Universität Ulm

Echtzeitfähiges Fusionssystem zur Fußgängererkennung bei Nacht

Dissertation zur Erlangung des Doktorgrades

Dr.rer.nat.

Fakultät für Ingenieurswissenschaften und Informatik
Universität Ulm

Roland Schweiger aus Kösching

2012

Amtierender Dekan: Prof. Dr.-Ing. Klaus Dietmayer
1. Gutachter: Prof. Dr.-Ing. Klaus Dietmayer
2. Gutachter: Prof. Dr. Heiko Neumann
Tag der Promotion: 05. April 2012

für Kerstin und Lukas

Danksagung

Die vorliegende Arbeit entstand im Rahmen einer Zusammenarbeit zwischen dem Institut für Mess-, Regel- und Mikrotechnik der Universität Ulm und der Abteilung GR/PAP des Vorentwicklungsstandortes der Daimler AG in Ulm.

Für die Annahme der Dissertation, seine Unterstützung und der Möglichkeit im Institut für Mess-, Regel- und Mikrotechnik der Universität Ulm als wissenschaftlicher Mitarbeiter an diesem Thema arbeiten zu können, danke ich Herrn Prof. Dr. Klaus Dietmayer. Insbesondere danke ich ihm für die zahlreichen Möglichkeiten meine Arbeit an internationalen Fachkonferenzen vorstellen zu können.

Für die freundliche Übernahme des Koreferats, das Interesse für meine Arbeit und der Zusammenarbeit bei der Betreuung vieler studentischer Arbeiten danke ich Prof. Dr. Heiko Neumann vom Institut für Neuroinformatik der Universität Ulm.

Seitens der Daimler AG gilt mein Dank vor allem meinem langjährigen Mentor Dr. Werner Ritter, der mir stets mit Rat und Tat zur Seite stand, sowie Dr. Otto Löhlein für seine außerordentlich wertvolle wissenschaftliche Betreuung und Freundschaft. Mein Dank geht insbesondere auch an Stefan Hahn, der mir auch nach meinem frühen Eintritt als Mitarbeiter bei Daimler ermöglichte, neben all den Projekten meine Dissertation zu beenden.

Für die vielen Diskussionen, die kollegiale Atmosphere und die daraus entstandenen Freundschaften möchte ich mich stellvertretend für alle beim Nightview-Dream-Team Markus Thom, Magdalena Szczot, Stefan Franz, Florian Schüle (dem besten Korrekturleser von allen) und Matthias Oberländer bedanken. Ein ganz besonderer Dank gilt dabei Matthias Serfling und Andreas Hallerbach zu denen sich in all den Jahren eine tiefe Freundschaft entwickelt hat.

Mein Dank geht außerdem für ihre studentischen Beiträge an Axel Roth, Ingo Kallenbach, Corvin Idler, Richard Arndt, Henning Hamer, Melanie Nemec und Andreas Hallerbach.

Besonders möchte ich mich bei meinen Eltern für ihre immer währende Ermutigung, Hilfe und verlässliche Unterstützung bedanken.

Zuletzt und nicht am wenigsten, sondern am meisten danke ich von ganzem Herzen meiner Frau Kerstin für ihre Liebe, für ihr Verständnis, ihre bedingungslose Unterstützung und dafür, dass sie meinem Sohn Lukas meine stark reduzierte Zeit für ihn immer wieder erklärt hat.

Inhaltsverzeichnis

Symbolverzeichnis

Punkte und Kamerasysteme

$\boldsymbol{p} = (\mathrm{col}, \mathrm{row})^{\mathrm{T}}$	Ein Punkt im Bild (Pixelkoordinaten), Seite 37
${}^{\mathrm{c}}\boldsymbol{P} = ({}^{\mathrm{c}}X, {}^{\mathrm{c}}Y, {}^{\mathrm{c}}Z)^{\mathrm{T}}$	Ein Punkt im Kamerakoordinatensystem, Seite 36
${}^{\mathrm{v}}\boldsymbol{P} = ({}^{\mathrm{v}}X, {}^{\mathrm{v}}Y, {}^{\mathrm{v}}Z)^{\mathrm{T}}$	Ein Punkt im Fahrzeugkoordinatensystem, Seite 35
$\boldsymbol{R}$	Rotationsmatrix, Seite 36
ϑ, ψ, ϕ	Nickwinkel,Gierwinkel und Wankwinkel, Seite 36
$\boldsymbol{\mathcal{P}}$	Projektionsmatrix, Seite 37
$\boldsymbol{l}_{\mathrm{up}}$	Epipolarlinie, Seite 43

Hilfsfunktionen im Umgang mit Hypothesen

$(\mathrm{col}, \mathrm{row}, h) = \mathrm{proj}_H({}^{\mathrm{v}}\boldsymbol{P}; \boldsymbol{\mathcal{P}})$	Projektion eines Fußgängers der Größe H und Scheitelpunkt ${}^{\mathrm{v}}\boldsymbol{P}$ ins Bild, Seite 39
$\mathrm{row} = \mathrm{reproj}_H(\mathrm{col}, h\,;\, \boldsymbol{\mathcal{P}})$	Hilfsfunktion zur Bestimmung von Fußgängerhypothesen im Bild, Seite 85
$o = \mathrm{proj\_stream2stream}_H(o^*\,;\, \boldsymbol{\mathcal{P}}^*, \boldsymbol{\mathcal{P}})$	Abbild eines Fußgängers aus dem Bild des Primärsensors im Bild des Sekundärsensors, Seite 91
$\mathrm{cov}(A, B)$	Überdeckungsmaß, Seite 47
$\mathrm{cov}_{\max}(A, B)$	Maximum-Überdeckungsmaß, Seite 47

Hypothesen und Label

$x \in \mathcal{X}$	Objekthypothese (Tupel von Suchfenstern), Seite 45
$y \in \mathcal{Y} = \{-1, +1\}$	Klassenlabel, Seite 46

$s = (\mathrm{col}', \mathrm{row}', h')$	Suchfenster (definiert einen rechteckigen Bildausschnitt), Seite 45
$o = (\mathrm{col}, \mathrm{row}, h)$	Objektfenster (definiert einen rechteckigen Bildausschnitt), Seite 46
$s = \chi(o)$	Funktion zur Umrechnung zwischen Objektfenster o und Suchfenster s, Seite 46
$Q_{\mathrm{col}}, Q_{\mathrm{row}}, Q_h$	Parametrisierung der skalierungsabhängigen Unterabtastung im Suchraum, Seite 89
$\mathcal{H}_{\mathrm{NIR}}(Q_{\mathrm{col}}, Q_{\mathrm{row}}, Q_h)$	Hypothesenmenge mit skalierungsabhängiger Unterabtastung, Seite 89
$k^{(l)}$	Schwellwert im Hypothesenbaum, Seite 98

Klassifikation

$m_t \colon \mathcal{X} \to \mathbb{R}$	Merkmalswert innerhalb eines Weaklearners, Seite 57
$h_t \colon \mathcal{X} \to \{-1, +1\}$	Weaklearner, Seite 57
$H \colon \mathcal{X} \to \{-1, +1\}$	Stronglearner, Seite 57
θ, Θ	Weaklearnerschwelle, Stronglearnerschwelle, Seite 57
$T, T_k^{\max}$	(vorgegebene maximale) Anzahl Weaklearner eines Stronglearners, Seite 57
$d_t^{(i)}$	Gewichte der Trainingsbeispiele, Seite 59
$A \colon \mathcal{X} \to \mathbb{R}$	Aktivierung, Seite 57
α_t	Gewicht eines Weaklearners innerhalb des Stronglearners, Seite 57
Z_t	Normalisierungsfaktor zur Normalisierung der Trainingsgewichte in AdaBoost, Seite 59
$\rho \colon \mathbb{R}^T \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$	Klassifikationsmargin eines Stronglearners, Seite 64
D, D_k, D_k^*	Detektionsraten, Seite 68
F, F_k, F_k^*	Falschalarmraten, Seite 69

Wahrscheinlichkeiten

$p(\cdot\|\cdot)$	bedingte Wahrscheinlichkeitsdichte, Seite 61
$p_{\mathrm{emp}}(\cdot\|\cdot)$	empirische Wahrscheinlichkeiten im PBT, Seite 73
$q(\cdot\|\cdot)$	bedingte Wahrscheinlichkeitsdichte, Rückschlusswahrscheinlichkeit, Seite 72

Zustandsschätzung und Partikelfilter

$\boldsymbol{x}_i \in \mathbb{R}^{N_x}$	Systemzustandsvektor zum Zeitpunkt i, Seite 104
$\boldsymbol{x}_{1:i}$	Menge der Systemzustände $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i$, Seite 104
$\boldsymbol{z}_i \in \mathbb{R}^{N_z}$	Messvektor zum Zeitpunkt i, Seite 104
$\boldsymbol{x}_i = \boldsymbol{f}_i(\boldsymbol{x}_{i-1}, \cdot)$	Systemmodell (Zustandsmodell 1. Ordnung), Seite 104
$\boldsymbol{z}_i = \boldsymbol{h}_i(\boldsymbol{x}_i, \cdot)$	Beobachtungsmodell, Seite 104
$\Xi_i = \{\xi_i^{(1)}, \ldots, \xi_i^{(N_s)}\}$	Menge gewichteter Partikel zum Zeitpunkt i, Seite 106

$\xi_i^{(j)} = (\boldsymbol{x}_i^{(j)}, w_i^{(j)})$	gewichteter Partikel im Partikelfilter, Seite 106
$p_{\text{prop}}(\boldsymbol{x}_i \vert \boldsymbol{z}_{1:i})$	Vorschlagsfunktion, engl. 'proposal distribution', Seite 107
$g\left(\boldsymbol{x}_i\right)$	Gewichtsfunktion zur Bewertung der Partikel im Partikelfilter, Seite 112

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmen

KAPITEL 1

Einleitung

Im Jahr 2007 ist in Deutschland erstmals die Zahl der Verkehrstoten unter 5 000 gesunken. Damit hat sich die Zahl der im Straßenverkehr tödlich verunglückten Personen seit dem Jahr 2000 um über ein Drittel (34%) verringert [Vor08]. Dieser Trend ist umso erfreulicher, da sich im selben Zeitraum der Bestand der motorisierten Kraftfahrzeuge (und damit auch das Verkehrsaufkommen insgesamt) um weitere 4.3 Millionen Fahrzeuge auf 57.4 Millionen Fahrzeuge erhöht hat. Diese positive Entwicklung lässt sich vor allem auf die breite Verfügbarkeit elektronischer Sicherheitssysteme im Fahrzeug, wie z.B. Airbags, ABS und ESP zurückführen.

Nachtsichtsysteme der ersten und zweiten Generation

Seit einigen Jahren zeichnet sich außerdem ein Forschungstrend in der Entwicklung sicherheitsrelevanter Systeme ab, die durch die Erfassung des Fahrzeugumfeldes das Unfallrisiko weiter senken werden [Ulm04]. Gerade unter schlechten Sichtverhältnissen wie z.B. bei Nacht können solche Systeme den Fahrer unterstützen. Eines dieser Systeme ist der Nachtsichtassistent, der seit 2005 in der Mercedes Benz S-Klasse als Sonderausstattung verfügbar ist [mer]. Um dem Fahrer bei Nacht eine bessere Sicht zu ermöglichen, wird die Szenerie vor dem Fahrzeug mit Infrarotlicht ausgeleuchtet, mit einer infrarotempfindlichen Kamera (Nahinfrarotkamera, kurz: NIR-Kamera) aufgenommen und dem Fahrer in seinem Sichtfeld als Grauwertbild in der Instrumententafel angezeigt. Während Abblendscheinwerfer nur eine begrenzte Sichtweite ermöglichen, ist der Sichtbereich mit diesem Nachtsichtsystem mehr als doppelt so groß (Abbildung 1.1). Eine Blendung des Gegenverkehrs ist ausgeschlossen, da das Infrarotlicht mit seinem Wellenlängenbereich mit über 780nm für das menschliche Auge nicht wahrnehmbar ist.

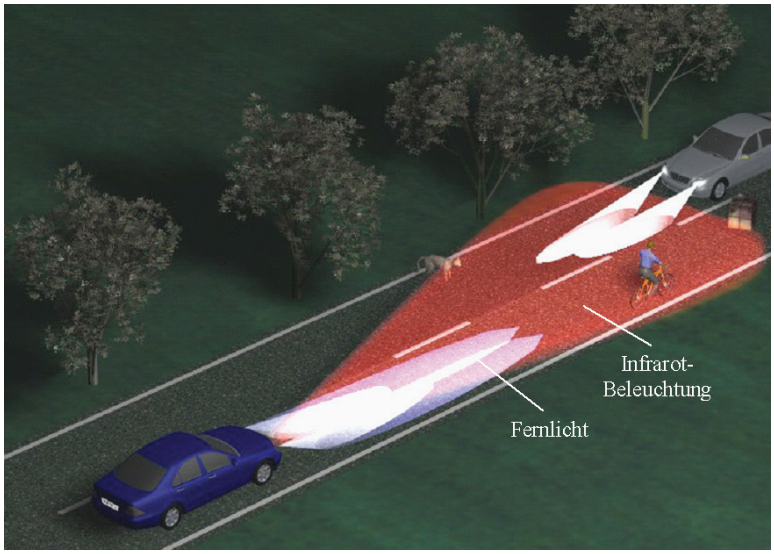


Abbildung 1.1.: Nachtsichtassistent von Mercedes. Eine Kamera filmt die durch einen blendfreien Infrarotscheinwerfer zusätzlich beleuchtete Fahrstrecke und sendet ihr Bild in Graustufendarstellung an das Multifunktionsdisplay. Quelle: [BB07].

Das Nachtsichtsystem von BMW („BMW Night Vision“, Markteinführung 2006, [bmw]) kommt ohne aktive Beleuchtung aus. Dem Fahrer wird in diesem System das Bild einer Wärmebildkamera (Ferninfrarotkamera, kurz: FIR-Kamera) präsentiert, so dass wärmere Objekte (z.B. Lebewesen) sich von ihrer kälteren Umgebung deutlich absetzen. Dieses künstliche Bild hat den Vorteil, dass potentiell gefährdete Verkehrsteilnehmer, z.B. Fußgänger, leicht vom Fahrer erkannt werden können. Der Nachteil liegt in der deutlich schlechteren Bildqualität und der für viele Fahrer ungewohnten technischen Darstellung, die die Zuordnung der möglichen Gefahrenpunkte zum tatsächlichen Fahrgeschehen erheblich erschweren.

Neben der reinen Sichtverbesserung bei Nacht durch darstellende Systeme gewinnt zunehmend auch der Fußgängerschutz bei Nacht an Bedeutung. Entgegen dem Trend ist der Nachtanteil der verunglückten Fußgänger seit 1991 nämlich nicht zurückgegangen [LAE05]. Dementsprechend erweitern die seit kurzem verfügbaren Nachtsichtsysteme der 2. Generation den rein darstellenden Aspekt um die Komponente einer Fußgängererkennung (Abbildung 1.2). Damit soll der Fahrer auf Fußgänger vor dem Fahrzeug direkt aufmerksam gemacht werden. Das FIR-basierte Nachtsichtsystem von BMW ist dabei im Gegensatz zum NIR-basierten System von Mercedes auch in der Lage Fußgänger in stark strukturierter Umgebung wie z.B. Innenstädten sicher zu detektieren. Dies ist vor allem durch den Wärmebildsensor begründet, der bei entsprechend kalter Umgebung in der Nacht generell besser zur Detektion von warmen Lebewesen geeignet ist.

Da ein erkannter Fußgänger im FIR-Bild vom Fahrer unter Umständen nur schwer zum Fahrgeschehen zugeordnet werden kann, ist im BMW-System ein einfaches Warnkonzept umgesetzt, das nur dann eine Warnung auslöst, wenn Fußgänger direkt im Bereich vor dem Fahrzeug erkannt werden oder Fußgänger diesen gerade betreten. Das NIR-

Abbildung 1.2.: Nachtsichtsysteme der 2. Generation. Oben das System von Mercedes-Benz, das Fußgänger im Bild graphisch hervorhebt. Unten das System von BMW, das durch ein Warnsymbol vor Fußgängern auf der Fahrbahn warnt.

System hat dagegen den Vorteil, durch die gute Bildqualität auch ohne eventuell störende Warnsignale auf Fußgänger im Gefahrenbereich aufmerksam machen zu können. Erkannte Fußgänger werden lediglich im Bild graphisch hervorgehoben um die Erkennung für den Fahrer zu erleichtern. Aufgrund des natürlichen Bildeindrucks kann der Fahrer die Gefahrensituation selbst sehr gut abschätzen.

Beide Systeme können Fußgänger bis in eine Entfernung von 90m vor dem Fahrzeug erkennen. Diese Grenze liegt beim FIR-System in der niedrigen Bildauflösung begründet. Beim NIR-System beschränkt die hohe benötigte Rechenleistung die Reichweite. Darüber hinaus sind beide Systeme nicht für den Einsatz bei Dämmerung geeignet. Die Algorithmen basierend auf dem NIR-Sensor sind dabei aufgrund der vielen Strukturen im Bild überfordert. Die Algorithmen basierend auf dem FIR-Sensor verlieren dagegen aufgrund der meist noch hohen Umgebungstemperatur in der Dämmerung ihre Trennkraft.

Zielstellung: Grundlage eines Nachtsichtsystems der dritten Generation

Nachtsichtsysteme der dritten Generation sollen in Zukunft dem Fahrer noch umfassender helfen, gefährliche Situationen in der Dunkelheit besser zu beurteilen, um früher auf Personen auf der Fahrbahn reagieren zu können. Solche zukünftigen Systeme erfordern

neben der notwendigen Sicherheit - also niedriger Falschalarmrate - sehr hohe Detektionsleistungen, um die Fahrzeugumgebung nicht nur sicher, sondern auch vollständig zu erfassen. Außerdem sollen Beschränkungen aufgrund zu hoher Temperaturen oder im Dämmerungsbereich wegfallen und Detektionen auch in größerer Entfernung als 90m möglich sein.

Um diesen Ansprüchen zu genügen, reicht ein einziger Sensor nicht mehr aus. Gegenstand dieser Arbeit ist deshalb die Realisierung eines Sensorfusionssystems zur echtzeitfähigen Fußgängererkennung als eine mögliche Basis für Nachtsichtsysteme der dritten Generation. Die Erkennung der Fußgänger findet dabei auf Basis der Nahinfrarotkamera und der Wärmebildkamera statt. Die Verknüpfung beider Sensoren erfolgt auf Merkmalsebene. Diese ermöglicht es, zu einer geeigneten Repräsentation der gesuchten Objekte zu gelangen, um somit die gesteigerte Komplexität des Detektionsproblems zu erfassen.

Die Auswahl und Parametrisierung der Merkmale erfolgt dabei - statistisch fundiert - mit einem Klassifikationsverfahren. Durch den unterschiedlichen Einbauort der Kameras und des dadurch entstehenden Parallaxeproblems war bisher eine gezielte Fusion auf Merkmalsebene nicht (oder nur unter hohem Rechenaufwand) möglich, da die einzelnen Merkmale der unterschiedlichen Bildausschnitte einander nicht zugeordnet werden konnten. Um sich dieser Problematik zu stellen und gleichzeitig die Echtzeitfähigkeit des Systems zu gewährleisten, werden in dieser Arbeit neuartige Suchstrategien vorgestellt, die die Eigenschaften der eingesetzten Klassifikatoren gezielt ausnutzen. Dazu wird unter anderem ein Partikelfilterverfahren entwickelt, das die Suche dynamisch über ganze Bildfolgen hinweg organisiert. Mit den unterschiedlichen Suchverfahren ist es möglich - trotz des durch den fusionierten Sensorraum erhöhten Suchaufwandes - einen Fußgängerdetektor zu realisieren, der bei deutlich besserer Detektionsleistung nicht viel mehr Rechenaufwand benötigt, als ein Einzelsensorsystem.

1.1. Stand der Technik und Beitrag der Arbeit

Verfahren zur Erkennung von Fußgängern in Videobildern gehören zu den zentralen Themen im Bereich Computer Vision und Mustererkennung. Einen guten Überblick aktueller Arbeiten sowie Vergleiche unterschiedlicher Ansätze im Fahrerassistenzbereich geben [Enz11, GLSG10, EG09] und [GT07]. Ein Fußgängererkennungssystem besteht dabei im allgemeinen aus den Komponenten einer Vorverarbeitung mit der Auswahl möglicher Objekthypothesen, der eigentlichen Klassifikationskomponente und einem Nachverarbeitungsschritt zur zeitlichen Filterung (Tracking) der Klassifikationsergebnisse [Enz11]. Letzterer ist nicht Bestandteil dieser Arbeit und wird deshalb hier nicht berücksichtigt.

Die Auswahl möglicher Objekthypothesen erfolgt im einfachsten Fall über eine erschöpfende Suche im Bild, d.h. es werden Objekthypothesen an allen Positionen im Bild erzeugt und von der Klassifikationskomponente validiert. Damit Fußgänger auch in unterschiedlichen Größen (Skalen) erkannt werden können, werden entweder

die Bilder bzw. Bildausschnitte in entsprechende Skalenrepräsentationen gebracht [DT05, DT06, WDSS08, PP00] oder die Objekthypothesen werden auch für alle möglichen Skalen erzeugt [VJS03, LBH08, ZAYC06, OTF+04]. Die Verfahren mit der erstgenannten Methode verwenden dann zwar Merkmale mit gleichbleibender Größe, müssen jedoch Multiskalenrepräsentationen berücksichtigen, wogegen die anderen Verfahren auf effizient skalierbare Merkmale angewiesen sind.

Eine erschöpfende Suche im Bild ist meist sehr aufwändig, deshalb wird in vielen Arbeiten der Suchraum bereits vor der Anwendung eines Klassifikators eingeschränkt, indem Objekthypothesen auf Basis einer initialen Segmentierung in Vordergrund (mögliche Objekte, also Fußgänger) und Hintergrund eingeteilt werden. Im Fahrerassistenzbereich geschieht dies in vielen Fällen auf Basis der Segmentierung von Stereokorrespondenzen [FK96, FGG+98, GM07, GGM04, ALS+07, SPFN06, SR06] oder unter Ausnutzung der Bewegung im Bild, sowohl merkmalsbasiert (engl. „Structure from Motion“, [LCCV07, LSCV08]) oder auch auf Objektebene, z.B. [LF04]. Diese Vorgehensweisen scheiden für den Anwendungsfall dieser Arbeit aus, da verlässliche Stereokorrespondenzen oder Flussmerkmale im Nachtsichtbild für den angestrebten Entfernungsbereich bis 130 m in der Regel nicht - oder nur bei großem technischen Aufwand (z.B. einem NIR-Stereosystem mit sehr großer Basisbreite oder hoch aufgelösten FIR-Sensoren) - verfügbar sind. In dieser Arbeit erfolgt die Auswahl möglicher Objekthypothesen deshalb über eine klassische Suche im Bild. Der größere Suchraum wird dabei zum einen durch gezielte Modellierung der Problemstellung eingeschränkt und zum anderen die Anzahl der zu prüfenden Hypothesen über neue, effiziente Suchstrategien deutlich reduziert.

Nach der Auswahl möglicher Objekthypothesen entscheidet eine Klassifikationskomponente, welche der Hypothesen „Hintergrund“ darstellen oder tatsächlich Fußgänger repräsentieren. Generell kann man zwischen zwei grundsätzlichen Ansätzen unterscheiden: auf Form bzw. Kontur basierende Methoden und erscheinungsbasierte Techniken.

Die bekannteste konturbasierte Methode ist sicherlich das sogenannte Chamfer-Matching [Gav00, GGM04, GM07], das einen Abgleich der Bildausschnitte mit Konturmodellen aus einer Hierarchie von Konturen vornimmt, die offline anhand von Beispielen erzeugt wurden. Der Abgleich basiert dabei auf einer Distanztransformation, der sogenannten Chamfer-Distanz [Bor86]. Über die Hierarchie der Konturen wird zunächst lediglich ein grobes Konturenmodell abgeglichen, um dann hierarchisch immer feinere Modelle anzuwenden. Dieses Vorgehen wird auch in [MOL+05] auf Basis von FIR-Bildern angewandt. Bei FIR-Bildern sind jedoch einfache binäre Konturenmodelle - meist über Korrelationstechniken - weiter verbeitet, z.B. [BBFS00] mit Fokus auf die in FIR-Bildern sehr ausgeprägte Kopf-Schulter-Kombination oder auch in [BFT06] unter Ausnutzung der Beinstellung. [ND02] adressiert mit probabilistisch motivierten Grauwerteschablonen vor allem das Problem, dass in FIR-Bildern Einzelteile von Fußgängern nicht immer mit ausreichendem Kontrast sichtbar sind (z.B. wegen isolierender Kleidung). [BBGM07] geht einen ähnlichen Weg, allerdings mit unterschiedlichen Masken für unterschiedliche Beinstellungen.

Generell eignen sich konturbasierte Verfahren mehrheitlich für die Detektion von (hochaufgelösten) Fußgängern im Nahbereich. Selbst in diesem Umfeld werden aber diese

Verfahren meist mit erscheinungsbasierten Methoden gekoppelt [MG06, GM07, EG09]. Für den Anwendungsfall dieser Arbeit, der Detektion von Fußgängern in sehr großen Entfernungen, sind diese Verfahren vor allem aufgrund der schlecht augelösten FIR-Bilder wenig geeignet. Darüber hinaus lassen sich in konturbasierten Verfahren mehrere Bildsensoren nicht auf früher Ebene miteinander kombinieren.

Im Gegensatz zu konturbasierten Verfahren lernen erscheinungsbasierte Verfahren anhand von Trainingsbeispielen die Parameter einer Entscheidungsfunktion (Klassifikator), um Fußgänger und Hintergrund voneinander zu trennen. Dazu kommen unterschiedliche Klassifikationstechniken zum Einsatz, die ihre Entscheidung wiederum auf eine große Zahl möglicher Merkmalstypen stützen. Als weit verbreitete Methode ist hier sicherlich die SVM (engl. „Support Vector Machine“, siehe z.B. [Bis06]) zu nennen. Dabei handelt es sich um einen Klassifikator, der die Merkmalsrepräsentation der Menge der positiven Beispiele (Fußgänger) und negativen Beispiele (Hintergrund) über eine Hyperebene (lineare SVM) bzw. eine Hyperfläche (nicht-lineare SVM) so trennt, dass der Abstand beider Klassen im Merkmalsraum maximal wird. [PP00] und [MPP01] verwenden nicht-lineare SVMs in Verbindung mit sogenannten Haarwavelet Merkmalen. Diese Merkmale bilden die Grauwertedifferenz benachbarter, rechteckiger Bildregionen, sind einfach und effizient mit Hilfe von Integralbildern [Cro84] zu berechnen und werden auch in dieser Arbeit zur Klassifikation verwendet. Ebenfalls nicht-lineare SVMs und Haarwavelets verwenden [SPFN06] und [ALS+07], allerdings in Kombination mit vielen weiteren Merkmalen, unter anderem den Einträgen eines Canny-Kantenbildes, Gradientenbeträge und -orientierungen sowie weitere Texturmerkmale. Speziell zur Detektion von Fußgängern bei Nacht in NIR-Bildern mit aktiver Beleuchtung nutzen [ABDL05] eine Kette von SVMs, jeweils mit einer Auswahl von Wavelet-Koeffizienten nach einer Wavelet-Transformation der Eingabebilder [Str01]. Im Kontext der Detektion von Fußgängern in FIR-Bildern nutzt [XLF05] SVMs in Verbindung mit Hot-Spot-Merkmalen, die heiße Regionen im FIR-Bild über Schwellwerte identifizieren.

Dalal und Triggs [DT05, DT06] nutzen mit großem und vielzitiertem Erfolg lineare SVMs in Verbindung mit HOG Merkmalen (engl. „Histogram of Oriented Gradients“). Dabei handelt es sich um Merkmale, die auf normalisierten Histogrammen lokal berechneter Intensitätsgradienten basieren. Dazu werden in lokalen Umgebungen Gradientenhistogramme über deren Orientierungen aufgestellt und in Relation zum Konrast in übergeordneten Flächen normalisiert. Diese Histogramme bilden damit die vorherrschenden Richtungen ab und repräsentieren die Form der Kanten in der entsprechenden Umgebung [ZZ10]. Eine echtzeitfähige Implementierung auf GPU-Basis findet sich beispielsweise in [WDSS08]. Eine Anwendung des Ansatzes von Dalal und Triggs auf FIR-Bildern wird in [SR06] und [ZWN07] beschrieben. Letztere verwenden Kaskaden von SVM-Klassifikatoren und neben HOG Merkmalen sogenannte Edgelet Merkmale, die Kurvenstücke der Linien im Bild repräsentieren.

Generell haben SVMs vor allem den Vorteil einer exzellenten Klassifikationsgüte. Nachteile sind jedoch ein sehr hoher Aufwand beim Training, eine schlechte Nachadaption mit neuen Daten, eine sehr hohe Speicheranforderung vor allem bei großen Datensätzen

und ein signifikanter Berechnungsaufwand bei der Klassifikation im nicht-linearen Fall.

Neben dem SVM-Klassifikator werden auch neuronale mehrschichtige feedforward-Netze (siehe z.B. [Bis06]) zur Klassifikation von Fußgängern in Kamerabildern verwendet. Diese bilden lineare Entscheidungsfunktionen im Merkmalsraum ab, in dem die Eingansdaten über nicht lineare Funkionen in der verdeckten Schicht transformiert wurden. Neuronale Netze zur Detektion von Fußgängern werden oft zusammen mit den bereits beschriebenen Merkmalen benutzt, aber auch direkt auf den Grauwerten des Hypothesenrechtecks [SYYO05], auf Gradientenbeträge [ZT00] und/oder in Verbindung mit lokalen rezeptiven Feldern in der verdeckten Schicht [WA99, MG06, GM07]. Diese Netze können eine gute Klassifikationsperformanz erreichen und können gut mit neuen Daten nachadaptiert werden (Gradientenabstiegsverfahren beim Training). Allerdings ist ein hoher Betreuungsaufwand beim Training nötig, da viele Parameter und Konfigurationen per Hand voreingestellt oder über aufwändige Bootstrapping-Verfahren und Parameter-Optimierungen evaluiert werden müssen. Oft sind erstellte erfolgreiche Topologien und Parametersätze bereits für nur leicht anders situierte Probleme nicht mehr optimal.

Die dritte, wichtige und weit verbreitete Klassifikationsarchitektur zur Detektion von Fußgängern in Bildern ist die Verwendung einer sogenannten AdaBoost-Kaskade, die von Viola und Jones [VJ01a] bereits sehr erfolgreich zur Gesichtserkennung eingesetzt wurde. Dazu werden mehrere „starke“ Klassifikatoren hintereinander geschaltet und die Anzahl der Objekthypothesen sukzessive ausgedünnt. Damit lassen sich auch große Mengen an Hypothesen pro Bild prüfen, da in den ersten Stufen bereits mit wenigen Merkmalen eine große Anzahl an Hypothesen als Hintergrund erkannt und verworfen werden kann. Jede Stufe wird dabei mit AdaBoost [FS97, SS99] trainiert, einem Greedy-Algorithmus, der iterativ einfache Klassifikatoren, die in der Regel jeweils lediglich aus einem Merkmal mit Schwellwert bestehen, zu einem Klassifikator kombiniert. Jeder dieser einfachen Klassifikatoren fokussiert sich dabei auf die bisher falsch klassifizierten Trainingsbeispiele. Im Kontext der Fußgängererkennung in Videobildern wurden AdaBoost-Kaskaden erstmals von [VJS03] vorgestellt, in Verbindung mit Haarwavelet Merkmalen, die auf Flussvektorbildern angewandt werden. [JS08] optimiert und erweitert denselben Ansatz um die ursprünglichen klassischen Haarwavelet Merkmale, die direkt auf Grauwertbilder angewandt werden. Ebenfalls klassische Haarwavelet Merkmale benutzt [MOL$^+$05] zur Detektion von Fußgängern in FIR-Bildern.

Nach dem großen Erfolg der HOG Merkmale von Dalal und Triggs haben sich diese auch im Zusammenhang mit AdaBoost etabliert: [ZAYC06] beispielsweise nutzen dazu HOGs mit 9 Orientierungen pro Histogramm, jeweils auf Basis unterschiedlicher Blockgrößen. Die Autoren übertragen dabei das Prinzip der Integralbilder zur effizienten Berechnung von Haarwavelets auch auf die Berechnung der HOG-Deskriptoren unterschiedlicher Größe und Konfiguration. AdaBoost wählt im Training dann iterativ die jeweils geeignetsten HOG-Deskriptoren aus. [GSLP07] benutzen neben Haarwavelets auch HOG-ähnliche Merkmale, sogenannte EOH Merkmale (engl. „Edge Orientation Histograms“, [LW04]). Diese Merkmale setzen die einzelnen Komponenten des Gradientenhistogramms - also die einzelnen Gradientenrichtungen - über Quotienten in

Beziehung zueinander. Wie in [ZAYC06] können auch sie effizient über Integralbilder realisiert werden. Schließlich werden auch in [ZWN07] die dort bereits im Zusammenhang mit SVMs benutzen Edgelet Merkmale auch mit AdaBoost-Klassifikatoren eingesetzt.

Generell liegt der Vorteil von AdaBoost-Kaskadenarchitekturen in der großen Geschwindigkeit bei der Klassifikation. Sie lassen sich leicht trainieren und haben im Gegensatz zu neuronalen feedforward-Netzen nur wenige Parameter, die justiert werden müssen. Nachadaption ist generell zwar schwierig, zumindest aber zur weiteren Reduktion von Falschalarmen durch Hinzufügen weiterer Stufen möglich. Nachteile sind eine leicht schlechtere Klassifikationsperformanz im Vergleich zur nicht-linearen SVM und die leichte Gefahr einer Überadaption. Allerdings kann auch eine AdaBoost-Kaskade in den letzten Stufen durch eine SVM unterstützt werden.

AdaBoost wählt im Training ihre Merkmale aus einem großen, meist überbestimmten Merkmalssatz sukzessive aus. Die Wahl, welche Merkmale konkret für die Klassifikation verwendet werden sollen, erfolgt damit auf statistischer Grundlage. Dies ist vor allem dann von Vorteil, wenn mehrere Sensoren kombiniert werden und von vornherein nicht klar ist, welche Merkmale von welchem Sensor signifikant sind. Vor allem wegen letztgenannter Eigenschaft erfolgt die Klassifikation in dieser Arbeit mittels einer AdaBoost-Kaskade, die entsprechend zu einem Fusionsframework zur Kombination der FIR- und NIR-Merkmale erweitert wird.

Folgt man den aktuellen Vergleichen [WDSS08, EG09, GLSG10], muss die Wahl in Bezug auf die verwendeten Merkmale sicherlich auf HOG Merkmale fallen. Dennoch werden diese Merkmale in dieser Arbeit nicht verwendet, denn wie auch [JS08] bemerkt, liegt der Fokus der meisten Arbeiten mit HOG Merkmalen auf dem nahen oder mittleren Entfernungsbereich, meist mit gut aufgelösten Bildern. Auch der direkte Vergleich einer Haarwavelet basierten Kaskade mit einer linearen SVM mit HOG Merkmalen in [EG09] zeigt, dass für schlecht aufgelöste Fußgänger-Beispiele mit etwa $18\,\mathrm{px} \times 36\,\mathrm{px}$ Haarwavelet Merkmale die bessere Option darstellen. In vorliegender Arbeit sind die Bildausschnitte sogar noch weit kleiner, nämlich $7\,\mathrm{px} \times 11\,\mathrm{px}$ im FIR-Bild für einen 1.80 m großen Fußgänger in 130 m Entfernung. Aus den genannten Gründen fällt die Wahl der Merkmale in dieser Arbeit auf Haarwavelet-ähnliche Merkmale. Es sei noch anzumerken, dass die Wahl dieses Merkmalstyps keine signifikate Einschränkung des gesamten Verfahrens darstellt, da die Klassifikationsarchitektur ohne Probleme auch um weitere Merkmale ergänzt oder sogar mit komplett unterschiedlichen Merkmalssätzen betrieben werden kann.

Trotz der zunehmenden Verbreitung mehrerer Sensoren im Fahrzeug gibt es bisher vergleichsweise wenige Arbeiten über Fusionsarchitekturen mehrerer Bildsensoren speziell zur Detektion von Fußgängern im Fahrerassistenzbereich. Die Herausforderung dabei ist die Registrierung der unterschiedlichen Bildmerkmale aufeinander [GT07]. In vielen Fällen wird zur Registrierung zuvor die Tiefeninformation auf einem Stereopaar verwendet: [KT07b] und [KT07a] im Zusammenhang mit HOG-ähnlichen Merkmalen und SVM, [BBF+06] mit zwei Stereo-Systemen, einer Hypothesengenerierung auf Basis der Disparitäten und abschließender Klassifikation auf den FIR-Bildern.

In vorliegender Arbeit wird dagegen erstmals eine Fusionsarchitektur vorgestellt, die ohne einen direkten Registrierungsschritt auskommt. Dem Klassifikator werden dabei auch nicht registierte Bildausschnitte präsentiert, die von diesem jedoch als Hintergrund verworfen werden. Die Bildregistrierung erfolgt damit implizit.

Beitrag der Arbeit

In dieser Arbeit wird ein Fußgängererkennungssystem entwickelt, das auf Merkmalsbasis einen Nahinfrarotsensor und einen Ferninfrarotsensor kombiniert. Die Arbeit umfasst dabei ...

- ... die theoretischen Grundlagen eines Kaskadenklassifikators, der zur Fusion auf Merkmalsbasis eingesetzt wird.
- ... die neu entwickelte Bestimmung von Rückschlusswahrscheinlichkeiten auf Basis des Kaskadenklassifikators. Dies ermöglicht den Einsatz solcher Klassifikatoren in probabilistischen Verfahren, z.B. einem Partikelfilter.
- ... die Entwicklung einer einfachen Suchstrategie auf Basis geeigneter Weltmodelle, um den Suchraum effizient einschränken zu können.
- ... die Optimierung des Suchraums durch Umsetzung einer grob-zu-fein Suchstrategie, die die Eigenschaften von Kaskadenklassifikatoren ausnutzt.
- ... ein Verfahren zur probabilistischen Mehrobjektverfolgung mit Partikelfilter, das es durch den Einsatz mehrerer Partikelfilterinstanzen, einer einfachen Datenassoziation sowie einer speziellen Abarbeitungsstrategie ermöglicht, mehrere (Fußgänger-)ziele gleichzeitig zu modellieren.
- ... die Umsetzung einer probabilistisch motivierten Suchstrategie unter Verwendung eines Partikelfilterverfahrens, das die Suche nach Fußgängern im Bild dynamisch über ganze Bildfolgen hinweg organisiert.

1.2. Systemübersicht

Das Fußgängererkennungssystem besteht im wesentlichen aus drei Stufen (siehe Abbildung 1.3). In der ersten Stufe (I - Hypothesengenerierung) werden mögliche Hypothesen erzeugt, für die dann in der zweiten Stufe (II - Merkmalsberechnung) Merkmale berechnet werden. Die dritte Stufe (III - Klassifikation) führt die Klassifikation durch und entscheidet, ob eine der Hypothesen einen Fußgänger darstellt oder nicht. Das Detektionsproblem wird also hypothesengetrieben angegangen, indem an allen sinnvollen Stellen im Suchraum Hypothesen generiert werden, die dann anhand der Messungen (also der Bildmerkmale) vom Klassifikator validiert werden.

Basiert die Detektion der Objekte auf nur einem Bildsensor, können die Hypothesen im Bildraum z.B. durch Suchfenster (das sind rechteckige Bildausschnitte) dargestellt werden. Bei dem Einsatz mehrerer Bildsensoren müssen zur Bildung der Hypothesen

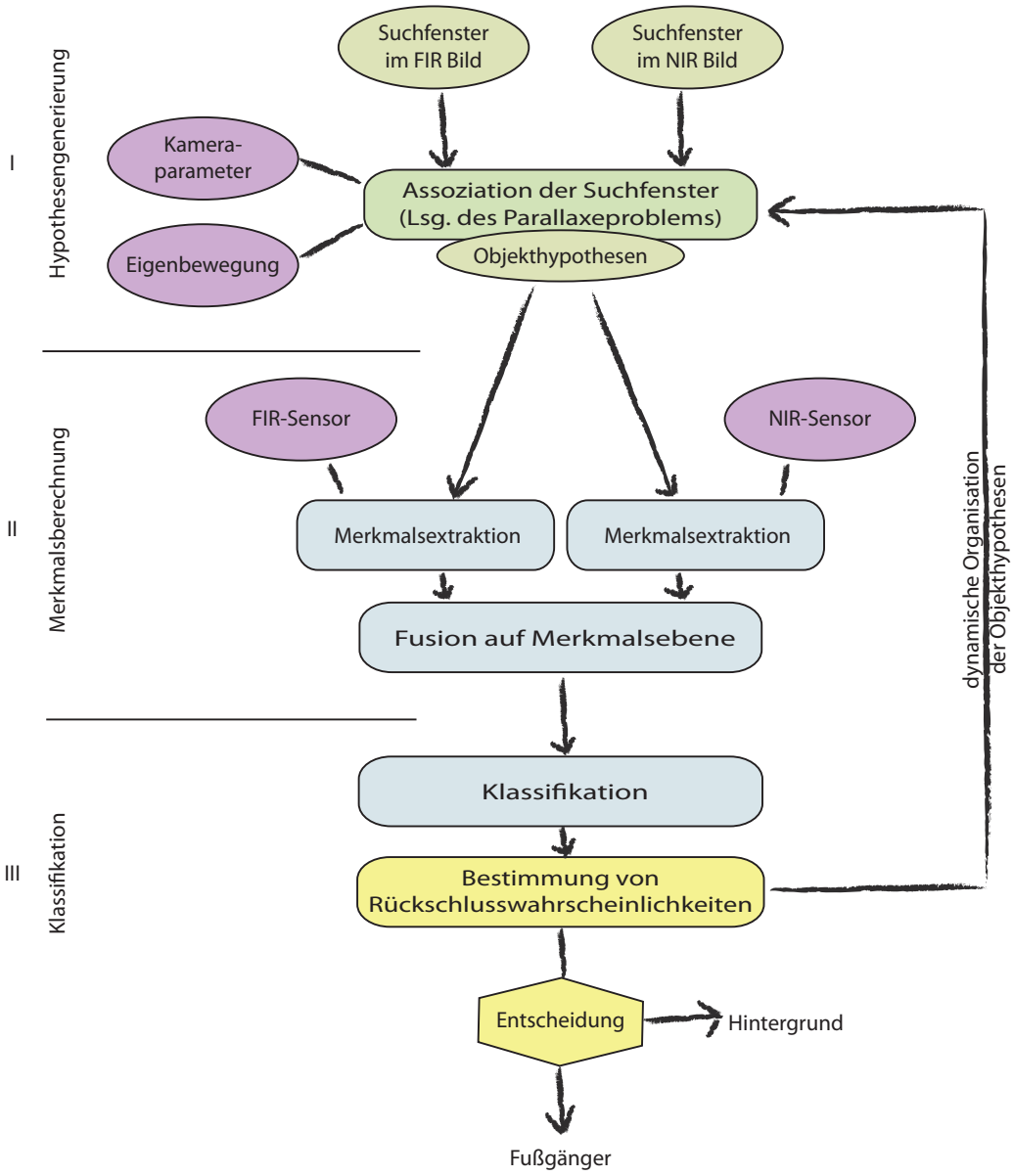


Abbildung 1.3.: Systemübersicht. Im wesentlichen besteht das Erkennungssystem aus drei Stufen. In der ersten Stufe (Hypothesengenerierung) werden Hypothesen erzeugt, für die in der zweiten Stufe (Merkmalsberechnung) Merkmale berechnet werden. Die Merkmale werden dabei zu einer einzigen Merkmalsmenge zusammengefasst. Auf Basis dieser wird in der letzten Stufe (Klassifikation) entschieden, ob es sich bei der jeweiligen Hypothese um einen Fußgänger handelt oder nicht.

die Suchfenster aus den unterschiedlichen Bildern in einem Assoziationsschritt einander zugeordnet werden. Dies geschieht durch ein geeignetes Weltmodell auf Basis der Modellparameter des Kamerasystems. Aufgrund von Mehrdeutigkeiten in der Zuordnung ist die Anzahl der Hypothesen im Fusionssystem im Gegensatz zum Einzelsensorsystem sehr groß. Zur Gewährleistung der Echtzeitfähigkeit werden in dieser Arbeit mehrere Suchstrategien vorgestellt, die im Falle einer Organisation der Hypothesen über ganze Bildfolgen hinweg auch die Eigenbewegung des Fahrzeugs mit berücksichtigen. Nur so können Objekthypothesen im Bild - von einem zum nächsten Zeitschritt - korrekt fortgeschaltet werden. Der Mechanismus zum Erzeugen der Objekthypothesen wird Hypothesengenerator genannt.

Die merkmalsbasierte Fusion findet in der zweiten Stufe des Detektionssystems statt. Von jedem der Sensordatenströme werden Merkmale extrahiert und zu einer einzigen Merkmalsmenge zusammengefasst. In einem Trainingsverfahren werden dann die besten Merkmale aus dieser Menge anhand eines statistischen Verfahrens ausgewählt und zu einem Klassifikator zusammengesetzt. Dabei spielt die Herkunft der Merkmale keine Rolle mehr. Welche Merkmale von welchem Sensor zur Klassifikation herangezogen werden, wird also nicht heuristisch, sondern in einem statistischen Lernverfahren entschieden. Der verwendete Klassifikator ist der bekannte Kaskadenklassifikator von Viola und Jones [VJ01a], deren Einzelklassifikatoren mit AdaBoost [FS97, SS99] trainiert wurden.

Ursprünglich liefert ein solcher Klassifikator nur eine binäre Entscheidung (nämlich „Detektion“ oder „keine Detektion“). In Stufe III des Fußgängererkennungssystems wird deshalb zusätzlich eine Rückschlusswahrscheinlichkeit auf Basis des Klassifikationsergebnisses berechnet. Sie gibt für eine gegebene Objekthypothese an, wie wahrscheinlich sie tatsächlich einen Fußgänger darstellt. Diese Wahrscheinlichkeit fließt dann implizit im Assoziationsschritt des nächsten Zeitschritts mit ein, um so eine probabilistische Suchstrategie über zeitlich aufeinander folgende Bilder zu ermöglichen.

1.3. Gliederung der Arbeit

Die Arbeit ist inhaltlich in sechs Teile untergliedert. Im ersten Teil (Kapitel 2) werden neben grundlegenen Begriffsdefinitionen die unterschiedlichen Kamerasysteme eingeführt und die Probleme der mehrdeutigen Projektion kurz skizziert.

Der zweite Teil der Arbeit (Kapitel 3) befasst sich mit der Klassifikationskomponente des Fußgängererkennungssystems. Neben der algorithmischen Einführung des Verfahrens werden auch die theoretischen Eigenschaften des Trainingsverfahrens dargelegt. Insbesondere wird eine Herleitung von Rückschlusswahrscheinlichkeiten aus dem Klassifikationsergebnis vorgenommen, die es erstmals erlauben in probabilistischen Zustandsschätzern die Ergebnisse eines Kaskadenklassifikators zu berücksichtigen.

Kapitel 4 stellt zwei Hypothesengeneratoren vor, die auf Basis geeigneter Weltmodelle den Suchraum zur Detektion von Fußgängern in sinnvoller Weise einschränken. Zur weiteren Optimierung des Suchraums benutzt der zweite Hypothesengenerator dabei

eine grob-zu-fein Suchstrategie, die spezielle Eigenschaften von Kaskadenklassifikatoren ausnutzt.

Der vierte Teil der Arbeit (Kapitel 5) beschreibt ein Verfahren zur probabilistischen Zustandsschätzung. Die Zustandsschätzung zur Realisierung einer dynamischen Suchstrategie wird durch den bekannten Condensation-Algorithmus [IB98a, BI98] umgesetzt. Das Kapitel schließt mit einem Lösungsansatz zur Problematik der probabilistischen Mehrobjektverfolgung mit Partikelfilter.

Die Umsetzung der probabilistisch motivierten Suchstrategie unter Verwendung des Partikelfilterverfahrens ist dann Gegenstand von Kapitel 6.

Im letzten Teil der Arbeit (Kapitel 7) erfolgt eine detaillierte Auswertung der verschiedenen Verfahren. Die Arbeit endet in Kapitel 8 mit einer Zusammenfassung und einem Resumé.

KAPITEL 2

Grundlagen

Zur robusten Detektion von Fußgängern bei Nacht wird hier ein Fusionsansatz auf Merkmalsbasis vorgestellt, der auf Basis einer Nah- und Ferninfrarotkamera eine Verbesserung der Detektionsleistung realisiert. Sowohl aus einem Bild der NIR-Kamera, als auch aus einem zeitlich zugeordneten Bild der FIR-Kamera werden Merkmale extrahiert und zur Fußgängererkennung verwendet. Um Merkmale einander zuordnen zu können, müssen Beziehungen zwischen den beiden Kamerabildern hergestellt werden: Welche Bildbereiche gehören zusammen? Wo findet sich ein Fußgänger aus dem einen Bild im anderen Bild wieder? Wie können ganz allgemein korrespondierende Bildpunkte und -bereiche in beiden Bildern identifiziert werden?

Darüber hinaus müssen Bildausschnitte auch zum bewegten Fahrzeug selbst zugeordnet werden können: In welcher Entfernung zum Fahrzeug befindet sich ein detektierter Fußgänger? Welche Ausschnitte im Bild sind für die gestellte Aufgabe der Warnung von Fußgängern im Straßenverkehr bei Nacht von Relevanz?

Für all diese Fragestellungen ist es notwendig die Lagebeziehungen der Kameras zueinander, sowie die Position und Ausrichtung der Kameras bezogen auf das Fahrzeug in geeigneten Koordinatensystemen zu definieren und zu bestimmen.

Im folgenden Abschnitt 2.1 werden zunächst die in dieser Arbeit verwendeten Kameras beschrieben. Abschnitt 2.2 geht dann im Detail auf die verwendeten Kamerasysteme ein. Abschnitt 2.3 skizziert die Problematik der mehrdeutigen Projektion und beschreibt kurz die Grundlagen der Epipolargeometrie. Das Kapitel schließt in Abschnitt 2.4 mit Begriffsdefinitionen zur Beschreibung von Fußgängerhypothesen im Bild.

	NIR-Kamera	FIR-Kamera
	CMOS-Kamera	Mikrobolometer
spektrale Empfindlichkeit	$380 - 1100\,\text{nm}$	$8 - 14\,\mu\text{m}$
Auflösung	$640\,\text{px} \times 480\,\text{px}$	$324\,\text{px} \times 256\,\text{px}$
Winkelauflösung	$26\,\text{px}/^\circ$	$9\,\text{px}/^\circ$
Quantisierungstiefe	12 bit	14 bit
Framerate	25 fps	30 fps

Tabelle 2.1.: Kenngrößen der verwendeten Kameras.

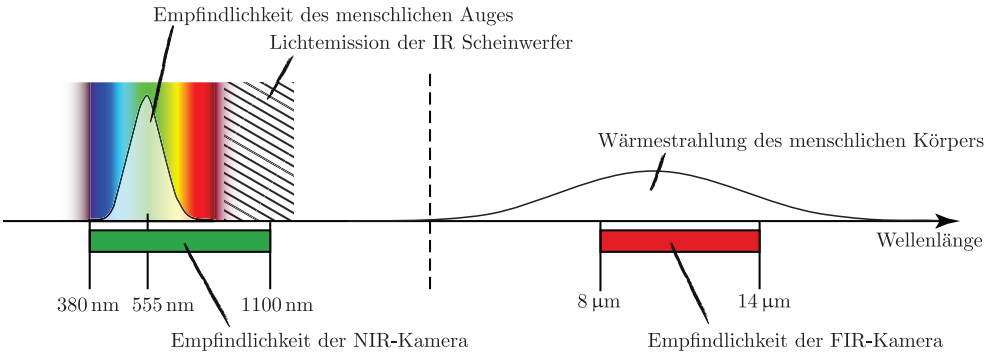


Abbildung 2.1.: Das elektromagnetische Spektrum. Illustrative Abbildung.

2.1. Sensoren zur Fußgängererkennung bei Nacht

Die für die vorliegende Arbeit verwendete Sensorik ist zum einen eine FIR-Kamera der Firma Autoliv, wie sie im rein darstellenden Nachtsichtsystem von BMW verwendet wird, zum anderen eine NIR-Kamera der Firma Bosch, wie sie im Nachtsichtassistenten von Daimler eingesetzt wird. Tabelle 2.1 gibt einen Überblick über die Kennzahlen beider Kameras, die zugehörigen spektralen Empfindlichkeiten sind in Abbildung 2.1 gegenübergestellt.

Die NIR-Kamera ist eine CMOS-Kamera mit einer spektralen Empfindlichkeit von $380 - 1100\,\text{nm}$. Sie zeichnet sich vor allem durch einen hohen Helligkeitsdynamikbereich [EK03] aus. Die Auflösung der Grauwertbilder beträgt $640\,\text{px} \times 480\,\text{px}$ mit $26\,\text{px}/^\circ$ Winkelauflösung und 12 bit Quantisierungstiefe. Die Framerate ist 25 fps. Bei Nacht werden zur Beleuchtung Infrarotscheinwerfer mit einer fernlichtähnlichen Charakteristik verwendet, die im Wellenlängenbereich von 800 nm bis 1100 nm arbeiten. Damit beträgt die direkte Sichtweite des Systems je nach Witterungsverhältnissen bis zu 120 m und mehr (vgl. Abbildung 2.2).

Bei der FIR-Kamera handelt es sich um einen ungekühlten Mikrobolometer mit $324\,\text{px} \times 256\,\text{px}$ Auflösung mit $9\,\text{px}/^\circ$ Winkelauflösung, 14 bit Quantisierungstiefe und einer Framerate von 30 fps. Der Spektralbereich ist $8 - 14\,\mu\text{m}$. Die thermische Empfindlichkeit beträgt laut Datenblatt 100 mK bei $+25\,^\circ\text{C}$. Die thermische Empfindlichkeit wird dabei definiert durch die Temperaturänderung eines schwarzen Strahlers, der den Sichtbereich der Kamera komplett ausfüllt und eine Änderung des Messsignals bewirkt, die dem

Abbildung 2.2.: Sichtbereich ohne und mit IR-Beleuchtung. Das Bild links wurde mit abgeschaltetem IR-Scheinwerfer bei normalem Abblendlicht aufgenommen, im Bild rechts sind die IR-Scheinwerfer aktiv. Die aufgestellten Puppen sind mit Stoff bezogen und stehen in Abständen von 50 m, 80 m, 100 m und 120 m vor dem Fahrzeug. Quelle: [Sch03b].

Effektivwert des elektrischen Rauschens des Sensors entspricht (NEDT, engl. „Noise Equivalent Temperature Difference“, [RAY]).

Beide Kameras, sowie die IR-Beleuchtung sind in einem Versuchsfahrzeug integriert. Die FIR-Kamera ist dabei im Kühlergrill und die NIR-Kamera hinter der Frontscheibe im Dachraum montiert. Die genauen intrinsischen und extrinsischen Kameraparameter, sowie die nötigen Nomenklaturen sind Gegenstand des folgenden Abschnitts.

2.2. Kamerakoordinatensysteme und Kalibrierung

Die Koordinatensysteme in dieser Arbeit folgen denen in DIN 70000 [DIN94], die dort in erster Linie für fahrdynamische Zwecke definiert wurden. Die Wahl des Kamerakoordinatensystems stellt dabei eine logische Erweiterung dar.

Koordinatensysteme

Das **Fahrzeugkoordinatensystem** bezeichnet das fahrzeugfeste Koordinatensystem (engl. vehicle axis system). Die ${}^{\mathrm{v}}X$-Achse zeigt in Fahrtrichtung entlang der Fahrzeuglängsmittelebene nach vorne (Distanz), die ${}^{\mathrm{v}}Y$-Achse steht senkrecht zur Fahrzeuglängsmittelebene und zeigt in Fahrtrichtung nach links (laterale Ablage), die ${}^{\mathrm{v}}Z$-Achse zeigt nach oben (Höhe). Der Koordinatenursprung ist in DIN 70000 nicht fest spezifiziert und ist in dieser Arbeit durch den Mittelpunkt der Hinterachse festgelegt (vgl. Abbildung 2.3). Alle Größen werden in [m] angegeben. Für Punkte und Koordinaten werden Großbuchstaben verwendet.

Die Reihenfolge der **Drehungen** in allen Koordinatensystemen erfolgt analog zu DIN 70000: Zuerst Gierwinkel ψ (engl. yaw angle), dann Nickwinkel ϑ (engl. pitch angle), dann Wankwinkel ϕ (engl. roll angle). Die Rotationsmatrix ist dann $\boldsymbol{R} = \boldsymbol{R}_{\phi} \cdot \boldsymbol{R}_{\vartheta} \cdot \boldsymbol{R}_{\psi}$,

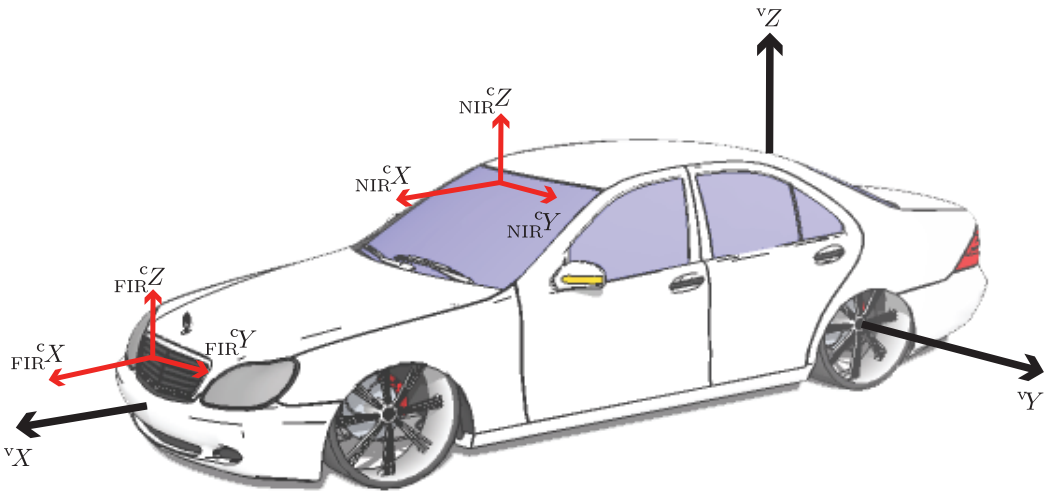


Abbildung 2.3.: Koordinatensysteme. Das Fahrzeugkoordinatensystem $({}^{\mathrm{v}}X, {}^{\mathrm{v}}Y, {}^{\mathrm{v}}Z)^{\mathrm{T}}$ hat seinen Koordinatenursprung im Mittelpunkt der Hinterachse. Die ${}^{\mathrm{c}}X$-Achsen der beiden Kamerakoordinatensysteme $({}_{\mathrm{FIR}}{}^{\mathrm{c}}X, {}_{\mathrm{FIR}}{}^{\mathrm{c}}Y, {}_{\mathrm{FIR}}{}^{\mathrm{c}}Z)^{\mathrm{T}}$ und $({}_{\mathrm{NIR}}{}^{\mathrm{c}}X, {}_{\mathrm{NIR}}{}^{\mathrm{c}}Y, {}_{\mathrm{NIR}}{}^{\mathrm{c}}Z)^{\mathrm{T}}$ zeigen wie die ${}^{\mathrm{v}}X$-Achse des Fahrzeugkoordinatensystems in Fahrtrichtung nach vorne. Die Bildebenen liegen dann jeweils in der ${}^{\mathrm{c}}Y$-${}^{\mathrm{c}}Z$-Ebene.

mit

$$\boldsymbol{R}_{\phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix},$$

$$\boldsymbol{R}_{\vartheta} = \begin{pmatrix} \cos\vartheta & 0 & \sin\vartheta \\ 0 & 1 & 0 \\ -\sin\vartheta & 0 & \cos\vartheta \end{pmatrix},$$

$$\boldsymbol{R}_{\psi} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Die angegebenen Drehmatrizen definieren hier aktive Drehungen, d.h. sie bestimmen die Drehung eines Punktes $\boldsymbol{X}$ durch $\boldsymbol{X'} = \boldsymbol{R}\boldsymbol{X}$.

Das **Kamerakoordinatensystem** (oder **Sensorkoordinatensystem**, engl. camera axis system) ist rechtshändig, die ${}^{\mathrm{c}}X$-Achse zeigt in dieselbe Richtung, wie das Fahrzeugkoordinatensystem, in dem der Sensor verbaut ist - im vorliegenden Fall also in Richtung der optischen Achse. Die ${}^{\mathrm{c}}Y$-Achse zeigt in Blickrichtung nach links, die ${}^{\mathrm{c}}Z$-Achse nach oben. Die Bildebene liegt damit also in der ${}^{\mathrm{c}}Y$-${}^{\mathrm{c}}Z$-Ebene. Dies erscheint auf den ersten Blick unnatürlich, doch mit einer Orientierung des Sensorsystems in der selben Weise wie das Trägersystem sind die Rotationswinkel, die die Lage des Sensors beschreiben leichter interpretierbar.

Auch im Kamerakoordinatensystem werden alle Größen in [m] angegeben. Für Punkte und Koordinaten werden Großbuchstaben verwendet.

Die Transformation einer Koordinate im Fahrzeugkoordinatensystem ins jeweilige Kamerakoordinatensystem erfolgt dann mittels

$$^{\mathrm{c}}\boldsymbol{P} = \boldsymbol{R}^{-1} \cdot (^{\mathrm{v}}\boldsymbol{P} - {}^{\mathrm{v}}\boldsymbol{C}),$$

mit ${}^{\mathrm{v}}\boldsymbol{C}$ die Koordinaten des Kamerazentrums. Die Umkehrung ist mit

$$^{\mathrm{v}}\boldsymbol{P} = \boldsymbol{R} \cdot {}^{\mathrm{c}}\boldsymbol{P} + {}^{\mathrm{v}}\boldsymbol{C} \tag{2.1}$$

gegeben.

Das **Kamerachip-Koordinatensystem** (oder **Pixelkoordinatensystem**) ist ein diskretes 2-D Koordinatensystem mit dem Ursprung links oben in der Ecke des Bildes. Die Koordinaten werden diskret durch row und col in [px] (Pixel) angegeben. Für Punkte werden Kleinbuchstaben verwendet.

Ein Punkt $\boldsymbol{p} = (\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ in Pixelkoordinaten ist im Kamerakoordinatensystem

$$\begin{pmatrix} {}^{\mathrm{c}}X' \\ {}^{\mathrm{c}}Y' \\ {}^{\mathrm{c}}Z' \end{pmatrix} = \begin{pmatrix} f \\ -(\mathrm{col} - \mathrm{col}_0)\, s_{\mathrm{col}} \\ -(\mathrm{row} - \mathrm{row}_0)\, s_{\mathrm{row}} \end{pmatrix}, \tag{2.2}$$

mit der Brennweite f in [m], Pixelgrößen s_{col}, s_{row} in $[\frac{\mathrm{m}}{\mathrm{px}}]$ und dem Hauptpunkt $(\mathrm{col}_0, \mathrm{row}_0)^{\mathrm{T}}$. Mit diesen intrinsischen Parametern ist dann die projektive Abbildung eines Punktes ${}^{\mathrm{c}}\boldsymbol{P} = ({}^{\mathrm{c}}X, {}^{\mathrm{c}}Y, {}^{\mathrm{c}}Z)^{\mathrm{T}}$ in Kamerakoordinaten auf den Punkt $\boldsymbol{p} = (\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ im Bild bestimmt durch

$$\begin{pmatrix} \mathrm{col} \\ \mathrm{row} \end{pmatrix} = \begin{pmatrix} -\frac{{}^{\mathrm{c}}Y}{{}^{\mathrm{c}}X} \cdot \frac{f}{s_{\mathrm{col}}} + \mathrm{col}_0 \\ -\frac{{}^{\mathrm{c}}Z}{{}^{\mathrm{c}}X} \cdot \frac{f}{s_{\mathrm{row}}} + \mathrm{row}_0 \end{pmatrix}. \tag{2.3}$$

Die Wahl der Schreibweise $(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ für eine Pixelkoordinate ist sicherlich ungewöhnlich, doch erleichtert diese Schreibweise die Lesbarkeit und das Verständnis vieler der Koordinatentransformationen im Rahmen dieser Arbeit.

Zusammenfassend lässt sich die gesamte Abbildung eines Punktes ${}^{\mathrm{v}}\boldsymbol{P} = ({}^{\mathrm{v}}X, {}^{\mathrm{v}}Y, {}^{\mathrm{v}}Z)^{\mathrm{T}}$ im Fahrzeugkoordinatensystem auf den Bildpunkt $\boldsymbol{p} = (\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ in homogenen Koordinaten darstellen mittels

$$\begin{pmatrix} \alpha \cdot \mathrm{col} \\ \alpha \cdot \mathrm{row} \\ \alpha \end{pmatrix} = \boldsymbol{\mathcal{P}} \cdot \begin{pmatrix} {}^{\mathrm{v}}\boldsymbol{P} \\ 1 \end{pmatrix}, \tag{2.4}$$

mit der Projektionsmatrix

$$\boldsymbol{\mathcal{P}} = \boldsymbol{K} \cdot \left[\boldsymbol{R}^{-1} \mid -\boldsymbol{R}^{-1} \cdot {}^{\mathrm{v}}\boldsymbol{C} \right],$$

den extrinsischen Parametern in $\boldsymbol{R}$, ${}^{\mathrm{v}}\boldsymbol{C}$, sowie der intrinsischen Kameramatrix

$$\boldsymbol{K} = \begin{pmatrix} \mathrm{col}_0 & -\frac{f}{s_{\mathrm{col}}} & 0 \\ \mathrm{row}_0 & 0 & -\frac{f}{s_{\mathrm{row}}} \\ 1 & 0 & 0 \end{pmatrix}.$$

Die Umkehrung der Abbildung in Gleichung (2.4) ist nicht eindeutig: mit (2.1) und der inversen Kameramatrix

$$\boldsymbol{K}^{-1} = \begin{pmatrix} 0 & 0 & 1 \\ -\frac{s_{\mathrm{col}}}{f} & 0 & \frac{\mathrm{col}_0 s_{\mathrm{col}}}{f} \\ 0 & -\frac{s_{\mathrm{row}}}{f} & \frac{\mathrm{row}_0 s_{\mathrm{row}}}{f} \end{pmatrix}$$

gilt für jedes $s \in \mathbb{R}$ mit $\boldsymbol{\mathcal{P}}^{-1} := \left[\boldsymbol{R}\boldsymbol{K}^{-1} \mid {}^{\mathrm{v}}\boldsymbol{C}\right]$:

$${}^{\mathrm{v}}\boldsymbol{P} = \boldsymbol{\mathcal{P}}^{-1} \begin{pmatrix} \mathrm{col} \cdot s \\ \mathrm{row} \cdot s \\ s \\ 1 \end{pmatrix}.$$

Die Umkehrung der Projektion eines Fußgängers ins Bild kann unter der Annahme einer ebenen Welt und bei bekannter Fußgängergröße H auch geometrisch hergeleitet werden (Abbildung 2.4). Ein Fußgänger im Bild ist über den Scheitelpunkt $(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ und der Größe h definiert. Mit den Definitionen $\Delta^{\mathrm{v}}\boldsymbol{P} := (0, 0, H)^{\mathrm{T}}$ und $\Delta^{\mathrm{c}}\boldsymbol{P} := (\Delta^{\mathrm{c}}X, \Delta^{\mathrm{c}}Y, \Delta^{\mathrm{c}}Z)^{\mathrm{T}} = \boldsymbol{R}^{-1}\Delta^{\mathrm{v}}\boldsymbol{P}$ gilt mit ${}^{\mathrm{c}}Z' \overset{(2.2)}{=} -\left((\mathrm{row} + h) - \mathrm{row}_0\right) s_{\mathrm{row}}$ und den Bezeichnungen aus Abbildung 2.4:

$$\frac{h s_{\mathrm{row}}}{f} = \frac{\Delta^{\mathrm{c}}Z - a}{{}^{\mathrm{c}}X} \quad \text{und} \quad \frac{{}^{\mathrm{c}}Z'}{f} = \frac{a}{\Delta^{\mathrm{c}}X}.$$

Daraus folgt:

$$\begin{aligned} {}^{\mathrm{c}}X &= \frac{f}{h s_{\mathrm{row}}} \left(\Delta^{\mathrm{c}}Z - \Delta^{\mathrm{c}}X \frac{{}^{\mathrm{c}}Z'}{f}\right), \\ {}^{\mathrm{c}}Y &\overset{(2.3)}{=} \left(\mathrm{col}_0 - \mathrm{col}\right) {}^{\mathrm{c}}X \cdot \frac{s_{\mathrm{col}}}{f}, \\ {}^{\mathrm{c}}Z &\overset{(2.3)}{=} \left(\mathrm{row}_0 - \mathrm{row}\right) {}^{\mathrm{c}}X \cdot \frac{s_{\mathrm{row}}}{f}, \end{aligned}$$

und schließlich

$${}^{\mathrm{v}}\boldsymbol{P} = \boldsymbol{R} \begin{pmatrix} {}^{\mathrm{c}}X \\ {}^{\mathrm{c}}Y \\ {}^{\mathrm{c}}Z \end{pmatrix} + {}^{\mathrm{v}}\boldsymbol{C}.$$

In dieser Arbeit wird die Projektion eines Fußgängers der Größe H und Scheitelpunkt ${}^{\mathrm{v}}\boldsymbol{P}$ ins Bild mit

$$(\mathrm{col}, \mathrm{row}, h) = \mathrm{proj}_H({}^{\mathrm{v}}\boldsymbol{P}; \boldsymbol{\mathcal{P}}) \tag{2.5}$$

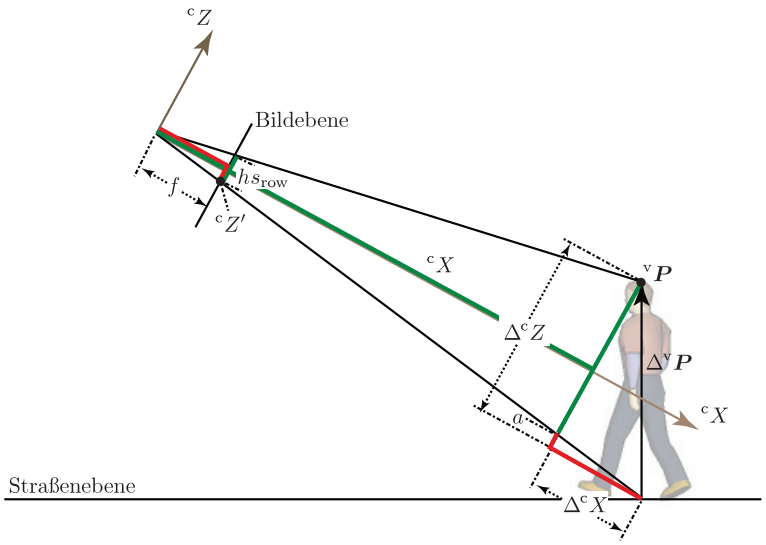


Abbildung 2.4.: Geometrische Skizze zur Umkehrung der Projektion eines Fußgängers ins Bild. Für den Spezialfall einer bekannten Fußgängerhöhe in der Welt, kann unter der Annahme einer ebenen Welt zu einem Abbild eines Fußgängers im Bild eindeutig die Entfernung ${}^{\mathrm{c}}X$ bestimmt werden.

bezeichnet. Der Zusatz $\boldsymbol{\mathcal{P}}$ wird weggelassen, wenn dies aus dem Kontext ersichtlich ist. Die geometrisch hergeleitete Umkehrung ist dann entsprechend

$$ {}^{\mathrm{v}}\boldsymbol{P} = \mathrm{proj}_H^{-1}\left(\mathrm{col}, \mathrm{row}, h\,;\, \boldsymbol{\mathcal{P}}\right). $$

Zusätzlich dazu wird in Kapitel 4 die Hilfsfunktion

$$ \mathrm{row} = \mathrm{reproj}_H\left(\mathrm{col}, h\,;\, \boldsymbol{\mathcal{P}}\right) $$

eingeführt, die die Zeile row des Fußgängerscheitelpunktes im Bild bestimmt, wenn dessen Spalte col, die Höhe h des Fußgängers im Bild und die Größe H des Fußgängers in der realen Welt bekannt sind (genaue Definition in (4.3), Seite 85, Kapitel 4.1).

