer gleich. Dies führt zu einer Art bedingten Entropie

$I(X, T)$:

$$I(X, T) = \sum_{j=1}^m \frac{|T_j|}{|T|} I(T_j) \quad (4.2)$$

Diese bedingte Entropie ist ein Maß dafür, wie hoch die Entropie der durch die Teilung bezüglich X entstehenden Teilmengen T_j ist. Dazu bildet man das gewichtete Mittel über alle Entropien.

Der Algorithmus versucht Entscheidungen durch nicht-kategorische Attribute X mit sehr kleinem $I(X, T)$ möglichst weit oben im Baum zu platzieren. Wenn $I(X, T)$ klein ist, bedeutet dies, dass die „Rest“-Entropie in den Teilmengen sehr klein ist. Das wiederum spricht dafür, dass das gewählte Attribut X gut mit der zuzuordnenden Klasse korreliert. Das Maß, wie gut ein nicht-kategorisches Attribut die Menge aller Trainingsdatensätze partitioniert, lässt sich folgendermaßen angeben:

$$\text{Gain}(X, T) = I(T) - I(X, T) \quad (4.3)$$

Algorithmus 4.1 zeigt den schematischen Ablauf des ID3-Algorithmus.

Algorithmus 4.1 : ID3-Trainings-Algorithmus für Entscheidungsbräume

Input : R : Menge von nicht-kategorischen Attributen;

C : Kategorisches Attribut;

S : Trainingsdatensatz

Result : Entscheidungsbaum

begin

 /* Behandlung der Sonderfälle */

if IstLeer(S) **then**

 | **return** Fehler-Knoten;

else if IstLeer(R) **then**

 | **return** Knoten mit der Klasse, die am häufigsten in S vorkommt;

end

let D = Das Attribut mit dem größten $\text{Gain}(D, S)$ unter den Attributen in R ;

let $\{d_j | j = 1, 2, \dots, m\}$ die verschiedenen Werte von D ;

let $\{S_j | j = 1, 2, \dots, m\}$ die Einträge aus S , die den Wert d_j für das Attribut D aufweisen;

 /* Rekursiver Aufruf mit der partitionierten Menge S und
 entferntem Attribut D */

 ID3($R - \{D\}, C, S_1$), ID3($R - \{D\}, C, S_2$), ..., ID3($R - \{D\}, C, S_m$);

return Baum mit der Wurzel D und den Ästen $d_1 \dots d_m$;

end

Da die Auswertung des $\text{Gain}(X, T)$ Attribute X bevorzugt, die sehr viele verschiedene Werte annehmen können (im Extremfall in jedem Trainingsdatensatz unterschiedlich), verwendet

man anstatt des *Gain* auch den sog. *GainRatio*:

$$GainRatio(X, T) = \frac{Gain(X, T)}{SplitInfo(X, T)} \quad (4.4)$$

$$SplitInfo(X, T) = I\left(\frac{|T_1|}{|T|}, \frac{|T_2|}{|T|}, \dots, \frac{|T_m|}{|T|}\right) \quad (4.5)$$

So werden Attribute, die die Menge T in sehr viele gleichgroße Teilmengen zerlegen, benachteiligt.

Der C4.5-Algorithmus, ebenfalls in [23] erklärt, beschreibt einige Erweiterungen zu ID3. Unter anderem wird das Verfahren so ergänzt, dass es mit kontinuierlichen Attributen umgehen kann. Anders als bei den Bayes'schen Netzen soll hier keine manuelle Aufteilung des Wertebereichs erfolgen.

Um eine optimale Schranke zu finden, betrachtet man zunächst alle Werte $x_1 \dots x_m$ eines kontinuierlichen Attributs X im Trainingsmaterial T . Man teilt für jedes x_j die Menge T in Elemente mit Werten kleiner oder gleich x_j und solchen größer x_j auf. Daraufhin wird der *GainRatio* der so erzeugten Partitionierungen untersucht. Bei maximalem *GainRatio* hat man eine passende Schranke zur Aufteilung des kontinuierlichen Wertebereichs gefunden.

IMPLEMENTIERUNG

Ein wesentlicher Teil der Arbeit besteht in der Implementierung der Verfahren und in der Integration der vorhandenen Routinen und Bibliotheken. Um die Vorgänge innerhalb der Software veranschaulichen zu können, ist eine entsprechende Visualisierung unabdingbar. Das folgende Kapitel gibt Aufschluss über die internen Strukturen der erstellten Software und bietet Interessierten einen Überblick über die eingebundenen Bibliotheken.

5.1 Architektur des Systems

Ausgehend von der Anforderung, dass die Software später auch ohne aufwändige Visualisierung im Fahrzeug lauffähig sein muss, wird eine strikte Trennung zwischen der Darstellungsebene und der Berechnungsebene angestrebt. Des Weiteren ist von Interesse, eine transparente Umschaltung zwischen aufgezeichneten Daten und Live-Daten zu erreichen. Abbildung 5.1 auf der nächsten Seite zeigt ein Blockdiagramm mit den wichtigsten Komponenten des Systems. Eine kurze Beschreibung gibt jeweils zusätzliche Informationen zu den jeweiligen Funktionen. Die umgesetzte Architektur basiert auf dem Model/View/Controller-Schema [4]. Dabei stellt das Fusions-Modul das Modell dar, welches die Ansicht über Änderungen im Datenbestand informiert („Observer Pattern“).

5.1.1 Visualisierung

Die Visualisierungs-Komponente besteht unter anderem aus einem leicht wiederverwendbaren Rahmen zur Darstellung von dreidimensionalen Szenen und Videos (linke Seite in Abbildung 5.2). Hinzu kommen einige, speziell für diese Arbeit benötigte, Benutzerschnittstellen. Dazu zählen beispielsweise das Zuweisen von Markierungen zu den gefundenen Tracks oder das Umschalten von verschiedenen Anzeigeeoptionen (rechte Seite in Abbildung 5.2).

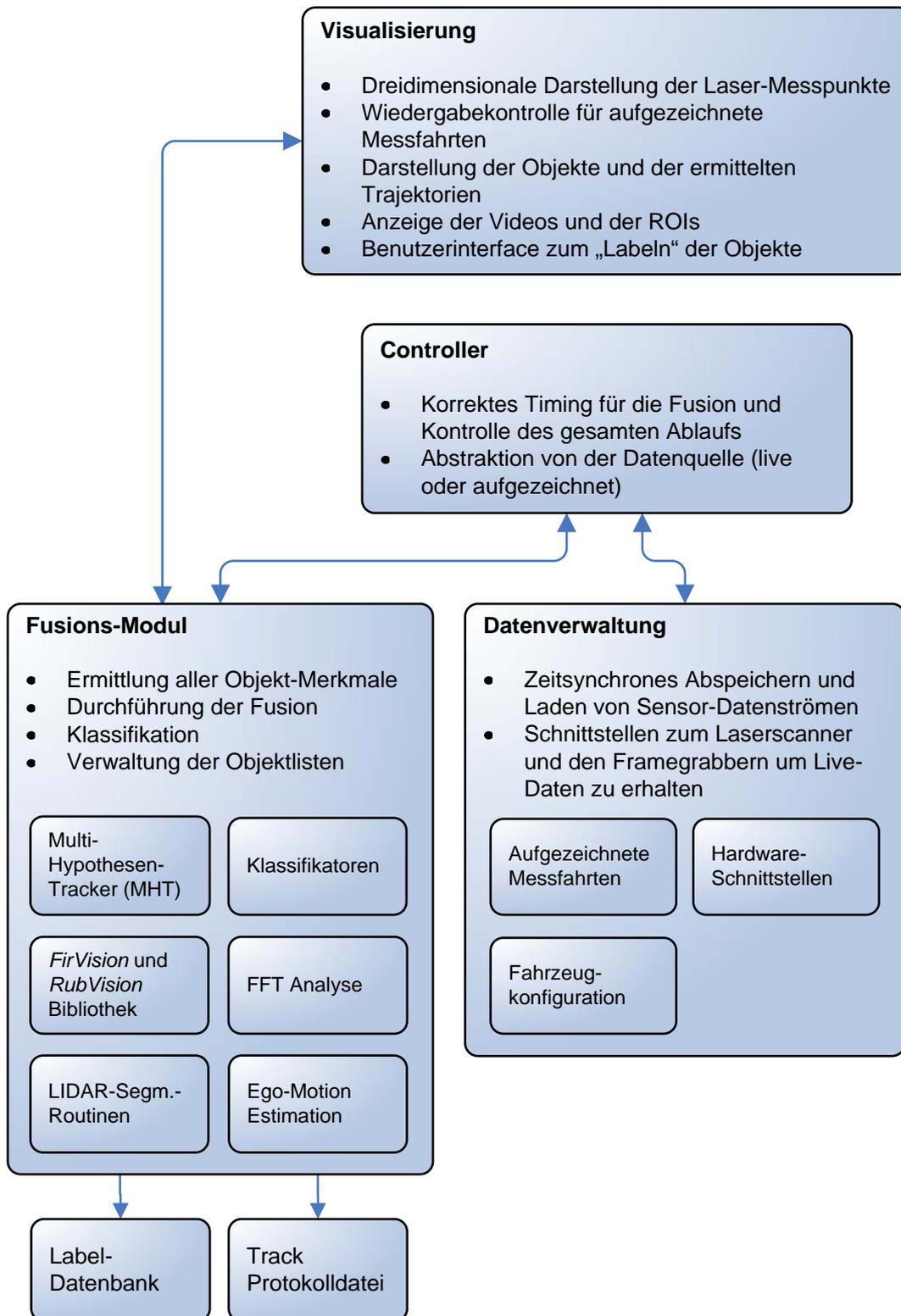


Abbildung 5.1: Darstellung der logischen Verknüpfung der Komponenten des Systems

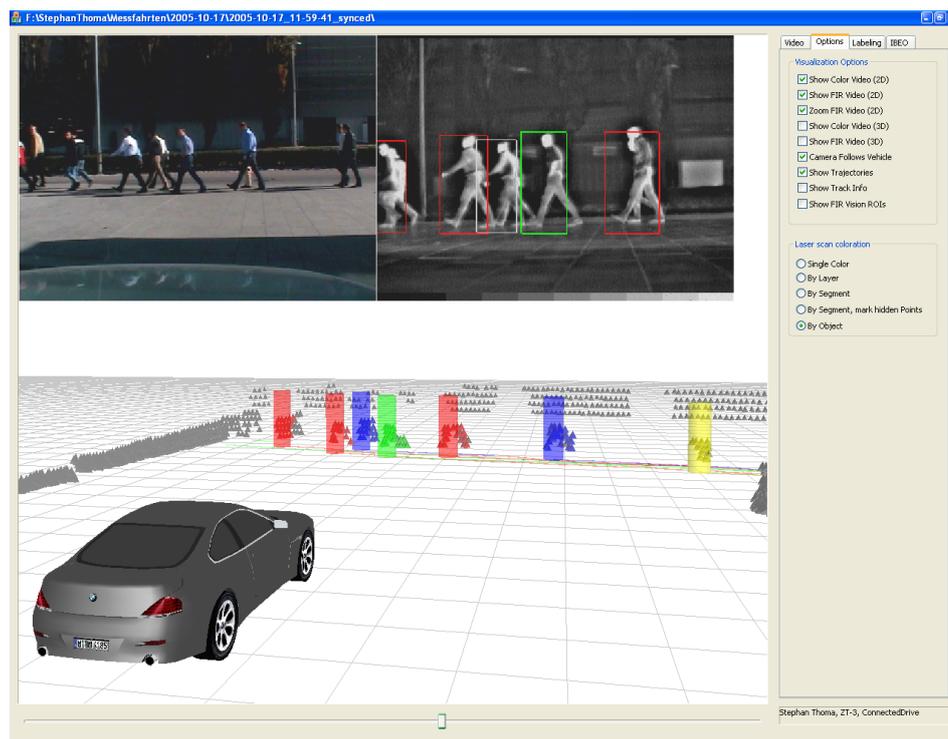


Abbildung 5.2: Bildschirmabdruck der Visualisierungs-Komponente

Da der Rahmen für die dreidimensionale Darstellung auch in anderen Projekten bei der BMW Forschung und Technik GmbH eingesetzt wird, ist darauf zu achten, die Schnittstelle so einfach und so generisch wie möglich zu halten. Realisiert wird ein wiederverwendbarer Rahmen (C3dFrame), der in andere Anwendungen als Dialogobjekt eingefügt werden kann. Abbildung 5.3 auf der nächsten Seite zeigt ein UML Klassendiagramm des umgesetzten Entwurfs. Es gibt eine generische Klasse für ein Zeichnungselement (CItem) und mehrere spezielle Implementierungen. Die Zeichnungselemente können in jeder Kombination in den Rahmen eingefügt werden. Dargestellt werden die Elemente dann über die Microsoft Direct3D-Schnittstellen¹. Einige Zeichnungselemente seien hier kurz erwähnt:

CPointList stellt eine Liste aus $[x,y,z]$ -Tupeln als Punktwolke im dreidimensionalen Koordinatensystem dar. Diese wird in dieser Arbeit beispielsweise für die Anzeige der Lasermesspunkte verwendet. Unterschiedliche Farben kennzeichnen, je nach Anwendungsfall, verschiedene Segmente, Objekte oder Scanebenen.

