Moving objects detection and tracking by omnidirectional sensors of a mobile robot

Marković, Ivan

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ivan Marković

MOVING OBJECTS DETECTION AND TRACKING BY OMNIDIRECTIONAL SENSORS OF A MOBILE ROBOT

DOCTORAL THESIS

Zagreb, 2014

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ivan Marković

MOVING OBJECTS DETECTION AND TRACKING BY OMNIDIRECTIONAL SENSORS OF A MOBILE ROBOT

DOCTORAL THESIS

Supervisor: Professor Ivan Petrović, PhD

Zagreb, 2014

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ivan Marković

OTKRIVANJE I PRAĆENJE GIBAJUĆIH OBJEKATA SVESMJERNIM SENZORIMA MOBILNOGA ROBOTA

DOKTORSKI RAD

Mentor: Prof. dr. sc. Ivan Petrović

Zagreb, 2014.

Doctoral thesis was written at the University of Zagreb, Faculty of Electrical Engineering and Computing, Departement of Control and Computer Engineering.

Supervisor: Professor Ivan Petrović, PhD

Thesis contains 153 pages

Thesis no.:

DOCTORAL THESIS EVALUATION COMMITTEE

Professor Davor Petrinović, PhD
University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia

Professor Nedjeljko Perić, PhD
University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia

Associate Professor Robert Cupec, PhD
Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Croatia

DOCTORAL THESIS DEFENCE COMMITTEE

Professor Davor Petrinović, PhD
University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia

Associate Professor Mato Baotić, PhD
University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia

Associate Professor Robert Cupec, PhD
Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Croatia

Professor François Chaumette, PhD
INRIA Rennes-Bretagne Atlantique, IRISA, France

Professor Patrick Danès, PhD
LAAS-CNRS & University of Toulouse III Paul Sabatier, France

Date of doctoral thesis defence: March 7, 2014.

ABOUT THE SUPERVISOR

IVAN PETROVIĆ was born in Klobuk, Bosnia and Herzegovina in 1961. He received B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia, in 1983, 1989 and 1998, respectively.

For the first ten years after graduation he was with the Institute of Electrical Engineering of Končar Corporation in Zagreb, where he had been working as a research and development engineer for control and automation systems of electrical drives and industrial plants. From 1994 he has been working at the Department of Control and Computer Engineering at FER. In November 2009 he was promoted to Full Professor. He has actively participated as a collaborator or principal investigator on 30 national and 15 international scientific projects. Currently, he coordinates EU FP7 project "Centre of Research Excellence for Advanced Cooperative Systems" (ACROSS). He published 40 papers in scientific journals and more than 150 papers in proceedings of international conferences in the area of control engineering and automation applied to control mobile robots and vehicles, power systems, electromechanical systems and other technical systems.

Prof. Petrović is a member of IEEE, Croatian Academy of Engineering (HATZ), president of the Croatian Robotics Society, vice-president of the Technical committee on Robotics of the International Federation of Automatic Control (IFAC), executive committee member of the Federation of International Robot-soccer Association (FIRA), and a founding member of the iSpace Laboratory Network. He is also a member of the Croatian Society for Communications, Computing, Electronics, Measurements and Control (KoREMA) and Editor-in-Chief of the Automatika journal. He received the award "Professor Vratislav Bedjanič" in Ljubljana for outstanding M.Sc. thesis in 1990 and silver medal "Josip Lončar" from FER for outstanding Ph.D. thesis in 1998. For scientific achievements he received the award "Rikard Podhorsky" from the Croatian Academy of Engineering, "National Science Award of the Republic of Croatia" and the gold plaque "Josip Lončar" from FER in 2008, 2011 and 2013, respectively.

O MENTORU

IVAN PETROVIĆ rođen je u Klobuku, Bosna i Hercegovina, 1961. godine. Diplomirao je, magistrirao i doktorirao u polju elektrotehnike na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva (FER), 1983., 1989. odnosno 1998. godine.

Prvih deset godina po završetku studija radio je na poslovima istraživanja i razvoja sustava upravljanja i automatizacije elektromotornih pogona i industrijskih postrojenja u Končar – Institutu za elektrotehniku. Od svibnja 1994. radi u Zavodu za automatiku i računalno inženjerstvo FER-a. U studenom 2009. godine izabran je u zvanje redovitog profesora. Sudjelovao je kao suradnik ili voditelj na 30 nacionalnih i 15 međunarodnih znanstvenih projekata. Trenutačno je koordinator EU FP7 projekta „Centre of Research Excellence for Advanced Cooperative Systems" (ACROSS). Objavio je 40 znanstvenih radova u časopisima i više od 150 znanstvenih radova u zbornicima skupova u području automatskog upravljanja i estimacije s primjenom u upravljanju mobilnim robotima i vozilima te energetskim, elektromehaničkim i drugim tehničkim sustavima.

Prof. Petrović član je stručne udruge IEEE, Akademije tehničkih znanosti Hrvatske (HATZ), predsjednik Hrvatskog društva za robotiku, dopredsjednik tehničkog odbora za robotiku međunarodne udruge IFAC, član izvršnog odbora međunarodne udruge FIRA te suutemeljitelj međunarodne udruge „The iSpace Laboratory Network". Član je i upravnog odbora Hrvatskog društva za komunikacije, računarstvo, elektroniku, mjerenja i automatiku (KoREMA) i glavni i odgovorni urednik časopisa Automatika. Godine 1990. primio je u Ljubljani nagradu „Prof. dr. Vratislav Bedjanič" za posebno istaknuti magistarski rad, 1998. srebrnu plaketu „Josip Lončar" FER-a za posebno istaknutu doktorsku disertaciju, a za znanstvena je postignuća dobio 2008. godine nagradu „Rikard Podhorsky" Akademije tehničkih znanosti Hrvatske, 2011. godine „Državnu nagradu za znanost" i 2013. godine zlatnu plaketu „Josip Lončar" FER-a.

ACKNOWLEDGEMENTS

I am greatly indebted to my mentor, Prof. Ivan Petrović, for giving me the opportunity of being a member of his research group, to learn from his great experience, and for going well beyond the call of duty in providing help and priceless advice.

I would like to express my thanks to friends and colleagues from the Department of Control and Computer Engineering for selfless sharing of knowledge, countless hours of discussions, and without whom this work would not have been attainable and would have been far less enjoyable.

I am also indebted to Prof. François Chaumette, the leader of the Lagadic group in INRIA Rennes-Bretagne Atlantique, for accepting me as a visiting researcher and with whom it has been an honor to work together. My thanks also go to all the members of the Lagadic group for providing aid and support in both scientific and francophone matters. I am truly privileged to have been a part of this excellent research group.

I would also like to thank Prof. Patrick Danès, Alban Portello and Sylvain Argentieri for our truly pleasant joint work and collaboration. I have learned a lot from our numerous discussions which have always given me a new perspective on the problems of robot audition.

It goes without saying that my deepest gratitude goes to my family for their infinite wisdom and support, without whom none of this would have been possible.

ABSTRACT

Directional data emerge often in many aspects of mobile robotics. Measurements from various sensors yield direction-only information of the objects of interest. Since probabilistic methods have been widely accepted and successfully utilized in many mobile robotics problems, question arises if such modeling could offer prospects in the context of probabilistic representation of directional data gathered by a mobile robot. One of the goals of this thesis is to develop directional statistics based methods for moving object tracking by omnidirectional sensors of a mobile robot. In that mindset the thesis addresses moving object tracking via two different problems, namely speaker detection, localization and tracking with a microphone array, and moving object detection, tracking and following with an omnidirectional camera. Furthermore, in the thesis we also address the challenge of heterogenous sensor fusion through the prism of moving object tracking.

The speaker localization and tracking problem is solved by modeling the measurement of a microphone array with a convex mixture of von Mises distributions, where the tracking is thus performed by way of particle filtering. This approach is later extended, to circumvent the sample based techniques, by keeping the tracking procedure fully in the analytical domain via a mixture filter based on the von Mises distribution. Furthermore, a prerequisite for robust speaker localization and tracking is voice activity detection. In the thesis we analyze this problem from the standpoint of model based voice activity detection methods which are enhanced by supervised learning algorithms. Specifically, a detector based on the Rayleigh and Rice distributions is coupled with a number of carefully chosen spectral and temporal features in a supervised classification approach. Apropos of omnidirectional camera, where spherical projection model coupled with displacement information from motor encoders is proposed to segment out features that do not belong to the static scene around the mobile robot, directional statistics is used in the context of movement tracking on the sphere with a Bayesian filter based on the von Mises-Fisher distribution. Finally, fusion of heterogenous sensors for object tracking is analyzed in a comparative study of the extended information filter, the unscented information filter and the particle filter.

KEYWORDS: moving object tracking, directional statistics, speaker localization, von Mises distribution, microphone array, voice activity detection, sensor fusion, mobile robotics

SAŽETAK

OTKRIVANJE I PRAĆENJE GIBAJUĆIH OBJEKATA SVESMJERNIM SENZORIMA MOBILNOGA ROBOTA

Informacije o smjeru, poput azimuta ili smjera gibanja te elevacije često se javljaju u mnogim primjenama uključujući i mobilnu robotiku. U mobilnoj je robotici problem modeliranja takvih veličina prisutan zbog korištenja senzora koji mogu mjeriti isključivo kutne veličine detektiranih objekata. Razmatrajući navigaciju mobilnih robota, kutne veličine prisutne su u odometriji, pošto je smjer gibanja robota kutna varijabla. Perspektivne kamere se koriste u problemu navigacije s kutnim vrijednostima radi ispravljanja i potiskivanja pogrešaka odometrije. Razmatrajući praćenje smjera gibajućih objekata, korištene su svesmjerne i perspektivne kamere postavljene na mobilnoga robota. Također, u području akustike postoji znatan broj radova koji se bave lokalizacijom i praćenjem govornika, problem koji se tipično rješava koristeći polje mikrofona od četiri, osam ili samo dva mikrofona (inspirirano biološkim auditornim funkcijama). Za sve navedene pristupe, bez obzira radi li se o odometriji, kamerama ili polju mikrofona na mobilnom robotu ili letjelici, svi oni moraju raditi s kutnim veličinama, tj. samo s azimutom ili azimutom i elevacijom. Radeći s ovakvim podacima, pogotovo u prisustvu nesigurnosti, predstavlja se problem njihovog prikazivanja u vjerojatnosnim okvirima. U većini se slučajeva kutne veličine tretiraju koristeći Gaussove slučajne varijable, što je opravdano kada su varijacije u iznosima kuteva male. U suprotnome, ne ulazeći trenutno u detalje nedostataka Gaussove razdiobe u slučaju modeliranja kutnih slučajnih varijabli, može se napomenuti da u području statistike posvećene kutnim veličinama postoji nekoliko kružnih razdioba: von Misesova, kardoidna, „namotana" Poissonova, Cauchijeva te Gaussova razdioba. „Namotane" razdiobe su analitički zahtjevne zbog elementa beskonačnog zbroja. Glavna upotreba kardoidne distribucije je aproksimacija von Misesove s velikom nesigurnosti. Stoga, najčešće se koristi upravo von Misesova razdioba za koju se može pokazati da proizlazi iz istih pretpostavki kao i Gaussova, ali u slučaju kada se razmatra definicija razdiobe na jediničnoj kružnici. Iako nije do sada korištena u velikom broju istraživanja, robotičarska je zajednica prepoznala prednosti te razdiobe u problematici lokalizacije i navigacije. Cilj je ovoga rada razviti metode praćenja gibajućih objekata svesmjernim senzorima mobilnoga robota koristeći kružne razdiobe i usmjernu statistiku. Iako ove razdiobe imaju mnoge prednosti, ipak ih nije jednostavno uključiti u postojeće paradigme.

Prvi razmatrani problem u disertaciji je problem lokalizacije i praćenja govornika poljem

mikrofona. Ovaj je problem riješen modelirajući mjerenje polja mikrofona konveksnim zbrojem von Misesovih razdioba gdje je praćenje ostvareno čestičnim filtrom. Ovaj je pristup potom nadalje razvijen u okviru Bayesovih metoda estimacije na način da se je praćenje riješilo u potpunosti zbrojem von Misesovih razdioba. Na taj je način cijeli postupak ostao analitički i u eksplicitnom obliku koristeći samo von Misesovu razdiobu. Nadalje, razmatran je postupak estimacije ne samo azimuta govornika već i same udaljenosti. Uzimajući u obzir i pomake robota moguće je implicitno raditi triangulaciju te na taj način estimirati udaljenost govornika od polja mikrofona. U ovome su dijelu korištena dva mikrofona na sfernoj glavi te je dobiveni model senzora bila tzv. funkcija pseudo-vjerodostojnosti. Kako bi se poboljšala estimacija udaljenosti, diskretna je funkcija pseudo-vjerodostojnosti opisana zbrojem kontinuiranih von Misesovih razdioba minimizirajući kvadratnu grešku te je korišten čestični filtar za postupak estimacije. Urađeni su eksperimenti koji su pokazali da u prosjeku ovakav postupak poboljšava točnost estimacije udaljenosti.

Kako bi metoda lokalizacije radila pouzdano i u uvjetima promjenjive okoline s prisutnim šumom, razvijen je i algoritam otkrivanja govorne aktivnosti. Algoritam je temeljen na modelsko-statističkim metodama koje modeliraju razdiobu koeficijenata diskretne Fourierove transformacije. U radu su modelirani koeficijenti kompleksnom Gaussovom razdiobom te je praćena distribucija ovojnice signala koja pod hipotezom prisutnosti samo šuma daje Rayleighovu razdiobu, a pod pretpostavkom prisutnosti i govora i šuma daje Riceovu distribuciju. Izračunat je omjer funkcija vjerodostojnosti ove dvije razdiobe te je dobivena metoda otkrivanja govora zasnovana na Rayleighovoj i Riceovoj razdiobi. Kako bi se nadalje unaprijedile performanse otkrivanja govora, metodi su dodane različite spektralne i temporalne značajke što je činio skup od 71 značajke na temelju kojeg je trebalo otkrivati prisutnost govora. Međutim, prije korištenja algoritama nadziranog učenja, skup je analiziran metodom parcijalne zajedničke informacije čime su eliminirane korelirane značajke. Time se je dobio reducirani skup koji se je sastojao od ukupno 13 značajki. Potom su korištena i uspoređena tri algoritma nadziranog učenja: stroj potpornih vektora, Boost i umjetne neuronske mreže. Usporedba je napravljena koristeći bazu koja je sadržavala šest različitih govornika s tri različita tipa šuma i tri različite razine signal-šum.

Sljedeće područje istraživanja koje je razmatrano u disertaciji je otkrivanje i praćenje gibajućih objekata svesmjernom kamerom. Svesmjerne kamere imaju prednost nad standardnim kamerama u vidu puno većeg kuta gledanja. Međutim, te prednosti prate i nedostaci poput distorzija uzrokovanih ogledalom ili lećom te nižom rezolucijom jer ipak prikazuju puno veće područje na jednakoj veličini senzora. U većini istraživanja svesmjerne kamere koristile su se za procjenu vlastitog gibanja i za lokalizaciju. Nadalje, problem detekcije gibajućih objekata u slici je dodatno otežan činjenicom da kamera na mobilnom robotu zbog vlastitog gibanja bilježi i vlastiti optički tok. U ovome je radu predložena metoda detekcije gibajućih objekata koja se zasniva na računanju optičkog toka u svesmjernoj slici. Nakon što je izračunat optički tok, pošto je korištena kalibrirana kamera, svaka točka toka može se prikazati na jediničnoj sferi. Potom, uzimajući podatke o rotaciji i translaciji koordinatnih sustava iz odometrije robota, na temelju prethodne i trenutne slike može se definirati luk glavne kružnice na kojoj bi krajnja točka toka trebala ležati ako je optički tok bio uzrokovan samo gibanjem robota. U suprotnome, smatra se da je optički tok bio uzrokovan gibajućim objektom. Nakon što su vektori optičkog toka grupirani na temelju sličnosti azimuta, ele-

vacije i duljine, izračunat je centar mase vektora što je dalo točku na jediničnoj sferi koja je predstavljala mjerenje ovoga senzora. To je mjerenje nadalje modelirano razdiobom na jediničnoj sferi, tj. von Mises-Fisherovom razdiobom. Kao i u slučaju praćenja govornika, u ovome je dijelu disertacije razvijena metoda praćenja gibajućeg objekta u potpunosti zasnovana na von Mises-Fisherovoj razdiobi. Na kraju, da bi mobilni robot pratio gibajući objekt, proračunat je algoritam upravljanja metodom vizualnog slijeđenja.

Posljednje poglavlje disertacije, pored zaključka, posvećeno je fuziji senzora. U ovome su poglavlju razmatrane dvije metode fuzije: centralizirana i hijerarhijska fuzija. Centralizirana fuzija se zasniva na ideji da svaki senzor centru fuzije odašilje svoje trenutno mjerenje, odnosno svoju funkciju vjerodostojnosti, koji potom izvršava algoritam estimacije tako što na temelju modela gibanja vrši predikciju i potom na temelju svih mjerenja radi korekciju. Pod pretpostavkom uvjetne nezavisnosti mjerenja, korekcija se može računati na način da se sve funkcije vjerodostojnosti međusobno pomnože. U slučaju hijerarhijske fuzije, svaki od senzora lokalno računa estimaciju te potom centru fuzije šalje procijenjenu vrijednost zajedno sa svojom nesigurnosti. Centar fuzije, koji također vrši estimaciju, prije nego li izvršava fuziju mora uzeti u obzir da radi sa estimacijama, a ne sa mjerenjima, jer bi u tom slučaju višestruko brojao informaciju iz predikcije svakog od senzora. U tom slučaju fuzija se ispravno radi na način da se svaka estimacija senzora podijeli sa svojom predikcijom kako bi se izlučila funkcija vjerodostojnosti i samo ona koristila pri korekciji estimacije centra fuzije. Ovaj problem je analiziran na primjeru praćenja gibajućih objekata dvama senzorima. Predstavljeno je rješenje zasnovano na proširenom informacijskom filtru, nederivacijskom informacijskom filtru te čestičnom filtru. Također, predloženo je i rješenje za slučaj ako se vrši fuzija različitih vrsta filtara, npr. čestičnog i proširenog informacijskog filtra. Na kraju, predstavljeni su rezultati eksperimenta praćenja više ljudi trima heterogenim senzorima: laserskim senzorom udaljenosti, poljem mikrofona i RGB-D kamerom.

KLJUČNE RIJEČI: praćenje gibajućih objekata, usmjerna statistika, lokalizacija govornika, von Misesova razdioba, polje mikrofona, otkrivanje govorne aktivnosti, fuzija senzora, mobilna robotika

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ML	maximum likelihood
MAP	maximum a posteriori
GCC	generalized cross-correlation
TDOA	time difference of arrival
DOA	direction of arrival
PHAT	phase transform
MUSIC	multiple signal classification
IPD	interaural phase difference
ILD	interaural level difference
SIR	sequential importance resampling
WGN	white Gaussian noise
LRS	laser range sensor
FOV	field-of-view
MHT	multiple hypothesis tracker
JPDAF	joint probabilistic data association filter
ASJPDAF	adaptive sample-based joint probabilistic data association filter
RMSE	root-mean-square error
VAD	voice activity detector
DFT	discrete Fourier transform
SVM	support vector machine
SNR	signal-to-noise ratio
MFCC	mel frequency cepstral coefficient
PMCC	power normalized cepstral coefficient
LR	likelihood ratio
ANN	artificial neural network
GD	Gaussian distribution
GGD	generalized Gaussian distribution
RRD	Rayleigh and Rice distribution
MCRA	minima-controlled recursive averaging
DD	decision directed
TP	true positive
TN	true negative
FP	false positive

FN	false negative
KDE	kernel density estimator
ROC	receiver operating characteristics
AUC	area under the receiver operating characteristics curve
SDR	speech detection rate
FAR	false alarm rate
MCC	Matthew's correlation coefficient
PMI	partial mutual information
IVS	input variable selection
ZCR	zero-crossing rate
SF	spectral flux
SR	spectral rolloff
SC	spectral centroid
SBW	spectral bandwidth
RBF	radial basis function
MLP	multilayer perceptron network
RPROP	resilient propagation
ERR	error rate
VM	von Mises
VMF	von Mises-Fisher
WC	wrapped Cauchy
PDF	probability density function
KF	Kalman filter
EKF	extended Kalman filter
UKF	unscented Kalman filter
DKF	decentralised Kalman filter
EIF	extended information filter
UIF	unscented information filter
PF	particle filter
KL	Kullback-Leibler
MC	Monte-Carlo

1

Introduction

Do not follow the path.
Go where there is no path and begin the trail.