Kamerakalibrierung

Zur Bestimmung der intrinsischen Kameraparameter sowie der Lagebeziehungen der beiden Kameras zueinander wurde auf die Arbeiten von [Kru07] zurückgegriffen. Dabei handelt es sich um ein Framework zur automatischen Kalibration von Kamerasystemen mit beliebiger Anzahl von Sensoren. Es bestimmt sowohl die intrinsischen, als auch die extrinsischen Parameter des Kamerasystems auf Basis von Bildaufnahmen eines Schachbrettmusters. Dazu wird in jedem Bild das Kalibriergitter (bestehend aus den Ecken von Schachbrettfeldern) durch Templatematching, Kreuzkorrelation und einer anschließenden topologischen Analyse der Menge der Eckknoten subpixelgenau bestimmt. Die intrinsische Kalibrierung erfolgt dann auf Basis von Bouguet [Bou99]. Anschließend erfolgt die Bestimmung der Lagebeziehungen aller Kameras in einem

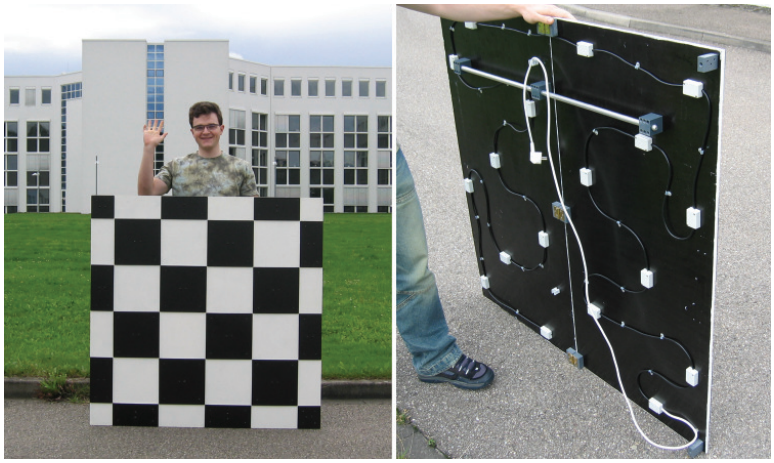


Abbildung 2.5.: Kalibrierkörper mit beheizten Kacheln. Die schwarzen Kacheln haben eine Kantenlänge von 25 cm und werden durch von hinten angebrachte PTC-Heizelemente auf ca. 50°C erhitzt. Quelle: [Rot06].

gemeinsamen System durch einen Optimierungsprozess (modifiziertes Newtonverfahren), das die räumliche Geometrie des Kalibrierkörpers (d.h. Größe und Anzahl der Schachbrettkacheln) mit berücksichtigt.

Voraussetzung für die Anwendung dieses Verfahrens ist ein Kalibrierkörper mit Schachbrettmuster, das in beiden Kamerabildern sichtbar ist. Da die Empfindlichkeitsbereiche der in dieser Arbeit verwendeten Sensoren sich nicht überlappen, wurde im Rahmen der Arbeit von [Rot06] ein Kalibrierkörper entwickelt, der sowohl im Nahinfrarotbereich, als auch im Wärmespektrum sichtbar ist (Abbildung 2.5). Das Schachbrett besteht aus schwarzen quadratischen Kacheln mit einer Kantenlänge von 25 cm, die auf einer weiß lackierten Tafel montiert sind. Im Nahinfrarotbereich ist das Muster als schwarz/weißes Schachbrett sehr gut zu erkennen. Die schwarzen Kacheln werden zusätzlich durch von hinten angebrachte PTC-Heizelemente[1] auf ca. 50°C erhitzt. Dadurch ist auch im Bild der FIR-Kamera ein kontrastreiches Schachbrettmuster abgebildet.

Das Framework von [Kru07] bestimmt neben den intrinsischen Parametern zwar die Lagebeziehung der Kameras zueinander, allerdings lediglich im Kamerakoordinatensystem einer der beiden Kameras. Um zusätzlich den Bezug zum Fahrzeugkoordinatensystem herzustellen, wurden in einer weiteren Kalibrierung Punkte im Fahrzeugkoordinatensystem vermessen und so die externe Kalibrierung der NIR-Kamera vorgenommen. Die externe Kalibrierung der FIR-Kamera ergibt sich dann aus dem ersten Kalibrationsverfahren.

Dazu wurde in [Hal07] der Boden vor dem Fahrzeug mit einem Schachbrettmuster ausgelegt und achsenparallel sowohl zur ${}^{v}Y$-Achse, als auch zur ${}^{v}X$-Achse des Fahrzeugkoordinatensystems ausgerichtet (Abbildung 2.6). Zusätzlich wurde das Kalibrierbrett mit den Flächen im 45°-Winkel zueinander vermessen.

[1] PTC-Heizelemente sind selbstregelnde Kaltleiter-Heizelemente auf Keramikbasis mit positivem Temperaturkoeffizienten (PTC, engl. „Positive Temperature Coefficient“, [PTC]).

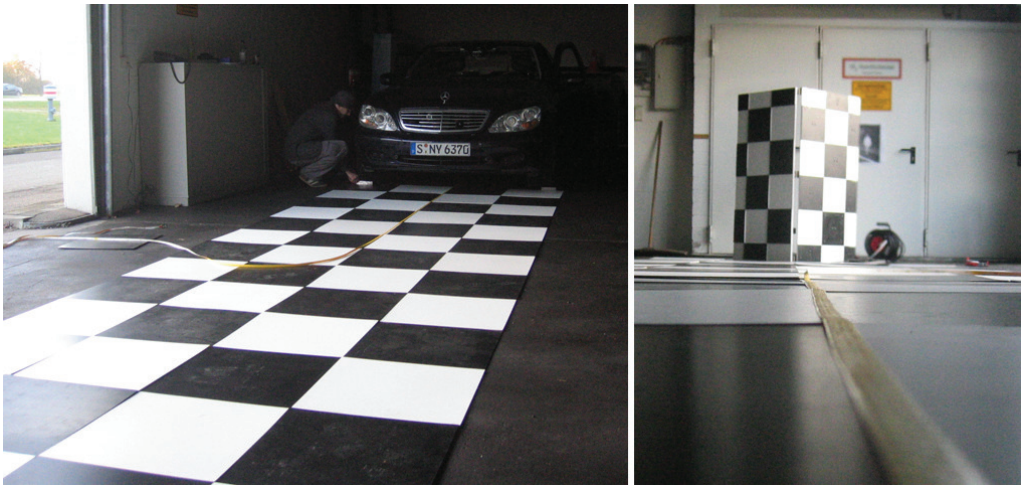


Abbildung 2.6.: Vermessung des Kalibrierkörpers im Fahrzeugkoordinatensystem. Um die Lage der Kameras im Fahrzeugkoordinatensystem zu bestimmen, wurde ein Kalibrierkörper im Fahrzeugkoordinatensystem vermessen. Quelle: [Hal07].

	NIR-Kamera	**FIR-Kamera**
$f[\mathrm{m}]$	0.012	0.01
$s_{\mathrm{col}}[\frac{\mathrm{m}}{\mathrm{px}}]$	$8.092 \cdot 10^{-6}$	$2.024 \cdot 10^{-5}$
$s_{\mathrm{row}}[\frac{\mathrm{m}}{\mathrm{px}}]$	$8.098 \cdot 10^{-6}$	$2.032 \cdot 10^{-5}$
$(\mathrm{row}_0, \mathrm{col}_0)^{\mathrm{T}}[(\mathrm{px}, \mathrm{px})^{\mathrm{T}}]$	$(212, 348)^{\mathrm{T}}$	$(131, 164)^{\mathrm{T}}$
${}^{\mathrm{v}}C^{\mathrm{T}}[(\mathrm{m}, \mathrm{m}, \mathrm{m})^{\mathrm{T}}]$	$(1.73, -0.26, 1.28)^{\mathrm{T}}$	$(3.75, -0.18, 0.58)^{\mathrm{T}}$
$\phi[^\circ]$ (Wankwinkel)	1.41°	0.58°
$\vartheta[^\circ]$ (Nickwinkel)	-0.48°	0.97°
$\psi[^\circ]$ (Gierwinkel)	-2.05°	-0.47°

Tabelle 2.2.: Kalibrierdaten des Kamerasystems. Die Verzeichnung beider Kameras ist vernachlässigbar. Deshalb wird hier von einem Lochkameramodell ausgegangen.

Die kalibrierten Kameraparameter sind in Tabelle 2.2 zusammengefasst. Die Verzeichnungen beider Kameras waren dabei vernachlässigbar, so dass von einem Lochkameramodell ausgegangen werden kann.

Eine genaue Kalibrierung der Kameras - auch der extrinsischen Parameter - ist wichtig, da nur so Objektmodelle im Fahrzeugkoordinatensystem in Bezug zu Modellen im Bildkoordinatensystem gesetzt werden können. Der Aufwand der Kalibrierung ist dabei in einem serienmäßigen Fertigungsprozess deutlich geringer als hier, da an einem Fertigungsband die Lagebeziehungen des Kalibrierkörpers zum fest eingespannten Fahrzeug in ausreichender Genauigkeit bekannt sind.

Dennoch muss man sich bewusst machen, dass eine Kalibrierung auch immer mit Fehlern behaftet bzw. nicht dauerhaft sein kann: Neben Dynamikeinflüssen im normalen Fahrbetrieb (z.B. Nickbewegungen des Fahrzeugs) können jederzeit Materialverschleiß

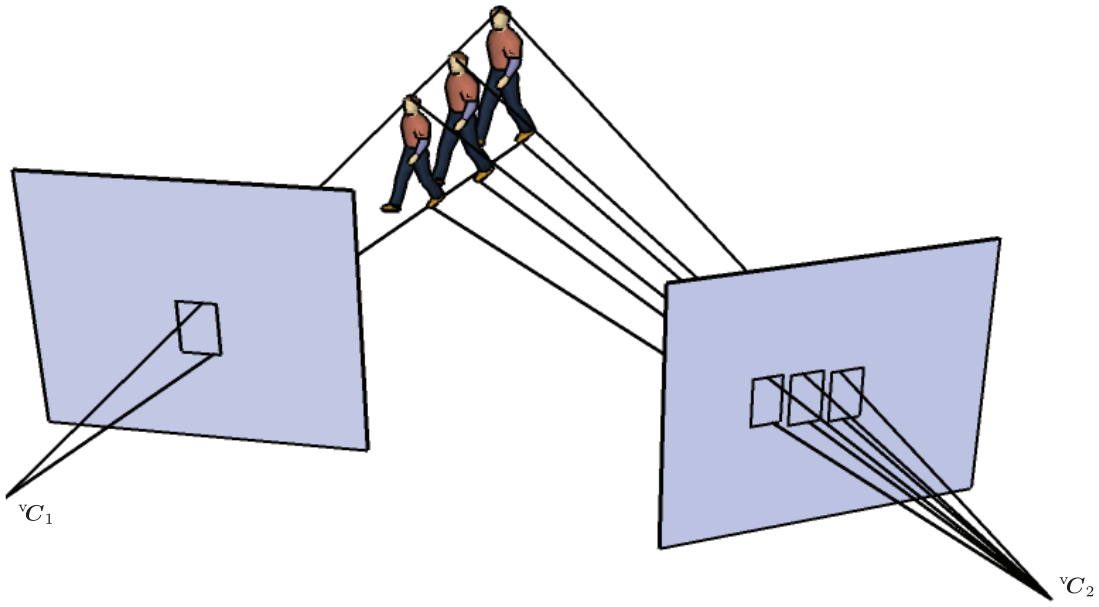


Abbildung 2.7.: Mehrdeutige Projektionen. Ausgehend vom Abbild eines Fußgängers im linken Bild gibt es mehrere mögliche Abbilder im rechten Bild, die potentiell denselben Fußgänger beschreiben. Da jedoch die reale Größe und damit genaue Position in Fahrzeugkoordinaten des Fußgängers nicht bekannt ist, ist die Abbildung mehrdeutig.

etc. Einfluss auf die Kalibrierdaten haben. Gute Fusionssysteme zeichnen sich deshalb durch Robustheit gegenüber fehlerhaften Kalibrierungen aus.

2.3. Mehrdeutige Projektionen

Die Detektionsaufgabe der Fußgängererkennung wird sich in dieser Arbeit auf die Informationen beider Kameras stützen. Zur Detektion im Bild wird dazu ein Suchfenster in Größe und Position variiert und über das Bild geführt. Die Klassifikation als „Fußgänger“ oder „Hintergrund“ erfolgt dabei auf Basis von Merkmalen, die innerhalb des Suchfensters ausgewertet werden. Sollen Merkmale mehrerer Kameras mit unterschiedlichen Blickwinkeln kombiniert werden, müssen die einzelnen Merkmale einander zugeordnet werden können, d.h. ausgehend vom Suchfenster aus einem Bild müssen die Suchfenster in den Bildern aller anderen Sensoren dasselbe Objekt repräsentieren. Ist die reale Größe eines Fußgängers bekannt, so ist die projektive Abbildung in Gleichung (2.4) eindeutig umkehrbar. Damit kann einem Suchfenster des einen Sensorstroms genau ein Suchfenster des anderen Sensorstroms zugeordnet werden. Leider trifft diese Annahme bei Fußgängern in der Regel nicht zu. Die Abbildung ist mehrdeutig, d.h. für ein Suchfenster im einen Bild sind mehrere Suchfenster im anderen Bild möglich (Abbildung 2.7).

Mathematisch - bezogen z.B. auf die Mittelpunkte der Oberkanten der jeweiligen Suchfenster - liegen alle möglichen zugeordneten Suchfenster auf derselben Epipolarlinie.

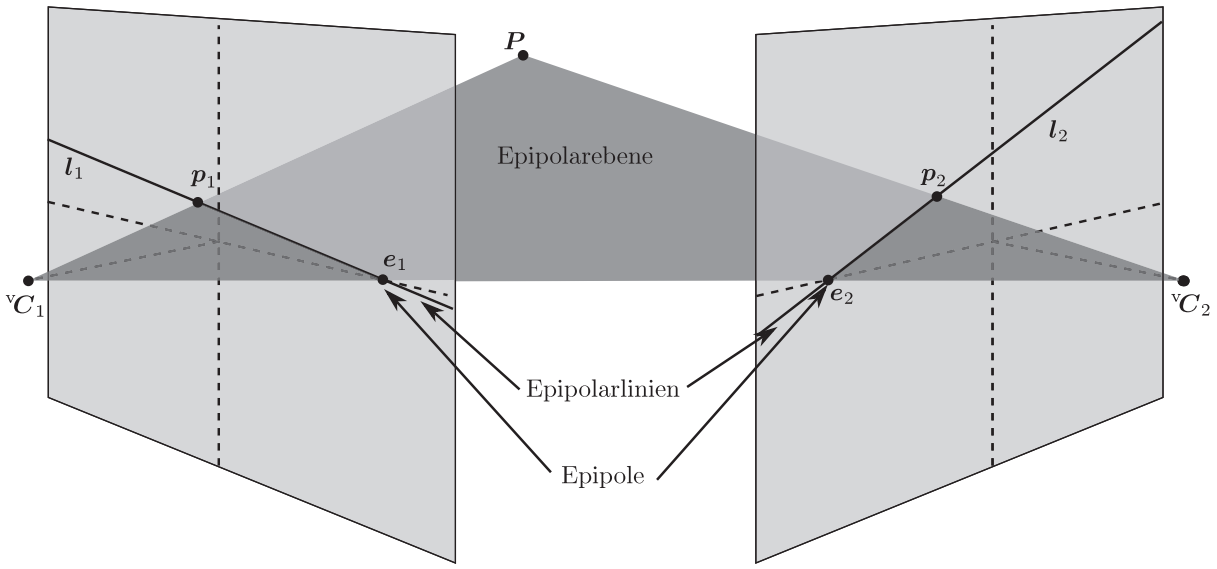


Abbildung 2.8.: Epipolargeometrie. Die Kamerazentren ${}^{v}\boldsymbol{C}_1$, ${}^{v}\boldsymbol{C}_2$ und der Punkt $\boldsymbol{P}$ spannen die Epipolarebene auf, die die Bildebenen in den Epipolarlinien schneidet. Die Epipolarlinie $\boldsymbol{l}_2$ enthält damit alle möglichen Korrespondenzpunkte im zweiten Kamerabild, die zu einem festen Punkt $\boldsymbol{p}_1$ im ersten Kamerabild gehören. Die Epipole sind jeweils die Bilder der Kamerazentren im jeweils anderen Bild.

Die Epipolarlinie $\boldsymbol{l}_2$ gibt dabei ganz allgemein alle möglichen Korrespondenzpunkte $\boldsymbol{p}_2$ im zweiten Kamerabild an, die zu einem festen Punkt $\boldsymbol{p}_1$ im ersten Kamerabild gehören (d.h. Punkte, die jeweils Abbild desselben Punktes im Kamerakoordinatensystem sind, siehe Abbildung 2.8). Die Überführung von einem Kamerabild ins andere findet (im Folgenden immer in homogenen Koordinaten) mit Hilfe der Fundamentalmatrix $\boldsymbol{F}$ statt:

$$\boldsymbol{l}_2 = \boldsymbol{F}\boldsymbol{p}_1.$$

Die Fundamentalmatrix lässt sich leicht algebraisch durch die beiden Projektionsmatrizen $\boldsymbol{\mathcal{P}}_1$ und $\boldsymbol{\mathcal{P}}_2$ aus (2.4) ausdrücken (vgl. [HZ04]): Alle Urbilder von $\boldsymbol{p}_1$ liegen nämlich auf einer Geraden

- durch das Kamerazentrum ${}^{v}\boldsymbol{C}_1$ und
- durch den Punkt ${}^{v}\boldsymbol{P}_1 = \boldsymbol{\mathcal{P}}_1^{+}\boldsymbol{p}_1$.

$\boldsymbol{\mathcal{P}}_1^{+}$ ist die Pseudo-Inverse von $\boldsymbol{\mathcal{P}}_1$ zur Least-Square-Lösung von $\boldsymbol{\mathcal{P}}_1\,{}^{v}\boldsymbol{P} = \boldsymbol{p}_1$, d.h. $\boldsymbol{\mathcal{P}}_1^{+}\boldsymbol{\mathcal{P}}_1 = \mathrm{diag}\,(1,1,1)$.

Die Bilder von ${}^{v}\boldsymbol{C}_1$ und ${}^{v}\boldsymbol{P}_1$ im Bild der zweiten Kamera sind

- der Epipol $\boldsymbol{e}_2 = \boldsymbol{\mathcal{P}}_2\,{}^{v}\boldsymbol{C}_1$ und
- der Punkt $\boldsymbol{\mathcal{P}}_2\,{}^{v}\boldsymbol{P}_1 = \boldsymbol{\mathcal{P}}_2\boldsymbol{\mathcal{P}}_1^{+}\boldsymbol{p}_1$.

Beide liegen auf der Epipolarlinie $\boldsymbol{l}_2$, und damit ist $\boldsymbol{l}_2 = \boldsymbol{e}_2 \times \boldsymbol{\mathcal{P}}_2\boldsymbol{\mathcal{P}}_1^{+}\boldsymbol{p}_1$, also

$$\boldsymbol{F} = \boldsymbol{\mathcal{P}}_2\,{}^{v}\boldsymbol{C}_1 \times \left(\boldsymbol{\mathcal{P}}_2\boldsymbol{\mathcal{P}}_1^{+}\right).$$

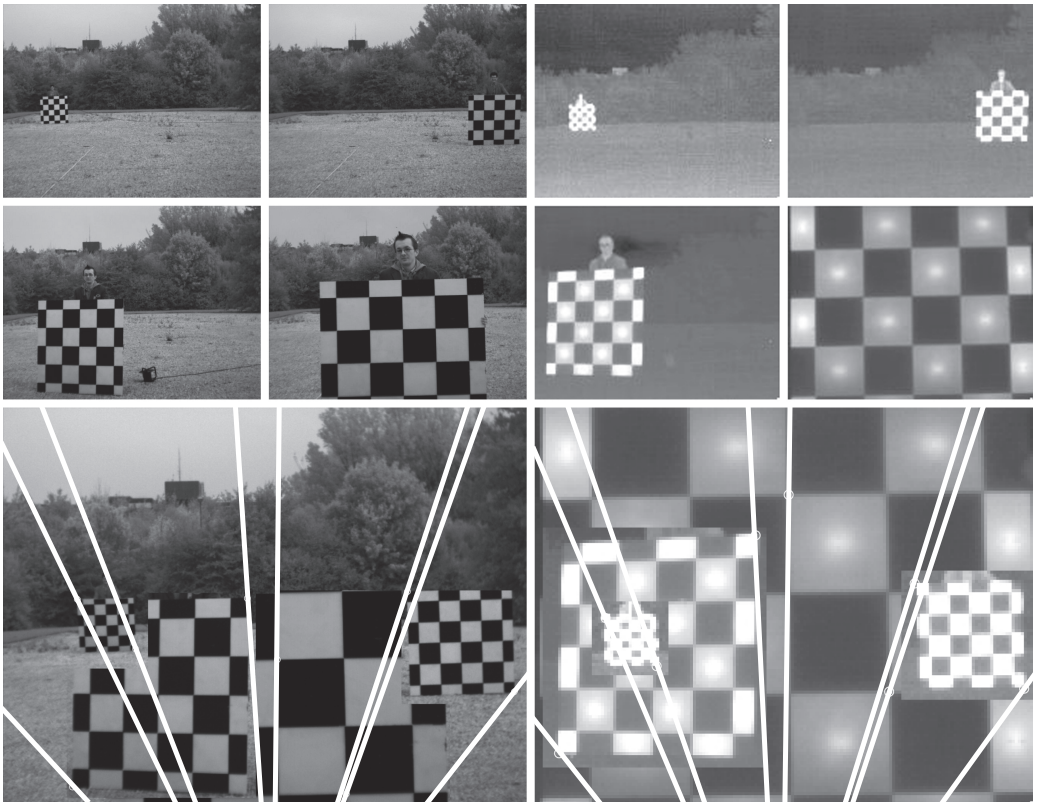


Abbildung 2.9.: Epipolarlinien. Die Bilder in der letzten Zeile stellen jeweils eine Montage der vier Einzelaufnahmen darüber dar, jeweils mit freigestellten Kalibrierkörpern. Links sind Bilder der NIR-Kamera dargestellt, rechts entsprechend die zugehörigen FIR-Bilder. Zu einigen ausgewählten Eckpunkten innerhalb der Kalibrierwand sind unterschiedliche Epipolarlinien dargestellt. Deutlich ist die Lage der Epipole unterhalb des dargestellten Bereichs zu erkennen. Quelle: [Hal07].

Bei dem in dieser Arbeit verwendeten Kameraaufbau liegen die Epipole beider Kameras jeweils unterhalb der Bildebenen: der Epipol der NIR-Kamera ist im Punkt $(243.4, 741.9)^{\mathrm{T}}$, der der FIR-Kamera im Punkt $(140.8, 292.4)^{\mathrm{T}}$. Abbildung 2.9 zeigt einige ausgewählte Epipolarlinien in einer Fotomontage unterschiedlicher Aufnahmen des Kalibrierkörpers.

Zur Vollständigkeit sei an dieser Stelle noch auf den in Kapitel 4 definierten Spezialfall hingewiesen, der mit

$$(\mathrm{col}_2, \mathrm{row}_2, h_2) = \mathrm{proj\_stream2stream}_H\,(\mathrm{col}_1, \mathrm{row}_1, h_1\,;\,\boldsymbol{\mathcal{P}}_1, \boldsymbol{\mathcal{P}}_2)$$

abgekürzt wird. Es handelt sich dabei um das Abbild eines Fußgängers (angegeben durch den Scheitelpunkt $(\mathrm{col}_1, \mathrm{row}_1)^{\mathrm{T}}$ und dessen Höhe h_1) aus dem Bild des 1. Sensors ins Bild des 2. Sensors, bei bekannter realer Fußgängergröße H. In diesem Fall ist die Projektion in Abbildung 2.7 eindeutig (genaue Definition in (4.4), Seite 91, Kapitel 4.2).

2.4. Begriffsdefinitionen

Die Detektion von Fußgängern vor dem Fahrzeug erfolgt in dieser Arbeit auf Basis von Bildern einer oder mehrerer Kameras. Dazu werden mit einem trainierten Klassifikator einzelne Hypothesen $x \in \mathcal{X}$ evaluiert. $\mathcal{X}$ bezeichnet dabei die Gesamtmenge aller möglichen Hypothesen für ein Bild (oder Bildtupel im Falle mehrerer Sensoren). Bei nur einem Sensor kann eine Hypothese in Form eines einzelnen Suchfensters spezifiziert werden, das Position und Größe eines rechteckigen Bereichs im Bild beschreibt. Das Ergebnis des Klassifikators ist dann jeweils eine Aussage, ob ein Fußgänger an genau dieser Position und in der jeweiligen Größe detektiert wurde. In dieser Arbeit weisen alle Suchfenster ein festes Seitenverhältnis r auf. Damit ist ein Suchfenster s über die Bildkoordinate der mittleren oberen Ecke $(\mathrm{col}', \mathrm{row}')^{\mathrm{T}}$ und dessen Suchfensterhöhe h' beschrieben:

$$s = (\mathrm{col}', \mathrm{row}', h') \,.$$

Die Breite des Suchfensters ist dann $w' = r \cdot h'$. Kommt mehr als nur eine Kamera zum Einsatz ist eine Hypothese nicht mehr nur durch ein einzelnes Suchfenster spezifiziert, sondern durch je ein Suchfenster pro Kamerabild. Bei einer Gesamtzahl von N_{C} Kameras ist

$$x = (s_1, \ldots, s_{N_{\mathrm{C}}}) = \left((\mathrm{col}'_1, \mathrm{row}'_1, h'_1), \ldots, \left(\mathrm{col}'_{N_{\mathrm{C}}}, \mathrm{row}'_{N_{\mathrm{C}}}, h'_{N_{\mathrm{C}}}\right)\right) ,$$

im Falle der Kombination von FIR- und NIR-Sensor also

$$x = (s_{\mathrm{FIR}}, s_{\mathrm{NIR}}) = ((\mathrm{col}'_{\mathrm{FIR}}, \mathrm{row}'_{\mathrm{FIR}}, h'_{\mathrm{FIR}}), (\mathrm{col}'_{\mathrm{NIR}}, \mathrm{row}'_{\mathrm{NIR}}, h'_{\mathrm{NIR}})) \,.$$

Sinnvollerweise stehen die einzelnen Suchfenster natürlich über die Kamerageometrie in Beziehung zueinander. Der Klassifikator wird dann auf beide Suchfenster gleichzeitig angewandt.

In jedem Bild können Fußgänger an vielen Positionen mit verschiedenen Skalierungen erscheinen. Deshalb muss bei der Verwendung eines Klassifikators als Detektor in jedem Bild (oder Bildtupel im Falle mehrerer Sensoren) eine große Menge an Hypothesen geprüft werden. Die Systemkomponente zur Erzeugung der Hypothesen wird Hypothesengenerator genannt und nimmt neben der Klassifikationskomponente (siehe Kapitel 3) einen zentralen Stellenwert im Gesamtsystem ein (siehe Kapitel 4 und Kapitel 6).

Die Klassifikation der Hypothese erfolgt in dieser Arbeit auf Basis von Merkmalen, die das Ergebnis einfacher Filteroperationen darstellen. Die Filter sind dabei immer im gleichen Raster eines sogenannten Basissuchfensters beschrieben und werden entsprechend der Höhe der Suchfenster mitskaliert (siehe Kapitel 3.1). Das Basissuchfenster stellt damit auch das Suchfenster mit der kleinsten detektierbaren Größe dar.

Um die Kontur der Fußgänger und deren Kontrast zur Umgebung richtig erfassen zu können, wird darüber hinaus im Suchfenster ein Rand um das eigentliche Objekt mit einbezogen. Ein Suchfenster ist also immer etwas größer als die tatsächliche Darstellung eines Fußgängers im Bild. Die Größe des Randbereichs ist dabei implizit durch die Objektfensterdefinition festgelegt. Diese beschreibt die Lage und Größe eines Fußgängerabbildes im Suchfenster (Abbildung 2.10). Die Definition erfolgt dabei

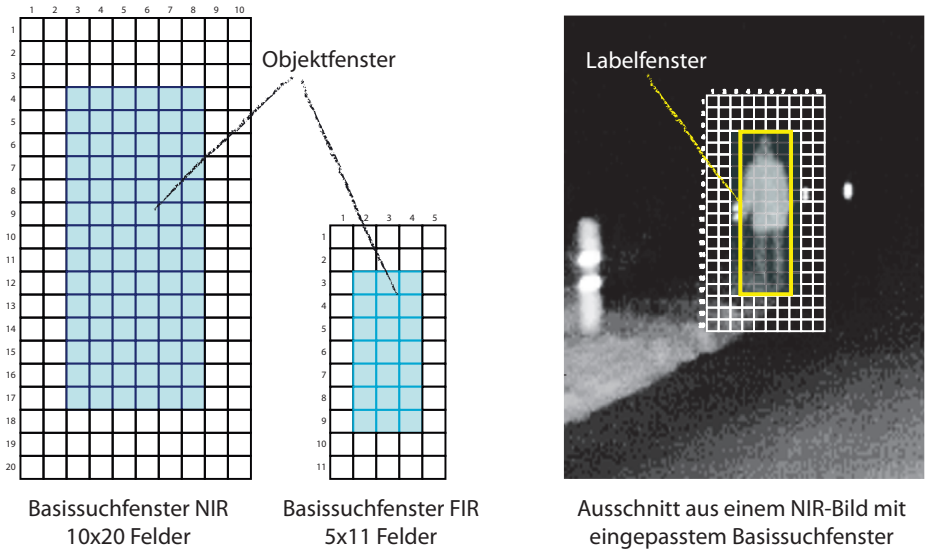


Abbildung 2.10.: Definition des Objektfensters innerhalb des Basissuchfensters. Das Objektfenster (blau markierter Bereich) definiert die Lage und Größe eines Fußgängerabbildes innerhalb des Suchfensters. Es ist in der Regel kleiner als das Suchfenster, um so einen Randbereich um den eigentlichen Fußgänger zu erhalten. Damit können bei der Klassifikation auch die Kontur der Fußgänger und Kontrastunterschiede zum Hintergrund berücksichtigt werden.

im Raster des Basissuchfensters. Die Größe des Randbereichs wird damit relativ zur Objektfenstergröße definiert und skaliert mit der Größe des Suchfensters mit.

Obwohl die Definition der Objektfenster im Raster des Basissuchfensters erfolgt, kann das Objektfenster o einer Hypothese genauso wie das Suchfenster s über den Mittelpunkt $(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ und dessen Höhe h angegeben werden:

$$o = (\mathrm{col}, \mathrm{row}, h)\,.$$

Die Höhe eines Objektfensters im Bild wird dabei auch als Skalierung bezeichnet. In dieser Arbeit ist für ein Suchfenster genau ein Objektfenster definiert (bei der gleichzeitigen Detektion von unterschiedlichen Objekttypen ist dies nicht der Fall, siehe z.B. [KSPL06]). Die Umrechnung zwischen Objektfenster und Suchfenster wird dann mit χ abgekürzt, d.h.

$$s = \chi\,(o) \quad \text{und entsprechend} \quad o = \chi^{-1}\,(s)\,.$$

Der Klassifikator entscheidet jeweils für eine Hypothese, ob sie einen Fußgänger darstellt, oder nicht. Damit der Algorithmus zwischen „Fußgänger“ und „Hintergrund“ unterscheiden kann, wird er mit Hilfe einer großen Anzahl an Trainingsbeispielen trainiert (siehe Kapitel 3.4). Diese Trainingsbeispiele sind Hypothesen aus einem Trainingsdatensatz, in dem alle Fußgänger per Hand mit einem rechteckigem Label und

einem Klassenlabel $y \in \mathcal{Y} = \{-1, +1\}$ versehen sind. Jedes Label wird dazu in das Seitenverhältnis eines Objektfensters gebracht und entsprechend der Definitionen des Objektfensters im Basissuchfenster zu einem Suchfenster erweitert.

Um die Güte des Klassifikationsalgorithmus bzw. des kompletten Fußgängerdetektors bestimmen zu können, wird er auf einem Testdatensatz angewandt und die Detektionen mit den Labels verglichen. Zum Vergleich kommt dabei ein Überdeckungsmaß cov zum Einsatz. Dieses errechnet sich aus dem Quotienten der Schnitt- und Vereinigungsmenge der Flächen zweier Rechtecke:

$$\mathrm{cov}\,(A, B) = \frac{A \cap B}{A \cup B} \in [0, 1]\,. \tag{2.6}$$

Mit einem Schwellwert für die Überdeckung der Objektfenster mit dem Labelfenster wird zwischen korrekter Detektion und Falschalarm unterschieden. Neben der einfachen Überdeckung in (2.6) kommt in dieser Arbeit noch ein weiteres Überdeckungsmaß zum Einsatz, nämlich die Maximum-Überdeckung $\mathrm{cov}_{\max}$:

$$\mathrm{cov}_{\max}\,(A, B) = \frac{A \cup B}{\min\,(A, B)} \in [0, 1]\,. \tag{2.7}$$

Der Term $\min\,(A, B)$ in (2.7) bezeichnet dabei die kleinere der beiden Flächen A und B. Im Gegensatz zur Überdeckung (2.6) berücksichtigt die Maximum-Überdeckung (2.7) die Unterschiede der Skalierung nicht, so dass $\mathrm{cov}_{\max} = 1$ genau dann resultiert, wenn eines der beiden Rechtecke komplett innerhalb des anderen liegt, unabhängig von ihrer Größe.

Zur Vereinfachung der Notation werden die Flächen A und B in $\mathrm{cov}\,(A, B)$ und $\mathrm{cov}_{\max}\,(A, B)$ in dieser Arbeit sowohl in der Form von Quadrupel $(\mathrm{col}, \mathrm{row}, w, h)$, als auch in Form von Objektfenstern $o = (\mathrm{col}, \mathrm{row}, h)$ angegeben. Das Quadrupel $(\mathrm{col}, \mathrm{row}, w, h)$ beschreibt dabei ein Rechteck mit der linken oberen Ecke $(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$, der Breite w und der Höhe h.

KAPITEL 3

Kaskadierte Klassifikatoren zur Detektion von Fußgängern

Die Fußgängererkennung erfolgt in dieser Arbeit in Form einer Top-Down-Strategie. Im Suchraum werden an allen sinnvollen Positionen Hypothesen generiert, für die dann ein Klassifikator entscheidet, ob es sich dabei um einen Fußgänger handelt oder nicht. Ein entsprechender Klassifikator muss also jeweils ein Zwei-Klassen-Problem lösen, um die Vordergrundklasse Fußgänger vom Hintergrund zu unterscheiden.

Die Lösung dieses Zwei-Klassen-Problems im Kontext einer Anwendung bei Nacht, stellt eine besondere Herausforderung dar: Zum Einen muss den speziellen Beleuchtungsverhältnissen bei Nacht Rechnung getragen werden, zum Anderen bewegen sich sowohl die Fußgängerobjekte, als auch der Hintergrund, dessen scheinbare Bewegung durch die Eigenbewegung des Fahrzeugs hervorgerufen wird. Die Bewegungseffekte werden in der Nacht aufgrund der langen Integrationszeiten der Kamera bei der Bildbelichtung sogar noch verstärkt (sog. „motion blur“, Abbildung 3.1). Darüber hinaus erscheinen Fußgänger im Bild in unterschiedlichen Skalierungen. Die Abbildung eines 1.80 m großen Fußgängers in 40 m Entfernung ist im NIR-Bild 67 px hoch, im FIR-Bild 22 px hoch. Dagegen ist derselbe Fußgänger in 120 m Entfernung im NIR-Bild noch 22 px, im FIR-Bild nur noch 8 px groß (Abbildung 3.2 und Abbildung 3.3). Abbildung 3.4 zeigt einen Querschnitt unterschiedlichster Fußgängerbilder, die in dieser Arbeit verwendet wurden.

Weitere Anforderungen an das Detektorsystem ergeben sich aus dem speziellen Anwendungsszenario im Automobilbereich mit dem Anspruch auf Echtzeitfähigkeit, Skalierbarkeit (also die Erweiterung auf andere Problemstellungen, z.B. der Detektion von Fahrzeugen) und Umsetzbarkeit in Hardware.

Grundlage des hier entwickelten Detektorsystems bildet der Kaskadenklassifikator von Viola und Jones [VJ01a, VJ01b, VJ04, AGW97]. Er wird im folgenden ohne Beschränkung der Allgemeinheit lediglich für den Ein-Sensor-Fall beschrieben. Eine

Abbildung 3.1.: Beispiele von „motion blur“ in Fußgängerbildern. Das mittlere und rechte Beispiel weisen einen hohen Anteil an Bewegungsunschärfe auf. Alle drei Beispiele haben in etwa dieselbe Skalierung.

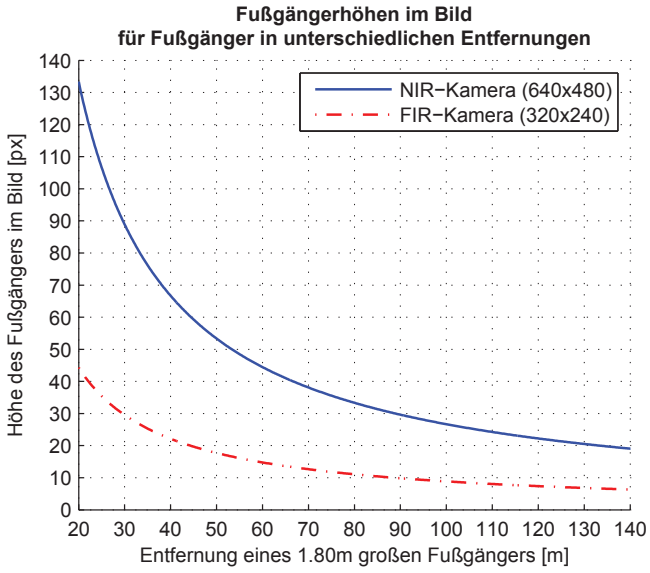


Abbildung 3.2.: Fußgängerhöhen im Bild. Fußgänger (1.80 m Körpergröße) in unterschiedlichen Entfernungen erscheinen im Bild in unterschiedlichen Skalierungen. Vor allem im FIR-Bild sind wegen der geringen Bildauflösung weit entfernte Fußgänger nur noch wenige Pixel hoch (8 px in einer Entfernung von 120 m).

Abbildung 3.3.: Beispielbilder der NIR-Kamera von Fußgängern aus unterschiedlichen Entfernungen. Die Darstellung links zeigt fünf Fußgänger aus unterschiedlichen Entfernungen (bis ca. 100 m) in realen Größenverhältnissen im Bild. Rechts sind alle Fußgänger auf dieselbe Größe skaliert dargestellt.

Abbildung 3.4.: Beispielbilder von Fußgängern. Dargestellt ist ein Querschnitt aus dem Trainingsdatensatz. Alle Fußgängerausschnitte wurden dabei auf diesselbe Größe skaliert und zur besseren Darstellung bezogen auf den jeweiligen Ausschnitt normiert. In der Matrix links sind jeweils die Ausschnitte aus den FIR-Bildern, rechts jeweils die dazugehörigen Ausschnitte aus den NIR-Bildern dargestellt.

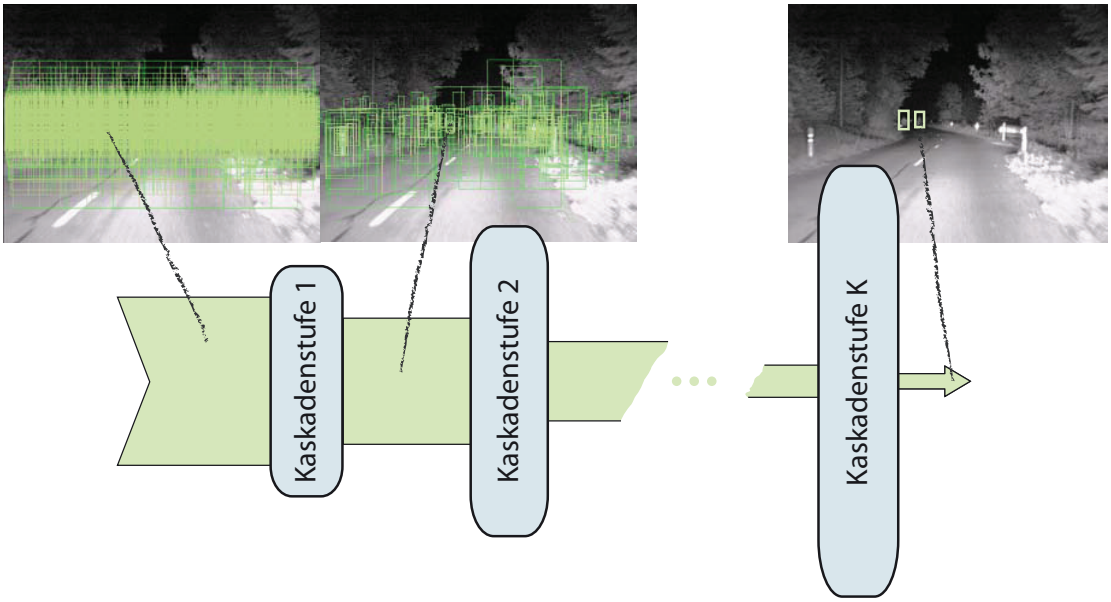


Abbildung 3.5.: Kaskadenklassifikator zur Detektion von Fußgängern. Objekthypothesen werden an allen möglichen Positionen im Bild in allen sinnvollen Größen erzeugt und dem ersten Klassifikator übergeben (linkes Bild). Dieser versucht mit möglichst wenig Aufwand möglichst viele Hypothesen sicher als Hintergrund zu klassifizieren. Der jeweils nachfolgende Klassifikator muss nur noch die verbleibenden Hypothesen betrachten. Während die Anzahl der zu untersuchenden Objekte mit jeder erreichten Stufe weiter abnimmt, werden die Einzelklassifikatoren immer komplexer. Durch den kaskadierten Ablauf bleibt der Gesamtaufwand aber sehr niedrig.

Hypothese $x \in \mathcal{X}$ besteht demnach jeweils aus genau einem Suchfenster im Bild. Die Erweiterung auf mehrere Sensoren ändert die Klassifikationsstruktur sowie das Lernverfahren nicht, so dass alle beschriebenen Eigenschaften direkt auf den im letzten Abschnitt 3.6 beschriebenen Mehr-Sensor-Fall übertragen werden können.

Der Kaskadenklassifikator von Viola und Jones arbeitet mit Grauwertbildern und zeichnet sich durch eine hohe Robustheit und Detektionsgeschwindigkeit aus. Dabei werden mehrere Einzelklassifikatoren in Reihe geschaltet (Abbildung 3.5). Jede dieser Klassifikatorstufen versucht mit möglichst wenig Aufwand möglichst viele Bereiche im Bild als Hintergrund zu deklarieren. Dazu werden an allen möglichen Positionen im Suchraum in allen sinnvollen Größen Hypothesen im Bild generiert und dem ersten Einzelklassifikator übergeben. Dieser untersucht die Hypothesen anhand weniger, einfacher Merkmale und verwirft diejenigen, bei denen es sich sicher nicht um Fußgänger handelt. Die jeweils nachfolgende Klassifikationsstufe muss nur noch die verbleibenden Hypothesen betrachten. Wie beim Training eines Entscheidungsbaumes [AGW97] wird jede Klassifikationsstufe nur mit den Beispielen trainiert, die in den vorangegangenen Stufen nicht verworfen wurden. Die Klassifikationsaufgabe, die Fußgänger vom Hintergrund zu unterscheiden, wird also mit jeder Stufe immer schwieriger. Aufgrund des deutlich reduzierten Suchbereichs können allerdings immer komplexere Klassifikatoren mit mehr Merkmalen eingesetzt werden. Hypothesen, die das Ende der Kaskade erreichen

werden als Fußgänger klassifiziert. Die möglichen Hypothesen werden also sukzessive ausgedünnt, während die Komplexität und damit der Aufwand der Klassifikatoren von Stufe zu Stufe wächst. Durch die Kaskadierung bleibt der Gesamtaufwand aber niedrig.

Dieses Grundprinzip eignet sich bei solchen Detektionsproblemen, bei denen der Hintergrund um ein vielfaches häufiger auftritt, als die zu detektierenden Objekte. Es eignet sich also insbesondere zur Detektion von Objekten in Bildern und Videoszenen: Man geht davon aus, dass ein Großteil der Hintergrundbeispiele leicht erkennbar ist und in frühen Kaskadenstufen aussortiert werden kann. Schwerer zu diskriminierende Beispiele benötigen dagegen komplexere und auch aufwändigere Klassifikatoren. Deren Klassenzugehörigkeit wird erst in den hinteren Stufen entschieden. Das Gesamtsystem ist damit erstaunlich schnell, da nur wenige Datensätze in diesen berechnungsintensiven Stufen der Kaskade bearbeitet werden müssen.

Jede Stufe kann prinzipiell der Kaskade aus anderen Klassifikatorstrukturen mit nahezu beliebig strukturierten Merkmalen zusammengesetzt sein. Unter dem Gesichtspunkt einer echtzeitfähigen Realisierung hat sich jedoch der von Viola und Jones vorgeschlagene Aufbau als effizient und praktikabel erwiesen. Analog zu [VJ01b] sind auch in dieser Arbeit alle Klassifikatorstufen identisch aufgebaut. Sie klassifizieren eine Hypothese aufgrund der Ergebnisse einfacher Kantenfilter, die den Haar-Basisfunktionen aus [OPS+97] ähneln und in Abschnitt 3.1 näher beschrieben werden. Sie sind von sehr einfacher Struktur und können mit Hilfe von Integralbildern in konstanter Zeit berechnet werden.

Mit Hilfe des sogenannten AdaBoost-Algorithmus [FS97, SS99] wird für jede Stufe aus einem Merkmalsatz auf Basis von Trainingsdaten eine Auswahl getroffen und zu einem sogenannten Stronglearner verknüpft. Dieser realisiert eine gewichtete Mehrheitsentscheidung verschiedener Weaklearner, die wiederum jeweils genau einen Merkmalswert mit einer trainierten Schwelle vergleichen. Die Anzahl der Merkmale (und damit die Anzahl der Weaklearner) bestimmt dann die Komplexität der Klassifikationsstufe. Der genaue Aufbau der Stronglearner sowie der AdaBoost-Algorithmus selbst werden im Detail in Abschnitt 3.2 beschrieben. In Abschnitt 3.3 werden dann die statistischen Grundlagen zur Herleitung von Rückschlusswahrscheinlichkeiten der gesamten Kaskade geschaffen. Diese, sowie die Besonderheiten beim Training und Anwendung der Kaskade sind Gegenstand von Kapitel 3.4. Rückschlusswahrscheinlichkeiten geben an, wie wahrscheinlich eine Hypothese auf Basis des Klassifikationsergebnisses tatsächlich einem Fußgänger entspricht. Diese Wahrscheinlichkeit wird in dieser Arbeit dann zur Gewichtung der Partikel in einem Partikelfilterverfahren benutzt, der durch den probabilistischen Ansatz die Anzahl der zu prüfenden Hypothesen im Bild reduzieren kann.

Abbildung 3.6.: Die drei unabhängigen 2D-Haarwavelets. Das Wavelet links repräsentiert horizontale, das in der Mitte vertikale und das rechts eckige Strukturen.

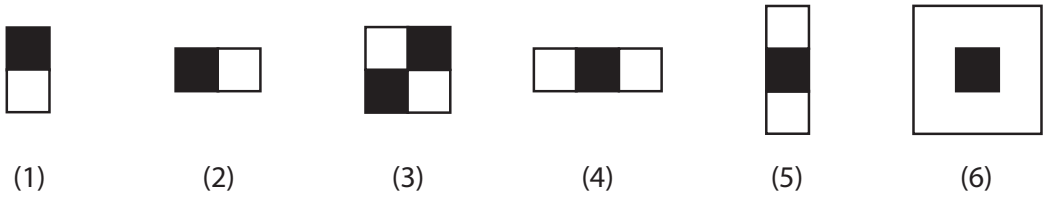


Abbildung 3.7.: Haarwavelet-ähnliche Basisfilter. Mit diesen sechs Basisfiltern wurden für diese Arbeit überbestimmte Filtersätze generiert. Durch die Erweiterung der 2D-Haarwavelets aus Abbildung 3.6 können objektspezifische Eigenschaften für die Klassifikation besser abgebildet werden. Die Filtertypen (4) und (5) repräsentieren dabei Liniensegmente und Typ (6) entspricht einem Center-surround Merkmal. Theoretisch sind diese weiteren Merkmale durch Kombinationen der Typen (1) und (2) bereits abgedeckt, können aber schneller berechnet werden da jeweils nur die Summen zweier Flächen voneinander subtrahiert werden müssen.

3.1. Haarwavelet-ähnliche Filter zur Objektdetektion

Personen unterscheiden sich sehr stark in Größe, Gestalt und Körperhaltung. Darüber hinaus weist z.B. die Abbildung weit entfernter Fußgänger aufgrund der niedrigen Auflösung im Gegensatz zu nahen Fußgängern wenig Textur auf. Texturunterschiede sind dabei auch bedingt durch die Reflexivitätseigenschaften der Kleidung (im NIR-Bild) bzw. durch deren unterschiedlicher Wärmeisolation (im FIR-Bild). Pixelbasierte Klassifikationsverfahren haben es damit sehr schwer im Einzelbild nur auf Grund der Intensitäten einzelner Pixel die Gemeinsamkeiten innerhalb der Objektklasse „Fußgänger“ zu erlernen. Das fundamentale Problem ist die genaue Charakterisierung dieser Klasse anhand von Merkmalen. In der Regel ist man dabei bestrebt, die Variabilität innerhalb der Objektklasse zu verringern.

In [OPS+97] wird durch eine Haarwaveletransformation die Information der Pixel im Kontext der direkten Nachbarschaft in verschiedene Anteile zerlegt, in der Hoffnung, damit die Variabilität innerhalb der Klasse zu verringern und so die Fußgänger besser vom Hintegrund trennen zu können. Die drei Typen der 2-dimensionalen Haarwavelets sind in Abbildung 3.6 dargestellt. Sie spiegeln lokale Intensitätsänderungen in horizontaler und vertikaler Richtung, sowie Ecken wieder. Der Ursprung dieser Filter liegt in der 2D-Haarwavelet Transformation (siehe z.B. [SDS02, Mal89]), die vor allem bei der Bildkompression breite Verwendung findet. Sie bilden dabei eine linear unabhängige Basis, die den Bildraum entsprechender Größe aufspannen. Für Details zur Haarwavelet Transformation sei z.B. auf [PL04] verwiesen. Diese Bestandteile sind voneinander

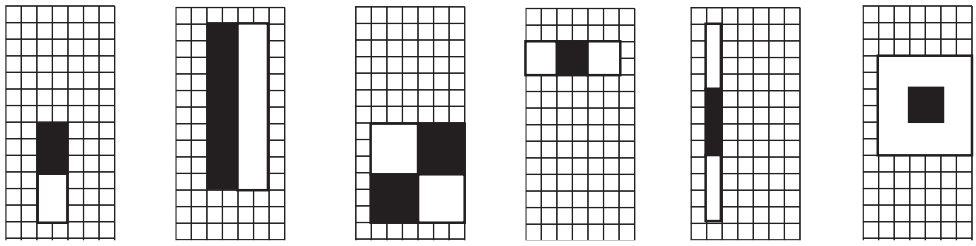


Abbildung 3.8.: Beispiele des überbestimmten Merkmalssatzes. Der Satz an Merkmalen ensteht, indem die Basisfilter aus Abbildung 3.7 an jeder möglichen Position mit allen zulässigen Größen im Basissuchfenster (hier 7px × 14px) plaziert werden. Insgesamt entstehen so 12 100 Merkmale. Der Merkmalssatz heißt überbestimmt, da zur exakten Codierung eines Bildausschnittes die drei Haarwavelets aus Abbildung 3.6 bereits ausreichend sind. Durch den überbestimmten Merkmalssatz ergeben sich allerdings bessere Repräsentationsmöglichkeiten des Signals, die die Klassifikationsaufgabe leichter lösbar machen.

unabhängig, allerdings können sie objektspezifische Eigenheiten teilweise immer noch sehr schlecht abbilden. In [POP98] und später in [LKP03] wurde deshalb der Filtersatz zu einem überbestimmten Satz aus Wavelet-ähnlichen Basisfunktionen erweitert.

Die einzelnen Filter sind damit zwar linear abhängig, dies führt aber aufgrund der dadurch besseren Repräsentationsmöglichkeiten des Signals zu Vorteilen bei der Lösung der Klassifikationsaufgabe. Zur Generierung der Filter werden im Basissuchfenster[1] alle Basisfilter an unterschiedlichen Positionen platziert und unterschiedlich skaliert. Die in dieser Arbeit verwendeten Haarwavelet-ähnlichen Basisfilter sind in Abbildung 3.7 dargestellt. Zusätzlich zu den drei unabhängigen 2D-Haarwavelet Typen werden dabei zwei Filtertypen verwendet, die Liniensegmente repräsentieren, sowie ein Center-surround Merkmal. Theoretisch sind diese zusätzlichen Filtertypen durch Kombinationen der drei unabhängigen 2D-Haarwavelets bereits abgedeckt, können aber schneller berechnet werden da jeweils nur die Summen zweier Flächen voneinander subtrahiert werden müssen. Beispiele aus dem überbestimmten Filtersatz zeigt Abbildung 3.8.

Zur Bestimmung der Filterantwort wird immer die Differenz zwischen der Summen der Grauwerte unter den weißen und den schwarzen Flächen gebildet. Um die Mittelwertfreiheit der Filter zu garantieren, werden die Summen jeweils mit den dazugehörigen Flächen normiert. Ein Merkmal m ist also jeweils über die Koordinaten der schwarzen und weißen Flächen im Basissuchfenster eindeutig definiert. Die Menge aller aus den Basisfiltern in Abbildung 3.7 im Basissuchfenster entstandenen Merkmale sei M.

Natürlich liegen reale Fußgängerhypothesen in der Regel nicht in der Skalierung dieses Basissuchfensters vor. Die meisten Detektionsalgorithmen verwenden deshalb Bildpyramiden, um das Bild in unterschiedlichen Skalierungen nach Objekten absuchen zu können. Die Erstellung solcher Bildpyramiden ist jedoch meist sehr rechenintensiv und stellt einen zusätzlichen Aufwand bei der Lösung der Detektionsaufgabe dar.

[1] Zur Definition von „Basissuchfenster“ siehe Kapitel 2.4.

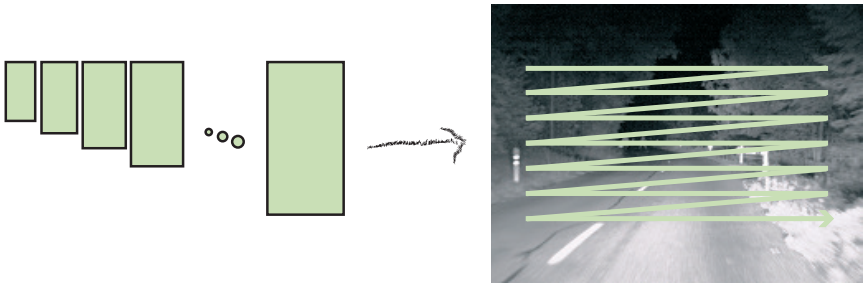


Abbildung 3.9.: Skalierung von Merkmalen. Zur Detektion von Fußgängern wird das Bild mit einem Suchfenster in allen sinnvollen Skalierungen abgescannt und die entsprechenden Hpyothesen der Kaskade präsentiert. Da die Merkmalsberechnung mit Hilfe des Integralbildes für skalierte Merkmale nicht aufwändiger als für nicht skalierte Merkmale ist, kann auf den Einsatz von Bildpyramiden verzichtet werden.

[VJ01a] geht deshalb den anderen Weg. Anstatt die Bilder zu skalieren, werden durch die Skalierung der Basissuchfenster die darin definierten Merkmale in die benötigte Größe gebracht (vgl. Abbildung 3.9). Da durch die Verwendung von Integralbildern [Cro84] die Größe der Filter keinen Einfluss auf den Berechnungsaufwand hat, lässt sich die Filterantwort in allen Fällen in konstanter Zeit berechnen.

Unter einem Integralbild $\boldsymbol{\mathcal{I}}(\boldsymbol{I})$ versteht man ein Bild, in dem an jedem Punkt $(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ die Summe aller Intensitätswerte des Bildes $\boldsymbol{I}$ abgelegt ist, die sich über und links dieses Punktes einschließlich des Punktes selbst befinden:

$$\boldsymbol{\mathcal{I}}(\mathrm{col}, \mathrm{row}) = \sum_{\substack{\mathrm{col}' \leq \mathrm{col} \\ \mathrm{col}' \leq \mathrm{row}}} \boldsymbol{I}(\mathrm{col}', \mathrm{row}') .$$

Die Pixelsumme eines rechteckigen Bereiches $(\mathrm{col}, \mathrm{row}, w, h)$ mit der linken oberen Ecke $(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$, der Breite w und der Höhe h kann dann unabhängig von dessen Größe mit

$$\begin{aligned} \sum_{\substack{\mathrm{col} \leq \mathrm{col}' \leq \mathrm{col}+w \\ \mathrm{row} \leq \mathrm{row}' \leq \mathrm{row}+h}} \boldsymbol{I}(\mathrm{col}', \mathrm{row}') = {} & \boldsymbol{\mathcal{I}}(\mathrm{col}, \mathrm{row}) - \boldsymbol{\mathcal{I}}(\mathrm{col}, \mathrm{row}+h) \\ & - \boldsymbol{\mathcal{I}}(\mathrm{col}+w, \mathrm{row}) + \boldsymbol{\mathcal{I}}(\mathrm{col}+w, \mathrm{row}+h) \end{aligned}$$

in konstanter Zeit mit drei Additionen berechnet werden. Somit spielt es keine Rolle, wie groß die Filter sind. Die Basissuchfenster zusammen mit den Filtern zu skalieren, ist daher effektiver, als eine aufwändige Bildpyramide zu berechnen. Voraussetzung ist lediglich ein Integralbild. Dieses kann durch

$$\begin{aligned} s(\mathrm{col}, \mathrm{row}) &= s(\mathrm{col}, \mathrm{row}-1) + \boldsymbol{I}(\mathrm{col}, \mathrm{row}) \\ \boldsymbol{\mathcal{I}}(\mathrm{col}, \mathrm{row}) &= \boldsymbol{\mathcal{I}}(\mathrm{col}-1, \mathrm{row}) + s(\mathrm{col}, \mathrm{row}) \end{aligned}$$

mit einem einzigen Bilddurchlauf sehr effizient berechnet werden. Der Merkmalswert $m(x)$ für eine gegebene Hypothese x wird also berechnet, indem die auf das Basissuchfenster bezogenen Koordinaten der Beschreibung des Merkmals $m \in M$ auf das Suchfenster skaliert wird und die entsprechenden Summen mit Hilfe des Integralbildes $\boldsymbol{\mathcal{I}}$ gebildet werden.

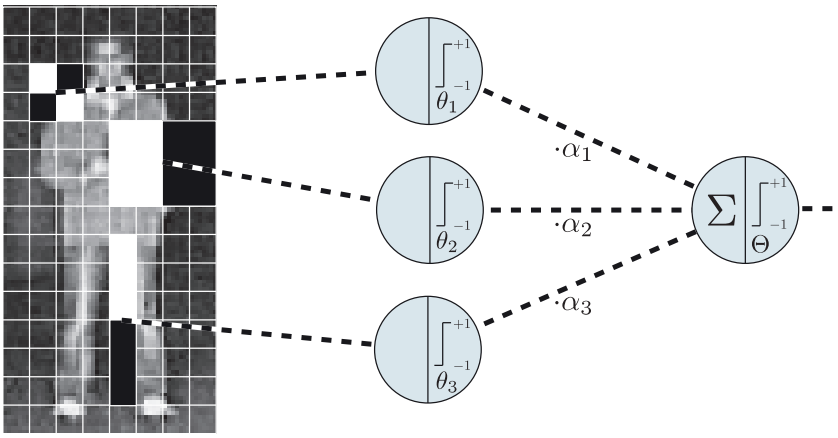


Abbildung 3.10.: Aufbau einer Kaskadenstufe. Jeder Merkmalswert wird im Weaklearner über eine einfache Schwellwertoperation auf +1 oder −1 abgebildet. Diese Weaklearnerergebnisse werden dann im finalen Stronglearner gewichtet aufsummiert und mit dem Stronglearnerschwellwert verglichen (für $\Theta = 0$: gewichtete Mehrheitsentscheidung).

Innerhalb des Klassifikators selbst wird das Ergebnis der einzelnen Merkmale mit einem Schwellwert verglichen. Auf diese Weise erhält man einen sehr einfachen Klassifikator, der im Folgenden deshalb auch als Weaklearner h bezeichnet wird:

$$h(x) = \begin{cases} +1 & \text{pol} \cdot m(x) < \text{pol} \cdot \theta \\ -1 & \text{sonst.} \end{cases} \tag{3.1}$$

Er weist einer Hypothese x eine Klasse in $\{-1, +1\}$ zu, abhängig vom Filterergebnis $m(x)$, $m \in M$, einem Schwellwert $\theta \in \mathbb{R}$ und der Polarität pol $\in \{-1, +1\}$. Innerhalb einer Kaskadenstufe werden also die Merkmale über solche Weaklearner abgebildet. Mehrere Weaklearner $h_t, t = 1, \ldots, T$ werden über eine gewichtete Mehrheitsentscheidung zu einem Stronglearner H kombiniert und bilden somit eine Stufe der Kaskade:

$$H(x) = \begin{cases} +1 & A(x) \geq \Theta \\ -1 & \text{sonst,} \end{cases} \tag{3.2}$$

mit

$$A(x) := \sum_{t=1}^{T} \alpha_t h_t(x).$$

T ist die Anzahl der einzelnen Weaklearner, α_t ist das Gewicht des Weaklearners h_t, der das Merkmal $m_t \in M$ abbildet. $A(x)$ heißt Aktivierung und wird mit der Stronglearnerschwelle Θ verglichen. Eine Hypothese x wird damit wieder einer Klasse in $\{-1, +1\}$ zugewiesen. Abbildung 3.10 zeigt eine Visualisierung dieser Stronglearnerstruktur.

Natürlich werden in einem Stronglearner nicht alle möglichen Weaklearner aus dem überbestimmten Merkmalssatz verwendet. Schließlich will man gerade in den ersten

Kaskadenstufen mit möglichst wenig Merkmalen auskommen. Andererseits ist die Auswahl der sinnvollen Weaklearner per Hand natürlich nicht möglich. Die Entscheidung, welche Weaklearner zu einem Stronglearner kombiniert werden erfolgt deshalb mit dem statistischen Lernverfahren AdaBoost, das gleichzeitig mit der Bestimmung aller Schwellen und Gewichten auch festlegt, in welcher Weise die Verknüpfung stattfindet.

3.2. Boosting

Generelle und allgemeingültige Regeln zur Lösung von Problemen zu finden ist meist sehr schwierig. Es ist wesentlich leichter einige Faustregeln aufzustellen, die menschlich intuitiv eingesetzt werden. Angenommen, man möchte z.B. einen Spam-Filter entwickeln, der unerwünschte Email im elektronischen Postfach automatisch ausfiltert, dann ist eine mögliche Faustregel z.B. „enthält eine Email die Phrase 'buy now', so klassifiziere sie als Spam“ [Sch03a]. Natürlich sind solche Faustregeln nicht immer zuverlässig und sie lösen die gestellte Aufgabe nur sehr unzureichend. Dennoch ist jede dieser Faustregeln meist besser als der Zufall.

Boosting im Allgemeinen beschäftigt sich mit der Aufgabe, aus einer Sammlung solcher Faustregeln einen zuverlässigen Klassifikator abzuleiten. Es ist ein Verfahren zur Erstellung einer effizienten Entscheidungsregel (Stronglearner), um mehrere einfache Klassifikatoren (Weaklearner) zu kombinieren. Erste theoretische Arbeiten zu diesem Gebiet entstanden von [Sch90] und [Fre95], die schließlich in [FS97] und [SS99] den praktisch relevanten AdaBoost-Algorithmus hervorbrachten. AdaBoost erweitert dabei sukzessive eine bestehende Kombination aus Weaklearnern, indem der Fehler auf einer gewichteten Beispielmenge minimiert wird. Nach jedem Auswahlschritt werden dabei die Beispiele in der Trainingsmenge neu gewichtet: falsch klassifizierte Beispiele werden für den nächsten Auswahlschritt höher gewichtet, richtig klassifizierte dagegen niedriger. Am Ende bildet eine gewichtete Mehrheitsentscheidung über alle Weaklearner das Ergebnis des Stronglearners (Gleichung (3.2)).

Allgemein lässt sich das Vorgehen aller Boosting-Algorithmen folgendermaßen zusammenfassen:

1. Versehe jedes Trainingsbeispiel $x^{(i)}$ mit einem Gewicht $d^{(i)}, i = 1, \ldots, N$.
2. Trainiere einen Weaklearner auf Basis dieser gewichteten Beispiele.
3. Überprüfe, wie gut der Weaklearner arbeitet und wähle entsprechend seine Gewichtung α_t im Stronglearner.
4. Gewichte die Trainingsmenge neu und wiederhole das Training mit dem nächsten Weaklearner.
5. Ergebnis des Stronglearners ist die gewichtete Mehrheitsentscheidung aller Weaklearner.

1. Ausgehend von der Trainingsmenge $\{(x_1, y_1), \ldots, (x_N, y_N)\}$, mit $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y} = \{-1, +1\}$, $i = 1, \ldots, N$ werden die Gewichte $d_1^{(1)}, \ldots, d_1^{(N)}$ initialisiert mit $d_1^{(i)} = \frac{1}{N}, i = 1, \ldots, N$.

2. Für $t = 1, \ldots, T$:

   - Trainiere alle Weaklearner entsprechend der aktuellen Gewichtung der Trainingsbeispiele.
   - Bestimme den besten Weaklearner $h_t \colon \mathcal{X} \to \{-1, +1\}$, der den gewichteten Trainingsfehler minimiert.
   - Berechne den gewichteten Trainingsfehler:

$$\epsilon_t = \sum_{i : y^{(i)} \neq h_t\left(x^{(i)}\right)} d_t^{(i)}.$$

   - Wähle

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}. \tag{3.3}$$

   - Aktualisiere

$$d_{t+1}^{(i)} = \frac{d_t^{(i)} \exp\left\{-\alpha_t y^{(i)} h_t\left(x^{(i)}\right)\right\}}{Z_t}, \tag{3.4}$$

     mit dem Normalisierungsfaktor Z_t, so dass $\left\{d_{t+1}^{(1)}, \ldots, d_{t+1}^{(N)}\right\}$ eine Verteilungsfunktion darstellt.

3. Ausgabe ist der Gesamtklassifikator

$$H(x) = \operatorname{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) = \begin{cases} +1 & A(x) \geq \Theta = 0 \\ -1 & \text{sonst.} \end{cases}$$

Algorithmus 3.1: Diskreter AdaBoost-Algorithmus.

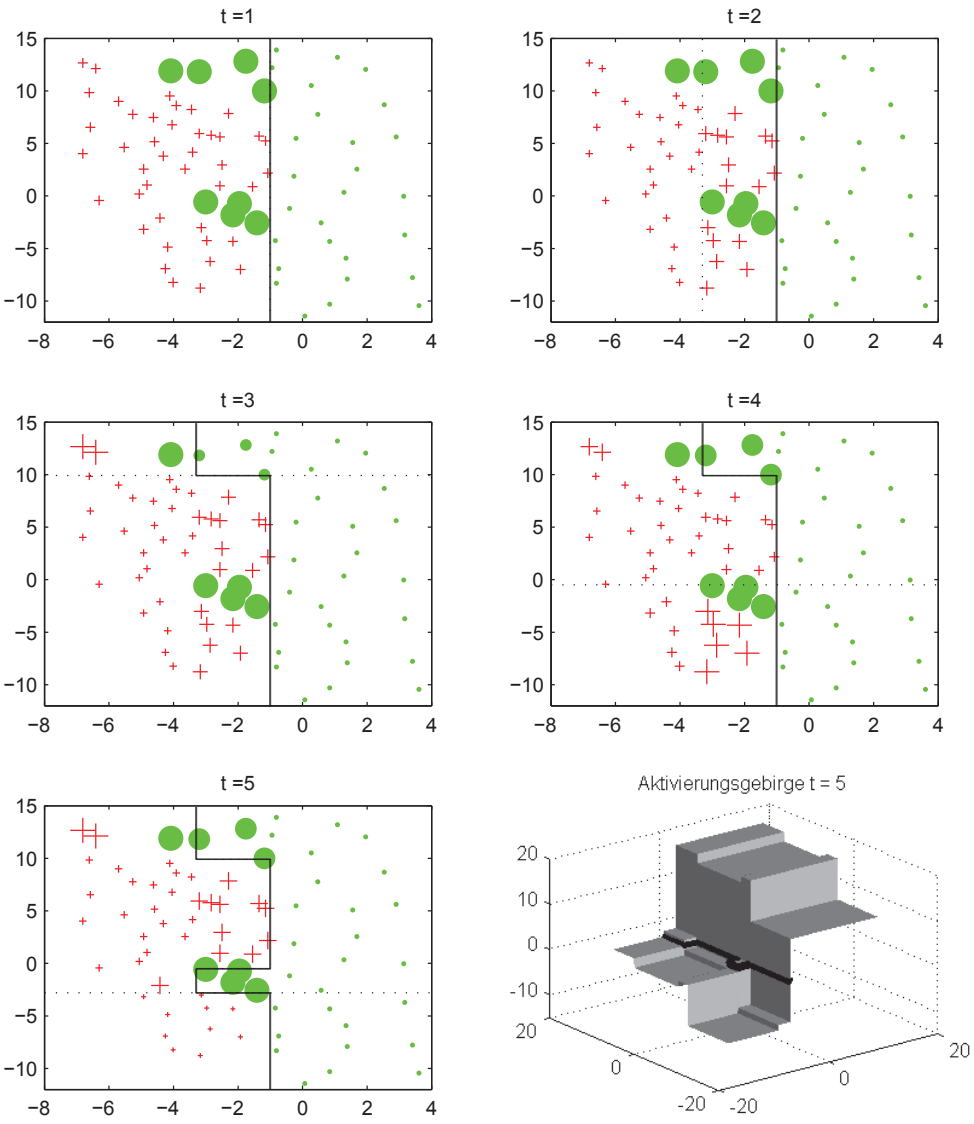


Abbildung 3.11.: Ablauf des AdaBoost-Algorithmus am Beispiel künstlicher Daten. Dargestellt sind jeweils die gewichteten Trainingsdaten, die Entscheidungsgrenze (durchgezogene Linie) und der zuletzt ausgewählte Weaklearner (gestrichelte Linie), nach $t = 1, 2, 3, 4$ und $t = 5$ Runden. Positivbeispiele sind dabei als rote Kreuze, Negativbeispiele als grüne Punkte dargestellt. Die Größe der Markierung spiegelt das jeweilige Gewicht $d_t^{(i)}$ wieder. Nach $t = 5$ Runden ist der Trainingsfehler bereits nahezu Null. Das Aktivierungsgebirge rechts unten zeigt den Verlauf von $A : \mathbb{R}^2 \to \mathbb{R}$, mit $x \mapsto A(x) = \sum_{t=1}^{5} \alpha_t h_t(x)$.