CVideoFrame ermöglicht die effiziente Darstellung von Videobildern auf dem Bildschirm. Unterstützt wird entweder eine überlagerte zweidimensionale Darstellung oder eine Projektion auf beliebige Flächen innerhalb der dreidimensionalen Szene. Zusätzlich können weitere zweidimensionale Zeichnungselemente über das Videobild gelegt werden. Dies wird beispielsweise zur Anzeige von ROIs oder erkannten Objekten im Bild genutzt.

¹<http://www.microsoft.com/windows/directx>

CTrajectory dient zur Darstellung von Objekt-Trajektorien. Es wird eine halbtransparente Darstellung unterstützt, sodass beispielsweise die Gitternetzlinien des Koordinatensystems nicht komplett verdeckt werden.

CCarMesh wird in diesem Projekt zur Darstellung des eigenen Fahrzeugs genutzt, kann jedoch auch zur Darstellung von anderen Verkehrsteilnehmern herangezogen werden.

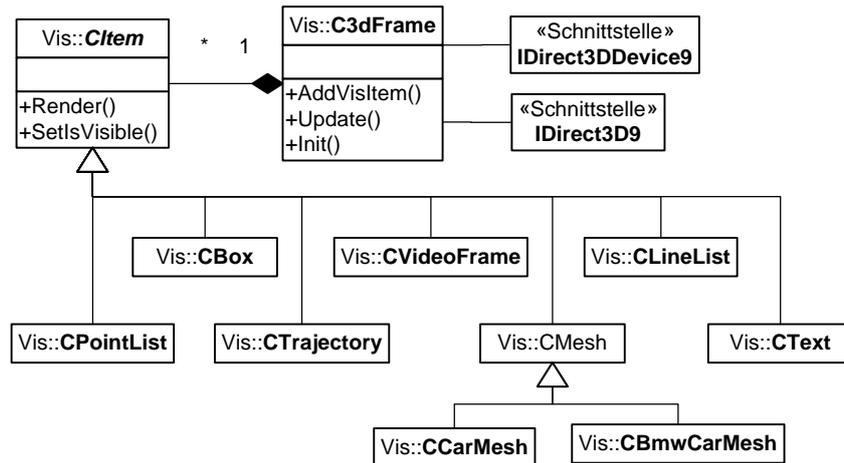


Abbildung 5.3: UML Klassendiagramm der Visualisierungskomponente

5.1.2 Fusions-Modul und Controller

Das Fusions-Modul implementiert zusammen mit dem Controller den in Abbildung 2.1 auf Seite 13 dargestellten Fusions-Prozess. Abbildung 5.4 zeigt einen vereinfachten Datenstrom der Sensoren in einer zeitlichen Abfolge. Zunächst fällt auf, dass nur für jeden dritten Kamera-Frame ein Laserscan zur Verfügung steht. Das liegt an den unterschiedlichen Frequenzen, mit denen Kamera und Laserscanner Daten liefern. Die Kamera sendet etwa alle 33 ms ein Bild. Der Laserscanner hingegen benötigt für eine komplette Messung 100 ms. Eine Fusion wird in dieser Arbeit nur dann durchgeführt, wenn beide Sensoren Daten liefern.

Zusätzlich ist bei der Realisierung darauf zu achten, dass die Daten der Sensoren immer zum richtigen Zeitpunkt fusioniert werden. Dabei sind alle Verzögerungen, die während der Verarbeitung auftreten, zu berücksichtigen. In der Abbildung ist zunächst die Verzögerung durch den MHT gezeigt. Erst wenn dieser Objekte liefert, kann eine Fusion stattfinden. Es ist deshalb notwendig, eine bestimmte Anzahl von FIR-Bildern und Laserscans in einem Ringpuffer abzulegen, um auch auf „ältere“ Daten zugreifen zu können. Die Fourier-Analyse benötigt noch weiter in der Vergangenheit liegende Messdaten, denn eine Analyse erstreckt sich über einen Zeitraum von 1,6 Sekunden.

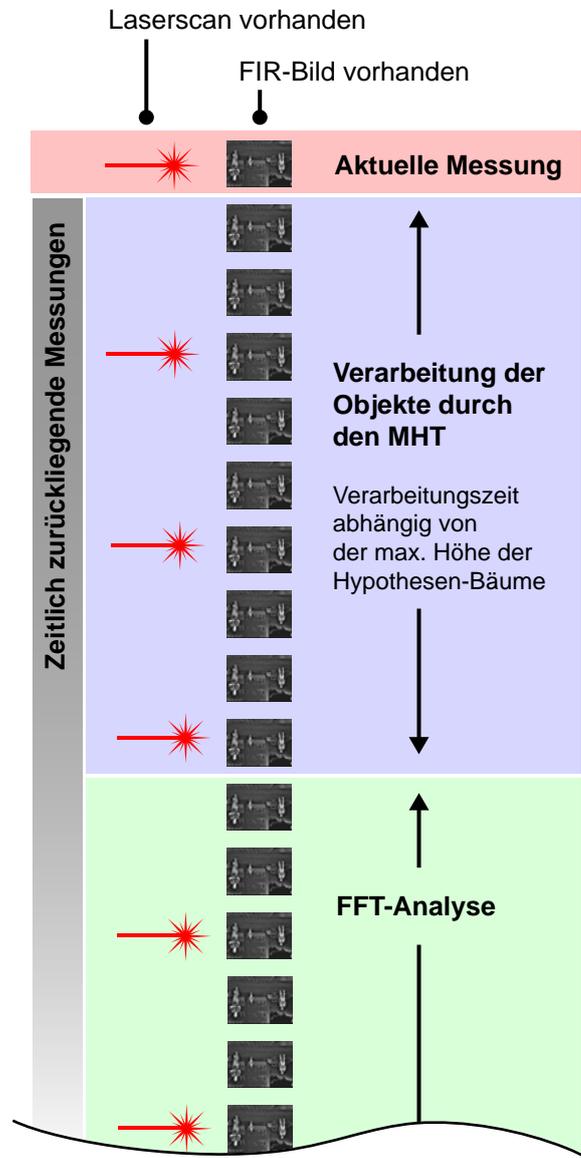


Abbildung 5.4: Zeitliche Abfolge der Datenverarbeitung im Fusions-Modul

5.1.3 Datenverwaltung und -aufzeichnung

Zur Aufzeichnung der Daten wird ein vorhandenes Aufnahmeprogramm so angepasst, dass auch eine Protokollierung der Messdaten des Laserscanners möglich ist. In diesem Fall diene die FIR-Kamera als Takt für die Aufzeichnung. In jedem Zyklus wird in einer Protokolldatei vermerkt, welche Sensoren Messdaten geliefert haben, und wie groß die Datenmenge ist. Die Nutzdaten der Sensoren werden sequentiell in eine separate Datendatei geschrieben. Dies hat den Vorteil, dass keine feste Datenrahmengröße pro Zyklus festgelegt werden muss. Die Daten können dadurch kompakt sequentiell auf die Festplatte geschrieben werden. In der Protokolldatei hingegen wird in jedem Zyklus ein neuer Eintrag hinzugefügt. Dieser Eintrag enthält Verweise auf die Byte-Positionen innerhalb der Datendatei.

Beim Einlesen der Daten kann zunächst das gesamte Protokoll in den Speicher geladen werden. Die eigentlichen Sensordaten liegen dann in sequentieller Anordnung in der Datendatei. So vermeidet man ein zufälliges Zugriffsmuster auf der Festplatte, denn bei den Aufzeichnungen in dieser Arbeit waren ca. 15 MB/s nötig.

5.2 Vorhandene Bibliotheken

Die Realisierung des Projekts wäre ohne der Vielzahl an bereits verfügbaren Bibliotheken nicht möglich gewesen. So werden in vielen Bereichen vorhandene Komponenten verwendet und angepasst. Die nachfolgende Tabelle gibt Aufschluss über die verwendeten Bibliotheken und die notwendigen Anpassungen.

Komponente	Erklärung
Visualisierung	Für die Darstellung der Messpunkte, der Fahrzeug- und Objekt-Trajektorien sowie der Klassifizierungs-Ergebnisse wird ein vollständig neues Frontend mit den <i>Microsoft Foundation Classes</i> (MFC) entwickelt (siehe [17]). Die 3D-Darstellung ist über die <i>Direct3D</i> -Schnittstelle realisiert (siehe [11]).
Tracking	Das in [6] entwickelte <i>Multi-Hypothesen Framework</i> dient als Grundlage für den in dieser Arbeit verwendeten Tracker. Die Implementierung stellt alle Grundfunktionen zur Verfügung, die zur Implementierung eines eigenen Trackers notwendig sind (Track-Trees verwalten, Hypothesen generieren, ...). Um alle speziellen Dinge, wie beispielsweise die Kalman-Filterung, muss sich der Entwickler selbst kümmern. Der Quellcode des Frameworks stammt aus dem GTMS-Projekt ² .

²<http://sourceforge.net/projects/gtms>

Komponente	Erklärung
Kalman-Filter	Die in Abschnitt 3.2.1 beschriebenen Kalman-Filter werden mittels der Bibliothek <i>OpenCV</i> implementiert ³ . Diese stellt Funktionen zur Verfügung, die das Verfahren nach [32] umsetzen.
Entscheidungsbäume	Für die Klassifikation mittels Entscheidungsbäumen wird die in [25] entwickelte Bibliothek <i>YaDT (Yet another Decision Tree builder)</i> ⁴ verwendet. Die Implementierung setzt das Verfahren in Anlehnung an den C4.5-Algorithmus aus [23] um. Ein Kommandozeilen-Programm kann zum Trainieren der Bäume und zur Auswertung genutzt werden. Leider liegt diese Bibliothek nicht im Quellcode vor.
Bildverarbeitung	Die Kaskaden aus [19] stützen sich auf die Bibliothek <i>OpenCV</i> ab. Das Verfahren nach [33] ist eine spezielle Implementierung in C++.
Matrizen-Operationen	Die gesamte Matrizen-Arithmetik (abgesehen von der Kalman-Filterung) wird mit Hilfe der UBLAS-Klassen aus der <i>Boost-Bibliothekensammlung</i> ⁵ realisiert.
Bayes'sche Netze	Der Propagations-Algorithmus nach [21] wurde ursprünglich in [18] implementiert und in dieser Arbeit wiederverwendet.
XML-Parser	Zum Einlesen und Abspeichern von XML-Dateien wird der von Kyung-min Cho entwickelte XML-Parser <i>XMLite</i> ⁶ verwendet.

