
Systematische Untersuchung von Radar Tracking-Algorithmen

Masterthesis Nr. 637/16

Bearbeiter: Stefan Martin Ackermann | 1546440

Betreuer: Martin Holder, M.Sc.

Co-Betreuer: Philipp Rosenberger, M.Sc.



TECHNISCHE
UNIVERSITÄT
DARMSTADT



FAHRZEUGTECHNIK
TU DARMSTADT

Stefan Martin Ackermann
Matrikelnummer: 1546440
Studiengang: Master Mechatronik

Masterthesis Nr. 637/16
Thema: Systematische Untersuchung von Radar Tracking-Algorithmen

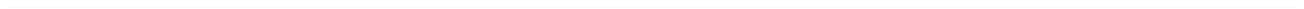
Eingereicht: 14. Juni 2017

Technische Universität Darmstadt
Fachgebiet Fahrzeugtechnik
Prof. Dr. rer. nat. Hermann Winner
Otto-Berndt-Straße 2
64287 Darmstadt



Lizenz: CC BY 4.0 International

<https://creativecommons.org/licenses/by/4.0/deed.de>



Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig angefertigt habe. Sämtliche aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und noch nicht veröffentlicht.

Darmstadt, den 14. Juni 2017

Zusammenfassung

In modernen Fahrzeugen werden vermehrt Fahrerassistenzsysteme zur Steigerung der Sicherheit und des Komforts eingesetzt. In Zukunft werden als Folge des technologischen Fortschritts die Anzahl und Komplexität dieser Systeme weiter zunehmen. Für das Testen und die Freigabe hochautomatisierter Fahrfunktionen besteht die Forderung, neue Qualitätsstandards und Methoden zu entwickeln. Zu diesem Zweck haben sich 17 Projektpartner aus der deutschen Automobilindustrie zu dem Verbundprojekt "Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen (PEGASUS)"¹ zusammengeschlossen. Das Ziel des Projekts ist die Entwicklung eines einheitlichen Vorgehens im Bereich Test und Erprobung. Darüber hinaus wird europaweit im "ENABLE-S3"-Projekt² an bereichsübergreifenden virtuellen Plattformen für die Validierung und Verifizierung hochautomatisierter Funktionen geforscht. Der Gültigkeitsbereich dieser Plattformen erstreckt sich von der Luftfahrt, über die Automobilindustrie, die Landwirtschaft, das Gesundheitswesen bis zur Schifffahrt und die Bahnindustrie.

Das Fachgebiet Fahrzeugtechnik Darmstadt (FZD) ist an beiden Projekten beteiligt. Für das PEGASUS-Projekt werden Validierungsmethoden von Simulationsmodellen für aktive Sensoren entwickelt. Im Zuge dessen wird das Modell eines Radarsensors erarbeitet, das eine hohe Modellgüte aufweist. Für die Verwendung der simulierten Radarsignale in nachgelagerten Fahrerassistenzsystemen ist die Detektion und zeitliche Verfolgung von statischen und dynamischen Objekten innerhalb der Signale erforderlich. Dies gelingt mithilfe sogenannter Tracking-Algorithmen. Zurzeit steht dem Fachgebiet FZD kein solcher Algorithmus zur Verfügung. In dieser Thesis wird daher die Analyse und Entwicklung eines Tracking-Algorithmus beschrieben.

Für die Implementierung wird zunächst eine Analyse des aktuellen Stands der Entwicklungen von Tracking-Verfahren durchgeführt. Im Anschluss wird ein Open Source Tracking-Algorithmus implementiert, der eine Identifizierung und Untersuchung seiner Schlüsselparameter zulässt. Für die Verwendung des Tracking-Algorithmus mit Signalen eines realen Radarsensors wird eine Schnittstelle zwischen beiden Bausteinen definiert. Um einen belastbaren Vergleich der Leistungsfähigkeit verschiedener Tracking-Algorithmen zu erhalten, wird eine Analyse bekannter Kriterien und Metriken durchgeführt. Mithilfe dieser Methoden werden die Einflüsse von Variationen der Schlüsselparameter auf die Qualität der Tracking-Ergebnisse untersucht. Gleichzeitig werden Konzepte zur Erstellung von Testfällen für Tracking-Algorithmen dargelegt. Zusammen mit den Metriken wird damit eine Bewertungsgrundlage für die Qualität von Tracking-Ergebnissen präsentiert.

Die Untersuchungen des implementierten Algorithmus zeigen, dass dieser die geforderten Funktionen eines Tracking-Verfahrens erfüllt. Durch die Auswertung der Tracking-Ergebnisse bei Variation der Schlüsselparameter werden die Einflüsse einzelner Parameter auf die Tracking-Qualität deutlich. Während dieser Analyse zeigt sich, dass die gewählten Metriken und Testfälle eine systematische Untersuchung der Tracking-Ergebnisse zulassen. Sie sind folglich auch für zukünftige Einschätzungen der Tracking-Qualität weiterer Algorithmen einsetzbar.

¹ Bundesministerium für Wirtschaft und Energie: Homepage Projekt Pegasus (2016)

² ECSEL Joint Undertaking: Homepage ENABLE-S3 (2016)

Inhaltsverzeichnis

Zusammenfassung	I
Inhaltsverzeichnis	II
Formelzeichen und Indizes	V
Abkürzungsverzeichnis	VI
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	X
1 Einleitung	1
1.1 Motivation	1
1.2 Konkretisierung der Aufgabenstellung	2
1.3 Methodik des Vorgehens	3
2 Der Radarsensor	5
2.1 Grundlagen des Radarverfahrens	6
2.1.1 Herleitung der Radargleichung	6
2.1.2 Der Doppler-Effekt	8
2.1.3 Verfahren zur Bestimmung des Azimuthwinkels	8
2.2 Radarsysteme	10
2.2.1 FM-CW Radar	10
2.2.2 Pulsradar	10
2.2.3 Pulskompressionsradar	15
2.3 Radarsignalverarbeitung des Pulskompressionsradars	17
2.4 Schnittstelle zwischen Radarsensor und Tracking-Algorithmus	19
3 Radar Tracking-Algorithmen	20
3.1 Einführung	20
3.2 Universelle Struktur eines Tracking-Algorithmus	22
3.3 Vorstellung verschiedener Tracking-Algorithmen	24
3.3.1 Sensordatenverarbeitung	24
3.3.2 Datenassoziation	26
3.3.3 Track-Management	29
3.3.4 Filterung und Prädiktion	29
3.3.5 Gate-Berechnung	32
3.4 Implementierung des Radar Tracking-Algorithmus	32
3.4.1 Mathematische Grundlagen	33
3.4.2 Tracking-Parameter	36
3.4.3 Definition der Objekt-Tracker	39
3.4.4 Reale Sensordaten	39
3.4.5 Simulierte Sensordaten	39
3.4.6 Clustergruppengenerierung	40
3.4.7 Datenassoziation	41

3.4.8	Aktualisierung der Objekt-Tracker	42
3.4.9	Löschen von Tracker-Duplikaten	43
3.4.10	Initialisierung neuer Objekt-Tracker.....	43
3.4.11	Bewertung der Tracking-Ergebnisse.....	44
4	Leistungsbewertung von Tracking-Algorithmen.....	45
4.1	Vergleichskriterien für die Analyse von Multiple-Target Tracking-Algorithmen	45
4.1.1	Zeitliche Kriterien.....	45
4.1.2	Genauigkeit der Tracking-Ergebnisse	46
4.1.3	Kontinuität der Objektverfolgung.....	47
4.2	Metriken für Multiple-Target Tracking-Algorithmen	47
4.2.1	Hausdorff-Metrik.....	48
4.2.2	Optimal Mass Transfer Metrik	49
4.2.3	Optimal Subpattern Assignment Metrik	50
4.2.4	Optimal Subpattern Assignment for Tracks Metrik	52
4.2.5	Generalized Optimal Subpattern Assignment Metrik	55
4.2.6	Bento et al. Metrik	56
4.2.7	Rahmathullah et al. Metrik.....	57
4.2.8	Weitere Metriken zum Vergleich von Tracking-Algorithmen	59
4.3	Testfälle für Multiple-Target Tracking-Algorithmen	59
4.3.1	Anforderungen an die Testfälle.....	59
4.3.2	Testfall 1: Formationsfahrt	61
4.3.3	Testfall 2: Autobahnfahrt.....	63
4.4	Erzeugen simulierter Radardaten	66
5	Analyse des implementierten Tracking-Algorithmus	67
5.1	Wahl der Metriken für den Leistungsvergleich von Tracking-Algorithmen.....	67
5.2	Vorstellung der Schlüsselparameter des Tracking-Algorithmus	67
5.2.1	Schlüsselparameter der Sensordatengenerierung und -verarbeitung	67
5.2.2	Schlüsselparameter der Clustergruppengenerierung.....	69
5.2.3	Schlüsselparameter der Datenassoziation.....	70
5.2.4	Schlüsselparameter der Objekt-Tracker-Aktualisierung	71
5.2.5	Schlüsselparameter der Löschung von Tracker-Duplikaten	73
5.2.6	Schlüsselparameter der Tracking-Algorithmus-Metriken	74
5.3	Ergebnisse der Tracking-Algorithmus-Analyse.....	75
5.3.1	Parametervariation der Sensordatengenerierung	81
5.3.2	Parametervariation der Clustergruppengenerierung	87
5.3.3	Parametervariation der Datenassoziation	88
5.3.4	Parametervariation der Objekt-Tracker.....	89
5.3.5	Parametervariation der Tracking-Algorithmus-Metriken	91
6	Tracking-Ergebnisse mit realen Radardaten.....	93
6.1	Testfahrt mit zwei Zielfahrzeugen	93

6.2	Testfahrt mit Corner Reflector.....	95
6.3	Vergleich des implementierten Algorithmus mit einer Industrielösung	97
7	Fazit und Ausblick	98
7.1	Fazit.....	98
7.2	Ausblick	98
7.2.1	Der Radarsensor	99
7.2.2	Der Tracking-Algorithmus	99
7.2.3	Die Leistungsbewertung der Tracking-Algorithmen	100
A	Anhang	101
A.1	Programmablaufpläne des implementierten Tracking-Algorithmus	101
A.2	Testfall 1: Bewegungsgleichungen Objekt 1	111
A.3	Testfall 1: Bewegungsgleichungen Objekt 2	112
A.4	Testfall 1: Bewegungsgleichungen Objekt 3	113
A.5	Einzelbild aus dem Objekt-Tracking 2	114
A.6	Einzelbild: Tracking mit realen Radardaten 2.....	115
	Literaturverzeichnis.....	116

Formelzeichen und Indizes

Lateinische Buchstaben:

Zeichen	Einheit	Bezeichnung
<i>A</i>	m ²	Fläche
<i>a</i>	$\frac{\text{m}}{\text{s}^2}$	Beschleunigung
<i>B</i>	Hz	Rauschbandbreite
<i>c</i>	$\frac{\text{m}}{\text{s}}$	Lichtgeschwindigkeit
<i>D</i>	m	Breite der Antenne
<i>d</i>	m	Abstand
<i>dT</i>	s	Zeitliche Differenz
<i>f</i>	Hz	Frequenz
<i>G</i>	dB	Antennengewinn
<i>k</i>	$\frac{\text{J}}{\text{K}}$	Boltzmann-Konstante
<i>L</i>	dB	Dämpfung
<i>N</i>	W	Noise (Rauschen)
<i>P</i>	W	Leistung
<i>p</i>	1	Wahrscheinlichkeit
<i>R</i>	m	Radius
<i>S</i>	$\frac{\text{W}}{\text{m}^2}$	Leistungsdichte
<i>T</i>	K; s	Rauschtemperatur; Zeitabstand
<i>t</i>	s	Zeit
<i>v</i>	$\frac{\text{m}}{\text{s}}$	Geschwindigkeit
<i>x</i>	m	Sensorfeste Koordinate
<i>y</i>	m	Sensorfeste Koordinate

Griechische Buchstaben:

Zeichen	Einheit	Bezeichnung
α	°	Winkel
λ	m	Wellenlänge
ρ	1	Apertur(Antennenfläche)-Wirkungsgrad
σ	m ² ; m oder $\frac{m}{s}$	Radarquerschnitt; Standardabweichung
τ	s	Pulsdauer; Laufzeit
θ	°	Keulenbreite
φ	°	Phasenwinkel
ϑ	°	Azimuthwinkel

Indizes:

Zeichen	Bezeichnung
<i>3dB</i>	Absenkung um 3 dB (50%)
<i>atm</i>	Atmosphäre
<i>COHO</i>	Coherent Oscillator
<i>d,D</i>	Doppler
<i>e</i>	Empfangsrichtung
<i>eff</i>	Effektiv
<i>k,i,j</i>	Laufindizes
<i>l</i>	links
<i>P</i>	Puls
<i>r</i>	rechts
<i>s</i>	Senderichtung
<i>STALO</i>	Stabilized Local Oscillator
<i>stör</i>	Störgröße
<i>sys</i>	System
<i>ZF</i>	Zwischenfrequenz

Abkürzungsverzeichnis

ACC	Adaptive Cruise Control
CA	Constant Acceleration
COHO	Coherent Oscillator
CTR	Constant Turn Rate
CV	Constant Velocity
CW	Continuous Wave
EKF	Extended Kalman Filter
ENABLE	European Initiative to Enable Validation for Highly Automated Safe and Secure Systems
FISST	Finite Set Statistic
FM-CW	Frequency-Modulated Continuous Wave
FZD	Fachgebiet Fahrzeugtechnik Darmstadt
GM-PHD	Gaussian Mixture - Probability Hypothesis Density filter
GOSPA	Generalized Optimal Subpattern Assignment
IEEE	Institute of Electrical and Electronics Engineers
IMM	Interacting Multiple Model
JPDAF	Joint probabilistic data association filter
LIDAR	Light detection and ranging
MHT	Multiple Hypothesis Tracking filter
MoE	Measures of Effectiveness
OMAT	Optimal Mass Transfer
OSPA	Optimal Subpattern Assignment
OSPA-T	Optimal Subpattern Assignment for Tracks
PDAF	Probabilistic data association filter
PEGASUS	Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen
PHD	Probability Hypothesis Density filter
RCS	Radar cross section
RFS	Random Finite Set
RMSE	Root-mean-square error
STALO	Stabilized Local Oscillator
SVSF	Smooth Variable Structure filter
ULB	Universitäts- und Landesbibliothek

Abbildungsverzeichnis

1-1	Signalverarbeitung zur Verdeutlichung der Motivation	2
1-2	Darstellung der Methodik des Vorgehens	3
2-1	Prinzip eines Pulsradars	5
2-2	Phased-Array-Antenne mit zwei Elementen	9
2-3	Sende- und Empfangssignale des Pulsradars mit einem Ziel	11
2-4	Sende- und Empfangssignale des Pulsradars mit zwei Zielen - Trennbar	11
2-5	Sende- und Empfangssignale des Pulsradars mit zwei Zielen 2 - Untrennbar	12
2-6	Auflösung der Distanzmehrdeutigkeit bei Pulsradaren	13
2-7	Radarsignalverarbeitung des Puls-Doppler-Radars	14
2-8	Darstellung der Entfernung-Doppler-Karte	14
2-9	Sende- und Empfangssignale bei Verwendung des Pulskompressionsverfahrens	16
2-10	Blockschaltbild der Radarsignalverarbeitung	18
3-1	Universelle Struktur eines Tracking-Algorithmus	23
3-2	Reflexionszentren einer Personenkraftwagenkarosserie	26
3-3	Datenassoziation mit einem Objekt und drei Messwerten	27
3-4	Datenassoziation mit zwei Objekten und vier Messwerten	27
3-5	Beispiel eines erweiterten Track-Managements	30
3-6	Blockschaltbild des implementierten Tracking-Algorithmus	33
4-1	Szenarien für den Vergleich der Hausdorff-, OMAT- und OSPA-Metrik	52
4-2	Beispielzuordnung von Trajektorien für die OSPA-T-Metrik	55
4-3	Beispielsituation: Vorteil GOSPA gegenüber OSPA-Metrik	56
4-4	Beispielsituation: Vorteil Rahmatullah et al.-Metrik gegenüber Bento-Metrik	58
4-5	Testfall 1: Formationsfahrt	62
4-6	Vektoren der Beschleunigungen	63
4-7	Testfall 2: Autobahnfahrt	64
5-1	Einzelbild aus dem Objekt-Tracking mit künstlichen Radardaten	77
5-2	OSPA-T Metrik: Anzahl Kardinalitätssprünge	78
5-3	OSPA-T Metrik: Übersicht Kardinalität	78
5-4	OSPA-T Metrik: Grafische Darstellung der OSPA-T-Bewertung	79
5-5	OSPA-T Metrik: Trajektorienverlauf mit Kardinalitätssprüngen	80
6-1	Einzelbild: Tracking mit realen Radardaten	94
6-2	Einzelbild: Tracking mit realen Radardaten (mit Corner Reflector)	96
A-1	Programmablaufplan: Einlesen der realen Sensordaten	101
A-2	Programmablaufplan: Generieren der Testtrajektorien	102
A-3	Programmablaufplan: Clustergruppengenerierung	103
A-4	Programmablaufplan: Rekursive Suche nach potentiellen Clustergruppenmitgliedern	104
A-5	Programmablaufplan: Datenassoziation	105
A-6	Programmablaufplan: Aktualisierung der Tracker-Status	106

A-7 Programmablaufplan: Korrektur und Prädiktion der Objekt-Tracker-Zustände	107
A-8 Programmablaufplan: Löschen von Objekt-Tracker-Duplikaten	108
A-9 Programmablaufplan: Initialisierung neuer Objekt-Tracker.....	109
A-10 Programmablaufplan: Bewertung der Tracking-Qualität	110
A-11 Einzelbild aus dem Objekt-Tracking 2	114
A-12 Einzelbild: Tracking mit realen Radardaten 2.....	115

Tabellenverzeichnis

3-1	Parameter des Tracking-Algorithmus 1	37
3-2	Parameter des Tracking-Algorithmus 2	38
3-3	Zusätzliche Parameter des Objekt-Trackers	39
3-4	Zwischenergebnis Clustergruppengenerierung.....	40
3-5	Beispiel Datenassoziation.....	42
3-6	Statuswerte der Objekt-Tracker	43
4-1	Parameter des Testfalls 1: Formationsfahrt	63
4-2	Parameter des Testfalls 2: Autobahnfahrt.....	65
5-1	Schlüsselparameter der Sensordatengenerierung	68
5-2	Schlüsselparameter der Sensordatengenerierung 2	69
5-3	Schlüsselparameter der Clustergruppengenerierung	70
5-4	Schlüsselparameter der Datenassoziation.....	71
5-5	Schlüsselparameter der Objekt-Tracker Aktualisierung 1	72
5-6	Schlüsselparameter der Objekt-Tracker Aktualisierung 2	73
5-7	Schlüsselparameter der Löschung von Tracker-Duplikaten	74
5-8	Schlüsselparameter der Tracking-Algorithmus-Metriken	75
5-9	Tracking-Qualität bei Standardparametrierung	75
5-10	Tracking-Qualität: Variation Objektbeschleunigung	81
5-11	Tracking-Qualität: Variation Objektbeschleunigung mit neuem Modell	82
5-12	Tracking-Qualität: Variation mittlere Objektgeschwindigkeit.....	82
5-13	Tracking-Qualität: Variation minimale Objektdistanzen.....	83
5-14	Tracking-Qualität: Variation des Sensorrauschens im Testfall 1.....	84
5-15	Tracking-Qualität: Variation des Sensorrauschens im Testfall 2.....	85
5-16	Tracking-Qualität: Variation der Sensorauflösung im Testfall 1	86
5-17	Tracking-Qualität: Variation der Sensorauflösung im Testfall 2	86
5-18	Tracking-Qualität: Variation der Clustergruppenparameter im Testfall 1	87
5-19	Tracking-Qualität: Variation der Clustergruppenparameter im Testfall 2	88
5-20	Tracking-Qualität: Variation der Datenassoziationsparameter im Testfall 1	89
5-21	Tracking-Qualität: Variation der Datenassoziationsparameter im Testfall 2	89
5-22	Tracking-Qualität: Variation der Objekt-Tracker Parameter im Testfall 1	90
5-23	Tracking-Qualität: Variation der Objekt-Tracker Parameter im Testfall 2	90
5-24	Tracking-Qualität: Variation der Tracking-Algorithmus-Metrik Parameter im Testfall 1.....	91
5-25	Tracking-Qualität: Variation der Tracking-Algorithmus-Metrik Parameter im Testfall 2.....	92

1 Einleitung

Ein Entwicklungsschwerpunkt im Automobilbereich liegt zurzeit auf der Einführung von Fahrerassistenzsystemen mit einer immer höheren Autonomiestufe³. Heutige Assistenzsysteme werden überwiegend in die Autonomiestufen 1 (z.B. Adaptive Cruise Control (ACC)) oder 2 (z.B. Spurhaltefunktion und Stauassistent) eingeteilt. In den kommenden Jahren werden auch hochautomatisierte Assistenzsysteme zur Serienreife gelangen. Das Testen und Freigeben dieser Fahrfunktionen erfordert gleichzeitig die Erforschung neuer Methoden und Standards zur Gewährleistung der Systemsicherheit.

1.1 Motivation

Die deutsche Automobilindustrie hat sich zu diesem Zweck zum Forschungsverbund PEGASUS⁴ zusammengeschlossen. Insgesamt 17 Automobilunternehmen, Zulieferer und Forschungseinrichtungen arbeiten zusammen an der Erarbeitung von Methoden und Standards zur Absicherung hochautomatisierter Fahrfunktionen. Ziel des Projekts ist es, die Möglichkeit zu schaffen, hochautomatisierte Fahrfunktionen schnell weiterentwickeln zu können, um die rasche Einführung des automatisierten Fahrens in der Praxis zu ermöglichen. Das FZD der TU Darmstadt ist einer dieser Projektpartner. Am Fachgebiet werden unter anderem Sensormodelle für aktive Sensoren erforscht. Diese Modelle ermöglichen es, reale Testfahrten durch Simulationen zu ersetzen und damit den Testaufwand neuer Assistenzsysteme zu reduzieren.

Ebenfalls beteiligt ist das FZD am europaweiten ENABLE-S3-Projekt⁵. Ziel dieses Projekts ist die Entwicklung von Methoden zur Absicherung hochautomatisierter Funktionen. Das Projekt beschränkt sich dabei nicht nur auf den automobilen Einsatz, sondern schließt auch Anwendungen in der Landwirtschaft, der Luftfahrt, der Schifffahrt, dem Gesundheitswesen und der Bahnindustrie mit ein.

Für die Messung der Distanzen und Relativgeschwindigkeiten zu Objekten im Umfeld eines Fahrzeugs werden Radarsensoren eingesetzt. Für automatisierte Fahrfunktionen ist es erforderlich, dass statische und dynamische Objekte in der Fahrzeugumgebung detektiert und über konsekutive Messungen verfolgt werden. Diese Objekterkennung und -verfolgung leisten sogenannte Tracking-Algorithmen.

Zur Entwicklung eines virtuellen Radarsensormodells steht dem FZD der Radarsensor eines Automobilzulieferers zur Verfügung. Die Software des Sensors erlaubt allerdings keinen Einblick in die genaue Funktionsweise des Objekt-Tracking-Moduls. Der Sensor gibt Rohdaten und das spätere Objekt-Tracking-Ergebnis aus, die eigentliche Objektverfolgung geschieht intern. Da kein Einblick in die genaue Funktionsweise des Algorithmus vorhanden ist, wird in der vorliegenden Master Thesis eine Analyse von Tracking-Algorithmen durchgeführt und anschließend ein Open Source-Objekt-Tracking-Modul für den Einsatz am Fachgebiet FZD implementiert.

Abbildung 1-1 zeigt eine Übersicht der Signalverläufe des Gesamtsystems. Die obere Reihe zeigt

³ Bundesanstalt für Straßenwesen: Rechtsfolgen zunehmender Fahrzeugautomatisierung (2012)

⁴ Bundesministerium für Wirtschaft und Energie: Homepage Projekt Pegasus (2016)

⁵ ECSEL Joint Undertaking: Homepage ENABLE-S3 (2016)

die Signalverarbeitung des realen Radarsensors mit einer Blackbox für das Objekt-Tracking-Modul der Industrielösung. In der unteren Reihe wird die Signalverarbeitung des zu implementierenden Tracking-Algorithmus gezeigt. Die Leistungsfähigkeit beider Objekt-Tracking-Pfade wird am Ende miteinander verglichen.

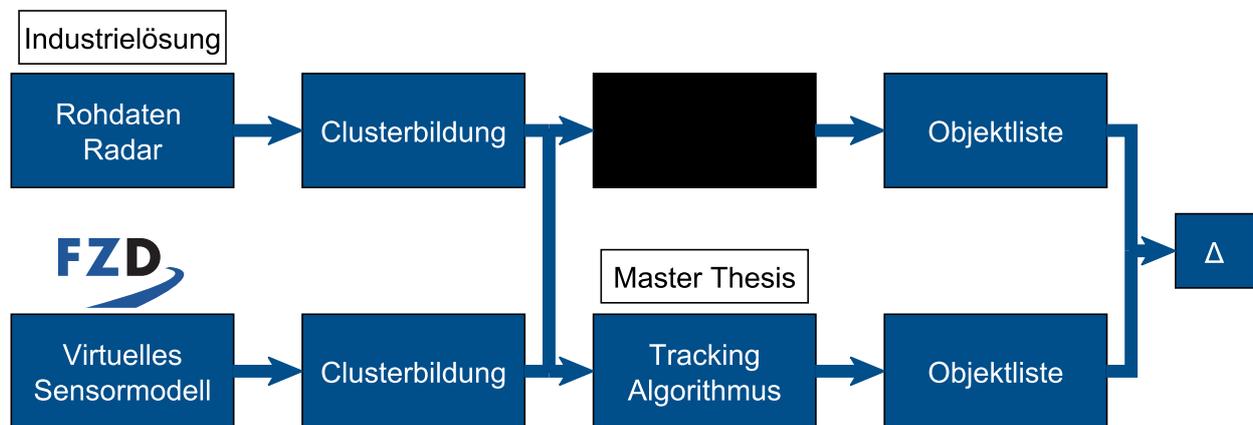


Abbildung 1-1: Signalverarbeitung des Gesamtsystems zur Verdeutlichung der Motivation. In der oberen Reihe ist die Blackbox der Industrielösung zu sehen, direkt darunter das zu implementierende Objekt-Tracking Modul

1.2 Konkretisierung der Aufgabenstellung

Im Rahmen dieser Master Thesis wird ein Radarsensor für die Detektion und Verfolgung von Objekten im Umfeld eines Fahrzeugs eingesetzt. Für ein Verständnis der Funktionsweise und Signalverarbeitung dieses Sensors wird eine Einarbeitung in dessen Hard- und Software durchgeführt. Hierfür wird sowohl die Dokumentation des Sensorherstellers als auch Fachliteratur herangezogen.

Für die Definition einer Schnittstelle zwischen Sensor und Tracking-Algorithmus werden die Ausgangssignale des Radarsensors analysiert. Gleichzeitig wird eine Liste mit Eingangssignalen erstellt, die für die Funktionsweise von Tracking-Algorithmen benötigt werden. Die endgültige Definition der Schnittstelle wird anschließend in Rücksprache mit dem Sensorhersteller festgelegt.

Für die Implementierung eines Tracking-Algorithmus wird eine Übersicht über Realisierungsmöglichkeiten erstellt. Hierfür wird eine Literaturrecherche durchgeführt. Im Anschluss daran wird ein Tracking-Algorithmus ausgewählt und in Matlab implementiert.

Durch eine weitere Literaturrecherche werden Kriterien für einen systematischen Vergleich der Leistungsfähigkeit verschiedener Tracking-Algorithmen identifiziert. Parallel dazu werden Methoden für die Erstellung von Testfällen dargelegt.

Als nächstes wird eine Analyse des implementierten Tracking-Algorithmus durchgeführt. Hierfür werden die Schlüsselparame-ter des Algorithmus identifiziert und die Einflüsse einer Variation dieser Parameter auf die Tracking-Ergebnisse untersucht.

Abschließend wird die Tracking-Qualität des implementierten Algorithmus mit der Industrielösung verglichen.

1.3 Methodik des Vorgehens

Für die Erstellung dieser Master Thesis werden im Vorfeld die durchzuführenden Arbeitsschritte definiert und festgehalten. Eine Darstellung dieser Schritte zeigt Abbildung 1-2.



Abbildung 1-2: Darstellung der Methodik des Vorgehens

Die Arbeit an dieser Master Thesis beginnt mit einer primären Literaturrecherche zu den Themen Radarsensoren, Tracking-Algorithmen und Tracking-Algorithmus-Metriken. Hierfür werden verschiedenen Datenbanken (bspw. Institute of Electrical and Electronics Engineers (IEEE)⁶, Universitäts- und Landesbibliothek (ULB) Darmstadt⁷ und Google Scholar⁸) durchsucht. Durch die Analyse der

⁶ IEEE: Homepage IEEE (2017)

⁷ ULB Darmstadt: Homepage ULB TU Darmstadt (2017)

⁸ Google: Homepage Google Scholar (2017)

Literaturverzeichnisse gefundener Medien wird weitere Literatur für eine Sekundärrecherche identifiziert.

Die Ergebnisse der Literaturrecherche zum Thema Radarsensoren werden in Kapitel 2 zusammengefasst. Da der Radarsensor des FZD auf dem Pulskompressionsverfahren beruht, wird dieses Verfahren bevorzugt dargestellt. Beendet wird das Kapitel mit der Vorstellung der Schnittstelle zwischen Radarsensor und Tracking-Algorithmus.

In Kapitel 3 wird eine Übersicht über verschiedene Tracking-Algorithmen präsentiert. Im Anschluss daran wird die Implementierung des ausgewählten Tracking-Algorithmus beschrieben.

Im folgenden Kapitel 4 wird eine Analyse von Kriterien und Metriken für einen Vergleich der Qualität von Tracking-Ergebnissen durchgeführt. In diesem Kapitel wird zusätzlich die Methodik zur Erstellung von Testfällen für Tracking-Algorithmen dargelegt.

In Kapitel 5 werden die Schlüsselparameter des implementierten Tracking-Algorithmus identifiziert. Mithilfe der in Kapitel 4 definierten Metriken und Testfälle werden anschließend die Einflüsse dieser Parameter auf die Tracking-Qualität untersucht.

Einen Vergleich der Tracking-Qualität zwischen dem implementierten Tracking-Algorithmus und der Industrielösung wird in Kapitel 6 durchgeführt. Zusätzlich werden die Tracking-Ergebnisse des implementierten Algorithmus bei der Verwendung von realen Radarsensordaten beleuchtet.

In Kapitel 7 wird abschließend ein Ausblick auf mögliche Erweiterungen des vorgestellten Tracking-Algorithmus gegeben.

2 Der Radarsensor

Die ursprüngliche Bezeichnung des Ortungssystems “Radar“ lautet “Radio Aircraft Detection and Ranging“⁹, heutzutage wird es meist “Radio Detection and Ranging“ genannt. Entwickelt wird dieses Sensorsystem in den 1920er Jahren, ursprünglich für die Ortung von Flugzeugen. Durch den zweiten Weltkrieg hat das Radar insbesondere im militärischen Bereich an Bedeutung gewonnen. Während dieser Zeit wird die Entwicklung stark vorangetrieben, um größere Reichweiten und höhere Genauigkeiten zu erreichen. Das Radar ist ein aktives Ortungsgerät, es ist daher nicht auf fremde Beleuchtungsquellen angewiesen. Ein weiterer Vorteil ist, dass die Frequenzen der Radarstrahlen in einem Spektrum liegen, das nicht durch äußere Umwelteinflüsse, wie Regen, Schnee oder Nebel, beeinflusst wird^{10a}. Der Nachteil ist die geringe Auflösung. Einzelheiten von detektierten Objekten werden nicht erkannt^{10b}. Abbildung 2-1 zeigt das Prinzip eines Pulsradars.

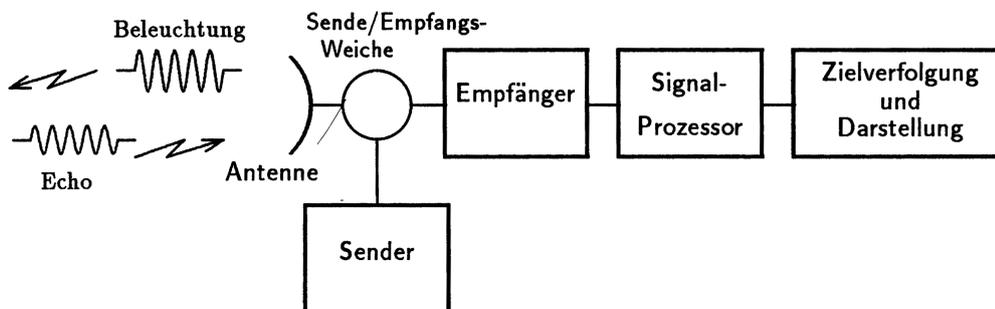


Abbildung 2-1: Prinzip eines Pulsradars¹¹

Für die Ortung von Objekten wird eine elektromagnetische Welle von einer Radarantenne ausgestrahlt. Diese wird an Objekten, die sich in Ausbreitungsrichtung befinden, reflektiert und zu einem Radarempfänger zurückgeschickt. Das Radarprinzip beruht dabei auf den Annahmen, dass die Wellenausbreitung geradlinig und die Ausbreitungsgeschwindigkeit bekannt ist¹². Auf diese Weise kann aus der Laufzeit der ausgesendeten Welle vom Sender über das Objekt zum Empfänger die Distanz des Objekts berechnet werden. Die Frequenzen der ausgestrahlten elektromagnetischen Wellen liegen bei automobilen Radarsensoren bei etwa 76,5 Ghz¹³.

Quellen für die folgenden Unterkapitel sind Bücher der Autoren Skolnik¹⁴, Richards¹⁵, Göbel¹⁶ und Ludloff¹⁷.

⁹ Göbel, J.: Radartechnik (2011) S.17

¹⁰ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.1-1

¹¹ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.1-2

¹² Göbel, J.: Radartechnik (2011) S.20

¹³ Winner, H. et al.: Handbuch Fahrerassistenzsysteme (2015) S.262

¹⁴ Skolnik, M.: Radar handbook (2008)

¹⁵ Richards, M. A.: Fundamentals of radar signal processing (2014)

¹⁶ Göbel, J.: Radartechnik (2011)

¹⁷ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008)

2.1 Grundlagen des Radarverfahrens

Im folgenden werden die Grundlagen des Radarverfahrens vorgestellt. Für den Entwurf eines Radars ist die Radargleichung von Bedeutung. Mit ihr wird mit bekannten Parametern annähernd die maximale Reichweite eines Radars berechnet¹⁸. Die Messung der Radialgeschwindigkeiten zwischen Radarsensor und Objekten im Fahrzeugumfeld basiert auf dem Doppler-Effekt. Darüber hinaus werden Verfahren zur Ermittlung des Azimutwinkels zwischen Sensorachse und beobachteten Objekten vorgestellt.

2.1.1 Herleitung der Radargleichung

Die maximale Reichweite eines Radars wird über die Radargleichung berechnet. Diese bildet die Basis des Radarentwurfs. Grundlage der Radargleichung ist ein hypothetischer kugelförmiger Radarstrahler in einem leeren Raum. Wird von diesem Kugelstrahler eine elektromagnetische Welle ausgestrahlt, so verteilt sich dessen Leistung auf einer Kugeloberfläche mit der Position des Strahlers als Mittelpunkt. Die Leistungsdichte S , die ein Objekt im Abstand R vom Radarsensor trifft, wenn die ausgestrahlte Leistung P_s beträgt, ist damit

$$S = \frac{P_s}{4\pi \cdot R^2}. \quad (2-1)$$

Die Antenne eines Radarsensors ermöglicht die Konzentration der ausgestrahlten elektromagnetischen Energie in eine gewünschte Richtung. Diese Konzentration wird Antennengewinn G genannt. Er ist definiert als das Verhältnis der Leistungsdichte der Radarantenne zur Leistungsdichte eines idealen Kugelstrahlers. Die Formel des Antennengewinns lautet

$$G = \frac{4\pi\rho A}{\lambda^2} = \frac{4\pi A_{\text{eff}}}{\lambda^2}. \quad (2-2)$$

Hierbei ist λ die Wellenlänge des ausgestrahlten Signals, ρ der Apertur(Antennenfläche)-Wirkungsgrad und A die geometrische Antennenoberfläche¹⁹. Durch diesen Antennengewinn wird ein Strahlungsdiagramm mit großer Hauptkeule und kleineren Nebenkeulen erzeugt. Die Wahl der Hauptkeulenbreite unterliegt einem Zielkonflikt. Für eine gute Clutterunterdrückung ist eine große Keulenbreite zu bevorzugen, für eine gute Winkelauflösung ist auf der anderen Seite eine möglichst geringe Breite anzustreben^{20a}. Die Keulenbreite ist definiert als der Bereich um das Maximum der Sendeleistung bis hin zu den Bereichen, bei denen die Leistung auf die Hälfte, also um 3dB, abgefal-

¹⁸ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.2-1

¹⁹ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.2-17

²⁰ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.2-18

len ist^{20b}. Die Keulenbreite wird durch Formel 2-3^{20c} abgeschätzt. D ist hier die Breite der Antenne in der Ebene des Keulenwinkels.

$$\theta_{3\text{dB}} \approx 65 \cdot \frac{\lambda}{D} \quad (2-3)$$

Trifft die ausgestrahlte Welle auf ein Objekt, so wird diese reflektiert. Die reflektierte Leistung ist dabei abhängig vom Radarquerschnitt σ des Objekts. Auch die reflektierte Welle breitet sich kugelförmig aus, sodass die Leistungsdichte beim Empfänger mit Einbezug des Antennengewinns

$$S = \frac{P_s \cdot G_s}{4\pi \cdot R^2} \cdot \frac{\sigma}{4\pi \cdot R^2} \quad (2-4)$$

beträgt.

Der Radarquerschnitt ist abhängig von der Frequenz des Sendesignals und dem Aspektwinkel. Eine Untersuchung der Radareigenschaften von Personenkraftwagenkarosserien wird von Marx²¹ durchgeführt.

Werden Verluste in den Signalleitungen (L_s) und durch die Atmosphäre (L_{atm}), sowie die effektive Fläche der Empfangsantenne (A_e) hinzugefügt, so ergibt sich die Empfangsleistung am Radarsensor nach Formel 2-5²².

$$P_e = \frac{P_s \cdot G_s \cdot \sigma \cdot A_e}{(4\pi)^2 \cdot R^4} \cdot \frac{1}{L_s \cdot L_{\text{atm}}} \quad (2-5)$$

Durch die Empfangsantenne und Auswerteelektronik wird ein Rauschen erzeugt, das die Empfangssignale überlagert. Dieses Rauschen wird als Systemrauschtemperatur T_{sys} zusammengefasst. Unter Einbezug der Boltzman-Konstante k und der Rauschbandbreite B wird das Signal-Rausch-Verhältnis am Ausgang des Empfängers durch Formel 2-6 angegeben. Die Eingangsleistung S entspricht dabei der Empfangsleistung P_e .

$$\left(\frac{S}{N}\right)_{\text{aus}} = \frac{P_s \cdot G_s \cdot \sigma \cdot A_e}{(4\pi)^2 \cdot R^4 \cdot k T_{\text{sys}} B} \cdot \frac{1}{L_s \cdot L_{\text{atm}}} \quad (2-6)$$

²¹ Marx, B. J.: Radareigenschaften von Personenkraftwagenkarosserien (2014)

²² Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.2-3

Wird A_e in Formel 2-6 mithilfe von Gleichung 2-2 ersetzt, so ergibt sich die allgemeine Radargleichung nach Formel 2-7²³. Diese gilt für alle Radarsignalförmigkeiten. Der Gewinn der Empfangsantenne wird G_e genannt.

$$\left(\frac{S}{N}\right)_{\text{aus}} = \frac{P_s \cdot G_s \cdot G_e \cdot \lambda^2 \cdot \sigma}{(4\pi)^3 \cdot R^4 \cdot kT_{\text{sys}} \cdot B} \cdot \frac{1}{L_s \cdot L_{\text{atm}}} \quad (2-7)$$

2.1.2 Der Doppler-Effekt

Neben der Laufzeit des ausgesendeten Radarsignals, aus der die Distanz des beobachteten Objekts hervorgeht, wird mit dem Radarsensor auch die Radialgeschwindigkeit ermittelt. Hierfür wird der Doppler-Effekt ausgenutzt. Dieser beschreibt die Veränderung der Frequenz des Reflexionssignals, wenn das Zielobjekt eine Relativgeschwindigkeit zum Sensor aufweist. Der Effekt zeigt sich als Frequenzverschiebung f_d zwischen dem Sende- und Empfangssignal. Mithilfe dieser Frequenzverschiebung wird mit Gleichung 2-8 die Relativgeschwindigkeit berechnet. f_s ist die Sendefrequenz und c die Lichtgeschwindigkeit.

$$v = \frac{1}{2} \lambda f_d = \frac{1}{2} c \frac{f_d}{f_s} \quad (2-8)$$

Die Abtastung der Frequenzverschiebung genügt dabei dem Abtasttheorem, da sonst Mehrdeutigkeiten durch Unterabtastung entstehen²⁴.

2.1.3 Verfahren zur Bestimmung des Azimutwinkels

Für die Messung des Azimutwinkels entfernter Objekte beim Einsatz von Radarsensoren stehen verschiedene Verfahren zur Verfügung. Angestrebt wird eine Radarkeule mit einer geringen Breite, um eine möglichst hohe Konzentration der abgestrahlten Energie zu erhalten. Mit dieser schmalen Keule ist die Abdeckung großer Azimutwinkel nicht möglich. Eine Lösung ist die mechanische Schwenkung der Radarantenne über den abzudeckenden Winkelbereich. Dieses Verfahren wird Scanning genannt²⁵. Während des Schwenkvorgangs werden die Radarwellen ausgestrahlt und auf den Empfang der Reflexionen gewartet. Aus dem aktuellen Winkel der Radarantenne und den ermittelten Signallaufzeiten werden die Azimutwinkel der Umfeldobjekte bestimmt. Nachteil dieses Verfahrens ist die verschleißende mechanische Bewegung der Radarantenne. Ein weiteres Verfahren ist der Monopuls bzw. der Mehrstrahler²⁶. Hierbei werden zwei oder mehr Empfangsantennen für die Winkelbestimmung eingesetzt. Das Sendesignal wird ausgestrahlt und die reflektierten Wellen in den

²³ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.2-19

²⁴ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.2-25ff

²⁵ Winner, H. et al.: Handbuch Fahrerassistenzsysteme (2015) S.281

²⁶ Winner, H. et al.: Handbuch Fahrerassistenzsysteme (2015) S.281ff

Empfangsantennen detektiert. Aus dem Phasenunterschied zwischen den Empfangssignalen wird der Azimuthwinkel bestimmt.

Beim Radarsensor des FZD wird der Azimuthwinkel über eine Phased-Array-Antenne (Digital Beam Forming) ermittelt^{27,28}. Für dieses Verfahren werden mehrere Empfangsantennen mit bekanntem Abstand auf der Breite des Sensors verteilt. Die Anordnung erzeugt einen elektronisch schwenkbaren Radarstrahl, bei dem die Nachteile einer mechanischen Bewegung entfallen. Der Azimuthwinkel wird bei dieser Antennenart über die Phasendifferenz der reflektierten Signale in den Empfangsleitungen der Antennen ermittelt. Abbildung 2-2 zeigt das Prinzip einer Phased-Array-Antenne mit zwei Antennenelementen.

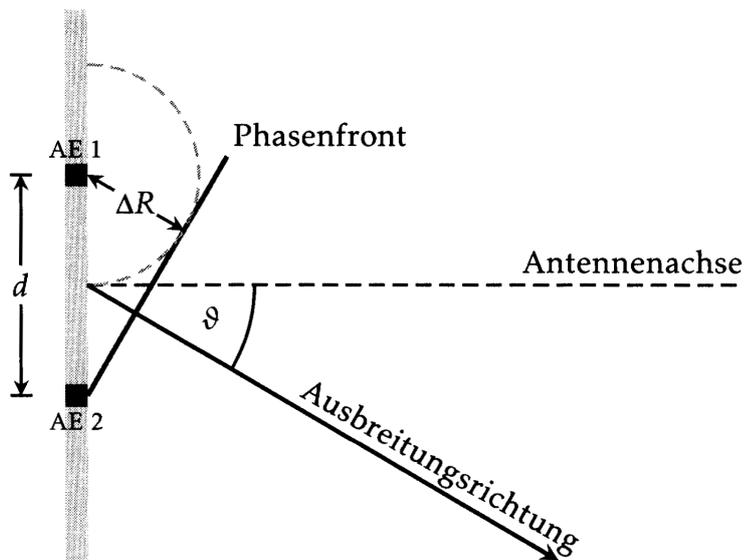


Abbildung 2-2: Phased-Array-Antenne mit zwei Elementen²⁹

Die sich einstellende Phasendifferenz wird mit Formel 2-10 unter Einsatz von Formel 2-9 berechnet.

$$\Delta R = d \sin \vartheta \quad (2-9)$$

$$\Delta \varphi = 2\pi \frac{\Delta R}{\lambda} = 2\pi \frac{d \sin \vartheta}{\lambda} \quad (2-10)$$

In die Berechnung geht der Abstand d der einzelnen Empfangsantennen und die Wellenlänge λ des Radarsignals ein. Für jedes weitere Empfangselement wird mithilfe obiger Gleichung die zu erwartende Phasendifferenz berechnet. Aus allen gemessenen Phasendifferenzen wird anschließend der Azimuthwinkel ϑ ermittelt.

²⁷ Göbel, J.: Radartechnik (2011) S.37ff

²⁸ Skolnik, M.: Radar handbook (2008) Kapitel 13

²⁹ Göbel, J.: Radartechnik (2011) S.39

2.2 Radarsysteme

Für die Gestaltung eines Radars sind zwei verschiedene Radarsysteme bekannt. Sowohl für das Dauerstrichradar (Continuous Wave (CW)), als auch für das Pulsradar existieren weitere Untervarianten. Im folgenden wird das Dauerstrichradarsystem und danach das, durch seine Mehrzielfähigkeit für den automobilen Einsatz besser geeignete, Pulsradar vorgestellt.

2.2.1 FM-CW Radar

Eine Erweiterung des allgemeinen CW-Radar ist das Frequency-Modulated Continuous Wave (FM-CW) Radar. Bei diesem Radarsystem wird ein kontinuierliches sinusförmiges frequenzmoduliertes Radarsignal ausgesendet. Die Frequenz des Signals wird dabei linear verändert. Nach der Reflexion dieses Signals an Umfeldobjekten wird es von der Empfangsantenne des Radarsensor detektiert. Aus der Laufzeit des Signal wird der Abstand des Objekts zum Sensor berechnet³⁰. Das FM-CW Radar ermöglicht zusätzlich die Bestimmung der Relativgeschwindigkeit mithilfe des Doppler-Effekts. Hierfür wird bei ansteigender Sendefrequenz eine andere Steigung verwendet als bei der abfallenden Frequenz. Ergebnis ist ein lineares Gleichungssystem, aus dem sowohl der radiale Abstand, als auch die radiale Geschwindigkeit hervorgeht. Großer Nachteil des FM-CW Radars ist die schlechte Mehrzielfähigkeit. Im automobilen Einsatz ist diese Fähigkeit allerdings gefordert, da in einem Verkehrsumfeld eine Vielzahl von Objekten zu verfolgen sind. Aus diesem Grund liegt der Fokus der Entwicklung von Radarsysteme auf Pulsradaren.

