Integrated Multi-Object Tracking and Classification for Vehicle Environment Perception

Dem Fachbereich Maschinenbau und Verfahrenstechnik
der Technische Universität Kaiserslautern
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)
vorgelegte
Dissertation

von

M.Sc. Ganesh Nageswaran
aus Erode (Indien)

D 386

Tag der Einreichung: October 26, 2017
Tag der mündlichen Prüfung : August 09, 2018

Dekan: Prof. Dr.-Ing. Jörg Seewig
Vorsitzender: Prof. Dr.-Ing. Michael Günthner
1. Gutachter: Prof. Dr.-Ing. Rudolf Flierl
2. Gutachter: Prof. Dr.-Ing. Jörg Seewig

Declaration of Authorship

I, Ganesh NAGESWARAN, declare that this thesis titled, “Integrated Multi-Object Tracking and Classification for Vehicle Environment Perception” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Doctoral degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“*If you cannot explain it simply, you don’t understand it well enough.*”

Albert Einstein

Abstract

Road accidents remain as one of the major causes of death and injuries globally. Several million people die every year due to road accidents all over the world. Although the number of accidents in European region have reduced in the past years, road safety still remains a major challenge. Especially in case of commercial trucks, due to the size and load of the vehicle, even minor collisions with other road users would lead to serious injuries or death. In order to reduce number of accidents, automotive industry is rapidly developing advanced driver assistance systems (ADAS) and automated driving technologies. Efficient and reliable solutions are required for these systems to sense, perceive and react to different environmental conditions. For vehicle safety applications such as collision avoidance with vulnerable road users (VRUs), it is not only important for the system to efficiently detect and track the objects in the vicinity of the vehicle but should also function robustly.

An environment perception solution for application in commercial truck safety systems and for future automated driving is developed in this work. Thereby a method for integrated tracking and classification of road users in the near vicinity of the vehicle is formulated. The drawbacks in conventional multi-object tracking algorithms with respect to state, measurement and data association uncertainties have been addressed with the recent advancements in the field of unified multi-object tracking solutions based on random finite sets (RFS). Gaussian mixture implementation of the recently developed labeled multi-Bernoulli (LMB) filter [RSD15] is used as the basis for multi-object tracking in this work. Measurement from an high-resolution radar sensor is used as the main input for detecting and tracking objects.

On one side, the focus of this work is on tracking VRUs in the near vicinity of the truck. As it is beneficial for most of the vehicle safety systems to also know the category that the object belongs to, the focus on the other side is also to classify the road users. All the radar detections believed to originate from a single object are clustered together with help of density based spatial clustering for application with noise (DBSCAN) algorithm. Each cluster of detections would have different properties based on the respective object characteristics. Sixteen distinct features based on radar detections, that are suitable for separating pedestrians, bicyclists and passenger car categories are selected and extracted for each of the cluster. A machine learning based classifier is constructed, trained and parameterised for distinguishing the road users based on the extracted features.

The class information derived from the radar detections can further be used by the tracking algorithm, to adapt the model parameters used for precisely predicting the object motion according to the category of the object. Multiple model labeled multi-Bernoulli filter (MMLMB) is used for modelling different object motions. Apart from the detection level, the estimated state of an object on the tracking level also provides information about the object class. Both these informations are fused using Dempster-Shafer theory (DST) of evidence, based on respective class probabilities Thereby, the output of the integrated tracking and classification with MMLMB filter are classified tracks that can be used by truck safety applications with better reliability.

The developed environment perception method is further implemented as a real-time prototypical system on a commercial truck. The performance of the tracking and classification approaches are evaluated with the help of simulation and multiple test scenarios. A comparison of the developed approaches to a conventional converted measurements Kalman filter with global nearest neighbour association (CMKF-GNN) shows significant advantages in the overall accuracy and performance.

Kurzfassung

Verkehrsunfälle bleiben weltweit eine der Hauptursachen für Tod und Verletzungen. Mehrere Millionen Menschen sterben jedes Jahr aufgrund von Verkehrsunfällen auf der ganzen Welt. Obwohl sich die Zahl der Unfälle in Europa in den letzten Jahren reduziert hat, bleibt die Sicherheit im Straßenverkehr noch eine große Herausforderung. Besonders im Nutzfahrzeugsegment führen, aufgrund der Größe und Masse des Fahrzeugs selbst geringfügige Kollisionen mit anderen Verkehrsteilnehmern zu schweren Verletzungen oder zum Tod. Um die Anzahl der Unfälle zu reduzieren, werden neue ADAS-Funktionen und automatisierte Fahrtechnologien entwickelt. Für diese Systeme sind effiziente und zuverlässige Lösungen erforderlich, um unterschiedliche Umgebungsbedingungen zu erfassen, wahrzunehmen und zu reagieren. Für Fahrzeugsicherheitsanwendungen wie die Kollisionsvermeidung bei gefährdeten Verkehrsteilnehmern ist es nicht nur wichtig, dass das System die Objekte in der Nähe des Fahrzeugs effizient erfasst und verfolgt, sondern auch robust funktioniert.

In dieser Arbeit wird eine Umgebungswahrnehmungslösung für die Anwendung in LKW-Sicherheitssystemen und für zukünftiges automatisiertes Fahren entwickelt. Dabei wird ein Verfahren zur integrierten Verfolgung und Klassifizierung von Verkehrsteilnehmern im Nahbereich des Fahrzeugs formuliert. Die Nachteile herkömmlicher Multi-Objekt Verfolgungsalgorithmen bezüglich Zustands-, Mess- und Datenassoziations unsicherheiten wurden mit den Fortschritten auf dem Gebiet Random Finite Set (RFS) basierten Multi-Objekt Verfolgungsmethoden gelöst. Gaussian mixture implementierung des jüngst entwickelten *labeled multi-Bernoulli* (LMB) Filters [RSD15] wird in dieser Arbeit als Grundlage für die Multi-Objekt Verfolgung verwendet. Die Messung von einem hochauflösenden Radarsensor dient als Hauptdatenquelle zur Erkennung und Verfolgung von Objekten.

Auf der einen Seite liegt der Fokus dieser Arbeit auf der Verfolgung von Verkehrsteilnehmer in der Nähe des LKWs. Da es für die meisten Fahrzeug-Sicherheitssysteme vorteilhaft ist, auch den Objekttyp zu wissen, liegt der Fokus auf der anderen Seite auch darin, die Verkehrsteilnehmer zu klassifizieren. Alle Radarmessungen, die wahrscheinlich von einem einzigen Objekt stammen, werden mit Hilfe von *density based spatial clustering for application with noise* (DBSCAN) gruppiert. Jeder solche Cluster hätte unterschiedliche Merkmale basierend auf den jeweiligen Objekteigenschaften. Sechzehn verschiedene Merkmale basiert auf den Radarmessungen werden für jeden Cluster ausgewählt und extrahiert. Die extrahierte Merkmale dienen als Basis zur Klassifizierung von Fußgänger, Radfahrer und Personenkraftwagen. Zur Klassifikation wird ein maschinelles Lernverfahren verwendet, trainiert und parametrisiert, um die Verkehrsteilnehmer basierend auf den extrahierten Merkmalen zu unterscheiden.

Die von den Radarerfassungen abgeleiteten Klasseninformationen können weiter innerhalb des Tracking-Algorithmus verwendet werden, um die Parameter des Bewegungsmodells anzupassen. Ein *Multiple model multi-Bernoulli* (MMLMB) Filter wird zur Modellierung verschiedener Objektbewegungen verwendet. Neben der Erkennungsebene liefert der geschätzte Zustand eines Objekts auf der Verfolgungsebene auch Informationen über die

Objektklasse. Beide Informationen werden unter Verwendung der Dempster-Shafer Evidenztheorie fusioniert. Die Ausgabe der integrierten Multi-Objekt Verfolgung und Klassifikation mit dem MMLMB-Filter sind dabei klassifizierte Tracks, die für LKW-Sicherheitsanwendungen mit höherer Zuverlässigkeit verwendet werden können.

Die entwickelte Umgebungswahrnehmungsmethode wird im Echtzeit-prototypisches System auf einem Nutzfahrzeug implementiert. Die Performance der Tracking- und Klassifikationsansätze wird mit Hilfe von Simulation und mehreren Testszenarien ausgewertet. Ein Vergleich der entwickelten Ansätze zu einem konventionellen *converted measurements Kalman filter* mit *global nearest neighbour* Assoziierung (CMKF-GNN) zeigt signifikante Vorteile hinsichtlich der Gesamtgenauigkeit und Leistung.

Acknowledgements

This Dissertation is the result of my research work conducted as a scientific employee at Technical University of Kaiserslautern. The work was carried out in cooperation with the company Knorr-Bremse SfN GmbH in Schwieberdingen.

This work would not have been possible without the support of many people. First and foremost, I would like to thank my advisor Prof. Dr.-Ing. Rudolf Flierl for his supervision. Thank you for giving me the right directions and technical advises. I would also like to thank Prof. Dr.-Ing. Jörg Seewig for being the second examiner of the Dissertation. I am also thankful to Dr.-Ing. Eduard Gerum und Prof. Dr. Laszlo Palkovics for initiating the project on behalf of Knorr-Bremse SfN GmbH.

I would like to sincerely thank Dr.-Ing. Ulrich Gücker for many of the fruitful technical discussions, guidance and insights. I would also like to thank my fellow colleagues Andreas Nohl, Jochen Banspach, Johannes Schnier, Alexej Ginsheimer and Markus Klein at Knorr-Bremse for their technical discussions and support in performing measurements and vehicle tests. I am also thankful to Dr.-Ing. Hermann Buchner for his advice and insights right from my time as a master student in Germany.

I would like to thank my sister and my entire family for their inspiration and encouragement. I am grateful to my parents for providing me good education and giving me complete freedom to pursue what I believed in.

Most importantly I want to sincerely thank my wife Vidya for her love, care and support. Thank you for being my emotional pillar throughout the different phases of my PhD life.

Contents

Declaration of Authorship iii

Abstract vii

Kurzfassung ix

Acknowledgements xi

List of Abbreviations xvii

List of Symbols xxi

List of Figures xxv

List of Tables xxix

1 Introduction 1

2 State-of-art 5
- 2.1 ADAS and Automated Driving 5
  - 2.1.1 Categorisation of ADAS and automated systems 5
  - 2.1.2 General system architecture 9
  - 2.1.3 Safety and Legal Regulations 10
- 2.2 Sensors 10
  - 2.2.1 Camera 11
  - 2.2.2 Laserscanner 12
  - 2.2.3 Radar 13
    - 2.2.3.1 Pulse-Doppler-Radar 14
    - 2.2.3.2 FMCW-Radar 15
  - 2.2.4 Ultrasonic sensor 18
- 2.3 State-of-art perception scheme 18
- 2.4 Related work 20

3 Objectives 23

4 Technical Preliminaries and Methodology 25
- 4.1 Approach and Structure 25
- 4.2 Test vehicles and system setup 27
  - 4.2.1 Coordinate systems 28
  - 4.2.2 On-board sensors 28
- 4.3 Radar Sensor 30
  - 4.3.1 Properties 30

4.3.2 Calibration . . . 31
4.4 Software framework . . . 31
4.5 CAN bus structure . . . 33
4.6 Measurement techniques . . . 34
4.7 System and Software Schema . . . 34

5 Random Finite Sets based Multi-Object Tracking 39
5.1 Random Finite Sets . . . 40
5.2 Statistics and Classes of RFS . . . 42
5.2.1 Poisson Random Finite Set . . . 44
5.2.2 Identical Independently Distributed Cluster Random Finite Set . . . 44
5.2.3 Bernoulli Random Finite Set . . . 44
5.2.4 Multi-Bernoulli Random Finite Set . . . 45
5.3 Labeled Random Finite Set . . . 45
5.3.1 Labeled Multi-Bernoulli Random Finite Set . . . 46
5.3.2 Generalised Labeled Multi-Bernoulli RFS . . . 46
5.3.3 δ-Generalised Labeled Multi-Bernoulli RFS . . . 47
5.4 Multi-Object Bayes Filter . . . 47
5.4.1 Multi-Object Likelihood Functions . . . 48
5.4.2 Multi-Object Markov Densities . . . 50
5.5 Probability Hypothesis Density Filter . . . 51
5.6 Labeled Multi-Bernoulli Filter . . . 54
5.6.1 Prediction . . . 54
5.6.2 Update . . . 57
5.6.3 LMB Approximation and Track Management . . . 60
5.6.4 Adaptive Birth Density . . . 61
5.7 Multiple Model Labeled Multi-Bernoulli Filter . . . 61
5.7.1 Prediction . . . 62
5.7.2 Update . . . 64
5.8 Integrating Class Information into MMLMB Filter . . . 67
5.8.1 Dempster-Shafer Theory of Evidence . . . 68
5.8.2 GM-MMLMB Filter with Classification . . . 70

6 Radar based Feature Extraction and Classification 75
6.1 Segmentation . . . 76
6.1.1 Pre-processing . . . 76
6.1.2 DBSCAN Clustering . . . 76
6.2 Feature Extraction . . . 78
6.2.1 Model Pre-selection . . . 79
6.2.2 Geometric Shape Fitting . . . 79
6.2.3 Characteristic Feature Selection . . . 81
6.3 Classification . . . 87
6.3.1 Support Vector Machine . . . 88
6.3.2 Class probabilities based on track velocity . . . 93

7 Application and Realisation of Environment Perception for Truck ADAS 95
7.1 Measurement Characteristics . . . 95
7.2 State Models . . . 96
7.3 Measurement Model . . . 99

7.4 Ego-motion Compensation 101
7.5 Implementation of DST 103

8 Evaluation 105
8.1 Tracker Evaluation Metrics 105
  8.1.1 RMSE 105
  8.1.2 Filter Consistency 106
  8.1.3 Optimal Subpattern Assignment Metric 107
8.2 Example Tracking Scenarios and Results 107
  8.2.1 Simulation 107
  8.2.2 Test Scenario 1 112
  8.2.3 Test Scenario 2 116
  8.2.4 Test Scenario 3 116
  8.2.5 Performance Evaluation 123
8.3 Classification Evaluation 127
  8.3.1 Metrics 127
  8.3.2 Classification Performance 128

9 Conclusion 131
9.1 Summary 131
9.2 Future Work 132

A Fundamentals of State Estimation and Conventional Object Tracking 133
A.1 Bayes Filter 133
A.2 Kalman Filter 135
  A.2.1 Constant Gain Kalman Filter 137
  A.2.2 Extension of Kalman Filter for Non-Linear Systems 138
  A.2.3 Interacting Multiple Model Kalman Filter 140
A.3 Multi-object Tracking 143
  A.3.1 Gating 144
  A.3.2 Nearest Neighbour Data Association 145
  A.3.3 Probabilistic Data Association Methods 146

B Ground truth solution 151
B.1 Camera calibration 151
B.2 Marker detection 152

C DBSCAN algorithm 155

D Zadeh's Paradox in Dempster's rule of combination 157

Bibliography 159

Lebenslauf 173

List of Abbreviations

δ-GLMB	δ-generalised labeled multi-Bernoulli
3D	three-dimensional
ABS	anti-lock braking system
ACC	adaptive cruise control
ADAS	advanced driver assistance systems
AEB	advanced emergency braking
ASIL	automotive safety integrity level
B	bicylicst
BB	bounding box
BBA	basic belief assignment
BBM	basic belief mass
BSD	blind spot detection
C	car
CA	constant acceleration
CAN	controller area network
CB-MeMBer	cardinality balanced multi-target multi-Bernoulli
CCD	charge-coupled device
CMKF	converted measurements Kalman filter
CMMPHD	classifying multiple model probability hypothesis density
CMOS	complementary metal oxide semiconductor
CPHD	constant probability hypothesis density
CTA	city turning assist
CTRV	constant turn rate and velocity
CV	constant velocity
DBSCAN	density based spatial clustering for applications with noise

DDF	decoupled double filter
DGPS	differential global positioning system
DST	Dempster-Shafer theory of evidence
EKF	extended Kalman filter
ESP	electronic stability program
FCW	forward collision warning
FFT	fast Fourier transform
FISST	finite set statistics
FMCW	frequency modulated continuous wave
FN	false negative
FOD	frame of discernment
FoV	field of view
FP	false positive
FPS	frames per second
GLMB	generalised labeled multi-Bernoulli
GM-LMB	Gaussian mixture labeled multi-Bernoulli
GM-MMLMB	Gaussian mixture multiple model labeled multi-Bernoulli
GM-MMPHD	Gaussian mixture classifying multiple model probability hypothesis density
GM-MMPHD	Gaussian mixture multiple model probability hypothesis density
GMc	gaussian mixture component
GNN	global nearest neighbour
GPS	global positioning system
HAD	highly automated driving
i.i.d	independent identically distributed
IEPF	iterative end point fit
IMM	interacting multiple model
JIPDA	joint integrated probabilistic data association
JMS	jump Markov system
JPDA	joint probabilistic data association

LDWS	lane departutre warning system
LMB	labeled multi-Bernoulli
MABX	micro autobox
MAP	maximum a posteriori
MHD	Mahalanobis distance
MMLMB	multiple model labeled multi-Bernoulli
MMPHD	multiple model probability hypothesis density
NEES	normalised state estimation error squared
NIS	normalised innovation squared
NN	nearest neighbour
OSPA	optimal subpattern assignment
PA	parking assist
PCA	principal component analysis
PDA	probabilistic data association
PDF	probability density function
PHD	probability hypothesis density
RBF	radial basis function
RCS	radar cross section
RFS	random finite set
RMSE	root mean squared error
RMSPE	root mean squared position error
RMSVE	root mean squared velocity error
RP	reference point
RPT	rapid prototyping
SLAM	simultaneous localisation and mapping
SMC	sequential Monte-Carlo
SNR	signal to nose ratio
SoC	system-on-chip
SVM	support vector machine

TBM	transferable belief model
TN	true negative
TP	true positive
UKF	unscented Kalman filter
VRU	vulnerable road users

List of Symbols

Spaces

$\mathbb{B}$	label space of born objects
$\mathbb{C}$	index space
$\mathbb{L}$	label space of existing objects
$\mathbb{M}$	index space of current measurements
$\mathbb{N}$	space of natural numbers
$\mathbb{R}$	space of real numbers
$\mathbb{X}$	state space
$\mathcal{F}(\mathbb{X})$	finite subsets of $\mathbb{X}$
$\mathcal{F}_n(\mathbb{X})$	finite subsets of $\mathbb{X}$ with cardinality n
$\mathcal{L}$	projection from $\mathbb{X} \times \mathbb{L}$ to $\mathbb{L}$
Θ	track label to measurement assignments space
Ξ	discrete space

Densities

$\kappa(z)$	probability hypothesis density of clutter process
π	multi-object probability density
$\rho(n)$	cardinality distribution
$c(z)$	clutter probability density
$D(x)$	probability hypothesis density
$f_+(x\|\cdot)$	Markov transition density
p	spatial distribution

Sets

$\Delta(\mathbf{X})$	distinct label indicator
X	set of labeled state vectors
X	set of state vectors
Z	set of measurement vectors
I	set of track labels in a single hypothesis

L	set of track labels

Functions

$1_A(\mathrm{B})$	inclusion function
$\delta_A(\mathrm{B})$	generalised Kronecker delta function
d_{MHD}	Mahalanobis distance function
f	state transition function
h	measurement function

Measurement likelihood

$\eta_{\mathrm{Z}}^{(\theta)}(\ell)$	likelihood of association of track ℓ to measurement $\theta(\ell)$
$\eta_{\mathrm{Z}}^{(\theta)}(\ell \vert o)$	likelihood of association of track ℓ condition on mode o to measurement $\theta(\ell)$
$\psi_{\mathrm{Z}}(x;\theta)$	generalised measurement likelihood
$g(z \vert x)$	likelihood of measurement z given the object state x

Matrices

$\underline{\chi}_{o,o'}$	motion model transition matrix
$\underline{\mathrm{A}}$	assignment matrix
$\underline{\mathrm{F}}$	state transition matrix
$\underline{\mathrm{G}}$	gating matrix
$\underline{\mathrm{H}}$	measurement matrix
$\underline{\mathrm{I}}$	identity matrix
$\underline{\mathrm{K}}$	Kalman gain
$\underline{\mathrm{P}}$	estimation error covariance matrix
$\underline{\mathrm{Q}}$	process noise covariance matrix
$\underline{\mathrm{R}}$	measurement noise covariance matrix
$\underline{\mathrm{S}}$	innovation covariance matrix

Notations

$\langle f, g \rangle$	inner product of functions f and g
A	labeled random finite set
$\mathbf{a}$	labeled vector
A	random finite set
σ	standard deviation
θ	association hypothesis
$\underline{\mathrm{A}}$	matrix

$\underline{\mathrm{A}}^{-1}$	inverse of a matrix
$\underline{\mathrm{A}}^{\top}$	transpose of a matrix
a	vector or scalar value
$F1$	feature number 1
m_C	class-BBA
p^A	multi-object exponential

Specific probabilities

$\eta_S(\ell)$	state independent survival probability of track ℓ
$p(o)$	model probability
$p(x\|o)$	spatial distribution conditioned on model o
$p_D(x)$	object detection probability
$p_S(x)$	object survival probability
$q_D(x)$	probability of missed detection
$q_D(x)$	probability of object disappearance
r	existence probability

Subscripts

$\square_{+}$	predicted variable or density
$\square_{0:k}$	elements upto time step k
$\square_{B}$	variable or density w.r.t new born object
$\square_{C}$	variable or density w.r.t clutter processs
$\square_{ego}$	relates to ego vehicle
$\square_{k}$	time step k
$\square_{S2V}$	transformation - sensor to vehicle coordinates
$\square_{sensor}$	relates to sensor coordinates
$\square_{S}$	variable or density w.r.t surviving object
$\square_{veh}$	relates to vehicle coordinates

Scalars and vectors

$\dot{\psi}$	vehicle yaw rate
ℓ	track label
γ	residual of predicted and received measurements
$\hat{N}$	expected number of objects
$\hat{x}$	expected value of object spatial distribution $p(x)$
λ	expected value of Poisson distribution

$\mathbf{x}$	labeled object state vector
μ	IMM mixing probability
J	number of Gaussian mixtures
o	motion model
w	weight of a hypothesis or component
x	object state vector
z	sensor measurement vector
z_+	predicted measurement vector

List of Figures

1.1 Commercial trucks accident statistics according to [Sta15]. 1
1.2 Death rate of road users when met with an accident against a commercial vehicle[Sta15]. 2
1.3 Driving scenarios causing major accident of trucks with road users [Sta15]. . 2
1.4 A typical hazardous situation between bicyclist and truck during a turning maneuver. 3

2.1 Categorisation of ADAS applications and roadmap for fully automated driving of commercial trucks according to ERTRAC [Dri17]. 8
2.2 Modules of a typical autonomous driving system. 9
2.3 Example of image based bicyclist detection dataset [LFY$^+$16] 11
2.4 An high-definition laser scanner from the company Velodyne and the corresponding raw point cloud data of the laserscanner (source: Velodyne product website) 13
2.5 Transmitted and received signals in a pulse-Doppler radar [Bü08]. 14
2.6 Transmitted and received signals in a linear FMCW radar [Bü08]. 16
2.7 Two frequency ramps with different slopes 16
2.8 R-v diagram for a single object case 17
2.9 R-v diagram for multiple object detection with a 4-Chirp linear FMCW radar. 18
2.10 One of the typical vehicle environment perception schemes 19
2.11 Development of RFS based tracking methods 21

4.1 Structure of the dissertation 26
4.2 Both the test vehicles are installed with sensors and measurement equipments. The radar sensor mounting is as represented in Figure (A), enclosed by a green circle. 28
4.3 Figure at the top depicts the vehicle Cartesian coordinate system and Figure at the bottom illustrates the radar sensor coordinate system with respect to the vehicle coordinate system. 29
4.4 Used wheel-speed and steering angle sensors. 30
4.5 Illustration of field of view of the used radar sensor. 30
4.6 Setup with corner reflector for sensor position calibration and estimation of measurement characteristics. 31
4.7 Rapid prototyping framework with corresponding signal flow. 32
4.8 CAN bus structure 33
4.9 XCP communication structure (source: Vector Informatik GmbH product website). 34
4.10 Proposed system scheme for tracking and classification of objects using radar sensor. 36

5.1 Illustration of RFS based multi-object tracking. 41

5.2 Multi-Object Likelihood [Mah07b] 49
5.3 Multi-Object Markov Densities [Mah07b] 50
5.4 Visual illustration of probability hypothesis density. 52
5.5 LMB filter algorithm (cf. [RVVD14]) 56
5.6 Visualisation of belief, uncertainty and plausibility of a BBA (cf. [Mun11]) 68
5.7 Integrating class information into tracking framework. The state of the track is predicted and updated with GM-MMLMB filter equations on the upper state level. The class-BBA is predicted and updated in the lower class level. Object class is then derived by fusing BBAs from both the levels. 71

6.1 Illustration of the DBSCAN algorithm 77
6.2 Snapshot of a test sequence showing the results of detections clustering with DBSCAN. In the top Figure the cluster detections belonging to bicyclist are indicated by red markers, cluster detections belonging to corner of building are indicated in blue markers and unclustered individual detections in yellow. Bottom Figure is the corresponding web-cam snapshot. 78
6.3 Considered geometric shapes for detection clusters. The black color cross represents the corresponding reference point (RP) chosen for the fitted shape 81
6.4 Example measurement scenario illustrating geometric shape fit. The green bounding boxes belong to the passenger car, blue to the bicyclist and magenta to the pedestrian. For clarity purpose, the shape fit of pedestrian and bicyclist are shown only for their forward motion. 82
6.5 Boxplot of all the features belonging to different object classes. 83
6.6 Biplot of all the features projected onto first 3 PCs. The features are numbered in the same order as in Figure 6.5. 86
6.7 Biplot of reduced features projected onto PCs. 86
6.8 16-D feature data set scaled onto 3-D space 87
6.9 Linear SVM. 89
6.10 Non-linear SVM with kernel trick. 89
6.11 Rough RBF-Kernel parameter search by cross-validation with step-size 2. . 91
6.12 Medium scale RBF-Kernel parameter search by cross-validation with step-size 1. 92
6.13 Small scale RBF-Kernel parameter search by cross-validation with step-size 0.25. 93
6.14 Approximation of belief mass according to track velocity 94

7.1 Variance of the radar azimuth angle and range measurement with respect to the polar azimuth and range values are shown in Figure (A) & (C). Azimuth and range variance overlayed on the Cartesian x and y coordinates, with the sensor mounted looking towards rear is shown in Figure (B) & (D). The figure depicts the higher variance of the azimuth measurement along the border area of FoV, due to side lobes of the antenna. 97
7.2 Ego-motion Compensation 101

8.1 Simulated object trajectories withing the sensor FoV region. The mounting position of the sensor is assumed to be same as in section 4.2. The start positions of the objects are marked by a triangle and end positions by cross. 108
8.2 Cardinality estimate of the filter for the simulated object trajectories as in Figure 8.1. 109

8.3 The top plot shows the x position estimates of the filter and the bottom plot shows the y position estimates for the simulation scenario. Simulated sensor measurements along with clutter are also plotted 110
8.4 The first plot shows the OSPA distance with cut-off $c = 3$ m and order $p = 1$ for the simulated scenario. The second plot depicts the corresponding OSPA localisation and the third plot the OSPA cardinality. 111
8.5 Probabilities of the CV and CTRV motion models for a pedestrian target moving away orthogonally from the ego-vehicle corresponding to the simulated scenario. 111
8.6 Radar detections from pedestrian, bicyclist and passenger car corresponding to the test sequence 1. 112
8.7 OSPA distance and cardinality estimate. The first plot shows the OSPA distance of scenario 1 with cut-off $c = 3$ m and order $p = 1$. The gray background in the plot illustrates time instances when the camera ground truth for bicyclist is not available. The second plot depicts the corresponding cardinality estimate. 113
8.8 Top plot shows the estimated x positions of the objects and the bottom plot shows the estimated y positions corresponding to the test scenario 1. Radar detections and the approximated ground truth are also shown 114
8.9 Top plot shows the estimated longitudinal relative velocities and the bottom plot shows the estimated lateral relative velocities corresponding to the test scenario 1. 115
8.10 Figure (A) depicts the detections of the pedestrian and the bicyclist projected on to the camera image depicted by magenta points corresponding to test scenario 2. Green cross illustrates the marker position detected from the camera. The right top corner of the image is the origin of the radar. All the detections, including also the ones from background objects are shown in Figure (B). The origin of the radar in Figure (B) is (0,0). The cluster of bicyclist and pedestrian are illustrated as blue bounding box and circle, respectively. 117
8.11 Figure (A) shows the OSPA distance. Again the gray background in the plot illustrates time instances when the camera ground truth for bicyclist is not available. Figure (B) shows the cardinality estimates corresponding to test scenario 2. 118
8.12 Top plot shows the estimated x positions of the objects and the bottom plot shows the estimated y positions corresponding to the test scenario 2. Radar detections and the approximated ground truth are also shown. 119
8.13 Top plot shows the estimated longitudinal relative velocities and the bottom plot shows the estimated lateral relative velocities corresponding to the test scenario 2. 120
8.14 RMS position and velocity errors of bicyclist corresponding to test scenario 2. 121
8.15 Model probabilities of the bicyclist track corresponding to test scenario 2. . 121
8.16 Radar detections from two pedestrians corresponding to the test sequence 3. 122
8.17 OSPA distance and cardinality estimate corresponding to test scenario 3. . 123
8.18 Top plot shows the estimated x positions of the objects and the bottom plot shows the estimated y positions corresponding to the test scenario 3. . . . 124
8.19 Top plot shows the estimated longitudinal relative velocities and the bottom plot shows the estimated lateral relative velocities of the pedestrians corresponding to the test scenario 3. 125

8.20 NEES consistency metric for a bicyclist state estimation sequence. 126
8.21 RMSE values for bicyclist test scenarios corresponding to the GM-MMLMB filter with integrated classification and CMKF-GNN filter. 127
8.22 Confusion Matrix representing the results of radar measurement and track based object classification. Numbers represent the probability values. 128

A.1 Dynamic Bayesian Network. 134
A.2 IMM filter with two motion models 141
A.3 Multi-object tracking scenario with a bank of single-object state estimators. 144
A.4 chi-square distribution. 145
A.5 Example of an association conflict situation where the measurement $z^{(2)}$ lies within the gating region of both the tracks $T^{(1)}$ and $T^{(2)}$. 145
A.6 JIPDA hypotheses tree for two measurements and two tracks. 147

B.1 Camera calibration with checker board 151
B.2 Ground projected camera image. To be seen are the bright red markers attached to the wheels of the bicyle 152
B.3 Steps invovlved in detection of red marker positions attached to the bicycle 153

List of Tables

2.1 SAE International's levels of driving automation for on-road vehicles (copyright SAE International J3016). 7

4.1 Specifications of the used radar sensor. 31

7.1 Binary representation of the elements in the power set according to their order. 104

8.1 Classification performance for each class based on accuracy, sensitivity and fall-out. 129

For my parents

&

my wife Vidya

Chapter 1

Introduction

Safety has always been the major concern in the automotive industry. With increasing number of vehicles, size and power, safety of the driver as well as the road users need to be improved. The Federal Statistical Office in Germany ("Statistiches Bundesamt") releases yearly a report containing the statistics of road accidents. According to the 2015 report, totally around 2.52 million road accidents were recorded by the police in Germany, out of which almost 0.39 million people have been injured. The number of deaths from these accidents is reported as 3459, out of which almost 23% of the deaths involved atleast one commercial vehicle.

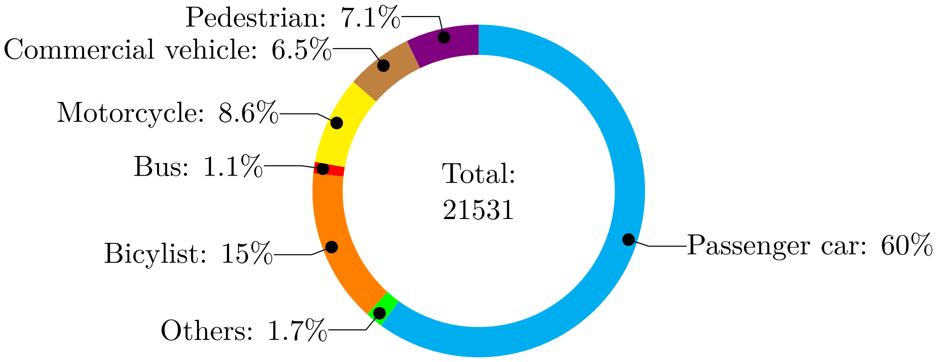


FIGURE 1.1: Commercial trucks accident statistics according to [Sta15].

In 2015, totally 21531 accidents between commercial vehicles and other road users have been reported. The fatality risk of other road users when involved in an accident with the commercial truck is four times higher than the fatality risk of the truck driver himself [Sta15]. This is majorly due to the size and load of a truck and the accident risk increases with increase in fleet. Figure 1.1 depicts the statistics of different road users involved in an accident with a commercial vehicle for the year 2015. Almost 60% of commercial vehicle accidents occurred with passenger car, followed by bicyclists at 15% and pedestrians at 7.1%. However, when involved in an accident with commercial vehicles, the death rate of pedestrians and bicyclists is higher than that of the occupants of a passenger car, as depicted in Figure 1.2.

Moreover, 49.1% of the accidents are reported to have occurred within the urban area compared to 27.1% in B-roads and 23% on the highways. The major cause of accidents is due to the error of commercial vehicle drivers by not maintaining a proper safety distance to the road users. The second major accident scenario, especially in urban conditions is

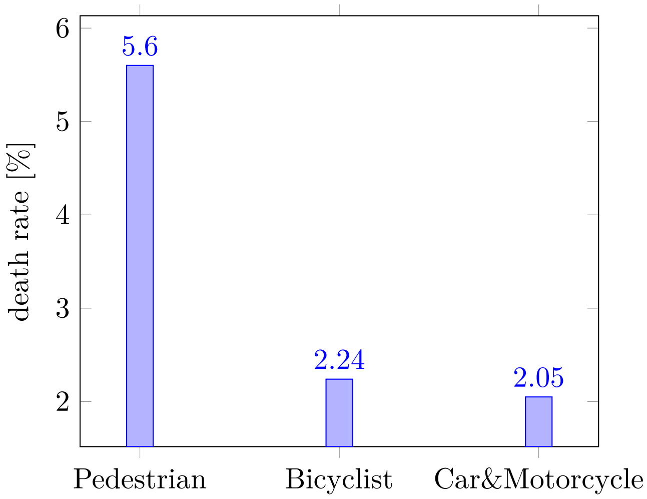


FIGURE 1.2: Death rate of road users when met with an accident against a commercial vehicle[Sta15].

during turning, reversing or driving in and out of the main street. Figure 1.3 depicts the various driving scenarios during which the error of the commercial vehicle driver causes accidents leading to injury and death of road users. One of the major reasons for higher

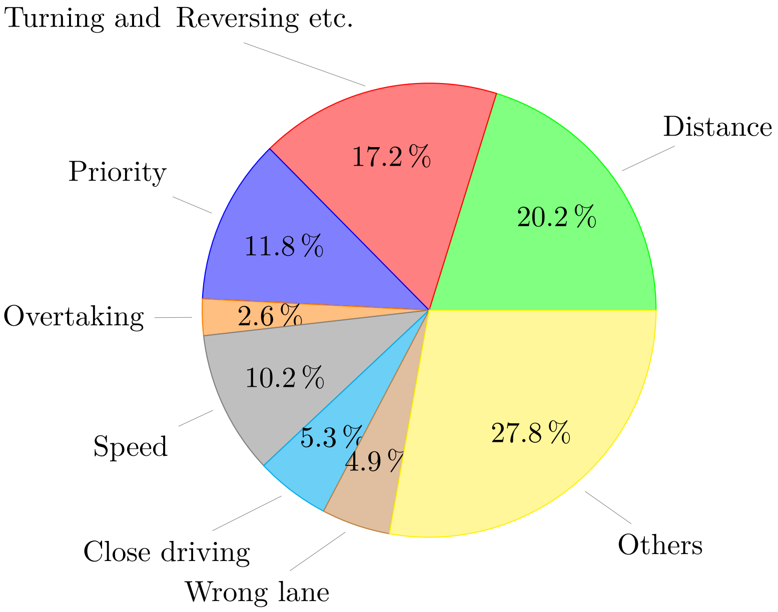


FIGURE 1.3: Driving scenarios causing major accident of trucks with road users [Sta15].

collision risk of commercial vehicles with road users in urban areas is due to the limited surrounding visibility for the truck drivers. Compared to the passenger car, the commercial truck drivers have a larger blind spot area surrounding the truck. The truck driver from his seating position or through the side mirrors cannot see most of the objects present in these close surrounding areas. Thereby the risk of accident is very high when a road user is present within any of the so called blind spot region. These problems motivate the need for developing safety systems which can avoid accidents of commercial vehicles involving other road users, majorly pedestrians and bicyclists in urban scenarios. The safety systems in such cases help to avoid accidents by detecting and perceiving objects in the near vicinity of truck through sensors, which can otherwise not be seen by the driver himself. From vehicle stability systems like anti-lock braking systems (ABS) and electronic stability program (ESP) to recent innovations in the field of advanced driver assistance systems (ADAS), the aim has been to reduce the number of accidents and improve traffic safety. As accidents are majorly caused due to human errors, many automotive manufacturers and suppliers are actively involved in research towards autonomous driving vehicles. New sensor technologies and intelligent algorithms play a vital role in achieving autonomous driving.

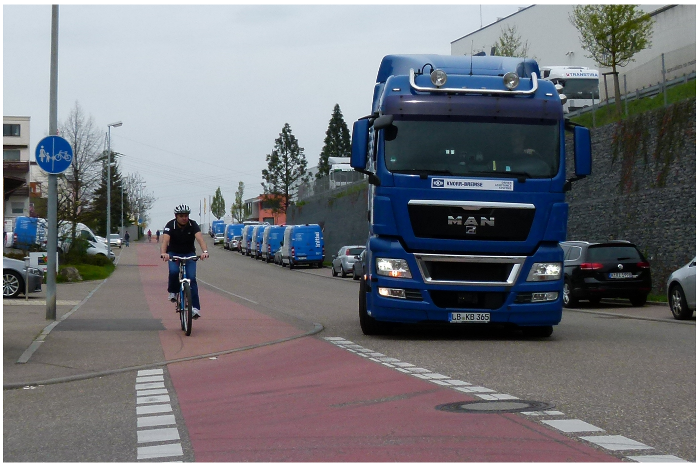


Figure 1.4: A typical hazardous situation between bicyclist and truck during a turning maneuver.

An ADAS application or an autonomous driving vehicle need to perform multiple tasks that a human would otherwise perform with errors. However, for achieving these tasks different functionalities need to be realised. One such high-level function is the environment perception which itself involves further low-level modules. Multi-object tracking is one of the important modules of the vehicle environment perception. Different aspects of multi-object tracking and classification required for realising vulnerable road user (VRU) safety applications in commercial trucks and future autonomous vehicles is the main focus area of this thesis. The aim of multi-object tracking in general is to estimate the number of objects in the environment and also their states, with uncertain sensor measurements as input. The estimation of object state is conventionally realised with the help of Bayes filters. Kalman filter is a well known example of the recursive Bayes filter. In case of single-object Bayes filter, an assumption of a single target and a corresponding single measurement is made. In order to track multiple objects, a bank of single-object Bayes filters are used. Additionally, which measurement arises from which object is not directly evident anymore. Therefore explicit track-to-measurement association is required. Data association algorithms are used to associate a track with a measurement, which would also consider the case that the measurement is just a clutter and does not belong to any real object or the object is not detected by sensor. In case of vehicle environment perception scenarios, typically many objects are present in the environment which means a complete consideration of all the possible track-to-measurement association hypotheses is not computationally tractable. Also, the tracker of an object doesn't directly have information about the tracker of another object and the interactions are often based on different heuristics. In the past decades, various advancements have been made in the field of multi-object tracking. Global nearest-neighbour (GNN), joint probabilistic data association (JPDA), joint integrated probabilistic data association (JIPDA) and multi-hypothesis tracking (MHT) are some of the prominent classical methods that have been developed to address the problem of data association and state estimation in target tracking.

Random Finite Sets (RFS) based multi-object tracking, since been proposed by Mahler, has seen many developments. The multi-object Bayes filter elaborated in [Mah07b], based on finite set statistics (FISST), extends the single-object distribution with multi-object distribution representing the uncertainty in the number of objects and their states. Many moment and parameter approximations of the Random Finite Set based multi-object Bayes filter have been derived. Probability hypothesis density (PHD) filter is an approximation of the multi-object Bayes filter which propagates only the first order moment of the unlabeled target RFS. A shortcoming of the PHD filter is the unstable cardinality estimation, which is improved in the cardinalized probability hypothesis density (CPHD) filter, which propagates the cardinality distribution along with the first order moment of the RFS. In contrast to the PHD and CPHD filters, the class of multi-Bernoulli filters propagates the parameters of the multi-Bernoulli distribution. Cardinality balanced multi-target multi-Bernoulli (CB-MeMBer) filter is once such implementation of multi-object Bayes recursion, where the multi-object posterior is approximated by the multi-Bernoulli distribution.

A drawback of the unlabeled RFS is the missing explicit consideration of the target identity. In order to address this, the labeled Random Finite Sets is introduced in [VV13a], where the object states are augmented with a label. Based on this, generalized labeled multi-Bernoulli (GLMB) filter and δ-generalized labeled multi-Bernoulli (δ-GLMB) filter are derived. Labeled multi-Bernoulli (LMB) filter introduced in [RVVD14] reduces the computational complexity of the δ-GLMB in the prediction step, but still maintaining the same update as in δ-GLMB. In LMB, the prior and posterior of the labeled RFS is approximated by a single hypothesis for each cardinality. In [Reu14] the performance of LMB is argued to be better than PHD and CPHD filters. Multiple model version of LMB (MMLMB) filter is derived in [RSD15] based on the Jump-Markov System.

In [MRSD14], the object class information is integrated into the multiple model PHD (MM-PHD) filter and proposes the classifying MMPHD (CMMPHD) filter. A Gaussian Mixture implementation of the CMMPHD is used for tracking road users using laser scanner and video camera at intersections. Dempster-Shafer theory is used for calculating certainty values of object classes, as it can effectively handle fuzzy data. The object class is updated in parallel to the PHD filter estimation. A similar approach is formulated in this work for the class of MMLMB filter. A method to integrate the object class information in the MMLMB filter is proposed and the feasibility is studied as an application for tracking road users in the near vicinity of the truck. Distinct object features are extracted from high-resolution radar detections. Machine learning based classifier is constructed with the help of object features, in order to classify different road users. The developed methods are further implemented in a prototype framework and evaluated with various test scenarios, with the focus of realising environment perception for ADAS applications and extending it to autonomous driving in the future.

Chapter 2

State-of-art

2.1 ADAS and Automated Driving

The introduction chapter reflects the increase in vehicle traffic since the advent of motor vehicles and the accident statistics of different driving and environment scenarios. On the other hand, it is also very important to find solutions to reduce accidents and improve overall road safety. Since the past decade, although the trend in traffic accidents has been downward, there is a constant need of new innovative electronic systems, not only to protect the vehicle passengers but also other road users. Passive safety systems like airbag and seatbelt protect the vehicle passengers by reducing the impact of collision on them, when the vehicle meets with an accident. Active safety systems like anti-lock braking system (ABS) and electronic stability program (ESP) significantly reduces the risk of accidents due to loss of vehicle stability, caused by environmental conditions and physical limits of vehicle dynamics. ABS supports the driver in maintaining the stability of the vehicle during braking, even in wet and icy road conditions. It operates mainly based on the wheel speed sensors, which indicate possible wheel lock during braking. ESP system on the other hand gets input from yawrate, lateral acceleration and steering sensors in addition to the wheel speed sensors, in order to detect the vehicle stability state. When the ESP system senses the instability of the vehicle, it brakes the wheels individually in order to correct under or overseer conditions and helps the driver to make the vehicle stable again.

However, the concept of ADAS came increasingly into safety applications, once the perception of the vehicle environment by sensors was made possible. By perception, one understands the ability of the system to detect, track and classify various objects in the vehicle environment. This information about the environment can then be used by the system to analyse the situation and react by warning the driver or intervening through actuators.

2.1.1 Categorisation of ADAS and automated systems

Based on the functionality, ADAS can further be grouped into three elementary categories:

- *Comfort and Efficiency enhancing systems:* ADAS which improves the driving comfort or helps in achieving better energy and fuel efficiency come under this category. Adaptive headlights is an example of a comfort system, which increases the visibility around curves by automatically redirecting the headlight beams based on the vehicle direction. ADAS like high-way truck platooning enables the trucks to drive one behind the other in a close distance, which results in an overall increase in fuel efficiency due to reduced air drag.

- *Driver information systems:* Systems which can provide information or warning to the driver, based on which the driver can react or control the vehicle come under this category. Lane departure warning system (LDWS), attention assist and blind spot detection (BSD) are some of the examples of this category of ADAS, where the driver is informed about the risks through flashing signals or acoustic warning.
- *Autonomous safety systems:* ADAS like advanced emergency braking (AEB), lane keeping assist (LKA) and city turning assist (CTA) with braking intervention are examples of systems that come under this category. When a collision or risk is predicted, the system can autonomously take over the control from the driver and react to the situation by braking or steering, in order to avoid accident.

Given the already high complexity of some of the ADAS, many automotive companies aim to make their vehicles fully autonomous in the coming decade. Autonomous driving is currently one of the hot research topics, as it involves many technical and scientific challenges, in order to be made a reality. Starting right from the PROMOTHEOUS project (PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety) in the eighties, the idea of autonomous driving is driven forward by many initiatives like DARPA (Defense Advanced Research Projects Agency) urban driving challenge, HAVEit and interactIVe research projects, to name a few. According to European Road Transport Research Advisory Council (ERTRAC), the main drivers of higher levels of automated driving given in [Dri17] are:

- *Safety:* Reduce accidents caused due to human errors.
- *Efficieny and environmental objectives:* Ensure smoother traffic by reducing traffic congestion and increase transport system efficiency by decreasing emissions and energy consumptions of vehicles.
- *Comfort:* The driver can spend time on other activities when the vehicle drives autonomously.
- *Social inclusion:* Ensure mobility for all, including elderly and impaired people.
- *Accessibility:* Improved access to urban areas and city centers.

In 2014, a new SAE international standard J3016: *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems* was established, defining levels of automation and common terminology for automated driving. Six levels of automation are defined from level 0 indicating "no automation" to level 5 indicating "full automation". A summary of SAE International's levels of driving automation for on-road vehicles is given in Table 2.1. Based on the J3016 levels of driving automation, Figure 2.1 shows the categorisation of different ADAS applications according to their respective automation level and the road map for complete autonomous driving of commercial trucks, as presented by ERTRAC [Dri17].

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/Deceleration	Monitoring of Driving Environment	Fallback Performance of Dynamic Driving Task	System Capability (Driving Modes)
Human driver monitors the driving environment						
0	No Automation	the full-time performance by the human driver of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	the driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	the driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task	System	Human driver	Human driver	Some driving modes
Automated driving system (“system”) monitors the driving environment						
3	Conditional Automation	the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene	System	System	Human driver	Some driving modes
4	High Automation	the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene	System	System	System	Some driving modes
5	Full Automation	the full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver	System	System	System	All driving modes

TABLE 2.1: SAE International’s levels of driving automation for on-road vehicles (copyright SAE International J3016).

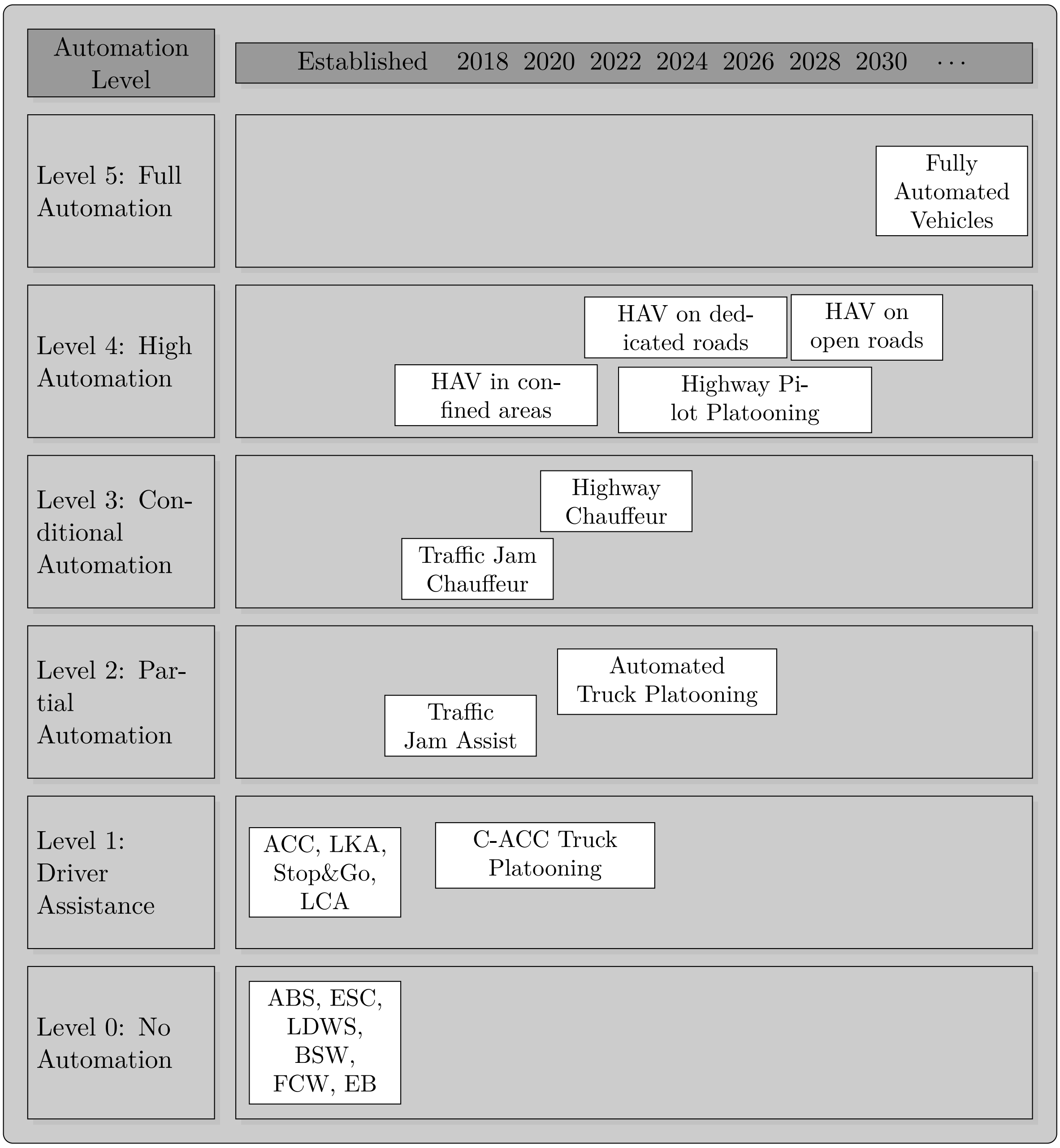


FIGURE 2.1: Categorisation of ADAS applications and roadmap for fully automated driving of commercial trucks according to ERTRAC [Dri17].

2.1.2 General system architecture

An autonomous driving system in general consists of many layers of modules based on different functionalities. A general architecture of an autonomous driving system consisting different functional modules is illustrated in Figure 2.2. Sensors and digital maps serve as the input to the system, containing information about the objects, obstacles in the vehicle environment and the geographical location. HMI module serves as an interface between the passenger and the vehicle. The perception layer is responsible for sensing and understanding of the environment, so that the host vehicle can know what are present in it's surrounding. Then the localisation module helps the host vehicle to know it's own position with respect to the extracted surroundings. The output of the fusion layer is well defined list of detected and tracked objects with their kinematic states like position and velocity, along with the kinematic states of the host vehicle itself. Based on the output of the perception and localisation modules, the motion planning layer generates a behaviour the host vehicle should follow. Moreover, a maneuverable trajectory is generated by the trajectory planner, considering the occupancy by objects and obstacles in the host vehicle's surrounding. The lowest layer is the motion control, where the controller tries to follow the generated trajectory from the previous layer, as close as possible. The motion controller comprises of the trajectory controller module responsible for the lateral and longitudinal control of the vehicle, as well as a reactive module responsible of brake functions.

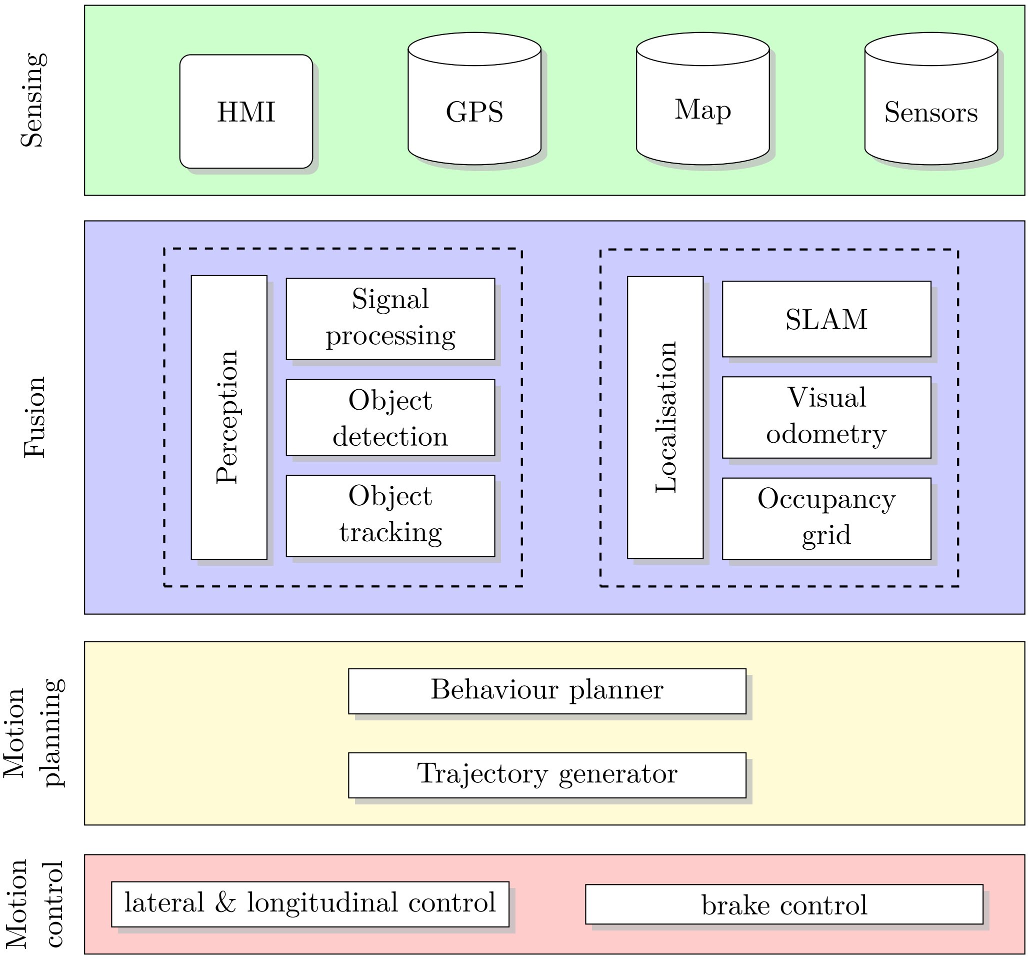


FIGURE 2.2: Modules of a typical autonomous driving system.

For partial automation systems and ADAS applications, the architecture need not contain all of the above mentioned functional modules. For example, a LDWS application which comes under SAE level 0 automation does not necessarily need behaviour or motion planing function. Therefore, a modular approach to the system and software development provides higher flexibility for realising various applications. Model-based development methods are being widely used in automotive system and embedded software development. This allows for a higher abstraction of various functionalities and formalises the development by use of models. Moreover the relationship between requirements, architecture and testing is made more transparent.

2.1.3 Safety and Legal Regulations

With the increasing complexity of systems from ADAS to automated driving, there is also a higher need to address the safety of these autonomous systems. The three levels of safety according to [MTBW17] are:

- *Safety level 1:* Safety with respect to product liability required for getting permission to launch a product in specific customer market.
- *Safety level 2:* Functional safety of mechatronic systems and mechanical parts.
- *Safety level 3:* Functional safety with respect to electrical and electronic (E/E) systems.

ISO 26262 *Road Vehicles-Functional Safety*, defined in 2011 is an international standard for functional safety of E/E systems in production automobiles. ISO 26262 standard defines guidelines regarding requirements and recommendations to reduce systematic development failures and handle the complexity of E/E systems [MTBW17]. Automotive safety integrity level (ASIL) is a risk classification scheme defined by ISO 26262 for E/E systems, based on the hazard level. ASIL is determined based on hazard analysis and risk assessment. Four categories of ASIL are ASIL A, ASIL B, ASIL C and ASIL D, with ASIL A representing the lowest integrity requirements and ASIL D representing the highest integrity requirements. Although ISO 26262 was initially defined for passenger car segment, it is being adapted in the commercial truck segment as well. Due to higher complexity of automated driving systems, enhancements are required to the existing ISO 26262 standards [MTBW17]. A newer version of the standard is expected to be officially released in 2018, addressing several aspects of autonomous driving.

2.2 Sensors

On of the main requirements to realise ADAS and automated driving functions is the ability to sense and gain detailed information about the vehicle surroundings. This is realised by using different sensors, with varying measurement techniques and capabilities. Based on the fundamental measurement principle, the sensors can be categorised as distance based sensors and image based sensors. While LIDAR, RADAR and Ultrasonic sensors are examples of distance based sensors, mono and stereo vision cameras rely on video and image processing. Each sensor has it’s own advantages and disadvantages and therefore in order to be efficient, the system should use the best combination of the sensors.

2.2.1 Camera

Cameras can capture high level of details about the surrounding. With camera, it is also possible to extract the color, contrast and texture of objects in the vehicle environment. Cameras are already widely used for realising many of the ADAS applications. From simple applications like LDWS, where a single front looking camera is mounted on the windshield, to advanced applications like 360° surround view, where multiple cameras are placed around the corners of the vehicle, cameras are vital for vehicle environment perception. Image formation in camera is achieved by it's photosensitive sensor. Light photons are converted into electrical charge, proportional to the intensity of light. Basically two types of sensors are used for this purpose, the charge-coupled device (CCD) and complementary metal oxide semiconductor (CMOS). In CCD sensor, the accumulated charge of a cell is read at one corner of the array, whereas in CMOS sensor the charge of each pixel is amplified by a transistor and is read individually. CCD sensors generally create high-quality images with low-noise compared to CMOS, but is highly sensitive to light. However, the CCD sensor consumes higher power and are costlier to produce, the CMOS sensor on the contrary requires lesser operating power and are cheaper to produce.

For automotive applications, both monocular and stereo vision cameras are used. A monocular camera captures the environment as a 2D image, either as a grayscale or color image. The grayscale camera in general has a better resolution than the color camera, as a separate red, blue and green color based filter for each pixel, called Bayer-matrix is required in color image. A monocular camera can be used for detecting objects based on template matching, pattern recognition or motion properties. The distance to the object is not given directly and is estimated by additional object model assumptions. A stereo camera has two mono-cameras placed with a small horizontal separation, which can be compared to the construction of an human eye. Both the cameras capture images of the environment separately, which can then be compared and synchronised to get depth information directly by means of disparity map. This also helps in a 3D representation of the environment.

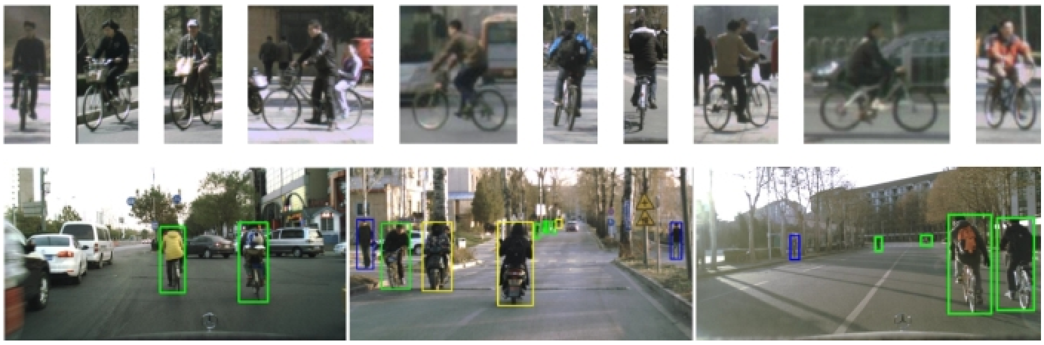


FIGURE 2.3: Example of image based bicyclist detection dataset [LFY+16]

Some of the important properties to be considered for the selection of a right camera and lens for an application are the focal length, resolution and frame-rate. Normal lenses on an average have a FoV of ≈ 45° and wide-angle lenses can have FoV > 100°. Although wide-angle lenses can cover a larger surrounding area around the vehicle, they have higher image distortion and poor resolution at farther distances, compared to normal lenses. In general, the camera resolution, usually represented in mega-pixels (MP), decides the distance upto which an object can be detected and extracted. Resolutions of 1MP and 2MP are widely used in automotive applications, with latest systems extending upto 8MP. While applications like forward collision warning (FCW) require lenses with narrower FoV and better resolution at higher distances, a Parking Assist (PA) application requires wide-angle or fish-eye lenses with a wider FoV. Similarly, frames per second (FPS) varies typically from 25 to 80 FPS, where an higher FPS is desirable for safety-critical systems like collision avoidance than for simple assisting systems. Another factor proportional to resolution and FPS is the computational effort. Higher data rate and resolution needs higher processing power. A constant development in the state-of-art system-on-chip (SoC) vision solutions would enable the usage of multiple cameras for future automated driving applications. Objects are detected by camera sensors based on various real-time image processing algorithms and corresponding object hypotheses. The object hypotheses are either based on object appearance characteristics like texture, edges, corners and shape, or based on motion characteristics extracted by optical flow algorithms. Further, constructing classifiers based on object characteristics, with the help of statistical and deep learning methods enable image based classification of objects. However, in case of extreme weather conditions like fog and heavy rain, it is difficult to detect objects and extract information about the environment. The functionality of the system will not be available in such conditions, if only camera is used. Therefore other variants of sensors and information sources are also required, to improve the system availability.

2.2.2 Laserscanner

Lidar is an acronym of light detection and ranging, where the distance to the object is measured with the help of laser light. The lidar sensor sends out pulsed laser light, part of which is then reflected back by objects and obstacles in the environment. The reflected pulse is then detected by the receiving diode. The distance to the object is then calculated by considering the time duration between transmission of the laser pulse and the reception of the corresponding pulse reflection from the object. If Δt is considered to be the time difference and c as speed of light, the distance to the object d can then be calculated with

$$d = \frac{\Delta t \cdot c}{2} \tag{2.1}$$

In a typical lidar sensor operating with infrared wavelength, the laser pulse is sent at a frequency of $\approx$ 900 nm. However, for calculating the object relative velocity based on Doppler frequency shift, high-frequency components and techniques are required, incurring high costs. Therefore lidar sensor based relative velocity measurement is not done for automotive applications [Ohl14]. Based on the operating principle, lidar sensors are basically categorised into two types as multibeam lidar and laserscanner. In a multibeam lidar, several photo diodes are arranged in parallel and are fixed. Whereas in a laserscanner, the laser beam from one or more diodes are dispersed in different directions by a rotating mirror. Laserscanners typically sends out laser beams in both horizontal and vertical directions. The horizontal angle is controlled by the rotation of the mirror and the elevation angle is controlled by tilt of the mirror. The laser beams can be ordered as several layers with angular offset as less as 0.1°. In general, laserscanners can have a horizontal FoV of more than 200°. High definition laserscanners with a rotating housing containing the diodes can reach FoV upto 360°. Therefore, lidar sensors in general have a higher angular resolution and therefore the form and shape of objects can be extracted very well. However, due to optical measurement principle of lidar sensors, they are very sensitive to rain, snow and fog, which leads to false sensor measurements in such climatic conditions. Also, objects or

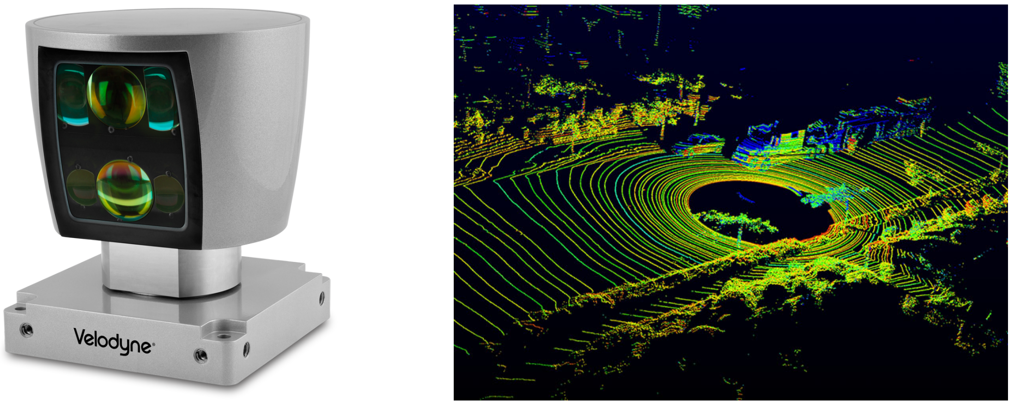

(A) Velodyne HDL-64-E laserscanner (B) Sensor raw data

Figure 2.4: An high-definition laser scanner from the company Velodyne and the corresponding raw point cloud data of the laserscanner (source: Velodyne product website)

.

obstacles with transparent surface cannot be detected by lidar sensors, as the light would pass through the object and will not be reflected back to the receiver diode. Moreover, due to bigger sizes of laserscanner, they can be mounted only in certain positions on the vehicle. The bigger size of laserscanner is mainly due to the housing required to contain it's spinning mechanical parts. This highly affects the practical integration of laserscanner into series systems. In order to overcome this disadvantage, newer lasercanner technologies with electronically steered laser beams instead of rotating mirrors are in development. Many companies like Velodyne and Quanergy have already showcased their prototypes of compact solid-state lidars, which are expected to be used in future ADAS and automated driving systems.