— Ashanti proverb

The introduction chapter presents and elaborates the motivation behind the research conducted in the thesis. We shall start by discussing what exactly do we mean by omnidirectional sensors and which in particular were used in the thesis. Ensuingly, we analyze the measurements potential of the omnidirectional sensors and their dimension. We shall also discuss and break down the title of thesis and see how it relates to the contents of the thesis research. Furthermore, we shall discuss why the focus was set on directional measurements and how they appear in the field of mobile robotics. Subsequently, a discussion is in order on why circular distributions were utilized as a basis for estimation, inlieu of the pervasive Gaussian distribution. Thereafter, the original contributions of the thesis will be presented and described in more detail, while at the end of the chapter, outline and structure of the thesis will be sketched along with a brief summary of the chapter contents.

1.1 Motivation and problem statement

What do we consider by omnidirectional sensors? Essentially these are sensors that enable us to capture the surrounding scene of the sensor in a single frame, i.e. sensors which capture data from all directions. This pertains more to the field-of-view in the horizontal plane, which for omnidirectional sensors is by definition 360°. In this thesis two omnidirectional sensors were used, namely a microphone array and an omnidirectional camera. What is in common for both of these sensors, in the application manner used in the thesis, is that they provide directional measurements. For the case of the omnidirectional cameras these are measurements of the direction in 3D, i.e. a vector on the unit sphere. For the case of the microphone array these are angular measurements in 2D, i.e. measurements of the bearing [1] (azimuth) or a vector on the unit sphere. Although the microphone array can be setup so as to provide direction measurements in 3D like the omnidirectional camera, and if the geometry of the room and reverberations are taken into account it can be used to provide also the distance to the object, in this thesis all the microphones are coplanar and we shall

[1] In the thesis this will denote the angle of the measured phenomena relative to the heading of the mobile robot or the x-axis of the local coordinate system.

analyze the case when the array provides bearing-only measurements. To summarize, by omnidirectional sensors we consider sensors with a 360° horizontal field-of-view which report directional measurements either in the form of the bearing or a vector on a unit sphere.

Let us continue now by decomposing the title of the thesis. The 'moving object' implies that the phenomena that we will be measuring will not be stationary. Our task will thus be to either detect this motion (in the camera image) or detect some other phenomena that the moving object produces (speaker's voice). Thereafter, these measurement will be used by a tracking method which will estimate the state of the moving object, whose dimension depends on the utilized sensor. Since the sensors we use are omnidirectional, appropriate modeling of such measurements will be required and this will lead us to the field of directional statistics. Furthermore, since the application scenarios involves a mobile robot we can expect that the sensors will be in motion. For the microphone array this will mean that the surroundings will not be stationary—the robot will experience different types of noise as well as ego-motion noise, while for the camera this will present an inconvenience in moving object detection since the movement of the robot will also induce optical flow in the image besides the one induced by the moving objects.

Directional measurements, like the bearing, emerge often in some of the most fundamental problems in mobile robotics, like SLAM and navigation. The origin of bearing-only measurements comes as an inherent property of commonly used sensors like cameras (both perspective and omnidirectional), which are widely utilized due to their low cost, small dimensions, power consumption etc. In navigation bearing values are often encountered in odometry, since essentially heading of the robot is an angular variable. By analyzing the pervasiveness of, not just bearing, but bearing-only scenarios in mobile robotics, we can conclude that the best possible modeling of angular values would surely bring an improvement or simplification to some of the mentioned mobile robotics problems. Probabilistic approaches to problems in mobile robotics have been widely accepted and tested in a variety of problems and scenarios. Working with directional data, especially under uncertainty, imposes a problem on how to represent them in probabilistic frameworks. Angular variables have hitherto been mostly modeled in a probabilistic manner as Gaussian random variables, which does not unfortunately capture well the non-euclidean properties of angular data. The first issue is that angular data is periodic and that, e.g., values like 1° and 361° should be equally probable. Furthermore, the problem becomes more involved if we simply want to calculate the maximum likelihood value of the mean of a set of independent identically distributed angular random variables, since we cannot simply take the mean value as the Gaussian distribution would suggest. In Bayesian inference, often two distributions are multiplied so as to yield a new updated belief in the state. How should the case be handled if we have equally certain but antipodal means like 90° and 270°? Should the product always yield a more certain distribution than any of the initial multiplicands were (like in the case of the Gaussian distribution)? Are two bad angular measurements always better than one bad angular measurement? Some of these problems can be solved by simple methods, like placing a modulo 2π operation on the considered values, but for the most the solution is not that simple and by using a proper directional distribution we can be at ease that some of these question are implicitly considered in the distribution itself. Therefore, directional

(circular) distributions constitute a fundamental part of the contributions of the thesis which also strives to offer answers to the aforementioned questions.

1.2 ORIGINAL CONTRIBUTIONS

The original contributions of the thesis essentially revolve about probabilistic methods in moving object tracking and voice activity detection. Concerning moving object tracking, this pertains to utilization of methods from the field of directional statistics. The contributions and a brief elaboration follow in the sequel.

- Speaker tracking method by an omnidirectional microphone array of a mobile robot based on a mixture of von Mises distributions.

  This contribution encompasses algorithms developed for speaker tracking by a microphone array placed on a mobile robot. Firstly, a convex combination of von Mises distributions is proposed to model the microphone array measurements, which is then employed in a particle filtering scheme to estimate the bearing of the active speaker. This approach is subsequently expanded in the frame of a Bayesian mixture filter based on the von Mises distribution. In the end, the concept is utilized in an active speaker localization approach where by exploiting the information from robot's ego-motion, both the bearing and the range are consequently estimated.

- Robust voice activity detection algorithm based on modeling of the signal envelope, likelihood ratio and supervised learning.

  This contribution comprises the voice activity detection algorithm that discriminates speech from noise by calculating the likelihood ratio of the distributions of the discrete Fourier transform coefficients conditioned on the speech/non-speech hypotheses and a supervised learning approach with carefully selected input variables. Three learning algorithms are compared for the task: support vector machine, Boost and artificial neural networks.

- Moving object detection and tracking method by an omnidirectional camera of a mobile robot based on optical flow estimation and directional statistics.

  This contribution covers the developed method for detecting motion in an omnidirectional image with a moving sensor. The optical flow is estimated and the flow vectors are lifted to a unit sphere where they are classified as either stemming from the static background or dynamic objects. The discrimination is based on determining the rigid transform between the two locations where the images were captured and on projecting hypothetical locations on the sphere where the static features might have been projected onto. Once the motion is detected the moving object is tracked on the sphere via a Bayesian filter based on the von Mises-Fisher distribution.

1.3 OUTLINE OF THE THESIS

The thesis is organized into seven chapters. Each chapter begins with a short abstract which serves to present generally the content of the chapter, results and the insights it offers.

Afterwards the reader is gradually introduced with the problem and with the related work in the field. After the body of the chapter, in the end, a summary is given which restates some the main results of the chapter and its contributions. Hereafter, we present the outline of the thesis with a short summary of the contents.

CHAPTER 2. This chapter presents the general mathematical background for the thesis and sets up the context of the problem. An emphasis is set on probabilistic methods for moving object tracking and especially on distributions stemming from the field of directional statistics. Hence, the von Mises distribution—a distribution on the unit circle—and the von Mises-Fisher distribution—a distribution on the unit sphere—are presented along with some of their characteristics and methods for parameter estimation and simulation. Furthermore, the Kalman and the particle filter are also presented and a framework for a Bayesian mixture filter with arbitrary densities is formulated.

CHAPTER 3. The chapter comprises the speaker localization methods. Firstly, the modeling of the microphone array with a mixture of von Mises distributions is presented and the experimental results of speaker tracking via a particle filter are presented. Subsequently, this approach is extended to tracking with a mixture filter based on the von Mises distribution. In other words, the estimator is based solely on the analytical densities to perform the tracking. In the end, an active speaker localization algorithm is presented which fuses the bearing measurements and the motion of the robot to estimate both the bearing and the range of the speaker.

CHAPTER 4. A novel approach to voice activity detection based on the likelihood ratio of two hypotheses producing the Rayleigh and Rice distributions and supervised learning is presented. Firstly, the Rayleigh and Rice detector is compared to two other statistical model based detectors. Then a feature set consisting of the likelihood ratio and seventy other spectral and temporal features is created and the most utile features are selected via a partial mutual information method. In the end three supervised learning algorithms are tested and their performance is compared for the task of voice activity detection: support vector machine, Boost and artificial neural networks.

CHAPTER 5. In this chapter a method based on processing on the unit sphere for moving object detection, tracking and following with an omnidirectional camera mounted on a mobile robot is presented. A projection model which enables us to lift the omnidirectional image to the unit sphere is coupled with displacement information from motor encoders and used to segment out vectors that do not belong to the static scene around the mobile robot. Once the motion is detected in the image, the center of gravity of the dynamic flow vectors is calculated and probabilistically structured in order to be included in the tracking framework based on the Bayesian estimation on the sphere with the von Mises-Fisher distribution. Given the estimated position a control law based on visual servoing is calculated which in turn makes the robot follow the moving object.

↬ CHAPTER 6. In this chapter Bayesian methods for sensor fusion are presented. The methods are divided in two groups based on the information that each sensor modality reported: a centralized independent likelihood fusion where each sensor only reported its measurement, and hierarchical fusion where each sensor ran its filter and reported its own estimate along with the uncertainty. The explicit expressions for the fusion solution are given in the form of the extended information filter, unscented information filter and the particle filter. Ensuingly, the problem of tracking multiple moving objects with multiple heterogeneous sensors is addressed. The integration of multiple sensors is solved by asynchronously updating the tracking filters as new data arrives and the data association is solved by applying the joint probabilistic data association filter. Experimental results are presented for the case of people tracking with a laser range sensor, microphone array and an RGB-D camera.

↬ CHAPTER 7. This chapter brings conclusions and summary of the scientific contributions. Some ideas for future work are given as well.

2

General background and problem setting

This chapter presents mathematical background and several tools utilized in the thesis. As we shall see an emphasis is set on probabilistic methods for moving object tracking and especially on distributions stemming from the field of directional statistics. We also give a setting to these mathematical tools, namely the omnidirectional sensors that were used in the thesis and the scenario of moving object tracking with such sensors. The mobile robot setup with a microphone array and an omnidirectional camera is presented and described. Furthermore, a distribution on the unit circle, the von Mises distribution, that is proposed to model angular measurements and system state is described. Methods for parameter estimation and distribution simulation are presented. Since the von Mises distribution suffers from numerical problems for the cases when it is very sharp, a numerically stable procedure is also derived and presented. For the case of 3D directional data a parametric distribution on the unit sphere, the von Mises-Fisher distribution, is presented. Just like the circular von Mises, this distribution is proposed to be utilized for directional measurements and system states. Methods for parameter estimation and simulation of the distribution are presented and a numerically stable procedure for distribution evaluation is described. Since the emphasis in this thesis is on probabilistic methods for object tracking, i.e. state estimation methods, a general description of the class of Bayesian estimation methods is presented and discussed. In the end, three explicit forms of the estimators are presented: the Kalman filter, the particle filter and the mixture filter.

2.1 Introduction

A mobile robot, if it is to behave autonomously in a changing highly-dynamic environment, is destined to be equipped with at least some of the sensors shown in Fig. 2.1. Each of the depicted sensors has its advantages and disadvantages, some are active and some are passive. The laser range sensor (LRS) is a very accurate sensor but it operates in a single horizontal plane, while the RGB-D and the stereo cameras offer an image rich with information by way of a point cloud—each operating on a different principle but with a somewhat smaller field-of-view (FOV). The omnidirectional camera and the microphone array, which are sensors of interest in this thesis, just by themselves cannot measure the distance, but they offer the prospects of a very large FOV, specificaly the microphone array if arranged in a 3D fashion can truly measure the phenomena in any given direction, while the omnidirectional camera, depending on the setup, has a 360° FOV in the horizontal plane and often more than 180° in

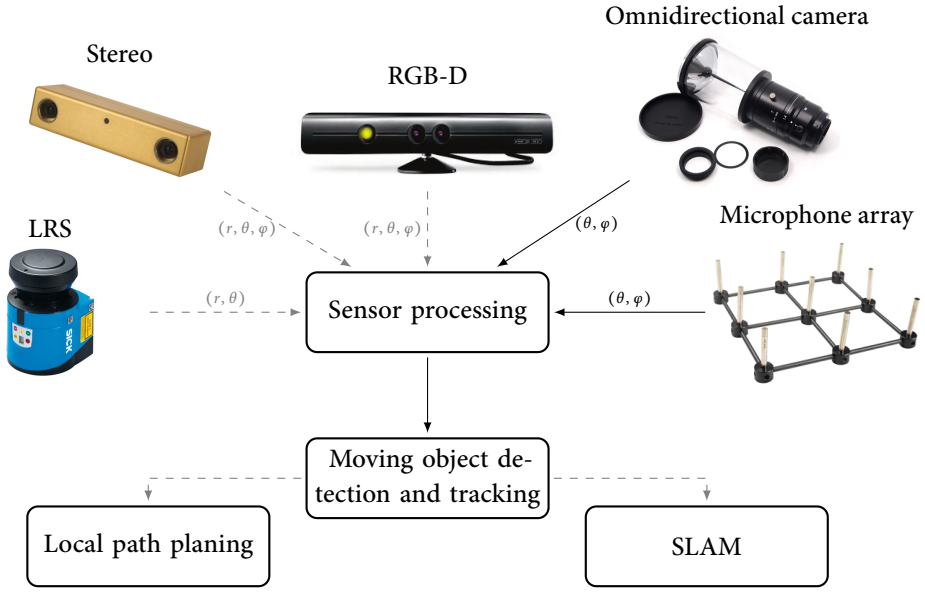


Figure 2.1: Example of different sensors utilized in mobile robotics

the vertical plane. Given that, we can imagine that an autonomous mobile robot, which is to behave robustly in a challenging environment, can only profit by rationally combining various heterogeneous sensors. The raw measurements of each of these sensors is subjected to signal processing which can then be utilized in the task of moving object detection and tracking which can further be exploited by the local path planing and localization and map building algorithms as shown in Fig. 2.1. As stated in Chapter 1 directional measurements, both in 2D (bearing/azimuth) and 3D, are often encountered in some of the fundamental problems in mobile robotics like SLAM, localization, navigation, moving object tracking, and in the sequel we present some of the examples.

An overview and comparison of various bearing-only SLAM algorithms can be found in [1, 2]. In [3] a uniform probability on range was imposed to yield a new probability density function for bearing-only measurements for SLAM (bearing was modeled as a Gaussian variable). Since in the pertinent problem landmarks are always bearing-only measurements, research on proper initialization is still active [4–6]. Furthermore, bearing-only SLAM was applied to acoustic landmarks which were detected with a microphone array [7]. A related problem is mobile robot localization in an already built map from bearings-only measurements. In [8] a Monte-Carlo localization was implemented based on measurements from an omnidirectional camera, in [9] localization was solved by an optimization on a set of linear constraints governed by each bearing measurement, while in [10] a triangulation method was used for localization based on bearing measurements of artificial landmarks. In [11] an optimal landmark placement for range-only, bearing-only and range-and-bearing sensors was analyzed and proposed in regard to achieving a bound on the maximal deviation from the desired trajectory. Moreover, there exists recent works on cooperative localization with bearing-only sensors [12–15]. In navigation bearing values are often encountered in odometry, since essentially heading of the robot is an angular variable. In [16] monocular images were used in a bearing-only navigation scenario to correct robot heading and suppress odometric error. In [17] bearing measurements were used for 2D/3D homing of

ground and aerial vehicles with an omnidirectional camera in a visual servoing framework.

When considering tracking with bearing-only values, the pertinent problem was tackled foremostly in naval warfare. In [18] it was shown that tracking in modified polar coordinates with an extended Kalman filter provided better and more stable results that when tracking in Cartesian coordinates. This model was further developed in [19] where the tracking was performed with a bank of range parametrized extended Kalman filters in modified polar coordinates. Although this problem has been researched for few decades, it still receives attention due to emerging new filtering methods. In [20] three different filters were compared for the task, while in [21] various methods for tracking and decentralized sensor fusion were studied, including bearing-only scenarios.

In mobile robotics this problem has been present, again, due to utilization of sensors like cameras and microphone arrays, which are able only to measure bearing values of the objects. In [22–25] an omnidirectional camera was utilized to track dynamic objects/humans with a mobile robot. If analyzing the field of acoustics, there exists a many papers on the problem of speaker tracking and localization which is usually solved by utilizing a microphone array of four, eight or just two microphones (inspired by biological auditory functions) [26–31]. In [26] the authors used a microphone array for tracking multiple speakers where the state was modeled and estimated with the particle filter. In [32] the authors propose a multiple hypothesis square root unscented Kalman filter for localization of intermittent moving sources, while in [33] the authors further develop general eigenvalue decomposition multiple signal classification (MUSIC) algorithm to meet real-time and high resolution requirements. Microphone arrays have even been used in outdoor environments and mounted on quadcopters to be used for sound source localization [34, 35]. Furthermore, some microphone arrays for robotics applications have reached impressive dimensions [36–38]. For all these approaches, regardless of a microphone array being mounted on a mobile robot, a quad-copter, consisting of two or 64 channels, they all work with angular variables, i.e. bearing, or bearing and elevation.

However, we need to stress out that in some research areas bearing-only tracking assumes that after some iterations the filter will converge to a location in 2D (determined by both bearing and range). This is not a problem if the tracked object is stationary, as in the case of landmarks in SLAM and localization, or if the moving platform can outmaneuver the tracked object. Indeed, optimal maneuvers for bearing-only tracking have been an interesting topic of research [39–41]. But in some aspects of mobile robotics, like human-robot interaction, it might be sufficient to estimate just the bearing of the tracked object. The limitations on the robot speed and maneuverability in closed office-like environments might make it sometimes difficult for the robot to outmaneuver the tracked object.

2.2 UTILIZED SENSORS

One of the main goals of the thesis was to develop tracking methods tailored to omnidirectional sensors. As stated in Chapter 1 by omnidirectional sensors we consider sensors which are able to make measurements in all directions and usually provide measurements in angles. In other words, in 2D this can be translated to making measurements on the unit circle, while in 3D we can consider the measurements as being made on the unit sphere. We

shall see in Section 2.3 how these measurements can be modeled with techniques from the field of directional statistics, but firstly we shall present the sensors utilized in this thesis and we shall refrain from using definitions and the vernacular of the field in order to first introduce the general concept behind utilized sensors.

The sensor that we primarily used and modeled as an omnidirectional sensor in 2D is the microphone array. A microphone array consists of several microphones arranged in a specific geometry, often omnidirectional themselves in the sense that they equally receive signals from all directions. The geometry depends, naturally, on the space constraints of the platform that the sensors are being placed on and, as we shall see in Chapter 3, the error sensitivity of the array depends on this geometry. The microphone signals are acquired by a multichannel synchronized card, which have become light and small in dimensions so that they can be incorporated on mobile robots [42] or even small flying vechicles [34, 35]. An example of a mobile robot equipped with a microphone array is shown in Fig. 2.2. The microphone array can be arranged in a scenario where there are no objects preventing direct path of sound waves from one microphone to the other. In such a set-up, most often the time it takes the signal to reach one microphone from the other is taken as the acoustical cue for calculating the direction from which the signal came. We can imagine that placing an object between the microphones can only perturb such measurements, but on the contrary, if we can model this specific 'perturbation' we can take it to our advantage and incorporate additional cues, like the difference in the level of the signals, into our signal processing framework. Indeed, approaches to determining the direction of the signal with a microphone array or just a pair of microphones are numerous, as we shall in Chapter 3. But the primary contribution of the thesis in the field of speaker localization is not in the development of new signal processing algorithms for determining the location of the speaker, but in probabilistic modeling and tracking with such measurements. In that sense, the methods presented in the thesis are universal and could be utilized with any of the localization approaches which yield directional measurements.