³<http://sourceforge.net/projects/opencvlibrary/>

⁴<http://www.di.unipi.it/~ruggieri/>

⁵<http://www.boost.org/>

⁶<http://www.codeproject.com/cpp/xmlite.asp>

BEWERTUNG UND ANALYSE DER ERGEBNISSE

Die bisher vorgestellten und implementierten Verfahren sollen in diesem Kapitel auf ihre Tauglichkeit hin untersucht werden.

Die folgenden Aspekte werden in diesem Kapitel untersucht:

- **Genaue Analyse der Verfahren im Hinblick auf den Abstand zum Fahrzeug**
Um abschätzen zu können, wie schnell auf einen Fußgänger reagiert werden kann, ist es notwendig, zu wissen, ab welchem Maximalabstand die Verfahren zur Bildererkennung und zur Verarbeitung des Laserscans brauchbare Ergebnisse liefern. Anhand einer Beispielszene wird untersucht, ab welcher Entfernung das Tracking das Objekt verfolgt und Pendelbewegungen detektierbar sind. Außerdem werden die beiden in dieser Arbeit verwendeten FIR-Bilderkennungsverfahren bezüglich ihrer Erkennungsleistung untersucht.
- **Untersuchung der Rechenzeit**
Damit eine Aussage getroffen werden kann, ob die Implementierung für den Echtzeiteinsatz geeignet ist, wurde anhand von komplexen Szenen das Laufzeitverhalten der einzelnen Komponenten unter die Lupe genommen.
- **Beurteilung der Klassifikation**
Um zu beurteilen, wie gut die verwendeten Klassifikatoren arbeiten, wurden einige Referenzszenen untersucht und die Trefferrate analysiert. Die für Abschnitt 4.3.3 vorgenommenen Markierungen in den Szenen können auch dazu verwendet werden, um die Qualität der Klassifikation objektiv beurteilen zu können.

6.1 Beurteilung der Verfahren bezüglich Abstand zum Fahrzeug

Zunächst soll genauer untersucht werden, in welchem Abstand zum Fahrzeug die Verfahren, auf die sich die Klassifikation abstützt, noch zuverlässig arbeiten. Dies ist für eine Abschätzung der maximal möglichen Reaktionszeiten wichtig. In diesem Fall dient eine gestellte Szene, in der sich ein Fußgänger auf das stehende Fahrzeug zu bewegt, als Testumgebung.

Abbildung 6.1 auf der nächsten Seite zeigt die graphische Auswertung dieser Referenzszene mit nur einem Fußgänger.

Da in diesem Experiment der Abstand zum Fahrzeug die entscheidende Größe darstellt, gibt dieser die x-Achse in allen Diagrammen vor. Die Diagramme beginnen bei etwa 37 Metern Abstand auf der linken Seite und enden bei etwa 2 Metern am rechten Rand. Der x-Maßstab für alle Diagramme ist identisch gewählt, um eine gute Vergleichbarkeit sicherzustellen. Über den Diagrammen ist zusätzlich eine Bildsequenz zu sehen, welche die Größe des Fußgängers im FIR-Video verdeutlichen soll.

Die Diagramme beginnen zu dem Zeitpunkt, zu dem der MHT das Objekt bestätigt. Das Objekt wird in dem gezeigten Bereich vom MHT nie verworfen, also kontinuierlich ab einem Abstand von 37 Metern verfolgt.

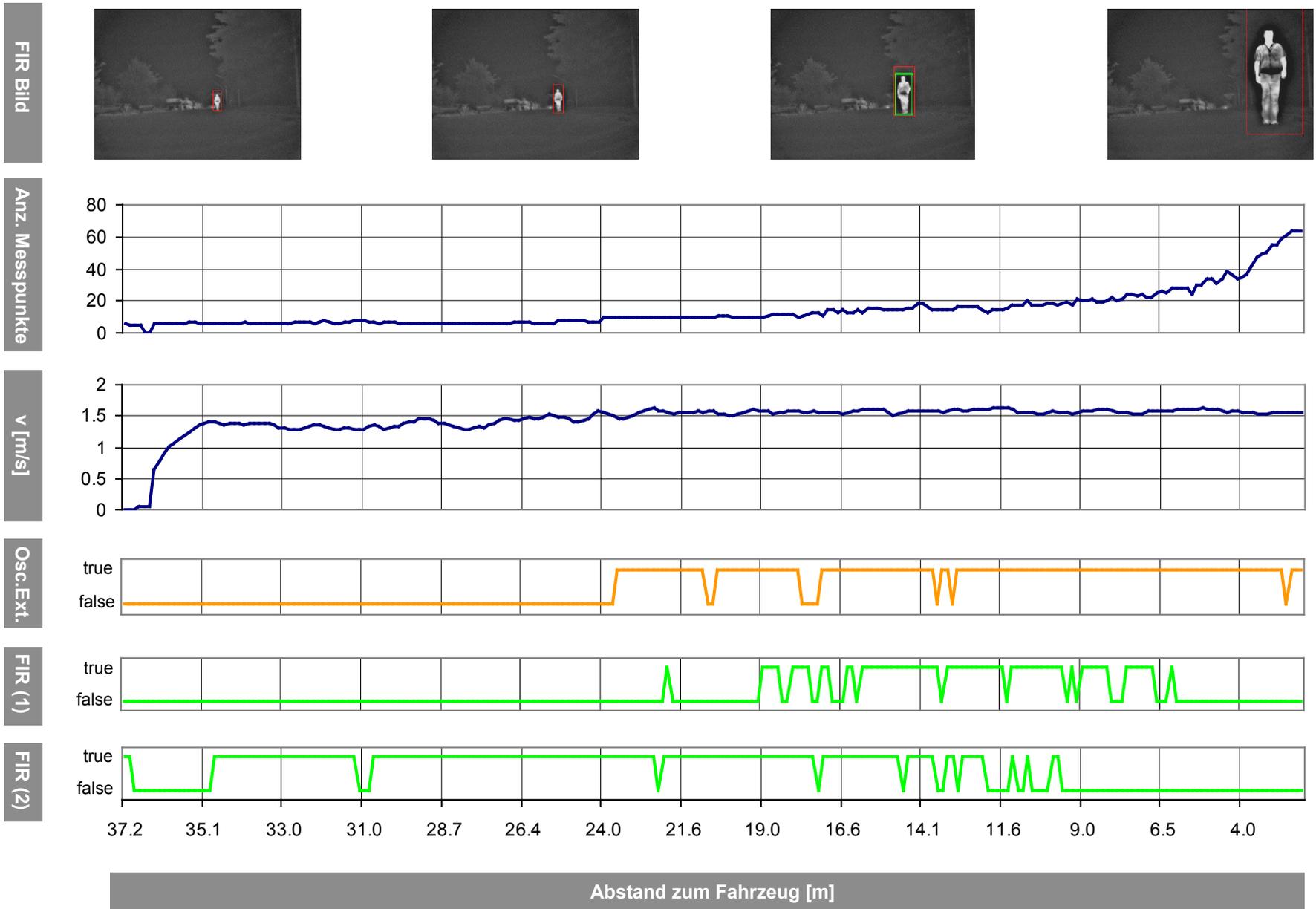


Abbildung 6.1: Auswertung der Verfahren bezüglich Abstand zum Fahrzeug

Die ausführliche Legende und die Interpretation der einzelnen Diagramme sieht folgendermaßen aus:

Anzahl Messpunkte Das Diagramm zeigt die Anzahl der Laser-Messpunkte, die der Fußgänger im Laserscan erzeugt. Das Segmentierungsverfahren gibt hier eine Unterschranke für die Anzahl vor, um nicht zu viele Störungen durch kleine Segmente zu erzeugen. Erst wenn die Segmentierung ein ausreichend großes Segment findet, beginnt der MHT das Objekt zu verfolgen. Unter Umständen überschreitet die Anzahl der Messpunkte bereits bei größerem Abstand die Schwelle; wenn allerdings in darauffolgenden Scans keine Bestätigung erfolgt, verwirft der MHT das Objekt als „False Alarm“.

Geschwindigkeit v Hier ist die Kalman gefilterte Objektgeschwindigkeit angetragen. Bereits nach wenigen Laserscans und einem Abstand von etwa 35 Metern schwingt sich diese auf einen Wert von ca. 1,5 m/s ein. Damit ist die Objektgeschwindigkeit ein wichtiges Kriterium für die Klassifikation, vor allem bei weiter entfernten Objekten.

Oscillating Extent Dieses Diagramm zeigt das Ergebnis der Fourier Analyse der Bein-Pendelbewegungen, welche erst viel später im Laserscan zu erkennen sind. In der gezeigten Szene erfolgt dies ab ca. 24 Metern, dann allerdings ohne nennenswerte „Lücken“, obwohl das Fußgänger-Segment bei dieser Entfernung nur aus etwa 10 Messpunkten besteht. Es ist anzunehmen, dass die Auflösung der Laserscanner weiter voranschreitet, und dieses Verfahren somit bei immer größeren Entfernungen anwendbar ist.

FIR (1) gibt an, ob der Fußgänger im FIR-Bild mittels des *FirVision*-Ansatzes aus [19] erkannt wurde (*FIR Match*). Es zeigt sich, dass dieser vor allem bei geringerer Entfernung zum Fahrzeug gute Ergebnisse liefert. Erst wenn Teile des Fußgängers über den Bildrand hinaustreten, versagt die Klassifikation. Derart geringe Abstände könnten mit dem ebenfalls in [19] entwickelten „Bein-Klassifikator“ besser abgedeckt werden.

FIR (2) gibt an, ob der Fußgänger im FIR-Bild mittels des *RubVision*-Ansatzes aus [33] erkannt wurde. Da das Verfahren ursprünglich für Situationen auf der Landstraße optimiert wurde, findet hier die Erkennung des Fußgängers wesentlich früher statt. Es ist bei komplexeren Szenen allerdings auch mit mehr False Positives zu rechnen. Deutlich schlechter wird die Erkennungsleistung bei Abständen unter 15 Metern.

6.2 Betrachtung der Rechenzeiten

Der folgende Abschnitt widmet sich den Laufzeiten der einzelnen Verfahren. Dies ist notwendig, um abschätzen zu können, welches Vorgehen für den Echtzeiteinsatz im Fahrzeug am besten geeignet ist. Bei der Implementierung in dieser Arbeit wird der Takt für den gesamten Fusions-Prozess durch den Laserscanner vorgegeben. Dies bedeutet, dass für einen Verarbeitungsschritt maximal 100 ms zur Verfügung stehen.

Alle Rechenzeiten werden auf einem aktuellen Pentium IV PC-System ermittelt.

Es zeigt sich, dass die folgenden Arbeitsschritte bezüglich ihrer Laufzeit eine untergeordnete Rolle spielen:

Arbeitsschritt	Durchschn. Laufzeit μ	Standardabweichung σ
Daten in Ringpuffer laden	2.4 ms	8.1 ms
Laserscan Segmentierung	1.4 ms	5.0 ms
Klassifikation	0.5 ms	1.9 ms
Analyse Pendelbewegungen	0.2 ms	0.7 ms

Die Werte werden anhand einer komplexen Szene ermittelt, um eine möglichst realistische obere Schranke angeben zu können. Um zu verdeutlichen, wie stark die Werte um den Mittelwert schwanken, zeigt die Tabelle auch die Standardabweichung.