Im konkreten Fall des diskreten AdaBoost-Algorithmus ergibt sich Algorithmus 3.1. Abbildung 3.11 stellt den Ablauf am Beispiel künstlicher Daten exemplarisch dar.

Die Aufgabe des Weaklearning-Algorithmus ist es, einen Weaklearner h_t entsprechend der Verteilung $\left\{d_t^{(1)}, \ldots, d_t^{(N)}\right\}$ zu trainieren. Im einfachsten Fall ist h_t wie in (3.1) binär, d.h. $h_t(x) \in \{-1, +1\}$. Der Weaklearning-Algorithmus minimiert dann den gewichteten Trainingsfehler

$$\epsilon_t := \mathbb{P}_{i \sim D_t}\left(h_t\left(x^{(i)}\right) \neq y^{(i)}\right) = \sum_{i: y^{(i)} \neq h_t\left(x^{(i)}\right)} d_t^{(i)} = 1 - \sum_{i: y^{(i)} = h_t\left(x^{(i)}\right)} d_t^{(i)}. \tag{3.5}$$

Wurde ein Weaklearner ausgewählt, wird mit α_t dessen Einfluss im finalen Stronglearner festgelegt. Die Wahl von α_t erfolgt dabei nicht willkürlich, sondern so, dass der Trainingsfehler des gesamten Klassifikators minimiert wird.

3.3. Theoretische Eigenschaften von AdaBoost

Die algorithmische Darstellung von AdaBoost in Algorithmus 3.1 gibt keinen Einblick in dessen eigentliche Funktionsweise. Zum Verständnis der Funktionsweise von AdaBoost werden im Folgenden nur die grundlegenden theoretischen Untersuchungen zu AdaBoost zusammengefasst, insbesondere wird aufgezeigt, welche Fehlerfunktion minimiert wird und woraus sich die Gewichtsaktualisierung in (3.4) ableitet. Für tiefergehende theoretische Einblicke in die Eigenschaften von AdaBoost sei als Einstiegspunkt auf die Übersichtsarbeiten von [Sch99, Sch02, Rid99], oder auch [MR03] verwiesen. Die wichtigsten Eigenschaften von AdaBoost sind dabei:

1. Der Trainingsfehler ist nach oben beschränkt (Satz 1).
2. Der Trainingsfehler nimmt unter bestimmten Voraussetzungen mit zunehmender Rundenzahl exponentiell ab (Satz 2).
3. Die Wahl von α_t in (3.3) sorgt dafür, dass die obere Schranke des Trainingsfehlers minimiert wird (Satz 3).
4. AdaBoost stellt ein Gradientenabstiegsverfahren dar. Im Prinzip ist es ein Verfahren zur Bestimmung einer Linearkombination von Basisklassifikatoren zur Minimierung der oberen Schranke des Trainingsfehlers.
5. Die Rückschlusswahrscheinlichkeiten von Beispielen bei der Klassifikation durch AdaBoost können durch

$$p(y = +1|x) = \frac{\exp\{2A(x)\}}{1 + \exp\{2A(x)\}}, \text{ mit } A(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

   angenähert werden (Satz 5).
6. Der Generalisierungsfehler von AdaBoost-Klassifikatoren ist beschränkt (Gleichung 3.14).

Von Bedeutung ist dabei natürlich zunächst die Frage nach dem Trainingsfehler, d.h. wie gut AdaBoost in der Lage ist, die Trainingsdaten zu repräsentieren und in die jeweils richtigen, aus dem Training bekannten Klassen einzuordnen. Formal ist der Trainingsfehler der Anteil an Falschklassifikationen im Trainingsdatensatz und ist gegeben durch

$$\mathrm{err}_{\mathrm{Training}} := \frac{1}{N} \left| \left\{ x^{(i)} \left| H\left(x^{(i)}\right) \neq y^{(i)} \right. \right\} \right|. \tag{3.6}$$

Für Klassifikatoren, die mit AdaBoost trainiert werden gilt, dass dieser Trainingsfehler nach oben beschränkt ist:

Satz 1 ([FS97])**.** *Mit der Notation aus Algorithmus 3.1 gilt, dass der Trainingsfehler (*3.6*) nach oben beschränkt ist, mit*

$$\mathrm{err}_{\mathrm{Training}} \leq \frac{1}{N} \sum_{i=1}^{N} \exp\left\{ -y^{(i)} A\left(x^{(i)}\right) \right\} = \prod_{t=1}^{T} Z_t, \tag{3.7}$$

$$A(x) = \sum_{t=1}^{T} \alpha_t h_t\left(x^{(i)}\right)$$

*und $Z_t, t = 1, \ldots, T$ die Normalisierungskonstante in (*3.4*), mit*

$$Z_t = \sum_{i=1}^{N} d_t^{(i)} \exp\left\{ -\alpha_t y^{(i)} h_t\left(x^{(i)}\right) \right\}.$$

Der Beweis findet sich in Anhang A, Seite 171.

Die obere Schranke (3.7) ist vergleichsweise schwach. In der Praxis verläuft der tatsächliche Trainingsfehler meist weit darunter (vgl. Abbildung 3.12). Allerdings ist die obere Schranke ein Garant dafür, dass AdaBoost den Fehler auf jeder beliebigen Trainingsmenge nach einer endlichen Anzahl von Runden unter eine vorgegebene Schwelle absenken kann, vorausgesetzt die einzelnen Weaklearner entscheiden alle nur ein klein wenig besser als der Zufall, wie der folgende Satz zeigt:

Satz 2. *Wenn jeder der Weaklearner nur ein klein wenig besser entscheidet als der Zufall, so dass $\gamma_t := \frac{1}{2} - \epsilon_t \geq \gamma$ für irgendein $\gamma > 0$, sinkt der Trainingsfehler exponentiell in T:*

$$\mathrm{err}_{\mathrm{Training}} \leq \prod_{t=1}^{T} Z_t \leq \exp\left\{ -2T\gamma^2 \right\}.$$

Der Trainingsfehler sinkt dabei also sogar exponentiell in T. Der Beweis findet sich in Anhang A, Seite 173.

Zusammen mit der am Ende dieses Kapitels diskutierten Schranke für den Generalisierungsfehler zeigen Satz 1 und Satz 2, dass AdaBoost wirklich in der Lage ist, effizient

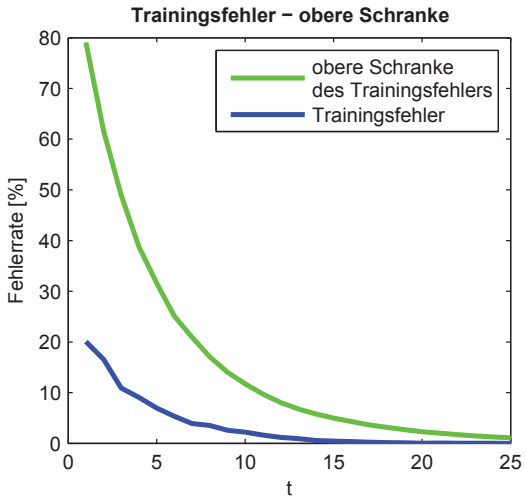


Abbildung 3.12.: Obere Schranke des Trainingsfehlers. Die grüne, obere Kurve zeigt die obere Schranke des Trainingsfehlers (blaue, untere Kurve) eines Trainings auf dem *twonorm*-Datensatz[1] von [Bre98].

und effektiv schwache Klassifikatoren zu einem starken Klassifikator zu kombinieren. Jeder ausgewählte Weaklearner wird dabei mit dem Gewicht α_t versehen, dessen Wahl sich über die Minimierung der oberen Schranke motivieren lässt:

Satz 3 (Wahl von α_t)**.** *Für* $h_t\left(x^{(i)}\right) \in \{-1, +1\}, i = 1, \ldots, N$ *minimiert*

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$$

in jeder Runde von AdaBoost die obere Schranke aus Satz 1.

Der Beweis findet sich in Anhang A, Seite 174.

Mit dieser Wahl von α_t wird also die obere Schranke des Trainingsfehlers in jeder Runde von AdaBoost minimiert.

Die obere Schranke (3.7) kann auch in Abhängigkeit des sogenannten Margins der Trainingsbeispiele formuliert werden. Der Margin ist dabei ein Maß für die Sicherheit, mit der ein Beispiel klassifiziert wird und repräsentiert dabei den Abstand der Trainingsbeispiele von der Entscheidungsgrenze im Merkmalsraum. Im binären Fall ist der Margin definiert mittels

$$\rho\left(\alpha_{1:t}, x, y\right) := yA\left(x\right) = y \sum_{r=1}^{t} \alpha_r h_r\left(x\right) \in \mathbb{R}. \tag{3.8}$$

[1] 20-dimensionaler, 2-Klassen-Datensatz, beschrieben in [Bre98]. Die Beispiele der ersten Klasse stammen von einer Normalverteilung um den Mittelwert $(a, a, \ldots, a)$, die der zweiten Klasse aus einer Normalverteilung um den Mittelwert $(-a, -a, \ldots, -a)$, jeweils mit $a = \frac{2}{\sqrt{20}}$ und der Einheitsmatrix als Kovarianzmatrix.

Der Margin ist für ein Beispiel x genau dann positiv, wenn das korrekte Klassenlabel y vorhergesagt wird. Je größer der Wert, desto eindeutiger ist dabei die Klassifikation.

Setzt man (3.8) in (3.7) ein, kann man die Formulierung der oberen Schranke des Trainingsfehlers durch den Margin ausdrücken und damit das Funktional

$$J(\rho) := \mathbb{E}\left[\exp\left\{-\rho\left(\alpha_{1:t}, x, y\right)\right\}\right] = \frac{1}{N}\sum_{j=1}^{N}\exp\left\{-\rho\left(\alpha_{1:t}, x^{(j)}, y^{(j)}\right)\right\} \tag{3.9}$$

aufstellen, das in allgemeinerer Form schon von [Bre97] (siehe auch [FD98, FHT00, MBBF99, Fri01]) formuliert wurde. $\mathbb{E}[\cdot]$ ist dabei der Erwartungswert des Margins aller Trainingsbeispiele.

Geht man davon aus, dass J eine Fehlerfunktion darstellt, die von AdaBoost minimiert wird, so stellt J eine Verlustfunktion dar, die abhängig vom Margin der jeweiligen Beispiele ist. Je größer der Margin der Beispiele, desto kleiner wird der Wert von J. Den Zusammenhang mit AdaBoost zeigt folgender Satz:

Satz 4 (Lemma 1 aus [ROM01])**.** *Die Gewichtsverteilung d_{t+1} in Algorithmus 3.1 leitet sich aus der Normalisierung des Gradienten $\frac{\partial J}{\partial\rho(\alpha_{1:t},x^{(i)},y^{(i)})}$ ab, d.h.*

$$d_{t+1}^{(i)} = \frac{\partial J(\rho)}{\partial\rho\left(\alpha_{1:t}, x^{(i)}, y^{(i)}\right)} \Big/ \sum_{j=1}^{N}\frac{\partial J(\rho)}{\partial\rho\left(\alpha_{1:t}, x^{(j)}, y^{(j)}\right)}.$$

Der Beweis folgt einer vervollständigten und korrigierten Herleitung aus [ROM01] und findet sich in Anhang A, Seite 174.

Der Gradient $\frac{\partial J}{\partial\rho(\alpha_{1:t},x^{(i)},y^{(i)})}$ gibt eine Antwort auf die Frage, welches Beispiel seinen Margin am stärksten erhöhen muss, damit J schnellstmöglich minimal wird (Gradientenabstieg). AdaBoost stellt damit ein Gradientenabstiegsverfahren zur Minimierung des Funktionals $J(\rho)$ dar. Die Richtung des Gradientenabstiegs ergibt sich aus dem Gradienten der Fehlerfunktion bezogen auf die zu optimierenden Parameter (hier eben $\frac{\partial J}{\partial\rho(\alpha_{1:t},x^{(i)},y^{(i)})}$, Satz 4). Die Schrittweite in Gradientenrichtung entspricht in Analogie dazu der Minimierung von J bezüglich α_t (Satz 3).

Im Grunde genommen ist AdaBoost damit ein Verfahren zur Bestimmung einer Linearkombination $A(x)$ von schwachen Klassifikatoren zur Minimierung von $J(\rho)$. Das Verfahren ist additiv, da die Linearkombination in jedem Zeitschritt über einen Gradientenabstieg (Newton-Update) um einen Term erweitert wird und es gilt:

Satz 5 (Lemma 1 in [FHT00])**.** $J(\rho) = J(A(x)) = \mathbb{E}\left[\exp\left\{-yA(x)\right\}\right]$ *nimmt sein Minimum an bei*

$$A(x) = \frac{1}{2}\ln\frac{p(y=1|x)}{p(y=-1|x)}.$$

Damit gilt:

$$p(y=1|x) = \frac{\exp\left\{2A(x)\right\}}{1+\exp\left\{2A(x)\right\}} \tag{3.10}$$

und

$$p(y=-1|x)=\frac{\exp\{-2A(x)\}}{1+\exp\{-2A(x)\}}.$$

Der Beweis von Satz 5 findet sich in Anhang A, Seite 176.

[FHT00] zeigen, dass AdaBoost in der Tat ein Verfahren darstellt, das ein additives logistisches Regressionsmodell der Form

$$\log\frac{p(y=1|x)}{p(y=-1|x)}=\sum_{t=1}^{T}\beta_t h_t(x)\text{, mit }\beta_t=2\alpha_t \tag{3.11}$$

aufbaut, indem schrittweise durch einen Gradientenabstieg zur Minimierung von J (3.11) um einen Term erweitert wird (vgl. auch die Bemerkungen zu Satz 4). Aufgelöst nach $p(y=1|x)$ ergibt wieder

$$p(y=1|x)=\frac{\exp\{2A(x)\}}{1+\exp\{2A(x)\}}. \tag{3.12}$$

Für die Motivation von J als Fehlerfunktional zeigen [FHT00] noch zusätzlich auf, dass die Minimierung von J äquivalent ist zur Minimierung des negativen Bernoulli Log-Likelihoods. Mit $p(x)$ wie in (3.12) definiert und $y^*:=\frac{1}{2}(y+1)\in\{0,1\}$ ist der negative Bernoulli Log-Likelihood

$$\begin{aligned}\mathbb{E}\left[-l\left(y^*,p(x)\right)\right]&=\mathbb{E}\left[-\left(y^*\ln p(x)+(1-y^*)\ln\left(1-p(x)\right)\right)\right]\\&=\mathbb{E}\left[\ln\left(1+\exp\{-2yA(x)\}\right)\right].\end{aligned} \tag{3.13}$$

Die Taylorerweiterung bis zum 2. Glied von $-l\left(y^*,p(x)\right)+1-\ln 2$ um $A(x)=0$ ist

$$-l\left(y^*,p(x)\right)+1-\ln 2\approx 1-yA(x)+\frac{1}{2}A(x)^2$$

und damit identisch der Taylorerweiterung von $\exp\{-yA(x)\}$. Der zusätzliche Term $1-\ln 2$ beeinflusst die Minimierung von (3.13) nicht und damit ist das Verhalten von (3.13) und $J=\exp\{-yA(x)\}$ nahe $A(x)=0$ sehr ähnlich. Darüber hinaus nimmt auch (3.13) sein Minimum an bei

$$A(x)=\frac{1}{2}\ln\frac{p(y=+1|x)}{p(y=-1|x)}.$$

Aufgrund all dieser Betrachtungen kann man davon ausgehen, dass (3.12) ein geeignetes Wahrscheinlichkeitsmodell zur Näherung der Rückschlusswahrscheinlichkeiten $p(y=+1|x)$ darstellt (Abbildung 3.13). Diese sind für die vorliegende Arbeit von zentraler Bedeutung, da zur probabilistischen Zustandsschätzung Wahrscheinlichkeiten vorliegen müssen. Dazu wird in Abschnitt 3.5 das Wahrscheinlichkeitsmodell auf die

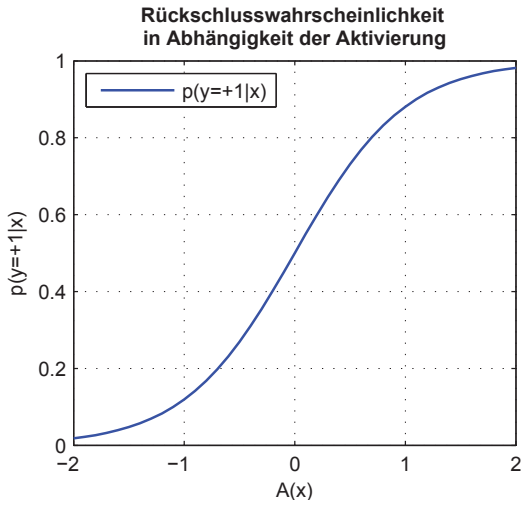


Abbildung 3.13.: Rückschlusswahrscheinlichkeit in Abhängigkeit der Aktivierung. Dargestellt ist der Verlauf von $p(y=1|x) = \frac{\exp\{2A(x)\}}{1+\exp\{2A(x)\}}$ aus Gleichung (3.10).

ganze Kaskade erweitert (die Betrachtungen hier sind jeweils auf eine Kaskadenstufe begrenzt).

Neben der probabilistischen Interpretation eines Beispiels bei der Klassifikation durch AdaBoost sind bei der Untersuchung der Eigenschaften von AdaBoost natürlich auch dessen Generalisierungseigenschaften von Bedeutung. Der Generalisierungsfehler ist die Wahrscheinlichkeit, dass ein neues, unbekanntes Beispiel falsch klassifiziert wird und spiegelt damit die Güte der Klassifizierung auf einem unbekannten Testset wieder.

In der Regel geht man davon aus, dass beim Erreichen eines Trainingsfehlers von null die Generalisierungseigenschaften mit zunehmender Rundenzahl eher schlechter werden, da ein Lernverfahren in solchen Situationen die Trainingsdaten meist „auswendig“ lernt. Dies legt nahe, dass auch Boosting zum Überlernen führen kann. Immerhin konvergiert bei AdaBoost der Trainingsfehler exponentiell schnell gegen null (vgl. Satz 2). Dennoch zeigten bereits verschiedene frühe empirische Untersuchungen [Qui96, DC96, Bre97], dass Boosting zumindest bei gering verrauschten Datensätzen in der Regel nicht überadaptiert. Der Fehler auf dem Testset nimmt sogar mit zunehmender Rundenzahl weiter ab, selbst dann, wenn der Trainingsfehler schon lange null erreicht hat. Dieses Verhalten haben [SFBL98] analysiert. Das Ergebnis dieser Untersuchungen ist eine obere Schranke des Generalisierungsfehlers $\mathbb{P}(yA(x) \leq 0) = \mathbb{P}(\rho \leq 0)$ in Abhängigkeit der VC-Dimension[2] d der Weaklearner und des Margins ρ auf der Trainingsmenge: Mit einer Wahrscheinlichkeit von mindestens $1-\delta$ und für alle $\hat{\rho} > 0$ ist der Generalisierungsfehler

[2]Die Vapnik-Chervonenkis (VC)-Dimension ist ein Maß für die „Komplexität“ eines Funktionenraums binärer Funktionen. Im Kontext von Klassifikatoren entspricht es der maximalen Anzahl von Datenpunkten, bei der eine einzelne Klassifikatorfunktion in der Lage ist unter allen denkbaren Möglichkeiten beide Klassen zu separieren. Die bei den Beispielen in diesem Kapitel verwendeten Weaklearner haben jeweils eine VC-Dimension von 2. Für weitere Details zur VC-Dimension sei auf [BEHW89] und [VC71] verwiesen.

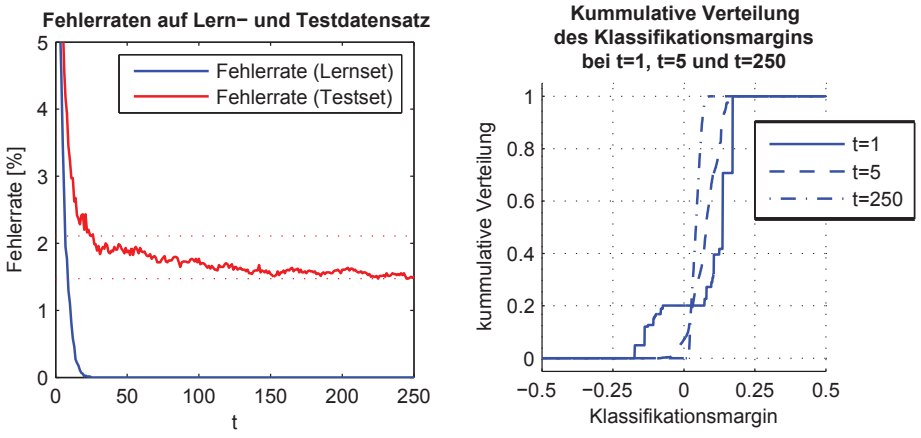


Abbildung 3.14.: Fehlerraten und kummulative Verteilung des Klassifikationsmargins auf dem *twonorm*-Datensatz (vgl. Abbildung 3.12) aus [Bre98]. *Links:* obwohl der Trainingsfehler (blaue, untere Kurve) bereits in Runde $t = 25$ verschwindet, sinkt der Fehler auf einem Testdatensatz mit zunehmender Rundenzahl t weiter (grüne, obere Kurve). Dies lässt auf gute Generalisierungseigenschaften von AdaBoost schließen. *Rechts:* kummulative Vereilung des Klassifikationsmargins (Visualisierung angelehnt an [SFBL98]).

beschränkt durch

$$\mathbb{P}\left(\rho \leq 0\right) \leq \mathbb{P}_{\text{emp}}\left(\rho \leq \hat{\rho}\right) + \mathcal{O}\left(\sqrt{\frac{1}{N}\left(\frac{d \ln^2 \frac{N}{d}}{\hat{\rho}^2} + \ln \frac{N}{\delta}\right)}\right). \tag{3.14}$$

Bemerkenswert ist, dass diese obere Schranke unabhängig von der Rundenzahl T und damit unabhängig von der Anzahl der Weaklearner ist. Da AdaBoost in jeder Runde die Kostenfunktion $J\left(\rho\right)$ von (3.9) minimiert und damit den Margin ρ maximiert, kann der Generalisierungsfehler auch dann noch sinken, wenn der Trainingsfehler schon null ist (vgl. 1. Term in (3.14)). Dieser Effekt lässt sich empirisch auch in Abbildung 3.14 beobachten. Auch nachdem der Trainingsfehler bereits null ist, wird durch Boosting in den weiteren Runden der Margin der Trainingsbeispiele größer und der Fehler auf dem Testset kleiner.

Allerdings darf nicht unerwähnt bleiben, dass sich dieses Verhalten bei stark verrauschten Daten (insbesondere bei Daten mit ungenauen oder gar vertauschten Klassenlabels) ins Gegenteil wendet - am eindrucksvollsten belegt durch die Untersuchungen von [Die00]. Interessanterweise ist dieses Verhalten ebenso aus der Tatsache begründet, dass AdaBoost versucht, den Margin zu maximieren. In erster Linie konzentriert sich AdaBoost nämlich auf die Beispiele mit dem kleinsten Margin und weist damit den Ausreißern ein unverhältnismäßig großes Gewicht zu [ORM98]. Eine mögliche Abhilfe bieten regularisierte Varianten von AdaBoost, z.B. [ROM01] oder auch [STL06].

In dieser Arbeit wird auf Regularisierung verzichtet, da dieser Effekt erst bei Klassifikatoren mit deutlich mehr Weaklearnern auftritt. Innerhalb der Kaskade werden pro Stufe aber die Anzahl der Merkmale - und damit die Anzahl der Weaklearner - beschränkt,

um den Rechenaufwand bei der Klassifikation möglichst gering zu halten. Das Problem der Überadaption lässt sich jedoch auch in dieser Arbeit nicht umgehen, allerdings aus einem anderen Grund: Vor allem in den letzten Kaskadenstufen stehen bei begrenzter Trainingsmenge nur noch wenige Beispiele zum Training zur Verfügung, so dass der zweite Term in (3.14) immer größer und damit die Generalisierungseigenschaft immer schlechter wird.

3.4. Training und Anwendung kaskadierter Klassifikatoren

Der in den Abschnitten 3.2 und 3.3 dargestellte AdaBoost-Algorithmus beschreibt das Training einer einzelnen Stufe des kaskadierten Klassifikators aus Abbildung 3.5. Die ersten Stufen sollen dabei mit möglichst wenig Weaklearnern auskommen, um mit geringem Aufwand einen Großteil der Objekthypothesen auszusortieren. Die jeweils nachfolgenden Klassifikatoren müssen dann noch die jeweils verbleibenden Hypothesen betrachten. Die Klassifikatoren jeder Stufe können so immer komplexer werden, da die Anzahl der zu untersuchenden Hypothesen mit jeder Stufe abnimmt und damit der Gesamtaufwand niedrig bleibt. Dabei geht man natürlich davon aus, dass die Vordergrundklasse „Fußgänger“ bezogen auf die Gesamtanzahl aller möglichen Hypothesen im Bild nur sehr selten auftritt und nur sehr wenige Hypothesen die aufwändigen letzten Kaskadenstufen erreichen.

Für das Training einer ganzen Kaskade ergeben sich folgende Rahmenbedingungen:

- Zur Reduktion des Gesamtaufwandes in der Anwendung müssen die ersten Kaskadenstufen auf nur wenige Weaklearner beschränkt werden. Die Anzahl der Weaklearner pro Stufe kann mit zunehmender Stufenzahl steigen.
- Um eine hohe Detektionsrate der gesamten Kaskade sicher zu stellen, muss die Detektionsrate jeder einzelnen Stufe auch bei beschränkter Weaklearnerzahl sehr hoch sein.
- Jede Stufe darf auch nur mit den Beispielen trainiert werden, die diese auch erreichen (Bootstrapping, [Efr79]).

Jede der Einzelstufen einer Kaskade der Länge K ist ein eigenständiger Klassifikator und kann daher eine unterschiedliche Detektionsrate $D_k, k = 1, \ldots, K$ aufweisen. Dann ist die Detektionsrate der gesamten Kaskade

$$D = \prod_{k=1}^{K} D_k. \tag{3.15}$$

Unter Detektionsrate versteht man dabei den Anteil der dem Klassifikator präsentierten Fußgänger, die vom Klassifikator auch richtigerweise als Fußgänger klassifiziert werden. Demgegenüber ist die Falschalarmrate der Anteil der dem Klassifikator präsentierten

Hypothesen, die vom Klassifikator fälschlicherweise als Fußgänger angesehen werden. Für die Falschalarmrate F einer Kaskade gilt analog

$$F = \prod_{k=1}^{K} F_k,$$

wobei F_k jeweils die Falschalarmrate der Stufe k bezeichnet.

Wird beispielsweise eine Detektionsrate von $D^* = 0.95$ angestrebt, muss jede Stufe einer Kaskade der Länge $K = 10$ eine Detektionsrate von etwa $D_k^* = 0.995$ erreichen. Dies erscheint auf den ersten Blick als unerreichbar hoch, allerdings reicht bereits $F_k^* = 0.25, k = 1, \ldots, K$, um eine Falschalarmrate von $F^* = 1 \cdot 10^{-6}$ für die gesamte Kaskade sicherzustellen.

Der AdaBoost-Algorithmus aus Abschnitt 3.2 sieht zunächst nicht vor, zielgerichtet vorgegebene Detektionsraten einzuhalten. In erster Linie ist AdaBoost nämlich ein Verfahren zur Maximierung des Margins (siehe Kapitel 3.3). Die Ausgabe des Klassifikators ist dabei das Vorzeichen der Aktivierung (Schritt 3 in Algorithmus 3.1). Hier bietet sich jedoch die Möglichkeit, die Schwelle dieses Stronglearner-Perzeptrons zu verschieben: Bei einer höheren Schwelle entsteht ein Klassifikator mit niedrigerer Falschalarmrate, dafür aber auch niedrigerer Detektionsrate. Eine niedrigere Schwelle resultiert in einen Klassifikator mit hoher Detektionsrate, aber auch höherer Falschalarmrate. Um eine vorgegebene Detektionsrate der Kaskadenstufe einzuhalten schlagen [VJ04] vor, im Training einer Stufe k nach jedem von AdaBoost ausgewählten Weaklearner die Schwelle des Stronglearners zu verkleinern, bis die geforderte Detektionsrate D_k^* sichergestellt ist. Ist die Falschalarmrate dann noch zu groß, werden in weiteren Runden von AdaBoost weitere Weaklearner ausgewählt. Das Training der Stufe endet, sobald die voreingestellte Falschalarmrate F_k^* nach Verschieben der Schwelle unterschritten wird oder die maximale Anzahl $T_k^{\max}$ von Weaklearnern für diese Stufe erreicht werden.

Die in Kapitel 3.3 diskutierten theoretischen Eigenschaften beim Training durch AdaBoost bleiben erhalten - das Training selbst wird nach wie vor mit einer Stronglearnerschwelle $\Theta = 0$ durchgeführt. Die Schwellen werden erst nach dem Training verschoben. Allerdings ändert sich nach dem Verschieben der Schwelle die Klassifikationsentscheidung und damit auch der gesamte Trainingsfehler $\text{err}_{\text{Training}} := \frac{1}{N} \left| \left\{ x^{(i)} \left| H\left(x^{(i)}\right) \neq y^{(i)} \right\} \right|$. Da die Motivation der Berechnung der Rückschlusswahrscheinlichkeit $p(y = 1|x)$ unter anderem auf Basis der oberen Schranke dieses Trainingsfehlers beruht, muss diese für den Fall der verschobenen Schwelle nochmals verifiziert werden (siehe Abschnitt 3.5).

Nach dem Training einer Kaskadenstufe werden für das Training der nachfolgenden Stufe in einem Bootstrapping-Schritt neue Trainingsbeispiele erzeugt, indem auf alle verfügbaren Beispiele die bereits bestehende Teilkaskade angewandt wird. Jede Stufe wird damit nur mit den Beispielen trainiert, die diese auch erreichen. Das Training stoppt, sobald aus dem verfügbaren Datenmaterial nicht mehr genügend Negativbeispiele die zu trainierende Stufe erreichen.

In diesem Bootstrapping-Schritt müssen vor allem in späteren Kaskadenstufen unter Umständen sehr viele Negativbeispiele überprüft werden, bis die geforderte Menge

an Beispielen zum Training der entsprechenden Stufe zur Verfügng steht. Ein Großteil der Hypothesen wird ja in den frühen Kaskadenstufen bereits aussortiert. Der Boostrapping-Schritt nimmt im Training sogar die meiste Zeit und Ressourcen in Anspruch (in Extremfällen mehrere Tage pro Kaskadenstufe). Aus diesem Grund werden die Detektions- und Falschalarmraten zur Verschiebung der Stronglearnerschwellen in dieser Arbeit nicht wie in [VJ04] vorgeschlagen anhand eines vom Trainingsdatensatz unabhängigen Testdatensatz, sondern anhand des Trainingsdatensatzes selbst bestimmt. Andernfalls müsste der langwierige Bootstrapping-Schritt auch für den Testdatensatz durchgeführt werden. Die vorgegebenen, zu erreichenden Detektionsraten D_k^* bzw. Falschalarmraten $F_k^*, k = 1, \ldots, K_{\max}$ sind in dieser Arbeit also immer im Kontext des Trainingsdatensatzes zu interpretieren (und entsprechend zu wählen).

Darüber hinaus wird die Anzahl von Weaklearnern in jeder Stufe durch $T_k^{\max}$ beschränkt. So kann sichergestellt werden, dass der Performance-Vorteil von wenigen Weaklearnern gerade in den ersten Kaskadenstufen nicht durch unrealistische Anforderungen an die Falschalarmraten verloren geht.

Der gesamte Ablauf ist in Abbildung 3.15 nochmals zusammengefasst. Die Funktionsweisen und Parametrisierungen unterschiedlicher Hypothesengeneratoren werden in Kapitel 4 ausführlich beschrieben. Wie in der Anwendung zur Detektion von Fußgängern erzeugt der Hypothesengenerator potentielle Hypothesen in Form von Suchfenstern im Bild, die vom bereits trainierten Kaskadenteil klassifiziert werden.

Zusammengefasst sind die Trainingsparameter zum Training einer Kaskade damit:

- $K_{\max}$: maximale Anzahl von Kaskadenstufen, die trainiert werden sollen.
- $D_k^*, k = 1, \ldots, K_{\max}$: vorgegebene minimale Detektionsraten auf dem Lernset.
- $F_k^*, k = 1, \ldots, K_{\max}$: vorgegebene maximale Falschalarmraten auf dem Lernset.
- $T_k^{\max}, k = 1, \ldots, K_{\max}$: maximale Anzahl von Weaklearnern für die Stufe k.
- $N_{\text{pos}}^*, N_{\text{neg}}^*$: Anzahl der positiven und negativen Beispiele, mit der jede Stufe trainiert werden soll (Bootstrapping).

Die Anwendung der Kaskade zur Detektion von Fußgängern ist bereits auch Teil des Trainings (oberer blauer Block in Abbildung 3.15). Ein Hypothesengenerator erzeugt in jedem Bild eine Menge von Hypothesen als Eingabe der Kaskade. Wird die Hypothese dann von einer der Kaskadenstufen verworfen, wird sie als Hintegrundbeispiel klassifiziert. Wird sie nie verworfen und erreicht eine bestimmte Stufe (Detektionsstufe), wird sie als Fußgänger klassifiziert. Die Detektionsstufe muss dabei nicht notwendigerweise die letzte Stufe der Kaskade darstellen. Indem Stufen am Ende der Kaskade an- oder abgeschaltet werden, kann der Arbeitspunkt der Kaskade grob justiert werden. Schaltet man Stufen ab, steigt die Detektionsrate (und mit ihr die Falschalarmrate). Fügt man Stufen hinzu, können mehr Falschalarme aussortiert und damit vermieden werden. Damit sinkt jedoch auch die Detektionsrate, da jede hinzugefügte Stufe in der Regel eine Detektionsrate $D_k < 1$ aufweist und multiplikativ in die gesamte Detektionsrate mit einfließt (siehe Gleichung (3.15)).

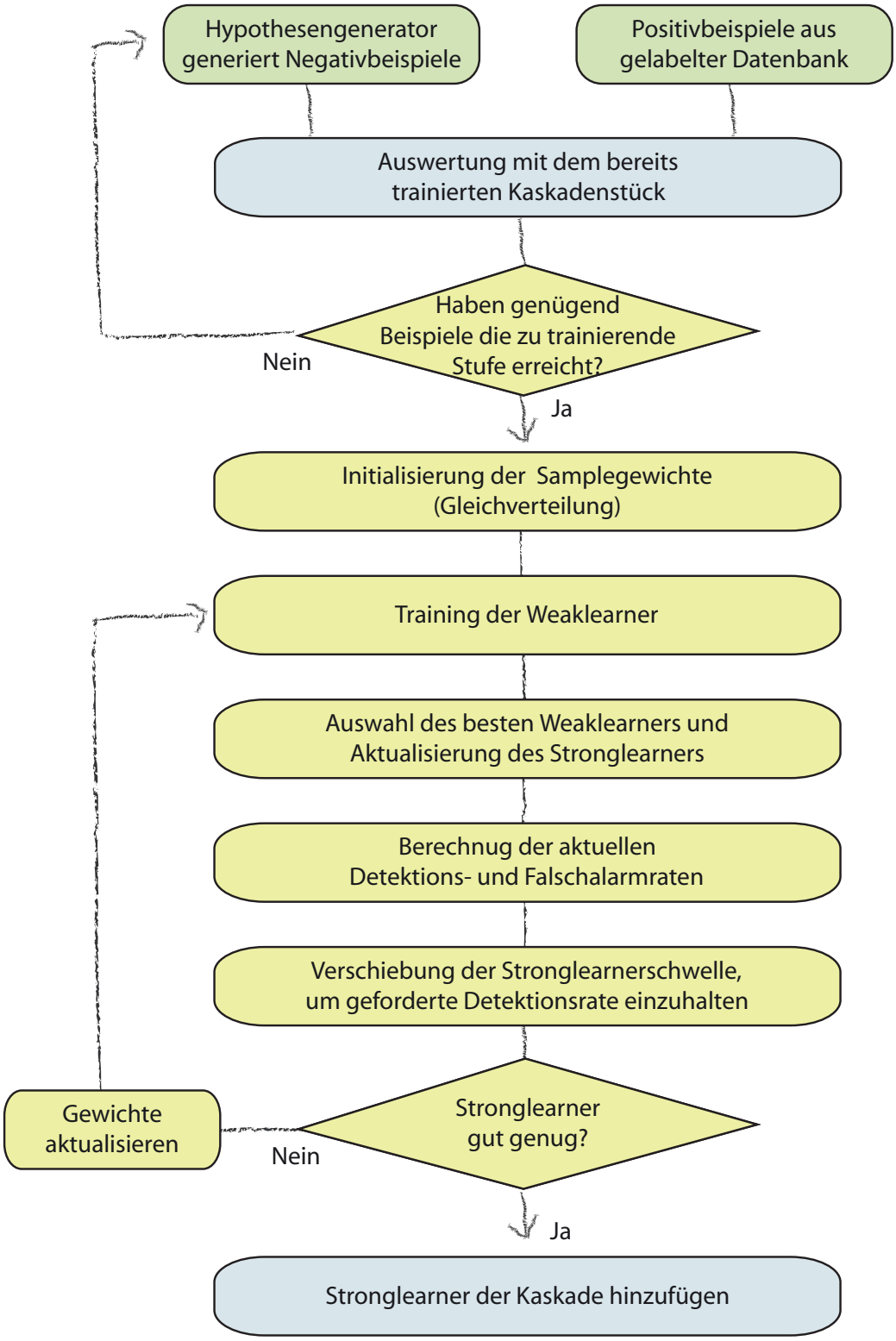


Abbildung 3.15.: Ablauf des Trainings einer Kaskadenstufe.

Die Einstellung dieses Arbeitspunktes ist natürlich nur in sehr grober Granularität möglich - es können lediglich ganze Stufen an- oder abgeschaltet werden. Nur mit der Verfügbarkeit von Rückschlusswahrscheinlichkeiten für jede Hypothese ist man in der Lage die Detetektionsrate über einen Schwellenvergleich direkt einzustellen.

3.5. Rückschlusswahrscheinlichkeiten

In einem Kaskadenklassifikator wird eine Hypothese entweder in einer der Stufen verworfen oder sie passiert die letzte Stufe und wird damit der Positivklasse zugeordnet. Der Klassifikator trifft damit eine binäre Entscheidung, und ist damit im Zusammenhang mit probabilistischen Verfahren - wie z.B. den in Kapitel 5 beschriebenen Partikelfiltern - nicht einsetzbar.

Im Folgenden wird deshalb ein mathematisches Modell zur Berechnung von Rückschlusswahrscheinlichkeiten bei der Klassifikation durch kaskadierte Klassifikatoren entwickelt. Ausgangspunkt sind die in [Tu05] beschriebenen probabilistischen Boosting-Bäume (engl. „Probabilistic Boosting Trees“, kurz PBT), die in gewisser Weise eine Verallgemeinerung von Kaskadenklassifikatoren darstellen. Wie die Stufen einer Kaskade bestehen die Knoten eines solchen Baumes aus Klassifikatoren, die mit AdaBoost trainiert werden, allerdings jeweils mit einer Stronglearnerschwelle von $\Theta = 0$. Dadurch können in PBTs die in Kapitel 3.3 hergeleiteten Rückschlusswahrscheinlichkeiten für AdaBoost-Klassifikatoren direkt verwendt werden.

Der Aufbau eines PBTs ist in Abbildung 3.16 schematisch dargestellt. Um Verwechslungen zu vermeiden, werden im Folgenden die Rückschlusswahrscheinlichkeiten des gesamten PBTs (bzw. der Kaskade) mit $p(y|x)$ bezeichnet, die der einzelnen mit AdaBoost trainierten Klassifikatoren jeweils mit $q(y|x)$. Für letztere gilt entsprechend Satz 5 in Abschnitt 3.3 (für $\Theta = 0$):

$$q(y = +1|x) = \frac{\exp\{2A(x)\}}{1 + \exp\{2A(x)\}}. \tag{3.16}$$

Im Training wird in jedem Knoten des binären Baumes die Hypothesenmenge auf Basis der Rückschlusswahrscheinlichkeit $q(y = +1|x)$ des entsprechenden AdaBoost-Klassifikators in zwei Untermengen aufgeteilt: Hypothesen mit $q(y = +1|x) < 0.5 - \kappa$, $\kappa > 0$, werden an den linken Teilbaum, Hypothesen mit $q(y = +1|x) > 0.5 + \kappa$ an den rechten Teilbaum und Hypothesen mit $0.5 - \kappa \leq q(y = +1|x) \leq 0.5 + \kappa$ werden an beide Teilbäume weitergegeben. Das Training erfolgt rekursiv bis der Baum bis zu einer vorgegebenen Baumtiefe K expandiert ist. Die Konstante κ realisiert hier auch eine Schwellwertverschiebung des Stronglearners, doch wird κ sehr klein gewählt und soll einer Überadaption entgegenwirken [Tu05]. Außerdem ist für die Anwendung des fertigen Baums $\kappa = 0$.

Zur Klassifikation eines Beispiels x wird der Baum in derselben Weise durchlaufen. In jedem Knoten wird dazu die Rückschlusswahrscheinlichkeit $q(y = +1|x)$ bestimmt.

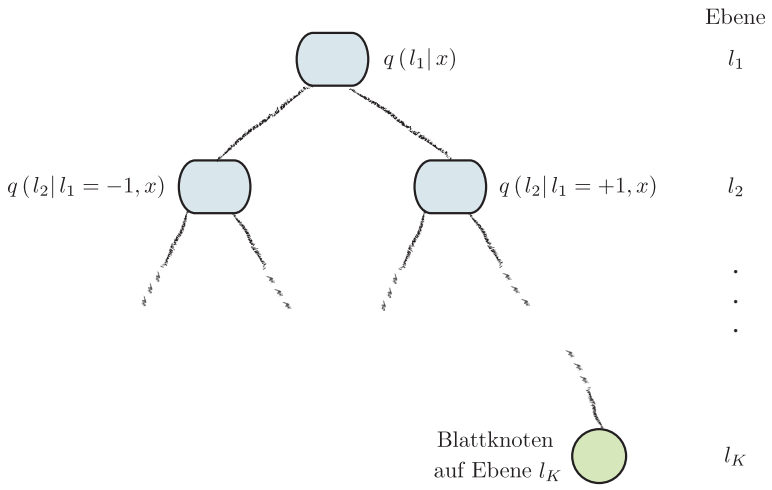


Abbildung 3.16.: Struktur eines probabilistischen Boosting-Baums. Jeder Knoten des binären Baumes stellt einen Klassifikator dar, der mit AdaBoost trainiert ist. Anhand der Rückschlusswahrscheinlichkeiten wird entschieden, ob eine Hypothese an den rechten oder linken Teilbaum weitergeleitet wird.

[Tu05] benutzt dann das Prinzip der totalen Wahrscheinlichkeit um eine Rückschlusswahrscheinlichkeit $p(y|x)$ für das Beispiel x anhand der Ergebnisse aller durchlaufenden Knoten im Baum anzunähern. Formal ergibt sich $p(y|x)$ aus der Rekursion

$$\begin{aligned} p(y|x) &= \sum_{l_1} q(l_1|x) \cdot p(y|l_1, x) \\ &= \sum_{l_1} q(l_1|x) \cdot \left(\sum_{l_2} q(l_2|l_1, x) \cdot p(y|l_2, l_1, x) \right) \\ &= \sum_{l_1, l_2} q(l_1|x) \cdot q(l_2|l_1, x) \cdot p(y|l_2, l_1, x) \\ &= \dots \\ &= \sum_{l_1, \dots, l_K} q(l_1|x) \cdot q(l_2|l_1, x) \cdot \dots \cdot q(l_K|l_{n-1}, \dots, l_1, x) \cdot p(y|l_K, \dots, l_1, x). \end{aligned} \tag{3.17}$$

$l_k \in \{-1, +1\}, k = 1, \dots, K$ steht dabei für den Zweig in der Baumtiefe k und $q(l_k|l_{k-1}, \dots, l_1, x)$ ist die Rückschlusswahrscheinlichkeit (3.16) des zugehörigen AdaBoost-Klassifikators.

In den Blättern des Baumes stoppt die Rekursion. Für $p(y|l_K, \dots, l_1, x)$ werden dann jeweils empirische Werte

$$p(y|l_K, \dots, l_1, x) := p_{\text{emp}}(y|l_K, \dots, l_1)$$

angenommen, die in [Tu05] während des Trainings bestimmt werden. Sie geben den gewichteten Anteil der Beispiele wieder, die den entsprechenden Blattknoten erreichen:

$$p_{\text{emp}}(y|l_K, \dots, l_1) = \sum_j d^{(j)}_{T_K+1} \delta(y_j = y).$$

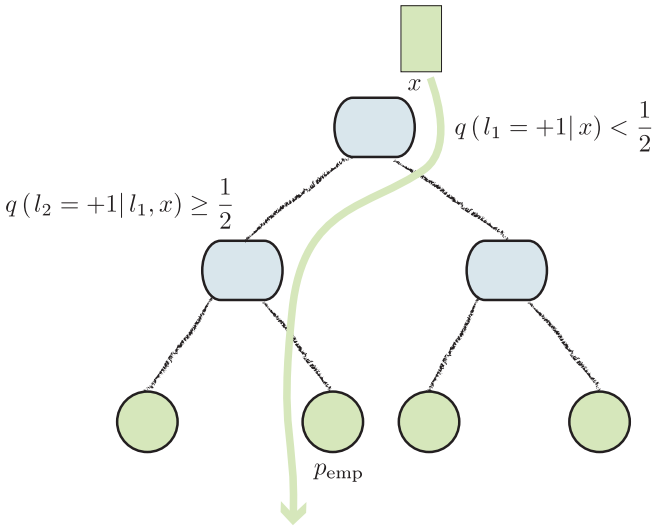


Abbildung 3.17.: Anwendung des probabilitischen Boosting-Baums. Eine Hypothese durchläuft den Baum, indem in jedem Knoten nach Anwendung des Klassifikators anhand der Rückschlusswahrscheinlichkeit entschieden wird, ob die Hypothese an den linken Teilbaum ($q(y = +1|x) < \frac{1}{2}$) oder an den rechten Teilbaum $q(y = +1|x) \geq \frac{1}{2}$ weitergegeben wird.

Abbildung 3.17 illustriert die Berechnung der Rückschlusswahrscheinlichkeit. Theoretisch müssen dabei alle Knoten im Baum ausgewertet werden. Um den Rechenaufwand zu vermindern können alternativ auch nur die Knoten berücksichtigt werden, die das Beispiel auf dem Weg zu einem der Blattknoten durchläuft. Der Beitrag der ignorierten Teilbäume muss dann durch empirische Wahrscheinlichkeiten entsprechend derer der Blattknoten abgeschätzt werden. Es sei außerdem nochmals angemerkt, dass bei der Anwendung des PBTs $\kappa = 0$ ist, d.h. die Entscheidung, ob ein Beispiel an den linken oder rechten Teilbaum weitergegeben wird basiert lediglich auf dem Vergleich der einzelnen Rückschlusswahrscheinlichkeiten mit 0.5. Das ist gleichbedeutend mit einem Vergleich der Aktivierung $A(x)$ mit $\Theta = 0$ (vgl. (3.16)).

Die Bestimmung von Rückschlusswahrscheinlichkeiten im PBT lässt sich nicht eins zu eins auf Kaskadenklassifikatoren übertragen. Zwar kann ein Kaskadenklassifikator als degenerierter PBT formuliert werden (siehe Abbildung 3.18), die Berechnungen unterscheiden sich allerdings in zwei wesentlichen Eigenschaften:

1. Kaskaden sind keine vollständigen binären Bäume (es gibt keine linken Teilbäume).
2. Nach dem Training einer Stufe wird die Schwelle Θ des Stronglearners so angepasst, dass eine bestimmte Detektionsrate sicher gestellt ist. In der Regel ist also $\Theta \neq 0$.

Die erste Eigenschaft stellt keine wesentliche Einschränkung dar: analog zum PBT können die fehlenden Teilbäume durch empirische Wahrscheinlichkeiten angenähert werden. Diese enstehen in einem Evaluationsschritt nach dem Training und geben die

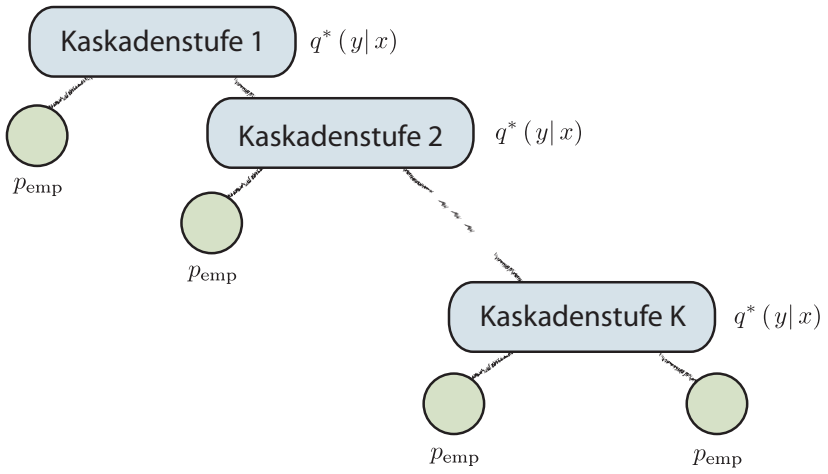


Abbildung 3.18.: Kaskade als degenerierter probabilistischer Boosting-Baum. Eine Kaskade kann auch als Baum aufgefasst werden, mit jeweils nur einem Kindknoten für jeden Klassifikator. Die im Vergleich zum klassischen probabilistischen Boosting-Baum fehlenden linken Teilbäume werden dabei durch empirische Wahrscheinlichkeiten angenähert.

Verteilung der Beispiele aus einem unabhängigen Testset (im folgenden Validierungsmenge) wieder, die in der entsprechenden Stufe als Hintergrund verworfen wurden.

Bei kritischer Betrachtung stellt sich dennoch die Frage, warum es nicht ausreicht, für jedes Beispiel ausschließlich die Rückschlusswahrscheinlichkeiten des jeweils letzten, erreichten Stronglearners zu betrachten. Jede Stufe der Kaskade wird nur mit den Beispielen trainiert, die diese auch erreichen. Die Problemstellung wird dadurch für jede Stufe immer spezieller, da nur noch Trainingsbeispiele aus den Randbereichen der Entscheidungsgrenze berücksichtigt werden (vgl. Abbildung 3.19). Die entsprechende Verteilung von $p(y = 1|x)$ modelliert daher nur einen Ausschnitt der gesamten Wahrscheinlichkeitsverteilung. Es ist also in diesem Fall besser, auch die Wahrscheinlichkeiten aus der gesamten, durchlaufenen Kaskade bei der Bestimmung von Rückschlusswahrscheinlichkeiten mit einzubeziehen.

Die zweite Eigenschaft wirkt sich allerdings direkt auf die Berechnung der Rückschlusswahrscheinlichkeiten aus. Die Rückschlusswahrscheinlichkeit $q(y|x)$ eines mit AdaBoost trainierten Klassifikators wurde in Lemma 5 auf Basis der oberen Schranke aus Satz 1 hergeleitet. Für den Fall verschobener Schwellen, d.h. $\Theta \neq 0$, gilt für die obere Schranke des Trainingsfehlers:

Satz 6. *Der Trainingsfehler* $\text{err}_{\text{Training}}$ *eines AdaBoost-Klassifikators mit adaptierter Schwelle* $\Theta \neq 0$ *ist beschränkt durch*

$$\text{err}_{\text{Training}} := \frac{1}{N} \left| \left\{ x^{(i)} \left| H\left(x^{(i)}\right) \neq y^{(i)} \right\} \right| \leq \frac{1}{N} \sum_i \exp\left\{ -y^{(i)} \left(A\left(x^{(i)}\right) - \Theta \right) \right\}.$$

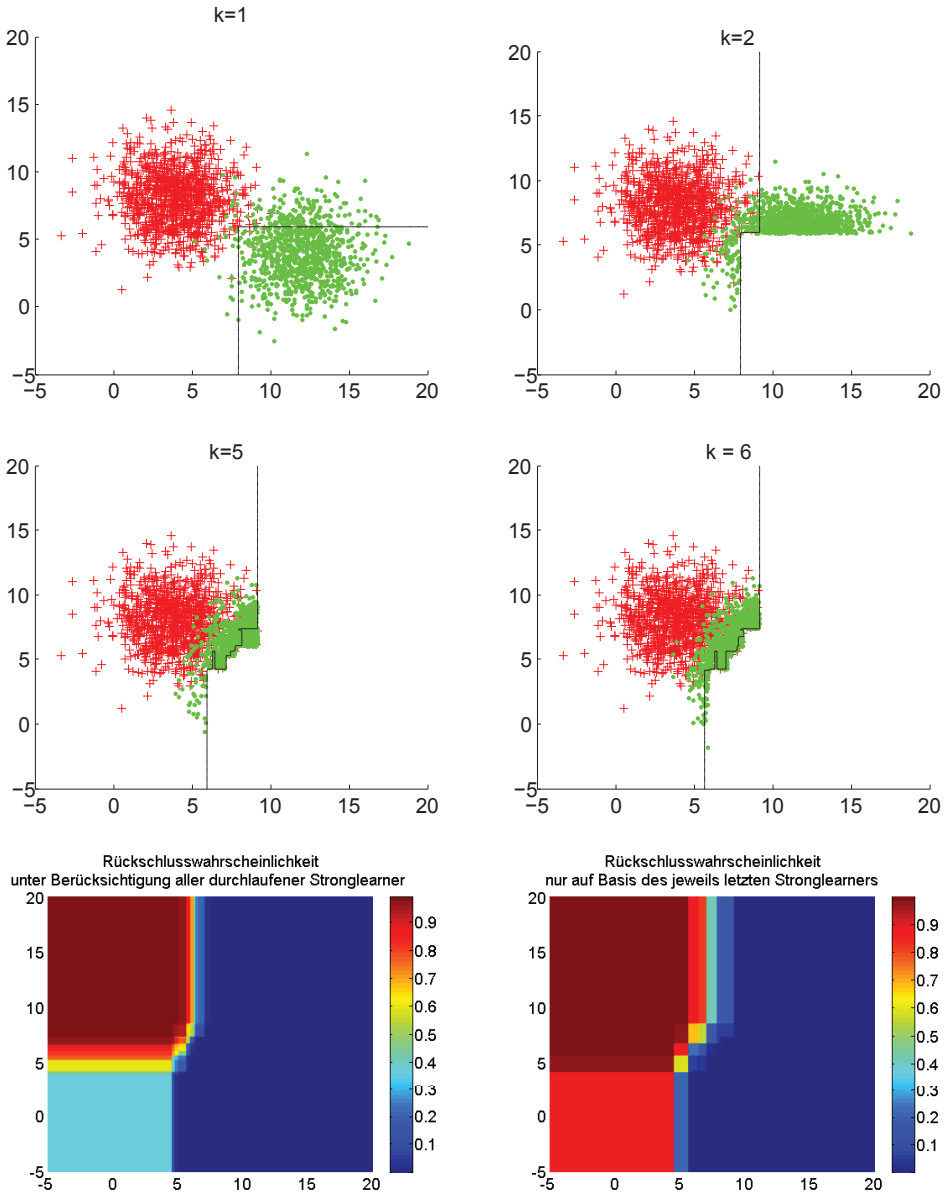


Abbildung 3.19.: Rückschlusswahrscheinlichkeiten am Beispiel künstlicher Daten. Dargestellt sind die Beispiele, die zum Training der Kaskadenstufe $k = 1, 2, 5, 6$ verwendet wurden. Die durchgezogene Linie stellt jeweils die Entscheidungsgrenze nach dem Training dieser Stufe dar. Die Positivbeispiele (rot) stammen dabei aus einer Normalverteilung mit $\mu = (4, 8)^{\mathrm{T}}$ mit $\Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}^{\mathrm{T}}$, die Negativbeispiele (grün) von einer Normalverteilung mit $\mu = (12, 4)^{\mathrm{T}}$ mit $\Sigma = (2, 2)^{\mathrm{T}}$. Links unten ist die Verteilung der Rückschlusswahrscheinlichkeit nach Gleichung (3.17), rechts unten nur auf Basis der Rückschlusswahrscheinlichkeit (3.16) des jeweils letzten, erreichten Stronglearners.

Der Beweis findet sich in Anhang A, Seite 177.

In Analogie zu Lemma 5 lässt sich dann eine Rückschlusswahrscheinlichkeit durch die Minimierung des Erwartungswertes

$$\mathbb{E}\left[\exp\left\{-y\left(A\left(x\right)-\Theta\right)\right\}\right]$$

herleiten:

Satz 7. *$J^*\left(A\left(x\right),\Theta\right) := \mathbb{E}\left[\exp\left\{-y\left(A\left(x\right)-\Theta\right)\right\}\right]$ nimmt sein Minimum an bei*

$$A\left(x\right) = \frac{1}{2}\ln\frac{p(y=1|x)}{p(y=-1|x)} + \Theta.$$

Damit gilt:

$$p(y=1|x) = \frac{\exp\left\{2A\left(x\right)-\Theta\right\}}{1+\exp\left\{2A\left(x\right)-\Theta\right\}}$$

und

$$p(y=-1|x) = \frac{\exp\left\{-2A\left(x\right)-\Theta\right\}}{1+\exp\left\{-2A\left(x\right)-\Theta\right\}}.$$

Der Beweis findet sich in Anhang A, Seite 178.

Zur praktischen Plausibilisierung sind in Abbildung 3.20 die Histogramme der Rückschlusswahrscheinlichkeiten einer Teststichporbe aus dem *twonorm*-Datensatz[3] von [Bre98] dargestellt. Dazu wurde der Datensatz in drei disjunkte Teilmengen, nämlich einer Trainingsmenge, einer Validierungsmenge zur Bestimmung der empirischen Wahrscheinlichkeiten und einer Teststichprobe aufgeteilt. Das Training der Kaskade erfolgte mit Entscheidungsbäumen (CARTs, siehe [BFSO84]) als Weaklearner und jeweils vorgegebener minimaler Detektionsrate $D_k^* = 99.5\%$, maximaler Falschalarmrate von $F_k^* = 20.0\%$ und maximal $T_k^{\max} = 20, k = 1,\ldots,4$ Weaklearnern pro Stufe. Die ermittelten empirischen Wahrscheinlichkeiten sind in Abbildung 3.21 aufgelistet. Abbildung 3.22 schließlich zeigt die Detektions- und Falschalarmrate der Kaskade für unterschiedliche Entscheidungsschwellen auf Basis der Rückschlusswahrscheinlichkeiten.

[3] 20-dimensionaler, 2-Klassen-Datensatz, beschrieben in [Bre98]. Die Beispiele der ersten Klasse stammen von einer Normalverteilung um den Mittelwert $(a, a, \ldots, a)$, die der zweiten Klasse aus einer Normalverteilung um den Mittelwert $(-a, -a, \ldots, -a)$, jeweils mit $a = \frac{2}{\sqrt{20}}$ und der Einheitsmatrix als Kovarianzmatrix. Siehe auch Abbildung 3.12, Seite 63 und Abbildung 3.14, Seite 67.

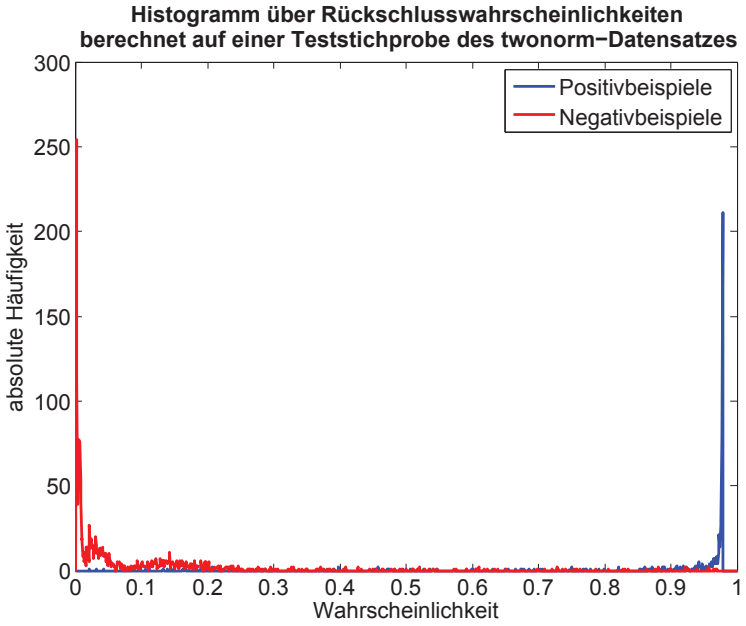


Abbildung 3.20.: Histogramme der Rückschlusswahrscheinlichkeiten der Beispiele aus einer Teststichprobe aus dem *twonorm*-Datensatz. Das Histogramm der Wahrscheinlichkeiten der Positivbeispiele ist blau (rechts), das der Negativbeispiele rot (links) dargestellt. Die Trennung beider Klassen ist deutlich ausgeprägt.

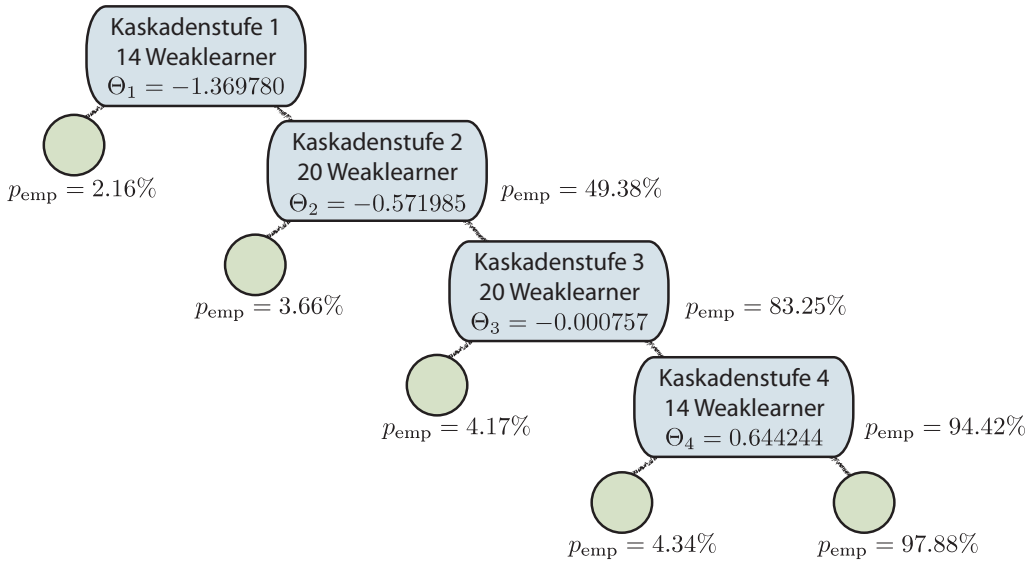


Abbildung 3.21.: Empirische Wahrscheinlichkeiten einer Kaskade. Das Training erfolgte auf einer Teilmenge des *twonorm*-Datensatzes. Die Bestimmung der empirischen Wahrscheinlichkeiten wurde dann auf einer von der Trainingsmenge disjunkten Validierungsmenge durchgeführt. Sie geben den Anteil der Positivbeispiele im entsprechenden Zweig wieder.

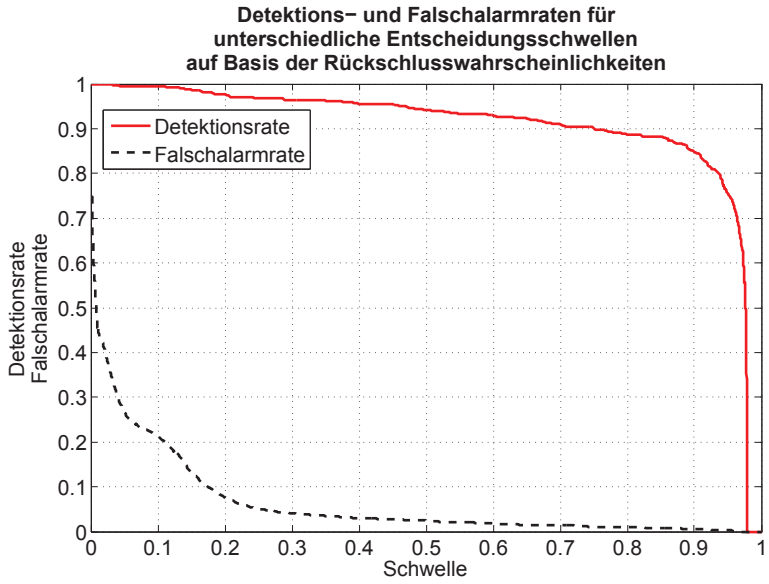


Abbildung 3.22.: Detektions- und Falschalarmraten für unterschiedliche Entscheidungsschwellen auf Basis der Rückschlusswahrscheinlichkeiten. Die Klassifikationsentscheidung auf Basis der Rückschlusswahrscheinlichkeiten kann durch den Vergleich mit einer Schwelle getroffen werden. Die Schwelle stellt dabei den Arbeitspunkt des Klassifikators ein. Bei einer hohen Entscheidungsschwelle sinkt die Falschalarmrate - allerdings sinkt auch die Detektionsrate. Bei niedriger Entscheidungsschwelle ist die Detektionsrate sehr hoch, allerdings werden dann auch mehr Falschalarme gemeldet.

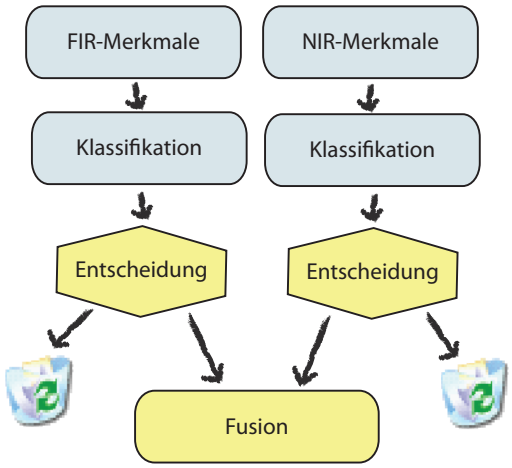


Abbildung 3.23.: Fusion auf Objektebene. Detektorsysteme, die jeweils nur auf Basis eines Sensors eine Objektbildung vornehmen werden in der Fusion miteinander kombiniert. Da die Fusion auf einer sehr späten Ebene stattfindet, kann diese die physikalischen Eigenschaften beider Sensoren nicht in optimaler Weise miteinander verknüpfen, da diese bei der Entscheidung nicht mehr zur Verfügung stehen.

3.6. Merkmalsbasierte Fusion mit AdaBoost

Prinzipiell kann eine Fusion mehrerer Bildsensoren zur Fußgängererkennung auf der Basis unterschiedlicher Konzepte erfolgen:

- Fusion auf Bildpunktebene (Fusion auf Rohdatenebene, frühe Fusion),
- Fusion auf Merkmalsebene (merkmalsbasierte Fusion),
- Fusion auf Objektebene (späte Fusion).

Eine Fusion auf Bildpunktebene setzt eine punktgenaue Zuordnung der Bildpunkte im FIR-Bild zu den Punkten im NIR-Bild voraus. Dieses fusionierte Bild kann dann z.B. als Eingabe für das in den vorangegangenen Abschnitten beschriebene Detektorsystem dienen. Der größte Vorteil eines solchen vollständig registrierten Bildes ist ein deutlich kleinerer Suchraum für die Klassifikation, da Mehrdeutigkeiten bei der Hypothesenerstellung vermieden werden (siehe auch Abschnitt 2.3 und Kapitel 4). Darüber hinaus können dann auch Merkmale definiert werden, die sich auf Basis mehrerer Sensoren gleichzeitig berechnen. Die hohe Texturinformation im NIR-Bild kann dann z.B. direkt mit der Wärmesignatur aus dem FIR-Bild gekoppelt werden. Aufgrund der unterschiedlichen Einbauorte und Einbauwinkel des FIR-Sensors und des NIR-Sensors, den unterschiedlichen Öffnungswinkel sowie den unterschiedlichen Auflösungen der Sensoren ist eine solche Zuordnung jedoch nicht praktikabel. Eine Fusion auf Bildpunktebene scheidet daher für den Anwendungsfall dieser Arbeit aus.

Eine Fusion auf Objektebene verknüpft auf unterschiedliche Weise mehrere Detektorsysteme, die jeweils nur auf Basis eines Sensors eine Objektbildung vornehmen. Das Grundprinzip ist in Abbildung 3.23 dargestellt. Aus den Einzelsensoren werden

Merkmale extrahiert, die dann unabhängig voneinander zur Klassifikation herangezogen werden. Dabei werden auf Basis der einzelnen Sensoren Entscheidungen getroffen, die jeweils die physikalischen Eigenschaften der anderen Sensoren unberücksichtigt lassen. Die eigentliche Fusion findet auf einer späteren Ebene statt und kann dabei nicht mehr die kompletten Informationen berücksichtigen. In der Regel sind solche Systeme hinsichtlich Detektionsrate und Reichweite nur wenig besser als die Einzelsysteme: Hat bei der Fusion von zwei Sensoren eines der Teilsysteme den Fußgänger nicht erkannt, kann das Gesamtsystem nur dann eine Detektion melden, wenn beide Teilsysteme entsprechende Existenzmaße mitteilen können. Nur wenn sich eines der Teilsysteme bei seiner Entscheidung sehr sicher ist, kann es das andere überstimmen. In der Regel werden mit solchen Systemen die Falschalarmrate deutlich gesenkt werden, da die Einzelsysteme sich in ihrer Detektionsenscheidung gegenseitig verifizieren. Deshalb werden solche Fusionsansätze häufig in sicherheitskritischen Anwendungen eingesetzt.

In dieser Arbeit sollen Detektionsrate und Reichweite des Systems gegenüber den Einzelsensorsystemen erhöht werden. Deshalb erfolgt die Fusion der beiden Sensoren bereits auf der Merkmalsebene. Alle verfügbaren Merkmale fließen dabei gleichzeitig in den Entscheidungsprozess mit ein. Dazu werden die Merkmale aller Sensoren in einer einzigen Merkmalsmenge M zusammengefasst, aus der im Training die jeweils besten Merkmale unabhängig vom zugehörigen Sensor ausgewählt werden. Die Architektur des gesamten Fusionssystems entspricht dabei der des Einzelsensorsystems: Ein Hypothesengenerator erzeugt Hypothesen, für die jeweils ein Kaskadenklassifikator entscheidet, ob es sich um einen Fußgänger handelt oder nicht. Eine Hypothese entspricht in diesem Fall aber nicht mehr einem einzelnen Suchfenster, sondern setzt sich aus je einem Suchfenster pro Sensorstrom zusammen. Die verwendeten Merkmale entsprechen in dieser Arbeit wieder den haarwaveletähnlichen Filtern aus Abschnitt 3.1, die nun jeweils auf jeden Sensor einzeln bezogen sind. Prinzipiell können für jeden Sensor auch komplett unterschiedliche Merkmalsarten zum Einsatz kommen.

Bei der Verwendung von haarwaveletähnlichen Filtern im Fusionssystem ist ein Merkmal nicht mehr nur durch den Merkmalstyp aus Abbildung 3.7, dessen Position und Skalierung im Basissuchfenster definiert, sondern zusätzlich noch durch einen Verweis auf den jeweiligen Sensor, auf dessen Daten der jeweilige Merkmalswert berechnet wird. Entsprechend wird auch aus dem einzelnen Basissuchfenster ein Basissuchfenstertupel.