The sensor that we utilized primarily in the context of an omnidirectional sensor with 3D directional measurements is the omnidirectional camera. The omnidirectional camera is a general concept and signifies a camera that provides a 360° view of the surrounding scene in a single image. The realizations of the omnidirectional camera are achieved either by a synchronized combination of multiple cameras facing different directions, or by a combination of a camera and a mirror or a wide-angle (fish-eye) lens. An example of a mobile robot equipped with an omnidirectional camera is shown in Fig 2.3. An important theoretical concept behind image formation in omnidirectional cameras is the so-called unified projection model which enables us to map each pixel in the image to a point on the unit sphere [43, 44]. This model is theoretically valid only for certain camera-mirror combinations, but it has been shown in practice to be valid also for fish-eye lenses [45]. Therefore, by focusing on the processing of the sphere we can obtain general approaches which can be applied to any omnidirectional image. In that spirit, so has the work in the thesis focused on developing algorithms for detection and tracking of moving objects with directional measurements on the unit sphere. In essence, all the measurements and states are considered as unit sphere vectors coupled with corresponding uncertainty, thus creating a framework for probabilistic state estimation on the sphere. The methods presented in

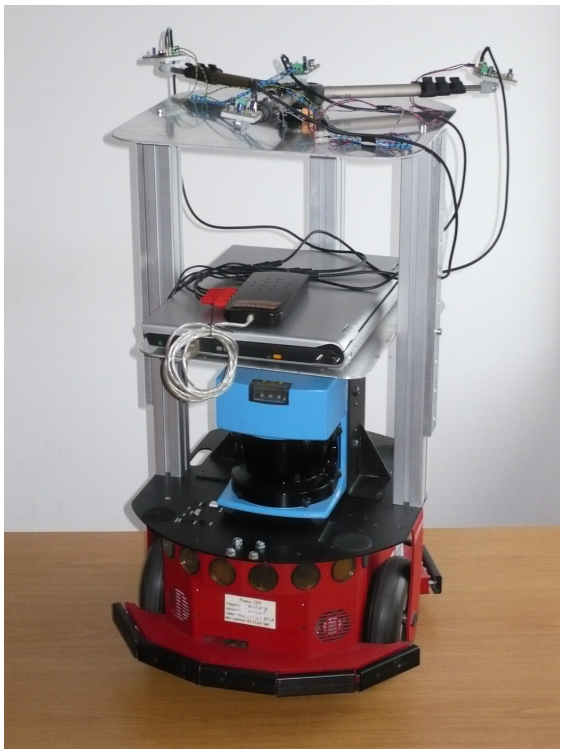


Figure 2.2: Mobile robot equipped with a four channel microphone array

this thesis concerning the tracking on the unit sphere are universal and could be applied to any direction-only state estimation problem, including the previously discussed problem of speaker localization when the microphone array in not planar and measures also the elevation angle.

2.3 DIRECTIONAL STATISTICS AND CIRCULAR DISTRIBUTIONS

Directional statistics studies mainly observations which are unit vectors either in the plane or in a three dimensional space. In the former case usually the sample space will be a circle, while in the latter case the sample space will be a sphere. Given the observation space, special methods are required in order to deal with the intricacies of such data. In this thesis, both the data on the circle and on the sphere will be analyzed and appropriate methods utilized. Majority of the tools and concepts come from monograms [46–49] from which also this section is constructed in order to give a gentle introduction to the subject matter.

Circular data can arise in many situations, commonly coming as measurements from two principal circular instruments, the compass and the clock. In meteorology wind direction provides a natural source of circular data, while the time of the day at which thunderstorms or heavy showers occur can also be treated as such. In biology studying animal navigation usually leads to circular data, e.g. turtle orientation after laying eggs, vanishing angles of birds to name but a few. Regarding the physics, one of the fundamental circular distributions arose when Richard von Mises proposed to test if measured atomic weights were indeed integers subject to error by verifying if the resulting distribution of fractional parts has a mode at $0°$ [50]. The robotics community has recognized the benefits of the von Mises distribution to model directional data. In [51] the von Mises (VM) distribution was used

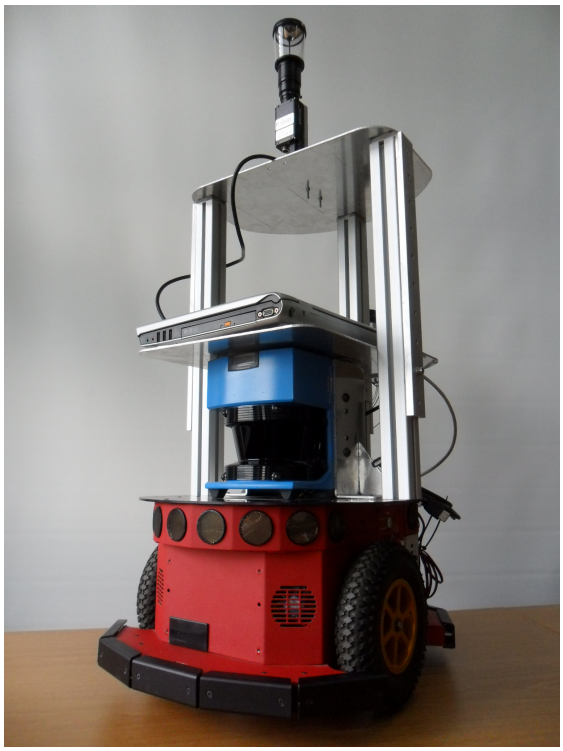


Figure 2.3: Mobile robot equipped with an omnidirectional camera (perspective camera and a hyperbolic mirror)

in odometry to deal with the heading changes for topological model learning. In [52] the authors proposed a solution for solving large-scale partially observable Markov decision processes and tested the algorithm first on a synthetic problem of a circular corridor, where the transition and observation probabilities were modeled with the VM distribution. In [53] the emission distribution of a hidden Markov model was learned by estimating parameters of the VM distribution in order to model compass measurements in a localization problem. In our previous work [54–56] we also utilized the VM distribution in the context of speaker localization and tracking in order to model the state and the microphone array measurements as a VM mixture in the context of Bayesian estimation framework.

Spherical data, since the surface of the earth is approximately a sphere, arise readily in earth sciences, e.g. the location of the earthquake's epicenter, the paleomagnetic directions of the earth's magnetic pole etc. Furthermore, many astronomical observations are points on the celestial sphere and as such yield spherical data. It is interesting to note that Gauss developed theory of errors primarily for the analysis of astronomical measurements. However, since the measurements were concentrated in a small region, it was reasonable to approximate the sphere locally with a tangential plane thus leading to the development of the theory of statistics on the Euclidean spaces rather than on the sphere [46]. One of the fundamental spherical distributions arose in statistical mechanics while studying the moment of weakly interacting dipoles subjected to external electric field [57]. In robotics the von Mises-Fisher (VMF) distribution has also been recognized and utilized. In [58] in the context of collaborative robot localization VMF was used to model the errors in relative orientation measurements. For the purpose of C-space sampling in humanoid robot configurations the authors in [59] used VMF for the orientation variable. In [60] the

distribution was used to introduce perturbations in the 3D rotation for the purpose of a dissimilarity measure between point cloud representations of segmented shapes near object grasping points. In our previous work [61] we used VMF to model the measurements of an omnidirectional camera and to represent the state (direction) of a moving object as a point on the unit sphere. In the sequel we present both the VM and the VMF distribution and show some of their properties.

2.3.1 *Von Mises distribution*

The von Mises distribution, also referred to as normal circular distribution, is a continuous parametric probability distribution defined on the unit circle, or equivalently on interval $[0, 2\pi)$, with probability density function (PDF) given by [50]

$$p(x; \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} \exp\{\kappa \cos(x - \mu)\}, \quad 0 \le x < 2\pi, \tag{2.1}$$

where $\mu \in [0, 2\pi)$ denotes the mean angle, $\kappa \ge 0$ is the concentration parameter, and I_0 is the modified Bessel function of the first kind and of order zero [46]. Recall, the modified Bessel function of the first kind and of order $n \in \mathbb{N}$ is defined by

$$I_n(\kappa) = \frac{1}{2\pi} \int_0^{2\pi} \exp(\kappa \cos \xi) \cos(n\xi)\, \mathrm{d}\xi. \tag{2.2}$$

In many ways the von Mises distribution is considered as the circular analogue of the normal distribution on the real line: it is unimodal, symmetric around mean angle μ, and the concentration parameter κ is analogous to the inverse of the variance. Furthermore, it is characterized by the maximum entropy principle in the sense that it maximizes the Boltzmann-Shannon entropy $-\int_0^{2\pi} f(x) \log f(x) \mathrm{d}x$ under prescribed circular mean (first trigonometric moment) equal to $I_1(\kappa)/I_0(\kappa) e^{i\mu}$ [46]. An illustration of several VM distributions with different concentration parameters can be seen in Fig. 2.4, while an algorithm for simulating the distribution is shown in Algorithm 1 [46]. The latter could be used in the example of adding von Mises noise to angular measurements. Also, note that values such as $x \pm 2k\pi, k \in \mathbb{N}$ all have equal probabilities.

Von Mises distribution, like many well known parametric distributions (Gaussian, Poisson, Gamma, Dirichlet etc.), is an exponential family [49]. A parametric set of probability distributions defined on a sample space $\mathcal{X}$ and parametrized by the natural parameter $\theta \in \Theta$ is called *exponential family* if their probability densities admit the following canonical representation

$$p(x; \theta) = \exp(T(x) \cdot \theta - F(\theta) + C(x)), \quad x \in \mathcal{X}. \tag{2.3}$$

Map $T : \mathcal{X} \to \mathbb{R}^d$ is called the minimal sufficient statistics, and functions F and C denote the log-normalizer (log-partition) and the carrier measure, respectively.

It can be readily checked from (2.1) that our study example, the von Mises distribution, $p(x; \mu, \kappa)$ with standard parameters μ and κ, is an exponential family parametrized by the natural parameter $\theta = (\kappa \cos \mu, \kappa \sin \mu) \in \Theta = \mathbb{R}^2$. The minimal sufficient statistics is the standard parametrization of the unit circle $T(x) = (\cos x, \sin x)$, the log-normalizer is given by $F(\theta) = \log(2\pi I_0(\sqrt{\theta_1^2 + \theta_2^2}))$, and the carrier measure is trivial, $C = 0$. Canonical parametrizations (2.3) for other exponential families can be found in [62].

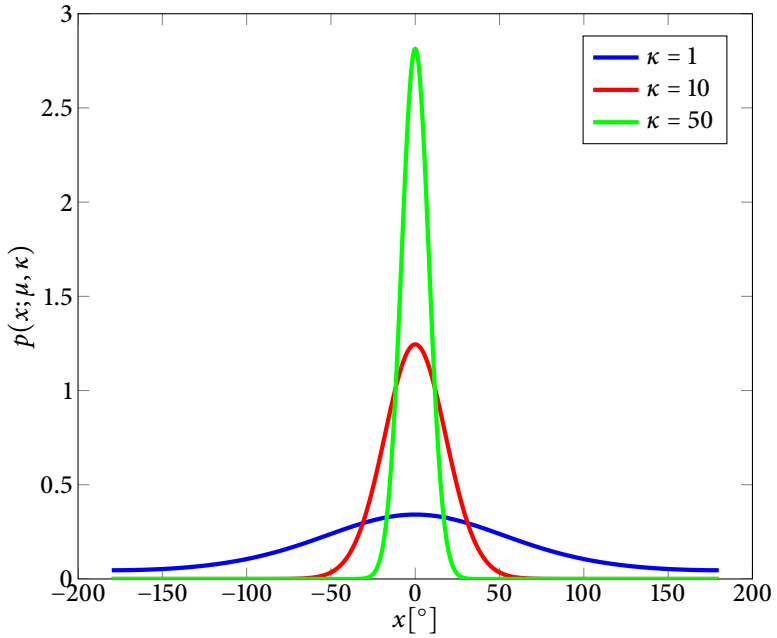


Figure 2.4: The von Mises distribution with $\mu = 0$ and for various concentration parameters

Algorithm 1: Simulation of the von Mises distribution

Require: Distribution parameters μ, κ

Ensure: A sample θ from the distribtion

1: $a \leftarrow 1 + \sqrt{1 + 4\kappa^2}, b \leftarrow \dfrac{a - \sqrt{2a}}{2\kappa}, r \leftarrow \dfrac{1 + b^2}{2b}$
2: *# Sample from the uniform distribution*
   $U_1 \leftarrow \mathcal{U}_{[0,1]}, U_2 \leftarrow \mathcal{U}_{[0,1]}$
3: $z \leftarrow \cos(\pi U_1), f \leftarrow (1 + rz)/(r + z), c \leftarrow \kappa(r - f)$
4: **if** $c(2 - c) - U_2 > 0$ **then**
5: go to (11)
6: **end if**
7: Sample $U_2 \leftarrow \mathcal{U}_{[0,1]}$
8: **if** $\log(c/U_2) + 1 - c < 0$ **then**
9: go to (2)
10: **end if**
11: sample $U_3 \leftarrow \mathcal{U}_{[0,1]}$
12: $\theta = \mu + \text{sign}(U_3 - 0.5) \arccos(f)$

VON MISES DISTRIBUTION PARAMETER ESTIMATION. When having a number of bearing measurements and if we reason that they are sampled from a unimodal distribution, then we can estimate the μ and κ parameters via maximum likelihood (ML) estimation. Firstly, some auxiliary values are calculated, namely the Cartesian coordinates of the center of mass

$$\bar{C} = \frac{1}{N} \sum_{i=1}^{N} \cos \theta_i, \quad \bar{S} = \frac{1}{N} \sum_{i=1}^{N} \sin \theta_i. \tag{2.4}$$

Furthermore, the mean resultant length is given by

$$\bar{R} = \sqrt{(\bar{C}^2 + \bar{S}^2)}. \tag{2.5}$$

Given the aforementioned variables, when $\bar{R} > 0$ we can calculate the ML estimate of the mean direction as [46]

$$\hat{\mu} = \text{atan2}(\bar{S}, \bar{C}). \tag{2.6}$$

The ML estimation of the concentration parameter is a solution to the following equation [46]

$$A(\hat{\kappa}) = \bar{R}, \tag{2.7}$$

where

$$A(\hat{\kappa}) = \frac{I_1(\hat{\kappa})}{I_0(\hat{\kappa})}, \tag{2.8}$$

is the ratio of the modified Bessel functions of order one and order zero. This equation is solved by numerically inverting $A(\,.\,)$. However, a simpler and computationally less expensive approach can be adopted. It is based on good approximations of the inverse of $A(\,.\,)$ for certain intervals of $\bar{R}$

$$\begin{aligned} \hat{\kappa} &= 2\bar{R} + \bar{R}^3 + \bar{R}^5 5/6, \quad && \bar{R} < 0.53 \\ \hat{\kappa} &= -0.4 + 1.39\bar{R} + \frac{0.43}{1-\bar{R}}, \quad && 0.53 \leq \bar{R} < 0.85 \\ \hat{\kappa} &= \frac{0.5}{1-\bar{R}}, \quad && \bar{R} \geq 0.85. \end{aligned} \tag{2.9}$$

Furthermore, in [63] a good approximation for estimation of the concentration parameter κ has been empirically discovered

$$\hat{\kappa} = \frac{2\bar{R} - \bar{R}^3}{1-\bar{R}^2}. \tag{2.10}$$

CALCULATION OF VON MISES DISTRIBUTIONS WITH LARGE κ. The direct form of the VM distribution suffers from numerical issues when working with large concentration parameter κ. The main problem is that for large κ both the exponent and the modified Bessel function of the first kind quickly reach the maximum value that can be stored in double precision floating point representation.

To solve this problem, we move the normalizer of the VM distribution in the exponent as follows

$$\begin{aligned} p(x;\mu,\kappa) &= \exp\left\{\kappa\cos(x-\mu) - \log(2\pi I_0(\kappa))\right\} \\ &= \exp\left\{\kappa\cos(x-\mu) - \log(2\pi) - \log(I_0(\kappa))\right\}. \end{aligned} \tag{2.11}$$

We can now see that in the exponent we have $\log(I_0(\kappa))$ on whose approximation we shall concentrate. In [64] a solution for computing the logarithm of the modified Bessel function of the first kind was presented in the context of parameter estimation of the von Mises-Fisher distribution. In this section, we utilize this idea and present a solution for computation of the VM with large κ.

The $I_0(x)$ has the following power series expansion [46]

$$I_0(x) = \sum_{k=0}^{\infty} \frac{1}{(k!)^2} \left(\frac{x}{2}\right)^{2k} = \sum_{k=0}^{\infty} f_k(x). \tag{2.12}$$

Since $f_k(x) \to 0$ when $k \to \infty$ we can expect to have a good approximation for large k. In order to evaluate $\log(I_0(x))$ we apply the following property of the exponential function

$$\begin{aligned} \log(I_0(x)) &= \log \sum_{k=0}^{\infty} \exp\left\{2k \log \frac{x}{2} - 2 \log k!\right\} \\ &= \log \sum_{k=0}^{\infty} \exp\left\{2k \log \frac{x}{2} - 2 \sum_{r=0}^{k} \log r\right\} \\ &= \log \sum_{k=0}^{\infty} \exp\left\{t_k(x)\right\}. \end{aligned} \tag{2.13}$$

The term $\log k!$ is still problematic from a numerical standpoint, which is why we used the equality $\log k! = \sum_{r=0}^{k} \log r$ to yield a computable solution. Please note that the aforementioned calculation of $\log k!$ can be calculated off-line for a given k and stored for a simple look-up operation.

At this point we can still expect large values in the exponent in (2.13) and in order to solve this problem we define $m(x) = \max\{t_k(x)\}$ and perform the following algebraic manipulation

$$\log(I_0(x)) = m(x) + \log \sum_{k=0}^{\infty} \exp\left\{t_k(x) - m(x)\right\}, \tag{2.14}$$

where $t_k(x) = 2k \log \frac{x}{2} - 2 \sum_{r=0}^{k} \log r$ and $m(x) = \max\{t_k(x)\}$. The number of the terms in (2.14) required to have an accurate approximation depends on the κ. Comparing this method of VM evaluation to Matlab implementation based on [65], which suffered from numerical problems for large κ, we did not notice any increase in the computational expense.

2.3.2 *Von Mises-Fisher distribution*

The direct generalization to the sphere of the von Mises distribution on the circle is the von Mises-Fisher distribution which serves as an all-purpose probability model for directions in space and directional measurement errors [48]. When considering directions in p dimensions, i.e. unit vectors in p dimensional Euclidean space $\mathbb{R}^p$, one can represent them as points on S^{p-1}, i.e. the $p-1$ dimensional sphere with unit radius and center at the origin. In other words, a p-sphere is defined as a set of points in $(p+1)$ dimensional Euclidean space, hence a 1-sphere is the circle and the 2-sphere is the surface of a ball in three-dimensional space. A three-dimensional unit random vector $\boldsymbol{x}$ is said to have a von Mises-Fisher distribution if its PDF is of the following form

$$p(\boldsymbol{x}; \kappa, \boldsymbol{\mu}) = \frac{\kappa}{4\pi \sinh \kappa} \exp\left(\kappa \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{x}\right), \quad \boldsymbol{x} \in S^2, \tag{2.15}$$

where $\boldsymbol{\mu}$, also a unit vector ($\|\boldsymbol{\mu}\| = 1$), is the mean direction, κ is the concentration parameter and S^2 is the unit 2-sphere. Because (2.15) is symmetrical about $\boldsymbol{\mu}$, the mean direction of $\boldsymbol{x}$ is $\boldsymbol{\mu}$. For $\kappa > 0$, the distribution has a mode at the mean direction $\boldsymbol{\mu}$, whereas when $\kappa = 0$

the distribution is uniform. The larger the κ the greater the clustering around the mean direction. Since (2.15) depends on $\boldsymbol{x}$ solely through $\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}$, the VMF is rotationally symmetric about $\boldsymbol{\mu}$. An example of several VMF distributions with different concentration parameters are depicted in Fig. 2.5.

For large concentration parameters double precision arithmetic can overflow easily. Therefore a numerically more reliable form of the PDF which works over a larger range of concentration parameters can be used [66]

$$p(\boldsymbol{x};\boldsymbol{\mu},\kappa) = \begin{cases} \dfrac{1}{4\pi}, & \kappa = 0 \\ \dfrac{\kappa}{2\pi(1-\exp(-2\kappa))}\exp\left(\kappa(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}-1)\right), & \kappa > 0. \end{cases} \tag{2.16}$$

It can be readily checked that the VMF distribution $p(\boldsymbol{x};\boldsymbol{\mu},\kappa)$ defined by (2.15) with standard parameters $\boldsymbol{\mu}$ and κ, is an exponential family parametrized by the natural parameter $\boldsymbol{\theta} = \kappa\boldsymbol{\mu}, \boldsymbol{\theta} \in \mathbb{R}^3$. The minimal sufficient statistics is $T(x) = \boldsymbol{x}$, the log-normalizer is given by $F(\boldsymbol{\theta}) = \log 4\pi \sinh(\kappa)/\kappa$, and the carrier measure is trivial $C = 0$. An algorithm for simulating VMF based on [66, 67] is shown in Algorithm 2. Note that the result is a random vector from the VMF with the given κ and $\boldsymbol{\mu} = [0,0,1]^{\mathrm{T}}$; in order to get the desired direction we only need to apply the appropriate rotation.