Daten in Ringpuffer laden Da der MHT und die Fourier Analyse auf zeitlich zurückliegende Daten zugreifen müssen, werden zunächst die Video-Frames und der Laserscan in eine spezielle Ringpuffer-Struktur geladen. Dieser Schritt ist hier der Vollständigkeit halber aufgeführt, da er zeitlich gesehen in der gleichen Größenordnung liegt. Dazu gehört auch die Umwandlung der Laser-Rohdaten in ein für die Anwendung taugliches Format.

Laserscan Segmentierung Ein typischer Laserscan besteht aus etwa 1200 Messpunkten. Diese werden durch die Segmentierung in einzelne Punktgruppen zerlegt. Das Verfahren scheint ausreichend effizient zu arbeiten.

Klassifikation Bei der Klassifikation ergibt sich kein nennenswerter Unterschied zwischen der Variante mit Bayes'schem Netz und der Methode mit Entscheidungsbaum. Beide Verfahren sind ausreichend effizient für einen Echtzeiteinsatz.

Analyse Pendelbewegungen Die Fourier-Analyse schlägt mit dem geringsten zeitlichen Aufwand zu Buche. Dies ist ein weiterer Grund, der für dieses Verfahren spricht.

6.2.1 Multi-Hypothesen-Tracker

Die Analyse der Laufzeit des Multi-Hypothesen-Trackers gestaltet sich etwas komplexer. Folgende Einflussfaktoren sind zu beachten:

- Die Anzahl der gleichzeitig getrackten Objekte
- Der Abstand zwischen den Objekten
- Die Unsicherheit in der Position durch die Kalman-Filterung

Gerade bei Fußgängergruppen treten viele Objekte mit geringem räumlichen Abstand auf. Dies führt dazu, dass der MHT-Algorithmus viele verschiedene Hypothesen für die einzelnen Objekt-Messwert-Zuordnungen bildet. Abbildung 6.2 zeigt die für die Auswertung verwendete Szene. Es handelt sich um eine perspektivische Ansicht, bei dem jeder Zylinder einen verfolgten Fußgänger repräsentiert. Die Positions-Historien sind durch Linien auf dem „Boden“ verdeutlicht. Das Gitternetz hat eine Auflösung von einem Meter in beide Richtungen.

Abbildung 6.3 zeigt ein Diagramm, bei dem die Laufzeit in Abhängigkeit von der Anzahl der Objekte angetragen ist. Da die Messpunkte halbtransparent überlagert sind, deutet eine dunklere Farbe eine höhere Dichte an. Es fällt auf, dass die maximale Zeitdauer mit größer werdender Objektanzahl ansteigt. Die Laufzeit ist jedoch in den meisten Fällen weitaus geringer als der Maximalwert. Da die gewählte Szene einen Extremfall darstellt, ist der Einsatz des MHT-Algorithmus bezüglich der Rechenzeit vertretbar.

6.2.2 FIR-Bildverarbeitung

Für die Laufzeitbetrachtung bezüglich der Bildverarbeitung sind drei Szenarien relevant:

- *FirVision*-Ansatz auf das ganze Bild ohne weitere Optimierungen
- *FirVision*-Ansatz beschränkt auf die LIDAR-ROIs und optimiertem Suchalgorithmus
- *RubVision*-Ansatz auf das gesamte Bild angewendet¹

Abbildung 6.4 zeigt eine Gegenüberstellung der Laufzeiten der einzelnen Szenarien. Die Rechenzeit für den *FirVision*-Ansatz auf das ganze Bild verhält sich nahezu unabhängig von der Anzahl der Objekte im Bild. Der Algorithmus sucht, unabhängig von der Szenenkomplexität, das gesamte Bild nach möglichen Fußgängern ab. Die ermittelten Rechenzeiten decken sich in etwa mit den in der Arbeit [19] angegebenen Werten für den „Ganz-Körper-Klassifikator“.

Beschränkt man den *FirVision*-Ansatz lediglich auf die gefundenen ROIs und wendet den optimierten Suchalgorithmus an, so sinkt die Rechenzeit bei kleiner Objektanzahl deutlich. Natürlich erhöht sich die Rechenzeit bei diesem Szenario, sobald die Anzahl der zu untersuchenden ROIs steigt. Sie bleibt jedoch, wie aus der Grafik hervorgeht, selbst bei einer hohen Objektdichte unter dem Wert für das Durchsuchen des gesamten Bildes.

Die Rechenzeit des *RubVision*-Ansatzes steigt ebenfalls mit wachsender Objektdichte. Dies erscheint plausibel, da das Verfahren zunächst eine Segmentierung anhand von vertikalen Strukturen im Bild durchführt. Je größer die Objektdichte, desto höher wird auch der Anteil an vertikalen Strukturen im Bild sein.

Insgesamt lässt sich sagen, dass die Fusion mit den Laserscanner-Daten einen deutlichen Vorteil bezüglich der Rechenzeit für die Bildverarbeitung mit sich bringt.

¹Für die *RubVision*-Methode wurde keine spezielle Optimierung auf ROIs entwickelt.

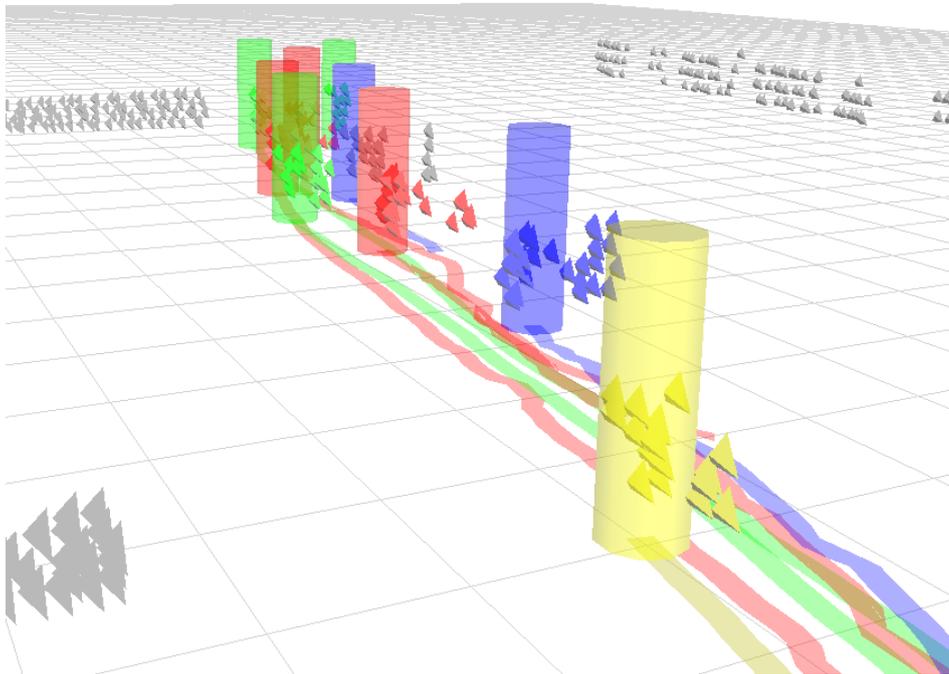


Abbildung 6.2: Szene mit Fußgängergruppe zur MHT Laufzeitauswertung

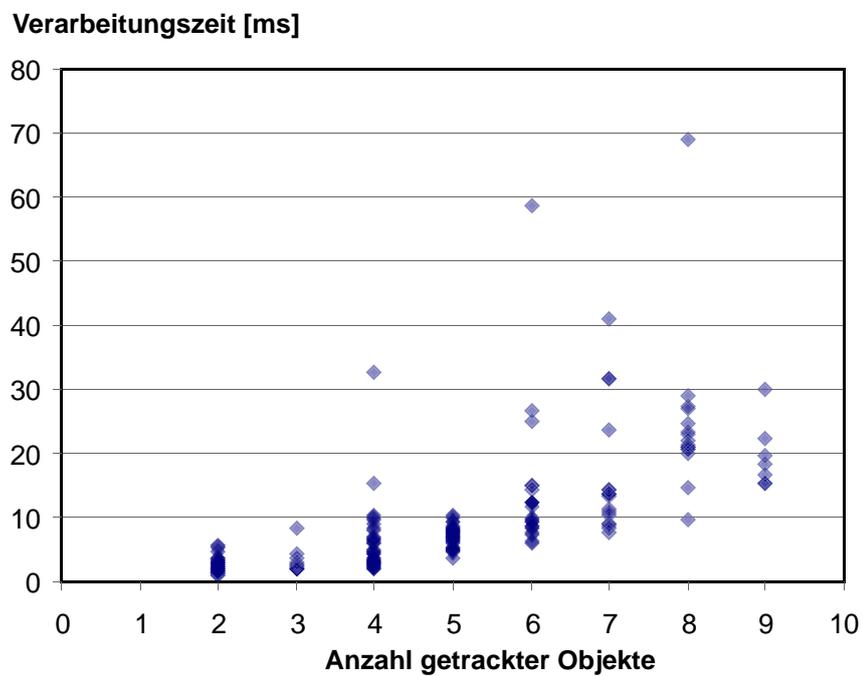


Abbildung 6.3: Laufzeiten des MHT-Algorithmus bezüglich der Anzahl der getrackten Objekte

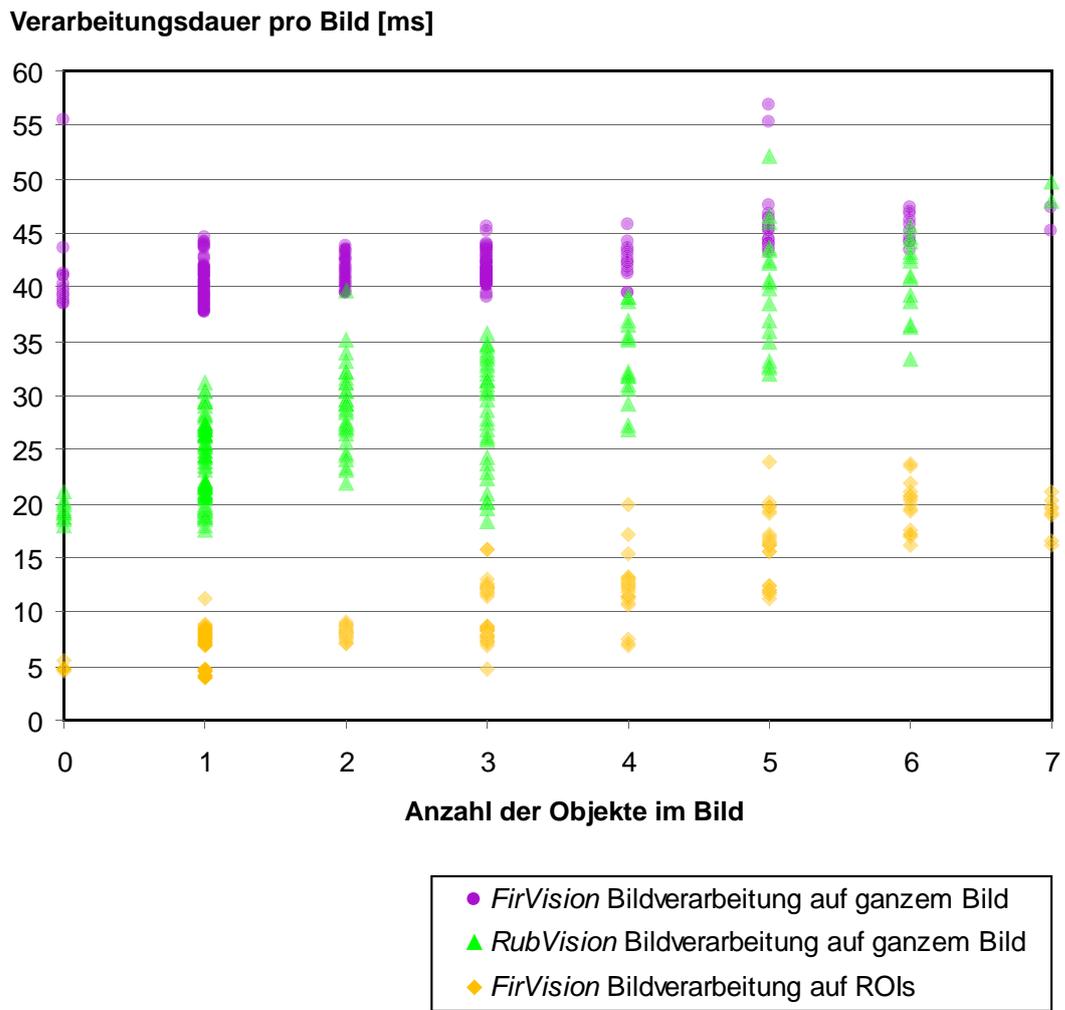


Abbildung 6.4: Vergleich der Rechenzeit der verschiedenen Bildverarbeitungs-Methoden