2.2.3 Radar

Radar (radio detection and ranging) is in use from as early as 1900s for marine and military applications for measuring distance. Radar works based on the principle of electro magnetic radiations for measuring the distance. Radar sensors for automotive applications typically sends out electromagnetic radiations in the frequency range of gigahertz. The transmitted electromagnetic radiations when reflected by an object or obstacle are absorbed back by a receiving antenna of the radar, also called as echo. The reflected wave absorbed by the receiver of the radar, contains a part of the transmitted wave energy, as some of the energy is absorbed by the air medium and the object itself. The relation between the transmitted power P_s and the received power after reflection from the object P_r can be given as [Ung94]

$$P_r = \frac{G \cdot \sigma \cdot A}{(4\pi)^2 \cdot R^4} \cdot P_s \tag{2.2}$$

where R is the distance to reflecting surface, G gain of transmitting antenna, A the effective area of receiving antenna and σ the radar cross section (RCS) of the object. RCS is typically the measure of object detectability. RCS of an object depends on it's size, type of surface material and reflective property. Generally, strongly reflecting objects like car and metal doors have higher RCS values than bicyclists and pedestrians. Distance measurement with automotive radar sensor is similar to the principle of lidar sensors as described in the previous section, but operates at a higher wavelength of electromagnetic spectrum. In addition to the range measurement, radar is also used for measuring angular position and relative velocity of the object. The relative velocity of the object is measured based on the Doppler effect. In automotive field, frequencies of 24 GHz, 77GHz and 79 GHz are used for various applications. However, as the 24 GHz frequency range is also used in astronomical applications, it is slowly getting outdated in automotive applications. Therefore new generation sensors use a frequency band of 76-77 GHz for long range applications and 77-81 GHz frequency band for short range applications.

Based on the functioning principle, radar can be separated into two categories as pulse-Doppler-radar and frequency modulated continuous wave (FMCW) radar.

2.2.3.1 Pulse-Doppler-Radar

In pulse-Doppler radars, the transmission antenna sends out high-frequency impulses periodically. When an object is present in the sensor FoV, the transmitted electromagnetic pulses hit the surface of the object and is reflected back to the radar. Figure 2.5 depicts the working principle of a pulse-Doppler-radar. By repeatedly switching the electromagnetic

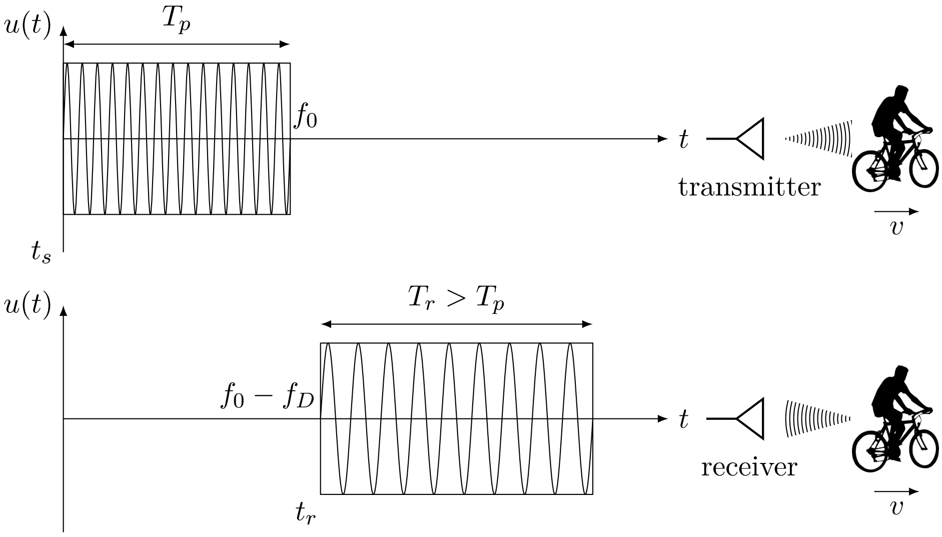


Figure 2.5: Transmitted and received signals in a pulse-Doppler radar [Bü08].

wave transmitter on and off, short pulses of frequency f_0 and time period T_p are sent out for every T_r time interval. The sent impulse is then reflected back after hitting the object. With t_s denoting the starting time of transmission and t_r as the time of reception of the reflected wave, the range R to the object can be calculated similar to 2.1 as

$$R = \frac{(t_s - t_r) \cdot c}{2} \tag{2.3}$$

The parameter T_p influences the range resolution of the radar. Two objects at different distances can be detected as separate objects only if the reflected signals do not overlap.

The interval parameter T_f decides the maximum range of the radar. Following [Fö06], the range resolution ΔR and maximum range of the radar can be given as

$$\begin{aligned} \Delta R &= \frac{c \cdot T_p}{2} \\ R_{\max} &= \frac{c \cdot T_f}{2} \end{aligned} \tag{2.4}$$

Moreover, the impulse reflected back from a moving object would have a shift in frequency f_D, with a difference from the actual transmitted frequency f_0 due to the Doppler effect. The relative velocity v_r of the object can then be calculated by

$$v_r = -\frac{1}{2} \cdot c \cdot \frac{f_D}{f_0} \tag{2.5}$$

Instead of a direct measurement of t_s and t_r, the transmitted and received signals are mixed, transformed and filtered as a matched filter and processed based on the in-phase and quadrature components.

In addition to the range and relative velocity, the azimuthal angular position of the object can also be extracted from the radar. In case of multiple radars, the azimuth of the object can be derived by trilateration method. On the other hand, azimuth can also measured with a single radar, which transmits multiple beams or contains multiple receiver antennas. When the transmitting antenna sends multiple beams, the azimuth can be calculated based on the amplitude difference in their corresponding echo signals. Each beam has different orientation angle with respect to the object and therefore has a difference in the reflected amplitude. Azimuth is then derived by considering the quotient of sum and difference of the amplitudes [Bü08]. In case, the radar contains multiple receiving antennas closely placed to each other, the echo signal received by each antenna would have the same amplitude but different phase. Azimuth in this case is calculated based on the phase difference. The difference in phase of echo signal between antennas depend on the distance between them and the signal wavelength.

2.2.3.2 FMCW-Radar

In contrast to the impulse signal transmitted in a pulse-Doppler-radar, linear frequency modulated continuous wave signal is transmitted in a FMCW-radar. The range R and relative velocity v_r of the object are then derived based on the frequency difference between the transmitted and received signals. The advantage of FMCW-radar is the continuous availability of measurements, compared to the pulsed signals sent in certain time intervals. Figure 2.6 depicts a linear frequency modulated signal with a single ramp. The solid red line represents the transmitted signal. The dotted black line represents the frequency of the received signal reflected back from a stationary object and the dashed red line illustrates the received signal reflected back from a moving object. T_R denotes the duration of the ramp and f_{hub} the frequency hub. The slope of the ramp signal is then $s_R = f_{hub}/T_R$. The reflected signal from a stationary object is received after a time τ since the transmitted signal and has a frequency shift Δf_s. In case of moving object, there is an additional shift in frequency f_D, due to the Doppler effect. The total shift in frequency is then given by

$$\Delta f = \Delta f_s - f_D = \frac{2}{c} \left(s_R \cdot R + f_0 \cdot v_r \right) \tag{2.6}$$

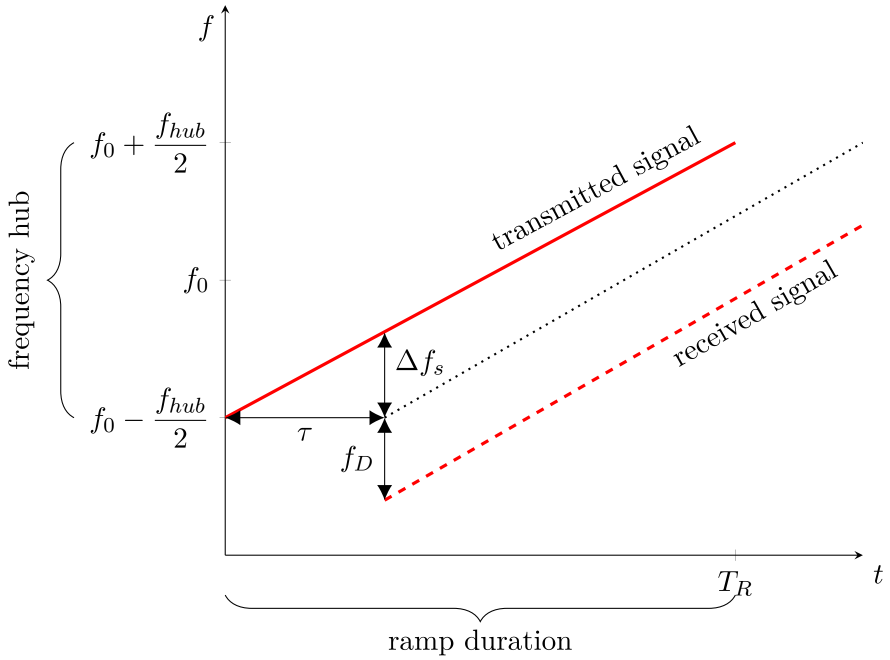


FIGURE 2.6: Transmitted and received signals in a linear FMCW radar [Bü08].

where the frequency shift is dependent both on the R and v_r.

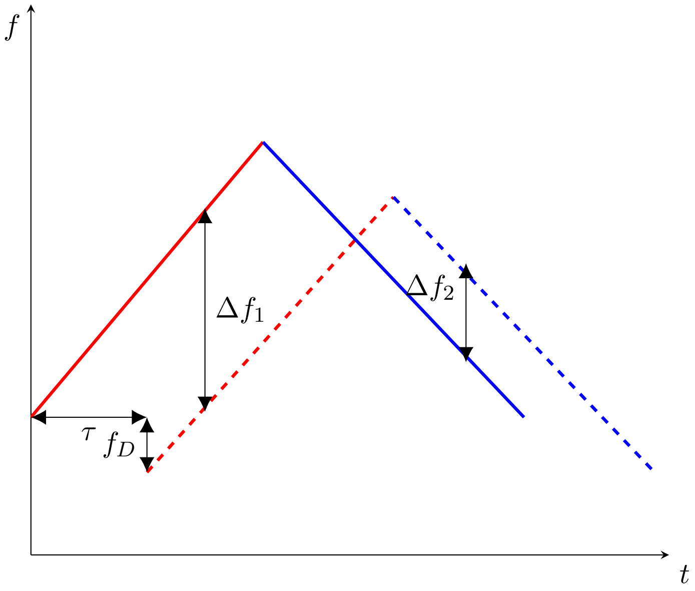
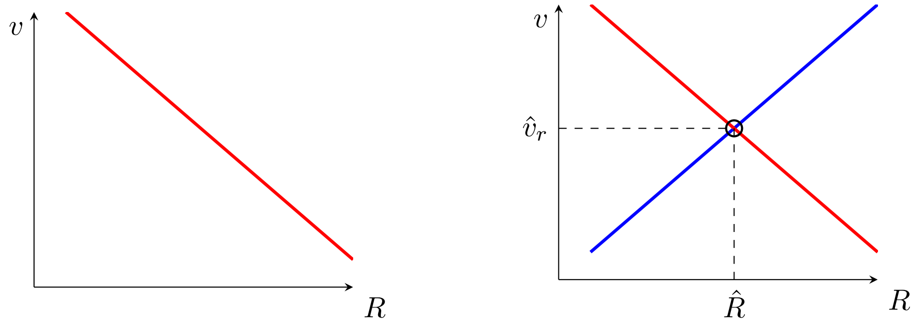


FIGURE 2.7: Two frequency ramps with different slopes

In a so called R-v diagram as in Figure 2.8a, a single ramp is not sufficient to estimate both range and relative velocity simultaneously from the frequency shift, as many combinations of range and relative velocity can be calculated for the same frequency shift. Therefore, an additional frequency ramp is required as illustrated in Figure 2.8b. Both the ramp signals with slopes s_{R_1} and s_{R_2} then would have corresponding echo signals with a frequency shift of Δf_1 and Δf_2 respectively as [Bü08]

$$\begin{aligned}\Delta f_1 &= \frac{2}{c}\left(s_{R_1} \cdot R + f_0 \cdot v_r\right) \\ \Delta f_2 &= \frac{2}{c}\left(s_{R_2} \cdot R + f_0 \cdot v_r\right)\end{aligned} \tag{2.7}$$

Solving the above equations, the range and relative velocity values are extracted from the intersection point of the two slopes in the R-v diagram. However with the above solution, when there are for example two objects in the sensor FoV, four intersection points would be possible in the R-v diagram. In that case, which values of range and relative velocity actually belongs to the objects is not direct anymore. In order to overcome this problem, multiple ramps with different slopes are used. For example, a linear frequency modulated signal with four ramps would result in distinct intersection points and the object individual measurements can be extracted from the intersections of all the ramps as illustrated in Figure 2.9. The range and relative velocity resolution of FMCW-radar depends on how well two frequency peaks in FFT can be separated.



(A) Extraction of R and v not possible with single ramp.
(B) Extraction of R and v with two ramps of different slopes.

FIGURE 2.8: R-v diagram for a single object case

Many companies like Delphi and Continental are offering multimode radars, where the radar can operate in both short and long range mode simultaneously. This is achieved by the concept of electronic beam scanning. This enables the usage of a single radar for many applications like ACC and FCW. It can detect vehicles at long range as well as vehicles and even pedestrians at short range simultaneously Also, the concept of pulse-compression-radar overcomes the disadvantages of classical pulse-Doppler and FMCW radars. Combining both the concepts, chirps with a certain time interval are transmitted in a pulse-compression-radar. Compared to the classical pulse-Doppler-radar, the pulse-compression-radar has a better overall signal-to-noise ratio (SNR) due to large duty cycle. Compared to the FMCW-radar it is easier in a pulse-compression-radar to separate the range and relative velocity information. In general, unlike camera or lidar sensors, radar sensors are robust against weather effects like rain, snow and fog.

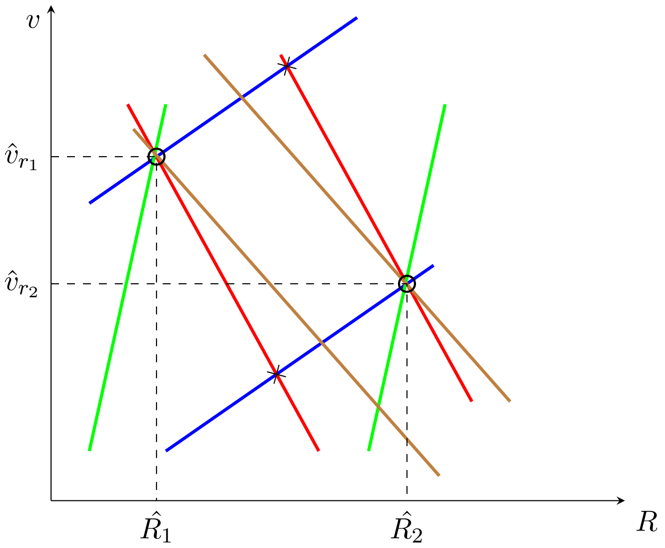


FIGURE 2.9: *R*-*v* diagram for multiple object detection with a 4-Chirp linear FMCW radar.

2.2.4 Ultrasonic sensor

Ultrasonic sensor measures distance to the object based almost on the same principle as radar. However, instead of electromagnetic waves, sound waves are transmitted and received. This means a medium, in this case air, is required for transporting the waves. Again, distance is measured by recording the time elapsed between the transmission and reception of the sound wave after being reflected back by the obstacle or object and considering the speed of sound as 344 m/s. Ultrasonic sensors mainly find their application in park assistance systems. Although ultrasonic sensors are cheaper in price, they have a very limited range. Moreover, the measurement quality is highly affected by temperature and weather conditions. Also, some objects cannot be detected as they could absorb most of the sound energy or their if they are small in size to reflect back enough energy.

2.3 State-of-art perception scheme

As stated in the previous sections, environment perception serves as the basis for ADAS and automated driving applications. For realisation of any vehicle safety application, information about the vehicle environment is essential. Objects are detected, followed and classified in the environment perception layer of the system architecture. Vehicle environment perception in general involves multiple steps. Figure 2.10 illustrates a typical sensor based vehicle environment perception.

The basis for further stages of processing is the sensor data. Raw detections in case of radar and lidar, and raw images in case of camera are the output of the respective sensors. These raw sensor signals are initially preprocessed, for example multiple reflections of a radar believed to be from the same object are filtered out or measurements can be transformed to another coordinate system. For distance based sensors, depending on the spatial closeness and characteristics, individual detections are grouped together as a cluster. This process is known as data clustering. Each object has it's own features and it is reflected in their

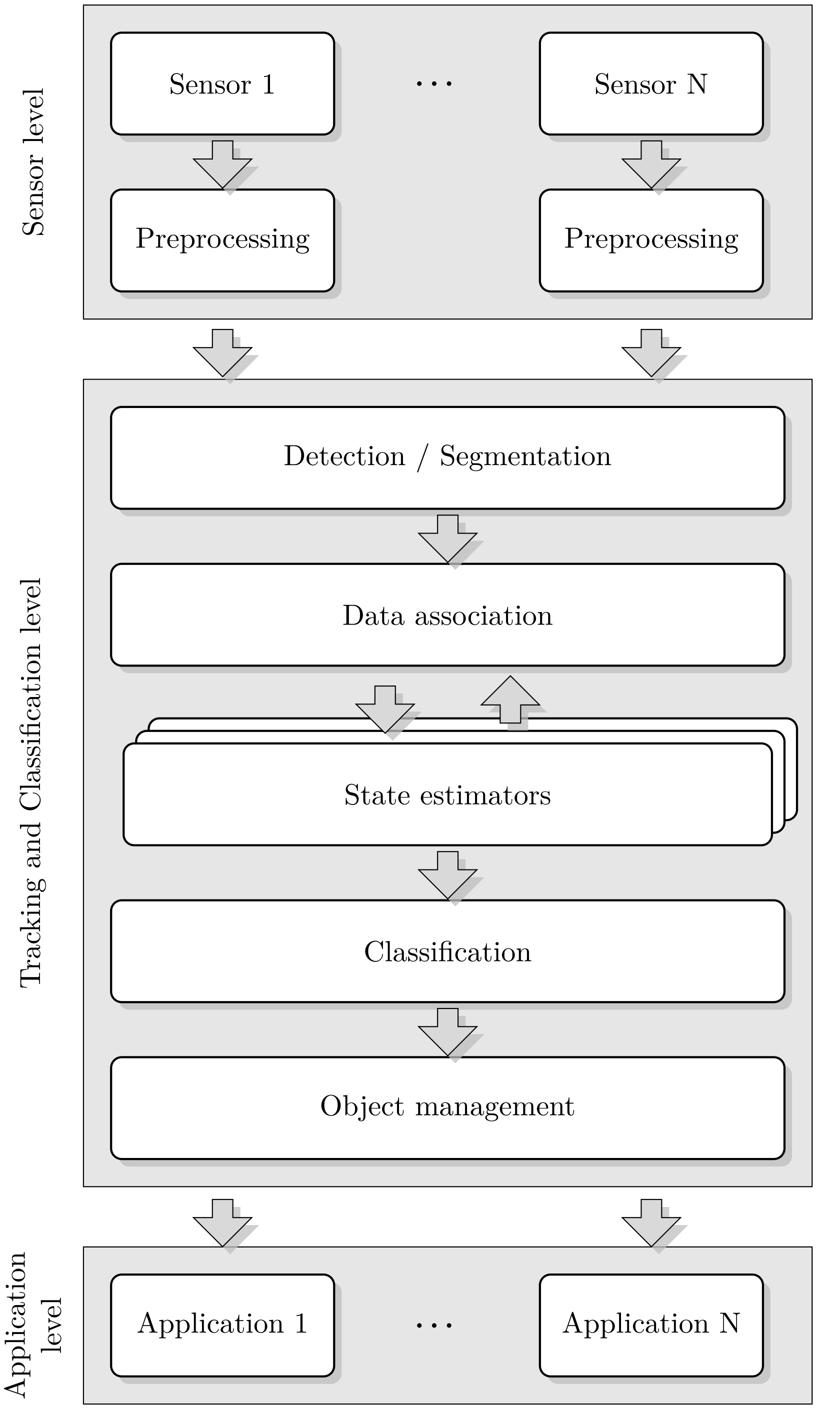


FIGURE 2.10: One of the typical vehicle environment perception schemes

detections cluster. A car has higher width and length than a pedestrian, and a pedestrian can show higher Doppler variance compared to bicyclist due to swinging arms. These features are extracted from the clusters based on a process called feature extraction. Objects can then be classified based on the extracted features from the detection clusters.

Apart from detection, each object needs to be tracked continuously so that application can use the object information. This is termed in general as object tracking. The state of each object like it's position and velocity are also to be estimated, which is required for further analysis and control functions. The object tracking step itself involves many intermediate steps. Object hypotheses and motion models serve as the basis for estimating the object state with the help of estimation algorithms. Sensor detections with similar characteristic for many time steps infers a strong evidence about an object existence and the object is initialised. After the object initialisation, the sensor detections need to be associated with the respective objects for every time step. This process is called data association and is one of the difficult and computationally demanding steps. Because, a sensor delivers typically hundreds of detections every time step and the source of the detection is not straight forward. Moreover, a detection maybe from the actual object or even a clutter measurement.

An optimal data association algorithm should associate the detections to the actual object that caused them. Mis-association or error in the data association step would affect the whole object tracking procedure, giving false information to the application layer. In addition to object appearance and association, confirmation of the object presence and object disappearance situations should also be considered. The object management step addresses this problem, where the object is confirmed or deleted based on evidence of existence. The required degree of object existence evidence depends on the false alarm rate acceptable for the application. For example, a safety critical application like emergency braking system (EBS) should have a lesser false alarm rate than a comfort application, where false alarm can be tolerated compared EBS [Mun11].

2.4 Related work

Radar and thermopile sensors' data are fused in [Lin06] for detecting pedestrians in real-time from a moving vehicle. Kalman filter and weighting technique are used for the fusion of object states and the object category is fused separately using DST. In [FC08] a generic multi-sensor fusion method for tracking pedestrians is presented. A standard Kalman filter with NN data association is used for tracking and the track management is based on Sittler's score [Sit64]. Moreover the detection and recognition confidences are combined using evidential fusion.They evaluate their approach only based on synthetic sensor data, however shows the advantage of considering the reliability of the information source.

Methods for improving the detection and tracking of moving objects, by including the class information at detection and tracking levels are presented in [Gar14]. Two different approaches are presented: in the first approach, the object class information from radar, lidar and mono-camera tracks are used for fusion at tracking level and in the second approach the class information from detections are used for fusion on the detection level. Pedestrian, bicyclist, car and truck object classes are considered and multi-sensor class information is derived based on evidential combination. However the existence information of the objects useful for adaptation in different application scenarios is not addressed.

A generic sensor fusion framework based on joint integrated probabilistic data association (JIPDA) filter is presented in [Mun11], where the existence of an object is also estimated simultaneously along with the object state. The JIPDA method is extended in [Ott13], for pedestrian tracking solutions in truck ADAS applications. Pedestrians are detected and tracked from a moving truck based on radar and mono-camera sensors. Data association is based on several heuristics and is solved by enumerating the association hypotheses using graph-based hypotheses tree. The posterior of the object state is however approximated by

2003, R. Mahler	Probability Hypothesis Density (PHD) Filter	+ Easy to implement - Error prone cardinality estimation
2007, R. Mahler	Cardinalised PHD (CPHD) Filter	+ Improved cardinality estimation - Spooky effect
2009, B.N. Vo and B.T. Vo	Cardinality Balanced Multi-Bernoulli (CB-MeMBer) Filter	+ Improved state estimation - Missing track labeling
2013, B.N. Vo and B.T. Vo	δ-Generalised Labeled Multi-Bernoulli (δ-GLMB)Filter	+ Exact closed form solution - High computational effort
2014, S. Reuter	Labeled Multi-Bernoulli (LMB) Filter	+ Simplified δ-GLMB - Missing object class information

FIGURE 2.11: Development of RFS based tracking methods

a unimodal Gaussian distribution. Moreover, the object class information is not considered in the tracking framework.

Random Finite Set (RFS) based object tracking methods are being further developed in order to overcome the disadvantages of the conventional tracking methods and improve state estimation. A Gaussian mixture probability hypothesis density (GM-PHD) filter for tracking extended targets is presented in [GLO10]. The results of tracking pedestrians with GM-PHD filter and laser scanner are presented. However, the results are provided as a simple proof of concept and is not evaluated for further complex test scenarios. Meissner in [Mei15] develops a method for tracking road users at intersections with classifying multiple model probability hypothesis density (CMMPHD) filter. He presents an approach for joint tracking and classification using multiple sensors and uses DST for combining class evidences. Additionally, the class information is also used for selecting the suitable motion model for a particular object. However, the involved application has stationary sensors mounted at intersections rather than a moving vehicle. In [Reu14], Reuter presents the labeled multi-Bernoulli (LMB) filter and compares it's performance to cardinalized probability hypothesis density (CPHD) and cardinality balanced multi-target multi-Bernoulli (CBMeMBer) filters. It is shown with the example of a passenger car automatic cruise control application that the LMB filter performs identical to the JIPDA filter in basic scenarios and at certain situations outperforms it. The object class information and extension of the approach to commercial truck applications for tracking VRUs are however not presented.

Chapter 3

Objectives

As presented in Chapter 1, the increase in vehicle traffic and freight increases the number of accidents caused due to trucks. This clearly motivates the need for highly efficient vehicle safety systems in order to improve road safety. One of the important aspects is to reduce truck accidents with vulnerable road users (VRU), where the survival rate of the VRU is very less when involved in an accident with truck.

A foremost requirement of a system for improving the safety of VRU would be it's ability to detect, localise and follow the VRU precisely, so that the safety application can take decisions based on these informations. Moreover with the advancement of autonomous driving concepts and sensor technologies as presented in Chapter 2, the need of robust methods for sensing, understanding and following objects in the vehicle environment clearly serves as a motivation for this work.

The main objective of this thesis is to develop an approach for reducing the number of commercial truck accidents involving collision with VRUs, especially in urban scenarios. Thereby, the particular aim is to develop robust environment perception solutions for detecting and tracking VRUs in the near vicinity of truck, based on which the safety application can then react to avoid collision.

A typical vehicle perception environment in urban scenario is highly dynamic, where different class of objects like pedestrians and bicyclists appear simultaneously in the sensor field-of-view (FoV). Each class of object has different characteristics and motion attributes. For example, a straight walking pedestrian can be assumed to move with constant velocity whereas a constant turn rate and velocity model would be more suitable for representing the motion of a car. Consequently, it is beneficial to integrate the class knowledge of the object into the tracking algorithm, in order to use a suitable motion model for object state estimation. Also, it is essential in most of the ADAS applications to know the class of the object, in order to adapt the system parameters and to use a suitable intervention strategy. Therefore an additional aim of the work is to classify the objects in the vehicle environment and to fuse the class information with the object tracker and the intervention strategy.

Further on the practical side, the aim is to implement a prototype of the developed approaches and to evaluate their performance and accuracy with the help of a series of real-time test scenarios.

Chapter 4

Technical Preliminaries and Methodology

4.1 Approach and Structure

The high-level goals stated in Chapter 3 can be translated into series of technical and scientific research questions that need to be answered. The technical questions to be answered from the system realisation point of view are:

- What are the available sensor technologies and how can the objects in the near vicinity of truck be sensed and detected?
- What are the technical preliminaries, measurement and communication techniques required for efficient transfer of data between sensors and the computation module? How can the data be processed, stored and analysed?
- How should the system and software be designed considering different levels of object information such as detection, tracking and classification? Which software tools and methods are to be used for implementation of algorithms?

The research questions to be answered from the scientific point of view, in order to develop a feasible solution are:

- What are the available mathematical methods and approaches to address the problem of multi-object tracking and classification?
- How well do the existing approaches satisfy the requirements of tracking and classifying VRUs in the near vicinity of truck?
- How to develop a robust approach for tracking and classifying objects in highly dynamic urban scenarios, considering random object appearance and disappearance, maneuvering objects, varying observability and uncertain sensor measurements. How to qualitatively and quantitatively evaluate the developed approaches, in order to improve and mature them for implementation in series development of truck ADAS and autonomous driving ECUs?

The main contribution of this work is the development of solutions to the above questions, by formulating systematic methods and novel approaches built on existing theoretical concepts, that can be implemented for VRU protection applications in trucks. The sensor used for object detection is an high-frequency radar with a frequency range of 77-81 GHz. Random Finite Set (RFS) based object tracking methods are used as a base for the tracker

Chapter 1: Introduction

Chapter 2: State-of-art

- Scope and development of ADAS & automated driving technologies
- Description of widely used sensors, their measurement principles and state-of-art vehicle environment perception scheme

Chapter 3: Objectives

Chapter 4: Technical Preliminaries and Methodology

- Approach and structure
- Description of test vehicle setup, coordinate systems, radar mounting & calibration, data communication and measurement techniques
- Proposal of a radar based environment perception scheme for application in truck safety systems

Chapter 5: Random Finite Sets Based Object Tracking

- Outline of RFS based unified multi-object tracking concepts
- Description of the LMB filter and it's multiple model version used in this work
- Presentation of an approach for joint tracking and classification of VRUs with MMLMB filter

Chapter 6: Radar based Feature Extraction and Classification

- Methods for radar detections clustering, geometric shape fitting and feature extraction
- Development of a machine learning based classifier for radar measurement based object classification

Chapter 7: Application & Realisation of Environment Perception for Truck ADAS

- Analysis of radar measurement characteristics
- Choice of suitable motion models, sensor model and other model assumptions required for the realisation of multi-object tracking

Chapter 8: Evaluation

- Accuracy and performance evaluation of the developed tracking and classification approaches with multiple real-time test scenarios

Chapter 9: Conclusion

FIGURE 4.1: Structure of the dissertation

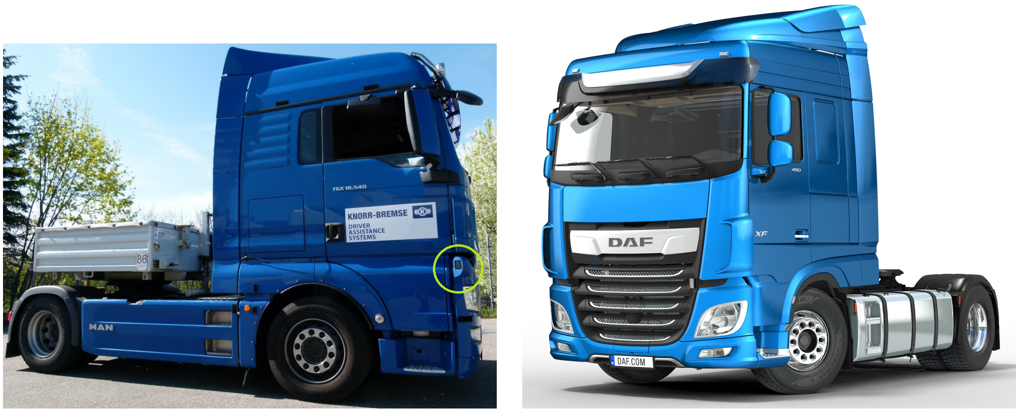
development. Recently developed multiple model labeled multi-Bernoulli filter (MMLMB) [RSD15] is used for considering different motion models in object tracking. Another aspect of this work is the development of a machine learning based classifier for classifying different objects in the environment of the truck, based on high-resolution radar detections. The constructed classifier is trained and tested with data from multiple measurement sequences. Moreover, an approach to integrate the derived class information within the MMLMB tracking algorithm is formulated, in order to adapt the tracking parameters based on the object class for improving estimation robustness. For this, the solution proposed in [MRSD14] for multiple model probability hypothesis density (MMPHD) filter is adapted to be used with the MMLMB filter. Further, an own implementation and parametrisation of the conventional converted measurements Kalman filter (CMKF) with global nearest neighbour (GNN) data association method is used to compare the performance of the developed approach.

The structure of this dissertation is illustrated in Figure 4.1. After a brief introduction in Chapter 1, the state-of-art ADAS and autonomous driving technologies along with the sensor measurement principles were outlined in Chapter 2. Subsequently, the objectives of the dissertation were stated in Chapter 3.

The methodological approaches for achieving the stated objectives, system setup and technical preliminaries required for communication, data acquisition and concept realisation are described in this Chapter. The basics of finite set statistics (FISST), multi-object Bayes filters and RFS based object tracking methods as alternative to the conventional multi-object tracking methods based on bank of single-object Bayes filters, are presented in first part of Chapter 5. The second part of Chapter 5 presents the LMB filter proposed in [RVVD14] based on the labeled RFS, along with it's multiple model version. Chapter 5 concludes with the presentation of an approach to integrate the object class information into object tracking, based on Dempster-Shafer theory (DST) of evidence. Chapter 6 explains the concepts of segmentation, feature extraction and machine learning based object classification. The choice of motion models, sensor model and filter parameters required for prototypical realisation of the developed methods and approaches as a truck safety application is presented in Chapter 7. Chapter 8 begins with the definition of metrics for evaluation of the developed object tracking and classification methods and concludes by presenting the evaluation results according to different test scenarios. Chapter 9 concludes the dissertation, providing a summary of the work and directions for future research.

4.2 Test vehicles and system setup

The test vehicles used for experiments in this work is shown in Figure 4.2. The vehicle in Figure 4.2a is MAN TGX series tractor unit. The vehicle has a 4x2 axle configuration with a total length 5.88 m and width 2.44 m. The unladen weight of the tractor unit is 7400 kg. The vehicle is equipped with a 6-cylinder 520 PS engine. Another test vehicle is a DAF XF series tractor unit, shown in Figure 4.2b. It also has a 4x2 axle configuration and total mass of 7500 kg. The DAF tractor unit has a total length of 5.96 m and width 2.55 m. It is pulled by a 6-cylinder 483 PS engine. Both vehicles have 24V power supply. The radar sensor is mounted on the right side corner of the vehicles with it's axis tilted ≈ 30 °towards the rear of the vehicle.



(A) Test vehicle - MAN TGX (B) Test vehicle - DAF XF

FIGURE 4.2: Both the test vehicles are installed with sensors and measurement equipments. The radar sensor mounting is as represented in Figure (A), enclosed by a green circle.

4.2.1 Coordinate systems

The vehicle and sensor coordinate systems are defined according to the DIN 70,000. The origin of the Cartesian vehicle coordinate system is at the center of rear axle of the truck represented by x_V, y_V and z_V. The coordinate system has it's x-axis pointing in the longitudinal direction towards the vehicle front, y-axis pointing in the lateral left direction and z-axis in the upward direction.

The origin of the radar sensor cartesian coordinate system is at the right corner of the vehicle and is represented by x_S and y_S. Figure 4.3 depicts the vehicle and sensor coordinate systems. As the radar has no elevation measurement, only the two-dimensional x and y axes are considered. Radar sensor delivers the azimuth angle, range and Doppler velocity of the target in polar coordinates. Therefore the measurements are transformed into sensor Cartesian coordinates and then to the vehicle coordinate system with the transformation matrices defined in equation 4.1. A point $\mathbf{p}_{sensor} = (x_{sensor}, y_{sensor})^\top$ in the sensor coordinate system is transformed as a point $\mathbf{p}_{veh} = (x_{veh}, y_{veh})^\top$ into vehicle coordinate system by

$$\mathbf{p}_{veh} = \mathbf{R}_{S2V} \cdot \mathbf{p}_{sensor} + \mathbf{t}_{S2V} \tag{4.1}$$

where the rotation matrix $\mathbf{R}_{S2V}$ is calculated from the yaw mounting angle ψ_{sensor} of the sensor, which is the rotation around z-axis and $\mathbf{t}_{S2V}$ is translational x and y mounting positions of the sensor with respect to the vehicle coordinate system.

$$\mathbf{R}_{S2V} = \begin{bmatrix} \cos(\psi_{sensor}) & -\sin(\psi_{sensor}) \\ \sin(\psi_{sensor}) & \cos(\psi_{sensor}) \end{bmatrix} \tag{4.2}$$

4.2.2 On-board sensors

Apart from the high-resolution short range radar sensor, the vehicles are also equipped with a front looking camera sensor and long range radar sensor. Only the high-resolution radar sensor is primarily use for this work. Additionally, the vehicles are also equipped with on-board wheel speed and yaw-rate sensors. Wheel speed sensors in MAN TGX and in DAF

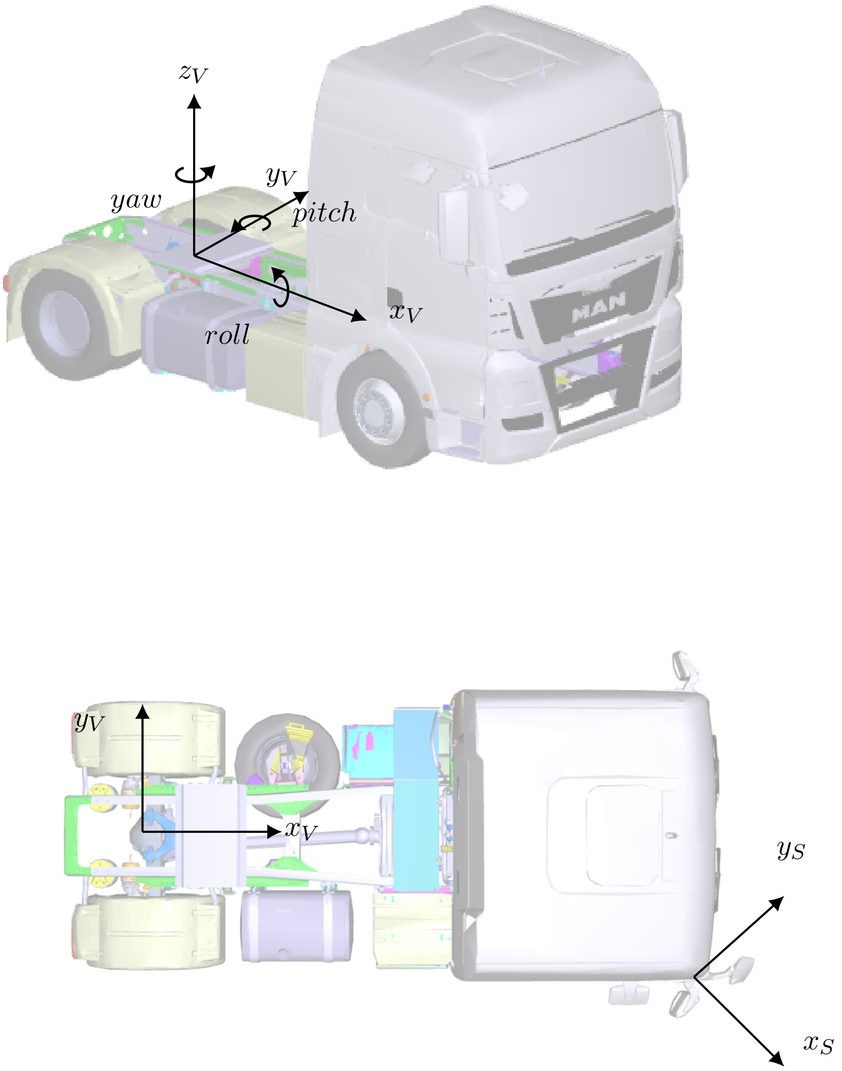
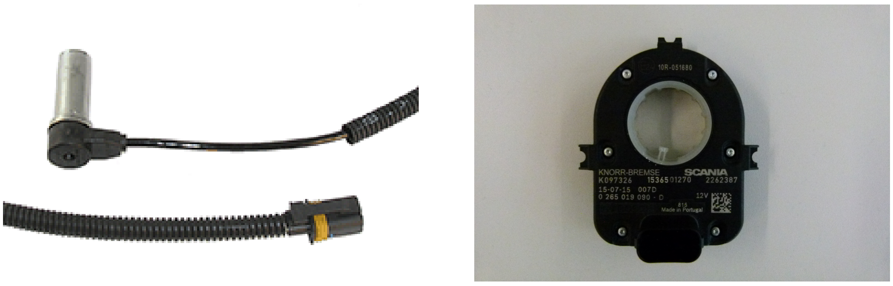


FIGURE 4.3: Figure at the top depicts the vehicle Cartesian coordinate system and Figure at the bottom illustrates the radar sensor coordinate system with respect to the vehicle coordinate system.

XF 460 are from the company Knorr-Bremse. The wheel speed sensors are used as part of series ABS and ESP systems. A wheel speed sensor installed in each of the four wheels of the truck and measures the wheel rotations per minute based on induction principle. Figure 4.4a shows one of the wheel speed sensors used in the test vehicles. The host vehicle speed is estimated based on the wheel speed sensor measurements. Another important sensor for measuring the angular velocity around the vertical z_V axis of the vehicle is the yaw-rate sensor. It is actually part of a sensor cluster required for the ESP system, which can also measure lateral and longitudinal acceleration of the truck. The used sensor is DRS MM 3.R8k from Bosch. In a micromechanical type gyroscopic yaw-rate sensor, yaw-rate measurement is based on the Coriolis-principle. The Coriolis acceleration is measured by micro-mechanical capacitive acceleration sensor placed on an oscillating element. The generated acceleration is proportional to the product of vehicle yaw-rate and the oscillatory



(A) Wheel-speed sensor (B) Steering angle sensor

FIGURE 4.4: Used wheel-speed and steering angle sensors.

velocity. Additionally the steering wheel angle is also measured and is available as input signal to the model functions. MAN TGX test vehicle is equipped with a steering angle sensor from the company Knorr-Bremse and DAF XF 460 is equipped with a Takata steering angle sensor. All the on-board sensor signals are available through the J1939 vehicle CAN bus.

4.3 Radar Sensor

4.3.1 Properties

A prototype radar sensor with a carrier frequency of the range 77-81 GHz is used in this work. The range and Doppler of a target are measured by FMCW principle. Theoretically the radar has an opening angle ±75°. The sensor specifications are summarized in Table 4.1. The radar sensor delivers measurements in polar coordinates with azimuth, range and relative velocity values of objects. The FoV of the radar sensor as mounted on the right corner of the vehicle is illustrated in Figure 4.5.

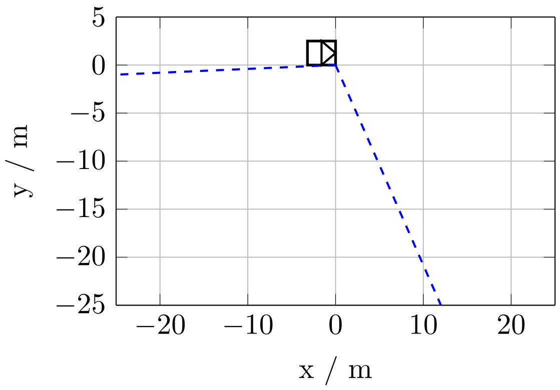


FIGURE 4.5: Illustration of field of view of the used radar sensor.

Property	Specification
Radar measurement principle	FMCW
Frequency / GHz	77-81
Maximum detection range / m	30
Range accuracy / m	0.028
Range resolution / m	0.1
Horizontal opening angle / deg	150
Angular accuracy / deg	2.61
Angular resolution / deg	1
Relative velocity accuracy / m/s	0.1
Relative velocity resolution / m/s	0.3

Table 4.1: Specifications of the used radar sensor.

4.3.2 Calibration

The orientation angle of the sensor with respect to the vehicle longitudinal axis is calibrated by series of tests. A corner reflector of know radar cross section (RCS) is placed at reference x and y positions in the FoV as shown in Figure 4.6, and the Cartesian position of the measured target point is calculated. The test is repeated with different reference positions and the sensor orientation angle is calibrated until the root mean-squared error (RMSE) in position is minimized.

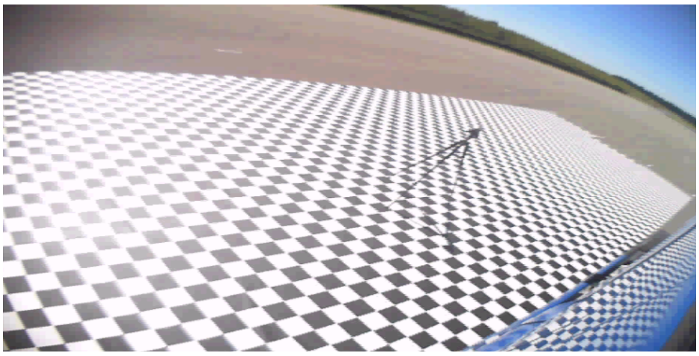


Figure 4.6: Setup with corner reflector for sensor position calibration and estimation of measurement characteristics.

4.4 Software framework

Algorithms for vehicle environment perception in this work are implemented and tested according to model-based software development methods. Matlab/Simulink, it's toolboxes along with some hand coded C-functions are used for implementation. Matlab and Simulink Coder enables the code generation from the models. Prototypical VRU protection system is realised in real-time on the test vehicles with help of rapid prototyping (RPT) method and Micro Autobox (MABX) from the company dSpace. Rapid prototyping is a development method enabling quick prototypical implementation, testing and iteration of real-time functions and control strategies. With RPT, new functions can be integrated to the system in real-time even in the early stages of development, with the help of prototype hardware. MABX is one of the prototype hardwares used for the deployment and testing of software code in real-time. The initial step of RPT method is to build software functions by means of graphical models. Matlab/Simulink is used for this purpose in this work. In the context of model-based development methods, the developed functions as models can be iteratively tested as Model-in-the-loop (MIL), Software-in-the-loop (SIL) and Hardware-in-the-loop (HIL) solutions. MIL and SIL testing do not need a real-time hardware for implementation, whereas HIL is more complex, which also requires the actual ECU or actuators within the testing loop. As the actual ECU is not available during the early development phases of the system, RPT facilitates the testing of software functions in real-time at early development stage, however with a prototype hardware. This helps in quick optimisation and improvement of functions. The RPT structure used in this work is illustrated in Figure 4.7

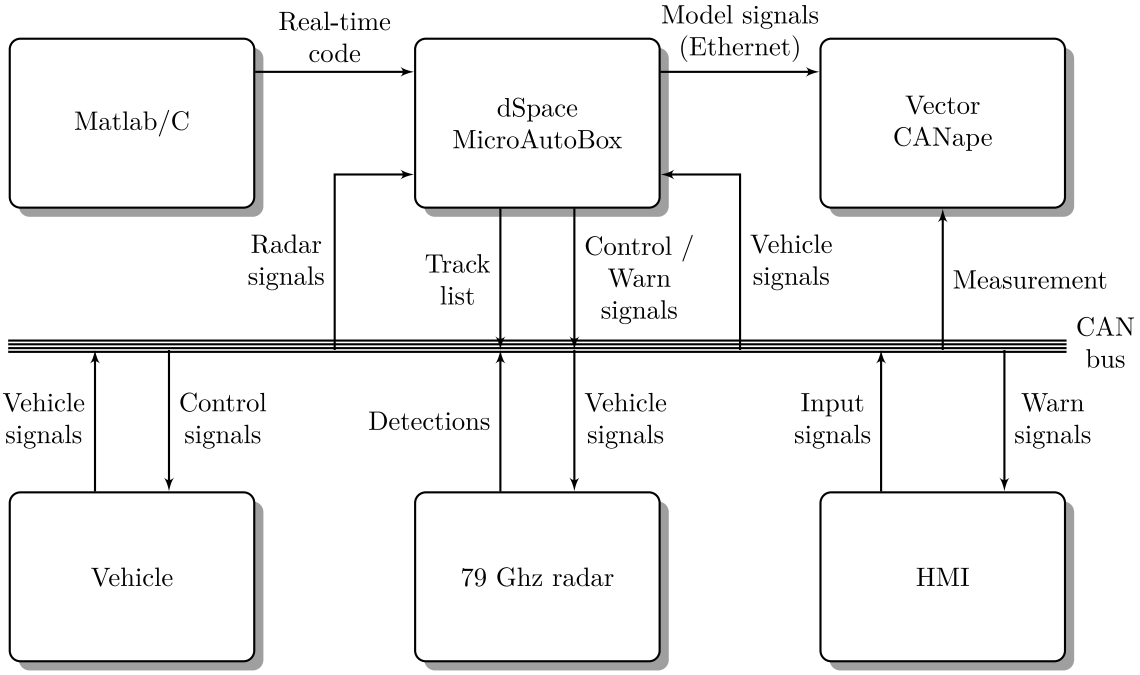


FIGURE 4.7: Rapid prototyping framework with corresponding signal flow.

Apart from implementation, various test scenarios are also defined in this work for testing the functionality of various software modules. Test data are logged with help of CANape measurement environment from the company Vector Informatik, to facilitate offline simulation and analysis of the developed algorithms. For visualisation and analysis, an own implementation of GUI based on Matlab, ControlDesk software from dSpace as well as Python based analysis tool "MASS" from Knorr-Bremse are used.

4.5 CAN bus structure

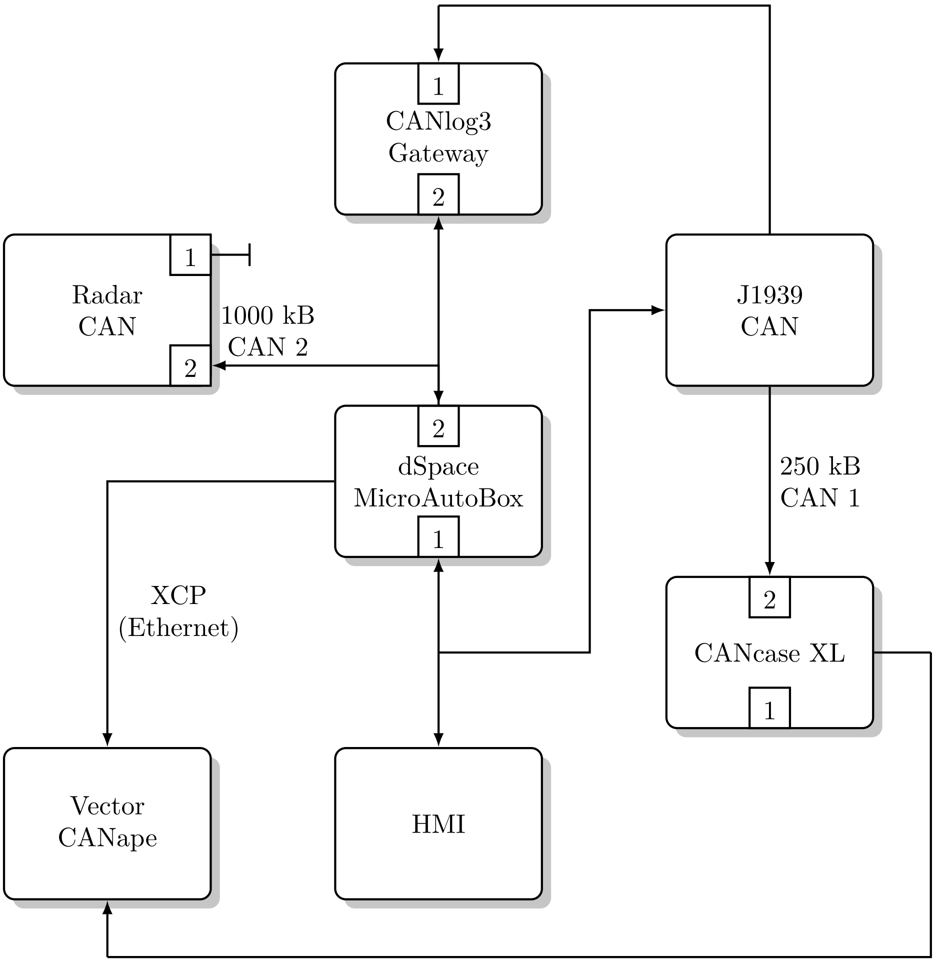


Figure 4.8: CAN bus structure

The communication and data transfer between sensor modules, processor and measurement computer is through a bunch of CAN buses. The CAN bus structure of the test vehicles are illustrated in Figure 4.8. J1939 is the transmission protocol defined by SAE, used for communication of ECUs in commercial vehicles. The vehicle signals like speed, steering wheel angle and yaw-rate are made available to the MABX and CANape through J1939 vehicle CAN 1 bus at a baud rate of 250 kB. The radar sends detections through the private CAN 2 at a baud rate of 1000 kB according to the radar manufacturer's protocol. The radar additionally requires some of the vehicle CAN 1 signals for it's functionality. But the radar CAN 2 cannot be directly connected to the vehicle CAN 1, as they both have different network protocols and baud rates. Moreover, J1939 protocol typically uses 29 bit extended CAN-identifier whereas the network protocol of the radar uses 11-bit CAN-identifier. Therefore a gateway is required for communication between the vehicle CAN 1 and radar CAN 2. The CAN-gateway is realised with a CANlog 3 datalogger from the company G.i.N. The datalogger can be programmed with Log Task Language (LTL) using configuration software GiNConf, provided by the company G.i.N.

4.6 Measurement techniques

The vehicle and radar signals from CAN 1 and CAN 2 are connected to CANcase XL and are logged with CANape software running on a measurement PC. Measurement PC is a Panasonic Toughbook. Additionally, internal signals and parameters of the model running on the MABX can be directly sent to the CANape measurement environment through the XCP protocol. XCP is the Universal Measurement and Calibration Protocol defined by the Association for Standardization of Automation and Measuring System (ASAM) in 2003. The RTI XCP on Ethernet blockset provided by dSpace enables the model data from MABX to be sent directly to the measurement system through ethernet connection. The RTI XCP blocksets are used to configure and setup XCP on Ethernet services and also to capture model data. When the blocksets are placed and setup in the Matlab/Simulink model, an A2L file is generated during the model build process. The measurement system then reads this A2L file generated for the MABX and configures the XCP on Ethernet Real-Time Service to acquire data from the memory of MABX. This on the other hand also helps in easier model debugging and offline resimulation. The communication structure of the Matlab/Simulink model with the CANape measurement environment through XCP on Ethernet is depicted in Figure 4.9. The measurement PC running CANape measurement software is additionally connected with a LifeCam Studio from Microsoft. The webcam has a 1080p-HD sensor, mounted with wide-angle lens having a 75° diagonal FoV. The webcam is capable of recording video sequences upto 30 frames per second (FPS). The webcam is primarily used for video recording of the test scenarios, which are then used for detailed analysis of the developed algorithms based on the actual scene. Moreover the webcam is also calibrated with a checker board for ground truth projection. The ground truth projected images are then used for detecting relevant objects in the scene with openCV libraries, thereby serving as a source of reference measurement.

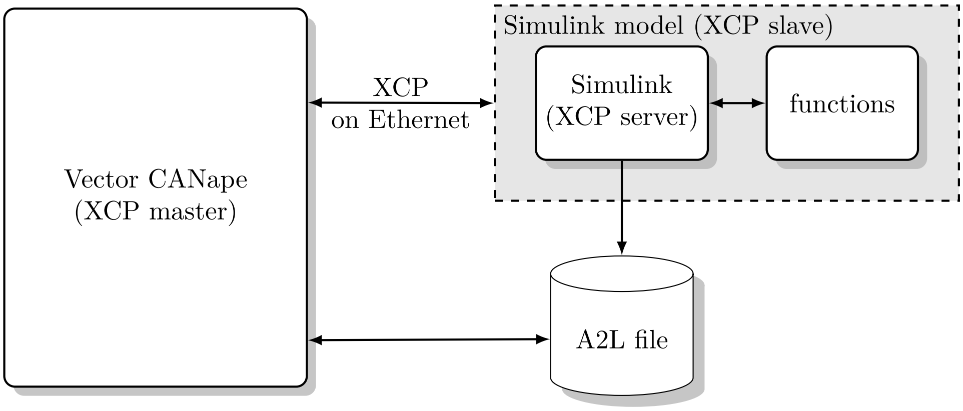


FIGURE 4.9: XCP communication structure (source: Vector Informatik GmbH product website).

4.7 System and Software Schema

The objectives stated in Chapter 3 are transformed into system and software solutions. They are realised with the help of different modules, each satisfying different functional requirements. Radar is the primarily used sensor used for object detection. Range, azimuth and Doppler measurement of objects in polar coordinates is the first level of input. Apart from the radar detections, vehicle speed, yaw-rate, steering angle are also some of the additionally required input signals, in order to estimate the host vehicle's state. Figure 4.10 depicts the software modules, scheme and data flow used in this work. The complete software scheme is divided into following modules:

> *Radar detection:* The interface between radar and the model running in MABX is realised by using RTI CAN MultiMessage blockset from dSpace, which can handle complex CAN setups and high number of messages. The blocksets are placed within the Matlab/Simulink models and are configured with the corresponding CAN database configurations. Both receive and transmit signals from the model can be configured within the RTI CAN MultiMessage blockset. The raw detections delivered by the radar are thereby directly integrated into the software running on prototype hardware and made available for other modules.
>
> *Measurement pre-processing:* Apart from coordinate transformations, additional pre-processing of raw detections is also required. If the reflected electromagnetic wave from the object is again reflected back from the host vehicle surface, it could act in a same way as a signal transmitted by radar. When this wave hits the object again and reflects back with enough power, it would be detected as an additional object. This multiple reflection from a same object would create ghost objects and therefore need to be pre-processed.
>
> *Detections clustering:* The used radar is capable of detecting multiple points of an extended object. The aim of clustering module is to segment detections believed to be from the same object as distinct group. DBSCAN algorithm proposed in [EKSX96] is used in this work for detections clustering.
>
> *Shape fitting and feature extraction:* Each group of detections have certain features, depending on the object they originated from. Various features are then extracted from the clustered groups. The features that are looked for in each group are defined and analysed in Chapter 6.
>
> *Measurement based classification:* A machine learning based method to classify objects, based on the extracted features of the detection clusters is implemented in this module.
>
> *Object reference point:* Although the detections from an extended object are clustered as a group, the object is tracked only with reference to a single point. This point representing the object should be stable and invariant to the object direction of motion with respect to the sensor. Different choices of object reference point are made, depending on the spatial extension of the detections cluster, result of shape fitting module as well as the position of the object.
>
> *Object state estimation and tracking:* Information about object position and velocity are essential for situation analysis and vehicle intervention strategies. These entities are in general called object states and the method to calculate them is called state estimation. Object tracking is the process to properly predict and correct their movement and position using motion and measurement models, with assumed object hypotheses [TBF05]. As motion models are required for predicting the object states, measurement models are required for correcting the predicted object states based on actual sensor measurements. Therefore for tracking an object, it is required to associate the

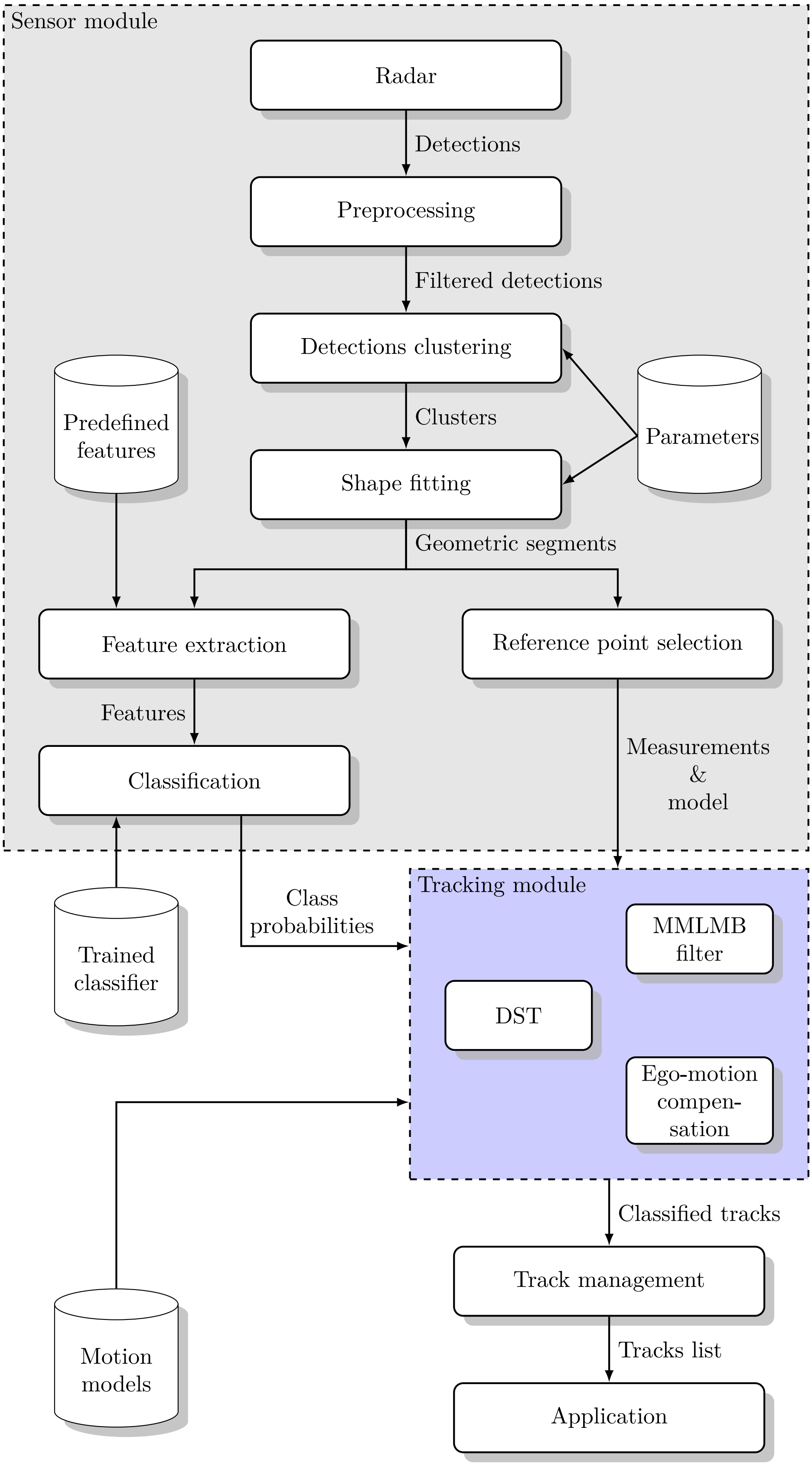


FIGURE 4.10: Proposed system scheme for tracking and classification of objects using radar sensor.

actual radar measurement with that object at each time step. This process is called data association. All the above steps are related to each other and are implemented within this module. For object tracking, a Gaussian mixture MMLMB filter based on RFS approach is implemented. Due to the implicit data association step of MMLMB filter, a separate data association sub-module is not required. Disadvantages of the state-of-art tracking methods, motivation, principles and development of RFS based tracking concepts, and implementation details of this module are further detailed in Chapter 5.