Algorithm 2: Simulation of the von Mises-Fisher distribution

Require: Distribution concentration parameter κ

Ensure: A sample $\boldsymbol{r}$ from the distribution with $\boldsymbol{\mu} = [0,0,1]^{\mathrm{T}}$

1: sample $\xi \sim \mathcal{U}_{[0,1]}$

2: $w \leftarrow 1 + \kappa^{-1}\left(\log\xi + \log\left(1 - \dfrac{\xi-1}{\xi}\exp(-2\kappa)\right)\right)$

3: sample $\theta \sim \mathcal{U}_{[0,2\pi]}$

4: $\boldsymbol{v} \leftarrow [\cos\theta, \sin\theta]^{\mathrm{T}}$

5: $\boldsymbol{r} \leftarrow [\boldsymbol{v}\sqrt{1-w^2}, w]$

In the literature this distribution is also referred to as the Fisher distribution [46], while the von Mises-Fisher distribution is used to denote a family of distributions on the $(p-1)$-sphere. In [48] it is also referred to as the Fisher distribution with a note that it is also called the von Mises-Fisher distribution, while the term Langevin distribution is used to represent a family of distributions on the $(p-1)$-sphere. A historical account on this distribution can be found in [48]. In the thesis, if not otherwise stated, henceforth we use the term von Mises-Fisher distribution to denote the distribution on the 2-sphere in the three-dimensional Euclidean space and the term sphere to denote simply the 2-sphere.

↬ VON MISES-FISHER DISTRIBUTION PARAMETER ESTIMATION. When facing a group of N unit vectors on the sphere, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, and wanting to infer about the VMF distribution, the need for calculating the parameters $\boldsymbol{\mu}$ and κ arises. The ML estimation of the parameters is then as follows [46]. If we denote the sample mean vector as

$$\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i \tag{2.17}$$

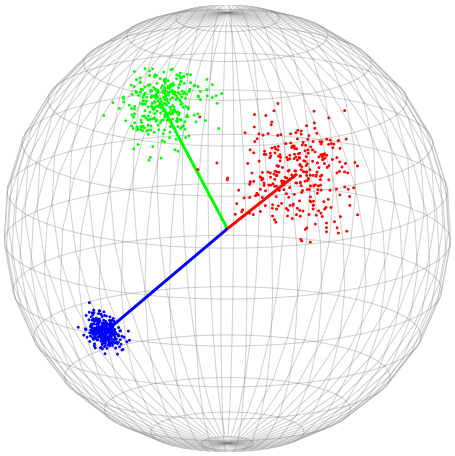


Figure 2.5: Illustration of samples from three von Mises-Fisher distributions: $\kappa = 50$ (red), $\kappa = 100$ (green), $\kappa = 500$ (blue)

and the mean resultant length as

$$\bar{R} = ||\bar{\boldsymbol{x}}|| \tag{2.18}$$

then the mean value and the concentration parameter are calculated via

$$\hat{\boldsymbol{\mu}} = \frac{\bar{\boldsymbol{x}}}{\bar{R}} \tag{2.19}$$

$$A_3(\hat{\kappa}) = \bar{R} = \coth \kappa - \frac{1}{\kappa}. \tag{2.20}$$

Since numerical methods are necessary to solve (2.20) simpler equations have been derived for some limiting cases. When κ is large, for $\bar{R} \geq 0.9$ the following approximation is satisfactory

$$\hat{\kappa} = \frac{1}{1 - \bar{R}}. \tag{2.21}$$

When κ is small, for $\bar{R} < 0.05$ the following approximation is satisfactory

$$\hat{\kappa} = 3\bar{R}\left(1 + \frac{3}{5}\bar{R}^2 + \frac{99}{175}\bar{R}^4\right). \tag{2.22}$$

Furthermore, in [63] a good approximation for estimation of κ has been empirically discovered

$$\hat{\kappa} = \frac{3\bar{R} - \bar{R}^3}{1 - \bar{R}^2}. \tag{2.23}$$

The previously presented methods for parameter estimation have been derived for general $p - 1$ dimensional spheres, while here they are presented only for the special case of $p = 3$. Please confer [46, 63, 68] for a more general approach.

2.4 BAYESIAN FILTERING

The general idea behind the approach to be presented in this section is the assumption that the quantity we are trying to estimate is a random variable. The approach is termed Bayesian since its implementation is explicitly based on the Bayes theorem, which enables us to incorporate some prior knowledge on the value we are trying to estimate. As it shall be

demonstrated, the resulting filtering approach has a recursive form with general applicability. It can be further said that the Bayes filter (estimator) is optimal on average, or with respect to the assumed prior PDF of the value we are striving to estimate [69]. In this section we present the general derivation of the Bayes filter and several of its realizations.

Let us say that x is a quantity we would like to infer from z in a probabilistic manner. In other words we would like to estimate $p(x\,|\,z)$, i.e. the PDF of the quantity x given the observed z (in our case usually the noisy sensor data). The Bayes rule [70] allows us to solve this problem as follows

$$p(x\,|\,z) = \frac{p(z\,|\,x)p(x)}{p(z)}. \tag{2.24}$$

In the sequel we define each term in (2.24) and offer an interpretation in the context of tracking and/or speaker localization. The quantity $p(z\,|\,x)$ is the likelihood function, i.e. in our context the sensor model to which we assign a PDF to model just how noisy our measurements are. This model does not have to necessarily capture just the noise of the sensor, but also the uncertainty (or ignorance) we might have due to the environmental conditions, e.g. in speaker localization problem the corruption due to the reverberation in the room. The quantity $p(x)$ is called the prior distribution, which represents our knowledge we had about x prior to making the observation z. In practice it could be a uniform distribution over the state space—signifying absolute ignorance, or a very wide distribution over the first measurement. For an example, in the context of speaker tracking we could place a uniform distribution over $[0, 2\pi)$, i.e. the von Mises with the concentration parameter $\kappa = 0$. To continue, note that the quantity $p(z)$ does not depend on x and actually serves as a normalizer in (2.24) to ensure that the resulting distribution integrates to unity. Analytically, it can be evaluated via

$$p(z) = \int p(z\,|\,x)p(x)\mathrm{d}x, \tag{2.25}$$

or as in the case of the particle filter taken into account by normalizing the weights of the particles. Finally, the quantity $p(x\,|\,z)$ is the posterior distribution, i.e. the PDF of the value we infer after incorporating the noisy observation z to the prior knowledge about x. An interesting interpretation of the former logic is that it is a quantitative form of weak syllogisms (epagoge) which deal not with absolutes, but with degrees of plausibility [71].

To derive the Bayes filter, we will turn to the quantities of interest as vectors with a temporal dimension, i.e. we will want to infer the state $\boldsymbol{x}_t$ given $\boldsymbol{z}_{1:t}$. In other words, with the Bayes filter we are striving to estimate the density $p(\boldsymbol{x}_t\,|\,\boldsymbol{z}_{1:t})$, i.e. the PDF of the state $\boldsymbol{x}_t$ at time instant t given the history of all the measurements $\boldsymbol{z}_{1:t}$. Firstly, we start by decomposing the measurement vector and applying the Bayes theorem

$$\begin{aligned} p(\boldsymbol{x}_t\,|\,\boldsymbol{z}_{1:t}) &= p(\boldsymbol{x}_t\,|\,\boldsymbol{z}_t, \boldsymbol{z}_{1:t-1}) \\ &= \frac{p(\boldsymbol{z}_t\,|\,\boldsymbol{x}_t, \boldsymbol{z}_{1:t-1})p(\boldsymbol{x}_t\,|\,\boldsymbol{z}_{1:t-1})}{p(\boldsymbol{z}_t\,|\,\boldsymbol{z}_{1:t-1})}. \end{aligned} \tag{2.26}$$

The current measurement $\boldsymbol{z}_t$ given the current state $\boldsymbol{x}_t$ is conditionally independent of all the previous measurements $\boldsymbol{z}_{1:t-1}$ and this results with the following simplification

$$p(\boldsymbol{z}_t\,|\,\boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}) = p(\boldsymbol{z}_t\,|\,\boldsymbol{x}_t). \tag{2.27}$$

At the moment the Bayes filter has the following form

$$p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t) p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1})}{p(\boldsymbol{z}_t \,|\, \boldsymbol{z}_{1:t-1})}. \tag{2.28}$$

The PDF $p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t)$ represents our sensor model (measurement probability) which describes the noisy measurements of the sensor

$$\boldsymbol{z}_t = h(\boldsymbol{x}_t) + \boldsymbol{n}_t, \tag{2.29}$$

where $h(\boldsymbol{x}_t)$ is a nonlinear function in the system state and $\boldsymbol{n}_t$ is measurement noise. The PDF $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1})$ is the prior, i.e. the knowledge about the state at time instant t before taking the measurement $\boldsymbol{z}_t$. It is actually a prediction of the state to time t and is evaluated by expanding $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1})$

$$p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}) = \int p(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1}) p(\boldsymbol{x}_{t-1} \,|\, \boldsymbol{z}_{1:t-1}) \mathrm{d}\boldsymbol{x}_{t-1}, \tag{2.30}$$

where $p(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1})$ is the probabilistic model of the state evolution and $p(\boldsymbol{x}_{t-1} \,|\, \boldsymbol{z}_{1:t-1})$ is the posterior from the time instant $t-1$. The model $p(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1})$ is defined by the state transition equation and known statistics of the process noise

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) + \boldsymbol{v}_t, \tag{2.31}$$

where $f(\boldsymbol{x}_t)$ is a nonlinear function in the system state and control actions and $\boldsymbol{v}_t$ is process noise. At this point we exploit the assumption that the state $\boldsymbol{x}_{t-1}$ is complete, i.e. that no variables prior to $\boldsymbol{x}_{t-1}$ may influence the stochastic evolution of future states, as all the information is already contained in $\boldsymbol{x}_{t-1}$. Expression (2.30) is an immediate consequence of the total probability theorem and is often known under the name Chapman-Kolmogorov equation. The normalizer $p(\boldsymbol{z}_t \,|\, \boldsymbol{z}_{1:t-1})$ can be analytically evaluated by expansion

$$p(\boldsymbol{z}_t \,|\, \boldsymbol{z}_{1:t-1}) = \int p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t) p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}) \mathrm{d}\boldsymbol{x}_t. \tag{2.32}$$

To conclude, the Bayes filter is a recursive algorithm that iterates between prediction (2.30) and update (2.28) steps.

2.4.1 *Kalman filter*

When the underlying distribution $\boldsymbol{x}_t$ of the state is Gaussian, and the transition and measurement equations are linear, then the Bayesian filter takes the form of the Kalman filter (KF). A complete derivation of the KF via (2.30) and (2.28) can be found in [72]. In this section we shall only present a brief treatment of the subject and final expressions.

Let us assume that the transition equation has the following form

$$\boldsymbol{x}_t = \boldsymbol{A}_t \boldsymbol{x}_{t-1} + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{G}\boldsymbol{v}_t, \tag{2.33}$$

where $\boldsymbol{x}_t \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}_t \in \mathbb{R}^c$ is the known control input, $\boldsymbol{v}_t \in \mathbb{R}^q$ represents a random variable describing the uncertainty in the evolution of the state, $\boldsymbol{A}_t$ is the $n \times n$ state matrix, $\boldsymbol{B}$ is the $n \times c$ input matrix, and $\boldsymbol{G}$ is the $n \times q$ noise matrix. If our estimation of

the state at time $t-1$ is a Gaussian distribution $\mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{t-1}, \boldsymbol{P}_{t-1})$, and (2.33) describes the transition equation, then solution to (2.30) is a Gaussian distribution with the following parameters

$$\begin{aligned} \boldsymbol{\mu}_{t|t-1} &= \boldsymbol{A}_t \boldsymbol{\mu}_{t-1} + \boldsymbol{B}_t \boldsymbol{u}_t \\ \boldsymbol{P}_{t|t-1} &= \boldsymbol{A}_t \boldsymbol{P}_{t-1} \boldsymbol{A}_t^{\mathrm{T}} + \boldsymbol{G} \boldsymbol{Q}_t \boldsymbol{G}^{\mathrm{T}}, \end{aligned} \tag{2.34}$$

where $\boldsymbol{Q}_t$ is the $q \times q$ process noise covariance matrix.

So far we have presented the equations for the prediction part. If we assume that the sensor model $p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t)$ is defined by

$$\boldsymbol{z}_t = \boldsymbol{H}_t \boldsymbol{x}_t + \boldsymbol{n}_t, \tag{2.35}$$

where $\boldsymbol{z}_t \in \mathbb{R}^m$ is the observation vector, $\boldsymbol{n}_t \in \mathbb{R}^m$ is the random variable describing uncertainty in the observation, and $\boldsymbol{H}_t$ is the $m \times n$ measurement matrix, then the solution to (2.28) takes the following form

$$\begin{aligned} \boldsymbol{K}_t &= \boldsymbol{P}_{t|t-1} \boldsymbol{H}_t^{\mathrm{T}} (\boldsymbol{H}_t \boldsymbol{P}_{t|t-1} \boldsymbol{H}_t^{\mathrm{T}} + \boldsymbol{R}_t)^{-1} \\ \boldsymbol{\mu}_{t|t} &= \boldsymbol{\mu}_{t|t-1} + \boldsymbol{K}_t (\boldsymbol{z}_t - \boldsymbol{H}_t \boldsymbol{x}_{t|t-1}) \\ \boldsymbol{P}_{t|t} &= (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t) \boldsymbol{P}_{t|t-1}, \end{aligned} \tag{2.36}$$

where $\boldsymbol{R}_t$ is the $m \times m$ measurement noise covariance matrix. Thus, our state estimation at time t is again a Gaussian distribution $\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|t}, \boldsymbol{P}_{t|t})$. It is important to mention that the random variables $\boldsymbol{v}_t$ and $\boldsymbol{n}_t$ describing process and observation noise are all Gaussian, zero-mean, white, and themselves uncorrelated. If some of the assumptions about the uncorrelatedness are violated, this can still be accounted for in the KF algorithm by, for example, whitening the observations, but such situations are not analyzed in the thesis. If the transition and/or measurement equations are not linear, then we need to relax the linearity assumption and the solution is to utilize the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) [72]. However, if the underlying distribution it not Gaussian and possibly multimodal then different approaches need to be utilized.

2.4.2 *Particle filter*

Particle filtering is a versatile method for recursive Bayesian state estimation. It can handle nonlinear dynamics and measurements models, as well as non-Gaussian noises. The posterior PDF of the state at any time t conditioned on the sequence of observed measurements up to t is estimated by means of a point-mass probability distribution with stochastic support, i.e. *weighted particle set*. Let $\{\boldsymbol{x}^p, w^p\}_{p=1}^P$ denote the random measure that characterizes the posterior state PDF $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t})$, where each particle in the set $\{\boldsymbol{x}^p\}_{p=1}^P$ is associated to the respective weight in $\{w^p\}_{p=1}^P$. The weights satisfy $\sum_p w^p = 1$, so that $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t})$ can be approximated as [73, 74]

$$p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}) \approx \sum_{p=1}^{P} w_t^p \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^p), \tag{2.37}$$

with $\delta(.)$ the Dirac delta measure. In other words, sampling from $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t})$ returns to sampling a particle with a probability equal to its associated weight. As the number of

samples becomes very large, this characterization becomes an equivalent representation to the usual function description of the posterior PDF, and the particle filter approaches the optimal Bayesian estimate. How to reach (2.37) via a Bayesian procedure is discussed in the sequel.

The expectation of some function $f(\boldsymbol{x})$ integrable with respect to the PDF $p(\boldsymbol{x})$ is

$$E\left[f(\boldsymbol{x})\right] = \int f(\boldsymbol{x})p(\boldsymbol{x})\,d\boldsymbol{x}, \tag{2.38}$$

and the approximation of the integral with particles is

$$E\left[f(\boldsymbol{x})\right] \approx \frac{1}{P}\sum_{p=1}^{P} f(\boldsymbol{x}^p), \tag{2.39}$$

where $\boldsymbol{x}^p \sim p(\boldsymbol{x})$ and the expectation converges to the true values as $P \to \infty$. Often, it is hard to sample from the true distribution, hence importance sampling is used. The main idea is to sample from the importance density $q(\boldsymbol{x})$ which encompasses the support space of $p(\boldsymbol{x})$, and then we can rewrite (2.38) as

$$E\left[f(\boldsymbol{x})\right] = \int f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,d\boldsymbol{x} = \int f(\boldsymbol{x})w(\boldsymbol{x})q(\boldsymbol{x})\,d\boldsymbol{x}, \tag{2.40}$$

where the importance weights $w(\boldsymbol{x})$ is given as $w(\boldsymbol{x}) = p(\boldsymbol{x})/q(\boldsymbol{x})$. An estimate of the expectation is then given by

$$E\left[f(\boldsymbol{x})\right] \approx \frac{1}{P}\sum_{p=1}^{P} f(\boldsymbol{x}^p)w(\boldsymbol{x}^p). \tag{2.41}$$

To summarize, the particles are drawn according to the importance density, then weighted so that the consequent random measure constitutes a sound approximation to the posterior PDF.

In particle filtering often the importance density is the one that matches the prior dynamics $p(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1})$, i.e. each particle $\boldsymbol{x}_t^p$ at time t is drawn from its predecessor $\boldsymbol{x}_{t-1}^p$ at time $t-1$ according to the proposal density $\boldsymbol{x}_t^p \sim p(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1}^p)$. Then, the weights of the filter are updated by evaluating the likelihood $p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t^p)$ prior to setting

$$w_t^p \propto w_{t-1}^p p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t^p), \tag{2.42}$$

where $p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t)$ represents the sensor model. In the end, all the particle weights are normalized so that they sum up to unity. The former procedure corresponds to the Bayes update (2.28) under the assumptions that the importance density is chosen to be the prior density [73]. It is also possible to perform particle filter size adaptation through the KLD-sampling procedure proposed in [75].

Since, for any recursive particle filter, the significant weights tend to concentrate on a limited set of particles after few iterations, a resampling step is inserted, which consists in turning $\{\boldsymbol{x}_t^p, w_t^p\}_{p=1}^P$ into the equivalent evenly weighted set $\{\boldsymbol{x}_t^{*p}, \frac{1}{P}\}_{p=1}^P$ by independently sampling (with replacement) $\boldsymbol{x}_t^{*p}$ according to $P(\boldsymbol{x}_t^{*p} = \boldsymbol{x}_t^p) = w_t^p$. The resampling step can be applied at each iteration or only when the number of effective weights $P_{\text{eff}} = 1/\sum_p (w^p)^2$

is less than a given threshold, e.g. 33% of the total number of particles P. One problem with the particle filtering is sample impoverishment which means that after resampling, in the worst case if there is no process noise, the filter might degenerate to having P copies of a single sample. Some of the approaches to solving this problem are the so-called auxiliary and regularized particle filter [73, 74]. Furthermore, particle filters are still affected by the problem of inadequate state-space exploration, especially if the prior distribution has little overlap with the likelihood function. One solution is to fit kernel or mixture models to the particle set, which being a form of regularization, also address the firstly described problem of sample impoverishment [21, 74].

2.4.3 *Mixture filter*

The idea behind mixture filtering is as follows. Not much unlike the particle filter where the state is represented by a set of particles with corresponding weights, in mixture filter the state is represented by a set of PDFs, or hypotheses, defined by their parameters and corresponding weights. Note that all the components are allowed to have different parameters, i.e. in the case of the VM distribution the means, concentration parameters and weights. More formally, the idea is to represent the state as a sum of N density functions

$$p(\boldsymbol{x}) = \sum_{i=1}^{N} w_i p_i(\boldsymbol{x}) \tag{2.43}$$

where the component weights w_i sum up to unity.