6.2.3 Gesamte Verarbeitungsdauer

Es wird nun noch die Rechenzeit für einen kompletten Zyklus untersucht. Abbildung 6.5 zeigt die Auswertung anhand einer komplexen Szene. Es wird deutlich, dass die Rechenzeit, bis auf wenige Ausnahmen, unterhalb der Schranke von 100 ms liegt. Da alle Messungen auf einem normalen Multitasking-Betriebssystem ausgeführt werden, lässt sich nicht vollständig ausschließen, dass Prozesswechsel zu scheinbar höheren Verarbeitungszeiten geführt haben. Aus diesem Grund wird jedoch auf die Angabe einer einzelnen Zahl verzichtet und stattdessen eine grafische Darstellung gewählt.

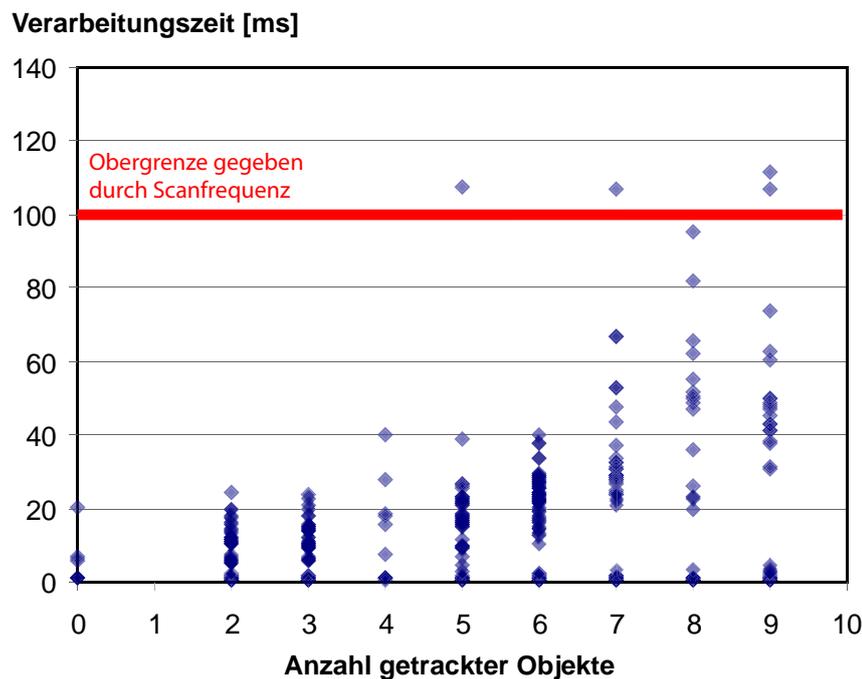


Abbildung 6.5: Laufzeit des gesamten Fusions-Prozesses in Abhängigkeit von der Anzahl der verfolgten Objekte

6.3 Beurteilung der Erkennungsleistung

Ein weiterer Baustein zur Bewertung der implementierten Verfahren ist die Zuverlässigkeit der Klassifikation.

Vorab werden die wichtigsten Begrifflichkeiten erklärt:

Hit Ein korrekt erkannter Fußgänger

False Positive Ein Objekt, welches in der Realität *kein* Fußgänger ist, wird dennoch als Fußgänger klassifiziert.

False Negative Ein Objekt, welches in der Realität ein Fußgänger ist, wird fälschlicherweise nicht als solcher erkannt.

Da hier nun etliche Faktoren eine Rolle spielen, ist die Definition einer Metrik zur objektiven Beurteilung der Erkennungsleistung schwierig. Zunächst werden einige dieser Faktoren zusammengefasst:

- Der Laserscanner hat einen wesentlich größeren *Sichtbereich* vor dem Fahrzeug als die FIR-Kamera. Eine Fusion ist somit nur im engen Sichtbereich (ca. 36° Öffnungswinkel) der Kamera möglich. In Bereichen, in denen keine Fusion mit der FIR-Bilderkennung möglich ist, stützt sich die Klassifikation alleine auf die Merkmale aus dem Laserscan (Geschwindigkeit, Größe, Pendelbewegungen).
- Durch das *Tracken der Objekte* im Laserscan ist es möglich, Objekte über einen längeren Zeitraum hinweg zu verfolgen und über die Zeit hinweg zusätzliche Evidenzen zu sammeln (für oder gegen Fußgänger). Keines der beiden FIR-Erkennungsverfahren implementiert ein ähnlich zuverlässig arbeitendes Trackingverfahren. Anders als bei der reinen Bildverarbeitung müssen Fußgänger also nicht in jedem Zeitschritt neu klassifiziert werden.
- Im Falle von *stehenden Fußgängern* bringt die Fusion mit der Bildverarbeitung den größten Nutzen. Das Tracking kann bei einem stehenden Fußgänger keine charakteristische Geschwindigkeit und auch keine Pendelbewegungen ermitteln. Die einzige Kenngröße, die bleibt, ist somit die räumliche Ausdehnung eines Objekts. Diese Angabe alleine führt allerdings zu unbrauchbar vielen Falschmeldungen, da im Straßenverkehr sehr viele Objekte von der Größe eines Fußgängers auftreten.

Aufgrund dieser Ausgangslage werden Szenen analysiert, in denen die meisten Fußgänger in Bewegung sind. Die folgenden Ausführungen beziehen sich auf die Erkennungsleistung des Entscheidungsbaum-Klassifikators. Für das Bayes'sche Netz werden keine weiteren Untersuchungen angestellt.

6.3.1 Untersuchung ohne Bildverarbeitung

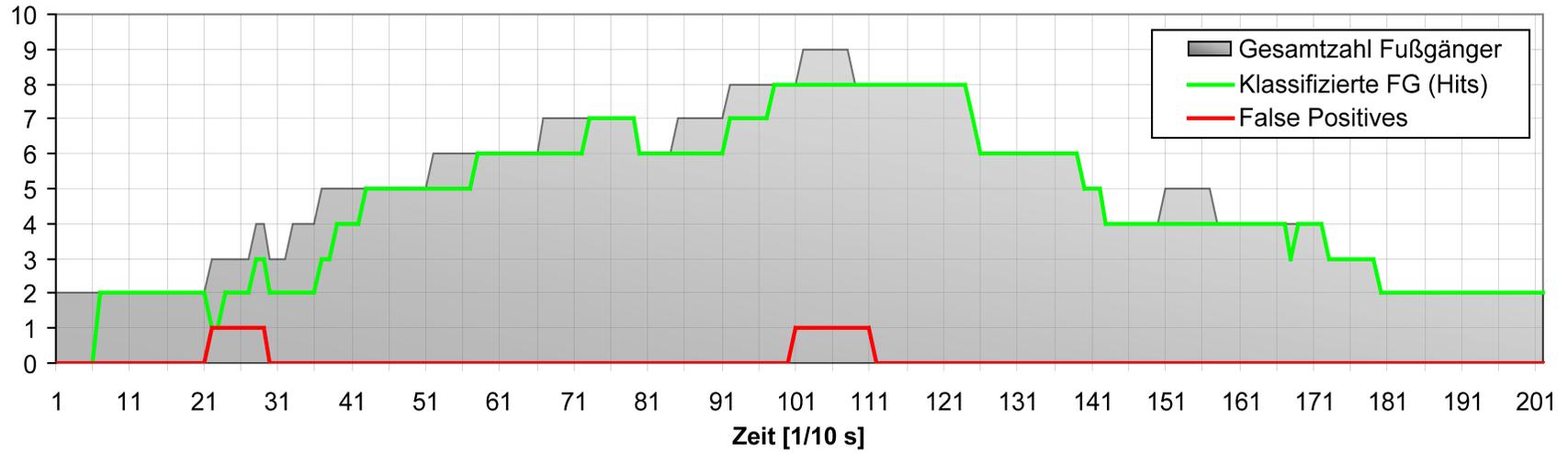
Abbildung 6.6(a) auf Seite 80 zeigt die Auswertung der Klassifikation in einer Szene, die sich nur auf den Laserscan abstützt. Die grau hinterlegte Kurve gibt die Anzahl der Fußgänger im Sichtbereich des Laserscanners an. Dieser Wert ergibt sich aus der manuellen Markierung aller Fußgänger in der Szene. Die korrekt klassifizierten Fußgänger (Hits) sind zu einer Summe zusammengefasst und ebenfalls in das Diagramm eingezeichnet. An dieser Darstellung lässt sich auch ablesen, dass die Klassifikation in allen Fällen etwa eine halbe Sekunde nach Erscheinen des Fußgängers erfolgt. Es treten jedoch noch einige „False Positives“ auf. Die Anzahl der „False Negatives“ ergibt sich implizit aus der Gesamtanzahl und der Anzahl der korrekt erkannten Fußgängern. Immer dann, wenn die grüne Linie in der Grafik unterhalb des grauen Bereichs liegt, treten folglich „False Negatives“ auf.