2.2.2 Pulsradar

Beim Pulsradar wird das Sendesignal in kurzen Pulsen ausgestrahlt. Nach Reflexion des Signals an Umfeldobjekten wird das Signal von der Empfangsantenne des Radarsensors detektiert. Aus der Laufzeit des Pulses vom Sender über das Objekt zum Empfänger wird der Abstand des Objekts berechnet. Abbildung 2-3 zeigt die Sende- und Empfangssignale eines Pulsradars bei Beobachtung eines Ziels. τ_p ist die Pulsdauer und T_p der Pulsabstand des Sendesignals.

Die Laufzeit des Sendesignals berechnet sich nach Formel 2-11

$$\tau = \frac{2R}{c}, \quad (2-11)$$

wobei R die Distanz des Objekts vom Sensor und c die Lichtgeschwindigkeit ist. Wird diese Formel umgestellt, so ergibt sich für den Abstand des Objekts

$$R = \frac{c\tau}{2}. \quad (2-12)$$

³⁰ Göbel, J.: Radartechnik (2011) S.172ff

³¹ Göbel, J.: Radartechnik (2011) S.116

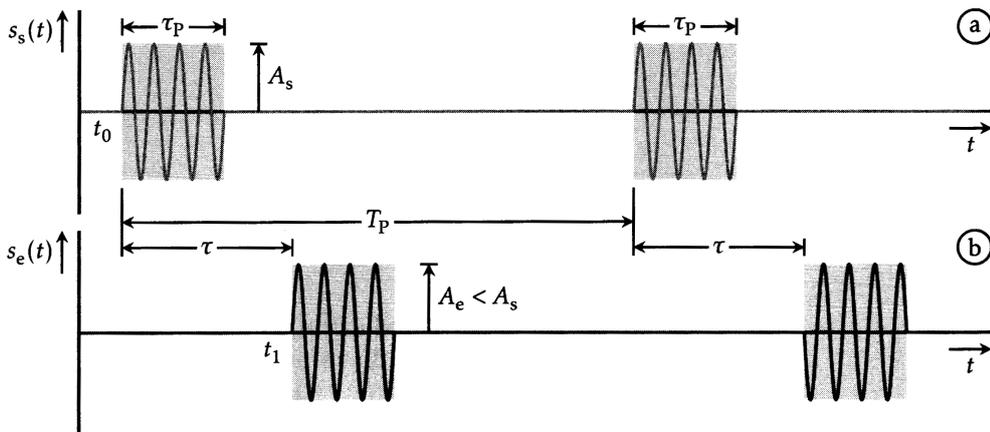


Abbildung 2-3: Sende- (a) und Empfangssignale (b) eines Pulsradars bei Beobachtung eines Ziels³¹

Bei mehreren Zielen werden auch mehrere Empfangssignale detektiert. In Abbildung 2-4 ist eine Trennung der beiden Empfangssignale möglich.

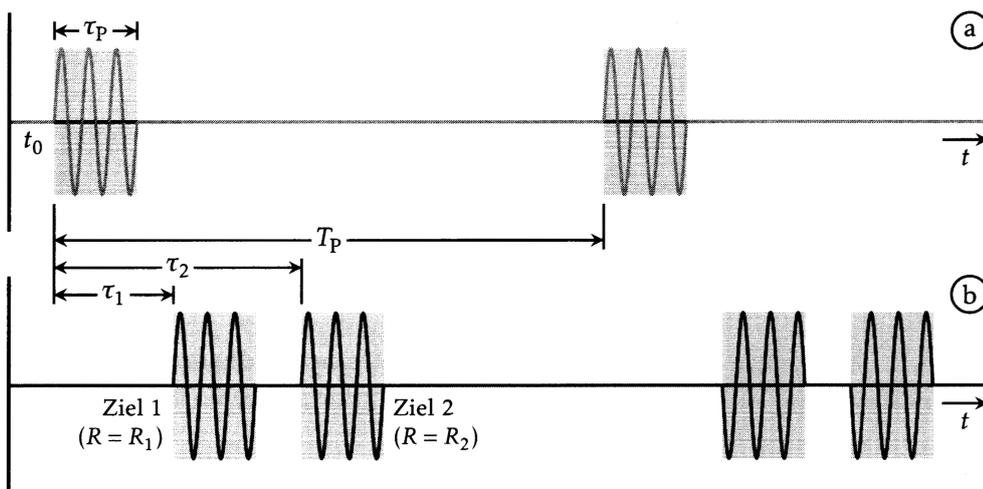


Abbildung 2-4: Sende- (a) und Empfangssignale (b) eines Pulsradars bei der Beobachtung von zwei Zielen und trennbaren Empfangssignalen.³²

Auf Abbildung 2-5 ist eine Trennung der Empfangssignale nicht möglich, daher ist eine Unterscheidung der beobachteten Objekte ausgeschlossen.

Die Pulsdauer τ_p beeinflusst damit die Trennfähigkeit des Pulsradars. Die Entfernungsauflösung eines Pulsradars berechnet sich nach Gleichung 2-13.

$$\delta R = \frac{c\tau_p}{2} \quad (2-13)$$

³² Göbel, J.: Radartechnik (2011) S.119

³³ Göbel, J.: Radartechnik (2011) S.120

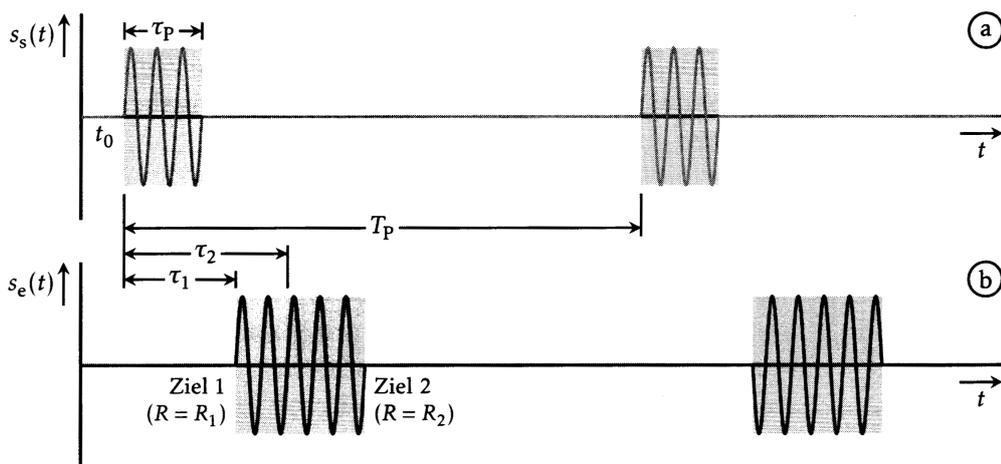


Abbildung 2-5: Sende- (a) und Empfangssignale (b) eines Pulsradars bei der Beobachtung von zwei Zielen und untrennbaren Empfangssignalen.³³

Aus diesem Grund wird eine möglichst geringe Pulsdauer angestrebt. Die Verkürzung der Pulsdauer ist allerdings physikalisch begrenzt³⁴.

Einen weiteren Einfluss auf die Leistungsfähigkeit eines Pulsradars hat die Pulsfolgefrequenz $f_p = 1/T_p$. Wird ein ausgestrahlter Puls von einem Objekt in größerer Entfernung reflektiert und erreicht die Empfangsantenne erst, wenn schon der Folgepuls ausgesendet wurde, so ergibt sich eine Mehrdeutigkeit in der Distanzmessung. Aufgelöst wird diese Mehrdeutigkeit durch Pulse mit verschiedenen Pulsfolgefrequenzen. Ein Beispiel zeigt Abbildung 2-6. Hier werden insgesamt drei verschiedene Pulsfolgefrequenzen verwendet. Durch die Kombination der Einzelergebnisse wird die wahre Distanz des Objekts bestimmt.

Für die Bestimmung der Relativgeschwindigkeit stehen verschiedene Methoden zur Verfügung. Eines dieser Verfahren wird Puls-Doppler genannt^{36,37}. Hierfür wird die maximal messbare Distanz des Sensors in Abschnitte, die Entfernungstore (Range Gates) genannt werden, eingeteilt. Die Pulse des ausgesendeten Signals werden aus einem kontinuierlichen Sinussignal ausgeschnitten und sind damit kohärent. Nach dem Aussenden der Pulse werden die Entfernungstore in der Empfangselektronik nacheinander durchgeschaltet. Jedes Tor deckt eine Distanzauflösungszelle ab. Trifft ein Empfangssignal eines der Tore, so wird im nächsten Schritt eine Dopplerfilterbank durchlaufen. Diese besteht aus Filtern, die jeweils eine Relativgeschwindigkeitsauflösungszelle abdecken. Für die Parametrierung der Filter wird der maximal detektierbare Relativgeschwindigkeitsbereich in Abschnitte eingeteilt. Abbildung 2-7 zeigt die Signalverarbeitung eines Puls-Doppler-Radars mit den Entfernungstoren und Dopplerfilterbanken.

³⁴ Göbel, J.: Radartechnik (2011) S.121

³⁵ Göbel, J.: Radartechnik (2011) S.131

³⁶ Göbel, J.: Radartechnik (2011) S.147ff

³⁷ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) 7-46ff

³⁸ Göbel, J.: Radartechnik (2011) S.148

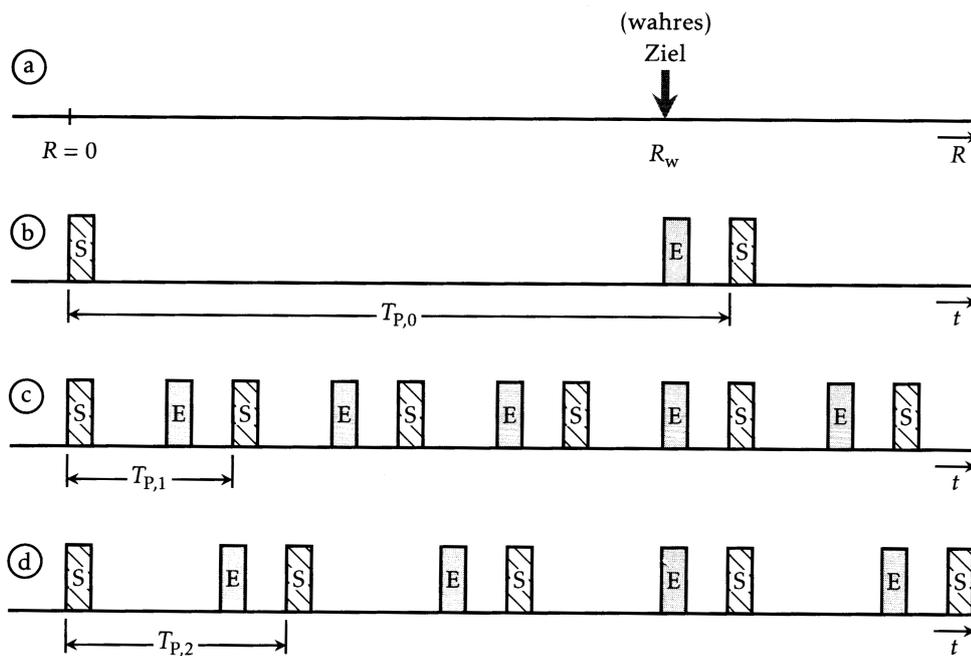


Abbildung 2-6: Auflösung der Distanzmehrdeutigkeit bei Pulsradaren. Zu sehen sind die Sende- und Empfangssignale mit drei verschiedenen Pulsfolgefrequenzen³⁵

Auf diese Weise entsteht eine zweidimensionale Matrix mit Distanz- und Relativgeschwindigkeitszellen. Diese Matrix wird Entfernung-Doppler-Karte genannt und ist auf Abbildung 2-8 dargestellt. Beobachtete Objekte belegen in dieser Karte eine oder mehrere Zellen. Durch die Unterscheidung nach Distanz und Geschwindigkeit wird die Trennung der Objekte anhand beider Kriterien ermöglicht.

³⁹ Göbel, J.: Radartechnik (2011) S.152

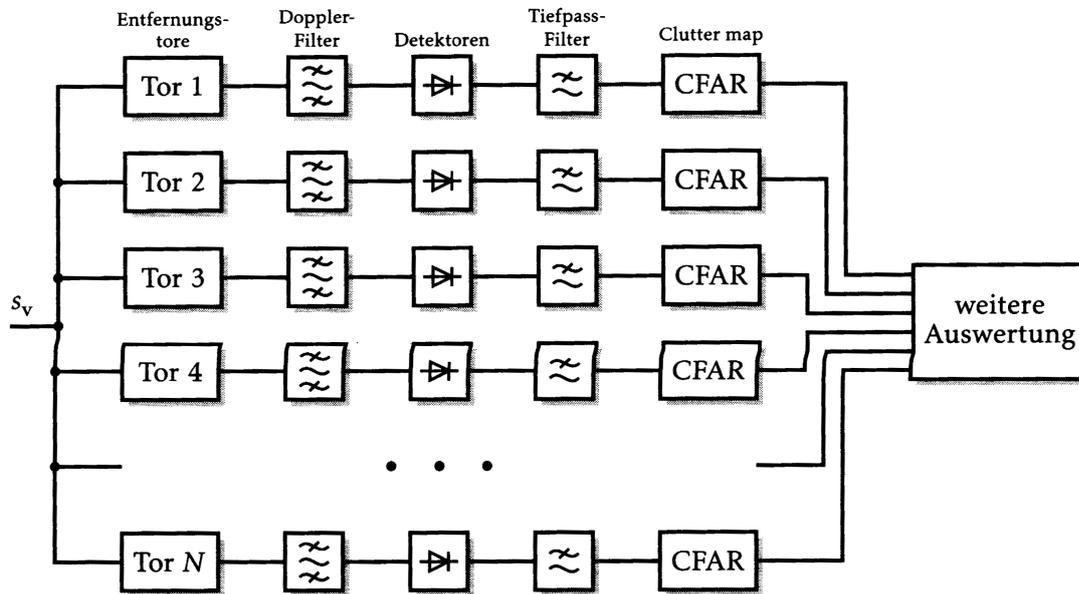


Abbildung 2-7: Darstellung der Radarsignalverarbeitung mit Entfernungstoren und Dopplerfilterbänken eines Puls-Doppler-Radars³⁸

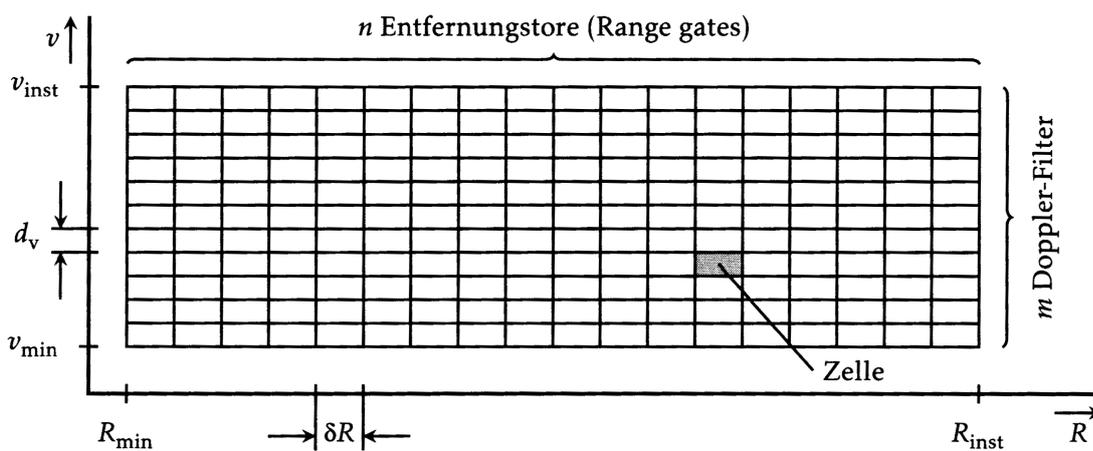


Abbildung 2-8: Darstellung der Entfernungs-Doppler-Karte³⁹

2.2.3 Pulskompressionsradar

Eine Erweiterung der Pulsradars stellt das Prinzip des Pulskompressionsradars^{40,41} dar. Hierzu wird beim Senden die Bandbreite des Sendesignals durch einen linearen Anstieg der Sendefrequenz vergrößert. Mithilfe dieses Tricks wird die Entfernungsauflösung verkleinert, so dass genauere Distanzmessungen möglich werden. Im Empfangsteil findet anschließend die Kompression des Signals statt. Eine dort befindliche Filterbank führt eine zeitliche Komprimierung des empfangenen Pulses durch und verbessert damit Entfernungsauflösung. Abbildung 2-9 zeigt die Sende- und komprimierten Empfangssignale bei Verwendung des Pulskompressionsverfahrens.

⁴⁰ Göbel, J.: Radartechnik (2011) S.158ff

⁴¹ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.8-1ff

⁴² Göbel, J.: Radartechnik (2011) S.163

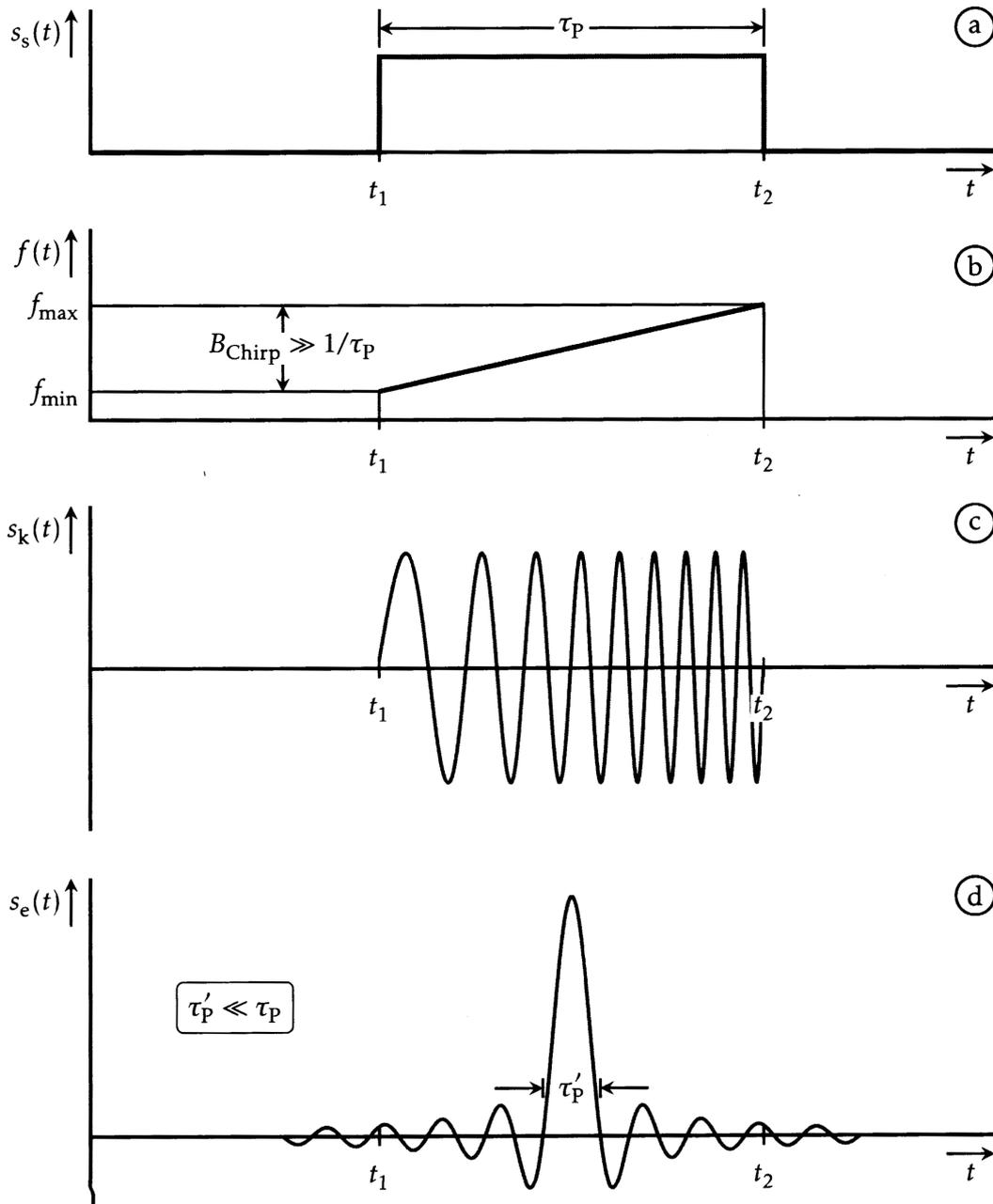


Abbildung 2-9: Sende- und Empfangssignale bei Verwendung des Pulskompressionsverfahrens. (a) ist die Hüllkurve des Sendesignals, (b) ist der Verlauf der Sendefrequenz über der Zeit, (c) ist die Zeitfunktion des Sendesignals und (d) ist das komprimierte Ausgangssignal⁴²

2.3 Radarsignalverarbeitung des Pulskompressionsradars

Dem FZD steht ein Radarsensor zur Verfügung, der das Pulskompressionsverfahren einsetzt. Dieser Sensor wird für die Gewinnung der realen Sensordaten für die Analyse der Tracking-Algorithmen verwendet. In diesem Unterkapitel wird die Signalverarbeitung des Radarsensors beschrieben. Da der Hersteller des Sensors keine detaillierten Informationen über die Struktur der Radarsignalverarbeitung zur Verfügung stellt, orientiert sich das Blockschaltbild auf Abbildung 2-10 an einem gewöhnlichen Aufbau eines Pulskompressionsradars, welches von Ludloff⁴³ beschrieben wird. Die Abbildung zeigt ein vollkohärentes Pulskompressionsradar. Dies bedeutet, dass sowohl zwischen den Sende- als auch zwischen den Empfangssignalen eine feste Phasenbeziehung besteht. Der Stabilized Local Oscillator (STALO) liefert ein hochstabiles Signal, das um 30 Mhz bis 60 Mhz (bzw. 100 MHz bis 200 MHz⁴⁴) unter oder über der Sendefrequenz f_t liegt. Der Coherent Oscillator (COHO) ist ebenfalls ein Oszillator, der allerdings mit einer Frequenz von 30 Mhz bis 60 Mhz (bzw. 100 MHz bis 200 MHz) schwingt und gleichzeitig die Zwischenfrequenz erzeugt. Die Sendefrequenz entsteht durch das Mischen der STALO- und COHO-Signale. Als nächstes wird der Sendepuls im Pulsmodulator aus dem kontinuierlichen Sendesignal ausgeschnitten und anschließend verstärkt. Der verstärkte Sendepuls wird über die Phased-Array-Antennen ausgesendet und die Echos des Signals von den Empfangsantennen detektiert. Werden dynamische Objekte beobachtet, so zeigt sich dies durch Frequenzverschiebungen im Empfangssignal. Die empfangene Signalfrequenz weicht in diesem Fall um f_D von der Sendefrequenz ab. Das Empfangssignal wird durch einen Signalbegrenzer limitiert, um die nachgelagerte Elektronik vor eventuellen Spannungsspitzen zu schützen. Gleichzeitig wird eine zeitabhängige Dämpfung durchgeführt, um Echos von Objekten mit geringer Distanz abzuschwächen. Diese Maßnahme verkleinert die Dynamik des Empfangssignals und vereinfacht somit die nachfolgende Realisierung der Signalverarbeitung⁴⁵. Die Vorverarbeitung des Empfangssignals wird durch eine abschließende Verstärkung abgeschlossen. Im nächsten Schritt wird das empfangene Signal auf die Zwischenfrequenz herunter gemischt. Dabei bleiben Frequenzverschiebungen durch den Doppler-Effekt erhalten. Die resultierende Zwischenfrequenz wird daraufhin im Phasendetektor mithilfe der Referenzphase des COHO in ein In-Phase- (I) und ein Quadrature-Signal (Q) zerlegt. Dieser Schritt ist notwendig, um bei der Digitalisierung des Signals die Phaseninformation nicht zu verlieren. Anschließend werden die beiden Signale in einem AD-Wandler digitalisiert.

Da es sich um ein Pulskompressionsradar handelt, wird im nächsten Schritt eine Pulskompression des digitalisierten Empfangssignals durchgeführt. Für die Detektion der Radialgeschwindigkeiten wird das Signal daraufhin in eine Doppler-Filterbank geleitet. Die weiteren Schritte der Signalverarbeitung geschehen intern und werden vom Hersteller des Sensors weder kommuniziert noch dokumentiert. Über eine Schnittstelle werden die Radarinformationen an den Tracking-Algorithmus übermittelt. Dieser detektiert statische und dynamische Objekte in den übertragenen Informationen und erstellt eine Objektliste.

⁴³ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.5-22ff

⁴⁴ Winner, H. et al.: Handbuch Fahrerassistenzsysteme (2015) S.268

⁴⁵ Göbel, J.: Radartechnik (2011) S.127f

⁴⁶ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.1-6 und S.5-26

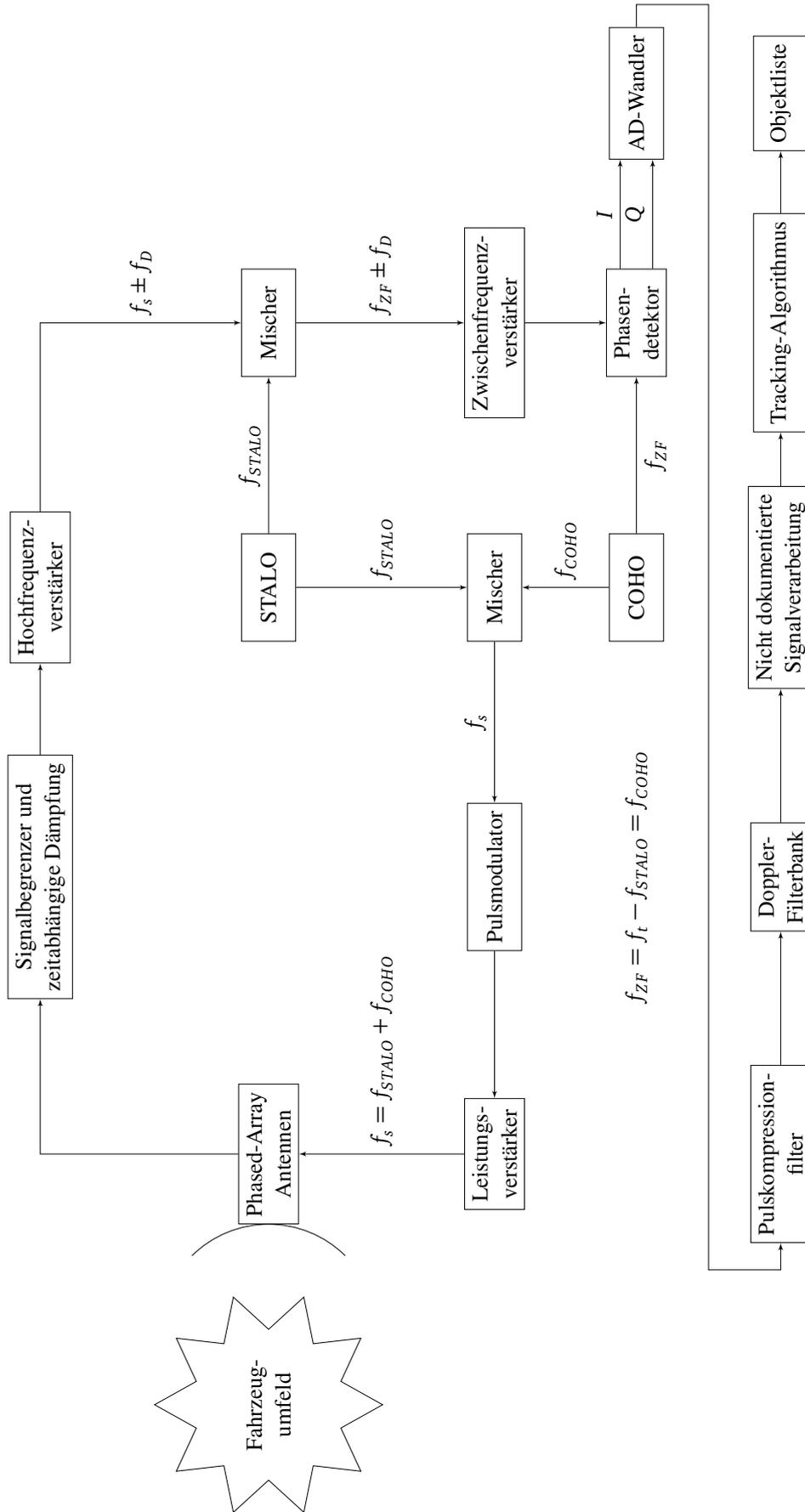


Abbildung 2-10: Blockschaltbild der Radarsignalverarbeitung⁴⁶

2.4 Schnittstelle zwischen Radarsensor und Tracking-Algorithmus

Für die Nutzung der Informationen des Radarsensors durch den Tracking-Algorithmus ist die Definition einer Schnittstelle zwischen beiden Bausteinen erforderlich. Über diese wird das vom Sensor aufbereitete Radarbild an den Tracking-Algorithmus übertragen. Für die Gestaltung der Schnittstelle wird eine Übersicht über benötigte Informationen von Tracking-Algorithmen erstellt. Ein Tracking-Algorithmus benötigt die Distanzen und Relativgeschwindigkeiten aller detektierten Objekt-Reflexionen (Cluster). Anschließend werden die Ausgangssignale des Radarsensors analysiert, um die benötigten Signale zu finden. Es werden verschiedene Ebenen der Signalverarbeitung des Sensors identifiziert, die als Quelle der Signale für den Tracking-Algorithmus geeignet sind. Die Wahl der endgültigen Ebene in der Signalverarbeitung wird in Zusammenarbeit mit dem Hersteller des Sensors getroffen. Wahlkriterien sind das benötigte interne Wissen für die Nutzung der Informationen und der Aufwand für die Aufbereitung der Signale zwischen den Ebenen der Signalverarbeitung.

Über die Schnittstelle werden die folgenden Informationen übertragen:

Anzahl der Reflexionen: Der Radarsensor überträgt die Anzahl der detektierten Reflexionen (Cluster).

Positionen der Reflexionen (Cluster): Die Positionen der Reflexionen (Cluster) relativ zum Sensor werden in kartesischen Koordinaten an den Tracking-Algorithmus übermittelt.

Relativgeschwindigkeiten aller Reflexionen (Cluster): Die Relativgeschwindigkeiten zwischen dem Sensor und den Reflexionen (Cluster) werden in kartesischen Koordinaten übertragen.

Nach Festlegung der Schnittstelle wird diese implementiert.

3 Radar Tracking-Algorithmen

In diesem Kapitel wird eine Auswahl von Algorithmen zur Detektion und Verfolgung von Objekten vorgestellt. Grundsätzlich stehen für die Sensierung des Fahrzeugumfelds verschiedene Sensorprinzipien zur Auswahl. In dieser Thesis werden im speziellen Radarsensoren zur Beobachtung von Objekten eingesetzt.

Im ersten Abschnitt werden die grundlegenden Aufgaben von Tracking-Algorithmen vorgestellt. Die Forschung an Radar Tracking-Algorithmen hat eine große Anzahl von Veröffentlichungen hervorgebracht. Einige der Ansätze werden im zweiten Teil dieses Kapitels präsentiert. Das Kapitel schließt mit der detaillierten Dokumentation der Entwicklung des implementierten Tracking-Algorithmus.

3.1 Einführung

Tracking-Verfahren sind Bausteine von Fahrerassistenzsystemen. Sie dienen zur Aufbereitung der Sensordaten von Umfeldsensoren. Die Sensierung des Fahrzeugumfelds erfolgt durch Kameras, Radarsensoren oder Ultraschallsensoren. Diese erfassen eine Vielzahl von Messwerten, die durch Reflexionen an Umfeldobjekten entstehen. Aufgabe von Radar Tracking-Algorithmen ist die Analyse der Empfangssignale eines Radarsensors und die Generierung einer Liste von Objekten, die sich zum Messzeitpunkt im Fahrzeugumfeld befinden. Für jedes Objekt dieser Liste werden die Zustände (z.B. Distanz und Relativgeschwindigkeit) ermittelt. Aus konsekutiven Messungen des Sensors werden Objekttrajektorien abgeschätzt⁴⁷.

Die Schätzung der Zustände und die Berechnung der Objekttrajektorien basiert auf Messwerten, die ungenau und zusätzlich von Rauschen überlagert sind.

Radarsensoren erzeugen Umfeldinformationen mit einer geringen Auflösung. Aus diesem Grund ist die Unterscheidung von Objekten, die eine geringe Distanz zueinander haben, eine Herausforderung. Sind die Distanzen zu gering, so werden Objekte vereint⁴⁸. In Radarsignalen werden zusätzlich Clutter-Reflexionen beobachtet, die durch Reflexionen an nicht-relevanten Objekten, wie beispielsweise Bodenwellen, entstehen. Auch eine Mehrwegeausbreitung der Sende- und Empfangswellen führt zu Clutter im Radarbild. Ein Tracking-Algorithmus filtert diesen Clutter heraus, so dass nur gewünschte Objektreflexionen verbleiben.

Die Verarbeitung der Empfangssignale erfolgt aufgrund von Messrauschen und -ungenauigkeiten auf Basis der Wahrscheinlichkeitstheorie⁴⁹. Der Satz von Bayes dient zur Berechnung von bedingten Wahrscheinlichkeiten. Zu jedem Messzeitpunkt werden die Zustände der beobachteten Objekte durch den Radarsensor erfasst. Jede konsekutive Messung erhöht den Umfang der Informationen über die

⁴⁷ Blackman, S.; Popoli, R.: Design and analysis of modern tracking systems (1999) S.1

⁴⁸ Challa, S. et al.: Fundamentals of object tracking (2011) S.1

⁴⁹ Challa, S. et al.: Fundamentals of object tracking (2011) S.8-15

Umfeldobjekte. Auf diese Weise werden die Zustände der beobachteten Objekte zu jedem Messzeitpunkt aktualisiert. Wird der Satz von Bayes zweimal angewendet ergibt sich Formel 3-1.

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (3-1)$$

Hierbei ist $p(x|y)$ die bedingte Wahrscheinlichkeit, dass Ereignis x eintritt, wenn zuvor Ereignis y eingetreten ist. $p(y|x)$ ist umgekehrt die bedingte Wahrscheinlichkeit des Ereignisses y , wenn vorher Ereignis x eingetreten ist. $p(x)$ und $p(y)$ sind die Wahrscheinlichkeiten der Ereignisse x und y . Angewandt auf das Objekt-Tracking ist x der gesuchte Zustand eines beobachteten Objekts und y die zum Objekt x gehörenden Messwerte. Mit neuen Messwerten y wird der Zustand x aktualisiert. Für Tracking-Verfahren wird $p(x)$ zu einer Wahrscheinlichkeitsdichtefunktion. Die Berechnung der Aktualisierung von $p(x)$ ist einer der Aufgaben eines Tracking-Algorithmus. Gelöst wird sie durch den Einsatz von Gleichung 3-1. $p(x)$ wird A-Priori (vorher) und $p(x|y)$ A-Posteriori (nachher) genannt. Die nachfolgende Herleitung des Tracking-Algorithmus-Integrals ist aus Challa et al.⁵⁰ entnommen.

Angewandt auf Tracking-Verfahren lautet das Theorem von Bayes

$$p(\mathbf{S}^k|\mathbf{y}^k) = \frac{p(\mathbf{y}^k|\mathbf{S}^k)p(\mathbf{S}^k)}{p(\mathbf{y}^k)}. \quad (3-2)$$

\mathbf{S}^k ist ein generischer Objektzustand, der die Zustände aller beobachteten Objekte enthält. $p(\mathbf{S}^k)$ ist die kombinierte Wahrscheinlichkeitsdichtefunktion von \mathbf{S}^k . $p(\mathbf{S}^k|\mathbf{y}^k)$ ist die A-Posteriori Dichtefunktion der Objekte unter Einbeziehung der neuen Messwerte \mathbf{y}^k zum Zeitpunkt k . $p(\mathbf{y}^k|\mathbf{S}^k)$ ist die sogenannte "Likelihoodfunktion". Diese beschreibt die Wahrscheinlichkeit, dass die Messwerte \mathbf{y}^k detektiert werden, wenn \mathbf{S}^k gilt. Der zweite Term $p(\mathbf{S}^k)$ beschreibt die Dichtefunktion der A-Priori-Wahrscheinlichkeit. Normiert auf Zahlen zwischen 0 und 1 wird anschließend durch Division mit $p(\mathbf{y}^k)$. Diese Schritte werden für jeden konsekutiven Messzeitpunkt wiederholt. Ergebnis ist eine rekursive Schätzung der Objektzustände.

Mit Gleichung 3-2 wird die bedingte Wahrscheinlichkeit von \mathbf{S}^k unter der Berücksichtigung aller vorherigen Zustände \mathbf{S}^k berechnet. In der Realität hängt der Zustand \mathbf{S}^k allerdings nur vom direkten Vorgänger \mathbf{S}^{k-1} ab. Daraus folgt die rekursive Gleichung 3-3.

$$p(\mathbf{S}^k|\mathbf{y}^k) = \frac{p(\mathbf{y}_k|\mathbf{S}_k)}{p(\mathbf{y}_k|\mathbf{y}^{k-1})}p(\mathbf{S}_k|\mathbf{S}_{k-1})p(\mathbf{S}^{k-1}|\mathbf{y}^{k-1}). \quad (3-3)$$

⁵⁰ Challa, S. et al.: Fundamentals of object tracking (2011) S.12ff

Für ein Objekt-Tracking wird die A-Posteriori Wahrscheinlichkeit über eine Integration dieser Wahrscheinlichkeitsdichtefunktion erreicht. Das Objekt-Tracking-Integral ist in Formel 3-4 dargestellt.

$$p(\mathbf{S}_k|\mathbf{y}^k) = \frac{1}{p(\mathbf{y}_k|\mathbf{y}^{k-1})} p(\mathbf{y}_k|\mathbf{S}_k) \int_{\mathbf{S}_{k-1}} p(\mathbf{S}_k|\mathbf{S}_{k-1}) p(\mathbf{S}_{k-1}|\mathbf{y}^{k-1}) d\mathbf{S}_{k-1} \quad (3-4)$$

Für die Lösung dieser Gleichung werden Modell- und Sensorgleichungen aufgestellt. Die Modellgleichung (Zustandsübergangsgleichung) beschreibt die Zustandsänderung zwischen Zeitpunkt $k-1$ und Zeitpunkt k . Da eine Modellbildung nicht exakt ist, wird zusätzlich ein Modellrauschen hinzugefügt. Damit ergibt sich eine Modellgleichung nach Gleichung 3-5.

$$\mathbf{S}_k = \mathbf{g}(\mathbf{S}_{k-1}, \mathbf{v}_k) \quad (3-5)$$

\mathbf{g} ist eine lineare oder nichtlineare Funktion und \mathbf{v}_k stellt das Modellrauschen dar. Die Aufstellung der Sensorgleichung erfolgt analog. Die Funktion \mathbf{l} des Sensors ist linear oder nichtlinear und zusätzlich tritt Sensorrauschen \mathbf{w}_k auf. Daraus folgt Gleichung 3-6 für das Sensormodell.

$$\mathbf{y}_k = \mathbf{l}(\mathbf{S}_k, \mathbf{w}_k) \quad (3-6)$$

Die Berechnung der Wahrscheinlichkeitsdichtefunktion $p(\mathbf{S}^k|\mathbf{y}^k)$ in Gleichung 3-4 erfolgt durch Lösung des Integrals, welches Chapman-Kolmogorov-Gleichung genannt wird. Filter für die Lösung dieser rekursiven Bayes-Gleichung sind beispielsweise der Kalman-Filter, der Extended Kalman-Filter, der Unscented Kalman-Filter oder der Partikelfilter. Die Wahl des Filters hängt von verschiedenen Faktoren ab. Sind die Modell- und Sensorgleichungen 3-5 und 3-6 linear und ihr modelliertes Rauschen gaußförmig, so wird der Kalman-Filter eingesetzt. Sind die Modellgleichungen nichtlinear wird auf den Extended Kalman-Filter oder den Unscented Kalman-Filter ausgewichen. Ist zusätzlich das gewählte Rauschen nicht-gaußförmig, wird der Partikelfilter gewählt.

3.2 Universelle Struktur eines Tracking-Algorithmus

Die universelle Struktur eines Tracking-Algorithmus zeigt Abbildung 3-1.

Die Struktur besteht aus den folgenden Bausteinen:

Sensordatenverarbeitung Bei der Nutzung von Radarsensoren zur Umfeldsensierung werden eine Vielzahl von Messdaten ermittelt. In der Sensordatenverarbeitung werden die vom Sensor ausgegebenen Signale für die Nutzung von Tracking-Algorithmen vorbereitet. Dazu wird eine Schnittstelle gewählt, welche die erforderlichen Signale des Sensors an den Algorithmus

⁵¹ Blackman, S.; Popoli, R.: Design and analysis of modern tracking systems (1999) S.6

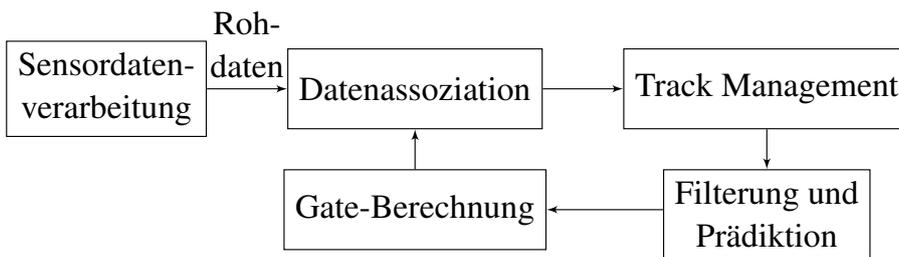


Abbildung 3-1: Universelle Struktur eines Tracking-Algorithmus⁵¹

überträgt. In diesem Schritt besteht weiterhin die Möglichkeit eine Filterung der Radarsignale durchzuführen, um beispielsweise Reflexionen mit schwacher Leistung zu entfernen.

Beim Einsatz eines Radarsensors werden von einem Objekt oftmals mehrere Reflexionen empfangen. In der Sensordatenaufbereitung werden diese Mehrfachreflexionen eines Objekts zusammengefasst, damit nicht jede Reflexion als ein eigenständiges Objekt behandelt wird. Für diesen Zweck werden verschiedene Verfahren eingesetzt. Einer der Ansätze ist ein “Extended Object Tracking“. Hier werden die Reflexionen eines Objekts genutzt, um sowohl die Form, als auch die Position und Bewegungsrichtung abzuschätzen. Ein anderer Ansatz durchsucht alle Reflexionen nach Gemeinsamkeiten, wie ähnliche Position und Geschwindigkeit. Diese Reflexionen werden zu “Clustergruppen“ zusammengefasst und aus ihren Zuständen ein Mittelwert gebildet. Dies vereinfacht die Objektverfolgung⁵².

Datenassoziation Während der Datenassoziatio werden die vorverarbeiteten Messwerte bereits verfolgten Objekten zugeordnet. Für diesen Schritt werden die Distanzen zwischen den vorhergesagten Objektzuständen und den Messwerten berechnet. Liegen diese Distanzen unterhalb einer Schwelle (innerhalb eines Suchbereichs), werden sie dem entsprechenden Objekt zugeordnet. Für die Datenassoziatio werden verschiedene Verfahren eingesetzt. Einer dieser Ansätze beruht auf den Berechnungen der Zugehörigkeitswahrscheinlichkeiten einzelner Reflexionen zum Objekt des betreffenden Suchbereichs. In Multiple-Target-Szenarien (mehrere Zielobjekte) wird zusätzlich sichergestellt, dass eine Reflexion nicht mehreren Objekten zugeordnet wird. Ein weiteres Verfahren wählt die Reflexion mit der geringsten Distanz zur prädierten Objektposition aus.

Track Management Im Track Management wird die Objektliste verwaltet. Alle Reflexionen, die während der Datenassoziatio keinem Objekt zugeordnet wurden, werden als potentiell neue Objekte betrachtet. Umgekehrt werden Objekte gelöscht, denen im vorherigen Schritt keine neuen Reflexionen zugeordnet wurden. Die Status aller weiteren Objekte der Liste werden bestätigt. Da nicht-zugeordnete Reflexionen auch durch Clutter entstehen, werden nicht alle Reflexionen unmittelbar als neue Objekte behandelt. Stattdessen wird das m/N-Verfahren eingesetzt. Reflexionen werden erst als Objekte eingeordnet, wenn sie in N Messungen m Reflexionen erzeugt haben. Die Erkennung von neuen Objekten ist damit verzögert. Umgekehrt werden die Objekte

⁵² Blackman, S.; Popoli, R.: Design and analysis of modern tracking systems (1999) S.6f

nicht direkt gelöscht, wenn einem Objekt keine neue Reflexion zugeordnet wurde, sondern erst wenn sie in M Messungen nicht erneut detektiert wurden.

Filterung und Prädiktion In der Filterung und Prädiktion werden die Zustände der Objekte mithilfe der zugeordneten Messwerte aktualisiert. Dazu wird beispielsweise der Kalman-Filter eingesetzt. Dieser korrigiert die prädizierten Objektzustände mithilfe der neuen Messwerte.

Im Teil der Prädiktion werden ausgehend von den aktuellen Objektzuständen die Zustände der Objekte zum nächsten Messzeitpunkt prädiziert.

Gate-Berechnung Nach der Prädiktion der Objektzustände werden in der Gate-Berechnung Suchbereiche (Gates) um die prädizierten Zustände bestimmt. Ein Gate beschreibt einen Bereich um prädizierte Zustände, in den, mit einer einstellbaren Wahrscheinlichkeit, die nächsten Messwerte dieses Objekts fallen werden. Dabei wird sowohl die Genauigkeit des Sensors, als auch des Bewegungsmodells einbezogen. Die Form des Gates wird durch die Mahalanobis-Distanz vorgegeben, welche die Größe auf Basis der χ^2 -Verteilung bestimmt⁵³.

3.3 Vorstellung verschiedener Tracking-Algorithmen

Jeder Block auf Abbildung 3-1 ist individuell gestaltbar. Aus diesem Grund werden im folgenden die Realisierungsmöglichkeiten der einzelnen Bausteine separat vorgestellt. Für die Implementierung eines Tracking-Algorithmus ist eine beliebige Kombination der Bausteine denkbar.

In einer Veröffentlichung von Granström et al.⁵⁴ wird eine umfangreiche Übersicht über Extended Object Tracking-Algorithmen vorgestellt. Die Autoren sprechen eine Empfehlung für zu wählende Zustände der Objekte aus, zeigen Möglichkeiten die Mehrfachreflexionen ausgedehnter Objekte zu vereinen und stellen verschiedene Verfahren zur Filterung der Radarsignale vor. Sie verweisen in ihrer Veröffentlichung auf eine Vielzahl von Publikationen, die das Thema Radar Tracking-Algorithmen behandeln.

Eine weitere umfangreiche Übersicht über alle Aspekte von Tracking-Algorithmen stammt von Pullford⁵⁵. Auch diese Veröffentlichung verweist auf eine Vielzahl weiterer Veröffentlichungen und bietet sich damit als Literaturquelle an.

3.3.1 Sensordatenverarbeitung

Der Block "Sensordatenverarbeitung" befasst sich mit der Verarbeitung der Rohsignale des Radarsensors.

Filterung der Radarsignale

Der Radarsensor empfängt die Reflexionen seiner ausgestrahlten Radarstrahlen. Mithilfe dieser Reflexionen werden Informationen über Winkel, Abstand und Geschwindigkeit von beobachteten Ob-

⁵³ Grisetti, G. et al.: Robotics 2 - Data Association (2009) S.7

⁵⁴ Granstrom, K., Baum, M.; Reuter, S.: Extended Object Tracking (2016)

⁵⁵ Pulford, G. W.: A Survey of Manoeuvring Target Tracking Methods (2015)

jekten bestimmt. Zusätzlich wird die Empfangsleistung gemessen, die von einigen Algorithmen zur Filterung der Signale verwendet wird. Dabei wird davon ausgegangen, dass schwache Signale durch Clutterreflexionen entstehen. Für die Filterung wird eine Schwelle festgelegt und entsprechende Reflexionen gelöscht, wenn deren Leistungen unterhalb dieser Schwelle liegen⁵⁶.