In order to perform the Bayesian filtering with mixtures, one needs to solve the prediction (2.30) and the update step (2.28) of the filter. Firstly, we shall analyze the prediction step. Let us assume that the posterior state at $t-1$ is represented by $p(\boldsymbol{x}_{t-1} \,|\, \boldsymbol{z}_{1:t-1})$ and that the state transition probability is a single PDF, then the prediction step amounts to the following nonlinear convolution [76]

$$\begin{aligned} p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}) &= \int p(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1}) p(\boldsymbol{x}_{t-1} \,|\, \boldsymbol{z}_{1:t-1}) \mathrm{d}\boldsymbol{x}_{t-1} \\ &= \int p_{\boldsymbol{v}_t}(\boldsymbol{x}_t - f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)) p(\boldsymbol{x}_{t-1} \,|\, \boldsymbol{z}_{1:t-1}) \mathrm{d}\boldsymbol{x}_{t-1}, \end{aligned} \tag{2.44}$$

where $p_{\boldsymbol{v}_t}(x)$ represent a density whose uncertainty parameter is defined by the noise statistics $\boldsymbol{v}_t$. If the state is represented by a density mixture then the prediction step becomes

$$\begin{aligned} p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}) &= \int p_{\boldsymbol{v}_t}(\boldsymbol{x}_t - f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)) \sum_{i=1}^{N} w_i p_{\boldsymbol{\sigma}_{i,t-1}}(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{i,t-1}) \mathrm{d}\boldsymbol{x}_{t-1} \\ &= \int \sum_{i=1}^{N} w_i p_{\boldsymbol{v}_t}(\boldsymbol{x}_t - f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)) p_{\boldsymbol{\sigma}_{i,t-1}}(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_{i,t-1}) \mathrm{d}\boldsymbol{x}_{t-1}. \end{aligned} \tag{2.45}$$

We can see from (2.45) that in order to calculate the prediction we need to convolve each component from the posterior at $t-1$ with the state transition probability. In this thesis we will focus on a simpler scenario where the motion model will be represented just by independent identically distributed additive noise. As we shall see in Section 3.5 for the von Mises distribution this can be evaluated analytically, while for the Gaussian mixture case the reader is directed to [21]. More elaborate methods for calculating the prediction lie in

solving the Fokker-Planck equation and for possible solutions the reader is referred to [77, 78].

Concerning the update step, let us assume that the likelihood $p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t)$, i.e. the sensor model, is also modeled as a density mixture. Since from the prediction step (2.45) our new prior is also a mixture, the update via Bayes rule amounts to the following

$$\begin{aligned} p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}) &\propto p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t) p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}) \\ &\propto \sum_{i=1}^{M} \gamma_i p_{\boldsymbol{n}_{i,t}}(\boldsymbol{z}_{i,t} - \boldsymbol{x}_t) \sum_{i=1}^{N} w_i p_{\boldsymbol{\sigma}_{i,t|t-1}}(\boldsymbol{x}_t - \boldsymbol{\mu}_{i,t|t-1}), \end{aligned} \tag{2.46}$$

where the normalizer $p(\boldsymbol{z}_t \,|\, \boldsymbol{z}_{1:t-1})$, omitted for clarity, is calculated via (2.32). By inspecting (2.28) we can see that the result of the update will be a component-wise multiplication of density mixtures, which in an ideal scenario results in a mixture consisting of the same densities numbering NM components. For von Mises mixtures a multiplication of two distributions will yield an unnormalized distribution, but due to the normalizer $p(\boldsymbol{z}_t \,|\, \boldsymbol{z}_{1:t-1})$ the overall result will be correctly normalized.

If we took the new posterior from time t with NM components and ran again the predict-update procedure, we would obtain a mixture with NM^2 components at time $t+1$. By continuing this trend we would be faced with a geometric increase in the number of components. Therefore, mixture filtering schemes usually employ a mixture reduction procedure after each update step. In our case, we can see that after the first update step we have ended up with a mixture consisting of NM components, and hence what we would like to achieve with the reduction is to reduce this number to just N components and loose as little information as possible.

Existing literature on mixture reduction schemes is mostly related to Gaussian mixture models. A reduction scheme for Gaussian mixtures in the context of Bayesian tracking systems in a cluttered environment, which successively merges the closest pair of components was proposed in [79]. A simplification of this algorithm can be done by first sorting the components according to their weights and then calculating the distance between the component with the smallest weight and all other components of the mixture. Once the components with the smallest distance are merged, the new component is inserted according to its resulting weight. The process is repeated until the required number of components is reached. The idea behind is that in each step we merge the component which brings the least information to the mixture. This approach, known as the West's algorithm, is one of the computationally most efficient and it was proposed in [80] for component number reduction of mixtures of Gaussian distributions. The main drawback of these schemes are their local character, which gives no information about the global deviation of the reduced mixture from the original one. In [81] the mixture reduction was formulated as an optimization problem for the integral square difference cost function. A better suited distance measure between probability distributions is the Kullback-Leibler (KL) distance [82], but it lacks a closed form formula between mixtures, what makes it computationally inconvenient. Several concepts have been employed to circumvent this problem. A new distance measure between mixture distributions, based on the KL distance, which can be expressed analytically was derived in [83], and utilized to solve the mixture reduction problem. In [84] an upper bound for the KL distance was obtained and used as dissimilarity measure in

a successive pairwise reduction of Gaussian mixtures, which gives a control of the global deviation of the reduced mixture from the original one. Introducing the notion of Bregman information, the authors in [85] generalized the previously developed Gaussian mixture reduction concepts to arbitrary exponential family mixtures. Further development of these techniques for exponential family mixtures can be found in [86–89].

2.5 SUMMARY

In this chapter we have discussed the pervasiveness of directional measurements in mobile robotics. We have seen that these measurements stem from the nature of the measurements of various utilized sensors. We have described the sensors that are utilized throughout the thesis—the microphone array and the omnidirectional camera—and how directional statistics and pertaining distributions relate to these sensors. Thereafter, we have presented the distribution on the unit circle and the unit sphere, namely the von Mises and the von Mises-Fisher distribution. As it was discussed, the von Mises distribution will be used for modeling the measurements of the microphone array, while the von Mises-Fisher distribution will serve as a measurement model in omnidirectional images. Furthermore, methods for parameter estimation, stable evaluation, simulation and illustrations of the two distributions were also presented. Subsequently, Bayesian estimation was introduced in a general manner by deriving the recursive equations of the estimator which results in successive application of the prediction and update steps. In the end, three methods for Bayesian state estimation were presented: (i) the Kalman filter, optimal estimator suitable for linear and Gaussian systems and measurements, (ii) the particle filter, which offers more versatility in the shape, modality and type of the state distribution and measurement, and (iii) the mixture filter, which is interesting since in place of particles it offers the opportunity to include probability density functions.

3

Speaker localization and tracking

This chapter deals with the problem of localizing and tracking a moving speaker over the full range around the mobile robot. The problem is initially solved by taking advantage of the phase shift between signals received at spatially separated microphones. The proposed algorithm is based on estimating the time difference of arrival by maximizing the weighted cross-correlation function in order to determine the bearing of the detected speaker (angle between robots heading and the speaker in question). The cross-correlation is enhanced with an adaptive signal-to-noise estimation algorithm to make the bearing estimation more robust in noisy surroundings. A post processing technique is proposed in which each of these microphone-pair determined bearings are further combined into a mixture of von Mises distributions, thus producing a practical probabilistic representation of the microphone array measurement. It is shown that this distribution is inherently multimodal and that the system at hand is non-linear. Therefore, particle filtering is applied for tracking task. Furthermore, two most common microphone array geometries are analyzed and exhaustive experiments were conducted in order to qualitatively and quantitatively test the algorithm and compare the two geometries. Also, a voice activity detection algorithm based on the aforementioned signal-to-noise estimator was implemented and incorporated into the existing speaker localization system. Moreover, another approach to tracking when dealing with multimodal distributions is presented; namely, tracking with a mixture of von Mises distributions. The algorithm offers the prospects of covering the whole state space with an analytical density mixture. The experimental results are compared to the implementation based on the particle filter. At the end of the chapter a maximum likelihood method for estimating the bearing from interaural phase and level difference between two microphones mounted on a spherical head is utilized in an active speaker localization scenario where both the range and the bearing are estimated.

3.1 Introduction

In biological lifeforms hearing, as one of the traditional five senses, elegantly supplement other senses as being omnidirectional, not limited by physical obstacles, and absence of light. Inspired by these unique properties, researchers strive towards endowing mobile robots with auditory systems to further enhance human–robot interaction, not only by means of communication but also, just as humans do, to make intelligent analysis of the surrounding environment. By providing speaker location to other mobile robot systems,

like path planning, speech and speaker recognition, such a system would be a step forward in developing fully functional human–aware mobile robots.

The auditory system must provide robust and non–ambiguous estimate of the speaker location, and must be updated frequently in order to be useful in practical tracking applications. Furthermore, the estimator must be computationally non-demanding and possess a short processing latency to make it practical for real-time systems. The aforementioned requirements and the fact of an auditory system being placed on a mobile platform, thus having to respond to constantly changing acoustic conditions, make speaker localization and tracking a formidable problem.

Existing speaker localization strategies can be roughly categorized in four general groups. The first group of algorithms refers to beamforming methods in which the array is steered to various locations of interest and searches for the peak in the output power [26, 27, 90–93]. The second group includes methods based upon analysis of spatiospectral correlation matrix derived from the signals received at the microphones [33, 34, 38, 94–97]. The third group relies on the inspiration from physiologically known parts of the hearing system, e.g. binaural cue processing [28, 98–103]. The fourth group of localization strategies is based on estimating the time difference of arrival (TDOA) of the speech signals relative to pairs of spatially separated microphones and then using that information to infer about the speaker location. Estimation of the TDOA and speaker localization from TDOA can be considered as two separate problems. The former is usually calculated by maximizing the weighted cross-correlation function [104], while the latter is commonly known as multilateration, i.e. hyperbolic positioning, which is a problem of calculating the source location by finding the intersection of at least two hyperbolae [105–108]. In mobile robotics, due to small microphone array dimensions, usually hyperbolae intersection is not calculated, only the angle (bearing and/or elevation) is estimated [29, 109–112]. However, in some approaches the range is estimated besides the bearing by exploiting the known movement of the robot [32, 101, 113, 114] or by exploiting sound reflections via a known geometry of the room [115–119]. In [120, 121] the von Mises distribution is used to model error between the predicted and observed interaural phase difference of a binaural system, but was later substituted with a Gaussian distribution since the deviations were much smaller than 2π. Even though the TDOA estimation based methods are outperformed to a certain degree by several more elaborate methods [30, 122, 123], they still prove to be extremely effective due to their elegance and low computational costs.

In this chapter we first propose a new speaker localization and tracking method based on TDOA estimation, probabilistic measurement modeling based on von Mises distribution, and particle filtering. Speaker localization and tracking based on particle filtering was also used in [26, 124–126], but the novelty of this approach is the proposed measurement model used for *a posteriori* inference about the speaker location. The benefits of the proposed approach are that it solves the front-back ambiguity, increases the robustness by using all the available measurements, and localizes and tracks a speaker over the full range around the mobile robot, while keeping low computational complexity of TDOA estimation based algorithms. Subsequently, we extend this approach by replacing the particle filter with a mixture filter based solely on the von Mises distribution. The benefits of this approach are that the state is represented in an analytical instead of sample-based manner, thus

covering the whole state space, and that relatively few parameters are required to describe the posterior distribution. In the end, an active speaker localization strategy based on ML estimation of the bearing with a binaural spherical head and von Mises mixture fitting is presented. The approach enables us to estimate both the bearing and the range of the speaker by fusing the information from the robot movement and bearing measurements. Some of the results and novelties from this chapter have been proposed in [54, 55, 109] which concern the analysis and comparison of different microphone array configurations and the application of the von Mises distribution to microphone array measurement modeling and particle filtering, in [56, 114] which regard the von Mises mixture tracking and active localization, and in [127] where exponential family mixture reduction with emphasis on the VM distribution is extensively analyzed under the notion of Rényi α-divergence and composite metric distances.

3.2 TIME DIFFERENCE OF ARRIVAL ESTIMATION

The main idea behind TDOA-based locators is a two step one. Firstly, TDOA estimation of the speech signals relative to pairs of spatially separated microphones is performed. Secondly, this data is used to infer about speaker location. The TDOA estimation algorithm for two microphones is described first.

3.2.1 *Principle of TDOA*

A windowed frame of L samples is considered. In order to determine the delay $\Delta\tau_{ij}$ in the signal captured by two different microphones (i and j), it is necessary to define a coherence measure which will yield an explicit global peak at the correct delay. Cross-correlation is the most common choice, since we have at two spatially separated microphones (in an ideal homogeneous, dispersion-free, far-field and lossless scenario) two identical time-shifted signals. Cross-correlation is defined by the following expression

$$R_{ij}(\Delta\tau) = \sum_{n=0}^{L-1} x_i[n]\, x_j[n - \Delta\tau], \tag{3.1}$$

where x_i and x_j are the signals received by microphone i and j, respectively. As stated earlier, R_{ij} is maximal when correlation lag in samples, $\Delta\tau$, is equal to the delay between the two received signals.

The most appealing property of the cross-correlation is the ability to perform calculation in the frequency domain, thus significantly lowering the computational intensity of the algorithm. Since we are dealing with finite signal frames, we can only estimate the cross-correlation

$$\hat{R}_{ij}(\Delta\tau) = \sum_{k=0}^{L-1} X_{i,k}\, X_{j,k}^{*}\, e^{j2\pi \frac{k\Delta\tau}{L}}, \tag{3.2}$$

where $X_{i,k}$ and $X_{j,k}$ are the discrete Fourier transforms (DFTs) of $x_i[n]$ and $x_j[n]$, $(.)^*$ denotes complex-conjugate, and k represents the frequency bin. We are windowing the frames with rectangular window and no overlap. Therefore, before applying Fourier transform to signals x_i and x_j, it is necessary to zero-pad them with at least L zeros, since we want to

calculate linear, and not circular convolution. A major limitation of the cross-correlation given by (3.2) is that the correlation between adjacent samples is high, which has an effect of wide cross-correlation peaks. Therefore, appropriate weighting should be used.

3.2.2 *Spectral weighting*

The problem of wide peaks in unweighted, i.e. generalized cross-correlation (GCC), can be solved by whitening the spectrum of signals prior to computing the cross-correlation. The most common weighting function is the phase transform (PHAT) [104]. What PHAT function ($\psi_{\text{PHAT}} = 1/|X_{i,k}||X^*_{j,k}|$) does, is that it whitens the cross-spectrum of signals x_i and x_j, thus giving a sharpened peak at the true delay. In the frequency domain, GCC-PHAT is computed as

$$\hat{R}^{\text{PHAT}}_{ij}(\Delta\tau) = \sum_{k=0}^{L-1} \frac{X_{i,k}X^*_{j,k}}{|X_{i,k}||X_{j,k}|} e^{j2\pi\frac{k\Delta\tau}{L}}. \tag{3.3}$$

The main drawback of the GCC with PHAT weighting is that it equally weights all frequency bins regardless of the signal-to-noise ratio (SNR), thus making the system less robust to noise, and this is especially noticeable when the sound source is narrowband. To overcome this issue, as proposed in [26], a modified weighting function based on SNR is incorporated into GCC framework.

Firstly, a gain function for such modification is introduced (this is simply the Wiener gain)

$$G_{i,k}(l) = \frac{\xi_{i,k}(l)}{1+\xi_{i,k}(l)}, \tag{3.4}$$

where $\xi_{i,k}(l)$ is the *a priori* SNR at the i-th microphone, at time frame l, for frequency bin k and $\xi^0_i = \xi_{min}$. The a priori SNR is defined as $\xi_{i,k}(l) = \lambda^s_{i,k}(l)/\lambda^n_{i,k}(l)$, where $\lambda^s_{i,k}(l)$ and $\lambda^n_{i,k}(l)$ are the speech and noise variance, respectively. It is calculated by using the decision directed (DD) approach proposed in [128]

$$\xi_{i,k}(l) = \alpha_e[G_{i,k}(l-1)]^2\gamma_{i,k}(l-1) + (1-\alpha_e)\max\{\gamma_{i,k}(l)-1, 0\}, \tag{3.5}$$

where α_e is the adaptation rate, $\gamma_{i,k}(l) = |X_{i,k}(l)|^2/\lambda^n_{i,k}(l)$ is the a posteriori SNR, and $\lambda_{i,k}(0) = |X_{i,k}(0)|^2$.

In stationary noise environments, the noise variance of each frequency bin is time invariant, i.e. $\lambda^n_{i,k}(l) = \lambda_{i,k}$ for all l. But if the microphone array is placed on a mobile robot, most surely due to robot's changing location, we will have to deal with non-stationary noise environments. An algorithm used to estimate $\lambda_{i,k}(l)$ is based on minima-controlled recursive averaging (MCRA) developed in [129, 130]. The noise spectrum is estimated by averaging past spectral power values, using a smoothing parameter that is adjusted by the speech presence probability. Speech absence in a given frame of a frequency bin is determined by the ratio between the local energy of the noisy signal and its minimum within a specified time window. The smaller the ratio in a given spectrum, more probable the absence of speech is. Further improvement can be made in (3.4) by using a different spectral gain function [131]. A more detailed description of the DD and MCRA algorithms is deferred to Chapter 4.

To make the TDOA estimation more robust to reverberation, it is possible to modify the noise estimate $\lambda_{i,k}(l)$ to include a reverberation term $\lambda^{\text{rev}}_{i,k}(l)$

$$\lambda_{i,k}(l) \leftarrow \lambda_{i,k}(l) + \lambda^{\text{rev}}_{i,k}(l), \tag{3.6}$$

where $\lambda^{\text{rev}}_{i,k}(l)$ is defined using reverberation model with exponential decay [26]

$$\lambda^{\text{rev}}_{i,k}(l) = \alpha^{\text{rev}}\lambda^{\text{rev}}_{i,k}(l-1) + (1-\alpha^{\text{rev}})\delta|G_{i,k}(l-1)X_{i,k}(l-1)|^2, \tag{3.7}$$

where α^{rev} is the reverberation decay, δ is the level of reverberation and $\lambda^{\text{rev}}_{i,k}(0) = 0$. Equation (3.7) can be seen as modeling the *precedence effect* [132, 133], in order to give less weight to frequencies where recently a loud sound was present.

Using just PHAT weighting, poor results were obtained and we concluded that the effect of the PHAT function should be tuned down. As it was explained and shown in [134], the main reason for this is that speech can exhibit both wide-band and narrow-band characteristics. For example, if uttering the word "shoe", "sh" component acts as a wide-band signal and voiced component "oe" as a narrow-band signal.

Based on the discussion above, the enhanced GCC-PHAT-β has the following form

$$\hat{R}^{\text{PHAT-}\beta\text{e}}_{ij}(\Delta\tau) = \sum_{k=0}^{L-1} \frac{G_{i,k}X_{i,k}G_{j,k}X^*_{j,k}}{\left(|X_{i,k}||X_{j,k}|\right)^{\beta}} e^{j2\pi\frac{k\Delta\tau}{L}}. \tag{3.8}$$

where $0 < \beta < 1$ is the tuning parameter.

3.2.3 *Direction of arrival estimation*

The TDOA between microphones i and j can be found by locating the peak in the cross-correlation

$$\Delta\tau_{ij} = \arg\max_{\Delta\tau} \hat{R}^{\text{PHAT}-\beta e}_{ij}(\Delta\tau). \tag{3.9}$$

Once TDOA estimation is performed, it is possible to compute the bearing of the sound source through series of geometrical calculations. It is assumed that the distance to the source is much larger than the array aperture, i.e. we assume the so called far-field scenario. Thus the expanding acoustical wavefront is modeled as a planar wavefront. Although this might not always be the case, being that human-robot interaction is actually a mixture of far-field and near-field scenarios, this mathematical simplification is still a reasonable one. Using the cosine law we can state the following (Fig. 3.1)

$$\varphi_{ij} = \pm\arccos\left(\frac{c\Delta\tau_{ij}}{a_{ij}}\right), \tag{3.10}$$

where a_{ij} is the distance between the microphones, c is the speed of sound, and φ_{ij} is the direction of arrival (DOA) angle.