6.3.2 Untersuchung mit Bildverarbeitung

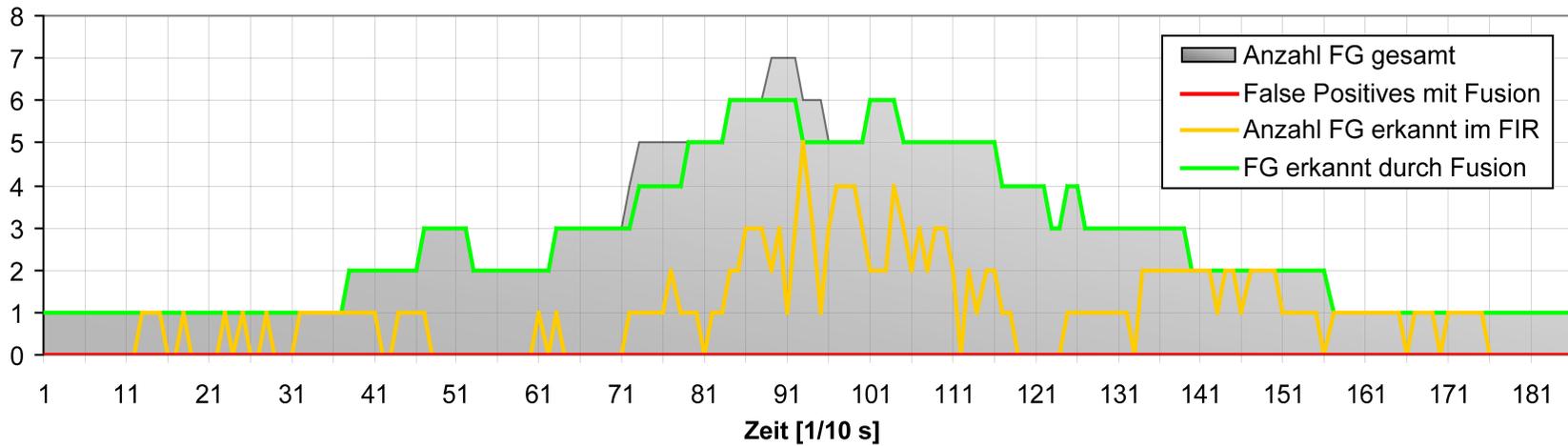
Abbildung 6.6(b) zeigt die Auswertung der gleichen Szene. Allerdings ist der Bereich für die Auswertung in diesem Fall *auf den Sichtbereich der FIR-Kamera beschränkt*. Die Klassifikationsleistung ergibt sich nun durch den Fusionsansatz. Das Tracking verfolgt die Fußgänger bereits außerhalb des Sichtbereichs der FIR-Kamera. Deshalb ist in dieser Darstellung keine Verzögerung in der Klassifikation zu erkennen. Mit der Fusion treten in dieser Szene keine „False Positives“ mehr auf.

Zusätzlich ist die Erkennungsleistung der reinen Bildverarbeitung (*FirVision*) angetragen. Wie bereits dargelegt, erfolgt die Erkennung im Bild in jedem Frame neu. Deshalb können Fußgänger, die bereits erfolgreich klassifiziert werden, unter Umständen später nicht wieder gefunden werden. Aufgrund dieser Tatsache springt die Linie für die Erkennungsleistung der reinen FIR-Bildverarbeitung relativ stark.

Im Anhang C auf Seite 97 ist die Auswertung einiger weiterer Szenen zu finden.



(a) Klassifikation ohne Bilderkennung über den gesamten Sichtbereich des Laserscanners



(b) Klassifikation durch Fusion und Vergleich mit reiner Bildverarbeitung

Abbildung 6.6: Analyse der Zuverlässigkeit der Klassifikation in einer Beispielszene

ZUSAMMENFASSUNG UND AUSBLICK

Diese Arbeit zeigt, dass die Kombination aus FIR-Kamera und Laserscanner gute Ergebnisse für die Fußgängererkennung ergibt. Die beiden Sensoren ergänzen sich in vielen Situationen und ermöglichen so eine höhere Sicherheit in der Entscheidung.

Segmentierung Das umgesetzte Segmentierungsverfahren für die Punktwolken des Laserscanners funktioniert in den untersuchten Szenen gut. Durch die adaptive Tiefensuche arbeitet der Algorithmus effizient. Die Anpassung der Segmentierungsschwelle bei größer werdendem Abstand zum Fahrzeug erweist sich als probates Mittel, um auch weiter entfernte Objekte noch gut zu trennen.

Tracking Das eingesetzte Multi-Hypothesen-Tracking bietet vor allem bei komplexen Szenen Vorteile. Auch bei Fußgängergruppen lassen sich die Trajektorien der einzelnen Personen noch gut unterscheiden, obwohl dort naturgemäß oft kurzzeitige Verdeckungen auftreten. Die Parameter für die Kalman-Filterung sind für die untersuchten Szenarien gut geeignet.

Bein-Pendelbewegungen Die Analyse der Bein-Pendelbewegungen mittels Fourier-Transformation liefert einen guten Anhaltspunkt, ob es sich bei einem Objekt um einen Fußgänger handelt. Das Verfahren funktioniert unabhängig von der Bewegungsrichtung des Fußgängers. Dazu werden alle Messpunkte, die zu einem Objekt gehören, in ein objekt-lokales Koordinatensystem transformiert. Die Lage dieses Koordinatensystems ist abhängig vom Geschwindigkeitsvektor, welcher durch das Tracking ermittelt wird. Es zeigt sich, dass auch bei einem relativ kurzen Beobachtungsfenster von 1,6 Sekunden noch ein ausreichend großer Ausschlag im Frequenz-Spektrum sichtbar ist.

Klassifikation Die Klassifikation mittels Entscheidungsbäumen und vorheriger Fusion auf Merkmalsebene arbeitet zuverlässig (siehe Abschnitt 6.3). Durch die Fusion mit der FIR-Kamera können einige „False Positives“ vermieden werden. Eine erste Klassifikation kommt in den untersuchten Szenen meist etwa eine halbe Sekunde nach Erscheinen

des Objekts zustande. Im Unterschied zur reinen Bildverarbeitung eröffnet das Verfolgen von Objekten auch die Möglichkeit, Informationen über einen längeren Zeitraum zu „sammeln“. Eine Klassifikation ist also nicht in jedem Zeitschritt von neuem notwendig.

Abstand zum Fahrzeug Die Analyse im Hinblick auf den maximalen Abstand zum Fahrzeug ist wichtig, um eine Aussage treffen zu können, wie schnell auf einen Fußgänger reagiert werden kann. Die Auswertung ergibt, dass ein Verfolgen im Laserscan bei einer Annäherung an den Fußgänger ab ca. 35 Meter und darunter zuverlässig möglich ist. Unterschreitet ein Objekt diesen Abstand, lässt sich bereits eine Aussage über die Größe und die Geschwindigkeit treffen. Zusammen mit der *RubVision*-Bildererkennung [33], welche in diesem Abstand ebenfalls bereits Fußgänger klassifiziert, lässt sich eine sichere Aussage treffen, ob es sich um einen Fußgänger handelt.

Kommt der Fußgänger näher, so lassen sich ab ca. 24 Meter Entfernung zum Fahrzeug auch Bein-Pendelbewegungen im Laserscan nachweisen. In diesem Entfernungsbereich ist eine Kombination mit der *FirVision*-Bildverarbeitung [19] am günstigsten. Mit der nächsten Generation von Laserscannern¹ ist eine Erkennung auch schon in größerem Abstand zum Fahrzeug möglich.

Rechenzeit Alle Verfahren sind für einen Echtzeiteinsatz bei einer Scanfrequenz des Laserscanners von 10 Hz geeignet. Vor allem die Bildverarbeitung konnte durch die Beschränkung auf interessante Bildbereiche deutlich beschleunigt werden.

Zukünftige Arbeiten können die ermittelten Trajektorien dazu nutzen, um entsprechende Warnfunktionen im Fahrzeug zu realisieren. Es lässt sich abschätzen, ob erkannte Fußgänger im aktuellen Fahrgeschehen zu einer Gefahr werden können. Abschnitt 6.1 zeigt, dass nicht jeder Ansatz für jeden Abstand zum Fahrzeug gleich gut geeignet ist. Die Information über den Abstand zum Fußgänger kann dazu genutzt werden, eine bestimmte Bildverarbeitungsmethode aus mehreren vorhandenen Ansätzen auszuwählen.

¹http://www.ibeo-as.de/html/rd/rd_dv_sensroam.html

GLOSSAR

Feature	Ein Merkmal, welches dazu genutzt wird, einem Objekt eine bestimmte Klasse zuzuordnen
FFT	Fast Fourier Transform Eine effiziente Umsetzung der Diskreten Fourier Transformation (DFT)
FIR	Ferninfrarot Bezeichnet den Bereich zwischen 0.7 und 1.4 μm Wellenlänge im infraroten Spektrum (siehe [1])
Fusion	Die kombinierte Verarbeitung mehrerer Sensor-Ströme
Klassifikation	Die Zuordnung eines Objekts zu einer Objektklasse
LIDAR	Light Detection and Ranging Oberbegriff für alle Messverfahren, die Entfernungen über Lichtimpulse ermitteln
MHT	Multi-Hypothesis-Tracker Trackingverfahren nach [24]
NIR	Nahinfrarot Bezeichnet den Bereich zwischen 15 und 1000 μm Wellenlänge im infraroten Spektrum (siehe [1])
Precrash	Bezeichnet Maßnahmen, die kurz vor einem unvermeidbaren Unfall zur Minderung der Unfallfolgen getroffen werden
ROI	Region of Interest

	Bezeichnet einen Bereich im Bild, der als interessant eingestuft wird (enthält z. B. einen potentiellen Fußgänger)
Segment	Bezeichnet in dieser Arbeit eine Menge von Messpunkten, die aufgrund ihrer räumlichen Lage, ein und demselben Objekt zugeordnet werden
Segmentierung	Bezeichnet den Vorgang, die Punktwolke eines Laserscanners in mehrere Segmente zu zerlegen
Tracking	Aus dem engl. „das Verfolgen“ Steht für das Beobachten von Objekten über einen längeren Zeitraum hinweg, um den Kurs des Objekts zu schätzen

ABBILDUNGSVERZEICHNIS

1.1	Beispielszene mit mehreren gleichzeitig verfolgten Fußgängern vor dem Fahrzeug	5
1.2	Verschiedene Ansätze für die Sensorfusion	7
1.3	Grundsätzlicher Aufbau des Systems und Datenfluss	8
2.1	Der Ablauf des Fusions-Prozesses	13
2.2	Frontansicht des Versuchsträgers	15
2.3	Schematische Darstellung des Versuchsträgers und der Sensorik	16
2.4	Darstellung der Scan-Ebenen und des Koordinatensystems, Quelle: IBEO AS	16
2.5	Versuchsreihe mit verschiedenen Isolations-Materialien auf einer Metallplatte	19
2.6	Zusammenbauskizze des Schachbretts	20
2.7	Ansicht des Schachbretts im FIR-Bild	20
2.8	FIR-Bild mit überlagerten Laser-Messpunkten	22
2.9	Der Szenenbaum mit allen Transformations-Matrizen	24
2.10	Herleitung der Projektionsmatrix für die Kameras	25
3.1	Ein vereinfachter Laserscan und die Organisation der Daten	29
3.2	Zwei Segmente und deren Schattenwurf in der Draufsicht	33
3.3	Entwicklung eines einzelnen Hypothesen-Baums	37
3.4	Tracking eines kurzzeitig verdeckten Fußgängers	39
3.5	Messpunkte eines Fußgängers über mehrere Zeitschritte hinweg	41
3.6	Ein Fußgänger und sein lokales Koordinatensystem in der Draufsicht	43
3.7	Die räumliche Ausdehnung eines Fußgänger-Segments	45
3.8	Diskrete Fourier-Transformation der räumlichen Ausdehnung eines Fußgänger-Segments	45
3.9	DFT Kurzzeitspektren der räumlichen Ausdehnung eines gehenden Fußgängers	46
3.10	DFT Kurzzeitspektren der räumlichen Ausdehnung eines Objekts, welches kein gehender Fußgänger ist	46