Im Fall der Fusion eines FIR- und NIR-Sensors setzt sich nun die Merkmalsmenge aus dem überbestimmten Merkmalssatz M_{FIR} haarwaveletähnlicher Filter des FIR-Sensors, sowie dem überbestimmten Merkmalssatz M_{NIR} haarwaveletähnlicher Filter des NIR-Sensors zusammen:

$$M = M_{\mathrm{FIR}} \cup M_{\mathrm{NIR}}.$$

Die Weaklearnerstruktur bleibt ansonsten gleich:

$$h(x) = \begin{cases} +1 & p \cdot m(x) < p \cdot \theta \\ -1 & \text{sonst.} \end{cases}, \quad m \in M_{\mathrm{FIR}} \cup M_{\mathrm{NIR}}, \theta \in \mathbb{R}, p \in \{-1, +1\}.$$

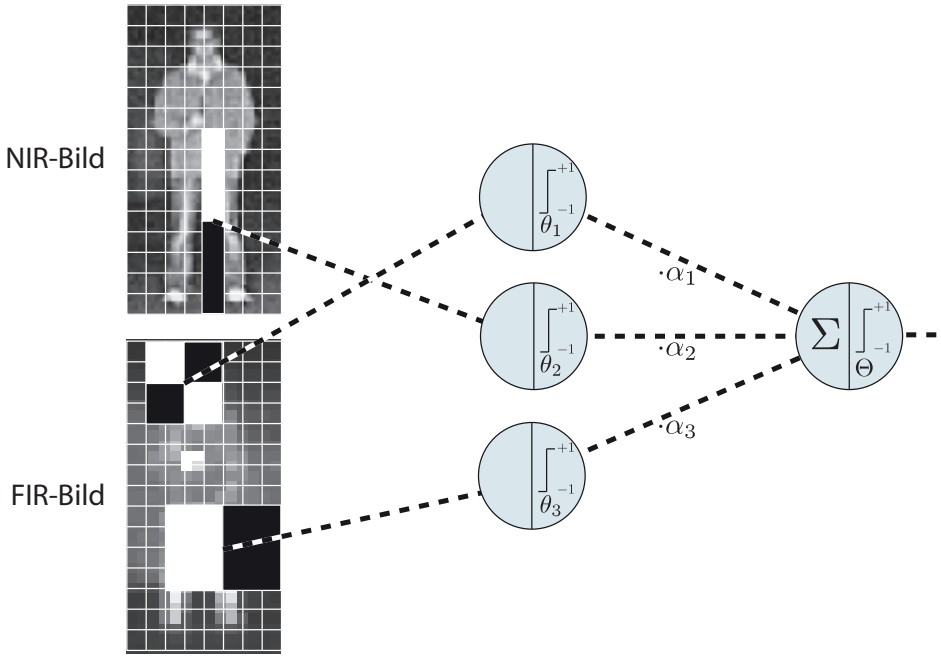


Abbildung 3.24.: Fusion auf Merkmalsebene. Der Aufbau des Klassifikators entspricht der normalen Stronglearnerstruktur aus Abbildung 3.10. Die Auswahl der Merkmale erfolgt im Training von AdaBoost auf Basis der vereinigten Merkmalsmenge der FIR- und NIR-Merkmale. Der fertige Klassifikator führt dadurch eine Fusion auf Merkmalsebene durch.

Entsprechend ändert sich auch nicht die Struktur der Stronglearner:

$$H(x) = \begin{cases} +1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \Theta \\ -1 & \text{sonst.} \end{cases}$$

Abbildung 3.24 zeigt eine Visualisierung der Stronglearnerstruktur für den Fall, dass AdaBoost im Training ein NIR-Merkmal und zwei FIR-Merkmale ausgewählt hat.

Da die Klassifikationsstruktur unverändert ist, gelten die Ergebnisse aus den vorangegangenen Abschnitten auch für den Fusionsfall. Damit stellt die vorgestellte Detektionsarchitektur eine flexibles Methode zur merkmalsbasierten Fusion dar, die auch leicht auf andere Anwendungsfälle adaptiert werden kann. So wird z.B. in [Ser11] das Konzept auf den Fall einer Fusion von Bild und bildgebendem Radar auf Merkmalsebene übertragen. Neben der Definition geeigneter Merkmale ist die Herausforderung auch hier eine geeignete Hypothesenbildung umzusetzen, die einen echtzeitfähigen Einsatz des Detektors möglich macht. Eine ausführliche Darstellung aller Aspekte der Fusion von Bild und bildgebendem Radar gibt die Arbeit von [Ser11].

Die Hypothesenbildung für den Fall der Fusion reiner Bildsensoren ist Gegenstand des nächsten Kapitels.

KAPITEL 4

Hypothesengenerierung

Zur Detektion von Fußgängern werden im Suchraum Hypothesen generiert, die dann mit dem Kaskadenklassifikator evaluiert werden. Je nach Ergebnis wird an Stelle der jeweiligen Hypothese eine Detektion gemeldet.

Der Hypothesengenerator ist damit ein wichtiger Bestandteil zur Lösung der Detektionsaufgabe: Detektionen können nur dann gemeldet werden, wenn zuvor entsprechende Hypothesen erzeugt wurden. Andererseits steigt der benötigte Rechenaufwand linear mit der Anzahl der Hypothesen. In der Regel wird deshalb die Hypothesenmenge mit Hilfe einer geeigneten Unterabtastung und zusätzlicher Suchbereichseinschränkungen reduziert. Ziel ist es, den Berechnungsaufwand so gering wie möglich zu halten, ohne dabei die Detektionsleistung zu beeinträchtigen.

Der Suchbereich kann z.B. leicht eingeschränkt werden, indem man annimmt, dass Fußgänger sich immer in derselben Ebene wie das Fahrzeug befinden. Um Robustheit gegenüber Kalibrationsfehlern oder Dynamikeinflüssen zu erreichen, sind jedoch zusätzliche Relaxationen solcher einfachen geometrischen Modelle notwendig. Diese, sowie die konkrete Abtaststrategie werden im nächsten Abschnitt 4.1 im Falle des einfachen Einzel-Sensor Hypothesengenerators präsentiert. Jede Hypothese wird dabei durch ein Suchfenster im Bild dargestellt. Der Einzel-Sensor Hypothesengenerator ist auch Ausgangspunkt für den Multi-Sensor Hypothesengenerator in Abschnitt 4.2. Anstatt einzelne Suchfenster umfasst in diesem Fall jede Hypothese mehrere Suchfenster, nämlich je ein Suchfenster pro Sensordatenstrom. Allerdings sind durch das Parallaxeproblem mehrere Zuordnungen eines Suchfensters des einen Sensordatenstroms zu einem aus dem Datenstrom des anderen Sensors möglich. Im Fusionsdetektor ist deshalb die Anzahl der Hypothesen gegenüber dem Einzel-Sensor Fall deutlich größer.

Um die Echtzeitfähigkeit des Detektors zu gewährleisten, werden in Abschnitt 4.3 Optimierungsstrategien vorgestellt, die spezielle Eigenschaften von Kaskadenklassifikatoren zur Reduktion des Suchaufwandes ausnutzen. Anstatt statisch mit einer festen Hypothesenmenge zu arbeiten, wird die Anzahl der Hypothesen über eine dynamische lokale Steuerung der Hypothesendichte effektiv reduziert.

Mit den probabilistischen Partikelfiltern schließlich werden in Kapitel 5 und Kapitel 6 die Hypothesen auch dynamisch über ganze Bildfolgen hinweg organisiert. Unter anderem wird dabei auch die Eigenbewegung des Fahrzeugs mit berücksichtigt. Über die in Abschnitt 3.5 hergeleiteten Rückschlusswahrscheinlichkeiten fließen die Ergebnisse des Kaskadenklassifikators aber direkt in den Partikelfilter mit ein. Über die Zeit hinweg fokussieren sich die Hypothesen damit automatisch in den Bereichen mit Fußgängern im Bild.

4.1. Einfacher Hypothesengenerator

Die einfachste Suchstrategie zum Finden von Objekten im Bild ist das pixelweise Abtasten des gesamten Bildes in allen möglichen Skalierungen. Das ergibt zum Beispiel im NIR-Sensor bei einer Bildgröße von $640\,\mathrm{px} \times 480\,\mathrm{px}$ und Skalierungen zwischen $h_0 = 10\,\mathrm{px}$ und $h_{\max} = 240\,\mathrm{px}$ eine Hypothesenmenge mit ca. 45 Millionen Elementen. Selbst mit den effizientesten Kaskadenklassifikatoren ist damit an eine Echtzeitanwendung nicht zu denken.

Tastet man das ganze Bild pixelweise ab, werden viele Hypothesen erzeugt, die nicht sinnvoll sind, z.B. kann man davon ausgehen, dass Fußgänger nicht plötzlich am Himmel erscheinen, Hypothesen direkt am oberen Bildrand machen daher wenig Sinn. Durch eine geeignete Modellierung der Welt kann daher die Zahl der Hypothesen deutlich eingeschränkt werden - ohne die Detektionsfähigkeit des gesamten Detektors einzuschränken. Darüber hinaus kann eine skalierungsabhängige Unterabtastung die Zahl der zu prüfenden Hypothesen weiter reduzieren. Im Folgenden wird in drei Schritten, ausgehend von einem stark vereinfachenden und im Grunde genommen unzureichenden Weltmodell, ein Modell entwickelt, das die Zahl der Hypothesen auf ca. 676 000 senkt.

Modell I, ebene Welt mit genormter Fußgängergröße

Die einfachste Modellannahme ist die Annahme einer ebenen Welt, d.h. die zu detektierenden Objekte und das Fahrzeug befinden sich auf gleicher Ebene (engl. Ground-Plane-Assumption). Nimmt man zusätzlich an, dass die Größe aller Objekte genormt ist (d.h. alle Fußgänger gleich groß sind), befinden sich in diesem stark vereinfachten Weltmodell alle Fußgänger in einem gleichmäßigen Korridor vor dem Fahrzeug (Abbildung 4.1). Dieses Modell ist natürlich sehr stark vereinfacht und unzureichend, z.B. sind Kinder mit sehr kleiner Körpergröße überhaupt nicht berücksichtigt.

Abbildung 4.1.: Suchkorridor im Ortsraum einer ebenen Welt. In einer ebenen Welt und unter der Voraussetzung einer einheitlichen Objektgröße, befinden sich alle Fußgänger in einem gleichmäßigem Korridor vor dem Fahrzeug.

Die Modellierung der Wirklichkeit als ebene Welt schränkt den Sichtbereich in horizontaler Richtung nicht ein, d.h. für den Spaltenwert col eines Objektfensters gibt es keinerlei Einschränkung. Wählt man col beliebig und hält diesen fest, so ist bei einer festen Skalierung h und der angenommenen Objektgröße H in Realweltkoordinaten der Zeilenwert row im Modell eindeutig festgelegt: row ergibt sich als Lösung des (nichtlinearen) Gleichungssystems

$$\begin{pmatrix} \alpha \cdot \mathrm{col} \\ \alpha \cdot \mathrm{row} \\ \alpha \end{pmatrix} = \boldsymbol{\mathcal{P}} \cdot \begin{pmatrix} X \\ Y \\ H \\ 1 \end{pmatrix}, \tag{4.1}$$

$$\begin{pmatrix} \beta \cdot \underset{\sim}{\mathrm{col}} \\ \beta \cdot (\mathrm{row} + h) \\ \beta \end{pmatrix} = \boldsymbol{\mathcal{P}} \cdot \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}, \tag{4.2}$$

mit den sechs Unbekannten row, $\underset{\sim}{\mathrm{col}}, X, Y, \alpha, \beta$. Gleichung (4.1) entspricht dabei der Abbildung des Scheitelpunktes $(X, Y, H)^{\mathrm{T}}$ eines Fußgängers auf den Bildpunkt $(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$. Gleichung (4.2) bildet entsprechend den Fußpunkt $(X, Y, 0)^{\mathrm{T}}$ auf $(\underset{\sim}{\mathrm{col}}, \mathrm{row} + h)^{\mathrm{T}}$ ab (vgl. Abbildung 4.2). $\underset{\sim}{\mathrm{col}}$ ist dabei nur eine Hilfsvariable im Modell. Trotz möglichen Rollwinkels ϕ der Kamera sind die Objektfenster nur durch den Mittelpunkt der Oberkante und einer Höhe h definitiert. Obwohl $\phi > 0$ möglich ist, wird davon ausgegangen, dass eine Drehung der Hypothese nicht notwendig ist. Die Teillösung (X, Y) des Gleichungssystems ist damit die Rückprojektion eines Fußgängers der Größe H, wenn sich dessen Abbild im Bild an der Spalte col befindet und die Skalierung h aufweist. Die Zeile row ergibt sich dann aus erneuter Projektion ins Bild. Diese Funktion wird mit reproj_H abgekürzt, d.h.

$$\mathrm{row} = \mathrm{reproj}_H(\mathrm{col}, h\,;\, \boldsymbol{\mathcal{P}}), \tag{4.3}$$

mit
- col : Spalte des Fußgängerscheitelpunktes im Bild,
- row : Zeile des Fußgängerscheitelpunktes im Bild,
- h : Größe des Fußgängers im Bild,
- H : Größe des Fußgängers in der Welt,
- $\boldsymbol{\mathcal{P}}$: Projektionsmatrix des Kameramodells.

Der Zusatz $\boldsymbol{\mathcal{P}}$ wird weggelassen, wenn dies aus dem Kontext ersichtlich ist. Alle Hypothesen einer Skalierung liegen in diesem Modell auf einer Linie. Algorithmus

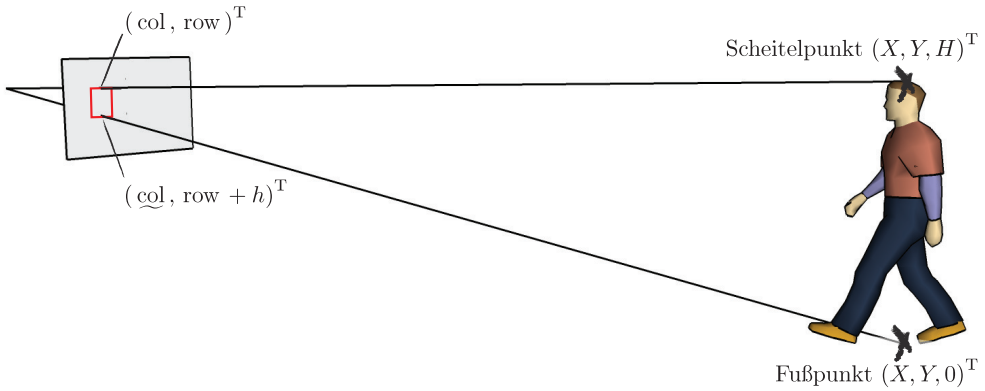


Abbildung 4.2.: Objektfenster im Bild. Ein Objektfenster $o = (\mathrm{col}, \mathrm{row}, h)$ ist durch den Mittelpunkt $(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ der Oberkannte und der Skalierung h definiert. Da die Kamera unter Umständen einen Rollwinkel verschieden von null aufweisen kann, ist der Mittelpunkt der Unterkante nicht notwendigerweise $(\mathrm{col}, \mathrm{row} + h)^{\mathrm{T}}$. Deshalb wird zusätzlich die Unbekannte $\underset{\sim}{\mathrm{col}}$ eingeführt, so dass $(\underset{\sim}{\mathrm{col}}, \mathrm{row} + h)^{\mathrm{T}}$ das Bild des Fußgängerfußpunktes ist.

4.1 fasst die Vorschrift zur Erzeugung der Hypothesenmenge im Modell I nochmals zusammen. Für den Fall der NIR-Kamera werden damit für $h_0 = 10\,\mathrm{px}, h_{\max} = 240\,\mathrm{px}$ und $H = 1.80\,\mathrm{m}$ 141 661 Hypothesen erzeugt.

Modell II: relaxierte ebene Welt mit variabler Fußgängergröße

Die Näherungen in Modell I, also die Annahme einer ebenen Welt sowie die Annahme, dass alle Fußgänger gleich groß sind, modelliert die Wirklichkeit natürlich nur unter harten Einschränkungen. Im Modell II wird die Annahme einer ebenen Welt aufgeweicht, indem der Nickwinkel ϑ im Kameramodell innerhalb eines Relaxationswinkel ρ variiert wird. Durch die Änderung des Nickwinkels entstehen jeweils neue Projektionsmatrizen, die im Folgenden mit $\boldsymbol{\mathcal{P}}_{\vartheta=\vartheta\pm\rho}$ bezeichnet werden[1]. Zusätzlich werden unterschiedliche Fußgängergrößen zugelassen, d.h. eine Hypothese mit der Skalierung h kann sowohl einen nahen kleinen Fußgänger, oder einen weiter entfernten größeren Fußgänger darstellen. Dadurch werden für die Fußgängerposition und deren Größe im Raum Toleranzbereiche zugelassen. Alle Objektfenster einer Skalierung h liegen dann nicht wie in Modell I auf einer Linie, sondern innerhalb eines Bildkorridors zwischen den Zeilen $\mathrm{row}_{\mathrm{up}}$ und $\mathrm{row}_{\mathrm{lo}}$, wie in Abbildung 4.3 illustriert. Der größere Suchbereich kompensiert darüber hinaus Orientierungsfehler im Kameramodell bzw. Nickbewegungen des Fahrzeugs.

Das Modell der relaxierten Welt mit variabler Fußgängergröße ist flexibel und modelliert sehr gut die Anforderungen im Fahrerassistenzbereich. Es kann z.B. auch Kinder mit sehr kleiner Körpergröße berücksichtigen. Im Rahmen dieser Arbeit wird $H_{\min} = 1.60\,\mathrm{m}$ gewählt. Kleinere Kinder werden trotzdem erkannt, da $H_{\min}$ und $H_{\max}$ zusammen mit dem Relaxationswinkel $\rho = 2°$ einen genügend großen vertikalen Suchbereich definieren.

[1] d.h. $\boldsymbol{\mathcal{P}}_{\vartheta=\vartheta\pm\rho}$ geht aus $\boldsymbol{\mathcal{P}}$ hervor, indem alle Kameraparameter außer dem Nickwinkel ϑ gleich bleiben und ϑ durch $\vartheta \pm \rho$ ersetzt wird.

1. Wähle Objektgröße H (z.B. $H = 1.80\,\mathrm{m}$) und initialisiere die Hypothesenmenge $\mathcal{H} := \{\}$. $\mathtt{col}_{\mathtt{min}}$ entspricht dem linken Bildrand, $\mathtt{col}_{\mathtt{max}}$ entsprechend dem rechten Bildrand.
2. Für $h = h_0, \ldots, h_{\mathtt{max}}$
   - $\mathtt{row}_{\mathtt{left}} = \mathtt{reproj}_H(\mathtt{col}_{\mathtt{min}}, h)$
   - $\mathtt{row}_{\mathtt{right}} = \mathtt{reproj}_H(\mathtt{col}_{\mathtt{max}}, h)$
   - Für $\mathtt{col} = \mathtt{col}_{\mathtt{min}}, \ldots, \mathtt{col}_{\mathtt{max}}$
     - $\mathtt{row} = \mathtt{row}_{\mathtt{left}} + \mathtt{col}\frac{\mathtt{col}_{\mathtt{max}} - \mathtt{col}_{\mathtt{min}}}{\mathtt{row}_{\mathtt{right}} - \mathtt{row}_{\mathtt{left}}}$.
     - Hypothese $x = \chi(\mathtt{col}, \mathtt{row}, h)$ und $\mathcal{H} = \mathcal{H} \cup \{x\}$.
3. Ausgabe ist die Hypothesenmenge $\mathcal{H}$. Alle so erzeugten Hypothesen entsprechen Fußgängern der festen Größe H, die sich in einer ebenen Welt im Sichtbereich der Kamera befinden.

Algorithmus 4.1: Hypothesenmenge im Modell I. Erzeugung einer Hypothesenmenge zur Detektion von Fußgängern im Modell einer ebenen Welt mit genormter Fußgängergröße (Modell I).

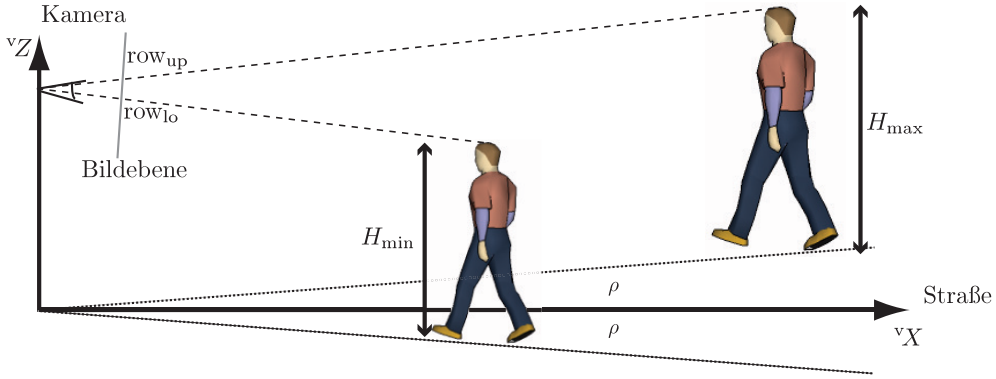


Abbildung 4.3.: Relaxation der ebenen Welt zur Bestimmung des Suchraums. Zur geometrischen Bestimmung des Suchraums wird für jede Skalierung die Ober- und Untergrenze für die obere Objektfensterkante im Bild bestimmt. Die Grenzen $\mathrm{row}_{\mathrm{up}}$ und $\mathrm{row}_{\mathrm{lo}}$ ergeben sich, wenn der Fußgänger einmal mit der kleinsten und einmal mit der größten erwarteten Objektgröße (H_{min} bzw. H_{max}) auf die Bildebene projeziert wird. Die ebene Welt wird dabei zusätzlich um den Relaxationswinkel ρ gekippt.

Mit $h_0 = 10\,\text{px}$ im NIR-Bild können theoretisch Kinder der Größe $H = 1.00\,\text{m}$ bis in eine Entfernung von 100 m erkannt werden. Im FIR Fall ist die Detektionsreichweite jedoch aufgrund der geringen Auflösung des Sensors kleiner. Mit $h_0 = 7\,\text{px}$ ist nur eine theoretische Reichweite von ca. 70 m möglich.

Die Erzeugung der Hypothesen nach Modell II ist in Algorithmus 4.2 dargestellt. Für $h_0 = 10\,\text{px}, h_{\max} = 240\,\text{px}, H_{\min} = 1.60\,\text{m}, H_{\max} = 2.00\,\text{m}, \rho = 2°$ werden damit im NIR-Bild 14 309 323 Hypothesen erzeugt.

1. Wähle minimale Objektgröße $H_{\min}$ (z.B. $H_{\min} = 1.60\,\text{m}$), wähle maximale Objektgröße $H_{\max}$ (z.B. $H_{\max} = 2.00\,\text{m}$) und initialisiere die Hypothesenmenge $\mathcal{H} := \{\}$. $\text{col}_{\min}$ entspricht dem linken Bildrand, $\text{col}_{\max}$ entsprechend dem rechten Bildrand.

   Zur Notation: $\boldsymbol{\mathcal{P}}_{\vartheta=\vartheta\pm\rho}$ geht aus $\boldsymbol{\mathcal{P}}$ hervor, indem alle Kameraparameter außer dem Nickwinkel ϑ gleich bleiben und ϑ durch $\vartheta \pm \rho$ ersetzt wird.

2. Für $h = h_0, \ldots, h_{\max}$
   - $\text{row}_{\text{up,left}} = \text{reproj}_{H_{\max}}(\text{col}_{\min}, h\,;\, \boldsymbol{\mathcal{P}}_{\vartheta=\vartheta-\rho})$
   - $\text{row}_{\text{up,right}} = \text{reproj}_{H_{\max}}(\text{col}_{\max}, h\,;\, \boldsymbol{\mathcal{P}}_{\vartheta=\vartheta-\rho})$
   - $\text{row}_{\text{lo,left}} = \text{reproj}_{H_{\min}}(\text{col}_{\min}, h\,;\, \boldsymbol{\mathcal{P}}_{\vartheta=\vartheta+\rho})$
   - $\text{row}_{\text{lo,right}} = \text{reproj}_{H_{\min}}(\text{col}_{\max}, h\,;\, \boldsymbol{\mathcal{P}}_{\vartheta=\vartheta+\rho})$
   - Für $\text{col} = \text{col}_{\min}, \ldots, \text{col}_{\max}$
     - $\text{row}_{\text{up}} = \text{row}_{\text{up,left}} + \text{col}\frac{\text{col}_{\max}-\text{col}_{\min}}{\text{row}_{\text{up,right}}-\text{row}_{\text{up,left}}}$.
     - $\text{row}_{\text{lo}} = \text{row}_{\text{lo,left}} + \text{col}\frac{\text{col}_{\max}-\text{col}_{\min}}{\text{row}_{\text{lo,right}}-\text{row}_{\text{lo,left}}}$.
     - Für $\text{row} = \text{row}_{\text{up}}, \ldots, \text{row}_{\text{lo}}$:

       Hypothese $x = \chi(\text{col}, \text{row}, h)$ und $\mathcal{H} = \mathcal{H} \cup \{x\}$.

3. Ausgabe ist die Hypothesenmenge $\mathcal{H}$. Alle so erzeugten Hypothesen entsprechen im Bild abgebildete Fußgänger mit einer Größe im Intervall $[H_{\min}, H_{\max}]$, die sich auf einer Ebene befinden, die zur ebenen Welt um einen Winkel $\leq \pm\rho$ geneigt ist.

Algorithmus 4.2: Hypothesenmenge im Modell II. Erzeugung einer Hypothesenmenge zur Detektion von Fußgängern im Modell einer relaxierten ebenen Welt mit variabler Fußgängergröße.

Modell III: relaxierte ebene Welt mit variabler Fußgängergröße und skalierungsabhängiger Unterabtastung

Die Relaxation der ebenen Welt und die Berücksichtigung variabler Objektgrößen in Modell II schränken den Suchbereich im Bild in sehr sinnvoller Weise ein. Dennoch ist durch die pixelweise Abtastung innerhalb des Suchbereichs sowie die Berücksichtigung aller Skalierungen im Bereich $[h_0, h_{\max}]$ die Anzahl der Hypothesen sehr groß. Ersetzt man die pixelgenaue Abtastung durch eine skalierungsabhängige Unterabtastung, so kann die Anzahl der Hypothesen deutlich reduziert werden. Im Modell III wird die Rasterschrittweite durch Q_{col} und Q_{row} proportional zur Skalierung h festgelegt. Innerhalb des Bildkorridors einer Skalierung liegen alle Objektfenster dann auf dem Raster

$$\left(k \cdot \lceil Q_{\text{col}} \cdot h \rceil \, , \text{row}_{\min}(h) + l \cdot \lceil Q_{\text{row}} \cdot h \rceil\right)^{\text{T}}, \quad k, l \in \mathbb{N}_0^+.$$

Die Rasterschrittweiten sind also jeweils $Q_{\text{col}} \cdot h$ und $Q_{\text{row}} \cdot h$. $\lceil \cdot \rceil$ entspricht dabei der Aufrundungsfunktion mit $\lceil x \rceil = \min_{k \in \mathbb{Z}, k \geq x}(k)$.

Neben der Quantisierung der Objektfenster innerhalb der Bildkorridore unterliegen die möglichen Skalierungen selbst einer Quantisierung. Ausgehend von einer minimalen Skalierung h_0 ist die nächst größere Skalierung jeweils um den Faktor Q_h größer, d.h. $h_{n+1} = \lceil h_n \cdot (1 + Q_h) \rceil$.

Die skalierungsabhängige Art der Quantisierung lässt sich mit einer Eigenschaft des Detektors motivieren, nämlich der Tatsache, dass mit der Größenskalierung der Merkmale auch die Unschärfe ihrer Lokalisation im Bild zunimmt[2]. Für größere Skalierungen reicht also eine große Rasterschrittweite, für kleinere Skalierungen ist dagegen eine feinere Rasterung notwendig.

Zusammenfassend wird die Hypothesenmenge $\mathcal{H}$ des einfachen Hypothesengenerators neben den Parametern des Kameramodells in erster Linie durch die unterschiedlichen Quantisierungen parametrisiert. Für eine feste Skalierung h ist dann die Rasterschrittweite der Objektfenster innerhalb einer Zeile $Q_{\text{col}} \cdot h$, bzw. innerhalb einer Spalte $Q_{\text{row}} \cdot h$. Die nächst größere Skalierung ist $\lceil h \cdot (1 + Q_h) \rceil$.

Algorithmus 4.3 zeigt den genauen Ablauf der Hypothesenerzeugung nach Modell III. Im Folgenden werden die Parameter der Hypothesenmenge - falls nicht aus dem Kontext ersichtlich - durch das Tripel $(Q_{\text{col}}, Q_{\text{row}}, Q_h)$ angezeigt. Eine typische Hypothesenmenge für den NIR-Sensor ist dann z.B. $\mathcal{H}_{\text{NIR}}(0.03, 0.05, 0.08)$. Mit dieser kann ohne signifikante Einbußen bzgl. Detektionssicherheit eine Reduktion der 45 Millionen Hypothesen des vollständigen Suchraums auf 676 088 Hypothesen im NIR-Bild erreicht werden.

[2] Zur Erinnerung: die Merkmale sind im Raster des Basissuchfensters und damit der Objektfenster definiert (siehe Kapitel 3.1) und werden entsprechend der Größe der Hypothesen mitskaliert.

1. Wähle minimale Objektgröße $H_{\mathtt{min}}$ (z.B. $H_{\mathtt{min}} = 1.60\mathrm{m}$), wähle maximale Objektgröße $H_{\mathtt{max}}$ (z.B. $H_{\mathtt{max}} = 2.00\,\mathrm{m}$) und initialisiere die Hypothesenmenge $\mathcal{H} := \{\}$. $\mathtt{col}_{\mathtt{min}}$ entspricht dem linken Bildrand, $\mathtt{col}_{\mathtt{max}}$ entsprechend dem rechten Bildrand.

2. Wähle Quantisierungen $Q_{\mathtt{col}}$, $Q_{\mathtt{row}}$, Q_h (z.B. $Q_{\mathtt{col}} = 0.03, Q_{\mathtt{row}} = 0.05, Q_h = 0.08$).

3. Wiederhole bis $h \geq h_{\mathtt{max}}$:
   - Bestimme $\mathtt{row}_{\mathtt{up,left}}$, $\mathtt{row}_{\mathtt{up,right}}$, $\mathtt{row}_{\mathtt{lo,left}}$ und $\mathtt{row}_{\mathtt{lo,right}}$ wie in Algorithmus 4.2 nach Modell II.
   - Für $\mathtt{col} = \mathtt{col}_{\mathtt{min}}, \ldots, \mathtt{col}_{\mathtt{max}}$, Schrittweite $\lceil Q_{\mathtt{col}} \cdot h \rceil$:
     - $\mathtt{row}_{\mathtt{up}} = \mathtt{row}_{\mathtt{up,left}} + \mathtt{col} \frac{\mathtt{col}_{\mathtt{max}} - \mathtt{col}_{\mathtt{min}}}{\mathtt{row}_{\mathtt{up,right}} - \mathtt{row}_{\mathtt{up,left}}}$.
     - $\mathtt{row}_{\mathtt{lo}} = \mathtt{row}_{\mathtt{lo,left}} + \mathtt{col} \frac{\mathtt{col}_{\mathtt{max}} - \mathtt{col}_{\mathtt{min}}}{\mathtt{row}_{\mathtt{lo,right}} - \mathtt{row}_{\mathtt{lo,left}}}$.
     - Für $\mathtt{row} = \mathtt{row}_{\mathtt{up}}, \ldots, \mathtt{row}_{\mathtt{lo}}$, Schrittweite $\lceil Q_{\mathtt{row}} \cdot h \rceil$:
       Hypothese $x = \chi(\mathtt{col}, \mathtt{row}, h)$ und $\mathcal{H} = \mathcal{H} \cup \{x\}$.
   - $h = \lceil h\,(1 + Q_h) \rceil$.

4. Ausgabe ist die Hypothesenmenge $\mathcal{H}$. Alle so erzeugten Hypothesen entsprechen im Bild abgebildete Fußgänger mit einer Größe im Intervall $[H_{\mathtt{min}}, H_{\mathtt{max}}]$, die sich auf einer Ebene befinden, die zur ebenen Welt um einen Winkel $\leq \pm\rho$ geneigt ist.

Algorithmus 4.3: Hypothesenmenge im Modell III. Erzeugung einer Hypothesenmenge zur Detektion von Fußgängern im Modell einer relaxierten ebenen Welt mit variabler Fußgängergröße und skalierungsabhängiger Unterabtastung.

4.2. Multi-Sensor Hypothesengenerator

Eine einzelne Hypothese im Fall mehrerer Sensoren wird durch ein Tupel von Suchfenstern mit je einem Suchfenster pro Sensorstrom beschrieben. Die einzelnen Suchfenster werden dabei wie im einfachen Hypothesengenerator durch die Mittelpunkte der Oberkanten $(\mathrm{col}'_i, \mathrm{row}'_i)^{\mathrm{T}}$ und den Suchfensterhöhe h'_i angegeben. Eine Hypothese im Fall der Fusion mit zwei Sensoren ist also $x = (s_1, s_2) = ((\mathrm{col}'_1, \mathrm{row}'_1, h'_1), (\mathrm{col}'_2, \mathrm{row}'_2, h'_2))$. Die zugehörigen Objektfenster sind $o_1 = (\mathrm{col}_1, \mathrm{row}_1, h_1)$ und $o_2 = (\mathrm{col}_2, \mathrm{row}_2, h_2)$. Die Seitenverhältnisse r_1, r_2 der zugehörigen Rechtecke werden wieder als konstant vorausgesetzt.

Der im Folgenden beschriebene Multi-Sensor Hypothesengenerator erzeugt Hypothesen durch geeignete Tupelbildung von Hypothesen des einfachen Hypothesengenerators. Dazu werden zunächst für alle einzelnen Sensoren entsprechend der Modellannahmen in Abschnitt 4.1 Einzel-Sensor Hypothesen erzeugt und dann einander zugeordnet.

Ohne Beschränkung der Allgemeinheit wird der Multi-Sensor Hypothesengenerator hier nur für den Anwendungsfall der Kombination von zwei Sensoren beschrieben. Die Techniken lassen sich jedoch leicht auch auf mehrere Sensoren übertragen. Ausgehend von einem Suchfenster aus den Einzel-Sensor Hypothesen des einen Sensors (im Folgenden als Primärsensor bezeichnet) wird ein Korrespondenzbereich im anderen Sensor (im Folgenden als Sekundärsensor bezeichnet) generiert, der die zugeordneten Suchfenster aus den Einzel-Sensor Hypothesen umfasst.

Ist vom Primärsensor das Objektfenster o^* im Modell I (also einer ebenen Welt mit fester Fußgängergröße H) gegeben, so ist unter diesem Modell das zugehörige Objektfenster o im Sekundärsensor eindeutig definiert: der Scheitelpunkt $(X, Y, H)^{\mathrm{T}}$ des Fußgängers ist als Teillösung von (4.1) und (4.2) bekannt und kann damit eindeutig im zweiten Sensor abgebildet werden. Diese Abbildung wird bezeichnet mit

$$o = \mathrm{proj\_stream2stream}_H\left(o^*\,;\,\boldsymbol{\mathcal{P}}^*, \boldsymbol{\mathcal{P}}\right), \tag{4.4}$$

mit
- o : Objektfenster im Sekundärsensor,
- $o*$: Objektfenster im Primärsensor,
- H : Größe des Fußgängers in der Welt,
- $\boldsymbol{\mathcal{P}}^*$: Projektionsmatrix des Primärsensor-Kameramodells,
- $\boldsymbol{\mathcal{P}}$: Projektionsmatrix des Sekundärsensor-Kameramodells.

Der Zusatz $\boldsymbol{\mathcal{P}}^*, \boldsymbol{\mathcal{P}}$ wird weggelassen, wenn aus dem Kontext ersichtlich. In Modell II und III, einer relaxierten ebenen Welt mit variabler, unbekannter Objektgröße, ist eine solche eindeutige Abbildung nicht mehr gegeben. Dem Objektfenster $o^* = (\mathrm{col}^*, \mathrm{row}^*, h^*)$ des Primärsensors können mehrere Objektfenster $\{o_1, o_2, \ldots\}$ im Sekundärsensor zugeordnet werden, je nachdem welche Objektgröße angenommen wird. Ohne Berücksichtigung eines Relaxationswinkels (d.h. $\rho = 0$) werden alle zugehörigen Objektfenster im zweiten Sensor durch die Epipolarlinien l_{up} und l_{lo} begrenzt, wobei l_{up} die Epipolarlinie des Scheitelpunktes $(\mathrm{col}^*, \mathrm{row}^*)^{\mathrm{T}}$ und l_{lo} entsprechend die Epipolarlinie des Fußpunktes

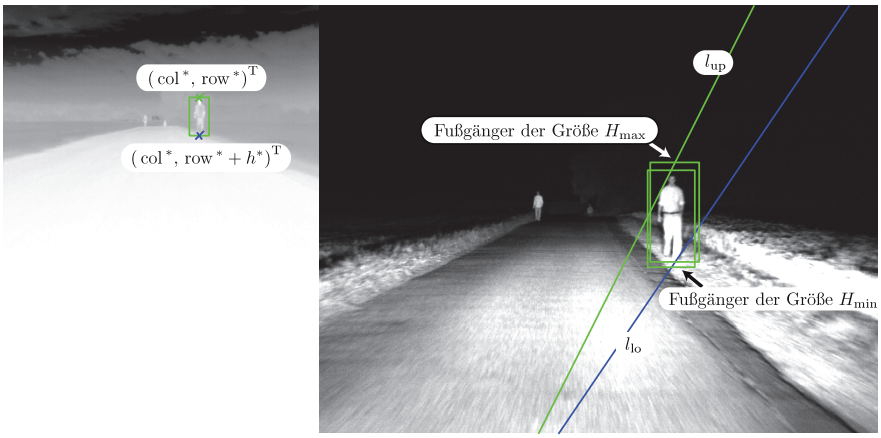


Abbildung 4.4.: Korrespondierende Suchfenster im Sekundärsensor für unterschiedliche Fußgängergrößen. Für den Relaxationswinkel $\rho = 0$ werden alle zugehörigen Objektfenster im zweiten Sensor durch die Epipolarlinien l_{up} des Fußgängerscheitelpunktes und l_{lo} des Fußgängerfußpunktes beschrieben. Die möglichen Skalierungen sind dabei durch die möglichen Fußgängergrößen H_{min} und H_{max} eingeschränkt (hier: $H_{\mathrm{min}} = 1.60\,\mathrm{m}, H_{\mathrm{max}} = 2.00\,\mathrm{m}$).

$(\mathrm{col}^*, \mathrm{row}^* + h^*)^{\mathrm{T}}$ darstellt (Abbildung 4.4). Hier wird angenommen, dass der Rotationswinkel der Kameras so klein ist, dass eine Drehung der Objektfenster nicht notwendig ist. Die möglichen Skalierungen sind dabei durch die möglichen Fußgängergrößen H_{min} und H_{max} eingeschränkt. Soll zusätzlich ein Relaxationswinkel $\rho \neq 0$ berücksichtigt werden, befinden sich die Scheitelpunkte der möglichen Korrespondenzobjektfenster nicht mehr auf einer Linie, sondern in einem Band um die Epipolarlinie mit $\rho = 0$ (Abbildung 4.5).

In der algorithmischen Umsetzung wird das Band in dieser Arbeit durch einen rechteckigen Korrespondenzbereich angenähert. Dessen Grenzen ergeben sich dadurch, dass bei der Projektion von einem Sensor in den anderen einmal von einem Fußgänger der Größe H_{min} bei einem Relaxationswinkel $-\rho$ und einmal von einem Fußgänger der Größe H_{max} bei einem Relaxationswinkel $+\rho$ ausgegangen wird. Darüber hinaus werden aus dem Korrespondenzbereich nur solche Suchfenster zugeordnet, die der mittleren Suchfensterhöhe im Korrespondenzbereich entsprechen. Diese Einschränkung ist zulässig, da die Spannweite der möglichen Objektfenstergrößen im Sekundärsensor ab einer Entfernung von 20 m vor dem Fahrzeug kleiner als die halbe Rasterschrittweite $\frac{Q_h}{2}$ bei der Erzeugung der Hypothesen im Sekundärsensor ist (Abbildung 4.6 mit $Q_h = 0.05$).

Zum Ausgleich der Quantisierungseffekte selbst muss der Korrespondenzbereich zusätzlich um die halbe, skalierungsabhängige Rasterschrittweite erweitert werden. Zusätzlich dazu wird ein minimaler Toleranzbereich angenommen, um vor allem bei kleinen Skalierungen Fehler aufgrund ungenauer Synchronisierung der Sensoren bzw. Fehler in der Kamerakalibrierung zu kompensieren.

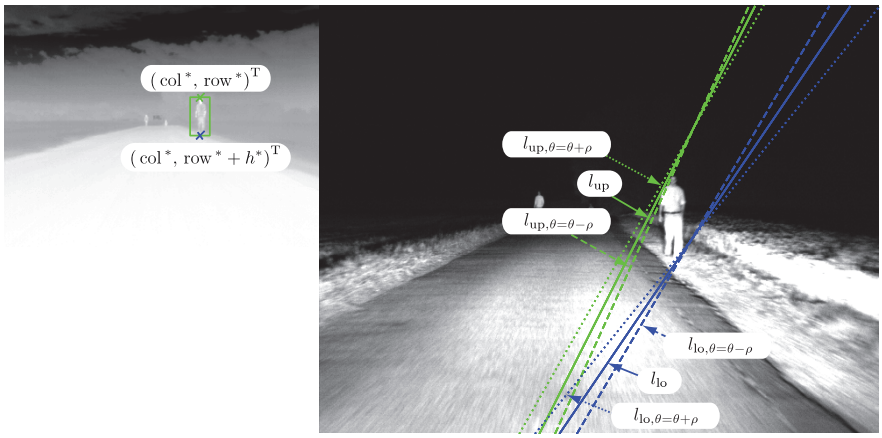


Abbildung 4.5.: Epipolarlinien zur Bestimmung des Korrespondenzbereiches bei unterschiedlichen Relaxationswinkeln. Der Relaxationswinkel wirkt sich direkt auf die Lage der Epipolarlinien aus. Dadurch liegen die Scheitelpunkte nicht mehr nur auf einer Linie, sondern innerhalb eines Bandes um die Epipolarlinie mit $\rho = 0$.

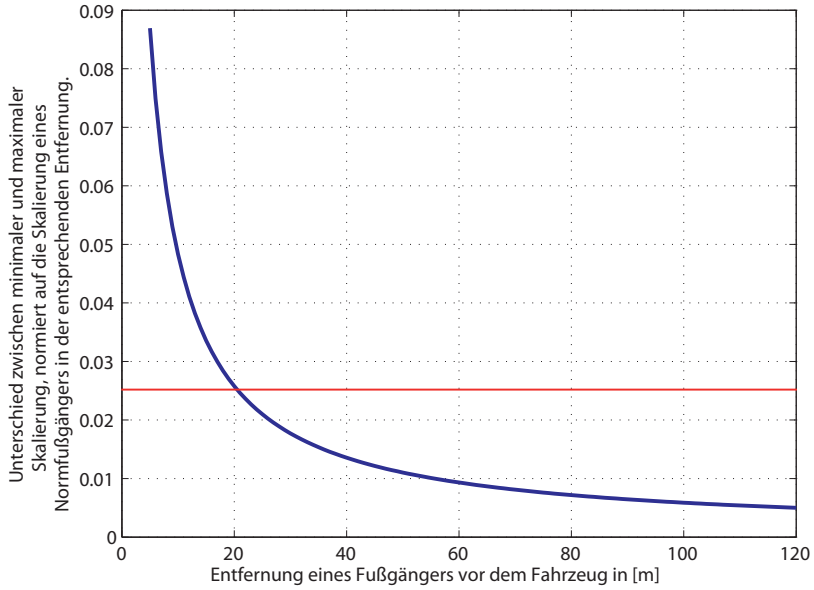


Abbildung 4.6.: Spannweite möglicher Skalierungen im Sekundärsensor. Geht man bei der Projektion einer Hypothese von einem Sensor in den anderen einmal von einem Fußgänger der Größe $H_{\min}$ und einmal von einem Fußgänger der Größe $H_{\max}$ aus (hier: $H_{\min} = 1.60\,\mathrm{m}, H_{\max} = 2.00\,\mathrm{m}$), so ist der Unterschied zwischen minimaler und maximaler Skalierung der Objektfenster - jeweils normiert auf die Skalierung eines Normfußgängers der Größe $\frac{1}{2}(H_{\min} + H_{\max})$ - ab einer Entfernung von 20 m kleiner als die halbe Rasterschrittweite $\frac{Q_h}{2}$ (bei $Q_h = 0.05$). Zur Korrespondenzbestimmung reicht es also, die Skalierung des Normfußgängers zu berücksichtigen.

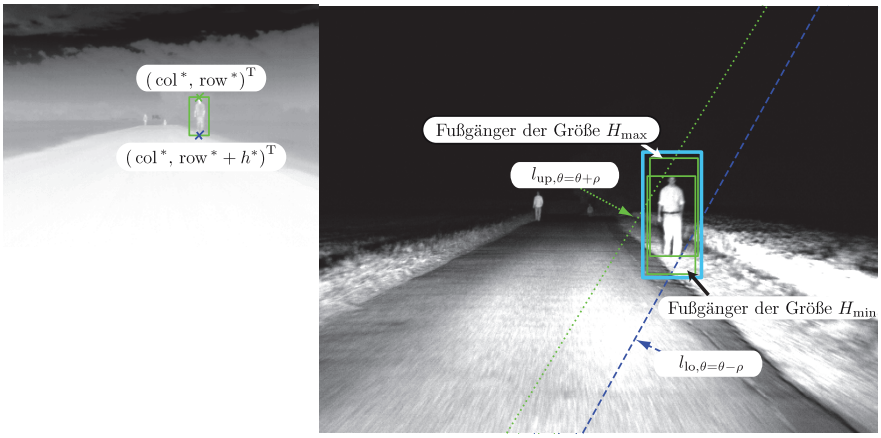


Abbildung 4.7.: Korrespondenzbereich zur Bestimmung von Objektfensterpaaren. Die Grenzen des rechteckigen Korrespondenzbereiches (umschreibendes Rechteck) ergeben sich dadurch, dass bei der Projektion von einem Sensor in den anderen einmal von einem Fußgänger der Größe $H_{\min}$ bei einem Relaxationswinkel $-\rho$ (unteres Rechteck) und einmal von einem Fußgänger der Größe $H_{\max}$ bei einem Relaxationswinkel $+\rho$ (oberes Rechteck) ausgegangen wird (hier: $H_{\min} = 1.60\,\mathrm{m}, H_{\max} = 2.00\,\mathrm{m}, \rho = 3°$). Der Korrelationsbereich wird zusätzlich um einen Toleranzbereich erweitert.

Das gesamte Vorgehen zur Bestimmung der Korrespondenzhypothesen $\hat{\mathcal{H}}(o^*)$ zum Objektfenster o^* ist in Abbildung 4.7 und Algorithmus 4.4 zusammengefasst. Ist $\mathcal{H}^*$ die Menge der Hypothesen im Primärsensor, so ist für den Multi-Sensor Fall:

$$\mathcal{H} = \bigcup_{\chi(o^*) \in \mathcal{H}^*} \hat{\mathcal{H}}(o^*) .$$

Wählt man den FIR-Sensor als Primärsensor ist mit $\mathcal{H}^* = \mathcal{H}_{\mathrm{FIR}}(0.03, 0.05, 0.08)$ und $\mathcal{H}' = \mathcal{H}_{\mathrm{NIR}}(0.03, 0.05, 0.08)$, sowie $h_{0,\mathrm{FIR}} = 7\,\mathrm{px}, h_{\max,\mathrm{FIR}} = 80\,\mathrm{px}, h_{0,\mathrm{NIR}} = 10\,\mathrm{px}, h_{\max,\mathrm{NIR}} = 240\,\mathrm{px}, H_{\min} = 1.60\,\mathrm{m}, H_{\max} = 2.00\,\mathrm{m}, \rho = 2°$ und $\mathrm{tol}_{\mathrm{col}} = \mathrm{tol}_{\mathrm{row}} = 1\,\mathrm{px}$ die Zahl der Hypothesen im Fusionsfall damit 1 395 330 pro Bild.

4.3. Hypothesenbaum

Der Suchraum des Detektors wurde mit den Hypothesengeneratoren aus Abschnitt 4.1 und 4.2 durch geeignete Modellannahmen so weit wie möglich eingeschränkt. Es bleibt jedoch das Problem der optimalen Parametrisierung, speziell die Festlegung der geeigneten Abtast-Schrittweiten. Eine zufriedenstellende Lösung der Detektionsaufgabe konnte mit den empirisch bestimmten Abtastschrittweiten $Q_{\mathrm{col}} = 0.03$, $Q_{\mathrm{row}} = 0.05$ und $Q_h = 0.08$ erreicht werden. Mit fast 1.4 Millionen Hypothesen im Fall der Fusion von FIR und NIR-Sensor ist der Aufwand für eine Echtzeitanwendung allerdings immer noch zu hoch. Die Verwendung großer Abtastschrittweiten reduziert die Anzahl der Hypothesen zwar deutlich, allerdings auf Kosten der Detektionssicherheit. Im Folgenden wird schrittweise das Konzept eines Raster-Hypothesenbaums entwickelt, das durch

1. Ausgangspunkt ist ein Objektenster o^* aus der Einzelstromhypothesenmenge des Primärsensors. Die Projektionsmatrix im Kameramodell des Primärsensors ist $\boldsymbol{\mathcal{P}}^*$, die des Sekundärsensors ist $\boldsymbol{\mathcal{P}}$. Die minimale Fußgängergröße sei $H_{\mathtt{min}}$ (z.B. $H_{\mathtt{min}} = 1.6\,\mathtt{m}$), die maximale Fußgängergröße sei $H_{\mathtt{max}}$ (z.B. $H_{\mathtt{max}} = 2.0\,\mathtt{m}$). Der minimale Toleranzbereich um den Korrespondenzbereich sei im Bildraum mit $\mathtt{tol}_{\mathtt{min,row}}$ und $\mathtt{tol}_{\mathtt{min,col}}$ (z.B. $\mathtt{tol}_{\mathtt{min,row}} = \mathtt{tol}_{\mathtt{min,col}} = 4\,\mathtt{px}$) angegeben. Die Einzelstromhypothesenmenge des Sekundärsensors sei $\mathcal{H}' = \mathcal{H}'\left(Q'_{\mathtt{col}}, Q'_{\mathtt{row}}, Q'_h\right)$.

   Zur Notation: $\boldsymbol{\mathcal{P}}_{\vartheta=\vartheta\pm\rho}$ geht aus $\boldsymbol{\mathcal{P}}$ hervor, indem alle Kameraparameter außer dem Nickwinkel ϑ gleich bleiben und ϑ durch $\vartheta \pm \rho$ ersetzt wird.

2. $o_1 = \mathtt{proj\_stream2stream}_{H_{\mathtt{min}}}\left(o^*\,;\, \boldsymbol{\mathcal{P}}^*_{\vartheta=\vartheta-\rho}, \boldsymbol{\mathcal{P}}_{\vartheta=\vartheta-\rho}\right)$
   $o_2 = \mathtt{proj\_stream2stream}_{H_{\mathtt{max}}}\left(o^*\,;\, \boldsymbol{\mathcal{P}}^*_{\vartheta=\vartheta+\rho}, \boldsymbol{\mathcal{P}}_{\vartheta=\vartheta+\rho}\right)$

3. Bestimme das umschreibende Rechteck $B' = (\mathtt{col}_{B'}, \mathtt{row}_{B'}, w_{B'}, h_{B'})$, das die beiden Objektfenster o_1 und o_2 enthält.

4. Der Bereich, der alle korrespondierenden Objektfenster enthält ist dann gegeben durch

$$B(o^*) = (\mathtt{col}_{B'} - \mathtt{tol}_{\mathtt{col}}, \mathtt{row}_{B'} - \mathtt{tol}_{\mathtt{row}}, \\ w_{B'} + 2 \cdot \mathtt{tol}_{\mathtt{col}}, h_{B'} + 2 \cdot \mathtt{tol}_{\mathtt{row}}),$$

   mit

$$\mathtt{tol}_{\mathtt{col}} = \max\left(\mathtt{tol}_{\mathtt{min,col}}, \frac{1}{2} Q'_{\mathtt{col}} \frac{h_1 + h_2}{2}\right)$$
$$\mathtt{tol}_{\mathtt{row}} = \max\left(\mathtt{tol}_{\mathtt{min,row}}, \frac{1}{2} Q'_{\mathtt{row}} \frac{h_1 + h_2}{2}\right).$$

5. Die Menge der Multi-Sensor Hypothesen $\hat{\mathcal{H}}(o^*)$, die zum Objektfenster o^* gehören ist dann mit $s^* = \chi(o^*)$:

$$\hat{\mathcal{H}}(o^*) = \left\{(s^*, s) \,\middle|\, \mathtt{cov}_{\mathtt{max}}(s, B(o^*)) = 1 \wedge s \in \mathcal{H}'\right\}.$$

Algorithmus 4.4: Bestimmung der Korrespondenzhypothesen zu einem Objektfenster im Primärsensor.

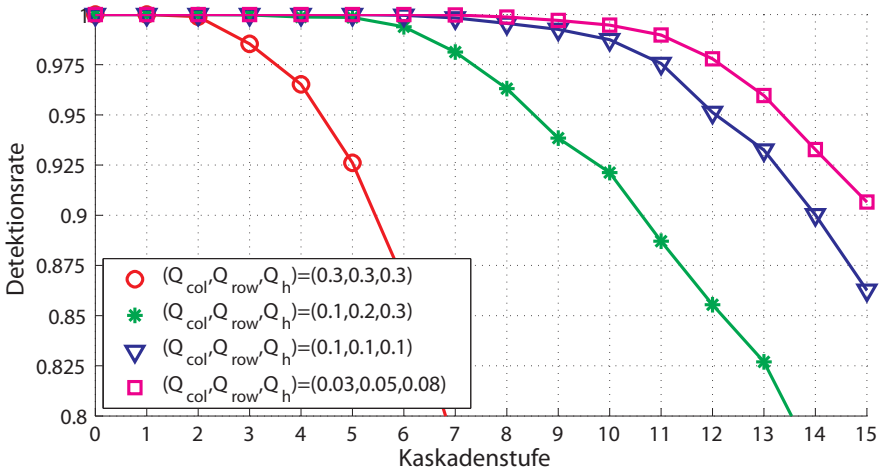


Abbildung 4.8.: Vergleich der Detektionsraten verschiedener Rasterdichten. Für jede der Rasterdichten ist die Detektionsrate eines NIR-Detektors über die Zahl der verwendeten Stufen aufgetragen.

eine dynamische lokale Steuerung der skalierungsabhängigen Unterabtastung in Modell III die Anzahl der zu überprüfenden Hypothesen effektiv ohne große Einbußen der Detektionssicherheit reduziert.

Charakteristische Detektorantwort

Zur Untersuchung des Zusammenhangs der Rasterdichte und der Detektionsleistung wurden in [Rot06] Experimente mit verschiedenen Rasterdichten durchgeführt. Dabei zeigte sich vor allem, dass bei sehr grober Abtastung Fußgänger zwar nicht mehr erkannt, Hypothesen im Umfeld des Fußgängers aber auch nicht bereits in den ersten Kaskadenstufen verworfen werden. Die Ergebnisse dieses Experiments sind in Abbildung 4.8 zu sehen. Selbst bei sehr grober Rasterung wird der größte Teil der zu detektierenden Fußgänger mit den ersten Stufen des Detektors erkannt.

Grund für dieses Phänomen ist eine Eigenschaft des Detektors, die im Folgenden als charakteristische Detektorantwort bezeichnet wird: Die Antwort des Detektors ist maximal für eine Hypothese, die exakt auf dem Fußgänger positioniert ist. Schiebt man die Hypothese schrittweise vom Fußgänger weg, fällt das Detektorergebnis nicht abrupt auf Null ab, sondern es gibt einen Bereich, in dem es stark variiert und tendenziell absinkt. Eine exemplarische Visualisierung der charakteristischen Detektorantwort ist in Abbildung 4.9 dargestellt. Dabei wurden Hypothesen mit fester Skalierung und pixelgenauer Abtastung erzeugt und die jeweilige Detektorantwort farblich an Stelle der Mittelpunkte der Hypothesen im Bild kodiert. Man kann den Bereich, in dem die Detektorantwort abfällt gut erkennen. Umgangssprachlich vergrößert sich bei niedrigen Stufen die „Trefferfläche für einen Fußgänger“. Es ist zu erwarten, dass der Detektor eine ähnliche Charakteristik bei variierender Skalierung aufweist.

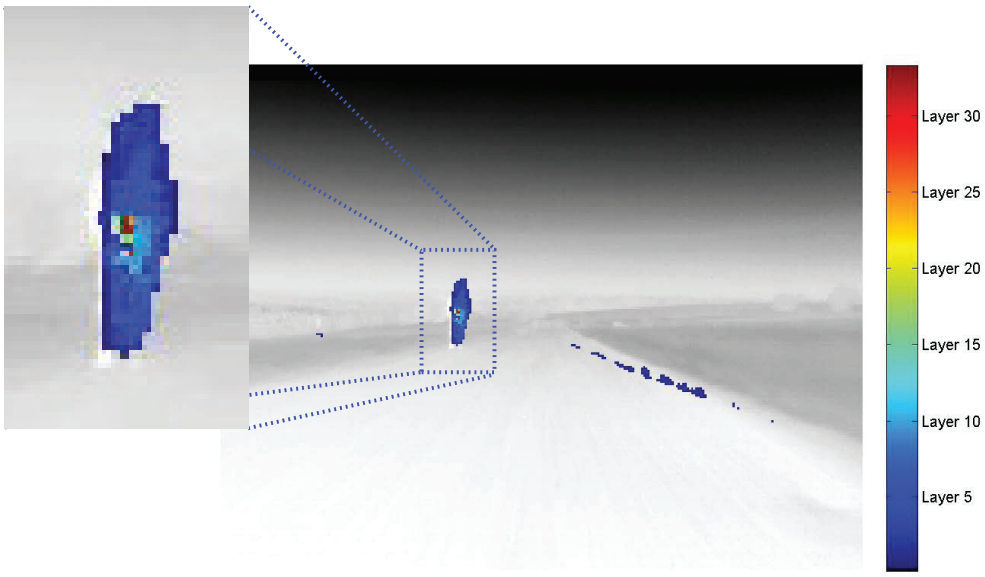


Abbildung 4.9.: Illustration der charakteristischen Detektorantwort. Zur Visualisierung wurden Hypothesen mit fester Skalierung und pixelgenauer Abtastung erzeugt und die jeweils erreichte Kaskadenstufe farblich im Bild kodiert. Dargestellt sind jeweils die Mittelpunkte der Hypothesen. Man kann den Bereich, in dem die Detektorantwort abfällt gut erkennen. In diesem Beispiel wurde ein FIR-Monoklassifikator auf einem FIR-Bild mit einem Fußgänger am linken Straßenrand angewandt.

Zur Erklärung der charakteristischen Detektorantwort sei angemerkt, dass während des Trainings bewusst keine zu einem Label leicht versetzten Trainingsbeispiele verwendet wurden. Dennoch muss man davon ausgehen, dass die Positivbeispiele aufgrund ungenauer Labels nicht immer exakt waren. Trotzdem ist die charakteristische Detektorantwort nicht konstruiert, sondern muss für jeden Detektor experimentell untersucht werden.

Grob-zu-fein Suchstrategie

Weist ein Detektor die beschriebene charakteristische Detektorantwort auf, kann diese zur Reduktion der Hypothesenanzahl in Form einer grob-zu-fein Suche verwendet werden. Initial wird das Bild mit einer grob aufgelösten Hypothesenmenge $\mathcal{H}^{(1)} := \mathcal{H}(Q_{\text{col}}^{(1)}, Q_{\text{row}}^{(1)}, Q_h^{(1)})$ abgesucht und die Rasterdichte dynamisch an den Stellen sukzessive verfeinert, an denen die Hypothesen eine bestimmte Kaskadenstufe erreichen oder überschreiten (Abbildung 4.10). Dadurch wird nur noch die lokale Nachbarschaft derjenigen Hypothesen untersucht, die einen Fußgänger in der Nähe vermuten lassen.

Die Kaskadenstufe als Kriterium zur Verfeinerung des Suchrasters motiviert sich dabei aus der charakteristischen Detektorantwort, d.h. nur an den Stellen im Bild, die im groben Raster eine genügend hohe Detektorantwort aufweisen, wird im nächst feineren Raster weitergesucht. Nach dem gleichen Prinzip kann dann erneut weiter verfeinert werden, usw. bis das feinste Suchraster erreicht ist.

Für jeden Verfeinerungsschritt muss dafür ein Schwellwert als Kriterium definiert sein. Diese Schwellwerte können leicht aus einer Auswertung wie in Abbildung 4.8 festgelegt werden. Es seien dazu $\mathcal{H}^{(l)} := \mathcal{H}(Q_{\text{col}}^{(l)}, Q_{\text{row}}^{(l)}, Q_h^{(l)}), l = 1, \ldots, L$ Hypothesenmengen mit unterschiedlichen Rasterschrittweiten, wobei $\mathcal{H}^{(1)}$ die Menge mit der gröbsten und $\mathcal{H}^{(L)}$ die Menge mit der feinsten Rasterung bezeichnet. Als Schwellwert $k^{(l)}, l = 1, \ldots, L-1$ wird jeweils die maximale Kaskadenstufe gewählt, für die die Hypothesenmenge $\mathcal{H}^{(l)}$ immer noch fast die gleiche Detektionsrate erreicht wie die feinste Hypothesenmenge $\mathcal{H}^{(L)}$, d.h. so dass für $\alpha \leq 1$ gilt:

$$D_{k^{(l)}}^{(l)} \geq \alpha \cdot D_{k^{(l)}}^{(L)}.$$

$D_{k^{(l)}}^{(L)}$ ist dabei die Detektionsrate im feinsten Raster $\mathcal{H}^{(L)}$ bei Kaskadenstufe $k^{(l)}$. Insgesamt ist damit sichergestellt, dass der gesamte Detektor eine Detektionsrate von

$$D_K \leq \alpha^{L-1} \cdot D_K^{(L)}$$

erreicht. In vorliegender Arbeit wurden für α Werte zwischen 0.98 und 0.995 verwendet.

Nachbarschaftsbeziehungen

Zur konkreten Umsetzung der lokalen Steuerung der Hypothesendichte muss eine geeignete Nachbarschaftsbeziehung im Hypothesenraum definiert werden. Die Nachbarn

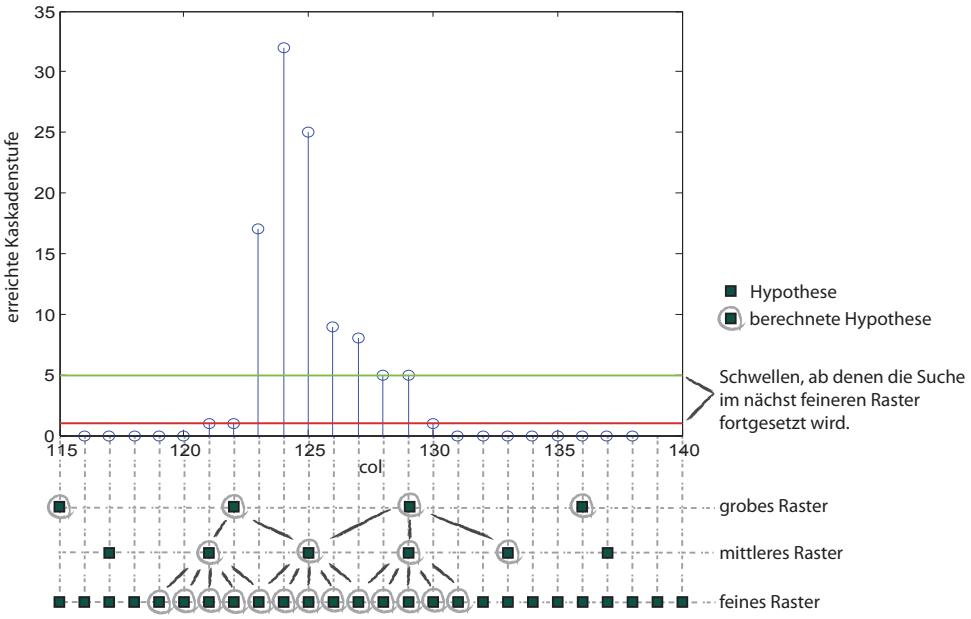


Abbildung 4.10.: Lokale Verfeinerung des Suchrasters (eindimensional). Dargestellt ist die Detektorantwort (erreichte Kaskadenstufe) einer Zeile aus Abbildung 4.9. Die Hypothesen der unterschiedlichen Rasterdichten sind unten als Rechtecke dargestellt. Alle tatsächlich berechneten Hypothesen sind mit einem Kreis markiert. Zunächst wird die Detektorantwort aller Hypothesen des gröbsten Rasters berechnet. Nur wenn deren erreichte Kaskadenstufe eine bestimmte Schwelle erreichen, wird in der Nachbarschaft im nächst feineren Raster weiter gesucht.

in $\mathcal{H}^{(l+1)}$ einer Hypothese $\left(\underset{\sim}{\text{col}}, \underset{\sim}{\text{row}}, \underset{\sim}{h}\right) \in \mathcal{H}^{(l)}$ sind für $\delta > 0$ gegeben durch

$$\begin{aligned}\Big\{(\text{col}', \text{row}', h') \in \mathcal{H}^{(l+1)} \quad \Big| \quad & (\text{col}, \text{row}, h) = \chi^{-1}\left(\text{col}', \text{row}', h'\right) \\ & \wedge \left|\text{col} - \underset{\sim}{\text{col}}\right| \leq \max\left(\left\lceil \delta Q^{(l)}_{\text{col}} \underset{\sim}{h} \right\rceil, \left\lceil Q^{(l+1)}_{\text{col}} \underset{\sim}{h} \right\rceil\right) \\ & \wedge \left|\text{row} - \underset{\sim}{\text{row}}\right| \leq \max\left(\left\lceil \delta Q^{(l)}_{\text{row}} \underset{\sim}{h} \right\rceil, \left\lceil Q^{(l+1)}_{\text{row}} \underset{\sim}{h} \right\rceil\right) \\ & \wedge \left\lfloor \underset{\sim}{h}\left(1 + Q^{(l)}_{h}\right)^{-\delta} \right\rfloor \leq h \leq \left\lceil \underset{\sim}{h}\left(1 + Q^{(l)}_{h}\right)^{\delta} \right\rceil \Big\},\end{aligned} \tag{4.5}$$

d.h. die Schachbrettdistanz wird auf normierten Positionen der jeweiligen Objektfenster angewandt, um so die Skalierungsinvarianz der Nachbarschaftsbeziehung sicher zu stellen (Abbildung 4.11). Es stellt mit $\delta \geq 0.5$ sicher, dass innerhalb derselben Skalierung jede Hypothese aus $\mathcal{H}^{(l+1)}$ auch mindestens einen Nachbarn in $\mathcal{H}^{(l)}$ derselben Skalierung hat. Darüber hinaus gibt es für jede Hypothese aus $\mathcal{H}^{(l)}$ mindestens einen linken sowie einen rechten (bzw. oberen und unteren) Nachbarn in $\mathcal{H}^{(l+1)}$. Die Überlappung zwischen den Nachbarschaften nimmt mit großem δ weiter zu. In dieser Arbeit wird $\delta = 0.75$ gewählt.

Im Fusionsfall gilt in jedem Strom ein mehrdimenionales Nachbarschaftskriterium. Für benachbarte Multi-Sensor Hypothesen muss die Nachbarschaftsbedingung (4.5) in allen Sensoren erfüllt sein.

Hypothesenbaum

Anstatt die Nachbarschaftsbeziehungen online für jede Hypothese umzusetzen, können dieselben Beziehungen effektiv vorab in einem statischen Baum abgebildet werden. Dazu werden alle Hypothesenmengen der unterschiedlichen Rasterdichten erzeugt und über Kanten entsprechend der Nachbarschaftsbeziehungen verknüpft (siehe Abbildung 4.12). Jede Ebene im resultierenden Baum entspricht damit einer vollständigen Hypothesenmenge in einer spezifischen Rasterdichte.

Bei der Anwendung des Detektors wird der Baum beginnend mit der Wurzelmenge rekursiv abgearbeitet (Tiefensuche). Der Abstieg zu den Kindknoten erfolgt dabei immer dann, wenn das entsprechende Schwellenkriterium erfüllt ist. Im Falle einer Detektion kann die Anzahl der zu prüfenden Hypothesen mit dem Backtracking genannten Verfahren weiter vermindert werden. Dabei wird die Suche im Baum abgebrochen und bei der nächsten Baumwurzel fortgesetzt (illustriert in Abbildung 4.13). Um dabei keinen systematischen Fehler zu erzeugen, werden bei der Erstellung eines Hypothesenbaumes alle Kindknoten zufällig permutiert.

Der Hypothesenbaum wird offline erstellt, d.h. die Hypothesen der verschiedenen Rasterdichten, sowie die Verbindungen zwischen den einzelnen Ebenen werden vorab berechnet und in einer Datei abgelegt. In der eigentlichen Anwendung wird der Baum dann geladen und entsprechend seiner vorberechneten Verknüpfungen abgearbeitet. Über einfache Caching-Mechanismen kann außerdem sichergestellt werden, dass jede Hypothese maximal einmal pro Bild (bzw. Bildtupel) berechnet wird. Der

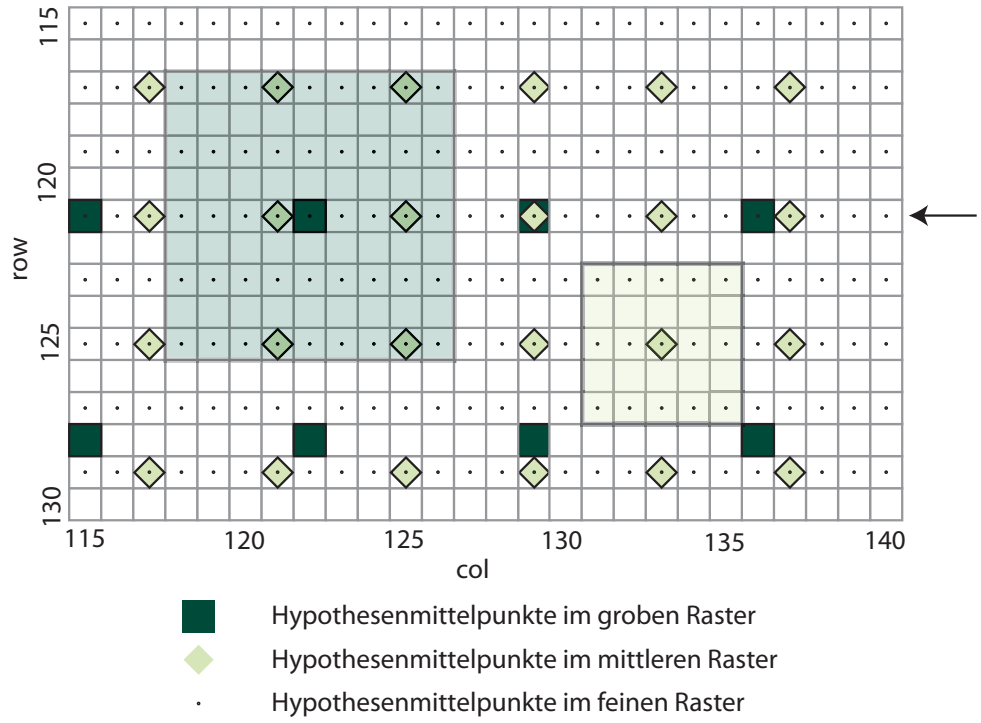


Abbildung 4.11.: Nachbarschaft im Hypothesenbaum. Dargestellt sind Hypothesenmittelpunkte von Hypothesen der festen Skalierung $h = 32\,\text{px}$ der Objektfenster in einem Pixelraster. Für das grobe Raster $\mathcal{H}^{(1)}$ wurde $Q^{(1)}_{\text{col}} = Q^{(1)}_{\text{row}} = 0.2$, für das mittlere Raster $\mathcal{H}^{(2)}$ wurde $Q^{(2)}_{\text{col}} = Q^{(2)}_{\text{row}} = 0.1$ und für das feinste Raster $\mathcal{H}^{(3)}$ wurde $Q^{(3)}_{\text{col}} = 0.03$, $Q^{(3)}_{\text{row}} = 0.05$ und $Q^{(3)}_{h} = 0.08$ gewählt. Beispielhaft ist links die Nachbarschaft in $\mathcal{H}^{(2)}$ zu einer Hypothese aus $\mathcal{H}^{(1)}$ und rechts die Nachbarschaft in $\mathcal{H}^{(3)}$ zu einer Hypothese aus $\mathcal{H}^{(3)}$ farblich hinterlegt. Die mit einem Pfeil markierte Zeile entspricht der aus Abbildung 4.10.