Ego-motion compensation: Apart from the motion of the objects in the environment, the ego-vehicle itself is in motion. This movement of the ego-vehicle should be considered in prediciting the object states and is realised within this module. Vehicle speed, steering angle and yaw-rate are available as input signals to the model again with the help of RTI CAN MultiMessage blockset and J1939 vehicle CAN database. Based on these signals and a kinematic single track model of the truck, the motion of ego-vehicle is predicted.

Integrating object class information: The output of measurement based classification module are confidence values for each of the object classes. These confidence values are merged with the output of the object state estimation, specifically the object velocity, with the help of Demspter-Shafer theory of evidence. Further, the weight for motion models in MMLMB are adapted according to predicted object class.

Track management: Object birth, confirmation and death are handled in track management module. Decision within track management module are based on evidence levels supporting object existence.

Interface to safety application: Output of the above processing steps is a list of established tracks. Each track has it's identity along with it's estimated state and class. These tracks can further be used by ADAS applications, based on situation and application dependent track selection criteria. The list of tracks are made available on the CAN bus through the transmit configurations of the integrated dSpace RTI CAN MultiMessage blocksets and can further be used by the application.

Chapter 5

Random Finite Sets based Multi-Object Tracking

A sensor delivers observation of certain properties of an object, based on it's measurement principle and capabilities. In order to be usable, the properties and characteristics of the object need to be continuously estimated over time, based on the uncertain sensor measurements. This process of continuous estimation of object state such as location and velocity, typically of one or more objects based on one or more sensors, together with the estimation of number of objects is called as object or target tracking. Additionally, the estimation process requires assumptions about the motion properties of the object which is governed by uncertain motion models, as well as sensor measurement characteristics which is modeled by corresponding sensor measurement models.

The theoretical foundations of the conventional multi-object tracking concepts are summarised in Appendix A. These methods use multiple instances of a single-object Bayes filter. Which means, the multi-object tracking problem is not exactly considered as a single closed entity, but is based on chain of separate single-object statistics. The independent single-object filters require an explicit association of the measurement data to each object. Moreover the conventional target tracking methods handle the estimation of object states and the estimation of number of actual objects separately. The data association and track management steps are only suboptimal as they are often based on assumptions.

Although multiple solutions for improving multi-object tracking with parallel single-object filters have been developed, increased complexity of these algorithms can lead to poor behaviour traceability [Mah04]. Therefore a systematic unification of object detection, tracking, classification, management and evaluation, based on finite set statistic (FISST) is proposed by Mahler. A generalisation of the single-object Bayesian methods to the multi-object tracking problem is facilitated by modeling the object states and measurements as random finite set (RFS), rather than individual random vectors. The motivation for multi-object RFS based target tracking approaches compared to the classical single-object Bayesian approaches according to [Mah03, Mah07b] are

- An unified approach for the estimation of the object states as well as the number of objects.
- A finite set approach unlike the vector representation can represent all possible occurrences of multi-object states, including the case where no objects are present, in which case the set will be empty.

- An explicit consideration of measurement-to-track association is not required as it is unified within the multi-object Bayesian formalisation.

This chapter is divided into six sections. The sections 5.1-5.3 presents the basics of RFS and various multi-object probability distributions. Also, the concepts of labeled RFSs from [VV13a] are presented. Section 5.4 introduces the multi-object Bayesian recursion with the help of multi-object Markov densities and multi-object Likelihood functions. The computationally tractable approximation of the multi-object Bayes filter, namely the probability hypothesis density (PHD) filter and it's variants are introduced in section 5.5. Section 5.6 describes the labeled multi-Bernoulli filter (LMB) introduced in [RVVD14] and the following section 5.7 presents the multiple model version of the LMB filter proposed in [RSD15] along with it's Gaussian mixture implementation. The last section 5.8 begins with the description of Demspter-Shafer theory (DST) of evidence. It also presents a method on how to further extend the MMLMB filter proposed in [RSD15], in order to enable the integration of class information into the tracking framework and to track VRUs.

5.1 Random Finite Sets

In general, every object state is modeled as a random vector, which is typical in the case of single-object Bayes filter. A set of individual single-object Bayes filters, for example Kalman filter, are used to estimate each of these state vectors. But the notion of random number of objects is missing in this representation, which is also essential for multi-object tracking, given the fact that objects can appear or disappear from time to time. This is made possible by the Random Finite Set representation of the object states. Compared to random vector representation, the advantage of RFS is that the number of objects at any time step can be random, unordered and distinct. The definition of RFS according to [Vo08] is given as

Definition 5.1.1. *random finite set: A random finite set is simply a random variable that take values as (unordered) finite sets, i.e. a finite-set-valued random variable. The essential difference between an RFS and a random vector is that: for an RFS the number of constituent points is random and the points themselves are random, distinct and unordered; whereas for a random vector there is exactly one constituent point which is random.*

In the context of multi-object tracking, all the object together can be represented in RFS form as

$$\mathrm{X} = \{x^{(1)}, x^{(2)}, x^{(3)}, \dots x^{(n)}\} \tag{5.1}$$

where $|\mathrm{X}| = n \geq 0$ gives the number of objects and $x^{(1)} \dots x^{(n)}$ the individual random object state vectors. Similarly, all the measurements observed by a sensor can be written as the RFS

$$\mathrm{Z} = \{z^{(1)}, z^{(2)}, z^{(3)}, \dots z^{(n)}\} \tag{5.2}$$

where $|\mathrm{Z}| = m \geq 0$ gives the number of measurements and $z^{(1)} \dots z^{(m)}$ the random measurement vectors. The term ≥ 0 in both state and measurement sets thus represents the possibility of considering the case where there are no objects and the case of missed detection or clutter respectively. In other words, the multi-object state X can take different instantiations according to number of objects as

$\mathrm{X} = \emptyset$ there are no objects present

$$\mathrm{X} = \left\{x^{(1)}\right\} \qquad \text{one object with state vector } x^{(1)} \text{ is present}$$
$$\mathrm{X} = \left\{x^{(1)}, x^{(2)}\right\} \qquad \text{two objects with state vector } x^{(1)} \neq x^{(2)} \text{ are present}$$
$$\vdots$$

and the measurement set Z can take different instantiations according to number of measurements as

$$\mathrm{Z} = \emptyset \qquad \text{there are no sensor measurements}$$
$$\mathrm{Z} = \left\{z^{(1)}\right\} \qquad \text{single measurement } z^{(1)} \text{ is available}$$
$$\mathrm{Z} = \left\{z^{(1)}, z^{(2)}\right\} \qquad \text{two measurements with vectors } z^{(1)} \neq z^{(2)} \text{ are available}$$
$$\vdots$$

The representation of states and measurements as RFS in the corresponding state and observation space are illustrated in Figure 5.1. This representation of RFS is used through out this work, for describing the RFS based multi-object Bayes filters in the following sections. Further in this work, similar notations and abbreviations as in [VV13a] and [RVVD14] are used. Single-object states are denoted by lower case letters (e.g. x), multi-object states represented as RFSs are denoted by upper case letters (e.g. X) and multi-object probability distribution is denoted by π. Spaces are denoted with blackboard upper case letters, as $\mathbb{X}$ for state space with object state vectors $x \in \mathbb{X}$ and as $\mathbb{Z}$ for measurement space with measurement vectors $z \in \mathbb{Z}$. In case of labeled RFS, which is briefly presented in the following sections, multi-object state is denoted by bold face upper case letter $\mathbf{X}$, state vector augmented with label as $\boldsymbol{x} = (x, l)$. Moreover, as proposed in [VV13a, Mah14], inner product of continuous functions are denoted by

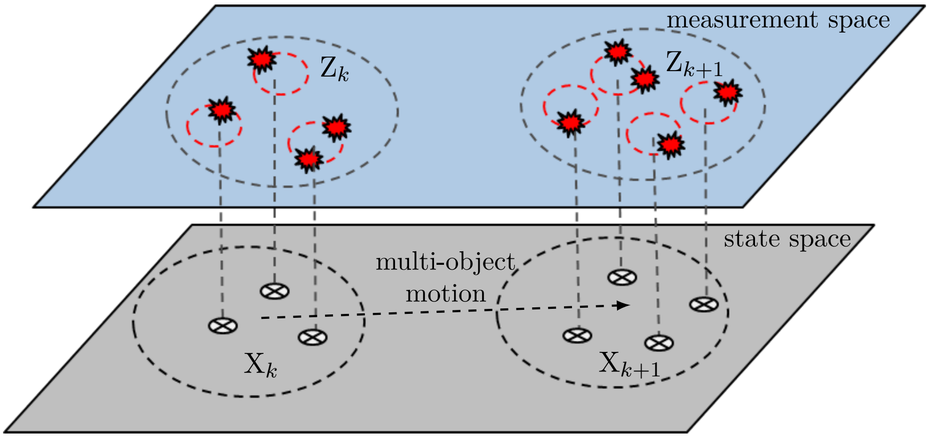


FIGURE 5.1: Illustration of RFS based multi-object tracking.

$$\langle f, g \rangle = \int f(x)g(x)dx \tag{5.3}$$

and the multi-object exponential notation of a real valued function $h(x)$ is given as

$$h^{\mathrm{X}} = \prod_{x \in \mathrm{X}} h(x) \tag{5.4}$$

which means the function has to be evaluated for all the vectors x in the RFS X and $h^{\mathrm{X}} = 1$ if $\mathrm{X} = \emptyset$. The generalized Kronecker delta function and the inclusion function (which denotes if a set is a subset of another set) are given by

$$\delta_{\mathrm{Y}}(\mathrm{X}) \triangleq \begin{cases} 1, & \text{if } \mathrm{X} = \mathrm{Y} \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

$$1_{\mathrm{Y}} \triangleq \begin{cases} 1, & \text{if } \mathrm{X} \subseteq \mathrm{Y} \\ 0, & \text{otherwise} \end{cases} \tag{5.6}$$

5.2 Statistics and Classes of RFS

The three fundamental statistical descriptors described in [Mah07b] for specifying the statistics of an RFS Ψ are

- belief-mass function $\beta_{\Psi}(T)$ - basis for deriving multi-object Markov densities and likelihood functions.
- multiobject probability density function $\pi(Y)$ - fundamental concept for multi-object Bayes filtering.
- probability generating functional $G_{\Psi}(h)$ - basis for deriving approximate multi-object Bayes filter variants.

The belief measure of the RFS $\Psi \subseteq \mathfrak{Y}$, also called as belief-mass function is the generalisation of probability-mass function, given in [Mah07b] as

$$\beta_{\Psi}(T) = \Pr(\Psi \subseteq T) \tag{5.7}$$

The multi-object probability density can be then derived based on the belief-mass function with the relation [Mah07b]

$$\pi(\mathrm{Y}) = \frac{\delta \beta_{\Psi}}{\delta \mathrm{Y}}(\emptyset) \tag{5.8}$$

For the multi-object tracking problem, considering all the possible number of elements the multi-object state X can have, the multi-object probability distribution $\pi(\mathrm{X})$ can take the forms

$$\pi(\mathrm{X}) = \begin{cases} \pi(\emptyset), & \text{if } \mathrm{X} = \emptyset \\ \pi\left(\left\{x^{(1)}\right\}\right), & \text{if } \mathrm{X} = \left\{x^{(1)}\right\} \\ \pi\left(\left\{x^{(1)}, x^{(2)}\right\}\right), & \text{if } \mathrm{X} = \left\{x^{(1)}, x^{(2)}\right\} \\ \vdots & \vdots \end{cases} \tag{5.9}$$

with

$$\int \pi(\mathrm{X}) \delta \mathrm{X} = 1 \tag{5.10}$$

The integration of the above equation however involves a set integral and can be represented considering all the possible events, i.e. summing over all the possible number of objects as

$$\int \pi(\mathrm{X}) \delta \mathrm{X} = \pi(\emptyset) + \sum_{i=1}^{\infty} \frac{1}{i!} \int \pi\left(\left\{x^{(1)}, \ldots, x^{(i)}\right\}\right) dx^{(1)} \ldots dx^{(i)} \tag{5.11}$$

which is computationally intractable [Mah07b]. In order to compute the set integral, the number of elements (vectors) n in the set should theoretically be $< \infty$. In that case, the cardinality distribution, i.e. the probability that there are n vectors in the RFS X is given as

$$\rho(n) = \Pr(|\mathrm{X}| = n) = \frac{1}{n!} \int \pi \left(\left\{ x^{(1)}, \ldots, x^{(n)} \right\} \right) dx^{(1)} \ldots dx^{(n)} \tag{5.12}$$

which is used for estimating the number of objects in the context of multi-object tracking. The probability generating functional of an RFS Ψ is the functional defined as the expected value and can be derived for the multi-object state X [Mah14] as

$$G_\Psi(h) = \mathbb{E}[h^\Psi] = \int h^{\mathrm{X}} \cdot \pi(\mathrm{X}) \delta \mathrm{X} \tag{5.13}$$

where h^X is the multi-object exponential notation of a test function $h(x)$ in equation 5.4 and $0 \leq G_\Psi(h) \leq 1$.

In [Mah07b] the above statistics of RFS are expressed with an example of a single twinkling star. Assuming that the twinkling star is observable with a probability r and has a spatial distribution $p(x)$, the belief-mass function can be given as

$$\begin{aligned} \beta(T) &= \Pr(\mathrm{X} = \emptyset) + \Pr(\mathrm{X} = \{x\}) \cdot \Pr(x \in T) \\ &= 1 - r + r \cdot p_{\mathrm{X}}(T) \end{aligned} \tag{5.14}$$

The multi-object probability density for the twinkling star can then be derived as

$$\pi(\mathrm{X}) = \begin{cases} \frac{\delta\beta}{\delta\emptyset}(\emptyset) = 1 - r, & \text{if } \mathrm{X} = \emptyset \\ \frac{\delta\beta}{\delta x}(\emptyset) = r \cdot p(x), & \text{if } \mathrm{X} = \{x\} \\ 0, & \text{if } |\mathrm{X}| \geq 2 \end{cases} \tag{5.15}$$

and the probability generating functional as

$$\begin{aligned} G(h) &= \int h^{\mathrm{X}} \cdot \pi(\mathrm{X}) \delta X \\ &= 1 \cdot \pi(\emptyset) + \int h(x) \cdot \pi\left(\{x\}\right) dx + 0 \\ &= 1 - r + r \int h(x) \cdot p(x) dx \end{aligned} \tag{5.16}$$

Based on these statistical properties, the RFS can further be categorised into different classes. These classes serve as the main foundations for the derivation of various RFS based multi-object filters.

1. *Poisson RFS* - foundation for the probability hypothesis density (PHD) filter.
2. *Identical Independently Distributed Cluster RFS* - foundation for the cardinalised probability hypothesis density (CPHD) filter.
3. *Multi-Bernoulli RFS* - basis for the multi-Bernoulli filters
4. *Labeled RFS* - basis for the class of labeled multi-Bernoulli filters.

5.2.1 Poisson Random Finite Set

In the context of representing the RFS $\mathrm{X} = \{x^{(1)}, x^{(2)}, x^{(3)}, \dots x^{(n)}\}$ in Poisson notation, if the expected number of objects can be considered as λ and spatial distribution of i^{th} object as $p(x^{(i)})$, then the multi-object distribution of the Poisson RFS is given by

$$\pi(\mathrm{X}) = e^{-\lambda} \cdot \lambda^n \, . \, p(x^{(1)}) \dots p(x^{(n)}) \tag{5.17}$$

In many implementations of multi-object tracking, the clutter process is modeled as Poisson distribution with λ_c expected clutter measurements per time step and a uniform clutter spatial distribution $c(z)$. The clutter density is then given as

$$k(z) = \lambda_c \cdot c(z) \tag{5.18}$$

5.2.2 Identical Independently Distributed Cluster Random Finite Set

A RFS X is an identical independently distributed cluster (i.i.d.c) RFS if it has the following two properties [Mah14]

1. The RFS has $n \geq 0$ number of objects, where n is an integer value drawn from the probability distribution $\rho(n)$ on the number of objects.
2. The objects of the RFS are spatially distributed according to a distribution $p_X(\cdot)$, where $\mathrm{X} = \left\{x^{(1)}, \dots, x^{(n)}\right\}$.

The multi-object probability density of the i.i.d.c RFS X is then given as

$$\pi(\mathrm{X}) = n! \cdot \rho(n) \cdot p(x^{(1)}) \dots p(x^{(n)}) \tag{5.19}$$

where $p(x^{(i)})$ represents the spatial distribution of the i^{th} object. Therefore, the i.i.d.c RFS can be understood as a generalisation of the Poisson RFS in which the cardinality distribution $\rho(n)$ is represented by a Poisson distribution.

5.2.3 Bernoulli Random Finite Set

A Bernoulli RFS X is characterised by spatial probability density $p_{\mathrm{X}}(.)$ as well as the probability of existence r. The probability that the Bernoulli RFS X is a singleton with an element x is given by the existence probability r and the probability that it is empty is given by $1 - r$. The spatial distribution is then given by the spatial distribution $p(x)$ of the object x. Therefore the parameters (r, p) completely define a Bernoulli RFS and it is used for representing a single object. The probability distribution of Bernoulli RFS is given as [Mah14]

$$\pi(\mathrm{X}) = \begin{cases} 1 - r, & \text{if } X = \emptyset \\ r \cdot p(x), & \text{if } X = \{x\} \\ 0, & \text{if } |X| \geq 2 \end{cases} \tag{5.20}$$

5.2.4 Multi-Bernoulli Random Finite Set

A multi-Bernoulli RFS is the union of fixed number of independent Bernoulli RFS [Mah14]. In case of multi-object tracking, if every object can be described as an individual Bernoulli RFS, then the multi-Bernoulli RFS of the complete state space X can be given as union of these independent Bernoulli RFSs. Suppose there are N objects, each represented by its individual Bernoulli RFS $\mathrm{X}^{(i)}$, then the multi-Bernoulli RFS is given as

$$\mathrm{X} = \bigcup_{i=1}^{N} \mathrm{X}^{(i)} \tag{5.21}$$

Corresponding to the parameter representation of a Bernoulli RFS, the multi-Bernoulli RFS can be represented by combining the individual object parameters as $\left\{\left(r^{(i)}, p^{(i)}\right)\right\}_{i=1}^{M}$. The multi-Bernoulli multiobject probability density is given by [Mah14]

$$\pi(x^{(1)} \dots x^{(n)}) = \prod_{j=1}^{N}(1 - r^{(j)}) \sum_{1<i_1 \neq \dots \neq i_n < N} \prod_{j=1}^{n} \frac{r^{(i_j)} p^{(i_j)}(x^{(j)})}{(1 - r^{(i_j)})} \tag{5.22}$$

and the cardinality distribution [Mah14]

$$\rho(n) = \prod_{j=1}^{N}(1 - r^{(j)}) \sum_{1<i_1 \neq \dots \neq i_n < N} \prod_{j=1}^{n} \frac{r^{(i_j)}}{(1 - r^{(i_j)})} \tag{5.23}$$

5.3 Labeled Random Finite Set

In multi-object tracking applications, it is mostly essential to have unique identities for each of the tracks. It is required for estimating object trajectories and know which track at time t_{k+1} was same as at time t_k, for association. The RFS classes and their multi-object probability distributions presented in previous section although covers the multi-object state estimation problem, do not explicitly consider a label (identity) for each object. In [VV13a] the class of Labeled RFS is proposed, where each object state is augmented with a distinct label $\ell \in \mathbb{L}$. The labels are drawn from a discrete label space $\mathbb{L} = \{\alpha_i : i \in \mathbb{N}\}$, where $\mathbb{N}$ is a set of positive integers and α_i's are distinct [VV13a].

The multi-object labeled RFS can therefore be written as

$$\mathbf{X} = \{(x^{(1)}, \ell^{(1)}), (x^{(2)}, \ell^{(2)}), \dots (x^{(n)}, \ell^{(n)})\} \subseteq \mathbb{X} \times \mathbb{L} \tag{5.24}$$

Moreover, each object should have an unique identity. Therefore a set of track labels $\mathcal{L}(\mathbf{X}) = \{\mathcal{L}(x, \ell) : (x, \ell) \in \mathbf{X}\}$ of the labeled multi-object state $\mathbf{X}$ is introduced in [VV13a]. $\mathcal{L}(x, \ell) = \ell$ is then the projection of labeled state (x, ℓ) on the space $\mathbb{X} \times \mathbb{L}$ to the label space $\mathbb{L}$ denoted as $\mathcal{L} : \mathbb{X} \times \mathbb{L} \mapsto \mathbb{L}$. Also, each object label should be distinct, which means two different objects cannot have the same label. A label can be assigned exactly to one object only if the condition $|\mathcal{L}(\mathbb{X})| = \mathbb{X}$ and the distinct label indicator is given as

$$\Delta(\mathbf{X}) = \delta_{|\mathbf{X}|}(\mathcal{L}(|\mathbf{X}|)) \tag{5.25}$$

which takes the value one if the object labels are distinct, zero otherwise. More detailed derivation and properties of labeled RFS are presented in [VV13a]

5.3.1 Labeled Multi-Bernoulli Random Finite Set

Augmenting each object of a multi-Bernoulli RFS with a distinct label gives the labeled multi-Bernoulli (LMB) RFS defined by the existence and spatial parameters of each object represented by it's corresponding label

$$\pi(\mathbf{X}) = \left\{(r^{(\ell)}, p^{(\ell)})\right\}_{\ell \in \mathbb{L}} \tag{5.26}$$

The multi-object probability density of LMB RFS on $\mathbb{X} \times \mathbb{L}$ is given by [VV13a]

$$\pi((x^{(1)}, \ell^{(1)}), \dots (x^{(n)}, \ell^{(n)})) = \delta_n(|l_1, \dots, l_n|) \times \prod_{i \in \mathbb{L}} (1 - r^{(i)}) \prod_{\ell=1}^{n} \frac{1_{\mathbb{L}} r^{(\ell)} p^{(\ell)}}{1 - r^{(\ell)}} \tag{5.27}$$

A compact notation of the LMB RFS as given in [VV13a] and [Reu14] using the multi-object exponential notation

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) w(\mathcal{L}(\mathbf{X})) p^{\mathbf{X}} \tag{5.28}$$

where

$$\begin{aligned} w(L) &= \prod_{i \in \mathbb{L}} (1 - r^{(i)}) \prod_{\ell \in L} \frac{1_{\mathbb{L}}(\ell) r^{(\ell)}}{1 - r^{(\ell)}}, \\ p(x, \ell) &= p^{(\ell)}(x). \end{aligned} \tag{5.29}$$

The multiple model version of the LMB filter derived in [RSD15] and used in this work is based on the LMB RFS.

5.3.2 Generalised Labeled Multi-Bernoulli RFS

The generalised labeled multi-Bernoulli (GLMB) RFS is a generalisation of the LMB RFS with the multi-object probability density given by [VV13a]

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{c \in \mathbb{C}} w^{(c)}(\mathcal{L}(\mathbf{X}))[p^{(c)}]^{\mathbf{X}} \tag{5.30}$$

where $\mathbb{C}$ is a discrete index set enabling multiple hypotheses of a set of track labels, weights $w^{(c)}$ and spatial distributions $p^{(c)}$ satisfy

$$\sum_{L \subseteq \mathbb{L}} \sum_{c \in \mathbb{C}} w^{(c)}(L) = 1, \tag{5.31}$$

$$\int p^{(c)}(x, \ell) dx = 1. \tag{5.32}$$

In GLMB RFS the tracks are statistically dependent unlike the LMB RFS. The sum component in equation 5.30 infers that in contrast to the LMB RFS, in the GLMB RFS multiple hypotheses of a set of track labels are considered facilitating arbitrary weights and cardinality distributions. In other words, a LMB RFS is a special case of GLMB RFS, where only one component c for each realisation of $\mathbf{X}$ is considered and weights depend on track existence probabilities $r^{(\ell)}$. As multiple hypotheses are considered for a set of track labels, with each hypothesis representing possible track to measurement associations, the data association uncertainty is directly included in the filter update [Reu14].

5.3.3 δ-Generalised Labeled Multi-Bernoulli RFS

The δ-Generalised Labeled Multi-Bernoulli (δ-GLMB) RFS is a special case of GLMB RFS, where an hypothesis or component c is represented by the association history ξ of a set of track labels I. The δ-GLMB RFS is realised from the GLMB RFS with the parameters,

$$\begin{aligned} \mathbb{C} &= \mathcal{F}(\mathbb{L}) \times \Xi \\ w^{(c)}(L) &= w^{(I,\xi)}(L) = w^{(I,\xi)}\delta_I(L) \\ p^{(c)} &= p^{(I,\xi)} = p^{\xi} \end{aligned}$$

where Ξ denotes a discrete space. The probability density of δ-GLMB is given as [VV13a]

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)}(\mathcal{L}(\mathbf{X}))[p^{(\xi)}]^{\mathbf{X}} \tag{5.33}$$

The update step of the LMB filter in section 5.6 uses the δ-GLMB representation of the multi-object state, to consider different association hypotheses. Further, the probability of number of objects n is obtained by summing all the weights of the hypotheses in which the number of tracks is $|I| = n$. The cardinality distribution as derived in [VV13a] is given by

$$\rho(n) = \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \sum_{L\in\mathcal{F}_n(\mathbb{L})} w^{(I,\xi)}\delta_I(L) = \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} \tag{5.34}$$

The PHD of the unlabeled version of δ-GLMB RFS is shown in [VV13a] to be

$$\begin{aligned} v(x) &= \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \sum_{l\in\mathbb{L}} p^{(\xi)}(x,\ell) L \subseteq \mathbb{L} 1_L(\ell) w^{(I,\xi)}\delta_I(L) \\ &= \sum_{l\in\mathbb{L}} \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} 1_I(\ell) p^{(\xi)}(x,\ell) \end{aligned} \tag{5.35}$$

The existence probability $r^{(\ell)}$ of each track in the context of multi-object tracking can then be extracted from the PHD of the δ-GLMB RFS, by considering the sum of weights of all the hypotheses that contain that particular track with label ℓ and is given by,

$$r^{(\ell)} = \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} 1_I(\ell) \tag{5.36}$$

5.4 Multi-Object Bayes Filter

As in case of a single-object Bayes filter, the multi-object Bayes filter can also be realised with a recursion of prediction and update steps. But the difference is, in case of multi-object Bayes filter, the multi-object likelihood function and multi-object Markov density need to be formulated. Mahler proposed the top-down formulation of the multi-object Bayes filtering based on the FISST. An object state represented as a random variable x in a single-object Bayes filter is a RFS X with set of target states in multi-object Bayes filter. The multi-object Bayes filter prediction and update according to [Mah07b] are:

$$\pi_+(\mathrm{X}|\mathrm{Z}^k) = \int f_+(\mathrm{X}_+|\mathrm{X})\pi(\mathrm{X}|\mathrm{Z}^k)\delta\mathrm{X}, \tag{5.37}$$

$$\pi(\mathrm{X}_+|\mathrm{Z}^{k+1}) = \frac{g(\mathrm{Z}|\mathrm{X}_+)\pi_+(\mathrm{X}|\mathrm{Z}^k)}{\int g(\mathrm{Z}|\mathrm{X}_+)\pi_+(\mathrm{X}|\mathrm{Z}^k)\delta\mathrm{X}}. \tag{5.38}$$

in which the multi-object Markov transition density is $f_+(\mathrm{X}_+|\mathrm{X})$ and the multi-object Likelihood function is $g(\mathrm{Z}|\mathrm{X}_+)$. Compared to the single-object Bayes filter, the multi-object Bayes filter recursion can be given as,

$$\begin{tikzcd}
\cdots \arrow[r] & \pi(\mathrm{X}|\mathrm{Z}_{1:k}) \arrow[r, "\text{predict}"] & \pi_+(\mathrm{X}_+|\mathrm{Z}_{1:k}) \arrow[r, "\text{correct}"] & \pi(\mathrm{X}_+|\mathrm{Z}_{1:k+1}) \arrow[r] & \cdots \\
 & & \begin{array}{c}\text{multi-object}\\ \text{Markov density}\\ f_+(\mathrm{X}_+|\mathrm{X})\end{array} \arrow[u] & \begin{array}{c}\text{multi-object}\\ \text{likelihood function}\\ g(\mathrm{Z}|\mathrm{X}_+)\end{array} \arrow[u] & \\
 & & \begin{array}{c}\text{multi-object}\\ \text{motion model}\\ \mathrm{X}_+ = \underbrace{T(\mathrm{X})}_{\text{surviving objects}} \cup \underbrace{B}_{\substack{\text{birth}\\ \text{objects}}}\end{array} \arrow[u] & \begin{array}{c}\text{multi-object}\\ \text{measurement model}\\ \mathrm{Z} = \underbrace{\Upsilon(\mathrm{X}_+)}_{\text{detections}} \cup \underbrace{C}_{\text{clutter}}\end{array} \arrow[u] &
\end{tikzcd}$$

Various approximations of the multi-object Bayes filter are realised based on the statistics of different RFS distributions as described in section 5.2. The probability hypothesis density (PHD) filter is realised by propagating the first order moment of the multi-object probability density. The cardinalised probability hypothesis density (CPHD) filter additionally approximates the second order moment in order to propagate the cardinality distribution. Cardinality balanced multi-Bernoulli (CB-MeMBer) filter is an approximation based on the statistics of the multi-Bernoulli RFS. [VV13a] proposed a closed form solution of the multi-object Bayes filter by introducing the generalised labeled multi-Bernoulli (GLMB) filter and the δ-generalised labeled multi-Bernoulli (δ-GLMB) filter, based on the statistics of the labeled RFS described in section 5.3. In order to address the computational complexity of the GLMB filter, labeled multi-Bernoulli (LMB) filter is introduced in [RVVD14] and is used as a basis in this work. After presenting the multi-object likelihood functions and multi-object Markov densities, concepts of the various multi-object Bayes filters are briefly presented and compared in the following sections.

5.4.1 Multi-Object Likelihood Functions

In contrast to the single-object measurement likelihood $g(z|x_+)$, which only represents the distance based closeness of a measurement to a track considering measurement uncertainties, the multi-object likelihood $g(\mathrm{Z}|\mathrm{X}_+)$ also considers the characteristics of the sensor like FoV, probability of detection, clutter etc. The multi-object measurement model introduced

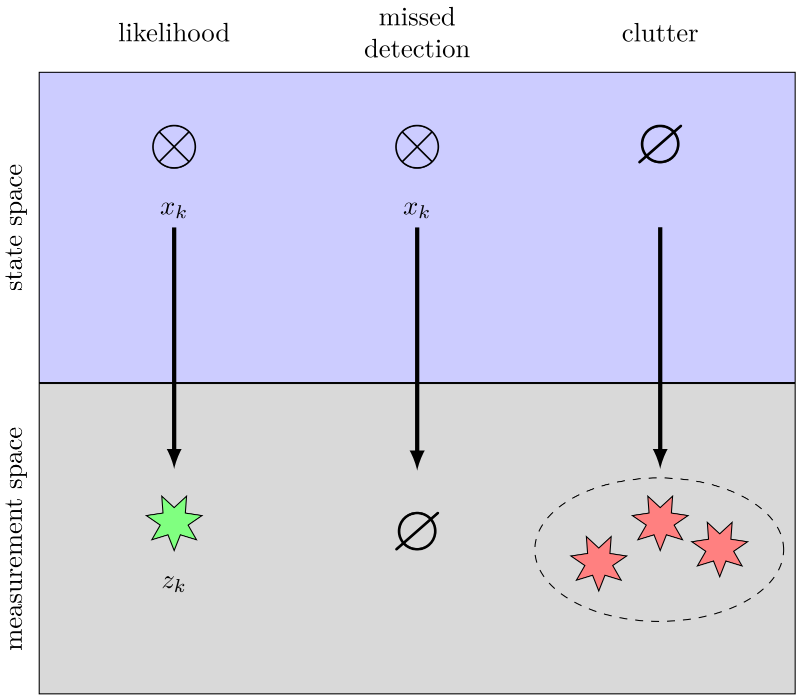


FIGURE 5.2: Multi-Object Likelihood [Mah07b]

in [Mah07b] is based on various assumptions. Every object is assumed to generate one detection and is detected by the sensor with a single-object measurement likelihood $g(z|x_+)$. As described in the Bernoulli RFS statistics, each object is assumed to be detected with a probability $p_D(x_+)$ and not detected with a probability $1-p_D(x_+)$. Moreover, clutter measurement is assumed to be Poisson distributed with λ_c number of clutter measurements and spatial density $c(z)$. The measurements are also assumed to be independent of the object's actual state. Thus if each object generates one measurement which can be represented as RFS $\Upsilon(x_+)$, it will be a singleton set with

$$\Upsilon(x_+) = \begin{cases} \emptyset & \text{if undetected} \quad (\text{probability } 1 - p_D(x_+)) \\ \{z\} & \text{if detected} \quad (\text{probability } p_D(x_+)), \end{cases} \tag{5.39}$$

which is a Bernoulli distribution as in (5.20). Therefore the RFS measurement model, assuming all the objects' measurement and clutter measurements to be statistically independent is given as [Mah07b]

$$\mathrm{Z} = \Upsilon(x_+^{(1)}) \cup \cdots \cup \Upsilon(x_+^{(n)}) \cup C, \tag{5.40}$$

with $\Upsilon(x_+^{(1)}) \cup \cdots \cup \Upsilon(x_+^{(n)}) = \Upsilon(\mathrm{X}_+)$ representing the detections from the predicted multi-object state X_+, which is a multi-Bernoulli RFS and C the clutter RFS with Poisson distribution. Different possible hypothesis of the measurement source are considered in the multi-object measurement modeling, as illustrated in Figure 5.2. The source of a measurement can be either an object or clutter and additionally the object may even be missed. Therefore the multi-object likelihood considers all the measurement-to-track associations between sets of measurements and tracks, including clutter and missed detection. The multi-object likelihood in [Mah07b] thus averages over all the measurement hypotheses and is given as

$$g(\mathrm{Z}|\mathrm{X}_+) = \pi_C(\mathrm{Z})\pi(\emptyset|\mathrm{X}_+) \sum_\theta \prod_{i:\theta(i)>0} \frac{p_D(x_+^{(i)}) \cdot g(z_{\theta(i)}|x_+^{(i)})}{(1 - p_D(x_+^{(i)})) \cdot \lambda_c c(z_{\theta(i)})}, \tag{5.41}$$

$$\pi(\emptyset|\mathrm{X}_+) = \prod_{i=1}^{n} (1 - p_D(x_+^{(i)})), \tag{5.42}$$

$$\pi_C(\mathrm{Z}) = e^{-\lambda_C} \prod_{z \in \mathrm{Z}} \lambda_c c(z). \tag{5.43}$$

where θ represents all the possible measurement-to-track association hypotheses, $\pi(\emptyset|\mathrm{X}_+)$ the probability that all detections are missed and $\pi_C(\mathrm{Z})$ the probability all measurements are clutter.

5.4.2 Multi-Object Markov Densities

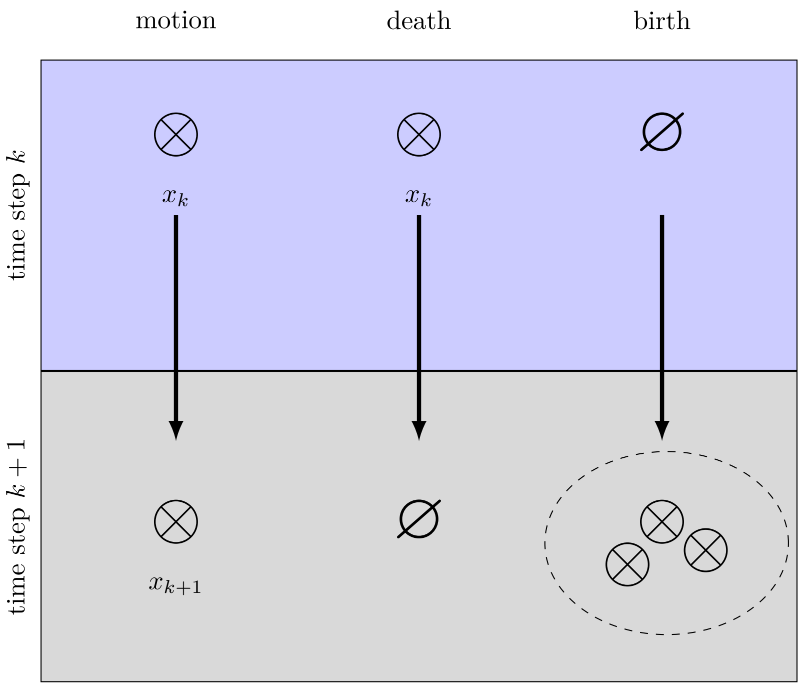


Figure 5.3: Multi-Object Markov Densities [Mah07b]

As the multi-object likelihood function considers the clutter and missed detections, appearance and disappearance of objects are considered in multi-object Markov density derived in [Mah07b]. Every object at the time step k is assumed to survive into time $k+1$ with a probability $p_S(x)$ or disappear with a probability $1-p_S(x)$. Also, each object is assumed to evolve independent of other objects with a single-object Markov transition density $f_+(x_+|x)$. Assuming the transition of each object into a RFS state $T(x)$, such that [Mah07b]

$$T(x) = \begin{cases} \emptyset & \text{if disappears (probability } 1 - p_S(x)) \\ \{x_+\} & \text{if survives (probability } p_S(x)), \end{cases} \tag{5.44}$$

which is again a Bernoulli RFS as in 5.20. Additionally, considering B as the RFS of new born objects at time $k+1$, the predicted RFS can be given as,

$$\mathrm{X}_{+} = T(x^{(1)}) \cup \cdots \cup T(x^{(n)}) \cup B, \tag{5.45}$$

with $T(x^{(1)}) \cup \cdots \cup T(x^{(n)}) = T(\mathrm{X})$ representing the multi-object state X as multi-Bernoulli RFS and B the birth RFS, which can be modeled with Poisson distribution. Similar to the case of a measurement likelihood, different prediction hypotheses are considered for multi-object Markov density as shown in Figure (5.3). An object a time step k can either evolve to survive or disappear at $k+1$. A new object may also appear at $k+1$, which denotes it's birth. Additionally, an object can spawn other new objects as well, which is however not considered in this work. The multi-object Markov density must consider all these object prediction hypotheses θ and is given by [Mah07b] as

$$f_{+}(\mathrm{X}_{+}|\mathrm{X}) = \pi_B(\mathrm{X}_{+})\pi_{+}(\emptyset|\mathrm{X}) \sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_S(x^{(i)}) \cdot f_{+}(x_{+}^{\theta(i)}|x^{(i)})}{(1-p_S(x^{(i)})) \cdot \lambda_B b(x_{+}^{\theta(i)})}, \tag{5.46}$$

$$\pi(\emptyset|\mathrm{X}) = \prod_{i=1}^{|\mathrm{X}|}(1-p_S(x^{(i)})), \tag{5.47}$$

$$\pi_B(\mathrm{X}_{+}) = e^{-\lambda_B} \prod_{i=1}^{|\mathrm{X}_{+}|} \lambda_B b(x_{+}^{(i)}). \tag{5.48}$$

where λ_B is the expected number of new born objects distributed according to probability density $b(\cdot)$. $\pi_{+}(\emptyset|\mathrm{X})$ represents the probability that none of the objects survive and $\pi_B(\mathrm{X}_{+})$ the probability that all the objects are newly born.

5.5 Probability Hypothesis Density Filter

An approximation of the multi-object Bayes filter by propagating only the first order moment of the multi-object probability density $\pi(\mathrm{X})$ was first proposed in [Mah03], called the probability hypothesis density (PHD) filter. In a single-object case, first order moment of the probability density function is the mean or the expected value of the object state. The object state is estimated by propagating the first order moment in time, like in the case of constant gain Kalman filter as given in (A.2.1).

The PHD filter can be considered as the multi-object counterpart of the constant gain Kalman filter, in which the first order moment of the multi-object probability density is propagated The first order moment of the multi-object probability density is called as the PHD $D(x)$ [Mah03], derived from the concept of intensity density in point process theory

[DVJ88, DVJ03]. The multi-object Bayes filter recursion in correspondence to the single-object Bayes filter can then be realised propagating the PHD as:

$$\begin{tikzcd}
\ldots \arrow[r] & \pi(\mathrm{X}|\mathrm{Z}_{1:k}) \arrow[r, "\text{predict}"] \arrow[d] & \pi_+(\mathrm{X}_+|\mathrm{Z}_{1:k}) \arrow[r, "\text{correct}"] \arrow[d] & \pi(\mathrm{X}_+|\mathrm{Z}_{1:k+1}) \arrow[r] \arrow[d] & \ldots \\
 & D_{k|k}(x|\mathrm{Z}_{1:k}) \arrow[r, "\text{predict}"] & D_{k+1|k}(x|\mathrm{Z}_{1:k}) \arrow[r, "\text{correct}"] & D_{k+1|k+1}(x|\mathrm{Z}_{1:k+1}) &
\end{tikzcd}$$

The naive definition of the expected value of the multi-object RFS X with multi-object probability density $\pi(\mathrm{X})$ can be given as

$$E[X] = \int \mathrm{X} \cdot \pi(\mathrm{X}) \delta \mathrm{X} \tag{5.49}$$

However, the above integral is not mathematically defined. But by substituting the RFS X with the Dirac delta function concentrated at X, the expected value for the multi-object case can be given as

$$D(x) = \int \delta_{\mathrm{X}}(x) \cdot \pi(\mathrm{X}) \delta \mathrm{X} \tag{5.50}$$

Considering a two dimensional multi-object state space, the PHD can therefore be visualised

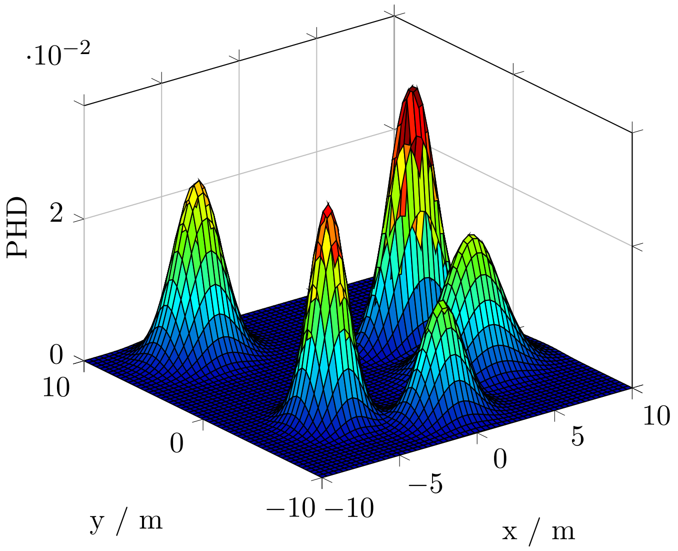


Figure 5.4: Visual illustration of probability hypothesis density.

as the expected value in Figure 5.4, where the peaks having the highest density denote the expected object locations. Consequently, the expected number of objects in the region S can be given based on the PHD as

$$\hat{N}_S = \int_S D(x) dx \tag{5.51}$$

The PHD filter is realised as an approximation of the multi-object Bayes filter with recursive prediction and correction of multi-object PHD. The prediction step of the PHD filter proposed in [Mah03] considers the motion of surviving objects, object disappearance, birth of new objects as well as spawning of new objects from existing objects. Object spawning is however not considered in this work. The prediction of the prior PHD is given by

$$D_+(x) = \int \underbrace{p_S(\xi)}_{\substack{\text{probability}\\ \text{of survival}}} \quad \underbrace{f_+(x|\xi)}_{\substack{\text{Markov}\\ \text{transition density}}} \quad \underbrace{D(\xi)}_{\text{prior PHD}} \; d\xi + \quad \underbrace{b(x)}_{\substack{\text{object}\\ \text{birth density}}} \tag{5.52}$$

The update step of the PHD filter considers the measurement likelihood, clutter and missed detection. The likelihood that an object generates a measurement is considered as $g(z|x)$, as in the standard measurement model. The object is assumed to be detected with a probability $p_D(x)$ and missed with a probability $1 - p_D(x)$. Clutter is assumed to be Poisson distributed with λ_c number of clutter measurements and a spatial distribution $c(z)$. Moreover, it is assumed that the milt-object prediction is Poisson distributed. The update of the predicted PHD is then given as

$$D(x) \approx \underbrace{1 - p_D(x)D_+(x)}_{\text{missed detection}} \quad + \underbrace{\sum_{z\in \mathrm{Z}} \frac{p_D(x)g(z|x)D_+(x)}{\lambda_c c(z) + \int p_D(\xi)g(z|\xi)D_+(\xi)d\xi}}_{\substack{\text{update predicted PHD}\\ \text{with all } |\mathrm{Z}| \text{ measurements}}} \tag{5.53}$$

A computationally tractable PHD filter can mainly be implemented using two methods, either as sequential Monte Carlo (SMC) approximation as in [Mah07b, SW03] or as Gaussian mixture implementation (GM-PHD) proposed in [VM06], which is based on the Gaussian sum filters in [SA71, AS72]. In case of SMC implementation the posterior density is approximated by sampling certain number of particles as

$$D(x) = \sum_{i=1}^{v} w^{(i)} \delta_{x^{(i)}}(x) \tag{5.54}$$

In case of GM-PHD filter, the PHDs are approximated by Gaussian mixtures as

$$D(x) = \sum_{i=1}^{J} w^{(i)} \mathcal{N}(x; \hat{x}^{(i)}, \underline{\mathrm{P}}^{(i)}), \tag{5.55}$$

where the statistics of each Gaussian component is given by it's mean $\hat{x}^{(i)}$ and covariance $\underline{\mathrm{P}}^{(i)}$, $w^{(i)}$ represents the weight of each Gaussian component and J the number of Gaussian components. The weights of particles in SMC and weights of Gaussians in GM implementation are not normalised, as the integral over the PHD denotes the number of objects. Compared to the SMC implementation of the PHD filter, the GM-PHD filter is easier to implement and computationally light. The filter recursion of GM-PHD can be given as

$$\begin{tikzcd}
\dots \arrow[r] & D_{k|k}(x|\mathrm{Z}_{1:k}) \arrow[r, "\text{predict}"] \arrow[d] & D_{k+1|k}(x|\mathrm{Z}_{1:k}) \arrow[r, "\text{correct}"] \arrow[d] & D_{k+1|k+1}(x|\mathrm{Z}_{1:k+1}) \arrow[d] \\
\dots \arrow[r] & \sum_{i=1}^{J_{k|k}} w^{(i)}\mathcal{N}(x;\hat{x},\underline{\mathrm{P}}^{(i)}) \arrow[r, "\text{predict}"] & \sum_{i=1}^{J_{k+1|k}} w_+^{(i)}\mathcal{N}(x;\hat{x}_+^{(i)},\underline{\mathrm{P}}_+^{(i)}) \arrow[r, "\text{correct}"] & \sum_{i=1}^{J_{k+1|k+1}} w_+^{(i)}\mathcal{N}(x;\hat{x}_+^{(i)},\underline{\mathrm{P}}_+^{(i)})
\end{tikzcd}$$

Although the PHD filter is easy to implement, it produces an unstable estimate of number of objects, as only the first order moment of the multi-object distribution is propagated in time [EWBS05a]. In order to improve the estimation of the number of objects, the cardinalised probability hypothesis density (CPHD) filter was introduced by Mahler [Mah07b], which is a partial second-order moment filter. In CPHD filter, along with the PHD $D(x)$, the cardinality distribution $\rho(n)$ of the multi-object density $\pi(x)$ is also propagated. Similar to the PHD filter, the CPHD filter can also be realised either with SMC or Gaussian mixture implementation. Further details and implementation of CPHD filter can be found in [Mah07b] and [VVC07]. Also a multiple model probability hypothesis density (MMPHD) is proposed in [VPT06], in which multiple object motion models are used in filter prediction, to address the motion of maneuvering objects. Various applications of the MMPHD can be found in literature, for example in [MRD13] the MMPHD filter is used for tracking road users in intersection scenarios, using laserscanners and camera.

5.6 Labeled Multi-Bernoulli Filter

As the PHD filter approximates the multi-object Bayes filter by propagating only the first order moment of the multi-object density, the CPHD filter also propagates it's cardinality distribution. Unlike the PHD and CPHD filters, the multi-object Bayes filter can also be realised by approximating the multi-object density by multi-Bernoulli distribution as in 5.22. The cardinality balanced multi-target multi-Bernoulli (CB-MeMBer) filter [VVC09] is one such filter, which propagates the parameters of multi-Bernoulli distribution in time. Details and implementations of the CB-MeMBer filter are described in [VVC09]. However, the CB-MeMBer filter can overestimate the number of objects if the clutter rate is high and requires a high detection probability [Reu14]. An exact closed form solution of the multi-object Bayes filter is proposed in [VV13a] on the basis of labeled RFS. Vo and Vo introduced the generalised labeled multi-Bernoulli (GLMB) filter and it is shown that the δ-generalised labeled multi-Bernoulli (δ-GLMB) density is closed under multi-object prediction and update. In case of δ-GLMB, both the prediction and update of tracks requires generating all the possible hypothesis, which means with the increase in number of objects and measurements, the computational complexity increases exponentially in both prediction and update. Reuter in [RVVD14] suggested the labeled multi-Bernoulli (LMB) filter, where multi-object posterior and prediction in δ-GLMB are approximated by LMB RFS. This approximation reduces the total required hypotheses in the prediction step. It is shown in [RVVD14] that the LMB reduces the computational complexity by approximation of posterior, but still produces identical results of δ-GLMB filter in many scenarios.

5.6.1 Prediction

In the multi-object Bayesian recursion implementation of LMB filter, the multi-object posterior density of the previous time step is an approximated LMB RFS as in (5.26) of the form

$$\pi(\mathbf{X}) = \Delta(\mathbf{X})w(\mathcal{L}(\mathbf{X}))p^{\mathbf{X}} \tag{5.56}$$

where

$$\begin{aligned} w(L) &= \prod_{i\in\mathbb{L}}(1-r^{(i)})\prod_{\ell\in L}\frac{1_{\mathbb{L}}(\ell)r^{(\ell)}}{1-r^{(\ell)}}, \\ p(x,\ell) &= p^{(\ell)}(x). \end{aligned} \tag{5.57}$$

Also the multi-object birth intensity is considered to be an LMB RFS of the form

$$\pi_B(\mathbf{X}) = \Delta(\mathbf{X})w_B(\mathcal{L}(\mathbf{X}))[p_B]^{\mathbf{X}} \tag{5.58}$$

where

$$w_B(I) = \prod_{i\in\mathbb{B}}(1-r_B^{(i)})\prod_{\ell\in I}\frac{1_{\mathbb{B}}(\ell)r_B^{(\ell)}}{1-r_B^{(\ell)}}, \tag{5.59}$$

$$p_B(x,\ell) = p_B^{(\ell)}(x). \tag{5.60}$$

with the labels $l \in \mathbb{B}$ of the new born objects distinct and doesn't have the same labels as the surviving objects i.e. $\mathbb{L}\cap\mathbb{B} = \emptyset$.

The prediction density in a current time step is union of the surviving tracks of the previous time step in a label space $\mathbb{L}$ and the newly born tracks in current time step in label space $\mathbb{B}$. The tracks from previous time step are considered to survive with a probability of $p_S(x,\ell)$ and evolve according to a standard Markov transition density $f_+(x|x',\ell)$ or disappear with a probability $q_S(x,\ell) = 1 - p_S(x,\ell)$. The predicted LMB distribution in label space $\mathbb{L}_+ = \mathbb{L}\cup\mathbb{B}$ is [RVVD14]

$$\pi_+ = \left\{(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)})\right\}_{\ell\in\mathbb{L}} \cup \left\{(r_B^{(\ell)}, p_B^{(\ell)})\right\}_{\ell\in\mathbb{B}}. \tag{5.61}$$

with

$$r_{+,S}^{(\ell)} = \eta_S(\ell)r^{(\ell)}, \tag{5.62}$$

$$p_{+,S}^{(\ell)} = \langle p_S(\cdot,\ell)f_+(x|\cdot,\ell), p(\cdot,\ell)\rangle / \eta_S(\ell), \tag{5.63}$$

$$\eta_S(\ell) = \int \langle p_S(\cdot,\ell)f_+(x|\cdot,\ell), p(\cdot,\ell)\rangle\, dx. \tag{5.64}$$

SMC and Gaussian mixture implementation of the LMB filter are presented in [RVVD14]. The GM implementation of LMB filter is used in this work and is therefore described briefly here. In a GM implementation, the posterior of the tracks are represented by mixture of Gaussian distributions. Posterior density $p^{(\ell)}(\cdot)$ of each track $\ell \in \mathbb{L}$ is given by a mixture of Gaussians

$$p^{(\ell)}(x) = \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)}\mathcal{N}(x;\hat{x}^{(\ell,j)}, \underline{\mathrm{P}}^{(\ell,j)}) \tag{5.65}$$

where J denotes the number of Gaussian components, $w^{(\ell,j)}$ is the weight of the j^{th} Gaussian component, which needs to be normalized, $\hat{x}^{(\ell,j)}$ is the mean value and $\underline{\mathrm{P}}^{(\ell,j)}$ the respective error covariance of the Gaussian component. For the filter prediction in GM-LMB implementation, each Gaussian component is predicted with Kalman filter prediction equations

$$\hat{x}_+^{(\ell,j)} = \underline{\mathrm{F}}\hat{x}^{(\ell,j)}, \tag{5.66}$$

$$\underline{\mathrm{P}}_+^{(\ell,j)} = \underline{\mathrm{F}}\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}^\top + \underline{\mathrm{Q}}. \tag{5.67}$$

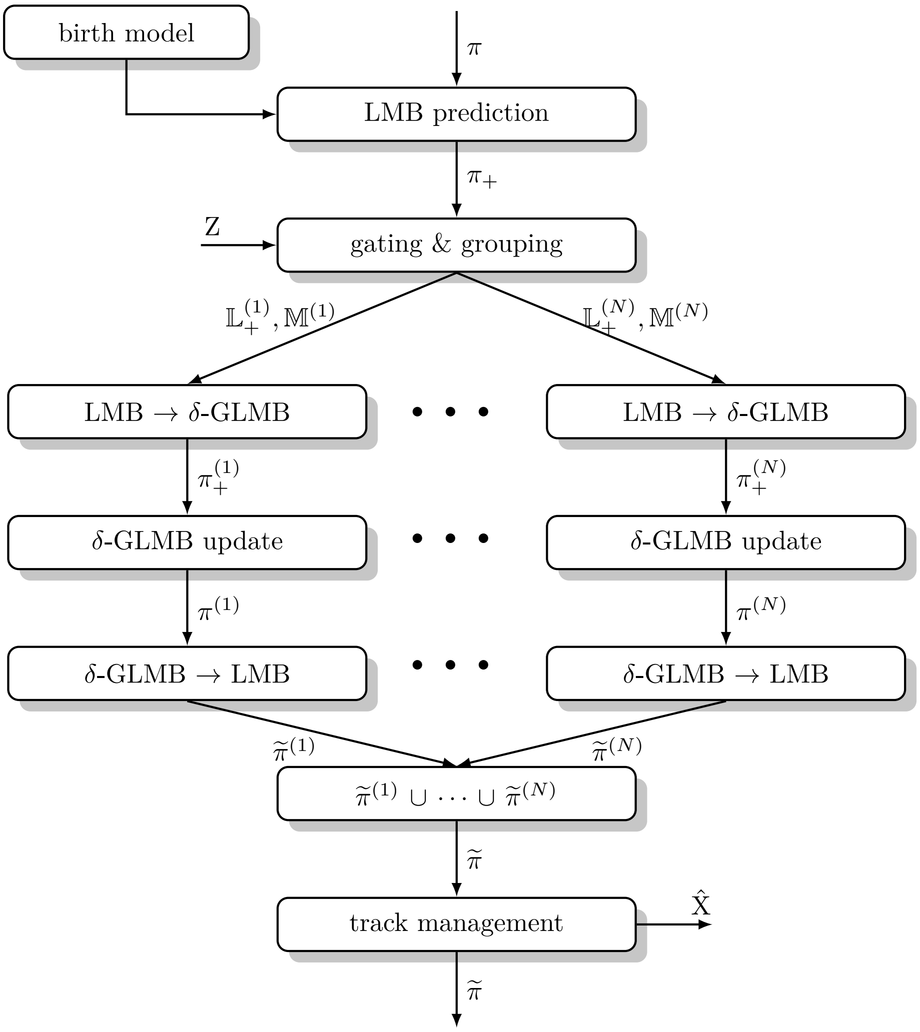


Figure 5.5: LMB filter algorithm (cf. [RVVD14])

with linear transition matrix $\underline{\mathrm{F}}$ of a single motion model and process noise $\underline{\mathrm{Q}}$. In case non-linear motion models are used for prediction, the EKF or UKF versions of the Kalman filter can be implemented. The survival probability of a track is considered as p_S for it's the existence prediction. The predicted existence probability and spatial distribution of a track are derived in [RVVD14] as

$$r_{+,S}^{(\ell)} = r^{(\ell)} p_S, \tag{5.68}$$

$$p_{+,S}^{(\ell)}(x) = \sum_{j=1}^{J^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}(x; \hat{x}_+^{(\ell,j)}, \underline{\mathrm{P}}_+^{(\ell,j)}) \tag{5.69}$$

Similar to the surviving tracks, $|\mathbb{B}|$ number of new new born tracks $\ell \in \mathbb{B}$ at current time step are also assumed to be a mixture of Gaussian components

$$p_B^{(\ell)}(x) = \sum_{j=1}^{J_B^{(\ell)}} w_B^{(\ell,j)} \mathcal{N}(x; \hat{x}_B^{(\ell,j)}, \underline{\mathrm{P}}_B^{(\ell,j)}) \tag{5.70}$$

where J_B denotes the number of birth Gaussian components, $w_B^{(\ell,j)}$ is the weight of the j^{th}

birth Gaussian component, $\hat{x}_B^{(\ell,j)}$ is the mean value and $\underline{\mathrm{P}}_B^{(\ell,j)}$ the respective error covariance of the birth Gaussian component, which are all specified in the birth model.

5.6.2 Update

It is shown in [RVVD14] that the LMB is not closed under update operation and the predicted LMB density needs to be expressed in δ-GLMB form. Therefore before the update step, the predicted LMB density (5.61) needs to be represented in δ-GLMB form and is given by [VV13a, RVVD14]

$$\pi_+(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{I_+ \in \mathcal{F}(\mathbb{L}_+)} w_+^{(I_+)} \delta_{I_+}(\mathcal{L}(\mathbf{X}_+)) \, [p_+]^{\mathbf{X}_+} \tag{5.71}$$

where I_+ denotes a hypothesis containing a set of track labels and for representing the LMB RFS in δ-GLMB form all the possible hypotheses corresponding to the permutations of track labels need to be generated. With the new set of measurements Z and no history of association maps available, the δ-GLMB posterior can then be given according to the δ-GLMB update from [VV13a] and [RVVD14] as

$$\pi_+(\mathbf{X}_+|\mathrm{Z}) = \Delta(\mathbf{X}) \sum_{(I_+,\theta) \in \mathcal{F}(\mathbb{L}_+) \times \Theta_{I_+}} w^{(I_+,\theta)}(\mathrm{Z}) \delta_{I_+}(\mathcal{L}(\mathbf{X})) \left[p^{(\theta)}(.|\mathrm{Z})\right]^{\mathbf{X}}. \tag{5.72}$$

where θ denotes the measurement to track association map for the track labels in each of the hypothesis I_+. The posterior weight for each of the hypothesis $w^{(I_+,\theta)}$ with the measurement set $\mathbb{M}$, the measurement updated posterior spatial distribution of each track $p^{(\theta)}(x,\ell|\mathrm{Z})$ and likelihoods $\eta_{\mathrm{Z}}^{(\theta)}(\ell)$ are given by [VV13a]

$$w^{(I_+,\theta)}(\mathrm{Z}) = \frac{\delta_{\theta^{-1}(\{0 \cup \mathbb{M}\})}(I_+) w_+(I_+) [\eta_{\mathrm{Z}}^{(\theta)}]^{I_+}}{\sum_{(I_+,\theta) \in \mathcal{F}(\mathbb{L}_+) \times \Theta_{I_+}} \delta_{\theta^{-1}(\{0 \cup \mathbb{M}\})}(I_+) w_+(I_+) [\eta_{\mathrm{Z}}^{(\theta)}]^{I_+}}, \tag{5.73}$$

$$p^{(\theta)}(x,\ell|\mathrm{Z}) = \frac{p_+(x,\ell)\psi_{\mathrm{Z}}(x,\ell;\theta)}{\eta_{\mathrm{Z}}^{(\theta)}(\ell)}, \tag{5.74}$$

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \langle p_+(\cdot,\ell), \psi_{\mathrm{Z}}(\cdot,\ell;\theta) \rangle \,, \tag{5.75}$$

$$\psi_{\mathrm{Z}}(\cdot,\ell;\theta) = \delta_0(\theta(\ell))(1-p_D)(x,\ell) + (1-\delta_0(\theta(\ell))) \frac{p_D(x,\ell) g(z_{\theta(\ell)}|x,\ell)}{\kappa(z_{\theta(\ell)})} \tag{5.76}$$

In case of GM implementation as in [RVVD14], $p_+(x,\ell)$ is the predicted density (5.69) with mean (5.66) and covariance (5.67). The association likelihood of a track to single measurement or missed detection then becomes

$$\psi_{\mathrm{Z}}(x,\ell;\theta) = \begin{cases} \dfrac{p_D}{\kappa(z_{\theta(\ell)})} \mathcal{N}(z_{\theta(\ell)}; \underline{\mathrm{H}}x, \underline{\mathrm{R}}) & \text{if } \theta(\ell) > 0 \\ (1-p_D) & \text{if } \theta(\ell) = 0. \end{cases} \tag{5.77}$$

where $\underline{\mathrm{H}}$ is the measurement matrix and $\underline{\mathrm{R}}$ the measurement noise covariance matrix from the standard Kalman filter. Multiplying spatial distribution of the track (5.69) with the likelihood in (5.77) gives

$$p_+(x,\ell)\cdot\psi_{\mathrm{Z}}(x,\ell;\theta) = \begin{cases} \dfrac{p_D}{\kappa(z_{\theta(\ell)})} \sum\limits_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}) \cdot \mathcal{N}(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}), \text{if } \theta(\ell) > 0 \\ (1-p_D) \cdot \sum\limits_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}(x; \hat{x}_+^{(\ell,j)}, \underline{\mathrm{P}}_+^{(\ell,j)}), \quad \text{if } \theta(\ell) = 0. \end{cases} \tag{5.78}$$

which in case a measurement is associated with the track ($\theta(\ell) > 0$) follows the innovation equations of Kalman filter with

$$z_+^{(\ell,j)} = \mathrm{H}\hat{x}_+^{(\ell,j)}, \tag{5.79}$$

$$\underline{\mathrm{S}}^{(\ell,j)} = \underline{\mathrm{H}}\underline{\mathrm{P}}_+^{(\ell,j)}\underline{\mathrm{H}}^\top + \underline{\mathrm{R}}, \tag{5.80}$$

$$\underline{\mathrm{K}}^{(\ell,j)} = \underline{\mathrm{P}}_+^{(\ell,j)}\underline{\mathrm{H}}^\top[\underline{\mathrm{S}}^{(\ell,j)}]^{-1}, \tag{5.81}$$

$$\hat{x}^{(\ell,j,\theta)}(\mathrm{Z}) = \hat{x}_+^{(\ell,j)} + \underline{\mathrm{K}}^{(\ell,j)}(z_{\theta(\ell)} - z_+^{(\ell,j)}), \tag{5.82}$$

$$\underline{\mathrm{P}}^{(\ell,j)} = \underline{\mathrm{P}}_+^{(\ell,j)} - \underline{\mathrm{K}}^{(\ell,j)}\underline{\mathrm{S}}^{(\ell,j)}[\underline{\mathrm{K}}^{(\ell,j)}]^\top. \tag{5.83}$$

and the updated state and covariances are same as the predicted mean and covariances in case of missed detection ($\theta(\ell) = 0$). Then the normalization constant is given as

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \begin{cases} \dfrac{p_D}{\kappa(z_{\theta(\ell)})} \sum\limits_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}) & \text{if } \theta(\ell) > 0 \\ (1-p_D) & \text{if } \theta(\ell) = 0. \end{cases} \tag{5.84}$$

Substituting the above equations in (5.74) gives the updated posterior spatial distribution of a track if $\theta(\ell) > 0$ as

$$p^{(\theta)}(x,\ell|\mathrm{Z}) = \frac{\dfrac{p_D}{\kappa(z_{\theta(\ell)})} \sum\limits_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}) \cdot \mathcal{N}(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)})}{\dfrac{p_D}{\kappa(z_{\theta(\ell)})} \sum\limits_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)})} \tag{5.85}$$

and if $\theta(\ell) = 0$ as

$$p^{(\theta)}(x,\ell|\mathrm{Z}) = \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}(x; \hat{x}_+^{(\ell,j)}, \underline{\mathrm{P}}_+^{(\ell,j)}) \tag{5.86}$$

In case of higher number of tracks and measurements, the computational complexity involved in the calculation of the above update step increases exponentially. In order to reduce the computational effort in such cases, a method for gating and grouping measurements and tracks is proposed in [RVVD14]. By gating, only the most relevant measurements $\mathrm{Z}^{(\ell)}$ for a track ℓ are considered. In case of GM implementation, the relevancy, i.e. whether a measurement $z_{\theta(\ell)}$ lies within the gate of a track ℓ is found based on the statistical Mahanalobis distance (MHD) given as

$$d_{MHD}^2(\ell) \triangleq \min_{j \in J^{(\ell)}} \left[\left(z_{\theta(\ell)} - z_+^{(\ell,j)} \right)^\top \underline{\mathrm{S}}^{(\ell,j)} \left(z_{\theta(\ell)} - z_+^{(\ell,j)} \right) \right] \tag{5.87}$$

The measurement $z_{\theta(\ell)}$ is considered to lie within the gate of the track ℓ, if any of the Gaussian component of the track among $J^{(\ell)}$ components satisfy the condition $d^2_{MHD}(\ell) < \vartheta$. The gating threshold ϑ can be chosen based on the application scenario. Grouping of predicted tracks is then done by considering the set of measurements that are commonly relevant to those tracks i.e. if atleast one of the measurements in the set lies within the gate of all those tracks. If there are N number of groups formed by the above grouping method, the predicted LMB distribution in (5.61) can then be partitioned as

$$\pi_+ = \bigcup_{i=1}^{N} \left\{ \left(r_{+,i}^{(\ell)}, p_{+,i}^{\ell} \right) \right\}_{l\in\mathbb{L}_+^{(i)}} \tag{5.88}$$

with $\mathbb{L}_+^{(i)}$ representing the track labels present in a group $\mathcal{G}^{(i)}$ and $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$.