4.1	Berechnung der ROI (gelb) aus der räumlichen Ausdehnung eines Segments (weiß)	49
4.2	Vergleich der realen Ausdehnung eines Segments und der gefilterten Objekt-Größe	50
4.3	Beispielszene zur Ermittlung der Tiefenreihenfolge der ROIs	51
4.4	Kombination der ROIs aus den Laserdaten und der unabhängig gefundenen FIR-Bildregionen	52
4.5	Beschränkung der Bildverarbeitung auf die ermittelten ROIs aus den Laserdaten	52
4.6	FIR-Bildverarbeitung mit Hilfe der ROIs aus den LIDAR-Daten	54
4.7	Bayes'sches Netz für die Klassifikation	56
5.1	Darstellung der logischen Verknüpfung der Komponenten des Systems	62
5.2	Bildschirmabdruck der Visualisierungs-Komponente	63
5.3	UML Klassendiagramm der Visualisierungskomponente	64
5.4	Zeitliche Abfolge der Datenverarbeitung im Fusions-Modul	65
6.1	Auswertung der Verfahren bezüglich Abstand zum Fahrzeug	71
6.2	Szene mit Fußgängergruppe zur MHT Laufzeitauswertung	75
6.3	Laufzeiten des MHT-Algorithmus bezüglich der Anzahl der getrackten Objekte	75
6.4	Vergleich der Rechenzeit der verschiedenen Bildverarbeitungs-Methoden . .	76
6.5	Laufzeit des gesamten Fusions-Prozesses in Abhängigkeit von der Anzahl der verfolgten Objekte	77
6.6	Analyse der Zuverlässigkeit der Klassifikation in einer Beispielszene	80
A.1	Verletzungsschwere in Abhängigkeit der Kollisionsgeschwindigkeit bei Fußgängern	94
B.1	UML-Klassendiagramm	96
C.1	Klassifikation durch Fusion und Vergleich mit reiner Bildverarbeitung in einer Beispielszene	98
C.2	Klassifikation durch Fusion und Vergleich mit reiner Bildverarbeitung in einer Beispielszene	99

TABELLENVERZEICHNIS

2.1	Zusammenhang zwischen Scanfrequenz und Winkelauflösung des Scanners .	17
2.2	Messpunkte am Fahrzeug für die extrinsische Kalibrierung	21
4.2	Beispiel eines Trainingsdatensatzes für den Entscheidungsbaum	58

LITERATURVERZEICHNIS

- [1] *Wikipedia, the free encyclopedia*. <http://en.wikipedia.org/wiki/>. 83
- [2] *Richtlinie 2003/102/EG des europäischen Parlaments und des Rates zum Schutz von Fußgängern und anderen ungeschützten Verkehrsteilnehmern vor und bei Kollisionen mit Kraftfahrzeugen*, November 2003. <http://europa.eu.int/comm/enterprise/automotive/pagesbackground/pedestrianprotection/>. 3
- [3] *Cooperation, Intel: Open CV Library - Versionen 3.1 bis 5.0 beta*. Technischer Bericht, June 2005. 18
- [4] BRUEGGE, BERND und ALLEN H. DUTOIT: *Object-Oriented Software Engineering*. Pearson Education Inc., 2004. 61
- [5] CORMEN, THOMAS H., CHARLES E. LEISERSON, RONALD L. RIVEST und CLIFFORD STEIN: *Introduction to Algorithms*. MIT Press, Zweite Auflage, 2001. 28
- [6] COX, I.J. und S.L. HINGORANI: *An Efficient Implementation and Evaluation of Reid's Multiple Hypothesis Tracking Algorithm for Visual Tracking*. In: *ICPR94*, Seiten A:437–442, 1994. 8, 66
- [7] FREUND, YOAV und ROBERT E. SCHAPIRE: *A short introduction to boosting*. Journal of Japanese Society for Artificial Intelligence, 1999. 18
- [8] FÜRSTENBERG, KAY, KLAUS DIETMAYER und ULRICH LAGES: *Offenlegungsschrift: Verfahren zur Erkennung von Fußgängern*. Technischer Bericht, IBEO Automobile Sensor GmbH, 2003. 8, 27, 41
- [9] GEE, L. A. und M. A. ABIDI: *Multisensor Fusion for Decision-based Control Cues*. 6
- [10] GONG, YIHONG: *Integrated Object Detection and Tracking by Multiple Hypothesis Analysis*. NEC Journal of Advanced Technology, 2005. 8

- [11] GRAY, KRIS: *Microsoft DirectX 9 Programmable Graphics Pipeline*. Microsoft Press, 2003. 66
- [12] GREHN, J. und J. KRAUSE: *Metzler Physik*. Schroeder Verlag GmbH, Hamburg, 1998. 20
- [13] HARTLEY, RICHARD und ANDREW ZISSERMAN: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Zweite Auflage, 2003. 23
- [14] HEDRICK, K. und D. CAVENEY: *Multiple Target Tracking using Multiple Models and Multiple Sensors UCB-BMW Collaboration - Final Report*. University of California at Berkeley, 2003. 36, 48
- [15] KLEIN, LAWRENCE A.: *Sensor and Data Fusion - A Tool for Information Assessment and Decision Making*. SPIE - The International Society for Optical Engineering, 2004. 6
- [16] KOCH, KARL-RUDOLF: *Einführung in die Bayes-Statistik*. Springer, 2000. 55, 56
- [17] KRUGLINSKI, DAVID J.: *Inside Visual C++ 6.0*. Microsoft Press Deutschland, September 2005. 66
- [18] KUGELE, STEFAN, ANDREAS SCHOLZ und STEPHAN THOMA: *Integrative Dialogführung in einer hybriden Systemarchitektur, Abschlussbericht des Interdisziplinären Projekts*, 2004. 56, 67
- [19] NEVE, ANTJE: *Fußgängererkennung in FIR und Video Sequenzen mithilfe entfernungsgebender Sensorik*. Diplomarbeit, Technische Universität München, 2005. 4, 8, 17, 18, 21, 53, 67, 72, 74, 82
- [20] OHLENBURG, JAN, THORSTEN FRÖHLICH und WOLFGANG BROLL: *Internal and External Scene Graphs: A New Approach for Flexible Distributed Render Engines*. In: *Proceedings of the IEEE Virtual Reality 2005 (VR2005)*. 22
- [21] PEARL, JUDEA: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988. 56, 67
- [22] QUINLAN, ROSS: *Simplifying decision trees*. International Journal of Man-Machine Studies, 27, 221-234, 1987. 8
- [23] QUINLAN, ROSS: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993. 58, 60, 67
- [24] REID, D. B.: *An algorithm for tracking multiple targets*. 1979. 8, 83
- [25] RUGGIERI, SALVATORE: *YaDT: Yet another Decision Tree builder*. 2003. <http://www.di.unipi.it/~ruggieri/Papers/yadt.pdf>. 67
- [26] SCHMITT, THORSTEN, MICHAEL BEETZ, ROBERT HANEK und SEBASTIAN BUCK: *Watch their Moves: Applying Probabilistic Multiple Object Tracking to Autonomous Robot Soccer*, 2001. 8

- [27] SCHRÜFER, ELMAR: *Elektrische Meßtechnik*. Carl Hanser Verlag München Wien, Siebte Auflage, 2001. 20
- [28] STRAUSS, PAUL S. und RIKK CAREY: *An object-oriented 3D graphics toolkit*. In: *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, Seiten 341–349, New York, NY, USA, 1992. ACM Press. 22
- [29] TAYLOR, GEOFFREY und LINDSAY KLEEMAN: *A Multiple Hypothesis Walking Person Tracker with Switched Dynamic Model*. 2003. 8, 41
- [30] TRONNIER, THORSTEN: *Fußgängerdetektion in Infrarot-Szenen für Fahrerassistenzsysteme*. Diplomarbeit, Technische Universität München, 2004. 4, 8, 41
- [31] WALLENTOWITZ, HENNING: *Umdruck zur Vorlesung Kraftfahrzeuge. II. Vertikal - /Querdynamik von Kraftfahrzeugen*. 1998. 25
- [32] WELCH, GREG und GARY BISHOP: *An Introduction to the Kalman Filter*. Technischer Bericht, 1995. 34, 67
- [33] WIEGERSMA, AALZEN: *Project Report BMW Pedestrian Detection in FIR Images*. Project Report, Ruhr Universität Bochum, März 2005. 17, 67, 72, 82
- [34] WILLHOEFT, V., R. SCHULZ, J. SCHOLZ und T. KLUGE: *ALASCA User Manual*. IBEO Automobile Sensor GmbH, März 2005. <http://www.ibeo-as.de>. 14
- [35] ZHANG, ZHENGYOU: *Flexible Camera Calibration by Viewing a Plane From Unknown Orientations*. 1999. 19

UNFALLSTATISTIKEN

MAIS steht für „Maximum Abbreviated Injury Scale“ und gibt die Schwere der schlimmsten Verletzung an (AIS 1 = Minor; AIS 2 = Moderate; AIS 3 = Serious; AIS 4 = Severe; AIS 5 = Critical; AIS 6 = Unsurvivable)

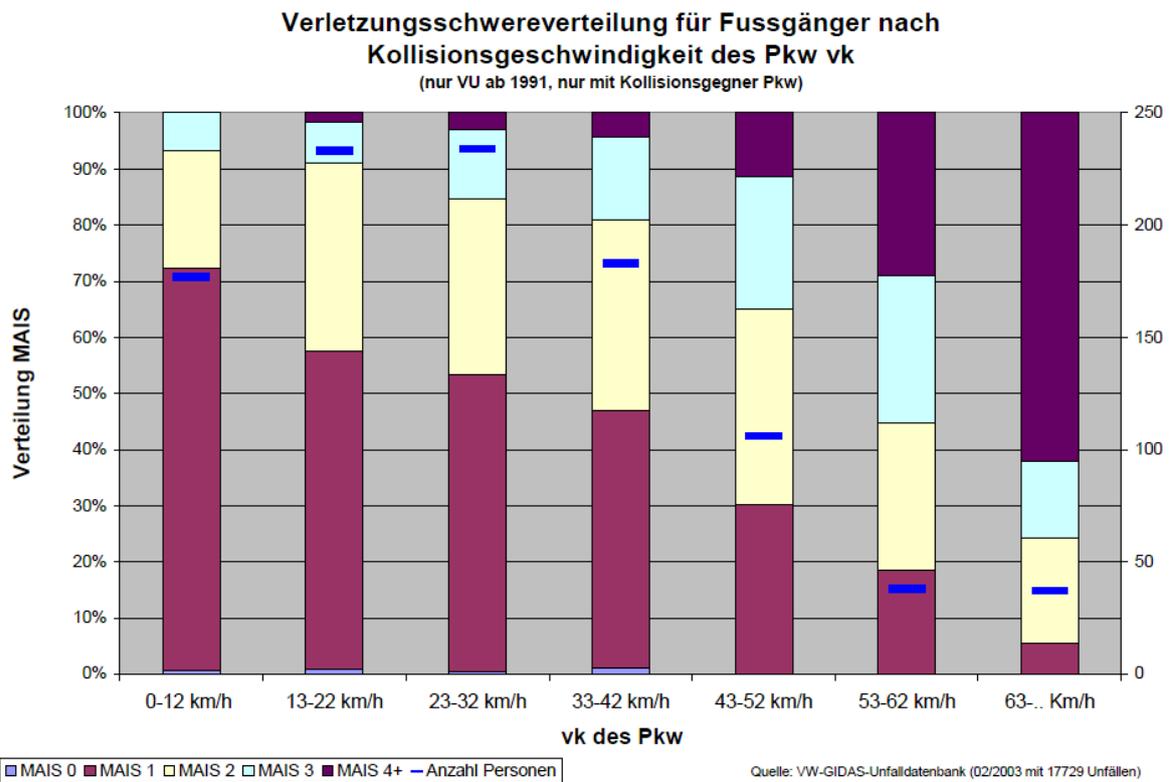


Abbildung A.1: Verletzungsschwere in Abhängigkeit der Kollisionsgeschwindigkeit (vk) bei Fußgängern