Since we will be using more than two microphones one must make the following transformation in order to fuse the estimated DOAs. Instead of measuring the angle φ_{ij} from the baseline of the microphones, transformation to bearing θ_{ij} measured from the x axis of the array coordinate system (bearing line is parallel with the x axis when $\theta_{ij} = 0°$) is

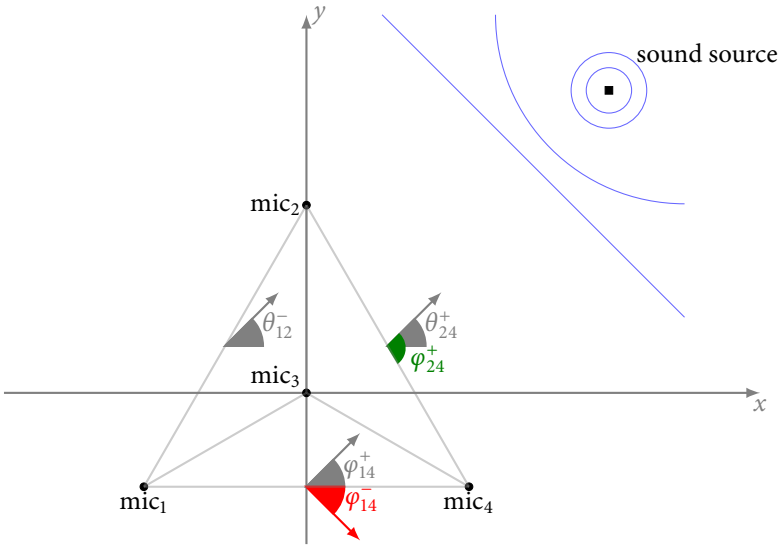


Figure 3.1: Direction of arrival angle transformation

performed. The transformation is done with the following equation (angles φ^{+}_{24} and θ^{+}_{24} in Fig. 3.1)

$$\begin{aligned}\theta^{\pm}_{ij} &= \alpha_{ij} \pm \varphi_{ij} \\ &= \operatorname{atan2}\left(\frac{y_j - y_i}{x_j - x_i}\right) \pm \arccos\left(\frac{c\Delta\tau_{ij}}{a_{ij}}\right).\end{aligned} \tag{3.11}$$

At this point we should note the following:

- under the far-field assumption, all the DOA angles measured anywhere on the baseline of the microphones are equal, since the bearing line is perpendicular to the expanding planar wavefront (angles θ^{-}_{12} and θ^{+}_{24} in Fig. 3.1)
- front-back ambiguity is inherent when using only two microphones (angles φ^{-}_{14} and φ^{+}_{14} in Fig. 3.1).

Having M microphones, (3.11) will yield $2 \cdot \binom{M}{2}$ possible bearing values. How to solve the front-back ambiguity and fuse the measurements is explained in Section 3.4, but at this point we turn to the analysis of microphone array geometry.

3.3 MICROPHONE ARRAY GEOMETRY

We find that microphone arrangement on a mobile robot is also an important issue and should be carefully analyzed. If we constrain the microphone placement in 2D, then two most common configurations present:

- square array – four microphones are placed on the vertices of a square. The origin of the reference coordinate system is at the intersection of the diagonals
- Y array – three microphones are placed on the vertices of an equilateral triangle, and the fourth is in the orthocenter which represents the origin of the reference coordinate system.

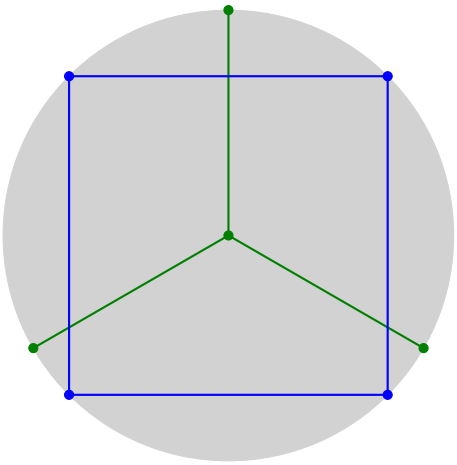


Figure 3.2: Possible microphone placement scenarios

The dimensions of the microphone array depend on the type of the surface it is placed on. In this section two microphone array configurations are compared as if placed on a circular surface with fixed radius r (see Fig. 3.2). Hence, both arrays are defined by their respective square and triangle side length a, which is equal to $a = r\sqrt{2}$ and $a = r\sqrt{3}$, respectively. Estimation of TDOA is influenced by the background noise, channel noise and reverberation, and the goal of (3.8) is to make the respective estimation as insensitive as possible to these influences. Under assumption that the microphone coordinates are measured accurately, we can see from (3.11) that the estimation of the bearing $\theta_{ij}^{\pm}$ depends solely on the estimation of the TDOA. Therefore, it is reasonable to analyze the sensitivity of bearing estimation to TDOA estimation error. Furthermore, it is shown that this sensitivity depends on the microphone array configuration. Firstly, we define the error sensitivity of bearing estimation to TDOA measurement, s_{ij}, as follows [133]

$$s_{ij} = \frac{\partial \theta_{ij}}{\partial(\Delta\tau_{ij})}. \tag{3.12}$$

By substituting (3.10) and (3.11) into (3.12) and applying simple trigonometric transformations, we gain the following expression

$$s_{ij} = \frac{c}{a_{ij}} \frac{1}{|\sin(\theta_{ij} - \alpha_{ij})|}. \tag{3.13}$$

From (3.13) we can see that there are two means by which error sensitivity can be decreased. The first is by increasing the distance between the microphones a_{ij}. This is kept under constraint of the robot dimensions and is analyzed for circle radius $r = 30$ cm, thus yielding square side length $a = 0.42$ cm and triangle side length $a = 0.52$ cm. The second is to keep the azimuth θ_{ij} as close to $90°$ relative to α_{ij} as possible. This way we are ensuring that the impinging source wave will be parallel to the microphones baseline. This condition could be satisfied if all the microphone pair baselines have the maximum variety of different orientations.

For the sake of the argument, let us set $c = 1$. The error sensitivity curves s_{ij}, as a function of bearing θ_{ij}, for Y and square array are shown in Fig. 3.3. We can see from Fig. 3.3 that the distance between the microphones a_{ij} mostly contributes to the offset of the sensitivity

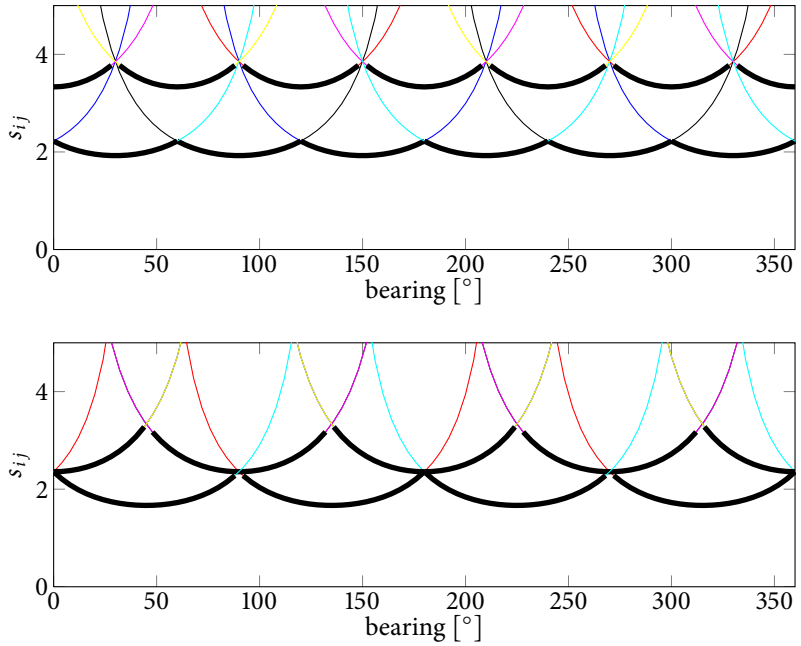


Figure 3.3: Error sensitivity of azimuth estimation for Y (upper plot) and square array (bottom plot)

curves, and that the variety of orientations affects the effectiveness of angle coverage. For Y array, Fig. 3.3 shows two groups of sensitivity curves: one for $a_{ij} = r$ (bottom marked by the upper black line), the length of the baseline connecting the microphones on the vertices with the microphone in the orthocenter, and other for $a_{ij} = r\sqrt{3}$ (bottom marked by the lower black line), the length of the baseline connecting the microphones on the vertices of the triangle. The first group has the largest error sensitivity value of 3.8 approximately, and the second group has the largest error sensitivity value of 2.2 approximately. For the square array, Fig. 3.3 shows also two groups of sensitivity curves: one for $a_{ij} = r\sqrt{2}$ (bottom marked by the upper black line), the side length of the square, and the other for $a_{ij} = 2r$ (bottom marked by the lower black line), the diagonal length of the square. The first group has the largest error sensitivity value of 3.3 approximately, and the second group has the largest error sensitivity value of 2.3 approximately. From the above discussion and figures we can see that the Y array maximizes baseline orientation variety, while the square array maximizes total baseline length (this length is defined as sum of all the distances between the microphones and is in favor by factor 1.2 for square array). This type of analysis can also be easily made for bigger and more complex microphone array systems in order to search for the best possible microphone placements. A possible scenario is that one of the microphones gets occluded and its measurement is unavailable or completely wrong. For Y array we have selected that one of the microphones on the vertices is occluded, since this is the most probable case, and for the square array it makes no difference, since the situation is only symmetrical for any microphone. Robustness of error sensitivity with respect to microphone occlusion is shown in Fig. 3.4 for both Y and square array, from which it can be seen that the result is far worse for Y array. This is logical, since we removed from the configuration two microphone pairs with largest baseline lengths. From the above discussion we can conclude that the square array is more robust to microphone occlusion.

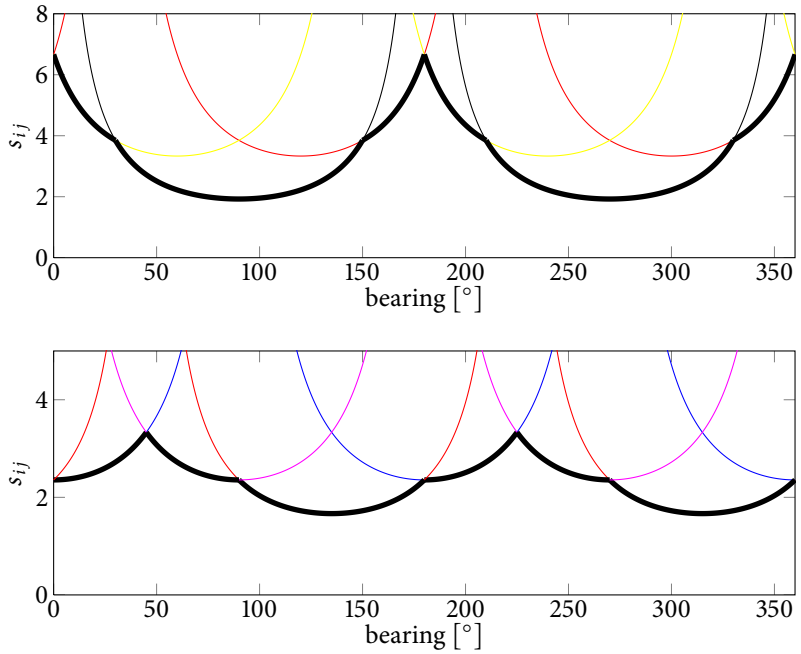


Figure 3.4: Error sensitivity of azimuth estimation for Y (upper plot) and square array (bottom plot) with one microphone occluded

To conclude, we can state the following; although Y array configuration places microphones in such a way that no two microphone-pair baselines are parallel (thus ensuring maximum orientation variety), square array has larger total baseline length, yielding smaller overall error sensitivity and greater robustness to microphone occlusion.

Furthermore, when considering microphone placement on a mobile robot from a practical point of view, square array has one more advantage. If the microphones are placed on the body of the robot (as opposed to the top of the robot, e.g. the head), problem occurs for Y array configuration considering the placement of the fourth microphone (the one in the orthocenter). However, the advantages of Y array should not be left out when considering tetrahedra microphone configurations (see [31]). Also if the two configurations are analysed with both having the same total baseline length, Y array would prove to have superior angle resolution [109].

3.4 TRACKING WITH THE PARTICLE FILTER

The problem at hand is to analyze and make inference about a dynamic system. For that, two models are required: one predicting the evolution of the speaker state over time (system model), and second relating the noisy measurements to the speaker state (measurement model). We assume that both models are available in probabilistic form. Thus, the approach to dynamic state estimation consists of constructing the a posteriori pdf of the state based on all available information, including the set of received measurements, which are further combined due to circular nature of the data, as a mixture of von Mises distributions.

Before presenting the models and the algorithm in details, we describe in general major successive steps of the algorithm. The algorithm starts with an initialization step at which we assume that the speaker can be located anywhere around the mobile robot, i.e. we

assume that the angle has a uniform distribution. At this point it would be practical to devise a way of discerning if the processed signal frame contains speech or not. This method would prevent misguided interpretations of the TDOA estimation due to speech absence, i.e. estimation from signal frames consisting of noise only. Hence, any further action is taken only if voice activity is detected by an appropriate algorithm (Chapter 4 is dedicated solely to voice activity detection algorithms). When the voice presence condition is fulfilled, the algorithm proceeds with predicting the state of the speaker trough the kinematics model described in Section 3.4.1. Once measurements are taken, a measurement model based on a mixture of von Mises distributions, described in Section 3.4.2, is constructed. Since this model is inherently multimodal, particle filtering approach, described in general form in Section 2.4.2 and particularly for this application in the ensuing Section 3.4.3, is utilized to represent the PDF of such measurement model and to effectively estimate the speaker bearing as the expected value of this PDF.

3.4.1 *Kinematics and state space equation*

The sound source kinematics is modeled by the well behaved Langevin motion model [125]

$$\begin{aligned} \begin{bmatrix} \dot{x}_t \\ \dot{y}_t \end{bmatrix} &= \alpha \begin{bmatrix} \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{bmatrix} + \beta \begin{bmatrix} v_x \\ v_y \end{bmatrix} \\ \begin{bmatrix} x_t \\ y_t \end{bmatrix} &= \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \delta \begin{bmatrix} \dot{x}_t \\ \dot{y}_t \end{bmatrix} \end{aligned} \tag{3.14}$$

where $[x_t, y_t]^{\mathrm{T}}$ is the location of the speaker, $[\dot{x}_t, \dot{y}_t]^{\mathrm{T}}$ is the velocity of the speaker at time index t, $v_x, v_y \sim \mathcal{N}(0, \sigma_v)$ is the stochastic velocity disturbance, α and β are model parameters, and δ is the time between update steps. Although the motion model of the speaker is defined in 2D by (3.14), which is found to describe well motion of the speaker [26, 124–126], in the end we will calculate the estimated speaker bearing via

$$\theta_t = \operatorname{atan2}(y_t, x_t). \tag{3.15}$$

3.4.2 *Von Mises distribution based measurement model*

Measurement of the sound source state with M microphones can be described by the following equation

$$\mathbf{z}_t = \mathbf{h}_t(\theta_t, n_t), \tag{3.16}$$

where $\mathbf{h}_t(.)$ is a non-linear function with noise term n_t, and $\mathbf{z}_t = [\theta^{\pm}_{ij}, \ldots, \theta^{\pm}_{M,M-1}]_t$, $i \neq j, \{i, j\} = \{j, i\}$ is the measurement vector defined as a set of bearings calculated from (3.11). Working with M microphones gives $N = \binom{M}{2}$ microphone pairs and $2N$ bearing measurements. Since $\mathbf{z}_t$ is a random variable of circular nature, we propose to model it with the von Mises distribution presented in Section 2.3.1. We restate the PDF here for completeness

$$p(\theta; \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} \exp\{\kappa \cos(\theta - \mu)\}, \tag{3.17}$$

where μ is the mean direction, κ is the concentration parameter and $I_0(\,.\,)$ is the modified Bessel function of the first kind and order zero. According to (3.11), a microphone pair

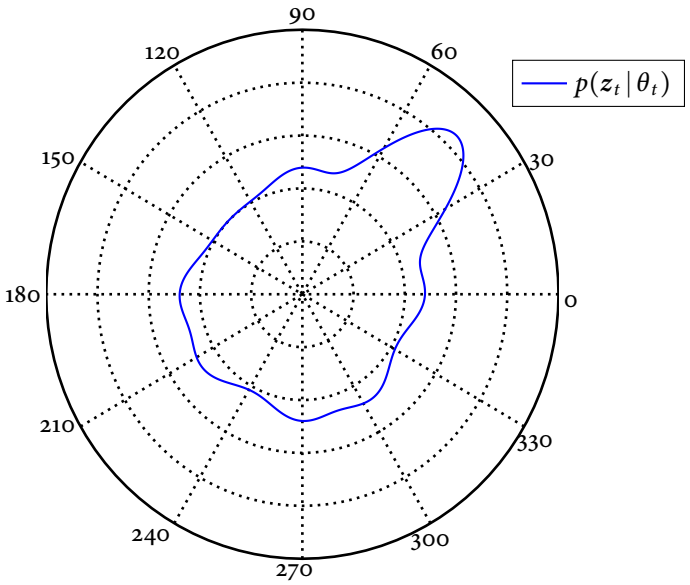


Figure 3.5: A mixture of several von Mises distributions wrapped on a unit circle (most of them having a mode at 45°)

$\{i, j\}$ measures two possible bearings θ^+_{ij} and θ^-_{ij}. Since we cannot discern from a single microphone pair which bearing is correct, we can say, from a probabilistic point of view, that both angles are equally probable. Therefore, we propose to model each microphone pair as a sum of two von Mises densities, yielding a bimodal PDF of the following form

$$\begin{aligned} p_{ij}\left(\theta_t; \theta^{\pm}_{ij,t}, \kappa\right) &= \frac{1}{2} p_{ij}\left(\theta_t; \theta^{+}_{ij,t}, \kappa\right) + \frac{1}{2} p_{ij}\left(\theta_t; \theta^{-}_{ij,t}, \kappa\right) \\ &= \frac{1}{4\pi I_0(\kappa)} \exp\left[\kappa \cos\left(\theta_t - \theta^{+}_{ij,t}\right)\right] + \\ &+ \frac{1}{4\pi I_0(\kappa)} \exp\left[\kappa \cos\left(\theta_t - \theta^{-}_{ij,t}\right)\right]. \end{aligned} \tag{3.18}$$

Having all pairs modeled as a sum of two von Mises densities, we propose a convex combination of all those pairs to represent the microphone array measurement model. Such a model has the following multimodal PDF

$$p(\mathbf{z}_t \mid \theta_t) = \sum_{\{i,j\}=1}^{N} w_{ij} p_{ij}\left(\theta_t; \theta^{\pm}_{ij,t}, \kappa\right), \tag{3.19}$$

where $\sum w_{ij} = 1$ is the mixture coefficient. These mixture coefficients are selected so as to minimize the overall error sensitivity. As it has been shown, the error sensitivity is function of the bearing. The goal of the coefficients w_{ij} is to give more weight in (3.19) to the most reliable PDF.

By looking at (3.13), we can see that the error sensitivity is the greatest when the argument in the sine function is zero. This corresponds to a situation when speaker is located at the baseline of a microphone pair. Furthermore, we can see that the error sensitivity is the smallest when speaker is on a line perpendicular to the microphone pair baseline. Since we need the coefficients w_{ij} to give the least weight to a microphone pair in the former situation

and the most weight to a microphone pair in the latter situation, it would be appropriate to calculate w_{ij} by inverting (3.13). However, we use scaled and inverted (3.13)

$$w_{ij} = \frac{0.5 + \left|\sin(\theta_{t-1} - \alpha_{ij})\right|}{1.5}. \tag{3.20}$$

where the ratio c/a_{ij} is set to one, since it is constant, and the coefficients are scaled so as to never cancel out completely a possibly unfavourable PDF. We can also see that the mixture coefficients are a function of the estimated bearing and that this form can only be applied after we have a reliable estimate of the bearing.

The model (3.19) represents our belief in the sound source bearing. A graphical representation of the analytical (3.19) is shown in Fig. 3.5. Of all the $2N$ measurements, half of them will measure the correct bearing, while their counterparts from (3.11) will have different (not equal) values. So, by forming such a linear opinion pool, PDF (3.19) will have a strong mode at the correct bearing value.