Zu Beachten ist, dass die Empfangsleistung bei Radarsensoren auch vom Radarquerschnitt des beobachteten Objekts abhängt. Dieser Querschnitt ist wiederum stark abhängig vom Auftreffwinkel des Radarstrahls auf das Objekt. Variationen um $0,2^\circ$ führen zu einer Schwankung des Rückstrahlquerschnitts von bis zu 20 dB⁵⁷. Am Fachgebiet Fahrzeugtechnik der TU Darmstadt wird der Einfluss des Aspektwinkels auf die Empfangsleistung eines Radarstrahls untersucht⁵⁸.

Ein anderer Ansatz zur Gestaltung dieses Blocks ist der Verzicht auf eine Filterung.

Aufbereitung der Radarsignale

In diesem Teil der Sensordatenverarbeitung werden die Radarsignale über eine Schnittstelle zum Tracking-Algorithmus transferiert. Dafür wird festgelegt, welche Informationen vom Algorithmus benötigt werden. Diese Informationen werden zu Zustandsvektoren zusammengefasst. Inhalt dieser Vektoren sind die Positionen und Geschwindigkeiten der betreffenden Reflexion in kartesischen oder polaren Koordinaten. Je nach Tracking-Algorithmus werden zusätzlich auch Informationen wie die Radarrückstrahlfläche, der Signal-zu-Rausch-Abstand und die Form des Objekts aufgenommen^{59,60}.

Mehrfachreflexionen

Typischerweise werden bei der Beobachtung von Objekten mehr als nur eine Reflexion empfangen. Damit nicht jede Reflexion als ein eigenständiges Objekt behandelt wird, werden diese Mehrfachreflexionen zusammengefasst. Hierfür stehen beispielsweise Algorithmen für ein Extended Object Tracking zur Verfügung.

Ein solcher Ansatz wird von Broßeit et al.⁶¹ beschrieben. In ihrer Veröffentlichung wird ein probabilistischer Weg aufgezeigt, der eine Bounding-Box über die Position des beobachteten Objekts legt. Durch diese Idee wird die Form des Objekts mithilfe der Mehrfachreflexionen ermittelt und durch konsekutive Messungen aktualisiert. In die Berechnung der Bounding-Box fließen die bekannten Sensorungenauigkeiten des Radarsensors ein. Bei Anwendung ihres Algorithmus auf real erhobene Sensordaten haben sich für die Position ein Root-mean-square error (RMSE) von 0,1 m und eine Orientierungsgenauigkeit von $1,7^\circ$ eingestellt.

Ein weiterer Extended Object Tracking-Algorithmus wird von Kellner et al.⁶² präsentiert. Als Sensor wird ein High-Resolution-Radar verwendet. Dieser erzeugt eine Vielzahl von Reflexionen an einem

⁵⁶ Blackman, S.; Popoli, R.: Design and analysis of modern tracking systems (1999) S.6f

⁵⁷ Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008) S.3-15

⁵⁸ Marx, B. J.: Radareigenschaften von Personenkraftwagenkarosserien (2014)

⁵⁹ Blackman, S.; Popoli, R.: Design and analysis of modern tracking systems (1999) S.7

⁶⁰ Granstrom, K., Baum, M.; Reuter, S.: Extended Object Tracking (2016) S.3f

⁶¹ Brosseit, P. et al.: Probabilistic rectangular-shape estimation for extended object tracking (2016)

⁶² Kellner, D. et al.: Tracking of Extended Objects with High-Resolution Doppler Radar (2016)

beobachteten Objekt. Durch die Verwendung dieses Radarsensors stehen zusätzlich zur Distanz und dem Azimutwinkel des Objekts auch dessen Geschwindigkeit und Orientierungsrichtung zur Verfügung. Diese werden verwendet, um die Fahrtrichtung und die Gierrate des beobachteten Objekts zu berechnen. Dies ermöglicht die hochdynamische Verfolgung von Objekten.

Hammarstrand et al.⁶³ beschreiben einen Extended Object Tracking Algorithmus auf Basis von Untersuchungen der Reflexionszentren von Fahrzeugen. Radarreflexionen entstehen beispielsweise im Bereich der Radkästen und am Heck von Fahrzeugen. Die Autoren unterscheiden dabei Punkt- und Flächenreflektoren. Eine Darstellung dieser Reflexionszentren zeigt Abbildung 3-2.

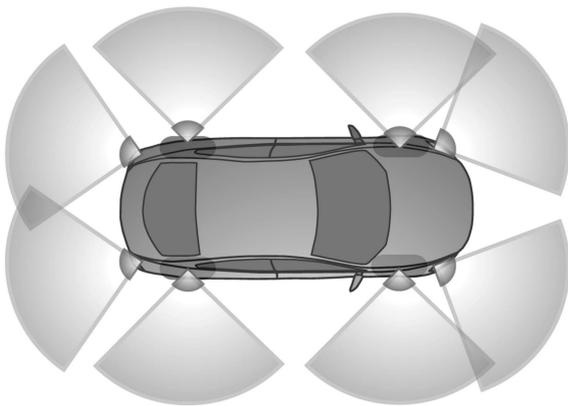


Abbildung 3-2: Reflexionszentren einer Personenkraftwagenkarosserie⁶⁴

Alle obigen Ansätze setzen eine ausreichende Anzahl von Reflexionen eines Objekts voraus. Stehen nicht genügend Reflexionen für die Formerkennung zur Verfügung wird auf ein anderes Verfahren zurück gegriffen. Hierbei werden alle Reflexionen gesucht, die eine ähnliche Position und Geschwindigkeit aufweisen und diese zu einer Clustergruppe zusammengefasst. Jede Gruppe repräsentiert dabei ein Objekt. Durch Bildung der Mittelwerte dieser Clustergruppen wird ein neuer Zustand erzeugt, der im nachgelagerten Teilen des Algorithmus verwendet wird. Durch diesen Schritt verringert sich die Anzahl der zu verarbeitenden Reflexionen.

3.3.2 Datenassoziation

Im Block "Datenassoziation" werden die erstellten Clustergruppen den bekannten Objekten der Objektliste zugeordnet. Abbildung 3-3 zeigt ein mögliches Szenario. Hier werden eine prädierte Objektposition \hat{Z}^1 und drei mögliche Sensorwerte (Z_1, Z_2, Z_3) dargestellt. Die Datenassoziation ordnet einen der drei Messwerte der prädierten Position zu.

Aufwendiger wird die Auswahl des Sensorwerts bei der Beobachtung mehrerer Objekte. Abbildung 3-4 zeigt ein solches Szenario. Wird hier Z_2 zu \hat{Z}^2 zugeordnet, ist darauf zu achten, dass Z_1 zu \hat{Z}^1 zugeordnet wird.

⁶³ Hammarstrand, L. et al.: Extended Object Tracking using a Radar Resolution Model (2012)

⁶⁴ Hammarstrand, L. et al.: Extended Object Tracking using a Radar Resolution Model (2012) S.2375

⁶⁵ Bar-Shalom, Y., Daum, F.; Huang, J.: The probabilistic data association filter (2009) S.83

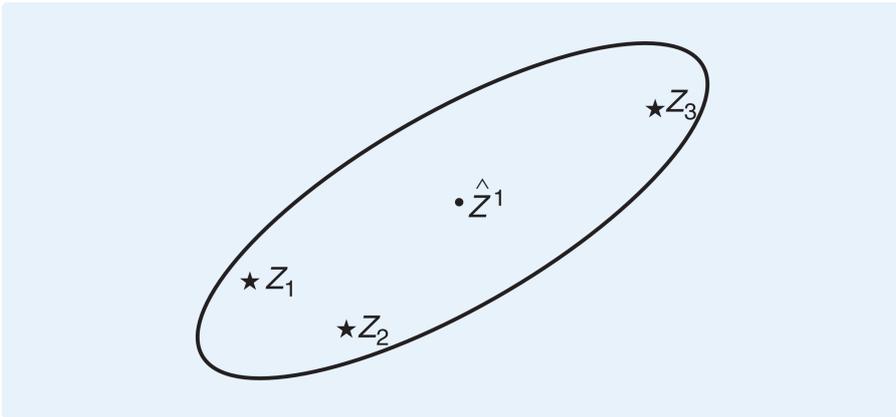


Abbildung 3-3: Datenassoziation mit einem Objekt und drei Messwerten⁶⁵

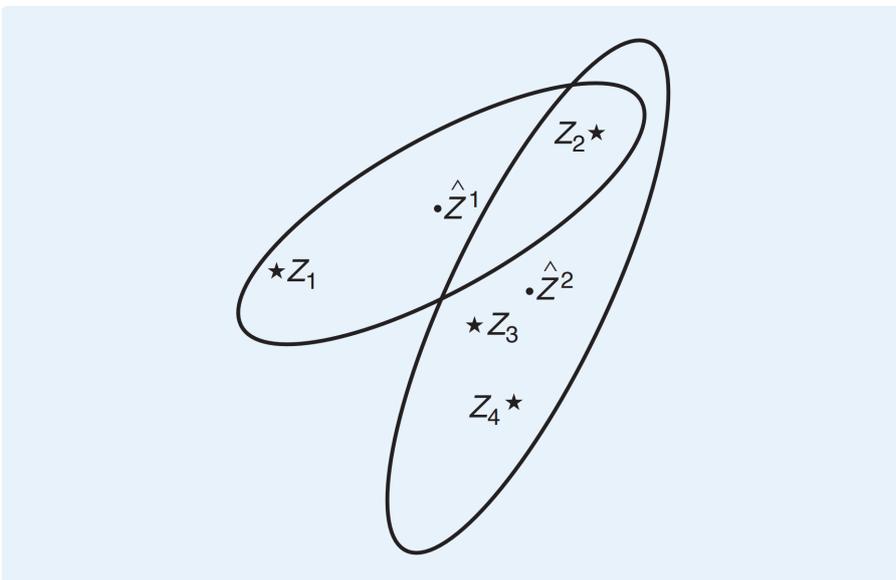


Abbildung 3-4: Datenassoziation mit zwei Objekten und vier Messwerten⁶⁶

Für die Gestaltung der Datenassoziation stehen verschiedene Ansätze zur Verfügung. Beim Verfahren “Nearest Neighbor“ wird derjenige Messwert ausgewählt, der die geringste Distanz zur prädizierten Objektposition aufweist. Der Berechnungsaufwand ist gering, allerdings besteht die Gefahr einen falschen Sensorwert auszuwählen, beispielsweise bei sich kreuzenden Objekten.

Einen aufwendigeren Ansatz beschreiben Bar-Shalom et al.⁶⁷. Der von ihnen entwickelte Probabilistic data association filter (PDAF) ist wahrscheinkeitsbasiert. Hier wird derjenige Messwert gewählt, der die größte Zugehörigkeitswahrscheinlichkeit aufweist. Unter Einbezug der zugrundeliegenden Modell- und Sensorungenauigkeiten werden die Datenassoziationswahrscheinlichkeiten eines jeden Messwerts berechnet. Der PDAF wird eingesetzt, wenn die folgenden Bedingungen erfüllt sind:

⁶⁶ Bar-Shalom, Y., Daum, F.; Huang, J.: The probabilistic data association filter (2009) S.84

⁶⁷ Bar-Shalom, Y., Daum, F.; Huang, J.: The probabilistic data association filter (2009)

-
1. In jedem Suchbereich befindet sich nur ein Objekt. Die Bewegungsgleichungen des Objekts sind linear und das angenommene Rauschen des Modells und Sensors ist gaußförmig.
 2. Das Objekt wurde bereits in einer vorherigen Algorithmen-Iteration entdeckt und die Verfolgung eingeleitet.
 3. Für jeden neuen Messzeitpunkt wird der Suchbereich des Objekts aktualisiert.
 4. Jedes beobachtete Objekt erzeugt nur einen einzelnen Sensormesswert.
 5. Alle weiteren Sensormesswerte werden als Fehlmessungen bzw. Clutter angenommen.

Bei der Verfolgung mehrerer Objekte wird der Einsatz des Joint probabilistic data association filter (JPDAF) empfohlen. Dieser berechnet die Zugehörigkeitswahrscheinlichkeiten der Messwerte global für die gesamte beobachtete Szene. Eine solche Szene wird in Abbildung 3-4 gezeigt. Die Verwendung ist an folgende Bedingungen geknüpft:

1. Die Zahl der beobachteten Objekte ist bekannt.
2. Messwerte eines Objekts können in dem Suchbereich eines anderen Objekts landen.
3. Die Bewegungsgleichungen sind linear und es wird gaußförmiges Rauschen angenommen.

Typischerweise ist die Anzahl der zu beobachtenden Objekte nicht bekannt, somit ist der JPDAF nicht in allen Fällen anwendbar.

Ein weiteres Verfahren für die Datenassoziation ist der Multiple Hypothesis Tracking filter (MHT)⁶⁸. Bei Anwendung dieses Verfahrens wird keine sofortige Entscheidung der Datenassoziation getroffen, sondern abgewartet, bis weitere Informationen durch zukünftige Messungen zur Verfügung stehen. Die Zuordnung erfolgt damit durch Einbezug von Messzeitpunkten aus der Vergangenheit. Dazu werden zu jedem Messzeitpunkt Zugehörigkeitshypothesen aufgestellt und konsekutiv aktualisiert. Der Einsatz der Hypothesen ermöglicht die Einordnung der Messwerte in verschiedene Gruppen, wie Clutter, bestätigte Objekte oder neue Objekte. Durch diesen Ansatz wird die Verfolgung einer unbekannt Anzahl von Objekten ermöglicht. Problem bei der Verwendung dieses Verfahrens ist der große Rechenaufwand für die Hypothesenbestimmung. Durch eine Vielzahl von Objekten und Messzeitpunkten entsteht eine exponentiell anwachsende Anzahl von Hypothesen, die überprüft werden müssen. Gelöst wird dieses Problem durch den Einsatz von Suchbereichen, um die prädierten Objektpositionen und über Versuche verschiedene Hypothesen zu vereinen⁶⁹.

Ein ganz anderer Ansatz basiert auf Random Finite Set (RFS). Dieser erlaubt die Verwendung der bedingten Wahrscheinlichkeitsrechnung von Bayes, umgeht allerdings das Problem der Datenassoziation. Ursprünglich vorgestellt hat diese Methode Mahler⁷⁰. In den letzten Jahren ist dieser Ansatz weiter verbessert worden. Eine aktuelle Veröffentlichung wird von Sharif⁷¹ präsentiert. Für den RFS-Filter wird die gesamte beobachtete Situation als ein individueller Zustand aufgefasst. Ebenso bilden

⁶⁸ Challa, S. et al.: Fundamentals of object tracking (2011) S.228ff

⁶⁹ Sharif, U.: Multi-Target Tracking using Random Finite Set based Bayesian Filtering in a Heterogeneous Platform (2016)

⁷⁰ Mahler, R.: Statistics 101 for multisensor, multitarget data fusion (2004)

⁷¹ Sharif, U.: Multi-Target Tracking using Random Finite Set based Bayesian Filtering in a Heterogeneous Platform (2016)

die Messwerte einen eigenen Zustand. Die Zusammenführung beider Zustände erfolgt mathematisch durch die Verwendung der Finite Set Statistic (FISST). Ein Lösungsverfahren für den Einsatz in der Filterung und Prädiktion von RFS Tracking-Algorithmen ist beispielsweise der Probability Hypothesis Density filter (PHD)⁷². Eine weitere ausführliche Vorstellung von RFS für die Objektverfolgung stammt von Reuter⁷³.

Eine letzte hier vorgestellte Möglichkeit für die Gestaltung der Datenassoziation ist die Mittelwertbildung aus allen Messwerten im Suchbereich. Angewendet auf das Szenario in Abbildung 3-3 bedeutet dies, dass der Mittelwert der Reflexionen Z_1 , Z_2 , und Z_3 gebildet wird.

3.3.3 Track-Management

Im Track-Management werden die Verwaltungsaufgaben des Tracking-Algorithmus durchgeführt. Dazu wird eine Liste mit bestätigten und potentiellen Objekten geführt und fortlaufend aktualisiert. Das Track-Management hat die Aufgabe potentielle Objekte mithilfe des m/N-Verfahrens zu überprüfen, um ihre Status zu bestätigen oder sie zu löschen.

Eine weitere Möglichkeit für die Verwaltung von Objekten wird von Hyun et al.⁷⁴ vorgestellt. Die Autoren stellen ein Track-Management auf Basis eines Zustandsautomaten vor. Die Zustände des Zustandsautomaten sehen vier möglichen Phasen (Ruhezustand, Initialisierungsphase, bestätigtes Objekt und Löschung des Objekts) von beobachteten Objekten vor. Abbildung 3-5 zeigt ein Beispiel einer Objektverfolgungssituation.

3.3.4 Filterung und Prädiktion

Während der Filterung und Prädiktion werden die Zustände der verfolgten Objekte geschätzt. Für die Lösung dieses Tracking-Problems stehen eine Vielzahl von Verfahren zur Verfügung. Im folgenden werden verschiedene Ansätze vorgestellt.

Prädiktion

Für die Prädiktion der Objektzustände werden die aktuellen Zustände in die Bewegungsmodelle eingesetzt und damit die Zustände zum nächsten Messzeitpunkt vorhergesagt. Für die Verfolgung von Fahrzeugen stehen verschiedene Bewegungsmodelle zur Verfügung⁷⁶. Für ein Bewegungsmodell wird beispielsweise davon ausgegangen, dass sich die Objekte mit einer konstanten Geschwindigkeit bewegen (Constant Velocity (CV)-Modell). Weitere Modelle gehen von konstanter Beschleunigung (Constant Acceleration (CA)-Modell) oder konstantem Kurvenradius (Constant Turn Rate (CTR)-Modell) aus.

Alle diese Modelle beschreiben jeweils nur eine Bewegungsart der Objekte. Ein Fahrzeug im Stra-

⁷² Mahler, R.: Multitarget bayes filtering via first-order multitarget moments (2003)

⁷³ Reuter, S.: Multi-object tracking using random finite sets (2014)

⁷⁴ Hyun, E.; Lee, J.-H.: Multi-target tracking scheme using a track management table (2016)

⁷⁵ Hyun, E.; Lee, J.-H.: Multi-target tracking scheme using a track management table (2016) S.2

⁷⁶ Grisetti, G. et al.: Robotics 2 Target Tracking (2009) S.39

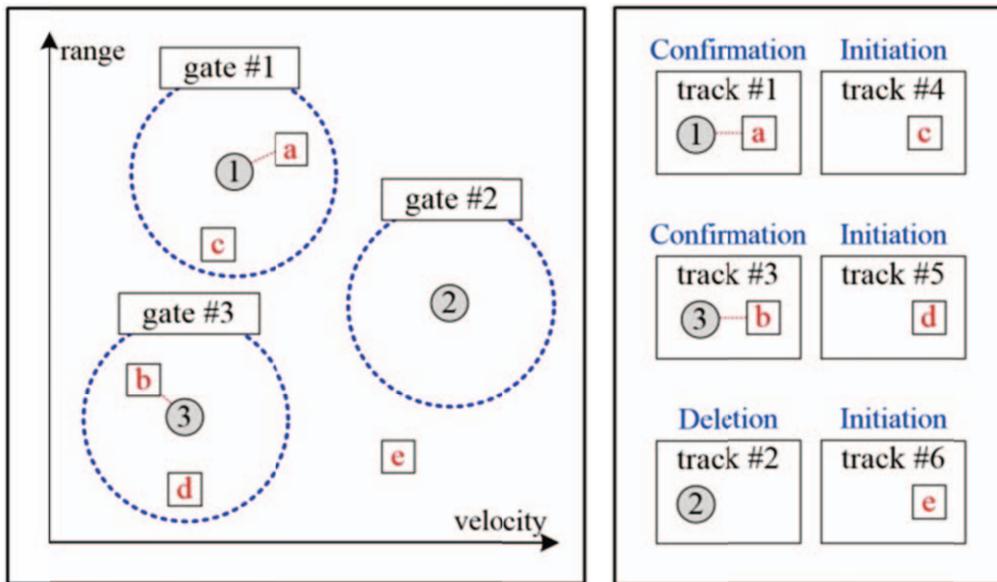


Abbildung 3-5: Beispiel eines erweiterten Track Managements. Zu sehen sind Objekte in drei verschiedenen Phasen. Bestätigt sind die Objekte 1 und 3. In der Löschphase befindet sich Objekt 2. Die Messungen c, d und e zeigen mögliche neue Objekte an und sind daher in der Initialisierungsphase.⁷⁵

ßenverkehr führt allerdings alle drei Bewegungsarten durch. Aus diesem Grund werden mehrere Bewegungsmodelle erstellt und je nach Fahrzeugmanöver zwischen den Modellen umgeschaltet. Ein solcher Ansatz ist das Interacting Multiple Model (IMM)⁷⁷. Die Parametrierung der Modelle kann entweder offline oder online während des Tracking-Verlaufs erfolgen. Die Übergänge von einem Modell auf ein anderes geschieht durch Markov-Ketten. Mithilfe dieser Ketten werden Wahrscheinlichkeiten für die Modellübergänge des IMM bestimmt. Ausgegeben wird schlussendlich eine gewichtete Kombination aller Wahrscheinlichkeiten der Modelle. Es wird damit keine harte Entscheidung zwischen den Modellen getroffen. Weiterentwicklungen des ursprünglichen IMM werden von Kim et al.⁷⁸ beschrieben.

Zusätzliche Verfahren, die mit mehreren parallelen Bewegungsmodellen arbeiten, sind in einer Übersicht von Gomes⁷⁹ zusammengefasst.

Die Autoren Bühren und Yang⁸⁰ stellen ein weiteres globales Bewegungsmodell für Tracking-Verfahren mit automobilen Radarsensoren vor. Anstatt die Bewegungen der Zielobjekte in beiden Koordinatenrichtungen unabhängig voneinander zu modellieren, schlagen sie ein Modell vor, bei denen die Bewegungen miteinander verknüpft sind. Ihr vorgeschlagener Zustandsvektor enthält zusätzlich zur Position und Geschwindigkeit auch die Beschleunigung, den Kurswinkel und den Lenkradwinkel des Objekts. Mit diesen Informationen wird ein Einspurmodell gespeist. Nachteil dieses

⁷⁷ P. Blom, H.: An efficient filter for abruptly changing systems (1984)

⁷⁸ Kim, Y.-S.; Hong, K.-S.: An IMM algorithm for tracking maneuvering vehicles (2004)

⁷⁹ Gomes, J. B. B.: An overview on target tracking using multiple model methods (2008)

⁸⁰ Bühren, M.; Yang, B.: A Global Motion Model (2007)

Modells ist die entstehende Nichtlinearität und der damit einhergehenden Wahl einer komplexeren Filterungsmethode. Zusätzlich wird für das Einspurmodell der Radstand und der Lenkwinkel für die Berechnungen benötigt. Die Autoren schlagen einen festen Wert für den Radstand vor. Die Erkennung des Lenkradwinkels ist allerdings ein noch zu lösendes Problem.

Filterung

Ein bekannter Filter, der bei linearen Systemen und gaußförmigem Messrauschen verwendet wird, ist der Kalman-Filter⁸¹. Dieser Filter wird in einer Vielzahl von Tracking-Algorithmen verwendet. In der Forschung werden allerdings auch weitere Verfahren für die Filterung der fehlerbehafteten Radarsignale verwendet.

Einer dieser Filter ist der sogenannte Smooth Variable Structure filter (SVSF). Nach einer Veröffentlichung von Attari⁸² liefert dieser Filter insbesondere bei großen Modellungenauigkeiten ein zuverlässiges Tracking-Ergebnis. Bei gleicher Modellungenauigkeit und der Verwendung von Kalman-Filtern werden deutlich schlechtere Tracking-Ergebnisse erzielt. Bei der Verfolgung von Objekten entstehen durch die variablen Geschwindigkeiten dieser Objekte Abweichungen zwischen der Realität und den Modellberechnungen. Die in diesem Fall auftretende Modellungenauigkeit wird durch den SVSF korrigiert.

Eine weitere Möglichkeit für die Filterung der Radarsignale ist die Verwendung eines Partikelfilters. Ein Einsatz dieses Filters erlaubt die Verwendung nichtlinearer Bewegungsmodelle und beliebig geformtes Rauschen in den Messsignalen. Diese Art von Filter ist nicht auf Linearisierungen der Modell- und Sensorgleichungen angewiesen, wie dies beim Extended Kalman-Filter und Unscented Kalman-Filter der Fall ist. Eine Veröffentlichung von Yu et al.⁸³ stellt einen Partikelfilter vor, der bei einer, von den Autoren durchgeführten Analyse, eine genauere Objektverfolgung ermöglicht, als andere untersuchte Filter. Für die Generierung der Partikel verwenden Yu et al. einen Extended Kalman Filter (EKF) und bestimmen damit die prädierten Objektpositionen. Die Partikel werden anschließend in Bereichen um diese Positionen ausgewählt. Eine Übersicht über die Möglichkeiten für die Verwendung von Partikelfiltern in Objektverfolgungsanwendungen bieten Ristic et al.⁸⁴.

Erfolgt das Objekt-Tracking mithilfe von RFS, werden andere Arten von Filtern zur Zustandsschätzung eingesetzt. Einer dieser Filter ist der PHD. Dieser Filter wird erstmals von Mahler⁸⁵ vorgestellt und basiert auf der rekursiven Berechnung der Wahrscheinlichkeitsdichten der beobachteten Objektsituationen. Vorteil dieses Filters ist der vergleichsweise geringe Rechenaufwand, da die Wahrscheinlichkeiten nicht komplett neu berechnet werden, sondern vorhandene Informationen iterativ aktualisiert werden.

Panta et al.⁸⁶ erweitern obigen PHD-Filter hin zu einer geschlossenen Lösung. Ihr Ansatz ist der

⁸¹ Kalman, R. E.: A New Approach to Linear Filtering and Prediction Problems (1960)

⁸² Attari, M.: SVSF Estimation for Target Tracking with Measurement Origin Uncertainty (2016)

⁸³ Yu, M. et al.: An enhanced particle filtering method for GMTI radar tracking (2016)

⁸⁴ Ristic, B., Arulampalam, S.; Gordon, N.: Beyond the Kalman filter (2004)

⁸⁵ Mahler, R.: Multitarget bayes filtering via first-order multitarget moments (2003)

⁸⁶ Panta, K., Clark, D. E.; Vo, B.-N.: Data Association and Track Management for the GMPHD (2009)

Gaussian Mixture - Probability Hypothesis Density filter (GM-PHD). Dieser basiert ebenfalls auf Mengen bekannter Objekte und Messwerte und ermöglicht die Verfolgung einer unbekannt Anzahl von Objekten in Clutter.

Weitere Objekt-Tracking-Filter gehen aus einer umfangreichen Übersicht von Vo et al.⁸⁷ hervor.

3.3.5 Gate-Berechnung

Die Größe und Form der Suchbereiche um prädizierte Objektpositionen wird auf Basis der Mahalanobis-Distanz berechnet. Bei Tracking-Algorithmen handelt es sich dabei um Ellipsen. Alle Punkte auf dem Rand einer Ellipse haben bei der Mahalanobis-Distanz die gleiche Auftretenswahrscheinlichkeit. Die Größe der Ellipse wird in Abhängigkeit eines Wahrscheinlichkeitsparameters bestimmt. Die Größe der Ellipse wird so gewählt, dass bei gegebenen Sensor- und Modellungenauigkeiten in (z.B.) 99 % aller Fälle die nächsten Messwerte innerhalb dieses Bereichs fallen werden.

Bei Filtern, die auf Basis von RFS arbeiten, ist die Berechnung eines Gates nicht notwendig, da die Objekte und Messwerte als zwei getrennte Gesamtzustände betrachtet werden.

3.4 Implementierung des Radar Tracking-Algorithmus

Der implementierte Tracking-Algorithmus basiert auf linearen Annahmen über die Sensor- und Modellgleichungen und setzt gaußförmiges Rauschen voraus. Abbildung 3-6 zeigt das Blockschaltbild des Tracking-Algorithmus.

Die Funktionen aller Blöcke innerhalb des gestrichelten Bereichs werden in einer Schleife wiederholt. Die Blöcke außerhalb werden einmalig aufgerufen. Innerhalb der Tracking-Algorithmus-Schleife wird zwischen der Verwendung von realen und simulierten Radardaten unterschieden. Werden die simulierten Radarsignale eingesetzt, so ist die Bewertung des Tracking-Ergebnisses nach Beendigung des Tracking-Durchlaufs möglich.

Die Implementierung der einzelnen Blöcke wird in den folgenden Abschnitten vorgestellt. Der Algorithmus wird in MathWorks MATLAB Version 2016b erstellt. Der vorgestellte Algorithmus verwendet Programmcode, der über den "MATLAB File Exchange"⁸⁸ öffentlich zur Verfügung steht. Teile der folgenden Programmcodes werden verwendet:

- Distribute figures⁸⁹
- gaussian prob⁹⁰
- matlab2tikz⁹¹
- OSPA-T tracking algorithm metric⁹²

⁸⁷ Vo, B.-.-.-n. et al.: Multitarget Tracking (2015)

⁸⁸ MATLAB: File Exchange (2017)

⁸⁹ Simonsen, A.: Distribute figures (2015)

⁹⁰ Autor unbekannt: gaussian prob (2017)

⁹¹ Schlömer, N.: matlab2tikz (2017)

⁹² Ristic, B., Sherrah, J.; García-Fernández, Á. F.: OSPA-T Quellcode (2012)

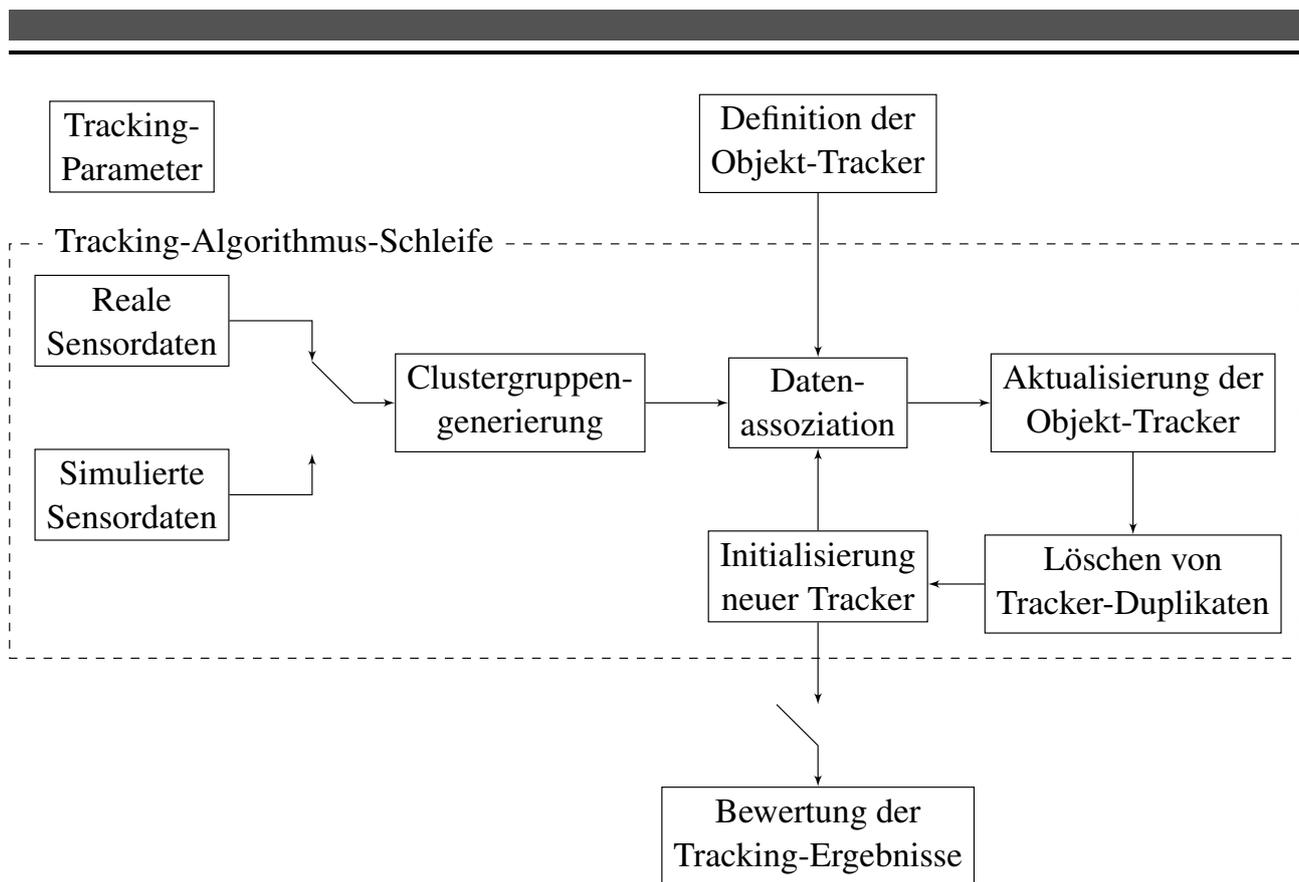


Abbildung 3-6: Blockschaltbild des implementierten Tracking-Algorithmus

- Probabilistic Data Association Filters (PDAF) - a tracking demo⁹³
- Rahmathullah et al. tracking algorithm metric⁹⁴
- Round Toward Vector of Values⁹⁵
- setdiff⁹⁶

3.4.1 Mathematische Grundlagen

Die Entwicklung des Tracking-Algorithmus basiert auf den Annahmen, dass die verwendeten Modell- und Sensorgleichungen linear sind und das auftretende Rauschen gaußförmig ist. In diesem Fall wird der Kalman-Filter zur Filterung und Prädiktion der Objekt-Zustände eingesetzt. Der Radarsensor

⁹³ Dubin, U.: Probabilistic Data Association Filters PDAF - a tracking demo (2016)

⁹⁴ Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: Rahmathullah et al. Metrik (2016), Quellcode erfragt

⁹⁵ R, T.: Round Toward Vector of Values (2012)

⁹⁶ Nick: setdiff (2017)

liefert Informationen über die beobachtete Szene, die zu Zuständen zusammengefasst werden. Der Zustand jeder Reflexion (Cluster) besitzt die folgende Form:

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} \quad (3-7)$$

x und y sind dabei die kartesischen Koordinaten des Clusters. Verwendet wird ein Fahrzeugkoordinatensystem mit Ursprung in der Radarsensorposition.

Für die Zustände der Objekte wird ebenfalls obiger Vektor eingesetzt. Für die Prädiktion der Objektzustände wird eine Zustandsübergangsmatrix aufgestellt. Mithilfe dieser Matrix und bekannten Zuständen zum Zeitpunkt t werden die Zustände der Objekte zum Messzeitpunkt $t + 1$ berechnet. Der Aufbau der Zustandsübergangsmatrix basiert auf der Annahme, dass sich das Objekt mit konstanter Geschwindigkeit bewegt. Die beobachteten Objekte werden als Punktmassen modelliert und bewegen sich damit unabhängig in beiden Koordinatenrichtungen. Die Zustandsübergangsmatrix A wird in Gleichung 3-8 gezeigt.

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}_{t+1} = \underbrace{\begin{bmatrix} 1 & dT & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dT \\ 0 & 0 & 0 & 1 \end{bmatrix}}_A \cdot \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}_t \quad (3-8)$$

Da eine Modellbildung die Realität nicht exakt abbildet, werden Modellungenauigkeiten durch eine Modellkovarianzmatrix erfasst. Als Störgröße auf die angenommene konstante Geschwindigkeit des Objekts wird die Beschleunigung $a_{\text{stör}}$ gewählt. Die Parametrierung der Störgröße richtet sich nach denkbaren ‘‘Sprüngen‘‘ der Beschleunigung zwischen zwei Messzeitpunkten. Der Einfluss der Störbeschleunigung auf die Zustände zeigt Gleichung 3-9. dT ist die zeitliche Differenz zwischen zwei Messpunkten. Für die y -Richtung gelten diese Zusammenhänge analog.

$$\begin{aligned} x &= \frac{1}{2}dT^2 \cdot \ddot{x} \\ \dot{x} &= dT \cdot \ddot{x} \end{aligned} \quad (3-9)$$

Eine Kovarianzmatrix enthält die folgenden Elemente:

$$\mathbf{Q} = \begin{bmatrix} \sigma_x^2 & \sigma_{x\dot{x}} & \sigma_{xy} & \sigma_{x\dot{y}} \\ \sigma_{\dot{x}x} & \sigma_{\dot{x}}^2 & \sigma_{\dot{x}y} & \sigma_{\dot{x}\dot{y}} \\ \sigma_{yx} & \sigma_{y\dot{x}} & \sigma_y^2 & \sigma_{y\dot{y}} \\ \sigma_{\dot{y}x} & \sigma_{\dot{y}\dot{x}} & \sigma_{\dot{y}y} & \sigma_{\dot{y}}^2 \end{bmatrix} \quad (3-10)$$

Eine Möglichkeit zur Berechnung dieser Matrix zeigt Formel 3-11⁹⁷.

$$\mathbf{Q} = \mathbf{G} \cdot \mathbf{G}^T \cdot a_{\text{stör}}^2, \quad (3-11)$$

Hierbei ist $\mathbf{G} = [0.5dT^2 \ dT \ 0.5dT^2 \ dT]^T$. Da die Bewegungen in x- und y-Richtung unabhängig voneinander sind, wird die obere rechte und untere linke Untermatrix zu Null gesetzt. Die entstehende Modellkovarianzmatrix mit parametrierbarer Störbeschleunigung wird in Formel 3-12 dargestellt.

$$\mathbf{Q} = a_{\text{stör}}^2 \cdot \begin{bmatrix} \frac{dT^4}{4} & \frac{dT^3}{2} & 0 & 0 \\ \frac{dT^3}{2} & dT^2 & 0 & 0 \\ 0 & 0 & \frac{dT^4}{4} & \frac{dT^3}{2} \\ 0 & 0 & \frac{dT^3}{2} & dT^2 \end{bmatrix} \quad (3-12)$$

Als nächstes wird die Messmatrix des Sensors aufgestellt. Die Schnittstelle zwischen Sensor und Tracking-Algorithmus ermöglicht eine direkte Übernahme der Sensordaten, so dass sich die Messmatrix nach Gleichung 3-13 ergibt.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-13)$$

Auch die Genauigkeit des Radarsensors fließt in Form einer Kovarianzmatrix in die Berechnungen des Kalman-Filter ein. Die Genauigkeit der Distanz- und Geschwindigkeitsmessung wird als Standardabweichung aus dem Datenblatt des Radarsensors entnommen. Durch eine Standardabweichung des Azimutwinkels vergrößert sich die Standardabweichung der y-Werte bei zunehmender Entfernung vom Sensor. Aus diesem Grund wird die Sensorkovarianzmatrix permanent in Abhängigkeit der Distanz aktualisiert. Formel 3-14 zeigt die Kovarianzmatrix. σ ist die Standardabweichung. Auf

⁹⁷ Balzer, P.: Das Kalman Filter einfach erklärt Teil 2 (2013)

der Hauptdiagonalen der Kovarianzmatrix werden die entsprechenden Varianzen der Distanz- und Geschwindigkeitsmessung eingetragen.

$$\mathbf{R} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_{\dot{x}}^2 & 0 & 0 \\ 0 & 0 & \sigma_y^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}}^2 \end{bmatrix} \quad (3-14)$$

Der Kalman-Filter arbeitet in zwei Schritten. In einem ersten Schritt wird eine Prädiktion der Objektzustände zum nächsten Messzeitpunkt durchgeführt. Im anschließenden zweiten Schritt werden diese prädizierten Zustände mithilfe der neuen Messwerte korrigiert. Im Prädiktionsschritt werden die folgenden Berechnungen durchgeführt:

$$\begin{aligned} x_{t+1} &= \mathbf{A}x \\ \mathbf{P}_{t+1} &= \mathbf{A}\mathbf{P}_t\mathbf{A}^T + \mathbf{Q}. \end{aligned} \quad (3-15)$$

Die Kovarianzmatrix \mathbf{P} beschreibt die Ungenauigkeit des berechneten Zustands und wird nach jedem Messzeitpunkt aktualisiert.

Für den Korrekturschritt wird zuerst der Kalman-Gain berechnet. Dieser beeinflusst, wie die prädizierten und gemessenen Zustände gewichtet werden und legt damit fest, ob eher den prädizierten oder den gemessenen Zuständen vertraut wird. Anschließend werden die korrigierten Zustände und die Kovarianzmatrix \mathbf{P} berechnet. \mathbf{I} ist dabei die Einheitsmatrix und z_t der Messvektor zum Messzeitpunkt t . Zusammen ergeben sich damit die in Formel 3-16 dargestellten Rechenschritte.

$$\begin{aligned} \mathbf{K}_t &= \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}_t\mathbf{H}^T + \mathbf{R})^{-1} \\ x_t &= x_t + \mathbf{K}_t(z_t - \mathbf{H}x_t) \\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t \end{aligned} \quad (3-16)$$

3.4.2 Tracking-Parameter

Beim Start des Tracking-Algorithmus wird die "Tracking Parameter"-Funktion aufgerufen. Diese legt alle für den Tracking-Algorithmus benötigten Variablen im MATLAB Workspace ab. In den Tabellen 3-1 und 3-2 sind diese Parameter notiert und beschrieben.

Tabelle 3-1: Parameter des Tracking-Algorithmus 1

Parameter	Beschreibung
Allgemeine Parameter	
MeasuredDataName	*.mat-Dateiname der realen Sensordaten
RadarModeSelection	Auswahl des Radar-Modus (Nahbereich, Fernbereich oder kombiniert)
MeasurementSource	Quelle der Sensordaten (real oder simuliert)
Niteration	Anzahl der Monte-Carlo-Durchläufe für die Algorithmus-Untersuchung
Allgemeine Objekt-Tracker Parameter	
NObjectTrackers	Anzahl Objekt-Tracker
ValidationGatePercentage	Größe des Suchbereichs um prädizierte Objektposition. xx % aller zu erwartenden Messwerte liegen innerhalb des Suchbereichs.
Parameter der grafischen Ergebnisdarstellung	
PlotAxis	Limit der Achsen des MATLAB Plots
ShowClusterGroupsPlot	Plot mit unverarbeiteten Rohdaten anzeigen
ShowPlot	Plot mit verfolgten Objekten anzeigen
Record	Tracking-Verlauf als Video abspeichern
RecordName	Dateiname des Tracking-Verlauf Videos
RecordFPS	Anzahl der Bilder pro Sekunde des Tracking-Verlauf Videos
Clustergruppen Parameter	
MaxDistanceForGroup	Maximaler Abstand zwischen Clustern einer Clustergruppe
MaxVRelDeviation	Maximale Differenz der Relativgeschwindigkeit zwischen Clustern einer Clustergruppe
Kalman-Filter Parameter	
TargetModellaccstdv	Standardabweichung der Beschleunigung des beobachteten Objekts
SensorDistancstdvx	Standardabweichung der Distanzmessung in x-Richtung
SensorSpeedstdvx	Standardabweichung der Geschwindigkeitsmessung in x-Richtung
AzimuthAccuracy	Standardabweichung der Azimuthbestimmung
SensorDistancstdvy	Standardabweichung der Distanzmessung in y-Richtung
SensorSpeedstdvy	Standardabweichung der Geschwindigkeitsmessung in y-Richtung
initPcorr	Initialisierung der P-Matrix. Getrennt nach x- und y-Richtung

Tabelle 3-2: Parameter des Tracking-Algorithmus 2

Parameter	Beschreibung
Parameter des Track-Managements	
mausNN	m/N Auswahl. Wahl der N Messungen
mausNm	m/N Auswahl. Wahl der Anzahl m
maxNundetected	Anzahl von nichtvorhandenen Messwerten bis beobachtetes Objekt gelöscht wird
MaxHistoryDistance	Maximale Distanz zwischen den Positionen zweier Objekt-Tracker
MaxHistoryVRelDeviation	Maximale Relativgeschwindigkeits-Differenz zweier Objekt-Tracker
Parameter der simulierten Trajektorien	
TrajectorySelection	Auswahl der Trajektorie (Formationsfahrt/Autobahnfahrt)
SingleManeuverTime	Zeit eines Einzelmanövers (Insgesamt 4 Einzelmanöver pro Durchlauf)
alpha	Winkel der Beschleunigung der seitlichen Objekte
Acceleration	Beschleunigung der seitlichen Objekte
VelocityMiddleVehicle	Geschwindigkeit des mittleren Objekts
MinDistance	Minimale Distanz zwischen den seitlichen und dem mittleren Objekt
dT	Abtastezeit des simulierten Sensors
NObjectTracks	Anzahl von simulierten Reflexionen
NStaticObjects	Anzahl statischer Objekte
FoVAngle	Simulierter Radar-Öffnungswinkel
rng	Seed für den MATLAB Random Number Generator
SensorParameter	Simulierte Standardabweichung der Position, Geschwindigkeit und des Azimutwinkels
SensorResolution	Simulierte Sensorauflösung durch Runden auf parametrierbare Werte
Parameter der Tracking-Algorithmus-Metrik	
ShowOSPATPlot	Anzeige des OSPA-Metrik-Plot
OSPA-T Parameter P, C, L	Parameter der OSPA-T Metrik
Rahmatullah et al. Parameter P, C, Maxiter, ρ , γ	Parameter der Rahmathullah et al. Metrik

3.4.3 Definition der Objekt-Tracker

Während der Initialisierungsphase des Tracking-Algorithmus werden die Objekt-Tracker definiert. Jeder Objekt-Tracker hat die Möglichkeit, ein bestätigtes oder potentiell Objekt zu verfolgen und setzt dafür einen eigenständigen Kalman-Filter ein. Die Erstellung dieser Kalman-Filter geschieht in diesem Programmschritt. Weitere Parameter der Objekt-Tracker sind in Tabelle 3-3 dargestellt.

Tabelle 3-3: Zusätzliche Parameter des Objekt-Trackers

Variable	Initialwert	Beschreibung
Lifetime	0	Lebensdauer des verfolgten Objekts
History	$[0, 0, 0, 0]^T$	Zustandshistorie des verfolgten Objekts
InitialisationHistory	$[0, 0, 0, 0]$	Binäre Notation der Objekt-Reflexionen in der Initialisierungsphase
InitialisationCounter	0	Länge der Initialisierungsphase
TrackerStatus	0	Status des Objekt-Trackers (0: Nicht initialisiert, 1: In der Initialisierungsphase, 2: Initialisiert, 3: In der Löschphase)
NUndetected	0	Anzahl der nicht gemessenen Objekt-Reflexionen des initialisierten Objekt-Trackers

3.4.4 Reale Sensordaten

Während dieses Programmschritts werden die Sensordaten des realen Radarsensors in MATLAB eingelesen. Dafür werden die aus der Radarsoftware exportierten Daten in ein Matlab-Datenformat überführt. Um diesen Schritt bei späteren Programmabläufen nicht erneut durchführen zu müssen, werden die vorverarbeiteten Sensordaten abgespeichert. Der Programmablaufplan dieses Blocks ist im Anhang A.1 auf Abbildung A-1 dargestellt.

Im ersten Schritt wird überprüft, ob der Dateiname der exportierten Radardaten schon einmal verwendet wurde. Ist dies der Fall werden die vorverarbeiteten Daten direkt von der Festplatte in den MATLAB Workspace geladen. Handelt es sich um neue Daten, so werden diese vorverarbeitet und anschließend auf der Festplatte abgespeichert.