UML-DIAGRAMM

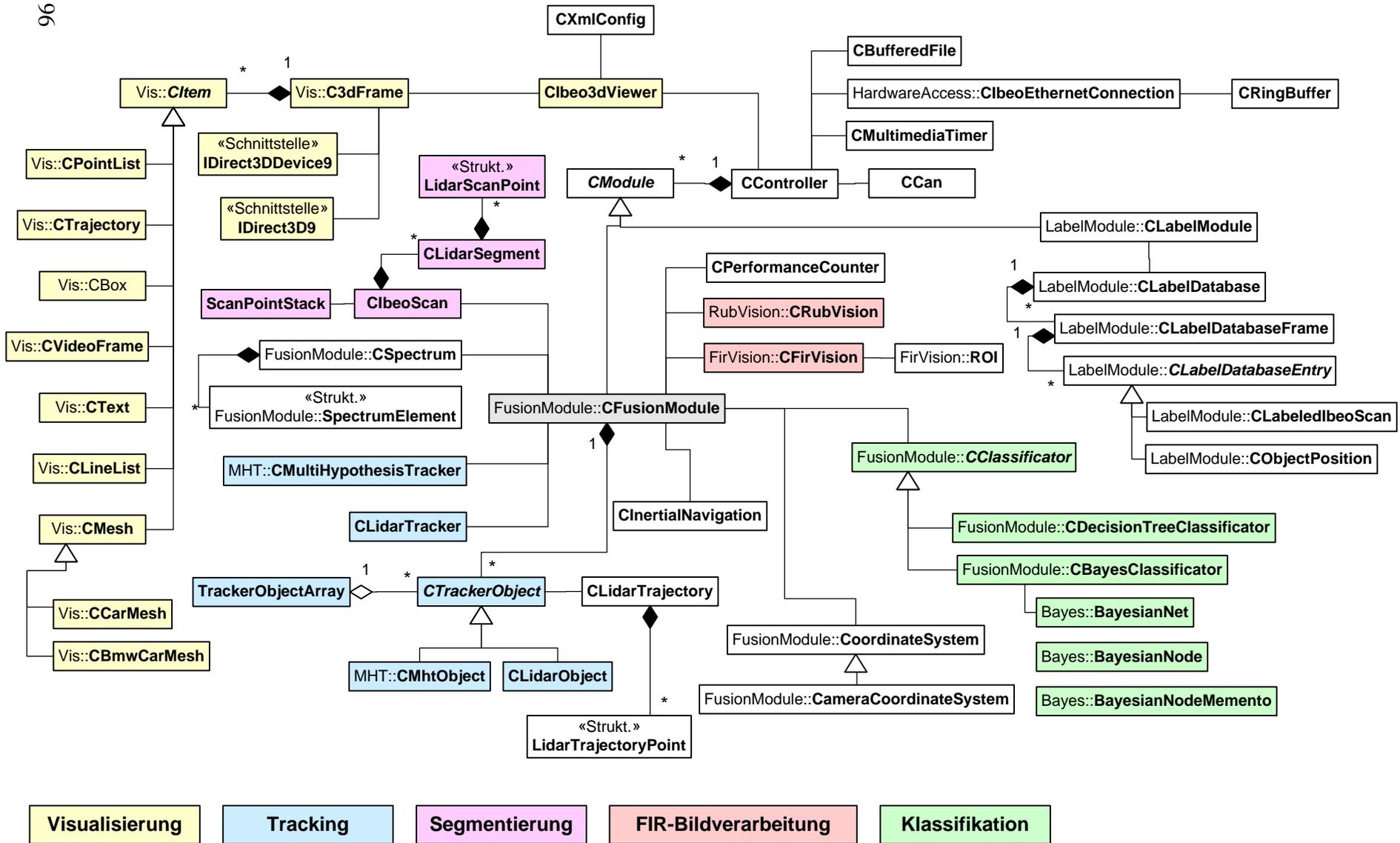


Abbildung B.1: UML-Klassendiagramm

KLASSIFIKATIONSERGEBNISSE

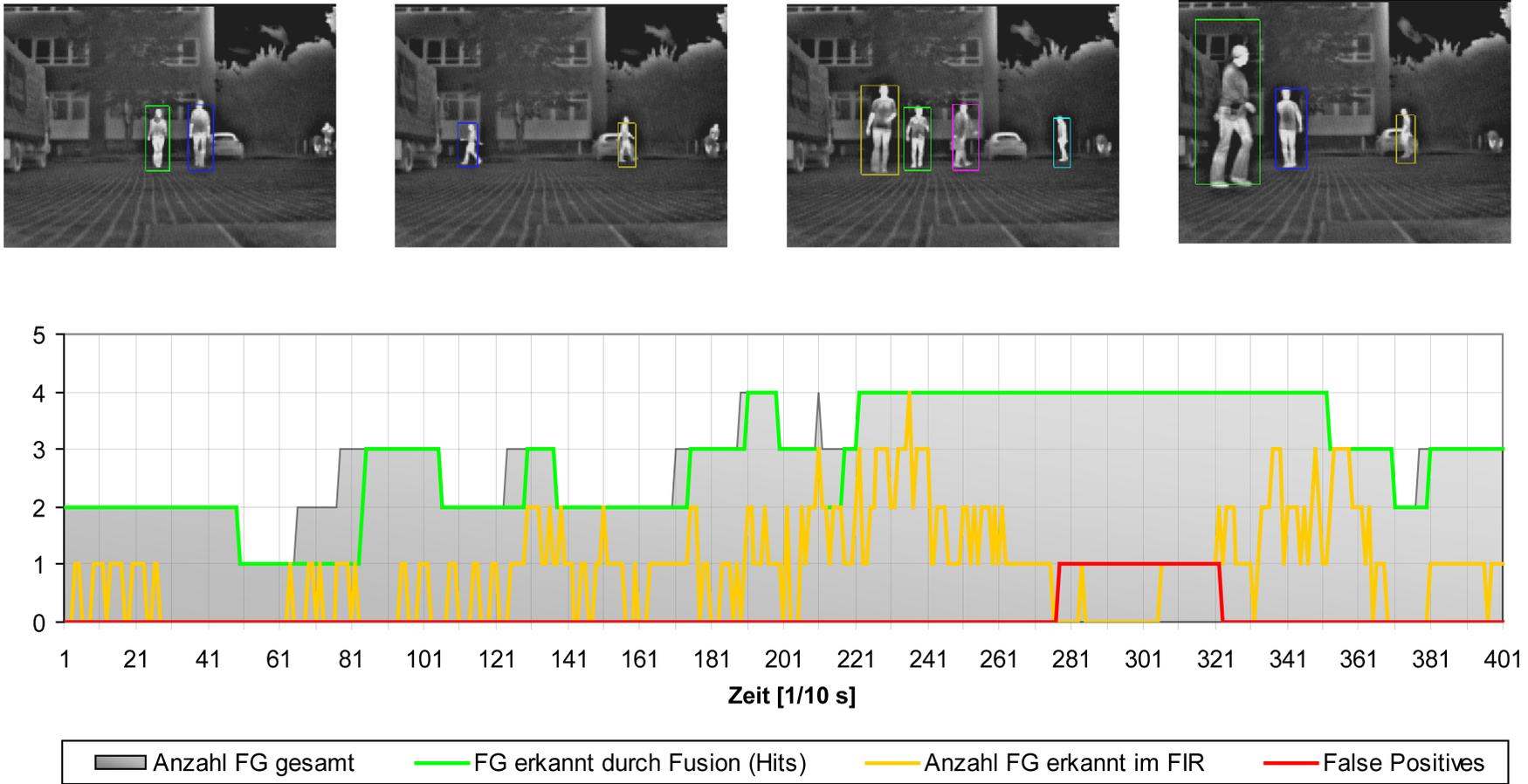


Abbildung C.1: Klassifikation durch Fusion und Vergleich mit reiner Bildverarbeitung in einer Beispielszene

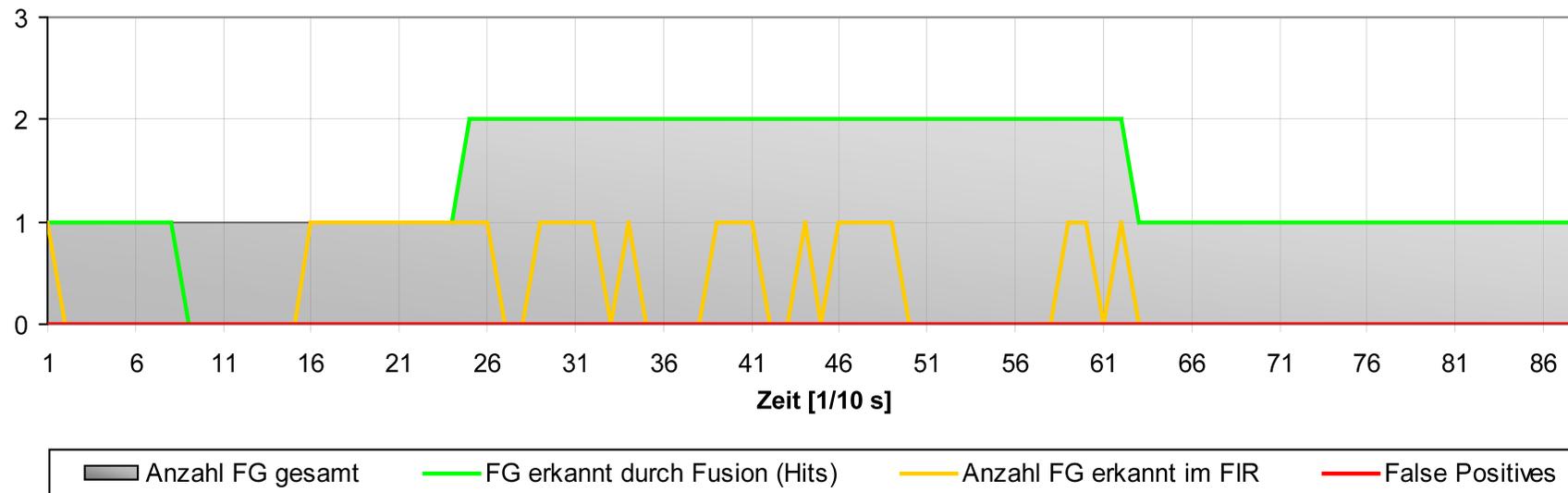


Abbildung C.2: Klassifikation durch Fusion und Vergleich mit reiner Bildverarbeitung in einer Beispielszene

ANHANG D

MATLAB CODE

Algorithmus D.1 : Berechnung der FFT in Matlab

```
function [result] = DoPedestrianFFT(values)
deltaT=0.1;

%In Anlehnung an: http://www.mathworks.com/support/tech-notes/1700/1702.html

% This is the length of the vector
NFFT = length(values));
% Take fft
FFTX = fft(values,NFFT);

% Calculate the number of unique points
NumUniquePts = ceil((NFFT+1)/2);
% FFT is symmetric, throw away second half
FFTX = FFTX(1:NumUniquePts);

% Take the magnitude of fft of x
MX = abs(FFTX);

% Scale the fft so that it is not a function of the length of x
MX = MX/length(values);

% Multiply by 2 to because you threw out the second half of FFTX above
MX = MX*2;

% DC Component should be unique.
MX(1) = MX(1)/2;
% Here NFFT is even; therefore, Nyquist point is included.
MX(end) = MX(end)/2;
% This is an evenly spaced frequency vector with NumUniquePts points.
f = (0:NumUniquePts-1)*(1/deltaT)/NFFT;

%plot values
figure
plot(1:length(values), values), grid on
xlabel('Scan Nummer')
ylabel('Amplitude in m')

%plot fft
figure
plot(f, MX), grid on
xlabel('Frequenz in Hz')
ylabel('Amplitude in m')
result = MX;
```