3.4.3 *Particle filtering for bearing estimation from a von Mises mixture*

We have presented the particle filter (PF) in its more general form in Section 2.4.2, while in this section we briefly recall it and connect it with the problem at hand. From a Bayesian perspective, we need to calculate some degree of belief in the state θ_t, given the measurements $\mathbf{z}_t$. Thus, it is required to construct the PDF $p(\theta_t \,|\, \mathbf{z}_t)$ which bears multimodal nature due to TDOA based localization algorithm. Therefore, particle filtering algorithm is utilized, since it is suitable for non-linear systems and measurement equations, non-Gaussian noise, multimodal distributions, and it has been shown in [26, 124–126] to be practical for sound source tracking. Moreover, in [26] it is successfully utilized to track multiple sound sources and in the same vein we will use the motion model (3.14) for the prediction part and later normalize the particles to a unit circle. In PF the posterior density function $p(\theta_t \,|\, \mathbf{z}_t)$ is represented by a set of random samples (particles) with associated weights and computes estimates based on these samples and weights.

Let $\{\theta_t^p, w_t^p\}_{p=1}^P$ denote a random measure that characterises the posterior PDF $p(\theta_t \,|\, \mathbf{z}_t)$, where $\{\theta_t^p,\ p = 1, \ldots, P\}$ is a set of particles with associated weights $\{w_t^p,\ p = 1, \ldots, P\}$. The weights are normalized so that $\sum_p w_t^p = 1$. Then, the posterior density at time instant t can be approximated as [73]

$$p(\theta_t \,|\, \mathbf{z}_t) \approx \sum_{p=1}^{P} w_t^p \delta(\theta_t - \theta_t^p), \tag{3.21}$$

where $\delta(.)$ is the Dirac delta measure. Thus, we have a discrete weighted approximation to the true posterior $p(\theta_t \,|\, \mathbf{z}_t)$.

The weights are calculated using the principle of importance resampling, where the proposal distribution is given by (3.14). In accordance to the sequential importance resampling (SIR) scheme, the weight update equation is given by [73]

$$w_t^p \propto w_{t-1}^p p(\mathbf{z}_t \,|\, \theta_t^p), \tag{3.22}$$

where $p(\mathbf{z}_t \,|\, \theta_t^p)$ is calculated by (3.19), thus replacing θ_t with particles θ_t^p. The next important step in PF is the resampling itself. The resampling step is solved by generating a new

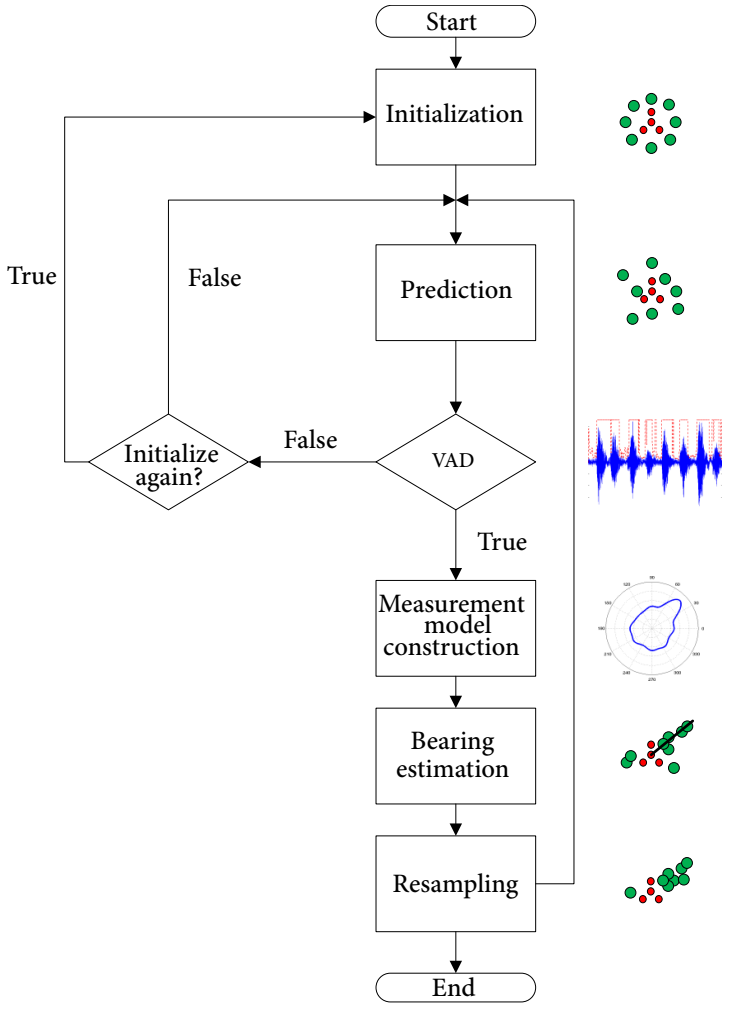


Figure 3.6: Flowchart diagram of the speaker localization and tracking algorithm based on the von Mises mixture likelihood and particle filtering

set of particles by resampling (with replacement) P times from an approximate discrete representation of $p(\theta_t \,|\, \mathbf{z}_t)$. After the resampling all the particles have equal weights, which are thus reset to $w_t^p = 1/P$. In the SIR scheme, resampling is applied at each time index. Since we have $w_{t-1}^p = 1/P \;\forall p$, the weights are simply calculated from

$$w_t^p \propto p(\mathbf{z}_t \,|\, \theta_t^p). \tag{3.23}$$

The weights given by the proportionality (3.23) are, of course, normalised before the resampling step.

The θ_t is then estimated simply via the following equation (3.15)

$$\begin{aligned} \hat{\theta}_t &= \mathrm{E}\left[\theta_t\right] = \operatorname{atan2}\left(\mathrm{E}[y_t], \mathrm{E}[x_t]\right) = \operatorname{atan2}\left(\mathrm{E}\left[\sin(\theta_t)\right], \mathrm{E}\left[\cos(\theta_t)\right]\right) \\ &= \operatorname{atan2}\left(\sum_{p=1}^{P} w_t^p \sin(\theta_t^p), \sum_{i=p}^{P} w_t^p \cos(\theta_t^p)\right), \end{aligned} \tag{3.24}$$

where $\mathrm{E}[\,.\,]$ is the expectation operator.

3.4.4 *Algorithm summary*

In order to get a clear overview of the complete algorithm, we present its flowchart diagram in Fig. 3.6, and hereafter describe each step of the algorithm with implementation details.

Initialization: At time instant $t = 0$ a particle set $\{\theta_0^p, w_0^p\}_{p=1}^P$ (velocities $\dot{x}_0$, $\dot{y}_0$ set to zero) is generated and distributed accordingly on a unit circle. Since the sound source can be located anywhere around the robot, all the particles have equal weights $w_0^p = 1/P$, $\forall\, p$, i.e. we assume that the angle has a uniform distribution.

Prediction: In this step all the particles are propagated according to the motion model given by (3.14).

Voice activity detection: In the speaker detection part a voice activity detector is applied to recorded signals. If no voice activity is detected, we proceed to a decision logic in which we either apply the motion model (3.14), in order to account for speaker moving during a silence period, or if this state lasts longer than a given threshold, the algorithm is reset and we simply go back to the initialization step. If voice activity is detected, then the algorithm proceeds to measurement model construction.

Measurement model construction: Upon receiving TDOA measurements, DOAs are calculated from (3.11) and for each DOA a bimodal PDF is constructed from (3.18). To form the proposed measurement model, all the bimodal PDFs are combined to form (3.19). The particle weights are calculated from (3.23), (3.19), and then normalized.

Bearing estimation: At this point we have the approximate discrete representation of the posterior density (3.19). The bearing is estimated via (3.24).

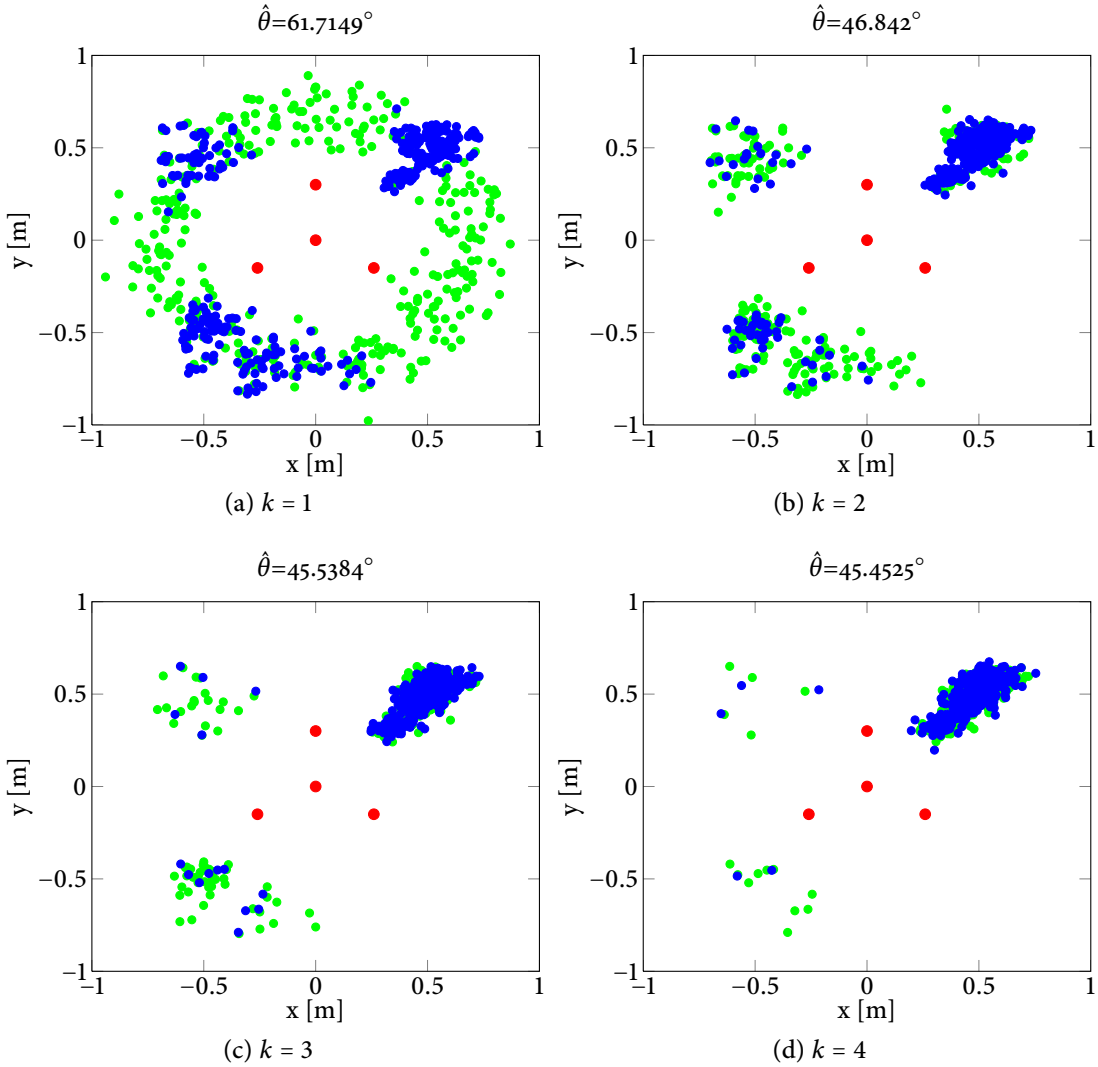
Resampling: This step is applied at each time index ensuring that the particles are resampled respective to their weights. After the resampling, all the particles have equal weights: $\{\theta_t^p, w_t^p\}_{p=1}^P \leftarrow \{\theta_t^p, 1/P\}_{p=1}^P$. The SIR algorithm is used (see [73]), but particle size adaptation is not performed, since we have a modest number of particles required for this algorithm. When the resampling is finished, the algorithm loops back to the speaker detection step.

3.4.5 *Experiments*

The proposed algorithm was thoroughly tested by simulation and experiments with a microphone array composed of four microphones arranged in either Y or square geometry (depending on the experiment). The circle radius for both array configurations was set to $r = 30$ cm, yielding side length of $a = 0.52$ cm for Y array and $a = 0.42$ cm for square array. Hereafter we present first an illustrative simulation and then the experimental results.

↬ SIMULATION. In order to get a deeper insight into particle behaviour, in this section we present an illustrative simulation. We constructed a measurement vector $\mathbf{z}_t$ similar to one that would be experienced during experiments. Six measurements were distributed close to the true value ($\theta = 45°$), while the other six were their counterparts, thus yielding

$$\begin{aligned}\mathbf{z}_t &= [\theta_{12}^- \ \theta_{13}^- \ \theta_{14}^+ \ \theta_{23}^+ \ \theta_{24}^+ \ \theta_{34}^+ \ \theta_{12}^+ \ \theta_{13}^+ \ \theta_{14}^- \ \theta_{23}^- \ \theta_{24}^- \ \theta_{34}^-] \\ &= [42° \ 44° \ 45° \ 45° \ 46° \ 48° \ 135° \ 75° \ 225° \ 15° \ 315° \ 255° \ ]\,.\end{aligned} \tag{3.25}$$



(a) $k = 1$ (b) $k = 2$

(c) $k = 3$ (d) $k = 4$

Figure 3.7: Simulation results of speaker localization with microphones in the Y configuration (red), the initial particle set (green) and the resampled particle set (blue)

The algorithm was tested with such $\mathbf{z}_t$ for the first four iterations of the algorithm execution. The results are shown in Fig. 3.7 where particles before and after the resampling are shown. We can see that in the first step the particles are spread uniformly around the microphone array. After the first measurement, the particle weights are calculated and the particles are resampled according to their respective weights. This procedure is repeated throughout the next iterations, and we can see in Fig. 3.7 that the particles converge to the true bearing value.

REAL-WORLD DATA. The microphone array consisting of four omnidirectional microphones was placed on a Pioneer 3DX robot as shown in Fig. 2.2. Audio interface is composed of low-cost microphones, pre-amplifiers and external USB soundcard (whole equipment costing circa 150 Euros). All the experiments were done in real-time, yielding $L/F_s = 21.33$ ms system response time. Real-time multichannel signal processing for the

Matlab implementation was realised with the Playrec[1] utility. The experiments were conducted in a classroom which has dimensions of 7 m × 7 m × 3.2 m, parquet wooden flooring, one side covered with windows and a reverberation time of 850 ms. During the experiments, typical noise conditions were present, like computer noise and air ventilation. In the experiments two types of sound sources were used; a white Gaussian noise (WGN) source and a single speaker uttering a test sequence.

The first set of experiments was conducted in order to qualitatively assess the performance of the algorithm. Two types of experiments were performed; one with a stationary robot and the other with a moving robot.

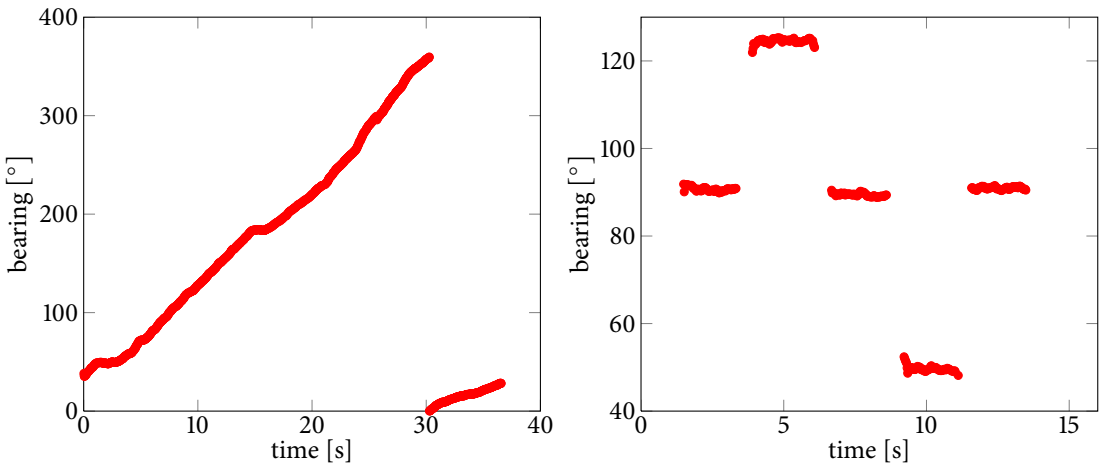
In the experiments with the stationary robot Y array configuration was used, and a loud white noise sound source, since it represents the best-case scenario in which all the frequency bins are dominated by the information about the sound source location. Two cases were analyzed. Figure 3.8a shows the first case in which a sound source moved around the mobile robot at a distance of 2 m making a full circle. Figure 3.8b shows the results from the second case, where a sound source made rapid angle changes under 0.5 s, thus simulating a turn-take scenario at a distance of 2 m . Both experiments were repeated with smaller array dimensions ($a = 30$ cm), resulting in smaller angle resolution, and no significant degradations to the algorithm were noticed.

The second set of experiments was conducted in order to quantitatively assess the performance of the algorithm. In order to do so, a ground truth system needed to be established. The Pioneer 3DX platform on which the microphone array was placed was also equipped with SICK LMS200 LRS. The adaptive sample-based joint probabilistic data association filter (ASJPDAF) for multiple moving objects developed in [135] was used for leg tracking. The authors find it to be a good reference system in controlled conditions. Measurement accuracy of the LMS200 LRS is ±35 mm, and due to determining the speaker location as the centre between the legs of the speaker, we estimate the accuracy of the ASJPDAF algorithm to be less than 0.5°. In the experiments, a human speaker walked around the robot uttering a sequence of words, or carried a mobile phone for white noise experiments, while the ASJPDAF algorithm measured range and bearing from the LRS scan.

In this set of experiments three parameters were calculated: detection reliability, root-mean-square error (RMSE) and standard deviation. To make comparison possible, the chosen parameters are similar to those in [26]. The detection reliability is defined as the percentage of samples that fall within ±5° from the ground truth bearing, RMSE is calculated as deviation from the ground truth bearing, while standard deviation is simply the deviation of the measured set from its mean value.

The experiments were performed at three different ranges for both the Y and square array configurations, and, furthermore, for each configuration voice and white noise source were used. The white noise source was a train of 50 element 100 ms long bursts, and for the voice source speaker uttered: "Test, one, two, three", until reaching the number of 50 words in a row. In both configurations the source changed angle in 15° or 25° intervals, depending on the range, thus yielding in total 4150 sounds played. The results of the experiments are summed up in Table. 3.1, from which we can see (for both array configurations) that for close

[1] http://www.playrec.co.uk/



(a) Full circle (b) Turn-take

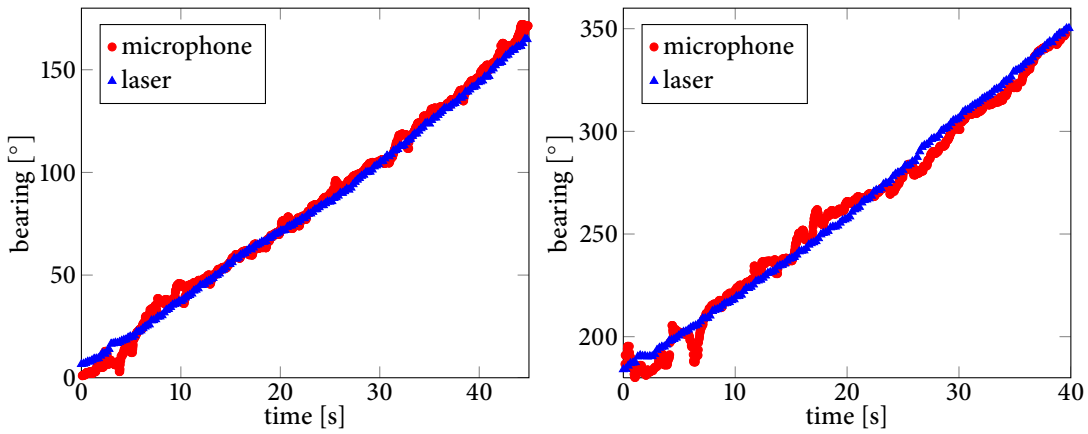
Figure 3.8: Tracking a moving white Gaussian noise sound source

Table 3.1: Experimental results of the algorithm performance for Y and square array configuration

	Y-array		Square array	
Range	W. noise	Voice	W. Noise	Voice
	Detection [%]			
1.50 [m]	97.43	98.93	99.43	97.71
2.25 [m]	97.71	92.86	98.00	96.0
3.00 [m]	94.57	86.86	96.00	91.43
	RMSE [°]			
1.50 [m]	1.90	2.20	1.72	2.19
2.25 [m]	1.61	3.07	1.99	2.83
3.00 [m]	2.38	4.58	1.80	3.95
	Std. deviation [°]			
1.50 [m]	0.96	1.59	0.94	1.36
2.25 [m]	1.10	2.78	1.04	2.30
3.00 [m]	1.65	3.85	1.14	3.01

interaction the results are near perfect. High detection rate and up to 2° error and standard deviation rate at distance of 1.5 m are negligible. In general, for both array configurations performance slowly degrades as the range increases. With the range increasing the far-field assumption does get stronger, but the angular resolution is lower, thus resulting in higher error and standard deviation. Concerning different array configurations, it can be seen that square array shows better results in all three parameters, on average up to 2.3% in detection, 0.4° in RMSE, and 0.4° in standard deviation.