3.4.5 Simulierte Sensordaten

In dieser Funktion werden die simulierten Radardaten erzeugt. Diese werden für den Vergleich der Leistungsfähigkeit von Tracking-Algorithmen benötigt. Im ersten Schritt werden die Testtrajektorien definiert. Die Verläufe der Trajektorien werden in Kapitel 4.3 vorgestellt. Über einen Parameter wird die Auswahl der Testtrajektorie getroffen. Anhand der Parameter im Workspace werden die Verläufe der Testtrajektorien festgelegt. Abbildung A-2 in Anhang A.1 zeigt den Programmablaufplan für die Generierung der Testtrajektorien.

3.4.6 Clustergruppengenerierung

Die im vorherigen Schritt eingelesenen realen bzw. simulierten Sensordaten beinhalten die Positionen und Relativgeschwindigkeiten aller Radar-Reflexionen (Cluster) zum Messzeitpunkt. Von ausgedehnten Objekten, wie beispielsweise von Fahrzeugen, werden teilweise mehrere Reflexionen erfasst. Durch Sensorungenauigkeiten und Messrauschen entstehen Distanzen von mehreren Metern zwischen den Clustern eines einzigen Objekts. Zusätzlich weisen Cluster eines Objekts Unterschiede in den gemessenen Relativgeschwindigkeiten auf. Die Clustergruppengenerierung hat das Ziel, alle Cluster eines Objekts zu identifizieren und zu gruppieren. Bei Objekten, die räumlich weit voneinander entfernt sind oder die große Relativgeschwindigkeitsunterschiede aufweisen, gelingt die Gruppengenerierung unkompliziert. Bei Objekten, die sich räumlich nah sind, wie beispielsweise Fahrzeuge auf zwei parallelen Fahrstreifen und zusätzlich eine ähnliche Relativgeschwindigkeit aufweisen, ist die Trennung der Objekte aufwendig.

Der entwickelte Algorithmus zur Clustergruppengenerierung wird im folgenden vorgestellt. Er nutzt die zur Verfügung stehenden Informationen über die Clusterpositionen und -relativgeschwindigkeiten. Im ersten Prozessschritt werden die euklidischen Distanzen zwischen allen Clustern zum Messzeitpunkt berechnet. Ergebnis ist eine symmetrische $N \times N$ Matrix, wobei N die Anzahl der Cluster ist. Eintrag (i, j) ist damit die Distanz zwischen Cluster i und Cluster j . Weiterhin werden Matrizen für die Relativgeschwindigkeitsdifferenzen der Cluster für beide Koordinatenrichtungen x und y aufgestellt. Im Gegensatz zur Distanz bleibt damit die Information der Geschwindigkeitsrichtung erhalten. Die Einträge aller drei Matrizen werden nachfolgend mit parametrierbaren Schwellwerten für die maximale Distanz und Relativgeschwindigkeit einer Clustergruppe verglichen. Durch diesen Schritt entstehen drei Binärmatrizen. Ihre Einträge sind 1, wenn die Distanz bzw. Relativgeschwindigkeitsdifferenz unterhalb der gewählten Schwelle liegt und 0, wenn sie darüber liegt. Alle drei Matrizen werden im letzten Schritt mit einer logischen UND-Verknüpfung vereint. Tabelle 3-4 zeigt ein beispielhaftes Ergebnis dieses Prozessschritts der Clustergruppengenerierung.

Tabelle 3-4: Zwischenergebnis Clustergruppengenerierung

	1	2	3	4	5
1	1	0	0	0	1
2	0	1	1	0	0
3	0	1	1	1	1
4	0	0	1	1	1
5	1	0	0	1	1

Im nächsten Prozessschritt werden alle Cluster gesucht, die miteinander in Verbindung stehen. Grundlage ist die obige Matrix. Ein Cluster wird einer Gruppe zugeordnet, wenn dieser mit mindestens einem Mitglied der Gruppe in Verbindung steht. Angewandt auf das Beispiel aus Tabelle 3-4 bedeutet dies, dass Cluster 1 und Cluster 5 eine Gruppe bilden. Cluster 5 wiederum steht in Verbindung

mit Cluster 4, so dass auch Cluster 4 in die Gruppe aufgenommen wird. Weiterhin steht Cluster 4 in Verbindung mit Cluster 3 und schließlich Cluster 3 mit Cluster 2. In diesem Fall wird also eine Clustergruppe aus allen fünf Clustern gebildet, obwohl beispielsweise Cluster 1 keine Verbindung zu Cluster 2 aufweist. Diese Logik wird eingesetzt, da die detailarmen Daten eines Radarsensors eine Trennung von zwei verschiedenen Objekten aufwendig macht. Entsteht beispielsweise ein Cluster an dem linken vorderen Radkasten eines Fahrzeugs und ein weiterer Cluster am rechten Teil des Fahrzeugshecks, so weisen diese beiden Cluster eine gewisse Distanz auf. Es besteht allerdings die Möglichkeit, dass diese Clusterdistanz auch durch zwei parallel fahrende Fahrzeuge erzeugt wurde. Ohne weitere Informationen und eine Sensorfusion, beispielsweise mit Daten einer Videokamera, ist die exakte Trennung von beobachteten Objekten anfällig für Fehler. Einfacher gestaltet sich die Clustergruppengenerierung bei Objekten, die eine unterschiedliche Relativgeschwindigkeit aufweisen. Beispiele hierfür sind zwei Fahrzeuge, die sich entgegengesetzt bewegen oder Fahrzeuge, die sich gegenseitig mit ausreichendem Geschwindigkeitsüberschuss überholen.

Die Suche der Mitglieder einer Clustergruppe erfolgt durch eine rekursive Funktion. Begonnen wird die Gruppenbildung mit Cluster 1. Im ersten Schritt werden alle Cluster von Spalte 1 zur ersten Gruppe hinzugefügt. Für jedes Cluster dieser Spalte wird wiederum die rekursive Funktion aufgerufen, um erneut alle Nachbarcluster zu finden. Diese Prozedur wird wiederholt, bis die Nachbarn aller Cluster der ersten Spalte zu Clustergruppe 1 hinzugefügt sind. Im nächsten Schritt wird der erste Cluster gesucht, der nicht in Gruppe 1 eingeordnet ist. Dieser Cluster bildet den ersten Eintrag der zweiten Gruppe und die rekursive Suche nach Gruppenmitgliedern beginnt erneut. Der Algorithmus endet, wenn alle Cluster einer Gruppe zugeordnet sind. In Anhang A.1 auf Abbildung A-3 wird der Programmablaufplan der Clustergruppengenerierung gezeigt. Die Unterfunktion "Rekursive Suche nach potentiellen Clustergruppenmitgliedern" wird im Anhang A.1 in Abbildung A-4 genauer vorgestellt.

Sind alle Cluster in Gruppen eingeteilt, werden die Mittelwerte der Positionen und Relativgeschwindigkeiten einer jeden Gruppe berechnet. Diese Mittelwerte bilden die neuen Sensordaten, die in den folgenden Programmschritten des Tracking-Algorithmus verarbeitet werden.

3.4.7 Datenassoziation

Im Datenassoziationsschritt werden die generierten Clustergruppen den Objekt-Trackern zugeordnet. Die Zuordnung erfolgt über die Mahalanobisdistanz zwischen der Clustergruppe und dem prädiertem Zustand des Objekt-Trackers. Über den Parameter "ValidationGatePercentage" (Tabelle 3-1) wird die maximale Distanz für eine Zuordnung festgelegt.

In einer Matrix werden die Mahalanobisdistanzen zwischen allen Clustergruppen und Objekt-Trackern eingetragen. Die Zeilennummern entsprechen den Objekt-Tracker-Nummern und die Spalten den Clustergruppennummern. Durch einen anschließenden Vergleich der Distanzen mit dem Schwellwert entsteht eine Binärmatrix. Jede 1 in einer Zeile ordnet die entsprechende Clustergruppe dem Objekt-Tracker der Zeile zu. Tabelle 3-5 zeigt ein Beispiel einer solchen Wahrheitstabelle.

Tabelle 3-5: Beispiel Datenassoziation. Die Zeilennummer repräsentiert die Objekt-Tracker-Nummer und die Spaltennummer die Clustergruppennummer

	1	2	3	4
1	0	0	1	0
2	1	1	0	0
3	0	1	0	1
4	1	0	0	0
5	0	0	0	0

Dem Objekt-Tracker 1 wird die Clustergruppe 3 zugeordnet und dem Objekt-Tracker 2 werden die beiden Clustergruppen 1 und 2 zugeteilt. Dem Objekt-Tracker 5 wird dagegen keine Clustergruppe zugeteilt. Der Programmablaufplan der Datenassoziation wird im Anhang A.1 auf Abbildung A-5 dargestellt.

Für die Berechnung der Mahalanobisdistanz wird die Innovationsmatrix \mathbf{S} benötigt. Diese berechnet sich nach Formel 3-17. Die Innovationsmatrix wird auch für die Berechnung der Kalman-Gain in Formel 3-16 verwendet und von dort übernommen.

$$\mathbf{S} = \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R} \quad (3-17)$$

Die Mahalanobisdistanz ist definiert durch

$$d(X, Y|S) = \sqrt{(X - Y)^T \mathbf{S} (X - Y)}, \quad (3-18)$$

wobei X und Y Vektoren (Zustände der Objekt-Tracker bzw. Clustergruppen) und \mathbf{S} die Innovationsmatrix ist.

3.4.8 Aktualisierung der Objekt-Tracker

Nach Zuordnung der Clustergruppen im vorherigen Schritt werden die Objekt-Tracker aktualisiert. Dazu werden im ersten Schritt ihre Status aktualisiert. Im anschließenden zweiten Schritt werden die zugeordneten Clustergruppen zur Korrektur der prädizierten Zustände verwendet.

Für die Aktualisierung jedes einzelnen Objekt-Tracker-Status wird überprüft, ob dem betreffenden Objekt-Tracker im vorherigen Schritt mindestens eine Clustergruppe zugeordnet wurde. Der neue Status des Trackers wird anschließend in Abhängigkeit vom ursprünglichen Status gesetzt. Alle möglichen Statuswerte sind in Tabelle 3-6 dargestellt.

Tabelle 3-6: Statuswerte der Objekt-Tracker

Status	Beschreibung
0	Objekt-Tracker nicht in Verwendung
1	Objekt-Tracker in Initialisierungsphase
2	Objekt-Tracker initialisiert
3	Objekt-Tracker in Löschphase

Abbildung A-6 im Anhang A.1 zeigt den Programmablauf der Status-Aktualisierung.

Im zweiten Teil der Tracker-Aktualisierung werden die Zustände der Objekt-Tracker korrigiert und anschließend eine Prädiktion der Zustände zum nächsten Messzeitpunkt durchgeführt. Die Korrektur der Zustände erfolgt durch den Korrekturschritt des Kalman-Filters. Anschließend werden die Zustände zum nächsten Messzeitpunkt mithilfe der Zustandsübergangsmatrix berechnet.

Zu beachten ist, dass für Objekt-Tracker, denen keine Clustergruppe zugeordnet ist, kein Korrekturschritt durchgeführt wird. Die Prädiktion erfolgt durch Verwendung der nicht-korrigierten Zustände. Dieser Schritt ist erforderlich, da ein temporär verdecktes Objekt seine Bewegung fortführt und sich damit der Suchbereich für weitere Objektreflexionen weiterbewegt.

In Anhang A.1 auf Abbildung A-7 ist der Programmablaufplan des zweiten Teils der Tracker-Aktualisierung abgebildet.

3.4.9 Löschen von Tracker-Duplikaten

Um zu Vermeiden, dass mehrere Objekt-Tracker das gleiche Objekt verfolgen, werden diese Tracker-Duplikate identifiziert und gelöscht. Beibehalten wird der Objekt-Tracker, der die längste Lebenszeit aufweist.

Die Identifizierung der Duplikate erfolgt durch Untersuchung der Objekt-Tracker-Zustände. Ähnlich wie bei der Clustergruppengenerierung werden drei Matrizen für die Positionen und Geschwindigkeiten der Tracker aufgestellt. Auch in diesem Fall erfolgt die Zuordnung der Tracker über eine Schwellwertbildung. Sind die Duplikate identifiziert, werden diese nach Lebensdauer sortiert. Anschließend werden die jüngsten Objekt-Tracker gelöscht. Abbildung A-8 in Anhang A.1 zeigt den dazugehörigen Programmablaufplan.

3.4.10 Initialisierung neuer Objekt-Tracker

Im letzten Programmschritt der Tracking-Algorithmus-Schleife werden mithilfe der verbleibenden Clustergruppen ungenutzte Objekt-Tracker in die Initialisierungsphase überführt.

Dazu werden im ersten Schritt alle Clustergruppen identifiziert, die im bisherigen Teil der Schleife keinem Objekt-Tracker zugeordnet wurden. Im Anschluss daran werden alle ungenutzten Objekt-Tracker gesucht. Jede Clustergruppe wird einem Objekt-Tracker zugewiesen, indem ihr Zustand als Initialisierungszustand des Objekt-Trackers verwendet wird. Überschreitet die Anzahl der nicht zu-

geordneten Clustergruppen die Anzahl der zur Verfügung stehenden Objekt-Tracker, so werden die Clustergruppen nach Distanz zum Sensor sortiert. Die Initialisierung der Tracker geschieht anschließend aufsteigend nach Distanz der Clustergruppen. Auf Abbildung A-9 im Anhang A.1 wird der entsprechende Programmablaufplan gezeigt.

3.4.11 Bewertung der Tracking-Ergebnisse

Für eine Leistungsbewertung von Tracking-Algorithmen wird die Funktion “Bewertung des Tracking-Ergebnisses“ eingesetzt. Die Bewertung erfolgt auf Basis von Metriken und Testfällen, die in Kapitel 4 vorgestellt werden.

Nach Ende des Objekt-Trackings erfolgt die Bewertung der Tracking-Qualität. Hierfür werden die Ground-Truth- und geschätzten Objekt-Trajektorien während des Programmablaufs abgespeichert und in das geforderte Datenformat überführt. Im Anschluss daran werden die Funktionen der Metriken ausgeführt und die Ergebnisse grafisch oder in Textform ausgegeben. Je kleiner die Bewertungszahl ausfällt, desto besser ist die Tracking-Qualität. Im Anhang A.1 auf Abbildung A-10 wird der Programmablaufplan zur Leistungsbewertung der Tracking-Algorithmen dargestellt.

4 Leistungsbewertung von Tracking-Algorithmen

In diesem Kapitel wird die systematische Analyse von Tracking-Algorithmen vorgestellt. Im ersten Unterkapitel werden verschiedene Kriterien (englisch: Measures of Effectiveness (MoE)) für den Vergleich von Multiple-Target Tracking-Algorithmen eingeführt. Im zweiten Teil werden Metriken vorgestellt, die diese Kriterien zusammenfassen und damit eine Bewertungszahl berechnen. Im letzten Unterkapitel werden zwei Testfälle für den Vergleich der Leistungsfähigkeit von Tracking-Algorithmen erläutert.

Die Analyse von Tracking-Algorithmen erfolgt in drei Schritten:

1. Definition von Testfällen
2. Anwendung des Tracking-Algorithmus auf die Testfälle
3. Berechnung einer Bewertungszahl nach den Metriken in Kapitel 4.2

Üblicherweise werden diese drei Schritte durch unabhängige Monte-Carlo-Durchläufe wiederholt und ein Mittelwert der berechneten Bewertungszahlen für die Gesamtbewertung der Algorithmen herangezogen⁹⁸.

4.1 Vergleichskriterien für die Analyse von Multiple-Target Tracking-Algorithmen

Für den Vergleich verschiedener Tracking-Algorithmen werden Kriterien benötigt. Der Erfüllungsgrad eines jeden Kriteriums fließt in die Gesamtbewertung des Algorithmus ein. Damit ergeben sich für alle Kriterien ein individuelles Ergebnis. Für eine abschließende Bewertung ist daher eine Gewichtung der einzelnen Kriterien obligatorisch. Diese ist abhängig vom gesetzten Schwerpunkt des Entwicklers. Die hier vorgestellten Kriterien sind in den Veröffentlichungen von Blackman et al.⁹⁹ und Ristic et al.¹⁰⁰ zu finden.

4.1.1 Zeitliche Kriterien

Diese Kriterien beziehen sich auf den zeitlichen Verlauf des Trackings von Objekten.

Track-Initialisierung

Die Track-Initialisierungsdauer beschreibt die benötigte Zeitspanne, um ein Objekt im beobachteten Sichtbereich des Sensors zu erkennen. Neue Objekte entstehen beispielsweise am Rand des Sensorsichtfelds durch das Eintauchen der Objekte in den beobachteten Bereich. Zusätzlich werden Objekte auch innerhalb des Sichtfelds geboren. Die Festlegung der Track-Initialisierungsdauer unterliegt einem Zielkonflikt. Das erste Ziel ist die schnelle Erfassung neuer Objekte im Sichtfeld. Das

⁹⁸ Ristic, B. et al.: A Metric for Performance Evaluation of Multi-Target Tracking Algorithms (2011)

⁹⁹ Blackman, S.; Popoli, R.: Design and analysis of modern tracking systems (1999), S.899-906

¹⁰⁰ Ristic, B. et al.: A Metric for Performance Evaluation of Multi-Target Tracking Algorithms (2011)

zweite Ziel ist eine möglichst kleine Falsch-Positive-Rate bei der Identifizierung von Objekten durch Sensorrauschen oder Clutter. Die Auflösung dieses Zielkonflikts geschieht durch die Nutzung des m/N-Verfahrens. N ist dabei die Anzahl konsekutiver Messungen und m die Anzahl von erkannten Reflexionen des potentiellen Objekts. Erfüllt eine Reihe von Messungen diese m/N-Voraussetzung, so wird das neue Objekt vom Tracking-Algorithmus bestätigt.

Der Erfüllungsgrad dieses Kriteriums ist umso größer, je schneller ein neues Objekt identifiziert wird.

Track-Löschung

Dieses Kriterium beschreibt die Zeitspanne bis ein Objekt aus der Objektliste gelöscht wird. Objekte werden gelöscht, wenn sie aus dem beobachteten Bereich des Sensors verschwinden. Dies ist beispielsweise der Fall, wenn sich das beobachtete Objekt aus dem Sichtfeld des Sensors herausbewegt oder wenn das Ego-Fahrzeug eine Gierbewegung ausführt und dadurch das Objekt das Sensorsichtfeld verlässt. Weitere Möglichkeiten sind die plötzliche Verdeckung von Objekten durch andere Objekte oder Messfehler des Sensors.

Die Wahl der Parameter für die Löschung von Tracks unterliegt ebenfalls einem Zielkonflikt. Durch Messfehler des Radarsensors kann es zu kurzzeitigen Falsch-Negativ Messungen kommen. In diesem Fall wird das Objekt nicht unverzüglich gelöscht, sondern weitere Messungen abgewartet. Das zweite Ziel ist die schnellstmögliche Löschung nicht-(mehr)-existenter Objekte. Der Erfüllungsgrad dieses Kriteriums ist umso größer, je schneller nicht-existente Objekte vom Tracking-Algorithmus gelöscht werden.

4.1.2 Genauigkeit der Tracking-Ergebnisse

Die verschiedenen Kriterien der Genauigkeit beschreiben die auftretenden Differenzen zwischen den vom Tracking-Algorithmus prädizierten Zuständen und den realen Zuständen der beobachteten Objekte.

Genauigkeit der Positions-Bestimmung

Das erste Kriterium ist die Genauigkeit der prädizierten Position eines beobachteten Objekts. Zwischen der realen und prädizierten Position des Objekts besteht eine Distanz. Gründe für diesen Unterschied sind sowohl Messfehler des Radarsensors als auch das ungenaue Bewegungsmodell des Objekts und der damit fehlerbehafteten Prädiktion der Position.

Die Genauigkeit der Positions-Bestimmung des Tracking-Algorithmus wird bestimmt, indem der euklidische Abstand zwischen der prädizierten und der realen Position berechnet wird. Diese Berechnung wird für alle Messpunkte einer Objekttrajektorie durchgeführt und die entstehenden Distanzen addiert. Nach diesem Kriterium wird der Tracker mit der geringsten Gesamtdistanz als bester Algorithmus bewertet.

Genauigkeit der Geschwindigkeits-Bestimmung

Das zweite Kriterium ist die Untersuchung der Genauigkeit der prädizierten Geschwindigkeit des Tracking-Algorithmus. Der Vergleich zwischen den realen und prädizierten Geschwindigkeiten verläuft analog zur Positionsgenauigkeit. Die Geschwindigkeitsdifferenzen werden für jeden Messzeitpunkt berechnet und anschließend addiert. Der Tracker mit der geringsten Gesamtdifferenz wird als bester Algorithmus bewertet.

4.1.3 Kontinuität der Objektverfolgung

Für die Untersuchung der Stetigkeit der Objekttrajektorien werden die Kriterien der Tracking-Kontinuität beschrieben. Ziel der Tracking-Algorithmen ist die Generierung von lückenlosen Trajektorien.

Track-Fragmentierung

Die Fragmentierung einer Objekttrajektorie führt dazu, dass die geschätzten Zustände der beobachteten Objekte nicht zu jedem Zeitpunkt bekannt sind. Der Erfüllungsgrad dieses Kriterium ist umso größer, je lückenloser die Trajektorien prädiziert werden.

Track-Bezeichnung

Die Track-Bezeichnung ist das zweite Kriterium der Track-Kontinuität. Neben der geringen Fragmentierung der Trajektorien sind auch Änderungen ihrer Bezeichnungen während des Beobachtungszeitraums zu vermeiden. Wird beispielsweise ein Objekt "A" und ein anderes "B" genannt, so ist die Bezeichnung der Objekte im besten Fall über die Dauer des Beobachtungszeitraums konstant und die Bezeichnungen werden nicht vertauscht. Ebenfalls zu vermeiden ist eine Umbenennung der Objekte, wenn es sich um das gleiche Objekt handelt. Beispielsweise sollte aus "A" nicht "C" werden. Das Kriterium wird umso besser erfüllt, je geringer die Anzahl von Bezeichnungswechsel während des Beobachtungszeitraums ist.

4.2 Metriken für Multiple-Target Tracking-Algorithmen

Die vorgestellten Kriterien aus Abschnitt 4.1 ermöglichen die Untersuchung der Leistungsfähigkeit verschiedener Tracking-Algorithmen. Die Auswertung mithilfe dieser Kriterien geschieht entweder durch einen direkten Vergleich der Erfüllungsgrade der einzelnen Kriterien oder mit sogenannten Tracking-Metriken. Diese Metriken fassen verschiedene Kriterien zusammen, definieren eine Gewichtung und bilden damit eine Bewertungszahl in einem metrischen Raum. Ein metrischer Raum erfüllt die mathematischen Anforderungen positive Definitheit (Gleichung 4-1), Symmetrie (Glei-

chung 4-2) und Dreiecksungleichung (Gleichung 4-3). Mathematisch erfüllt damit die Distanz ρ auf dem kartesischen Produkt $\mathbf{X} \times \mathbf{X}$ für alle $\mathbf{x}, \mathbf{y} \in \mathbf{X}$ die folgenden Axiome¹⁰¹:

$$\rho(x, y) = 0, \text{ wenn } x = y \quad (4-1)$$

$$\rho(x, y) = \rho(y, x) \quad (4-2)$$

$$\rho(x, y) + \rho(y, z) \leq \rho(x, z) \quad (4-3)$$

Jede vorgestellte Metrik definiert einen eigenen metrischen Raum und damit eine eigene Maßzahl für die Leistungsfähigkeit der Tracking-Algorithmen. Alle Metriken erzeugen eine dimensionslose Größe, mit dem die Algorithmen miteinander verglichen werden. Mithilfe dieser Maßzahl lässt sich eine Rangfolge der Leistungsfähigkeit der Algorithmen erstellen. Zu beachten ist, dass jede Metrik die einzelnen Kriterien unterschiedlich gewichtet und damit der Vergleich der Algorithmen über verschiedene Metriken hinweg nicht möglich ist. Je kleiner die berechneten Maßzahlen der Metriken sind, umso besser ist der Tracking-Algorithmus zu bewerten. In den folgenden Abschnitten werden einige der entwickelten Metriken für Multiple-Target Tracking-Algorithmen vorgestellt und ihre Vor- und Nachteile diskutiert.

4.2.1 Hausdorff-Metrik

Eine Metrik zum Vergleich der Leistungsfähigkeit von Tracking-Algorithmen ist die Hausdorff Metrik¹⁰². Diese wird überwiegend in der Bildverarbeitung eingesetzt und erfasst den Unterschied zwischen den Pixeln zweier Bilder. Mithilfe dieser Metrik wird der Grad der Übereinstimmung zwischen untersuchten Bildern bestimmt. In anderen Worten ermittelt die Metrik die wechselseitige Deckungsgleichheit der Bilder. Übertragen auf die Anwendung auf die Bewertung von Multiple-Target Tracking-Algorithmen werden die realen Trajektorien und die prädizierten Trajektorien übereinander gelegt und ihre Deckungsgleichheit untersucht. Nachteil der Hausdorff Metrik ist der ungünstige Umgang mit Kardinalitätsfehlern, also der fehlerhaften Erkennung der Anzahl von Objekten in der beobachteten Szene. Die Metrik reagiert auf diese Art von Fehlern nur gering, so dass die Leistungsfähigkeit der Tracking-Algorithmen nicht korrekt wiedergegeben wird (Siehe Szenario C auf Abbildung 4-1). Eine weitere Schwachstelle ist die fehlende Beachtung von Track-Kontinuitätskriterien bei der Bewertung der Algorithmen. Ein zusätzlicher Kritikpunkt ist der große Einfluss einzelner Messausreißer auf die Gesamtbewertung der Tracking-Algorithmen.

Seien S und T zwei nichtleere kompakte Teilmengen eines metrischen Raums und $d(\mathbf{x}, \mathbf{y})$ eine Di-

¹⁰¹Encyclopedia of Mathematics: Definition einer Metrik (2017)

¹⁰²Hazewinkel, M.: Hausdorff Metrik (2001)

stanz (beispielsweise die Euklidische- oder Mahalanobis-Distanz) in diesem metrischen Raum. Dann ist die Hausdorff-Distanz definiert durch

$$d^H(S, T) = \max \left\{ \max_{x \in S} \min_{y \in T} d(x, y), \max_{y \in T} \min_{x \in S} d(x, y) \right\}. \quad (4-4)$$

4.2.2 Optimal Mass Transfer Metrik

Im Jahr 2004 stellten Hoffmann und Mahler die Optimal Mass Transfer (OMAT)-Metrik für einen Vergleich der Leistungsfähigkeit von Multiple-Target Tracking-Algorithmen vor¹⁰³. Diese Metrik basiert auf der Wasserstein-Distanz. Ziel bei der Entwicklung ist die Beseitigung einiger Schwachstellen der Hausdorff-Metrik.

Für $1 \leq p \leq \infty$ und finiten nichtleeren Teilmengen $X = \{x_1, \dots, x_m\}$ und $Y = \{y_1, \dots, y_n\}$ im metrischen Raum W sind die Wasserstein-Distanzen definiert als

$$d_p(X, Y) := \min_{\mathbf{C}} \left(\sum_{i=1}^m \sum_{j=1}^n \mathbf{C}_{i,j} d(x_i, y_j)^p \right)^{1/p} \quad (4-5)$$

und

$$d_\infty(X, Y) := \min_{\mathbf{C}} \max_{1 \leq i \leq m, 1 \leq j \leq n} \tilde{\mathbf{C}}_{i,j} d(x_i, y_j). \quad (4-6)$$

Das Minimum wird aus der $m \times n$ Transportmatrix $\mathbf{C} = (\mathbf{C}_{i,j})$ übernommen, wobei $\tilde{\mathbf{C}}_{i,j} = 1$ wenn $\mathbf{C}_{i,j} \neq 0$ und $\tilde{\mathbf{C}}_{i,j} = 0$ im anderen Fall ist. Eine $m \times n$ Matrix \mathbf{C} ist eine Transportmatrix wenn alle ihre Einträge nicht negativ sind und wenn

$$\sum_{j=1}^n \mathbf{C}_{i,j} = \frac{1}{m} \text{ für } 1 \leq i \leq m, \quad \sum_{i=1}^m \mathbf{C}_{i,j} = \frac{1}{n} \text{ für } 1 \leq j \leq n$$

ist. d_p in Gleichung 4-5 wird hierbei die OMAT-Metrik der Ordnung p genannt. In vielen Fällen wird entweder $p = 1$ oder $p = 2$ gesetzt^{104a}.

Eine Untersuchung dieser Metrik^{104b} zeigt, dass die Probleme der Hausdorff-Metrik in Teilen korrigiert werden. In einigen Situationen wird die Problematik bei unterschiedlichen Kardinalitäten behoben. Weiterhin ermöglicht der Einsatz des Parameters p die Auswirkungen von Messausreißern auf die Gesamtbewertung der Tracking-Algorithmen zu begrenzen. Die Untersuchung von Schuhmacher

¹⁰³Hoffman, J. R.; Mahler, R.: Multitarget Miss Distance via Optimal Assignment (2004)

¹⁰⁴Schuhmacher, D., Vo, B.-T.; Vo, B.-N.: OSPA-Metrik (2008)

et al.^{104c} in Abbildung 4-1 zeigt auf der anderen Seite einige Szenarien, bei denen die OMAT-Metrik kein zufriedenstellendes Ergebnis liefert. Schuhmacher et al. beschreiben in ihrer Veröffentlichung 5 Kritikpunkte:

1. Widersprüchlichkeit der Metrik
 - In bestimmten Szenarien mit Kardinalitätsfehlern verhält sich das Ergebnis der Metrik nicht wie intuitiv erwartet
2. Kardinalitätsfehler werden umständlich korrigiert
3. Abhängigkeit der Metrik von der geometrischen Verteilung der Objekte im Raum
4. Metrik ist nicht definiert bei Kardinalität 0
5. Metrik ist mathematisch nicht eindeutig zuzuordnen
 - Die Metrik entspricht weder der Stochastik noch der Punktprozess-Theorie

4.2.3 Optimal Subpattern Assignment Metrik

Die Optimal Subpattern Assignment (OSPA)-Metrik wird von Schumacher et al.^{104d} beschrieben. Ziel dieser Metrik ist die Ausräumung der im vorherigen Abschnitt gezeigten Kritikpunkte der OMAT-Metrik. Die OSPA-Metrik basiert weiterhin auf der Wasserstein-Distanz, beseitigt allerdings die meisten Schwächen der OMAT-Metrik.

Bezeichne $\bar{d}^{(c)}(x, y) := \min(c, d(x, y))$ als die Distanz zwischen $x, y \in W$ mit Maximum $c > 0$ und Π_k als Menge von Permutationen auf $\{1, 2, \dots, k\}$ für alle $k \in \mathbb{N} = \{1, 2, \dots\}$. Für $1 \leq p \leq \infty$, $c > 0$ und beliebigen finiten Teilmengen $X = \{x_1, \dots, x_m\}$ und $Y = \{y_1, \dots, y_n\}$ im Raum W , wobei $m, n \in \mathbb{N}_0 = \{0, 1, 2, \dots\}$ ist die OSPA-Distanz definiert als

$$\bar{d}_p^{(c)}(X, Y) := \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m \bar{d}^{(c)}(x_i, y_{\pi(i)})^p + c^p(n-m) \right) \right)^{1/p}, \quad (4-7)$$

wenn $m \leq n$ und $\bar{d}_p^{(c)}(X, Y) := \bar{d}_p^{(c)}(Y, X)$ wenn $m > n$.

Außerdem gilt

$$\bar{d}_\infty^{(c)}(X, Y) := \begin{cases} \min_{\pi \in \Pi_n} \max_{1 \leq i \leq n} \bar{d}^{(c)}(x_i, y_{\pi(i)}), & \text{wenn } m = n \\ c, & \text{wenn } m \neq n \end{cases}. \quad (4-8)$$

In jedem Fall wird die Distanz auf 0 gesetzt, wenn $m = n = 0$.

Die praktische Berechnung der OSPA-Metrik für $p < \infty$ und $m \leq n$ erfolgt durch die folgenden 3 Schritte:

1. Finde die optimale Zuordnung aller Objekte der beiden Teilmengen X und Y . Eine optimale Lösung weist die geringste globale Wasserstein-Distanz zwischen alle Punkten der Teilmengen X und Y auf. Dieser Schritt ist ein Zuordnungsproblem und kann beispielsweise durch den Einsatz der Ungarischen Methode¹⁰⁵ berechnet werden. Der Algorithmus ermittelt die optimale Zuordnung aller Objekte aus X zu Y .
2. Auf Basis der optimalen Zuordnung aller Objekte aus den beiden Teilmengen werden für jedes Punktpaar die Distanzen a_j berechnet. Für alle nicht-zugeordneten Punkte der Teilmengen wird die Distanz auf c gesetzt.
3. Berechnung der p -ten-Ordnung der OSPA-Metrik durch $((1/n) \sum_{j=1}^n a_j^p)^{1/p}$.

In der Veröffentlichung von Schuhmacher et al. werden die Parameter p und c weiter untersucht. Wird p vergrößert und c konstant gehalten, so erhöht sich der Einfluss von Messausreißern auf die Gesamtbewertung der Tracking-Algorithmen. Mithilfe des Parameters p lässt sich damit die Gewichtung der Punktpaar-Distanzen verändern. Mit Parameter c andererseits lässt sich die Höhe des Einflusses von Kardinalitätsfehlern auf die Gesamtbewertung einstellen. Fehlt für einen Punkt einer der Teilmengen das Gegenstück in der anderen Teilmenge, so wird die Distanz dieses Punkts auf c gesetzt. Kardinalitätsfehler der Tracking-Algorithmen werden umso stärker gewichtet, je größer Parameter c ist.

Mit der Entwicklung der OSPA-Metrik werden einige der Kritikpunkte der OMAT-Metrik korrigiert. Abbildung 4-1 zeigt verschiedene Szenarien für den Vergleich der Hausdorff-, OMAT-, und OSPA-Metrik.

Die Szenarien der Abbildung 4-1 zeigen die Unterschiede der untersuchten Metriken in speziellen Anwendungsfällen. Szenario **A** zeigt den Fall mit zwei fehlerhaften prädierten Objektpositionen. Die Hausdorff-Metrik ergibt in diesem Fall ∞ und die OMAT-Metrik ist nicht definiert. Durch den Einsatz des Parameters c der OSPA-Metrik kann für dieses Szenario ein OSPA-Werte angegeben werden. Szenario **B** zeigt eine Vielzahl ähnlicher Punktpaare aus beiden Teilmengen und einen Ausreißer in der prädierten Punktmenge. Die Hausdorff-Metrik reagiert empfindlich auf diesen Ausreißer und berechnet einen hohen Wert der Metrik, obwohl das Tracking-Ergebnis intuitiv nicht schlecht gesehen wird. Durch den Parameter p in der OMAT- und OSPA-Metrik wird der Einfluss des Ausreißers begrenzt und damit dem guten Tracking-Ergebnis der anderen Punkte Rechnung getragen. Szenario **C** zeigt den geringen Einfluss von Kardinalitätsfehlern auf das OMAT-Ergebnis. Jedem realen Objekt werden eine Vielzahl von prädierten Objekten zugewiesen und damit intuitiv ein schlechtes Tracking-Ergebnis zugeordnet. Die OMAT-Metrik spiegelt diese Bewertung nicht wider. Das OSPA-Ergebnis nimmt dagegen den Kardinalitätsfehler in die Bewertung auf. Die Szenarien **C** bis **F** weisen alle die gleiche Hausdorff-Distanz auf, obwohl sich die Szenarien teils deutlich unterscheiden. Betrachtet man die OMAT-Ergebnisse der Szenarien **D** bis **F**, so fällt auf, dass Szenario **E** am schlechtesten bewertet wird. Da es bei Szenario **E** nur beim linken realen Objekt zu einer zweifachen prädierten Position kommt, wird dieses Szenario intuitiv besser bewertet, als beispielsweise Szenario **F**. Diese Bewertung spiegelt sich allerdings nicht im OMAT-Ergebnis wider, da Szenario **F**

¹⁰⁵Kuhn, H. W.: The Hungarian method for the assignment problem (1955)

¹⁰⁶ Schuhmacher, D., Vo, B.-T.; Vo, B.-N.: OSPA-Metrik (2008) S. 3449

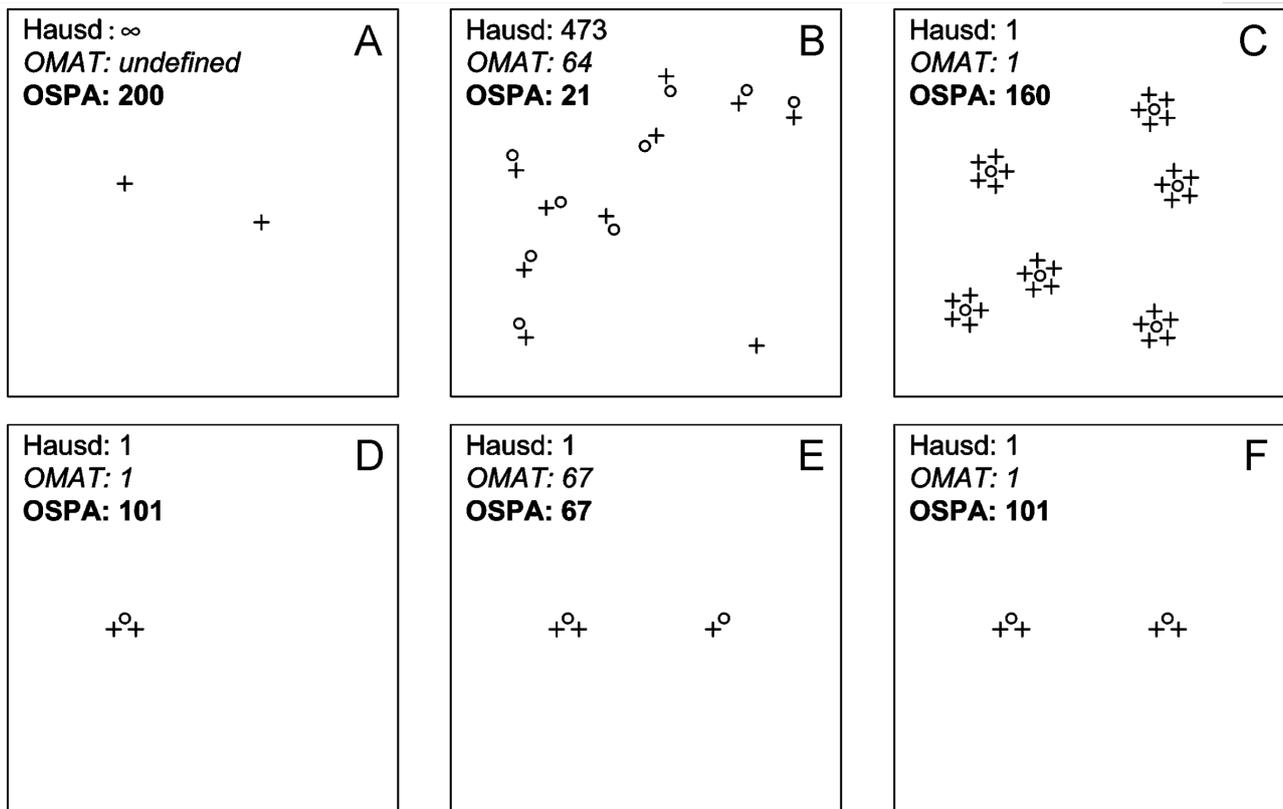


Abbildung 4-1: Szenarien für den Vergleich der Hausdorff-, OMAT- und OSPA-Metrik. \circ sind reale Objekte, $+$ sind prädizierte Objekte. $p = 1$ (OMAT und OSPA) und $c = 200$ (OSPA)¹⁰⁶

einen geringeren OMAT-Wert aufweist. Beim Einsatz der OSPA-Metrik dagegen wird diese intuitive Rangfolge erreicht.

Die OSPA-Metrik ist eine konsequente Weiterentwicklung vorheriger Metriken und korrigiert einige der Schwachstellen der Hausdorff- und OMAT-Metriken. Nachteil beim Einsatz dieser Metrik ist die weiterhin fehlende Beachtung weiterer Kriterien für den Vergleich von Tracking-Algorithmen. Die Kriterien der Track-Kontinuität, wie beispielsweise die Fragmentierung der Tracks gehen auch bei der OSPA-Metrik nicht in die Berechnung mit ein. Trotz dieser Nachteile basieren einige Tracking-Algorithmen auf der Minimierung der OSPA-Gesamtbewertung. Beispiele hierfür sind die beiden Tracking-Algorithmen Set JPDAF¹⁰⁷ und minimum mean OSPA estimator¹⁰⁸.

4.2.4 Optimal Subpattern Assignment for Tracks Metrik

Die OSPA-Metrik ist aktueller Standard für die Bewertung von Multiple-Target Tracking-Algorithmen¹⁰⁹. Sie dient als Grundlage für eine Vielzahl weiterer Metriken. Eine davon ist die Optimal Subpattern

¹⁰⁷Svensson, L., Svensson, D.; Willett, P.: Set JPDA algorithm for tracking unordered sets of targets (2009)

¹⁰⁸Guerriero, M. et al.: Shooting two birds with two bullets: How to find Minimum Mean OSPA estimates (2010)

¹⁰⁹Scheel, A.: Expertengespräch Tracking-Algorithmus-Metriken (2017)

Assignment for Tracks (OSPA-T)-Metrik¹¹⁰. Sie ergänzt die OSPA-Metrik und bezieht weitere Kriterien wie die "Track-Kontinuität" und den "Zeitlichen Ablauf" in ihre Berechnungen ein. Ristic et al. beschreiben in ihrer Veröffentlichung die grundlegende Problematik beim Einsatz von Metriken für den Vergleich von Tracking-Algorithmen. Eine wichtige Entscheidung ist die Wahl der MoE und wie diese Kriterien für die Berechnung einer einzelnen Bewertungszahl kombiniert werden. Bei den bisher betrachteten Metriken ist diese Überlegung nicht notwendig, da die Metriken allein auf Basis der globalen Gesamtdistanz aller Punkte aus den beiden Teilmengen berechnet werden. Die OSPA-T-Metrik bezieht die Erkennungsdauer von neuen Objekten und Benennungsfehler in ihre Berechnung mit ein.

Die Metrik basiert auf endlichen Mengen von Objekttrajektorien. Jede Trajektorie besteht aus einer endlichen Anzahl von Zuständen, die zeitlich konsekutiv geordnet sind. Jedem Zeitpunkt $\tau = (t_1, t_2, \dots, t_k)$ wird ein Zustand der Trajektorie zugeordnet. Eine Trajektorie X ist dann eine eindeutig markierte und sortierte Sequenz von Zuständen der Länge K .

$$X = (X_1, X_2, \dots, X_K), \quad (4-9)$$

wobei X_k , $k = 1, \dots, K$ entweder leer oder eine Menge (l, x_k) ist. l ist hierbei die Benennung der Trajektorie. Der Zustand der Trajektorie definiert sich durch die Position und Geschwindigkeit des Objekts in kartesischen Koordinaten.

Für die Berechnung wird außerdem ein Parameter e_k eingeführt. Dieser hat den Wert 1 wenn die Trajektorie zum Zeitpunkt k existiert und 0 im anderen Fall. Damit ergibt sich

$$X_k = \begin{cases} \emptyset, & \text{wenn } e_k = 0 \\ \{(l, x_k)\}, & \text{wenn } e_k = 1 \end{cases} \quad (4-10)$$

Ähnlich wie bei der Berechnung der OSPA-Metrik basiert auch die Berechnung der OSPA-T-Metrik auf der Betrachtung von zwei Mengen von Objekttrajektorien. Die eine Menge enthält die realen Trajektorien und die andere die von dem Tracking-Algorithmus prädizierten Trajektorien. Zum Zeitpunkt k ergeben sich damit die beiden Teilmengen

$$X_k = \{(l_1, x_{k,1}), \dots, (l_m, x_{k,m})\} \quad (4-11)$$

$$Y_k = \{(s_1, y_{k,1}), \dots, (s_n, y_{k,n})\}. \quad (4-12)$$

¹¹⁰Ristic, B. et al.: A Metric for Performance Evaluation of Multi-Target Tracking Algorithms (2011)

Die OSPA-T-Distanz ist definiert als

$$D_{p,c}(X_k, Y_k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m (d_c(\tilde{x}_{k,i}, \tilde{y}_{\pi(i)}))^p + (n-m)c^p \right) \right]^{\frac{1}{p}}. \quad (4-13)$$

Hierbei ist $d_c(\tilde{x}_k, \tilde{y}_k) = \min(c, d(\tilde{x}, \tilde{y}))$ und $d(\tilde{x}, \tilde{y})$ der sogenannte Basisabstand zwischen den beiden Teilmengen zum Zeitpunkt k . Dieser Basisabstand wird für die Berechnung der OSPA-T-Metrik neu definiert und ist damit einer der Unterschiede zwischen der OSPA- und der OSPA-T-Metrik.

Der Basisabstand ist definiert durch

$$d(\tilde{x}, \tilde{y}) = \left(d(x, y)^{p'} + d(l, s)^{p'} \right)^{\frac{1}{p'}}, \quad (4-14)$$

wobei $1 \leq p' < \infty$, $d(x, y)$ die Lokalisationsdistanz (die Gesamtdistanz zwischen den Objektpaaren beider Teilmengen X und Y) und $d(l, s)$ der OSPA-T-Benennungsfehler ist. Der Benennungsfehler ist definiert durch

$$d(s, t) = \alpha \bar{\delta}[s, t]. \quad (4-15)$$

$\bar{\delta}[s, t]$ ist das Komplement zum Kronecker-Symbol und damit $\bar{\delta}[i, j] = 0$, wenn $i = j$ und $\bar{\delta}[i, j] = 1$ anderenfalls.

Der Parameter $\alpha \in [0, c]$ legt die Gewichtung des Benennungsfehlers fest. Ist $\alpha = c$ so ist dieser maximal. Für $\alpha = 0$ geht der Benennungsfehler nicht die Berechnung des Gesamtergebnisses ein.

Die Berechnung der OSPA-T-Metrik basiert auf den gleichen Schritten wie bei der OSPA-Metrik. Im ersten Schritt wird die optimale Zuordnung zwischen den realen und den prädierten Trajektorien ermittelt. Diese Aufgabe wird durch einen Zuordnungsalgorithmus (z.B. Ungarische Methode) erledigt. Im Anschluss daran wird die OSPA-T-Distanz mithilfe von Gleichung 4-13 berechnet.

Die OSPA-T-Metrik wird als eine Standard-Metrik für den Vergleich von Tracking-Algorithmen eingesetzt¹¹¹. Sie erweitert die ursprüngliche OSPA-Metrik um eine weitere MoE. Die Berechnung des Benennungsfehlers erfolgt durch einen Einsatz des Kroneckersymbols und hat damit nur einen geringen Einfluss auf die Berechnungsdauer der Metrik.