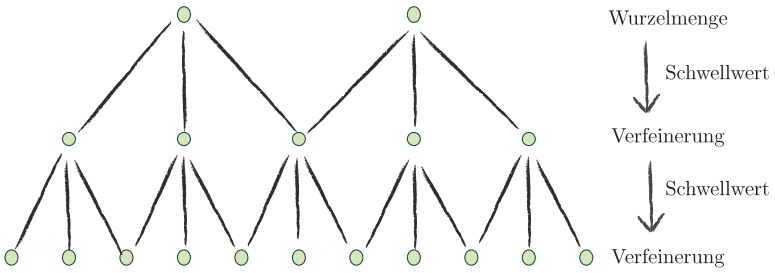


Abbildung 4.12.: Hypothesenbaum. Jede Ebene entspricht einer vollständigen Hypothesenmenge, wobei die Rasterdichten jeweils mit der Tiefe des Baumes zunehmen. Die Kanten entsprechen der Nachbarschaftsbeziehung (4.5). Bei der Anwendung des Detektors wird der Baum in einer Tiefensuche durchlaufen. Der Abstieg in die Kindknoten erfolgt dabei nur, wenn die Hypothese im Vaterknoten die entsprechende Detektionsschwelle erreicht hat.

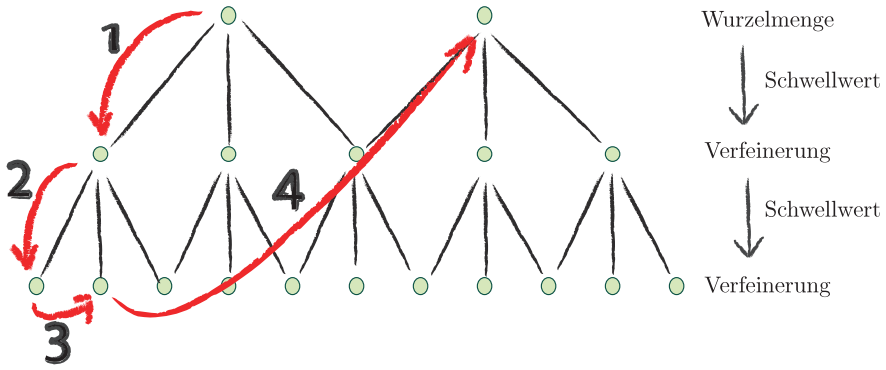


Abbildung 4.13.: Backtracking im Hypothesenbaum. Im Falle einer Detektion (Schritt 3 im Bild) wird die Suche im Baum abgebrochen und bei der nächsten Baumwurzel (Schritt 4 im Bild) fortgesetzt. Dadurch kann die Anzahl der zu prüfenden Hypothesen noch weiter reduziert werden.

Hypothesenbaum stellt also eine effiziente Umsetzung der grob-zu-fein Suche im Bild dar.

Zur Parametrisierung der unterschiedlichen Rasterdichten und Schwellen wird die Kaskade mit Beispielen unterschiedlicher Rasterdichten wie in Abbildung 4.8 evaluiert. Für die Rasterdichten aus Abbildung 4.8 ergeben sich für den NIR-Detektor:

$$\begin{aligned}
\mathcal{H}^{(1)} &= \mathcal{H}\left(0.3, 0.3, 0.3\right), \quad k^{(1)} = 2 \\
\mathcal{H}^{(2)} &= \mathcal{H}\left(0.1, 0.2, 0.3\right), \quad k^{(2)} = 5 \\
\mathcal{H}^{(3)} &= \mathcal{H}\left(0.1, 0.1, 0.1\right), \quad k^{(3)} = 9 \\
\mathcal{H}^{(4)} &= \mathcal{H}\left(0.03, 0.05, 0.08\right).
\end{aligned}$$

In der gröbsten Ebene hat dieser Hypothesenbaum 4 733 Hypothesen. Die feinste Ebene entspricht dem einfachen Hypothesengenerator aus Kapitel 4.1 mit 676 088 Hypothesen. Insgesamt hat der Baum 804 513 Knoten und 6 638 973 Kanten. Im Mittel werden damit 26 884 Hypothesen berechnet. Dies entspricht einer Reduzierung der Hypothesenanzahl gegenüber dem einfachen Hypothesengenerator um 96% (zur Evaluierung siehe auch Kapitel 7.3).

KAPITEL 5

Probabilistische Zustandsschätzung

Zur Fußgängererkennung werden Hypothesen (also Tupel von Suchfenstern im Falle der Fußgängererkennung mit mehr als einem Bildsensor) in Größe und Positition im Suchraum variiert und vom Kaskadenklassifikator ausgewertet. Die Objektlokalisierung folgt damit einem top-down Verfahren, indem Objekt- bzw. Zustandshypothesen (in diesem Fall die Suchfenster bzw. Suchfensterpaare) anhand von Messungen (also dem Ergebnis des Klassifikators) validiert werden. Bei der Verwendung einer der Hypothesengeneratoren aus den vorangegangenen Kapiteln wird unabhängig vom Ergebnis des letzten Bildes die Hypothesenmenge bei jedem Bild komplett neu ausgewertet. Die Hypothesen können jedoch auch auf Basis eines a priori Modells der möglichen bzw. wahrscheinlichen Fußgängerhypothesen abhängig vom vorangegangenen Zeitschritt vollzogen werden. Insbesondere können Hypothesen dann auch mit Hilfe eines Bewegungsmodells dynamisch über ganze Bildfolgen hinweg fortgeschaltet werden. Die Basis dafür bilden Partikelfilter, wie sie in Abschnitt 5.2 beschrieben werden. Eine konkrete Umsetzung ist in Abschnitt 5.3 dann der Condensation-Algorithmus, der schließlich in Kapitel 6 auch als Hypothesengenerator zur Fußgängererkennung eingesetzt wird.

Allgemein modelliert im Kontext der Fußgängererkennung der Systemzustand die Position, Größe und Geschwindigkeit und ein dynamisches Systemmodell die Bewegung der Fußgängers, also die Prädiktion des Systemzustandes von einem Zeitschritt zum nächsten. Das Beobachtungsmodell wird - zusammen mit der Annahme, dass das Abbild eines Fußgängers im Bild durch einen genügend großen rechteckigen Bereich (eben dem Suchfenster) beschränkt ist - durch die Klassifikationskomponente bestimmt. Alle in dieser Weise modellierten Zustandsschätzer folgen dem allgemeinen Prinzip des Bayesschen Trackings, als Grundlage kurz dargelegt im folgenden Abschnitt 5.1.

Solche Zustandsschätzer beschreiben in der Regel genau ein Ziel, in der Praxis müssen jedoch auch mehrere Fußgänger in einem Zeitschritt detektiert und verfolgt wer-

den können. Diese Problematik der probabilistischen Mehrobjektverfolgung sowie ein Lösungsvorschlag für den Anwendungsfall dieser Arbeit wird in Abschnitt 5.4 aufgezeigt. Erst damit sind die Voraussetzungen für den in Kapitel 6 beschriebenden Fußgängerdetektor auf Basis von Partikelfiltern gegeben.

5.1. Bayessches Tracking

Die Aufgabe eines Zustandsschätzers besteht allgemein darin, einen Zustand $\boldsymbol{x}_i \in \mathbb{R}^{N_x}$ aus Messdaten $\boldsymbol{z}_i \in \mathbb{R}^{N_z}$ zu schätzen, die zum Zeitpunkt i aus der Umgebung vorliegen und beobachtet werden können [Den04]. Dazu wird die Fahrzeugumgebung als ein dynamisches physikalisches System aufgefasst, in dem der Zustandsvektor $\boldsymbol{x}_i \in \mathbb{R}^{N_x}$ formal einen nicht-stationären homogenen zeitdiskretisierten Markovprozess erster Ordnung modelliert [BSLK01]. Im Kontext der Fußgängererkennung steht der Systemzustand für die Position des Fußgängers. Die Messdaten entsprechen dann den Positionen der vom Klassifikator als Fußgänger klassifizierten Hypothesen.

Die zeitliche Veränderung des Zustandsvektors wird durch ein dynamisches Systemmodell beschrieben:

$$\boldsymbol{x}_i = \boldsymbol{f}_i\left(\boldsymbol{x}_{1:i-1}, \boldsymbol{v}_{i-1}\right), \tag{5.1}$$

mit $\boldsymbol{x}_{1:i-1} := \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}$ die Menge aller vorangegangenen Zustände. Der Vektor $\boldsymbol{v}_{i-1} \in \mathbb{R}^{N_v}$ repräsentiert das System- oder Prozessrauschen, um Unsicherheiten und Ungenauigkeiten bzgl. des Systemmodells explizit mit einzubeziehen. Da der Zustandsschätzer einen Markov-Prozess erster Ordnung abbilden soll, ist (5.1) gleichbedeutend mit

$$\boldsymbol{x}_i = \boldsymbol{f}_i\left(\boldsymbol{x}_{i-1}, \boldsymbol{v}_{i-1}\right), \tag{5.2}$$

d.h. der Zustand umfasst die Vergangenheit des Systems hinreichend genau, so dass zukünftige Zustände lediglich aus dem aktuellen Zustand berechnet werden können (Markov-Eigenschaft, [MT93]).

Neben dem Systemmodell beschreibt

$$\boldsymbol{z}_i = \boldsymbol{h}_i\left(\boldsymbol{x}_i, \boldsymbol{r}_i\right) \tag{5.3}$$

formal das Beobachtungsmodell, also den Zusammenhang zwischen dem Systemzustand zum Zeitpunkt i und der entsprechenden Sensorantwort in Form des Messvektors $\boldsymbol{z}_i \in \mathbb{R}^{N_z}$. Entsprechend wird mit $\boldsymbol{r}_i \in \mathbb{R}^{N_r}$ das Mess- bzw. Beobachtungsrauschen mit berücksichtigt. Die beiden Rauschvektoren $\boldsymbol{v}_{i-1}$ und $\boldsymbol{r}_i$ werden dabei als statistisch unabhängig und mittelwertfrei vorausgesetzt.

Allgemein ist der Zustandsschätzer zur Schätzung des Zustandes eines dynamischen Systems eine Funktion $\hat{\boldsymbol{x}}_i\left(\hat{\boldsymbol{x}}_{1:i-1}, \boldsymbol{z}_{1:i}\right)$, die zum Zeitpunkt i einen Schätzwert $\hat{\boldsymbol{x}}_i$ des wahren Zustands $\boldsymbol{x}_i$ liefert [Den04]. Der Schätzwert basiert dabei auf der Folge der Beobachtungen $\boldsymbol{z}_{1:i}$, sowie der Folge der vorangegangenen Zustandsschätzwerte $\hat{\boldsymbol{x}}_{1:i-1}$. Da der jeweilige Schätzwert $\hat{\boldsymbol{x}}_i$ zu jedem Zeitpunkt über (5.2) und (5.3) von den

Rauschgrößen $\boldsymbol{v}_{i-1}$ und $\boldsymbol{r}_i$ abhängt ist $\hat{\boldsymbol{x}}_i$ eine Zufallsgröße. Ebenso ist der wahre Zustand $\boldsymbol{x}_i$ selbst ein Zufallsvektor mit der Wahrscheinlichkeitsdichte

$$p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i}). \tag{5.4}$$

Analog kann mit der Zustandsübergangsfunktion (5.2) eine bedingte Dichte

$$p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})$$

assoziiert werden. Diese bedingte Dichte kann in vielen Fällen analytisch aus der Zustandsübergangsfunktion abgeleitet werden: Geht man z.B. von einem zeitinvarianten zeitdiskreten linearen stochastischem System aus, ist $\boldsymbol{f}_i$ beschrieben durch einen Satz von stochastischen Differenzengleichungen der Form

$$\boldsymbol{x}_i = \boldsymbol{F}\boldsymbol{x}_{i-1} + \boldsymbol{v}, \quad \boldsymbol{v} \sim \mathcal{N}(\boldsymbol{o}, \boldsymbol{\Sigma}).$$

In diesem Fall kann dem Systemmodell eine bedingte Wahrscheinlichkeit der Form

$$p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}) = (2\pi)^{-\frac{N_x}{2}} \left|\boldsymbol{\Sigma}^{-\frac{1}{2}}\right| \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{F}\boldsymbol{x}_{i-1})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{F}\boldsymbol{x}_{i-1})\right)$$

zugeordnet werden (siehe z.B. [Bis06]).

Zur Zustandsschätzung muss also die Wahrscheinlichkeitsdichte (5.4) geschätzt werden. Mit der bayesschen Regel und der Markov-Eigenschaft $p(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{z}_{1:i-1}) = p(\boldsymbol{z}_i|\boldsymbol{x}_i)$, d.h. die aktuelle Messung $\boldsymbol{z}_i$ ist nur vom aktuellen Zustand $\boldsymbol{x}_i$ abhängig, ergibt sich

$$\begin{aligned} p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i}) &= \frac{p(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{z}_{1:i-1})p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i-1})}{p(\boldsymbol{z}_i|\boldsymbol{z}_{1:i-1})} \\ &= \frac{p(\boldsymbol{z}_i|\boldsymbol{x}_i)p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i-1})}{p(\boldsymbol{z}_i|\boldsymbol{z}_{1:i-1})}. \end{aligned} \tag{5.5}$$

Die Normalisierung $p(\boldsymbol{z}_i|\boldsymbol{z}_{1:i-1})$ steht in Abhängigkeit zum Beobachtungsmodell und wird im Folgenden durch η_i^{-1} abgekürzt:

$$\eta_i^{-1} := p(\boldsymbol{z}_i|\boldsymbol{z}_{1:i-1}) = \int p(\boldsymbol{z}_i|\boldsymbol{x}_i)p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i-1})d\boldsymbol{x}_i.$$

Eine wichtige Eigenschaft von Markov-Prozessen ist die Chapman-Kolmogorov-Gleichung (siehe z.B. [Pap91]):

$$p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i-1}) = \int p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})p(\boldsymbol{x}_{i-1}|\boldsymbol{z}_{1:i-1})d\boldsymbol{x}_{i-1}.$$

Sie bestimmt die Randverteilung des Markov-Prozesses und führt so die a posteriori Verteilung $p(\boldsymbol{x}_{i-1}|\boldsymbol{z}_{1:i-1})$ des vorangegangenen Zeitpunktes $i-1$ über in die a priori Verteilung $p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i-1})$ zum aktuellen Zeitpunkt i. Damit kann (5.5) weiter umgeformt werden zu

$$p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i}) = \eta_i \underbrace{p(\boldsymbol{z}_i|\boldsymbol{x}_i)}_{\text{Beobachtungsmodell}} \int \underbrace{p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})}_{\text{Systemmodell}} p(\boldsymbol{x}_{i-1}|\boldsymbol{z}_{1:i-1})d\boldsymbol{x}_{i-1}. \tag{5.6}$$

Dies ist die im Allgemeinen als Bayesfilter bezeichnete Gleichung zur rekursiven Abschätzung des aktuellen Systemzustandes. Algorithmisch läuft er in zwei Schritten ab:

$$\begin{aligned} &\text{1. Prädiktion:} && \overbrace{p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i-1})}^{\text{a priori}} = \int \overbrace{p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})}^{\text{Systemmodell}} p(\boldsymbol{x}_{i-1}|\boldsymbol{z}_{1:i-1}) d\boldsymbol{x}_{i-1}, \\ &\text{2. Innovation:} && \underbrace{p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})}_{\text{a posteriori}} = \eta_i \underbrace{p(\boldsymbol{z}_i|\boldsymbol{x}_i)}_{\text{Beobachtungsmodell}} p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i-1}). \end{aligned} \tag{5.7}$$

Zu jedem Zeitpunkt wird also über das dynamische Systemmodell $p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})$ eine Vorhersage über den aktuellen Zustand des Systems getroffen (Prädiktion). Beim Eintreffen neuer Sensordaten wird im zweiten Schritt auf Basis dieser prädizierten a priori Dichte mit Hilfe der Beobachtungswahrscheinlichkeit $p(\boldsymbol{z}_i|\boldsymbol{x}_i)$ die a posteriori Dichte zum Zeitpunkt nach der Beobachtung $\boldsymbol{z}_i$ bestimmt (Innovation).

Der eigentliche Schätzwert kann dann z.B. durch den MAP-Schätzer (engl. „maximum a posteriori“)

$$\hat{\boldsymbol{x}}_i^{\text{MAP}} = \arg\max_{\boldsymbol{x}_i} p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i}) = \arg\max_{\boldsymbol{x}_i} p(\boldsymbol{z}_i|\boldsymbol{x}_i) p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i-1}), \tag{5.8}$$

oder durch den MMSE-Schätzer (engl. „minimum mean square error“)

$$\hat{\boldsymbol{x}}_i^{\text{MMSE}} = \mathbb{E}\left[\boldsymbol{x}_i \,|\boldsymbol{z}_{1:i}\right] = \int \boldsymbol{x}_i p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i}) d\boldsymbol{x}_i$$

bestimmt werden.

5.2. Der Partikel-Filter

Die Grundidee von sequentiellen Monte-Carlo-Methoden ist, die Wahrscheinlichkeitsdichte $p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$ durch eine Menge von gewichteten Beispielen (Samples, Partikel) $\xi_i^{(j)} = (\boldsymbol{x}_i^{(j)}, w_i^{(j)})$, $j = 1, \ldots, N_s$ mit $\sum_{j=1}^{N_s} w_i^{(j)} = 1$ darzustellen und über die Zeit hinweg fortzuschalten. Jede dieser Partikelmengen $\Xi_i = \{\xi_i^{(1)}, \ldots, \xi_i^{(N_s)}\}$ repräsentiert dann eine sogenannte empirische Dichte [Den04, S. 83] mit

$$p_{\Xi_i}(\boldsymbol{x}) = \sum_{j=1}^{N_s} w_i^{(j)} \delta(\boldsymbol{x} - \boldsymbol{x}_i^{(j)}), \tag{5.9}$$

mit $\delta(\cdot)$ als Diracfunktion. Die Partikelmenge soll dabei die gewünschte Dichtefunktion in der Art und Weise repräsentieren, dass die Auswahl eines $\boldsymbol{x}_i^{(j)}$ mit der Wahrscheinlichkeit $w_i^{(j)}$ in etwa dem Ziehen einer zufälligen Probe aus der echten Verteilung von $\boldsymbol{x}$ gleich zu setzten ist [Mac00] (vgl. auch Abbildung 5.1).

In der Regel ist das effiziente generieren zufälliger Stichprobenelemente $\boldsymbol{x}_i^{(j)}$ aus $p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$ nicht möglich, da dazu die Dichtefunktion in analytischer Form vorliegen muss. In der

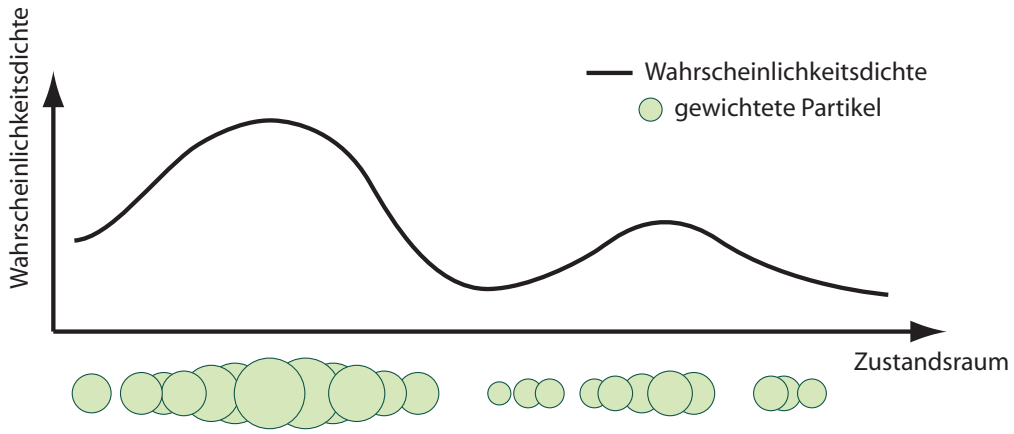


Abbildung 5.1.: Approximation einer Wahrscheinlichkeitsdichte durch gewichtete Partikel.

Praxis wird deshalb zur Realisierung ein Verfahren aus dem Bereich der sequentiellen Monte-Carlo-Simulationen benutzt: Die Samples und Gewichte in (5.9) können nämlich anhand des Verfahrens der gewichteten Stichprobenentnahme (engl. „importance sampling“, [Dou98]) bestimmt werden.

Dazu erzeugt man Beispiele nicht aus $p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$, sondern aus einer Vorschlagsfunktion $p_{\text{prop}}(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$ (engl. „importance function“ oder „proposal distribution“), von der es konstruktionsbedingt möglich sein muss Beispiele zu erzeugen. Allgemein gilt dann

$$p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i}) \propto w_i(\boldsymbol{x}_i)\, p_{\text{prop}}(\boldsymbol{x}_i|\boldsymbol{z}_{1:i}).$$

$\propto$ bezeichnet dabei die Proportionalität. Die sogenannten Einflussgewichte $w_i(\boldsymbol{x}_i)$ sind definiert durch

$$w_i(\boldsymbol{x}_i) = \frac{p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})}{p_{\text{prop}}(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})}. \tag{5.10}$$

Wurde also ein Sample $\boldsymbol{x}_i^{(j)}$ entsprechend $p_{\text{prop}}(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$ ausgewählt, gibt $w_i(\boldsymbol{x}_i^{(j)})$ die Wahrscheinlichkeit an, dass dieses Sample zufällig aus $p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$ gezogen wurde. Ist die Wahrscheinlichkeit an der Selle $\boldsymbol{x}_i^{(j)}$ anhand der Vorschlagsfunktion ein Beispiel zu generieren höher, als es nach der wahren Dichte sein sollte, so ist $w_i(\boldsymbol{x}_i^{(j)})$ entsprechend kleiner. Anschaulich hat $w_i(\boldsymbol{x}_i^{(j)})$ die Funktion einer Akzeptanzwahrscheinlichkeit [Sch06], und kann dazu verwendet werden zu entscheiden, ob $\boldsymbol{x}_i^{(j)}$ als ein Sample von $p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$ angesehen werden kann, oder nicht.

Wichtig ist noch zu bemerken, dass zur Bestimmung der Gewichte in (5.10) $p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$ nicht in analytischer Form vorliegen muss, sondern nur an den Stellen $\boldsymbol{x}_i^{(j)}$ auswertbar sein muss. In den meisten Fällen ist dabei die Normalisierungskonstante von $p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$

unbekannt, so dass die Gewichte durch

$$w_i^{(j)} = \frac{w_i(\boldsymbol{x}_i^{(j)})}{\sum_{\nu=1}^{N_s} w_i(\boldsymbol{x}_i^{(\nu)})}$$

definiert sind. Die a posteriori Wahrscheinlichkeit kann dann also zu jedem Zeitschritt nach (5.9) approximiert werden:

$$p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i}) \approx \sum_{j=1}^{N_s} w_i^{(j)} \delta(\boldsymbol{x}_i - \boldsymbol{x}_i^{(j)}). \tag{5.11}$$

Bei der sequenziellen Stichprobenentnahme (engl. „sequential importance sampling“) werden die Gewichte nicht in jedem Zeitschritt neu bestimmt, sondern sequentiell fortgeschaltet. Die Vorschlagsfunktion $p_{\text{prop}}(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})$ wird so modelliert, dass sie mittels

$$p_{\text{prop}}(\boldsymbol{x}_i^{(j)}|\boldsymbol{z}_{1:i}) = p_{\text{prop}}(\boldsymbol{x}_{i-1}^{(j)}|\boldsymbol{z}_{1:i-1}) \cdot p_{\text{prop}}(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{i-1}^{(j)}, \boldsymbol{z}_{1:i}) \tag{5.12}$$

faktorisiert werden kann. Durch Anwendung der Bayesrekursion aus (5.6) kann (5.10) in die Form

$$w_i\left(\boldsymbol{x}_i\right) \propto \frac{p(\boldsymbol{z}_i|\boldsymbol{x}_i) \int p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}) p(\boldsymbol{x}_{i-1}|\boldsymbol{z}_{1:i-1}) d\boldsymbol{x}_{i-1}}{p_{\text{prop}}(\boldsymbol{x}_i|\boldsymbol{z}_{1:i})} \tag{5.13}$$

gebracht werden. Aus dem vorangegangenen Zeitschritt ist die Partikelmenge $\Xi_{i-1} = \{(\boldsymbol{x}_{i-1}^{(1)}, w_{i-1}^{(1)}), \ldots, (\boldsymbol{x}_{i-1}^{(N_s)}, w_{i-1}^{(N_s)})\}$ vorhanden, die $p(\boldsymbol{x}_{i-1}|\boldsymbol{z}_{1:i-1})$ anhand (5.11) approximiert, d.h.

$$p(\boldsymbol{x}_{i-1}|\boldsymbol{z}_{1:i-1}) \approx \sum_{j=1}^{N_s} w_{i-1}^{(j)} \delta(\boldsymbol{x}_{i-1} - \boldsymbol{x}_{i-1}^{(j)}). \tag{5.14}$$

Substituiert man (5.14) in (5.13) kann mit (5.12) für jedes *einzelne* Partikel die Rekursion

$$\begin{aligned} w_i^{(j)} &\propto \frac{p(\boldsymbol{z}_i|\boldsymbol{x}_i^{(j)}) p(\boldsymbol{x}_i^{(j)}|\boldsymbol{x}_{i-1}^{(j)}) p(\boldsymbol{x}_{i-1}^{(j)}|\boldsymbol{z}_{1:i-1})}{p_{\text{prop}}(\boldsymbol{x}_{i-1}^{(j)}|\boldsymbol{z}_{1:i-1}) p_{\text{prop}}(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{i-1}^{(j)}, \boldsymbol{z}_{1:i})} \\ &\overset{(5.10)}{=} w_{i-1}^{(j)} \cdot \frac{p(\boldsymbol{z}_i|\boldsymbol{x}_i^{(j)}) p(\boldsymbol{x}_i^{(j)}|\boldsymbol{x}_{i-1}^{(j)})}{p_{\text{prop}}(\boldsymbol{x}_i^{(j)}|\boldsymbol{x}_{i-1}^{(j)}, \boldsymbol{z}_{1:i})} \end{aligned} \tag{5.15}$$

aufgestellt werden (siehe z.B. [Vih04, S. 34ff], [Fre99, S. 105] oder [Hau05, S. 17]). Damit können die Einflussgewichte rekursiv über die Zeit hinweg bestimmt werden.

Wählt man als Vorschlagsfunktion

$$p_{\text{prop}}(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}, \boldsymbol{z}_{1:i}) = p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}),$$

die den Zustandsübergang des dynamischen Systems repräsentiert, vereinfacht sich (5.15) zu

$$w_i^{(j)} \propto w_{i-1}^{(j)} \cdot p(\boldsymbol{z}_i|\boldsymbol{x}_i^{(j)}). \tag{5.16}$$

Obwohl diese Wahl eher suboptimal ist, da damit $p_{\text{prop}}(\boldsymbol{x}_i^{(j)}|\boldsymbol{x}_{i-1}^{(j)}, \boldsymbol{z}_{1:i})$ die Sensordaten unberücksichtigt lässt, wird diese Vorschlagsfunktion auf Grund ihrer Einfachheit und leichten Realisierbarkeit in der Praxis häufig verwendet. Sie bildet auch die Grundlage für die in dieser Arbeit verwendeten Realisierung, nämlich den in Abschnitt 5.3 beschriebenen Condensation Algorithmus.

Die durch (5.15) definierte rekursive Bestimmung der Einflussgewichte ist in Algorithmus 5.1 in Form des SIS-Filters (engl. „Sequence Importance Sampling"-Algorithmus) nochmals zusammengefasst. Er bildet die Basis der meisten Partikelfilterverfahren, hat aber in dieser Form den Nachteil der Degeneration. Das bedeutet, dass nach wenigen Zeitschritten nur ein Partikel ein Gewicht nahe dem Wert Eins besitzt, alle anderen aber Gewichte nahe Null haben. [Dou98] zeigt, dass sich dieses Phänomen (engl. „sample impoverishment" oder „sample degeneracy") auch nicht verhindern lässt. Um aber dieser unerwünschten Eigenschaft des SIS-Filters entgegenzuwirken, eleminiert

1. Zum Zeitpunkt $i = 0$: Initialisiere die Partikelmenge $\Xi_0 = \{(\boldsymbol{x}_0^{(0)}, w_0^{(0)}), \ldots, (\boldsymbol{x}_0^{(N_s)}, w_0^{(N_s)})\}$ anhand der a priori Dichte $p(\boldsymbol{x}_0)$.

2. Zum Zeitpunt $i \geq 1$:

   (a) Für $k = 1, \ldots, N_s$: Sample $\boldsymbol{x}_i^{(j)} \sim p_{\text{prop}}(\boldsymbol{x}_i|\boldsymbol{x}_i^{(j)}, \boldsymbol{z}_i)$.

   (b) Für $k = 1, \ldots, N_s$: Bestimme die unnormalisierten Einflussgewichte mittels

$$\bar{w}_i^{(j)} = w_{i-1}^{(j)} \cdot \frac{p(\boldsymbol{z}_i|\boldsymbol{x}_i^{(j)}) p(\boldsymbol{x}_i^{(j)}|\boldsymbol{x}_{i-1}^{(j)})}{p_{\text{prop}}(\boldsymbol{x}_i^{(j)}|\boldsymbol{x}_{i-1}^{(j)}, \boldsymbol{z}_{1:i})}.$$

   (c) Für $j = 1, \ldots, N_s$: Normalisiere die Einflussgewichte:

$$w_i^{(j)} = \frac{\bar{w}_i^{(j)}}{\sum_{\nu=1}^{N_s} \bar{w}_i^{(\nu)}}$$

Algorithmus 5.1: SIS-Algorithmus (engl. „sequence importance sampling").

man in der Praxis Partikel mit geringem Gewicht und vervielfältigt solche mit hohen Gewichten, d.h. man führt eine wiederholte Stichprobenentnahme (Resampling) durch. Dabei wird eine neue Partikelmenge $\tilde{\Xi}_i^{(j)} = \{(\tilde{\boldsymbol{x}}_i^{(j)}, \tilde{w}_i^{(j)}), \ldots, (\tilde{\boldsymbol{x}}_i^{(N_s)}, \tilde{w}_i^{(N_s)})\}$ erzeugt, indem auf Basis von

$$\mathbb{P}\left(\tilde{\boldsymbol{x}}_i^{(j)} = \boldsymbol{x}_i^{(\nu)}\right) = w_i^{(\nu)}, \quad \forall\, \nu = 1, \ldots, N_s$$

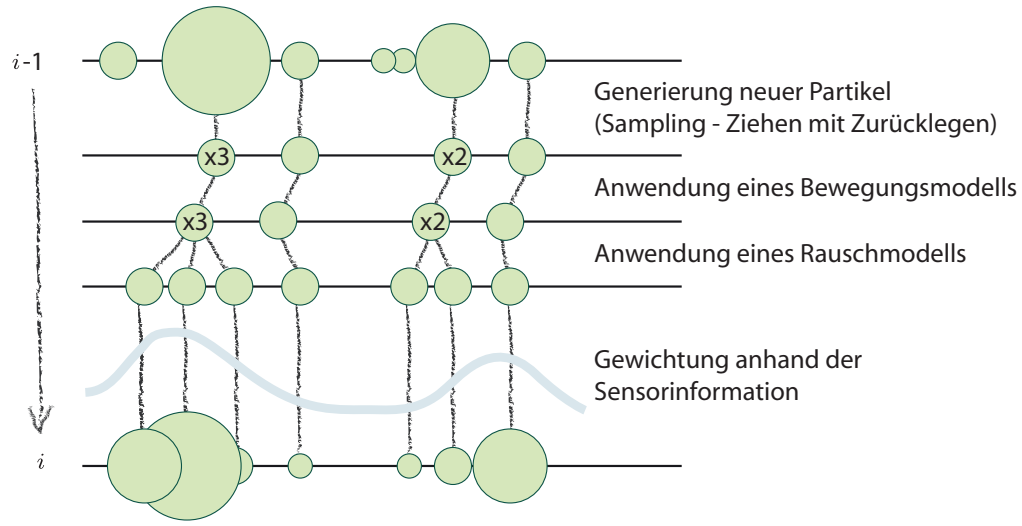


Abbildung 5.2.: Ein Iterationsschritt des Condensation-Algorithmus vom Zeitpunkt $i-1$ zum Zeitpunkt i. Die Gewichte der Partikel sind jeweils durch unterschiedliche große Kreise dargestellt. Partikel, die im Sampling-Schritt beim „Ziehen mit Zurücklegen“ mehrmals ausgewählt wurden, sind mit einem Multiplikator gekennzeichnet.

Beispiele $\tilde{\boldsymbol{x}}_i^{(j)}$ durch Ziehen mit Zurücklegen erzeugt werden. Nach einem solchen Schritt sind die Partikel natürlich ungewichtet, so dass $\tilde{w}_i^{(j)} = {N_s}^{-1} \ \forall\, j$ gilt. In Verbindung mit dem SIS-Filter ergibt sich so der SIR-Filter (engl. „Sampling Importance Resampling“), der erstmals schon von [Rub88] vorgeschlagen wurde.

5.3. Der Condensation Algorithmus

Der Condensaion Algorithmus nach [IB98a] und [BI98] stellt eine Realisierung des SIR-Filters mit (5.16) als Vorschlagsfunktion dar. Der Ablauf ist in Algorithmus 5.2 zusammengefasst und in Abbildung 5.2 illustriert.

Im Ablauf gleicht er den in [GSS93] eingeführten Bayesschen Bootstrap-Filtern. In jedem Iterationsschritt liegt die gewichtete Partikelmenge Ξ_{i-1} aus der letzten Iteration vor (in Abbildung 5.2 dargestellt durch unterschiedlich große Kreise). Anhand der Gewichtung werden dann neue Samples $\tilde{\boldsymbol{x}}_{i-1}^{(j)}$ generiert (Resampling) und im Prädiktionsschritt anhand ihres Systemmodells im Zustandsraum verschoben (ergibt $\boldsymbol{x}_i^{(j)}$). Das dynamische Systemmodell ist dabei in eine deterministische und eine stochastische Komponente aufgeteilt. Die stochastische modelliert dabei das Zustandsübergangsrauschen. Durch die Anwendung beider Komponenten auf die Elemente der Partikelmenge $\tilde{\Xi}_{i-1}$ werden so Stichproben der Dichte $p(\boldsymbol{x}_i|\boldsymbol{z}_{1:i-1})$ erzeugt. Die Neugewichtung $w_i^{(j)}$ jedes Partikels erfolgt dann auf Grundlage der gegenwärtigen Beobachtung $\boldsymbol{z}_i$ durch Auswertung der Likelihood $p(\boldsymbol{z}_i|\boldsymbol{x}_i)$ und anschließender Normierung.

1. Zum Zeitpunkt $i = 0$:
   (a) Initialisiere die Partikelmenge $\Xi_0 = \{(\boldsymbol{x}_0^{(0)}, w_0^{(0)}), \ldots, (\boldsymbol{x}_0^{(N_s)}, w_0^{(N_s)})\}$ anhand der a priori Dichte $p(\boldsymbol{x}_0)$.
   (b) Initial werden die Gewichte gleichverteilt, d.h. $w_0^{(j)} = \frac{1}{N_s} \;\forall\; j$.
   (c) Initialisiere die kummulativen Wahrscheinlichkeiten durch $c_0^{(0)} = 0$ und für $j = 1, \ldots, N_s$: $c_0^{(j)} = c_0^{(j-1)} + w_0^{(j)}$.
2. Zum Zeitpunkt $i \geq 1$:
   (a) Resampling
   Für $j = 1, \ldots, N_s$: Generiere eine uniform verteilte Zufallszahl $r \in_{\mathrm{R}} [0, 1]$ und wähle $\tilde{\boldsymbol{x}}_{i-1}^{(j)} = \boldsymbol{x}_{i-1}^{(\nu)}$ mit dem kleinsten Index ν, für den gilt $c_{i-1}^{(\nu)} \geq r$ (binäre Suche).
   (b) Prädikion
   Für $j = 1, \ldots, N_s$: Wende das dynamische Systemmodell $p(\boldsymbol{x}_i^{(j)} | \tilde{\boldsymbol{x}}_{i-1}^{(j)})$ in Form eines deterministischen Drift und einer stochastischen Diffusion an.
   (c) Gewichtung
   Für $j = 1, \ldots, N_s$: Führe an der Stelle $\boldsymbol{x}_i^{(j)}$ im Zustandsraum die Messungen durch und ermittle die neuen unnormalisierten Einflussgewichte mittels $\bar{w}_i^{(j)} = p(\boldsymbol{z}_i | \boldsymbol{x}_i^{(j)})$.
   (d) Für $j = 1, \ldots, N_s$: Normalisiere die Gewichte:

$$w_i^{(j)} = \frac{\bar{w}_i^{(j)}}{\sum_{\nu=1}^{N_s} \bar{w}_i^{(\nu)}}.$$

   (e) Berechne die kummulativen Wahrscheinlichkeiten durch $c_i^{(0)} = 0$ und für $j = 1, \ldots, N_s$: $c_i^{(j)} = c_i^{(j-1)} + w_i^{(j)}$.

Algorithmus 5.2: Ablauf des Condensation-Algorithmus.

Bemerkenswert ist, dass in Schritt 2c von Algorithmus 5.2 die Gewichte aus dem Resampling-Schritt, anders als in (5.16), auf Grund der abschließenden Normierung nicht berücksichtigt werden müssen, da $\tilde{w}_i^{(j)} = {N_s}^{-1} \ \forall \ j$.

Im Condensation-Algorithmus muss die Likelihood-Funktion nicht in parametrischer Form vorliegen. Gerade deshalb ist der Condensation Algorithmus im Bereich des Rechnersehens so beliebt, da die Likelihood-Funktion stattdessen direkt aus der Bildinformation bzw. direkt aus dem Klassifikationsergebnis in Form von Rückschlusswahrscheinlichkeiten gewonnen werden kann. Unter Umständen erfolgt die Bewertung aber auch über eine heuristische Gewichtsfunktion $g(\boldsymbol{x}_i)$, die ein beliebiges Wahrscheinlichkeitsmaß darstellt. Damit lässt sich der Condensation-Algrithmus für viele Problemkreise adaptieren, wie z.B. [SNR05] und [ISP+06] im Falle der Detektion von Fahrzeugen und in [SSNR06] im Falle einer durch Digitaler Karten gestützter Straßenverlaufserkennung demonstriert wurde.

Dieses Vorgehen hat allerdings den Nachteil, dass eine solche generierte Likelihood-Funktion in den meisten Fällen nicht glatt ist, sondern viele isolierte Maxima aufweist und damit die Approximationsgenauigkeit des Filters abnimmt [Fea98]. Dem kann praktisch entgegengewirkt werden, indem z.B. ein gewisser Anteil von Samples zu jedem Zeitschritt unabhängig von der jeweiligen Messung gleichverteilt im Zustandsraum generiert wird. In vielen Fällen bleibt allerdings nur noch die Reinitialisierung des Filters.

Ein Vorteil des Condensation-Algorithmus dagegen ist, dass bei der Gewichtung der Partikel in sehr strukturierter Art und Weise mehrere (Bild-)Merkmale durch eine multiplikative Verknüpfung von Gewichtsfunktionen zu einer Likelihood verknüpft werden können:

$$g(\boldsymbol{x}_i) = \prod_{\nu=1}^{m} g_\nu(\boldsymbol{x}_i). \tag{5.17}$$

Die Bedeutung des Condensation-Algorithmus für diese Arbeit liegt vor allem in der Tatsache begründet, dass er eine Objektsuche wie bei der Fußgängerdetektion mit Hypothesengeneratoren durch ein Verifikationsproblem ersetzt, indem Zustandshypothesen im Bildraum validiert werden. Die Hypothesen „clustern“ sich dabei innerhalb von Bereichen mit einer hohen Wahrscheinlichkeit für das Vorhandensein eines Objektes [Loy03]. Der Condensation-Algorithmus stellt somit eine effiziente Umsetzung des top-down Paradigmas dar, indem Objekthypothesen zu jedem Zeitschritt validiert und über die Zeit hinweig fortgeschaltet werden.

Der Aufwand für den Condensation-Algorithmus selbst kann im Falle eines linearen Systemmodells mit Zustandsübergangsmarix $F \in \mathbb{R}^{n \times n}$ und der Verwendung von m Merkmalen in (5.17) mit $\mathcal{O}(N_s n^2 + N_s m)$ abgeschätzt werden [KM00]. $N_s n^2$ resultiert dabei aus der N_s-fachen Auswertung der Übergangsmatrix F, $N_s m$ entspricht der N_s-fachen Anwendung der m Merkmale. Die Komplexität des Resampling-Schrittes ist mit $\mathcal{O}(N_s \log N_s)$ beschränkt [GSS93]. Der Gesamtaufwand ist damit $\mathcal{O}(N_s n^2 + N_s m + N_s \log N_s)$.

Aus Geschwindigkeitsgründen wurde in der praktischen Umsetzung dieser Arbeit der stochastische Resampling-Schritt darüber hinaus durch eine deterministische Variante ersetzt. Für ein Partikel $\boldsymbol{x}_i^{(j)}$ mit dem normalisierten Gewicht $w_i^{(j)}$ werden dabei genau $\lfloor w_i^{(j)} \cdot N_s \rfloor$ Partikel $\tilde{\boldsymbol{x}}_i$ erzeugt. Lediglich die noch fehlenden $K = N_s - \sum \lfloor w_i^{(j)} \cdot N_s \rfloor$ Partikel werden mit Hilfe der kummulativen Gewichte stochastisch erzeugt. Einen Überblick über weitere Realisierungsmöglichkeiten des Resampling-Schrittes gibt [BMH04].

5.4. Probabilistische Mehrobjektverfolgung

Die Modellierungen der vorangegangenen probabilistischen Zustandsschätzer sind für die Detektion und Verfolgung genau eines einzelnen Fußgängers geeignet. In der Praxis müssen jedoch auch mehrere Fußgänger innerhalb der Szene korrekt detektiert und verfolgt werden können. Im Folgenden wird deshalb zunächst die Problematik der probabilistischen Mehrobjektverfolgung dargestellt und anschließend eine realisierbare Lösung für den Anwendungsfall dieser Arbeit beschrieben.

Sei Ω der Zustandsraum zur Detektion und Verfolgung genau eines Objektes. Dann ist

$$\Omega^M = \Omega_1 \cup \Omega_2 \cup \ldots \cup \Omega_M = \bigcup_{\nu=1}^{M} \Omega_\nu \tag{5.18}$$

der Zustandsraum für eine feste und bekannte Anzahl von M Objekten. Bei einer variablen Anzahl von Objekten müssen zusätzlich Existenz- oder Indikatorvariablen (wie z.B. in [VMB05]) berücksichtigt werden. Anschaulich beschränken diese dann den Zustandsraum auf die aktuell geschätzte Anzahl von Objekten (die aber zu jedem Zeitschritt in irgendeiner Weise bekannt sein muss).

Die praktische Realisierung durch vereinigte Zustandsräume gestaltet sich jedoch sehr schwierig, da der Zustandsraum zu groß wird. Eine Möglichkeit zur Reduktion des Berechnungsaufwandes bei der Suche in solchen hochdimensionalen Zustandsräumen ist die sogenannte partitionierte Stichprobenentnahme (engl. „partitioned sampling“, [MB00]). Dafür nimmt man an, dass sich der Zustandsraum geeignet partitionieren lässt, so dass gilt

$$p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}) = \int p(\boldsymbol{x}_i|\boldsymbol{x}') \cdot p(\boldsymbol{x}'|\boldsymbol{x}_{i-1}) d\boldsymbol{x}',$$

mit $\boldsymbol{x}' \in \mathbb{R}^{N_{x'}}$ und $N_{x'} < N_x$.

Die jeweiligen Stichproben werden dann in jeder Partition in Abhängigkeit von gewichteten Stichproben der vorangegangenen Partition gewählt und mittles $p(\boldsymbol{x}_i|\boldsymbol{x}')$ fortgeschaltet. Man nutzt also die Abhängigkeiten zwischen den Partitionen aus um einen hochdimensionalen Zustandsraum abzusuchen. Will man zum Beispiel gleichzeitig zur Position die Bein- oder Armstellung eines Fußgängers modellieren, kann man in der ersten Partition zuerst die Position anhand des Torsos bestimmen und hat damit die Suche für die Bestimmung der Bein- und Armstellungen in den weiteren Partitionen deutlich einschränken. Übertragen auf den Fall der Mehrobjektverfolgung

macht dies jedoch nur Sinn, wenn auch solche Abhängigkeiten zwischen den Objekten identifiziert werden können (z.B. in [SGP04]). Im Anwendungsfall dieser Arbeit mit dem Fokus auf große Entfernungen sind Abhängigkeiten wie sie beispielsweise aus der Innenraumüberwachung bekannt sind (z.B. Regeln über den Abstand der Personen zueinander) nicht verwendbar. Deshalb wird davon ausgegangen, dass sich Fußgänger unabhängig voneinander durch die Szenerie bewegen.

Ein weiterer Nachteil bei der Suche im vereinigten Zustandsraum besteht darin, dass Hypothesen sowohl wahre als auch falsche Einzelobjekthypothesen gleichzeitig umfassen können. Einzelne Objekte mit hoher Bewertung können so für ein hohes Gewicht der Hypothese verantwortlich sein, obwohl alle anderen Einzelobjekte Falschalarme darstellen (siehe auch [Spe05, Kapitel 8]). Umgekehrt können schlecht bewertete Einzelobjekte der Grund dafür sein, dass ansonsten gute Hypothesen verworfen werden [VGP05]. Dieses Problem wird umso größer, je mehr Objekte im Zustandsraum gleichzeitig modelliert werden.

In gewisser Weise liegt dieses Problem auch in dimensionsvariablen Zustandsräumen der Form

$$\Omega_{\text{Gesamt}} = \{\boldsymbol{o}\} \cup \bigcup_{\nu=1}^{M} \Omega^{\nu}, \quad \text{mit } \Omega^{\nu} \text{ wie in (5.18)}$$

vor. $\{\boldsymbol{o}\}$ modelliert dabei aber explizit auch das Nichtvorhandensein eines Objektes. Dieser Formalismus zum Tracken mehrerer Objekte wird in aktuellen Veröffentlichungen zum einen zusammen mit transdimensionalen Markov-Ketten, zum anderen im Zusammenhang mit endlichen Zufallsmengen (Finite Set Statsistics - FISST) verwendet. Erstere verwenden Abwandlungen des Metropolis-Hastings-Algorithmus [Has70] um neben dem Systemzustand im Parameterraum auch den Rückschluss auf die Modellparameter in Bayesscher Weise zu schätzen. Die dabei verwendeten transdimensionalen Markov-Ketten bzw. Monte-Carlo-Methoden ermöglichen nicht nur die Schätzung aller unbekannten Größen, sondern auch eine Abschätzung über die Anzahl der unbekannten Größen. Einen sehr guten Einstieg in dieses große Feld der Monte-Carlo-Methoden findet sich z.B. im Anhang von [Fre99, S. 183ff].

Im Gegensatz dazu wird in [Mah04] FISST vorgestellt, die es ermöglicht Detektion, Datenassoziation und Tracking in einem geschlossenenen mathematischen Ansatz zu lösen (siehe auch [VSD05, VS04]). Die Grundidee besteht darin, ein Wahrscheinlichkeitsmodell auf Basis zufälliger endlicher Mengen (engl. „random finite sets“) $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $n \geq 0, \boldsymbol{x} \in \mathbb{R}^{N_x}$ zu definieren, wobei sowohl die einzelnen Zufallsvektoren $\boldsymbol{x}_\nu, \nu = 1, \ldots, N$ als auch die Anzahl n der Ziele Zufallsvariablen darstellen. Insbesondere ist explizit mit $n = 0$, $\boldsymbol{X} = \{\}$ erlaubt, d.h. die Unsicherheit über die Existenz eines Objektes ist im Modell direkt berücksichtigt. Auf Basis dieses FISST-Wahrscheinlichkeitsmodells wird dann die Bayesrekursion (5.7) für mehrere - in der Anzahl variierender - Ziele neu formuliert. Dabei umfasst z.B. $p(\boldsymbol{X}_i|\boldsymbol{X}_{i-1})$ auch ein Geburten-, Sterbe- und Überdeckungsmodell, da $\boldsymbol{X}_i$ und $\boldsymbol{X}_{i-1}$ unterschiedliche Kardinalitäten aufweisen dürfen. Entsprechend fließen in $p(\boldsymbol{Z}_i|\boldsymbol{X}_i)$ auch fehlende, falsche oder mehrfache Detektionen mit ein.

Anwendungen von FISST finden sich z.B. in [Sid03, MSRD06]. Alle diese Verfahren haben den Nachteil, dass sie bisher noch zu komplex für Echtzeitanwendungen im Fahrzeug sind, obwohl bereits erste Implementierungen auf GPU-Basis für die Fußgängererkennung im Innenraum existieren.

Eine echtzeitfähige Variante ist der zunächst unabhängig von FISST entstandene Integrated Probabilistic Data Association (IPDA) Filter von [MES94] (zum Zusammenhang mit FISST, siehe [CYW02]). Analog zu FISST setzt der IPDA-Filter die Existenz des Ziels nicht mehr voraus und behandelt implizit Datenassoziation zusammen mit Existenzwahrscheinlichkeiten in einem rekursiven Verfahren zur Zustandsschätzung. Eine Anwendung im automobilen Umfeld findet sich in [MRD07]. Der IPDA-Filter stellt damit eine elegante und effiziente Methode zur probabilistischen Mehrobjektverfolgung dar. Allerdings ermöglicht er nicht die hypothesengetriebene Vorgehensweise bei der Fußgängerdetektion.

Im Rahmen dieser Arbeit wird die visuelle Mehrobjektverfolgung deshalb wie in [Spe05] vorgeschlagen über einen Multiinstanzen-Tracker realisiert, der auch im Rahmen der Detektion und Verfolgung mehrerer Fahrzeuge in Nachtsichszenarien in [ISP+06] demonstriert wurde. Dasselbe Vorgehen findet sich auch in [KMA01, OTF+04, VDP03].

Mehrere Objekte werden dabei nicht mehr gemeinsam im vereinigten Zustandsraum betrachtet, sondern unter der Annahme, dass die Objekte unabhängig voneinander seien, mit je einer eigenen Trackerinstanz verfolgt. Hauptproblem dabei ist es zu verhindern, dass zwei oder mehr Trackerinstanzen ein und dasselbe Objekte verfolgen. Die Problematik ist ähnlich der Datenassoziationsproblematik im Allgemeinen, wo entschieden werden muss, welche Messung zu welchen schon bekanntem Ziel gehört. In diesem Fall ist die Problemformulierung allerdings umgedreht: Die Partikel einer Instanz müssen sich von den Zielen anderer Filter loslösen, d.h. dürfen diese nicht in Betracht ziehen.

In [VDP03] werden dazu alle Samples zu Cluster gruppiert, um dann zu entscheiden welche Instanz zu welchem Cluster gehört. Allerdings arbeitet [VDP03] nicht mit mehreren Partikelfilterinstanzen, sondern mit Partikelmischverteilungen die bei Bedarf zusammengelegt werden. Die Verwendung von Clusterverfahren soll in dieser Arbeit vermieden werden, da die erforderlichen Maße zur Aufteilung in Cluster nur schwer begründbar sind und in vielen Fällen eher willkürlich erscheinen.

Zur Lösung des Problems wird in dieser Arbeit auf das in der Datenassoziation oft verwendete Prinzip des „hard gatings“ zurückgegriffen: Jede Trackerinstanz definiert eine Verbotszone um sein Ziel, in der andere Trackerinstanzen keine Samples generieren dürfen. Wie in [Loy03] vorgeschlagen, werden dazu jeweils die Gewichte der Partikel auf Null zurückgesetzt, wenn sie eine solche Verbotszone verletzen. Die Modellierung der Verbotszonen im konkreten Anwendungsfall erfolgt im Bildraum und basiert auf der Maximum-Überdeckung (2.7). Die genaue Umsetzung ist in Kapitel 6 beschrieben.

Werden mehrere Trackerinstanzen simultan ausgewertet und definiert jede seine eigene Verbotszone, stellt sich die Frage, in welcher Reihenfolge sie abgearbeitet werden sollen. Dabei sind mehrere Ordnungskriterien denkbar. [SGP04] schlägt eine zufällige Reihenfolge vor, was sich in [ISP+06] allerdings als unvorteilhaft erwiesen hat, da so

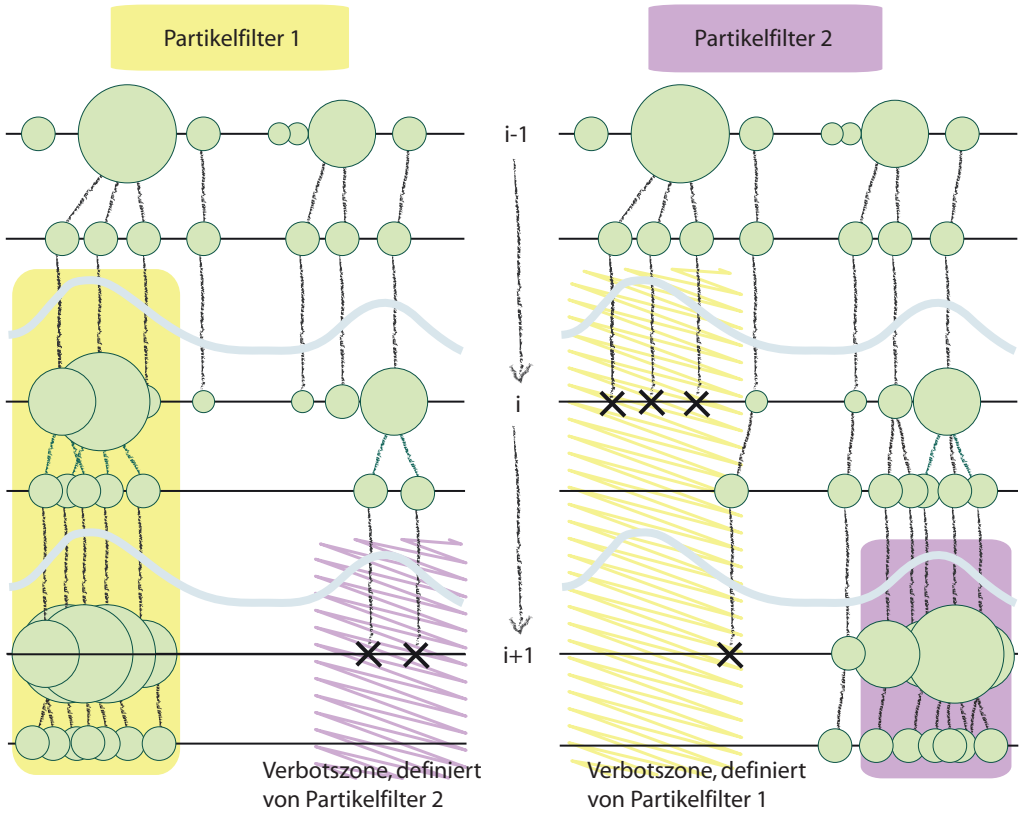


Abbildung 5.3.: Multiinstanzen Tracking ohne Priorisierung. Jeder Partikelfiler definiert auf seinem stärksten Ziel eine Verbotszone, die alle anderen Partikelfilter berücksichtigen müssen. Partikel innerhalb solcher Verbotszonen bekommen jeweils das Gewicht Null (dargestellt durch Kreuze). Die unterschiedlichen Instanzen werden untereinander nicht priorisiert, d.h. jeder Filter muss alle Verbotszonen aller anderen Filter berücksichtigen.

Trackerinstanzen laufend ihr Ziel auf dasjenige mit den stärksten Merkmalen gewechselt haben, sofern es nicht schon „belegt“ war. Auch eine Koppelung der Reihenfolge an die Partikelgewichte der jeweiligen Instanzen war nicht zielführend, da die große Varianz der Gewichte eher zu instabilem Verhalten geführt hat. Letztendlich hat sich eine fest vorgegebene statische Reihenfolge als robust erwiesen.

In [Loy03] wird vorgeschlagen, dass alle Trackerinstanzen die Verbotszonen aller anderen berücksichtigen sollen. In Abbildung 5.3 muss also die erste Trackerinstanz die Verbotszone des zweiten Trackers respektieren und umgekehrt. Im Rahmen der Arbeit von [Idl05] hat sich jedoch herausgestellt, dass dieser Mechanismus nur für eine kleine Anzahl von Zielen (< 5) funktioniert, da sich bei vielen Zielen die einzelnen Trackerinstanzen gegenseitig blockieren. Dies ist vor allem der Fall in Szenen mit vielen Falschmeldungen oder in Szenen mit eng beieinanderstehenden und überlappenden Zielen.

Deshalb werden in dieser Arbeit wie in [Idl05] die Trackerinstanzen neben der festen Abarbeitungsreihenfolge priorisiert (Abbildung 5.4). Jede Trackerinstanz muss

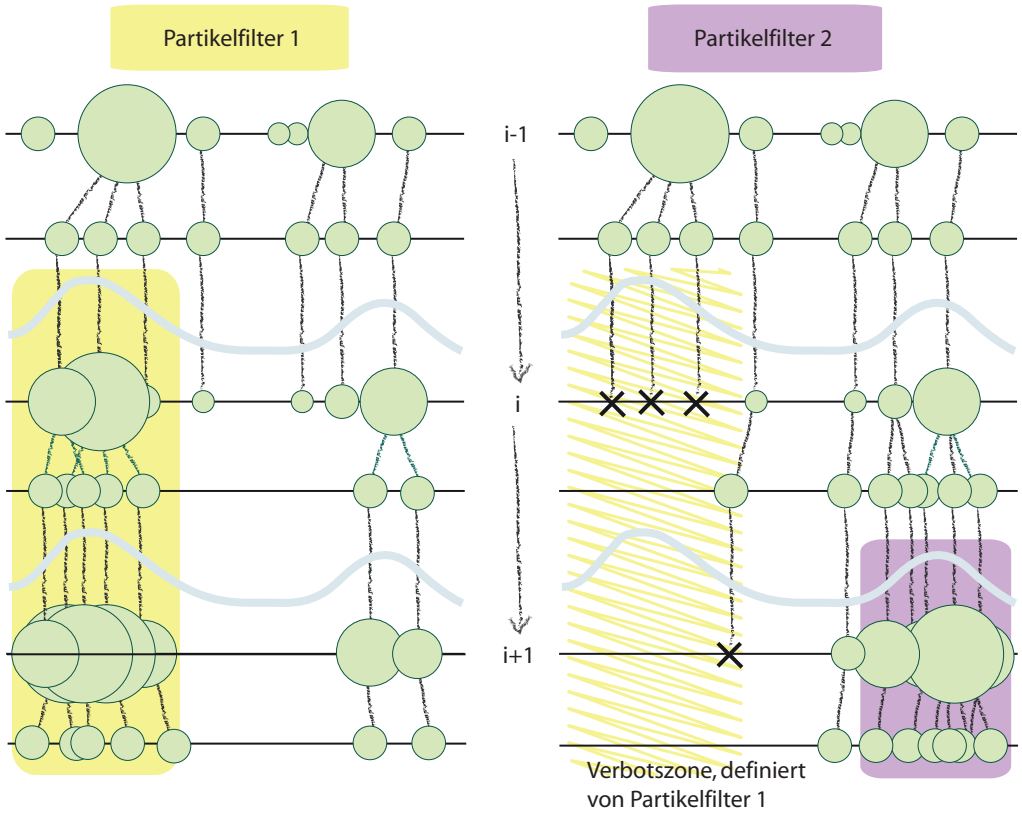


Abbildung 5.4.: Multiinstanzen Tracking mit Priorisierung. Jeder Partikelfilter definiert auf seinem stärksten Ziel eine Verbotszone, die im Unterschied zu Abbildung 5.3 nur auf niedriger priorisierte Partikelfilter Einfluss haben. Der höchst priorisierte Filter muss deshalb keine Verbotszonen berücksichtigen. Dadurch wird vermieden, dass sich einzelne Trackerinstanzen gegenseitig blockieren.

damit nur noch die Verbotszonen der höher priorisierten Instanzen berücksichtigen. Stehen zwei Trackerinstanzen bzgl. eines bestimmten Ziels in Konkurrenz, wird sich so immer der höher priorisierte durchsetzen können. Ohne Priorisierung würden die Trackerinstanzen immer im Wettbewerb zueinander stehen.

Der Partikelfilter wird in dieser Arbeit zur Organisation der Hypothesen zur Fußgängererkennung benutzt. Jedem Partikel ist dabei ein Suchfenstertupel zugeordnet, das von einem Klassifikator evaluiert wird. Die Besonderheit dabei ist, dass das Ergebnis des Klassifikators explizit in Form einer Rückschlusswahrscheinlichkeit als Messung im Partikelfilter berücksichtigt wird.

KAPITEL 6

Fußgängererkennung mit Partikelfilter

Obwohl die Ergebnisse kaskadierter Klassifikatoren auf Grund ihrer Struktur sehr schnell berechenbar sind, sind echtzeitfähige Anwendungen im Automobilbereich nur durch geeignete, intelligente Suchstrategien möglich. Gerade im Fall mehrerer Sensoren ist die Anzahl der zu prüfenden Hypothesen sehr groß (Abschnitt 4.2). Erst die Ausnutzung der charakteristischen Detektorantwort in Abschnitt 4.3 ermöglicht in einer dynamisch organisierten grob-zu-fein Suche die Realisierung eines echtzeitfähigen Fußgängerdetektors. Durch den Einsatz des Hypothesenbaums, der vorab offline erstellt werden kann, ist der Mehraufwand zur Organisation der Hypothesen gering. Im Mittel muss damit pro Bild nur noch ein Bruchteil der gesamten Hypothesenmenge berechnet werden. Allerdings steigt der Aufwand an, wenn das Bild viele Strukturen aufweist oder viele Fußgänger enthält. Im schlechtesten Fall müssen sogar in etwa gleich viele Hypothesen untersucht werden, wie mit dem statischen Hypothesengenerator.

Das in diesem Kapitel vorgestellte Verfahren nutzt den im vorausgegangenen Kapitel 5 beschriebenen probabilistischen Partikelfilter um die Hypothesen auch dynamisch über ganze Bildfolgen hinweg zu organisieren. Die Anzahl der Partikel und damit auch der Aufwand pro Bild bleiben gleich.

Die Hypothesen, die in jedem Bild vom Kaskadenklassifikator untersucht werden, werden nicht mehr vorab erzeugt, sondern durch Partikel ersetzt, die vom Partikelfilter über die Zeit hinweg im Zustandsraum fortgeschaltet werden. Die Zustandsmodellierung wird in Abschnitt 6.1 beschrieben. Über ein Systemmodell unter Berücksichtigung der Eigenbewegung des Fahrzeugs werden die Partikel in jedem Iterationsschritt des Filters entsprechend dem dynamischem Systemmodell im Zustandsraum verschoben und verrauscht. Über die in Kapitel 3.5 hergeleiteten Rückschlusswahrscheinlichkeiten fließen die Ergebnisse des Kaskadenklassifikators direkt als Wahrscheinlichkeiten zur Gewichtung der Partikel im Filter mit ein (Abschnitt 6.2). Nach bereits wenigen Bildern

fokussieren sich die Partikel damit automatisch in den Bereichen mit Fußgängern im Bild. Die Initialisierung der Partikelmenge erfolgt, indem ein statischer Hypothesengenerator mit sehr grober Rasterung verwendet wird, um die jeweiligen Partikelmengen in einem Importance a priori Sampling Schritt zu initialisieren. Die Detektionsentscheidung übernimmt dann ein MAP-Schätzer.

Da in einer Szenerie in der Regel mehrere Fußgänger auftreten können, wird das Konzept der in Kapitel 5.4 beschriebenen visuellen Mehrobjektverfolgung mit Hilfe von mehreren Instanzen von Partikelfiltern umgesetzt. Das dazu benötigte Ausschlusskriterium zur Definition von Verbotszonen wird in Abschnitt 6.3 erläutert.

6.1. Zustandsmodellierung

Der Aufwand zur Schätzung eines Zustandes mit Partikelfilter steigt in erheblichem Maße mit der Anzahl der Systemvariablen im Zustandsvektor $\boldsymbol{x}$. Je größer die Dimension des Zustandsraumes, desto mehr Partikel sind nötig, um diesen ausreichend abdecken zu können. Um die Effizienz des Verfahrens sicher zu stellen, ist deshalb ein möglichst kompakter Zustandsraum erforderlich.

Der Zustandsraum kann sowohl in Bildkoordinaten, als auch in Weltkoordinaten definiert werden. Die Modellierung in Weltkoordinaten hat dabei den Vorteil, dass damit gleichzeitig die Entfernung eines Fußgängers mit geschätzt wird. Geht man zunächst von einer ebenen Welt aus, kann der Scheitelpunkt eines Fußgängers durch die Distanz X und die laterale Ablage Y relativ zum Fahrzeug angegeben werden. Die entsprechenden Geschwindigkeitskomponenten sind $\dot{X}$, bzw. $\dot{Y}$. Neben der Fußgängerposition und der Geschwindigkeit muss auch die räumliche Ausdehnung eines Fußgängers berücksichtigt werden, damit im Bildraum geeignete Suchfenster zur Klassifikation festgelegt werden können. Die Größe H der Fußgänger muss also prinzipiell mit berücksichtigt werden. Eine nahe liegende Modellierung in Fahrzeugkoordinaten[1] ist dann

$$\boldsymbol{x}_i = \left(X_i, Y_i, \dot{X}_i, \dot{Y}_i, H_i\right)^{\mathrm{T}}.$$

Zur Gewichtung der Partikel durch den Kaskadenklassifikator ist es dann notwendig, den Systemzustand in den Bildraum zu projizieren, um so Suchfenster (bzw. Suchfenstertupel im Fall mehrerer Sensoren) zu erhalten. Das dynamische System ist durch die Bewegungen des Fußgängers und die des Fahrzeuges definiert.

In der Praxis hat sich dieses Modell im Zusammenhang mit der Bewertung durch den Kaskadenklassifikator jedoch als nicht praktikabel erwiesen. Die Gründe dafür sind wie folgt:

- Die Annahme einer ebenen Welt ist in der Regel nicht gerechtfertigt. Insbesondere aufgrund der Nickbewegungen des Fahrzeuges selbst ist dieses Vorgehen problematisch: Abbildung 6.1 zeigt, dass ein Nickwinkel ρ die Schätzung unrealistischer

[1] Zur Erinnerung: Fahrzeugkoordinaten werden in dieser Arbeit in Großbuchstaben angegeben.

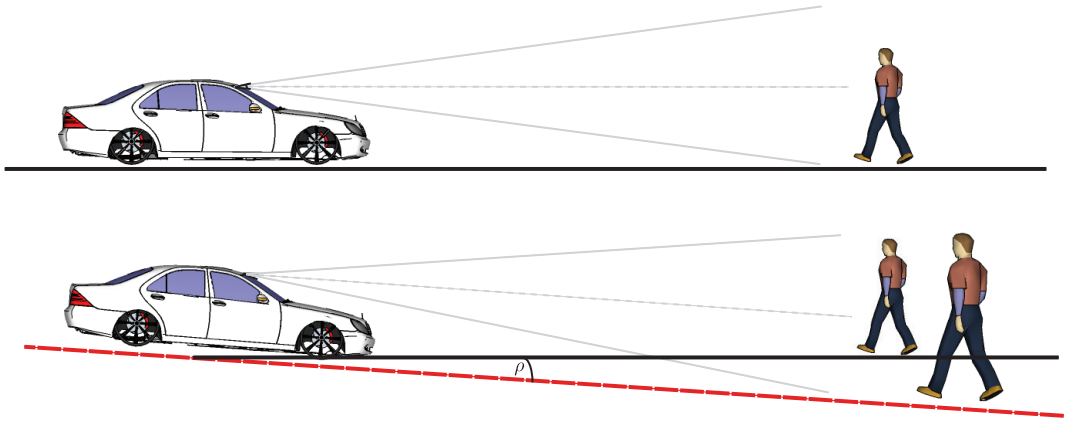


Abbildung 6.1.: Einfluss des Nickwinkels auf die Größenschätzung von Fußgängern. Durch die Nickbewegung des Fahrzeugs wird die X-Y-Ebene (rote, leicht gestrichelte Linie) unter die Straße (schwarze Linie) verschoben. Wird trotz der Nickbewegung von einer ebenen Welt ausgegangen, ist eine Zustandsschätzung nur unter der Annahme unrealistischer Fußgängergrößen (rechter Fußgänger) möglich.

Fußgängergrößen nach sich zieht. In [Arn06] wurde dieses Problem dadurch umgangen, dass die Größe des Fußgängers als konstant angenommen und stattdessen ein Relaxationswinkel ρ mit im System aufgenommen wurde. Problematisch ist dabei jedoch, dass der Relaxationswinkel durch keine Messung gestützt werden kann und zusätzliche Heuristiken nötig wurden, um das System zu stabilisieren (z.B. indem – leider erfolglos – versucht wurde, den Nickwinkel aus der vertikalen Verteilung der Antworten des Kaskadenklassifikators zu schätzen).

- Eine sinnvolle Modellierung der Messunsicherheit erzwingt eine große Unsicherheit in der Entfernungsschätzung. Die Messung erfolgt im Bildraum, d.h. die Messunsicherheit muss im Bildraum definiert werden. Geht man im Bild z.B. von einem additiven gaußverteiltem Rauschen der Suchfensterscheitelpunkte und Suchfensterhöhen aus, so ist die Rückprojektion dieser Rauschkomponente in den Zustandsraum (bei Annahme einer ebenen Welt) eine asymmetrische Verteilung mit hoher Varianz in der Entfernungskomponente (Abbildung 6.2).

- Der Kaskadenklassifikator ist für eine hochgenaue Objektlokalisierung nicht geeignet. Insbesondere durch die Größenskalierung der Merkmale ist die Lokalisierung im Bild unscharf[2]. Zusätzlich kommen durch die Posenvielfalt im Trainingsdatensatz, sowie eventuellen Ungenauigkeiten beim Labeln Unsicherheiten in der genauen Lokalisierung hinzu. Oft sind die Detektionsboxen vor allem bei Mehrfachdetektionen zu groß (Abbildung 6.3) und fehlen für die eigentlich richtige Skalierung. Damit ist eine genaue Entfernungsschätzung nahezu unmöglich. Die Lokalisierungsunsicherheit, die beim Hypothesenbaum in Kapitel 4.3 sogar expli-

[2]Zur Erinnerung: die Merkmale sind im Raster des Basissuchfensters definiert (siehe Kapitel 3.1) und werden entsprechend der Größe der Hypothesen mit skaliert.