Now each of the group $\mathcal{G}^{(i)}$ containing predicted tracks with labels $\mathbb{L}_+^{(i)}$ is converted into δ-GLMB RFS representation equivalent to (5.71) as

$$\pi_+^{(i)}(\widetilde{\mathbf{X}}_+^{(i)}) = \Delta(\widetilde{\mathbf{X}}_+^{(i)}) \sum_{I_+\in\mathcal{F}(\mathbb{L}_+^{(i)})} w_{+,i}^{(I_+)} \delta_{I_+}(\mathcal{L}(\widetilde{\mathbf{X}}_+^{(i)}))\, [p_+]^{\widetilde{\mathbf{X}}_+^{(i)}} \tag{5.89}$$

On the other hand, with the increase in the number of tracks $\left|\mathbb{L}_+^{(i)}\right|$ in a group $\mathcal{G}^{(i)}$, the number of hypotheses $|I_+|$ that need to be generated also increases, which should ideally cover all the possible combinations of predicted labels $\mathbb{L}_+^{(i)}$. In order to ensure computational tractability, the maximum number of hypotheses should not exceed a certain limit, depending on the used computing platform. Which means the LMB RFS can be converted into an approximate δ-GLMB representation, covering only the prediction hypotheses I_+ with highest weights $w_+^{(i)}(I_+)$. For this approximation, k-shortest path algorithm [Epp98] and hypotheses sampling methods are suggested in [RVVD14]. With the LMB predicted density for each group $\mathcal{G}^{(i)}$ being represented as δ-GLMB distribution, the update step for each group can be performed in parallel considering only the subset $\mathrm{Z}^{(i)}$ relevant to that group, out of the new set of measurements Z. Therefore, equivalent to (5.72), the δ-GLMB posterior for each group $\mathcal{G}^{(i)}$ can be given as

$$\pi^{(i)}(\widetilde{\mathbf{X}}_+^{(i)}|\mathrm{Z}^{(i)}) = \Delta(\widetilde{\mathbf{X}}^{(i)}) \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+^{(i)})\times\Theta_{I_+}^{(i)}} w^{(I_+,\theta)}(\mathrm{Z})\delta_{I_+}(\mathcal{L}(\widetilde{\mathbf{X}}^{(i)})) \left[p^{(\theta))}(.|\mathrm{Z}^{(i)})\right]^{\widetilde{\mathbf{X}}^{(i)}} \tag{5.90}$$

where the posterior weight $w^{(I_+,\theta)}$ for the set of track labels I_+ contained in the group is given by

$$w^{(I_+,\theta)}(\mathrm{Z}^{(i)}) = \frac{\delta_{\theta^{-1}(\{0\cup\mathbb{M}^{(i)}\})}(I_+) w_+^{(i)}(I_+)[\eta_{\mathrm{Z}^{(i)}}^{(\theta)}]^{I_+}}{\sum\limits_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+^{(i)})\times\Theta_{I_+}^{(i)}} \delta_{\theta^{-1}(\{0\cup\mathbb{M}^{(i)}\})}(I_+) w_+^{(i)}(I_+)[\eta_{\mathrm{Z}^{(i)}}^{(\theta)}]^{I_+}}, \tag{5.91}$$

As the association map θ contains measurement to track association only for the subset $\mathrm{Z}^{(i)}$, it is shown in [RVVD14] that the association likelihood $\eta_{\mathrm{Z}^{(i)}}^{\theta}$ and the posterior spatial density $p^{\theta}(x, \ell|\mathrm{Z}^{(i)})$ of the group $\mathcal{G}^{(i)}$ are equivalent to the case in (5.75) and (5.74), respectively. Further, if the number of track labels in I_+ and measurements in $\mathrm{Z}^{(i)}$ are high, the calculation of weights $w^{(I_+,\theta)}(\mathrm{Z}^{(i)})$ of the association hypotheses becomes computationally very expensive. Therefore the usage of Murty's algorithm [Mur68] is suggested in [VV13a] and [RVVD14] in order to truncate the posterior distribution by choosing only the most significant hypotheses. A cost matrix for the track labels in the predicted hypotheses I_+ and the measurements in $\mathrm{Z}^{(i)}$ is constructed based on statistical distance and the weights of only the optimal association hypotheses are further considered.

5.6.3 LMB Approximation and Track Management

The δ-GLMB measurement updated posterior (5.72) needs to be again represented in LMB RFS, so that it can be predicted again in the next time step. The posterior is approximated by the existence and spatial parameters of all the tracks in $\mathbf{X}$, with a matching PHD and mean cardinality [RVVD14]

$$\pi(\mathbf{X}|\mathrm{Z}) = \left\{(r^{(\ell)}, p^{(\ell)})\right\}_{\ell \in \mathbb{L}_+} \tag{5.92}$$

where the posterior existence probability $r^{(\ell)}$ and posterior spatial distribution $p^{(\ell)}$ of each track is given as

$$r^{(\ell)} = \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta_{I_+}} w^{(I_+,\theta)}(\mathrm{Z})1_{I_+}(\ell) \tag{5.93}$$

$$p^{(\ell)}(x) = \frac{1}{r^{(\ell)}} \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta_{I_+}} w^{(I_+,\theta)}(\mathrm{Z})1_{I_+}(\ell)p^{(\theta)}(x,\ell) \tag{5.94}$$

The inclusion function $1_{I_+}(\ell)$ in the above equations infers that only the hypotheses I_+ which contains the track label ℓ in it are considered for calculating the posterior existence and posterior spatial distribution of that track. Therefore, the existence probability of a track $r^{(\ell)}$ is obtained by summing up the weights of all the association hypotheses that contains that track label. In case of GM implementation, the spatial distribution $p^{(\ell)}(x)$ of a track, is then a mixture of Gaussians. In case of gating and grouping of the tracks and measurements, the multi-object posterior density is approximated as a union of the individual posterior densities of each of the group $\mathcal{G}^{(i)} : i = 1, \ldots, N$, and is given as [RVVD14]

$$\pi(\cdot|\mathrm{Z}) \approx \tilde{\pi}(\cdot|\mathrm{Z}) = \bigcup_{i=1}^{N} \left\{(r^{(\ell)}, p^{(\ell)})\right\}_{\ell \in \mathbb{L}_+^i} \tag{5.95}$$

As an additional maintenance of individual tracks, Gaussians with weights lesser than a threshold can be pruned and Gaussians with close mean values can be merged similar to the GM-PHD filter, in order to reduce computational complexity. Two ways of track extractions are proposed in [Reu14]. Tracks can be extracted either based on the maximum a posterior (MAP) estimate of cardinality distribution or by selecting only the tracks which have a probability of existence higher than a certain threshold given by

$$\hat{\mathbf{X}} = \left\{(\hat{x}, \ell) | r^{(\ell)} > e\right\} \tag{5.96}$$

where the threshold e can be parametrised based on the application. By this, the existence probability can also be used for track management in the LMB filter. Tracks can be initialised if their existence probability increases a birth threshold parameter or be deleted if their existence probability falls below the death threshold parameter.

5.6.4 Adaptive Birth Density

The birth distribution needs to be defined for every prediction step in the LMB filter recursion. An intuitive way is to define the spatial birth distribution concentrated at the edges of the sensor FoV, where the objects would appear and disappear. But in case of vehicle perception systems, the environment is highly dynamic and locations of object birth are uncertain. Moreover, a prior defined object birth distribution does not support reinitialisation of lost tracks in other regions of the sensor FoV. In order to address this, an adaptive birth model is suggested in [RVVD14]. The distribution of newly born tracks used for prediction to time instance $k+1$ is based on the measurements $\mathrm{Z_k}$ at time instance k. The adaptive LMB birth distribution for each of the measurements is represented in [RVVD14] as

$$\pi_{B,k+1} = \left\{ r^{(\alpha_i)}_{B,k+1}(z_i), p^{(\alpha_i)}_{B,k+1}(x|z_i) \right\}_{i=1}^{|\mathrm{Z_k}|} \tag{5.97}$$

The birth track is assigned with the label α_i in accordance to the order i of the respective measurement z_i in the measurement set Z_k. The spatial component $p^{\alpha_i}_{B,k+1}(x|z_i)$ of the birth track is initialised by converting the measurement values from measurement space into state space. Generaly speaking, the measurement has a high probability of belonging to a new object if it is spatially far away from any of the existing tracks. So the probability of existence of the birth component $r^{\alpha_i}_{B,k+1}$ should be high if the measurement z_i is spatially well separated from other existing track and should be low in other case. The spatial closeness of the measurement z_i to all the existing tracks can be inferred by summing up weights of all the hypotheses in which that measurement was associated to any of the tracks. This association probability can then be used for initialising the existence probability of the birth track from measurement z_i, where the existence probability should be high if the association probability is high and vice-versa. With λ_B expected number of new objects, maximum existence probability capping parameter $r^{\max}_B$ and the association probability $r_{A,k}(z_i)$, the existence probability of birth component is given as [Reu14]

$$r_{B,k+1}(z_i) = \min \left(r^{\max}_{B,k+1}, \frac{1 - r_{A,k}(z_i)}{\sum\limits_{\xi \in \mathrm{Z}_k} 1 - r_{A,k}(\xi)} \cdot \lambda_B \right) \tag{5.98}$$

5.7 Multiple Model Labeled Multi-Bernoulli Filter

In a typical multi-object tracking application scenario, many classes of objects appear in the sensor's FoV. Especially in urban scenarios, the object can be a pedestrian, bicyclist or even other vehicles. For predicting the motion of the object, using a single motion model for all the scenarios and all the object classes would be a wrong assumption. Pedestrians walking straight along a road for example can be assumed to move with constant velocity, whereas bicyclists and other vehicles can also make turn and accelerate rapidly. Therefore it makes sense to consider more than one motion model for different scenarios and object classes. Classically interacting multiple model (IMM) filters are used for this purpose, where the estimated state is from a mixture of all the motion models. Multiple model versions for the class of RFS filters based on jump Markov system (JMS), specifically for PHD filter is proposed as in [Mah12]. The basic idea in JMS is to append a discrete mode variable $o \in \mathbb{O}$ to the kinematic state x, giving an augmented state of the single-object $\mathbf{x} = (x, o)$, with $x \in \mathbb{X}$. The discrete mode variable represents the different motion models considered for modeling the object motion with $\mathbb{O}$ representing the discrete space of all the possible motion models. The prediction of the augmented state of the previous time step (x', o') to the current time step (x, o) according to the Bayes' rule can be factorised as in [Mah14] by

$$f_+(x, o|x', o') = \chi_{o,o'} \cdot f_+(x|x', o') \tag{5.99}$$

with $f_+(x|x', o')$ representing the Markov transition density corresponding to the mode variable o' and $\chi_{o,o'}$ is the Markov transition matrix for the Markov chain defined by the jump variable $o = 1, \ldots, O$. Consequently, the multi-object state of a JMS is defined in [Mah14] as a finite set of augmented states, given by

$$\mathbf{X} = \{(x^{(1)}, o^{(1)}), \ldots, (x^{(n)}, o^{(n)})\} \subseteq \mathbb{X} \times \mathbb{O} \tag{5.100}$$

A closed form GM solution of MMPHD is presented in [PVTM09], which is shown in [Mah12] to be the only mathematically correct implementation of MMPHD.

In [RSD15] the multiple model labeled multi-Bernoulli (MMLMB) filter is introduced, based again on JMS. It uses the same architecture as the LMB filter, but the spatial distribution of each track is given by joint distribution from all the considered motion modes. The multiple model representation for the labeled RFS case is then given by additionally including the object's label ℓ in the augmented state as [RSD15]

$$\mathbf{X} = \{(x^{(1)}, \ell^{(1)}, o^{(1)}), \ldots, (x^{(n)}, \ell^{(n)}, o^{(n)})\} \subseteq \mathbb{X} \times \mathbb{L} \times \mathbb{O} \tag{5.101}$$

The elements $t_{o,o'}$ of the Markov transition matrix $\underline{\chi}_{o,o'}$ models the transition probability from modes o' to o. In comparison to a single model LMB filter, the difference in the MMLMB filter is that the spatial distribution of each track is given by the joint distribution, augmenting the kinematic state with the motion model as

$$p^{(\ell)}(x, o) = p^{(\ell)}(o)p^{(\ell)}(x|o) \;\; \forall o \in \mathbb{O} \tag{5.102}$$

Consequently, the multiple model representation multi-object probability density of the LMB RFS (5.26) is given in as [RSD15]

$$\pi(\mathbf{X}) = \left\{(r^{(\ell)}, p^{(\ell)}(o)p^{(\ell)}(\cdot|o)\right\}_{\ell \in \mathbb{L}} \tag{5.103}$$

and the birth distribution corresponding to the form in (5.58) as,

$$\pi_B(\mathbf{X}) = \left\{(r_B^{(\ell)}, p_B^{(\ell)}(o)p_B^{(\ell)}(\cdot|o)\right\}_{\ell \in \mathbb{B}} \tag{5.104}$$

where the spatial distribution of the tracks is a joint distribution defined in the space $\mathbb{X} \times \mathbb{O}$ and the labels of surviving and birth tracks are distinct such that the property $\mathbb{L} \cap \mathbb{B} = \emptyset$ holds.

5.7.1 Prediction

The prediction step of MMLMB filter follows the prediction of the single model LMB filter, but the spatial distribution of the tracks is a joint distribution conditioned on all the motion models present in the system, i.e. the objects' state are predicted with all the considered motion models. The predicted density with respect to the prior in (5.103) and birth density in (5.104) is therefore given by

$$\pi_+ = \left\{(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)}(o)p_{+,S}^{(\ell)}(\cdot|o))\right\}_{\ell\in\mathbb{L}} \cup \left\{(r_B^{(\ell)}, p_B^{(\ell)}(o)p_B^{(\ell)}(\cdot|o))\right\}_{\ell\in\mathbb{B}}. \tag{5.105}$$

where the spatial density of the surviving tracks is predicted in line with (5.63) by [RSD15]

$$\begin{aligned} p_{+,S}^{(\ell)}(x,o) &= \frac{\int p_S^{(\ell)}(x')f(x,o|x',o')p^{(\ell)}(x',o')d(x',o')}{\eta_S(\ell)} && (5.106)\\ &= \frac{\sum\limits_{o'\in\mathbb{O}} \int p_S^{(\ell)}(x')t_{o,o'}f(x,o|x',o')p^{(\ell)}(x',o')p^{(\ell)}(o')dx'}{\eta_S(\ell)} && (5.107)\\ &= \sum_{o'\in\mathbb{O}} t_{o,o'}p^{(\ell)}(o')\frac{\int p_S^{(\ell)}(x')f(x|x',o')p^{(\ell)}(x'|o')dx'}{\eta_S(\ell)}, && (5.108) \end{aligned}$$

with the survival probability of the tracks $p_S^{(\ell)}(x',o')$ is assumed to be independent of the current motion model and therefore represented only as $p_S^{(\ell)}(x')$. Further the above equation (5.106) can be factorized into the probability $p_{+,S}^{(\ell)}(o)$ that track ℓ is in mode o and the corresponding predicted spatial distribution $p_{+,S}^{(\ell)}(x|o)$ conditioned on mode o as

$$p_{+,S}^{(\ell)}(o) = \sum_{o'\in\mathbb{O}} f(o|o')p^{(\ell)}(o') \tag{5.109}$$

$$p_{+,S}^{(\ell)}(x|o) = \frac{\int p_S^{(\ell)}(x')f(x|x',o')p^{(\ell)}(x'|o')dx'}{\eta_S(\ell)} \tag{5.110}$$

and the normalisation constant is given as

$$\begin{aligned} \eta_S(\ell) &= \int p_S^{(\ell)}(x')p^{(\ell)}(x',o')d(x',o')\\ &= \sum_{o'\in\mathbb{O}} p^{(\ell)}(o') \int p_S(x')p^{(\ell)}(x',o')dx' \end{aligned} \tag{5.111}$$

Consequently the predicted existence probability of the track ℓ is given as

$$r_{+,S}^{(\ell)} = \eta_S(\ell) \cdot r^{(\ell)} \tag{5.112}$$

A GM implementation of the MMLMB is primarily used in this work. In case of GM implementation, assuming the survival probability p_S of the track to be independent of it's state, the normalisation constant and the corresponding predicted spatial density of a track

ℓ are of the form

$$\begin{aligned}
\eta_S(\ell) &= \sum_{o'\in\mathbb{O}} p^{(\ell)}(o') \int p_S \sum_{j=1}^{J^{(\ell)}(o')} w^{(\ell,j)}(o') \mathcal{N}\left(x; \underline{\mathrm{F}}(o)\hat{x}^{(\ell,j)}, \underline{\mathrm{F}}(o)\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}(o)^\top + \underline{\mathrm{Q}}(o)\right) dx \\
&= \sum_{o'\in\mathbb{O}} p^{(\ell)}(o') p_S \sum_{j=1}^{J^{(\ell)}(o')} w^{(\ell,j)}(o') \int \mathcal{N}\left(x; \underline{\mathrm{F}}(o)\hat{x}^{(\ell,j)}, \underline{\mathrm{F}}(o)\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}(o)^\top + \underline{\mathrm{Q}}(o)\right) dx \\
&= p_S \sum_{o'\in\mathbb{O}} p^{(\ell)}(o') \sum_{j=1}^{J^{(\ell)}(o')} w^{(\ell,j)}(o') \int \mathcal{N}\left(x; \underline{\mathrm{F}}(o)\hat{x}^{(\ell,j)}, \underline{\mathrm{F}}(o)\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}(o)^\top + \underline{\mathrm{Q}}(o)\right) dx \\
&= p_S
\end{aligned} \tag{5.113}$$

$$\begin{aligned}
p^{(\ell)}_{+,S}(x,o) &= \frac{\sum\limits_{o'\in\mathbb{O}} t_{o,o'} p^{(\ell)}(o') p_S \sum\limits_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \underline{\mathrm{F}}(o)\hat{x}^{(\ell,j)}, \underline{\mathrm{F}}(o)\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}(o)^\top + \underline{\mathrm{Q}}(o)\right)}{p_S} \\
&= \frac{p_S \sum\limits_{o'\in\mathbb{O}} t_{o,o'} p^{(\ell)}(o') \sum\limits_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \underline{\mathrm{F}}(o)\hat{x}^{(\ell,j)}, \underline{\mathrm{F}}(o)\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}(o)^\top + \underline{\mathrm{Q}}(o)\right)}{p_S} \\
&= \sum_{o'\in\mathbb{O}} t_{o,o'} p^{(\ell)}(o') \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \underline{\mathrm{F}}(o)\hat{x}^{(\ell,j)}, \underline{\mathrm{F}}(o)\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}(o)^\top + \underline{\mathrm{Q}}(o)\right)
\end{aligned} \tag{5.114}$$

which can then be factorised into two parts according to (5.109) and (5.110) as

$$p^{(\ell)}_{+,S}(o) = \sum_{o'\in\mathbb{O}} t_{o,o'} p^{(\ell)}(o') \tag{5.115}$$

$$p^{(\ell)}_{+,S}(x|o) = \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \underline{\mathrm{F}}(o)\hat{x}^{(\ell,j)}, \underline{\mathrm{F}}(o)\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}(o)^\top + \underline{\mathrm{Q}}(o)\right) \tag{5.116}$$

5.7.2 Update

The conversion of the LMB RFS into δ-GLMB representation and the update procedure is same as that of the LMB filter explained in the previous section. The updated multiple model multi-object posterior is [RSD15]

$$\pi(\cdot|\mathrm{Z}) = \left\{(r^{(\ell)}, p^{(\ell)}(o) p^{(\ell)}(\cdot|o)\right\}_{\ell\in\mathbb{L}} \tag{5.117}$$

with the parameters

$$r^{(\ell)} = \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta_{I_+}} w^{(I_+,\theta)}(\mathrm{Z}) 1_{I_+}(\ell) \tag{5.118}$$

$$p^{(\ell)}(x|o) = \frac{1}{r^{(\ell)}} \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta_{I_+}} w^{(I_+,\theta)}(\mathrm{Z}) 1_{I_+}(\ell) p^{(\ell,\theta)}(x|o) \tag{5.119}$$

$$p^{(\ell)}(o) = \frac{1}{r^{(\ell)}} \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta_{I_+}} w^{(I_+,\theta)}(\mathrm{Z}) 1_{I_+}(\ell) p^{(\ell,\theta)}(o). \tag{5.120}$$

The updated spatial distribution and measurement likelihood are similar to that of the LMB filter (5.74) and (5.75), but are conditioned on the mode o [RSD15]

$$p^{(\ell,\theta)}(x|o) = \frac{p_+^{(\ell)}(x|o) \cdot \psi_{\mathrm{Z}}(x,\ell;\theta)}{\eta_{\mathrm{Z}}^{(\theta)}(\ell|o)} \tag{5.121}$$

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell|o) = \int p_+^{(\ell)}(x|o) \cdot \psi_{\mathrm{Z}}(x,\ell;\theta)dx \tag{5.122}$$

The likelihood ψ_{Z} of a measurement to track is same as that in (5.76). The updated probability of the track ℓ being in mode o is derived in [RSD15] by marginalizing state x out of the updated joint probability density $p^{(\ell,\theta)}(x,o)$

$$\begin{aligned} p^{(\ell,\theta)}(o) &= \frac{\int p_+^{(\ell)}(x|o)p_+^{(\ell)}(o) \cdot \psi_{\mathrm{Z}}(x,\ell;\theta)dx}{\eta_{\mathrm{Z}}^{(\theta)}(\ell)} && (5.123)\\ &= \frac{p_+^{(\ell)}(o)\eta_{\mathrm{Z}}^{(\theta)}(\ell|o)}{\eta_{\mathrm{Z}}^{(\theta)}(\ell)} && (5.124) \end{aligned}$$

The normalisation constant required in the above equation, for the probabilities of individual motion modes to sum up to one is given as

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \sum_{o\in\mathbb{O}} \eta_{\mathrm{Z}}^{(\theta)}(\ell|o)p_+^{\ell}(o) \tag{5.125}$$

The updated component weights are then given similar to the form in (5.73), but using the normalisation constant in the above equation (5.125).

In case of GM implementation assuming state independent detection probability p_D, the update likelihood of the measurement $z_\theta(\ell)$ for the track ℓ conditioned on mode o, in case the object being detected ($\theta(\ell) > 0$) is of the form

$$\begin{aligned} \eta_{\mathrm{Z}}^{(\theta)}(\ell|o) &= \int \frac{p_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}(o)} w_+^{(\ell,j)}(o)\mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right)\mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right)dx\\ &= \frac{p_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}(o)} w_+^{(\ell,j)}(o)\mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right)\int \mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right)dx && (5.126)\\ &= \frac{p_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}(o)} w_+^{(\ell,j)}(o)\mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right), \end{aligned}$$

where the innovation of the predicted Gaussian distribution conditioned on mode o is given according to the Kalman filter equations as

$$\begin{aligned} z_+^{(\ell,j)} &= \mathrm{H}(o)\hat{x}_+^{(\ell,j)}, && (5.127)\\ \underline{\mathrm{S}}^{(\ell,j)} &= \underline{\mathrm{H}}(o)\underline{\mathrm{P}}_+^{(\ell,j)}\underline{\mathrm{H}}^\top(o) + \underline{\mathrm{R}}, && (5.128)\\ \underline{\mathrm{K}}^{(\ell,j)} &= \underline{\mathrm{P}}_+^{(\ell,j)}\underline{\mathrm{H}}^\top(o)[\underline{\mathrm{S}}^{(\ell,j)}]^{-1}, && (5.129)\\ \hat{x}^{(\ell,j,\theta)}(\mathrm{Z}) &= \hat{x}_+^{(\ell,j)} + \underline{\mathrm{K}}^{(\ell,j)}(z_{\theta(\ell)} - z_+^{(\ell,j)}), && (5.130) \end{aligned}$$

$$\underline{\mathrm{P}}^{(\ell,j)} = \underline{\mathrm{P}}_{+}^{(\ell,j)} - \underline{\mathrm{K}}^{(\ell,j)}\underline{\mathrm{S}}^{(\ell,j)}[\underline{\mathrm{K}}^{(\ell,j)}]^{\top}. \tag{5.131}$$

The updated posterior spatial density of the track ℓ conditioned on mode o is then given for the case $\theta(\ell) > 0$ as,

$$\begin{aligned}
p^{(\ell,\theta)}(x|o) &= \frac{\frac{p_D}{\kappa(z_{\theta(\ell)})}\sum_{j=1}^{J_{+}^{(\ell)}(o)} w_{+}^{(\ell,j)}(o)\mathcal{N}\left(z_{\theta(\ell)}; z_{+}^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right)\mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right)}{\frac{p_D}{\kappa(z_{\theta(\ell)})}\sum_{j=1}^{J_{+}^{(\ell)}(o)} w_{+}^{(\ell,j)}(o)\mathcal{N}\left(z_{\theta(\ell)}; z_{+}^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right)} && (5.132)\\
&= \sum_{j=1}^{J_{+}^{(\ell)}(o)} w_{Z}^{(\ell,j,\theta)}(o)\mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right) && (5.133)
\end{aligned}$$

with the posterior weights of the Gaussian components given as

$$w_{Z}^{(\ell,j,\theta)}(o) = \frac{\frac{p_D}{\kappa(z_{\theta(\ell)})} w_{+}^{(\ell,j)}(o)\mathcal{N}\left(z_{\theta(\ell)}; z_{+}^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right)}{\eta_{\mathrm{Z}}^{(\theta)}(\ell|o)} \tag{5.134}$$

In case the object is not detected ($\theta(\ell) = 0$), the likelihood is of the form

$$\begin{aligned}
\eta_{\mathrm{Z}}^{(\theta)}(\ell|o) &= \int (1 - p_D) \sum_{j=1}^{J_{+}^{(\ell)}(o)} w_{+}^{(\ell,j)}(o)\mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right) dx && (5.135)\\
&= (1 - p_D) \sum_{j=1}^{J_{+}^{(\ell)}(o)} w_{+}^{(\ell,j)}(o) \int \mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right) dx && (5.136)\\
&= (1 - p_D) \sum_{j=1}^{J_{+}^{(\ell)}(o)} w_{+}^{(\ell,j)}(o) && (5.137)\\
&= (1 - p_D), && (5.138)
\end{aligned}$$

and the corresponding posterior spatial density is same as the prediction, given as

$$p^{(\ell,\theta)}(x|o) = \sum_{j=1}^{J_{+}^{(\ell)}(o)} w^{(\ell,j,\theta)}(o)\mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right) \tag{5.139}$$

where

$$\begin{aligned}
w^{(\ell,j,\theta)}(o) &= w_{+}^{(\ell,j)}, && (5.140)\\
\hat{x}^{(\ell,j,\theta)} &= \hat{x}_{+}^{(\ell,j)}, && (5.141)\\
\underline{\mathrm{P}}^{(\ell,j)} &= \underline{\mathrm{P}}_{+}^{(\ell,j)}. && (5.142)
\end{aligned}$$

From the relations (5.125), (5.126) and (5.138), the normalisation constant considering the measurement likelihood for track ℓ averaged over all the motion modes $o \in \mathbb{O}$ is given as

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \begin{cases} \sum\limits_{o \in \mathbb{O}} p_{+}^{(\ell)}(o) \frac{p_D}{\kappa(z_{\theta(\ell)})} \sum\limits_{j=1}^{J_{+}^{(\ell)}(o)} w_{+}^{(\ell,j)}(o) \mathcal{N}\left(z_{\theta(\ell)}; z_{+}^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right), & if\ \theta(\ell) > 0 \\ (1 - p_D), & if\ \theta(\ell) = 0, \end{cases} \tag{5.143}$$

which is then used for calculating the updated weight $w^{I+,\theta}(\mathrm{Z})$ of the hypothesis component used in (5.118)-(5.120), and to represent the joint probability density $p^{(\theta)}(x, \ell|\mathrm{Z})$ of the track ℓ, considering all the motion models.

5.8 Integrating Class Information into MMLMB Filter

There exists a number of state-of-art methods to derive or predict the class information of an object, depending on the sensor used. An object's class can be extracted based on the features of the measurements they generate or their states or a combination of both. For example, a video sensor can extract the edge and contour features of the object from the image, a laser scanner provides a point cloud which can be analysed to derive the class information based on extracted shapes or a radar sensor measures the frequency pattern of an object which can be used to differentiate the object classes. The basis of object state estimation is an assumption that the object follows a particular motion model, like Constant Velocity (CV), Constant Acceleration (CA) or Constant Turn Rate and Velocity (CTRV) models, to name a few. But not all objects have the same motion behaviour. As described in section 5.7, the motion characteristics vary depending on the class of object. The motion of a passenger car is different from that of a pedestrian. Therefore in multi-object tracking it is beneficial to know the class of the object in order to use an appropriate motion model, to predict it's motion.

The idea of integrating class information into the object state estimation has been proposed widely, for example in [GW12, Mei15, Mun11]. In [GW12] a classification aided CPHD is introduced, where the class information is used for data association in the update step of the filter. The mode variable in the likelihood function is replaced with a class variable and the confusion matrix from the classifier models the class probabilities. In [Mun11], the JIPDA filter is combined with Dempster-Shafer Theory (DST) of evidence, in order to predict and update the existence probability of an object along with it's state. The considered class of objects in [Mun11] are rather categorized as "relevant" or "not-relevant" objects for the application, although not the actual type of the object. Similarly, [Mei15] proposes the GM-CMMPHD filter, where each Gaussian mixture component (GMC) is augmented with an object class information derived from a laser scanner measurements based classifier and DST. The object classes are categorized as pedestrian, bicyclist, car and truck. The prior class information is used together with the object state estimation step, for adapting the transition probabilities of the object's motion models. The predicted class information is then updated using both the measurement based features and state of the object, following the update step. The class information is used only in parallel to the filter itself and doesn't affect the state estimation procedure [MRSD14, Mei15].

A similar approach is proposed in this work, but for the class of MMLMB filter explained in section 5.7. In GM-MMLMB filter, the distribution of each track is represented by a mixture of many Gaussian components, which is based on the consideration of different cardinality, different motion models as well as different measurement to track association hypotheses. The updated state of the object as well as it's existence probability are extracted from the parameters of multi-Bernoulli distribution. In the following sections, an introduction to the DST of evidence and a method to integrate the class information of the object into the GM-MMLMB filter using DST, in parallel to the state prediction and update steps, is described. As in [MRSD14], the class information is used, for adapting the transition probabilities of the considered motion models as well as to classify the object based both on it's state and measurement features.

5.8.1 Dempster-Shafer Theory of Evidence

The Dempster-Shafer-Evidence Theory (DST) is generalised probability theory, which defines the occurrence of an event as combination of evidences from many sources, rather than individual probability constituents. The DST provides a framework for reasoning uncertainty with the help of belief functions. The set of all the states or hypotheses under consideration is represented by a finite set of n elements, called the frame of discernment (FOD)

$$\Omega = \{e_1, e_2, \dots, e_n\} \tag{5.144}$$

The occurrence of an event is then calculated from the power set $\mathbb{P}(\Omega)$ of the order 2^{Ω}, consisting all the elements and subsets of the elements in FOD, including the null-set. A belief for each element in the power set, expressing the evidence from different sources, is given as a basic belief mass (BBM) $m(E_i) \geq 0$. Every $E \in \Omega$ with $m(E) > 0$ is called the focal hypotheses. The masses of all the subsets of power set equals to 1 and is defined by the basic belief assignment (BBA) function given as:

$$\begin{aligned} m(\emptyset) &= 0 \\ \sum_{E \in \mathbb{P}(\Omega)} m(E) &= 1 \end{aligned} \tag{5.145}$$

Unlike the probability theory based approaches where the masses can be given only to the elements of Ω, in DST the masses can be given to any subsets of Ω. Moreover, the BBA in (5.145) can support a set E, without supporting any of it's subsets, which represents the limited knowledge capacity of DST. The mass function $m(E)$ is called a categorical mass function in case there is only one focal set, such that $m(E) = 1$ with $E \subseteq \Omega$. If $E = \Omega$ in the above case, then the condition $m(\Omega)$ represents the state of total ignorance, which means none of the individual subsets of Ω are supported with an evidence.

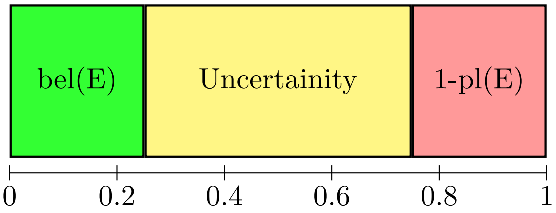


Figure 5.6: Visualisation of belief, uncertainty and plausibility of a BBA (cf. [Mun11])

The degree of belief $bel(E)$ is defined in the transferable belief model (TBM) in [SK94] as the lower bound of the probability interval, which is the sum of all masses that support E

$$\begin{aligned} bel &: 2^{\Omega} \mapsto [0,1] \\ bel(E) &= \sum_{\emptyset \neq X \subseteq E} m(X) \end{aligned} \tag{5.146}$$

Moreover, a concept on reliability of an evidence is proposed, in which the masses for evidences from less reliable sources are multiplied with a discounting factor, to represent a lesser weight for those unreliable sources. The degree of plausibility $pl(E)$ is given in [SK94] as the total amount of belief that supports E

$$\begin{aligned} pl &: 2^{\Omega} \mapsto [0,1] \\ pl(E) &= \sum_{X \cap E \neq \emptyset} m(X) \end{aligned} \tag{5.147}$$

The degree of plausibility represents the upper bound of the uncertainty and therefore it is given as

$$bel(E) \leq u(E) \leq pl(E) \tag{5.148}$$

Demspter's rule of combination provides a basis for the combining the masses of different pieces of evidence. The combination is given as:

$$(m_1 \oplus m_2)(E) = \begin{cases} 0 & \text{if } E = \emptyset \\ \dfrac{\sum\limits_{X \cap Y = E} m(X)m(Y)}{1 - \sum\limits_{X \cap Y = \emptyset} m(X)m(Y)} \; \forall E \in 2^{\Omega} & \text{if } E \neq \emptyset \end{cases} \tag{5.149}$$

The method of discounting in the TBM enables to model the reliability of BBA from a particular source. If the belief masses in BBA are considered to be coming from imprecise source, they should be used only with a certain degree of confidence. Consequently, discounting of a BBA with a parameter d depending on it's source can be given as

$$m^d(E) = \begin{cases} d \cdot m(E), & \text{if} E \neq \emptyset \\ 1 - d + d \cdot m(E), & \text{if} E = \Omega. \end{cases} \tag{5.150}$$

In the TBM in [SK94], two levels are defined as the credal level and the pignistic level. The formulation of belief functions and combination of evidences from many sources are all said to be in the so called credal level. Then the actual decisions based on the belief functions in the credal level are made in the so called pignistic level. Further, the pignistic transformation is given as:

$$BetP_m(E) = \sum_{B \subseteq \Omega, B \neq \emptyset} \frac{|E \cap B|}{|B|} m(B) \tag{5.151}$$

where only the BBMs $m(B)$ that do not contradict E are considered for calculating $BetP_m(E)$.

5.8.2 GM-MMLMB Filter with Classification

Every track in the GM-MMLMB filter realisation described in section 5.7 is augmented with the motion mode o along with it's label ℓ and state x. Moreover, in the GM implementation, the density of each track is represented by a mixture of Gaussians. With the aim of integrating object class information into the GM-MMLMB filter, let each track be additionally augmented with the class information of the track. Thereby, a track ℓ which has state x, motion mode o and class-BBA $m_C^{(\ell)}$, can be represented as $\left(x, \ell, o, m_C^{(\ell)}\right)$ and the i^{th} Gaussian mixture component (GMC) of it as

$$\left(w^{(i)}(o), \mathcal{N}(x; \hat{x}^{(i)}, \underline{\mathrm{P}}^{(i)}), m_C^{(\ell)}\right) \tag{5.152}$$

where it holds the class-BBA of it's parent track. The class-BBA $m_C^{(\ell)}$ of a track represents the certainty values of different object classes that are in question. The focal elements of the FOD Ω used in the DST are then the considered object classes. Therefore, the surviving tracks and the birth tracks contain their class-BBA along with their joint probability density. Figure 5.7 illustrates the method for integrating the class information alongside the multi-object filter recursion with the GM-MMLMB filter. Following [Mun11] and [Mei15], the prediction and update steps are divided into state and class levels, with interactions in between both the levels. The class-BBA information from time k is used by the state level for adapting parameters of multiple model object state prediction into time $k+1$ and the measurement updated object state information is used for the posterior class-BBA at time $k+1$. Consequently, the interactions denotes that the class-BBA is used for two purposes as in [Mei15]: to adapt the mode transition probabilities and to classify the object based on both measurement and track features.

Prediction

In the track prediction step of the GM-MMLMB filter, mean and covariance of each of the track's GMC is predicted, for each of the considered motion mode $o \in \mathbb{O}$, thereby new GMCs are generated. The weight of the predicted GMC is obtained by multiplying the weight of it's parent GMC with the transition and model probabilities of the parent track. The state prediction step follows the prediction equations of the GM-MMLMB filter, as in (5.114). The difference here is, instead of a fixed value, the model transition probabilities $t_{o,o'}$ used for the track prediction in GM-MMLMB filter are adapted in every time step, according to the posterior class-BBA of the track. Therefore, the class-BBA $m_C^{(\ell)}$ of the track should represent the likelihood of a motion model $o_i \in \mathbb{O}$, that is more suitable for the class of the object. In order to obtain the probability for each of the classes in the FOD Ω, the class-BBA needs to be transformed from it's credal level to the pignistic level as in (5.151). As in case of the general Markov transition matrix, with each row covering all the motion models, the transformed probabilities in each row must sum to one. The Markov transition matrix $\underline{\chi}_{o,o'}$ used for track ℓ is therefore recalculated with each update and is given as

$$\underline{\chi}_{o,o'}^{(\ell)} = \begin{bmatrix} BetP_{m_C^{(\ell)}}(E_{o_1}^{(\ell)}) & \dots & BetP_{m_C^{(\ell)}}(E_{o_N}^{(\ell)}) \\ \vdots & \ddots & \vdots \\ BetP_{m_C^{(\ell)}}(E_{o_1}^{(\ell)}) & \dots & BetP_{m_C^{(\ell)}}(E_{o_N}^{(\ell)}) \end{bmatrix} \tag{5.153}$$

As illustrated in [Mei15], the class hypotheses $E_{o_i}^{(\ell)}$ in the above transition probability matrix should cover atleast one of the object classes which follow the same motion model $o_i(i \in \{1, \dots, N_o\})$ and all the considered classes need to be covered by the hypotheses $\left\{E_{o_i}^{(\ell)}, \dots, E_{o_j}^{(\ell)}\right\}$ with $E_{o_i}^{(\ell)} \cap E_{o_j}^{(\ell)} = \emptyset$ for $i \neq j$.

The class-BBA of the track on the other side is predicted from time k to $k+1$ independent of the state of the track, by discounting with a parameter d as

$$m_{C_+}^{(\ell)}(E) = \left(m_C^{(\ell)}(E)\right)^d \tag{5.154}$$

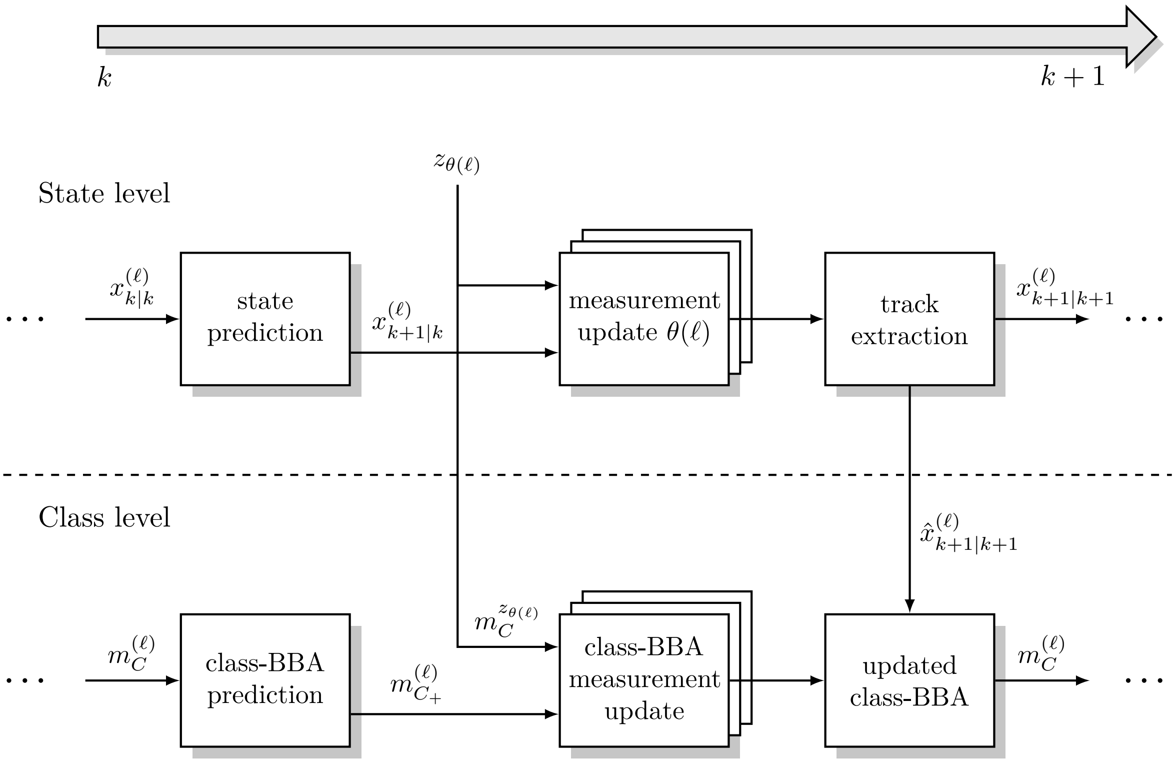


FIGURE 5.7: Integrating class information into tracking framework. The state of the track is predicted and updated with GM-MMLMB filter equations on the upper state level. The class-BBA is predicted and updated in the lower class level. Object class is then derived by fusing BBAs from both the levels.

Update

As described in section 5.7.2, the update step of the GM-MMLMB filter considers the various measurement to track association hypotheses, where an hypothesis is given by the mapping $\theta : I_+ \mapsto \{0, 1, \dots, |\mathrm{Z}|\}$, with I_+ representing a predicted set of track labels and Z the measurement set. The update of a track ℓ therefore involves innovations with all the measurements available at time $k+1$. The likelihoods of assignment of a measurement $z_{\theta(\ell)}$ to a track ℓ (given by the case $\theta(\ell) > 0$) and also association of the track to a missed detection (given by the case $\theta(\ell) = 0$) are considered. For track innovation, each measurement along with it's Gaussian distribution also contains the measurement based class-BBA $m_C^{z_{\theta(\ell)}}$ of the object, delivered by the classifier which can be given as

$$z^{(i)} = \left(\mathcal{N}(z; z_+^{(i)}, \underline{\mathrm{R}}^{(i)}), m_C^{z^{(i)}}\right),\ i \in \{1, \dots, |Z|\} \tag{5.155}$$

On the state level, the measurement $z_{\theta(\ell)}$ updates the spatial distribution of a track ℓ for an association $\theta(\ell)$ by innovation of each of the predicted GMCs of that track. The equations used for state and weight update of GMCs are same as (5.132) and (5.134). Additionally, on the class level, the predicted class-BBA $m_{C_+}^{(\ell)}(E)$ is updated by the measurement based class-BBA for an association $\theta(\ell)$ according to the DST rule of combination in (5.149) as

$$\begin{aligned} m_C^{(\ell,\theta)}(E) &= m_{C_+}^{(\ell)} \oplus m_C^{z_{\theta(\ell)},\alpha} \\ &= \frac{\sum\limits_{S_+ \cap S_z = E} m_{C_+}^{(\ell)}(S_+) m_C^{z_{\theta(\ell)},\alpha}(S_z)}{1 - \sum\limits_{S_+ \cap S_z = \emptyset} m_{C_+}^{(\ell)}(S_+) m_C^{z_{\theta(\ell)},\alpha}(S_z)}\ \forall E \in 2^{\Omega} \end{aligned} \tag{5.156}$$

The measurement based class-BBA is discounted with a factor α before being fused with the predicted class-BBA of the track. This discounting facilitates the consideration of confidence in classification output by the classifier. Each update hypothesis therefore has a corresponding updated class-BBA for the track. Further, the weights of the hypotheses $w^{(I_+,\theta)}(\mathrm{Z})$ are updated according to update step of GM-MMLMB described in section 5.7.2. On the state level, the posterior existence probability, posterior spatial density and model probabilities of a track are calculated according to the equations (5.118) to (5.120), which means the posterior of a track is from various hypotheses which include that track in them. On the class level, the class-BBA of a track ℓ is obtained by combining the updated class-BBA from the hypotheses that contains that track. Before combining, the class-BBA from an hypothesis is discounted with the normalised updated weight of that hypothesis and the combination of BBAs is given as

$$m_C^{(\ell)} = \left(m_C^{(\ell,\theta)}\right)^{\widetilde{w}_1^{(I_+,\theta)}(\mathrm{Z})} \oplus \left(m_C^{(\ell,\theta)}\right)^{\widetilde{w}_2^{(I_+,\theta)}(\mathrm{Z})} \cdots \oplus \left(m_C^{(\ell,\theta)}\right)^{\widetilde{w}_N^{(I_+,\theta)}(\mathrm{Z})}, \tag{5.157}$$

where $(I_+, \theta) \in \mathcal{F}(\mathbb{L}_+) \times \Theta_{I_+}$and $\ell \in I_+$. The track extraction step follows that of the standard LMB filter family, based on MAP estimate of cardinality distribution or track existence probabilities. The state estimates of a track can be approximated by the mean of it's GMC which has the highest weight. Therefore, this estimated state of the track from the state level can also be used to further update the class-BBA of the track, apart from the update by measurement based class-BBA. One of the track features which can be used to classify the object is it's velocity. The class-BBA can then be constructed based on the velocity of the track and be fused in the same way as the measurement based class-BBA in order to get a posterior class-BBA of the track on the class level.

The updated class-BBA from the above steps is further used for estimating the class of the track. For this the class-BBA of the track needs to be transformed to the pignistic level according to (5.151). The object class and the corresponding class probability can then be derived as

$$C^{(\ell)} = \max_{e=\Omega} BetP_{m_C^{(\ell)}}(e) \tag{5.158}$$

$$p_C^{(\ell)} = BetP_{m_C^{(\ell)}}(C^{(\ell)}). \tag{5.159}$$

Chapter 6

Radar based Feature Extraction and Classification

High-frequency radars in the frequency range 77-81 GHz are recently developed for usage in many automotive applications. With an increased resolution, multiple reflection points are detected from an extended object. Therefore the usual assumption of an object as point target is not valid anymore and the spatial extent of the object also needs to be considered. Many methods have been developed for considering the multiple reflections of an object. The basic idea is to segment the detection points as sets of clusters, based on heuristics exploring the similarity between the detection points. Each cluster is then believed to belong to a particular object. Consequently, each cluster contains information regarding the class of the object persistent to that particular cluster.

Every object has it's own distinct features. For example, a pedestrian walks with a velocity which is very distinct from the velocity of a car, or a dimensions of a cyclist are narrower compared to a truck. Extracting such features out of the measurement clusters of an object would help to classify that object. Vast number of methods have already been proposed for detecting and classifying object based on range sensors. Objects can be detected and classified based on shape models [Stü06, Mun11], measurement characteristics [SKFM14, Fö06, PLN09] or formal hypothesis [ZZC$^+$09]. Distance based clustering of detections and fitting one of the pre-defined shape models to the cluster to infer the object type is proposed in [Kä07]. In [ZSKS06] a method for joint object tracking and classification based on laser scanner is proposed, where the defined object classes are 0-axis, 1-axis or 2-axis objects. A Markov chain based on different observation positions of the object is constructed. The transition probabilities are defined based on the object maneuver and the corresponding observation of the sensor. In [PLN09], a method for segmentation of measurements from laser scanner and consecutive feature extraction from these segments is proposed. The extracted features are then used for training different machine learning based classifiers to detect pedestrians.

In this work, density based spatial clustering of applications with noise (DBSCAN) [EKSX96] algorithm is used for segmenting high-resolution radar detections into distinct clusters. The radar sensor setup in the ego-vehicle and it's measurement properties are as described in Chapter 4. Methods for extracting related features from clusters, are explained in section 6.2. Object classification is achieved by training a support vector machine (SVM) with the defined features of different moving object classes - pedestrians, cyclists and other vehicles. Libsvm [CL13] is used for implementation of the SVM classifier. The object class information based on the estimated object velocity is fused with the measurement based classification output of the SVM. For this, DST as explained in section 5.8.1 is used. Consequently the tracker's frame of discernment (FOD) according to the considered classes can be given as

$$\Omega = \{P, B, C\} \tag{6.1}$$

where P denotes pedestrian, B bicyclist and C passenger car.

6.1 Segmentation

The aim of segmentation is to divide the raw radar reflections into distinct groups. Each group or cluster can then be represented by a reference point and would have certain properties relevant to the object they belong to. The following subsections describe the procedures used for segmenting the radar detections.

6.1.1 Pre-processing

The radar sensor outputs detection points from the objects after internal signal processing. Each detection point has a range, azimuth and Doppler velocity values extracted according to the FMCW detection principle presented in section 2.2.3.2. As multiple detection points can originate from an extended object, the segmentation algorithm tries to group the detection points as clusters. Ideally each cluster should contain all the detections belonging to a single object. A typical problem with radar sensor is the so called double reflection, where the electro-magnetic waves from the radar reflected by the object surface is strong enough to be reflected back again by the radar mounting or the ego-vehicle surface. These reflections would create the notion of ghost objects by causing a peak in the FFT, if their amplitude are high enough. Before clustering, the detection points due to double reflections from the same object need to filtered out, in order to avoid false objects. The azimuth measured in the case of double reflection would be nearly same, but the range would have values in multiples of two

$$\begin{aligned} \theta_{double} &\approx \theta_{single} \\ r_{double} &= 2 \cdot r_{single} \end{aligned} \tag{6.2}$$

Additionally, detection points which have an unrealistic Doppler velocity values are filtered out before clustering. These are typically clutter measurements which is prevalent with radar sensors.

6.1.2 DBSCAN Clustering

After the pre-processing of reflections, the remaining detection points can be understood as belonging to the objects in the environment, either pedestrians, bicyclists, vehicles or other infrastructures. The basic concept of the DBSCAN algorithm is that, any point in a cluster should contain a certain number of other points within a particular defined radius. Which means, a group of points make a cluster if their density is higher than particular threshold. The distance between two points p and q can basically be described by any distance function $dist(p, q)$. Euclidean distance function is used in this work. Basic definitions are made in [EKSX96] which defines *Eps*, *MinPts*, *core point* and *border points.*

Eps is defined as the threshold radius which represents the neighbourhood of a point. The *Eps-neighbourhood* of a point p denoted as $\mathrm{N}_{Eps}(p)$ is defined by $\mathrm{N}_{Eps}(p) = [q \in D|dist(p,q) \leq Eps]$. Further as illustrated in the Figure 6.1, the point p is a *core point* if it has atleast *MinPts* number of points in it's neighbourhood and also those neighbouring points are *directly density-reachable* from p. If there is a point q_1 within the Eps-neighbourhood of p but do not have *MinPts* number of points that are directly density-reachable from it, then q_1 would be a border point. q_1 is directly density-reachable from p but not the other way, as directly density-reachable is defined to be symmetric only for the connections between the core points. Considering another point r in the figure (dbscan), q_1 is said to be *density-reachable* from r through the path of points $p_1, \ldots p_n$, with $p_1 = r$, $p_n = q_1$, and rest of the points p_{i+1} also being core points. The definition of *density-connected* connects two border points q_2 and q_1 through the point r as they are both density-reachable from the core point r. Now starting from point p, including all the core points that directly density-reachable from p and the border points q_1 and q_2 that are density-connected between them and density-reachable from p form a cluster.

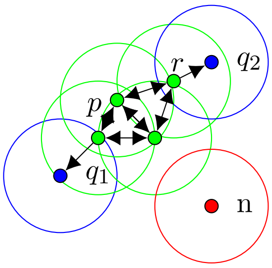


FIGURE 6.1: Illustration of the DBSCAN algorithm

The parameters *Eps* and *MinPts* need to be selected based on the application. The *k-dist* graph method is proposed in [EKSX96] for the selection of these parameters from the thinnest cluster of points. However, the suggestion is mainly for unknown situations. In our case, the objects in the environments are more or less known, which of all the pedestrian class has the least spatial extension. Another factor in selecting the parameter is that, the density and number of points reflected from an object depends strongly on the position of the object with respect to sensor. When the object is closer to the radar sensor it generates more reflection points, whereas when it is detected at the farthest position in the FoV, generates only a point. From many measurements it was noted that all the object classes generated as less as one detection point at the edge of FoV. At closer positions to the sensor, pedestrians generated upto six detections, bicyclists upto seven detection points and a passenger car upto twenty detection points. Intuitively, for urban scenario safety systems, the area very close to the ego-vehicle is of high relevance and the clustering in that area needs to be reliable, than at a higher distance. The range and azimuth resolution of the radar sensor used for this work is 0.028 m and 1°respectively. Considering these, *Eps* is chosen to be 0.05 m and *MinPts* as 3. However, single reflection points are not rejected after cluster formation and are still considered to cover the case when an object is at a higher distance from the sensor. A snapshot of the scenario for testing detections clustering is shown in Figure 6.2. In the shown test scenario, a bicyclist moves parallel to the ego-vehicle. The sequence is recorded in the yard, where a large wall and metal doors are also detected and grouped as distinct clusters.

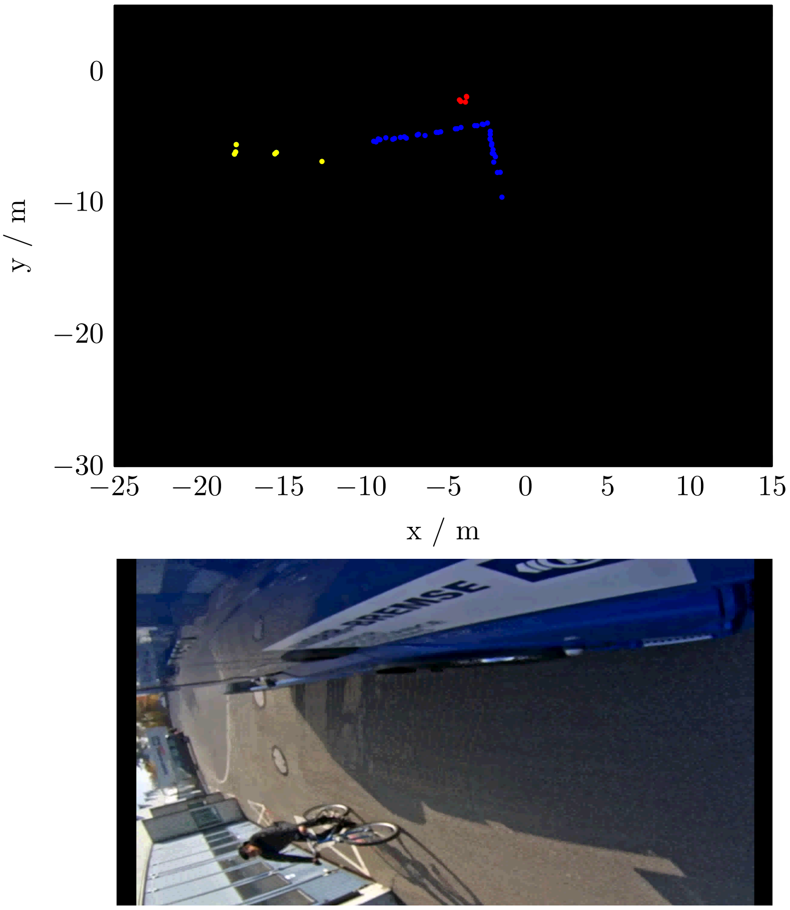


FIGURE 6.2: Snapshot of a test sequence showing the results of detections clustering with DBSCAN. In the top Figure the cluster detections belonging to bicyclist are indicated by red markers, cluster detections belonging to corner of building are indicated in blue markers and unclustered individual detections in yellow. Bottom Figure is the corresponding web-cam snapshot.

Although the radar delivers only lesser detection points compared to other high resolution sensors like laser scanner, the results show that the clustering is still feasible with effective heuristics. Distinction of objects when they are too close to each other is possible only to a certain extent using a single radar sensor. Also, when a poor reflecting class like pedestrian is very close to other object class with high reflectivity like a passenger car, it may happen that the interference is high and the pedestrian may not be detected atall. In addition to the radar sensor, using multiple sensors like camera or laser scanner and fusion of detections from these multiple sensors would improve the object distinction. However, these effects are not in the focus of the work and is recommended for improvement in the future.

6.2 Feature Extraction

The output of segmentation is well defined distinct clusters, each representing a distinct object. The feature extraction step searches for defined object hypotheses in each segment. The first hypothesis is made regarding the shape of the object. Pedestrians are considered in this work to have a circular extension, bicyclists stick extension and other vehicles as rectangular boxes. Therefore three shape model assumptions are made with L-shape representing two sides of vehicle, I-shape representing one side of vehicle and bicyclist and O-shape representing pedestrians. Similar kind of approach is used in many of the earlier works [Kä07, Pie15]. The parameters of the shape feature are then extracted by fitting the detection points in each cluster to one of the above defined models sequentially.

6.2.1 Model Pre-selection

In [Kä07] a method to reduce the computational effort of model fitting is proposed and the same is used here. Principal component analysis (PCA) is applied on every segment. It is to check whether the points are scattered only in one direction or both x- and y- axis directions. The covariance Cov of the Cartesian x- and y- coordinates of N radar detection points in that segment are calculated according to

$$Cov = \frac{1}{N-1} \sum_{i=1}^{N} \begin{bmatrix} x_i - \bar{x} \\ y_i - \bar{y} \end{bmatrix} \begin{bmatrix} x_i - \bar{x} \; y_i - \bar{y} \end{bmatrix} \tag{6.3}$$

The ratio of eigen values $\frac{\lambda_1}{\lambda_2}$ of the covariance matrix Cov, where $\lambda_1 \geq \lambda_2$ denotes the spread of the data in x- and y- coordinate directions. If $\frac{\lambda_1}{\lambda_2}$ is greater than a certain threshold parameter, it means only one of the corresponding eigenvectors is dominating and therefore only I-shape is considered further for that segment. If the ratio $\frac{\lambda_1}{\lambda_2}$ is lesser than the threshold parameter, both L-shape and I-shape are considered. Additionally, if there are not enough number of points N_{min} in the segment, it is then assumed to have only O-shape. The minimum required number of points and the threshold value for the eigenvalue ratio are tuned empirically as 3 and 10, respectively.

6.2.2 Geometric Shape Fitting

The aim of fitting a shape to the points in an segment is to overlay a bounding box or fit a line, in order to estimate the orientation of the object and to chose a reference measurement point for that segment. There are many methods and algorithms available from the field of robotics for extracting lines from a set of points. Some important methods are summarized in [NMTS05]. If the model pre-selection denotes that a segment must be tested for both L-shape and I-shape, iterative end point fit (IEPF) algorithm is first applied on that segment. The algorithm is also known in literature under the variants Ramer-Douglas-Peucker (RDP) algorithm or split-and-merge algorithm. The basic idea is to fit a line with first and the last point of the segment and find a point which has the maximum orthogonal distance to this line. This point intuitively is the corner point and the set is split into two individual set of points at this point. Lines are then fit individually to these sets of points. Now we are left only with the segments that need to be looked for I-shape. Similar methods of shape fitting are used in many previous works [Mun11, Kä07, Wen08], to list a few.

Hough-Transform introduced by Richard Duda and Peter Hart is a widely used method in the field of computer vision for extracting line segments out of images. The key idea is to represent a line in it's Hesse normal form

$$r = x\cos\theta + y\sin\theta \tag{6.4}$$

where r is the perpendicular distance from origin to the line and θ the angle made by the line with x-axis. The line is represented in the above normal form instead of general form $y = mx + b$ because in case of vertical line the slope parameter would lead to singularity. Whereas a vertical line in normal form can be represented with $\theta = 90°$and r equal to the x-axis intercept. The parameter space or Hough space is given by (r, θ). Generally speaking, any number of lines can be drawn through a given point in xy-plane. This would mean a sinusoidal curve for each point on the Hough-space with varying θ and r. Such sinusoidal segment is constructed for all the points in the cluster. Then the value of r' and θ' in the Hough-space, where the sinusoidal curves of all points intersect would give the line in the xy-plane, which passes through all those points or in other words a line which fits the points in the cluster.