Abbildung 4-2 zeigt eine beispielhafte Zuordnung von prädierten zu realen Trajektorien. In diesem Beispiel wird s_1 zu l_1 zugeordnet und s_3 zu l_2 . Es ist sicherzustellen, dass die beiden verbleibenden prädierten Trajektorien anderen realen Trajektorien zugeordnet werden. Anhand dieses Beispiels wird eine Problematik der OSPA-T-Metrik sichtbar. Der realen Trajektorie l_2 wird nach dem Ende

¹¹¹Scheel, A.: Expertengespräch Tracking-Algorithmus-Metriken (2017)

¹¹² Ristic, B. et al.: A Metric for Performance Evaluation of Multi-Target Tracking Algorithms (2011) S. 3454

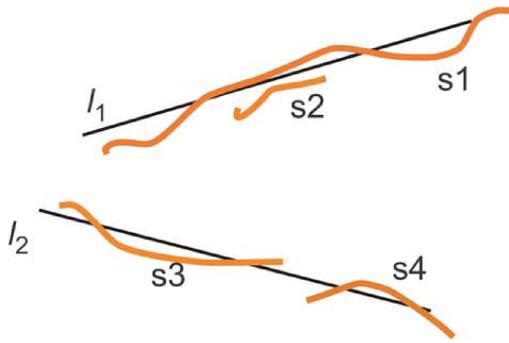


Abbildung 4-2: Beispielzuordnung Trajektorien für die OSPA-T-Metrik. l_x sind reale Trajektorien, s_x sind prädizierte Trajektorien.¹¹²

der prädizierten Trajektorie s_3 nicht die neue Trajektorie s_4 zugeordnet. Dies bedeutet, dass Lücken in den prädizierten Trajektorien einen großen Einfluss auf die Berechnung der Gesamtbewertung haben. Intuitiv ist die Zuordnung beider Trajektorien s_3 und s_4 zur realen Trajektorie l_2 zu erwarten. Die OSPA-T-Metrik erlaubt allerdings nur die Zuordnung einer prädizierten zu einer realen Trajektorie. Nachteil der OSPA-T-Metrik ist damit der Umgang mit fragmentierten Trajektorien. Trotz dieser Nachteile ist die OSPA-T-Metrik gut für den Vergleich von Tracking-Algorithmen geeignet, da sie weitere MoE in die Berechnung aufnimmt und damit einige der Nachteile der ursprünglichen OSPA-Metrik korrigiert.

4.2.5 Generalized Optimal Subpattern Assignment Metrik

Die Generalized Optimal Subpattern Assignment (GOSPA)-Metrik¹¹³ ist eine weitere Metrik, die auf der OSPA-Metrik basiert. Diese nutzt einen anderen Ansatz zur Berechnung des Kardinalitätsfehlers. Eine weitere Änderung ist die Nutzung von Zuordnungsoptimierungsansätzen anstatt der beim OSPA verwendeten Permutation der Trajektorien. Die Autoren bemängeln die Berechnung der Metriken auf Basis des globalen Minimums der Distanzen aus den beiden Teilmengen der realen und prädizierten Objekttrajektorien in Summe mit dem Kardinalitätsfehler. Ihr Vorschlag ist eine zweistufige Berechnung der Algorithmen-Bewertung. Im ersten Schritt erfolgt die Berechnung der globalen Distanz zwischen korrekt zugeordneten Trajektorienpaaren aus beiden Teilmengen. Im Anschluss daran werden Strafen für fehlerhafte und fehlende Trajektorienverläufe addiert.

Definiert wird die GOSPA-Metrik mithilfe von $c > 0$ und $1 \leq p < \infty$. Sind $X = \{x_1, \dots, x_{|X|}\}$ und $Y = \{y_1, \dots, y_{|Y|}\}$ finite Teilmengen mit den realen und prädizierten Trajektorienverläufen und $|X| \leq |Y|$ ist die GOSPA-Metrik definiert als

$$d_p^{(c,2)}(X, Y) = \left[\min_{\gamma \in \Gamma} \left(\sum_{(i,j) \in \gamma} d(x_i, y_j)^p + \frac{c^p}{2} (|X| + |Y| - 2|\gamma|) \right) \right]^{\frac{1}{p}}. \quad (4-16)$$

¹¹³Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: GOSPA-Metrik (2016)

Hier wird anstatt der Permutation die Zuordnungsmenge γ genutzt. Diese Zuordnungsmenge hat die folgenden Eigenschaften: $\gamma \subseteq \{1, \dots, |X|\} \times \{1, \dots, |Y|\}$, $(i, j), (i, j') \in \gamma \Rightarrow j = j'$ und $(i, j), (i', j) \in \gamma \Rightarrow i = i'$. Die letzten beiden Eigenschaften stellen sicher, dass jedes i und j mindestens eine Zuordnung erhält. Γ ist die Menge aller möglicher Zuordnungsmengen γ .

Die Änderung von c^p aus der OSPA-Metrik zu $\frac{c^p}{2}$ in der GOSPA-Metrik ermöglicht einen verbesserten Umgang mit falschen oder fehlerhaften Trajektorienverläufen.

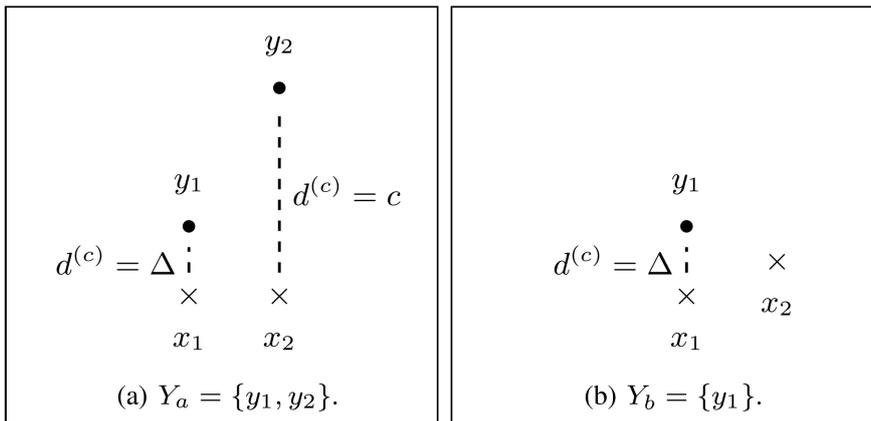


Abbildung 4-3: Beispiel Vorteil GOSPA gegenüber OSPA-Metrik. \times sind reale und \bullet prädizierte Objektpositionen¹¹⁴

Abbildung 4-3 zeigt ein solches Beispiel. Auf Abbildung 4-3(a) sind zwei reale Objekte und zwei prädizierte Objekte eingetragen. Korrekt zugeordnet wird y_1 zu x_1 . In diesem Beispiel wird y_2 fälschlicherweise x_2 zugeordnet. In dieser Situation ist y_2 eine fehlerhaft prädizierte Position und die Prädiktion für x_2 fehlt vollständig. 4-3(b) zeigt die gleiche Situation ohne die fehlerhafte Prädiktion y_2 . Intuitiv wird dieser Abbildung eine bessere Gesamtbewertung zugeordnet. Allerdings ist das OSPA-Ergebnis in beiden Fällen identisch. Der GOSPA-Algorithmus löst diese Situation dagegen zufriedenstellend. Er ordnet der linken Abbildung einen höheren Wert zu als der rechten.

$|\gamma|$ ist die Anzahl von korrekt zugeordneten Objekten. Daher sind die Ergebnisse $|X| - |\gamma|$ und $|Y| - |\gamma|$ die Anzahl der fehlenden und fehlerhaften Objektzuordnungen. Aus $\frac{c^p}{2} (|X| + |Y| - 2|\gamma|)$ folgt, dass jedes fehlerhafte oder fehlende Objekt eine Strafe von $\frac{c^p}{2}$ erzeugt.

4.2.6 Bento et al. Metrik

Die Metrik zum Vergleich der Leistungsfähigkeit von Multiple-Target Tracking-Algorithmen von Bento¹¹⁵ nutzt einen Ansatz aus der Bildverarbeitung. Auch diese Metrik hat das Ziel Ähnlichkeiten in den beiden Teilmengen der realen und prädizierten Trajektorien zu entdecken. Eine gute Übereinstimmung wird durch ein "kleines" Gesamtergebnis und eine schlechte Übereinstimmung durch ein "großes" Gesamtergebnis ausgedrückt. Die Metrik von Bento verwendet die CLEAR MOT¹¹⁶.

¹¹⁴ Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: GOSPA-Metrik (2016) S. 3

¹¹⁵ Bento, J.: A metric for sets of trajectories that is practical and mathematically consistent (2016)

¹¹⁶ Bernardin, K.; Stiefelhagen, R.: Evaluating Multiple Object Tracking Performance (2008)

Metrik, die vor allem in der Bildverarbeitung eingesetzt wird. Die OSPA-Metrik hat den Nachteil, dass ganze Trajektorien der einen Teilmenge ganzen Trajektorien der anderen Teilmenge zugeordnet werden. Bento beschreibt ein Beispiel mit 100 Fledermäusen, die in einer Höhle verfolgt werden. Kommt es im Verlauf der Beobachtung zu Verwechslungen von zwei Fledermäusen, beispielsweise wenn diese nah aneinander vorbeifliegen, so berechnet die OSPA-Metrik ein intuitiv zu schlechtes Ergebnis. Schon kleine Benennungsfehler führen damit zu einer großen Bestrafung in der Gesamtbewertung. Als menschlicher Betrachter des Beispiels der beobachteten Fledermäuse haben einzelne Verwechslungen von Fledermäusen keinen großen Einfluss auf die wahrgenommene Leistungsfähigkeit des Tracking-Algorithmus. Die CLEAR MOT-Metrik teilt die einzelnen Trajektorien dagegen in Abschnitte ein und ordnet damit Teile von Trajektorien aus der einen Menge Teilen von Trajektorien der anderen Menge zu. Diese Zuteilung erfolgt unter dem Gesichtspunkt der Minimierung der Gesamtdistanz zwischen den Trajektorienabschnitten beider Mengen. Bento zeigt in seiner Veröffentlichung das Problem auf, dass die CLEAR MOT-Metrik mathematisch nicht konsistent ist und damit nicht-intuitive Gesamtbewertungen berechnet. Außerdem führt er den Beweis an, dass CLEAR MOT keine Metrik im eigentlichen Sinn darstellt.

Aus diesem Grund schlägt er zwei neue Metriken vor. Eine der Metriken nennt er “Natürliche“-Metrik. Diese stellt eine geschlossene Lösung der Bewertung von Tracking-Algorithmen auf Basis von CLEAR MOT dar. Die zweite Metrik lässt sich dagegen schneller berechnen und nutzt dafür einen Optimierungsansatz für die Lösung des Zuordnungsproblems. Der Rechenaufwand verringert sich nach Bentos Angaben erheblich.

Die Ansätze von Bento lösen eine Vielzahl der Probleme des ursprünglichen CLEAR MOT-Ansatzes. Beispielsweise wird der Beweis erbracht, dass es sich hierbei tatsächlich um eine Metrik handelt und diese mathematisch konsistent ist. Nachteile bleiben der Einsatz von Dummy-Trajektorien, wenn die Kardinalität der realen und prädizierten Trajektorien nicht übereinstimmen. Ein weiterer Nachteil ist die Berechnung der Strafen für den Wechsel von Bezeichnungen während des Beobachtungszeitraums. Bei unterschiedlicher Kardinalität entstehen so Wechsel der Bezeichnungen zwischen Dummy-Trajektorien und realen Trajektorien. Diese Wechsel sind nicht intuitiv und verfälschen damit das zu erwartende Ergebnis. Schließlich ist ein letzter Nachteil der Metrik von Bento das Wegfallen des Parameters p aus der OSPA-Metrik. Mithilfe dieses Parameters wird eingestellt, wie stark der Einfluss von Messausreißern auf die Gesamtbewertung ist. Dieser Parameter wird für die Optimierung von Tracking-Algorithmen gebraucht^{117,118}.

4.2.7 Rahmathullah et al. Metrik

Die letzte hier vorgestellte Metrik wird von Rahmathullah et al.¹¹⁹ präsentiert. Diese Metrik stellt den aktuellen Stand der Technik dar und ist eine Weiterentwicklung der Ansätze von Bento aus dem vorherigen Abschnitt. Motivation zur Entwicklung dieser Metrik ist Berechnung der Gesamtbewertung von Tracking-Algorithmen auf Basis von intuitiven Kosten für Lokalisierungsfehler, fehlende

¹¹⁷Guerriero, M. et al.: Shooting two birds with two bullets: How to find Minimum Mean OSPA estimates (2010)

¹¹⁸Baum, M., Willett, P. K.; Hanebeck, U. D.: On Wasserstein Barycenters and MMOSPA Estimation (2015)

¹¹⁹Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: Rahmathullah et al. Metrik (2016)

und fehlerhafte Trajektorienabschnitte und auftretende Wechsel der Bezeichnungen einzelner Objekte während des Beobachtungszeitraums. Auch dieser Algorithmus nutzt Optimierungsansätze zur Berechnung der Zuordnung. Es werden dafür sowohl eine geschlossene Lösung mit hohem Rechenaufwand als auch eine numerische Lösung vorgestellt.

Die vorgestellte Metrik basiert auf der Idee der GOSPA-Metrik. Im ersten Schritt werden nur diejenigen Trajektorienabschnitte betrachtet, die anderen Trajektorienabschnitten korrekt zugeordnet werden können. Unterschiedliche Kardinalitäten in beiden Mengen der realen und prädizierten Trajektorienabschnitten führen nicht dazu, dass Dummy-Trajektorien eingefügt werden. Der Verlauf einer Trajektorie enthält damit korrekt zugeordnete Teilabschnitte, Wechsel zwischen nicht-zugeordneten und zugeordneten Teilabschnitten und Wechsel der Benennung von Teilabschnitten. Rahmathullah et al. beweisen in ihrer Veröffentlichung, dass die Strafe für den Wechsel von nicht-zugeordneten zu zugeordneten Trajektorienabschnitten und umgekehrt nur halb so groß sein sollte, wie ein Benennungswechsel. Dies ist ein Unterschied zur Metrik von Bento, da dort beide Fälle gleich bestraft werden.

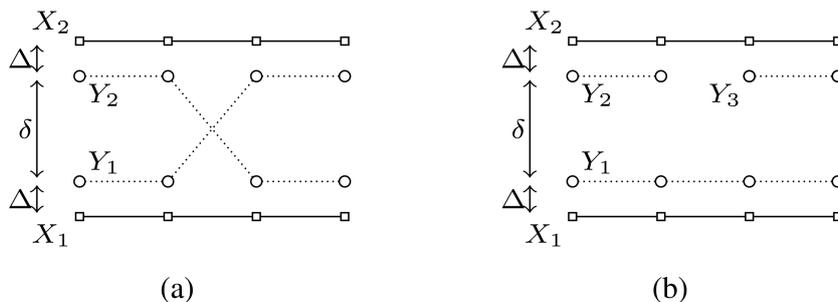


Abbildung 4-4: Beispielsituation: Vorteil Rahmathullah et al.-Metrik gegenüber Bento-Metrik¹²⁰

Abbildung 4-4 zeigt ein Beispiel für den Vergleich der Metrik von Bento mit der neuen Metrik von Rahmathullah et al. Nach Betrachtung der Abbildung scheint das Tracking-Ergebnis von Abbildung 4-4(b) besser als das Tracking-Ergebnis von 4-4(a) zu sein. Für den Fall $\delta \gg c$ und $\Delta \ll c$ sollte die Strafe daher in Abbildung 4-4(a) größer sein als in 4-4(b). Wendet man in diesem Beispiel die Metrik von Bento an, so werden die Lücken durch Dummy-Daten gefüllt. Die Berechnung des Ergebnisses der Bento-Metrik ergibt eine identische Bewertung beider Tracking-Ergebnisse. Grund hierfür ist eine Strafe für den Wechsel der Benennung zwischen einem Dummy-Zustand und einem realen Zustand der Trajektorie. Dieser Wechsel tritt auf, da die Trajektorie künstlich mit Dummy-Daten gefüllt wird. Die Veröffentlichung von Rahmathullah et al. stellt eine Metrik vor, die eine Vielzahl von Problemen anderer Metriken behebt. Sie ist bewiesenermaßen eine Metrik im betrachteten Raum und mathematisch konsistent definiert. Sie kombiniert eine Vielzahl von Kriterien für den Vergleich der Leistungsfähigkeit von Tracking-Algorithmen auf einer intuitiv natürlichen Art und Weise. Sie bezieht sowohl Lokalisierungsfehler, als auch Trajektorienfragmentation und Wechsel der Benennung von Trajektorien in die Bewertung von Tracking-Algorithmen ein. Nachteil ist der benötigte Rechenaufwand zur Berechnung der geschlossenen Lösung. Da die Berechnung über alle Dimensionen der einzelnen Zu-

¹²⁰ Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: Rahmathullah et al. Metrik (2016) S. 3

stände eines jedes Messzeitpunkts erfolgt, steigt der Rechenaufwand für größere Messzeiträume stark an. Lösung des Problems ist der ebenfalls vorgestellte numerische Berechnungsansatz, welcher einen geringeren Rechenaufwand benötigt.

4.2.8 Weitere Metriken zum Vergleich von Tracking-Algorithmen

Für die Vervollständigung der obigen Übersicht über verschiedene Entwicklungen im Bereich der Tracking-Algorithmus-Metriken werden im folgenden noch weitere Metriken erwähnt. Diese haben die OSPA-Metrik als Ursprung.

- OSPA-MT¹²¹. Keine klare Definition der Metrik bei fehlerhaften oder fehlenden Trajektorienabschnitten oder Wechsel in der Benennung.
- OSPA-TS¹²². Einsatz nur bei bekannter und identischer Länge der untersuchten Trajektorien möglich.
- H-OSPA¹²³. Einsatz nur bei punktwisem Vergleich der Trajektorien, nicht für komplette Trajektorien geeignet
- Q-OSPA¹²⁴. Einsatz nur bei punktwisem Vergleich der Trajektorien, nicht über für Trajektorien geeignet

4.3 Testfälle für Multiple-Target Tracking-Algorithmen

Dieses Unterkapitel beschreibt den Vorgang zur Erstellung von Testfällen für die systematische Untersuchung von Tracking-Algorithmen. Das Ziel bei der Erstellung der Testfälle ist die Untersuchung der Tracking-Algorithmen in herausfordernden Situationen. Die Nutzung von anspruchslosen Testfällen führt dazu, dass identische Bewertungszahlen für verschiedene Algorithmen ermittelt werden, obwohl sich die Ergebnisse in herausfordernden Situationen unterscheiden. Weiterhin ist bei der Erstellung zu beachten, dass eine Vielzahl von kritischen Situationen in die Testfälle aufgenommen werden.

4.3.1 Anforderungen an die Testfälle

Testfälle für die Bewertung von Tracking-Algorithmen haben eine Vielzahl von Kriterien zu erfüllen. Für den Vergleich der Algorithmen wird im ersten Schritt bestimmt, ob sie auf Basis von realen oder simulierten Radardaten bewertet werden. Nachteil bei der Nutzung von realen Testdaten ist das Fehlen der so genannten Ground-Truth-Verläufe der Testobjekte. Diese Ground-Truth-Informationen enthalten die tatsächlichen Trajektorieverläufe der beobachteten Objekte. Die Daten des Radarsensors sind fehlerbehaftet und damit als Ground-Truth nicht einsetzbar. Ein weiterer Nachteil bei der Nutzung von realen Radardaten ist die fehlende Möglichkeit, einzelne Testfahrten zu wiederholen. Es ist

¹²¹Vu, T.; Evans, R.: A new performance metric for multiple target tracking based on optimal subpattern assignment (2014)

¹²²Lau, R. A.; Williams, J. L.: Tracking a coordinated group using expectation maximisation (2013)

¹²³Nagappa, S., Clark, D. E.; Mahler, R.: Incorporating track uncertainty into the OSPA metric (2011)

¹²⁴He, X. et al.: A Track Quality Based Metric for Evaluating Performance of Multitarget Filters (2013)

nur mit großem Aufwand, wie beispielsweise mit Lenkrobotern, möglich, die Trajektorien der Objekte wiederholbar zu machen. Die Wiederholbarkeit bei der Bewertung von Tracking-Algorithmen ist notwendig, da alle Berechnungen und Prädiktionen auf statistischen Annahmen beruhen. Aus diesem Grund werden die Testfahrten in einem Monte-Carlo-Verfahren wiederholt, um statistische Ausreißer auszugleichen. Zu diesem Zweck ist es notwendig, dass die Ground-Truth-Trajektorieverläufe wiederholbar sind.

Ein weiterer Grund für die Nutzung von simulierten Trajektorienverläufen ist die Möglichkeit, verschiedene Sensoreinflüsse zu simulieren. Mithilfe der simulierten Verläufe lassen sich beispielsweise verschiedene Abtastraten, Rauscheinflüsse und Auflösungen des Sensors einstellen und damit unterschiedliche Einflüsse auf das Tracking-Ergebnis untersuchen. Bei realen Sensoren sind diese Parameter nur in gewissen Bereichen beeinflussbar.

Aus den genannten Gründen werden die Trajektorienverläufe in einer Simulation erzeugt. In einem ersten Schritt werden die Ground-Truth-Verläufe festgelegt. In einem zweiten Schritt werden diese Ground-Truth-Verläufe künstlich abgetastet. Dabei wird Einfluss auf die verwendete Abtastrate, die Auflösung und die Genauigkeit genommen und damit die Eigenschaften verschiedener Radarsensoren simuliert.

Für die Erstellung der Ground-Truth-Verläufe werden die Schwachstellen von Tracking-Algorithmen herangezogen. Ein Schwachpunkt ist die räumliche Trennung von Objekten, wenn diese nur eine geringe Distanz zueinander aufweisen. Bei einem Radar Tracking-Algorithmus wird zusätzlich zum Abstand zwischen zwei Objekten auch ihre Relativgeschwindigkeit mit in das Tracking aufgenommen. Sind allerdings die Geschwindigkeiten beider Objekte ähnlich und zusätzlich der Abstand gering, so ist eine Trennung der Objekte durch den Tracking-Algorithmus nicht möglich. Für eine genauere Untersuchung dieser Trennfähigkeit wird daher der minimale Abstand von Objekten im Ground-Truth parametrisiert. Eine weitere Schwachstelle von Tracking-Algorithmen ist die Verfolgung von beschleunigenden oder verzögernden Objekten. Ein Radarsensor ermöglicht das Messen der Geschwindigkeit, aber kein direktes Messen der Beschleunigung. Zur Untersuchung der Tracking-Ergebnisse bei beschleunigenden und verzögernden Objekten werden Objekte mit einer solchen Bewegung in die Testfälle aufgenommen. Ein weitere Schwachstelle von Tracking-Algorithmen sind Multiple-Target Situationen. Dies bedeutet, dass sich mehrere Objekte im Sichtbereich des Sensors befinden. Diese Objekte können sowohl statisch als auch dynamisch sein. Die Tracking-Algorithmen haben die Aufgabe, beide Arten von Objekten zu entdecken und ihre Bewegungen nachzuvollziehen. Der letzte Untersuchungsgegenstand ist das Verhalten der Algorithmen bei der Geburt und dem Sterben von Objekten im Sichtbereich des Sensors. Dabei wird untersucht, wie viel Zeit vergeht, bis ein neues Objekt erkannt wird und andersherum wie viel Zeit vergeht, bis ein nicht mehr sichtbares Objekt gelöscht wird.

Für die Analyse verschiedener Tracking-Algorithmen auf Basis der im vorherigen Kapitel vorgestellten Metriken werden zwei Testfälle vorgestellt. Diese beiden Fälle decken zusammen die gerade gestellten Anforderungen für die Untersuchung von Tracking-Algorithmen ab.

4.3.2 Testfall 1: Formationsfahrt

Der erste Testfall wird "Formationsfahrt" genannt. Beobachtet werden Objekte in einem simulierten Radarsichtfeld. Drei Objekte führen eine dynamische Bewegung aus. Die Anzahl der statischen Objekte und die Intensität des Clutters ist frei parametrierbar.

Der Testfall besteht aus vier Abschnitten. Im ersten Abschnitt führen die dynamischen Objekte 2 und 3 eine Bewegung mit konstanter Verzögerung durch. Objekt 1 bewegt sich mit konstanter Geschwindigkeit. Alle drei dynamischen Objekte bewegen sich vom simulierten Radarsensor weg. Im zweiten Abschnitt bewegt sich Objekt 2 parallel zu Objekt 1 mit konstanter Distanz. Die Geschwindigkeiten der beiden Objekte sind identisch. Das dritte Objekt führt weiterhin eine konstante Verzögerung durch. Im dritten Abschnitt bewegen sich alle drei dynamischen Objekte parallel mit identischer Geschwindigkeit. Im vierten Abschnitt entfernen sich die Objekte 2 und 3 mit der gleichen Beschleunigung und dem gleichen Beschleunigungswinkel wie in Abschnitt 1. Ziel dieses Testfalls ist die Untersuchung der Tracking-Ergebnisse bei Objekten mit konstanter Verzögerung bzw. Beschleunigung und das Verhalten der Tracking-Algorithmen bei Objekten mit identischer Geschwindigkeit und gleichzeitig geringem Abstand. Wird die einstellbare minimale Distanz der Objekte während der Parallelfahrt in den Abschnitten 2 und 3 zu gering, so werden die Objekte miteinander verschmolzen. Mithilfe dieses Testfalls wird damit zusätzlich das Sterben (Objektverschmelzung) und die Geburt (Trennung der Objekte in Abschnitt 4) von Objekten untersuchbar. Durch den parametrierbaren Winkel der Beschleunigungen von Objekt 2 und 3 wird eingestellt, mit welcher y-Geschwindigkeit sich die beiden Objekte dem mittleren Objekt annähern. Damit wird die Untersuchung seitlich auftauchender Objekte im Randbereich des Sensorsichtfelds möglich.

Abbildung 4-5 zeigt die Ground-Truth-Trajektorienverläufe der drei dynamischen Objekte. Der simulierte Radarsensor befindet sich an Position (0,0). Das mittlere Objekt startet mit konstanter Geschwindigkeit 5 m vor dem Sensor. Die Startpunkte der anderen beiden Objekte werden in Abhängigkeit der eingestellten Verzögerung und dem Winkel der Verzögerung berechnet. Die Bewegungen der Objekte werden als Punktmassen modelliert. Tabelle 4-1 zeigt alle Parameter dieses Testfalls.

Die Vektoren der x- und y-Beschleunigung in Abhängigkeit der Beschleunigung a_l bzw. a_r und des Winkels α_l bzw. α_r sind auf Abbildung 4-6 zu sehen.

In Anhang A.2 sind die Bewegungsgleichungen von Objekt 1 dargestellt. Für die Erstellung der Gleichungen von Objekt 2 wird im ersten Schritt der Zustand des Objekts zum Zeitpunkt $t = 0$ berechnet. Der Start-Zustand ist abhängig von den Parametern aus Tabelle 4-1. Objekt 2 hat zum Zeitpunkt t_1 die gleiche Geschwindigkeit wie Objekt 1. Für den Startzustand ergeben sich damit folgenden Gleichungen.

$$\begin{aligned}\dot{x}_{2,t=0} &= \dot{x}_{1,t=1} + a_l \cdot \cos(\alpha_l) \cdot t_1 \\ x_{2,t=0} &= x_{1,t=1} + \frac{1}{2}a_l \cdot \cos(\alpha_l) \cdot t_1^2 - \dot{x}_{2,t=0} \cdot t_1\end{aligned}\tag{4-17}$$

Testfall 1: Formationsfahrt

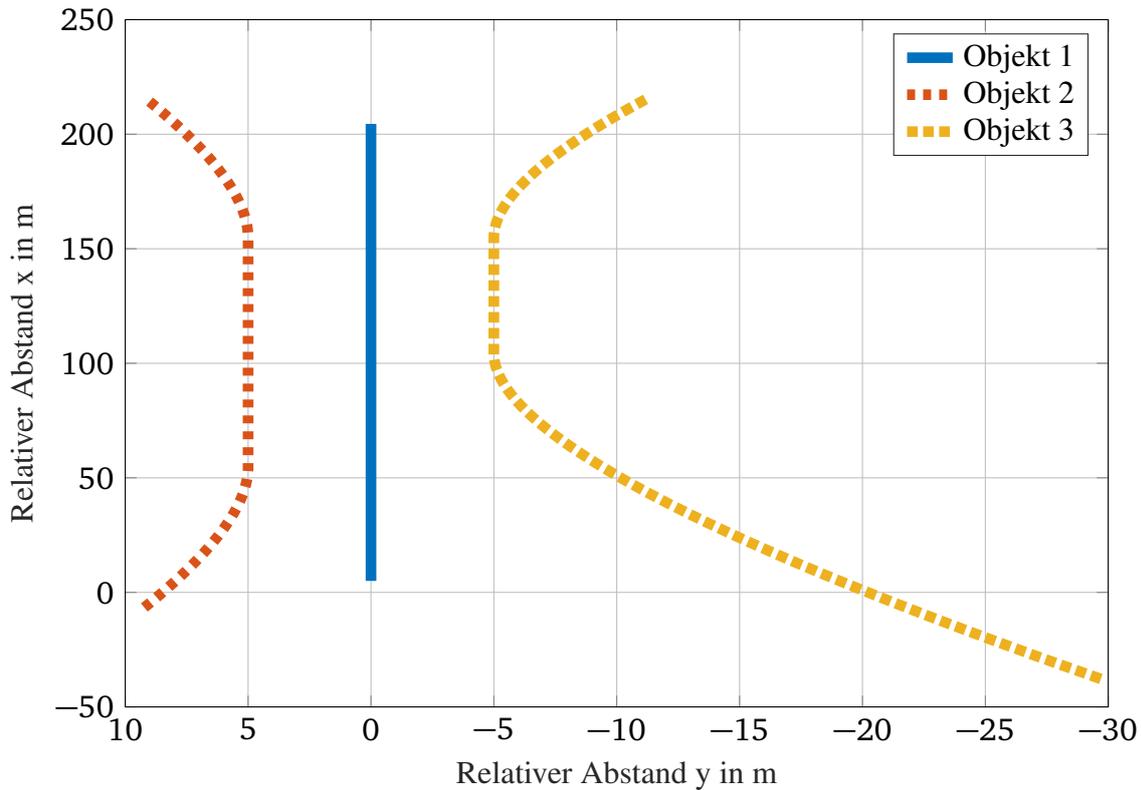


Abbildung 4-5: Formationsfahrt: Trajektorienverläufe der dynamischen Objekte. Die Verzögerung/Beschleunigung der Objekte 2 und 3 ist $1\frac{m}{s^2}$. Der Winkel der Beschleunigung des linken Objekts beträgt 20° . Der Winkel der Beschleunigung des rechten Objekts beträgt 30° . Die minimalen Abstände der Objekte 2 und 3 zum mittleren Objekt beträgt jeweils 5 m

$$\begin{aligned} \dot{y}_{2,t=0} &= \dot{y}_{1,t=1} - a_l \cdot \sin(\alpha_l) \cdot t_1 \\ y_{2,t=0} &= y_{l,min} - \frac{1}{2}a_l \cdot \sin(\alpha_l) \cdot t_1^2 - \dot{y}_{2,t=0} \cdot t_1 \end{aligned} \quad (4-18)$$

Die resultierenden Bewegungsgleichungen für Objekt 2 sind im Anhang A.3 notiert.

Die Gleichungen des dritten Objekts ähneln denen des zweiten Objekts. Der Start-Zustand von Objekt 3 wird berechnet durch:

$$\begin{aligned} \dot{x}_{3,t=0} &= \dot{x}_{1,t=2} + a_r \cdot \cos(\alpha_r) \cdot t_2 \\ x_{3,t=0} &= x_{1,t=2} + \frac{1}{2}a_r \cdot \cos(\alpha_r) \cdot t_2^2 - \dot{x}_{3,t=0} \cdot t_2 \end{aligned} \quad (4-19)$$

Tabelle 4-1: Parameter des Testfalls 1: Formationsfahrt

Parameter	Beschreibung
T	in s. Dauer eines Einzelmanövers. Gesamtmanöverdauer = $4 \cdot T$
α_l	in $^\circ$. Winkel zwischen x-Achse und Beschleunigungsrichtung Objekt 2
α_r	in $^\circ$. Winkel zwischen x-Achse und Beschleunigungsrichtung Objekt 3
a_l	in m/s^2 . Beschleunigung Objekt 2
a_r	in m/s^2 . Beschleunigung Objekt 3
v_1	in m/s. x-Geschwindigkeit Objekt 1
$y_{l,min}$	in m. Minimaler Abstand zwischen Objekt 1 und 2
$y_{r,min}$	in m. Minimaler Abstand zwischen Objekt 1 und 3
dT	in s. Abtastzyklus Radarsensor
NObjects	Gesamtzahl aller Objekte und Cluster pro Messzyklus
NStaticObjects	Anzahl statischer Objekte pro Messzyklus
FoVAngle	in $^\circ$. Öffnungswinkel des Radar-Sichtfelds
Seed	Seed für den Random Number Generator in Matlab

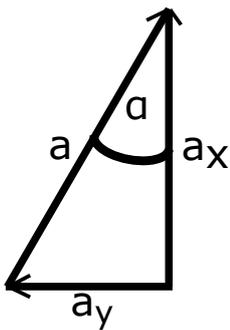


Abbildung 4-6: Vektoren der Beschleunigungen

$$\begin{aligned}
 \dot{y}_{3,t=0} &= \dot{y}_{1,t=2} - a_r \cdot \sin(\alpha_r) \cdot t_2 \\
 y_{3,t=0} &= y_{r,min} - \frac{1}{2} a_r \cdot \sin(\alpha_r) \cdot t_2^2 - \dot{y}_{3,t=0} \cdot t_2
 \end{aligned}
 \tag{4-20}$$

Die im zweiten Schritt erstellten Bewegungsgleichungen von Objekt 3 sind in Anhang A.4 zu finden.

Die Zustände der statischen Objekte werden zufällig erzeugt. Die statischen Objekte haben keine Geschwindigkeit. Die außerdem erzeugten Clutter-Signale werden zu jedem Messzeitpunkt zufällig erzeugt und weisen ebenfalls nur eine Position, aber keine Geschwindigkeit auf.

4.3.3 Testfall 2: Autobahnfahrt

Der zweite Testfall wird ‘‘Autobahnfahrt‘‘ genannt. Auch hier befinden sich erneut drei dynamische und beliebig viele statische und Clutter-Objekte vor dem Radarsensor. In diesem Fall bewegen sich

zwei dynamische Objekte vom Sensor weg (Objekte 1 und 3) und ein Objekt bewegt sich auf den Sensor zu (Objekt 2). Das linke der beiden sich entfernenden Objekte bewegt sich mit der doppelten Relativgeschwindigkeit gegenüber dem rechten Objekt. Die Trajektorien der drei Objekte verlaufen parallel. Die konstante Distanz zwischen den Objekten ist parametrierbar, ebenso wie die Geschwindigkeit der Objekte. Ziel dieses Testfalls ist die Untersuchung einer autobahnähnlichen Situation. Das mittlere Fahrzeug führt einen Überholvorgang eines langsameren Fahrzeugs aus, während ihm ein weiteres Fahrzeug auf dem Gegenfahrstreifen entgegenkommt. Ein Tracking-Algorithmus hat die Aufgabe, die drei beobachteten dynamischen Objekte nicht zu verschmelzen, auch wenn die Objekte eine geringe Distanz zueinander haben. Eine Trennung der Objekte ist möglich, da die Objekte eine unterschiedliche Geschwindigkeit aufweisen. Mithilfe der einstellbaren Distanzen und Geschwindigkeiten lassen sich Grenzfälle der Objekttrennung und -verschmelzung untersuchen.

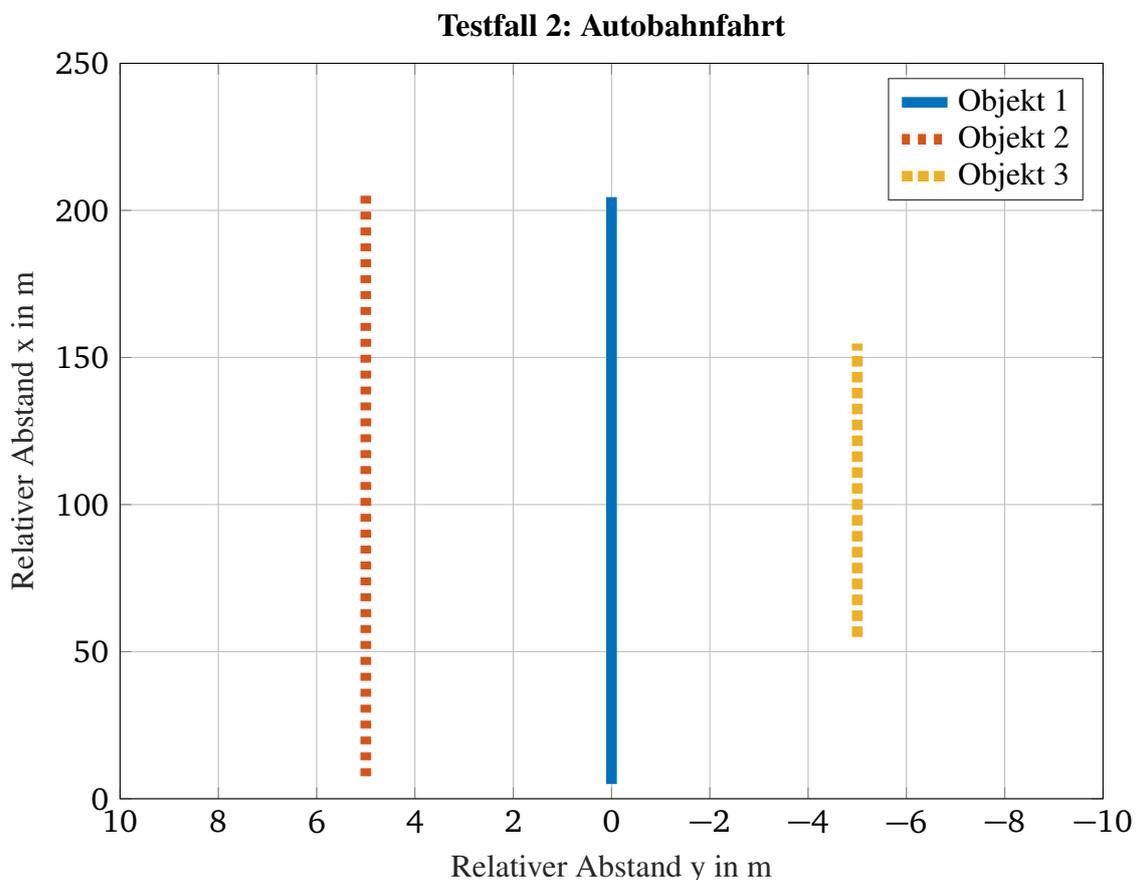


Abbildung 4-7: Autobahnfahrt: Trajektorienverläufe der dynamischen Objekte. Die minimalen Abstände der Objekte 2 und 3 zum mittleren Objekt betragen jeweils 5 m

Abbildung 4-7 zeigt die Ground-Truth-Trajektorienverläufe der drei dynamischen Objekte des Testfalls. Da sich Objekt 3 langsamer bewegt als Objekt 1 hat dieses zu Beginn des Testfalls einen größeren Abstand zum Radarsensor, der sich erneut bei Position (0,0) befindet. Der Testfall ist in vier Abschnitte unterteilt. Die Geschwindigkeiten der Objekte und die Startposition von Objekt 3 sind so gewählt, dass alle drei dynamischen Objekte zu Beginn des dritten Abschnitts den gleichen Abstand

zum Radarsensor haben. Damit findet der Überholvorgang gleichzeitig mit dem Passieren des entgegenkommenden Fahrzeugs statt.

Tabelle 4-2 listet alle Parameter des Testfalls auf.

Tabelle 4-2: Parameter des Testfalls 2: Autobahnfahrt

Parameter	Beschreibung
T	in s. Dauer eines Einzelmanövers. Gesamtmanöverdauer = $4 \cdot T$
v_1	in m/s. x-Geschwindigkeit Objekt 1 und 2. Geschwindigkeit von Objekt 3 ist direkt $v_1/2$
$y_{l,min}$	in m. Minimaler Abstand zwischen Objekt 1 und 2
$y_{r,min}$	in m. Minimaler Abstand zwischen Objekt 1 und 3
dT	in s. Abtastzyklus Radarsensor
NObjects	Gesamtzahl aller Objekte und Cluster pro Messzyklus
NStaticObjects	Anzahl statischer Objekte pro Messzyklus
FoVAngle	in °. Öffnungswinkel des Radar-Sichtfelds
Seed	Seed für den Random Number Generator in Matlab

Für Objekt 1 gelten die gleichen Bewegungsgleichungen wie in Testfall 1 und sind damit ebenfalls im Anhang A.2 zu finden.

Objekt 2 hat eine entgegen gerichtete Bewegungsrichtung. Es bewegt sich mit der gleichen Geschwindigkeit wie Objekt 1. Die Formeln 4-21 und 4-22 zeigen die Bewegungsgleichungen von Objekt 2:

$$\begin{aligned}
 \ddot{x}_2 &= 0 \\
 \dot{x}_2 &= -v_1 \\
 x_2 &= -v_1 \cdot t + 5 + v_1 \cdot 4 \cdot T
 \end{aligned}
 \tag{4-21}$$

$$\begin{aligned}
 \ddot{y}_2 &= 0 \\
 \dot{y}_2 &= 0 \\
 y_2 &= y_{l,min}
 \end{aligned}
 \tag{4-22}$$

Schließlich zeigen die Formeln 4-23 und 4-24 die Gleichungen von Objekt 3:

$$\begin{aligned}
 \ddot{x}_3 &= 0 \\
 \dot{x}_3 &= v_1 \\
 x_3 &= \frac{v_1}{2} \cdot t + 5 + v_1 \cdot T
 \end{aligned}
 \tag{4-23}$$

$$\begin{aligned}\ddot{y}_3 &= 0 \\ \dot{y}_3 &= 0 \\ y_3 &= y_{r,min}\end{aligned}\tag{4-24}$$

4.4 Erzeugen simulierter Radardaten

Die Wege zur Erzeugung von Testtrajektorien werden im vorherigen Unterkapitel beschrieben. Diese Trajektorien beschreiben die tatsächlich von den Objekten verfolgten Pfade. Die Positionen und Geschwindigkeiten werden in der Realität durch Sensoren erfasst, beispielsweise durch Radarsensoren. Die Genauigkeit dieser Messwerte hängt von der Sensorcharakteristik des eingesetzten Sensors ab. Diese wird durch Messfehler, die Auflösung und die Abtastrate des Sensors beeinflusst.

Vereinfachend wird angenommen, dass von jedem beobachteten Objekt nur eine Reflexion vom Radar empfangen wird. Die Generierung der simulierten Radarsignale erfolgt in zwei Schritten. Im ersten Schritt werden die Testtrajektorien äquidistant abgetastet. Die Abtastrate orientiert sich dabei an Werten, die von realen Radarsensoren bekannt sind. Abgetastet werden sowohl die Geschwindigkeiten, als auch die Positionen aller Objekte. Im zweiten Schritt werden diese Messwerte von einem gaußförmigen Rauschen überlagert. Die Größe des Rauschens orientiert sich dabei an den im Datenblatt des realen Radarsensors angegebenen Standardabweichungen der Positions- und Geschwindigkeitsmessung. Die Rauschwerte der Positionen in y-Richtung werden dabei über die Standardabweichung des Azimuthwinkels berechnet. Damit werden den y-Positionen weit entfernter Objekte ein größeres Rauschen überlagert, als nahe gelegenen Objekten. Auf diese Weise liegen zu jedem Abtastzeitpunkt verrauschte Zustände der drei dynamischen Objekte vor. Ergänzt werden diese durch statische Objekte, die zufällig im Beobachtungsbereich platziert werden. Ihre Geschwindigkeit beträgt in beide Richtungen 0 m/s. Auch die Zustände der statischen Objekte werden durch das oben vorgestellte Vorgehen verrauscht. Abschließend wird für jeden Messzeitpunkt eine frei parametrierbare Anzahl von Clutterpunkten erzeugt. Diese Clutterpunkte werden zufällig im Beobachtungsbereich platziert und zu jedem Messzeitpunkt unabhängig voneinander neu erzeugt.

5 Analyse des implementierten Tracking-Algorithmus

Nachdem alle Bausteine, die für eine Leistungsbewertung von Tracking-Algorithmen benötigt werden, vorgestellt sind, wird in diesem Kapitel eine Analyse des implementierten Tracking-Algorithmus durchgeführt. Einfluss auf die Bewertung der Tracking-Ergebnisse haben sowohl Parameter des verwendeten Algorithmus, als auch die realen oder simulierten Radardaten und die Parameter der eingesetzten Metriken und Testfälle.

Im ersten Teil dieses Kapitels wird die Wahl der Vergleichsmetriken getroffen. Anschließend werden die Schlüsselparameter der Tracking-Algorithmus vorgestellt und ihre Auswirkungen auf das Tracking-Ergebnis beschrieben. Die Anzahl der Schlüsselparameter schließt eine kombinatorische Untersuchung aller Parameter aus. Aus diesem Grund werden die Parameter im letzten Teil des Kapitels separat untersucht.

5.1 Wahl der Metriken für den Leistungsvergleich von Tracking-Algorithmen

Im vorherigen Kapitel werden eine Vielzahl von etablierten Tracking-Algorithmus-Metriken vorgestellt. Langjähriger Standard bei der Bewertung von Tracking-Ergebnissen ist die OSPA-T Metrik¹²⁵ und wird daher für die Leistungsbewertung ausgewählt. Neuere Metriken, wie beispielsweise die Metrik von Rahmathullah et al. stellen eine Verbesserung der OSPA-T-Metrik dar. Aus diesem Grund wird als zweite Bewertungsgrundlage die Metrik von Rahmathullah et al. verwendet. Damit spiegelt die OSPA-T-Metrik den aktuelle Standard und die Rahmathullah et al. den aktuellen Stand der Technik wider.

5.2 Vorstellung der Schlüsselparameter des Tracking-Algorithmus

Die Identifizierung der Schlüsselparameter erfolgt durch eine Analyse der Parameter des implementierten Tracking-Algorithmus. Dazu wird noch einmal Abbildung 3-6 in Kapitel 3 betrachtet. Diese zeigt die Basiselemente des implementierten Tracking-Algorithmus, die in den folgenden Unterkapiteln untersucht werden.

5.2.1 Schlüsselparameter der Sensordatengenerierung und -verarbeitung

Die Bewertung der Tracking-Qualität erfolgt auf Basis simulierter Radardaten. Grundlage sind die in Kapitel 4.3 vorgestellten Testfälle. Eine Veränderung der minimalen Distanzen bei gleichzeitig identischer Geschwindigkeit von unterschiedlichen Objekten hat Einfluss auf die Tracking-Qualität. Auch starke Verzögerungen und Beschleunigungen haben Auswirkungen auf die Bewertung des Algorithmus. Schlüsselparameter sind damit alle Parameter, die die Verläufe der Testtrajektorien beschreiben. Zusätzlich werden auch die Parameter des simulierten Radarsensors variiert. Damit hat auch die simulierte Genauigkeit, Abtastrate und Auflösung des Sensors Einfluss auf die Tracking-Qualität. Die Tabellen 5-1 und 5-2 listen die Schlüsselparameter der Sensordatengenerierung und -verarbeitung auf.