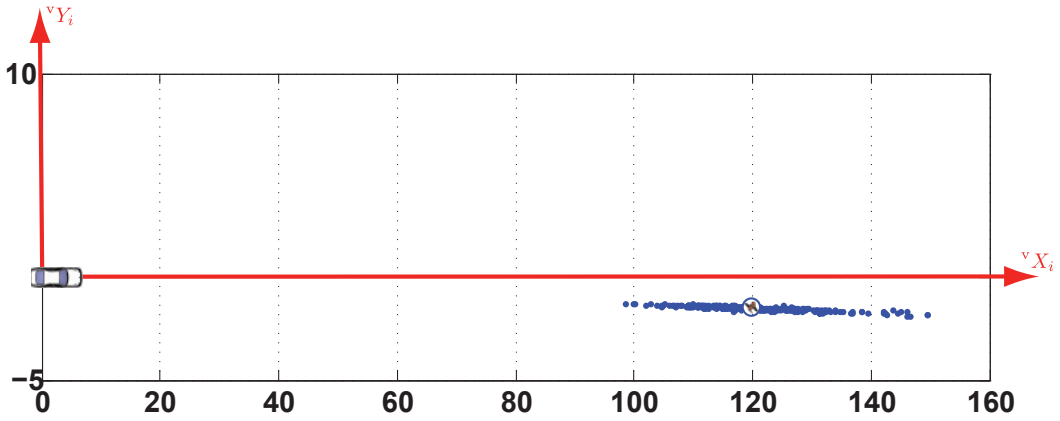


Abbildung 6.2.: Unsicherheit in der Entfernungsschätzung. Die Messunsicherheit wird sinnvollerweise immer im Bildraum definiert werden. Dadurch entsteht in einem Zustandsraum, der in Weltkoordinaten modelliert ist, eine große Varianz in der Entfernungskomponente. Dargestellt sind Fußgängerposition eines Fußgängers in 120m Entfernung, die aus der Rückprojektion verrauschter Suchfenster im Bild entstehen. Das normalverteilte Rauschen ist im Bildraum additiv mit $\sigma_{\text{col}} = 0.03 \cdot h$, $\sigma_{\text{row}} = 0.05 \cdot h$ und $\sigma_h = 0.08 \cdot h$.

Abbildung 6.3.: Mehrfachdetektionen mit falscher Skalierung. Dargestellt sind die Objektfenster von drei Detektionen auf diesem Fußgänger. Der Fußgänger gilt zwar als richtig erkannt, doch alle drei Objektfenster sind zu groß, so dass eine direkte Entfernungsschätzung auf Basis der Skalierung nicht möglich ist.

zit in Form der charakteristischen Detektorantwort ausgenutzt wurde, verhindert also eine genaue Lokalisierung.

Ein Tracking der Einzelbilddetektionen im Weltkoordinatensystem erscheint damit lediglich in einem nachgeschalteten Prozess sinnvoll, indem die Detektionen in einem Nachverarbeitungsschritt an die Größe der Fußgänger im Bild angepasst und dann erst an eine Trackingkomponente übergeben werden. Darüber hinaus ist das Hauptziel des Einsatzes von Partikelfiltern in diesem Fall ja nicht das robuste Tracking von Fußgängern an sich, sondern die intelligente Steuerung einer Hypothesenmenge zur Detektion eines Fußgängers.

Der Zustandsraum wird in dieser Arbeit deshalb bewusst auf ein Minimum reduziert:

$$\boldsymbol{x}_i = (\mathrm{col}_i, \mathrm{row}_i, h_i, H_i)^{\mathrm{T}}. \tag{6.1}$$

$(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ ist der Scheitelpunkt des Fußgängers im Bild, h dessen Skalierung und H die tatsächliche Größe des Fußgängers in der Welt. Die Hinzunahme von H hat sich dabei in Experimenten als vorteilhaft erwiesen, da zur Berücksichtigung der Eigenbewegung des Fahrzeugs die Partikel mittels $\mathrm{proj}_H(\cdot)$ (siehe Kapitel 2.2), und damit abhängig von H, ins Fahrzeugkoordinatensystem gebracht werden müssen. Alternativ kann man jedoch auch von einer festen Objekthöhe (z.B. $H = 1.80\mathrm{m}$) ausgehen.

Die Eigenbewegung des Fahrzeugs, also die Verschiebung $(\Delta^{\mathrm{v}}Y_i, \Delta^{\mathrm{v}}Y_i)^{\mathrm{T}}$ sowie die Drehung $\Delta\psi_i$ vom Zeitschritt $i-1$ zum Zeitschritt i, werden unabhängig vom Partikelfilter mit Hilfe eines Kalmanfilters geschätzt. Das Systemmodell $\boldsymbol{f}_i$ im Partikelfilter hat damit die Gestalt

$$\boldsymbol{f}_{i+1}(\boldsymbol{x}_i) = \boldsymbol{f}_{i+1}(\mathrm{col}_i, \mathrm{row}_i, h_i, H_i) = \begin{pmatrix} \mathrm{proj}_{H_i}\left(\boldsymbol{R}^*_{\Delta\psi_i} \cdot \mathrm{proj}^{-1}_{H_i}(\mathrm{col}_i, \mathrm{row}_i, h_i)\right) \\ H_i \end{pmatrix},$$

mit

$$\boldsymbol{R}^*_{\Delta\psi_i} = \begin{pmatrix} \cos\Delta\psi_i & \sin\Delta\psi_i & 0 \\ -\sin\Delta\psi_i & \cos\Delta\psi_i & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

und $\mathrm{proj}_H(\cdot)$ wie in (2.5), Seite 38, Kapitel 2.2 definiert. $\boldsymbol{R}^*_{\Delta\psi_i}$ stellt die in Abbildung 6.4 dargestellte Drehung des Koordinatensystems, hervorgerufen durch die Drehung $\Delta\psi_i$ des Fahrzeugs, dar. Die Fußgängerbewegungen selbst werden in diesem Modell nicht direkt berücksichtigt. Zwar könnten Geschwindigkeitskomponenten $\left(\dot{X}, \dot{Y}\right)$ im Systemzustand zusätzlich berücksichtigt werden, doch würden damit weitaus mehr Partikel benötigt, um den dann sechsdimensionalen Zustandsraum vollständig abzudecken. Darüber hinaus zeichnet sich das Anwendungsszenario einer Fußgängerdetektion von einem fahrenden Fahrzeug durch eine sehr hohe Dynamik aus, da im Gegensatz zur Eigenbewegung des Fahrzeugs die Fußgängerbewegungen selbst nur schwer vorhersehbar sind. Fußgänger können jederzeit Bewegungsrichtung und Geschwindigkeit aprupt ändern. Im Zusammenhang mit Partikelfiltern werden unter diesem Gesichtspunkt oft Markov-Modelle vorgeschlagen um die hohe Dynamik von Fußgängerbewegungen abzubilden (siehe z.B. [IB98b] oder [GGS04]). Die Wahrscheinlichkeiten für plötzliche

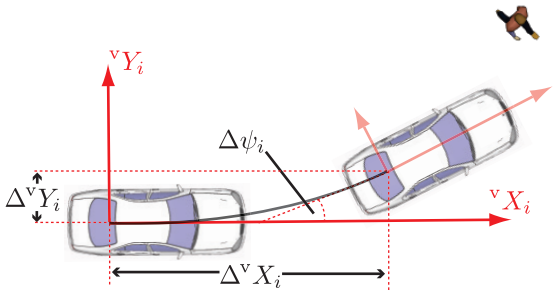


Abbildung 6.4.: Koordinatentransformation im Systemmodell des Partikelfilters. Durch die Eigenbewegung des Fahrzeugs ändert sich das Koordinatensystem durch die Translation $(\Delta^{\mathrm{v}}X_i, \Delta^{\mathrm{v}}Y_i)^{\mathrm{T}}$ und die Drehung $\Delta\psi_i$.

Zustandsänderungen sind dabei meist Schätzungen auf Basis statistischer Auswertungen. Aufgrund der Vielzahl verschiedener Situationen und Bewegungsänderungen wird dieser Ansatz in dieser Arbeit nicht weiter verfolgt. Wird ein Track aufgrund einer plötzlichen Bewegungsänderung abgebrochen, so wird der Fußgänger durch eine andere Partikelfilterinstanz neu erkannt und weiter verfolgt. Ansonsten werden alle Fußgängerbewegungen über die stochastische Diffusion im Prädiktionsschritt $p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})$ des Partikelfilters modelliert.

Im Gegensatz zum Kalmanfilter, der explizit zwischen Zustandsübergangsrauschen und Messrauschen unterscheidet und diese getrennt und unabhängig voneinander modelliert, wird im Partikelfilter auf diese Unterscheidung verzichtet. In dieser Arbeit werden zur Modellierung eines Systemübergangsrauschens mittelwertfreie Normalverteilungen $\mathcal{N}(\boldsymbol{o}, \boldsymbol{\Sigma})$ benutzt. Da die Bewegungen der Fußgänger aufgrund der Projektion ins Bild im Nahbereich einen stärkeren Einfluss auf den Scheitelpunkt $(\mathrm{col}, \mathrm{row})^{\mathrm{T}}$ der Fußgänger im Bild haben als Bewegungen der Fußgänger im Fernbereich, muss das Systemrauschen für nahe Fußgänger größer gewählt werden als für weiter entfernte. Ebenso ist aufgrund der skalierungsabhängigen Eigenschaften des Klassifikators bei Fußgängern mit großer Skalierung eine größere Messunsicherheit zu erwarten. Deshalb werden die Standardabweichungen für $\mathrm{col}, \mathrm{row}$ und h abhängig von der Skalierung gewählt und über α_{col}, α_{row} und α_h parametrisiert, d.h.

$$\begin{aligned} \sigma_{\mathrm{col},i} &:= \alpha_{\mathrm{col}} h_i, \\ \sigma_{\mathrm{row},i} &:= \alpha_{\mathrm{row}} h_i \quad \text{und} \\ \sigma_{h,i} &:= \alpha_h h_i. \end{aligned}$$

Zusätzlich werden die Rauschprozesse zu h und H als stark korreliert angenommen. Mit dem Korrelationskoeffizienten ρ_{hH} ergeben sich die entsprechenden Nebendiagonaleinträge in $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{x}_i) = \boldsymbol{\Sigma}(h_i) = \begin{pmatrix} (\alpha_{\mathrm{col}} h_i)^2 & 0 & 0 & 0 \\ 0 & (\alpha_{\mathrm{row}} h_i)^2 & 0 & 0 \\ 0 & 0 & (\alpha_h h_i)^2 & \rho_{hH}(\alpha_h h_i)\sigma_H \\ 0 & 0 & \rho_{hH}(\alpha_h h_i)\sigma_H & \sigma_H^2 \end{pmatrix}. \tag{6.2}$$

In dieser Arbeit ist $\alpha_{\text{col}} = 0.03$, $\alpha_{\text{row}} = 0.05$, α_h=0.08 und $\rho_{hH} = 0.9$ gewählt.

6.2. Initialisierung, Gewichtung und Detektionsentscheidung

Als konkrete Umsetzung des Partikelfilterverfahrens kommt hier der Condensation-Algorithmus aus Kapitel 5.3 zum Einsatz. Dessen Initialisierung, die Gewichtung der Partikel und wie letztendlich eine Detektionsentscheidung getroffen werden kann, wird im Folgenden beschrieben.

Initialisierung

Normalerweise erfolgt die Initialisierung der Partikelmenge durch eine zufällige gleichverteilte Streuung der Partikel im gesamten Zustandsraum. Um eine schnelle Detektion insbesondere weit entfernter Fußgänger zu ermöglichen, werden alternativ dazu Importance Samples benutzt, die eine gezielte Initialisierung einer Partikelfilterinstanz im Bildraum ermöglichen. Dazu werden zunächst Hypothesen mit sehr grober Rasterung (Modell III in Kapitel 4.1 bzw. Kapitel 4.2) erzeugt und vom Kaskadenklassifikator bewertet. Auf Basis der Rückschlusswahrscheinlichkeiten werden dann die N_s^* besten Hypothesen ausgewählt. Damit neue Partikel nicht alle im selben Bildbereich erzeugt werden, sondern im ganzen Bild verteilt sind, werden diese N_s^* besten Hypothesen auf Basis ihrer Überdeckung zu Clustern zusammengefasst (Abbildung 6.5).

Als Clusterkriterium dient hier das Überdeckungsmaß $\text{cov}(\cdot,\cdot)$. Iterativ werden diejenigen Hypothesen zusammengefasst, deren Überdeckung eine vorgegebene Schwelle $\text{cov}_{\text{cluster}}$ (in dieser Arbeit $\text{cov}_{\text{cluster}} = 0.1$) überschreiten. Die jeweils beste Hypothese aus einem Cluster dienst dann als Mittelwert einer normalverteilten a priori Dichte zur Initialisierung der Partikelmenge. Die Höhe H im Zustandsraum wird dann über eine Gleichverteilung innerhalb des Intervalls $[H_{\min}, H_{\max}]$ festgelegt.

Durch die zusätzliche Auswertung einer grob gerasterten Hypothesenmenge entsteht einerseits zusätzlicher Aufwand. Andererseits kann die Anzahl der Partikel durch eine gute Initialisierung deutlich reduziert werden. In dieser Arbeit werden deshalb nur 750 Partikel für eine Partikelfilterinstanz benötigt.

Gewichtung der Partikel

Die Gewichtung der Partikel erfolgt mit Hilfe des Kaskadenklassifikators. Im Einzel-Sensor Fall ist dabei die Anwendung des Klassifikators direkt möglich, da jeder Zustand $\boldsymbol{x}_i$ einem Objektfenster $o_i = (\text{col}_i, \text{row}_i, h_i)$ entspricht. Die Gewichtung ist dann die Rückschlusswahrscheinlichkeit des jeweiligen Kaskadenergebnisses:

$$g(\boldsymbol{x}_i) = g(\text{col}_i, \text{row}_i, h_i, H_i) = g(o_i, H_i) = p(y = +1 | x = \chi(o_i)).$$

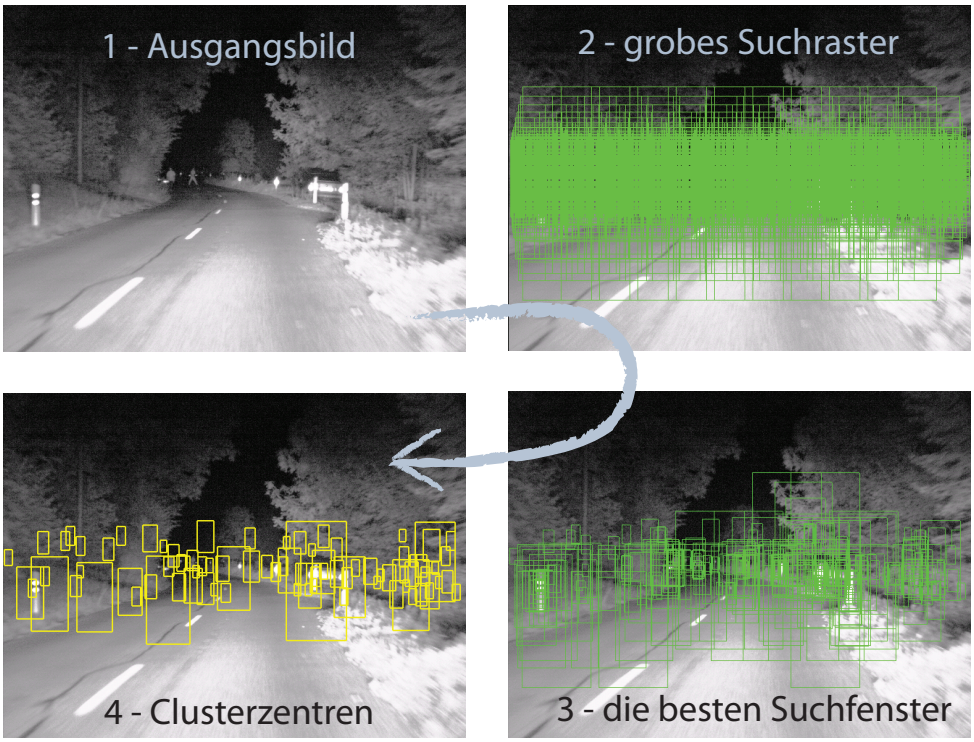


Abbildung 6.5.: Initialisierung der Partikelmenge. Zunächst werden Hypothesen mit sehr grober Rasterung erzeugt (2) und vom Kaskadenklassifikator bewertet. Auf Basis der Rückschlusswahrscheinlichkeiten werden dann die besten Hypothesen ausgewählt (3). Damit neue Partikel nicht alle im selben Bildbereich erzeugt werden, sondern im ganzen Bild verteilt sind, werden diese dann auf Basis ihrer Überdeckung zu Clustern zusammengefasst (4).

Im Multi-Sensor Fall ist durch einen Zustand $\boldsymbol{x}_i$ lediglich das Objektfenster o_i^* des Primärsensors definiert. Über die Höhe H kann das zugehörige Objektfenster im Sekundärsensor bestimmt werden und der Zustand wie im Einzel-Sensor Fall gewichtet werden. Mit der Hilfsfunktion $\text{proj\_stream2stream}_{H_i}(\cdot)$, definiert in (4.4), Seite 91, Kapitel 4.2 zur Bestimmung des Abbildes eines Fußgängers aus dem Bild des ersten Sensors ins Bild des zweiten Sensors, und der Funktion $\chi(\cdot)$ aus Kapitel 2.4 gilt formal

$$g(\boldsymbol{x}_i) = g(o_i, H_i) = p(y = +1 | x = (s_i^*, s_i)),$$

mit

$$\begin{aligned} s_i^* &= \chi(o_i^*), \\ s_i &= \chi\left(\text{proj\_stream2stream}_{H_i}(o_i^*)\right). \end{aligned}$$

Dieses Vorgehen ist natürlich suboptimal, da hier wieder von einer ebenen Welt ausgegangen wird. Zwar ist der Nickwinkel implizit auch durch H im Zustand mit abgedeckt (siehe Abbildung 6.1, Seite 121), doch die Wechselwirkungen müssen in der Praxis mit einer großen Anzahl an Partikeln kompensiert werden. Deshalb werden hier die Konzepte des Multi-Sensor Hypothesengenerators wieder aufgegriffen und ausgehend von einem Objektfenster im Primärsensor eine Menge an korrespondierenden Objektfenstern im Sekundärsensor erzeugt (vgl. Algorithmus 4.4, Seite 95, Abschnitt 4.2).

Sei $\hat{\mathcal{H}}(o_i\,;\,H_i)$ die Korrespondenzhypothesenmenge aus Algorithmus 4.4 mit $H_{\max} = H_{\min} = H_i$, dann ist die Gewichtsfunktion zur Bewertung der Partikel gegeben durch

$$g(\boldsymbol{x}_i) = \max_{x \in \hat{\mathcal{H}}(o_i\,;\,H_i)} p(y = +1|x).$$

Detektionsentscheidung

Der Partikelfilter propagiert die Wahrscheinlichkeitsdichtefunktion über die Position eines eventuell existierenden Fußgängers über die Zeit. Zunächst wird dabei keine Aussage darüber gemacht, ob es sich tatsächlich um einen Fußgänger handelt oder nicht. Zu jedem Zeitpunkt gibt es lediglich eine Menge von gewichteten Partikeln. Die Detektionsentscheidung muss also auf Basis der Partikelmenge gefällt werden.

Für den Schätzwert des eigentlichen Zustandes eignet sich in dieser Arbeit das Ergebnis des MAP-Schätzers (5.8), d.h.

$$\hat{\boldsymbol{x}}_i^{\text{MAP}} = \arg\max_i p(\boldsymbol{x}_i | \boldsymbol{z}_{1:i}).$$

Übertragen auf die Partikelmenge wird ein Fußgänger an der Stelle im Zustandsraum vermutet, die durch das Partikel mit dem höchsten Gewicht dargestellt wird, also

$$\hat{\boldsymbol{x}}_i^{\text{MAP}} = \boldsymbol{x}_i^{(j^*)}, \quad \text{mit} \quad w_i^{(j^*)} = \arg\max_i w_i^{(j)}.$$

Damit ist die Entscheidung über eine Detektion sehr nahe an die Rückschlusswahrscheinlichkeiten des Klassifikators genküpft. Dies ist nur konsequent, da so die Entscheidung über einen Fußgänger ohne zusätzliche Heuristiken auskommt und der Partikelfilter gezielt dafür eingesetzt werden kann den Suchraum optimal nach Fußgängern abzusuchen. Die Detektion wird damit anhand eines Schwellwertes w^* auf Basis der Gewichtung $w_i^{(j^*)}$ des Schätzwertes $\hat{\boldsymbol{x}}_i^{\mathrm{MAP}}$ bestimmt.

6.3. Multiinstanzen Fußgängerverfolgung

Der in den vorangegangenen Abschnitten beschriebene Partikelfilter ist erst einmal nur in der Lage maximal einen Fußgänger zu detektieren. Um mehrere Fußgänger gleichzeitig in einem Bild erkennen zu können, wird das Verfahren aus Kapitel 5.4 eingesetzt, d.h. mehrere Partikelfilterinstanzen suchen gleichzeitig den Zustandsraum ab und werden durch Verbotszonen und einem Priorisierungskriterium davon abgehalten alle dasselbe Objekt zu verfolgen. Zusätzlich stellt ein Reinitialisierungskriterium sicher, das divergierende Partikelfilterinstanzen schnell aufgelöst werden.

Definition von Verbotszonen

Die Verbotszonen werden anhand eines Überdeckungskriteriums der Suchfenster festgelegt. Dazu dient die maximale Überdeckung $\mathrm{cov}_{\max}$ zwischen zwei Objektfenstern. Die Verbotszone wird dann durch die Vorgabe einer maximal zulässigen Überdeckung $\mathrm{cov}^*_{\max}$ zwischen dem Suchfenster eines Partikels und dem Suchfenster eines bereits detektierten Ziels einer anderern Trackerinstanz definiert. Innerhalb dieser Arbeit wurde $\mathrm{cov}^*_{\max} = 0.8$ gewählt.

Überschreiben sich die Partikelmengen zweier Instanzen, die beide eine Detektion darstellen, wird die Verbotszone darüber hinaus im Zustandsraum so erweitert, dass die räumlich näher am Fahrzeug gelegene Schätzung gegenüber der anderen im Vorteil ist. Das heißt wenn eine aktive Trackerinstanz mit einer anderen aktiven Trackerinstanz kollidiert, dessen Partikel eine höhere Skalierung aufweisen, werden alle ihre Partikel mit Null gewichtet. Dieses zusätzliche Kriterium gilt nur für aktive Instanzen untereinander, also solchen, deren Likelihood-Ratio groß genug ist. Abbildung 6.6 illustriert anhand eines Beispiels die unterschiedlichen Verbotszonen.

Likelihood-Ratio als Reinitialisierungskriterium

Das Reinitialisierungskriterium im Kontext einer robusten Fußgängerdetektion sollte folgenden Anforderungen genügen:

- Damit Fußgänger möglichst frühzeitig bereits in großen Entfernungen erkannt werden, sollte das Reinitialisierungskriterium das Verfolgen schwacher Ziele ermöglichen.

Abbildung 6.6.: Verbotszone. In der Darstellung ist jede Partikelfilterinstanz in einer anderen Farbe dargestellt. Detektionen sind durch transparente Flächen gekennzeichnet. Durch die bevorstehende Kollision zwischen dem gelben und dem grünen Filter werden die Partikel des gelben mit 0 gewichtet. Die restlichen dargestellten Instanzen können entsprechend der Überdeckung $\text{cov}_{\max}$ keine Zustandshypothesen in der Nähe der beiden detektierten Fußgänger generieren.

- Wurde durch eine Reinitialisierung bereits zu Beginn ein aussichtsreiches Ziel erfasst, sollte die entsprechende Trackerinstanz eine andere Instanz verdrängen können, die eventuell schon längere Zeit ein schlechtes Ziel verfolgt.

Die Formulierungen „schwache“ bzw. „aussichtsreiche“ Ziele implizieren dabei das Vorhandensein eines Gütekriteriums für eine Partikelmenge. Eine MAP-Schätzung, wie sie auch zur Detektionsentscheidung verwendet wird, ist dazu nur bedingt geeignet. Eine Instanz mit nur sehr wenigen überdurchschnittlich gut bewerteten Partikeln sticht damit alle anderen Instanzen aus, obwohl diese unter Umständen „aussichtsreicher“ sind. Die Güte einer Instanz wird besser auf Basis der gesamten Partikelmenge bestimmt.

Das in dieser Arbeit zum Einsatz kommende Likelihood-Ratio Kriterium erfüllt diese Anforderung eines Gütekriteriums und ermöglicht darüber hinaus eine Reinitialisierung, die sich dynamisch an die gegenwärtige Szene anpassen kann. Grundidee ist, anhand eines Signal-zu-Rausch Verhältnisses auf das Vorhandensein eines Fußgängers zu schließen. Vorbild ist dabei die Radartechnik, da dort die Eingabedaten (Beobachtungen) oft sehr stark verrauscht sind (siehe z.B. [BD01]). Auf Basis solcher unsicheren Daten soll dann entschieden werden, ob diese eher Hintergrund- oder Objektbeobachtungen enthalten. Formal unterscheidet man dabei zwischen zwei Hypothesen H_1 und H_0 (Hypothesen im statistischen Sinn):

- **Hypothese H_0**: Bei der Beobachtung $\boldsymbol{z}_i$ handelt es sich um ein aufgrund von Rauschen verursachtes Phänomen (also Hintergrund). Die Dichte $p(\boldsymbol{z}_i|\text{H}_0)$ drückt für die Beobachtung $\boldsymbol{z}_i$ aus, mit welcher Wahrscheinlichkeit diese aufgrund eines Hintergrundphänomens entstanden sein könnte.

- **Hypothese $\mathbf{H}_1$**: Die Beobachtung $\boldsymbol{z}_i$ wurde durch den aktuellen Systemzustand $\boldsymbol{x}_i$ des verfolgten Ziels verursacht. Die entsprechende Wahrscheinlichkeitsdichte ist durch $p(\boldsymbol{z}_i|\mathrm{H}_1)$ gegeben.

Die Likelihood-Ratio $\mathcal{L}$ einer Beobachtung $\boldsymbol{z}_i$ ist definiert durch

$$\mathcal{L}(\boldsymbol{z}_i) = \frac{p(\boldsymbol{z}_i|\mathrm{H}_1)}{p(\boldsymbol{z}_i|\mathrm{H}_0)}.$$

Die Güte einer Trackerinstanz spiegelt sich sehr gut in dieser Likelihood-Ratio $\mathcal{L}$ wieder. Es wird jeweils eine Anzahl von M^* Partikelfilterinstanzen reinitialisiert, und zwar diejenigen mit den niedrigsten Werten von $\mathcal{L}(\boldsymbol{z}_{i-1})$. Natürlich wird darüber hinaus eine Partikelfilterinstanz auch dann reinitialisiert, wenn die zugehörige MAP-Schätzung den Zustandsraum verlässt, d.h. nach der Anwendung des dynamischen Systemmodells zu viele Partikel außerhalb der Bildgrenzen liegen.

Im Allgemeinen liegen die zugrundeliegenden Dichten $p(\boldsymbol{z}_i|\mathrm{H}_0)$ und $p(\boldsymbol{z}_i|\mathrm{H}_1)$ nicht in auswertbarer Form vor. [BD01] hat jedoch gezeigt, dass die Wahrscheinlichkeit für das Vorhandensein eines Objektes auf Grundlage der Partikelmenge Ξ_i approximiert werden kann, indem der Mittelwert aller unnormalisierter Partikelgewichte herangezogen wird:

$$p(\boldsymbol{z}_i|\mathrm{H}_1) \approx \frac{1}{N_s} \sum_{k=1}^{N_s} \bar{w}_k \tag{6.3}$$

Eine anschauliche Interpretation für Gleichung (6.3) erhält man, wenn man sich klar macht, dass eine Partikelmenge den Zustandsraum hypothetisiert. Im Kontext der Detektion von Fußgängern bedeutet dies, dass ein Partikel in Form einer gewichteten Stichprobe einen möglichen Fußgänger im Zustandsraum darstellt. Die Likelihood $p(\boldsymbol{z}_i|\mathrm{H}_1)$ validiert dies anhand der gegenwärtigen Beobachtung $\boldsymbol{z}_i$. Die Partikelgewichtung bewertet, wie wahrscheinlich die Hypothese H_1 für einige Bildausschnitte ist. Aus diesem Grund stellt der Mittelwert aller unnormalisierter Partikelgewichte eine Approximation der Wahrscheinlichkeit $p(\boldsymbol{z}_i|\mathrm{H}_1)$ dar.

Zu einer möglichen Approximation von $p(\boldsymbol{z}_i|\mathrm{H}_0)$ macht [BD01] keine Angaben. Diese Information ist in der Radartechnik in aller Regel durch die Charakteristika des messenden Sensors direkt verfügbar. In Analogie zu den Überlegungen zur Approximation von $p(\boldsymbol{z}_i|\mathrm{H}_1)$ durch (6.3) erfordert die Berechnung der Wahrscheinlichkeit $p(\boldsymbol{z}_i|\mathrm{H}_0)$ die Untersuchung der Hypothese H_0 mit Hinblick auf die Beobachtung $\boldsymbol{z}_i$. Sie kann angenähert werden, indem eine Partikelmenge ausgewertet wird, die gleichverteilt über den gesamten Suchraum verteilt ist. Da das Anwendungsszenario dieser Arbeit die Detektion von Fußgängern auf Landstraßen bei Nacht darstellt, enthällt eine solche Menge fast ausschließlich Hintegrundhypothesen. Das Ereignis „Fußgänger“ ist schließlich vergleichsweise selten.

Ein gleichmäßig verteilter Hypothesensatz repräsentiert also eine Menge von H_0-Hypothesen. Werden diese durch die Likelihood des Partikelfilters gewichtet, kann $p(\boldsymbol{z}_i|\mathrm{H}_0)$ in Analogie zu (6.3) durch den Mittelwert der resultierenden Gewichte approximiert werden.

Ein solcher gleichmäßig verteilter Hypothesensatz ist aus der Initialisierung in Kapitel 6.2 bereits vorhanden. Die Berechnung von $p(\boldsymbol{z}_i|\mathrm{H}_0)$ lässt sich damit ohne nennenswerten Mehraufwand realisieren. Ein weiterer Vorteil ist, dass die Approximationen sowohl für $p(\boldsymbol{z}_i|\mathrm{H}_1)$ als auch für $p(\boldsymbol{z}_i|\mathrm{H}_0)$ ohne zusätzliche Parameter auskommen. Darüber hinaus ist die Reinitialisierung unabhängig von einer Schwelle. Durch die Likelihood-Ratio erfolgt eine dynamische Anpassung des Hintergrundrauschens an den Bildinhalt. In schwierigen Szenen mit stark strukturierten Bildern wird $p(\boldsymbol{z}_i|\mathrm{H}_0)$ groß und die Fußgängererkennung entsprechend schwieriger.

Der gesamte Algorithmus zur Fußgängererkennung mit Partikelfilter ist nochmals in Algorithmus 6.1 zusammengefasst. Mit $\mathcal{H}^{\mathrm{init}}\,(0.6, 0.6, 0.6)$, $M = 10$ und $M^* = 4$ werden damit im FIR-solo Fall zur Fußgängerdetektion lediglich 7 823 Hypothesen, im Fusionsfall im Mittel 40 410 Hypothesen berechnet. Eine genaue Auswertung - insbesondere auch der Erkennungsleistung - gibt das folgende Kapitel im Abschnitt 7.4.

Sei $\mathcal{H}^{\text{init}}$ eine Hypothesenmenge mit grober Rasterung (z.B. $\mathcal{H}^{\text{init}}(0.6, 0.6, 0.6)$) und sei M die Anzahl der Trackerinstanzen sowie M^* die Anzahl der Trackerinstanzen, die zu jedem Zeitschritt maximal reinitialisiert werden sollen.
Für jede Iteration $i \geq 0$:

1. Berechne Rückschlusswahrscheinlichkeiten der Hypothesen in $\mathcal{H}^{\text{init}}$.
2. Führe eine Clusterung dieser Hypothesen auf Basis $\text{cov}_{\text{cluster}}$ (in dieser Arbeit $\text{cov}_{\text{cluster}} = 0.1$) zur Bestimmung der a priori Dichten zur Initialisierung durch.
3. Für jede der M Trackerinstanzen:
   (a) Muss Trackerinstanz reinitialisiert werden?
   Ja: Initialisiere Partikelmenge anhand a priori Dichte aus Schritt 2.
   Nein: Führe Resampling und Prädiktion durch (vgl. Condensation Algorithmus 5.2).
   (b) Gewichte Partikel anhand der Rückschlusswahrscheinlichkeiten der zugehörigen Hypothesen.
   (c) Behandle Kollisionen mit höher priorisierten (d.h. bereits abgearbeiteten) Partikelfilterinstanzen und setze die entsprechenden Partikelgewichte auf null.
4. Bestimme Gewinner anhand des MAP-Schätzers und entscheide über Detektion anhand des unnormalisierten Partikelgewichtes.
5. Markiere die schlechtesten Trackerinstanzen zur Reinitialisierung, so dass insgesamt bis zu M^* Trackerinstanzen zur Reinitialisierung vorgesehen sind.
6. Normiere die Gewichte aller Trackerinstanzen.

Algorithmus 6.1: Fußgängererkennung mit Partikelfilter.

KAPITEL 7

Systemevaluierung

Für die technische Sicht der Bewertung eines Erkennungssystems stehen zwei Fragen im Vordergrund:

- Wie oft hat das System relevante Ereignisse nicht erkannt, die es gemäß Spezifikation hätte erkennen sollen?
- Wie oft hat das System vermeintliche Ereignisse gemeldet, die in Wirklichkeit gar nicht stattgefunden haben?

Bei der sinnvollen Bewertung eines Gesamtsystems ist als Ereignis das generelle Vorhandensein eines Fußgängers zu benennen - unabhängig davon, wie lange der jeweilige Fußgänger im Bild sichtbar ist. Die 100 Fußgänger, die auf 100 zeitlich zusammenhängenden Bildern vorkommen, entsprechen in Wirklichkeit unter Umständen nur einer einzigen Person. Sie ist bei der Vorbeifahrt mit dem Fahrzeug ca. 4 Sekunden zu sehen. Sie dabei 100 mal immer wieder zu erkennen, darf nicht als großartige Leistung bewertet werden, sofern eine Minute später ein anderer Fußgänger komplett übersehen wird. Diese Betrachtungsweise allein reicht jedoch nicht aus, denn im Hinblick auf ein warnendes Nachtsichtsystem genügt es nicht, den Fußgänger einfach nur zu erkennen, sondern er muss so früh wie möglich erkannt werden. Zusätzlich muss man berücksichtigen, dass die Qualität eines solchen Gesamtsystems von vielen verschiedenen Einzelkomponenten abhängt: neben der eigentlichen Detektorkomponente sind hier natürlich die Suchstrategie, das Tracking der Einzeldetektionen, aber auch die Warnstrategie selbst zu nennen. In dieser Arbeit kommt es dabei vor allem auf den Vergleich der Qualität der unterschiedlichen Detektoren (NIR-Solo, FIR-Solo und Fusionsdetektor), sowie die Performanz der verschiedenen Suchstrategien an.

Um den Einfluss der jeweils anderen Komponenten auszuschließen, werden diese Teilsysteme einzeln bewertet. Insbesondere wird für einen fairen Vergleich die Tracking-Komponente außer Acht gelassen und Detektoren, sowie Aufwand der Suchstrategien auf Einzelbildbasis bewertet. Dabei muss sichergestellt sein, dass alle Experimente unter den gleichen Rahmenbedingungen auf den gleichen Datensätzen durchgeführt werden.

Man muss sich bewusst sein, dass damit zwar ein quantitativer Vergleich unterschiedlicher Umsetzungen einer Einzelkomponente möglich ist, alle Zahlen jedoch in Bezug auf die Bewertung des Gesamtsystems nur beschränkt aussagekräftig sind.

Der folgende Abschnitt 7.1 führt Maßzahlen zur bildbasierten Bewertung der Detektionskomponente ein und stellt die verwendeten Datensätze vor. Auf Basis dieser Auswertemethodik werden in Abschnitt 7.2 die Einzelsensordetektoren dem Fusionsdetektor gegenüber gestellt. Abschnitt 7.3 und 7.4 beleuchten dann die unterschiedlichen Suchstrategien in erster Linie unter dem Aspekt des Detektionsaufwandes. Jeder der Abschnitte endet dabei mit einer Zusammenfassung und einem Fazit über die jeweiligen Vergleiche.

7.1. Auswertungsmethodik

Primäres Merkmal zum Vergleich der unterschiedlichen Detektoren in dieser Arbeit sind sogenannte ROC-Kurven (engl. „Receiver Operating Characteristic“). Sie basieren auf folgenden Maßzahlen:

- Erkennungsrate:
$$D = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad \text{mit}$$
TP (engl. „True Positive“): Anzahl richtig detektierter Objekte.
FN (engl. „False Negative“): Anzahl der Objekte, die nicht vom Detektor gefunden worden sind.
- Falschalarme pro Bild:
$$F = \frac{\mathrm{FP}}{N}, \quad \text{mit}$$
FP (engl. „False Positive“): Anzahl der Detektionen, die nicht zu einem Objekt zugeordnet werden können.
N: Anzahl der Bilder im Testdatensatz.

Die Erkennungsrate ist der Anteil der richtig detektierten Fußgänger bezogen auf die Gesamtzahl der zu detektierenden Fußgänger im Datensatz. Die Falschalarme pro Bild sind die fälschlicherweise gemeldeten Detektionen im Mittel pro Bild. In ROC-Kurven ist die Erkennungsrate gegenüber der Falschalarmrate aufgetragen und zwar jeweils für unterschiedliche Arbeitspunkte des Detektors. Klassischerweise werden diese Arbeitspunkte durch sukzessives Abschalten der letzten Kaskadenstufen erhalten.

Dabei wird die Schwelle des jeweils letzten Stronglearners kontinuierlich gesenkt, bis er komplett abgeschaltet ist. Dies bewirkt, dass immer mehr Hypothesen die nächste Stufe erreichen bis keine Hypothesen mehr verworfen werden. Angefangen bei der letzten Kaskadenstufe gewinnt man so alle Arbeitspunkte, welche in die ROC eingetragen werden.

Durch die Verfügbarkeit von Rückschlusswahrscheinlichkeiten kann an Stelle der erreichten Kaskadenstufe die Rückschlusswahrscheinlichkeit selbst als Kriterium für eine Detektionsentscheidung gewählt werden. Der Arbeitspunkt ist dann eine Schwelle der Rückschlusswahrscheinlichkeit, ab der eine Hypothese als Detektion gewertet wird. Durch sukzessives Absenken der Schwelle erhält man wiederum die ROC-Kurve. Das bedeutet jedoch, dass alle Hypothesen (also auch diejenigen, die bereits in der ersten Kaskadenstufe verworfen wurden) protokolliert werden müssen. Im Gegensatz dazu reicht es bei der Layer-basierten Auswertung aus, erst ab einer bestimmten Kaskadenstufe zu protokollieren. Aus diesem Grund - und aufgrund des zusätzlich hohen Aufwandes bei der Bestimmung der empirischen Wahrscheinlichkeiten anhand eines sowohl vom Lernset als auch vom Testset unabhängigen dritten Datensatzes (Validierungsdatensatz, vgl. Kapitel 3.5), wird zum Vergleich des Fusionsklassifikators mit den Einzelsensor-Klassifikatoren das klassische Verfahren zur Bestimmung der ROC-Kurve verwendet.

Zur Ermittelung der ROC-Kurve wird der jeweilige Detektor auf dem Testdatensatz angewandt und zu jeder Hypothese die erreichte Kaskadenstufe, die Aktivierung des jeweils letzten Stronglearners und die Rückschlusswahrscheinlichkeit protokolliert. Die Hypothesen dieser Ist-Daten werden dann mit den jeweiligen Rechtecken der Soll-Daten (also den gelabelten Daten) verglichen um jeweils zu entscheiden, ob es sich um einen Falschalarm oder eine richtige Detektion handelt. Umgekehrt wird für jedes Rechteck der Soll-Daten überprüft, ob es dazu auch eine entsprechende Hypothese aus den Ist-Daten gibt. Als Deckungskriterium eines Rechtecks aus den Ist-Daten mit einem Rechteck aus den Soll-Daten dient dabei das Überdeckungsmaß (2.6) aus Kapitel 2.4. Eine Detektion gilt genau dann als einem Label zugeordnet, wenn die Überdeckung des jeweiligen Hypothesenfensters größer als 0.3 ist. Zusätzlich gelten - vor allem im Hinblick auf Mehrfachdetektionen - folgende Regeln:

1. a) Eine Menge von Detektionen, die alle zu ein und demselben Label zugeordnet werden, werden nur als eine einzelne, richtige Detektion gewertet. Dies entspricht damit der gängigen Praxis, im Gesamtsystem Mehrfachdetektionen zu einer einzelnen Detektion zusammenzufassen (z.B. durch einen einfachen Clusteringschritt oder der Datenassoziationskomponente eines nachgeschalteten Trackers).

   b) Entsprechend werden Mehrfachdetektionen, die keinem Label zugeordnet werden können, lediglich als ein False Positive gewertet. Dies geschieht vor allem im Hinblick auf die Vergleichbarkeit der unterschiedlichen Klassifikatoren, die durch unterschiedliche Suchraster (der FIR-Sensor hat z.B. eine deutlich geringere Auflösung als der NIR-Sensor) ein unterschiedliches Verhalten in Bezug auf Mehrfachdetektionen hervorrufen.

2. a) Ein Label mit den Attributen „occluded“ oder „part“ wird nicht als False Negative gewertet, da nicht davon ausgegangen werden kann, dass ein verdeckter oder nur teilweise sichtbarer Fußgänger erkannt wird.

   b) Entsprechend wird eine Detektion, die einem solchen Label zugeordnet wird, nicht als „False Positive“ gewertet.

Schränkt man die jeweiligen Soll- und Ist-Daten auf nur bestimmte Skalierungen ein, kann mit den damit entstehenden ROC-Kurven auch die Detektionsreichweite der unterschiedlichen Detektoren verglichen werden.

Zum Vergleich von Fusions- und Solodetektoren wird zusätzlich die Struktur der unterschiedlichen Klassifikatoren hinsichtlich der Anzahl der Merkmale pro Hypothese untersucht.

Zum Vergleich der unterschiedlichen Suchstrategien ist außerdem die Anzahl der berechneten Hypothesen pro Bild von Interesse.

Datensätze

Alle Untersuchungen wurden jeweils mit denselben Lern- und Testdatensätzen durchgeführt. Der Lerndatensatz umfasst dabei 900 einzelne Bildsequenzen der Länge 5s bis 45s, die innerhalb eines Zeitraumes von über 2 Jahren bei unterschiedlichen Temperatur- und Witterungsbedingungen aufgenommen wurden. Insgesamt besteht der Lerndatensatz aus 159 599 Bildpaaren (FIR/NIR) mit 163 691 markierten Fußgängern (Label). Trotz der großen Anzahl der Bilder handelt es sich dabei lediglich um Bildmaterial der Länge 1h 46min und umfasst 711 verschiedene Fußgängertracks. Bei 22% der Bildpaare handelt es sich um Aufnahmen aus stark beleuchteten Innenstädten. Die restlichen Szenen wurden hauptsächlich auf dunklen Landstraßen oder mäßig beleuchteten Vorstädten bzw. Industriegebieten aufgezeichnet.

Der vom Lerndatensatz unabhängige Testdatensatz umfasst 40 790 Bildpaare mit 61 471 markierten Fußgängern bei einem Anteil der Bilder von 28% aus Innenstädten. Der Testdatensatz entspricht einem Bildmaterial der Länge 27 min und umfasst 297 Fußgängertracks. Alle Szenen stammen dabei von Tagen, die im Lerndatensatz überhaupt nicht vertreten sind. Dadurch wird sichergestellt, dass keine statistischen Abhängigkeiten zwischen Lern- und Testdatensatz bestehen (z.B. über die Außentemperatur oder der Kleidung der Probanden).

Einen detaillierten Überblick über die verschiedenen Datensätze geben Tabelle 7.1 und Tabelle 7.2. Abbildung 7.1 zeigt aus zufällig ausgewählten Sequenzen jeweils ein Bild vom Anfang, aus der Mitte und vom Ende der Sequenz.

Insgesamt ist der Datensatz als sehr schwierig einzustufen, mit großer Posenvielfalt der Fußgängerbeispiele, unterschiedlichen Wetterbedingungen (insbesondere auch Regen) und teilweise sehr stark strukturierten Szenen aus Innenstädten. Der Datensatz ist dabei auch keineswegs repräsentativ, da z.B. viele der Landstraßenszenen selektiv auf Basis von Falschalarmen früherer Detektorvarianten aufgenommen wurden. Es wird auch aus

	Bildpaare		Labels		Fußgängertracks	
	Anzahl	Anteil	Anzahl	Anteil	Anzahl	Anteil
Innenstadt	34 400	22%	50 294	31%	310	44%
Landstraße	125 199	78%	113 397	69%	401	56%
Gesamt	159 599		163 691		711	

Tabelle 7.1.: Lerndatensatz.

	Bildpaare		Labels		Fußgängertracks	
	Anzahl	Anteil	Anzahl	Anteil	Anzahl	Anteil
Innenstadt	11 438	28%	16 667	27%	105	35%
Landstraße	29 352	72%	44 804	73%	192	65%
Gesamt	40 790		61 471		297	

Tabelle 7.2.: Testdatensatz.

der Verteilung der Fußgänger im Landstraßendatensatz klar, dass sich eine Übertragung der Kennzahlen aus diesem Kapitel auf ein mögliches Gesamtsystem verbietet: fast 200 Fußgängertracks in nur knapp 30 000 Bildern (das entspricht lediglich 20 Minuten) wird in Deutschland eher zur Ausnahme gehören. Dennoch ist gerade dieser schwierige Datensatz geeignet, Fusionssystem und Einzel-Sensor Systeme gegeneinander antreten zu lassen.

7.2. Fusion vs. FIR-Solo vs. NIR-Solo

Zum Vergleich des Multi-Sensor Fußgängererkennungssystems mit den jeweiligen Einzel-Sensor Fußgängererkennungssystemen werden jeweils Einzel-Sensor Fußgängerklassifikatoren (Solo-Klassifikatoren) auf Basis von FIR- bzw. NIR-Bildern trainiert und dem Fusionsklassifikator gegenübergestellt. Die Parametrisierung beim Training von AdaBoost ist dabei (sofern nicht sensorspezifisch) gleich und ist in Tabelle 7.3 aufgelistet.

Eine besondere Schwierigkeit stellt die Wahl des Basissuchfensters dar. Wichtigste Einschränkung ist dabei die zu erzielende Reichweite des Detektors in der Anwendung, denn das Basissuchfenster ist gleichzeitig das kleinste mögliche Suchfenster und schränkt damit die erzielbare Reichweite ein. Die Wahl eines kleinen Suchfensters ist also Voraussetzung für eine hohe Detektionsreichweite. Andererseits hat die Größe des Basissuchfensters Einfluss auf den überbestimmten Merkmalssatz (vgl. Kapitel 3.1). Ein größeres Basissuchfenster hat einen größeren Merkmalssatz zur Folge, aus dem AdaBoost Merkmale während des Trainings auswählen kann. Dies führt tendenziell zu besseren Klassifikatoren. Schließlich gibt es noch praktische Einschränkungen beim Training der Klassifikatoren, denn ein zu großer Merkmalssatz (also ein zu großes Basissuchfenster) erschöpft die Ressourcen (insbesondere den zur Verfügung stehenden Arbeitsspeicher) des Trainingsrechners.

(a) Landstraße

(b) Stadt

Abbildung 7.1.: Beispielsequenzen aus dem Datensatz. In jeder Zeile ist jeweils ein Bild vom Anfang, ein Bild aus der Mitte und ein Bild vom Ende der Sequenz dargestellt. Abbildung 7.1(a) zeigt vier Landstraßensequenzen, Abbildung 7.1(b) zeigt Sequenzen, die in Innenstädten aufgenommen wurden.

Parameterwahl	Beschreibung (vgl. Kapitel 3.4)
$d_k^* = \begin{cases} 0.1 & 1 \leq k \leq 10, \\ 0.3 & k > 10 \end{cases}$	vorgegebene minimale Detektionsraten auf dem Lerndatensatz
$f_k^* = \begin{cases} 0.99 & 1 \leq k \leq 2, \\ 0.995 & k > 2 \end{cases}$	vorgegebene maximale Falschalarmraten auf dem Lerndatensatz
$N_{\text{pos}}^* = 40\,000$	Anzahl der positiven Beispiele, mit der jede Stufe trainiert werden soll
$N_{\text{neg}}^* = 100\,000$	Anzahl der negativen Beispiele, mit der jede Stufe trainiert werden soll (Bootrapping)
$T_k^{\max} = \begin{cases} 4 & k = 1, \\ 6 & k = 2, \\ 8 & k = 3, \\ 15 & k = 4, \\ 25 & k = 5, \\ 50 & 6 \leq k \leq 8, \\ 100 & 9 \leq k \leq 12, \\ 200 & 16 \leq k \leq 13 \end{cases}$	maximale Anzahl von Weaklearnern für Stufe k

Tabelle 7.3.: Parametrisierung beim Training von AdaBoost.

Für den Vergleich der Einzel-Sensor Fußgängerklassifikatoren mit dem Fusionsklassifikator werden folgende Basissuchfenster verwendet:

- Basissuchfenster in FIR-Bildern: 7px × 11px.
- Basissuchfenster in NIR-Bildern: 7px × 14px

Die Größen der Objektfenster (vgl. Kapitel 2.4) sind:

- Objektfenster im FIR-Basissuchfenster: 5px × 7px.
- Objektfenster im NIR-Basissuchfenster: 5px × 10px.

Die Objektfenster liegen immer zentriert innerhalb der Suchfenster. Mit dieser Parametrisierung kann aus theoretischer Sicht ein Fußgänger der Größe 1.80m frühestens in einer Entfernung von 127m erkannt werden.

Der Bootstrapping-Schritt zur Auswahl der N_{neg}^* Negativ-Beispiele zum Training einer Kaskadenstufe wird jeweils mit Hilfe des Hypothesengenerators aus Kapitel 4.1 bzw. Kapitel 4.2 durchgeführt. In allen Fällen ist die Rasterschrittweite $(Q_{\text{col}}, Q_{\text{row}}, Q_h) = (0.03, 0.05, 0.08)$ bei einem Relaxationswinkel von $\rho = 2°$. Das Training ist beendet, wenn für eine Stufe weniger als 100 Negativbeispiele gefunden werden (erschöpfendes Training).

Damit werden im NIR-Fall pro Bild 676 088 Hypothesen, im FIR-Fall pro Bild 302 859 Hypothesen und im Fusionsfall pro Bildpaar 1 395 330 Hypothesen ausgewertet. Mit dem in Abschnitt 7.1 beschriebenen Lerndatensatz mit 159 599 Bildpaaren werden beim Training aller Kaskadenstufen (also bis mittels Bootstrapping keine geeigneten Negativbeispiele mehr gefunden werden können) bis über 200 000 000 000 (in Worten 200 Milliarden) Negativbeispiele berücksichtigt. Das Training von jeweils einem der Klassifikatoren dauerte jeweils zwischen 168h (7 Tage) und 480h (20 Tage).

Struktureller Vergleich

Der fertig trainierte FIR-Solo Klassifikator hat 24 Stufen mit insgesamt 2 284 Merkmalen (Abbildung 7.2(a)). Der NIR-Solo Klassifikator ist mit 40 Stufen und 5 891 Merkmalen deutlich aufwändiger (Abbildung 7.2(b)). Der Fusionsklassifikator weist 28 Stufen mit lediglich 1 571 Merkmalen auf. Bei letzterem wurden von AdaBoost in allen Stufen (ausgenommen der ersten) sowohl FIR- als auch NIR-Merkmale ausgewählt (Abbildung 7.2(c)).

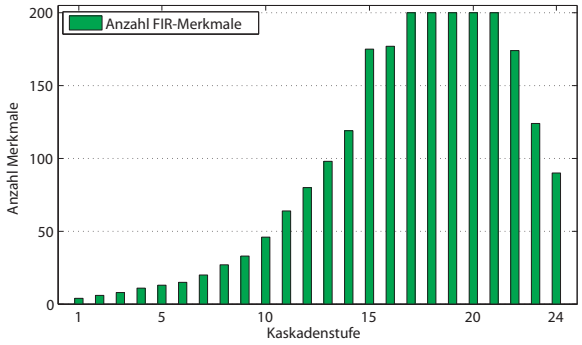
Bereits diese strukturellen Kennzahlen geben erste Hinweise auf die Qualität der unterschiedlichen Klassifikatoren. Offensichtlich hat der NIR-Klassifikator im Vergleich zum FIR-Klassifikator größere Mühe, die Positiv- und Negativbeispiele im Trainingsdatensatz zu trennen. Darüber hinaus scheint der Fusionsklassifikator besser zu sein als beide Solo-Varianten. Es findet auch tatsächlich eine Fusion auf Merkmalsebene statt, denn obwohl die FIR-Merkmale für die gestellte Aufgabe scheinbar geeigneter sind als die NIR-Merkmale, wählt AdaBoost beim Training des Fusionsklassifikators einen signifikanten Anteil (30% - 50%) an NIR-Merkmalen aus (Abbildung 7.2). In der Kombination kommt der Fusionsklassifikator mit weniger Merkmalen aus als jeder der Solo-Klassifikatoren. Ein Blick auf die konkreten Merkmale der ersten Stufen des Fusionsklassifikators gibt Abbildung 7.3.

Auswertungen auf dem Testdatensatz

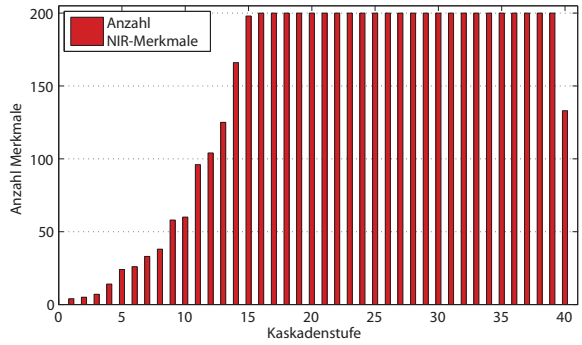
Die Einschätzung, dass der Fusionsklassifikator besser ist, als beide Solo-Klassifikatoren, erhärtet sich mit Blick auf die Ergebnisse des Testdatensatzes. Abbildung 7.4 zeigt die ROC-Kurven aller drei Klassifikatoren. Der FIR-Klassifikator wurde dabei ab Stufe 18, der NIR-Klassifikator ab Stufe 30 und der Fusionsklassifikator ab Stufe 21 ausgewertet[1]. Die Detektionsrate des Fusionsdetektors ist mit 93% bei 0.025 Falschalarmen pro Bild um fast 15 Prozentpunkte besser als der FIR-Solo Klassifikator (mit 78% Detektionsrate). Der NIR-Klassifikator erreicht auf diesem schwierigen Datensatz lediglich 60% Erkennungsrate, das aber auch nur bei fast fünf mal höherer Falschalarmrate.

Schränkt man die Auswertung nur auf Landstraßenszenarien ein, sieht das Bild für den NIR-Klassifikator nicht mehr ganz so negativ aus (Abbildung 7.5). Der NIR-Detektor

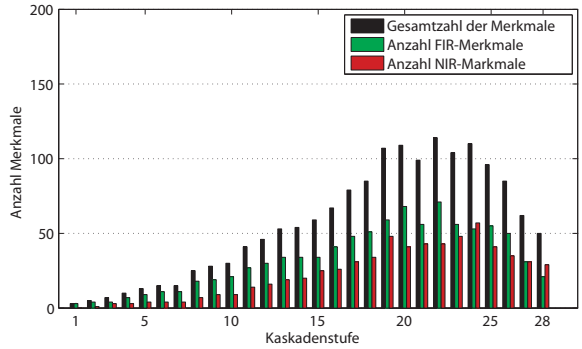
[1] Zur Erinnerung: zur Berechnung der ROC-Kurven müssen alle Detektionen protokolliert werden. Handhabbar ist dies erst ab einer vergleichsweise späten Stufe (vgl. Kapitel 7.1).



(a) Struktur des FIR-Klassifikators

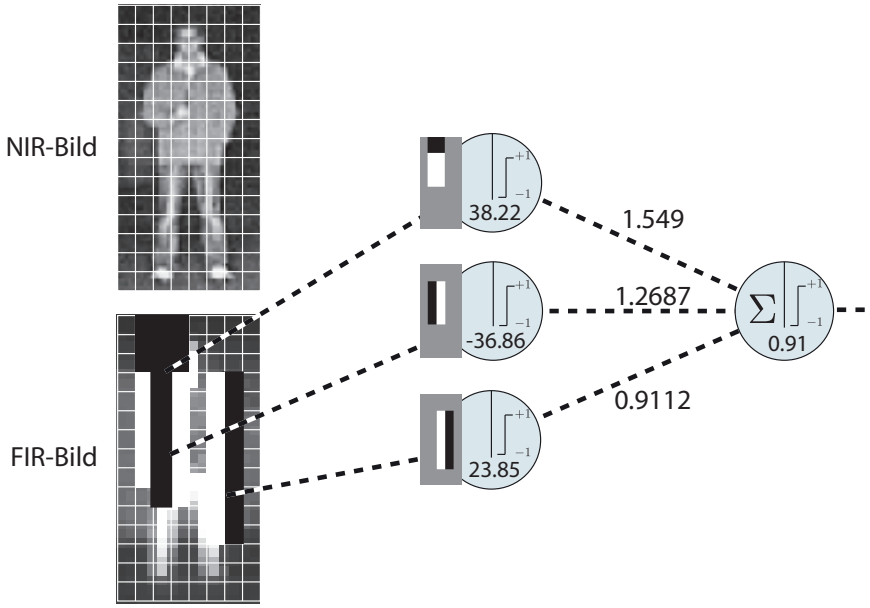


(b) Struktur des NIR-Klassifikators

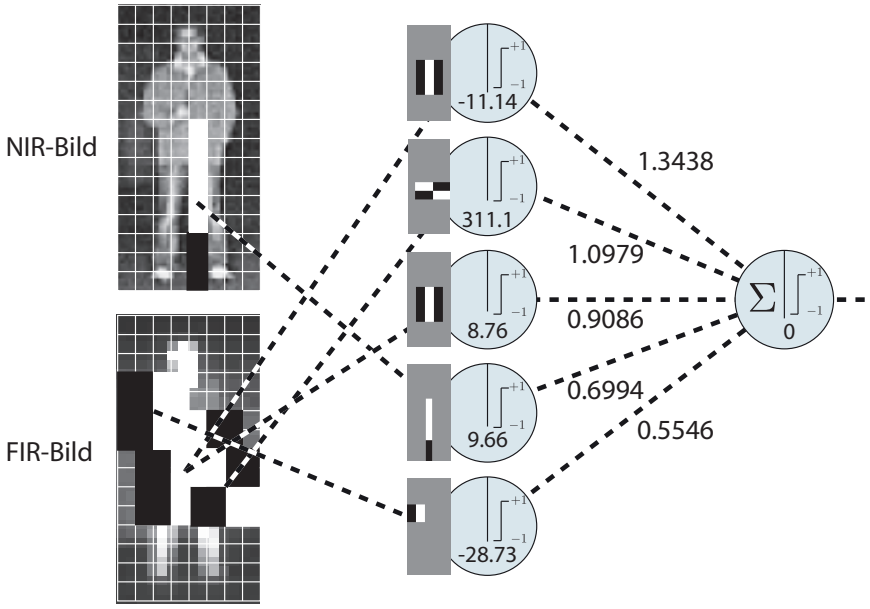


(c) Struktur des Fusionsklassifikators

Abbildung 7.2.: Struktur der trainierten Klassifikatoren. Dargestellt ist jeweils die Anzahl der Merkmale pro Kaskadenstufe. Der FIR-Klassifikator (7.2(a)) hat 2 284 Merkmale auf 24 Kaskadenstufen, der NIR-Klassifikator (7.2(b)) benötigt für diesselbe Aufgabe 5 891 Merkmale in 40 Stufen. Im Fusionsfall (7.2(c)) werden nur 1 571 Merkmale in 28 Stufen benötigt. Die Aufschlüsselung der verwendeten Merkmale im Fusionsklassifikator zeigt außerdem, dass in jeder Stufe (ausgenommen der ersten) zwischen 30% und 50% der Merkmale aus den NIR-Bildern stammen.



(a) 1. Kaskadenstufe



(b) 2. Kaskadenstufe

Abbildung 7.3.: Merkmale, Weaklearner und Stronglearner der ersten Stufen des Fusionsdetektors. Oben sind die Merkmale und Weaklearner (einschließlich der Weaklearnerschwellen und Gewichte) der ersten Kaskadenstufe, unten diejenigen der zweiten Kaskadenstufe dargestellt. Im FIR-Bild wurden tendenziell eher große Filter gewählt, im NIR-Datenstrom erstmals in der zweiten Stufe ein kleiner Filter, der den Schritt des Fußgängers beschreibt, welcher nur im NIR-Bild sichtbar ist.

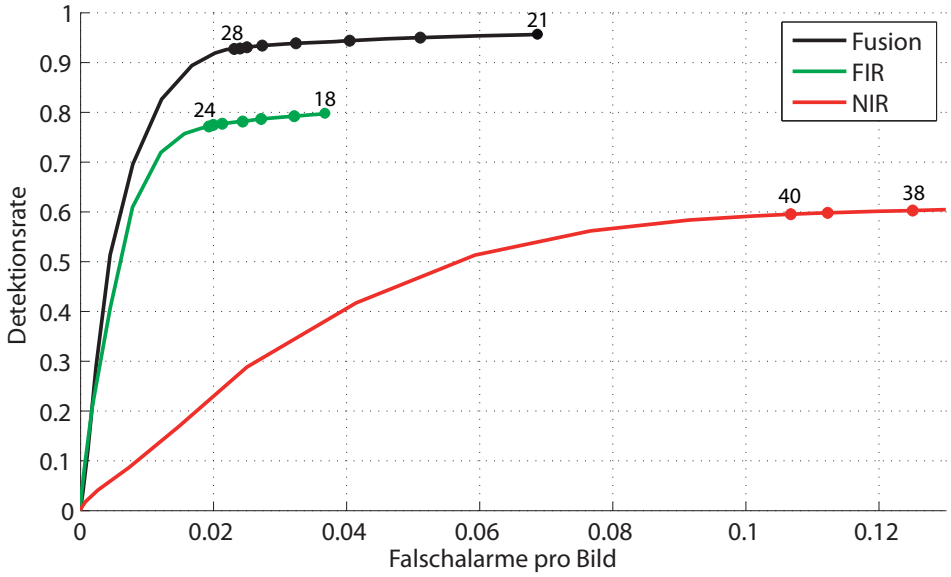


Abbildung 7.4.: Vergleich der ROC-Kurven des FIR-, NIR- und Fusionsklassifikators. Die Auswertung basiert auf dem kompletten Testdatensatz im maximal möglichen Entfernungsbereich bis 127m vor dem Fahrzeug. Die (nummerierten) Punkte stellen jeweils die einzelnen Kaskadenstufen dar.

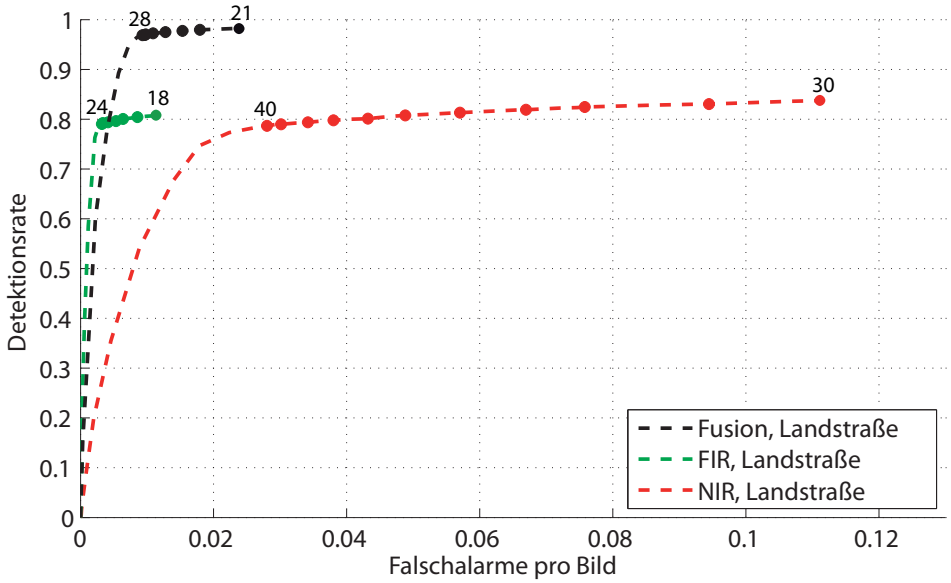


Abbildung 7.5.: Vergleich der ROC-Kurven des FIR-, NIR- und Fusionsklassifikators auf Landstraßenszenen. Die Auswertung basiert auf dem kompletten Testdatensatz im maximal möglichen Entfernungsbereich bis 127m vor dem Fahrzeug. Die (nummerierten) Punkte stellen jeweils die einzelnen Kaskadenstufen dar.

erreicht hier mit 79% dieselbe Erkennungsrate wie die FIR-Variante, wenn auch nach wie vor mit einer deutlich höheren Zahl an Falschalarmen im Vergleich (0.028 Falschalarme pro Bild beim NIR-Detektor gegenüber 0.003 Falschalarmen beim FIR-Detektor). Die NIR-Solo Variante ist also für den Einsatz in der Stadt nicht geeignet, kann aber auf Landstraßenszenen seine Leistung deutlich steigern. Die Erkennungsrate von fast 80% – einzelbildbasiert und auf einem schwierigen Datensatz – ist durchaus für den Einsatz in einem Komfortsystem im Fahrzeug denkbar.

Überraschend ist, dass auch der FIR-Klassifikator nicht über 80% Erkennungsrate auf Landstraßen kommt. Lediglich die Zahl der Falschalarme ist deutlich reduziert und ist hier auch um ein Vielfaches kleiner als im Fusionsfall, in dem allerdings mit über 95% Erkennungsrate nahezu kein Fußgänger übersehen wird.

Das überraschend schlechte Abschneiden des FIR-Klassifikators erscheint in einem anderen Licht, wenn man sich klar macht, dass die minimale Objektfenstergröße nur 5px $\times$ 7px ist. Der limitierende Faktor ist in erster Linie die geringe Auflösung des Sensors. Dies wird deutlich, wenn man bei der Berechnung der ROC-Kurven nur Fußgänger bis in eine Entfernung von 90m berücksichtigt. Abbildung 7.6 stellt die ROC-Kurven des FIR-Klassifikators für die verschiedenen Datensätze „Landstraße“, „Stadt“ und „gesamter Testdatensatz“ jeweils einmal für den gesamten Entfernungsbereich bis 127m, und einmal für den eingeschränkten Entfernungsbereich bis 90m dar[2]. Vor allem in Landstraßenszenarien ist die Erkennungsrate nun bis zu 10 Prozentpunkte gesteigert, wenn nur Fußgänger aus einer Entfernung $\leq$ 90m erkannt werden sollen. Die Unterschiede sind dabei in der Stadt nicht so ausgeprägt, da von vornherein ein kleinerer Anteil weit entfernter Fußgänger aus der Stadt im Datensatz (und in der Realität) vorhanden ist.

Interessanterweise zielt die Einschränkung auf nahe Fußgänger keine nennenswerten Änderungen der Kennzahlen des NIR-Klassifikators nach sich (Abbildung 7.7). Dies legt den Schluss nahe, dass die Auflösung des NIR-Sensors zur Detektion von Fußgängern bis in 127m zumindest ausreichend ist. Im Gegensatz dazu ist offensichtlich die geringe Auflösung des FIR Sensors der limitierende Faktor bei der Reichweite eines FIR-Fußgängererkennungssystems. Es ist also zu erwarten, dass mit höher aufgelösten FIR-Bildern auch deutlich bessere FIR-Klassifikatoren trainiert werden können.

Interessant ist, dass bei denselben Betrachtungen in Bezug auf das Fusionssystem (Abbildung 7.8) die Auflösung des FIR-Sensors keine Rolle mehr zu spielen scheint. Offensichtlich kann das Fusionssystem in diesem Fall den limitierenden Faktor „Auflösung FIR-Sensor“ kompensieren, d.h. für das Fusionssystem ist kein höher aufgelöster FIR-Sensor nötig.

[2]Die Einschränkung des Entfernungsbereichs fließt dabei nur bei der Bestimmung der Erkennungsrate mit ein, da nicht klar ist, welche Entfernungen einem Falschalarm(-cluster) zugeordnet werden soll (vgl. Diskussion in Kapitel 6.1 zum Thema Entfernungsschätzung aus Kaskadendetektionen). Aus den selben Gründen ist es auch nicht sinnvoll für die Auswertung die Größe der Suchfenster einzuschränken.

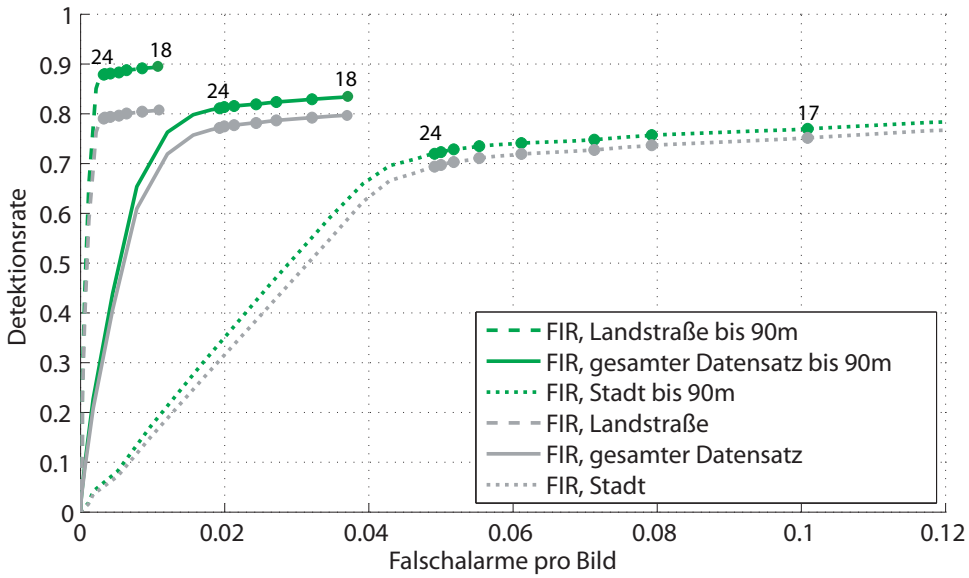


Abbildung 7.6.: Vergleich der ROC-Kurven des FIR-Klassifikators in unterschiedlichen Entfernungen. ROC-Kurven des FIR-Klassifikators für die verschiedenen Datensätze „Landstraße“, „Stadt“ und „gesamter Testdatensatz“, jeweils einmal für den gesamten Entfernungsbereich bis 127m und einmal für den eingeschränkten Entfernungsbereich bis 90m. Die Auswertung der Falschalarme bleibt davon unberührt. Die (nummerierten) Punkte stellen die einzelnen Kaskadenstufen dar.