However due to highly varying detections and noise in the measurement, finding an intersection point in the Hough-space for a best line fit is not direct. Therefore the uncertainty in the x and y positions, correspondingly the uncertainty in r and θ values should also be considered. An alternative method to handle uncertainty for line fitting in the weighted least squares sense is proposed in [Arr03] and also used in [Mun11]. Weight for each point is determined based on the uncertainty in range $\sigma^2_{r,i}$ and angle $\sigma^2_{\theta,i}$. It is further suggested in [Arr03] that the implementation complexity is lower if the line parameters θ' and r' are calculated in terms of Cartesian coordinates as:

$$\tan 2\theta' = \frac{-2\sum w_i(\bar{y}_w - y_i)(\bar{x}_w - x_i)}{\sum w_i\left[(\bar{y}_w - y_i)2(\bar{x}_w - x_i)^2\right]} \tag{6.5}$$

$$r' = \bar{x}_w \cos\theta' + \bar{y}_w \sin\theta' \tag{6.6}$$

where $\bar{x}_w = \frac{1}{\sum w_i}\sum w_i r_i \cos\theta_i$ and $\bar{y}_w = \frac{1}{\sum w_i}\sum w_i r_i \sin\theta_i$ are the weighted means. In order to give higher weights to the points with lesser deviation, the weights are chosen inversely proportional to the variances $\sigma^2_{r,i}$ and $\sigma^2_{\theta,i}$

$$w_i = \frac{1}{\sigma^2_{r,i} + \sigma^2_{\theta,i}} \tag{6.7}$$

Additionally the covariance equations are derived in [Arr03] from Taylor expansion of the equations for r and θ in polar coordinate forms

$$\begin{aligned}\sigma_{\theta'\theta'} = &\frac{1}{D^2 + N^2}\sum w_i^2[N(\bar{x}_w \cos\theta_i - \bar{y}_w \sin\theta_i - r_i \cos 2\theta_i) \\ &- D(\bar{x}_w \sin\theta_i - \bar{y}_w \cos\theta_i - r_i \sin 2\theta_i)]^2 \sigma^2_{r_i}\end{aligned} \tag{6.8}$$

$$\begin{aligned}\sigma_{r'r'} = \sum[\frac{w_i}{\sum w_i}\cos(\theta_i - \theta') + \nabla f_{r'}(\theta')(\bar{y_w}\cos\theta' - \bar{x}_w \sin\theta')]^2 \\ \cdot \sigma^2_{r_i}\end{aligned} \tag{6.9}$$

$$\sigma_{\theta' r'} = \sum \nabla f_r(\theta') f_r(r') \sigma^2_{r_i} \tag{6.10}$$

where N and D are the numerator and denominator in equation (6.5). After fitting a line to the points, the clusters which showed a possibility of L-shape are checked for line

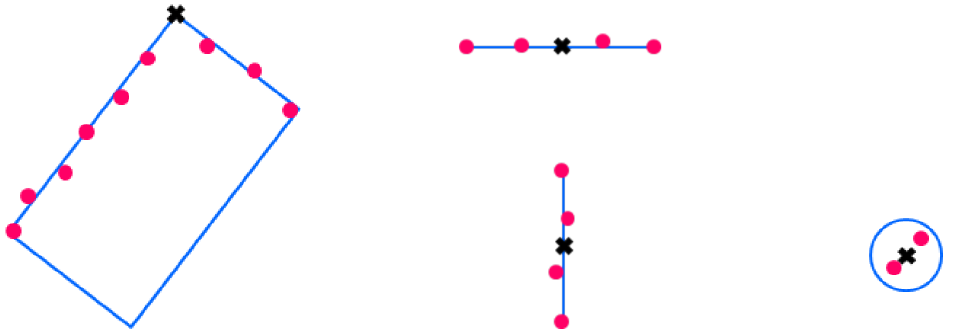

(A) L-shape (B) I-shape (C) O-shape

Figure 6.3: Considered geometric shapes for detection clusters. The black color cross represents the corresponding reference point (RP) chosen for the fitted shape

segments which are almost perpendicular to each other. In that case, the line segments are merged to fit the L-shape and a rectangular bounding box covering the nearest and farthest coordinate in the cluster is overlayed on the cluster as represented in Figure 6.3 to represent it's shape hypothesis. The orientation of the cluster is calculated by PCA. The covariance and eigenvectors are calculated as described in section 6.2.1. The orientation of the cluster is then given by the direction of the eigenvector $\mathbf{v_1}$ with highest eigenvalue λ_1

$$\psi_L = \tan^{-1}\left(\frac{\mathbf{v_1}(y)}{\mathbf{v_1}(x)}\right) \tag{6.11}$$

In order to calculate the four corner points of the bounding box, all the points in the cluster are first rotated by the orientation angle, so that the eigen vector $\mathbf{v_1}$ is parallel to the x-axis. After rotation, the maximum and minimum values of the x and y coordinates in the cluster represent the length BB_l and width BB_w of the bounding box. The coordinates of the corner points (p_1, p_2, p_3, p_4) representing the bounding box are finally calculated by again rotating the points of the cluster by it's orientation angle in the opposite direction. The corner point of the bounding box closest to the sensor is then chosen as reference measurement point for the cluster. If there are no perpendicular line segments that are to be merged, the cluster is then denoted with I-shape hypothesis and the middle point of the line is chosen as the reference point. The orientation of the I-shape is then same as the estimated angle θ'. As mentioned in the previous section, if there are not enough points for both of the L-shape and I-shape, the cluster is denoted with O-shape and the center point is chosen as reference and the orientation is set to zero. Figure 6.4 depicts a measurement sequence with a shape fitted for each of the detection clusters, belonging to different object classes.

6.2.3 Characteristic Feature Selection

Apart from the geometric features, each cluster basically contains other characteristics features as well, which is representative of the actual object class that the cluster belongs to. Extraction of characteristic features from measurement clusters and using these features as a basis for object classification has previously been studied in many works [AMB07, PLN09, Pie15, Fö06]. Most of the features that are searched for in each of the cluster are derived

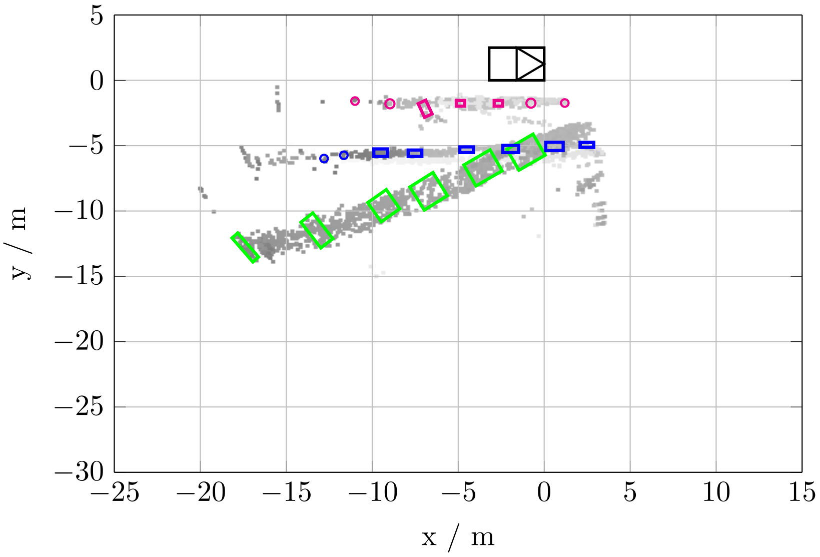


FIGURE 6.4: Example measurement scenario illustrating geometric shape fit. The green bounding boxes belong to the passenger car, blue to the bicyclist and magenta to the pedestrian. For clarity purpose, the shape fit of pedestrian and bicyclist are shown only for their forward motion.

from [AMB07], in addition to radar specific features. The features that are extracted for each cluster C are:

1. *Number of points:* denotes the total number of detection points in that cluster

$$NumOfPoints = |C_i| \tag{6.12}$$

2. *Standard deviation:* The standard deviation of the position of points is given by

$$Compactness = \sqrt{\frac{1}{n-1}\sum_j \|\mathrm{x_j} - \bar{\mathrm{x}}\|^2} \tag{6.13}$$

   where $\bar{\mathrm{x}}$ is the centroid of the cluster and $\mathrm{x_j}$ the positions of each point.

3. *Linearity:* This feature denotes the residual sum of squares between the cluster points and a line fitted to the cluster points [AMB07]. The parameters of the fitted line are calculated similar to the equations (6.5) and the linearity is given as

$$Linearity = \sum_j (x_j \cos(\theta') + y_j \sin\theta' - r')^2 \tag{6.14}$$

4. *Circularity:* The residual of sum of squares to a circle fitted for points in the cluster is denoted by this feature. If a circle of radius R and center (x_c, y_c) is fitted to the points by least squares method, then the circularity of the cluster is given by [AMB07]

$$Circularity = \sum_{j=1}^{n} \left(R - \sqrt{(x_c - x_j)^2 + (y_c - y_j)^2}\right)^2 \tag{6.15}$$

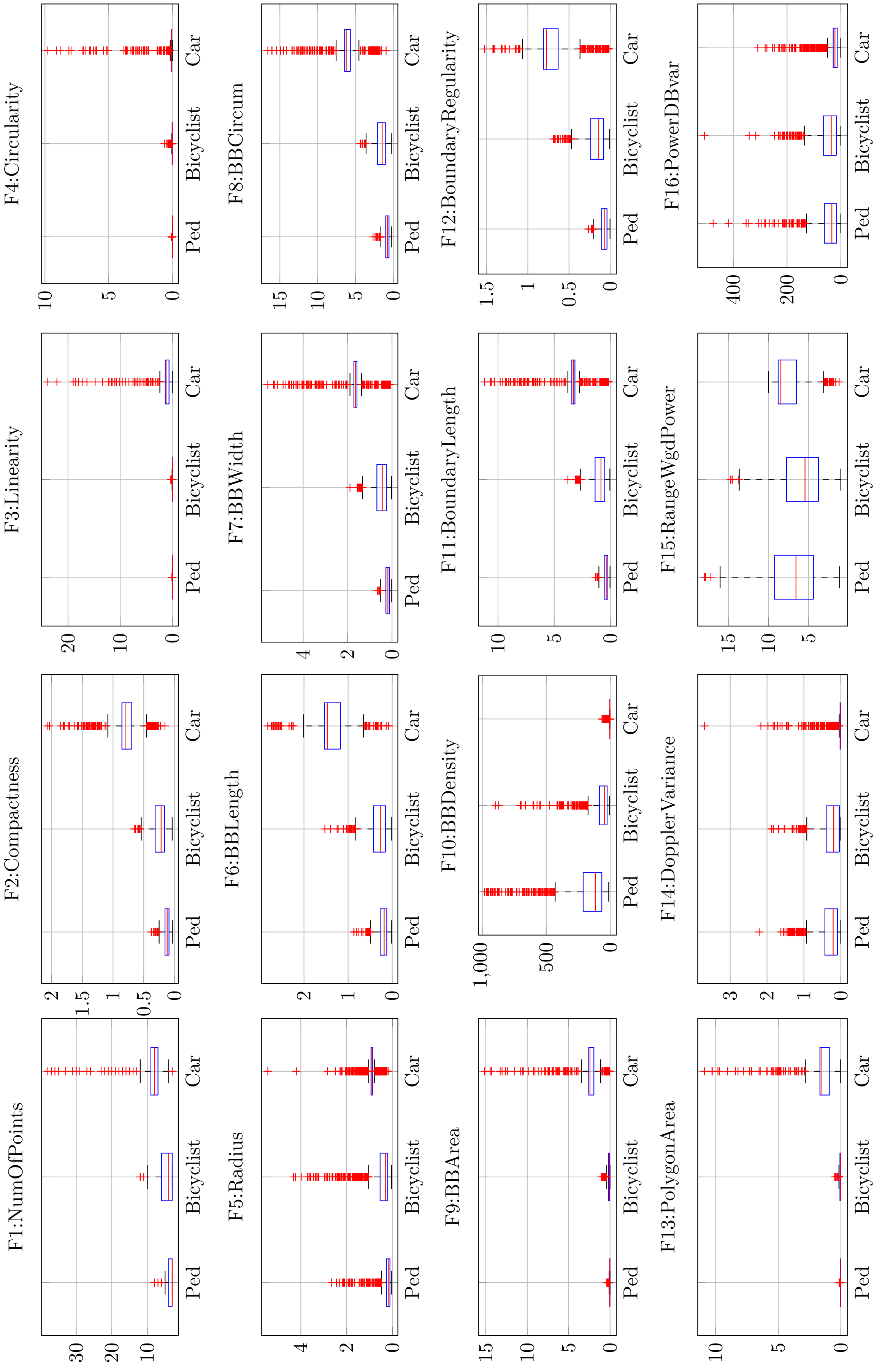


FIGURE 6.5: Boxplot of all the features belonging to different object classes.

with (x_j, y_j) denoting the positions of detection points in Cartesian coordinates.

5. *Radius:* Denotes the radius R of the circle fitted to the cluster for calculating circularity feature.
6. *Length of assumed bounding box:* The length of a bounding box $BBLength$ that would cover the points in the cluster is calculated as explained in section 6.2.2. For perfect lines, this value would be zero.
7. *Width of assumed bounding box:* This feature represents the width of the assumed bounding box $BBWidth$, again as calculated in section 6.2.2.
8. *Circumference of assumed bounding box:* Circumference of the bounding box
$$BBCircum = 2 \cdot (BBLength + BBWidth) \tag{6.16}$$
9. *Area of assumed bounding box:* The area of the rectangluar bounding box is calculated as
$$BBArea = BBLength \cdot BBWidth \tag{6.17}$$
10. *Density of assumed bounding box:* Density of the bounding box is the ratio of number of points in the cluster n to its area $area_{BB}$
$$BBDensity = \frac{NumOfPoints}{BBArea} \tag{6.18}$$
11. *Length of the boundary:* The length of the contour formed by connecting the points in a cluster is calculated by
$$\begin{aligned} BoundaryLength &= \sum_j d(p_j, p_{j-1}) \\ d(p_j, p_{j-1}) &= \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2} \end{aligned} \tag{6.19}$$
12. *Boundary regularity:* The standard deviation of the distances $d(p_j, p_{j-1})$ [AMB07].
13. *Polygon Area:* The area of the polygon fitted to the cluster of detection points.
14. *Doppler variance:* This feature denotes the variance of the Doppler velocity within a cluster. Due to the movement of the arms and legs, the Doppler variance of the pedestrian class is expected to be higher than other classes.
15. *Range weighted mean power:* The amplitude of electro-magnetic wave reflected from metal surfaces, as in case of vehicles, is expected to be higher than that of weak reflecting surfaces like pedestrian body. This feature is therefore used for representing the reflection characteristic of an object. However, the reflection power of objects that are in a closer range to the sensor is greater than that of the object at a higher range. In order to consider this range dependency, a weight is given to the power of a reflection point in the cluster. The weight should be inversely proportional to the

range

$$\begin{aligned} RangeWgdPower &= \frac{\sum_j w_j A(j)}{n} \\ w_j &= \frac{1}{r_j} \end{aligned} \tag{6.20}$$

16. *Power variance:* This feature represents the variance of reflection power in a cluster and is denoted by $PowerDBvar$. Objects with regular reflecting surfaces like passenger car is expected to have a lesser variance of the reflected power through out it's contour, whereas a bicycle for example can have a higher variance.

The system setup, for measuring the representative features of each object class, is same as described in Chapter 4. The object classes of interest are pedestrian, bicyclist and passenger car. All the features need to be calculated for every cluster. The cluster should be labeled in order to know which class of object the features belong to. In order to make this labeling task easier, separate measurements are made for each class. Many sets of measurement are made for each object class, with different maneuvers, in order to capture a wide spread of feature data. A bigger test area of radius more than 60 m is chosen in order to avoid the reflections that could come from other objects in the test area, which would cause a bias in feature extraction. However, sensor clutter cannot be avoided explicitly but are removed from measurements similar to the preprocessing method explained in section 6.1. After these preprocessing steps, a total of 2658 labeled cluster samples are considered further for feature extraction and also for the classifier construction, described in the following section. Boxplot is a convenient tool for representing the distribution of data which shows the minimum value in the data set, first quartile, median, third quartile and the maximum value. The distribution of each extracted feature for each class is represented and compared as boxplot in Figure 6.5. It shows the possibility of distinguishing object classes based on the extracted features.

The relevance of all the features for class separation is usually not known explicitly. PCA is a widely used method in machine learning field, to analyse the contribution of each feature to the classification problem and subsequently for dimension reduction [LCZT07]. The feature vectors are projected onto a lower dimension with the help of the principal components (PC). The so called PCs are the vectors that capture the highest variance of the data. Figure 6.6 shows the biplot with corresponding data projected onto the first three PCs and also the coefficients representing the variance contribution. The features $BBDensity$ and $PowerDBvar$ dominates the first two PCs and other features are entangled in the subsequent higher dimensional PCs. The feature number of points $NumOfPoints$ and the features representing dimensions, area and density are highly correlated, pointing in similar direction in the biplot. A biplot illustrating the contribution of these features, neglecting the $BBDensity$ and $PowerDBvar$ is given in Figure 6.7.

However, many of the features like $NumOfPoints$ are dependent on the actual position of the object with respect to sensor. Consequently many features like circularity, radius and area depend on the $NumOfPoints$ in the cluster. But at a higher distance from sensor, as even a passenger car may only show same characteristic features as a pedestrian, considering these features in such conditions would cause a bias in class separability. Therefore, the classification is divided into two separate methods, considering the position dependability of the features. As the radar can measure the Doppler velocity of the object, it could

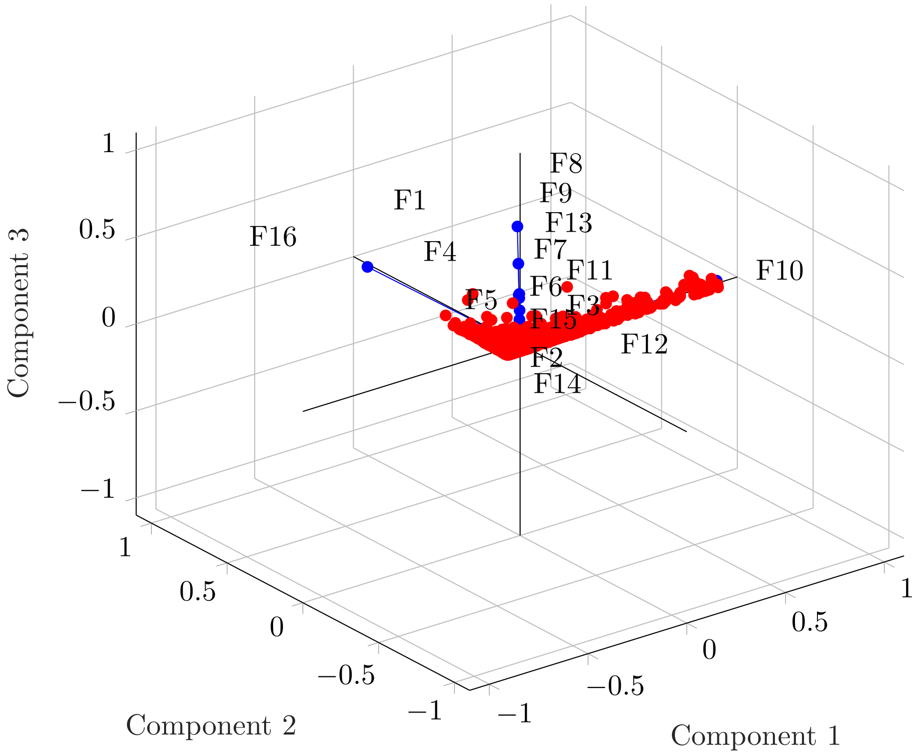


FIGURE 6.6: Biplot of all the features projected onto first 3 PCs. The features are numbered in the same order as in Figure 6.5.

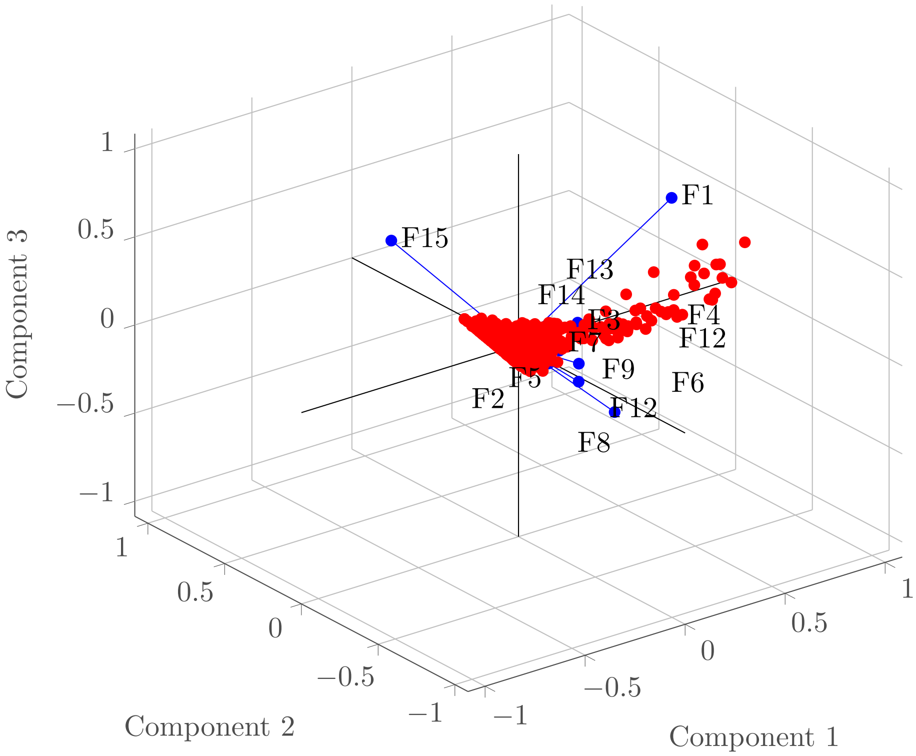


FIGURE 6.7: Biplot of reduced features projected onto PCs.

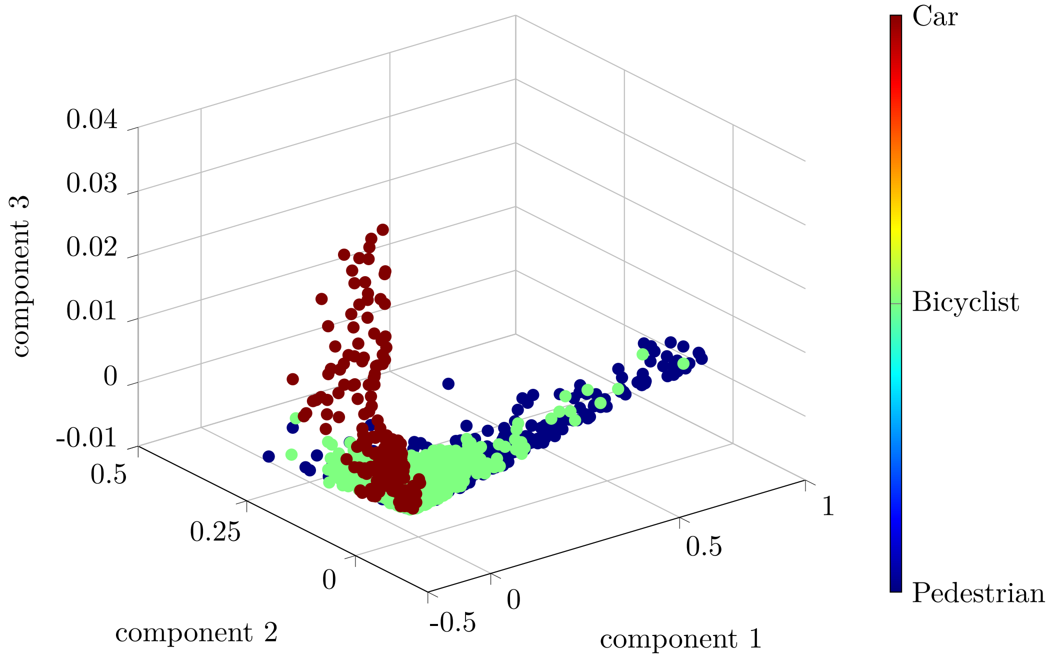


FIGURE 6.8: 16-D feature data set scaled onto 3-D space

be used for class-BBA in cases where the main classifier cannot give class probabilities based on all the defined features. But in conditions where the object moves with the same velocity as the ego-vehicle, this would cause a higher bias in class-BBA. Therefore the condition with $NumOfPoints < 3$ is synonymous to total ignorance of the classifier and full mass is given to $m(\Omega) = 1$. In that case, the class-BBA is dependent mainly on the estimated velocity of the tracked object. In other cases, the classification is based on the combination of assignments from track velocity as well as the output of the classifier. Design and implementation of the object classifier based on measurements and track attributes are described in the following sections.

6.3 Classification

For the given sensor characteristics and system setup, an object is expected to be in the sensor FoV for only a relatively shorter time interval. Which means, in order to have many training samples, many hours of test data needs to be logged. Support vector machine (SVM) is one of the classification methods least affected by the training data sample size [LWW$^+$14]. Therefore for object classification, SVM is used in this work. As mentioned in the previous section, the validity of feature extraction depends highly on the position of the object in the sensor's FoV. The azimuth separation for the classifier is empirically selected at $-60°$and $+60°$, based on many measurement observations.

6.3.1 Support Vector Machine

SVM is one of the well known supervised learning methods used for object classification. Since it's introduction in [CV95] it has been widely used in vehicle environment perception applications like image based detection of objects, vehicle maneuver prediction [Ott13] etc. SVM is a maximum margin classifier, which is able to derive a decision boundary between various classes in question. Basically the boundary is an hyperplane, calculated by mapping the inputs to higher dimensional feature space. More formally the problem can be stated as, given a set of training data $\{(\mathbf{x}_1, y_1), \ldots (\mathbf{x}_n, y_n)\}$ with $\mathbf{x}_i \in \mathcal{X}$ as input data and $y_i \in \mathbb{C} = \{-1, 1\}$ the class labels, the SVM calculates a decision function $F(x)$, which can then be used for predicting the value y_k for an input $\mathbf{x_k}$ at the time k [BGV92]. If the data in $\mathbf{x}$ are linearly separable, the decision boundary separating the classes would be the optimal hyperplane in the input space $\mathcal{X}$, which is given by the decision function

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i(\mathbf{x}_i) + b = 0 \tag{6.21}$$

where the parameters $\mathbf{w}$ and b are the parameters to be determined by an optimisation problem. The distance between the origin and the optimal hyperplane is then given by $\frac{b}{\|\mathbf{w}\|_2}$ with $\|\mathbf{w}\|_2$ representing the least-squares norm of $\mathbf{w}$. As the hyperplane is the boundary that separates the two classes, the condition satisfying each of the class can be written as

$$\begin{aligned} \mathbf{w}^\top \mathbf{x}_i + b &\geq +1, \text{ for } y_i = +1 \\ \mathbf{w}^\top \mathbf{x}_i + b &\leq -1, \text{ for } y_i = -1 \end{aligned} \tag{6.22}$$

which can be combinedly written as an inequality

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \tag{6.23}$$

Intuitively, for a well distinction of the classes, the margin between two classes separated by the optimal hyperplane need to be maximized. The distance between the optimal hyperplane and $\mathbf{x}$ is $\frac{y_i(\mathbf{w}^\top \mathbf{x} + b)}{\|\mathbf{w}\|_2}$. Therefore the margin distance between the two classes from (6.22) are

$$M = \frac{1}{\|\mathbf{w}\|_2} - \frac{-1}{\|\mathbf{w}\|_2} \Rightarrow \frac{2}{\|\mathbf{w}\|_2} \tag{6.24}$$

The optimisation problem for linear case is then to maximize the margin M or in other words to minimize $\|\mathbf{w}\|_2^2$, which is formulated as [BGV92]

$$\begin{gathered} \min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{subject to } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \\ i = 1, 2, \ldots N \end{gathered} \tag{6.25}$$

The above problem is called the *hard-margin* SVM. However, in many cases there is no direct linear separability of the classes. In that case the problem is solved by including a

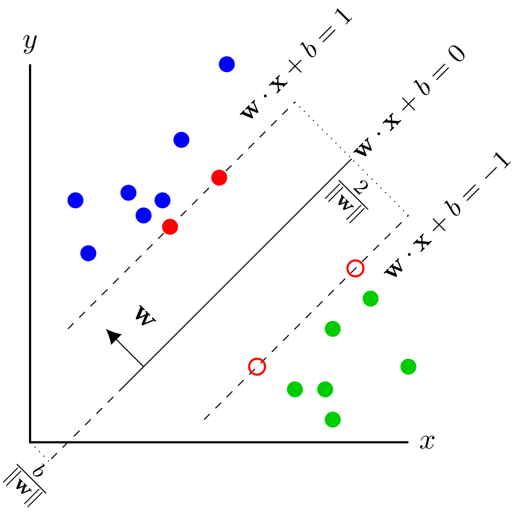


FIGURE 6.9: Linear SVM.

slack variable ξ_i representing the margin failure and regularisation constant C

$$\begin{aligned} &\min_{\mathbf{w}} \ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ &\xi_i \geq 0, i = 1, 2, \dots N \end{aligned} \tag{6.26}$$

and is called as *soft-margin* SVM. Moreover, when the classes are not linearly separable in the input space $\mathcal{X}$, they can however be mapped using to a higher dimensional feature space $\mathcal{F}$ by the mapping $\Phi : \mathcal{X} \mapsto \mathcal{F}$. The classes are then again linearly separable in the higher dimensional feature space $\mathcal{F}$ as illustrated in Figure 6.10.

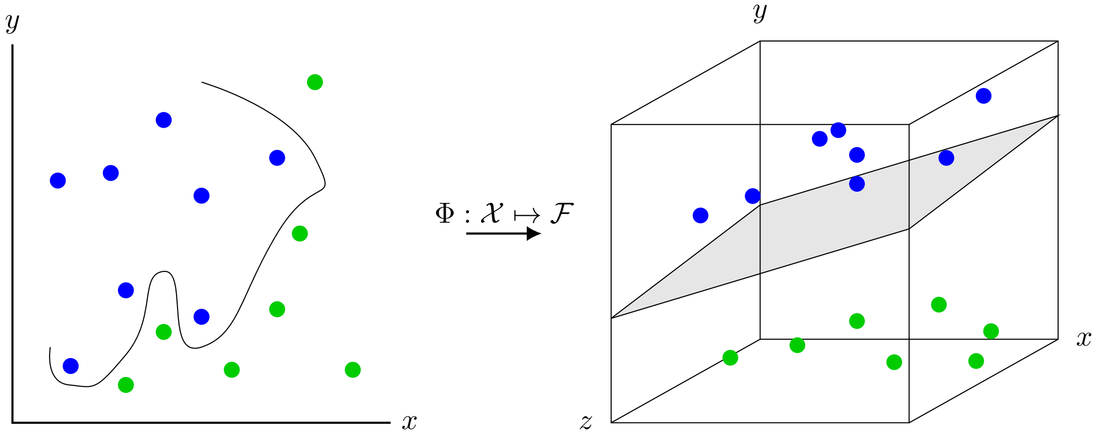


FIGURE 6.10: Non-linear SVM with kernel trick.

In [CV95], Lagrangian method to solve the above optimisation problem (6.26) in the dual space is derived. With Lagrangian multipliers α_i and β_i, the Lagrange functional is given as

$$L = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i(y_i(\mathbf{w}^\top \ell(\mathbf{x}_i)) + b) - \sum_i \beta_i\xi_i \tag{6.27}$$

and hyperplane is the solution of the optimisation problem in it's dual form

$$\begin{aligned} &\max_{\alpha}\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x_i})^\top\phi(\mathbf{x_j}) \\ &\text{subject to } \sum_{i=1} N\alpha_i y_i = 0 \text{ and } 0 \le \alpha_i \le C \end{aligned} \tag{6.28}$$

The mapping and the dot product computation is realised using a kernel function $K(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i})^\top\phi(\mathbf{x_j})$. There are many kernel functions that can be used for SVM [Hof06]. Radial basis function (RBF) kernel is used in this work. Therefore, the classification decision function is given as

$$F(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i e^{-\gamma\|x_i - x\|^2} + b\right) \tag{6.29}$$

So for the implementation of the above formulated SVM problem, two parameters need to found: C which controls the softness of the margin and γ which controls the hyperplane curvature for non-linearity. In order to build the classifier, the SVM needs to be trained first. The entire feature data with 2658 samples is scaled to the range $[0, 1]$ as recommended by the authors of libsvm [CL13]. 66.6% of the data set is divided as training data and the remaining 33.3% as testing data. The multi-class classification problem is solved as one-vs-rest classification problem in libsvm. Before the actual training of classifier, the parameters need to be estimated by cross-validation method. Cross-validation is the method of dividing the training set into certain fold of subsets, train the classifier with a set of subsets and sequentially test the classifier with one subset which was not used in training.

Grid-search on C and γ using cross-validation is recommended in [CL13] and used so in this work. Sequences of (C, γ) are tried and the one with the best cross-validation accuracy is selected. Initially a rough grid-search with 3-fold cross-validation and step-size 2 is made with broad (C, γ) ranges $C = 2^{-20}, 2^{-18}, \ldots, 2^{20}$ and $\gamma = 2^{-20}, 2^{-18}, \ldots, 2^{20}$. This is to roughly find the parameter regions where the cross-validation accuracy is high. As depicted in Figure 6.11, the best pair obtained is $(2^{18}, 2^2)$ with cross-validation accuracy of 84.3115% and the regions bounded by $C = 2^{-4}, \ldots, 2^{20}$ and $\gamma = 2^{-12}, \ldots, 2^{12}$ has higher cross-validation accuracy. However, the number of support vectors (SV) increases by increasing the slack value i.e. C. This consequently increases the prediction complexity because for every test point, a dot product with each SV needs to be computed. On the other hand, a very low value of C would increase misclassification. A higher value for γ would result in overfitting and a very low value would not reflect the non-linearity in the decision boundary. Therefore a trade-off is to be made between the cross-validation accuracy and kernel parameters. So in the second medium-scale grid-search the scale of (C, γ) is reduced to the range $C = 2^{-2}, 2^{-1}, \ldots, 2^{18}$ and $\gamma = 2^{-6}, 2^{-5}, \ldots, 2^8$ with a step-size 1. The best pair from medium scale search is $(2^{16}, 2^3)$, with cross-validation accuracy of 84.0293%, as shown in Figure 6.12. Finally, the search is narrowed down to the region $C = 2^4, 2^{3.75}, \ldots, 2^{13}$ and $\gamma = 2^{-2}, 2^{-1.75}, \ldots, 2^6$ with a step-size 0.25, as depicted in Figure 6.13 and the best parameter is found to be $(2^{13}, 2^3.5)$ with a cross-validation accuracy of 82.9007%. Considering the trade-off, the parameter pair $(2^10.5, 2^4.25)$ providing a cross-validation accuracy of 81% is

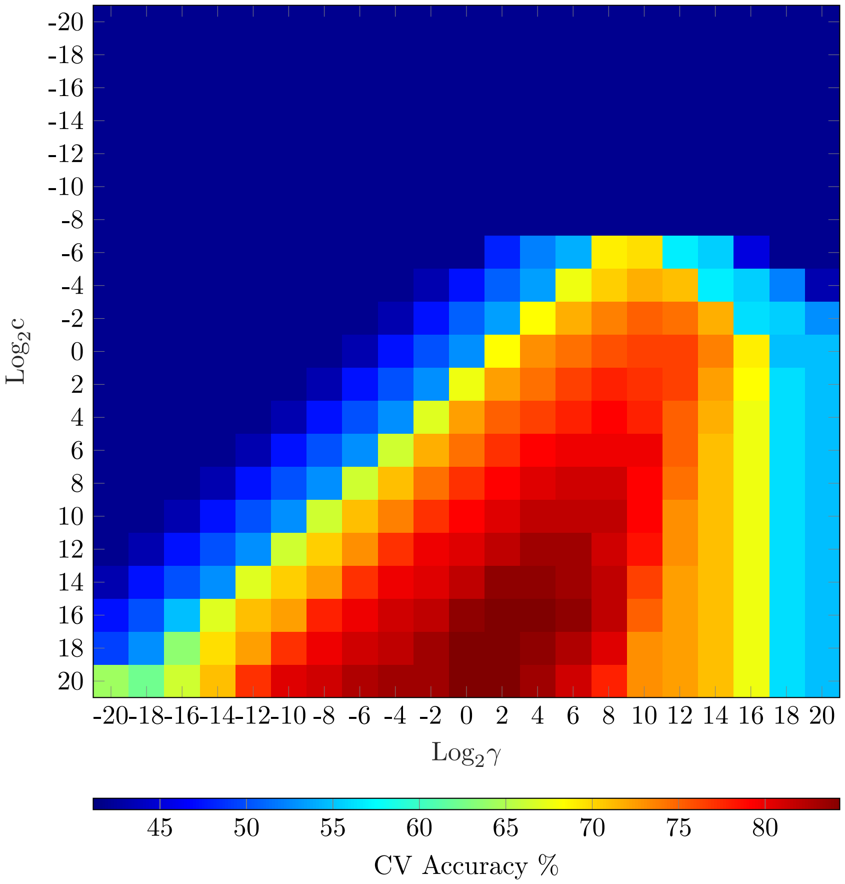


FIGURE 6.11: Rough RBF-Kernel parameter search by cross-validation with step-size 2.

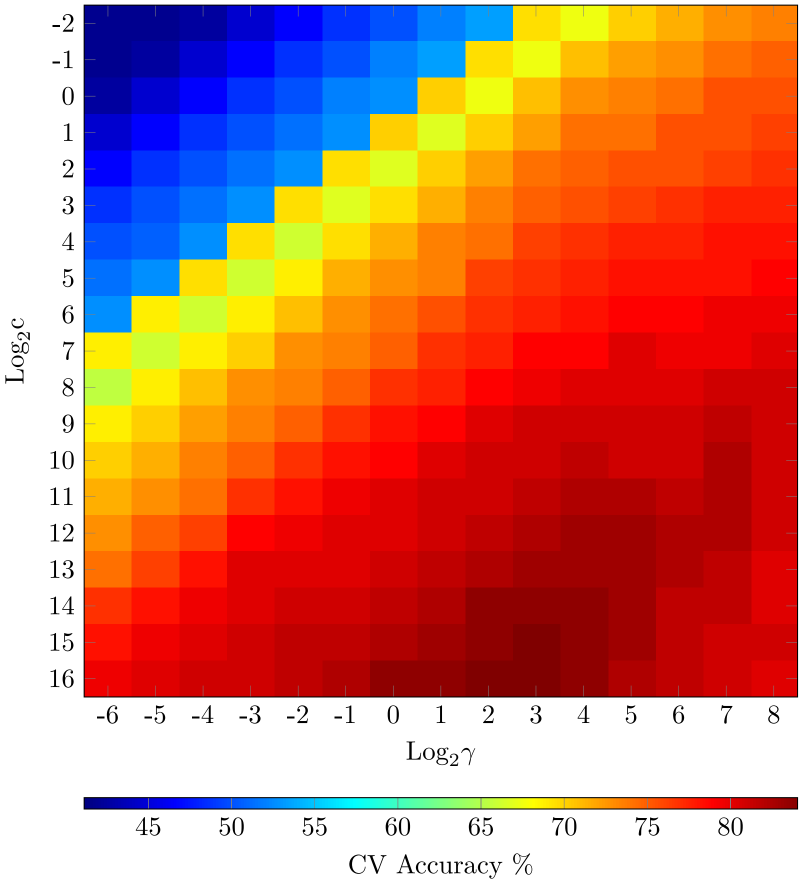


FIGURE 6.12: Medium scale RBF-Kernel parameter search by cross-validation with step-size 1.

chosen for training the model, along the direction of best parameter in the grid.

The evaluation of the trained SVM classifier is presented in section 8.3. The SVM outputs the class of the object and the decision or confidence values for each of the classes. But however it does not directly give the probabilities for each of the class. As the confidence values would be used in combination with the DST, they need not necessarily be probability values. In [Pla99] a method termed as 'Platt Scaling' is proposed which fits the confidence output of SVM onto a sigmoid function, in order to approximate the confidence values as probabilities. The probability is calculated as an optimisation problem with two parameters A and B to be minimised and the probability for positive class is given as

$$P\left(y=1 \mid f\right) = \frac{1}{1+e^{(Af+B)}} \tag{6.30}$$

However the above method is proposed mainly for two class problems and do not directly address multiclass problems. A method for transforming the confidence values output of SVM into accurate multiclass probability estimates is proposed in [ZE02]. The idea is to break the multiclass problem into many binary problems. Then each of them is calibrated separately into probabilities and their probabilities are combined. In case of one-vs-all method, the probabilities are normalised to 1. For one-vs-one method, a code matrix denoting the positive, negative and not considered classes of each binary classifier is constructed and the

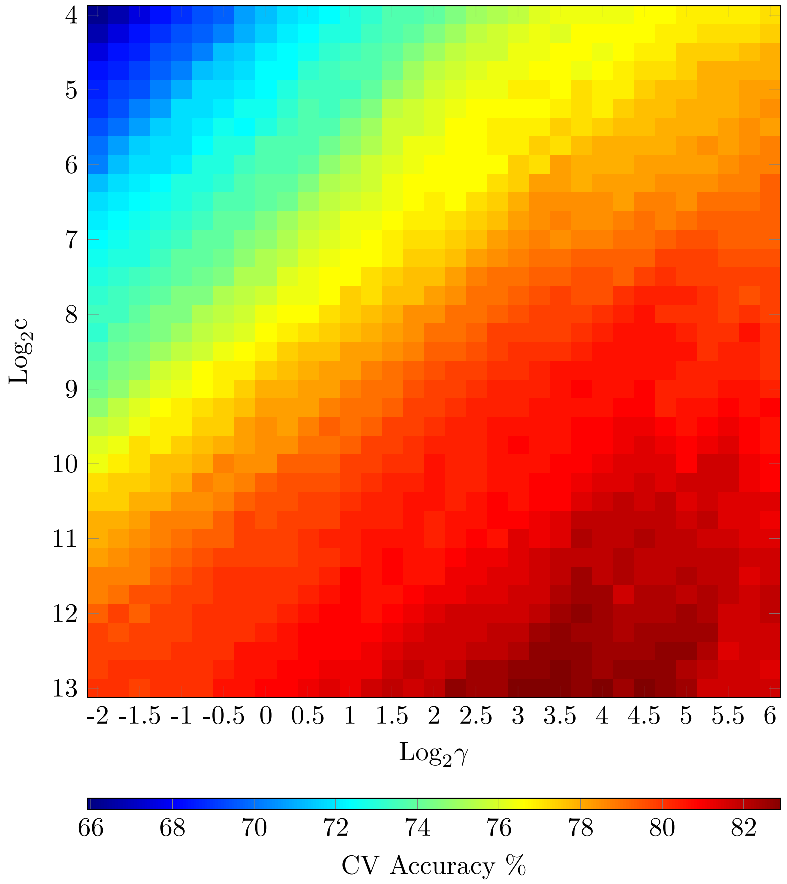


FIGURE 6.13: Small scale RBF-Kernel parameter search by cross-validation with step-size 0.25.

probabilities are calculated either by least-squares method or an iterative log-loss minimisation method called coupling.

6.3.2 Class probabilities based on track velocity

Apart from the measurement features based class probability estimation with SVM, the velocity information from the object tracks can also be used for estimating the class of the object. According to the experiments and studies in [KR10], pedestrians are expected to walk with mean velocity of ≈ 1.42 m/s. The characteristics of bicyclists' speed is studied in [JROR10] and states that the average daytime speed of a bicyclist is ≈ 4 m/s. However the speed of a bicyclist can vary between a lower speed of ≈ 2.5 m/s to higher speed of ≈ 5.6 m/s, making them difficult to differentiate from a slow moving passenger car. Considering theses studies, the belief masses based on track velocities can assigned to the elements of the power set according to the assumed function illustrated in Figure 6.14. This is based on the similar method proposed in [Gar14] and in [MRSD14] for combining measurement and track based class information. The combination of class information based on measurement features and track attributes is then realised by the integrated tracking and classification method presented in section 5.8.

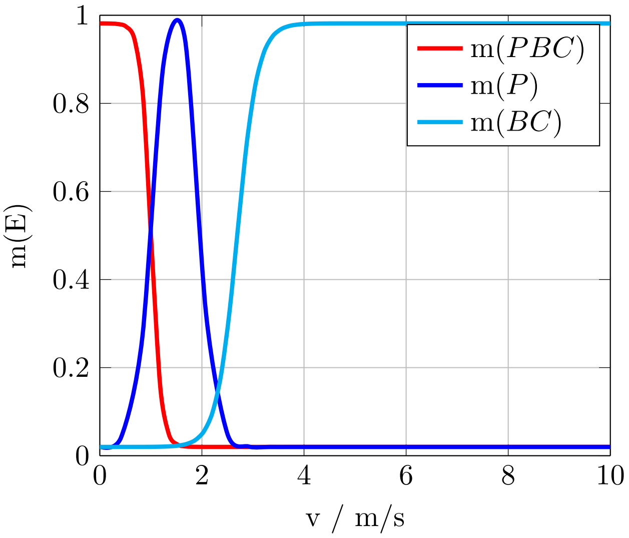


FIGURE 6.14: Approximation of belief mass according to track velocity

Chapter 7

Application and Realisation of Environment Perception for Truck ADAS

As described in Chapter 5, the recursive Bayesian state estimators require a motion model representing the movement of the object and a measurement model representing the sensor observations. Moreover, the estimators also require the knowledge of sensor measurement characteristics, in order to parameterise the measurement noise distribution. Measurement analysis, object motion models, sensor and ego-vehicle model assumptions, that are required in addition to the object classification techniques described in section 6.3 for the realisation of vehicle environment perception are described in this Chapter. The analysis of sensor measurement characteristics by means of experiments is presented in section 7.1. Assumptions about object motion models and the sensor measurement models are described in sections 7.2 and 7.3. A method for compensating the ego-vehicle motion from object motion is described in section 7.4. The Chapter concludes by describing a scheme for implementing the DST required to integrate the class information into the tracking framework.

7.1 Measurement Characteristics

The measurement characteristics of the used radar sensor are also evaluated experimentally. Thereby, the variances of azimuth σ_θ^2, range σ_r^2 and Doppler velocity $\sigma_{\dot{r}}^2$ measured by the radar are estimated. The measurement setup is similar to the calibration setup described in section 4.3.2. The corner reflector of a known RCS value is placed along various coordinate positions in the FoV and the corresponding reflections are measured. Additionally measurement sequences of a pedestrian walking with predefined velocity are also recorded. The variances are calculated from many measurement samples and the corresponding reference position of the target. Figure 7.1 depicts the sensor measurement characteristics corresponding to different positions of the target. The radar has higher variance in measured values along the border areas of the FoV, especially for the azimuth angle. This can understood as due to the side lobe characteristics of the radar antenna. Note that the variance of Doppler velocity is estimated only from the tests where the target is stationary or moves radially to the sensor at zero azimuth. This is because the reference Doppler velocity is same as the target velocity only for these two conditions and the Doppler reference for other target maneuvers is not available. Thereby the variance of Doppler velocity is estimated as 0.055

m/s. Also from the various measurements through the detection area, the opening angle of the radar sensor is experimentally verified to be approximately 140°.

7.2 State Models

For the application of MMLMB filter, multiple state models are required for modeling the motion characteristics of different objects. CV and CTRV models are used in this work where the CV model represents the pedestrian motion and CTRV model represents bicyclists and cars.

CV model

In a CV model the object is assumed to move with a constant linear velocity. The state-space constitutes the object longitudinal and lateral positions (x, y) and corresponding velocities (v_x, v_y) in the Cartesian vehicle coordinates. The states and the corresponding covariance matrix at any time instance k are given as,

$$\mathbf{x}_k^{cv} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \tag{7.1}$$

$$\underline{\mathrm{P}}_k^{cv} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & 0 & 0 \\ \sigma_{xy} & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_x}^2 & \sigma_{v_x v_y} \\ 0 & 0 & \sigma_{v_x v_y} & \sigma_{v_y}^2 \end{bmatrix} \tag{7.2}$$

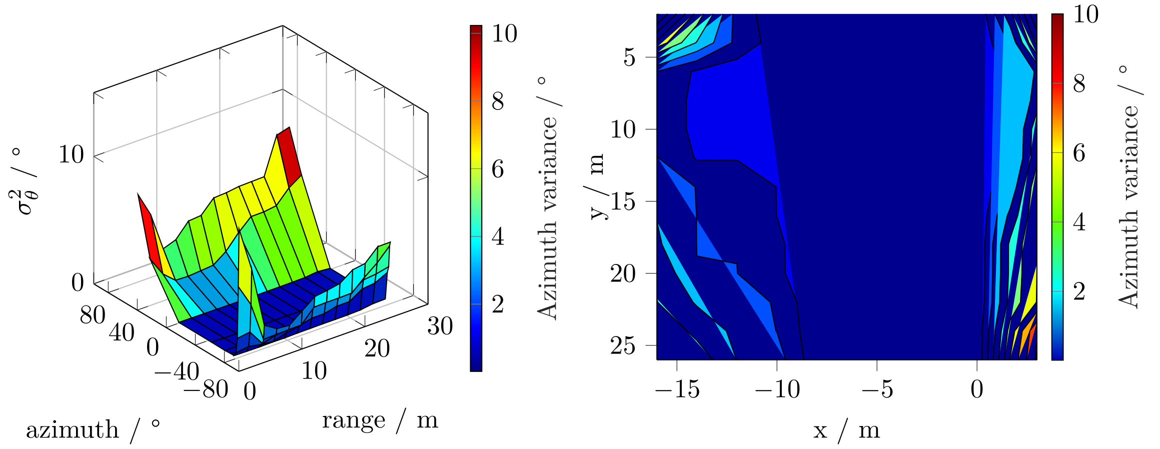
The states are then predicted with a discrete time step T according to the equation

$$\mathbf{x}_{k+1}^{cv} = \underline{\mathrm{F}}_{cv} \mathbf{x}_k + w_k \tag{7.3}$$

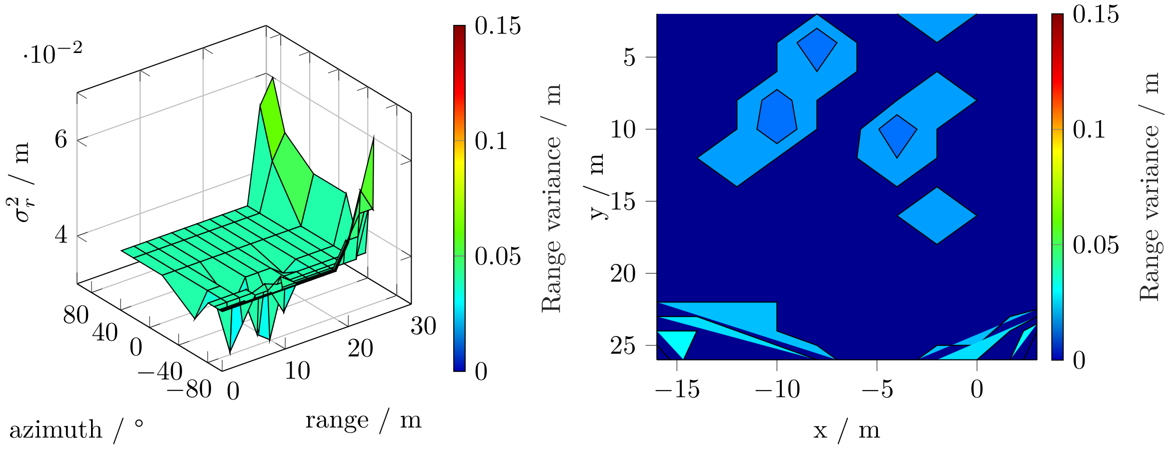
$$\underline{\mathrm{P}}_{k+1}^{cv} = \underline{\mathrm{F}}_{cv} \underline{\mathrm{P}}_k^{cv} \underline{\mathrm{F}}_{cv}^{\top} + \underline{\mathrm{Q}}_k^{cv} \tag{7.4}$$

where the state transition matrix $\underline{\mathrm{F}}_{cv}$ and the process noise matrix $\underline{\mathrm{Q}}^{cv}$ according to the Wiener-sequence acceleration model are given as

$$\underline{\mathrm{F}}_{cv} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.5}$$



(A) Polar coordinates (B) Cartesian coordinates



(C) Polar coordinates (D) Cartesian coordinates

FIGURE 7.1: Variance of the radar azimuth angle and range measurement with respect to the polar azimuth and range values are shown in Figure (A) & (C). Azimuth and range variance overlayed on the Cartesian x and y coordinates, with the sensor mounted looking towards rear is shown in Figure (B) & (D). The figure depicts the higher variance of the azimuth measurement along the border area of FoV, due to side lobes of the antenna.

$$\underline{Q}^{cv} = \sigma_a^2 \begin{bmatrix} \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 \\ 0 & \frac{T^4}{4} & 0 & 0 \\ \frac{T^3}{2} & 0 & T^2 & 0 \\ 0 & \frac{T^3}{2} & 0 & T^2 \end{bmatrix} \tag{7.6}$$

CTRV model

In case of a CTRV model, in addition to a constant velocity, the object is also assumed to have a constant turn rate ω. The state therefore consists of the positions and velocities denoted in Cartesian vehicle coordinates as in CV model, additionally augmented with the turn rate ω and can be given as

$$\mathbf{x}_k^{ctrv} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \\ \omega \end{bmatrix} \tag{7.7}$$

$$\underline{P}_k^{ctrv} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 & 0 & 0 & \sigma_{x\omega} \\ \sigma_{xy}^2 & \sigma_y^2 & 0 & 0 & \sigma_{y\omega} \\ 0 & 0 & \sigma_{v_x}^2 & \sigma_{v_x v_y}^2 & \sigma_{v_x\omega} \\ 0 & 0 & \sigma_{v_x v_y}^2 & \sigma_{v_y}^2 & \sigma_{v_y\omega} \\ \sigma_{x\omega} & \sigma_{y\omega} & \sigma_{v_x\omega} & \sigma_{v_y\omega} & \sigma_\omega^2 \end{bmatrix} \tag{7.8}$$

Consequently the states are predicted for a discrete time step T as

$$\mathbf{x}_{k+1}^{cv} = \begin{bmatrix} \underline{F}_{ctrv}(\omega) & 0 \\ 0 & e^{\frac{-T}{\tau_\omega}} \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} w_k & 0 \\ 0 & w_{\omega,k} \end{bmatrix} \tag{7.9}$$

$$\begin{aligned} \underline{P}_{k+1}^{ctrv} = & \begin{bmatrix} \underline{F}_{ctrv}(\omega) & 0 \\ 0 & e^{\frac{-T}{\tau_\omega}} \end{bmatrix} \underline{P}_k^{ctrv} \begin{bmatrix} \underline{F}_{ctrv}(\omega) & 0 \\ 0 & e^{\frac{-T}{\tau_\omega}} \end{bmatrix}^\top \\ & + \begin{bmatrix} \underline{Q}_k^{ctrv} & 0 \\ 0 & T^2\sigma_\omega^2 \end{bmatrix} \end{aligned} \tag{7.10}$$

where the turn rate ω is predicted with an assumption of first-order Markov process model and τ_ω is the corresponding correlation time constant for the turn rate. The transition matrix $\underline{F}_{ctrv}(\omega)$ in terms of turn rate and the process noise matrix $\underline{Q}^{ctrv}$ are given as

$$\underline{F}_{ctrv}(\omega) = \begin{bmatrix} 1 & 0 & \frac{\sin \omega T}{\omega} & \frac{1-\cos \omega T}{\omega} \\ 0 & 1 & \frac{1-\cos \omega T}{\omega} & \frac{\sin \omega T}{\omega} \\ 0 & 0 & \cos \omega T & -\sin \omega T \\ 0 & 0 & \sin \omega T & \cos \omega T \end{bmatrix} \tag{7.11}$$

$$\underline{Q}^{ctrv} = \sigma_a^2 \begin{bmatrix} \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 \\ 0 & \frac{T^4}{4} & 0 & 0 \\ \frac{T^3}{2} & 0 & T^2 & 0 \\ 0 & \frac{T^3}{2} & 0 & T^2 \end{bmatrix} \tag{7.12}$$

Turn rate in the current time step ω_k is used for calculating the matrix $\underline{F}_{ctrv}(\omega)$ [LJ03].

For the implementation of the GM-MMLMB filter with integrated classification described in section 5.8, each track, thereby their GMCs are predicted with respect to each motion model $o_i(i \in \{1, \ldots, N_o\})$. Also, in order to reduce the computational complexity, GMCs of a track with close mean values are merged as described in section 5.6.3. Consequently, for predicting each GMC of a track with both the motion models and also for merging the GMCs of a track conditioned on different motion models, the transformation of the states and covariances between both the motion models should be realisable. The transformation of the states from CTRV model to the CV model is achieved by augmenting the CV state space additionally with zero and also setting the corresponding element of the process transformation and covariance matrices as zero. The CV model is transformed into CTRV model where ω is initialised as zero and is propagated with help of a noise model. Additionally, all the objects are initialised with a higher probability for the CV model and the transition occurs by eventually updating the class-BBAs.

7.3 Measurement Model

The radar delivers the measurements in polar coordinates. Azimuth angle, range and Doppler velocity of a point target are measured and transmitted through CAN interface according to the system setup in Chapter 4. The measurements are transformed into Cartesian coordinates along with it's uncertainties. The measurement vector and it's noise covariance matrix in it's polar form is given as

$$\mathbf{z}_p = \begin{bmatrix} \theta \\ r \\ \dot{r} \end{bmatrix} \tag{7.13}$$

$$\underline{R}_p = \begin{bmatrix} \sigma_\theta^2 & 0 & 0 \\ 0 & \sigma_r^2 & 0 \\ 0 & 0 & \sigma_{\dot{r}}^2 \end{bmatrix} \tag{7.14}$$

The measurement vector and noise covariance matrix transformed into Cartesian coordinates at any time step k from the relation

$$r = \sqrt{x^2 + y^2} \tag{7.15}$$

$$\theta = \arctan(\frac{y}{x}) \tag{7.16}$$

$$\dot{r} = \sqrt{v_x^2 + v_y^2} \tag{7.17}$$

and is given as

$$\mathbf{z}_k = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} = \begin{bmatrix} r_k \cos\theta_k \\ r_k \sin\theta_k \\ \dot{r}_k \cos\theta_k \\ \dot{r}_k \sin\theta_k \end{bmatrix} \tag{7.18}$$

$$\underline{\mathrm{R}}_k = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 & 0 & 0 \\ \sigma_{xy}^2 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_x}^2 & \sigma_{v_x v_y}^2 \\ 0 & 0 & \sigma_{v_x v_y}^2 & \sigma_{v_y}^2 \end{bmatrix} \tag{7.19}$$

The elements of the converted covariance matrix corresponding to the Cartesian positions are considered as derived in [LBS93] as

$$\sigma_x^2 = \sigma_r^2 \cos^2\theta_k + \sigma_\theta^2 r_k^2 \sin^2\theta_k \tag{7.20}$$

$$\sigma_y^2 = \sigma_r^2 \sin^2\theta_k + \sigma_\theta^2 r_k^2 \cos^2\theta_k \tag{7.21}$$

$$\sigma_{xy}^2 = \frac{1}{2} \sin 2\theta_k \left(\sigma_r^2 - r_k^2 \sigma_\theta^2\right) \tag{7.22}$$

and the elements corresponding to the velocity components $\sigma_{v_x}^2$ and $\sigma_{v_y}^2$ are approximated with the constant value of 0.05 m/s derived from experiments in section 7.1. The observation matrix for the filter update for both the models can then be given as

$$\underline{\mathrm{H}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{7.23}$$

However, the radar can measure only the radial component of the object's velocity. The Doppler velocity measured by the radar would be same as the object velocity only when it is moving radially to the ego-vehicle as in case of a car driving in a straight highway. But for the given application setup in this work, the sensor is mounted on the side of the ego-vehicle and oriented towards the rear. Given typical scenarios where the object would mostly move parallel to the ego-vehicle, the velocity of the object is not always directly observable, especially when the object is closer to the sensor normal axis. In [APA04] a method to use the decoupled double filter (DDF) from [BP99] is suggested, where the radial and tangential motion of the object are tracked separately with two filters. The tangential component and radial component of the object velocity are then estimated by an assumption that the object moves with constant azimuthal acceleration. This method is situation specific and is not generic and requires additional association step. In [KBD+13], [FR06] a method to estimate a mean velocity based on best fit of the Doppler velocities of all the reflection points of the object is presented. However, the method is only suitable for large object like car which could give many reflection points. Therefore the Doppler velocity is used only in the regions of FoV away from normal axis of the sensor and is otherwise derived from the position. This is a loss of advantage of the radar sensor, but still the Doppler velocity can be used to initialise the state vector. The measurement vector and the observation matrix in that case are reduced to

$$\mathbf{z}_{r_k} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r_k \cos\theta_k \\ r_k \sin\theta_k \end{bmatrix} \tag{7.24}$$

$$\underline{\mathrm{H}_r} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{7.25}$$

Further, the transformation of measurement to the vehicle Cartesian coordinates is calculated as described in section 4.2.1, according to the mounting position of the sensor.

7.4 Ego-motion Compensation

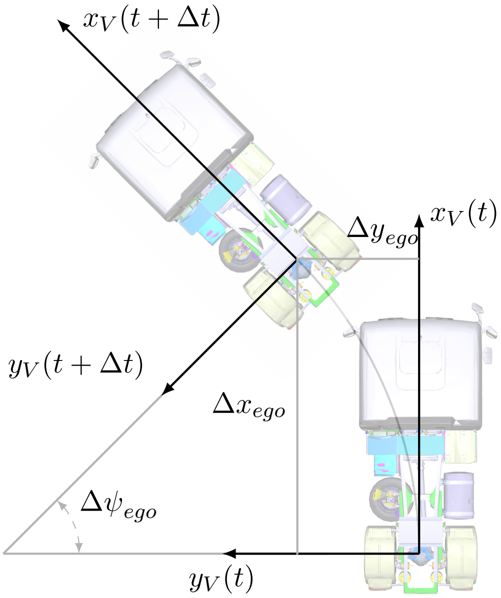


FIGURE 7.2: Ego-motion Compensation

As the ego-vehicle carrying the sensor itself is moving, the object motion should be compensated with the motion of the ego-vehicle. The object states predicted from time instance

k to $k+1$ with respect to the vehicle coordinates should be compensated with the respective motion of the ego-vehicle from k to $k+1$. As the radar sensor measures only in the azimuthal direction, it is sufficient to consider only the two-dimensional components of the ego-motion. The translational position change $(\Delta x_{ego}, \Delta y_{ego})$ and the rotational position change $(\Delta\psi_{ego})$ of the ego-vehicle between two time steps are calculated by using a kinematic single track model, neglecting the drift and wheel-slip. Therefore additional inputs required for the object state prediction are the change in ego-vehicle positions and the respective covariances. With a simple kinematic model assumption, the change in positions and yaw can be calculated from the tangential velocity v_{ego} and yaw-rate $\dot{\psi}_{ego}$ signals of the ego-vehicle which are received through the J1939 vehicle CAN interface.

$$\begin{aligned}
\Delta x_{ego} &= \frac{v_{ego,k}}{\dot{\psi}_{ego,k}} \cdot \sin\left(\dot{\psi}_{ego} \cdot T\right) \\
\Delta y_{ego} &= \frac{v_{ego,k}}{\dot{\psi}_{ego,k}} \cdot \left(1 - \cos(\dot{\psi}_{ego} \cdot T\right) \\
\Delta \psi_{ego} &= \dot{\psi}_{ego,k} \cdot T
\end{aligned} \tag{7.26}$$

Considering $\mathbf{u}_{ego,k+1} = [\Delta x, \Delta y, \Delta\psi]^\top$ as the additional input to the object state prediction from the ego-vehicle, the state prediction in equations (7.3) and (7.9) becomes

$$\mathbf{x}^{(\cdot)}_{comp,k+1} = \mathbf{f}_{ego}(\mathbf{x}^{(\cdot)}_{k+1}, \mathbf{u}_{ego,k+1}) \tag{7.27}$$

where the function $\mathbf{f}_{ego}$ represents the transformation of the predicted object states into the new ego-vehicle coordinate frame at $k+1$, with translation $\mathbf{t} = (\Delta x_{ego}, \Delta y_{ego})$ and rotation matrix

$$\mathbf{R}_{ego} = \begin{bmatrix} \cos(\Delta\psi_{ego}) & \sin(\Delta\psi_{ego}) \\ -\sin(\Delta\psi_{ego}) & \cos(\Delta\psi_{ego}) \end{bmatrix} \tag{7.28}$$

Both the positions and velocities are transformed, giving the ego-motion compensated predicted state of object as

$$\mathbf{x}^{(\cdot)}_{comp,k+1} = \begin{bmatrix} \cos(\Delta\psi_{ego})x + \sin(\Delta\psi_{ego})y - \Delta x_{ego} \\ -\sin(\Delta\psi_{ego})x + \cos(\Delta\psi_{ego})y - \Delta y_{ego} \\ \cos(\Delta\psi_{ego})v_x + \sin(\Delta\psi_{ego})v_y \\ -\sin(\Delta\psi_{ego})v_y + \cos(\Delta\psi_{ego})v_y \\ \omega \end{bmatrix} \tag{7.29}$$

with Δx_{ego} Δy_{ego} and $\Delta\psi_{ego}$ from equation (7.26). Apart from the object states, the predicted state covariances of equations (7.10) and (7.4) should also compensated with the tangential velocity and yaw-rate uncertainties of the ego-vehicle $\underline{\mathrm{P}}_{ego,k+1}$. Therefore the transformed state covariance matrix is given as

$$\underline{\mathrm{P}}^{(\cdot)}_{comp,k+1} = \mathbf{F}_{ego} \begin{bmatrix} \underline{\mathrm{P}}^{(\cdot)}_{k+1} & 0 \\ 0 & \underline{\mathrm{P}}_{ego,k+1} \end{bmatrix} \mathbf{F}^\top_{ego} \tag{7.30}$$

where $\mathbf{F}_{ego}$ is the Jacobian of $\mathbf{f}_{ego}$ calculated by partial differentiation

$$\mathbf{F}_{ego} = \frac{\partial \mathbf{f}_{ego}}{\partial(x, y, v_x, v_y, \omega)}\Big|_{\mathbf{x}_{k+1}, v_{ego}, \dot{\psi}_{ego}} \tag{7.31}$$

and the covariance matrix $\underline{\mathrm{P}}_{ego,k+1}$ depends on the uncertainties of the vehicle ESP sensors, given as

$$\underline{\mathrm{P}}_{ego,k+1} = \begin{bmatrix} \sigma^2_{v_{ego}} & \sigma_{v_{ego}}\sigma_{\dot{\psi}_{ego}} \\ \sigma_{v_{ego}}\sigma_{\dot{\psi}_{ego}} & \sigma^2_{\dot{\psi}_{ego}} \end{bmatrix} \tag{7.32}$$

7.5 Implementation of DST

As the above sections described the assumptions and model characteristics used in this work for object state estimation, this section describes a method to implement DST used for realising integrated tracking and classification according to the method presented in 5.8. The numerical operations involved in implementing DST are usually of high computational complexity. Binary representation of the elements in the power set 2^{Ω} is one of the methods to implement DST, in order to realise the combination rule [Lin06, Mun11]. Based on the three considered object classes - pedestrians, bicyclists and passenger cars, the corresponding power set can be given as

$$2^{\Omega} = \{\emptyset, \{P\}, \{B\}, \{P, B\}, \{C\}, \{P, C\}, \{B, C\}, \{P, B, C\}\}$$

The basic idea is then to represent all the elements of the power set with a binary value corresponding to the decimal value representing it's position order in the power set, in this case 0 to 7. The combination rule can then be realised by binary AND operation, which is equivalent to the union function $\cap$ and can be represented as [Mun11]

$$S_1 \cap S_2 = (S_1)_2 \& (S_2)_2 \tag{7.33}$$

The binary representation is given in Table (7.1). The rule of combination is therefore implemented by calculating the equation (5.149) as

$$m_C^{(i)}(E) = \frac{\sum\limits_{(S_{1_+})_2 \& (S_1)_2 = (E)_2} m_{C_+}^{(i)}(S_{1_+}) m_C^{(i)}(S_1)}{1 - \sum\limits_{(S_{1_+})_2 \& (S_1)_2 = \emptyset} m_{C_+}^{(i)}(S_{1_+}) m_C^{(i)}(S_1)} \tag{7.34}$$

The subset and non-empty intersection functions required for calculating the belief and plausibility as in equations (5.146) and (5.147) are realised as

$$(X \subseteq E) \Leftrightarrow ((X)_2 \& (E)_2 = (X)_2), \tag{7.35}$$

$$(X \neq \emptyset) \Leftrightarrow \left(\sum (X)_2 > 0\right) \tag{7.36}$$

2^2	2^1	2^0	
$\{C\}$	$\{B\}$	$\{P\}$	
0	0	0	$\emptyset$
0	0	1	$\{P\}$
0	1	0	$\{B\}$
0	1	1	$\{P, B\}$
1	0	0	$\{C\}$
1	0	1	$\{P, C\}$
1	1	0	$\{B, C\}$
1	1	1	$\{P, B, C\}$

TABLE 7.1: Binary representation of the elements in the power set according to their order.

where the operation $\sum (X)_2$ is the sum of all the bits of X represented in it's binary form. The pignistic transformation is realised by using the relation

$$|X| = \sum (X)_2 \tag{7.37}$$

Chapter 8

Evaluation

The results of the proposed tracking and classification methods are presented in this section. Metrics for the evaluation of tracking and classification performance are defined in the first section. Performance of the tracker is evaluated first with synthetic radar measurements as simulation and is also evaluated with real-world radar measurements. Different real-time test scenarios of object motion are defined and an analysis of object tracking and classification for these scenarios are presented in sections 8.2 and 8.3.