¹²⁵Scheel, A.: Expertengespräch Tracking-Algorithmus-Metriken (2017)

Tabelle 5-1: Schlüsselparameter der Sensordatengenerierung

Parameter	Standardwert	Beschreibung	Auswirkung auf das Tracking-Ergebnis
a_l	1	in m/s^2 . Beschleunigung Objekt links	Tracker kann Objekte mit starker Geschwindigkeitsänderung nicht verfolgen
a_r	1	in m/s^2 . Beschleunigung Objekt rechts	Tracker kann Objekte mit starker Geschwindigkeitsänderung nicht verfolgen
v_1	10	in m/s . x-Geschwindigkeit des mittleren Objekts	Bei Testfall 2 ist die Geschwindigkeit von Objekt rechts $\frac{v_1}{2}$. Durch eine kleine Geschwindigkeit v_1 entsteht eine kleine Geschwindigkeitsdifferenz zwischen Objekt Mitte und Objekt Rechts. Bei gleichzeitig geringem Abstand werden die Objekte beim Überholvorgang verschmolzen
$y_{l,min}$	5	in m. Minimaler Abstand zwischen Objekt Mitte und links	Bei identischer Geschwindigkeit und geringem Abstand werden die Objekte verschmolzen
$y_{r,min}$	4	in m. Minimaler Abstand zwischen Objekt Mitte und rechts	Bei identischer Geschwindigkeit und geringem Abstand werden die Objekte verschmolzen
SensorDistancstd- vxTraj	0,3	in m. Größe des Sensorrauschens der Position in x-Richtung	Bei stark verrauschten Positionsmessungen werden die Objekte nicht mehr verfolgt
SensorSpeedstdvx- Traj	0,0556	in m/s . Größe des Sensorrauschens der Geschwindigkeit in x-Richtung	Bei stark verrauschten Geschwindigkeitsmessungen werden die Objekte nicht mehr verfolgt

Tabelle 5-2: Schlüsselparameter der Sensordatengenerierung 2

Parameter	Standardwert	Beschreibung	Auswirkung auf das Tracking-Ergebnis
SensorSpeedstdvy-Traj	0,0556	in m/s. Größe des Sensorrauschens der Geschwindigkeit in y-Richtung	Bei stark verrauschten Geschwindigkeitsmessungen werden die Objekte nicht mehr verfolgt
AzimuthAccuracy-Traj	0,1	in °. Größe des Rauschens in der Azimuthbestimmung	Je größer das Azimuthwinkelrauschen, desto größer die Standardabweichungen der y-Positionen weit entfernter Objekte
SensorResolution-Pos	0,5	in m. Auflösung der Sensordaten Distanzmessung	Je größer, desto geringer die Auflösung der Distanzmessung in x- und y- Richtung
SensorResolution-Vel	0,1389	in m/s. Auflösung der Sensordaten Geschwindigkeitsmessung	Je größer, desto geringer die Auflösung der Geschwindigkeitsmessung in x- und y- Richtung

5.2.2 Schlüsselparameter der Clustergruppengenerierung

Im Schritt der Clustergruppengenerierung werden die Rohdaten der Cluster verarbeitet. Alle Cluster, die eine ähnliche Position und eine ähnliche Geschwindigkeit aufweisen, werden zu Gruppen zusammengefasst. Durch diesen Schritt werden Mehrfachreflexionen eines Objekts zu einer einzelnen Reflexion zusammengefasst. Die vorgestellte Implementierung der simulierten Radardatenerzeugung liefert standardmäßig nur eine Reflexion pro Objekt, aus diesem Grund ist eine Clustergruppengenerierung für die aktuelle Implementierung nicht nötig. Durch die Beibehaltung der Gruppengenerierung eröffnet sich allerdings die Möglichkeit, die Algorithmus-Trennfähigkeit zu untersuchen. Simulierte Cluster, die einen zu geringen Abstand oder eine zu geringe Geschwindigkeitsdifferenz aufweisen, werden in der Clustergruppengenerierung zusammengefasst und die Objekte somit verschmolzen. Tabelle 5-3 zeigt die Schlüsselparameter der Clustergruppengenerierung.

Tabelle 5-3: Schlüsselparameter der Clustergruppengenerierung

Parameter	Standardwert	Beschreibung	Auswirkung auf das Tracking-Ergebnis
MaxDistanceFor-Group	3	in m. Maximale Distanz zwischen Clustern um noch in dieselbe Gruppe eingeordnet zu werden	Wird die maximale Distanz der Cluster einer Clustergruppe zu groß gewählt, so besteht die Möglichkeit der Verschmelzung mehrerer Objekte. Kleine Werte führen dagegen zur Aufspaltung eines Objekts in mehrere Gruppen
MaxVRelDeviation	3	in m/s. Maximale Geschwindigkeitsdifferenz zwischen Clustern um noch in dieselbe Gruppe eingeordnet zu werden	Wird die maximale Geschwindigkeitsdifferenz der Cluster einer Clustergruppe zu groß gewählt, so besteht die Möglichkeit der Verschmelzung mehrerer Objekte. Kleine Werte führen dagegen zur Aufspaltung eines Objekts in mehrere Gruppen

5.2.3 Schlüsselparameter der Datenassoziation

Aufgabe der Datenassoziation ist die Verknüpfung der Radarmessungen eines neuen Abtastzyklus mit schon bekannten Objekten. Mithilfe dieser Zuordnung werden die neuen Zustände der beobachteten Objekte berechnet. Die Datenassoziation basiert auf der Verwendung der Mahalanobisdistanz. Diese bezieht die Kovarianzmatrix der Zustände in ihre Berechnung ein. Dadurch ergibt sich ein ellipsenförmiger Suchbereich um die prädierten Objektpositionen. Die Größe der Ellipse wird über die Inverse einer Chi-Quadrat-Verteilung festgelegt¹²⁶. Alle Messwerte, die innerhalb dieser Ellipse liegen, werden dem entsprechenden Objekt zugeordnet. Der Schlüsselparameter ist in diesem Fall die gewünschte Wahrscheinlichkeit, dass der nächste Messwert des Objekts in die Ellipse fällt. Tabelle 5-4 zeigt den Schlüsselparameter der Datenassoziation.

¹²⁶Grisetti, G. et al.: Robotics 2 - Data Association (2009) S.7

Tabelle 5-4: Schlüsselparameter der Datenassoziation

Parameter	Standardwert	Beschreibung	Auswirkung auf das Tracking-Ergebnis
ValidationGatePercentage	99	in %. Größe des Suchbereichs um prädizierte Objektposition. xx % aller zu erwartenden Messwerte liegen innerhalb des Suchbereichs.	Bei zu kleinen Werten besteht die Gefahr, dass Cluster eines Objekts nicht innerhalb des Suchbereichs liegen und damit nicht zugeordnet werden

5.2.4 Schlüsselparameter der Objekt-Tracker-Aktualisierung

Während dieses Programmschritts werden die Zustände der einzelnen Objekt-Tracker aktualisiert. Hierfür wird in einem ersten Schritt überprüft, welchen Objekt-Trackern während der Datenassoziation neue Messungen ihres verfolgten Objekts zugeordnet worden sind. Wurde dem Tracker kein neues Radar Cluster zugeordnet, so wird ein Zähler inkrementiert. Überschreitet dieser Zähler eine Schwelle, so wird das Objekt aus der Objektliste gelöscht. Diese Schwelle hat damit Einfluss auf die Lebensdauer von Objekten, die über mehrere Messungen nicht mehr erfasst wurden. Je nach Wert wird damit Einfluss auf die Kardinalität des beobachteten Szenarios genommen. Die Parameter "mausNN" und "mausNm" werden kombiniert betrachtet. Sie bilden die Basis des m/N-Verfahrens zum Filtern von Clutter aus den Radarsignalen. Schlüsselparameter ist das Verhältnis beider Werte. Im zweiten Schritt der Objekt-Tracker Aktualisierung werden die prädizierten Tracker-Zustände mit den neuen Radardaten korrigiert. Diese Filterung und Prädiktion erfolgt durch Einsatz eines Kalman-Filters. Für die Funktionsweise dieses Filters sind Definitionen einiger Matrizen notwendig. Für die Wahl der Kovarianzmatrizen des Modells bzw. des Kalman-Filters existiert kein bekanntes Parametrierungsverfahren. Ein Ausprobieren verschiedener Matrizen ist daher unerlässlich. Tabellen 5-5 und 5-6 zeigen die Schlüsselparameter der Objekt-Tracker Aktualisierung. Die Standardwerte der Kalman Matrizen sind in Kapitel 3.4.1 dargestellt.

Tabelle 5-5: Schlüsselparameter der Objekt-Tracker Aktualisierung 1

Parameter	Standardwert	Beschreibung	Auswirkung auf das Tracking-Ergebnis
maxNundetected	10	Anzahl von fehlenden Messungen bis Objekt aus Objektliste gelöscht wird	Bei großen Werten werden nicht mehr vorhandene Objekte weiterhin in der Objektliste behalten. Bei kleinen Werten werden die Objekte schon bei wenigen Fehlmessungen gelöscht, obwohl sie weiterhin existent sind
mausNm/mausNN	5/7	Wahl der Werte für das m/N-Verfahren	Verhältnisse nahe 1 tolerieren Fehlmessungen weniger. Verhältnisse nahe 0 filtern Clutter schlechter heraus.
NObjectTrackers	15	Anzahl Objekt-Tracker	Ist die Anzahl der Objekt-Trackern kleiner als die Anzahl der zu beobachtenden Cluster, so wird keine Verfolgung von weit entfernten Objekten durchgeführt. Nahe Objekte haben in diesem Fall Priorität

Tabelle 5-6: Schlüsselparameter der Objekt-Tracker Aktualisierung 2

Kalman-Filter Matrizen		
Matrix	Beschreibung	Auswirkung auf das Tracking-Ergebnis
P	Kovarianzmatrix des Filters. Legt die Genauigkeit der Zustände des Filters fest	Große Werte bedeuten größere erwartete Ungenauigkeiten
Q	Kovarianzmatrix des Modells. Legt die Genauigkeit des Prozessmodells fest	Große Werte ermöglichen die Verfolgung von Objekten mit nicht-konstanter Geschwindigkeit. Gleichzeitig nimmt die Prädiktionsgenauigkeit des Filters ab. Durch kleine Werte wird dem Prozessmodell eher vertraut, allerdings werden Objekte mit starker Geschwindigkeitsänderung nicht verfolgbar
R	Kovarianzmatrix des Sensors. Legt die Genauigkeit des Sensors fest	Große Werte der Einträge zeugen von einer geringeren Genauigkeit der Messergebnisse. Bei kleinen Werten wird die Sensorwerte eher vertraut
A	Zustandsübergangsmatrix. Beschreibt das Verhalten des Modells	Die Modellierung des Prozess hat Einfluss auf die Prädiktion der Zustände. Aufwendigere nichtlineare Prozessmodelle erfordern größere Anstrengungen und Rechenzeiten zum Lösen der Gleichungen

5.2.5 Schlüsselparameter der Löschung von Tracker-Duplikaten

In diesem Prozessschritt werden alle initialisierten Objekt-Tracker auf Duplikate untersucht. Duplikate mit einer kürzeren Lebenszeit werden gelöscht. Die Identifizierung von Tracker-Duplikaten erfolgt über den Vergleich ihrer Zustände. Hierzu werden die Differenzen zwischen den Positionen und Geschwindigkeiten der Objekt-Tracker-Zustände berechnet. Liegen diese unterhalb einer parametrierbaren Schwelle, so handelt es sich um Duplikate. Die Schlüsselparameter sind damit die genannten Schwellen. Diese sind in Tabelle 5-7 aufgelistet.

Tabelle 5-7: Schlüsselparameter der Löschung von Tracker-Duplikaten

Parameter	Standardwert	Beschreibung	Auswirkung auf das Tracking-Ergebnis
MaxHistoryDistance	2	Maximale Distanz zwischen den Positionen zweier Objekt-Tracker	Bei großem Wert werden Tracker von separaten aber nahe beieinander liegenden Objekten gelöscht
MaxHistoryVRel-Deviation	2	Maximale Geschwindigkeitsdifferenz zweier Objekt-Tracker	Bei großem Wert werden Tracker von separaten aber nahe beieinander liegenden Objekten mit gleichzeitig geringer Geschwindigkeitsdifferenz gelöscht

5.2.6 Schlüsselparameter der Tracking-Algorithmus-Metriken

Abschließend werden die Schlüsselparameter der eingesetzten Tracking-Algorithmus-Metriken vorgestellt. Auch die Wahl dieser Parameter hat Einfluss auf die finale Bewertung der Algorithmen. Für die Auswertung werden zwei verschiedene Metriken verwendet. Ihre Schlüsselparameter sind in Tabelle 5-8 dargestellt.

Tabelle 5-8: Schlüsselparameter der Tracking-Algorithmus-Metriken

Parameter	Standardwert	Beschreibung	Auswirkung auf das Tracking-Ergebnis
OSPATC	5	Cutoff-Parameter der OSPA-T Metrik	Je größer dieser Wert, desto größer die Strafe bei Kardinalitätsfehlern
OSPATP	1	Parameter P der OSPA-T Metrik	Je größer dieser Wert, desto größer die Strafe bei Messfehlern
RahmC	5	Cutoff-Parameter der Rahmathullah et al. Metrik	Je größer dieser Wert, desto größer die Strafe bei Kardinalitätsfehlern
RahmP	1	Parameter P der Rahmathullah et al. Metrik	Je größer dieser Wert, desto größer die Strafe bei Messfehlern
RahmGamma	1	Strafe für TrackSwitch	Je größer dieser Wert, desto größer die Strafe bei sich ändernden Benennungen der Objekte während der Objektverfolgung
RahmMaxIter	20	Anzahl der Iterationen für den Optimierungsprozess	Je größer dieser Wert, desto größer die erforderliche Rechenzeit. Größere Werte führen zu genaueren Ergebnissen

5.3 Ergebnisse der Tracking-Algorithmus-Analyse

Auf Basis der vorgestellten Testfälle wird die Leistungsbewertung der Tracking-Algorithmen durchgeführt. Durch die große Anzahl an Einflussgrößen auf das Tracking-Ergebnis, wird jeder Parameter separat untersucht und alle restlichen Parameter auf den Standardwerten belassen. Zur Vermeidung von statistischen Einflüssen auf die Bewertungszahlen werden die Berechnungen mehrfach wiederholt und alle Ergebnisse gemittelt.

Vor Beginn der Parametervariation werden die Ergebnisse der Metriken bei Standardparametrierung berechnet. Diese sind in Tabelle 5-9 dargestellt.

Tabelle 5-9: Ergebnisse der Metriken bei Standardparametrierung

	OSPA-T Metrik	Rahmathullah et al. Metrik
Testfall 1	83,68	455,08
Testfall 2	70,9	441,15

Der Ablauf des Objekt-Trackings ist grafisch darstellbar und wird als Video-Datei abgespeichert. Im folgenden wird ein Einzelbild einer Objektverfolgung mit Standardparametrierung im Testfall 1 dargestellt. Abbildung 5-1 zeigt ein Einzelbild des Objekt-Trackings während der ersten Phase von Testfall 1. Das linke Objekt (Nr. 5) ist annähernd auf Höhe des mittleren Objekts (Nr. 1). Das rechte Objekt (Nr. 6) befindet sich noch in der Verzögerungsphase. Bild A-11 in Anhang A.5 zeigt die beobachtete Situation ein wenig später. Der Radarsensor befindet sich jeweils auf Position (0,0). Video-Dateien der Testfälle mit Standardparametrierung sind in einem digitalen Anhang zu dieser Master Thesis als "Testfall1_x1.avi" und "Testfall2_x1.avi" zu finden.

Die OSPA-T-Metrik besitzt die Besonderheit, abseits von rein numerischen Ergebnissen auch grafische Ergebnisse zu erzeugen. Da eine grafische Darstellung aller Ergebnisse der Parametervariationen nicht zielführend ist, werden die erzeugbaren Grafiken beispielhaft bei Standardparametrierung für Testfall 1 gezeigt. Abbildung 5-2 zeigt die Anzahl der Sprünge der Kardinalität während des Objekt-Trackings.

Auf Abbildung 5-3 wird die reale und prädizierte Anzahl der Objekte in der beobachteten Szene zu jedem Messzeitpunkt angezeigt. Es zeigt sich die erwartete Verzögerung zwischen dem Eintauchen der Objekte in den beobachteten Bereich und die Detektion durch den Tracking-Algorithmus.

Auf Abbildung 5-4 ist die grafische Bewertung der Tracking-Qualität auf Basis der OSPA-T Metrik dargestellt. Die kummulierte Gesamtbewertung entsteht durch Addition der Distanz- und Kardinalitäts-Ergebnisse.

Abbildung 5-5 zeigt die Trajektorienverläufe der Objekte im Testfall 1. Zusätzlich sind die Stellen, an denen es zu Kardinalitätssprüngen kommt, markiert. Neben den Markierungen wird die Größe des Sprungs notiert, der sowohl positiv, als auch negativ sein kann. Zusätzlich sind auch die verrauschten Positionen der statischen Objekte zu sehen.

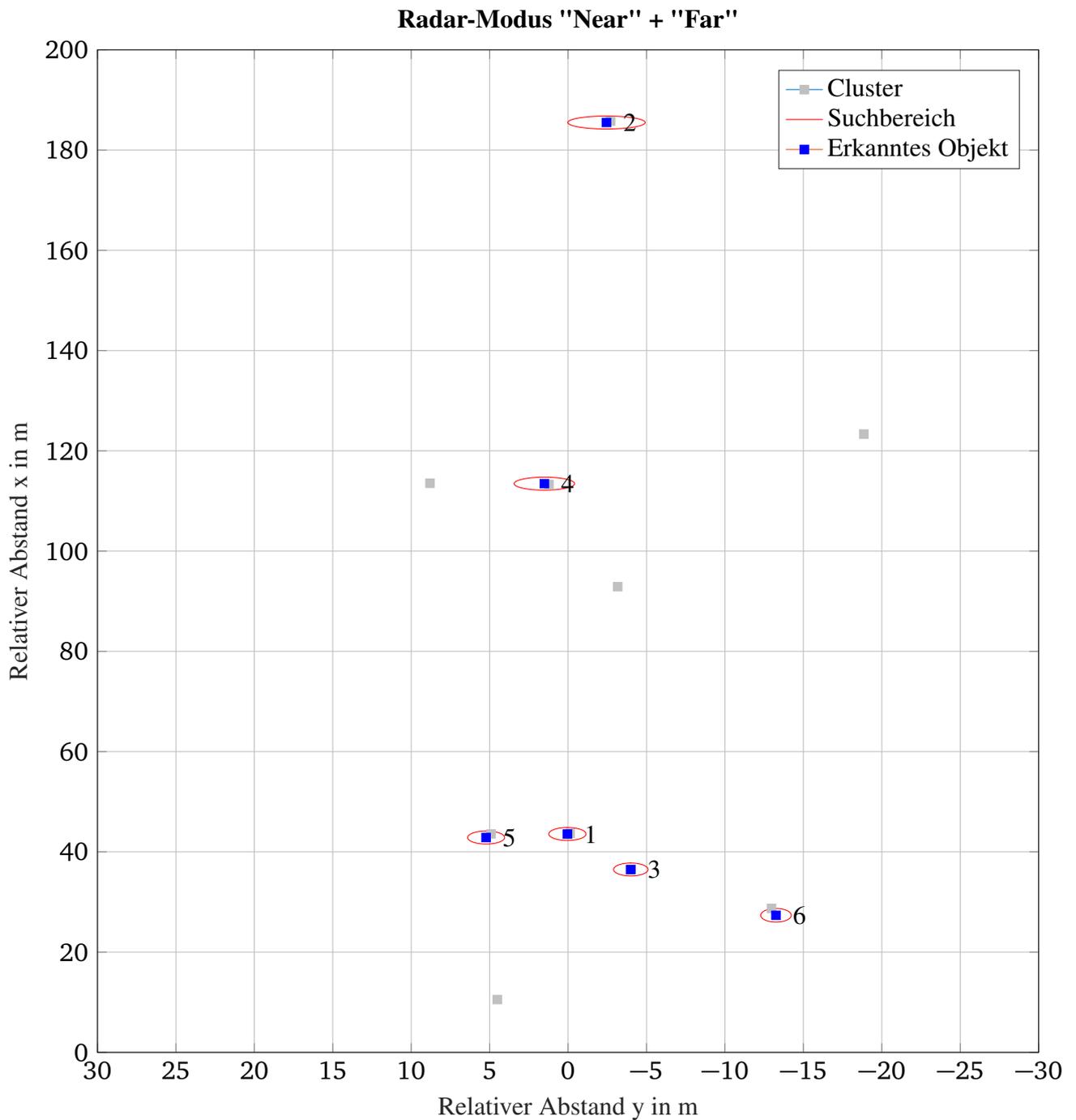


Abbildung 5-1: Einzelbild aus dem Objekt-Tracking. Objekte 2, 3 und 4 sind statisch. Objekt 1 ist das mittlere und die Objekte 5 und 6 sind die jeweils rechten bzw. linken Objekte von Testfall 1. Alle verbleibenden grauen Kästchen sind Clutter.

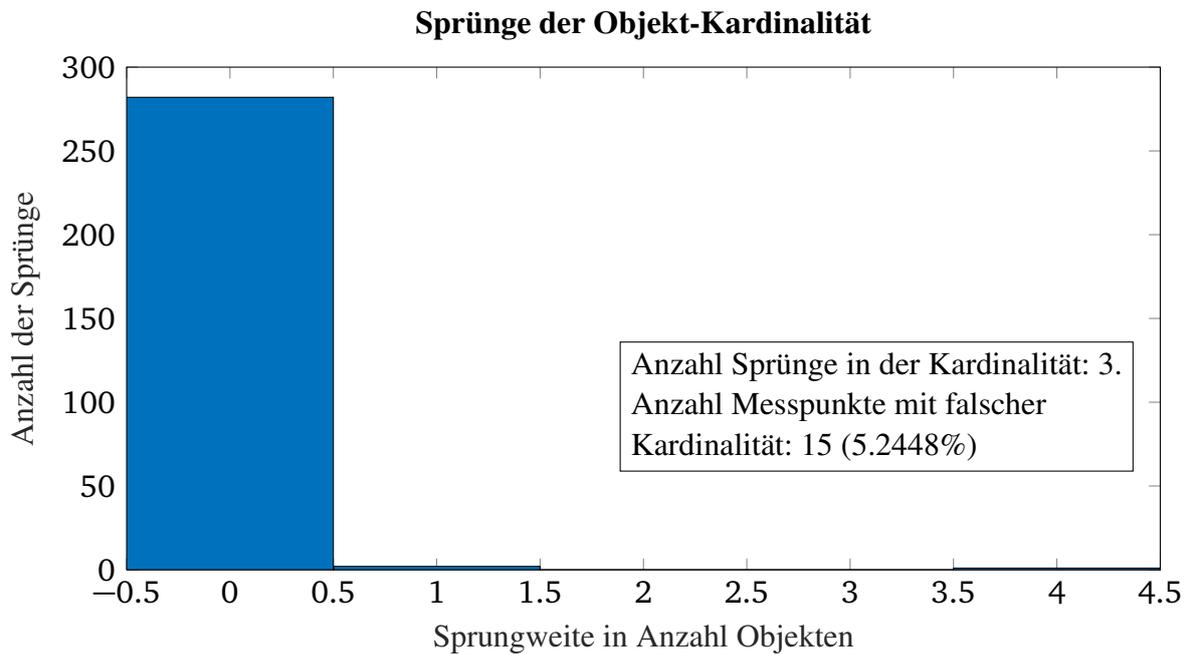


Abbildung 5-2: OSPA-T Metrik: Anzahl Kardinalitätssprünge. Zeigt Informationen über die Kardinalitätssprünge während des Objekt-Trackings an. Zusätzlich werden weitere Informationen zur Qualität des Objekt-Trackings dargestellt.

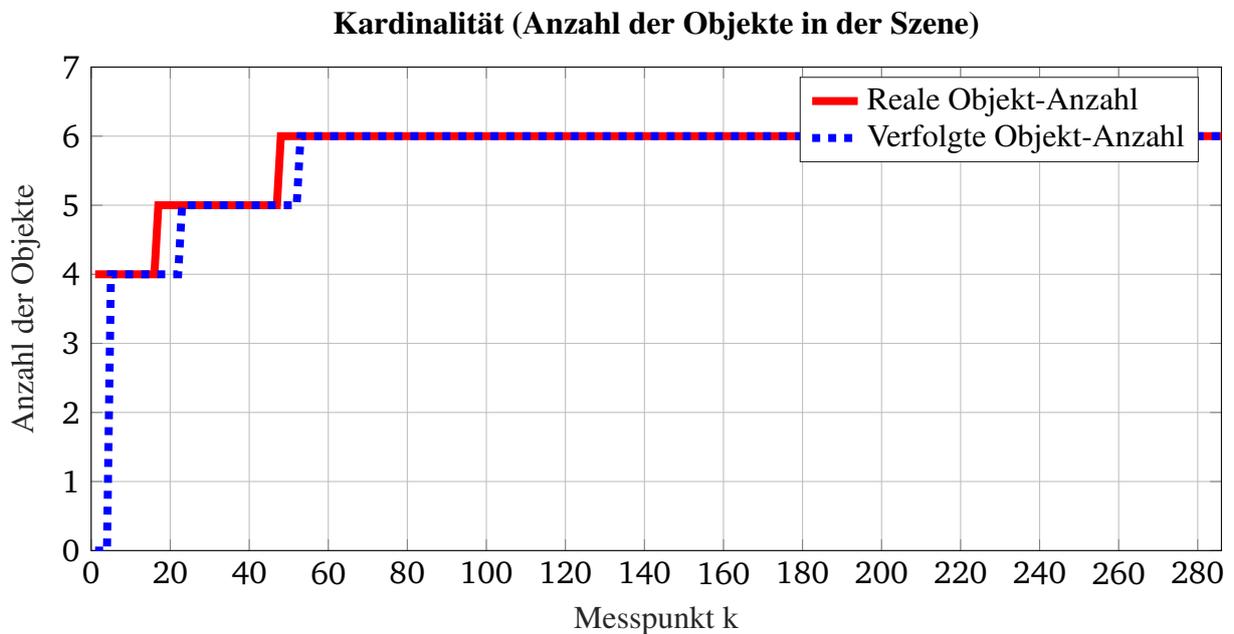


Abbildung 5-3: OSPA-T Metrik: Übersicht über Kardinalität der Objekte. Zeigt die reale und prädizierte Anzahl von Objekten während Tracking-Verlaufs

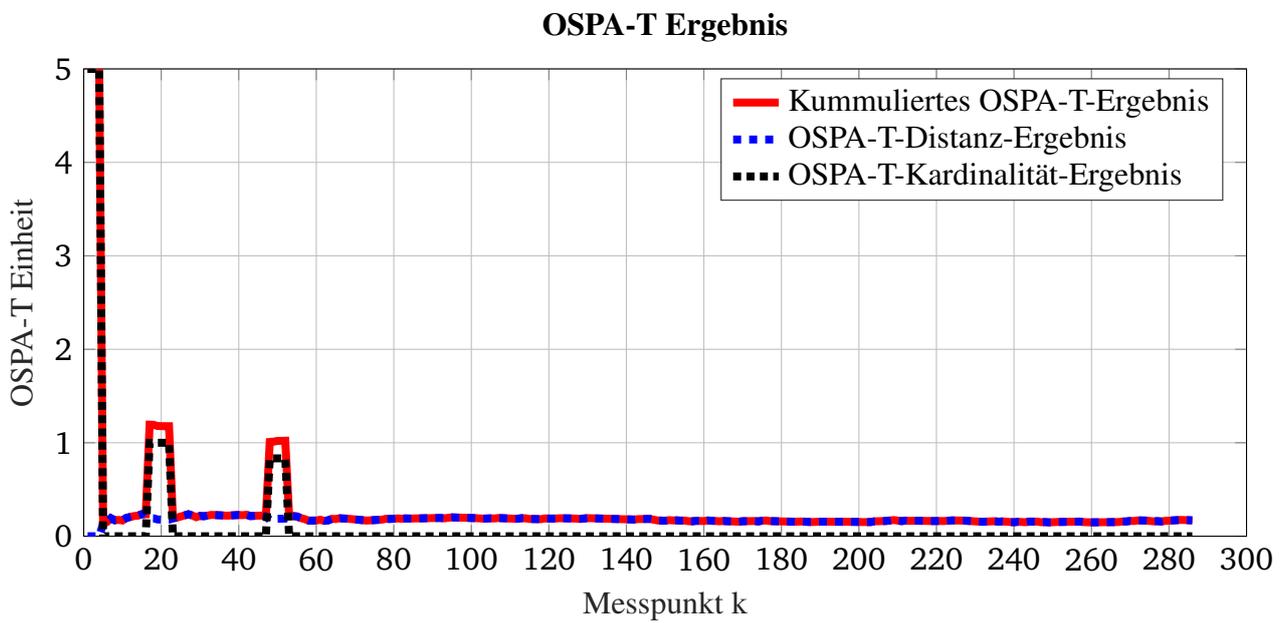


Abbildung 5-4: OSPA-T Metrik: Grafische Darstellung der OSPA-T Bewertung. Zeigt für jeden Messzeitpunkt die Verläufe der beiden Werte für die Berechnung der OSPA-T Bewertung an. Zu jedem Zeitpunkt werden die beiden Verläufe addiert und am Ende über alle Messzeitpunkte kummuliert.

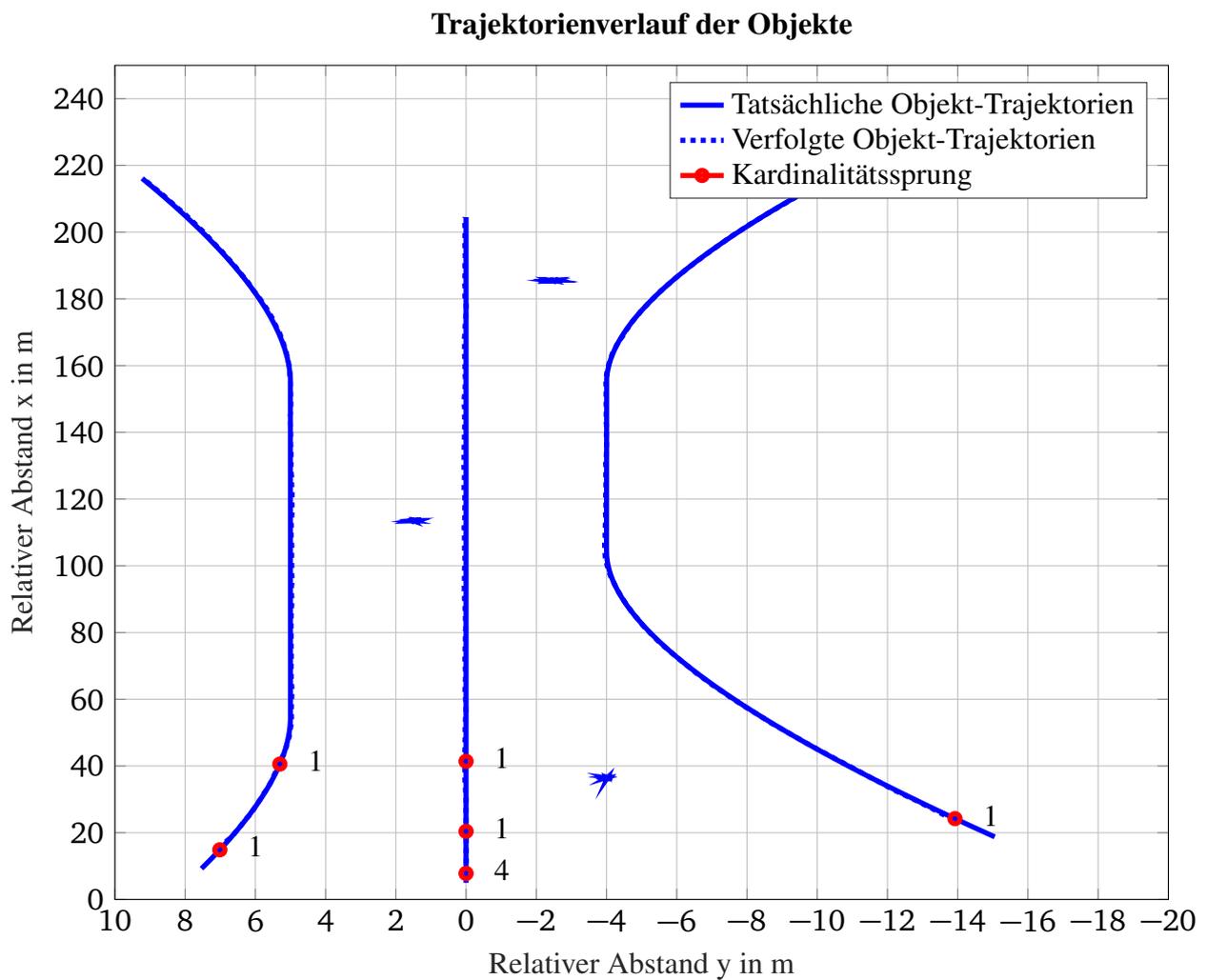


Abbildung 5-5: OSPA-T Metrik: Übersicht Trajektorienverlauf mit Kardinalitätssprüngen. Zeigt die Trajektorienverläufe der Objekte an. Zusätzlich werden die Stellen, an denen Kardinalitätssprünge auftreten markiert und die Größe des Sprungs daneben notiert. Weiterhin zu sehen sind die verrauchten Positionen der drei statischen Objekte.

5.3.1 Parametervariation der Sensordatengenerierung

Hier wird der Einfluss der Parametervariation bei der Sensordatengenerierung untersucht. Durch die Variation entstehen verschiedene Objekttrajektorien. Nicht nur die Trajektorienverläufe haben Einfluss auf die Sensordatengenerierung, sondern auch die Charakteristika der Radarsensoren. Diese Charakteristika werden im zweiten Teil dieses Unterkapitels variiert.

Parametervariation der Trajektoriengenerierung

Begonnen wird mit der Variation der Beschleunigungen bzw. Verzögerungen der Objekte 2 und 3 in Testfall 1. Die Objekte in Testfall 2 weisen eine konstante Geschwindigkeit auf, daher wird dieser Fall nicht untersucht. Tabelle 5-10 zeigt die Ergebnisse der Bewertung bei Variation der Beschleunigungen von Objekt 2 und 3 in Testfall 1. Ein beispielhaftes Video der Tracking-Verläufe bei starken Objektverzögerungen sind im digitalen Anhang unter "Testfall1_acceleration_x1.avi" zu finden.

Tabelle 5-10: Ergebnisse der Metriken bei variiertes Objektbeschleunigung im Testfall 1

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
$a_l = 0,5 \text{ m/s}^2$	79,49	450,91
$a_l = 2 \text{ m/s}^2$	136,42	509,86
$a_l = 3 \text{ m/s}^2$	142,98	669,87
$a_l = 4 \text{ m/s}^2$	163,61	697,09
$a_l = 5 \text{ m/s}^2$	164,59	693,87
$a_r = 0,5 \text{ m/s}^2$	79,09	447
$a_r = 2 \text{ m/s}^2$	133,16	532,24
$a_r = 3 \text{ m/s}^2$	166,54	754,11
$a_r = 4 \text{ m/s}^2$	177,26	740,16
$a_r = 5 \text{ m/s}^2$	183,05	737,66

Die Bewertungszahlen der Tabelle zeigen, dass sich bei höheren Objektbeschleunigungen bzw. -verzögerungen ein schlechteres Tracking-Ergebnis einstellt. Die Beschleunigungen der Objekte sind höher als von den Matrizen des Kalman-Filters erwartet. Die Bewertungszahlen steigen mit erhöhter Beschleunigung an, bis sie ihr Maximum erreichen. Im Bereich des Maximums werden die Objekte in den Beschleunigungs- bzw. Verzögerungsbereichen der Trajektorien nicht mehr detektiert. Aus diesem Grund hat eine Erhöhung der Beschleunigungswerte keinen weiteren Einfluss auf das Bewertungsergebnis.

Wird für den Fall $a_r = 5 \text{ m/s}^2$ die erwartete Störbeschleunigung "TargetModellaccstdv" des Bewegungsmodells (Tabelle 3-1) ebenfalls auf 5 m/s^2 gesetzt, so verbessert sich die Tracking-Qualität. In diesem Fall wird der Kalman-Filter so parametrisiert, dass auch Objekte mit starker Geschwindigkeitsänderung detektiert und verfolgt werden. Die Bewertungszahlen sind in

Tabelle 5-11 zu sehen. Videos dieser Untersuchung sind im digitalen Anhang unter “Testfall1_acceleration5ms2_altesModell_x1.avi“ und “Testfall1_acceleration5ms2_neuesModell_x1.avi“ zu finden.

Tabelle 5-11: Ergebnisse der Metriken bei variiertes Objektbeschleunigung im Testfall 1 mit neuem Modell

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
$a_r = 5 \text{ m/s}^2$	82,37	483,54

Ein Bewegungsmodell mit dauerhaft vergrößerter erwarteter Beschleunigung verschlechtert die Genauigkeit des Tracking-Ergebnisses bei konstanter Geschwindigkeit des Objekts. Diese Untersuchung zeigt, dass der Einsatz von mehreren parallelen Bewegungsmodellen, wie beispielsweise beim IMM Tracking-Algorithmus, gerechtfertigt ist.

Wird die standardmäßige Beschleunigung der Objekte auf $0,5 \text{ m/s}^2$ verringert, so verbessert sich das Tracking-Ergebnis. Grund hierfür ist die Annahme konstanter Geschwindigkeit des Objektbewegungsmodells.

Für Testfall 2 ist die Geschwindigkeit des mittleren Fahrzeugs essenziell. Die Geschwindigkeit des rechten Objekts beträgt die Hälfte des mittleren Objekts. Daraus resultiert eine immer kleinere Differenzgeschwindigkeit, wenn die Geschwindigkeit des mittleren Objekts verringert wird. Erreicht die Differenzgeschwindigkeit ein Minimum, werden die Objekte verschmolzen. Für die Auswertung dieser Untersuchung werden die minimalen Distanzen auf 1 m reduziert. Einfluss auf das Bewertungsergebnis haben damit nur die Geschwindigkeitsdifferenzen der Objekte. Tabelle 5-12 zeigt die Bewertungszahlen bei variiertes mittlerer Objektgeschwindigkeit.

Tabelle 5-12: Ergebnisse der Metriken bei variiertes mittlerer Objektgeschwindigkeit im Testfall 2

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
$v_1 = 10 \text{ m/s}$	72,5	452,84
$v_1 = 8 \text{ m/s}$	82,13	530,2
$v_1 = 7 \text{ m/s}$	90,43	579,56
$v_1 = 6 \text{ m/s}$	298,16	543,11
$v_1 = 5 \text{ m/s}$	502,97	739,75

Die Verringerung der mittleren Objektgeschwindigkeit auf Werte bis 7 m/s verändert die Bewertungszahlen nur gering. Ab 6 m/s und damit einer Differenzgeschwindigkeit von 3 m/s werden die Objekte teilweise verschmolzen. Teilweise deshalb, weil zusätzliches Rauschen in den Geschwindigkeitsmessungen vorhanden ist. Ab 5 m/s werden die sich überholenden Objekte verschmolzen und die Bewertungszahlen erhöhen sich erheblich. Hier ist die Geschwindigkeitsdifferenz so gering, dass ein Trennen der Objekte nicht möglich ist. Ein beispielhaftes Video dieser Untersuchung ist im digi-

talen Anhang unter “Testfall2_slowspeed_x1.avi“ zu finden.

Bei der nächsten Untersuchung werden die minimalen Distanzen während der Parallelfahrt der Objekte in Testfall 1 variiert. Durch die eingestellte Parametrierung der Clustergruppengenerierung werden alle Objekte, die eine Distanz von kleiner 3 m und einer ähnlichen Geschwindigkeit aufweisen, miteinander verschmolzen. Die Ergebnisse dieser Analyse sind in Tabelle 5-13 notiert.

Tabelle 5-13: Ergebnisse der Metriken bei variierten minimalen Objektdistanzen im Testfall 1

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
$y_{l,min} = 4 \text{ m}$	86,95	488,16
$y_{l,min} = 3 \text{ m}$	234,82	766,6
$y_{l,min} = 2 \text{ m}$	428,84	1189,08
$y_{l,min} = 1 \text{ m}$	477,12	1098,76
$y_{r,min} = 3 \text{ m}$	160,39	668,72
$y_{r,min} = 2 \text{ m}$	293,91	903,32
$y_{r,min} = 1 \text{ m}$	335,18	929,71

Bis zur Grenze von 3 m werden die Objekte eindeutig voneinander getrennt. Wird die eingestellte Distanz auf 3 m gesetzt, so ergibt sich eine teilweise Objektverschmelzung. Bei Werten kleiner als 3 m werden die Objekte vereint und die Bewertungszahlen steigen stark an. Ein beispielhaftes Video dieser Untersuchung ist im digitalen Anhang unter “Testfall1_distance_x1.avi“ zu finden.

Parametervariation der Sensorcharakteristika

Für die Sensordatengenerierung werden die Charakteristiken des simulierten Radarsensors variiert. Hierzu zählen beispielsweise die Genauigkeit, die Auflösung und die Abtastrate. Diese Parameter haben Einfluss auf die Qualität der Sensordaten und damit auf die Möglichkeiten der Objektverfolgung. Die Einflüsse dieser Charakteristika werden im folgenden untersucht.

Begonnen wird mit der Variation des Sensorrauschens in Testfall 1 und 2. Die Tabellen 5-14 und 5-15 zeigen die Ergebnisse.

Tabelle 5-14: Ergebnisse der Metriken bei variiertem Sensorrauschen im Testfall 1

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
SensorDistancestdvxTraj = 0,1 m	78,11	443,30
SensorDistancestdvxTraj = 0,2 m	79,01	448,28
SensorDistancestdvxTraj = 0,5 m	104,4	488,07
SensorDistancestdvxTraj = 1 m	248,86	893,32
SensorDistancestdvxTraj = 1,5 m	516,16	1926,45
SensorDistancestdvxTraj = 2 m	777,2	3207,9
AzimuthAccuracyTraj = 0,01°	83,27	452,8
AzimuthAccuracyTraj = 0,05°	83,40	453,59
AzimuthAccuracyTraj = 0,2°	89,69	468,09
AzimuthAccuracyTraj = 0,3°	106,91	492,32
AzimuthAccuracyTraj = 0,4°	156,98	582,83
AzimuthAccuracyTraj = 0,5°	245,59	752,06
SensorSpeedstdvxTraj = 0,0278 m/s	79,63	435,3
SensorSpeedstdvxTraj = 0,1389 m/s	156,49	572,48
SensorSpeedstdvxTraj = 0,2778 m/s	284,33	1042,29
SensorSpeedstdvxTraj = 0,5556 m/s	874,04	2742,50
SensorSpeedstdvxTraj = 0,8333 m/s	1143	3508,71
SensorSpeedstdvyTraj = 0,0278 m/s	78,81	415,04
SensorSpeedstdvyTraj = 0,1389 m/s	83,2	507,33
SensorSpeedstdvyTraj = 0,2778 m/s	216,77	790,64
SensorSpeedstdvyTraj = 0,5556 m/s	616,33	1996,9
SensorSpeedstdvyTraj = 0,8333 m/s	1035,6	3321,94

Tabelle 5-15: Ergebnisse der Metriken bei variiertem Sensorrauschen im Testfall 2

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
SensorDistancestdvxTraj = 0.1 m	68,60	427,37
SensorDistancestdvxTraj = 0.2 m	69,61	433,47
SensorDistancestdvxTraj = 0,5 m	74,92	464,83
SensorDistancestdvxTraj = 1 m	130,3	692,91
SensorDistancestdvxTraj = 1,5 m	401,36	1751,36
SensorDistancestdvxTraj = 2 m	696,34	2945,13
AzimuthAccuracyTraj = 0,01°	69,76	434,33
AzimuthAccuracyTraj = 0,05°	70,18	436,82
AzimuthAccuracyTraj = 0,2°	73,3	455,59
AzimuthAccuracyTraj = 0,3°	76,47	474,7
AzimuthAccuracyTraj = 0,4°	91,34	540,92
AzimuthAccuracyTraj = 0,5°	126,56	634,61
SensorSpeedstdvxTraj = 0,0278 m/s	70,53	418,49
SensorSpeedstdvxTraj = 0,1389 m/s	73,45	517,66
SensorSpeedstdvxTraj = 0,2778 m/s	130,03	772,75
SensorSpeedstdvxTraj = 0,5556 m/s	716,68	2228,15
SensorSpeedstdvxTraj = 0,8333 m/s	1084,83	3355,08
SensorSpeedstdvyTraj = 0,0278 m/s	70,14	418,24
SensorSpeedstdvyTraj = 0,1389 m/s	73,94	514,65
SensorSpeedstdvyTraj = 0,2778 m/s	93,67	701,25
SensorSpeedstdvyTraj = 0,5556 m/s	488,35	1762,26
SensorSpeedstdvyTraj = 0,8333 m/s	982,13	3189,65

Hinsichtlich der Bewertungszahlen verschlechtert sich das Tracking-Ergebnis in beiden Testfällen bei höherem Rauschen in der Distanzmessung. Ab einer Standardabweichung von 1 m steigen die Bewertungszahlen stark an. Werden Werte für die Genauigkeit unterhalb der Standardwerte gesetzt, so wird das Tracking-Ergebnis nur leicht verbessert. Eine Verdreifachung der Genauigkeit des Sensors von 0,3 m auf 0,1 m verbessert das Tracking-Ergebnis nur marginal.

Eine Verbesserung der Genauigkeit der Azimuthwinkelmessung verbessert das Tracking-Ergebnis minimal. Auf der anderen Seiten erhöht eine Verschlechterung die Bewertungszahlen. Insbesondere bei großen Objektdistanzen kommt es zu vermehrten Fehlmessungen und der damit einhergehenden Verschlechterung der Gesamtbewertung. Ab einer Standardabweichung von 0,5° in der Azimuthwinkelmessung steigt die Bewertungszahl überproportional an.

Wird die Genauigkeit der Geschwindigkeitsmessungen in x- und y-Richtung verschlechtert, so steigen die Bewertungszahlen stark an. Ein Fehler von 2 km/h (0,5556 m/s) erzeugt große Bewertungszah-

len der Metriken. Das schlechte Tracking-Ergebnis entsteht, da die Parametrierung des verwendeten Kalman-Filter auf genauere Geschwindigkeitsmessungen ausgerichtet ist.

Als nächstes wird die Auflösung des simulierten Radarsensors verändert. Die Ergebnisse der Variation sind in den Tabellen 5-16 und 5-17 dargestellt. Bei höheren Auflösungen der Distanzmessung verbessert sich das Tracking-Ergebnis. Umgekehrt führt eine niedrigere Auflösung zu schlechteren Tracking-Ergebnissen. Insbesondere bei einer Distanzauflösung ab 2 m sinkt die Qualität des Tracking-Ergebnisses überproportional. Wird die Auflösung der Geschwindigkeitsmessung variiert, so verändert sich das Tracking-Ergebnis nur marginal. Erst bei einer Auflösung von 2 km/h (0,5556 m/s) erhöhen sich die Bewertungszahlen.

Tabelle 5-16: Ergebnisse der Metriken bei variiertes Sensorauflösung im Testfall 1

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
SensorResolutionPos = 0,1 m	50,74	284,57
SensorResolutionPos = 0,3 m	57,61	324,51
SensorResolutionPos = 0,7 m	88,44	504,29
SensorResolutionPos = 1 m	135,91	776,17
SensorResolutionPos = 2 m	312,63	1499,11
SensorResolutionVel = 0,0278 m/s	80,04	453,98
SensorResolutionVel = 0,0833 m/s	80,016	453,80
SensorResolutionVel = 0,1944 m/s	96,16	601,61
SensorResolutionVel = 0,2778 m/s	82,53	465,92
SensorResolutionVel = 0,5556 m/s	275,51	863,61

Tabelle 5-17: Ergebnisse der Metriken bei variiertes Sensorauflösung im Testfall 2

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
SensorResolutionPos = 0,1 m	41,15	262,70
SensorResolutionPos = 0,3 m	53,24	335,20
SensorResolutionPos = 0,7 m	89,08	550,28
SensorResolutionPos = 1 m	121,41	744,24
SensorResolutionPos = 2 m	235,29	1427,18
SensorResolutionVel = 0,0278 m/s	70,9	441,15
SensorResolutionVel = 0,0833 m/s	70,9	441,15
SensorResolutionVel = 0,1944 m/s	91,39	616,38
SensorResolutionVel = 0,2778 m/s	70,9	441,15
SensorResolutionVel = 0,5556 m/s	70,9	441,15

Die Auswirkungen verschiedener Sensorabtastraten auf das Tracking-Ergebnis kann mithilfe der Metriken nicht untersucht werden, da die Metriken zum Teil auf der Distanz zwischen realer und prädizierter Position basieren. Diese Distanz wird zu jedem Messpunkt berechnet. Durch eine veränderte Abtastrate verändert sich die Anzahl der Messpunkte und damit die Anzahl der Punkte zur Berechnung der Gesamtdistanz. Auf eine Darstellung der Ergebnisse bei Abtastratenvariation wird daher verzichtet.