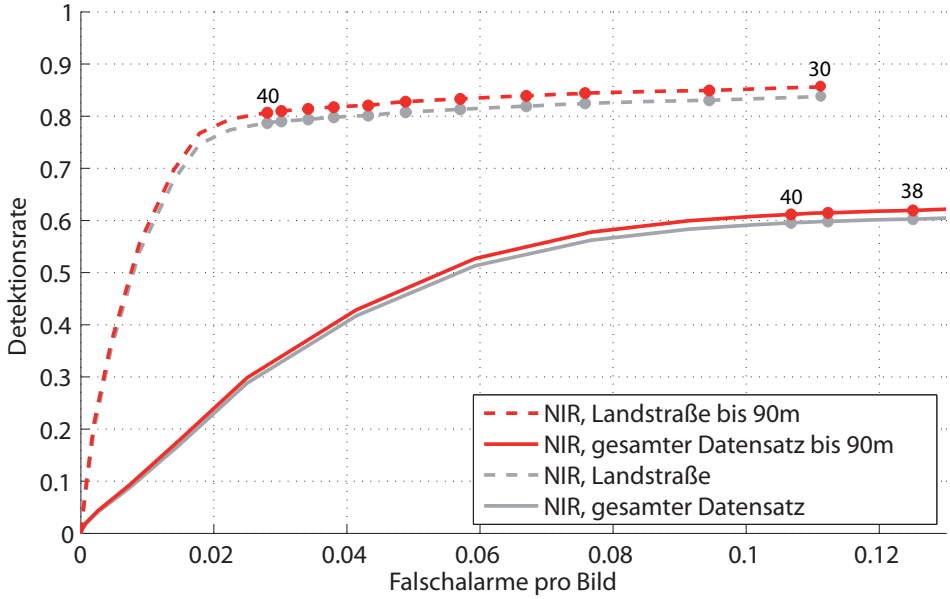


Abbildung 7.7.: Vergleich der ROC-Kurven des NIR-Klassifikators in unterschiedlichen Entfernungen. ROC-Kurven des NIR-Klassifikators für die Datensätze „Landstraße“ und „gesamter Testdatensatz“, jeweils einmal für den gesamten Entfernungsbereich bis 127m und einmal für den eingeschränkten Entfernungsbereich bis 90m. Die Auswertung der Falschalarme bleibt davon unberührt. Die (nummerierten) Punkte stellen die einzelnen Kaskadenstufen dar. Die ROC-Kurve zum Datensatz „Stadt“ wurde nicht dargestellt. Die Erkennungsrate bleibt auf diesem Datensatz unter 35%.

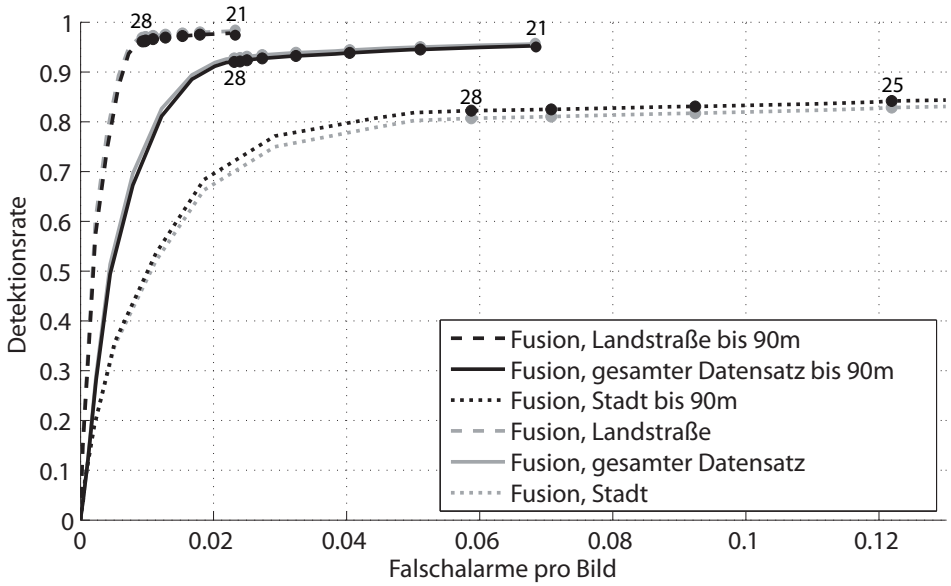


Abbildung 7.8.: Vergleich der ROC-Kurven des Fusionsklassifikators in unterschiedlichen Entfernungen. ROC-Kurven des Fusionsklassifikators für die verschiedenen Datensätze „Landstraße“, „Stadt“ und „gesamter Testdatensatz“ jeweils einmal für den gesamten Entfernungsbereich bis 127m und einmal für den eingeschränkten Entfernungsbereich bis 90m. Die Auswertung der Falschalarme bleibt davon unberührt. Die (nummerierten) Punkte stellen die einzelnen Kaskadenstufen dar.

Fazit Fusion vs. FIR-Solo vs. NIR-Solo

Zusammenfassend können auf Basis der Auswertungen folgende Aussagen getroffen werden:

- Der NIR-Solo Klassifikator ist mit 5 891 Merkmalen in 40 Stufen der aufwändigste Detektor und trotzdem deutlich schlechter als der FIR-Solo Klassifikator. Lediglich in Landstraßenszenen kann der NIR-Klassifikator diesselben Erkennungsraten erreichen, allerdings mit einer deutlich höheren Falschalarmrate. Für eine Anwendung in der Stadt ist der NIR-Klassifikator nicht geeignet.
- Der limitierende Faktor im FIR-Klassifikator ist die geringe Auflösung der FIR-Bilder. Im Bereich bis $\leq$ 90m vor dem Fahrzeug ist der FIR-Solo Klassifikator deutlich besser als im gesamten Entfernungsbereich bis 127m. Er erreicht dabei in Landstraßenszenen Detektionsraten von 90%.
- Der Fusionsklassifikator erreicht in allen Szenerien eine deutlich höhere Erkennungsrate als die beiden Einzel-Sensor Klassifikatoren, mit lediglich 1 571 Merkmalen in 28 Stufen. Bei nur 0.025 Falschalarmen pro Bild werden 93% der Fußgänger im Entfernungsbereich bis 127m erkannt. Der FIR-Klassifikator kann bei 0.025 Falschalarmen pro Bild lediglich 78% der Fußgänger erkennen.
- Auf Landstraßen erreicht der Fusionsklassifikator Erkennungsraten von 95%.
- Der Fusionsklassifikator ist anfälliger für Falschalarme als der FIR-Solo Klassifikator. Im optimalen Szenario für den FIR-Detektor (Entfernungsbereich $\leq$ 90m auf Landstraßen) weist der Fusionsklassifikator bei einer festen Erkennungsrate von 90% etwa doppelt so viele Falschalarme auf, wie der FIR-Klassifikator.
- Der Fusionsklassifikator kann die geringe Auflösung des FIR-Sensors kompensieren und ist im Entfernungsbereich bis 127m nicht wesentlich schlechter wie im Entfernungsbereich bis 90m. Weitere Untersuchungen auf anderen Datensätzen ([SFL+10]) haben jedoch gezeigt, dass noch kleinere Auflösungen des FIR-Sensors auch für den Fusionsklassifikator schwierig zu handhaben sind.

Alles in allem ist also der Fusionsklassifikator deutlich besser als die beiden Einzel-Sensor Varianten, und zwar bei deutlich kleinerem Aufwand (bezogen auf die Anzahl der zu berechnenden Merkmale) pro Hypothese.

Bezogen auf den Aufwand pro Bild (mit dem einfachen Hypothesengenerator, Kapitel 4.1 bzw. dem Multi-Sensor Hypothesengenerator, Kapitel 4.2) ist im Mittel der Aufwand für den Fusionsdetektor (mit $6.79 \cdot 10^6$ berechneten Merkmalen pro Bildpaar) nur unwesentlich kleiner als der des NIR-Klassifikators (mit im Mittel $7.26 \cdot 10^6$ Merkmalen pro Bild). Im Gegensatz dazu benötigt der FIR-Klassifikator im Mittel nur $0.89 \cdot 10^6$ Merkmalsberechnungen pro Bild. Der Aufwand ist natürlich stark abhängig von der Anzahl der Hpyothesen und damit vom verwendeten Hypothesengenerator. Inwieweit der Aufwand - bei gleich bleibender Qualität - mit der geeigneten Suchstrategie reduziert werden kann, ist Gegenstand des folgenden Abschnitts.

7.3. Hypothesengenerator vs. Hypothesenbaum

Neben einer guten Klassifikationsleistung ist für die Entwicklung eines Fahrerassistenzsystems auch der Berechnungsaufwand zur Detektion von Fußgängern im Bild von zentraler Bedeutung. Von Interesse sind dabei in erster Linie die Anzahl der zu berechnenden Hypothesen, sowie die Anzahl der zu berechnenden Merkmale pro Bild bzw. Bildpaar.

Die Anzahl der Hypothesen ist dabei abhängig

- von der gewählten Suchstrategie (Hypothesengenerator, Hypothesenbaum oder Partikelfilterverfahren) und dessen Parametrisierung, sowie
- von der Auflösung des Bildes bzw. der Bildpaare.

Die Anzahl der Merkmalsberechnungen pro Bild(-paar) ist dagegen abhängig von

- der Anzahl der Hypothesen,
- der Struktur des Klassifikators und
- dem Anteil der Hypothesen, welche eine hohe Kaskadenstufe erreichen (und damit abhängig von der Szenerie selbst).

Es kann durchaus vorkommen, dass ein Detektor zwar nur wenige Hypothesen pro Bild berechnet, aber viele Merkmalswerte berechnen muss, da viele der Hypothesen hohe Kaskadenstufen erreichen. Im Folgenden wird deshalb sowohl die mittlere und maximale Anzahl von Hypothesen im Bild, als auch die mittlere und maximale Anzahl von Merkmale pro Bild ausgewertet. Dabei wird in diesem Abschnitt zunächst der einfache Hypothesengenerator (Kapitel 4.1) dem Multi-Sensor Hypothesengenerator (Kapitel 4.2) gegenübergestellt, sowie die einfachen Hypothesengeneratoren mit dem Hypothesenbaum (Kapitel 4.3) verglichen. Die Auswertungen des Partikelfilterverfahrens erfolgen in Abschnitt 7.4.

Einfacher Hypothesengenerator

Beim Einsatz des einfachen Hypothesengenerators hängt die Anzahl der Hypothesen in starkem Maße vom Relaxationswinkel ρ und von den Abtastschrittweiten Q_{col}, Q_{row} und Q_h der skalierungsabhängigen Unterabtastung ab. Als Relaxationswinkel wurde in der gesamten Arbeit $\rho = 2°$ gewählt. Diese Wahl stellt eher die Untergrenze dar, da zur Kompensation der Annahme einer ebenen Welt vor allem beim Einfahren in eine Senke aber auch beim Überfahren einer Tempohemmschwelle ein weit höherer Relaxationswinkel nötig wäre. Die Wahl von $\rho = 2°$ reicht jedoch aus, um die typischen Nickbewegungen des Fahrzeugs bei normaler Fahrt weitgehend auszugleichen.

Dennoch darf der Relaxationswinkel bei der Systemauslegung nicht außer Acht gelassen werden, da er einen linearen Einfluss auf die Anzahl der Hypothesen hat (Abbildung 7.9 am Beispiel des einfachen NIR Hypothesengenerators). Eine Verdoppelung des

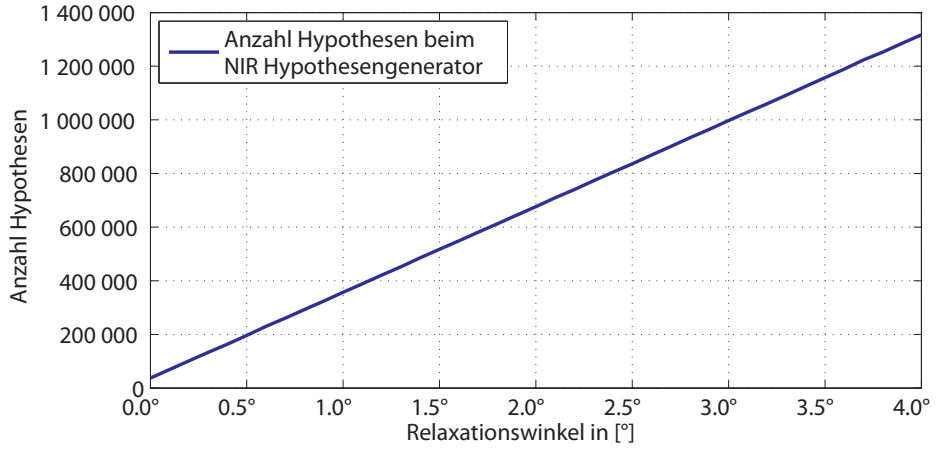


Abbildung 7.9.: Anzahl Hypothesen im einfachen NIR-Hypothesengenerator in Abhängigkeit vom Relaxationswinkel.

Relaxationswinkels geht auch mit einer Verdoppelung der Hypothesenanzahl einher. Kosten und Nutzen sind hierbei also sorgfältig abzuwägen.

Die feinste in dieser Arbeit verwendete Abtastschrittweite ist

$$\begin{aligned} Q_{\text{col}} &= 0.03, \\ Q_{\text{row}} &= 0.05, \\ Q_h &= 0.08. \end{aligned}$$

Die Wahl von $Q_h = 0.08$ motiviert sich vor allem durch die hohe Skalierungsinvarianz aller Klassifikatoren, die sich auch in den Mehrfachdetektionen bemerkbar macht (vgl. auch Diskussion in Kapitel 6.1). Die horizontale Abtastung mit $Q_{\text{col}} = 0.03$ ist darüber hinaus gegenüber der vertikalen Abtastung mit $Q_{\text{row}} = 0.05$ erhöht. Dies ist vor allem zur Detektion weit entfernter, seitlich zur Kamera mit geschlossenen Beinen stehenden Fußgängern von Vorteil. Die weiteren Modellparameter sind in Tabelle 7.4 aufgelistet. Aufgrund der unterschiedlichen Auflösungen der Kameras ergeben sich damit 676 088 Hypothesen im NIR-Bild und 180 408 Hypothesen im FIR Bild. Vor allem aufgrund der geringen Auflösung des FIR Sensors ist dort der Aufwand zur Detektion von Fußgängern per se also deutlich geringer. Betrachtet man die mittlere Anzahl der Merkmalsberechnungen pro Bild ist der Unterschied sogar noch gravierender: $7.26 \cdot 10^6$ Merkmalsberechnungen im NIR Bild gegenüber lediglich $0.89 \cdot 10^6$ Merkmalsberechnungen im FIR Bild.

Die Anzahl der Merkmalsberechnungen pro Bild ist dabei abhängig von der jeweiligen Szene. Vor allem in stark strukturierten Szenen ist die Anzahl der Merkmalsberechnungen (bei gleich bleibender Anzahl an Hypothesen) pro Bild deutlich größer. So benötigte der NIR Detektors auf Innenstadtszenarien angewandt im Mittel $10.30 \cdot 10^6$ Merkmale, der FIR Detektor $1.12 \cdot 10^6$ Merkmale. Das heißt jedoch nicht, dass es auf Landstraßen nicht auch Szenarien gibt, die eine hohe Anzahl an Merkmalsberechnungen erfordern. Ganz im Gegenteil - gemessen an der Anzahl an Merkmalsberechnungen

Parameterwahl	Beschreibung (vgl. Kapitel 4)
$h_{0,\mathrm{FIR}} = 7\,\mathrm{px}$	minimale Skalierung eines Fußgängers im FIR Bild
$h_{\max,\mathrm{FIR}} = 50\,\mathrm{px}$	maximale Skalierung eines Fußgängers im FIR Bild
$h_{0,\mathrm{NIR}} = 10\,\mathrm{px}$	minimale Skalierung eines Fußgängers im NIR Bild
$h_{\max,\mathrm{NIR}} = 170\,\mathrm{px}$	maximale Skalierung eines Fußgängers im NIR Bild
$H_{\min} = 1.60\,\mathrm{m}$	minimale Größe eines Fußgängers in der Welt
$H_{\max} = 2.00\,\mathrm{m}$	maximale Größe eines Fußgängers in der Welt
$\rho = 2.0°$	Relaxationswinkel
$\mathrm{tol}_{\mathrm{col}} = \mathrm{tol}_{\mathrm{row}} = 1\,\mathrm{px}$	Toleranzbereich beim Mapping von einem Sensor zum anderen (vgl. Algorithmus 4.4, Seite 95).

Tabelle 7.4.: Parametrisierung der Hypothesengeneratoren.

Abbildung 7.10.: Bilder mit den meisten Merkmalsberechnungen im Test. Das linke (NIR-) Bild stellt vor allem für den NIR Detektor eine Herausforderung dar. Er muss dazu $15.11 \cdot 10^6$ Merkmale berechnen. Für das mittlere (NIR-) Bild werden sogar $15.21 \cdot 10^6$ benötigt. Es handelt sich dabei um eine Aufnahme im Sommer mit Natriumdampflampen in den Straßenlaternen, die offensichtlich zu Problemen führen. Auch der Fusionsdetektor hat mit dieser Szenerie (das rechte Bild stellt das zugehörige FIR-Bild dar) Schwierigkeiten. Er muss $16.51 \cdot 10^6$ Merkmale berechnen.

stammen die schwierigsten Bilder im Test von Landstraßen bzw. aus vorstädtischen Bereichen (Abbildung 7.10).

Multi-Sensor Hypothesengenerator

Der Fusionsklassifikator ist den beiden Einzel-Sensor Klassifikatoren in seiner Leistung deutlich überlegen (Abschnitt 7.2). Um den Suchraum vollständig abzudecken, benötigt er jedoch deutlich mehr Hypothesen, nämlich 1 395 330 Hypothesen pro Bildpaar. Dabei wurde der FIR Sensor als Primärsensor verwendet, mit $\mathcal{H}^* = \mathcal{H}_{\mathrm{FIR}}\,(0.03, 0.05, 0.08)$ und $\mathcal{H}' = \mathcal{H}_{\mathrm{NIR}}\,(0.03, 0.05, 0.08)$, sowie $\mathrm{tol}_{\mathrm{col}} = \mathrm{tol}_{\mathrm{row}} = 1\mathrm{px}$. Wählt man den NIR Sensor als Primärsensor, ergeben sich sogar 5 041 340 Hypothesen. Trotz der großen Zahl an Hypothesen pro Bildpaar beim Fusionsdetektor ist im Mittel die Anzahl der berechneten Merkmale mit $6.79 \cdot 10^6$ sogar geringer als beim NIR-Detektor (mit im Mittel $7.26 \cdot 10^6$ Merkmalsberechnungen pro Bild). Im schlimmsten Fall ist der NIR-Detektor dennoch

			Anzahl Merkmale		
		Anzahl	im Mittel		im Maximum
	Auflösung	Hypothesen	Gesamt	Stadt	Gesamt
FIR	324×256	180 408	$0.89 \cdot 10^6$	$1.12 \cdot 10^6$	$3.03 \cdot 10^6$
NIR	640×480	676 088	$7.26 \cdot 10^6$	$10.30 \cdot 10^6$	$15.21 \cdot 10^6$
Fusion	s.o.	1 395 330	$6.79 \cdot 10^6$	$9.22 \cdot 10^6$	$16.51 \cdot 10^6$

Tabelle 7.5.: Vergleich der Aufwände durch den einfachen Hypothesengenerator. Auf Basis der Teststichprobe wurden Anzahl der Hypothesen pro Bild und Anzahl der berechneten Merkmale (im Mittel und im Maximum) pro Bild ermittelt. Die Anzahl der Merkmale ist dabei abhängig vom jeweiligen Bildinhalt. Generell ist es auf Bildern mit stark strukturierten Hintegründen (Stadt) aufwändiger Fußgänger zu detektieren.

leicht im Vorteil. Bei einem der Bildpaare aus dem Evaluationsdatensatz benötigte der NIR-Klassifikator $15.21 \cdot 10^6$ Merkmalsberechnungen, der Fusionsdetektor jedoch $16.51 \cdot 10^6$ (Abbildung 7.10, rechtes Bildpaar).

Bezogen auf die Anzahl der Merkmalsberechnungen ist im Vergleich von Fusionsdetektor und FIR-Detektor dagegen der FIR-Detektor dem Fusionsdetektor deutlich überlegen. Er benötigt pro Bild im Mittel nur $0.89 \cdot 10^6$, im schlechtesten Fall $3.03 \cdot 10^6$ Merkmale pro Bild. Tabelle 7.5 fasst den Vergleich einfacher Hypothesengenerator vs. Multi-Sensor Hypothesengenerator nochmals zusammen.

Hypothesenbaum

Um den Aufwand vor allem im Fusionsfall deutlich zu reduzieren wurde in dieser Arbeit der Hypothesenbaum entwickelt (Kapitel 4.3), der den Suchraum in Form einer grob-zu-fein Strategie abtastet. Dies wird mit Hilfe einer Baumstruktur realisiert, deren Ebenen jeweils einer Hypothesenmenge mit unterschiedlich feiner Rasterung entsprechen, und dessen Kanten die Nachbarschaftbeziehungen der grob-zu-fein Suche abbilden. In diesem Abschnitt werden Bäume mit $L = 4$ Ebenen untersucht, wobei die feinste Ebene $\mathcal{H}^{(4)}$ dieselbe Rasterdichte aufweist, wie der einfache Hypothesengenerator:

$$\begin{aligned}
\mathcal{H}^{(1)} &= \mathcal{H}\,(0.3, 0.3, 0.3)\,, \\
\mathcal{H}^{(2)} &= \mathcal{H}\,(0.1, 0.2, 0.3)\,, \\
\mathcal{H}^{(3)} &= \mathcal{H}\,(0.1, 0.1, 0.1)\,, \\
\mathcal{H}^{(4)} &= \mathcal{H}\,(0.03, 0.05, 0.08)\,.
\end{aligned}$$

Für den Fusionsdetektor ist jeweils der FIR Sensor der Primärsensor, die Rasterdichten sind auf jeder Ebene identisch. Für die Nachbarschaftsbeziehung (Gleichung (4.5), Seite 100, Abschnitt 4.3) wird jeweils $\delta = 0.75$ angenommen. Alle anderen Parameter bleiben gleich (Tabelle 7.4). Aufgrund der unterschiedlichen Auflösungen der Kamerabilder

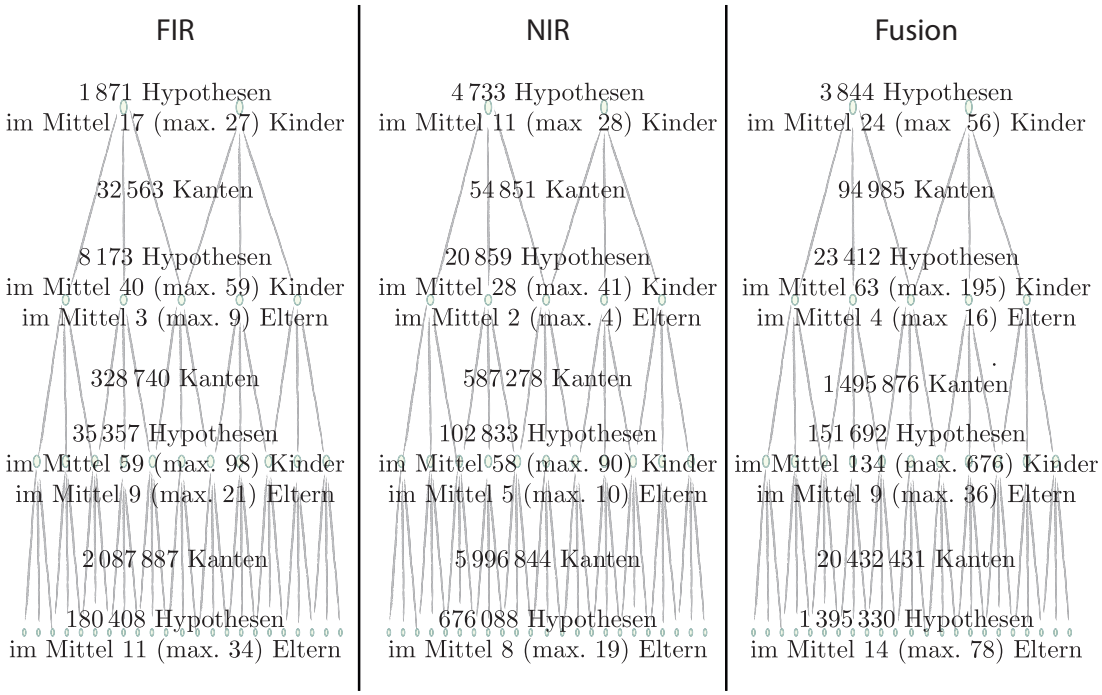


Abbildung 7.11.: Struktur der evaluierten Hypothesenbäume. Da die Raster der vier Baumebenen zueinander verschoben sind, ist die Anzahl der Kindknoten innerhalb einer Ebene nicht für jeden Knoten gleich. Dei verwendeten Ebenen sind $\mathcal{H}^{(1)} = \mathcal{H}(0.3, 0.3, 0.3), \mathcal{H}^{(2)} = \mathcal{H}(0.1, 0.2, 0.3), \mathcal{H}^{(3)} = \mathcal{H}(0.1, 0.1, 0.1)$ und $\mathcal{H}^{(4)} = \mathcal{H}(0.03, 0.05, 0.08)$.

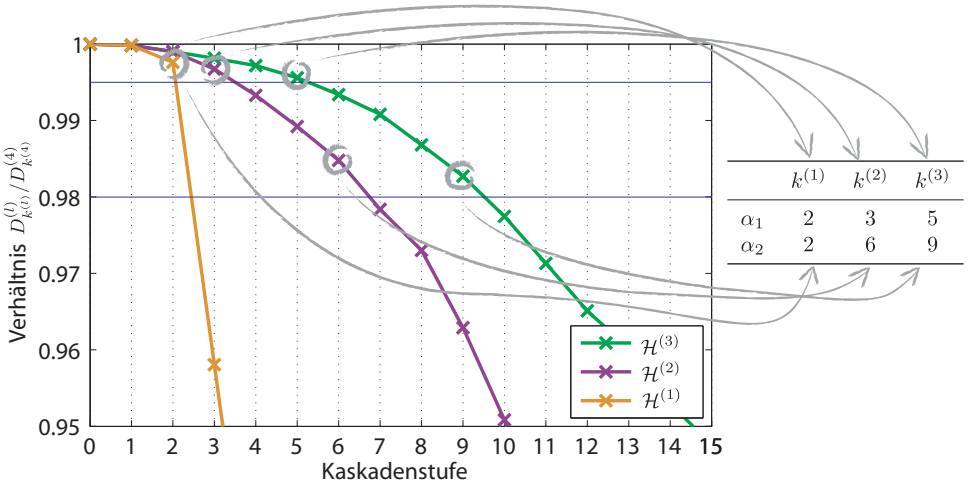


Abbildung 7.12.: Bestimmung der Schwellen für den FIR Hypothesenbaum. Das Verhältnis $D^{(l)}_{k^{(l)}} \geq \alpha \cdot D^{(L)}_{k^{(l)}}$ ist hier gegen die Kaskadenstufen aufgetragen. So können direkt die Schwellen für $\alpha_1 = 0.995$, obere Linie, und $\alpha_2 = 0.98$, untere Linie, abgelesen werden (siehe auch Gleichung (7.1)).

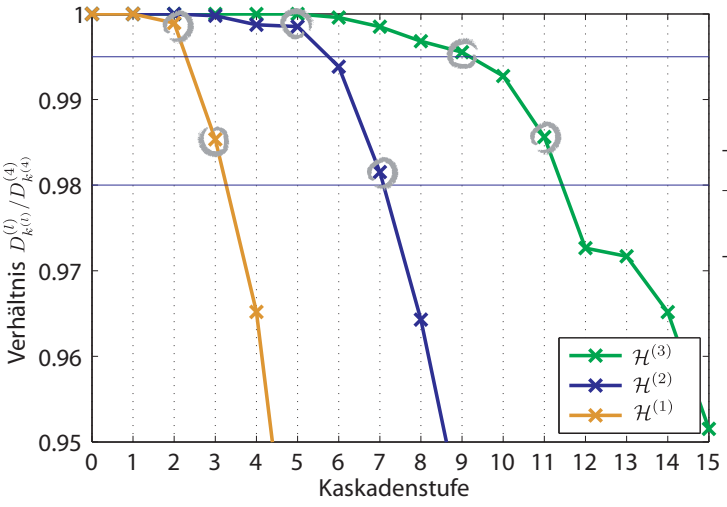
sind alle drei Bäume (FIR, NIR und Fusion) von unterschiedlicher Gestalt. Abbildung 7.11 gibt einen Überblick über die Struktur der drei Bäume.

Zur Bestimmung der Schwellen $k^{(1)}$, $k^{(2)}$ und $k^{(3)}$ wird jeder der Klassifikatoren (FIR, NIR und Fusion) mit den Hypothesengeneratoren $\mathcal{H}^{(1)}$, $\mathcal{H}^{(2)}$, $\mathcal{H}^{(3)}$ und $\mathcal{H}^{(4)}$ auf einem vom Lern- und Testdatensatz unabhängigen dritten Validierungsdatensatz (mit 9 730 Bildpaaren) evaluiert und bereits ab der ersten Kaskadenstufe die Detektionsrate protokolliert und ausgewertet. Die Schwellen werden dann jeweils so gewählt, dass

$$D^{(l)}_{k^{(l)}} \geq \alpha \cdot D^{(L)}_{k^{(l)}}. \tag{7.1}$$

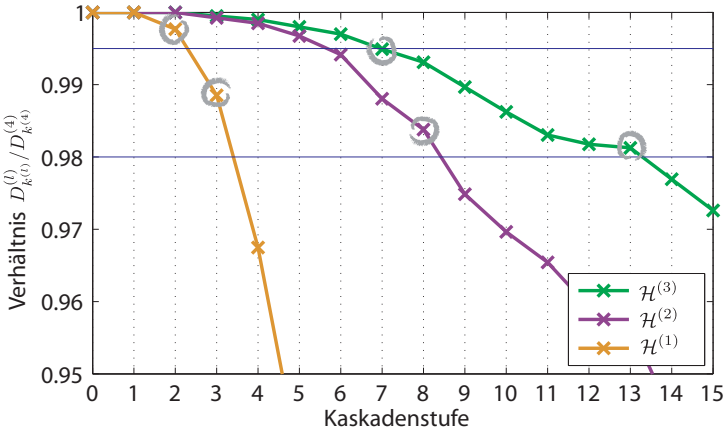
$D_{k^{(l)}}$ ist dabei die Detektionsrate im Raster $\mathcal{H}^{(l)}$ bei Kaskadenstufe $k^{(l)}$. Für α werden zwei unterschiedliche Varianten, $\alpha_1 = 0.995$ und $\alpha_2 = 0.98$, untersucht. Beim FIR Detektor ergeben sich so für $\alpha = \alpha_1$, $k^{(1)} = 2$, $k^{(2)} = 3$ und $k^{(2)} = 5$, und für $\alpha = \alpha_2$, $k^{(1)} = 2$, $k^{(2)} = 6$ und $k^{(2)} = 9$ (siehe Abbildung 7.12). Für den NIR Detektor und den Fusionsdetektor sind die entsprechenden Schwellen jeweils in Abbildung 7.13 und Abbildung 7.14 angegeben. Auffällig ist dabei vor allem, dass die Schwellen für den Fusionsdetektor in Bezug auf die Gesamtzahl von nur 28 Stufen im Vergleich zu den Einzeldetektoren deutlich größer sind.

Bevor näher auf die Anzahl der berechneten Hypothesen bzw. Merkmale pro Bildpaar eingegangen werden kann, muss zunächst die Detektionsleistung der unterschiedlichen Detektoren analysiert und der der einfachen Hypothesengeneratoren gegenübergestellt werden. Abbildung 7.15 zeigt dazu den Vergleich der ROC-Kurven für den FIR Fall. Interessanterweise ist für $\alpha_1 = 0.995$ die Detektionsrate besser als beim Detektor mit dem einfachen Hypothesengenerator. Auch die Falschalarmrate ist leicht erhöht. Für $\alpha_2 = 0.98$ ist die Detektionsleistung insgesamt schlechter. Ersteres mag überraschend erscheinen, doch obwohl die ersten Ebenen des Baumes ein sehr grobes Raster aufweisen,



	$k^{(1)}$	$k^{(2)}$	$k^{(3)}$
α_1	2	5	9
α_2	3	7	11

Abbildung 7.13.: Bestimmung der Schwellen für den NIR Hypothesenbaum. Das Verhältnis $D^{(l)}_{k^{(l)}} \geq \alpha \cdot D^{(L)}_{k^{(l)}}$ ist hier gegen die Kaskadenstufen aufgetragen. So können direkt die Schwellen für $\alpha_1 = 0.995$, obere Linie, und $\alpha_2 = 0.98$, untere Linie, abgelesen werden (siehe auch Gleichung (7.1) und Abbildung 7.12).



	$k^{(1)}$	$k^{(2)}$	$k^{(3)}$
α_1	2	5	7
α_2	3	8	13

Abbildung 7.14.: Bestimmung der Schwellen für den Fusion Hypothesenbaum. Das Verhältnis $D^{(l)}_{k^{(l)}} \geq \alpha \cdot D^{(L)}_{k^{(l)}}$ ist hier gegen die Kaskadenstufen aufgetragen. So können direkt die Schwellen für $\alpha_1 = 0.995$, obere Linie, und $\alpha_2 = 0.98$, untere Linie, abgelesen werden (siehe auch Gleichung (7.1) und Abbildung 7.12).

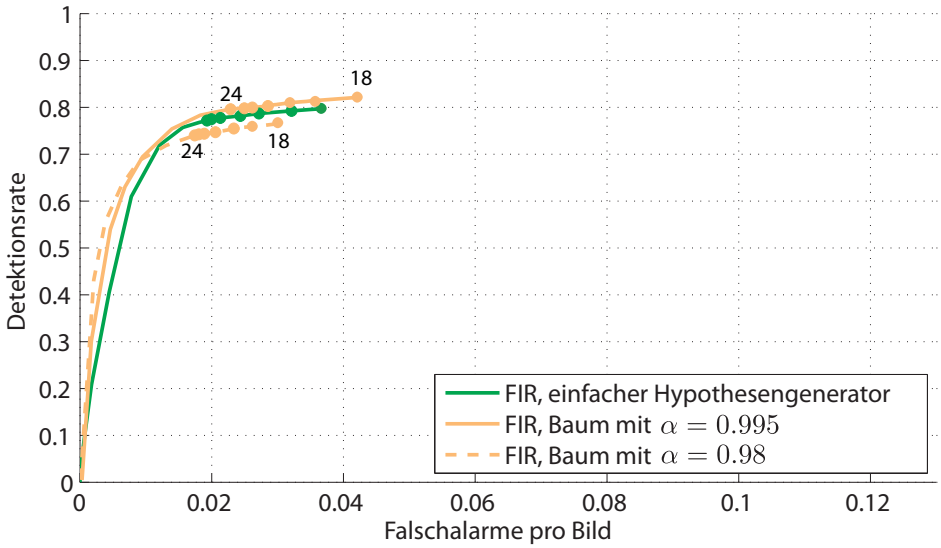


Abbildung 7.15.: Vergleich der ROC-Kurven bei der Detektion mit FIR Hypothesenbäumen.

ist das Raster - insgesamt betrachtet - feiner als in $\mathcal{H}^{(4)}$. Da die Hypothesen der unterschiedlichen Ebenen teilweise „auf Lücke“ zueinander angeordnet sind, werden insgesamt Hypothesen an mehreren Positionen im Suchraum geprüft. Dies wurde durch die Wahl der Rasterschrittweiten in Bezug auf $\mathcal{H}^{(1)}$ und $\mathcal{H}^{(2)}$ bzw. $\mathcal{H}^{(3)}$ und $\mathcal{H}^{(4)}$ gezielt forciert.

Abbildung 7.16 zeigt den Vergleich der ROC-Kurven für den NIR-Fall. Auch in diesem Fall ist die Detektionsrate und Anzahl der Falschalarme für $\alpha_1 = 0.995$ leicht erhöht. Für $\alpha_2 = 0.98$ bricht die Detektionsrate deutlich ein.

Für den Fusionsdetektor (Abbildung 7.17) ist der Vorteil der theoretisch feineren Abtastung im Suchraum nicht mehr sichtbar. Dennoch erreicht der Detektor mit Hypothesenbaum und $\alpha_1 = 0.995$ eine beachtliche Detektionsrate von 91% bei 0.025 Falschalarmen pro Bild.

Die teilweise sehr unterschiedlichen Verläufe der ROC-Kurven zeigen zweierlei. Erstens ist zumindest für $\alpha_1 = 0.995$ die Performanz der Detektoren mit Hypothesenbaum nicht signifikant schlechter als mit einfachem Hypothesengenerator. Zum Zweiten wird einmal mehr deutlich, dass die ROC-Kurve eines Fußgängerdetektors nicht nur vom Klassifikator selbst, sondern auch von der gewählten Suchstrategie abhängig ist. Diese hat zudem einen erheblichen Einfluss auf den benötigten Aufwand.

Für den Vergleich einfacher Hypothesengenerator vs. Hypothesenbaum gilt dies in besonderem Maße. Die Anzahl der im Mittel pro Bild bzw. Bildpaar berechneten Hypothesen ist um eine Größenordnung kleiner, als mit den entsprechenden einfachen Hypothesengeneratoren - und das bei vergleichbarer Performanz. So sind für $\alpha_1 = 0.995$

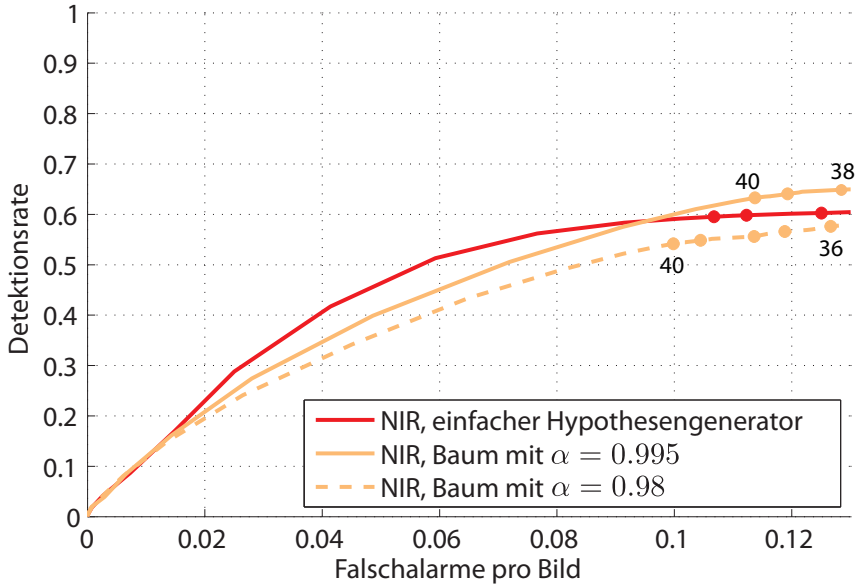


Abbildung 7.16.: Vergleich der ROC-Kurven bei der Detektion mit NIR Hypothesenbäumen.

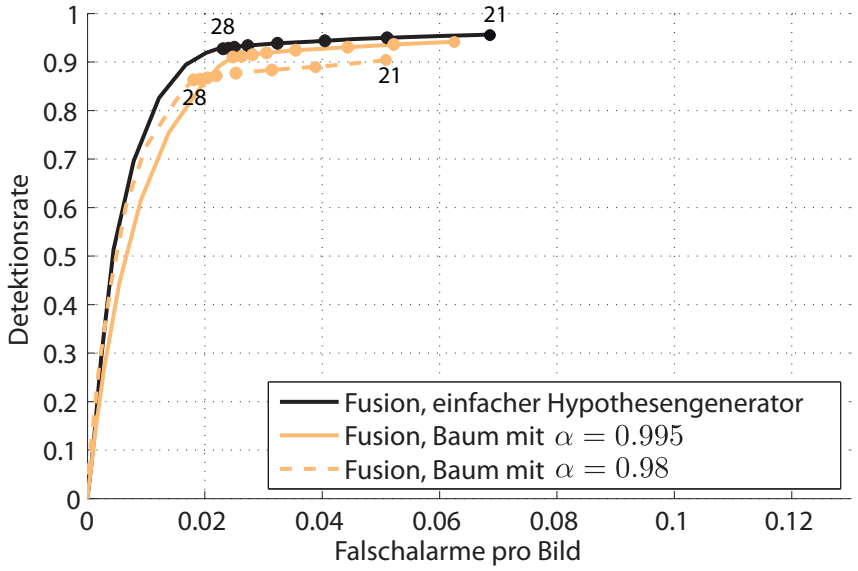


Abbildung 7.17.: Vergleich der ROC-Kurven bei der Detektion mit Multi-Sensor Hypothesenbäumen.

	einfacher	Hypothesenbaum			
		$\alpha_1 = 0.995$		$\alpha_2 = 0.98$	
	Generator	im Mittel	maximal	im Mittel	maximal
FIR	180 408	5 390(3%)	23 008(13%)	2 536(1.5%)	8 124(5%)
NIR	676 088	26 884(4%)	84 103(12%)	14 008(2%)	44 748(7%)
Fusion	1 395 330	14 408(1%)	99 539(7%)	5 589(0.5%)	22 856(2%)

Tabelle 7.6.: Anzahl berechneter Hypothesen mit einfachem Hypothesengenerator bzw. mit Hypothesenbaum. Auf Basis des Testdatensatzes wurden die mittlere und maximale Anzahl der Hypothesen pro Bild bzw. Bildpaar bei Verwendung eines Hypothesenbaumes ausgewertet (für $\alpha_1 = 0.995$ und $\alpha_2 = 0.98$) und der Hypothesen im einfachen Hypothesengenerator gegenüber gestellt.

	einfacher	Hypothesenbaum	
	Generator	$\alpha_1 = 0.995$	$\alpha_2 = 0.98$
FIR	$0.89 \cdot 10^6$	$0.12 \cdot 10^6(13\%)$	$0.05 \cdot 10^6(6\%)$
NIR	$7.26 \cdot 10^6$	$1.34 \cdot 10^6(18\%)$	$0.73 \cdot 10^6(10\%)$
Fusion	$6.79 \cdot 10^6$	$0.46 \cdot 10^6(7\%)$	$0.13 \cdot 10^6(2\%)$

Tabelle 7.7.: Anzahl berechneter Merkmale mit einfachem Hypothesengenerator bzw. mit Hypothesenbaum. Auf Basis des Testdatensatzes wurde die mittlere Anzahl der berechneten Merkmale pro Bild bzw. Bildpaar bei Verwendung eines Hypothesenbaumes ausgewertet und den Detektoren mit einfachem Hypothesengenerator gegenüber gestellt.

für den Fusionsdetektor statt 1 395 330 im Mittel nur noch 14 408 Hypothesen pro Bildpaar nötig. Dabei ist nun die Anzahl der Hypothesen jedoch abhängig vom Bildinhalt. Je schwieriger die Szenerie, desto mehr Hypothesen werden benötigt. Theoretisch könnten dabei sogar alle Knoten im Hypothesenbaum durchlaufen werden - im Fusionsfall sind das 1 574 278 Knoten[3]. In der Praxis waren jedoch nie mehr als 99 539 Hypothesen pro Bildpaar nötig.

Tabelle 7.6 fasst die anhand des Testdatensatz ausgewerteten mittleren und maximalen Hypothesenanzahlen pro Bild bzw. Bildpaar zusammen. Auffällig ist, dass der Fusionsdetektor offensichtlich am deutlichsten vom Hypothesenbaum profitiert. Er kommt nun im Mittel sogar mit weniger Hypothesen aus, als der NIR-Detektor. In allen Fällen (FIR, NIR und Fusion) kann der Aufwand mit $\alpha_2 = 0.98$ noch weiter reduziert werden (hier jedoch einhergehend mit Einbußen in der Detektionsleistung, siehe oben).

In Tabelle 7.7 werden anstatt der Anzahl der Hypothesen, die Anzahl der berechneten Merkmale verglichen. Mit einem Hypothesenbaum werden erwartungsgemäß pro Hypothese (gemittelt über ein Bild) etwas mehr Merkmale berechnet als mit dem einfachen

[3]Einige davon bilden jedoch ein und dieselbe Hypothese ab, da nicht alle Hypothesen einer Ebene „auf Lücke“ zur nächsten Hypothese angeordnet sind.

Hypothesengenerator. Obwohl im Mittel die Anzahl der Hypothesen im Fusionsfall um 99% sinkt, sinkt die Anzahl der Merkmale lediglich um 93%.

Fazit einfacher Hypothesengenerator vs. Hypothesenbaum

Zusammenfassend können auf Basis der Auswertungen folgende Aussagen getroffen werden:

- Der Fusionsdetektor muss mit dem einfachen Hypothesengenerator deutlich mehr Hypothesen (nämlich 1 395 330) pro Bild berechnen, als die beiden Einzel-Sensor Detektoren (mit 676 088 für den NIR- bzw. lediglich 180 408 für den FIR-Detektor).
- Betrachtet man die Anzahl der berechneten Merkmale pro Bild, ist der Fusionsdetektor auch mit dem einfachen Hypothesengenerator nicht viel aufwändiger, als der NIR-Detektor (Tabelle 7.5). Dabei weist der Fusionsdetektor eine deutlich höhere Detektionsrate und eine bessere Falschalarmrate auf.
- Aufgrund der geringen Bildauflösung des FIR-Sensors ist die Detektion von Fußgängern mit dem FIR-Sensor am wenigsten aufwändig. NIR-Detektor und Fusionsdetektor benötigen beide mehr als 8 mal so viele Merkmalsberechnungen pro Bild bzw. Bildpaar.
- Der Einsatz von Hypothesenbäumen anstatt einfacher Hypothesengeneratoren hat (zumindest für $\alpha_1 = 0.995$) keinen nachteiligen Einfluss auf die Detektionsleistung der drei Detektoren (FIR, NIR und Fusion).
- Mit Hypothesenbäumen reduziert sich der Aufwand für alle drei Detektoren erheblich. So werden bei der Anwendung im Testdatensatz für den Fusionsdetektor statt 1 395 330 Hypothesen pro Bild im Mittel nur noch 14 408 und maximal 99 539 Hypothesen benötigt (für $\alpha_1 = 0.995$). Der Aufwand reduziert sich damit nahezu um den Faktor 100.
- Der Fusionsdetektor profitiert am meisten vom Einsatz des Hypothesenbaumes, so dass er mit Hypothesenbaum nun merklich weniger aufwändig ist, als der NIR-Klassifikator - bei signifikant besserer Detektionsleistung des Fusionsdetektors.
- Der FIR-Detektor benötigt mit Hypothesenbaum nur noch 5 390 (für $\alpha_1 = 0.995$) bzw. 2 536 (für $\alpha_2 = 0.98$) Hypothesen.

Eine illustrative Übersicht über den Aufwand der verschiedenen Detektoren zeigt Abbildung 7.18.

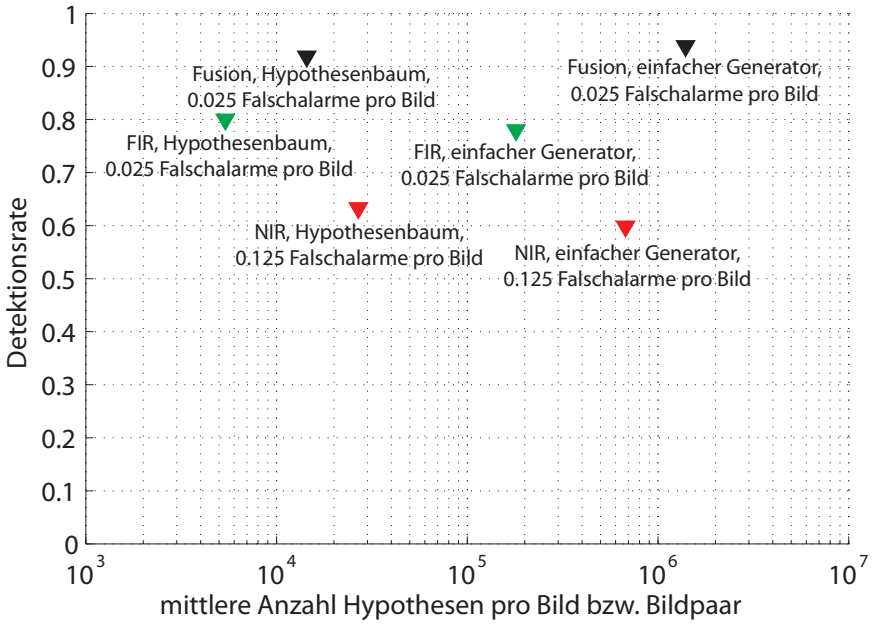


Abbildung 7.18.: Vergleich der verschiedenen Detektoren (FIR, NIR und Fusion) in Bezug auf Aufwand und Detektionsleistung. Der FIR- und Fusionsdetektor sind jeweils für die Arbeitspunkte bei 0.025 Falschalarmen pro Bild, der NIR-Detektor mit wurden mit 0.125 Falschalarmen pro Bild dargestellt.

7.4. Partikelfilter

Die Anwendung des Hypothesenbaumes zur Fußgängererkennung erlaubt die Realisierung eines Detektors, der im Mittel mit sehr wenigen Merkmalsberechnungen pro Bild auskommt. Allerdings hat er den Nachteil, dass die Anzahl der Hypothesen pro Bild von der Szenerie abhängig ist. Im ungünstigsten Fall ist der Berchnungsaufwand genauso hoch, wie bei der Detektion mit einfachen Hypothesengeneratoren. In der praktischen Umsetzung für ein Fahrerassistenzsystem müssen deshalb weitere Strategien entwickelt werden, um den Aufwand im worst-case Szenario zu begrenzen (z.B. durch Priorisierung bestimmter Bildbereiche). Darüber hinaus sind für diese worst-case Szenarien im Steuergerät Ressourcen vorzuhalten, die in den meisten Fällen aber gar nicht benötigt werden. Das Partikelfilterverfahren aus Kapitel 6 arbeitet dagegen in jedem Bild mit gleich vielen Partikeln. Die Anzahl der Hypothesen ist dadurch also per se beschränkt. Dazu muss jedoch - genauso wie bei der Detektion mit Hypothesenbäumen - überprüft werden, ob mit dieser Suchstrategie auch die gleiche Detektionsleistung erreicht werden kann. Die Evaluierung gestaltet sich in diesem Fall jedoch deutlich schwieriger, da ein Teilsystem bewertet werden muss, das sich je nach Parametrisierung in sehr unterschiedlicher Weise verhalten kann. Es ist außerdem nicht möglich, die vom Verfahren einmal bewerteten Hypothesen abzuspeichern, um dann durch Veränderung eines Arbeitspunktes (einer Schwelle) eine ROC-Kurve zu bestimmen. Vielmehr muss für jeden Arbeitspunkt das Verfahren von neuem auf dem Testdatensatz angewandt

Parameterwahl	Beschreibung (vgl. Kapitel 6)
$N_s = 300$	Anzahl Partikel pro Partikelfilterinstanz
$\mathcal{H}^{(\text{init})} = \mathcal{H}(0.3, 0.3, 0.3)$	Hypothesenmenge zur Initialisierung der Partikelfilterinstanzen
$\alpha_{\text{col}} = 0.03$ $\alpha_{\text{row}} = 0.05$ $\alpha_h = 0.08$	Faktoren zur skalierungsabhängigen Modellierung eines normalverteilten Rauschprozesses, mit $\sigma_\cdot = \alpha_\cdot \cdot h$, vgl. (6.2), Seite 124
$M = 10$	Anzahl Partikelfilterinstanzen
$M^* = 4$	Anzahl Partikelfilterinstanzen, die zu jedem Zeitschritt reinitialisiert werden
$w^* = 0.9$	Schwellwert zur Detektionsentscheidung
$\rho_{h,H} = 0.9$	Korrelationskoeffizient im normalverteilten Rauschprozess
$\text{cov}_{\text{cluster}} = 0.1$	Schwelle zur Clusterung der Hypothesenmenge zur Initialisierung der Partikelfilterinstanzen
$\text{cov}^*_{\text{max}} = 0.8$	maximal zulässige Überdeckung zur Definition von Verbotszonen

Tabelle 7.8.: Parametrisierung der Fußgängererkennung mit Partikelfilter.

werden, da die Parametrisierung immer auch Einfluss auf Verbotszonen, Reinitialisierung und Resampling der Partikelfilterinstanzen hat. In dieser Arbeit werden daher nur ausgewählt Arbeitspunkte ausgewertet.

Die Parametrisierung des Partikelfilterverfahrens hängt dabei im Wesentlichen von den in Tabelle 7.8 aufgeführten Parametern ab. Insbesondere sei an dieser Stelle auf die beiden Parameter M und M^* hingewiesen. M ist die Anzahl der Filterinstanzen, die zur Fußgängerdetektion verwendet werden sollen und beschränkt dadurch die Anzahl der Fußgänger, die in einem Bild gleichzeitig erkannt werden können. Dies stellt eine Einschränkung dar, da auch Szenen mit mehreren Fußgängern vorkommen können (z.B. in Innenstädten). Durch die Definition der Verbotszonen in dieser Arbeit (siehe Abschnitt 6.3) ist zumindest sichergestellt, dass nahe Fußgänger höher priorisiert werden. Das Verhalten muss jedoch anhand der konkreten Anforderungen in einem Fahrerassistenzsystem angepasst werden. Mit $M = 10$ ist in dieser Arbeit sichergestellt, dass theoretisch alle Fußgänger im Testdatensatz vom System erkannt werden könnten.

M^* ist die Anzahl der Trackerinstanzen, die zu jedem Zeitschritt neu im Zustandsraum initialisiert werden. Die Reinitialisierung ist insofern wichtig, als auch beim verwendeten Condensation-Algorithmus eine Degeneration der Partikelmengen („sample impoverishment“, siehe Abschnitt 5.2, Seite 109) nicht generell verhindert werden kann. Werden genügend Partikelfilterinstanzen zu jedem Zeitschritt neu initialisiert, ist sichergestellt, dass der Zustandsraum an weitgehend allen Stellen nach Fußgängern abgesucht wird. Das Vorgehen in jedem Zeitschritt die $M^* = 4$ „schlechtesten“ Instanzen (sofern sie

nicht einen Fußgänger verfolgen) zu reinitialisieren stellt sicher, dass immer genügend Partikelfilterinstanzen „auf der Suche“ nach weiteren Fußgängern im Bild sind.

FIR-solo

Im Folgenden werden zunächst anhand der Fußgängererkennung mit einem FIR Sensor unterschiedliche Arbeitspunkte des Verfahrens diskutiert und erst dann auch für den Fusionsfall evaluiert. Der NIR Detektor wird in diesem Fall nicht betrachtet, da in der Praxis eine Anwendung des Partikelfilterverfahrens aufgrund der schlechten Erkennungsleistung des Klassifikators auf den dafür zu schwierigen Testdatensatz (und damit Vergleichsdatensatz) nur unzureichend funktionierte.

Die sinnvollste[4] Parameterwahl für den FIR Detektor mit Partikelfilter ist in Tabelle 7.8 bereits aufgelistet. Damit werden 82.1% aller Fußgänger des gesamten Testdatensatzes erkannt, mit 0.028 Falschalarmen pro Bild. Bei den einfacheren Landstraßenszenen beträgt die Detektionsrate 85.5% mit 0.006 Falschalarmen pro Bild, in Innenstädten ist die Detektionsrate 73.0% mit 0.084 Falschalarmen pro Bild. Damit ist die Erkennungsrate um bis zu fünf Prozentpunkte höher als bei der Detektion mit einfachem Hypothesengenerator. Allerdings kann das Partikelfilterverfahren nicht die niedrigen Falschalarmraten erreichen. Insbesondere in Innenstädten treten mit Partikelfilter nahezu doppelt so viele Falschalarme auf. Den direkten Vergleich Partikelfilter gegenüber dem einfachen Hypothesengenerator stellt Abbildung 7.19 (Punkte mit $\mathcal{H}^{(\text{init})}\,(0.6, 0.6, 0.6)$) dar.

Aus systematischer Sicht wäre im übrigen auch eine niedrigere Detektionsrate begründbar, da die Filter beim erstmaligen Erscheinen eines Fußgängers im Bild erst nach einigen Iterationen eine zuverlässige Schätzung liefern können[5]. Durch die bereits sehr gute Initialisierung der Filterinstanzen mit Hilfe von $\mathcal{H}^{(\text{init})}$ werden allerdings nur wenige (in der Regel weniger als 4) Iterationen benötigt, um einen Fußgänger sicher zu detektieren, so dass dieser systematische Effekt stark abgemildert ist. Da der Partikelfilter nicht auf eine Quantisierung der Rasterschrittweiten der Hypothesen angewiesen ist, kann dieser den Suchraum in den „interessanten“ Regionen pixelgenau und in allen Skalierungen absuchen und so sogar eine höhere Detektionsrate erreichen. Außerdem stellt sich mit dem Partikelfilter ein Filtereffekt ein, der sporadisch auftretende Lücken bei den Detektionen verhindern. Aufgrund des im Prinzip feinst möglichen Suchrasters treten allerdings auch mehr Falschalarme auf.

Mit $N_s = 750$ Partikel in $M = 10$ Filterinstanzen, sowie $\mathcal{H}^{(\text{init})}\,(0.6, 0.6, 0.6)$ mit 323 Hypothesen, werden pro Bild 7 823 Hypothesen ausgewertet. Das sind mehr Hypothesen als im Mittel beim Hypothesenbaum mit $\alpha_1 = 0.995$ benötigt werden (im Mittel 5 390 Hypothesen), aber deutlich weniger als in dessen ungünstigstem Fall (23 008 Hypothesen).

Die Initialisierung der Partikelfilterinstanzen mit Hilfe der Auswertung der Hypothesenmenge $\mathcal{H}^{(\text{init})}$ ist dabei essentiell. Lässt man sie weg und initialisiert die jeweiligen

[4] „sinnvoll“ im Sinne von „mit dem besten Kosten-Nutzen-Verhältnis“

[5] Die grob-zu-fein Suche ist sozusagen über die Zeit hinweg organisiert.

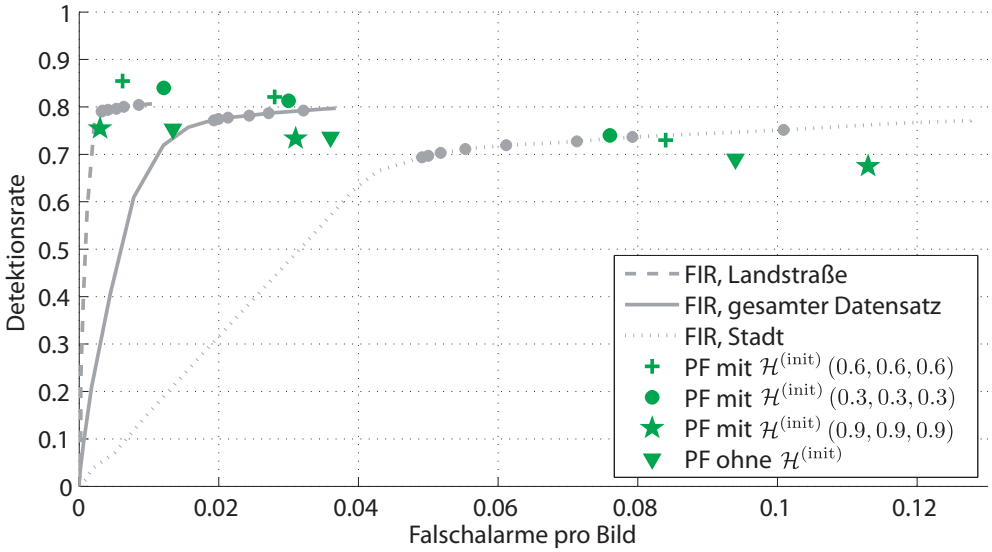


Abbildung 7.19.: Vergleich Partikelfilterverfahren und einfacher Hypothesengenerator bei unterschiedlicher Parametrisierung von $\mathcal{H}^{(\text{init})}$. Generell erreicht das Verfahren mit Partikelfilter höhere Detektionsraten, allerdings einhergehend mit einer größeren Zahl an Falschalarmen. Der Einfluss der Initialisierung durch $\mathcal{H}^{(\text{init})}$ ist deutlich sichtbar. Ohne diese Vorstufe liegt die Detektionsrate auf dem gesamten Datensatz bis zu zehn Prozentpunkte niedriger.

Filter zufällig auf Basis einer Gleichverteilung im Zustandsraum, ist lediglich eine Detektionsrate von nur 73.6% mit 0.036 Falschalarmen pro Bild möglich (Abbildung 7.19). Dabei müssen die Hypothesen in $\mathcal{H}^{(\text{init})}$ auch nicht sehr fein quantisiert sein. Durch den Clusterschritt zur Generierung der a priori Beispiele (vgl. Kapitel 6.2) werden die Hypothesen so oder so nochmals zusammengefasst. Erst ab etwa $\mathcal{H}^{(\text{init})}\,(0.9, 0.9, 0.9)$ verschwindet der positive Effekt schlagartig. Dann kann eine höhere Detektionsrate nur durch Erhöhung der Partikelanzahl realisiert werden. Ohne $\mathcal{H}^{(\text{init})}$ kann eine Detektionsrate von etwa 80% nur erreicht werden, wenn etwa 3 000 Partikel pro Instanz verwendet werden (Abbildung 7.20). Erhöht man die Zahl der Partikel weiter, wird das System instabil, da sich die unterschiedlichen Filterinstanzen dann gegenseitig behindern. Mit Initialisierung der Partikelfilterinstanzen auf Basis der Hypothesenmenge $\mathcal{H}^{(\text{init})}\,(0.6, 0.6, 0.6)$ sind dagegen $M = 750$ Partikel pro Instanz ausreichend (Abbildung 7.21).

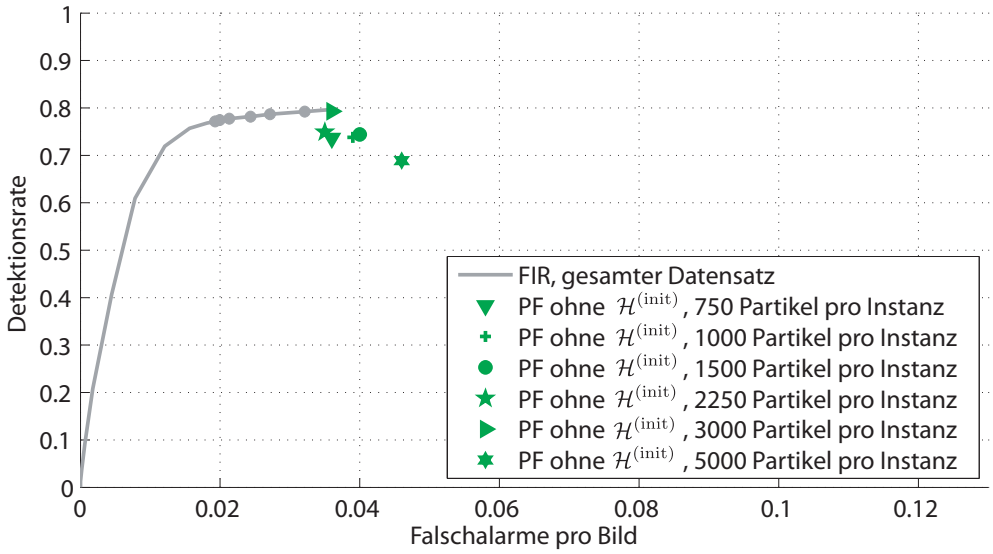


Abbildung 7.20.: Anzahl Partikel ohne a priori Initialisierung durch $\mathcal{H}^{(\text{init})}$. Ohne a priori Initialisierung der Partikelmengen werden 3 000 Partikel benötigt, um eine Detektionsrate von fast 80% zu erreichen (zum Vergleich: mit Initialisierung durch $\mathcal{H}^{(\text{init})}$ können mit 750 Partikel 82.1% der Fußgänger erkannt werden, siehe Abildung 7.19)

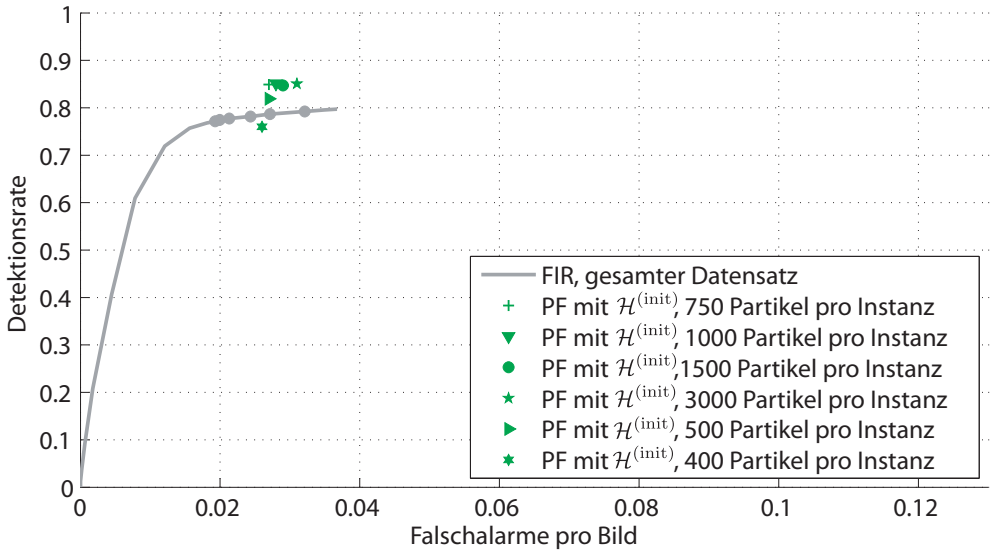


Abbildung 7.21.: Anzahl Partikel mit a priori Initialisierung durch $\mathcal{H}^{(\text{init})}$. 750 Partikel pro Filterinstanz sind nötig, um 82.1% der Fußgänger detektieren zu können.

Fusion

Der Zustand $\boldsymbol{x}_i = (\mathrm{col}_i, \mathrm{row}_i, h_i, H_i)^{\mathrm{T}}$ (Gleichung (6.1), Seite 123, Abschnitt 6.1) im Partikelfilterverfahren zur Fußgängererkennung modelliert nur das Objektfenster in genau einem Sensordatenstrom. Im Multi-Sensor Fall ist durch den Zustand damit lediglich das Objektfenster des Primärsensors definiert. Die zugehörigen Objektfenster des Sekundärsensors entstehen dann anhand derselben Mechanismen, wie sie beim einfachen Hypothesengenerator zum Einsatz kommen. Dies bedeutet jedoch, dass pro Partikel nicht mehr nur eine Hypothese ausgewertet werden muss, sondern mehrere, da die Zuordnungen nicht eindeutig sind (vgl. Abschnitt 6.2). Da die reale Größe eines Fußgängers im Zustand mit enthalten ist, ist der Korrespondenzsuchbereich zu einem Objektfenster kleiner, als beim einfachen Hypothesengenerator, da in Algorithmus 4.4 (Seite 95, Abschnitt 4.2) $H_{\min} = H_{\max} = H_i$ gilt. Die Anzahl der korrespondierenden Objektfenster hängt dann hauptsächlich vom Suchraster $\mathcal{H}'$ der Einzelsensorhypothesenmenge des Sekundärsensors ab. Dessen Parametrisierung muss also zusätzlich berücksichtigt werden, wobei entsprechend des einfachen Hypothesengenerators die Höhe der Objektfenster nicht quantisiert ist, sondern sich direkt aus der Epipolargeometrie bestimmt (vgl. Diskussion in Abschnitt 4.2). Wählt man dazu $\mathcal{H}\,(0.1, 0.2, \cdot)$ und belässt die restlichen Parameter wie in Tabelle 7.8, werden auf dem Testdatensatz 89.1% der Fußgänger bei 0.025 Falschalarmen pro Bild erkannt. Dies ist zwar besser, als im FIR-Solo Fall mit Partikelfilter (mit 82.1% Detektionsrate bei 0.028 Falschalarmen), jedoch hinsichtlich der Detektionsrate um fast vier Prozentpunkte schlechter als mit einfachem Hypothesengenerator (mit 93% Detektionsrate). Die Aufteilung der Evaluation in Landstraßen- und Stadtsequenzen (Abbildung 7.22) zeigt, dass die Detektionsrate vor allem in der Stadt im Vergleich deutlich niedriger ist. Allerdings kann bereits durch eine Erhöhung der Partikelzahl um jeweils ein Drittel auf 1 000 Partikel pro Trackerinstanz eine deutlich bessere Erkennungsrate mit 92.3% bei 0.036 Falschalarmen erreicht werden. Außerdem zeigt Abbildung 7.22, das die Erkennungsleistung auf Landstraßen sowohl hinsichtlich Detektionsrate, als auch hinsichtlich der Anzahl der Falschalarme vergleichbar ist.

Der Einfluss der Quantisierung zur Bestimmung der Korrespondenzhypothesen ist in Abbildung 7.23 anhand einer Auswahl an Arbeitspunkten dargestellt. Mit einem feineren Raster kann die Detektionsrate zwar leicht verbessert werden, doch steigen damit auch die Kosten, da mehr Korrespondenzhypothesen betrachtet werden müssen. Mit $\mathcal{H}\,(0.03, 0.05, \cdot)$ werden im Mittel 9 Hypothesen mit $\mathcal{H}\,(0.1, 0.2, \cdot)$ im Mittel lediglich 4 Hypothesen pro Partikel erzeugt[6]. Die Wahl von $\mathcal{H}\,(0.1, 0.2, \cdot)$ stellt dabei einen guten Kompromiss dar und hat sich auch in der Praxis mehrfach bestätigt. Bei $N_s = 1\,000$ Partikeln in $M = 10$ Trackerinstanzen und $\mathcal{H}^{\mathrm{init}}\,(0.6, 0.6, 0.6)$ mit 410 Hypothesen werden im Mittel pro Bild 40 410 Hypothesen berechnet. Das sind weniger Hypothesen als beim Hypothesenbaum mit $\alpha_1 = 0.995$ auf dem Testdatensatz maximal berechnet wurden (99 539 Hypothesen), jedoch um ein Vielfaches mehr als dieser im Mittel (14 408 Hypothesen) berechnete.

[6] Die Anzahl der Korrespondenzen ist im Mittel tatsächlich so gering, da durch die Höhe H_i im Zustand der Suchbereich im Sekundärsensor deutlich eingeschränkt werden kann.

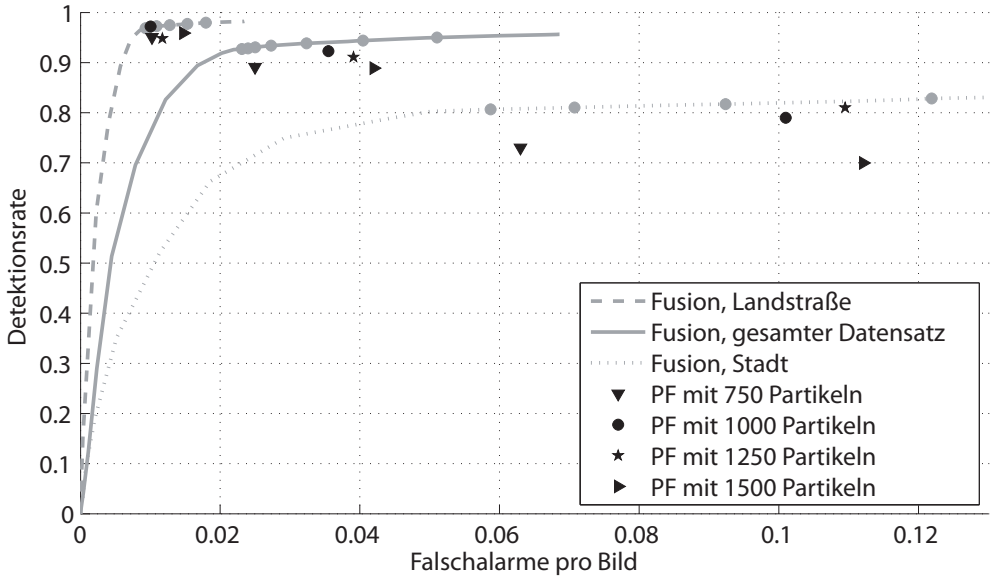


Abbildung 7.22.: Erkennungsleistung des Partikelfilterverfahrens bei unterschiedlicher Partikelzahl. $N_s = 750$ Partikel pro Filterinstanz reichen nicht ganz aus, um diesselbe Detektionsrate wie mit einfachen Hypothesengeneratoren zu erreichen. Bereits mit $N_s = 1\,000$ Partikeln pro Filterinstanz ist die Erkennungsrate vergleichbar, allerdings mit einer höheren Zahl an Falschalarmen in der Stadt.