8.1 Tracker Evaluation Metrics

A reliable reference ground truth data is required for the evaluation of the tracking performance. A DGPS reference would provide a good ground truth reference, but as the object in most of the scenarios moves very close to the truck, the DGPS signal appeared very unstable. A high resolution camera mounted on the top right side of the truck is used for recording the test scenarios. The camera is calibrated in order to get the ground truth projection of the image. The test bicycle is attached with vivid color markers, which can be explicitly detected in the image. The markers, thereby their positions are detected with image processing algorithms, using the openCV libraries. The calibration setup and detection process are detailed in Appendix B. However, the camera has a narrow FoV than the used radar sensor, therefore the ground truth values are available only for a certain region of the radar FoV. Apart from the bicyclist, a dedicated ground truth reference is not available for the car and pedestrian class. Therefore additional markings for pre-defined object trajectory were made in the test area before making measurements and the reference markings were followed as close as possible.

8.1.1 RMSE

The difference between the estimated state value and it's corresponding reference value is called the error in state estimation and is given as

$$\widetilde{\mathbf{x}}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k \tag{8.1}$$

Considering N samples in the complete measurement sequence, the RMSE based on 2D Euclidean distance is given by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \frac{(\hat{x}(i) - x(i))^2 + (\hat{y}(i) - y(i))^2}{2}} \tag{8.2}$$

The equation (8.2) is general form of RMSE and is calculated separately for position, velocity and other estimated states.

8.1.2 Filter Consistency

Apart from the error in state-estimation, the consistency of the filter also serves as an important performance measure. Conditions for filter consistency are given in [BSWT11] and the first condition is that the state errors should be acceptable as zero mean and the covariance should match the mean square error. This is defined by means of the measure normalised state estimation error squared (NEES), given as

$$\epsilon_e = \tilde{\mathbf{x}}^\top \underline{P}^{-1} \mathbf{x} \tag{8.3}$$

The second condition is that the innovation error should also have the same property as above and is defined by normalised innovation square (NIS), given as

$$\epsilon_v = \mathbf{v}^\top \underline{S}^{-1} \mathbf{v} \tag{8.4}$$

where $\mathbf{v} = \hat{\mathbf{z}}_k - \mathbf{z}_k$ is the innovation error. Moreover, NIS has an advantage that it can be used for filter consistency check, even in cases where the reference state value is not available. In order for the filter to be consistent, the NEES and NIS being chi-squared distributed with n degrees of freedom, should lie within a certain acceptance interval $[r_1, r_2]$ which is determined based on a certain significance level α and is given by

$$P\left\{\epsilon_{(\cdot)} \in [r_1, r_2]\right\} = 1 - \alpha \tag{8.5}$$

If the NEES value for example is larger than the upper bound r_2, it denotes the possibility that the considered process or measurement noise covariances could be lower than the actual noise values. In case NEES is smaller than the lower bound r_1, it could denote that the noise covariances are over estimated.

The third condition is that the innovation should be white and is checked through auto-correlation statistics given by

$$\tilde{\rho}(k-a, k) = \sum_{i=1}^{a} v(k-a)^\top v(k) \left[\sum_{i=1}^{a} v(k-a)^\top v(k-a) \sum_{i=1}^{a} v(k)^\top v(k)\right]^{\frac{-1}{2}} \tag{8.6}$$

$\tilde{\rho}$ is white if it has zero mean and variance of $\frac{1}{a}$, where a is the number of accumulated time steps. In order for the filter to be consistent, $\tilde{\rho}$ should also be within the acceptance interval $[-r, r]$, where r is determined from the condition

$$P\left\{\tilde{\rho} \in \left[-r \cdot (\frac{1}{a}), r \cdot (\frac{1}{a})\right]\right\} = 1 - \alpha \tag{8.7}$$

8.1.3 Optimal Subpattern Assignment Metric

The metrics defined above are useful for determining the deviation and consistency of the tracker only with respect to the state estimation. But in multi-object tracking applications, a metric also for evaluating the cardinality estimate is required. The optimal subpattern assignment metric (OSPA) is proposed in [SVV08]. The distance between two state vectors x and y in the context of OSPA metric is calculated by

$$d_c(x, y) = \min(c, d(x, y)) \tag{8.8}$$

where $c > 0$ is the cut-off parameter. In order to represent the OSPA metric for two sets $\mathrm{X} = \left\{x^{(1)}, \dots, x^{(m)}\right\}$ and $\mathrm{Y} = \left\{y^{(1)}, \dots, y^{(m)}\right\}$ additionally an order parameter p is used and the OSPA distance is given as [SVV08]

$$d_p^c(\mathrm{X}, \mathrm{Y}) = \left(\frac{1}{n}\left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d_c(x^{(i)}, y^{(\pi(i))})^p + c^p(n - m)\right)\right)^{\frac{1}{p}} \tag{8.9}$$

where $m, n \in \mathbb{N}_0 = \{0, 1, 2, \dots\}$, Π_k is the set of permutations on $\{1, 2, \dots, k\}$ for any $k \in \mathbb{N} = \{0, 1, 2, \dots\}$. In the above equation the given condition is $m < n$. In case $m > n$ then the representation becomes $d_p^c(\mathrm{Y}, \mathrm{X})$. The function d_p^c is then the OSPA metric of order $p \in [1; \infty]$ and cut-off $c > 0$ [SVV08]. The first term in the summation of equations (8.9) is representation of the error in target state and the second term the error in cardinality. As stated in [SVV08] the cut-off parameter c mitigates the effect of higher OSPA distances resulting from state errors due to an higher value of p and adds weight to OSPA distances from the cardinality error. To evaluate the OSPA distances due to cardinality error and state error separately, the equation (8.9) can be split into two parts as

$$d_p^{c,loc}(\mathrm{X}, \mathrm{Y}) = \left(\frac{1}{n}(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d_c(x^{(i)}, y^{(\pi(i))})^p\right)^{\frac{1}{p}} \tag{8.10}$$

$$d_p^{c,card}(\mathrm{X}, \mathrm{Y}) = \left(\frac{1}{n} c^p (n - m)\right)^{\frac{1}{p}} \tag{8.11}$$

with equation (8.10) representing the localisation error and equation (8.11) representing the cardinality error.

8.2 Example Tracking Scenarios and Results

8.2.1 Simulation

The results of the developed tracking approaches are first illustrated in simulation with GM implementation. Totally 6 objects are simulated out of which 1 object is a car, 1 bicyclist and 4 pedestrians. The synthetic ground truth trajectories of the objects are shown in Figure 8.1. The objects are defined within the region of the radar FoV. The car moves with a constant velocity and turns in towards the ego-vehicle. The bicyclist moves parallel to the ego-vehicle with constant velocity. 2 pedestrians move parallel to the ego-vehicle starting from the rear end, 1 pedestrian moves from front end towards rear and 1

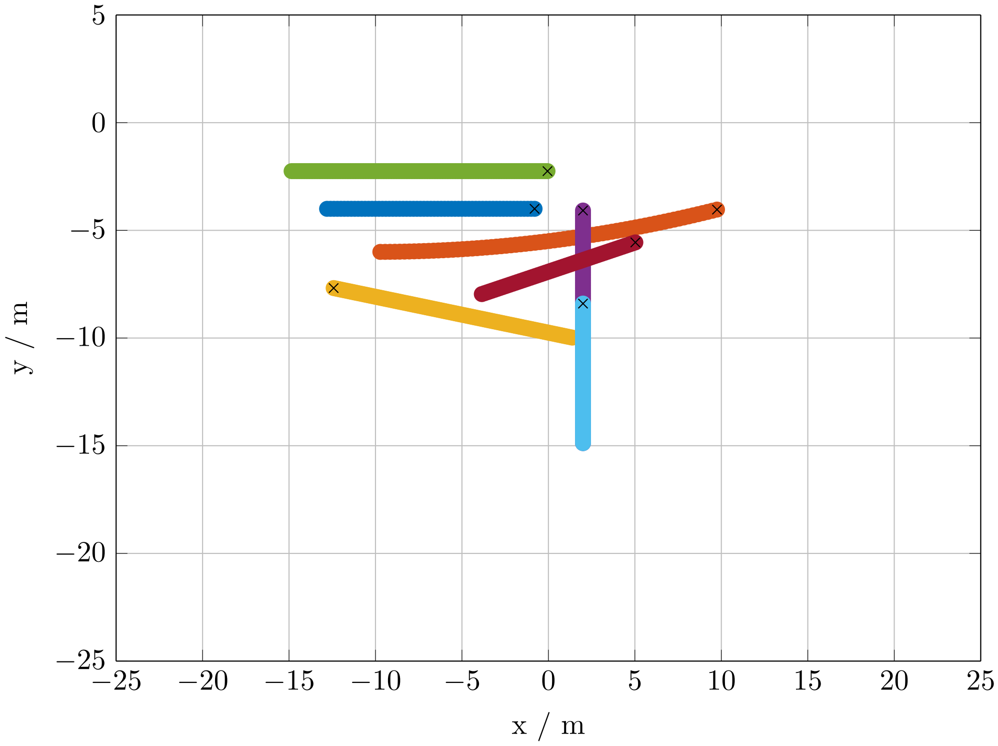


FIGURE 8.1: Simulated object trajectories withing the sensor FoV region. The mounting position of the sensor is assumed to be same as in section 4.2. The start positions of the objects are marked by a triangle and end positions by cross.

pedestrian moves forth and back orthogonal to the ego-vehicle. Two motion models, CV and CTRV are defined with standard deviation of the process noise in longitudinal x and lateral y directions as $\sigma_a = 1$ m/s^2. The standard deviation of the turn rate noise for CTRV is $\sigma_\omega = \pi/180$ rad/s. The radar measurements are simulated as reference points of the object with measurement variances σ_θ^2, σ_r^2 and $\sigma_{\dot{r}}^2$ as identified in section 7.1. The measurements and covariances are converted into Cartesian coordinates according to equations (7.18) to (7.19) and the sample time is $T = 0.08$ s. The survival probability of the tracks $p_s = 0.99$ and the detection probability of the sensor $p_D = 0.98$ are assumed to be state independent. The clutter measurements follow Poisson distribution with intensity $\kappa(z) = 1.3 \cdot 10^{-3}$, with average rate of $\lambda_c = 2$ clutter measurements per scan. The belief mass for the objects are defined with higher values for elements supporting their actual class and lower values are assigned for other elements of the power set. The belief mass of the clutter measurements are assumed with equal value for all the elements.

The objects are usually expected to appear only on the edges of the FoV. But in order to allow for a generic object initialisation throughout the FoV, an adaptive LMB birth intensity as described in section 5.6.4 is used. Moreover it allows re-initialisation of the lost tracks at any arbitrary region in the FoV [Reu14]. The LMB birth intensity is then defined by the parameters $\left\{r_B^{(i)}, p_B^{(i)}(\cdot, o)\right\}_{i=1,o=1}^{|\mathrm{Z_k}|,|\mathrm{O}|}$, where the existence probabilities $r_B^{(i)}$ are calculated as in equation (5.98). The measurements are initialised with higher probability of existence only if they have a higher Mahalanobis distance to any of the existing tracks. The spatial distribution $p_B^{(i)}(\cdot, o)$ is defined by a Gaussian distribution $\mathcal{N}\left(x; \hat{x}_B^{(i)}, \underline{\mathrm{P}}_B\right)$ conditioned on

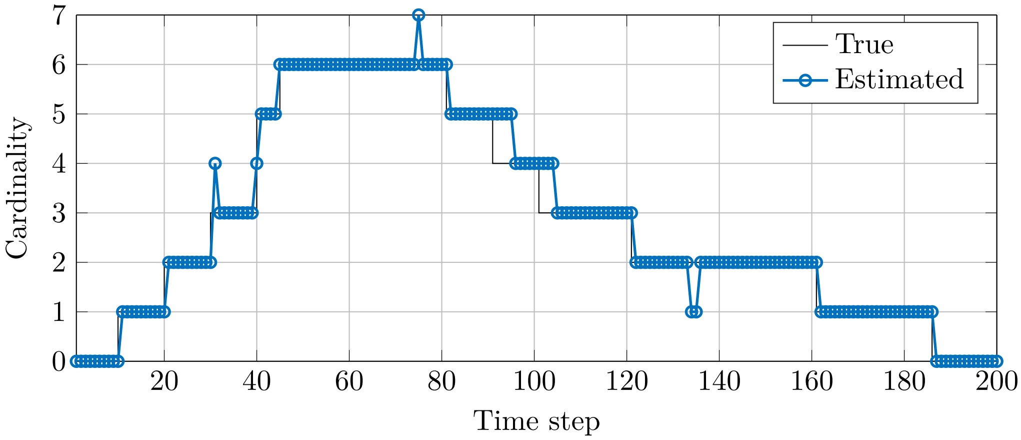


Figure 8.2: Cardinality estimate of the filter for the simulated object trajectories as in Figure 8.1.

each of the motion model, with it's mean position equal to converted x and y positions of the measurement. The turn rate is set to zero and the velocity is initialised with the x and y components of the Doppler velocity if the measurement has an absolute value of azimuth greater than 60°, else set to zero. The uncertainties of the position and velocities of the Gaussian $\underline{\mathrm{P}}_B$ are initialised with the Cartesian converted covariances of the measurements and the variance of turn rate is set as $(\pi/180)^2$. The initial model probabilities are set as $[p(o)] = [0.9\ 0.1]$. The class-BBAs of the birth tracks are set to the class-BBAs of the measurement and the model transition probabilities are calculated from these class-BBAs. Additionally the track management parameters are defined such that all the tracks with existence probabilities $r < 0.001$ are pruned. For reducing the computational complexity all the Gaussian components of the track which have weights $w < 1 \cdot 10^{-4}$ are pruned and Gaussian components with a Mahalanobis distance $d_{MHD} < 0.2$ are merged.

The cardinality estimates of the filter is shown in Figure 8.2. The estimated cardinality overall matches the true cardinality very well. The cardinality is overestimated at $k = 31$ and $k = 75$ due to initialisation of a false track from clutter. The pedestrian moving orthogonal to the ego-vehicle is lost at $k = 132$ due to simulated instantaneous change of direction, but is re-initialised again $k = 136$. Figure 8.3 shows the estimated x and y positions of the objects along with their true values. The simulated radar measurements and the clutter are also plotted. The OSPA distances of the filter are shown in Figure 8.4. The peaks in the OSPA distance are mainly due to cardinality changes and overall the OSPA distance is well under 0.5 m, which shows that the filter performs well. Moreover the CV and CTRV model probabilities of one of the pedestrians moving orthogonal to the ego-vehicle is illustrated in Figure 8.5. The pedestrian is assumed to be classified correctly by the classifier and the correct CV model is chosen all the time by updating the model transition probabilities according to the class-BBAs. The following subsections present the evaluation of the tracker with real-time sensor measurements. The test setup is same as described in Chapter 4.

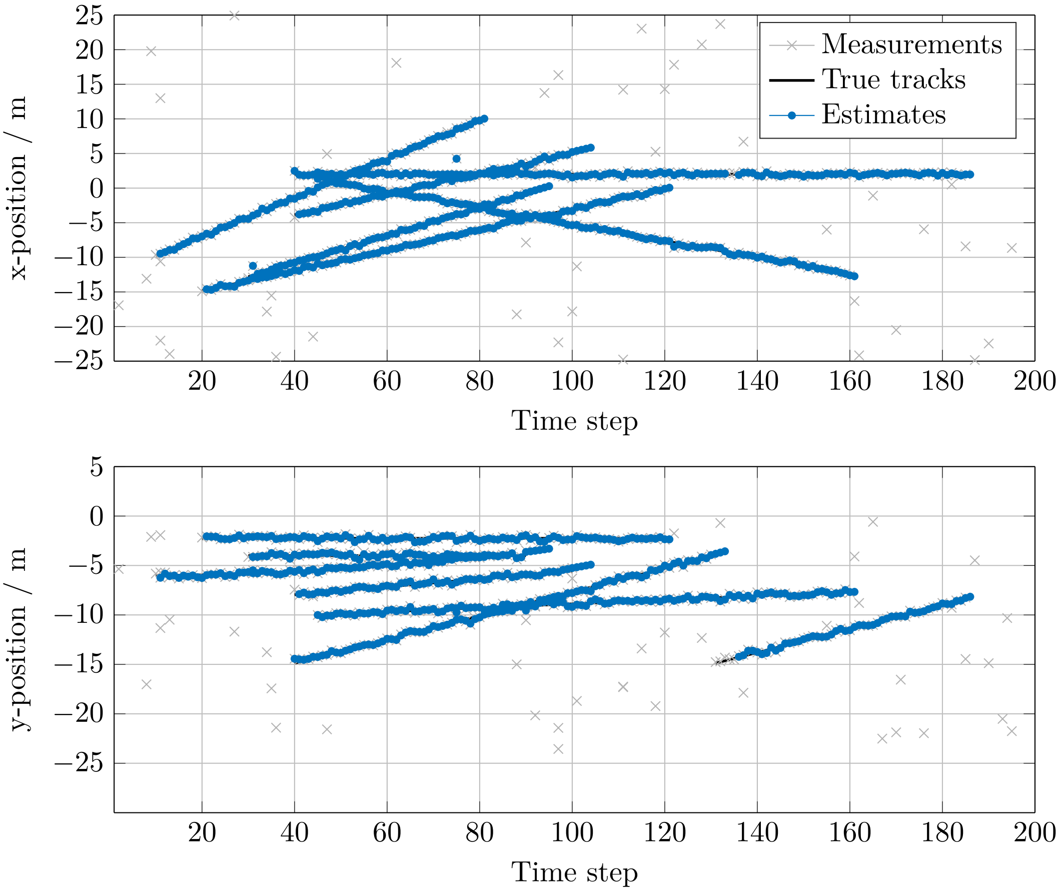


FIGURE 8.3: The top plot shows the x position estimates of the filter and the bottom plot shows the y position estimates for the simulation scenario. Simulated sensor measurements along with clutter are also plotted

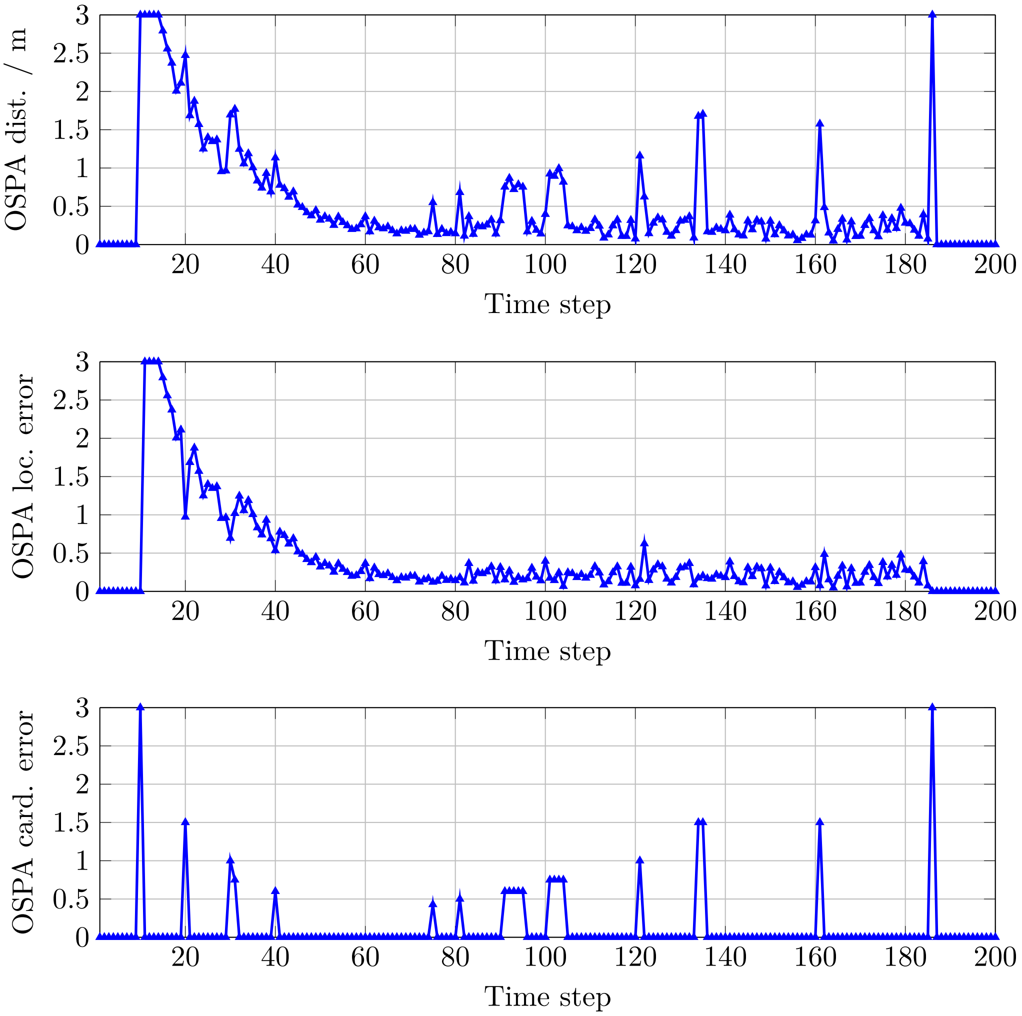


FIGURE 8.4: The first plot shows the OSPA distance with cut-off $c = 3$ m and order $p = 1$ for the simulated scenario. The second plot depicts the corresponding OSPA localisation and the third plot the OSPA cardinality.

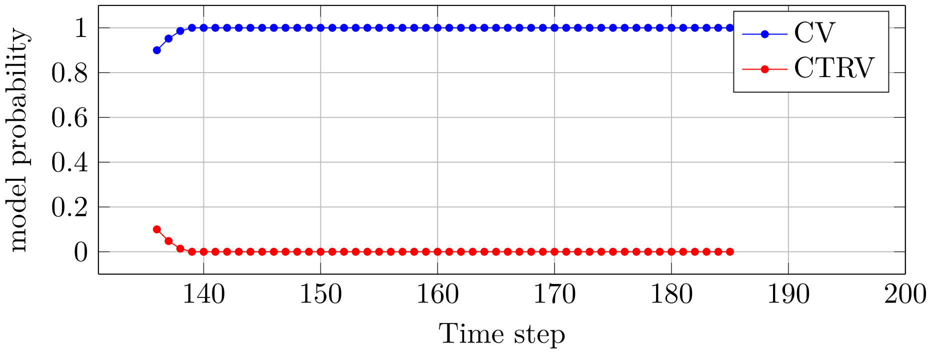


FIGURE 8.5: Probabilities of the CV and CTRV motion models for a pedestrian target moving away orthogonally from the ego-vehicle corresponding to the simulated scenario.

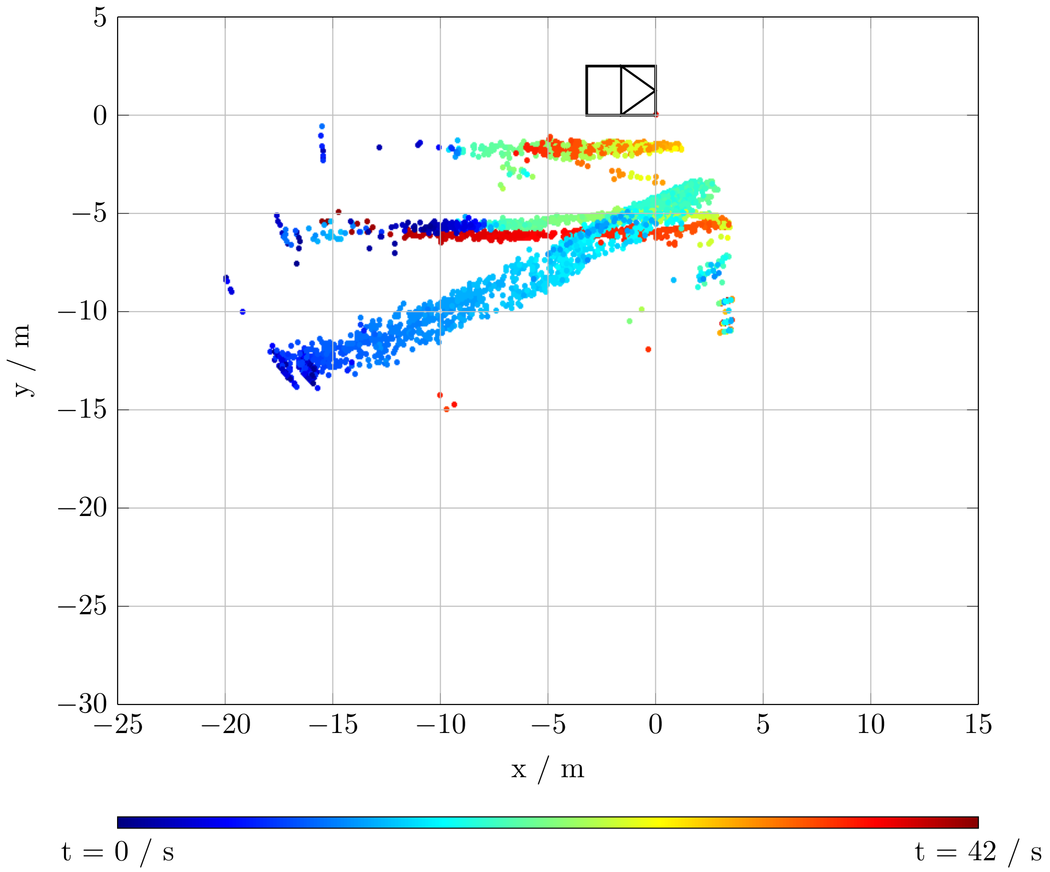
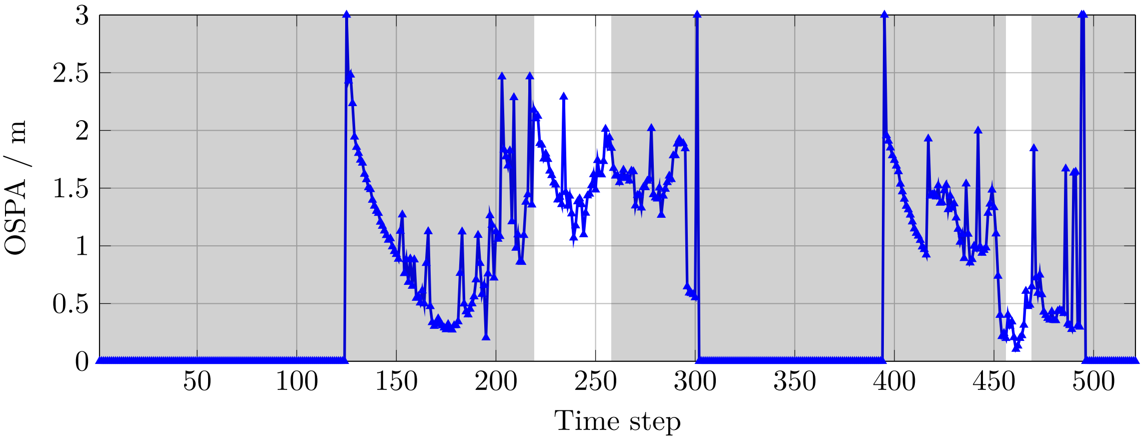


FIGURE 8.6: Radar detections from pedestrian, bicyclist and passenger car corresponding to the test sequence 1.

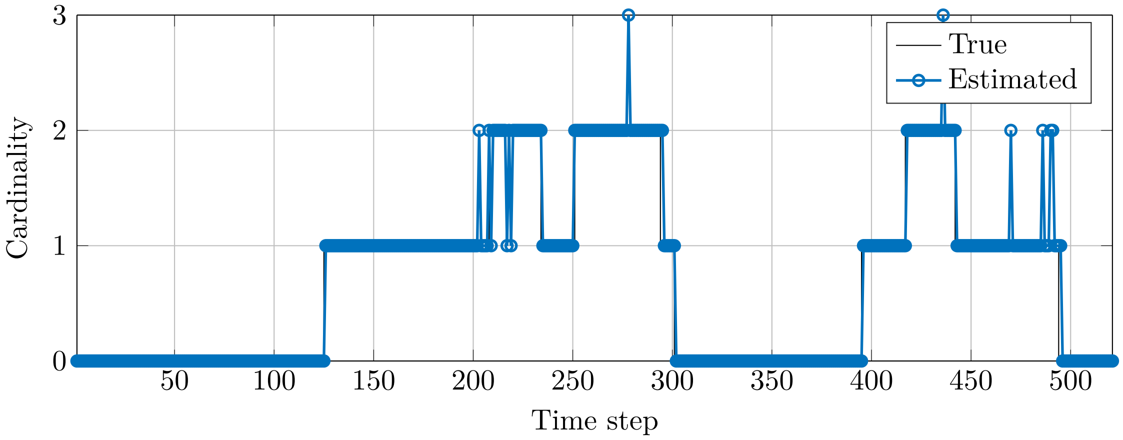
8.2.2 Test Scenario 1

In the first test scenario, a pedestrian, a bicyclist and a passenger car move in the sensor FoV area according to predefined trajectories. The passenger car is driven towards the ego-vehicle at a velocity of around 2.8 m/s, starting from a lateral distance of around 12 m and stops at front end of the FoV for a few time steps, before moving out. A bicyclist drives parallel to the ego-vehicle at a lateral distance of 5 m at a velocity of 2 m/s. A pedestrian moves in a straight line at a lateral distance of 1.5 m trying to maintain a constant velocity of approximately 1.5 m/s. Both the pedestrian and the bicyclist first move from the rear end of the ego-vehicle towards the front, exit the sensor FoV from the front end of the ego-vehicle and return back to move from the front towards the rear end. The test was performed in an area, where other static objects were also present in the background. The detections from the background static objects are pre-filtered before forming clusters. The raw background subtracted detections of the radar evolve as shown in Figure 8.6.

As described in the introduction of this Chapter, the ground truth reference is available only for certain area of the radar FoV, where it is overlapped by the FoV of the camera and the bright marker is detected. The reference for other time instances are approximated by predefined trajectory ground markings and velocity. Although not accurate, this would help in an overall analysis of the tracker performance through out the radar FoV. Moreover, the objects are defined to move in such a way that there is almost no occlusion of an object by



(A) OSPA distance



(B) Cardinality estimate

FIGURE 8.7: OSPA distance and cardinality estimate. The first plot shows the OSPA distance of scenario 1 with cut-off $c = 3$ m and order $p = 1$. The gray background in the plot illustrates time instances when the camera ground truth for bicyclist is not available. The second plot depicts the corresponding cardinality estimate.

another object, as this needs further considerations in the detections clustering and thereby makes the tracker analysis not straightforward.

The filter parameters used are same as in the simulation scenario and uses an adaptive birth distribution. The model probabilities for the CV and CTRV models are initialised as $[p(o)] = [0.5\ 0.5]$. The class-BBAs are discounted with a factor of 0.95, based on the class confidence estimates of the classifier. Figure 8.7b shows the cardinality estimate of the filter for the defined scenario. Cardinality ground truth is approximated by looking into measurements along with a synchronised video recording at every time step, for detections from the car, pedestrian and the bicyclist. According to the defined trajectory, if the car, pedestrian and the bicyclist are detected, the detections should lie within a lateral distance of 15 m. The car was always moving longitudinaly ahead of the bicyclist, and the bicyclist was always moving longitudinaly ahead of the pedestrian. Therefore, when there are no detections available for certain time steps from the objects, it can be approximated that the

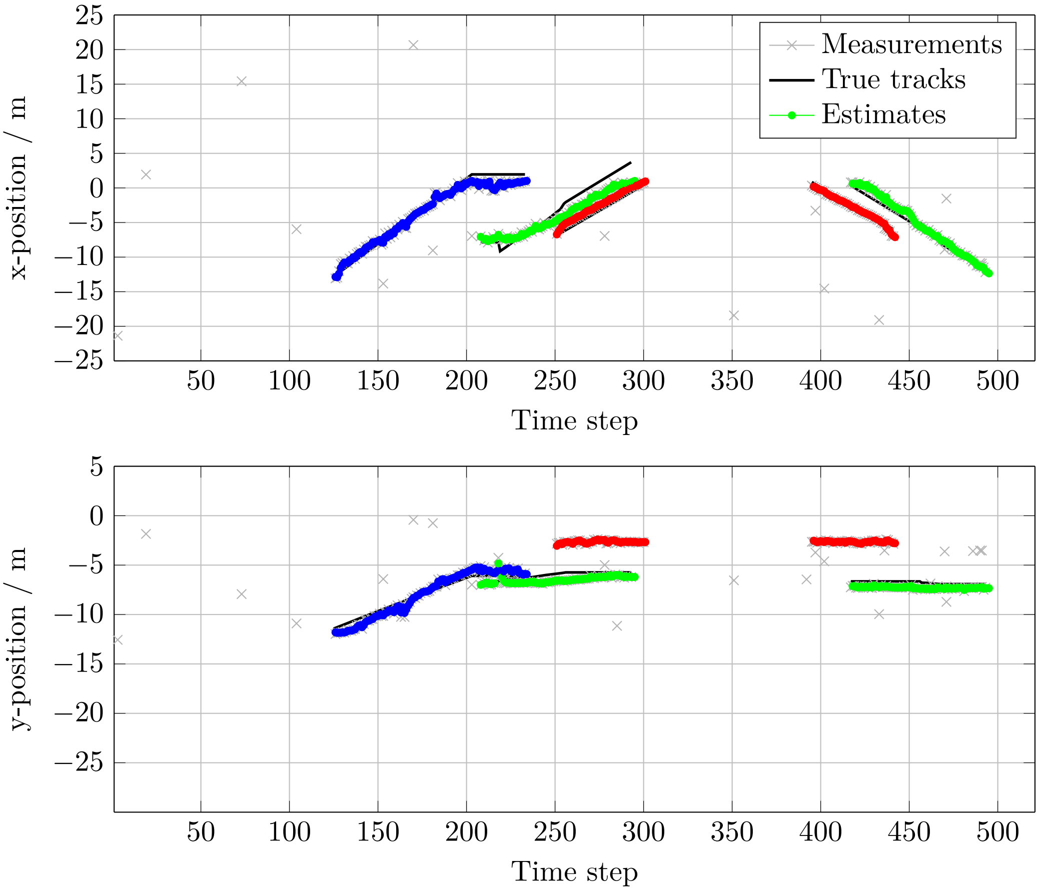


FIGURE 8.8: Top plot shows the estimated x positions of the objects and the bottom plot shows the estimated y positions corresponding to the test scenario 1. Radar detections and the approximated ground truth are also shown

target was not within the sensor FoV. Considering such an approximation, the cardinality estimate very well follows the true expected number of targets. At time step $k \approx 200$, $k \approx 270$ and $k \approx 470$ the cardinality deviation is due to a false track initialised from a clutter measurement, which is later terminated. The other deviations in cardinalities are due delayed track initialisation and termination, in either case as the probability of existence increases or reduces through certain time steps. The position and velocity estimates of the tracks are depicted in Figure 8.8 and 8.9 respectively. The corresponding OSPA distance for the scenario is shown in Figure 8.7a. The peaks in the OSPA distance are mainly due to cardinality errors. As the ground truth reference is only known approximately, the OSPA localisation is also an approximation. The time instances when the camera ground truth for bicyclist is not available are illustrated with a gray background in the plot. The ground truth values for these time instances are approximated based on the known test sequence. The OSPA distance is well below 1.5 m for most of the time instances.

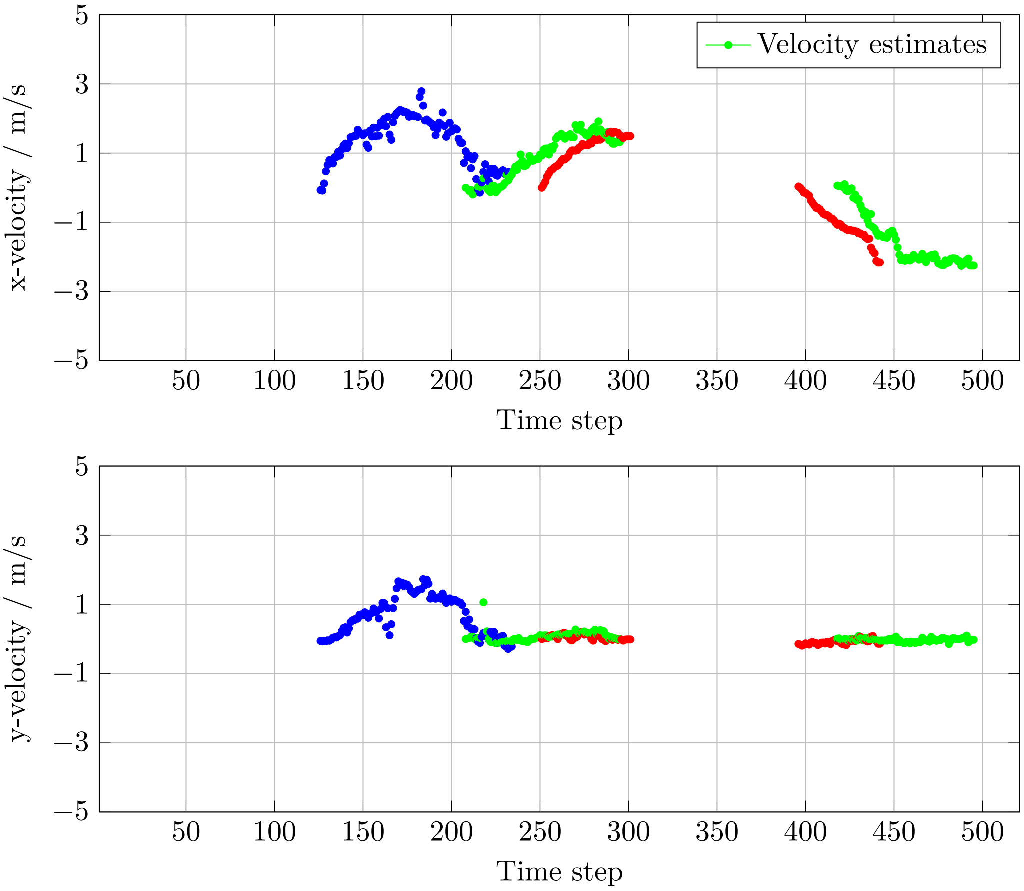


FIGURE 8.9: Top plot shows the estimated longitudinal relative velocities and the bottom plot shows the estimated lateral relative velocities corresponding to the test scenario 1.

8.2.3 Test Scenario 2

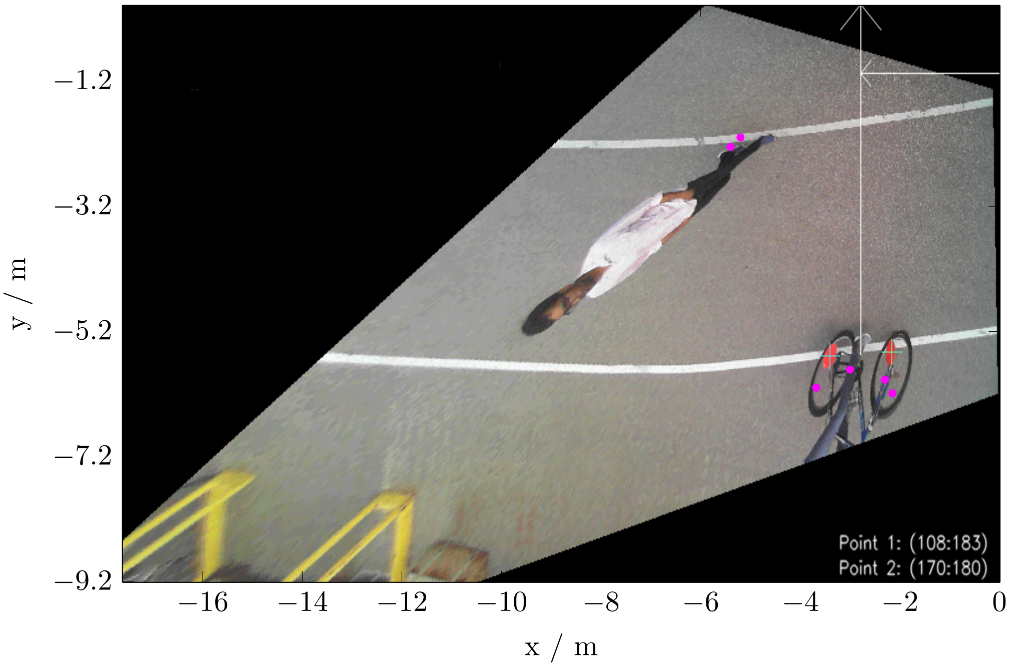
In the second test scenario, the ego-vehicle moves at a speed of 2.8 m/s, starting from a stationary position at the beginning of the test. The objects that move in the sensor FoV are a pedestrian and a bicyclist. Compared to the previous test scenarios, this test scenario is more challenging as it was performed in a yard, where there were also other large objects like door and container present in the background. The detections from the background are filtered out from the detections of the bicyclist and the pedestrian, before clustering. Figure 8.10 illustrates the radar detections of the pedestrian and bicyclist projected on to the calibrated camera image for the time instance $k = 16$ and the corresponding clusters. The pedestrian starts from the rear end of the truck and the bicyclist has a longitudinal and lateral separation of more than 3 m from the pedestrian. The filter parameters are the same as used for the previous test scenario.

The pedestrian is initially detected by the radar for certain time steps, but exists the FoV at around time step $k = 37$ as the vehicle moves forward faster than the pedestrian. The pedestrian starts walking faster and catches up with the ego-vehicle at $k = 83$ and starts walking again with almost constant velocity of 1.85 m/s. The bicyclist is within the sensor FoV for the complete test sequence and is tracked continuously. The OSPA distance and cardinality estimate are shown in Figure 8.11. Reference positions for the bicycle are available for the majority of the sequence. The reference for the pedestrian is not available. However based on the defined trajectory, at the time instances when the pedestrian is detected, the longitudinal positions are approximated by considering a nearly constant relative speed of 0.93 m/s with respect to the ego-vehicle and the lateral position at 1.5 m. The time instances when the reference values from the camera were not available are illustrated again by gray background in the plot.

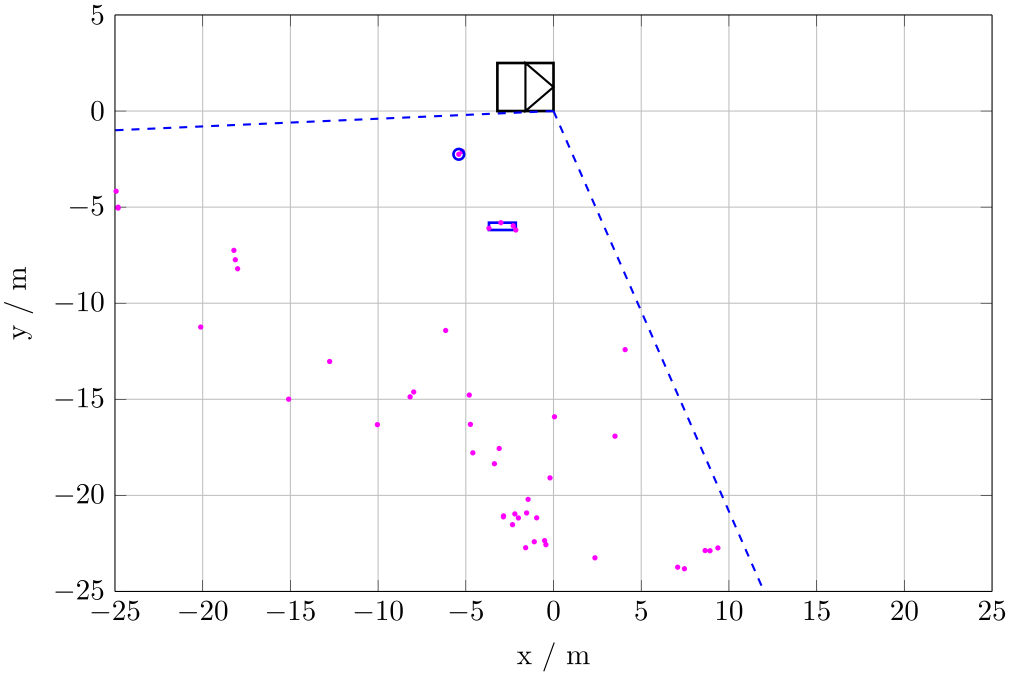
Peaks and higher deviations in OSPA distances are due to the change in cardinality and approximated reference for the pedestrian. The estimated positions of the objects as they start moving are depicted in Figure 8.12 and the relative velocities in Figure 8.13. The strong deviation of the bicyclist track at $k = 72$ is due to the imprecise clustering of detections. The reflections from a metal container in the background at this time point is not separable from the bicyclist detection resulting in a wandering RP. The RMSE values of the bicyclist using the camera detection as reference is shown in 8.14. The reference velocity is derived from the camera reference positions by smoothing. The ground truth point, when available, is transformed to the object RP for calculating the RMSE. Time instances when the ground truth were not available are again indicated by a gray background in the RMSE plot as well. Apart from the higher values due to approximated ground truth and imprecise clustering at $k = 72$, the RMSE are low with a mean position error of 0.896 m and mean velocity error of 0.5644 m/s. As the pedestrian most of the time remains towards the rear end of the truck, generates only one or two detection points and it's class-BBA is majorly from the track velocity part. Consequently, the pedestrian track is smoother than the bicyclist track as it is most of the time tracked as a point target in this scenario. The model probabilities of the bicyclist track are depicted in Figure 8.15.

8.2.4 Test Scenario 3

The third test scenario involves two pedestrians walking in the sensor FoV. Two pedestrians walk in opposite direction repeatedly towards and away from the ego-vehicle. The scenario

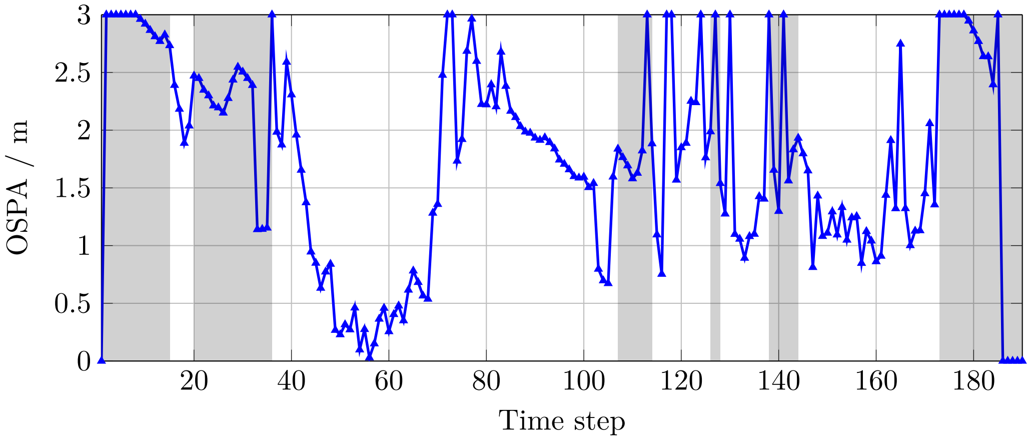


(A) Background subtracted detections projected on to the ground truth camera image.

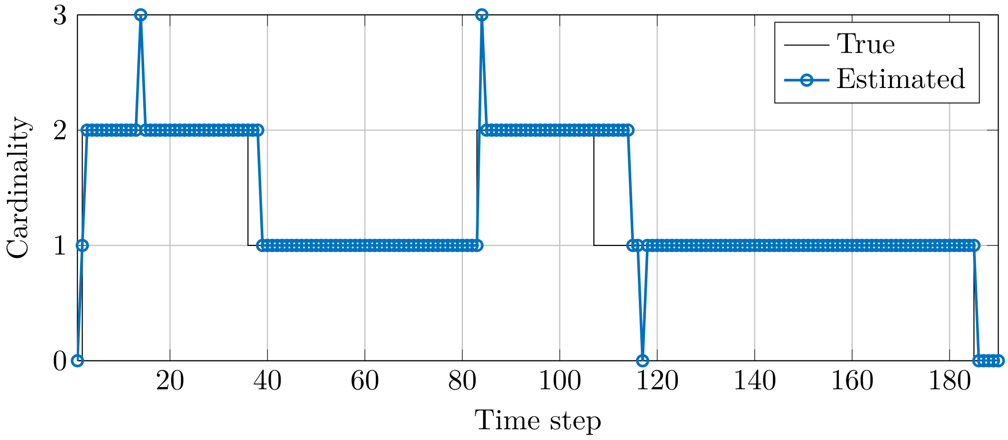


(B) Radar detections and clusters

FIGURE 8.10: Figure (A) depicts the detections of the pedestrian and the bicyclist projected on to the camera image depicted by magenta points corresponding to test scenario 2. Green cross illustrates the marker position detected from the camera. The right top corner of the image is the origin of the radar. All the detections, including also the ones from background objects are shown in Figure (B). The origin of the radar in Figure (B) is (0,0). The cluster of bicyclist and pedestrian are illustrated as blue bounding box and circle, respectively.



(A) OSPA distance



(B) Cardinality estimate

FIGURE 8.11: Figure (A) shows the OSPA distance. Again the gray background in the plot illustrates time instances when the camera ground truth for bicyclist is not available. Figure (B) shows the cardinality estimates corresponding to test scenario 2.

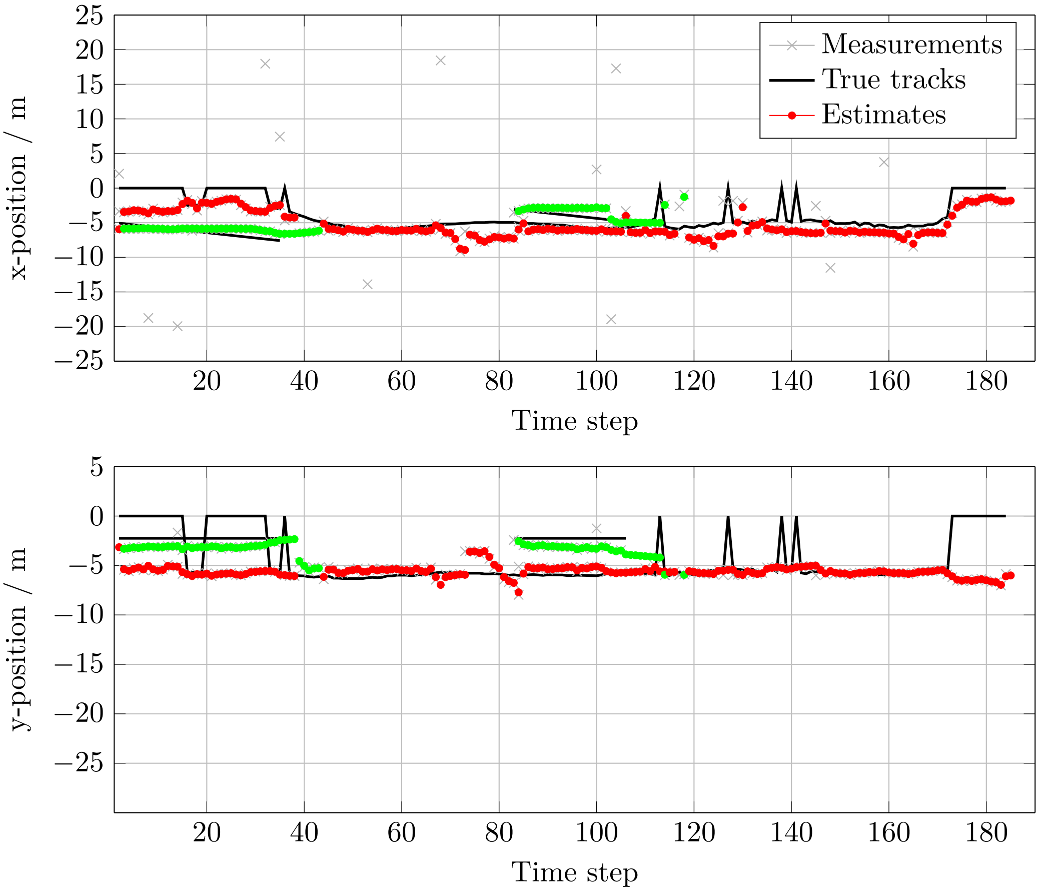


FIGURE 8.12: Top plot shows the estimated x positions of the objects and the bottom plot shows the estimated y positions corresponding to the test scenario 2. Radar detections and the approximated ground truth are also shown.

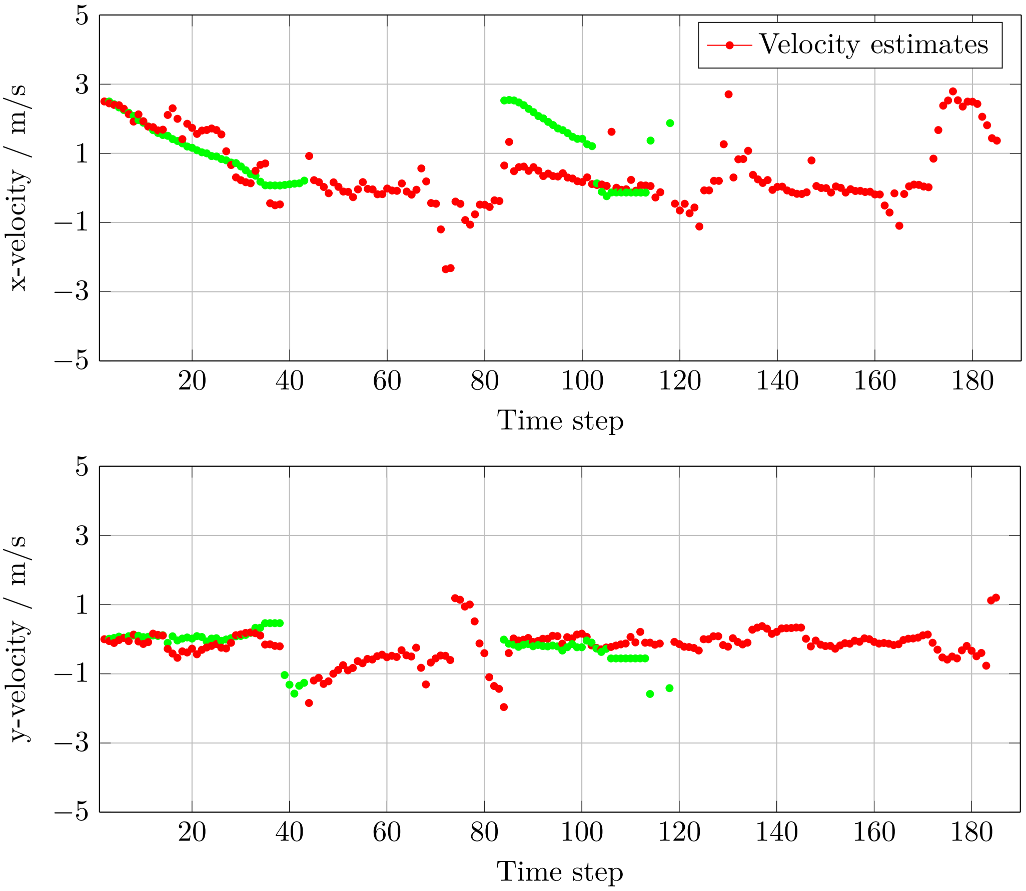
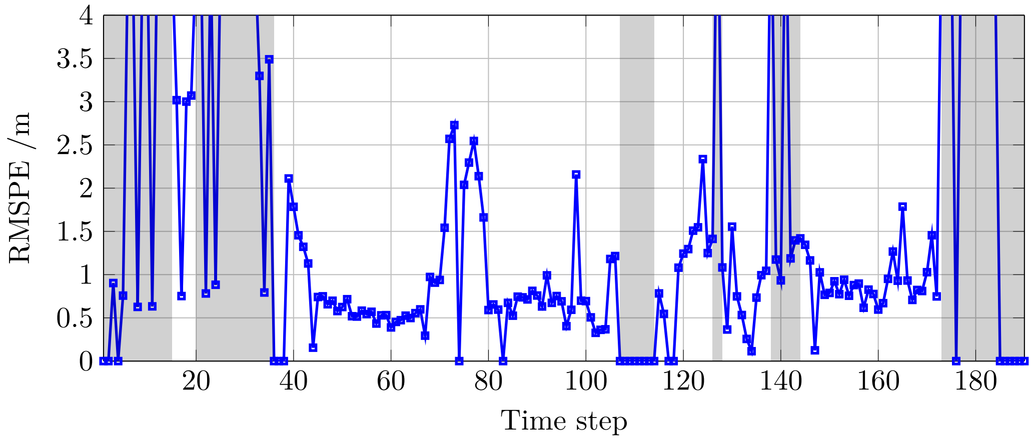
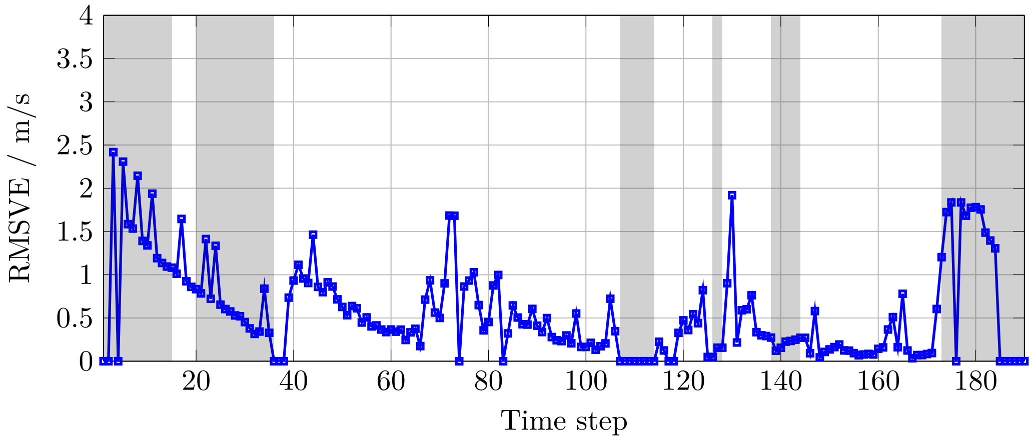


FIGURE 8.13: Top plot shows the estimated longitudinal relative velocities and the bottom plot shows the estimated lateral relative velocities corresponding to the test scenario 2.



(A) RMS position error



(B) RMS velocity error

FIGURE 8.14: RMS position and velocity errors of bicyclist corresponding to test scenario 2.

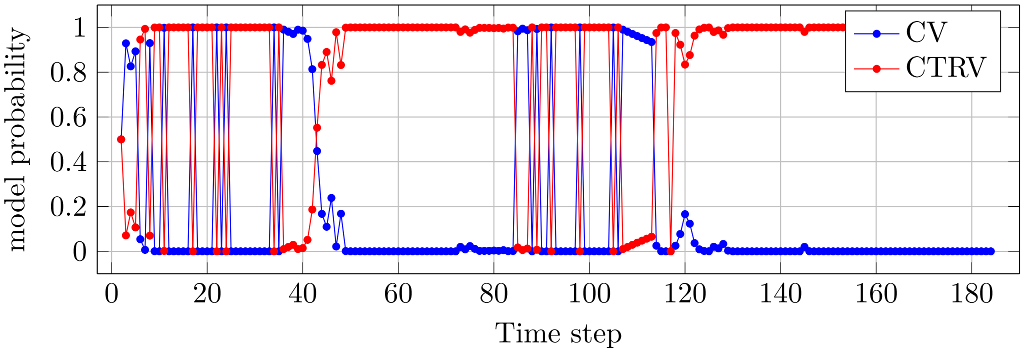


FIGURE 8.15: Model probabilities of the bicyclist track corresponding to test scenario 2.