5.3.2 Parametervariation der Clustergruppengenerierung

Die Parameter der Clustergruppengenerierung haben Einfluss auf die Verschmelzung beobachteter Objekte. Verschmolzen werden diese, wenn sie eine geringe Distanz zueinander aufweisen und ihre Geschwindigkeitsdifferenzen gleichzeitig unterhalb einer parametrisierten Schwelle liegen. Die Ergebnisse bei Variation der Clustergruppengenerierungsparameter zeigen die Tabellen 5-18 und 5-19.

Tabelle 5-18: Ergebnisse der Metriken bei variierten Clustergruppenparameter im Testfall 1

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
MaxDistanceForGroup = 1 m	79,68	452,29
MaxDistanceForGroup = 2 m	83,43	453,77
MaxDistanceForGroup = 4 m	131,99	569,63
MaxDistanceForGroup = 5 m	336,89	960,78
MaxDistanceForGroup = 6 m	693,54	1625,47
MaxVRelDeviation = 1 m/s	83,68	455,08
MaxVRelDeviation = 2 m/s	86,68	455,08
MaxVRelDeviation = 4 m/s	83,68	455,08
MaxVRelDeviation = 5 m/s	83,68	455,08
MaxVRelDeviation = 6 m/s	83,68	455,08

Tabelle 5-19: Ergebnisse der Metriken bei variierten Clustergruppenparameter im Testfall 2

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
MaxDistanceForGroup = 1 m	69,33	432,13
MaxDistanceForGroup = 2 m	70,13	436,71
MaxDistanceForGroup = 4 m	70,84	440,71
MaxDistanceForGroup = 5 m	71,23	442,72
MaxDistanceForGroup = 6 m	71,98	444,96
MaxVRelDeviation = 1 m/s	70,9	441,15
MaxVRelDeviation = 2 m/s	70,9	441,15
MaxVRelDeviation = 4 m/s	70,9	441,15
MaxVRelDeviation = 5 m/s	70,95	441,48
MaxVRelDeviation = 6 m/s	70,91	441,23

In Testfall 1 vergrößern sich die Bewertungszahlen bei steigender maximaler Clustergruppendifferenz. Während der Parallelfahrt der Objekte beträgt die minimale Distanz bei Standardparametrierung 4 m und 5 m. In diesen Bereichen sind die Objektgeschwindigkeiten identisch. Bei einer maximalen Clustergruppendifferenz von 5 m und 6 m werden die parallelen Objekte daher vereint. Damit entstehen Kardinalitätsfehler, die zu großen Bewertungszahlen führen. Bei Standardparametrierung der minimalen Distanz sind die Objekte so weit voneinander entfernt, dass diese auch bei Erhöhung der maximalen Geschwindigkeitsdifferenz nicht verschmolzen werden.

In Testfall 2 mit Objekten, die unterschiedliche Geschwindigkeiten aufweisen, hat die Variation der Clustergruppendifferenzparameter keinen Einfluss auf die Bewertungszahlen. Wird die minimale Distanz variiert, so sind die Geschwindigkeitsdifferenzen ausreichend groß, so dass die Objekte nicht verschmolzen werden.

5.3.3 Parametervariation der Datenassoziation

Schlüsselparameter der Datenassoziation ist die Größe des Suchgebiets für neue Messwerte rund um die prädierten Objektpositionen. Dieses Gebiet wird durch die Mahalanobisdistanz aufgespannt. Parameter für die Größe des Suchgebiets ist eine Prozentzahl. Dieser definiert den Anteil der Messwerte, die zum nächsten Messzeitpunkt innerhalb des Suchgebiets landen. Die Bewertung der Tracking-Ergebnisse für die beiden Testfälle sind in den Tabellen 5-20 und 5-21 notiert.

Tabelle 5-20: Ergebnisse der Metriken bei variierten Datenassoziationsparameter im Testfall 1

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
ValidationGatePercentage = 95 %	142,29	473,99
ValidationGatePercentage = 80 %	283,67	803,58
ValidationGatePercentage = 60 %	320,21	1045,81
ValidationGatePercentage = 40 %	390,76	1251,52

Tabelle 5-21: Ergebnisse der Metriken bei variierten Datenassoziationsparameter im Testfall 2

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
ValidationGatePercentage = 95 %	70,51	438,8
ValidationGatePercentage = 80 %	70,93	440,75
ValidationGatePercentage = 60 %	84,30	470,47
ValidationGatePercentage = 40 %	159	630,42

Bei der Analyse der Ergebnisse von Testfall 1 zeigt sich eine kontinuierliche Verschlechterung des Tracking-Ergebnisses bei Verkleinerung des Suchbereichs. Bei Testfall 2 dagegen sind die Werte überwiegend konstant und erst bei 40 % wird das Ergebnis verschlechtert. Begründung hierfür ist die nicht-konstante Geschwindigkeit der Objekte im Testfall 1. Durch die Verzögerung der Objekte weichen die prädizierten Zustände von den realen Zuständen ab. Dies wird durch einen größeren Suchbereich ausgeglichen. Wird dieser allerdings verkleinert, so geraten einige der Messwerte außerhalb dieses Bereiches.

Im Testfall 2 sind die Objektgeschwindigkeiten konstant. Aus diesem Grund sind die prädizierten Zustände der Objekte genauer und die Suchbereiche daher an den richtigen Stellen platziert. Eine Verkleinerung der Suchbereiche hat daher einen geringeren Einfluss.

5.3.4 Parametervariation der Objekt-Tracker

Die Parametrierung der Objekt-Tracker hat Einfluss auf das implementierte Track-Management. Mit diesen Parametern werden die Reflexionen von realen Objekten von Clutter unterschieden. Die Ergebnisse dieser Parametervariation sind in den Tabellen 5-22 und 5-23 dargestellt.

Tabelle 5-22: Ergebnisse der Metriken bei variierten Objekt-Tracker Parameter im Testfall 1

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
maxNundetected = 15	83,68	455,08
maxNundetected = 12	83,68	455,08
maxNundetected = 8	83,68	455,08
maxNundetected = 5	83,68	455,08
maxNundetected = 1	83,68	455,08
mausNm/mausNN = 6/7	90,2	468,55
mausNm/mausNN = 3/7	71,09	428,75
mausNm/mausNN = 1/7	145,72	713,91
mausNm/mausNN = 3/5	73,57	422,34

Tabelle 5-23: Ergebnisse der Metriken bei variierten Objekt-Tracker Parameter im Testfall 2

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
maxNundetected = 15	70,9	441,15
maxNundetected = 12	70,9	441,15
maxNundetected = 8	70,9	441,15
maxNundetected = 5	70,9	441,15
maxNundetected = 1	70,9	441,15
mausNm/mausNN = 6/7	75,63	454,11
mausNm/mausNN = 3/7	81,07	488,37
mausNm/mausNN = 1/7	368,93	4340,89
mausNm/mausNN = 3/5	87,86	513,13

Durch die künstliche Generierung der Objektrajektorien und Sensordaten zeigen die Variationen der Parameter für die maximale Anzahl an Fehlmessungen eines Objekts keine Auswirkungen auf das Tracking-Ergebnis. Für eine Einschätzung der Einflüsse dieser Parameter sind genauere Sensormodelle für die Generierung der Sensordaten erforderlich. Auch die Einflüsse der Parametervariation des m/N-Verfahrens sind nur bedingt analysierbar. Auch hierfür wird auf ein genaueres Modell verwiesen. Einzig die Wahl auf ein 1/7-Verfahren zeigt eine Verschlechterung des Tracking-Ergebnisses. Eine bessere Einschätzung der Einflüsse der untersuchten Parameter liefern Testfahrten mit einem realen Radarsensor. Für den realen Fall existieren zurzeit keine veröffentlichten Metriken, die mehr als nur die reinen Distanzen zwischen prädizierten und realen Trajektorien einbeziehen.

5.3.5 Parametervariation der Tracking-Algorithmus-Metriken

Die Parameter der verwendeten Metriken haben ebenfalls Einfluss auf die Bewertung der Tracking-Ergebnisse. Sowohl in der Veröffentlichung der OSPA-T-Metrik¹²⁷, als auch in der Veröffentlichung der Rahmathullah et al.-Metrik¹²⁸ werden Werte empfohlen. Die Bewertungszahlen bei Parametervariation der Metriken sind für die beiden Testfälle in den Tabellen 5-24 und 5-25 notiert.

Tabelle 5-24: Ergebnisse der Metriken bei variierten Tracking-Algorithmus-Metrik Parameter im Testfall 1

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
OSPATC = 7	97,18	
OSPATC = 3	70,18	
OSPATP = 0.5	104,4	
OSPATP = 2	97,13	
RahmC = 7		482,44
RahmC = 3		429,21
RahmP = 0.5		1993,08
RahmP = 2		19,85
RahmGamma = 0.5		455,2
RahmGamma = 2		454,19
RahmMaxIter = 10		455,08
RahmMaxIter = 30		455,08

¹²⁷Ristic, B. et al.: A Metric for Performance Evaluation of Multi-Target Tracking Algorithms (2011)

¹²⁸Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: Rahmathullah et al. Metrik (2016)

Tabelle 5-25: Ergebnisse der Metriken bei variierten Tracking-Algorithmus-Metrik Parameter im Testfall 2

Parameter	OSPA-T Metrik	Rahmathullah et al. Metrik
OSPATC = 7	79,23	
OSPATC = 3	62,56	
OSPATP = 0.5	97,82	
OSPATP = 2	69,85	
RahmC = 7		466,17
RahmC = 3		416,15
RahmP = 0.5		1916,79
RahmP = 2		19,24
RahmGamma = 0.5		442,03
RahmGamma = 2		440,74
RahmMaxIter = 10		441,15
RahmMaxIter = 30		441,15

Durch den Parameter C werden bei beiden Metriken die maximale Strafe durch Kardinalitätsfehler festgelegt. Bei kleinerem C werden bei beiden Testfällen geringere Bewertungszahlen und damit ein besseres Tracking-Ergebnis attestiert. Die Parameter P beider Metriken definieren den Einfluss von Messausreißern auf die Gesamtbewertung. Insbesondere bei der Rahmathullah et al. Metrik hat dieser Parameter einen großen Einfluss auf die Tracking-Bewertung. Der Parameter γ der Rahmathullah et al.-Metrik beschreibt die Strafe bei einem Wechsel der Benennung der Objekte. Während des Tracking-Verlaufs der Testfälle treten diese nicht auf, daher ändern sich die Bewertungszahlen nur marginal. Schlussendlich wird die maximale Anzahl von Iterationen der Rahmathullah et al.-Metrik variiert. Hier ergeben sich bei Anwendung der Metriken auf die beiden Testfälle keine Unterschiede in den Bewertungszahlen. Daraus lässt sich schließen, dass der numerische Algorithmus der Rahmathullah et al.-Metrik schnell konvergiert.

6 Tracking-Ergebnisse mit realen Radardaten

Abseits von simulierten Radardaten werden in diesem Kapitel die Tracking-Ergebnisse bei Nutzung von realen Radardaten dargestellt. Da für die Bewertung der Tracking-Qualität die Ground-Truth-Informationen fehlen, wird auf die Bewertung der Tracking-Ergebnisse verzichtet. Der Tracking-Algorithmus wird mit Standardparametrierung betrieben.

6.1 Testfahrt mit zwei Zielfahrzeugen

Für die erste Aufzeichnung von realen Radardaten bewegen sich zwei Fahrzeuge auf einer freien Fläche und der Radarsensor ist an einer festen Position montiert. Im ersten Teil der Testfahrt entfernen sich die Fahrzeuge vom Sensor, führen anschließend ein Wendemanöver durch und bewegen sich zum Sensor zurück. Der Tracking-Algorithmus detektiert und verfolgt eine Vielzahl von statischen Objekten. Die dynamischen Fahrzeuge bewegen sich dazwischen. Abbildung 6-1 zeigt ein Einzelbild während der Objektverfolgung. Zu sehen sind eine Vielzahl von statischen Objekten und die beiden dynamischen Objekte 8 und 53. Zu diesem Zeitpunkt bewegen sich die Fahrzeuge vom Sensor weg.

In Anhang A.6 wird ein späterer Zeitpunkt der Testfahrt dargestellt, bei dem sich die Fahrzeuge dem Sensor wieder nähern. Ein Video des Tracking-Verlaufes ist im digitalen Anhang unter "Raw_TwoCar_Continuous_2016.10.28_x1.avi" zu finden.

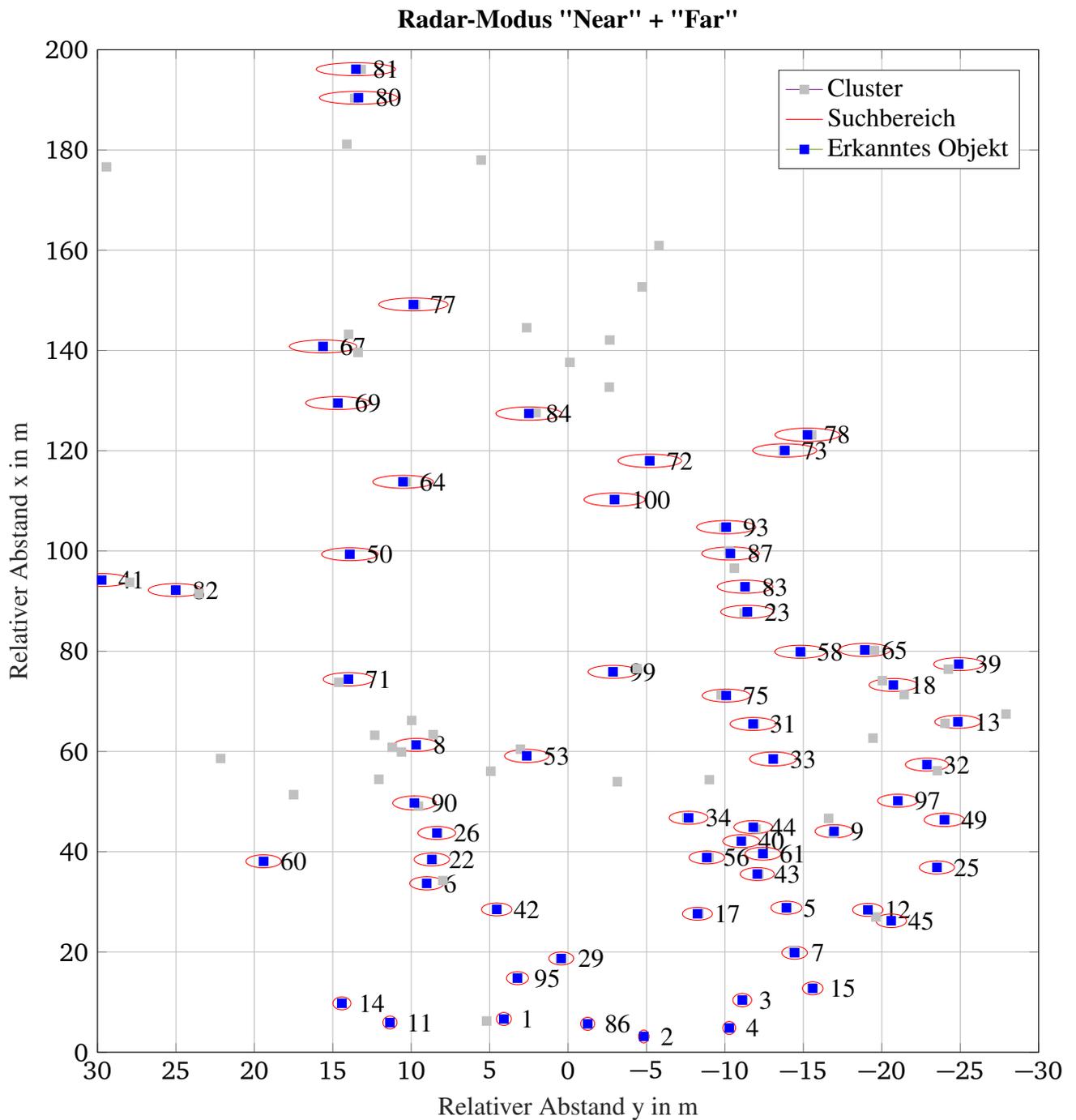


Abbildung 6-1: Einzelbild: Tracking mit realen Radardaten 1. Das Bild zeigt eine Vielzahl von statischen Objekten. Objekte 8 (Position ca: $x=60\text{m}$, $y=10\text{m}$) und 53 (Position ca: $x=60\text{m}$, $y=3\text{m}$) sind die beiden verfolgten dynamischen Fahrzeuge. Sie entfernen sich vom Sensor.

6.2 Testfahrt mit Corner Reflector

Für weitere Testfahrten wird der Radarsensor an der Front eines Versuchsfahrzeugs befestigt, um Radarmessungen mit einem dynamischen Egofahrzeug aufnehmen zu können. Dieses Versuchsfahrzeug nähert sich in den Testfahrten einem Corner-Reflector auf einer geraden Straße mit unterschiedlichen Geschwindigkeiten. Der aktuelle Versuchsaufbau ermöglicht keinen Zugriff auf die Sensoren des Versuchsfahrzeugs. Aus diesem Grund stehen die Beschleunigungen und Gierraten des Egofahrzeugs nicht für den Tracking-Algorithmus zur Verfügung. Bei einem dynamisch bewegten Radarsensor sind diese Informationen entscheidend für die Einbeziehung der Egofahrzeugbewegung in die Tracking-Verläufe. Durch eine Eigendrehung verändert sich das Radarbild und damit befinden sich die Objekte nicht an den vom Tracking-Algorithmus prädizierten Positionen. Daher wird eine Objektverfolgung unmöglich.

Exemplarisch wird das Tracking-Ergebnis einer Testfahrt mit dem Versuchsfahrzeug vorgestellt. Die gefahrene Geschwindigkeit beträgt 10 km/h. Auf Abbildung 6-2 ist die Freifläche der Straße gut zu erkennen. Seitlich dieser Freifläche werden statische Objekte vom Tracking-Algorithmus identifiziert. Das Versuchsfahrzeug nähert sich dem Corner Reflector (Objekt 5).

Bei Beobachtung des Tracking-Verlaufs zeigt sich, dass der Corner Reflector als ein Objekt erkannt und verfolgt wird. Nähert sich das Egofahrzeug dem Corner Reflector, so werden Lücken in der Objektverfolgung sichtbar. Grund hierfür ist die geringe Geschwindigkeit des Egofahrzeugs und der damit auftretenden Interferenzeffekte des Radarsensors. Ein weiteres Phänomen ist ebenfalls auf der genannten Abbildung zu sehen. Ab einer Distanz von etwa 25 m wird zusätzlich zu dem Corner-Reflector ein weiteres Objekt (Nummer 73) mit doppelter Distanz registriert. Dieses Objekt weist in etwa die doppelte Relativgeschwindigkeit auf.

Bei Testfahrten mit höheren Geschwindigkeiten treten die beobachteten Interferenzeffekte nicht auf, die Zweitreflexionen des Corner Reflectors dagegen schon.

Eine Untersuchung der Cluster-Anzahl, die durch Reflexionen am Corner Reflector entstehen, zeigen, dass pro Messzyklus eine Reflexion entsteht. Der Corner Reflector befindet sich damit über den gesamten Beobachtungszeitraum immer nur in einer Auflösungszelle des Radarsensors.

Videos der Tracking-Ergebnisse aller hier vorgestellten Testfahrten sind im digitalen Anhang unter "10kmh2_2017.03.30_x1.avi", "30kmh2_2017.03.30_x1.avi" und "60kmh2_2017.03.30_x1.avi" zu finden.

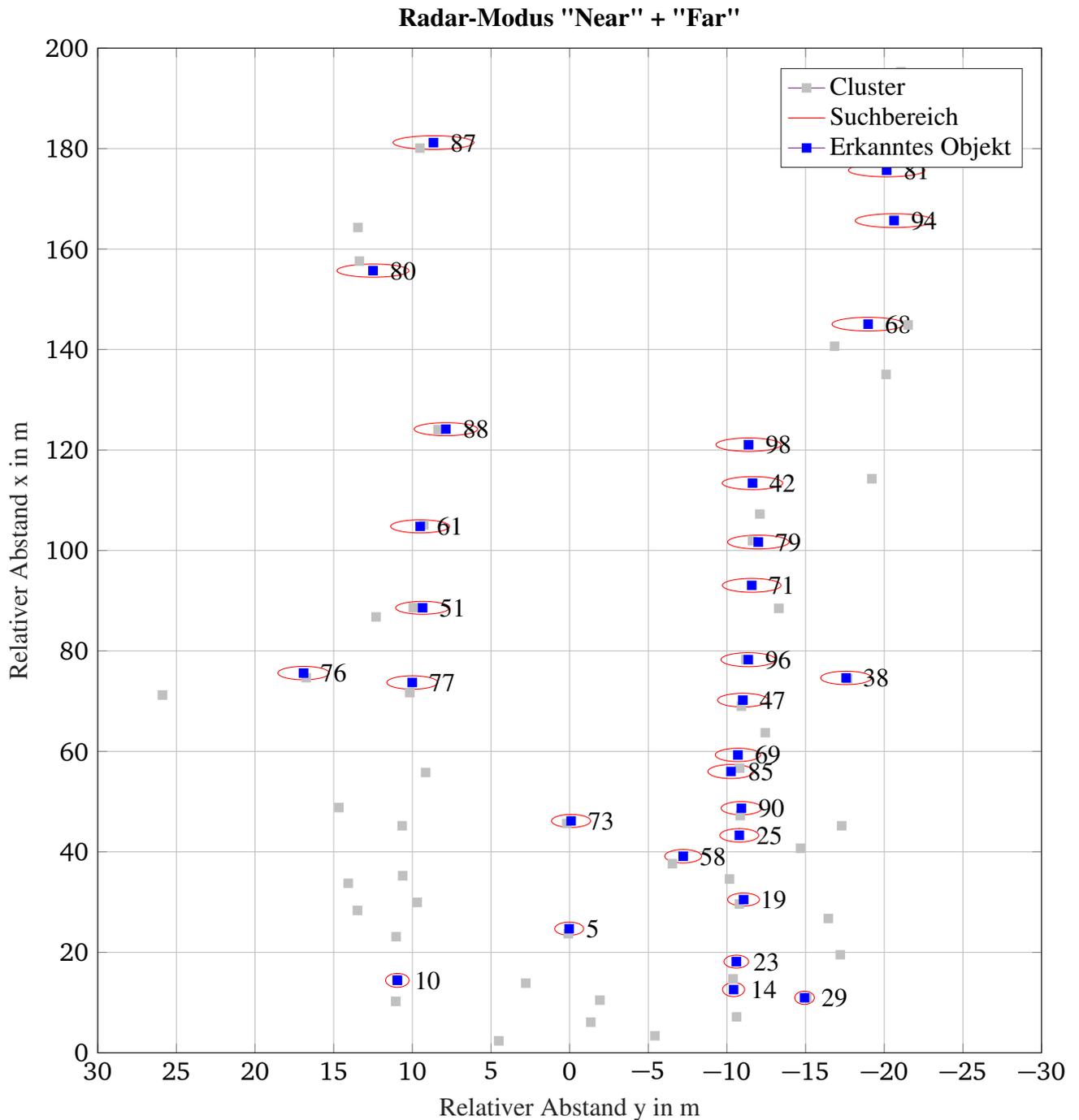


Abbildung 6-2: Einzelbild: Tracking mit realen Radardaten mit Corner Reflector. Das Bild zeigt die Freifläche der Straße. Im Seitenbereich werden statische Objekte identifiziert. Objekt 5 ist der detektierte Corner Reflector. Ab etwa 25m wird zusätzlich Objekt 73 entdeckt. Dieses hat in etwa die doppelte Distanz und Relativgeschwindigkeit wie Objekt 5

6.3 Vergleich des implementierten Algorithmus mit einer Industrielösung

Zum Abschluss dieser Thesis wird ein Vergleich der Tracking-Ergebnisse zwischen dem vorgestellten implementierten Tracking-Algorithmus und dem Algorithmus des Radarsensor-Herstellers durchgeführt. Da für den Leistungsvergleich von Tracking-Algorithmen bei der Verwendung von realen Sensordaten keine bekannte Metrik existiert, erfolgt der Vergleich durch subjektives Empfinden. Untersucht wird das Tracking-Verhalten beider Algorithmen bei Verwendung der Testfahrt mit einem Corner Reflector.

Durch auftretende Interferenzeffekte bei einer Fahrzeuggeschwindigkeit von 10 km/h und den damit einhergehenden Schwierigkeiten des implementierten Algorithmus, wird dieser Testfall für einen Vergleich herangezogen. In der Standardparametrierung des implementierten Algorithmus wird der Corner Reflector während der Annäherung des Fahrzeugs ab einer Entfernung von etwa 70 m mehrfach kurzfristig nicht mehr erkannt. Infolgedessen ändert sich die Bezeichnung des Objekts mehrfach, da der Algorithmus jede neue Detektion als Geburt eines Objekts behandelt. Die Industrielösung erkennt den Reflektor dagegen stetig über den gesamten Beobachtungszeitraum. Es scheint, dass dieser Algorithmus länger an bekannten Objekten festhält, als dies bei dem eigenen Algorithmus der Fall ist.

7 Fazit und Ausblick

7.1 Fazit

Diese Master Thesis gewährt einen umfassenden Einblick in die Analyse von Tracking-Algorithmen. Nach einer Einführung in die Grundlagen eines Radarsensors werden die Funktionsprinzipien der Distanz-, Winkel- und Radialgeschwindigkeitsmessungen vorgestellt. Dabei wird der Fokus insbesondere auf die Funktionsprinzipien des zur Verfügung stehenden Pulskompressionsradars gelegt.

Im Anschluss daran werden die verschiedenen Bestandteile von Tracking-Algorithmen vorgestellt. Für jedes dieser Bauteile werden Realisierungsmöglichkeiten präsentiert. Anschließend wird eine ausführliche Dokumentation der Implementierung eines Tracking-Algorithmus bereitgestellt.

Im folgenden Kapitel werden Kriterien für den Leistungsvergleich von Tracking-Algorithmen erläutert. Anschließend werden Wege gesucht, einzelne Kriterien miteinander zu verknüpfen und damit die Tracking-Algorithmus-Metriken eingeführt.

Weiterhin werden in dieser Thesis Testfälle für den Leistungsvergleich der Algorithmen vorgestellt. Es wird gezeigt, dass die Testfälle zu komplexen Umfeldsituation für Radarsensoren führen. Damit qualifizieren sich diese Testfälle für zukünftige Analysen der Leistungsfähigkeit von Radar Tracking-Algorithmen.

Im darauf folgenden Kapitel werden die Auswirkungen von Parametervariationen auf die Tracking-Qualität untersucht. Die Ergebnisse zeigen, dass insbesondere die Trajektorien der Objekte und die Qualität der Radarsensordaten Einfluss auf die Tracking-Qualität haben.

Abschließend werden Vergleiche zwischen dem implementierten und dem Hardwarehersteller Tracking-Algorithmus durchgeführt. Es zeigt sich, dass die Industrielösung bei komplexen Situationen, wie auftretende Interferenzeffekte, eine bessere Tracking-Qualität liefert.

7.2 Ausblick

Der vorgestellte Tracking-Algorithmus stellt einen Einstieg in die Entwicklung eines komplexeren Algorithmus dar. In der aktuellen Form lassen sich ohne großen Aufwand die Einflüsse der Schlüsselparameter auf das Tracking-Ergebnis untersuchen. Als Bewertungsgrundlage dienen Testfälle, die durch simulierte Radardaten gespeist werden. Die aktuelle Erzeugung ist rudimentär und kann zukünftig durch genauere Radarmodelle abgelöst werden. Grundlage der Algorithmus-Bewertung sind die vorgestellten Metriken. Nach aktuellem Stand nutzen diese Metriken nicht alle denkbaren MoE für die Einschätzung der Tracking-Qualität. Bei der Entwicklung zukünftiger Metriken können diese in die Bewertung aufgenommen werden.

Insgesamt lassen sich durch diese Thesis viele zukünftige Erweiterungen der vorgestellten Algorithmen, Metriken und Testfälle erforschen. In den folgenden Unterkapiteln werden die Themen dieser Arbeit durchlaufen und potentielle Weiterentwicklungen angesprochen.

7.2.1 Der Radarsensor

Der am FZD zur Verfügung stehende Radarsensor liefert über die definierte Schnittstelle die Distanz- und Geschwindigkeitsinformationen aller Reflexionen im beleuchteten Radarbereich. In der aktuellen Ausbaustufe wird allerdings von einem stehenden Sensor bzw. Egofahrzeug ausgegangen. Dem Algorithmus stehen keine Informationen über die Eigengeschwindigkeit oder die Gierrate des Egofahrzeugs zur Verfügung. Für zukünftige Weiterentwicklungen des Tracking-Algorithmus ist die Erfassung dieser Informationen durch Auswertung der entsprechenden Busse des Egofahrzeugs erforderlich. Durch die bekannte Eigengeschwindigkeit werden die Berechnungen der Absolutgeschwindigkeiten der Objekte möglich. Großen Einfluss auf das Tracking-Ergebnis hat auch die Gierbewegung des Egofahrzeugs. Die Auswertung der Ergebnisse zeigt, dass bei den realen Testfällen die Objekte verloren werden, wenn eine Gierbewegung ausgeführt wird. Grund hierfür ist, dass die prädierten Objektpositionen nicht mit dem Radarbild übereinstimmen und die entsprechenden Objekte daher gelöscht werden. Der Einbezug der Gierbewegung erfolgt nach Reuter¹²⁹ in zwei Schritten. Im ersten Schritt werden die Objektzustände wie bekannt prädiert. Anschließend wird die Eigenbewegung des Egofahrzeugs durch eine weitere Zustandsübergangsmatrix und eine dazugehörige Kovarianzmatrix für die Messfehler der Gierrate kompensiert.

Auch der Einsatz eines anderen Radarsensors ist möglich. In der vorgestellten Literatur wird oftmals ein High-Resolution-Radarsystem für die Sensierung des Umfelds eingesetzt. Diese Radare erzeugen ein detaillierteres Radarbild, bei dem die Objekte eine größere Anzahl von Reflexionen erzeugen. Die Anordnung der Reflexionen im Radarbild ermöglicht die Schätzung der Objektform und damit den Einsatz von Extended Object Tracking Algorithmen.

Zusätzlich ist auch der Einsatz eines Sensors mit einem komplett anderen Messprinzip möglich. Für den automobilen Einsatz ist beispielsweise das Light detection and ranging (LIDAR) zu nennen.

7.2.2 Der Tracking-Algorithmus

Neben der angesprochenen Einbeziehung der Gierrate und Eigengeschwindigkeit des Egofahrzeugs sind auch zukünftige Erweiterungen des Tracking-Algorithmus vorstellbar. Alle Blöcke des implementierten Tracking-Algorithmus auf Abbildung 3-6 lassen sich individuell weiterentwickeln.

Beim Extended Object Tracking kann durch den Einsatz eines High-Resolution-Radars oder eines LIDAR-Sensors auf die Clustergruppengenerierung verzichtet werden. Für diesen Fall stehen eine Vielzahl von Verfahren für die Schätzung der Objektform und -bewegung zur Verfügung. Einige dieser Verfahren werden in Kapitel 3 vorgestellt.

Für die Datenassoziation kann geprüft werden, ob das JPDAF-Verfahren eine Verbesserung gegenüber der aktuellen Implementierung darstellt. Dem Trend der aktuellen Tracking-Entwicklung folgend wird allerdings der Einsatz von RFS für das Tracking empfohlen, um die Probleme der Datenassoziation zu umgehen.

Für die Prädiktion der verfolgten Objektzustände bieten sich komplexere Target-Bewegungsmodelle

¹²⁹Reuter, S.: Multi-object tracking using random finite sets (2014) S.162

an. Bei einem Einsatz eines High-Resolution-Radars, bei dem der Momentanpol des beobachteten Objekts ermittelt wird, kann der Einsatz eines Einspurmodells für die Bewegungsvorhersage überprüft werden. Damit einhergehend wird die Filterung der Objekte auf Basis von Extended bzw. Unscented Kalman-Filter oder direkt über Partikelfilter gefordert. Werden für das Tracking RFS verwendet, so ist wiederum eine andere Art Filter zu wählen, beispielsweise der PHD.

Die Untersuchungen in Kapitel 5 zeigen die Verschlechterung des Tracking-Ergebnisses, wenn die verfolgten Objekte stark verzögern oder beschleunigen. Für diesen Fall bietet sich der Einsatz mehrerer Bewegungsmodelle an. Beispielsweise eines für konstante Geschwindigkeit und eines für Fahrmanöver. Ein solches Verfahren ist beispielsweise das IMM.

Die Untersuchungen der Tracking-Ergebnisse mit realen Sensordaten zeigen, dass eine vorgelagerte Filterung der Cluster sinnvoll sein kann. Für den Einsatz solcher Filter sind allerdings Kriterien zu definieren, anhand derer ungewünschte Cluster identifiziert werden. Möglicherweise ist die Radar cross section (RCS) (Radarquerschnitt) des Clusters ein solches Kriterium.

Auch die Verwaltung der Objekt-Liste von der Geburt bis zum Sterben der Objekte kann weiter verbessert werden. Hierfür ist beispielsweise die Veröffentlichung von Hyun¹³⁰ erwähnt.

7.2.3 Die Leistungsbewertung der Tracking-Algorithmen

Die Leistungsbewertung der Tracking-Algorithmen erfolgt mithilfe von künstlich erzeugten Radardaten. Weicht die Güte dieser Radarwerte von denen eines realen Radarsensors ab, so kann sich in der Bewertung der Algorithmen ein unterschiedliches Bild zwischen der Simulation und der Realität einstellen. Empfohlen wird daher, die Bewertung der Tracking-Algorithmen anhand von realen Sensordaten durchzuführen. Nach dem aktuellen Stand der Technik sind keine Metriken für den Einsatz mit realen Sensordaten verfügbar. Zukünftig ist daher zu beobachten, ob sich dieser Zustand ändert. Für den Einsatz von Metriken sind allerdings weitere Probleme zu lösen. So müssen die Ground-Truth aller beobachteten Objekte verfügbar sein, um Aussagen über die Genauigkeit der untersuchten Algorithmen zu erhalten. Weiterhin sind Wiederholungen der Testfahrten durchzuführen, um den Einfluss von Rauschen auf die Bewertungszahlen zu kompensieren. Um bei diesen Wiederholungen die Fahrzeugführer als Einflussfaktoren auf die gefahrenen Trajektorien auszuschließen, wird der Einsatz von Fahrrobotern vorgeschlagen.

Die Rechenzeit der Metriken steigt exponentiell mit der Anzahl der beobachteten Objekte an. Aus diesem Grund sollten bei der Nutzung von realen Sensordaten alle irrelevanten Objekte aus den Daten gefiltert werden.

¹³⁰Hyun, E.; Lee, J.-H.: Multi-target tracking scheme using a track management table (2016)

A Anhang

A.1 Programmablaufpläne des implementierten Tracking-Algorithmus

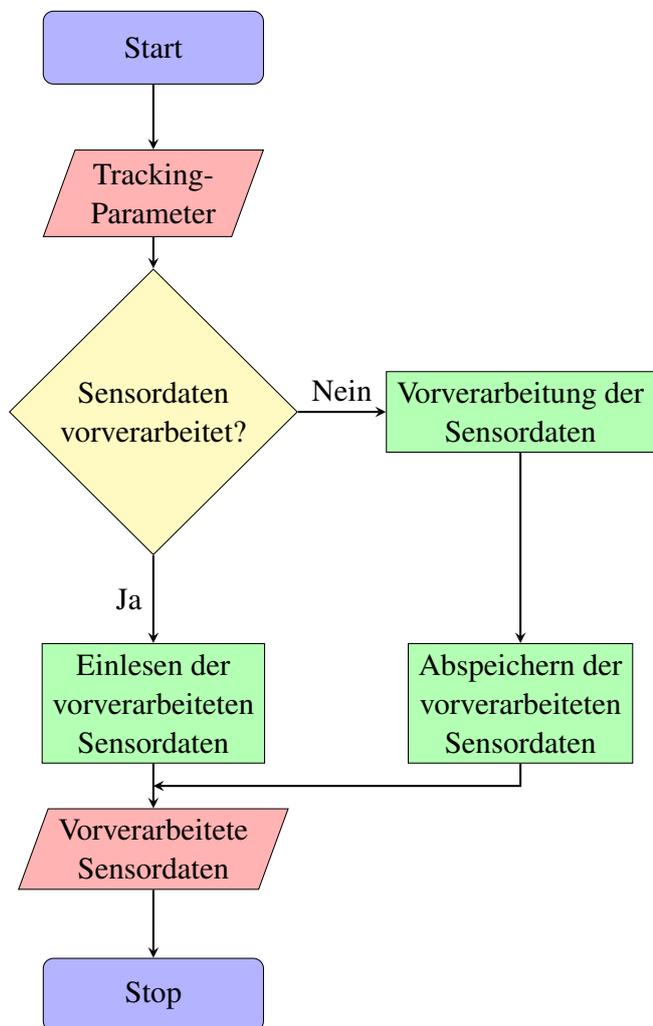


Abbildung A-1: Programmablaufplan (nach ISO 5807/DIN 66001): Einlesen der realen Sensordaten

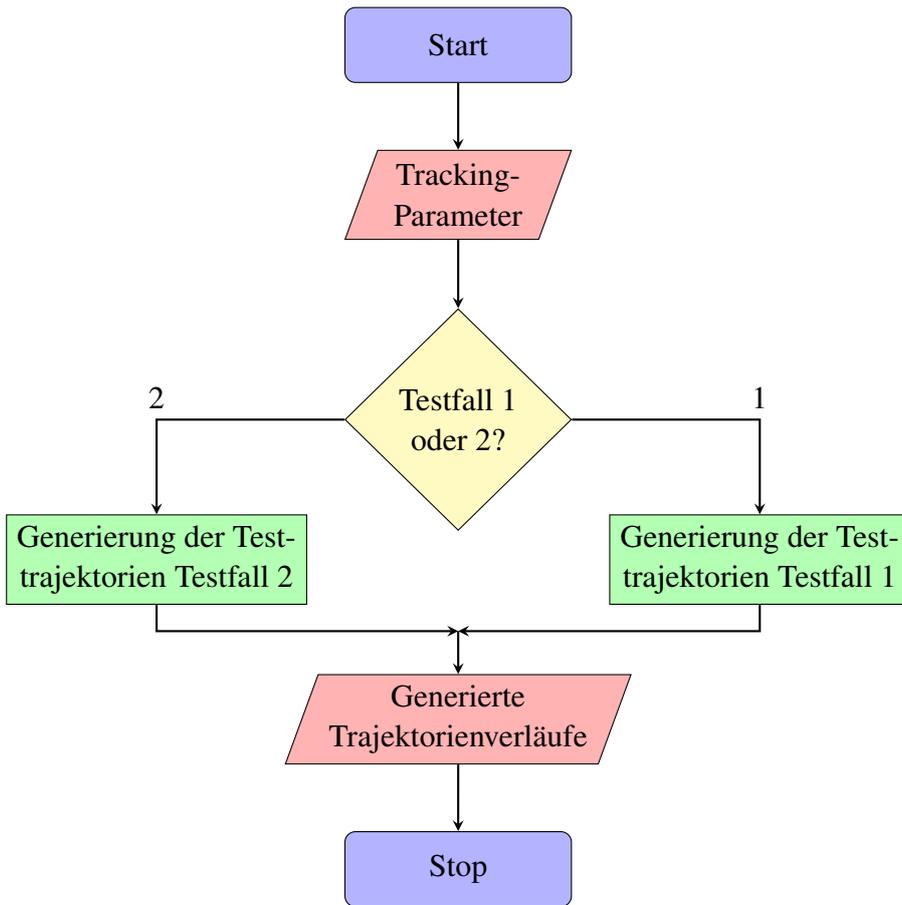


Abbildung A-2: Programmablaufplan (nach ISO 5807/DIN 66001): Generieren der Testtrajektorien

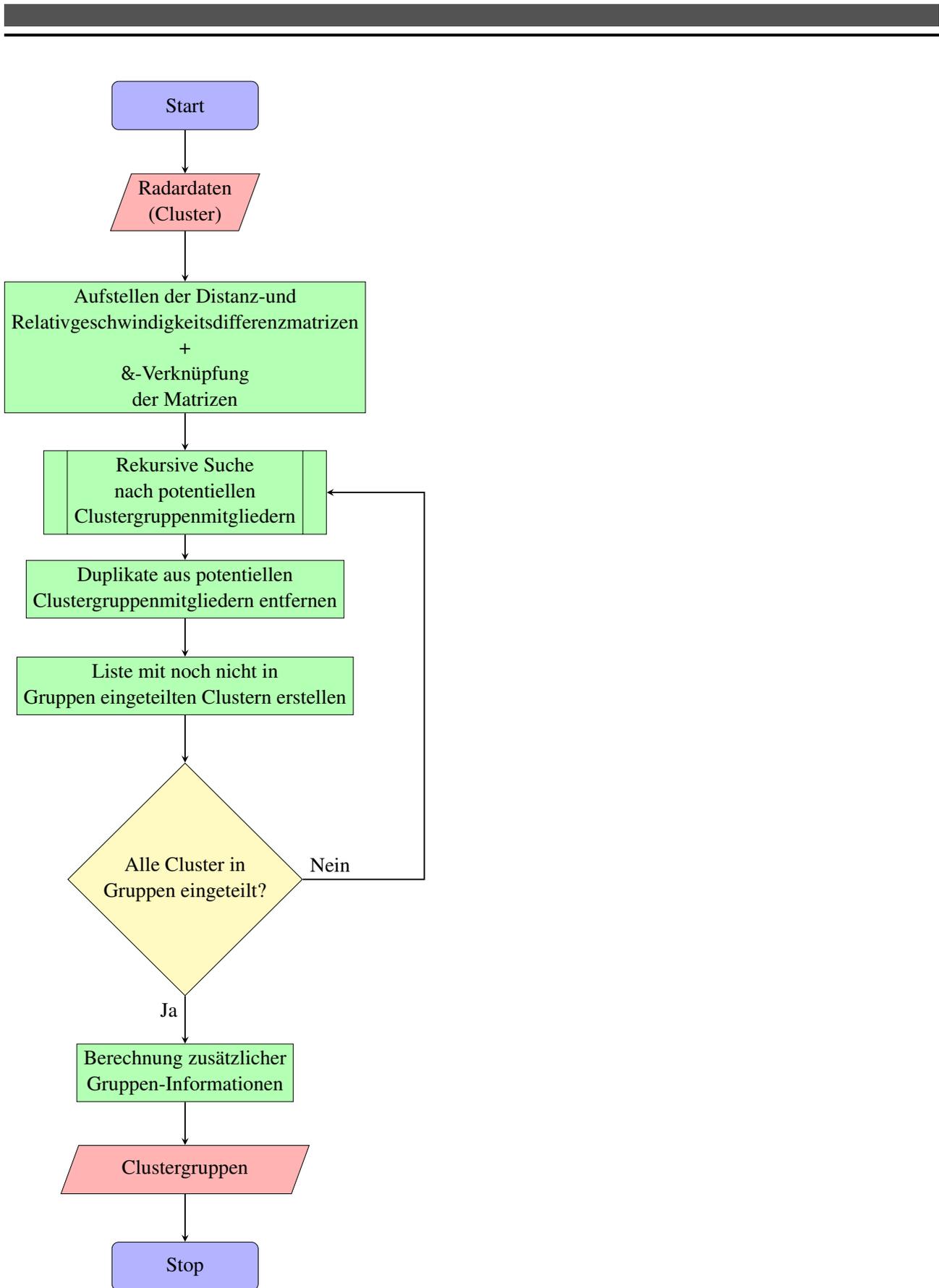


Abbildung A-3: Programmablaufplan (nach ISO 5807/DIN 66001): Clustergruppengenerierung

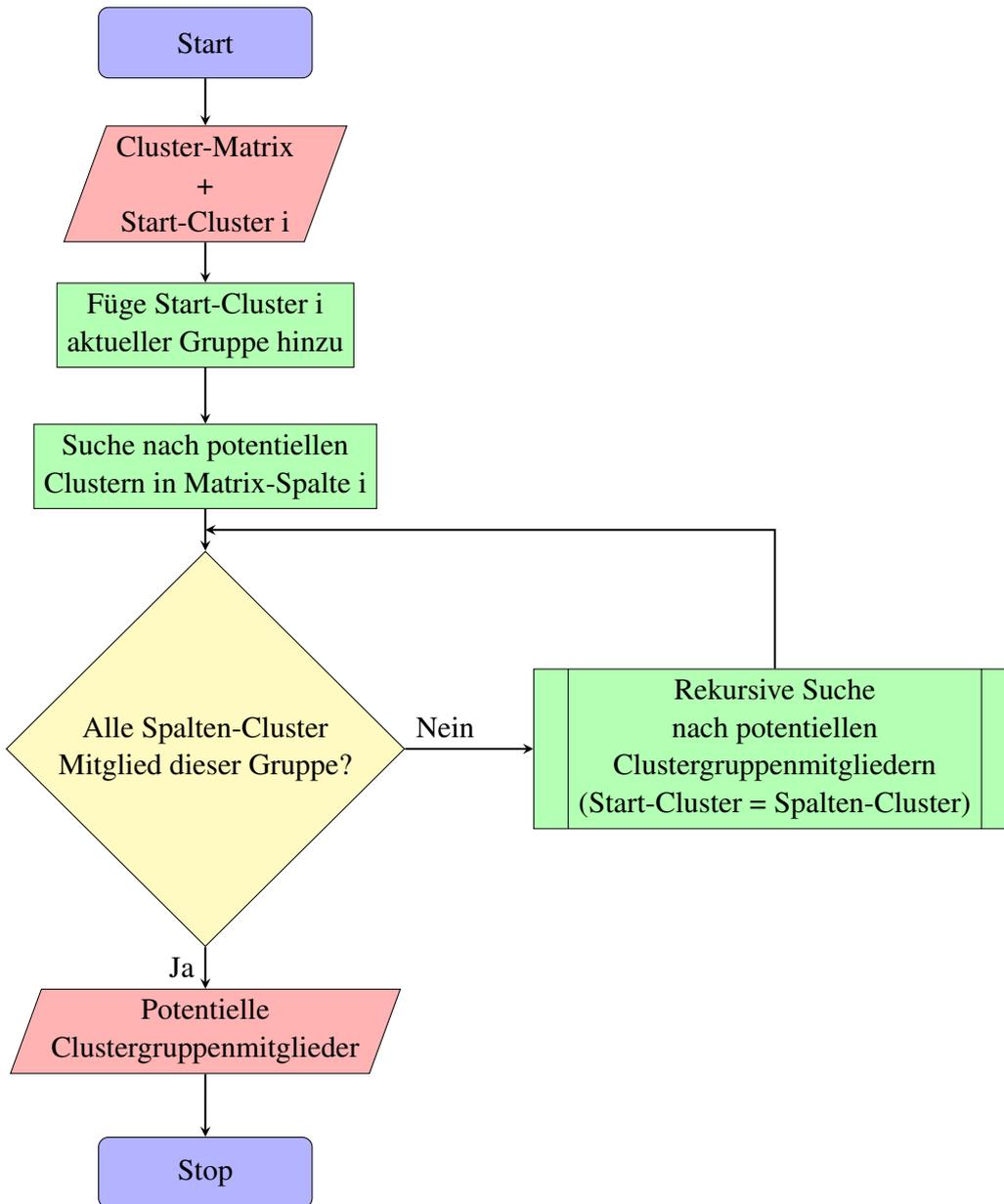


Abbildung A-4: Programmablaufplan (nach ISO 5807/DIN 66001): Rekursive Suche nach potentiellen Clustergruppenmitgliedern