Fazit Partikelfilterverfahren

Zusammenfassend können auf Basis der Auswertungen folgende Aussagen getroffen werden:

- Mit geeigneter Parametrisierung kann bei der Detektion von Fußgängern mit Hilfe von Partikelfilterverfahren die gleiche (beim FIR-solo Fall sogar eine leicht höhere) Detektionsrate erreicht werden wie mit dem einfachen Hypothesengenerator.
- Insbesondere in Stadtszenarien treten mit Partikelfilter etwas mehr Falschalarme auf.
- Durch die Verwendung einer a priori Initialisierung auf Basis der Auswertung einer sehr grob quantisierten Hypothesenmenge kann die Anzahl der Partikel auf nur $750 - 1\,000$ Partikel pro Trackerinstanz reduziert werden.
- Im FIR-solo Fall werden damit in jedem Bild nur 7 823 Hypothesen ausgewertet (zum Vergleich Hypothesenbaum mit $\alpha_1 = 0.995$: im Mittel 5 390, maximal 23 008). Die Anzahl der berechneten Hypothesen ist in jedem Bild gleich und unabhängig von der Szenerie.
- Im Fusionsfall werden in jedem Bildpaar im Mittel 40 410 ausgewertet (zum Vergleich Hypothesenbaum mit $\alpha_1 = 0.995$: im Mittel 14 408, maximal 99 539).
- Der Partikelfilter stellt vor allem im FIR-solo Fall sicher, dass die Detektion von Fußgängern in jedem Bild mit dem gleichen Aufwand betrieben werden kann. Der Aufwand ist nicht abhängig von der Szenerie.

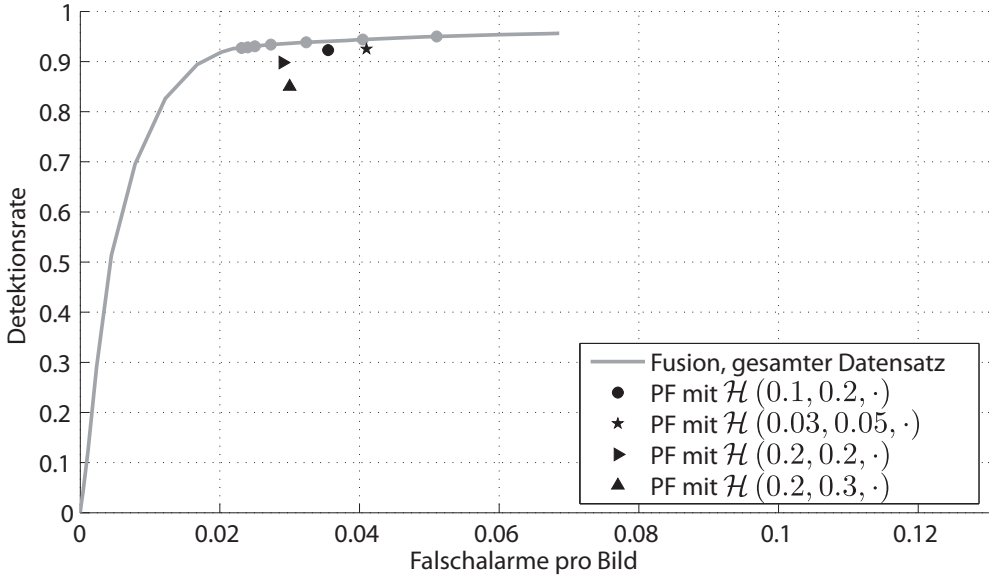


Abbildung 7.23.: Einfluss der Quantisierung zur Bestimmung der Korrespondenzhypothesen auf das Ergebnis des Partikelfilterverfahrens. Die Erkennungsleistung des Verfahrens hängt natürlich von der Quantisierung zur Bestimmung der Korrespondenzhypothesen ab. Je feiner das Raster, desto größer der Aufwand, da pro Partikel mehr (Korrespondenz-)Hypothesen ausgewertet werden müssen. Da die Skalierung der Objektfenster im Sekundärsensor auf Basis der Epipolargeometrie direkt bestimmt wird und konstant ist, ist die Quantisierung Q_h nicht relevant. Ein guter Kompromiss ist die Wahl von $\mathcal{H}(0.1, 0.2, \cdot)$, die im Mittel zu 4 korrespondierenden Objektfensterpaaren pro Partikel führt (zum Vergleich: bei $\mathcal{H}(0.03, 0.05, \cdot)$ werden im Mittel 9 Hypothesen gebildet).

KAPITEL 8

Zusammenfassung und Resumé

Gegenstand dieser Arbeit war die Realisierung eines Sensorfusionssystems zur echtzeitfähigen Fußgängererkennung als eine mögliche Basis für Nachtsichtsysteme der dritten Generation. Dazu wurden die Informationen einer Nahinfrarot-Kamera (mit aktiver Beleuchtung der Szene im nahinfraroten Wellenlängenbereich) und die einer Wärmebildkamera auf Merkmalsbasis miteinander kombiniert. Es wurden Kameras verwendet, wie sie bereits seit der ersten Generation von Nachtsichtsystemen unterschiedlicher Automobilfirmen im Fahrzeug zum Einsatz kommen.

Durch die Merkmalsfusion mit Hilfe von AdaBoost entstand ein Fußgängerklassifikator, der in Evaluierungen auf großen Datensätzen signifikant bessere Ergebnisse erreicht als entsprechende Verfahren mit nur einem Sensor. Bei gleich bleibender Falschalarmrate konnte so die Erkennungsrate im Entfernungsbereich bis ca. 130 m vor dem Fahrzeug gegenüber einem FIR-Solo Klassifikator um 15 Prozentpunkte auf 93% gesteigert werden. Der Klassifikator selbst ist dabei deutlich weniger aufwändig und umfasst etwa ein Drittel weniger Merkmale als der FIR-Solo Klassifikator. Die Fusion auf Merkmalsebene bringt also in vielerlei Hinsicht deutliche Vorteile mit sich.

Bisher war eine solche Fusion im Fahrerassistenzbereich nicht umsetzbar, da durch den unterschiedlichen Einbauort und des dadurch entstehenden Parallaxeproblems der Suchraum im Vergleich zu Einzel-Sensor Systemen um ein Vielfaches größer ist. Um sich dieser Problematik zu stellen und gleichzeitig die Echtzeitfähigkeit des Systems zu gewährleisten, wurden in dieser Arbeit neuartige Suchstrategien entwickelt, die die Eigenschaften der eingesetzten Klassifikatoren gezielt ausnutzen. Ausgehend von einem modellbasierten statischen Hypothesengenerator wurde die Struktur eines Hypothesenbaumes entwickelt, der den Suchraum in einer grob-zu-fein Suche gezielt nach Fußgängern absucht. Bei gleichbleibender Detektionsleistung müssen damit im Mittel

nur noch 1% der Hypothesen der ursprünglichen, vollständigen Hypothesenmenge betrachtet werden.

Allerdings ist die Anzahl der Hypothesen von der Schwierigkeit der Szenerie abhängig und nimmt bei stark strukturierten Szenen wieder zu. Um dem entgegen zu wirken, wurde die Idee der grob-zu-fein Suche zu einer probabilistisch motivierten Suchstrategie unter Verwendung eines Partikelfilterverfahrens weiter entwickelt. Dieses Verfahren organisiert die Suche nach Fußgängern dynamisch über ganze Bildfolgen hinweg und hält so den Berechnungsaufwand pro Bildpaar konstant auf niedrigem Niveau. Dazu werden mehrere Partikelfilterinstanzen zur Organisation der Hypothesen zur Fußgängererkennung benutzt. Jedem Partikel sind Hypothesen zugeordnet, die vom Fusionsklassifikator evaluiert werden. Die Besonderheit dabei ist, dass das Ergebnis des Klassifikators explizit in Form einer Rückschlusswahrscheinlichkeit als Messung im Partikelfilter berücksichtigt wird. Bisher war nur bekannt, wie Rückschlusswahrscheinlichkeiten von AdaBoost-Klassifikatoren bestimmt werden können. In dieser Arbeit wurde diese Theorie erweitert und erstmals auf die eingesetzten AdaBoost-Kaskaden ausgebaut.

Alle in dieser Arbeit vorgestellten Neuerungen, nämlich

- merkmalsbasierte Fusion mit AdaBoost,
- Berechnung von Rückschlusswahrscheinlichkeiten von AdaBoost-Kaskaden,
- Hypothesenbäume und
- Fußgängerdetektion mit Partikelfilter

haben eine hohe praktische Relevanz in vielerlei Hinsicht. So sind Rückschlusswahrscheinlichkeiten von AdaBoost-Kaskaden nicht nur für die Verwendung in Partikelfilterverfahren von Interesse, sondern werden auch in vielen anderen probabilistischen Trackingverfahren (z.B. [Mäh10]) Einzug halten. Darüber hinaus sind solche Rückschlusswahrscheinlichkeiten auch für komplexere Klassifikationsaufgaben, wie z.B. Kontextklassifikatoren (z.B. [SLSP09, SLM10]) von Interesse.

Die entwickelten Suchstrategien sind vor allem für die praktische Realisierung eines Detektionssystems von entscheidender Bedeutung. Fusion auf Merkmalsebene ist ohne geeignete Modellierung und Optimierung der Suchstrategien derzeit nicht umsetzbar. Dabei beschränkt sich die Verwendung nicht nur auf den Einsatz zusammen mit AdaBoost-Kaskaden. Beide Verfahren lassen sich auch in Kombination mit anderen Klassifikationsverfahren einsetzen. Sowohl der Hypothesenbaum, als auch das probabilistische Verfahren mit Partikelfilter haben beide eine gleichermaßen hohe praktische Relevanz. Der Hypothesenbaum erlaubt eine teils drastische Reduzierung des Suchaufwandes, wogegen das Partikelfilterverfahren einen konstanten - wenn auch höheren - Suchaufwand zu jedem Zeitpunkt sicher stellt.

Das wichtigste Ergebnis dieser Arbeit bleibt jedoch sicherlich der Nachweis, dass es enorme Vorteile mit sich bringt, NIR- und FIR-Kameras in einem System miteinander zu kombinieren. Die eindrucksvolle Performanzsteigerung vor allem in hohen Entfernungen sprechen dabei für sich.

ANHANG A

Beweise zu Kapitel 3

Beweis zu Satz 1, Seite 62

Satz 1 ([FS97])**.** *Mit der Notation aus Algorithmus 3.1 gilt, dass der Trainingsfehler (*3.6*) nach oben beschränkt ist, mit*

$$\begin{aligned} \mathrm{err}_{\mathrm{Training}} &\leq \frac{1}{N}\sum_{i=1}^{N}\exp\left\{-y^{(i)}A\left(x^{(i)}\right)\right\} = \prod_{t=1}^{T} Z_t, \qquad \text{(A.1)} \\ A(x) &= \sum_{t=1}^{T}\alpha_t h_t\left(x^{(i)}\right) \end{aligned}$$

*und $Z_t, t = 1, \ldots, T$ die Normalisierungskonstante in (*3.4*), mit*

$$Z_t = \sum_{i=1}^{N} d_t^{(i)} \exp\left\{-\alpha_t y^{(i)} h_t\left(x^{(i)}\right)\right\}. \qquad \text{(A.2)}$$

Beweis. Für $y^{(i)} \neq H\left(x^{(i)}\right)$ ist $y^{(i)}A\left(x^{(i)}\right) \leq 0$, d.h. $-y^{(i)}A\left(x^{(i)}\right) \geq 0$ und $\exp\left\{-y^{(i)}A\left(x^{(i)}\right)\right\} \geq 1$. Damit gilt:

$$\mathrm{err}_{\mathrm{Training}} = \frac{1}{N}\left|\left\{x^{(i)}\,\middle|\,H\left(x^{(i)}\right) \neq y^{(i)}\right\}\right| \leq \frac{1}{N}\sum_{i=1}^{N}\exp\left\{-y^{(i)}A\left(x^{(i)}\right)\right\}.$$

Die Gleichung in (A.1) lässt sich rekursiv wie folgt zeigen:

$$
\begin{aligned}
d_{T+1}^{(i)} &\overset{(3.4)}{=} \frac{1}{Z_T} d_T^{(i)} \exp\left\{-\alpha_T y^{(i)} h_T\left(x^{(i)}\right)\right\} \\
&\overset{(3.4)}{=} \frac{1}{Z_T}\left(\frac{1}{Z_{T-1}} d_{T-1}^{(i)} \exp\left\{-\alpha_{T-1} y^{(i)} h_{T-1}\left(x^{(i)}\right)\right\}\right) \exp\left\{-\alpha_T y^{(i)} h_T\left(x^{(i)}\right)\right\} \\
&= \frac{1}{Z_T \cdot Z_{T-1}} d_{T-1}^{(i)} \cdot \exp\left\{-y^{(i)}\left(\alpha_{T-1} h_{T-1}\left(x^{(i)}\right) + \alpha_T h_T\left(x^{(i)}\right)\right)\right\} \\
&\overset{(3.4)}{=} \ldots = \frac{1}{\prod_{t=1}^{T} Z_t} \cdot d_1^{(i)} \cdot \exp\left\{-y^{(i)} \sum_{t=1}^{T} \alpha_t h_t\left(x^{(i)}\right)\right\}
\end{aligned}
$$

$$
\begin{aligned}
&\overset{d_1^{(i)} = \frac{1}{N}}{\Longleftrightarrow} \qquad \frac{1}{N} \exp\left\{-y^{(i)} A\left(x^{(i)}\right)\right\} = d_{T+1}^{(i)} \cdot \prod_{t=1}^{T} Z_t \\
&\Longrightarrow \quad \frac{1}{N} \sum_{i=1}^{N} \exp\left\{-y^{(i)} A\left(x^{(i)}\right)\right\} = \underbrace{\sum_{i=1}^{N} d_1^{(i)}}_{=1} \cdot \prod_{t=1}^{T} Z_t.
\end{aligned}
$$

□

Beweis zu Satz 2, Seite 62

Satz 2. *Wenn jeder der Weaklearner nur ein klein wenig besser entscheidet als der Zufall, so dass* $\gamma_t := \frac{1}{2} - \epsilon_t \geq \gamma$ *für irgendein* $\gamma > 0$*, sinkt der Trainingsfehler exponentiell in* T*:*

$$\mathrm{err}_{\mathrm{Training}} \leq \prod_{t=1}^{T} Z_t \leq \exp\left\{-2T\gamma^2\right\}.$$

Beweis.

$$\begin{aligned}
Z_t &= \sum_{i=1}^{N} d_t^{(i)} \exp\left\{-\alpha_t y^{(i)} h_t\left(x^{(i)}\right)\right\} \\
&= \sum_{i:y^{(i)}=h_t\left(x^{(i)}\right)} d_t^{(i)} \exp\left\{-\alpha_t\right\} + \sum_{i:y^{(i)}\neq h_t\left(x^{(i)}\right)} d_t^{(i)} \exp\left\{\alpha_t\right\} \\
&\overset{(3.5)}{=} (1-\epsilon_t)\exp\left\{-\alpha_t\right\} + \epsilon_t \exp\left\{\alpha_t\right\} \\
&\overset{(3.3)}{=} (1-\epsilon_t)\exp\left\{-\frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}\right\} + \epsilon_t \exp\left\{\frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}\right\} \\
&= (1-\epsilon_t)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}} \\
&= (1-\epsilon_t)\sqrt{\frac{\epsilon_t\left(1-\epsilon_t\right)}{\left(1-\epsilon_t\right)^2}} + \sqrt{\frac{\epsilon_t^2\left(1-\epsilon_t\right)}{\epsilon_t}} \\
&= \sqrt{\epsilon_t\left(1-\epsilon_t\right)} + \sqrt{\epsilon_t\left(1-\epsilon_t\right)} \\
&= 2\sqrt{\epsilon_t\left(1-\epsilon_t\right)}.
\end{aligned} \tag{A.3}$$

Aus $\gamma_t = \frac{1}{2} - \epsilon_t \geq \gamma$ für $\gamma > 0$ und $\epsilon_t > 0$ folgt unmittelbar $\gamma_t \geq \gamma$, $\gamma < \frac{1}{2}$ und $\gamma_t < \frac{1}{2}$. Mit $\ln(1-x) \leq -x$ für $x < 1$ gilt dann

$$\begin{aligned}
\mathrm{err}_{\mathrm{Training}} \leq \prod_{t=1}^{T} Z_t &= \prod_{t=1}^{T} 2\sqrt{\epsilon_t\left(1-\epsilon_t\right)} = \prod_{t=1}^{T} 2\sqrt{\left(\frac{1}{2}-\gamma_t\right)\left(\frac{1}{2}+\gamma_t\right)} \\
&= \prod_{t=1}^{T}\sqrt{4\left(\frac{1}{4}-\gamma_t^2\right)} = \prod_{t=1}^{T}\sqrt{1-4\gamma_t^2} \leq \prod_{t=1}^{T}\sqrt{1-4\gamma^2} \\
&= \prod_{t=1}^{T}\exp\left\{\frac{1}{2}\ln\left(1-4\gamma^2\right)\right\} \leq \prod_{t=1}^{T}\exp\frac{1}{2}\left(-4\gamma^2\right) \\
&= \prod_{t=1}^{T}\exp\left\{-2\gamma^2\right\} = \exp\left\{-2T\gamma^2\right\}.
\end{aligned}$$

□

Beweis zu Satz 3, Seite 63

Satz 3 (Wahl von α_t)**.** *Für* $h_t\left(x^{(i)}\right) \in \{-1,+1\}, i = 1,\ldots,N$ *minimiert*

$$\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}.$$

in jeder Runde von AdaBoost die obere Schranke aus Satz 1.

Beweis. Eine Minimierung der Normalisieruntskonstante Z_t eines jeden Weaklearners führt auf ein minimales Produkt $\prod_{t=1}^{T} T_t$ und damit auf eine minimale obere Schranke des Trainingsfehlers (Satz 1). Für $t = 1,\ldots,T$ gilt:

$$Z_t \overset{(A.3)}{=} (1-\epsilon_t)\exp\{-\alpha_t\} + \epsilon_t \exp\{\alpha_t\}.$$

Algebraische Minimierung durch Differentiation nach α_t ergibt:

$$\begin{aligned}
&\frac{\partial Z_t}{\partial \alpha_t} = -(1-\epsilon_t)\exp\{-\alpha_t\} + \epsilon_t\exp\{\alpha_t\} \overset{!}{=} 0 \\
\iff\quad & \epsilon_t\exp\{\alpha_t\} = (1-\epsilon_t)\exp\{-\alpha_t\} \\
\iff\quad & \epsilon_t\exp\{2\alpha_t\} = 1-\epsilon_t; \\
\iff\quad & \alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}.
\end{aligned}$$

□

Beweis zu Satz 4, Seite 64

Satz 4 (Lemma 1 aus [ROM01])**.** *Die Gewichtsverteilung* d_{t+1} *in Algorithmus 3.1 leitet sich aus der Normalisierung des Gradienten* $\frac{\partial J}{\partial\rho(\alpha_{1:t},x^{(i)},y^{(i)})}$ *ab, d.h.*

$$d_{t+1}^{(i)} = \frac{\partial J(\rho)}{\partial\rho\left(\alpha_{1:t},x^{(i)},y^{(i)}\right)} \Big/ \sum_{j=1}^{N}\frac{\partial J(\rho)}{\partial\rho\left(\alpha_{1:t},x^{(j)},y^{(j)}\right)}.$$

Beweis. Sei für $i = 1,\ldots,N$:

$$\pi_t^{(i)} := \prod_{r=1}^{t}\exp\left\{-y^{(i)}\alpha_r h_r\left(x^{(i)}\right)\right\}. \tag{A.4}$$

Es gilt:

$$\pi_t^{(i)} = \pi_{t-1}^{(i)}\exp\left\{-y^{(i)}\alpha_t h_t\left(x^{(i)}\right)\right\}, \text{ mit } \pi_0^{(i)} := 1. \tag{A.5}$$

Das Funktional $J(\rho)$ ist in (3.9) definiert mit

$$J(\rho) := \frac{1}{N}\sum_{j=1}^{N}\exp\left\{-\rho\left(\alpha_{1:t}, x^{(j)}, y^{(j)}\right)\right\}.$$

Der normalisierte Gradient ist dann

$$\begin{aligned}
\frac{\frac{\partial J(\rho)}{\partial\rho\left(\alpha_{1:t},x^{(i)},y^{(i)}\right)}}{\sum_{j=1}^{N}\frac{\partial J(\rho)}{\partial\rho\left(\alpha_{1:t},x^{(j)},y^{(j)}\right)}} &= \frac{-\frac{1}{N}\exp\left\{-\rho\left(\alpha_{1:t}, x^{(i)}, y^{(i)}\right)\right\}}{-\frac{1}{N}\sum_{j=1}^{N}\exp\left\{-\rho\left(\alpha_{1:t}, x^{(j)}, y^{(j)}\right)\right\}} \\
&\overset{(3.8)}{=} \frac{\exp\left\{-y^{(i)}\sum_{r=1}^{t}\alpha_r h_r\left(x^{(i)}\right)\right\}}{\sum_{j=1}^{N}\exp\left\{-y^{(j)}\sum_{r=1}^{t}\alpha_r h_r\left(x^{(j)}\right)\right\}} \\
&= \frac{\prod_{r=1}^{t}\exp\left\{-y^{(i)}\alpha_r h_r\left(x^{(i)}\right)\right\}}{\sum_{j=1}^{N}\prod_{r=1}^{t}\exp\left\{-y^{(j)}\alpha_r h_r\left(x^{(j)}\right)\right\}} \\
&\overset{(A.4)}{=} \frac{\pi_t^{(i)}}{\sum_{j=1}^{N}\pi_t^{(j)}}.
\end{aligned}$$

Zu zeigen ist also $d_{t+1}^{(i)} = \frac{\pi_t^{(i)}}{\sum_{j=1}^{N}\pi_t^{(j)}}$.

Mit $\pi_0^{(i)} = 1, i = 1, \ldots, N$ gilt

$$\frac{\pi_0^{(i)}}{\sum_{j=1}^{N}\pi_0^{(j)}} = \frac{1}{N} = d_1^{(i)}.$$

Durch Induktion, also $d_t^{(i)} = \frac{\pi_{t-1}^{(i)}}{\sum_{j=1}^{N}\pi_{t-1}^{(j)}}$, folgt

$$\begin{aligned}
\frac{\pi_t^{(i)}}{\sum_{j=1}^{N}\pi_t^{(j)}} &\overset{(A.5)}{=} \frac{\pi_{t-1}^{(i)}\exp\left\{-y^{(i)}\alpha_t h_t\left(x^{(i)}\right)\right\}}{\sum_{j=1}^{N}\pi_t^{(j)}} \\
&= \frac{d_t^{(i)}\left(\sum_{k=1}^{N}\pi_{t-1}^{(k)}\right)\exp\left\{-y^{(i)}\alpha_t h_t\left(x^{(i)}\right)\right\}}{\sum_{j=1}^{N}\pi_t^{(j)}}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{d_t^{(i)} \exp\left\{-y^{(i)}\alpha_t h_t\left(x^{(i)}\right)\right\}}{\left(\sum_{k=1}^{N} \pi_{t-1}^{(k)}\right)^{-1} \sum_{j=1}^{N} \pi_t^{(j)}} \\
&= \frac{d_t^{(i)} \exp\left\{-y^{(i)}\alpha_t h_t\left(x^{(i)}\right)\right\}}{\sum_{j=1}^{N} \frac{\pi_t^{(j)}}{\sum_{k=1}^{N} \pi_{t-1}^{(k)}}} \\
&\overset{(A.5)}{=} \frac{d_t^{(i)} \exp\left\{-y^{(i)}\alpha_t h_t\left(x^{(i)}\right)\right\}}{\sum_{j=1}^{N} \frac{\pi_{t-1}^{(j)}}{\sum_{k=1}^{N} \pi_{t-1}^{(k)}} \exp\left\{-y^{(j)}\alpha_t h_t\left(x^{(j)}\right)\right\}} \\
&= \frac{d_t^{(i)} \exp\left\{-y^{(i)}\alpha_t h_t\left(x^{(i)}\right)\right\}}{\sum_{j=1}^{N} d_t^{(j)} \exp\left\{-y^{(j)}\alpha_t h_t\left(x^{(j)}\right)\right\}} \\
&\overset{(A.2)}{=} \frac{d_t^{(i)} \exp\left\{-y^{(i)}\alpha_t h_t\left(x^{(i)}\right)\right\}}{Z_t} \\
&= d_{t+1}^{(i)}.
\end{aligned}
$$

□

Beweis zu Satz 5, Seite 65

Satz 5 (Lemma 1 in [FHT00]). $J(\rho) = J(A(x)) = \mathbb{E}\left[\exp\left\{-yA(x)\right\}\right]$ *nimmt sein Minimum an bei*

$$A(x) = \frac{1}{2} \ln \frac{p(y=1|x)}{p(y=-1|x)}.$$

Damit gilt:

$$p(y=1|x) = \frac{\exp\left\{2A(x)\right\}}{1+\exp\left\{2A(x)\right\}}$$

und

$$p(y=-1|x) = \frac{\exp\left\{-2A(x)\right\}}{1+\exp\left\{-2A(x)\right\}}.$$

Beweis.

$$J(\rho) = J(A(x)) = \mathbb{E}\left[\exp\left\{-yA(x)\right\}\right] \to \min.$$

Die Minimierungsaufgabe ist gleichbedeutend mit

$$\mathbb{E}\left[\exp\left\{-yA(x)\right\} | \, x\right] \to \min.$$

Es gilt

$$\begin{aligned}
\mathbb{E}\left[\exp\left\{-yA\left(x\right)\right\}\middle|\, x\right] &= p(y=1|x)\exp\left\{-A\left(x\right)\right\} \\
&\quad + p(y=-1|x)\exp\left\{A\left(x\right)\right\} \\
\frac{\partial\mathbb{E}\left[\exp\left\{-yA\left(x\right)\right\}\middle|\, x\right]}{\partial A\left(x\right)} &= -p(y=1|x)\exp\left\{-A\left(x\right)\right\}. \\
&\quad + p(y=-1|x)\exp\left\{A\left(x\right)\right\} \stackrel{!}{=} 0 \\
\iff \quad p(y=-1|x)\exp\left\{A\left(x\right)\right\} &= p(y=1|x)\exp\left\{-A\left(x\right)\right\} \\
\iff \quad p(y=-1|x)\exp\left\{2A\left(x\right)\right\} &= p(y=1|x) \qquad \text{(A.6)} \\
\iff \quad \exp\left\{2A\left(x\right)\right\} &= \frac{p(y=1|x)}{p(y=-1|x)} \\
\implies \quad A\left(x\right) &= \frac{1}{2}\ln\frac{p(y=1|x)}{p(y=-1|x)}.
\end{aligned}$$

Mit (A.6) und $p(y=-1|x)+p(y=1|x)=1$ folgt

$$\begin{aligned}
(1-p(y=1|x))\exp\left\{2A\left(x\right)\right\} &= p(y=1|x) \\
\implies \quad p(y=1|x) &= \frac{\exp\left\{2A\left(x\right)\right\}}{1+\exp\left\{2A\left(x\right)\right\}},
\end{aligned}$$

sowie

$$\begin{aligned}
p(y=-1|x)\exp\left\{2A\left(x\right)\right\} &= 1-p(y=-1|x) \\
\iff \quad p(y=-1|x) &= (1-p(y=-1|x))\exp\left\{-2A\left(x\right)\right\} \\
\implies \quad p(y=-1|x) &= \frac{\exp\left\{-2A\left(x\right)\right\}}{1+\exp\left\{-2A\left(x\right)\right\}}.
\end{aligned}$$

□

Beweis zu Satz 6, Seite 75

Satz 6. *Der Trainingsfehler* $\mathrm{err}_{\mathrm{Training}}$ *eines AdaBoost-Klassifikators mit adaptierter Schwelle* $\Theta \neq 0$ *ist beschränkt durch*

$$\mathrm{err}_{\mathrm{Training}} := \frac{1}{N}\left|\left\{x^{(i)}\middle|H\left(x^{(i)}\right)\neq y^{(i)}\right\}\right| \leq \frac{1}{N}\sum_i \exp\left\{-y^{(i)}\left(A\left(x^{(i)}\right)-\Theta\right)\right\}.$$

Beweis. Mit

$$H\left(x^{(i)}\right) = \begin{cases} +1 & A\left(x\right)\geq\Theta \\ -1 & \text{sonst} \end{cases}, i=1,\ldots,N$$

folgt die Ungleichung direkt, da für $y^{(i)} \neq H\left(x^{(i)}\right) \in \{-1,+1\}$ gilt:

$$\exp\Big\{\underbrace{-y^{(i)}\left(A\left(x^{(i)}\right)-\Theta\right)}_{<0}\Big\} \geq 1.$$

□

Beweis zu Satz 7, Seite 77

Satz 7. *$J^*(A(x),\Theta) := \mathbb{E}[\exp\{-y(A(x)-\Theta)\}]$ nimmt sein Minimum an bei*

$$A(x) = \frac{1}{2}\ln\frac{p(y=1|x)}{p(y=-1|x)} + \Theta.$$

Damit gilt:

$$p(y=1|x) = \frac{\exp\{2A(x)-\Theta\}}{1+\exp\{2A(x)-\Theta\}}$$

und

$$p(y=-1|x) = \frac{\exp\{-2A(x)-\Theta\}}{1+\exp\{-2A(x)-\Theta\}}.$$

Beweis.

$$\mathbb{E}[\exp\{-yA(x)-\Theta\}] \to \min.$$

$$\begin{aligned}
\mathbb{E}[\exp\{-yA(x)-\Theta\}|\,x] &= p(y=1|x)\exp\{-(A(x)-\Theta)\} \\
&\quad + p(y=-1|x)\exp\{A(x)-\Theta\} \\
\frac{\partial\mathbb{E}[\exp\{-yA(x)-\Theta\}|\,x]}{\partial A(x)} &= -p(y=1|x)\exp\{-(A(x)-\Theta)\} \\
&\quad + p(y=-1|x)\exp\{A(x)-\Theta\} \stackrel{!}{=} 0
\end{aligned}$$

$$\begin{aligned}
\iff \quad & p(y=-1|x)\exp\{2(A(x)-\Theta)\} = p(y=1|x) \qquad (A.7)\\
\implies \quad & A(x) = \frac{1}{2}\ln\frac{p(y=1|x)}{p(y=-1|x)} + \Theta.
\end{aligned}$$

Außerdem folgt mit (A.7):

$$\begin{aligned}
& (1-p(y=1|x))\exp\{2(A(x)-\Theta)\} = p(y=1|x) \\
\implies \quad & p(y=1|x) = \frac{\exp\{2(A(x)-\Theta)\}}{1+\exp\{2(A(x)-\Theta)\}},
\end{aligned}$$

und analog

$$p(y=-1|x) = \frac{\exp\{-2(A(x)-\Theta)\}}{1+\exp\{-2(A(x)-\Theta)\}}.$$

□

ANHANG B

Liste eigener Publikationen

[1] Schweiger, Roland ; Neumann, Heiko ; Ritter, Werner: Multiple-Cue Data Fusion with Particle Filters for Vehicle Detection in Night View Automotive Applications. In: *Proceedings of 2005 IEEE Intelligent Vehicles Symposium*. Las Vegas, USA, 2005, S. 753–758

[2] Idler, Corvin ; Schweiger, Roland ; Paulus, Dietrich ; Mählisch, Mirko ; Ritter, Werner: Realtime Vision Based Multi-Target-Tracking with Particle Filters in Automotive Applications. In: *Proceedings of 2006 IEEE Intelligent Vehicles Symposium*. Tokio, Japan, 2006, S. 188–193

[3] Kallenbach, I. ; Schweiger, R. ; Palm, G. ; Löhlein, O.: Multi-class Object Detection in Vision Systems Using a Hierarchy of Cascaded Classifiers. In: *Proceedings of 2006 IEEE Intelligent Vehicles Symposium*. Tokio, Japan, 2006, S. 383–387

[4] Mählisch, Mirko ; Schweiger, Roland ; Ritter, Werner ; Dietmayer, Klaus: Multisensor Vehicle Tracking with the Probability Hypothesis Density Filter. In: *Proceedings of 9th International Conference on Information Fusion*. Florenz, Italien, 2006, S. 1–8

[5] Smuda, Peer ; Schweiger, Roland ; Neumann, Heiko ; Ritter, Werner: Multiple Cue Data Fusion with Particle Filters for Road Cource Detection in Vision Systems. In: *Proceedings of 2006 IEEE Intelligent Vehicles Symposium*. Tokio, Japan, 2006, S. 400–405

[6] Mählisch, Mirko ; Schweiger, Roland ; Ritter, Werner ; Dietmayer, Klaus: Sensorfusion Using Spatio-Temporal Aligned Video and Lidar for Improved Vehicle Detection. In: *Proceedings of 2006 IEEE Intelligent Vehicles Symposium*. Tokio, Japan, 2006, S. 424–429

[7] Schweiger, Roland ; Serfling, Matthias ; Dietmayer, Klaus ; Ritter, Werner: A Survey on the Usability of Digital Maps for Automotive Safety Applications. In: *Proceedings of the 9th International IEEE Conference on Intelligent Transportation Systems*. London, GB, 2006

[8] Schweiger, Roland ; Hamer, Henning ; Löhlein, Otto: Determining Posterior Probabilities on the Basis of Cascaded Classifiers as used in Pedestrian Detection Systems. In: *Proceedings of 2007 IEEE Intelligent Vehicles Symposium*. Istanbul, Türkei, 2007, S. 1284–1289

[9] Arndt, Richard ; Schweiger, Roland ; Ritter, Werner ; Paulus, D. ; Löhlein, Otto: Detection and Tracking of Multiple Pedestrians in Automotive Applications. In: *Proceedings of 2007 IEEE Intelligent Vehicles Symposium*. Istanbul, Türkei, 2007, S. 13–18

[10] Serfling, Matthias ; Schweiger, Roland ; Ritter, Werner: Road course estimation in a night vision application using a digital map, a camera sensor and a prototypical imaging radar system. In: *Proceedings of IEEE Intelligent Vehicles Symposium*. Eindhoven, Niederlande, 2008, S. 810–815

[11] Serfling, Matthias ; Löhlein, Otto ; Schweiger, Roland ; Dietmayer, Klaus: Camera and imaging radar feature level sensorfusion for night vision pedestrian recognition. In: *Proceedings of 2009 IEEE Intelligent Vehicles Symposium*. Xi'an, China, 2009, S. 597–603

[12] Franz, Stefan ; Schweiger, Roland ; Löhlein, Otto ; Kroschel, Kristian: Analysis and assessment of far infrared sensor performance parameters and their impact on pedestrian detection. In: *Proceedings of 13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Funchal, Portugal, 2010, S. 119–124

[13] Schweiger, R. ; Franz, S. ; Löhlein, O. ; Ritter, W. ; Källhammer, J.-E. ; Franks, J. ; Krekels, T.: Sensor fusion to enable next generation low cost Night Vision systems. In: *Proceedings SPIE 7726*, 2010, S. 772610–772610–11

[14] Franz, S. ; Schweiger, R. ; Löhlein, O. ; Willersin, D. ; Kroschel, K.: Performance evaluation of FIR sensor systems applied to pedestrian detection. In: *Proceedings of the SPIE Infrared Imaging Systems: Design, Analysis, Modelling, and Testing XXI* Bd. 7662, 2010, S. 766217–766217–11

[15] Schweiger, Roland ; Löhlein, Otto ; Ritter, Werner ; Källhammer, Jan-Erik: Low cost next generation multi sensor Night Vision System. In: *Transport Research Arena Europe*. Brüssel, Belgien, 2010, S. III–4

ANHANG C

Liste studentischer Arbeiten

[1] Idler, Corvin: *Visuelle probabilistische Mehrobjektverfolgung im Rahmen der Fahrzeugumfeldbeobachtung.* Koblenz, Universität Koblenz-Landau, Diplomarbeit, 2005

[2] Spörl, Benjamin: Detektion von Fahrzeugen in Nachtsicht- Videosequenzen mit Polynomklassifikator in Verbindung mit Partikelfitler als Hypothesengenerator / Technische Universität Ilmenau, SG Informatik. Ilmenau, 2005. – Forschungsbericht

[3] Smuda, Peer: *Fusionierung von Bildinformationen und digitaler Karte zur Spurverlaufsdetektion auf Landstraßen bei Nacht.* Ulm, Universität Ulm, Diplomarbeit, 2005

[4] Kallenbach, Ingo: *Multiklassen-Detektion mit verketteten Kaskadenklassifikatoren.* Ulm, Universität Ulm, Diplomarbeit, 2005

[5] Roth, Axel: *Fußgängerdetektion mit einem NIR-FIR-Fusionsansatz auf Basis eines Kaskadenklassifikators.* Ulm, Universität Ulm, Diplomarbeit, 2006

[6] Arndt, Richard: *Erkennung und Verfolgung mehrerer Fußgänger in Nachtsichtaufnahmen.* Koblenz, Universität Koblenz-Landau, Diplomarbeit, 2006

[7] Ewig, Andreas: *Fusion von Bild- und Radarsensordaten zur komponentenweisen Objektdetektion*, Universität Ulm, Diplomarbeit, 2006

[8] Serfling, Matthias: *Straßenverlaufserkennung mit Partikelfiltern unter Verwendung von Bildinformationen und digitaler Karte in Nachtsichtanwendungen.* Ilmenau, Technische Universität Ilmenau, Diplomarbeit, 2006

[9] Hamer, Henning: *Approximation of the Posterior Probability on the Basis of a Cascade Classifier for the Integration of Tracking into an Intelligent Pedestrian Searching Strategy.* Ulm, Universität Ulm, Diplomarbeit, 2007

[10] Hallerbach, Andreas: *Stereorekonstruktion aus korrespondierenden Objektdetektionen in kalibrierten NIR-/FIR-Bildfolgen.* Ulm, Universität Ulm, Diplomarbeit, 2007

[11] Christmann, Constantin: *Lokal-Adaptive Boosting-Trees.* Ulm, Universität Ulm, Diplomarbeit, 2007

[12] Brandsmeier, Holger: Datensynchronisation zur Sensordatenfusion / Universität Ulm. Ulm, 2007. – Forschungsbericht

[13] Nemec, Melanie: *Time-delayed cascaded classifier*, Universität Ulm, Diplomarbeit, 2007

[14] Pedak, Koidu: *Merkmalsuntersuchung und -erweiterung eines Straßenverlaufserkennungssystems in einer Nachtsichtanwendung.* Ulm, Universität Ulm, Diplomarbeit, 2007

[15] Thom, Markus: *Fußgängererkennung mit spärlicher Informationsverarbeitung.* Ulm, Universität Ulm, Diplomarbeit, 2008

[16] Hartmann, Oliver: *Nickwinkelbestimmung auf Basis des optischen Flusses in einer Nachtsichtumgebung*, Universität Ulm, Diplomarbeit, 2008

Literaturverzeichnis

[ABDL05] Andreone, L. ; Bellotti, F. ; De Gloria, A. ; Lauletta, R.: SVM-based pedestrian recognition on near-infrared images. In: *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis.*, 2005, 274–278

[AGW97] Amit, Jali ; Geman, Donald ; Wilder, Kenneth: Joint Induction of Shape Features and Tree Classifiers. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997), Nr. 11, S. 1300–1305

[ALS+07] Alonso, Ignacio P. ; Llorca, David F. ; Sotelo, Miguel ¦ Bergasa, Luis M. ; Toro, Pedro Revenga D. ; Nuevo, Jesús ; Ocaña, Manuel ; Ángel, Miguel ; Garrido, García: Combination of Feature Extraction Methods for SVM Pedestrian Detection. In: *Transportation* 8 (2007), Nr. 2, S. 292–307

[Arn06] Arndt, Richard: *Erkennung und Verfolgung mehrerer Fußgänger in Nachtsichtaufnahmen*, Universität Koblenz-Landau, Diplomarbeit, 2006

[BB07] Beutnagel-Buchner, Uwe et. a.: *NIRWARN: BMBF-Projekt; Gesamtschlussbericht.* Technische Informationsbibliothek u. Universitätsbibliothek Hannover, 2007

[BBF+06] Bertozzi, M ; Broggi, A ; Felisa, M ; Vezzoni, G ; Del Rose, M: Low-level Pedestrian Detection by means of Visible and Far Infra-red Tetra-vision. In: *Proceedings of 2006 IEEE Intelligent Vehicles Symposium.* Tokio, Japan, 2006, S. 231–236

[BBFS00] Broggi, A ; Bertozzi, M ; Fascioli, A ; Sechi, M: Shape-based pedestrian detection. In: *Proceedings of 2000 IEEE Intelligent Vehicles Symposium.* Dearborn, USA, 2000, S. 215–220

[BBGM07] Bertozzi, M ; Broggi, A ; Ghidoni, S ; Meinecke, M.M.: A night vision module for the detection of distant pedestrians. In: *Proceedings of 2007 IEEE Intelligent Vehicles Symposium*. Istanbul, Türkei, 2007, S. 25–30

[BD01] Boers, Yvo ; Driessen, J.N.: Particle Filter Based Detection for Tracking. In: *Proceedings of the American Control Conference* Bd. 6. Arlington, USA, 2001, S. 4393–4397

[BEHW89] Blumer, Anselm ; Ehrenfeucht, Andrzej ; Haussler, David ; Warmuth, K. Manfred: Learnability and the Vapnik-Chervonenkis Dimension. In: *Journal of the Association for Computing Machinery* 36 (1989), Nr. 4, S. 929–965

[BFSO84] Breiman, Leo ; Friedmann, Jerome ; Stone, Charles T. ; Olshen, R. A.: *Classification and Regression Trees*. Chapman and Hall/CRC, 1984

[BFT06] Broggi, A ; Fedriga, RL ; Tagliati, A: Pedestrian detection on a moving vehicle: an investigation about near infra-red images. In: *Proceedings of 2006 Intelligent Vehicles Symposium*. Tokio, Japan, 2006, S. 431–436

[BI98] Blake, Andrew ; Isard, Michael: *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. New York, USA : Springer-Verlag New York, Inc., 1998. – ISBN 3540762175

[Bis06] Bishop, Christopher M.: *Pattern Recognition and Machine Learning*. New York, USA : Springer Science+Business Media, LLC, 2006

[BMH04] Bolič, Miodrag ; M., Djurič P. ; Hong, Sangjin: Resampling Algorithms for Particle Filters: A Computational Complexity Perspective. In: *EURASIP Journal of Applied Signal Processing* 2004 (2004), Nr. 15, S. 2267–2277

[bmw] BMW AG: *BMW Deutschland*. `http://www.bmw.de`. – Elektronisches Dokument http://www.bmw.de. Datum letzter Zugriff: 25.03.2009.

[Bor86] Borgefors, Gunilla: Distance transformations in digital images. In: *Computer Vision Graphics and Image Processing* 34 (1986), Nr. 3, S. 344–371

[Bou99] Bouguet, J.: *Visual methods for three-dimensional modeling*. Pasadena, USA, California Institute of Technology, Diss., 1999

[Bre97] Breiman, Leo: Prediction Games and Arcing Algorithms / Statistics Department, University of California. Berkeley, USA, December 1997 (504). – Forschungsbericht

[Bre98] Breiman, Leo: Arcing Classifiers. In: *The Annals of Statistics* 26 (1998), Nr. 3, S. 801–849

[BSLK01] Bar-Shalom, Yaakov ; Li, Xiao-Rong ; Kirubarajan, Thiagalingam: *Estimation with applications to tracking and navigation: Theory Algorithms and Software*. New York, USA : John Wiley and Sons, INC., 2001

[Cro84] Crow, Franklin C.: Summed-Area Tables for Texture Mapping. In: *Proceedings of SIGGRAPH* Bd. 18, 1984, S. 207–212

[CYW02] Challa, S. ; Yo, Ba-Ngu ; Wang, Xuezhi: Bayesian approaches to track existence - IPDA and random sets. In: *Proceedings of the 5th International Conference on Information Fusion* Bd. 2. Annapolis, USA, 2002, S. 1228–1235

[DC96] Drucker, H. ; Cortes, C.: Boosting Decision Trees. In: *Proceedings of Neural Information Processing Systems* Bd. 8, 1996, S. 479–485

[Den04] Denzler, Joachim: *Probabilistische Zustandsschätzung und Aktionsauswahl im Rechnersehen*. Berlin : Logos Verlag Berlin, 2004

[Die00] Dietterich, Thomas G.: A Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. In: *Machine Learning* 40 (2000), Nr. 2, S. 139–157

[DIN94] *DIN 70000:1994-01: Straßenfahrzeuge - Fahrzeugdynamik und Fahrverhalten - Begriffe (ISO 8855:1991, modifiziert)*. Beuth Verlag, Berlin, 1994

[Dou98] Doucet, Arnaud: On Sequential Simulation-Based Methods for Bayesian Filtering / Signal Processing Group, Cambridge University Engineering Department. 1998. – Forschungsbericht. – Technical Report CUED/F-INFENG/TR310

[DT05] Dalal, N. ; Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. San Diego, USA, 2005. – ISBN 0–7695–2372–2, S. 886–893

[DT06] Dalal, Navneet ; Triggs, Bill: Human Detection Using Oriented Histograms of Flow and Appearance. In: *Lecture Notes in Computer Science* 3952/2006 (2006), S. 428–441

[Efr79] Efron, Bradley: Bootstrap Methods: Another Look at the Jackknife. In: *The Annals of Statistics* 7 (1979), Nr. 1, S. 1–26

[EG09] Enzweiler, Markus ; Gavrila, Dariu M.: Monocular pedestrian detection: survey and experiments. In: *IEEE transactions on pattern analysis and machine intelligence* 31 (2009), Nr. 12, S. 2179–95

[EK03] Egelhaaf, Jan C. ; Knoll, Peter M.: The „Night Sensitive" Vehicle. In: *Der Fahrer im 21. Jahrhundert - Anforderungen, Anwendungen, Aspekte für Mensch-Maschine-Systeme*. Düsseldorf : VDI-Verlag, 2003 (VDI-Berichte 1768), S. 247–257

[Enz11] Enzweiler, Markus: *Compaund Models for Vision-Based Pedestrian Recognition*, Ruprecht-Karls-Universität Heidelberg, Diss., 2011

[FD98] Frean, Marcus ; Downs, Tom: A simple cost funcion for boosting / Dep. of Computer Science and Electrical Engineering, University of Queensland. Brisbane, Australien, 1998. – Forschungsbericht

[Fea98] Fearnhead, Paul: *Sequential Monte Carlo methods in filter theory*. Oxford, UK, University of Oxford, Diss., 1998

[FGG+98] Franke, Uwe ; Gavrila, Dariu ; Gorzig, Steffen ; Lindner, Frank ; Puetzold, F ; Wohler, Christian: Autonomous driving goes downtown. In: *Intelligent Systems and their Applications IEEE* 13 (1998), Nr. 6, S. 40–48

[FHT00] Friedman, Jerome ; Hastie, Trevor ; Tibshirani, Robert: Additive Logistic Regression: A Statistical View of Boosting. In: *The Annals of Statistics* 28 (2000), Nr. 2, S. 337–407

[FK96] Franke, U ; Kutzbach, I: Fast Stereo Based Object Detection for Stop and Go Traffic. In: *Proceedings 1996 IEEE Intelligent Vehicles Conference*. Tokio, Japan, 1996, S. 339–344

[Fre95] Freund, Yoav: Boosting a weak learning algorithm by majority. In: *Information and Computation* 121 (1995), Nr. 2, S. 256–285

[Fre99] Freitas, Nando de: *Bayesian Methods for Neural Networks*. Cambridge, UK, Trinity College, University of Cambridge, Diss., 1999

[Fri01] Friedman, Jerome H.: Greedy Function Approximation: A Gradient Boosting Machine. In: *Annals of Statistics* 29 (2001), Nr. 5, S. 1189–1232

[FS97] Freund, Yoav ; Schapire, Robert E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. In: *Journal of Computer and System Sciences* 55 (1997), S. 119–139

[Gav00] Gavrila, D M.: Pedestrian Detection from a Moving Vehicle. In: *ECCV European Conference on Computer Vision* 2 (2000), S. 37–49

[GGM04] Gavrila, DM ; Giebel, J ; Munder, S: Vision-based pedestrian detection: The protector system. In: *Intelligent Vehicles Symposium, 2004 IEEE*, IEEE, 2004, 13–18

[GGS04] Giebel, Jan ; Gavrila, Dariu M. ; Schnörr, Christoph: A Bayesian Framework for Multi-cue 3D Object Tracking. In: *Proceesings of 8th European Conference on Computer Vision (ECCV)*. Prag, Tschechische Repuplik, 2004, S. 241–252

[GLSG10] Gerónimo, David ; López, Antonio M. ; Sappa, Angel D. ; Graf, Thorsten: Survey of pedestrian detection for advanced driver assistance systems. In: *IEEE transactions on pattern analysis and machine intelligence* 32 (2010), Nr. 7, S. 1239–58

[GM07] Gavrila, D M. ; Munder, S: Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle. In: *International Journal of Computer Vision* 73 (2007), Nr. 1, S. 41–59

[GSLP07] Gerónimo, David ; Sappa, Angel D. ; López, Antonio M. ; Ponsa, Daniel: Adaptive Image Sampling and Windows Classification for On-board Pedestrian Detection. In: *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)*, 2007

[GSS93] Gordon, Neil ; Salmond, David ; Smith, Adrian F. M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation,. In: *IEE Proceedings Part F: Radar and Signal Processing* Bd. 140, 1993, S. 107–113

[GT07] Gandhi, T. ; Trivedi, M.M.: Pedestrian Protection Systems: Issues, Survey, and Challenges. In: *IEEE Transactions on Intelligent Transportation Systems* 8 (2007), September, Nr. 3, 413–430. `http://dx.doi.org/10.1109/TITS.2007.903444`. – DOI 10.1109/TITS.2007.903444. – ISSN 1524–9050

[Hal07] Hallerbach, Andreas: *Stereorekonstruktion aus korrespondierenden Objektdetektionen in kalibrierten NIR-/FIR-Bildfolgen.* Ulm, Universität Ulm, Diplomarbeit, 2007

[Has70] Hastings, W. K.: Monte Carlo Sampling Methods Using Markov Chains and Their Applications. In: *Biometrika* 57 (1970), Nr. 1, S. 97–109

[Hau05] Haug, A.J.: A Tutorial on Bayesian Estimation and Tracking Techniques Applicable to Nonlinear and Non-Gaussian Processes / MITRE. 2005 (05-0211). – Forschungsbericht

[HZ04] Hartley, Richard ; Zissermann, Andrew: *Multiple View Geometry in Computer Vision.* 2. Cambridge, UK : Cambridge University Press, 2004

[IB98a] Isard, Michael ; Blake, Andrew: CONDENSATION – conditional density propagation for visual tracking. In: *International Journal of Computer Vision* 29 (1998), Nr. 1, S. 5–28

[IB98b] Isard, Michael ; Blake, Andrew: ICONDENSATION: Unifying low-level and high-level Tracking in a Stochastic Framework. In: *Proceesings of 5th European Conference on Computer Vision (ECCV).* Freiburg, 1998, S. 893–908

[Idl05] Idler, Corvin: *Visuelle probabilistische Mehrobjektverfolgung im Rahmen der Fahrzeugumfeldbeobachtung.* Koblenz, Universität Koblenz-Landau, Diplomarbeit, 2005

[ISP+06] Idler, Corvin ; Schweiger, Roland ; Paulus, Dietrich ; Mählisch, Mirko ; Ritter, Werner: Realtime Vision Based Multi-Target-Tracking with Particle Filters in Automotive Applications. In: *Proceedings of 2008 IEEE Intelligent Vehicles Symposium.* Tokio, Japan, 2006, S. 188–193

[JS08] Jones, Michael J. ; Snow, Daniel: Pedestrian detection using boosted features over many frames. In: *2008 19th International Conference on Pattern Recognition*. Tampa, US, 2008, S. 1–4

[KM00] Koller-Meier, Esther B.: *Extending the condensation algorithm for tracking multiple objects in range image sequences*. Konstanz, Schweitz, ETH Zürich, Diss., 2000

[KMA01] Koller-Meier, Esther B. ; Ade, Frank: Tracking multiple objects using the Condensation algorithm. 34 (2001), Nr. 2-3, S. 93–105

[Kru07] Krueger, Lars: *Model Based Object Classification and Localisation in Multiocular Images*. Bielefeld, University of Bielefeld, Diss., 2007

[KSPL06] Kallenbach, I. ; Schweiger, R. ; Palm, G. ; Löhlein, O.: Multi-class Object Detection in Vision Systems Using a Hierarchy of Cascaded Classifiers. In: *Proceedings of 2006 IEEE Intelligent Vehicles Symposium*. Tokio, Japan, 2006, S. 383–387

[KT07a] Krotosky, S ; Trivedi, M: Mutual information based registration of multimodal stereo videos for person tracking. In: *Computer Vision and Image Understanding* 106 (2007), Nr. 2-3, S. 270–287

[KT07b] Krotosky, S J. ; Trivedi, M M.: On Color-, Infrared-, and Multimodal-Stereo Approaches to Pedestrian Detection. In: *IEEE Transactions on Intelligent Transportation Systems* 8 (2007), Nr. 4, S. 619–629

[LAE05] Lerner, Markus ; Albrecht, Martina ; Evers, Claudia: Das Unfallgeschehen bei Nacht. In: *Berichte der Bundesanstalt für Straßenwesen* Bd. M 172. Bremerhaven : Bundesanstalt für Straßenwesen, 2005

[LBH08] Lampert, Christoph ; Blaschko, Matthew ; Hofmann, Thomas: Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In: *IEEE Conference on Computer Vision and Pattern Recognition (2008)*, 2008, S. 1–8

[LCCV07] Leibe, Bastian ; Cornelis, N ; Cornelis, Kurt ; Van Gool, Luc: Dynamic 3d scene analysis from a moving vehicle. In: *2007 IEEE Conference on Pattern Recognition*, 2007, S. 1–8

[LF04] Liu, Xia ; Fujimura, Kikuo: Pedestrian Detection Using Stereo Night Vision. In: *IEEE Transactions on Vehicular Technology* 53 (2004), Nr. 6, S. 1657–1665

[LKP03] Lienhart, Rainer ; Kuranov, Alexander ; Pisarevsky, Vadim: Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. In: *25th Pattern Recognition Symposium (DAGM '03)*. Magdeburg, 2003, S. 297–304

[Loy03] Loy, Gareth: *Computer Vision to See People: a basis for enhanced human computer interaction*. Canberra, Australien, Australian National University, Diss., 2003

[LSCV08] Leibe, Bastian ; Schindler, Konrad ; Cornelis, Nico ; Van Gool, Luc: Coupled object detection and tracking from static cameras and moving vehicles. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2008), Nr. 10, S. 1683–1698

[LW04] Levi, K. ; Weiss, Y.: Learning object detection from a small number of examples: the importance of good features. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 (2004), S. 53–60

[Mac00] MacCormick, John: *Probabilistic models and stochastic algorithms for visual tracking*. Oxford, GB, University of Oxford, Diss., 2000

[Mah04] Mahler, Ronald P.: Random Sets: Unification and computation for information fusion - A retrospective assessment. In: *Proceedings of the 7th International Conference on Information Fusion*. Stockholm, Schweden, 2004, S. 1–20

[Mal89] Mallat, Stephane G.: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (1989), Nr. 7, S. 674–693

[MB00] MacCormick, John ; Blake, Andrew: A Probabilistic Exclusion Principle for Tracking Multiple Objects. In: *International Journal of Computer Vision* 39 (2000), Nr. 1, S. 57–71

[MBBF99] *Kapitel* Functional Gradient Techniques for Combining Hypotheses. In: Mason, L. ; Baxter, J. ; Bartlett, P. L. ; Frean, M.: *Advances in Large Margin Classifiers*. Cambridge, UK : MIT Press, 1999, S. 221–247

[mer] Daimler AG: *Mercedes Deutschland*. `http://www.mercedes-benz.de`. – Elektronisches Dokument http://www.mercedes-benz.de. Datum letzter Zugriff: 25.03.2009.

[MES94] Musicki, D. ; Evans, R. ; Stankovic, S.: Integrated probabilistic data association. In: *IEEE Transactions on Automatic Control* Bd. 39, 1994, S. 1237–1241

[MG06] Munder, S ; Gavrila, D M.: An experimental study on pedestrian classification. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), Nr. 11, S. 1863–1868

[Mäh10] Mählisch, Mirko: *Filtersynthese zur simultanen Minimierung von Existenz-, Assoziations- und Zustandsunsicherheiten in der Fahrzeugumfelderfassung mit heterogenen Sensordaten*. Ulm, Universität Ulm, Diss., 2010

[MOL+05] Mählisch, M. ; Oberlander, M. ; Lohlein, O. ; Gavrila, Dariu ; Ritter, Werner: A multiple detector approach to low-resolution FIR pedestrian recognition. In: *Proceedings of 2005 IEEE Intelligent Vehicles Symposium*. Las Vegas, USA, 2005, S. 325–330

[MPP01] Mohan, Anuj ; Papageorgiou, Constantine ; Poggio, Tomaso: Example-based object detection in images by components. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001), Nr. 4, S. 349–361

[MR03] Meir, R. ; Rätsch, G.: An Introduction to Boosting and Leveraging. In: *Advanced Lectures on Machine Learning*. Mendelson, S. and Smola, A. J., 2003, S. 118–183

[MRD07] Mählisch, Mirko ; Ritter, Werner ; Dietmayer, Klaus: De-cluttering with Integrated Probabilistic Data Association for Multisensor Multitarget ACC Vehicle Tracking. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*. Istambul, Türkei, 2007, S. 178–183

[MSRD06] Mählisch, Mirko ; Schweiger, Roland ; Ritter, Werner ; Dietmayer, Klaus: Multisensor Vehicle Tracking with the Probability Hypothesis Density Filter. In: *Proceedings of 9th International Conference on Information Fusion*. Florenz, Italien, 2006, S. 1–8

[MT93] Meyn, S.P. ; Tweedie, R.L.: *Markov Chains and Stochastic Stability*. Online-Edition, 2005. London, UK : Springer-Verlag, 1993 `http://www.probability.ca/MT/`

[ND02] Nanda, H ; Davis, L: Probabilistic template based pedestrian detection in infrared videos. In: *Proceedings of 2002 IEEE Intelligent Vehicle Symposium* Bd. 1. Versailles, Frankreich, 2002, S. 15–20

[OPS+97] Oren, Michael ; Papageorgiou, Constantine ; Sinha, Pawan ; Osuna, Edgar ; Poggio, Tomaso: Pedestriaon Detection Using Wavelet Templates. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*. Puerto Rico, USA, 1997, S. 193–199

[ORM98] Onoda, T. ; Rätsch, G. ; Müller, K.-R.: An asymptotic analysis of AdaBoost in the binary classification case. In: *Proceedings of the International Conference on Artificial Neuronal Networks*. Skövde, Schweden, 1998, S. 195–200

[OTF+04] Okuma, Kenji ; Taleghani, Ali ; Freitas, Nando de ; Little, James J. ; Lowe, David G.: A Boosted Particle Filter: Multitarget Detection and Tracking. In: *Proceedings of the 8th European Conference on Computer Vision*. Prag, Tschechische Republik, 2004, S. 28–39

[Pap91] Papoulis, Athanasios: *Probability, Random Variables and Stochastic Processes*. 3. McGraw-Hill Companies, 1991

[PL04] Porwik, Piotr ; Lisowska, Agnieszka: The Haar-Wavelet Transform in Digital Image Processing: Its Status and Achievements. In: *Machine Graphics and Vision* 13 (2004), Nr. 1/2, S. 27–98

[POP98] Papageorgiou, C. ; Oren, M. ; Poggio, T.: A general framework for Object Detection. In: *Proceedings of International Conference on Computer Vision*. Bombay, Indien, 1998, S. 555–562

[PP00] Papageorgiou, Constantine ; Poggio, Tomaso: A trainable system for object detection. In: *International Journal of Computer Vision* 38 (2000), Nr. 1, S. 15–33

[PTC] *The Principles of PTC Heating*. `http://www.europeanthermodynamics.com/heaters/The%20Principles%20of%20PTC%20Heating.pdf`. – Elektronisches Dokument. Datum letzter Zugriff: 19.08.2011.

[Qui96] Quinlan, J. R.: Bagging, Boosting and C4.5. In: *Proceedings of the 13th National Conference on Artificial Intelligence*. Portland, USA, 1996, S. 725–730

[RAY] *Berührungslose Temperaturmessung, IR Wissen - Fachbegriffe*. `http://www.raytek.de/Raytek/de-r0/IREducation/GlossaryTerms.htm`. – Elektronisches Dokument http://www.raytek.de/Raytek/de-r0/IREducation/GlossaryTerms.htm. Datum letzter Zugriff: 19.08.2011.

[Rid99] Ridgeway, Greg: The State of Boosting. In: *Computing Science and Statistics* 31 (1999), S. 172–181

[ROM01] Rätsch, G. ; Onoda, T. ; Müller, K.-R.: Soft Margins for AdaBoost. In: *Machine Learning* 42 (2001), Nr. 3, S. 287–320

[Rot06] Roth, Axel: *Fußgängerdetektion mit einem NIR-FIR-Fusionsansatz auf Basis eines Kaskadenklassifikators*. Ulm, Universität Ulm, Diplomarbeit, 2006

[Rub88] Rubin, D.B.: Using the SIR Algorithm to Simulate Posterior Distributions. In: Bernardo, J. (Hrsg.) ; DeGroot, M. (Hrsg.) ; Lindley, D. (Hrsg.) ; Smith, A. (Hrsg.): *Bayesian Statistics* Bd. 3. Amsterdam, Holland : Oxford University Press, 1988, S. 395–402

[Sch90] Schapire, Robert E.: The Strength of Weak Learnability. In: *Machine Learning* 5 (1990), Nr. 2, S. 197–227

[Sch99] Schapire, Robert E.: A Brief Introduction to Boosting. In: *Proceedings of the 16th International Joint Conference on Artificial Intelligence*. Stockholm, Schweden, 1999, S. 1401–1406

[Sch02] Schapire, Robert E.: Advances in Boosting. In: *Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference*. Alberta, Kanada, 2002, S. 446–452

[Sch03a] Schapire, Robert E.: The Boosting Approach to Machine Learning: An Overview. In: Denison, D. D. (Hrsg.) ; Hansen, M. H. (Hrsg.) ; Holmes, C. (Hrsg.) ; Mallick, B. (Hrsg.) ; Yu, B. (Hrsg.): *Lecture Notes in Statistics: Nonlinear Estimation and Classification* Bd. 171. New York, USA : Springer-Verlag New York Inc., 2003, S. S. 149–172

[Sch03b] Schick, Jens: Night Vision Improvements. 2003. – Forschungsbericht. – 4th ADASE II Expert Workshop on Sensors & Actuators

[Sch06] Schön, Thomas B.: *Estimation of Nonlinear Dynamic Systems - Theory and Applications*. Linköping, Schweden, Linköpings Universitet, Diss., 2006

[SDS02] Stollnitz, E.J. ; DeRose, A.D. ; Salesin, D.H.: Wavelets for computer graphics: a primer.1. In: *IEEE Computer Graphics and Applications* 15 (2002), Nr. 3, S. 76–84

[Ser11] Serfling, Matthias: *Merkmalsbasierte Fusion einer NIR-Kamera und eines bildgebenden Radarsensors zur Fußgängerwarnung bei Nacht*. Chemnitz, Technische Universität Chemnitz, Diss., 2011

[SFBL98] Schapire, Robert E. ; Freund, Yoav ; Bartlett, Peter ; Lee, Wee S.: Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. In: *The Annals of Statistics* 36 (1998), Nr. 5, S. 1651–1686

[SFL+10] Schweiger, R. ; Franz, S. ; Löhlein, O. ; Ritter, W. ; Källhammer, J.-E. ; Franks, J. ; Krekels, T.: Sensor fusion to enable next generation low cost Night Vision systems. In: *Proceedings SPIE 7726*, 2010, S. 772610–772610–11

[SGP04] Smith, Kevin ; Gatica-Perez, Daniel: Order Matters: A Distributed Sampling Method for Multi-Object Tracking. In: *Proceedings of the British Machine Vision Conference (BMVC)*. London, GB, 2004, S. 25–32

[Sid03] Sidenbladh, Hedwig: Multi-Target Particle Filtering for the Probability Hypothesis Density. In: *Proceedings of the 6th International Conference on Information Fusion*. Cairns, Australien, 2003, S. 800–806

[SLM10] Szczot, Magdalena ; Loehlein, Otto ; Mählisch, Mirko: Incorporating Contextual Information in Pedestrian Tracking. In: *WIT 2010: 7th International Workshop on Intelligent Transportation*, 2010

[SLSP09] Szczot, M. ; Löhlein, O. ; Serfling, M. ; Palm, G.: Incorporating contextual information in pedestrian recognition. In: *Proceedings of 2009 IEEE Intelligent Vehicles Symposium*. Xi'an, China, 2009, S. 364–369

[SNR05] Schweiger, Roland ; Neumann, Heiko ; Ritter, Werner: Multiple-Cue Data Fusion with Particle Filters for Vehicle Detection in Night View Automotive Applications. In: *Proceedings of 2005 IEEE Intelligent Vehicles Symposium*. Las Vegas, USA, 2005, S. 753–758

[Spe05] Spengler, Martin: *On the applicability of Sequential Monte Carlo methods to multiple target tracking*. Zürich, Schweiz, Swiss Federal Institute of Technology Zurich, Diss., 2005

[SPFN06] Sotelo, M.A.l. ; Parra, I. ; Fernandez, D. ; Naranjo, E.: Pedestrian Detection Using SVM and Multi-Feature Combination. In: *2006 IEEE Intelligent Transportation Systems Conference*, 2006. – ISBN 1–4244–0093–7, 103–108

[SR06] Suard, F ; Rakotomamonjy, A: Pedestrian detection using infrared images and histograms of oriented gradients. In: *2006 IEEE Intelligent Vehicles Symposium Proceedings*, 2006, 206–212

[SS99] Schapire, Robert E. ; Singer, Yoram: Improved Boosting Algorithms Using Confidence-rated Predictions. In: *Machine Learning* 37 (1999), Nr. 3, S. 297–336

[SSNR06] Smuda, Peer ; Schweiger, Roland ; Neumann, Heiko ; Ritter, Werner: Multiple Cue Data Fusion with Particle Filters for Road Cource Detection in Vision Systems. In: *Proceedings of 2006 IEEE Intelligent Vehicles Symposium*. Tokio, Japan, 2006, S. 400–405

[STL06] Sun, Yijun ; Todorovic, Sinisa ; Li, Jian: Reducing the Overfitting of AdaBoost by Controlling its Data Distribution Skewness. In: *International Journal of Pattern Recognition* 20 (2006), Nr. 7, S. 1093–1116

[Str01] Struzik, Z.R.: Oversampling the Haar Wavelet Transform / Centrum voor Wiskunde en Informatica. Centrum voor Wiskunde en Informatica, 2001 (2). – Forschungsbericht

[SYYO05] Szarvas, M. ; Yoshizawa, A. ; Yamamoto, M. ; Ogata, J.: Pedestrian detection with convolutional neural networks. In: *Proceedings of 2005 IEEE Intelligent Vehicles Symposium*. Las Vegas, USA, 2005, S. 224–229

[Tu05] Tu, Zhuowen: Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. In: *Proceedings of the 10th IEEE International Conference on Computer Vision* Bd. 2. Beijing, China, 2005, S. 1589–1596

[Ulm04] Ulmer, B.: ADASE - European Industrial Roadmap for the Integrated Road Safety Systems. In: *Proceedings of the 12th International Symposium ATA*. Parma, Italien, 2004

[VC71] Vapnik, V. N. ; Chervonenkis, A. Y.: On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. In: *Theory of Probability and its Applications* 16 (1971), Nr. 2, S. 264–280

[VDP03] Vermaak, Jaco ; Doucet, Arnaud ; Pérez, Patrick: Maintaining Multi-Modality through Mixture Tracking. In: *Proceedings of 9th International Conference on Computer Vision* Bd. 2. Nizza, France, 2003, 1110–1116

[VGP05] Vermaak, Jaco ; Godsill, Simon J. ; Pérez, Patrick: Monte Carlo Filtering for Multi-Target Tracking and Data Association. In: *IEEE Transactions on Aerospace and Electronic Systems* 41 (2005), Nr. 1, S. 309–332

[Vih04] Vihola, Matti: *Random Sets for Multitarget Tracking and Data Fusion*. Tampere, Finnland, Tampere University of Technology, Department of Engineering, Diss., 2004

[VJ01a] Viola, Paul ; Jones, Michael: Rapid Object Detection Using a Boosted Cascade of Simple Features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* Bd. 1. Kauai, USA, 2001, S. 511–518

[VJ01b] Viola, Paul ; Jones, Michael: Robust Real-time Object Detection. In: *Second International Worksop on Statistical and Computional Theories of Vision - Modeling, Learning, Computing, and Sampling.* Vancouver, Kanada, 2001

[VJ04] Viola, Paul ; Jones, Michael J.: Robust Real-Time Face Detection. In: *International Journal of Computer Vision* 57 (2004), Nr. 2, S. 137–154

[VJS03] Viola, Paul ; Jones, Michael ; Snow, Daniel: Detecting pedestrians using patterns of motion and appearance. In: *Proceedings Ninth IEEE International Conference on Computer Vision* 2 (2003), S. 734–741

[VMB05] Vermaak, Jaco ; Maskell, Simon ; Briers, Mark: Tracking a Variable Number of Targets using the Existence Joint Probabilistic Data Association Filter / Signal Processing Group, Cambridge University Engineering Department. 2005. – Forschungsbericht. – CUED/F-INFENG/TR.514

[Vor08] Vorndran, Ingeborg: Unfallgeschehen im Straßenverkehr 2007. In: Radermacher, Walter (Hrsg.): *Wirtschaft und Statistik* Bd. 7. Wiesbaden : Statistisches Bundesamt, 2008, S. 592–602

[VS04] Vo, Ba-Ngu ; Singh, Sumeetpal: On The Bayes Filtering Equations of Finite Set Statistics. In: *Proceedings of 5th Asian Control Conference ASCC'2004.* Melbourne, Australien, 2004, 1273–1278

[VSD05] Vo, Ba-Ngu ; Singh, Sumeetpal ; Doucet, Arnaud: Sequential Monte Carlo methods for Multi-target Filtering with Random Finite Sets. In: *IEEE Transactions on Aerospace and Electronic Systems* 41 (2005), Nr. 4, S. S. 1224–1245

[WA99] Wöler, C. ; Anlauf, J.: An Adaptable Time-Delay Neural-Network Algorithm for Image Sequence Analysis. In: *IEEE Transactions on Neural Networks* 10 (1999), Nr. 6, S. 1531–1536

[WDSS08] Wojek, Christian ; Dorkó, Gyuri ; Schulz, André ; Schiele, Bernt: Sliding-Windows for Rapid Object Class Localization: A Parallel Technique. In: *Pattern Recognition* Bd. 5096. Springer Berlin Heidelberg, 2008, S. 71–81

[XLF05] Xu, F ; Liu, X ; Fujimura, K: Pedestrian Detection and Tracking With Night Vision. In: *IEEE Transactions on Intelligent Transportation Systems* 6 (2005), Nr. 1, S. 63–71

[ZAYC06] Zhu, Q. ; Avidan, S. ; Yeh, M.-C. ; Cheng, K.-T.: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* Bd. 2, 2006. – ISBN 0–7695–2597–0, S. 1491–1498

[ZT00] Zhao, Liang ; Thorpe, C E.: Stereo- and neural network-based pedestrian detection. In: *IEEE Transactions on Intelligent Transportation Systems* 1 (2000), Nr. 3, 148–154. http://dx.doi.org/10.1109/6979.892151. – DOI 10.1109/6979.892151. – ISSN 15249050

[ZWN07] Zhang, Li ; Wu, Bo ; Nevatia, Ram: Pedestrian Detection in Infrared Images based on Local Shape Features. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, S. 1–8

[ZZ10] Zhang, C. ; Zhang, Zhengyou: A survey of recent advances in face detection / Microsoft Research. Citeseer, 2010 (June). – Forschungsbericht

Index