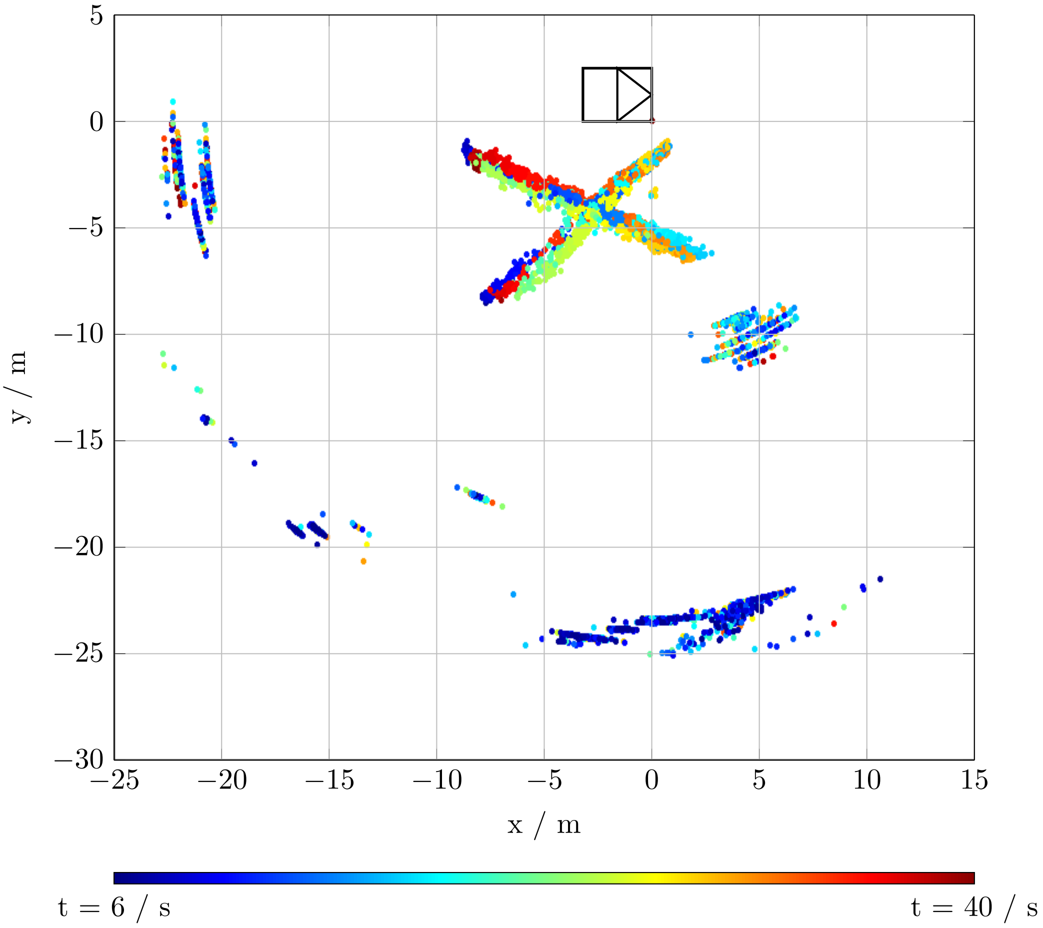
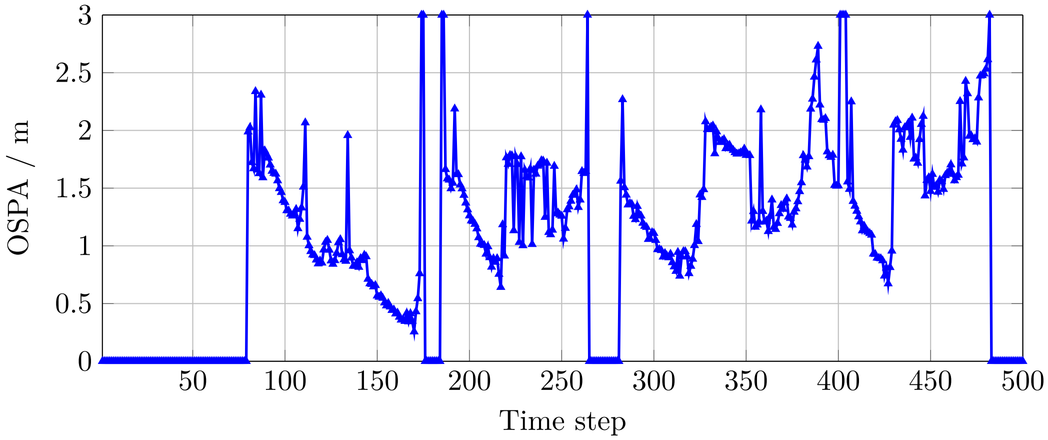


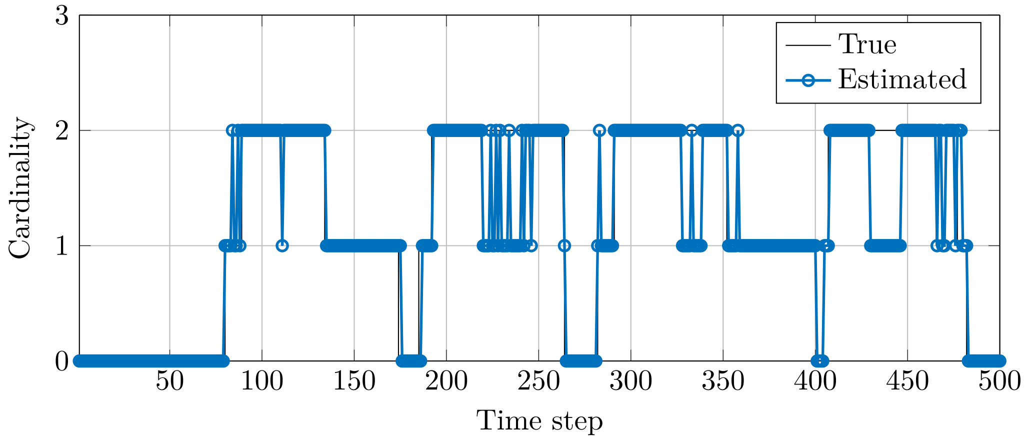
FIGURE 8.16: Radar detections from two pedestrians corresponding to the test sequence 3.

involves occlusion of a pedestrian by another pedestrian for a short time, when they cross each other. The pedestrians exit and enter the FoV several times. There were also other objects like metal doors, container and other trucks present in the sensor FoV during the test. Time evolution of the raw radar detections are illustrated in Figure 8.16. The parameters of the filter are same as of test scenario 1. As no ground truth reference for object states and cardinality are available, they are approximated by labeling the detections synchronised with the video recorded during the test.

Figure 8.17b depicts the estimated and approximated expected cardinality of the test scenario. The deviation in the estimated cardinality is mainly due to two reasons. First, when the pedestrians are too close to each other, detections belonging to each of the pedestrians are clustered together as a single cluster. This causes a deviation in the estimated cardinality, for example between $k = 220$ to $k = 233$, which is mainly due to the disadvantage of detections clustering. The pedestrians could not be spatially separated as two individual clusters. Another cause of deviation in the estimated cardinality is due to the occlusion of a pedestrian by another pedestrian when crossing each other, which can be observed in the same plot from $k = 234$ to $k = 246$. As the GM implementation does not facilitate the usage of state dependent detection and survival probabilities, track of the occluded pedestrian is deleted in case the occlusion lasts for certain number of time steps. However, the lost track



(a) OSPA distance



(b) Cardinality estimate

Figure 8.17: OSPA distance and cardinality estimate corresponding to test scenario 3.

is re-initialised, once the detection clusters are spatially well separable again.

Figure 8.17a depicts the OSPA distance with respect to the approximated ground truth values. Estimated position and velocities of the tracks are shown in Figure 8.18 and Figure 8.19 respectively. The major peaks in OSPA distances are mainly due to the cardinality errors. Apart from the cardinality errors due to occlusion and unresolved detection clusters, the pedestrians are tracked continuously.

8.2.5 Performance Evaluation

As the ground truth solution is directly available only for the bicyclist class, the overall performance of the designed tracker is evaluated from test scenarios involving bicyclist. However, the camera based ground truth is not available at all time steps. Therefore only the time instances where the reference ground truth from the camera were available are considered for the performance evaluation. As described in section 8.1.2, the consistency of the filter can be evaluated with the NEES metric. Figure 8.20 shows the NEES in estimated position and velocities from a certain time sequence of test scenario 2 for the

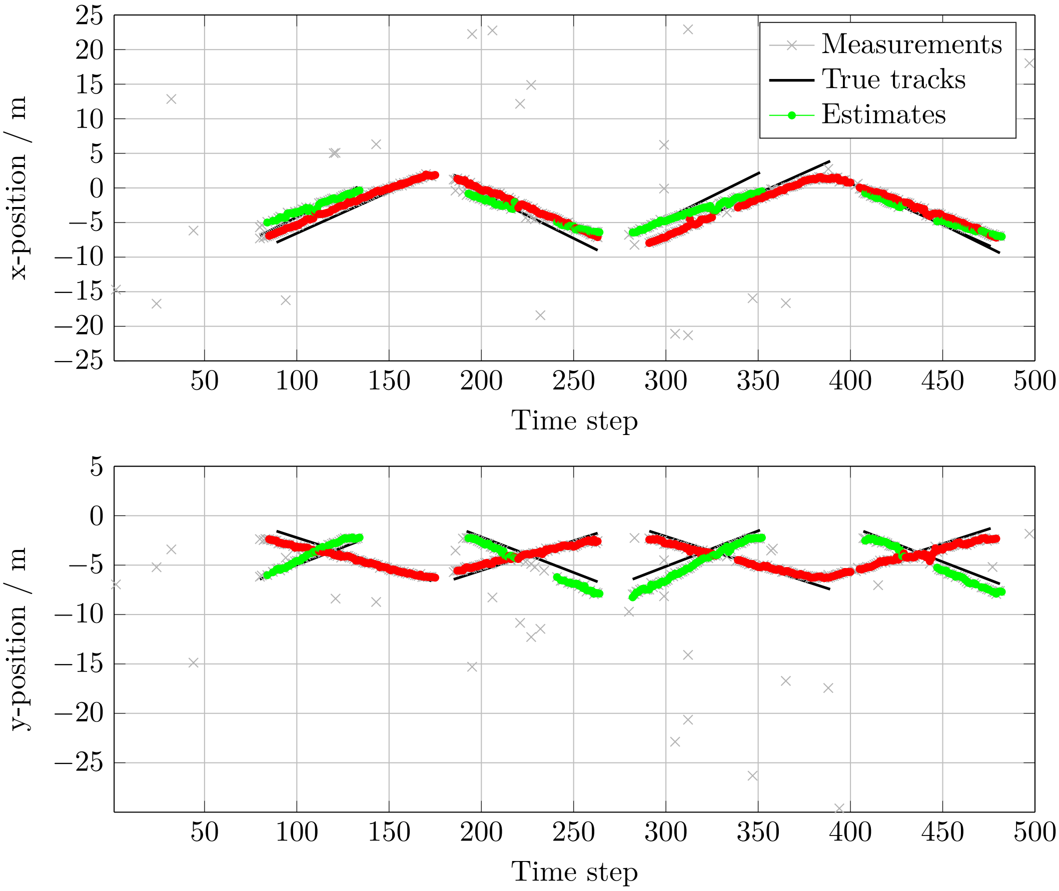


FIGURE 8.18: Top plot shows the estimated x positions of the objects and the bottom plot shows the estimated y positions corresponding to the test scenario 3.

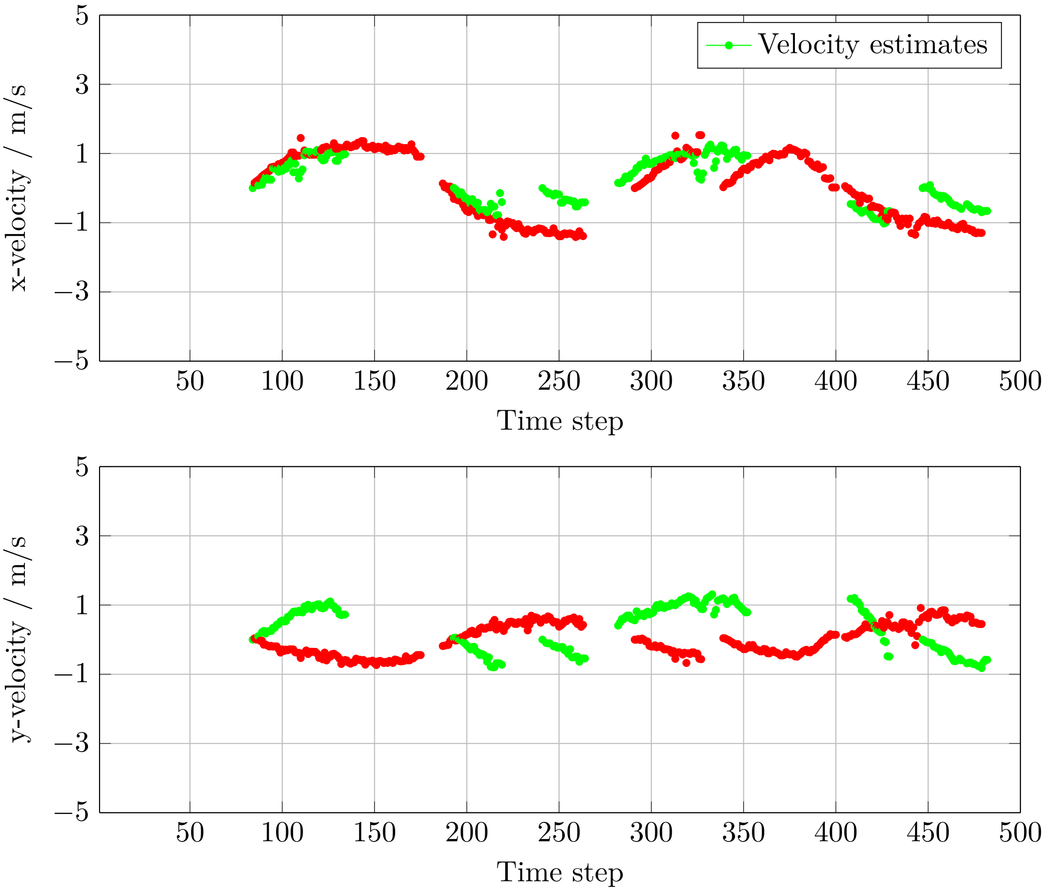


FIGURE 8.19: Top plot shows the estimated longitudinal relative velocities and the bottom plot shows the estimated lateral relative velocities of the pedestrians corresponding to the test scenario 3.

implemented GM-MMLMB filter with integrated classification. Subsequently, the NEES can be compared to a χ^2 distribution with 4 degrees of freedom. Considering a significance level of $\alpha = 0.05$, the filter can be understood to be consistent as NEES majorly lies within the interval $[0.48, 11.14]$ corresponding to the χ^2 distribution table.

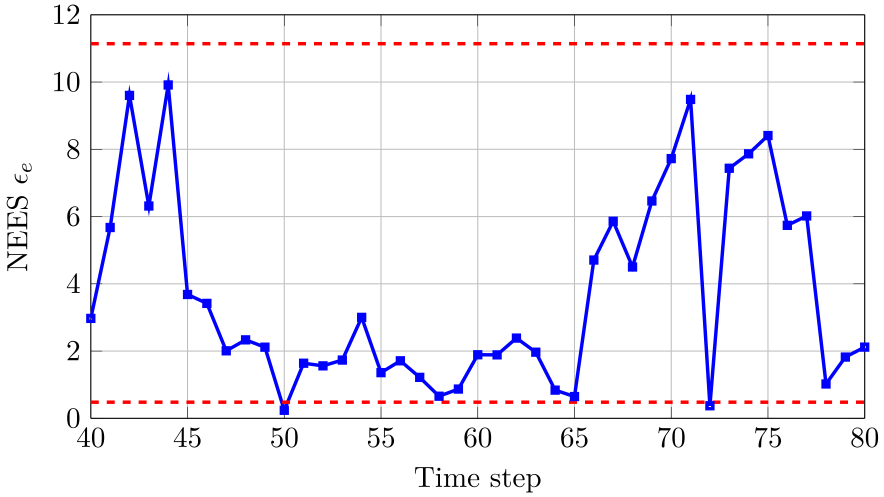
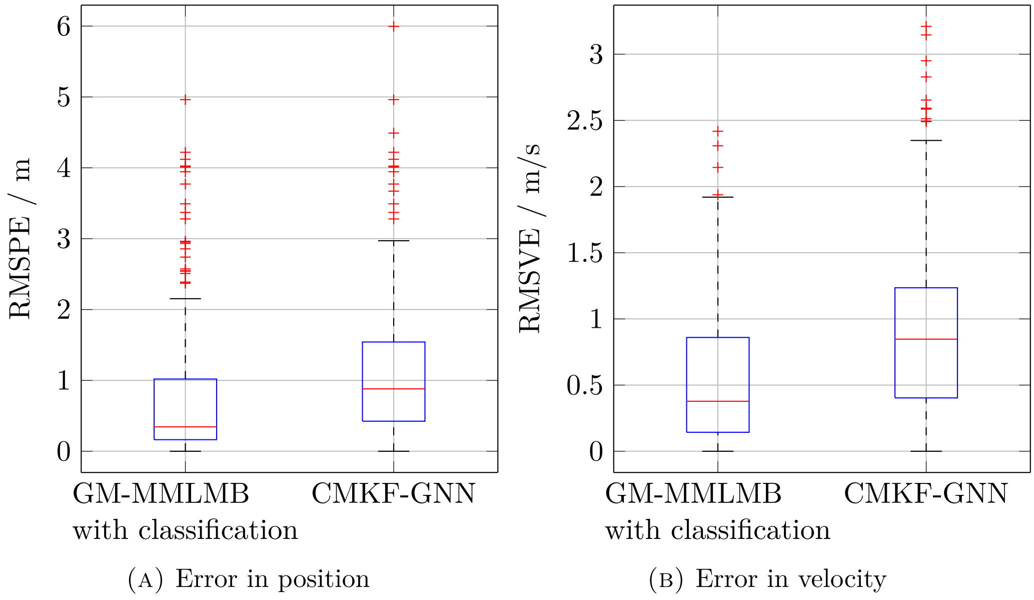


FIGURE 8.20: NEES consistency metric for a bicyclist state estimation sequence.

Additionally, the accuracy of state estimation is analysed with the help of multiple test scenarios. Different tests with varied maneuvers, distances and conditions were performed with the bicyclist. The tests covered the relative distances of the bicyclist to the ego-vehicle from 0.5 m to 8 m and velocities upto 5.6 m/s. The tests also covered both the parallel and radial movement of the bicyclist with respect to the ego-vehicle. Moreover, the basic performance of the filter for these scenarios in terms of state estimation is compared with an own implementation of converted measurements Kalman filter (CMKF) considering a CV model and GNN algorithm for data association. RMSE of position and velocity estimates are computed for all the tests and time instances when the reference ground truth from the camera are available. Figure 8.21 depicts the box-plot of RMSE position and velocity values of different tests scenarios. Given the simplicity of the test scenarios in which there were no additional objects and the limited availability of reference values, the estimation of the filters appear to be same. For the GM-MMLMB filter with integrated classification, the deviation in the estimated position has a median value of 0.345 m and the deviation in estimated velocity has a median value of 0.377 m/s, demonstrating good accuracy. The extremities in the deviation values can be understood as the effect of improper clustering and imprecise reference values. However, the deviation in CMKF-GNN filter is higher with a median of 0.88 m for position and 0.846 m/s for velocity. The higher deviation at some instances with CMKF-GNN filter are when the track gets associated with clutter measurement and subsequently resulting in track discontinuities. Also due to usage of basic M/N rule for track management, the number of false tracks initialised in the CMKF-GNN filter is more, causing the measurement actually belonging to a confirmed track being stolen by the false track, resulting in track loss.



(A) Error in position (B) Error in velocity

FIGURE 8.21: RMSE values for bicyclist test scenarios corresponding to the GM-MMLMB filter with integrated classification and CMKF-GNN filter.

8.3 Classification Evaluation

The metrics used to analyse the classification performance are described in the first part of this section. The second part presents the evaluation of object classification based on SVM and DST combination of track velocity described in section 6.3.

8.3.1 Metrics

A very well known representation of the classifier performance is the class confusion matrix. Before constructing the confusion matrix, defining other metrics like true positive (TP), true negative (TN), false positive (FP) and false negative (FN) are essential. TP is the case when the predicted object class and the actual object class are the same. TN is the case where the object actually doesn't belong to a particular class and is correctly classified as not belonging to that class. FP is when the the object is classified as belonging to a certain class, when it actually doesn't belong to that class. FN is the case where the object belongs to a certain class but is not classified as belonging to that class. A confusion matrix is then constructed with these values for the different object classes in consideration. Additional metrics which can be used for evaluating the performance of a classifier are

1. *Accuracy:* Denotes the the probability of correct classification and is given by

$$Accuracy = \frac{\sum TP + \sum TN}{P + N} \tag{8.12}$$

2. *Sensitivity:* Represents the true positive rate (TPR) of the classifier and is given by

$$Sensitivity(TPR) = \frac{\sum TP}{TP + FN} \tag{8.13}$$

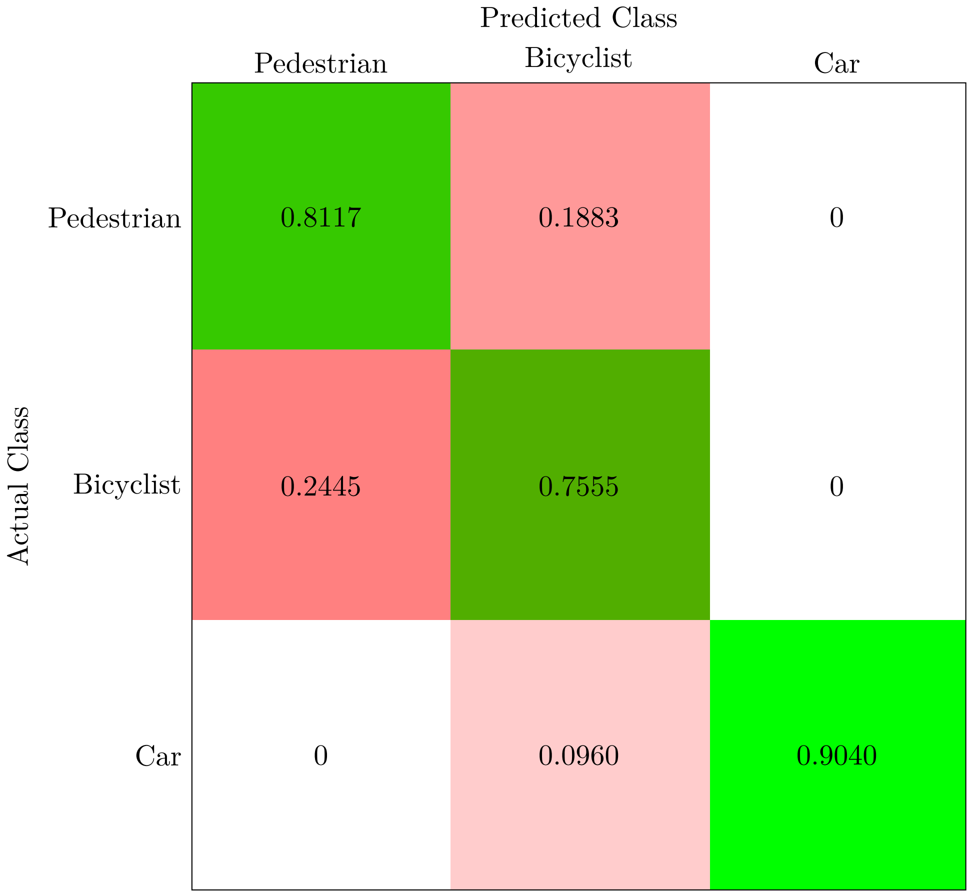
3. *Fall-out(1−specificity):* Represents the false positive rate (FPR) of the classifier and is given by

$$Fall\text{-}out(FPR) = \frac{\sum TP}{FP + TN} \tag{8.14}$$

The above defined metrics are basically for binary classification problem. In case of multiclass problem, the positive statistics are calculated by summing over the rows and the negative statistics by summing over the columns of the confusion matrix.

8.3.2 Classification Performance

More than 8000 samples of labeled cluster data are used for the purpose of classifier evaluation. Apart from clustering and feature extraction, the reference points of the cluster samples were also tracked by the designed filter, which enables the integration of the track velocity by DST. The confusion matrix of the classifier for the pedestrian, bicyclist and car classes are depicted in Figure 8.22. The rows of the confusion matrix represent the true class of the object and the columns represent the predicted class. The diagonal elements of the confusion matrix represent the TPs of that specific class and the corresponding row elements represent the FNs.



	Pedestrian	Bicyclist	Car
Pedestrian	0.8117	0.1883	0
Bicyclist	0.2445	0.7555	0
Car	0	0.0960	0.9040

FIGURE 8.22: Confusion Matrix representing the results of radar measurement and track based object classification. Numbers represent the probability values.

The car class has the highest probability of TPs, where it is classified correctly for 90.4% of the time. The confusion of the car class with pedestrian class is 0 %, whereas there is a small confusion of 9.6 % with the bicyclist class because the bicyclist samples include velocities overlapping with the car class. The pedestrian class is classified correctly 81.17 % of the time and the bicyclist is classified correctly 75.55 % of the time. There is relatively larger confusion between the pedestrian and the bicyclist class because many of the features from radar detections are identical to both of the classes, although the velocity difference is wider. As both pedestrians and bicyclists are relevant objects for most of the accident avoidance applications like an urban turning assist, it is important to reduce the false alarms due to irrelevant objects like a passenger car. This requirement can very well be satisfied by the developed classifier. The accuracy, sensitivity and fall-out of the classifier for each of the class are given in Table 8.1.

Class	Accuracy	Sensitivity	Fall-out
Pedestrian	0.8306	0.8117	0.1584
Bicyclist	0.8092	0.7555	0.1533
Car	0.9783	0.9040	0

TABLE 8.1: Classification performance for each class based on accuracy, sensitivity and fall-out.

Chapter 9

Conclusion

9.1 Summary

Because of the size and load of a commercial truck, an accident involving trucks even at low speeds can cause severe impacts. Compared to a passenger car, a truck driver usually has bigger blind spot areas around the truck. In urban scenarios, the collision risk of a truck with bicyclist or pedestrian is higher especially during turning or reversing maneuvers The survival rate of bicyclists and pedestrians is very low when met with an accident against the truck. Moreover the environment surrounding the truck in such situations are very complex where different categories of object move with different characteristics.

Environment perception solutions for application in commercial truck accident avoidance systems and future automated driving is developed in this work. The objective of the work was to develop robust methods for tracking and classification of VRUs in the near the vicinity of the truck, based on which the safety system can react to prevent accidents. In a first step to satisfy this objective, analysis of state-of-art multi-object tracking methods and their drawbacks along with related works in the field of vehicle environment perception has been performed. The unified approach in multi-object tracking based on RFS, expected to overcome the data association uncertainties in conventional target tracking methods have been studied and presented. The recently developed LMB filter based on labeled RFS has been used as the basis for tracking VRUs around the truck. Gaussian mixture implementation of MMLMB filter has been parametrised with multiple motion models to track pedestrians and bicyclists in the surrounding of the truck. High-resolution radar has been used as the primary sensor for detecting objects. The radar measurement characteristics are analysed in detail for a better parametrisation of the state estimator.

A rapid prototype system is setup on the truck for a real-time realisation of the developed approaches. Thereby, communication structure for transmission of radar detections, vehicle signals and measurement acquisition have been defined based on CAN and ethernet networks. A method for integrating radar detections based class information into MMLMB tracking framework is presented. High-resolution radar is capable of detecting multiple points of an object. As an initial step for deriving the object class based on radar measurements, density based clustering method has been used to group the radar detections originating from the same object. The cluster groups contain many useful informations about the object properties, like the number of reflection points, dimensions and Doppler variance etc. Machine learning based classifier has been constructed by training it with different features of the detection clusters, belonging to different classes of objects like pedestrian, bicyclist and passenger car. The trained classifier is able to predict the class of an object in the environment, whose identity is not known beforehand. The measurement based class information of the object helps in selecting an appropriate motion model for it, so that it can be tracked precisely. The model probabilities for predicting the motion of an object in MMLMB filter are adapted according to the measurement based predicted class of the object. Fusion of information from estimated state of the track with the measurement based class knowledge using DST helps in a reliable prediction of the objects' class. The safety application can then warn the driver or take decisions based on these classified tracks. For example, in a city turning assist application for trucks, the driver will be warned if the object in the blind spot is a pedestrian or bicyclist and other objects can be suppressed to reduce false warning rate. In the end, the developed approaches have been evaluated with simulation and various real-time test scenarios which show promising results. Also, it shown that the presented methods have better performance and accuracy compared to a conventional CMKF-GNN filter in similar environment scenarios.

9.2 Future Work

In order to mature the environment perception method presented in this work for implementation in series commercial vehicle safety systems, further intensive testing and validation are required. Collection of many relevant measurement sequences from public roads through durability runs, especially from dense urban scenarios are needed for analysis and improvement of tracking and classification performance. Moreover, the developed tracking and classification method depends highly on the efficiency of detections clustering. A geometric model is fitted to each of the cluster and a reference point for the object is chosen based on the best fitting shape. In situations when the object is far away from the ego-vehicle, the radar sensor is not capable of capturing the extension of the object and the number of detections received would be very less to extract object properties by clustering. Also in case of higher clutter measurements, clustering would give unreliable information about object extent. This has a higher influence on the object classification and eventually tracking itself. Therefore alternative methods for modeling extended objects are required. Methods to address this problem are already proposed in [Koc08, GLO10, GLO11, LGO13] and [SGM^{+}14, SKRD16]. Observation models for a passenger car are proposed in [SKRD16] and methods for assuming number of detection points from a target are suggested in [GLO10]. However, efficient implementation for a wider class of objects indicate an essential direction for future research. Another typical problem in tracking multiple objects is the object occlusion. Robust approaches for handling occlusion of objects by other moving or stationary objects would improve the overall tracking stability. Further, the integrated class information can also be used to make more accurate assumptions on number of detections returned by the object and also to adapt the parameters of measurement models, according to the class of the object. Also with the development of advanced sensor and mapping technologies, development of consistent methods for fusion of information from various sources would increase the overall stability and robustness of the ADAS and automated driving systems.

Appendix A

Fundamentals of State Estimation and Conventional Object Tracking

The theoritical foundations of object state estimation and multi-object tracking are presented in this Appendix, which can be further referred in detail in [BP99, BSWT11, Mah07b, Reu14]. The concepts of Bayesian state estimation and the recursive Bayes filter for single-object tracking are presented in section A.1. The section following that presents the Kalman filter, which is a linear closed-form implementation of the recursive Bayes filter based on Gaussian distribution. Various versions of the Kalman filter which can handle non-linear object motions and assumptions of different motion models and are presented from sections A.2.2 through A.2.3. Section A.3 describe approaches for realising multi-object tracking based on single-object Bayes filter and the additionally required data association methods. The last section of this Appendix outlines the methods for integrated object existence and state estimation.

A.1 Bayes Filter

According to the probability theory, the probability that a random variable X takes a value x can be denoted as $P\left(X=x\right)$. In other words, it can be denoted as the probability of occurence of the event x given as $p(x)$. Consequently, the probability of non-occurence of the event x is given as $1-p(x)$. Considering a range of values x_1 to x_2 the continuous random variable X can take, the probability can then given by the integral over a function of the variable X called the probability density function (PDF).

Considering two random variables X and Y, the joint probability distribution can be given as

$$p(x,y)=P(X=x,Y=y) \tag{A.1}$$

In case X and Y are independent random variables, i.e. if the events $X=x$ and $Y=y$ are independent events for all x and y, the joint probability distribution takes the form

$$p(x,y)=P(X=x)P(Y=y)=p(x)p(y) \tag{A.2}$$

Moreover the probability that the event $X=x$ occurs, given the event $Y=y$ has already occured and the probability $P(Y=y)\neq 0$ can be given by the relation of conditional probability as

$$p(x|y)=\frac{P(X=x)P(Y=y)}{P(Y=y)}=\frac{p(x)p(y)}{p(y)} \tag{A.3}$$

Given the commutative property $p(x, y) = p(y, x)$, and substituting the relation of conditional probability $p(y|x)$ in (A.3) with the condition $p(y) \neq 0$, gives the Bayes' theorem

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{A.4}$$

The above equation means that, given a prior probability $p(x)$ and a likelihood function $p(y|x)$, the posterior probability $p(x|y)$ can be calculated.

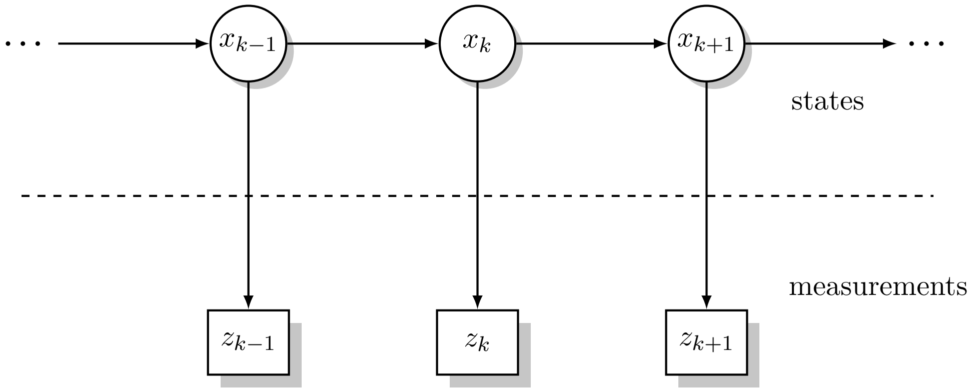


FIGURE A.1: Dynamic Bayesian Network.

The Bayes' theorem can be applied to the problem of state estimation by considering the object state x and the corresponding sensor measurements z as the random variables. In case of a dynamic system, the object state changes with time and so as the measurement originating from it. Therefore a time index k can be given to the state x_k and measurement z_k. For a discrete case, the motion of the object between two time steps k and $k+1$ can be modeled with a transition function as

$$x_{k+1} = f(x_k, v_k) \tag{A.5}$$

where v_k denotes the process noise representing uncertainity of the motion model. The measurement can be represented by a function which transforms the corresponding object state into measurement space as

$$z_{k+1} = h(x_{k+1}, w_{k+1}) \tag{A.6}$$

where w_{k+1} represents the measurement noise. The probability density of a state x_{k+1} at time $k+1$, given the states from previous time steps $x_{0:k}$ and the sequence of measurements $z_{1:k}$ can be expressed by the transition density $p(x_{k+1}|x_{0:k}, z_{1:k})$. Similarly, the probability density of a measurement z_{k+1}, given the time sequence of object states $x_{0:k+1}$ and measurements $z_{1:k}$ can be given by the likelihood $p(z_{k+1}|x_{0:k+1}, z_{1:k})$. Assuming the state x to be an unobserved Markov process and measurements z as observed states of hidden Markov model (HMM) as depicted in Figure A.1, the probability of the current state x_{k+1} would only depend on the previous state x_k and the probability of current measurement z_{k+1} would only depend on the current object state x_{k+1}. Therefore the corresponding probability densities can be given as

$$p(x_{k+1}|x_{0:k}, z_{1:k}) = f(x_{k+1}|x_k) \tag{A.7}$$
$$p(z_{k+1}|x_{0:k+1}, z_{1:k}) = g(z_{k+1}|x_{k+1}) \tag{A.8}$$

which are called as Markov transition density and measurement likelihood function respectively.

The continuous estimation of the object state through time is then achieved by recursive prediction and update, which forms the Bayes filter. The posterior or updated probability density of the object state x_{k+1} at time $k+1$ can be given as the probability density of the state after considering also the measurement z_{k+1} and is given as

$$p_{k+1}(x_{k+1}) \triangleq p_{k+1}(x_{k+1}|z_{1:k+1}) \tag{A.9}$$

When the measurement z_{k+1} was not yet available, the above probability density of the object state x_{k+1} is then called as the a priori or predicted probability density, where the prediction is given by Chapman-Kolmogorov equation. Consequently, the prediction and update equations of the Bayes filter are given as

$$p_{k+1|k}(x_{k+1}|z_{1:k}) = \int f_{k+1|k}\,(x_{k+1}|x_k)\, p_k(x_k|z_{1:k})dx_k \tag{A.10}$$

$$p_{k+1}(x_{k+1}) = \frac{g(z_{k+1}|x_{k+1}) \cdot p_{k+1|k}(x_{k+1})}{\int g(z_{k+1}|x_{k+1}) \cdot p_{k+1|k}(x_{k+1})dx_{k+1}} \tag{A.11}$$

Therefore the recursion of a single-object Bayes filter can be illustrated as

$$\begin{tikzcd}
\cdots \arrow[r] & p_{k|k}(x_k|z_{1:k}) \arrow[r, "\text{predict}"] & p_{k+1|k}(x_{k+1}|z_{1:k}) \arrow[r, "\text{correct}"] & p_{k+1|k+1}(x_{k+1}|z_{1:k+1}) \arrow[r] & \cdots \\
 & & \begin{array}{c}\text{Markov transition density}\\ f_{k+1|k}(x_{k+1}|x_k)\end{array} \arrow[u] & \begin{array}{c}\text{likelihood function}\\ g(z_{k+1}|x_{k+1})\end{array} \arrow[u] & \\
 & & \begin{array}{c}\text{motion model}\\ x_{k+1} = f(x_k, v_k)\end{array} \arrow[u] & \begin{array}{c}\text{measurement model}\\ z_{k+1} = h(x_{k+1}, w_{k+1})\end{array} \arrow[u] &
\end{tikzcd}$$

A.2 Kalman Filter

The Kalman filter derived in [Kal60] by R.E.Kalman is one of the widely used methods for state estimation, which is based on the least-squares. The Kalman filter enables a closed-form implementation of the recursive Bayes filter. In the Kalman filter, the process and measurement models are assumed to be linear, with Gaussian distributed signals and probability densities. A Gaussian density can be completely parametrised by the mean (first order moment) and covariance (second order moment) which can be written of the form

$$\mathcal{N}(x; \hat{x}, \underline{P}) = \frac{1}{\sqrt{det(2\pi P)}} \cdot \exp\left(-\frac{1}{2}(x-\hat{x})^\top \underline{P}^{-1}(x-\hat{x})\right) \tag{A.12}$$

The basic idea in the Kalman filter is to achieve the Bayesian recursion by only propogating the mean and covariance of the Gaussian densities through time. The recursion of the

Kalman filter corresponding to the Bayesian recursion is given as

$$\begin{tikzcd}
\dots \arrow[r] & \mathcal{N}(x;\hat{x}_{k|k},\underline{\mathrm{P}}_{k|k}) \arrow[r, "\text{predict}"] & \mathcal{N}(x;\hat{x}_{k+1|k},\underline{\mathrm{P}}_{k+1|k}) \arrow[r, "\text{correct}"] & \mathcal{N}(x;\hat{x}_{k+1|k+1},\underline{\mathrm{P}}_{k+1|k+1}) \arrow[r] & \dots \\
 & & \begin{array}{c}\text{Markov}\\ \text{transition density}\\ \mathcal{N}(x_{k+1};\underline{\mathrm{F}}x_k,\underline{\mathrm{Q}}_k)\end{array} \arrow[u] & \begin{array}{c}\text{likelihood function}\\ \mathcal{N}(z_{k+1};\underline{\mathrm{H}}x_{k+1},\underline{\mathrm{R}}_{k+1})\end{array} \arrow[u] & \\
 & & \begin{array}{c}\text{motion model}\\ x_{k+1}=\underline{\mathrm{F}}x_k+v_k\end{array} \arrow[u] & \begin{array}{c}\text{measurement model}\\ z_{k+1}=\underline{\mathrm{H}}x_{k+1}+w_{k+1}\end{array} \arrow[u] &
\end{tikzcd}$$

where $\underline{\mathrm{F}}$ denotes the linear motion model and $\underline{\mathrm{H}}$ the linear measurement model. The process noise v and the measurement noise w are assumed to be white noise with zero mean and covariance matrices $\underline{\mathrm{Q}}$ and $\underline{\mathrm{R}}$ respectively. Moreover, the Gaussian distribution of the object state is represented with it's mean $\hat{x}$ and covariance $\underline{\mathrm{P}}$. Based on these properties, the recursion in the Kalman filter can be implemented with repeated prediction and update steps as given in Algorithm 1.

Algorithm 1 Linear Kalman filter

```
Prediction
    $\hat{x}_{k+1|k} = \underline{\mathrm{F}}\hat{x}_{k|k} + \mathrm{B}_k u_k$
    $\underline{\mathrm{P}}_{k+1|k} = \underline{\mathrm{F}}\underline{\mathrm{P}}_{k|k}\underline{\mathrm{F}}^\top + \underline{\mathrm{Q}}_k$
Update
    $\hat{z}_{k+1} = \underline{\mathrm{H}}\hat{x}_{k+1|k}$
    $\gamma = z_{k+1} - \hat{z}_{k+1}$
    $\underline{\mathrm{S}} = \underline{\mathrm{H}}\underline{\mathrm{P}}_{k+1|k}\underline{\mathrm{H}}^\top + \underline{\mathrm{R}}_{k+1}$
    $\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + \underline{\mathrm{K}}\gamma$
    $\underline{\mathrm{P}}_{k+1|k+1} = (\underline{\mathrm{I}} - \underline{\mathrm{K}}\underline{\mathrm{H}})\,\underline{\mathrm{P}}_{k+1|k}$
```

In the prediction step of the Kalman filter algorithm, the object state from a previous time step k is predicted into the time step $k+1$ by using the linear motion model $\underline{\mathrm{F}}$. Along with the mean value of the state, it's covariance is also predicted according to

$$\hat{x}_{k+1|k} = \underline{\mathrm{F}}\hat{x}_{k|k} \tag{A.13}$$

$$\underline{\mathrm{P}}_{k+1|k} = \underline{\mathrm{F}}\underline{\mathrm{P}}_{k|k}\underline{\mathrm{F}}^\top + \underline{\mathrm{Q}}_k \tag{A.14}$$

After the state prediction, an expected value for the measurement based on the predicted state is calculated in the innovation step. The expected measurement $\hat{z}_{k+1}$ according to the predicted state, the residual γ between predicted and actual measurement at time $k+1$, the corresponding innovation covariance $\underline{\mathrm{S}}$ of the residual based on the predicted covariance $\underline{\mathrm{P}}_{k+1}$ and the actual measurement noise covariance matrix $\underline{\mathrm{R}}_{k+1}$ are given by

$$\hat{z}_{k+1} = \underline{\mathrm{H}}\hat{x}_{k+1|k} \tag{A.15}$$

$$\gamma = z_{k+1} - \hat{z}_{k+1} \tag{A.16}$$

$$\underline{S} = \underline{HP}_{k+1|k}\underline{H}^{\top} + \underline{R}_{k+1} \tag{A.17}$$

The Kalman gain $\underline{K}$ is then calculated as

$$\underline{K} = \underline{P}_{k+1|k}\underline{H}^{\top}\underline{S}^{-1} \tag{A.18}$$

The updated estimate of the object state $\hat{x}_{k+1|k+1}$ and the corresponding updated covariance $\underline{P}_{k+1|k+1}$ are then given based on the predicted state and the uncertainities in the prediction and innovation steps, weighted by the Kalman gain $\underline{K}$ as

$$\begin{align}
\hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + \underline{K}\gamma \tag{A.19}\\
\underline{P}_{k+1|k+1} &= \underline{P}_{k+1|k} - \underline{KHP}_{k+1|k} \tag{A.20}\\
&= \underline{P}_{k+1|k} - \underline{KSK}^{\top} \tag{A.21}\\
&= (\underline{I} - \underline{KH})\underline{P}_{k+1|k}(\underline{I} - \underline{KH})^{\top} + \underline{KR}_{k+1}\underline{K}^{\top} \tag{A.22}
\end{align}$$

where $\underline{I}$ denotes the identity matrix. In the above equations, the Kalman gain regulates the contribution of the current measurement and the predicted object state to the calculation of the posterior object state. In case the measurement is believed to be more accurate i.e. the measurement has a low error covariance $\underline{R}_{k+1}$, more weight is given to current measurement's contribution in calculating the posterior state and covariance. On the other hand, if the prediction is believed to more accurate i.e. if the predicted state covariance $\underline{P}_{k+1|k}$ approaches low value, more weight is given to the predicted state's contribution. The filter recursion is initialised with an assumption of the initial object state x_0, which is also Gaussian distributed, represented by it's mean value $\hat{x}_0$ and corresponding covariance $\underline{P}_0$.

A.2.1 Constant Gain Kalman Filter

In the general form of the Kalman filter explained in the previous section, the Kalman gain $\underline{K}$ is recalculated at every time step according to (A.18). This recalculation of the Kalman gain is computationally expensive due to the required matrix inversion. On the contrary, the computational effort can be reduced if the Kalman gain is considered to be constant for all time steps, which gives the constant gain Kalman filter [Mah07b]. As the Kalman gain is assumed to be constant, only the posterior object state in (A.19) needs to be calculated and the covariance update step (A.20) is not required. Therefore the Bayesian recursion is achieved by only propagating the first order moment, i.e. the mean value of the object state $\hat{x}$ through time. The corresponding Bayesian recursion can be approximated as

$$\begin{tikzcd}
\dots \arrow[r] & p(x_k|z_{1:k}) \arrow[r, "\text{predict}"] \arrow[d] & p(x_{k+1}|z_{1:k}) \arrow[r, "\text{correct}"] \arrow[d] & p(x_{k+1}|z_{1:k+1}) \arrow[r] \arrow[d] & \dots \\
 & \hat{x}_{k|k} \arrow[r, "\text{predict}"] & \hat{x}_{k+1|k} \arrow[r, "\text{correct}"] & \hat{x}_{k+1|k+1} &
\end{tikzcd}$$

As only the object state is propagated in time, the performance of the constant gain Kalman filter is lower than the Kalman filter and are suitable only for applications where the uncertainities in the process and measurements models are lesser [Mahler07]. One of the well known constant gain Kalman filter is the α-β-γ filter [HM04a, Kal84].

A.2.2 Extension of Kalman Filter for Non-Linear Systems

In case the process model (A.5) and the measurement model (A.6) used in the Kalman filter cannot be represened as linear functions, the essential assumptions of the Kalman filter explained in above sections does not hold anymore. Therefore in such cases, the extended Kalman filter is used, which includes the same prediction and update steps of the general Kalman filter but involves an additional step of linearising the non-linear process and measurement models. The linear approximation of the non-linear functions $f(x)$ and $h(x)$ are obtained by the Taylor series expansion around the current state estimate. Due to the computational complexity usually only a first order term of the Taylor expansion is considered, however in case required by the application even higher order terms can be included [BSF88, BP99]. For a first order approximation, the linear process matrix $\underline{F}$ is derived from the Jacobian of the function $f(x)$ as

$$\underline{F}^{J} = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_k} = \left.\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}\right|_{x=\hat{x}_k} \tag{A.23}$$

and the linear measurement matrix $\underline{H}$ is approximated with the Jacobian of the measurement function $h(x)$ as

$$\underline{H}^{J} = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_{k+1}} = \left.\begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \cdots & \frac{\partial h_n}{\partial x_n} \end{pmatrix}\right|_{x=\hat{x}_{k+1}} \tag{A.24}$$

In case a second order EKF is required, additionally the Hessian of the functions $\underline{F}^{H} = \left.\frac{\partial^2 f}{\partial x_i \partial x_j}\right|_{x=\hat{x}_k}$ should also be computed. Recursion of the EKF with the filter prediction and update is given in Algorithm 2.

Algorithm 2 Extended Kalman filter

```
Prediction
    $\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_k)$
    $\underline{P}_{k+1|k} = \underline{F}\underline{P}_{k|k}\underline{F}^{\top} + \underline{Q}_k$ with $\underline{F} = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_k}$
Update
    $\hat{z}_{k+1} = h(\hat{x}_{k+1|k})$
    $\gamma = z_{k+1} - \hat{z}_{k+1}$
    $\underline{S} = \underline{H}\underline{P}_{k+1|k}\underline{H}^{\top} + \underline{R}_{k+1}$ with $\underline{H} = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_{k+1}}$
    $\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + \underline{K}\gamma$
    $\underline{P}_{k+1|k+1} = (\underline{I} - \underline{K}\underline{H})\,\underline{P}_{k+1|k}$
```

An alternative filter method to handle non-linear functions is the unsceneted Kalman filter (UKF) proposed in [JUDW95, JUDW00]. Instead of linearising the functions $f(x)$ and $h(x)$ as in the case of EKF, the UKF works by approximating the probability density function with the help of unscented transformation. The probability distribution $\mathcal{N}(x; \hat{x}, \underline{P})$ is approximated by selecting a symmetric set of points called sigma points χ. These sigma points are then transformed using the non-linear process and measurement equations and the mean and covariance of the distribution are derived from this transformation. The sigma points χ are chosen in a way that they represent the essential statistics of the Gaussian distribution, i.e. the mean and variance of the sample points should represent the expected value of the state $\hat{x}$ and the covariance $\underline{\text{P}}$ respectively. The different possibilities of choosing sigma points are presented in [JU04]. One of the possibilities is to choose $2n_x + 1$ sigma points, out of which one sigma point is at the mean of the Gaussian distribution $\hat{x}$ and the other $2n_x$ symmetric sigma points on the level curve of the Gaussian distribution representing the covariance $\underline{\text{P}}$. Additionally each sigma point carries a weight w_i to it. In [JU04] a value of $1 - \frac{n_x}{3}$ is recommended for the weight $w_{(0)}$ of the mean sigma point and the weights of other sigma points are calculated based on this. Consequently, the sigma points $\chi_{x,(i),k}$ at a time k along with their weights can be given as

$$\chi_0 = \hat{x}_k \qquad w_{(0)} = 1 - \frac{n_x}{3} \tag{A.25}$$

$$\chi_{x,(i),k} = \hat{x}_k + \left(\sqrt{\frac{n_x}{1 - w_{(0)}}\underline{\text{P}}_k}\right)_i \qquad w_{(i)} = \frac{1 - w_{(0)}}{2n_x} \tag{A.26}$$

$$\chi_{x,(i+n_x),k} = \hat{x}_k - \left(\sqrt{\frac{n_x}{1 - w_{(0)}}\underline{\text{P}}_k}\right)_i \qquad w_{(i+n_x)} = \frac{1 - w_{(0)}}{2n_x} \tag{A.27}$$

where $i = 1, \ldots, n_x$ and $\left(\sqrt{\frac{n_x}{1-w_{(0)}}\underline{\text{P}}_k}\right)_i$ is the i^{th} column from the square root of the matrix $\left(\frac{n_x}{1-w_{(0)}}\underline{\text{P}}_k\right)$. The matrix square root can be determined by Cholesky decomposition. Once the sigma points are calculated, the propagation of the Gaussian distribution can be performed by the non-linear transformation of the set of sigma points and the linearisation of the non-linear functions $f(x)$ and $h(x)$ are not required anymore. The set of state sigma points $\chi_{x,(i),k|k}$ from previous time step k are transformed into the set of sigma points $\chi_{x,(i),k+1|k}$ at time step $k + 1$ corresponding to the non-linear equation in (A.5) with the relation

$$\chi_{x,(i),k+1|k} = f(\chi_{x,(i),k|k}) \tag{A.28}$$

The mean and covariance of the state $x_{k+1|k}$ are then calculated from the transformed set of sigma points as

$$\hat{x}_{k+1|k} = \sum_{i=0}^{2n_x} w_{(i)} \chi_{x,(i),k+1|k} \tag{A.29}$$

$$\underline{\text{P}}_{k+1|k} = \sum_{i=0}^{2n_x} w_{(i)} \left(\chi_{x,(i),k+1|k} - \hat{x}_{k+1|k}\right) \left(\chi_{x,(i),k+1|k} - \hat{x}_{k+1|k}\right)^{\top} \tag{A.30}$$

Similar to the state prediction, the distribution of the predicted measurement is computed from non-linear transformation of the set of state sigma points $\chi_{x,(i),k+1|k}$ from state space to the measurement space, given by the relation

$$\chi_{z,(i),k+1} = h(\chi_{x,(i),k+1|k}) \tag{A.31}$$

Consequently the mean and covariances of the expected measurement distribution can be calculated from sigma points as

$$\hat{z}_{k+1} = \sum_{i=0}^{2n_x} w_{(i)} \chi_{z,(i),k+1} \tag{A.32}$$

$$\underline{\mathrm{P}}_{\hat{z}z} = \sum_{i=0}^{2n_x} w_{(i)} \left(\chi_{z,(i),k+1} - \hat{z}_{k+1}\right) \left(\chi_{z,(i),k+1} - \hat{z}_{k+1}\right)^\top \tag{A.33}$$

$$\underline{\mathrm{P}}_{xz} = \sum_{i=0}^{2n_x} w_{(i)} \left(\chi_{x,(i),k+1} - \hat{x}_{k+1}\right) \left(\chi_{z,(i),k+1} - \hat{z}_{k+1}\right)^\top \tag{A.34}$$

and the update of the state corresponding to the equations (A.19) and (A.20) are given as

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + \underline{\mathrm{P}}_{xz} \underline{\mathrm{P}}_{\hat{z}z}^{-1} \gamma \tag{A.35}$$

$$\underline{\mathrm{P}}_{k+1|k+1} = \underline{\mathrm{P}}_{k+1|k} - \underline{\mathrm{P}}_{xz} \underline{\mathrm{P}}_{\hat{z}z}^{-1} \underline{\mathrm{P}}_{xz}^\top \tag{A.36}$$

With the above update, accuracies equivalent to second order EKF can be reached but with even lesser computational effort [VdM04]. Further details of the UKF can be found in [JU04] and in general the UKF outperforms the first order EKF [JUDW95].

The EKF and UKF are suitable for applications where the process and measurement models are moderately non-linear. Also for the Kalman filters it is prerequiste that the probability density can be parameterised by it's first and second order moments. In case of strong non-linearities and non-parametric representation of the probability density, the particle filter also known as the sequential Monte-Carlo (SMC) method can be used [GSS93, DdFG01]. In contrast to Kalman filter, the particle filter is a non-parametric implementation of the Bayes filter in which the probability distributions are approximated with a certain number of samples called particles $\left\{x^{(i)}\right\}_{i=1}^{N}$ carrying individual weights as

$$p(x|z_{1:k}) \approx \sum_{i=1}^{N} w^{(i)} \cdot \delta_{x^{(i)}}(x), \tag{A.37}$$

where Kronecker delta function is given as

$$\delta_{x^{(i)}}(x) = \begin{cases} 1 & \text{if } x = x^{(i)} \\ 0 & \text{otherwise} \end{cases} \tag{A.38}$$

For additional details and implementation of particel filter, refer [GSS93, DdFG01, RAG04].

A.2.3 Interacting Multiple Model Kalman Filter

The filter methods explained in the above sections uses only a single motion model to represent the motion behaviour of the object. This assumption that the object motion can always be represented by a single motion model is not true in all situations. In the real world scenario different motion assumptions are required to model the movement of manuevering objects. Moreover, different types of objects would be present in the vehicle surrounding like a pedestrians, bicyclists, infrastructure or other passenger cars. Each of the objects would exhibit different motion characteristics and therefore depending on the scenario a motion model $f(x, v)$ should be selected from a set of motion models, which would then best describe the object movement. This consideration of different motion models like constant velocity (CV), constant acceleration (CA) and constant turn (CT) etc. is achieved with the interacting multiple-model (IMM) Kalman filter [Blo84]. The basic concept in the IMM filter is to run multiple filters in parallel, which use different object motion models. The object states estimated by each of the individual filters are then mixed, where each of them are given different weights according to the validity of the motion model for the current scenario. The structure of IMM filter for the case with two Kalman filters with the corresponding motion models o_1 and o_2 is illustrated in Figure A.2.

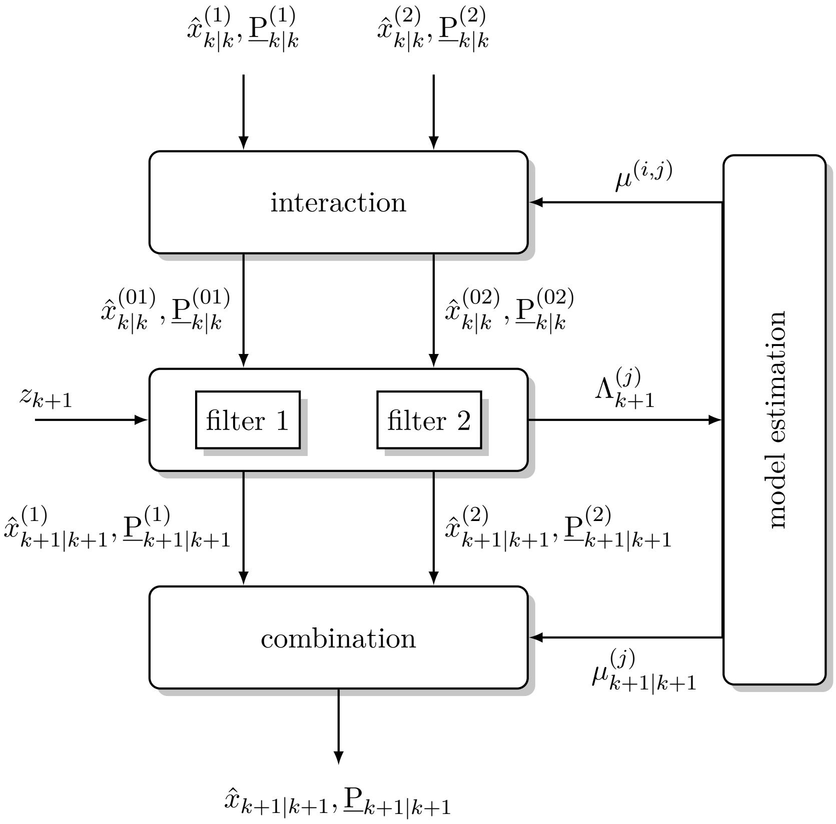

$\hat{x}^{(1)}_{k|k}, \underline{P}^{(1)}_{k|k}$
$\hat{x}^{(2)}_{k|k}, \underline{P}^{(2)}_{k|k}$
interaction
$\mu^{(i,j)}$
$\hat{x}^{(01)}_{k|k}, \underline{P}^{(01)}_{k|k}$
$\hat{x}^{(02)}_{k|k}, \underline{P}^{(02)}_{k|k}$
z_{k+1}
filter 1
filter 2
$\Lambda^{(j)}_{k+1}$
model estimation
$\hat{x}^{(1)}_{k+1|k+1}, \underline{P}^{(1)}_{k+1|k+1}$
$\hat{x}^{(2)}_{k+1|k+1}, \underline{P}^{(2)}_{k+1|k+1}$
combination
$\mu^{(j)}_{k+1|k+1}$
$\hat{x}_{k+1|k+1}, \underline{P}_{k+1|k+1}$

FIGURE A.2: IMM filter with two motion models

The object state estimation with IMM filter involves three sequential steps. The first step handles interaction between the filters based on predicted model probabilities also called as initial condition mixing. Based on the output of the interaction step, in the second step each of the filter predicts the object state according to it's motion model. The innovation of the predicted states with the received measurement in the current time step is then used for the update of the model probabilities and finally the estimated posterior object state is obtained as a combination of all the filters, based on the updated model probabilities.

Interaction

In the initial step of the IMM filter cycle, as illustrated in Figure A.2, the posterior state of each filter $(\hat{x}^{(i)}_{k|k}, \underline{P}^{(i)}_{k|k})$ from previous time step k are transformed into mixed state

$(\hat{x}^{(0j)}_{k|k}, \underline{P}^{(0j)}_{k|k})$ by mixing the estimates from all the filters, each representing different models. The mixing step is established by computing the mixing probabilties $\mu^{(i,j)}$ representing the weights with which the estimates from the previous time step of i^{th} filter are given to new mixed state of j^{th} filter as

$$\mu^{(i,j)} = \frac{t_{j,i}\mu^{(i)}_{k|k}}{\bar{c}^{(j)}}, \tag{A.39}$$

where $\bar{c}^{(j)}$ is the normalisation factor given by

$$\bar{c}^{(j)} = \sum_{i} t_{j,i}\mu^{(i)}_{k|k}, \tag{A.40}$$

representing the predicted model probability $\mu^{(j)}_{k+1|k}$. The transition probabilities $t_{j,i}$ are the elements of the Markov transition matrix $\chi_{j,i}$ corresponding to N_o number of motion models given as

$$\chi_{j,i} = \begin{bmatrix} t_{1,1} & \dots & t_{1,N_o} \\ \vdots & \ddots & \vdots \\ t_{N_o,1} & \dots & t_{N_o,N_o} \end{bmatrix} \tag{A.41}$$

Then the mixed states that goes into the j^{th} filter can be given as

$$\hat{x}^{(0j)}_{k|k} = \sum_{i} \mu^{(i,j)} \hat{x}^{(i)}_{k|k} \tag{A.42}$$

$$\underline{P}^{(0j)}_{k|k} = \sum_{i} \mu^{(i,j)} \left(\underline{P}^{(i)}_{k|k} + (\hat{x}^{(i)}_{k|k} - \hat{x}^{(0j)}_{k|k})(\hat{x}^{(i)}_{k|k} - \hat{x}^{(0j)}_{k|k})^{\top} \right) \tag{A.43}$$

Prediction and Update

Once the mixed states are computed in the interaction step for all the filters, the repsective state is predicted by each filter according to it's motion model. The predicted state $(\hat{x}^{(j)}_{k+1|k}, \underline{P}^{(j)}_{k+1|k})$ is then updated with the arrival of a measurement z_{k+1}. In case of IMMKF, the innovation of the predicted state in each filter follows the update equations of the Kalman filter (A.15)-(A.20). The model probabilities are updated based on the current measurement likelihood $\Lambda^{(j)}_{k+1}$ by

$$\mu^{(j)}_{k+1|k+1} = \frac{\Lambda^{(j)}_{k+1}\mu^{(j)}_{k+1|k}}{\sum\limits_{i} \Lambda^{(i)}_{k+1}\mu^{(i)}_{k+1|k}} \tag{A.44}$$

where the likelihood is calculated based on the measurement residual and innovation covariance of the Kalman filter as

$$\Lambda^{(j)}_{k+1} = \frac{1}{\sqrt{(2\pi)^m \left|\underline{S}^{(j)}_{k+1}\right|}} \exp\left(-\frac{1}{2} \left((\gamma^{(j)}_{k+1})^{\top} (\underline{S}^{(j)}_{k+1})^{-1} \gamma^{(j)}_{k+1} \right) \right) \tag{A.45}$$

Combination

In the final step of the IMM filter cycle, the output state estimate of the object is obtained by combining the updated estimates $(\hat{x}^{(j)}_{k+1|k+1}, \underline{\mathrm{P}}^{(j)}_{k+1|k+1})$ from each of the filter, weighted with the respective model probabilities

$$\hat{x}_{k+1|k+1} = \sum_j \mu^{(j)}_{k+1|k+1} \hat{x}^{(j)}_{k+1|k+1} \tag{A.46}$$

$$\underline{\mathrm{P}}_{k+1|k+1} = \sum_j \mu^{(j)}_{k+1|k+1} \left(\underline{\mathrm{P}}^{(j)}_{k+1|k+1} + (\hat{x}_{k+1|k+1} - \hat{x}^{(j)}_{k+1|k+1})(\hat{x}_{k+1|k+1} - \hat{x}^{(j)}_{k+1|k+1})^{\top} \right) \tag{A.47}$$

A.3 Multi-object Tracking

The estimation methods presented in the previous sections so far considered the presence of only a single object and assumed that this object gives rise to a single sensor detection. However, in the real world scenarios, multiple objects would be present in the vehicle environment. Therefore it is required to simultaneously estimate the states of all the objects present in the surrounding of the vehicle. Apart from the object states, the number of objects present should also be estimated. In comparison to the tracking of a single-object, multi-object tracking is more complicated because of the need to address the following issues:

- The number of objects present in the vehicle environment changes from time to time, because of the appearance of new objects or disappearance of already tracked objects. Also the number of objects varies according to the traffic scenarios.
- Due to the presence of multiple objects, although it can be assumed that each object would give rise to a sensor detection, the source of the detection is not evident anymore i.e. it is not known directly which measurement belongs to which object. Therefore additional data association approaches need to be used in multi-object tracking.
- In case of high-resolution sensors, multiple detections can originate from an object. Moreover, there may even be clutter measurements from the sensor or the object may not be detected despite it's presence. Such additional aspects also need to be considered in multi-object tracking algorithms.

An example scenario illustrating the change in number of objects with the evolution of objects in the state space $\mathbb{X}$ and the corresponding measurements in the measurement space $\mathbb{Z}$ are shown in Figure A.3. In convential multi-object tracking algorithms a bank of single-object Bayes filters (Kalman filters for example) run in parallel, where each filter instance is used for estimating the state of an individual object. The issue of estimating the number of objects, i.e. birth, confirmation and death of objects are handled separately by track management algorithms, which are usually based on evidences. Also for state estimation, the filter needs the measurement associated to that object. The most important part of the multi-object tracking algorithms is the data association, as wrong associations usually lead to erroneous estimations and track losses. The following sections describe different data association algorithms that are used for multi-object tracking since many years.