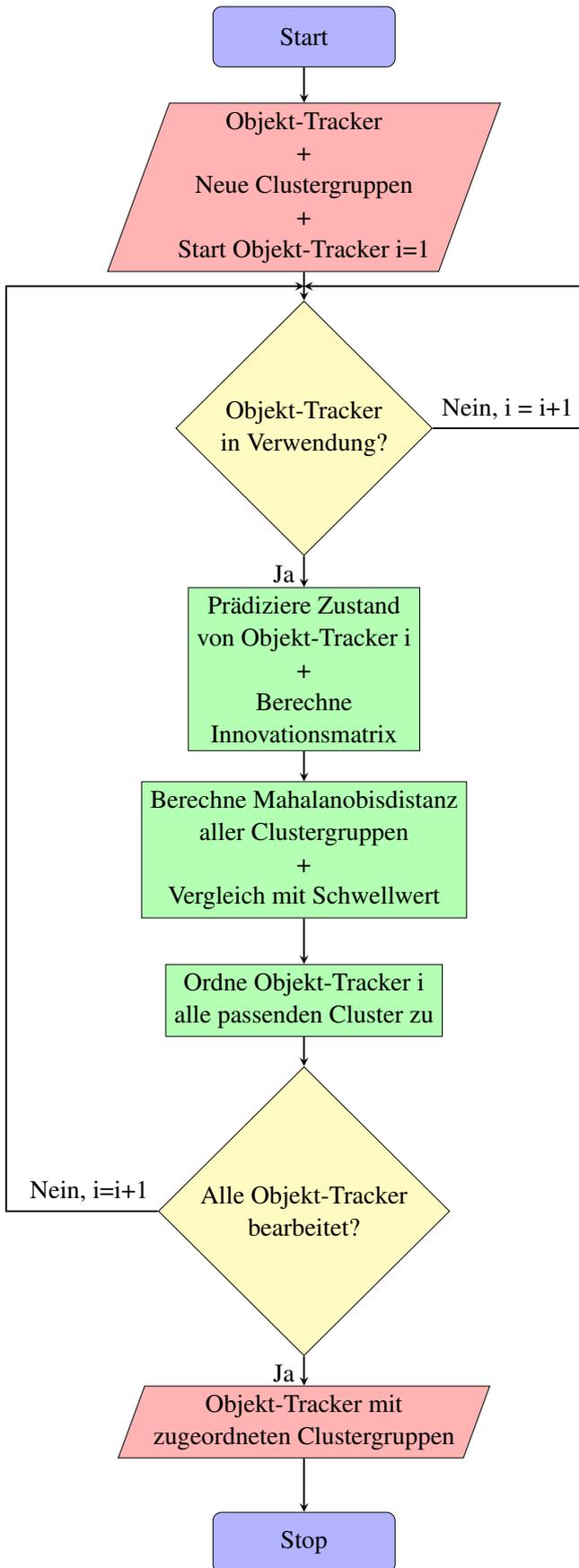


Abbildung A-5: Programmablaufplan (nach ISO 5807/DIN 66001): Datenassoziation

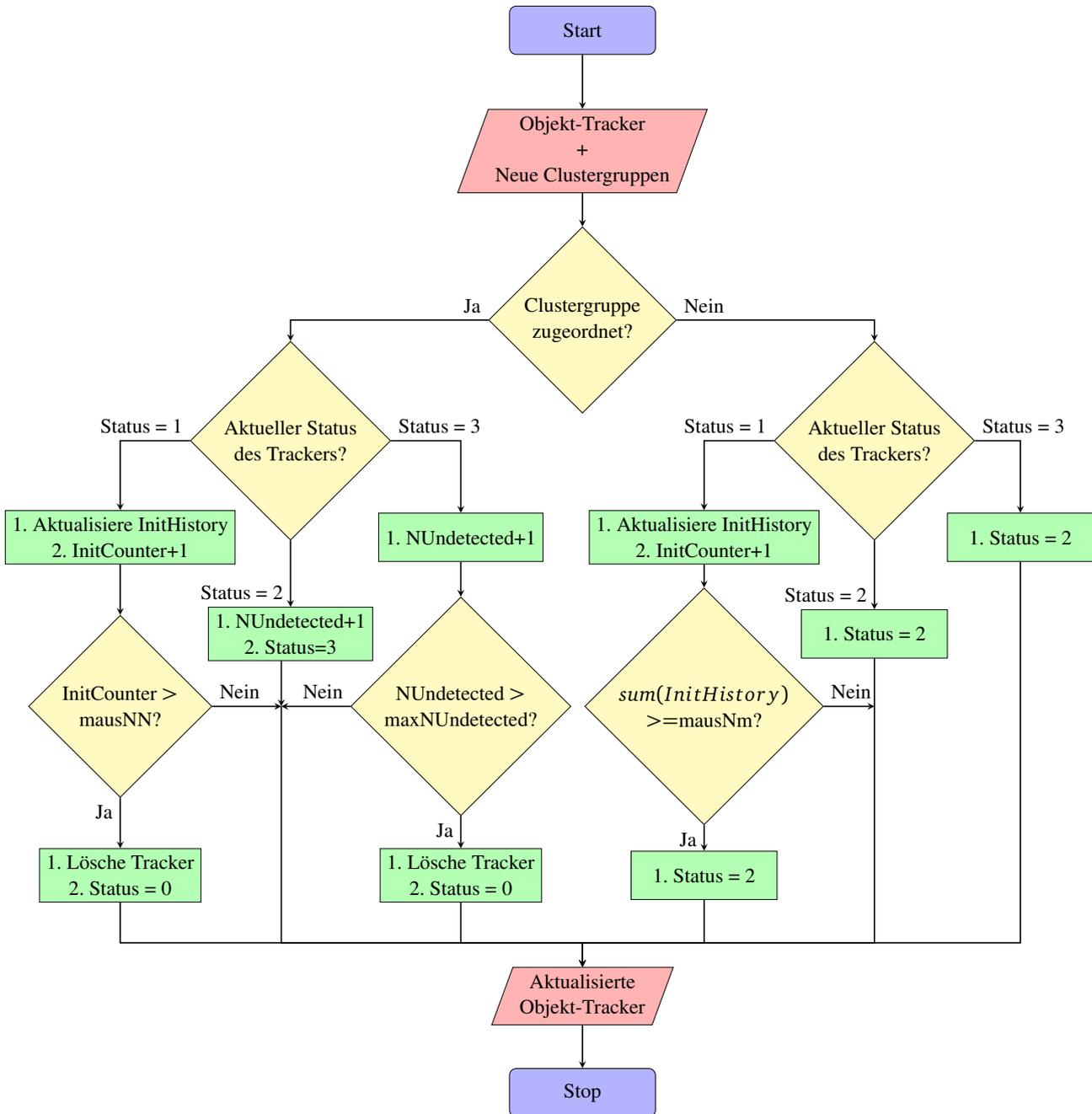


Abbildung A-6: Programmablaufplan (nach ISO 5807/DIN 66001): Aktualisierung der Tracker-Status

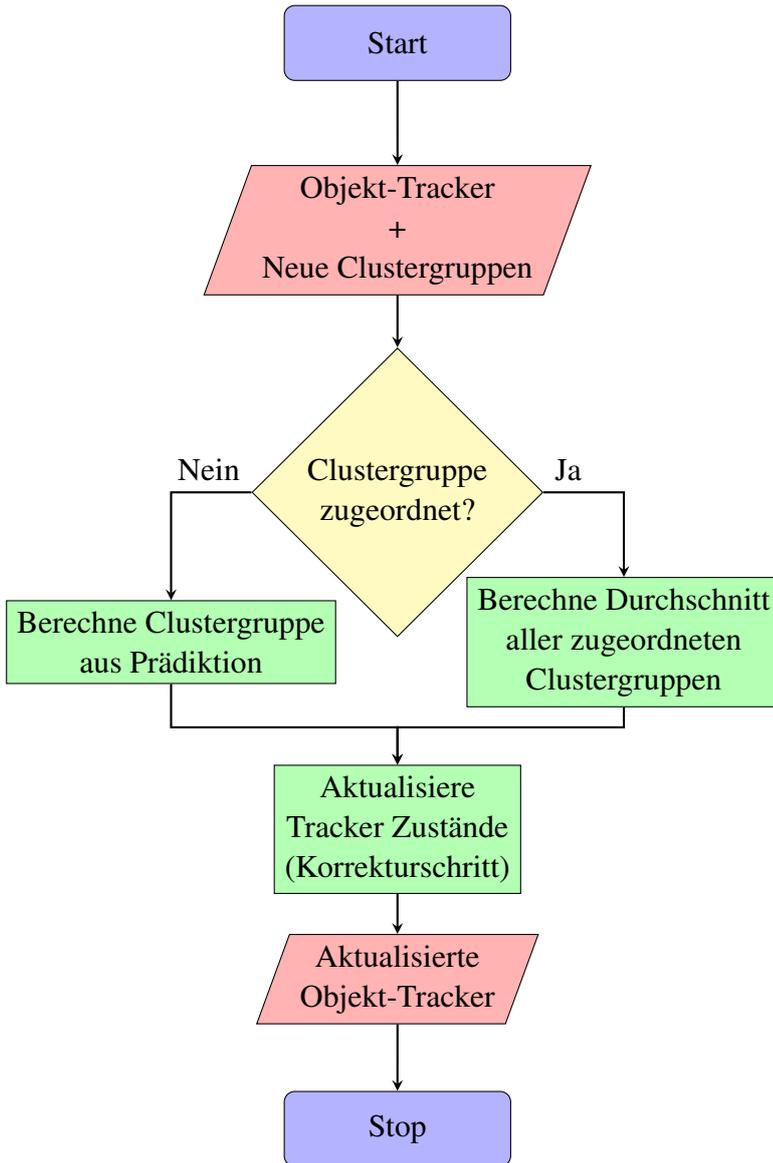


Abbildung A-7: Programmablaufplan (nach ISO 5807/DIN 66001): Korrektur und Prädiktion der Objekt-Tracker-Zustände



Abbildung A-8: Programmablaufplan (nach ISO 5807/DIN 66001): Löschen von Tracker-Duplikaten

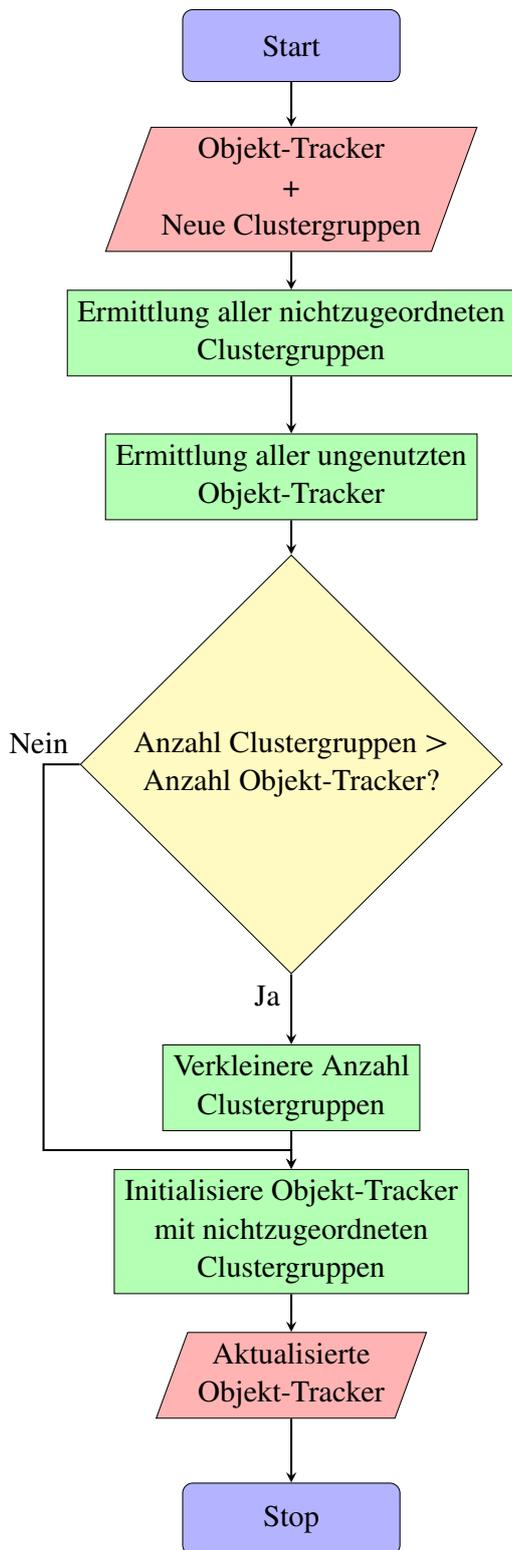


Abbildung A-9: Programmablaufplan (nach ISO 5807/DIN 66001): Initialisierung neuer Objekt-Tracker

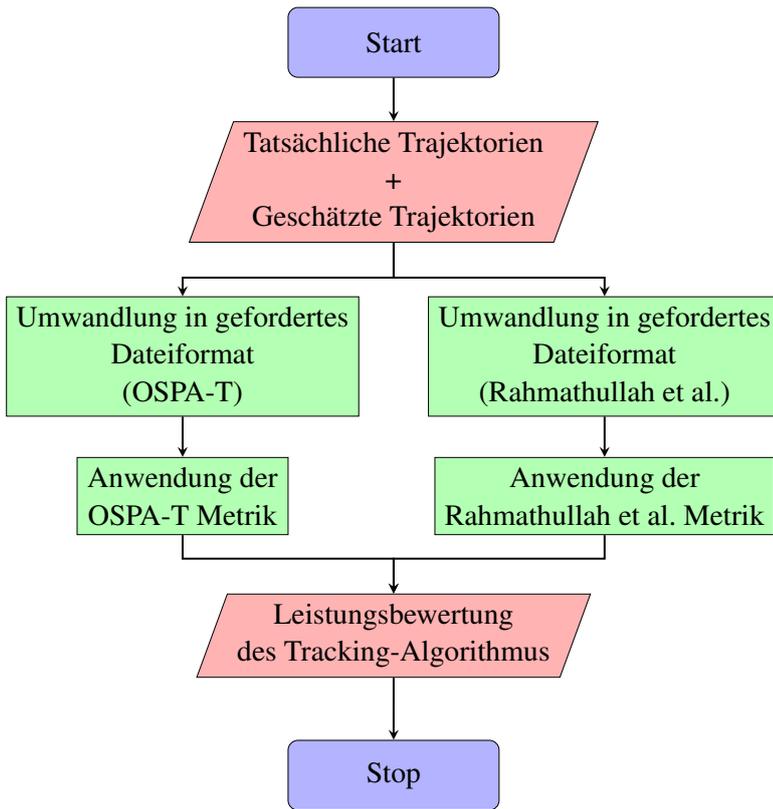


Abbildung A-10: Programmablaufplan (nach ISO 5807/DIN 66001): Bewertung der Tracking-Qualität

A.2 Testfall 1: Bewegungsgleichungen Objekt 1

$$\begin{aligned}\ddot{x}_1 &= 0 \\ \dot{x}_1 &= v_1 \\ x_1 &= v_1 \cdot t + 5\end{aligned}\tag{A-1}$$

$$\begin{aligned}\ddot{y}_1 &= 0 \\ \dot{y}_1 &= 0 \\ y_1 &= 0\end{aligned}\tag{A-2}$$

A.3 Testfall 1: Bewegungsgleichungen Objekt 2

$$\begin{aligned}
 \ddot{x}_2 &= \begin{cases} a_l \cdot \cos(\alpha_l), & \text{für } t \leq t_1 \\ 0, & \text{für } t_1 < t \leq t_3 \\ a_l \cdot \cos(\alpha_l), & \text{für } t_3 < t \leq t_4 \end{cases} \\
 \dot{x}_2 &= \begin{cases} -a_l \cdot \cos(\alpha_l) \cdot t + \dot{x}_{2,t=0}, & \text{für } t \leq t_1 \\ v_1, & \text{für } t_1 < t \leq t_3 \\ a_l \cdot \cos(\alpha_l) \cdot (t - t_3) + v_1, & \text{für } t_3 < t \leq t_4 \end{cases} \quad (\text{A-3}) \\
 x_2 &= \begin{cases} -\frac{1}{2}a_l \cdot \cos(\alpha_l) \cdot t^2 + \dot{x}_{2,t=0} \cdot t + x_{2,t=0}, & \text{für } t \leq t_1 \\ v_1 \cdot (t - t_1) + x_1(t_1), & \text{für } t_1 < t \leq t_3 \\ \frac{1}{2}a_l \cdot \cos(\alpha_l) \cdot (t - t_3)^2 + v_1 \cdot (t - t_3) + x_1(t_3), & \text{für } t_3 < t \leq t_4 \end{cases}
 \end{aligned}$$

$$\begin{aligned}
 \ddot{y}_2 &= \begin{cases} -a_l \cdot \sin(\alpha_l), & \text{für } t \leq t_1 \\ 0, & \text{für } t_1 < t \leq t_3 \\ -a_l \cdot \sin(\alpha_l), & \text{für } t_3 < t \leq t_4 \end{cases} \\
 \dot{y}_2 &= \begin{cases} a_l \cdot \sin(\alpha_l) \cdot t + \dot{y}_{2,t=0}, & \text{für } t \leq t_1 \\ 0, & \text{für } t_1 < t \leq t_3 \\ a_l \cdot \sin(\alpha_l) \cdot (t - t_3), & \text{für } t_3 < t \leq t_4 \end{cases} \quad (\text{A-4}) \\
 y_2 &= \begin{cases} \frac{1}{2}a_l \cdot \sin(\alpha_l) \cdot t^2 + \dot{y}_{2,t=0} \cdot t + y_{2,t=0}, & \text{für } t \leq t_1 \\ y_{l,min}, & \text{für } t_1 < t \leq t_3 \\ \frac{1}{2}a_l \cdot \sin(\alpha_l) \cdot (t - t_3)^2 + y_{l,min}, & \text{für } t_3 < t \leq t_4 \end{cases}
 \end{aligned}$$

A.4 Testfall 1: Bewegungsgleichungen Objekt 3

$$\begin{aligned}
 \ddot{x}_3 &= \begin{cases} a_r \cdot \cos(\alpha_r), & \text{für } t \leq t_2 \\ 0, & \text{für } t_2 < t \leq t_3 \\ a_r \cdot \cos(\alpha_r), & \text{für } t_3 < t \leq t_4 \end{cases} \\
 \dot{x}_3 &= \begin{cases} -a_r \cdot \cos(\alpha_r) \cdot t + \dot{x}_{3,t=0}, & \text{für } t \leq t_2 \\ v_1, & \text{für } t_2 < t \leq t_3 \\ a_r \cdot \cos(\alpha_r) \cdot (t - t_3) + v_1, & \text{für } t_3 < t \leq t_4 \end{cases} \quad (\text{A-5}) \\
 x_3 &= \begin{cases} -\frac{1}{2}a_r \cdot \cos(\alpha_r) \cdot t^2 + \dot{x}_{3,t=0} \cdot t + x_{3,t=0}, & \text{für } t \leq t_2 \\ v_1 \cdot (t - t_1) + x_1(t_1), & \text{für } t_2 < t \leq t_3 \\ \frac{1}{2}a_r \cdot \cos(\alpha_r) \cdot (t - t_3)^2 + v_1 \cdot (t - t_3) + x_1(t_3), & \text{für } t_3 < t \leq t_4 \end{cases}
 \end{aligned}$$

$$\begin{aligned}
 \ddot{y}_3 &= \begin{cases} -a_r \cdot \sin(\alpha_r), & \text{für } t \leq t_2 \\ 0, & \text{für } t_2 < t \leq t_3 \\ -a_r \cdot \sin(\alpha_r), & \text{für } t_3 < t \leq t_4 \end{cases} \\
 \dot{y}_3 &= \begin{cases} a_r \cdot \sin(\alpha_r) \cdot t + \dot{y}_{3,t=0}, & \text{für } t \leq t_2 \\ 0, & \text{für } t_2 < t \leq t_3 \\ a_r \cdot \sin(\alpha_r) \cdot (t - t_3), & \text{für } t_3 < t \leq t_4 \end{cases} \quad (\text{A-6}) \\
 y_3 &= \begin{cases} \frac{1}{2}a_r \cdot \sin(\alpha_r) \cdot t^2 + \dot{y}_{3,t=0} \cdot t + y_{3,t=0}, & \text{für } t \leq t_2 \\ y_{r,min}, & \text{für } t_2 < t \leq t_3 \\ \frac{1}{2}a_r \cdot \sin(\alpha_r) \cdot (t - t_3)^2 + y_{r,min}, & \text{für } t_3 < t \leq t_4 \end{cases}
 \end{aligned}$$

Abschließend werden die Bewegungsgleichungen der y-Richtung von Objekt 3 an der x-Achse gespiegelt.

$$\begin{aligned}
 \dot{y}_3 &= -\dot{y}_3 \\
 y_3 &= -y_3
 \end{aligned} \quad (\text{A-7})$$

A.5 Einzelbild aus dem Objekt-Tracking 2

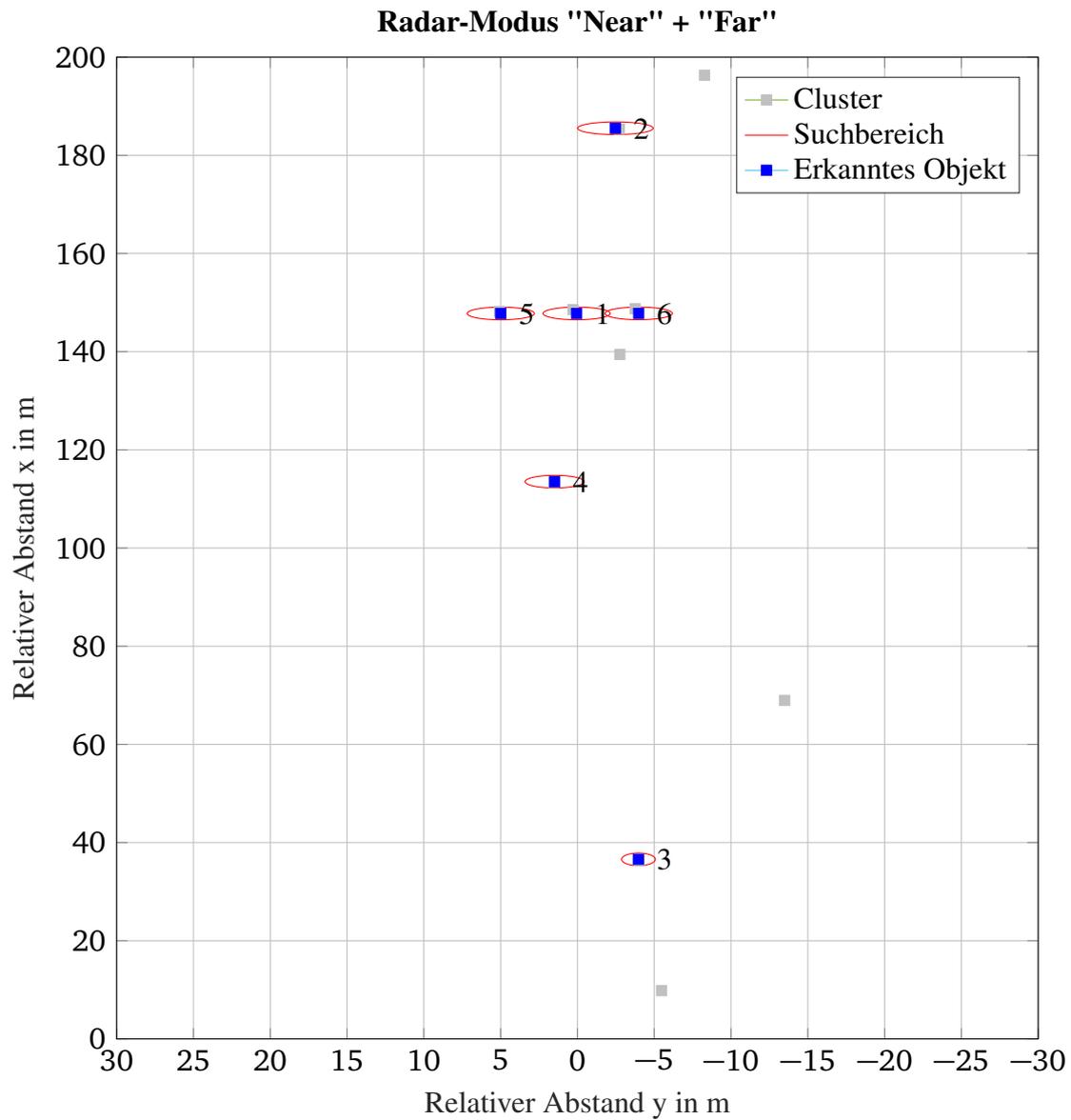


Abbildung A-11: Einzelbild aus dem Objekt-Tracking. Objekte 2,3 und 4 sind statisch. Objekt 1 ist das mittlere und die Objekte 5 und 6 sind die jeweils rechten bzw. linken Objekte von Testfall 1. Alle verbleibenden grauen Kästchen sind Clutter.

Alle drei dynamischen Objekte befinden sich auf gleicher Höhe und bewegen sich mit identischer Geschwindigkeit.

A.6 Einzelbild: Tracking mit realen Radardaten 2

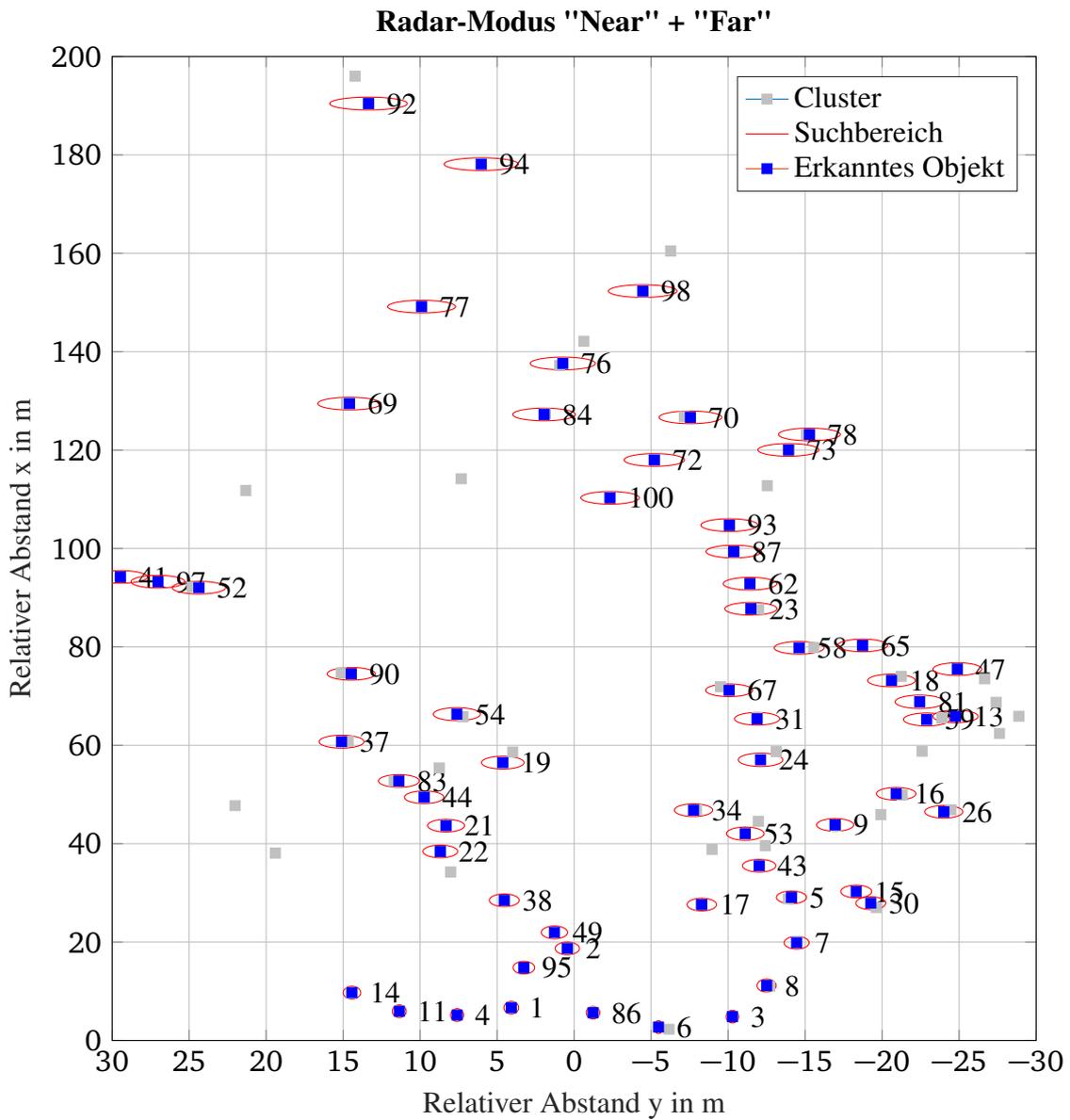


Abbildung A-12: Einzelbild: Tracking mit realen Radardaten 2. Das Bild zeigt eine Vielzahl von statischen Objekten. Objekte 37 (Position ca: $x=60\text{m}$, $y=15\text{m}$) und 54 (Position ca: $x=65\text{m}$, $y=8\text{m}$) sind die beiden verfolgten dynamischen Fahrzeuge. Sie nähern sich dem Sensor.

Literaturverzeichnis

Attari, M.: SVSF Estimation for Target Tracking with Measurement Origin Uncertainty (2016)

Attari, M.: “SVSF Estimation for Target Tracking with Measurement Origin Uncertainty”, (2016)

Autor unbekannt: gaussian prob (2017)

Autor unbekannt: gaussian prob, URL: https://www.ee.columbia.edu/~dpwe/e4896/code/prac10/gaussian_prob.m.html, (besucht am 20.02.2017), (2017)

Balzer, P.: Das Kalman Filter einfach erklärt Teil 2 (2013)

Balzer, P.: Das Kalman Filter einfach erklärt Teil 2, URL: <http://www.cbcity.de/das-kalman-filter-einfach-erklart-teil-2>, (besucht am 13.02.2017), (2013)

Bar-Shalom, Y., Daum, F.; Huang, J.: The probabilistic data association filter (2009)

Bar-Shalom, Y., Daum, F.; Huang, J.: “The probabilistic data association filter”, in: *IEEE Control Systems*, 29, (2009)

Baum, M., Willett, P. K.; Hanebeck, U. D.: On Wasserstein Barycenters and MMOSPA Estimation (2015)

Baum, M., Willett, P. K.; Hanebeck, U. D.: “On Wasserstein Barycenters and MMOSPA Estimation”, in: *IEEE Signal Processing Letters*, 22, S. 1511–1515, DOI: 10.1109/LSP.2015.2410217, (2015)

Bento, J.: A metric for sets of trajectories that is practical and mathematically consistent (2016)

Bento, J.: A metric for sets of trajectories that is practical and mathematically consistent, URL: <http://arxiv.org/pdf/1601.03094>, (2016)

Bernardin, K.; Stiefelhagen, R.: Evaluating Multiple Object Tracking Performance (2008)

Bernardin, K.; Stiefelhagen, R.: “Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics”, in: *EURASIP Journal on Image and Video Processing*, 2008, S. 1–10, DOI: 10.1155/2008/246309, (2008)

Blackman, S.; Popoli, R.: Design and analysis of modern tracking systems (1999)

Blackman, S.; Popoli, R.: Design and analysis of modern tracking systems, Artech House radar library, Artech House, ISBN: 9781580530064, (1999)

Brosseit, P., Rapp, M., Appenrodt, N.; Dickmann, J.: Probabilistic rectangular-shape estimation for extended object tracking (2016)

Brosseit, P., Rapp, M., Appenrodt, N.; Dickmann, J.: “Probabilistic rectangular-shape estimation for extended object tracking”, in: *2016 IEEE Intelligent Vehicles Symposium (IV)*, S. 279–285, DOI: 10.1109/IVS.2016.7535398, (2016)

Buhren, M.; Yang, B.: A Global Motion Model (2007)

Buhren, M.; Yang, B.: “A Global Motion Model for Target Tracking in Automotive Applications”, in: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, DOI: 10.1109/ICASSP.2007.366235, (2007)

Bundesanstalt für Straßenwesen: Rechtsfolgen zunehmender Fahrzeugautomatisierung (2012)

Bundesanstalt für Straßenwesen: Rechtsfolgen zunehmender Fahrzeugautomatisierung, URL: <http://www.bast.de/DE/Publikationen/Foko/2013-2012/2012-11.html>, (besucht am 03.04.2017), (2012)

Bundesministerium für Wirtschaft und Energie: Homepage Projekt Pegasus (2016)

Bundesministerium für Wirtschaft und Energie: Projekt PEGASUS, URL: <http://www.pegasus-projekt.info/de/about-PEGASUS>, (besucht am 19.02.2017), (2016)

Challa, S., Morelande, M., Musicki, D.; Evans, R. J.: Fundamentals of object tracking (2011)

Challa, S., Morelande, M., Musicki, D.; Evans, R. J.: Fundamentals of object tracking, Cambridge University Press, ISBN: 9780521876285, (2011)

Dubin, U.: Probabilistic Data Association Filters PDAF - a tracking demo (2016)

Dubin, U.: Probabilistic Data Association Filters PDAF - a tracking demo, URL: <https://de.mathworks.com/matlabcentral/fileexchange/34146-probabilistic-data-association-filters--pdaf--a-tracking-demo>, (besucht am 01.02.2017), (2016)

ECSEL Joint Undertaking: Homepage ENABLE-S3 (2016)

ECSEL Joint Undertaking: Homepage ENABLE-S3, URL: <http://www.enable-s3.eu/>, (besucht am 29.05.2017), (2016)

Encyclopedia of Mathematics: Definition einer Metrik (2017)

Encyclopedia of Mathematics: Definition einer Metrik, URL: <https://www.encyclopediaofmath.org/index.php/Metric>, (besucht am 24.04.2017), (2017)

Göbel, J.: Radartechnik (2011)

Göbel, J.: Radartechnik: Grundlagen und Anwendungen, 2. Auflage, VDE-Verl., ISBN: 978-3-8007-3141-1, (2011)

Gomes, J. B. B.: An overview on target tracking using multiple model methods (2008)

Gomes, J. B. B.: "An overview on target tracking using multiple model methods", in: *Instituto Superior Tecnico*, (2008)

Google: Homepage Google Scholar (2017)

Google: Homepage Google Scholar, URL: <https://scholar.google.de/>, (besucht am 25.01.2017), (2017)

Granstrom, K., Baum, M.; Reuter, S.: Extended Object Tracking (2016)

Granstrom, K., Baum, M.; Reuter, S.: Extended Object Tracking: Introduction, Overview and Applications, URL: <http://arxiv.org/pdf/1604.00970>, (2016)

Grisetti, G., Stachniss, C., Arras, K.; Burgard, W.: Robotics 2 - Data Association (2009)

Grisetti, G., Stachniss, C., Arras, K.; Burgard, W.: Robotics 2 - Data Association, (besucht am 21.04.2017), (2009)

Grisetti, G., Stachniss, C., Arras, K.; Burgard, W.: Robotics 2 Target Tracking (2009)

Grisetti, G., Stachniss, C., Arras, K.; Burgard, W.: Robotics 2 Target Tracking, URL: <http://ais.informatik.uni-freiburg.de/teaching/ws09/robotics2/pdfs/rob2-12-tracking.pdf>, (besucht am 23.05.2017), (2009)

Guerrero, M., Svensson, L., Svensson, D.; Willett, P.: Shooting two birds with two bullets: How to find Minimum Mean OSPA estimates (2010)

Guerrero, M., Svensson, L., Svensson, D.; Willett, P.: “Shooting two birds with two bullets: How to find Minimum Mean OSPA estimates”, in: *2010 13th International Conference on Information Fusion*, S. 1–8, DOI: 10.1109/ICIF.2010.5712056, (2010)

Hammarstrand, L., Svensson, L., Sandblom, F.; Sorstedt, J.: Extended Object Tracking using a Radar Resolution Model (2012)

Hammarstrand, L., Svensson, L., Sandblom, F.; Sorstedt, J.: “Extended Object Tracking using a Radar Resolution Model”, in: *IEEE Transactions on Aerospace and Electronic Systems*, 48, S. 2371–2386, DOI: 10.1109/TAES.2012.6237597, (2012)

Hazewinkel, M.: Hausdorff Metrik (2001)

Hazewinkel, M.: Encyclopaedia of mathematics - Hausdorff Metrik, URL: https://www.encyclopediaofmath.org/index.php/Hausdorff_metric, (besucht am 05.03.2017), (2001)

He, X., Tharmarasa, R., Kirubarajan, T.; Thayaparan, T.: A Track Quality Based Metric for Evaluating Performance of Multitarget Filters (2013)

He, X., Tharmarasa, R., Kirubarajan, T.; Thayaparan, T.: “A Track Quality Based Metric for Evaluating Performance of Multitarget Filters”, in: *IEEE Transactions on Aerospace and Electronic Systems*, 49, S. 610–616, DOI: 10.1109/TAES.2013.6404124, (2013)

Hoffman, J. R.; Mahler, R.: Multitarget Miss Distance via Optimal Assignment (2004)

Hoffman, J. R.; Mahler, R.: “Multitarget Miss Distance via Optimal Assignment”, in: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34, S. 327–336, DOI: 10.1109/TSMCA.2004.824848, (2004)

Hyun, E.; Lee, J.-H.: Multi-target tracking scheme using a track management table (2016)

Hyun, E.; Lee, J.-H.: “Multi-target tracking scheme using a track management table for automotive radar systems”, (2016)

IEEE: Homepage IEEE (2017)

IEEE: Homepage IEEE, URL: <https://www.ieee.org/index.html>, (besucht am 15.01.2017), (2017)

Kalman, R. E.: A New Approach to Linear Filtering and Prediction Problems (1960)

Kalman, R. E.: “A New Approach to Linear Filtering and Prediction Problems”, in: *Journal of Basic Engineering*, 82, S. 35, DOI: 10.1115/1.3662552, (1960)

Kellner, D., Barjenbruch, M., Klappstein, J., Dickmann, J.; Dietmayer, K.: Tracking of Extended Objects with High-Resolution Doppler Radar (2016)

Kellner, D., Barjenbruch, M., Klappstein, J., Dickmann, J.; Dietmayer, K.: "Tracking of Extended Objects with High-Resolution Doppler Radar", in: *IEEE Transactions on Intelligent Transportation Systems*, Jahrgang 17, S. 1341–1353, DOI: 10.1109/TITS.2015.2501759, (2016)

Kim, Y.-S.; Hong, K.-S.: An IMM algorithm for tracking maneuvering vehicles (2004)

Kim, Y.-S.; Hong, K.-S.: "An IMM algorithm for tracking maneuvering vehicles in an adaptive cruise control environment", in: *International Journal of Control Automation and Systems*, 2, S. 310–318, (2004)

Kuhn, H. W.: The Hungarian method for the assignment problem (1955)

Kuhn, H. W.: "The Hungarian method for the assignment problem", in: *Naval Research Logistics Quarterly*, 2, S. 83–97, DOI: 10.1002/nav.3800020109, (1955)

Lau, R. A.; Williams, J. L.: Tracking a coordinated group using expectation maximisation (2013)

Lau, R. A.; Williams, J. L.: "Tracking a coordinated group using expectation maximisation", in: *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, S. 282–287, DOI: 10.1109/ISSNIP.2013.6529803, (2013)

Ludloff, A.: Praxiswissen Radar und Radarsignalverarbeitung (2008)

Ludloff, A.: *Praxiswissen Radar und Radarsignalverarbeitung: Mit 22 Tabellen*, Praxis, 4. Auflage, Vieweg + Teubner, ISBN: 978-3-8348-0597-3, (2008)

Mahler, R.: Multitarget bayes filtering via first-order multitarget moments (2003)

Mahler, R.: "Multitarget bayes filtering via first-order multitarget moments", in: *IEEE Transactions on Aerospace and Electronic Systems*, 39, S. 1152–1178, DOI: 10.1109/TAES.2003.1261119, (2003)

Mahler, R.: Statistics 101 for multisensor, multitarget data fusion (2004)

Mahler, R.: "Statistics 101 for multisensor, multitarget data fusion", in: *IEEE Aerospace and Electronic Systems Magazine*, 19, S. 53–64, DOI: 10.1109/MAES.2004.1263231, (2004)

Marx, B. J.: Radareigenschaften von Personenkraftwagenkarosserien (2014)

Marx, B. J.: *Bewertungsverfahren für die Radareigenschaften von Personenkraftwagenkarosserien: Zugl.: Darmstadt, Techn. Univ., Diss., 2013, Fortschrittberichte VDI : Reihe 12, Verkehrstechnik, Fahrzeugtechnik, VDI-Verl., ISBN: 978-3-18-377512-5, (2014)*

MATLAB: File Exchange (2017)

MATLAB: File Exchange, URL: <https://de.mathworks.com/matlabcentral/fileexchange/>, (besucht am 26.04.2017), (2017)

Nagappa, S., Clark, D. E.; Mahler, R.: Incorporating track uncertainty into the OSPA metric (2011)

Nagappa, S., Clark, D. E.; Mahler, R.: "Incorporating track uncertainty into the OSPA metric", in: *14th International Conference on Information Fusion*, S. 1–8, (2011)

Nick: setdiff (2017)

Nick: setdiff, URL: <https://de.mathworks.com/matlabcentral/fileexchange/23172-setdiff>, (besucht am 04.02.2017), (2017)

P. Blom, H.: An efficient filter for abruptly changing systems (1984)

P. Blom, H.: “An efficient filter for abruptly changing systems”, in: *The 23rd IEEE Conference on Decision and Control. IEEE*, S. 656–658, DOI: 10.1109/CDC.1984.272089, (1984)

Panta, K., Clark, D. E.; Vo, B.-N.: Data Association and Track Management for the GMPHD (2009)

Panta, K., Clark, D. E.; Vo, B.-N.: “Data Association and Track Management for the Gaussian Mixture Probability Hypothesis Density Filter”, in: *IEEE Transactions on Aerospace and Electronic Systems*, 45, S. 1003–1016, DOI: 10.1109/TAES.2009.5259179, (2009)

Pulford, G. W.: A Survey of Manoeuvring Target Tracking Methods (2015)

Pulford, G. W.: “A Survey of Manoeuvring Target Tracking Methods”, in: *arXiv preprint arXiv:1503.07828*, (2015)

R, T.: Round Toward Vector of Values (2012)

R, T.: Round Toward Vector of Values, URL: <https://de.mathworks.com/matlabcentral/fileexchange/37674-round-toward-vector-of-values>, (besucht am 12.05.2017), (2012)

Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: Rahmathullah et al. Metrik (2016)

Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: A metric on the space of finite sets of trajectories for evaluation of multi-target tracking algorithms, URL: <http://arxiv.org/pdf/1605.01177>, (2016)

Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: GOSPA-Metrik (2016)

Rahmathullah, A. S., García-Fernández, Á. F.; Svensson, L.: Generalized optimal sub-pattern assignment metric, URL: <http://arxiv.org/pdf/1601.05585>, (2016)

Reuter, S.: Multi-object tracking using random finite sets (2014)

Reuter, S.: Multi-object tracking using random finite sets, Universität Ulm, DOI: 10.18725/OPARU-3204, (2014)

Richards, M. A.: Fundamentals of radar signal processing (2014)

Richards, M. A.: Fundamentals of radar signal processing, 2. Auflage, McGraw-Hill Education, ISBN: 978-0-07-179832-7, (2014)

Ristic, B., Vo, B.-N., Clark, D.; Vo, B.-T.: A Metric for Performance Evaluation of Multi-Target Tracking Algorithms (2011)

Ristic, B., Vo, B.-N., Clark, D.; Vo, B.-T.: “A Metric for Performance Evaluation of Multi-Target Tracking Algorithms”, in: *IEEE Transactions on Signal Processing*, 59, S. 3452–3457, DOI: 10.1109/TSP.2011.2140111, (2011)

Ristic, B., Arulampalam, S.; Gordon, N.: Beyond the Kalman filter (2004)

Ristic, B., Arulampalam, S.; Gordon, N.: Beyond the Kalman filter: Particle filters for tracking applications, Artech House radar library, Artech House, ISBN: 978-1-58053-631-8, (2004)

Ristic, B., Sherrah, J.; García-Fernández, Á. F.: OSPA-T Quellcode (2012)

Ristic, B., Sherrah, J.; García-Fernández, Á. F.: Performance Evaluation of Random Set Based Pedestrian Tracking Algorithms - Quellcode, URL: http://ba-ngu.vo-au.com/vo/OSPA_for_Tracks.zip, (2012)

Scheel, A.: Expertengespräch Tracking-Algorithmus-Metriken (2017)

Scheel, A.: "Expertengespräch Tracking-Algorithmus-Metriken, Telefonat", in: *Universität Ulm*, (2017)

Schlömer, N.: matlab2tikz (2017)

Schlömer, N.: matlab2tikz, URL: <https://de.mathworks.com/matlabcentral/fileexchange/22022-matlab2tikz-matlab2tikz>, (besucht am 04.04.2017), (2017)

Schuhmacher, D., Vo, B.-T.; Vo, B.-N.: OSPA-Metrik (2008)

Schuhmacher, D., Vo, B.-T.; Vo, B.-N.: "A Consistent Metric for Performance Evaluation of Multi-Object Filters", in: *IEEE Transactions on Signal Processing*, 56, S. 3447–3457, DOI: 10.1109/TSP.2008.920469, (2008)

Sharif, U.: Multi-Target Tracking using Random Finite Set based Bayesian Filtering in a Heterogeneous Platform (2016)

Sharif, U.: "Multi-Target Tracking using Random Finite Set based Bayesian Filtering in a Heterogeneous Platform: Masterthesis", (2016)

Simonsen, A.: Distribute figures (2015)

Simonsen, A.: Distribute figures, URL: <https://de.mathworks.com/matlabcentral/fileexchange/37176-distribute-figures>, (besucht am 01.02.2017), (2015)

Skolnik, M.: Radar handbook (2008)

Skolnik, M.: Radar handbook, 3. Auflage, McGraw Hill, ISBN: 978-0-07-148547-0, (2008)

Svensson, L., Svensson, D.; Willett, P.: Set JPDA algorithm for tracking unordered sets of targets (2009)

Svensson, L., Svensson, D.; Willett, P.: "Set JPDA algorithm for tracking unordered sets of targets", in: *2009 12th International Conference on Information Fusion*, S. 1187–1194, (2009)

ULB Darmstadt: Homepage ULB TU Darmstadt (2017)

ULB Darmstadt: Homepage ULB Darmstadt, URL: <https://www.ulb.tu-darmstadt.de/service/start/index.de.jsp>, (besucht am 17.01.2017), (2017)

Vo, B.-.-.n., Mallick, M., Bar-shalom, Y., Coraluppi, S., Osborne, R., Mahler, R., Vo, B.-.-.-t.; Osborne III, R.: Multitarget Tracking (2015)

Vo, B.-.-.n., Mallick, M., Bar-shalom, Y., Coraluppi, S., Osborne, R., Mahler, R., Vo, B.-.-.-t.; Osborne III, R.: "Multitarget Tracking", DOI: 10.1002/047134608X.W8275, (2015)

Vu, T.; Evans, R.: A new performance metric for multiple target tracking based on optimal subpattern assignment (2014)

Vu, T.; Evans, R.: “A new performance metric for multiple target tracking based on optimal subpattern assignment”, in: *17th International Conference on Information Fusion (FUSION)*, S. 1–8, (2014)

Winner, H., Hakuli, S., Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme (2015)

Winner, H., Hakuli, S., Lotz, F.; Singer, C.: *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*, ATZ/MTZ-Fachbuch, 3. Auflage, Springer Vieweg, ISBN: 978-3-658-05734-3, (2015)

Yu, M., Liu, C., Li, B.; Chen, W.-H.: An enhanced particle filtering method for GMTI radar tracking (2016)

Yu, M., Liu, C., Li, B.; Chen, W.-H.: “An enhanced particle filtering method for GMTI radar tracking”, (2016)