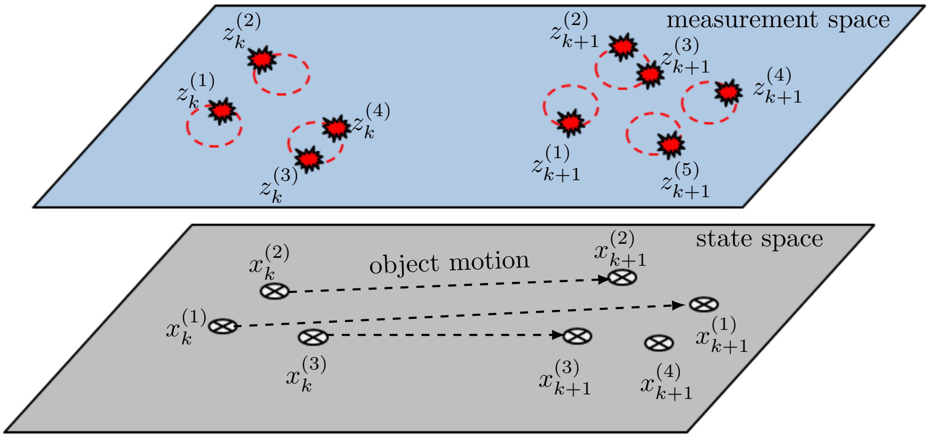


FIGURE A.3: Multi-object tracking scenario with a bank of single-object state estimators.

A.3.1 Gating

The data association step in general involves higher computation, especially when the number of of measurements and tracks are higher. In order to reduce it's computation effort, only the measurements which are within the gate of the track i.e. which are within a certain statistical distance to the track are considered for the update of that track. Therefore gating eliminates the unlikely measurement-to-track pairings going into the data association algorithm. Many techniques for constructing a gate around the predicted measurement are available (refer [BP99]). The gate is defined in such a way that it can be interpreted as the probability p_G that a measurement falls within the validation or gating region of the track. Utilising the assumption of Gaussian distribution in case of Kalman filter, ellipsoidal gate can be used, which is based on the statistical Mahalanobis distance d_{MHD} between the predicted measurement $\hat{z}_{k+1}$ of a track and the received measurement z_{k+1} given as

$$d_{MHD} = \sqrt{\gamma^\top \underline{S}^{-1} \gamma} \tag{A.48}$$

where $\gamma = z_{k+1} - \hat{z}_{k+1}$ is the measurement residual and $\underline{S}$ the innovation covariance as in Kalman fiter equations (A.16)-(A.17). Given the residual γ is also Gaussian, the square of Mahalanobis distance d_{MHD} would be χ^2 distributed as represented in Figure A.4. Based on this relation, the gating threshold υ denoting the maximum distance a measurement can be from the track in order to be considered lying within the gating region of the track is derived by defining the gate probability p_G. The gating threshold υ is therefore obtained from the inverse cumulative χ^2 distribution according to the defined probability p_G and the dimension of measurement vector n_z by

$$\upsilon = F_{\chi^2}^{-1}\left(p_G | n_z\right) \tag{A.49}$$

Considering all the tracks N and measurements M at a given time step, a $N \times M$ binary matrix called the gating matrix $\underline{G}_{ij}$ can be constructed. An element g_{ij} representing i^{th} track and j^{th} measurement would have the form

$$g_{ij} = \begin{cases} 1, & \text{if } d^2_{MHD_{(ij)}} \leq \upsilon \\ 0, & \text{otherwise.} \end{cases} \tag{A.50}$$

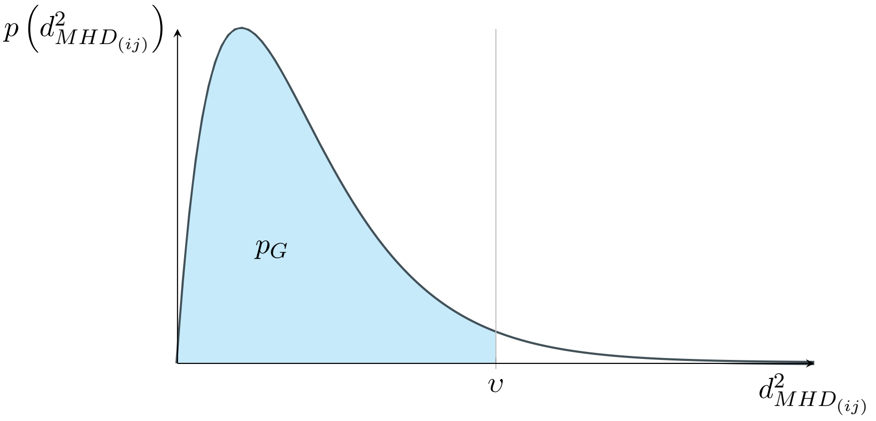


FIGURE A.4: chi-square distribution.

For each track, only the measurements corresponding to it's non-zero column elements would then be further considered as candidates for that track's update.

A.3.2 Nearest Neighbour Data Association

The simplest approach to associate a measurement to a track is the nearest-neighbour (NN). Out of all the measurements that are within the gating region of a track, the closest measurement with the shortest Euclidean or Mahalanobis distance is assigned to that track. But a problem here is, a measurement can be present within the gating region of multiple tracks which means there is a possibility that the same measurement can be assigned to many tracks as illustrated in Figure A.5. This would mean that the measurement belongs to all those objects, which is not true.

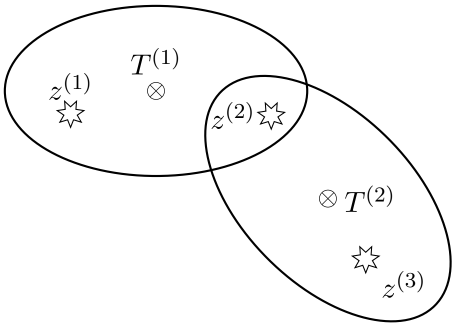


FIGURE A.5: Example of an association conflict situation where the measurement $z^{(2)}$ lies within the gating region of both the tracks $T^{(1)}$ and $T^{(2)}$.

The erroneous association in NN approach can be overcome by the global nearest-neighbour (GNN) approach [BP99], in which scores for all the measurements to track associations are computed and only the best association hypothesis out of them is selected. The association problem is solved in GNN method with help of constructing an assignment matrix $\underline{A}_a$, with each of it's element a_{ij} representing the association of i^{th} track to j^{th} measurement. Also, the score for a measurement-to-track association is computed which can be based on $d^2_{MHD_{(ij)}}$.

The best association hypothesis is then found by solving the assignment problem using optimal assignment algorithms like the auction algorithm [Ber88, Ber90], the Hungarian method [Kuh55], Munkres algorithm [Mun57], etc. The track mangement in such cases can be implemented with a M/N rule, meaning that a track is confirmed if it's measurement is prevelant for M times out of N time steps and similarly the confirmed track can be deleted if it does not get associated to a measurement for M times out of N tme steps.

A.3.3 Probabilistic Data Association Methods

Due to the hard association decision of NN approaches, where only a single association hypothesis is considered, the performance of data association would degrade in the presence of high clutter. The probabilistic data association (PDA) approach presented in [BSF88] makes a soft association decision, by considering all the measurement-to-track associations for each track individually. Thereby, all the measurements $\mathrm{Z}_{k+1} = \{z_1, \dots, z_M\}$ that are present within the gate of a track at a given time step influences the state update of that track with a certain weight. The weighted state update is performed by determining association probabilites $\beta^{(i,j)}$ for each track i with all the measurements $j = 1, \dots, M$. Therefore an element $a_{i,j}$ of the assignment matrix $\underline{\mathrm{A}}$ would contain the association probability $\beta^{(i,j)}$ between track i^{th} track and j^{th} measurement. The probability of missed detection is represented by $\beta^{(i,0)}$. The association probabilities for each track should sum up to one. The posterior density of a track $x^{(i)}$ can therefore be given as

$$p(x^{(i)}|z_1, \dots z_M) = \sum_{j=0}^{M} \beta^{(i,j)} p(x^{(i)}|z_j), \tag{A.51}$$

where $p(x^{(i)}|z_j)$ represents the posterior of the state $x^{(i)}$ updated by the measurement z_j. In case of missed detection the posterior of the state is same as the predicted state. The multi-modal density in (A.51) is not Gaussian anymore. Therefore, the posterior density needs to be approximated by a single Gaussian distribution in order to propagate Kalman filter recursion. The approximation is achieved by weighted innovation of each association hypothesis and computing the properties of a unimodal Gaussian distribution from these individual innovations. The posterior state of each association hypothesis is given according to (A.19) as

$$\hat{x}^{(i,j)}_{k+1|k+1} = \hat{x}^{(i)}_{k+1|k} + \underline{\mathrm{K}}^{(i,j)}(z_{j,k+1} - \hat{z}^{(i)}_{k+1}) \tag{A.52}$$

where $i = 1, \dots, N$ and $j = 1, \dots, M$. For the case of missed detection hypothesis ($j = 0$) the Kalman gain and residual are zero and the posterior state is same as the predicted state

$$\hat{x}^{(i,0)}_{k+1|k+1} = \hat{x}^{(i)}_{k+1|k}. \tag{A.53}$$

The posterior object state $\hat{x}^{(i)}_{k+1|k+1}$ is then approximated as weighted mean of the individual innovations

$$\hat{x}^{(i)}_{k+1|k+1} = \sum_{j=0}^{M} \beta^{(i,j)} \hat{x}^{(i,j)}_{k+1|k+1} \tag{A.54}$$

The corresponding covariance matrix is updated by

$$\underline{\mathrm{P}}^{(i)}_{k+1|k+1} = \sum_{j=0}^{M} \beta^{(i,j)} \Big(\underline{\mathrm{P}}^{(i)}_{k+1|k} - \underline{\mathrm{K}}^{(i,j)} \underline{\mathrm{S}}^{(i,j)} (\underline{\mathrm{K}}^{(i,j)})^\top + (\hat{x}^{(i,j)}_{k+1|k+1} - \hat{x}^{(i)}_{k+1|k+1})(\hat{x}^{(i,j)}_{k+1|k+1} - \hat{x}^{(i)}_{k+1|k+1})^\top \Big) \tag{A.55}$$

In PDA, the basic assumption is the presence of only a single object in clutter and therefore the presence of multiple objects is not considered in the data association. As in the case of NN data association, this would have degraded performance in a conflict situation when a measurement lies within the gating region of multiple tracks. Therefore the PDA method was extended to joint probability data association (JPDA), in which the association probabilites $\beta^{(i,j)}$ are computed using all measurements and all tracks [FBSS83, BSWT11]. In case of a conflict situation as in Figure A.5, all the tracks that have common measurements in their gate are grouped into a cluster, including the respective measurements. The posterior of the tracks are then determined by enumerating all the possible measurement-to-track association hypotheses of that cluster group including the cases of missed detection. The probability of each association hypothesis influencing the state update of the tracks need to be determined. A possible method to put down all the association hypotheses is by constructing a hypotheses tree. If the hypotheses are represented in such a tree like structure, each node in the tree would represent an elementary association e and each path starting from the root node e_0 till it's end node e_L would denote an association hypothesis constituting a particular set of elementary associations as $E = \{e_0, \ldots, e_L\}$, which could also include the case of missed detection and clutter.

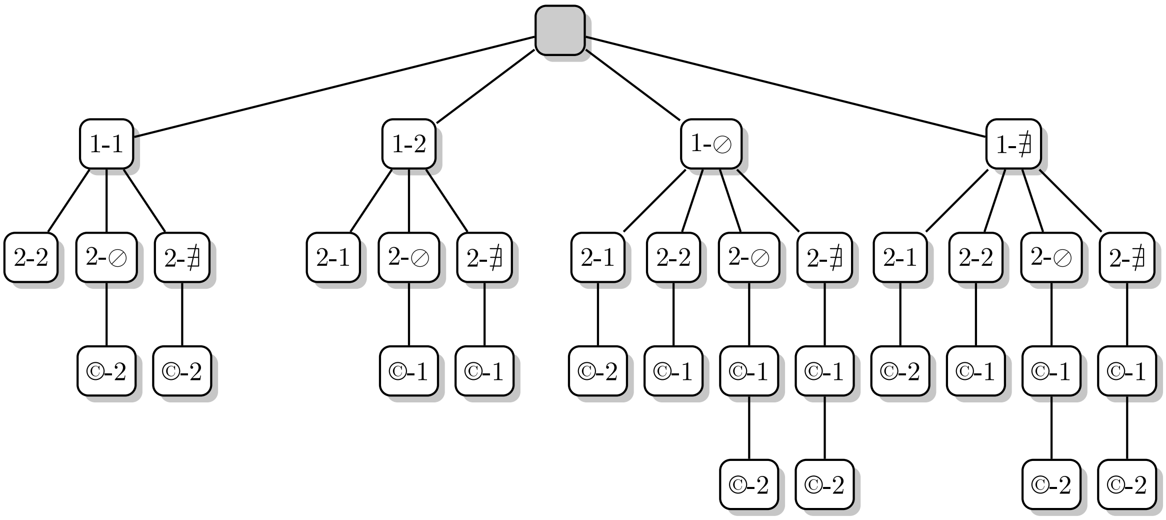


Figure A.6: JIPDA hypotheses tree for two measurements and two tracks.

However, an assumption in PDA and JPDA methods is that the object already exists. The joint integrated probabilistic data association (JIPDA) introduced in [ME04a] extends the JPDA approach by explicitly considering the object existence. In JIPDA method, along with the data association probabilites, the probability of object existence $p(\exists)$ is also calculated at every time step. The object existence is modeled as a Markov chain and is predicted and updated like the states of a Kalman filter. Other than the object existence consideration, the data association step of the JIPDA uses the enumeration of all association hypotheses like the JPDA filter. One such enumeration of all association hypotheses for the case of two measurements and two tracks is illustrated in Figure A.6. Along with the case of a object not being detected $\oslash$, the case of non-existence of the object $\nexists$ is also considered for track association. Similarly the case of a detection being a clutter © is also considered. Consequently the measurement set can be augmented as $\mathrm{Z}^*_{k+1} = \{z_1, z_2, \oslash, \nexists\}$ and the state set as $\mathrm{X}^*_{k+1|k} = \left\{x^{(1)}, x^{(2)}, \text{©}\right\}$. Subsequently, the probability of an association hypothesis E can be calculated by considering the set of objects $X_{na}(e)$ that have not been detected and set of objects $X_a(e)$ that are assigned with a measurement as

$$p(E) = \eta^{-1} \prod_{x^{(i)} \in X_{na}(e)} \left(1 - p_D(x^{(i)}) p_G p_{k+1|k}(\exists x^{(i)})\right) \cdot \prod_{x^{(i)} \in X_a(e)} \left(\frac{V}{\hat{m}_c} \Lambda(z_{i,k+1}|x^{(i)}) p_D(x^{(i)}) p_G p_{k+1|k}(\exists x^{(i)})\right) \tag{A.56}$$

where η is the normalisation constant which ensures that the probability of all the hypotheses sum up to one, V the gating volume of the cluster and $\hat{m}_c$ the number of false alarms. Further p_D denotes the detection probability of the object, $z_{i,k+1}$ denotes the measurement associated to track $x^{(i)}$ in the hypothesis E and $\Lambda(z_{i,k+1}|x^{(i)})$ the likelihood of measurement. Further details of the derivation can be found in [ME04a, BSWT11]. Going back to the tree based hypotheses represention, the probability of an association hypothesis corresponding to (A.56) can be determined from the individual node probabilities of the path as

$$p(E) = \prod_{e \in E} p(e) \tag{A.57}$$

After computing the probabilities of all the association hypotheses E, the existence probability $p(\exists)_{k+1|k+1}$ of the track $x^{(i)}$ can be extracted from the sum of probabilities of the hypotheses which say that the object exists, normalised by the sum of probabilities of all the association hypotheses. Similarly, the association probability $\beta^{(i,j)}$ of a track $x^{(i)}$ to the measurement z_j can be extracted from the sum of probabilities of hypotheses in which the j^{th} measurement is actually associated to track i^{th}, normalised by the sum of probabilities of the hypotheses which say that the object exists. Therefore the posterior existence probability of the object can be given as [Mä09]

$$p_{k+1|k+1}(\exists x^{(i)}) = \frac{\sum\limits_{E \in E^{(i)}} p(E)}{\sum\limits_{E \in E} p(E)} \tag{A.58}$$

and the association probability of measurement j to track i as

$$\beta^{(i,j)} = \frac{\sum\limits_{E \in E^{(i,j)}} p(E)}{\sum\limits_{E \in E^{(i,\exists)}} p(E)} \tag{A.59}$$

The updated posterior state of the track $\hat{x}^{(i)}_{k+1|k+1}$ is then approximated as a single Gaussian distribution as in the case of PDA according to (A.54), by substituting the value of $\beta^{(i,j)}$ obtained from the above equation (A.59). The state covariance update follows (A.55). The computational complexity for enumerating all the association hypotheses in the hpotheses tree clearly increases with the increase in the number of measurements and tracks. Methods to ensure computational tractability of the JIPDA filter in such cases by extensive gating techniques and calculating a upper limit on the number of hypotheses are proposed in [Mä09, Mun11].

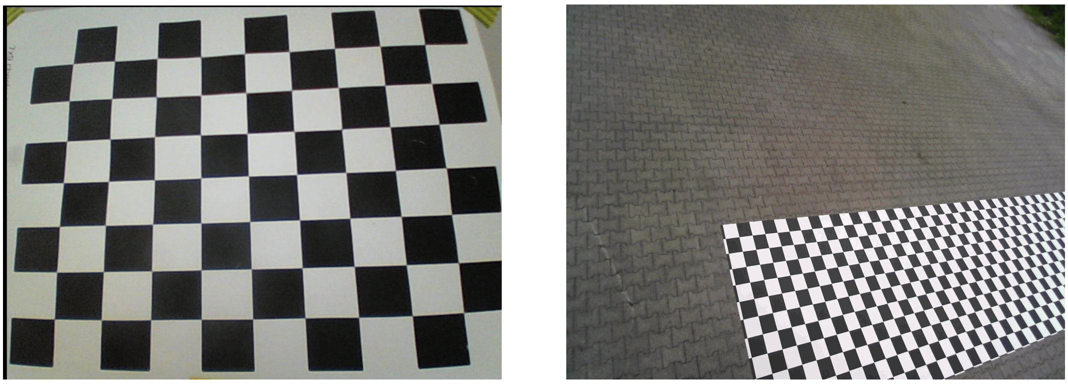
Appendix B

Ground truth solution

B.1 Camera calibration

In order to evaluate the tracker, image based ground truth reference solution is used. During the test sequences, webcam video is recorded along with the radar measurements with their respective timestamps. The recorded images are synchronised by associating the timestamps and are then processed offline to extract the ground truth information. The derived ground truth is then used for evaluating the tracker performance and accuracy. However, image based reference data is designed only for the bicyclist class of objects. For this, four bright red colored markers(pallets) are attached on either sides of the front and rear wheels of the bicycle used for testing purposes. OpenCV libraries in Python are used for calibration and offline image processing to extract position of the bright red colored pallet.

Before the reference could be extracted from the webcam images, the camera needs to be spatially calibrated for ground truth projection, thereby the intrinsic and extrinsic camera parameters need to determined. Checkerboard method [Zha00] is used for this purpose. The instrinsic camera matrix is first calculated by capturing various images of a checkerboard, from different orientations as shown in Figure B.1a. Then during the test sequences, a large checker board with each box size 17x17 cm is spread in the test area as shown in Figure B.1b, for calculating the homography matrix.



(A) (B)

Figure B.1: Camera calibration with checker board

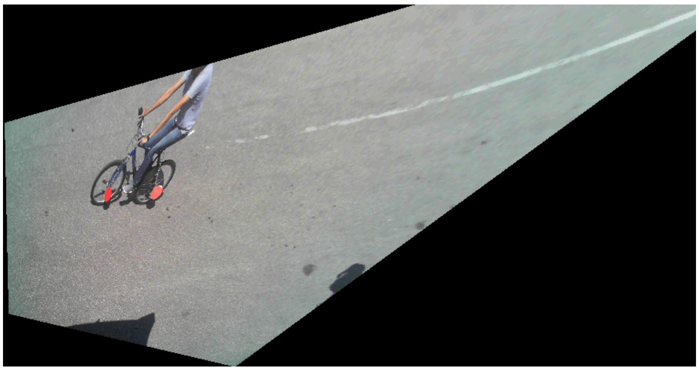


FIGURE B.2: Ground projected camera image. To be seen are the bright red markers attached to the wheels of the bicyle

B.2 Marker detection

Once the camera is calibrated, further steps are required to find the position of the bright red markers in the ground projected image. The marker detection for all the test sequences involving bicyclists are performed offline. The recorded and eventually ground projected image is first read in. Then the image is smoothened using the median blur. The smoothened images are then converted from RGB to HSV color-space. After the conversion step, a lower and upper threshold values of [0,140,120] and [180,255,255] respectively are applied for the seeking the red markers. The next step is to find the contours in the image and the corresponding minimum enclosing circle for those contours. The red marker positions are finally extracted from the center points of the enclosing circles. The position of the front right corner of the checker board is also measured with respect to the radar mounting coordinates, in order to find the transformation between radar and image coordinates. Finally, the transformed marker positions are used as the reference ground truth values.

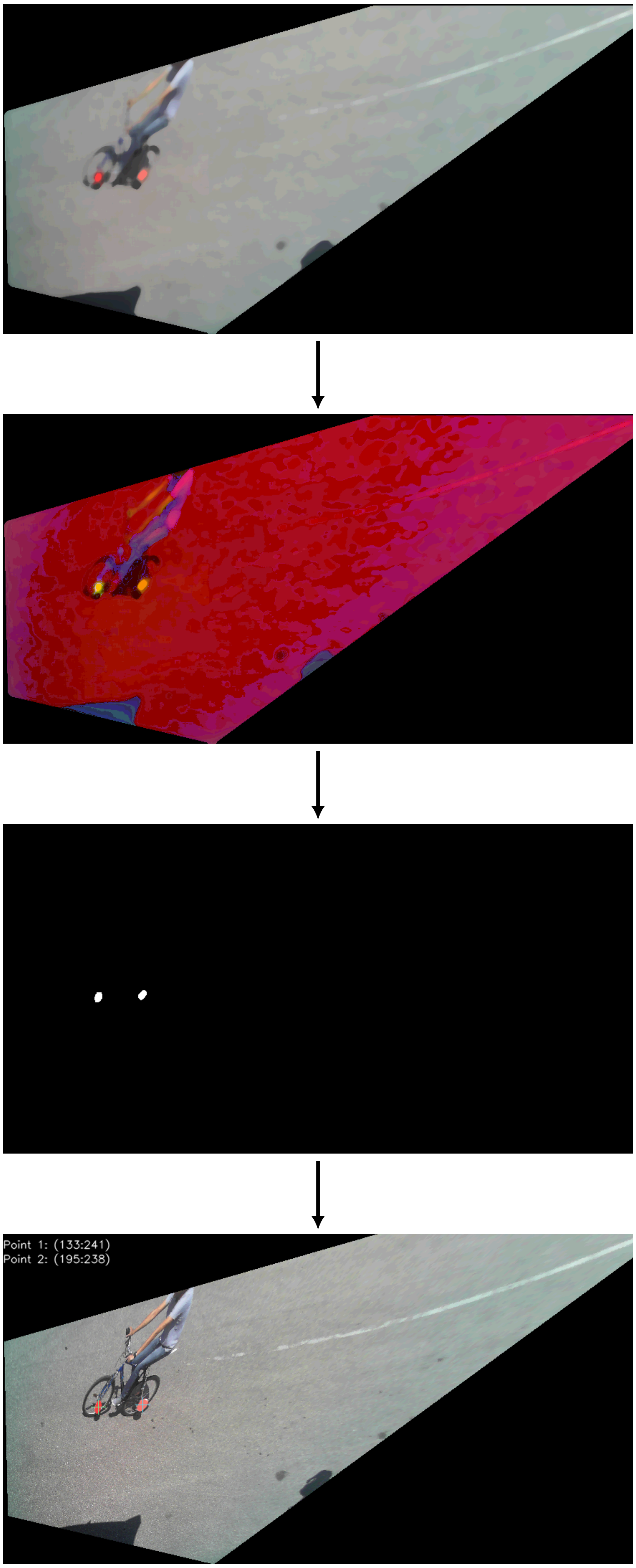


FIGURE B.3: Steps invovlved in detection of red marker positions attached to the bicycle

Appendix C

DBSCAN algorithm

Algorithm 3 DBSCAN Clustering

```
procedure DBSCAN(SetOfPts, Eps, MinPts)
    initialise $C = 0$
    for i from 1 to size(SetOfPts) do
        if point P is unvisited then
            mark P as visited
            NeighPts $=$ regionQuery(SetOfPts, P, Eps)
            if size(NeighPts) $<$ MinPts then
                mark P as Noise
            else
                $C = C + 1$
                expandCluster(SetOfPts, P, NeighPts, C, Eps, MinPts)
            end if
        end if
    end for
end procedure
procedure expandCluster(SetOfPts, P, NeighPts, C, Eps, MinPts)
    add P to cluster C
    for i from 1 to size(NeighPts) do
        if point P' $\in$ NeighPts is unvisited then
            mark P' as visited
            NeighPts' $=$ regionQuery(SetOfPts, P', Eps)
            if size(NeighPts') $\geq$ MinPts then
                NeighPts $=$ NeighPts $\cup$ NeighPts'
            end if
            if P' $\notin$ any cluster then
                include P' in cluster C
            end if
        end if
    end for
end procedure
```

```
procedure REGIONQUERY(SetOfPts, P, Eps)
    initialise NeighPts = $\emptyset$
    for $\forall$ Q $\in$ SetOfPts do
        if dist(P, Q) $\leq$ Eps then
            NeighPts = NeighPts $\cup$ {Q}
        end if
    end for
    return NeighPts
end procedure
```

Appendix D

Zadeh's Paradox in Dempster's rule of combination

The Dempster's rule of combination in regard to strong conflicting evidences has been highly critisized mainly based on the "Zadeh's Paradox" [Zad84]. Zadeh's paradox is given as: Assuming two doctors, Dr. A and Dr. B, independently examining a patient for symptoms of meningitis (μ), brain tumor (τ) or concussion (χ). Dr. A is almost sure that it is meningitis with a slight possibility of also being a brain tumor but definitely not concussion. Whereas Dr. B contradicts Dr. A's belief saying that is almost concussion with a slight possibility of being a brain tumor, but is definitely not meningitis. The corresponding BBAs according to Dr. A and Dr. B for the elements $\Omega = \{\mu, \tau, \chi\}$ can be given as

$$\begin{aligned} m_A(\mu) = 0.99, \quad m_A(\tau) = 0.01, \quad m_A(\chi) = 0.0 \\ m_B(\mu) = 0.0, \quad m_B(\tau) = 0.01, \quad m_B(\chi) = 0.99 \end{aligned} \tag{D.1}$$

with the belief for all the other elements considered as zero. The combination of the above stated BBAs according to Dempster's rule of combination can be given as [Mah07b]

$$m_A \oplus m_B(\mu) = \frac{m_A(\mu) \cdot m_B(\mu)}{m_A(\mu)m_B(\mu) + m_A(\tau)m_B(\tau) + m_A(\chi)m_B(\chi)} = 0.0 \tag{D.2}$$

$$m_A \oplus m_B(\tau) = \frac{m_A(\tau) \cdot m_B(\tau)}{m_A(\mu)m_B(\mu) + m_A(\tau)m_B(\tau) + m_A(\chi)m_B(\chi)} = 1.0 \tag{D.3}$$

$$m_A \oplus m_B(\chi) = \frac{m_A(\chi) \cdot m_B(\chi)}{m_A(\mu)m_B(\mu) + m_A(\tau)m_B(\tau) + m_A(\chi)m_B(\chi)} = 0.0 \tag{D.4}$$

$$\tag{D.5}$$

The result out of the above combination implies that the patient definitely has brain tumor. This is contradicting as both Dr. A and Dr. B gave only a very small support for the possibility of brain tumor. This contradictory result was majorly accepted as the main argument against the validity of Dempster's rule of combination. Mahler in [Mah07a] argues that the Zadeh's paradox is an unknowing criticism and states that the Demspter's rule of combination in the above example can be represented equivalent to the Bayes' rule, as the belief is concentrated only the elementary hypotheses. Further Mahler claims that it is unwise to assign zero probability to any of the state. In that way, the paradox can be resolved by assigning a very small non-zero values to the not supported states. Therefore in the above stated doctors example, considering a reliability of $\alpha = 0.99$, very small values are also assigned to $m_A(\chi)$ and $m_B(\mu)$ and the discounted BBAs can be given as

$$\begin{aligned} m_A(\mu) = 0.99, \quad m_A(\tau) = 0.009, \quad m_A(\chi) = 0.001 \\ m_B(\mu) = 0.001, \quad m_B(\tau) = 0.009, \quad m_B(\chi) = 0.99 \end{aligned} \tag{D.6}$$

Now combining the above BBAs according to the Bayes' rule gives

$$m_A \oplus m_B(\mu) = 0.48, \quad m_A \oplus m_B(\tau) = 0.04, \quad m_A \oplus m_B(\chi) = 0.48 \tag{D.7}$$

which infers equal possibility for both meningitis and concussion and least possibility of brain tumor. Thereby, discounting the BBAs according to it's reliability helps in a correct combination of contradicting evidences.

Bibliography

[AMB07] Arras, K. O., Mozos, O. M., and Burgard, W. *Using Boosted Features for the Detection of People in 2D Range Data.* In: *IEEE International Conference on Robotics and Automation*, pages 3402–3407, 2007.

[APA04] Aoudas, A., Polychronopoulos, A., and Amdifis, A. *Lateral motion tracking of automobiles.* In: *7th International Conference on Information Fusion*, 2004.

[Arr03] Arras, K. O. *Feature-Based Robot Navigation in Known and Unknown Environments.* PhD dissertation, École polytechnique fédérale de Lausanne, 2003.

[AS72] Alspach, D. and Sorenson, H. *Nonlinear Bayesian estimation using Gaussian sum approximations.* In: *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pages 439–448, 1972.

[Bü08] Bühren, M. *Simulation und Verarbeitung von Radarziellisten im Automobil.* PhD dissertation, University of Stuttgart, 2008.

[Ber88] Bertsekas, D. P. *The auction algorithm: a distributed relaxation method for the assignment problem.* In: *Annals of Operations Research*, vol. 14, no. 1, pages 105–123, 1988.

[Ber90] Bertsekas, D. P. *The auction algorithm for assignment and other network flow problems: a tutorial.* In: *Interfaces*, vol. 20, no. 4, pages 133–149, 1990.

[BGV92] Boser, B. E., Guyon, I. M., and Vapnik, V. N. *A Training Algorithm for Optimal Margin Classifiers.* In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[Bis06] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer, 2006.

[BL71] Bourgeois, F. and Lassalle, J.-C. *An extension of the Munkres algorithm for the assignment problem to rectangular matrices.* In: *Communications of the ACM*, vol. 14, no. 12, pages 802–804, 1971.

[Blo84] Blom, H. P. A. *An Efficient Filter for Abruptly Changing System.* In: *Proceedings of the 23rd IEEE Conference on Decision and Control*, pages 656–658, 1984.

[BP99] Blackman, S. and Popoli, R. *Design and Analysis of Modern Tracking Systems.* Artech House Publishers, 1999.

[BRG$^+$16] Beard, M., Reuter, S., Granström, K., Vo, B.-T., Vo, B.-N., and Scheel, A. *Multiple Extended Target Tracking With Labeled Random Finite Sets.* In: *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pages 1638–1653, 2016.

[BS02] Bar-Shalom, Y. *Update with out-of-sequence measurements in tracking: exact solution.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pages 769–777, 2002.

[BSB82] Bar-Shalom, Y. and Birmiwal, K. *Variable Dimension Filter for Maneuvering Target Tracking.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-18, no. 5, pages 621–629, 1982.

[BSCM04] Bar-Shalom, Y., Chen, H., and Mallick, M. *One-step solution for the multistep out-of-sequence-measurement problem in tracking.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 1, pages 27–37, 2004.

[BSF88] Bar-Shalom, Y. and Fortmann, T. *Tracking and Data Association.* Academic Press Inc., 1988.

[BSL95] Bar-Shalom, Y. and Li, X.-R. *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS Publishing, Storrs , CT , USA, 1995.

[BST75] Bar-Shalom, Y. and Tse, E. *Tracking in a cluttered environment with probabilistic data association.* In: *Automatica*, vol. 11, no. 5, pages 451–460, 1975.

[BSWT11] Bar-Shalom, Y., Willet, P., and Tian, X. *Tracking and Data Fusion: A Handbook of Algorithms.* YBS Publishing, Storrs , CT , USA, 2011.

[BVV12] Beard, M., Vo, B.-T., and Vo, B.-N. *Multitarget filtering with unknown clutter density using a bootstrap GMCPHD filter.* In: *Proceedings of the 15th International Conference on Information Fusion*, pages 535–541, 2012.

[BVV13] Beard, M., Vo, B.-T., and Vo, B.-N. *Multitarget filtering with unknown clutter density using a bootstrap GMCPHD filter.* In: *IEEE Signal Processing Letters*, vol. 20, no. 4, pages 323–326, 2013.

[CL13] Chang, C.-C. and Lin, C.-J. *LIBSVM: a Library for Support Vector Machines*, Initial version: 2001 & Updated version 2013. https://www.csie.ntu.edu.tw/ cjlin/libsvm/.

[CV95] Cortes, C. and Vapnik, V. *Support-Vector Networks.* In: *Machine Learning*, pages 273–297, 1995.

[DBS93] Dezert, J. and Bar-Shalom, Y. *Joint probabilistic data association for autonomous navigation.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 4, pages 1275–1286, 1993.

[DdFG01] Doucet, A., de Freitas, N., and Gordon, N. *Sequential Monte Carlo methods in practice.* Springer, New York, 2001.

[Deu15] Deusch, H. *Random Finite Set-Based Localization and SLAM for Highly Automated Vehicles.* PhD dissertation, Ulm University, 2015.

[DHG01] Duda, R. O., Hart, P. E., and G, S. D. *Pattern Classification*, volume 2. John Wiley & Sons, Inc., 2001.

[DK13] Dunne, D. and Kirubarajan, T. *Multiple model multi-Bernoulli filters for manoeuvering targets.* In: *IEEE Transactions on Aerospace and Electronic Systems,*, vol. 49, no. 4, pages 2679–2692, 2013.

[DRD16] Danzer, A., Reuter, S., and Dietmayer, K. *The Adaptive Labeled Multi-Bernoulli Filter.* In: *Proceedings of the 19th International Conference on Information Fusion*, 2016.

[Dri17] Driving, E. W. G. C. . A. *Automated Driving Roadmap.*, 2017.

[DVJ88] Daley, D. and Vere-Jones, D. *An Introduction to the Theory of Point Processes.* Springer, New York, 1988.

[DVJ03] Daley, D. and Vere-Jones, D. *An Introduction to the Theory of Point Processes*, volume 1: Elementary Theory and Methods. Springer, 2nd edition, 2003.

[EAGG13] Edman, V., Andersson, M., Granström, K., and Gustafsson, F. *Pedestrian group tracking using the GM-PHD filter.* In: *Proceedings of the European Signal Processing Conference*, Marrakech, Morocco, 2013.

[EKSX96] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.* In: *Proceedings of KDD*, vol. 96, pages 226–231, 1996.

[Epp98] Eppstein, D. *Finding the k shortest paths.* In: *SIAM Journal on Computing*, vol. 28, no. 2, pages 652–673, 1998.

[EWBS05a] Erdinc, O., Willett, P., and Bar-Shalom, Y. *Probability hypothesis density filter for multitarget multisensor tracking.* In: *Proceedings of the 8th International Conference on Information Fusion*, pages 1–8, 2005.

[EWBS05b] Erdinc, O., Willett, P., and Bar-Shalom, Y. *Probability hypothesis density filter for multitarget multisensor tracking.* In: *Proceedings of the 8th International Conference on Information Fusion*, pages 1–8, 2005.

[EWC08] Erdinc, O., Willett, P., and Coraluppi, S. *The Gaussian mixture cardinalized PHD tracker on MSTWG and SEABAR'07 datasets.* In: *Proceedings of the 11th International Conference on Information Fusion*, pages 1–8, 2008.

[Fö06] Fölster, F. *Erfassung ausgedehnter Objekte durch ein Automobil-Radar.* PhD dissertation, Technical University Hamburg-Harburg, 2006.

[FBSS83] Fortmann, T., Bar-Shalom, Y., and Scheffe, M. *Sonar tracking of multiple targets using joint probabilistic data association.* In: *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pages 173–184, 1983.

[FC08] Fayad, F. and Cherfaoui, V. *Object-level fusion and confidence management in a multi-sensor pedestrian tracking system.* In: *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 58–63, 2008.

[FR06] Folster, F. and Rohling, H. *Lateral velocity estimation based on automotive radar sensors.* In: *International Conference on Radar*, 2006.

[fWLW03] fan Wu, T., Lin, C.-J., and Weng, R. C. *Probability Estimates for Multi-class Classification by Pairwise Coupling.* In: *Journal of Machine Learning Research*, vol. 5, pages 975–1005, 2003.

[Gar14] Garcia, C. *Multiple Sensor Fusion for Detection, Classification and Tracking of Moving Objects in Driving Environments.* PhD dissertation, University of Grenoble, France, 2014.

[GE03] Guyon, I. and Elisseeff, A. *An Introduction to Variable and Feature Selection.* In: *Journal of Machine Learning*, vol. 3, pages 1157–1182, 2003.

[GLO10] Granström, K., Lundquist, C., and Orguner, U. *A Gaussian mixture PHD filter for extended target tracking.* In: *Proceedings of the 13th International Conference on Information Fusion*, pages 1–8, 2010.

[GLO11] Granström, K., Lundquist, C., and Orguner, U. *Tracking rectangular and elliptical extended targets using laser measurements.* In: *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, 2011.

[GMN97] Goodman, I., Mahler, R., and Nguyen, H. *Mathematics of Data Fusion: From the series Theory and decision library: Series B: Mathematical and Statistical Methods.* Springer, 1997.

[GO12] Granström, K. and Orguner, U. *A PHD filter for tracking multiple extended targets using random matrices.* In: , vol. 60, no. 11, pages 5657–5671, 2012.

[Gor97] Gordon, N. *A hybrid bootstrap filter for target tracking in clutter.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pages 353–358, 1997.

[GSS93] Gordon, N., Salmond, D., and Smith, A. F. M. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation.* In: *IEEE Proceedings Radar and Signal Processing*, vol. 140, no. 2, pages 107–113, 1993.

[GW12] Georgescu, R. and Willet, P. *Classification aided cardinalized probability hypothesis density filter.* In: *Signal Processing, Sensor Fusion, and Target Recognition XXI*, 2012.

[GWBS15] Granström, K., Willett, P., and Bar-Shalom, Y. *Systematic approach to IMM mixing for unequal dimension states.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 4, pages 2975–2986, 2015.

[HL64] Ho, Y.-C. and Lee, R. *A Bayesian approach to problems in stochastic estimation and control.* In: *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pages 333–339, 1964.

[HM04a] Hall, D. L. and McMullen, S. A. H. *Mathematical Techniques in Multisensor Data Fusion.* Artech House Inc., Norwood, 2004.

[HM04b] Hoffman, J. and Mahler, R. *Multitarget miss distance via optimal assignment.* In: *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 34, no. 3, pages 327–336, 2004.

[HM11] Horridge, P. and Maskell, S. *Using a probabilistic hypothesis density filter to confirm tracks in a multi-target environment.* In: *Proceedings of the 6th Workshop Sensor Data Fusion: Trends, Solutions, Applications*, 2011.

[Hof06] Hoffmann, M. *Support Vector Machines — Kernels and the Kernel Trick*, 2006.

[HR10] Heuel, S. and Rohling, H. *Pedestrian recognition based on 24 GHz radar sensors.* In: *Proceedings of the 11th International Radar Symposium (IRS)*, 2010.

[HSRD16] Hirsch, T., Scheel, A., Reuter, S., and Dietmayer, K. *Multiple Extended Object Tracking Using Gaussian Processes.* In: *Proceedings of the 19th International Conference on Information Fusion*, 2016.

[HSSS12] Hammarstrand, L., Svensson, L., Sandblom, F., and Sorstedt, J. *Exploiting LIDAR-based Features on Pedestrian Detection in Urban Scenarios.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pages 2371–2386, 2012.

[HVV13] Hoseinnezhad, R., Vo, B.-T., and Vo, B.-N. *Visual tracking in background subtracted image sequences via multi-Bernoulli filtering.* In: *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pages 392–397, 2013.

[HVV15] Hoang, H. G., Vo, B.-T., and Vo, B.-N. *A Generalized Labeled Multi-Bernoulli Filter Implementation using Gibbs Sampling.* In: *International Conference on Information Fusion*, Washington DC , USA, 2015.

[HVVS11] Hoseinnezhad, R., Vo, B.-T., Vo, B.-N., and Suter, D. *Bayesian integration of audio and visual information for multi-target tracking using a CB-MeMBer filter.* In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2300–2303, 2011.

[JROR10] Jensen, P., Rouquier, J.-B., Ovtracht, N., and Robardet, C. *Characterizing the speed and paths of shared bicycle use in Lyon.* In: *Transportation Research Part D: Transport and Environment*, vol. 15, no. 8, pages 522 – 524, 2010.

[JU04] Julier, S. and Uhlmann, J. *Unscented filtering and nonlinear estimation.* In: *Proceedings of the IEEE*, vol. 92, no. 3, pages 401–422, 2004.

[JUDW95] Julier, S., Uhlmann, J., and Durrant-Whyte, H. *A new approach for filtering nonlinear systems.* In: *Proceedings of the 1995 American Control Conference*, vol. 3, pages 1628–1632, 1995.

[JUDW00] Julier, S., Uhlmann, J., and Durrant-Whyte, H. *A new method for the nonlinear transformation of means and covariances in filters and estimators.* In: *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pages 477–482, 2000.

[JV15] Jones, B. A. and Vo, B.-N. *A Labeled Multi-Bernoulli Filter For Space Object Tracking.* In: *AAS/AIAA Space Flight Mechanics Meeting*, Willamsburg, VA, 2015. Preprint: AAS 15-413.

[Kä07] Kämpchen, N. *Feature-level fusion of laser scanner and video data for advanced driver assistance systems.* PhD dissertation, Ulm University, 2007.

[Kal60] Kalman, R. E. *A new approach to linear filtering and prediction problems.* In: *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, no. series D, pages 35–45, 1960.

[Kal84] Kalata, P. R. *The tracking index: a generalized parameter for* $\alpha - \beta$ *and* $\alpha - \beta - \gamma$ *target trackers.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 20, no. 2, pages 174–182, 1984.

[KBD$^+$13] Kellner, D., Barjenbruch, M., Dietmayer, K., Klappstein, J., and Dickmann, J. *Instantaneous lateral velocity estimation of a vehicle using Doppler radar.* In: *Proceedings of the 16th International Conference on Information Fusion*, pages 877–884, 2013.

[KD03] Kämpchen, N. and Dietmayer, K. *Data synchronization strategies for multi-sensor fusion.* In: *10th World Congress on Intelligent Transportation Systems*, Spain, 2003.

[Koc08] Koch, J. W. *Bayesian approach to extended object and cluster tracking using random matrices.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pages 1042–1059, 2008.

[KPN96] Knoblauch, R., Pietrucha, M., and Nitzburg, M. *Field studies of pedestrian walking speed and start-up time.* In: *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1538, no. 1, pages 27–38, 1996.

[KR10] Kramer, F. and Raddatz, M. *Das Bewegungsverhalten von fußgängern im Straßenverkehr auf Basis einer experimentellen Reihenuntersuchung.* In: *AT-Zonline: Verkehrsunfall und Fahrzeugtechnik*, vol. 12, pages 382–388, 2010.

[KSD16] Knill, C., Scheel, A., and Dietmayer, K. *A Direct Scattering Model for Tracking Vehicles with High-Resolution Radars.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 298–303, 2016.

[Kuh55] Kuhn, H. W. *The Hungarian method for the assignment problem.* In: *Naval Research Logistics Quarterly*, vol. 2, pages 83–97, 1955.

[LBS93] Lerro, D. and Bar-Shalom, Y. *Tracking With Debiased Consistent Converted Measurements Versus EKF.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 3, pages 1015–1022, 1993.

[LCB12] Lamard, L., Chapuis, R., and Boyer, J.-P. *A comparison of two different tracking algorithms is provided for real application.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 414–419, 2012.

[LCZT07] Lu, Y., Cohen, I., Zhou, X. S., and Tian, Q. *Feature Selection Using Principal Feature Analysis.* In: *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, pages 301–304, New York, NY, USA, 2007. ACM.

[LFY$^+$16] Li, X., Flohr, F., Yang, Y., Xiong, H., Braun, M., Pan, S., Li, K., and Gavrila, D. M. *A New Benchmark for Vison-Based Cyclist Detection.* In: *Proceedings of IEEE Intelligent Vehicle Symposium (IV)*, pages 1028–1033, 2016.

[LGO13] Lundquist, C., Granström, K., and Orguner, U. *An Extended Target CPHD Filter and a Gamma Gaussian Inverse Wishart Implementation.* In: *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pages 472–483, 2013.

[Lin06] Linzmeyer, D. *Real-Time Detection of Pedestrians from a Moving Vehicle Using Thermopile and Radar Sensor Fusion.* PhD dissertation, Ulm University, 2006.

[Liu10] Liu, F. *Objektverfolgung durch Fusion von Radar- und Monokameradaten auf Merkmalsebene für zukünftige Fahrerassistenzsysteme.* PhD dissertation, Karlsruhe Insitute for Technology, 2010.

[LJ03] Li, R. X. and Jilkov, P. V. *Survey of Maneuvering Target Tracking Part I: Dynamic Models.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pages 1333–1364, 2003.

[Lun11] Lundquist, C. *Sensor Fusion for Automotive Applications.* PhD dissertation, Linköping University, Sweden, 2011.

[LWW$^+$14] Li, C., Wang, J., Wang, L., Hu, L., and Gong, P. *Comparison of Classification Algorithms and Training Sample Sizes in Urban Land Classification with Landsat Thematic Mapper Imagery.* In: *Remote Sensing*, vol. 6, no. 2, pages 964–983, 2014.

[Mä09] Mählisch, M. *Filtersynthese zur simultanen Minimierung von Existenz-, Assoziations- und Zustandsunsicherheiten in der Fahrzeugumfelderfassung mit heterogenen Sensordaten.* PhD dissertation, Ulm University, 2009.

[MABSD98] Mazor, P., Averbuch, A., Bar-Shalom, Y., and Dayan, J. *Interacting Multiple Model Methods in Target Tracking: A Survey.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pages 103–123, 1998.

[Mah03] Mahler, R. *Multitarget Bayes filtering via first-order multitarget moments.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pages 1152–1178, 2003.

[Mah04] Mahler, R. *"Statistics 101" for multisensor, multitarget data fusion.* In: *IEEE Transactions on Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pages 53–64, 2004.

[Mah07a] Mahler, R. *PHD filters of higher order in target number.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pages 1523–1543, 2007.

[Mah07b] Mahler, R. *Statistical Multisource-Multitarget Information Fusion.* Artech House Inc., Norwood, 2007.

[Mah09] Mahler, R. *PHD filters for nonstandard targets, I: extended targets.* In: *12th International Conference on Information Fusion*, Seattle, WA, USA, 2009.

[Mah12] Mahler, R. *On multitarget jump-Markov filters.* In: *Proceedings of the 15th International Conference on Information Fusion*, pages 149–156, 2012.

[Mah13a] Mahler, R. *Divergence detectors for multitarget tracking algorithms.* In: *Proceedings SPIE 8745 , Signal Processing , Sensor Fusion and Target Recognition XXII*, 2013.

[Mah13b] Mahler, R. *"Statistics 102" for multisource-multitarget detection and tracking.* In: *IEEE Journal of selected topics in Signal Processing*, vol. 7, no. 3, pages 376–389, 2013.

[Mah14] Mahler, R. *Advances In Statistical Multisource-Multitarget Information Fusion.* Artech House Inc., Norwood, 2014.

[ME04a] Musicki, D. and Evans, R. *Joint integrated probabilistic data association: JIPDA.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, pages 1093–1099, 2004.

[ME04b] Musicki, D. and Evans, R. *Joint integrated probabilistic data association: JIPDA.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, pages 1093–1099, 2004.

[Mei15] Meissner, D. *Intersection-Based Road User Tracking Using a Classifying-Multiple-Model PHD Filter.* PhD dissertation, Ulm University, 2015.

[MES94] Musicki, D., Evans, R., and Stankovic, S. *Integrated probabilistic data association.* In: *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pages 1237–1241, 1994.

[ML90] McMillan, C. J. and Lim, S. S. *Data Association Algorithms for Multiple Target Tracking.* Ad-a231 688, Defence Research Establischment Ottawa, Ottawa, 1990.

[MMD10] Munz, M., Mählisch, M., and Dietmayer, K. *Generic centralized multi sensor data fusion based on probabilistic sensor and environment models for driver assistance systems.* In: *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 1, pages 6–7, 2010.

[MRD13] Meissner, D., Reuter, S., and Dietmayer, K. *Road user tracking at intersections using a multiple-model PHD filter.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 377–382, 2013.

[MRSD14] Meissner, D., Reuter, S., Strigel, E., and Dietmayer, K. *Intersection-Based Road User Tracking Using a Classifying Multiple-Model PHD Filter.* In: *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pages 21–33, 2014.

[MRWD13] Meissner, D., Reuter, S., Wilking, B., and Dietmayer, K. *Road user tracking using a Dempster-Shafer based classifying multiple-model PHD filter.* In: *Proceedings of the 16th International Conference on Information Fusion*, pages 1236–1242, 2013.

[MSRD06] Mählisch, M., Schweiger, R., Ritter, W., and Dietmayer, K. *Multisensor vehicle tracking with the probability hypothesis density filter.* In: *Proceedings of the 9th International Conference on Information Fusion*, pages 1–8, 2006.

[MTBW17] Martin, H., Tschabuschnig, K., Bridal, O., and Watzenig, D. *Functional Safety of Automated Driving Systems: Does ISO 26262 Meet the Challenges?*, pages 387–416. Springer International Publishing, Cham, 2017.

[Mun57] Munkres, J. *Algorithms for the assignment and transportation problems.* In: *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pages 32–38, 1957.

[Mun11] Munz, M. *Generisches Sensorfusionsframework zur gleichzeitigen Zustands- und Existenzschätzung für die Fahrzeugumfelderkennung.* PhD dissertation, Ulm University, 2011.

[Mur68] Murty, K. *An algorithm for ranking all the assignments in order of increasing cost.* In: *Operations Research*, vol. 16, pages 682–687, 1968.

[MVV11] Mahler, R., Vo, B.-T., and Vo, B.-N. *CPHD filtering with unknown clutter rate and detection profile.* In: *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pages 3497–3513, 2011.

[NMTS05] Nguyen, V., Martinelli, A., Tomatis, N., and Siegwart, R. *A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics.* In: *IEEE/RSJ Conference on Intelligent Robotics and Systems*, 2005.

[Nus16] Nuss, D. S. *A Random Finite Set Approach for Dynamic Occupancy Grid Maps.* PhD dissertation, Ulm University, 2016.

[Ohl14] Ohl, S. *Fusion von Umfeld wahrnehmenden Sensoren in städtischer Umgebung.* PhD dissertation, Technical University Carolo-Wilhelmina Braunschweig, 2014.

[Ott13] Otto, C. *Fusion of Data from Heterogeneous Sensors with Distributed Fields of View and Situation Evaluation for Advanced Driver Assistance Systems.* PhD dissertation, Karlsruhe Institute of Technology, 2013.

[Pan07] Panta, K. *Multi-Target Tracking Using 1st Moment of Random Finite Sets.* PhD dissertation, The University of Melbourne, Australia, 2007.

[Pie15] Pietzsch, S. *Modellgestützte Sensordatenfusion von Laserscanner und Radar zur Erfassung komplexer Fahrzeugumgebungen.* PhD dissertation, Technical Univeristy Munich, 2015.

[PJ05] Pulford, G. W. and J, S. D. *A Gaussian Mixture Filter for Near-Far Object Tracking.* In: *Proceedings of the 8th International Conference on Information Fusion*, 2005.

[PKS08] Punithakumar, K., Kirubarajan, T., and Sinha, A. *Multiple-model probability hypothesis density filter for tracking maneuvering targets.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pages 87–98, 2008.

[Pla99] Platt, J. C. *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.* In: *Advances In Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[PLN09] Premebida, C., Ludwig, O., and Nunes, U. *Exploiting LIDAR-based Features on Pedestrian Detection in Urban Scenarios.* In: *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems*, pages 18–23, 2009.

[Pra11] Prat, C. *Sensordatenfusion und Bildverarbeitung zur Objekt- und Gefahrenerkennung.* PhD dissertation, Technical University Carolo-Wilhelmina Braunschweig, 2011.

[PVTM09] Pasha, A., Vo, B.-N., Tuan, H. D., and Ma, W.-K. *A Gaussian Mixture PHD Filter for Jump Markov System Models.* In: , vol. 45, no. 3, pages 919–936, 2009.

[PVV16] Punchihewa, Y., Vo, B.-N., and Vo, B.-T. *A Generalized Labeled Multi-Bernoulli filter for maneuvering targets.* In: *19th International Conference on Information Fusion*, 2016.

[RAG04] Ristic, B., Arulampalam, S., and Gordon, N. *Beyond the Kalman Filter: Particle Filters for Tracking Applications.* Artech House Inc., 2004.

[RD09] Reuter, S. and Dietmayer, K. *Fuzzy estimation and segmentation for laser range scans.* In: *Proceedings of the 12th International Conference on Information Fusion*, pages 1974–1981, 2009.

[Rei79] Reid, D. *An algorithm for tracking multiple targets.* In: *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pages 843–854, 1979.

[Reu14] Reuter, S. *Multi-Object Tracking Using Random Finite Sets.* PhD dissertation, Ulm University, 2014.

[RGV$^+$13] Rezatofighi, S. H., Gould, S., Vo, B.-N., et al. *A multiple model probability hypothesis density tracker for time-lapse cell microscopy sequences.* In: *Information Processing in Medical Imaging (IPMI)*, 2013.

[RHG14] Roth, M., Hendeby, G., and Gustafsson, F. *EKF/UKF maneuvering target tracking using coordinated turn models with polar/Cartesian velocity.* In: *Proceedings of the 17th International Conference on Information Fusion*, 2014.

[RMD12] Reuter, S., Meissner, D., and Dietmayer, K. *Multi-object tracking at intersections using the cardinalized probability hypothesis density filter.* In: *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1172–1177, 2012.

[RMWD13] Reuter, S., Meissner, D., Wilking, B., and Dietmayer, K. *Cardinality balanced multi-target multi-Bernoulli filtering using adaptive birth distributions.* In: *Proceedings of the 16th International Conference on Information Fusion*, pages 1608–1615, 2013.

[RSD15] Reuter, S., Scheel, A., and Dietmayer, K. *The Multiple Model Labeled Multi-Bernoulli Filter.* In: *Proceedings of the 18th International Conference on Information Fusion*, Washington DC, 2015.

[RVVD14] Reuter, S., Vo, B.-T., Vo, B.-N., and Dietmayer, K. *The labeled multi-Bernoulli filter.* In: *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pages 3246–3260, 2014.

[SA71] Sorenson, H. and Alspach, D. *Recursive Bayesian estimation using Gaussian sums.* In: *Automatica*, vol. 7, no. 4, pages 465–479, 1971.

[Sch93] Schölkopf, B. *The kernel trick for distances.* In: *TR MSR 2000-51, Microsoft Research*, pages 5–3, 1993.

[SD04] Streller, D. and Dietmayer, K. *Object Tracking and Classification Using a Multiple Hypothesis Approach.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 808–812, 2004.

[SFD02] Streller, D., Fürstenberger, K., and Dietmayer, K. *Vehicle and Object Models for Robust Tracking in Traffic Scenes using Laser Range Images.* In: *Proceedings of the 5th IEEE International Conference on Intelligent Transportation Systems*, pages 118–123, 2002.

[SGM10] Song, F., Guo, Z., and Mei, D. *Feature selection using principal component analysis.* In: *International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM)*, pages 28–30, 2010.

[SGM$^+$14] Scheel, A., Granström, K., Meissner, D., Reuter, S., and Dietmayer, K. *Tracking and data segmentation using a GGIW filter with mixture clustering.* In: *Proceedings of the 17th International Conference on Information Fusion*, 2014.

[Sha76] Shafer, G. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, 1976.

[Sit64] Sittler, R. *An optimal data association problem in surveillance theory.* In: *IEEE Transactions on Military Electronics*, vol. 8, no. 2, pages 125–139, 1964.

[SK94] Smets, P. and Kennes, R. *The transferable belief model.* In: *Artificial Intelligence*, vol. 66, no. 2, pages 191–234, 1994.

[SKFM14] Schubert, E., Kunert, M., Frischen, A., and Menzel, W. *A Multi-Reflection-Point Target Model for Classification of Pedestrians by Automotive Radar.* In: *Proceedings of the 11th European Radar Conference*, pages 181–184, 2014.

[SKRD16] Scheel, A., Knill, C., Reuter, S., and Dietmayer, K. *Multi-sensor multi-object tracking of vehicles using high-resolution radars.* In: *IEEE Intelligent Vehicles Symposium (IV)*, pages 558–565, 2016.

[Sme90] Smets, P. *The combination of evidence in the transferable belief model.* In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pages 447–458, 1990.

[SMKM15] Schubert, E., Meinl, F., Kunert, M., and Menzel, W. *High Resolution Automotive Radar Measurements of Vulnerable Road Users - Pedestrians & Cyclists.* In: *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4, 2015.

[SRKW16] Schlichenmaier, J., Roos, F., Kunert, M., and Waldschmidt, C. *Adaptive clustering for contour estimation of vehicles for high-resolution radar.* In: *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 2016.

[Sta15] Statistiches-Bundesamt. *Unfälle von Güterkraftfahrzeugen im Straßenverkehr.*, 2015.

[Stü04] Stüker, D. *Heterogene Sensordatenfusion zur robusten Objektverfolgung im automobilen Straßenverkehr.* PhD dissertation, Carl von Ossietzky-University Oldenburg, 2004.

[Stü06] Stüker, D. *Heterogene Sensordatenfusion zur robusten Objektverfolgung im automobilen Straßenverkehr.* PhD dissertation, Carl von Ossietzky-University Oldenburg, 2006.

[SVV08] Schuhmacher, D., Vo, B.-N., and Vo, B.-T. *A consistent metric for performance evaluation of multi-object filters.* In: *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pages 3447–3457, 2008.

[SW03] Sidenbladh, H. and Wirkander, S.-L. *Tracking random sets of vehicles in terrain.* In: *Conference on Computer Vision and Pattern Recognition Workshop*, page 98, 2003.

[TBF05] Thrun, S., Burgard, W., and Fox, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).* The MIT Press, 2005.

[TK10] Tu, P.-J. and Kiang, J.-F. *Estimation on Location, Velocity, and Acceleration With High Precision for Collision Avoidance.* In: *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pages 374–379, 2010.

[Ung94] Unger, H.-G. *Hochfrequenztechnik in Funk und Radar.* Springer, 1994.

[VdM04] Van der Merwe, R. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models.* PhD dissertation, Oregon Health & Science University, 2004.

[VJ01] Viola, P. and Jones, M. *Rapid object detection using a boosted cascade of simple features.* In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, 2001.

[VM06] Vo, B.-N. and Ma, W.-K. *The Gaussian mixture probability hypothesis density filter.* In: *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pages 4091–4104, 2006.

[Vo08] Vo, B.-T. *Random Finite Sets in Multi-Object Filtering.* PhD dissertation, The University of Western Australia, School of Electrical, Electronic and Computer Engineering, 2008.

[VPT06] Vo, B.-N., Pasha, A., and Tuan, H. D. *A Gaussian mixture PHD filter for nonlinear jump Markov models.* In: *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3162–3167, 2006.

[VSD05] Vo, B.-N., Singh, S., and Doucet, A. *Sequential Monte Carlo methods for multitarget filtering with random finite sets.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pages 1224–1245, 2005.

[Vu09] Vu, T.-D. *Vehicle Perception: Localization, Mapping with Detection, Classification and Tracking of Moving Objects.* PhD dissertation, Grenoble Insitute of Technology, France, 2009.

[VV11] Vo, B.-T. and Vo, B.-N. *A random finite set conjugate prior and application to multi-target tracking.* In: *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 431–436, 2011.

[VV13a] Vo, B.-T. and Vo, B.-N. *Labeled random finite sets and multi-object conjugate priors.* In: *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pages 3460–3475, 2013.

[VV13b] Vo, B.-T. and Vo, B.-N. *Tracking, identification, and classification with random finite sets.* In: *Proceedings of the SPIE 8745, Signal Processing, Sensor Fusion, and Target Recognition XXII.* International Society for Optics and Photonics, 2013.

[VV17] Vo, B.-N. and Vo, B.-T. *Multi-Sensor Multi-object Tracking with the Generalized Labeled Multi-Bernoulli Filter*, 2017. Eprint-arXiv:1702.08849v1.

[VVC07] Vo, B.-T., Vo, B.-N., and Cantoni, A. *Analytic implementations of the cardinalized probability hypothesis density filter.* In: *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pages 3553–3567, 2007.

[VVC09] Vo, B.-T., Vo, B.-N., and Cantoni, A. *The cardinality balanced multitarget multi-Bernoulli filter and its implementations.* In: *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pages 409–423, 2009.

[VVH17] Vo, B.-T., Vo, B.-N., and Hoang, H. G. *An Efficient Implementation of the Generalized Labeled Multi-Bernoulli Filter.* In: *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pages 1975–1987, 2017.

[VVHM13] Vo, B.-T., Vo, B.-N., Hoseinnezhad, R., and Mahler, R. *Robust multi-Bernoulli filtering.* In: *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pages 399–409, 2013.

[VVPS10] Vo, B.-T., Vo, B.-N., Pham, N.-T., and Suter, D. *Joint detection and estimation of multiple objects from image observations.* In: *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pages 5129–5141, 2010.

[Wen08] Wender, S. *Multisensorsystem zur erweiterten Fahrzeugumfelderfassung.* PhD dissertation, Ulm University, 2008.

[XZZ05] Xu, X., Zhou, D., and Zhou, Y. *A New Clustering Algorithm Based on Distance and Density.* In: *Proceedings of International Conference on Services Systems and Services Management*, vol. 2, pages 1016–1021, 2005.

[YJH17] Yi, W., Jiang, M., and Hoseinnezhad, R. *The Multiple Model Vo–Vo Filter.* In: *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 2, pages 1045–1054, 2017.

[Zad84] Zadeh, L. A. *Review of A Mathematical Theory of Evidence.* In: *AI Magazine*, vol. 5, no. 3, pages 81–83, 1984.

[ZBS$^+$14] Ziegler, J., Bender, P., Schreiber, M., et al. *Making Bertha Drive - An Autonomous Journey on a Historic Route.* In: *IEEE Intelligent Transportation Systems Magazine*, pages 8–20, 2014.

[ZE02] Zadrozny, B. and Elkan, C. *Transforming Classifier Scores into Accurate Multiclass Probability Estimates*, 2002.

[Zha00] Zhang, Z. *A flexible new technique for camera calibration.* In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pages 1330–1334, 2000.

[ZR99] Zollo, S. and Ristic, B. *On the Choice of the Coordinate System and Tracking Filter for the Track-while-scan Mode of an Airborne Pulse Doppler Radar.* Dsto-tr-0926, DSTO Electronics and Surveillance Research Laboratory, Salisbury, Australia, 1999.

[ZSKS06] Zhao, H., Shao, X. W., Katabira, K., and Shibasaki, R. *Joint tracking and classification of moving objects at intersection using a single-row laser range* *scanner*. In: *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 287–294, 2006.

[ZZC$^+$09] Zhao, H., Zhang, Q., Chiba, M., Shibasaki, R., Cui, J., and Zha, H. *Moving Object Classification using Horizontal Laser Scan Data.* In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2424–2430, 2009.

Lebenslauf

Name	Ganesh Nageswaran
Berufstätigkeit	
seit 08/2014	Wissenschaftlicher Mitarbeiter, Technische Universität Kaiserslautern
06/2009 - 09/2010	Ingenieur, TVS Brakes India GmbH, Chennai, Indien
Hochschulbildung	
10/2010 – 03/2014	Masterstudium, Commercial Vehicle Technology, Technische Universität Kaiserslautern, Abschluss: Master of Science
08/2005 – 05/2009	Bachelorstudium, Maschinenbau, Anna Universität, Indien, Abschluss: Bachelor of Engineering
Schulbildung	
06/2004 – 05/2005	Higher Secondary School, Computer Science, Prince Matriculation Schule, Indien, Abschluss: Higher Secondary Examination (Class XII)