In [26], where an array of eight microphones was used and a beamforming approach, similar experiments were performed with an open and closed array configuration. For the open configuration, our algorithm shows smaller detection reliability of less than 4%



(a) Y array (b) Square array

Figure 3.9: Tracking a moving speaker with the microphone array and laser range sensor

on average, and larger RMSE of less than 2° on average. For the closed configuration, our algorithm shows the same detection reliability on average, and larger RMSE of less then 1° on average.

From the previous discussion we can see that the algorithm proposed in [26] shows better or equal performance, on average, in both detection reliability and RMSE. However, in [26] an array of eight, compared to four, microphones was used and a beamforming approach was utilized. The beamforming approach is based on dividing the space around the mobile robot into a direction grid, steering the microphone array to all possible directions, and for each direction an expression like (3.8) is calculated for all microphone pairs. Although more complex, it does however have an advantage of being able to track multiple simultaneously talking speakers.

The third set of experiments was conducted in order to assess the tracking performance of the algorithm. A speaker made a semicircle at approximately 2 m range around the robot uttering: "Test, one, two, three", while at the same time legs were tracked using LRS. The experiment was made for both array configurations. Figures 3.9a and 3.9b show the bearing measured with the leg tracker and with the microphone array arranged in the Y and square configurations, respectively. It can be seen that the square array, in this case, shows bigger deviations from the laser measured bearing than the Y array does. In Fig. 3.9b at 6.3 s, one of the drawbacks of the algorithm can be seen. It is possible that at an occasion, erroneous measurements might outnumber the correct ones. In this case, wrong bearing will be estimated for that time, but as can be seen in Fig. 3.9b the algorithm gets back on track in a short time period.

3.5 TRACKING WITH THE VON MISES MIXTURE

In this section, we propose to model the complete bearing-only tracking process with the von Mises distribution; from the state representation and transition probability to the measurement likelihood. Compared to the PF, the benefits of the proposed approach lie in representing the function and not just the density, and in the fact that less components are needed to model the state. For the classical Bayesian tracking procedure with a mixture of

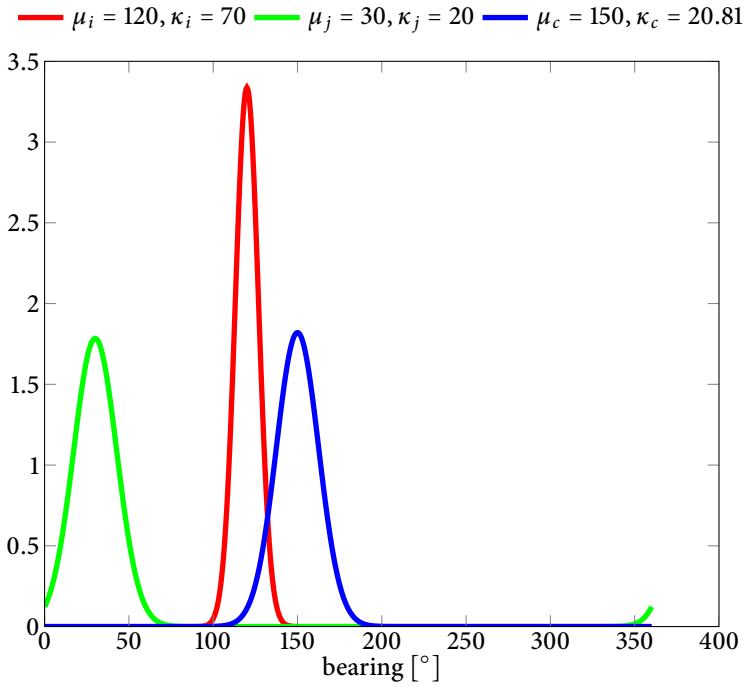


Figure 3.10: Convolution $p(x; \mu_c, \kappa_c)$ of two von Mises distributions $p(x; \mu_i, \kappa_i)$ and $p(x; \mu_j, \kappa_j)$

von Mises densities we show the solution to the following problems: (i) the convolution and (ii) the product of two von Mises distributions, (iii) the algorithm for component reduction of a mixture of von Mises distributions, and (iv) the analytical expression for the entropy of a mixture of von Mises distributions in order to have a measurement of the state uncertainty. The solution for the first two problems are presented from the literature, the third problem is solved by adapting a component reduction technique for Gaussian distributions, while the fourth problem is solved by deriving the entropy from the beginning.

As stated in Section 2.4 Bayesian tracking procedure consists of two steps: prediction and update [72, 73]. The prediction step involves calculating the prior PDF via the total probability theorem (or Chapman-Kolmogorov equation). In this section we shall assume that our motion model consists only of adding von Mises noise to the previous state which will enable us to calculate the prediction step analytically (since a sum of independent random variables has a distribution which is result of the convolution of the summands). In the next, update, step, the posterior at time t is calculated via the Bayes theorem which includes the product of two distributions and their respective normalization. Given that, we now explicitly calculate the relations (2.28) and (2.30) for von Mises distributions.

3.5.1 *Convolution of the von Mises distributions*

Given two von Mises PDFs, $p(\theta; \mu_i, \kappa_i)$ and $p(\theta; \mu_j, \kappa_j)$, the resulting convolution of a predicted state will be of the following form [46]

$$h(\theta) = \frac{1}{2\pi I_0(\kappa_i) I_0(\kappa_j)} I_0\left(\left\{\kappa_i^2 + \kappa_j^2 + 2\kappa_i\kappa_j \cos(\theta - [\mu_i + \mu_j])\right\}^{1/2}\right), \tag{3.26}$$

which in fact is not a von Mises distribution, but can be well approximated by the following von Mises PDF [46]

$$h(\theta) \approx p(\theta; \mu_i + \mu_j, A^{-1}(A(\kappa_i)A(\kappa_j)) \tag{3.27}$$

where we recall the function $A(\,.\,)$ from Chapter 2

$$A(\kappa) = \frac{I_1(\kappa)}{I_0(\kappa)}, \; \kappa \geq 0 \tag{3.28}$$

as the ratio of the modified Bessel functions of order one and order zero, and $A^{-1}(\,.\,)$ is its inverse. An illustration of the convolution is shown in Fig. 3.10.

3.5.2 *Product of the von Mises distributions*

The numerator in the update step given by (2.28) involves a calculation of a product of two von Mises distributions. Given the following von Mises PDFs, $p(\theta; \mu_i, \kappa_i)$ and $p(\theta; \mu_j, \kappa_j)$, the resulting product is of the following form (see Appendix A.1) [136]

$$\begin{aligned} g(\theta) &= \frac{1}{4\pi^2 I_0(\kappa_i) I_0(\kappa_j)} \exp\left[\kappa_{ij} \cos(\theta - \mu_{ij})\right] \\ &= \frac{I_0(\kappa_{ij})}{2\pi I_0(\kappa_i) I_0(\kappa_j)} \frac{1}{2\pi I_0(\kappa_{ij})} \exp\{\kappa_{ij} \cos(x - \mu_{ij})\}, \end{aligned} \tag{3.29}$$

where

$$\mu_{ij} = \mu_i + \operatorname{atan2}\left(-\sin\Delta\mu, \kappa_i/\kappa_j + \cos\Delta\mu\right) \tag{3.30}$$

$$\kappa_{ij} = \sqrt{\kappa_i^2 + \kappa_j^2 + 2\kappa_i\kappa_j \cos\Delta\mu}, \tag{3.31}$$

and $\Delta\mu = \mu_i - \mu_j$. The product in (3.29) is an unnormalized von Mises distribution scaled by the factor

$$\frac{I_0(\kappa_{ij})}{2\pi I_0(\kappa_i) I_0(\kappa_j)}. \tag{3.32}$$

Note that in order to complete the update step, we still need to calculate (2.32) which will normalize the result from (3.29) and yield a true density. An illustration of the normalized product is shown in Fig. 3.11.

It is interesting to note at this point that the product of von Mises distributions calculated by (3.29) has very different properties than the product of Gaussian distributions. For an example, the concentration parameter of the product is a function of the factor pair mean directions and concentrations, while in the case of Gaussian distributions, the variance of the product is only function of the factor pair variances. Given that, if the distance between factor pair mean directions is large enough, it is possible that the concentration parameter of the product will be smaller (representing higher uncertainty) than any concentration parameter of the factor pair. Indeed, a product of von Mises distributions with equal concentration parameters and antipode mean directions will yield a uniform distribution.

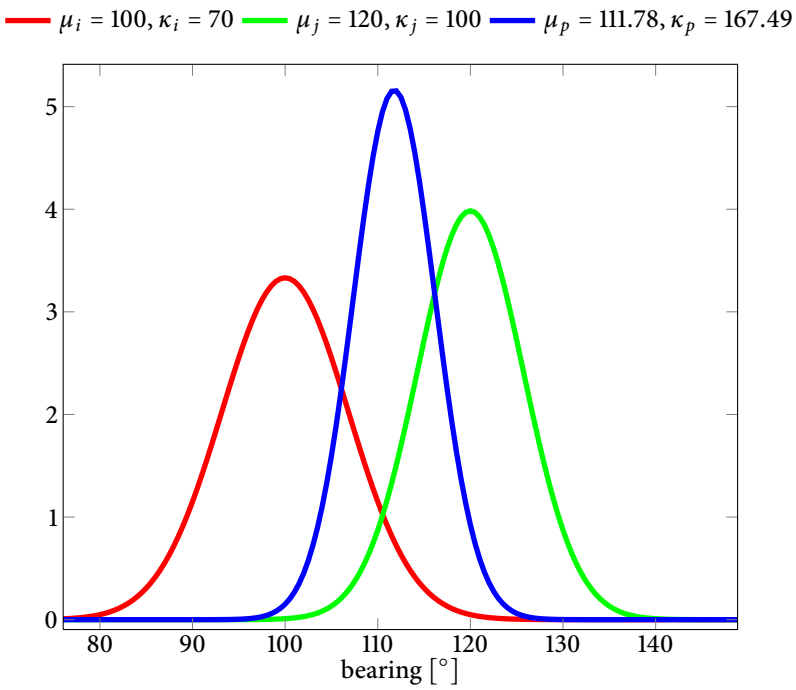


Figure 3.11: Product $p(x; \mu_p, \kappa_p)$ of two von Mises distributions $p(x; \mu_i, \kappa_i)$ and $p(x; \mu_j, \kappa_j)$

3.5.3 *Von Mises mixture filtering for bearing estimation*

Although there are several distributions appropriate for circular models [46], the von Mises distribution is the most commonly used and studied, since it provides a closed-form analytical framework for many applications. Given that, we represent the posterior at time $t-1$ as a convex combination of N von Mises PDFs

$$p(\theta_{t-1} \,|\, \boldsymbol{z}_{1:t-1}) = \sum_{i=1}^{N} w_{t-1,i} \frac{1}{2\pi I_0(\kappa_{t-1,i})} \exp\left[\kappa_{t-1,i} \cos(\theta_{t-1} - \mu_{t-1,i})\right], \tag{3.33}$$

where $\sum_i w_{t-1,i} = 1$. As stated earlier, the state transition involves just adding von Mises noise to the previous state, which, in effect spreads the posterior (increases the uncertainty) in the prediction step. Thus the state evolution density is given by

$$p(\theta_t|\theta_{t-1}) = \frac{1}{2\pi I_0(\kappa^q)} \exp\left[\kappa^q \cos(\theta_t - \theta_{t-1})\right]. \tag{3.34}$$

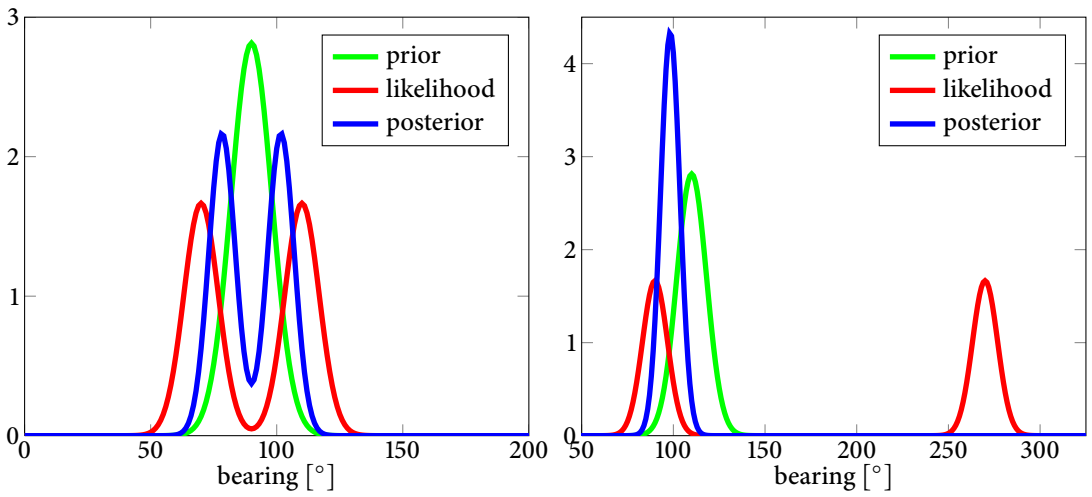
Now, the prediction step of the mixture filter consist of convolving (3.33) with (3.34) which yields the predicted VM mixture

$$p(\theta_t \,|\, \boldsymbol{z}_{1:t-1}) = \sum_{i=1}^{N} w_{t|t-1,i} \frac{1}{2\pi I_0(\kappa_{t|t-1,i})} \exp\left[\kappa_{t|t-1,i} \cos(\theta_t - \mu_{t|t-1,i})\right], \tag{3.35}$$

where $\mu_{t|t-1,i} = \mu_{t-1,i}$, $w_{t|t-1,i} = w_{t-1,i}$ and $\kappa_{t|t-1,i} = A^{-1}(A(\kappa_{t-1,i})A(\kappa^q))$.

Following a similar train of thought as for (3.33), we also write the sensor model as a convex combination of M von Mises PDFs

$$p(\boldsymbol{z}_t \,|\, \theta_t) = \sum_{i=1}^{M} w_i^z \frac{1}{2\pi I_0(\kappa_i^z)} \exp\left[\kappa_i^z \cos(\theta_t - z_{t,i})\right], \tag{3.36}$$



(a) 'Splitting' the prior (b) 'Ignoring' the distant component at 270°

Figure 3.12: Bayesian update examples of a single von Mises prior with a 2-component von Mises likelihood

where $\sum w_i^z = 1$. Note that by doing so, we also allow the sensor model to be a multimodal PDF. The concentration parameters of the sensor model and the state evolution density are determined empirically. Now, the update step of the VM mixture filter consists of multiplying (3.36) and (3.35) and normalizing the product according to the Bayes rule. The update step normalizes properly the posterior mixture, but the process affects also the component weights, i.e. the weight of the resulting component is not just the product of the individual factor weights but is also scaled by (3.32). This scaling factor bears a striking similarity to the convolution of two von Mises densities (3.26), and indeed by integrating the product of the prediction and likelihood we are convolving the pairs of components in θ_t. Just as (3.26) is well approximated by (3.27) so can the scaling factor (3.32) be approximated by a von Mises distribution which has the following form for a combination of a predicted mixture component i and a likelihood component j

$$\frac{1}{2\pi I_0(\tilde{\kappa})} \exp\left[\tilde{\kappa}\cos(z_{t,j} - \mu_{t|t-1,i})\right], \tag{3.37}$$

where $\tilde{\kappa} = A^{-1}(A(\kappa_j^z)A(\kappa_{t|t-1,i}))$. This result is similar to the case of Gaussian distributions [137] and in the form of (3.37) can be seen as a kind of von Mises 'innovation'. In the actual implementation it is not necessary to use the approximation in (3.37)—it simply offers an intuitive form for an interpretation. In the end, the updated VM mixture is as follows

$$p(\theta_t \,|\, \boldsymbol{z}_{1:t}) = \sum_{i=k}^{NM} w_{t|t,k} \frac{1}{2\pi I_0(\kappa_{t|t,k})} \exp\left[\kappa_{t|t,k}\cos(\theta_t - \mu_{t|t,k})\right], \tag{3.38}$$

where the posterior $\mu_{t|t,i}$ and $\kappa_{t|t,i}$ are calculated via (3.30) and (3.31), while the posterior component weights are evaluated as

$$\begin{aligned} w_{t|t,k} &= w_{t|t-1,i} w_j^z \cdot \frac{I_0\left(\left\{\kappa_{t|t-1,i}^2 + {\kappa_j^{(z)}}^2 + 2\kappa_{t|t-1,i}\kappa_j^z \cos(z_{t,j} - \mu_{t|t-1,i})\right\}^{1/2}\right)}{2\pi I_0(\kappa_{t|t-1,i}) I_0(\kappa_j^z)} \\ &\approx w_{t|t-1,i} w_j^z \cdot \frac{1}{2\pi I_0(\tilde{\kappa})} \exp\left[\tilde{\kappa}\cos(z_{t,j} - \mu_{t|t-1,i})\right] \end{aligned} \tag{3.39}$$

An illustration of the Bayes update process of a single von Mises prior with a 2-component von Mises likelihood is shown in Fig. 3.12. The first example in Fig. 3.12a shows update with a likelihood whose components of equal concentration parameters were symmetrically situated around the prior's mean direction, which resulted in 'splitting' the prior in the direction of the two likelihood components. The second example in Fig 3.12b shows update with a likelihood where one component was close to being antipodal to the prior's mean direction. This basically resulted with the update ignoring this distant component, since, indeed, given the prior such measurement is deemed as unlikely.

Finally, from a multimodal distribution we infer the state θ_t as a maximum a posteriori (MAP) estimate from the posterior $p(\theta_t \,|\, \boldsymbol{z}_{1:t})$

$$\hat{\theta}_t = \arg\max_{\theta_t} p(\theta_t \,|\, \boldsymbol{z}_{1:t}). \tag{3.40}$$

This is solved by numerically evaluating the mixture with a suitably chosen resolution and then taking the maximum value of the density.

Basically, a Bayesian tracking algorithm with previously defined state representation, motion model and sensor model, would consist of: (i) initially setting up an a priori distribution via (3.33) (N von Mises PDFs uniformly spread with small κ), (ii) convolving (3.33) with the state evolution PDF (3.34), (iii) multiplying the result of the convolution with (3.36), (iv) estimating the state, and then repeating steps (ii), (iii), and (iv) over time. The only problem with the previous procedure is the step (iii), where the state representation consisting of N von Mises PDFs is multiplied with M von Mises PDFs of the sensor model. This yields MN von Mises PDFs and would hence grow geometrically in time. In order to solve this problem, we need to reduce the number of the components in the mixture.

3.5.4 *Reducing the number of mixture components*

Since in this thesis the main goal is to utilize the reduction in the context of tracking, thus putting a constraint on the execution time, we propose a variant of the West's algorithm [80] for reduction of the number of von Mises components, which in its original form has computational complexity of $\mathcal{O}(N \log N)$ [21]. For a more in-depth study of the problem of von Mises mixture reduction please confer [127], while for a comparison of reduction algorithms in Gaussian mixtures confer [21]. West's algorithm, in essence, reduces the number of components by searching for the nearest neighbour, and then replaces the pair with a single component whose parameters are an average of the pair's values. Originally, this algorithm was developed to reduce the number of components with equal variances, with similarity criteria being the nearest neighbour in the mean value. In order to adapt the algorithm for reducing the mixture of von Mises components, we introduce the following modifications.

Let us assume that we have the following two VM components: $w_i p_i = w_i p(\theta; \mu_i, \kappa_i)$ and $w_j p_j = w_j p(\theta; \mu_j, \kappa_j)$. Since we are not working with true PDFs, but with mixture components, which are scaled PDFs, as a measure of similarity we consider the *scaled symmetrized KL distance* [138], defined by

$$D_{s\mathrm{KL}}(w_i p_i, w_j p_j) = \frac{1}{2}(w_i D_{\mathrm{KL}}(p_i, p_j) + w_j D_{\mathrm{KL}}(p_j, p_i)) + \frac{1}{2}(w_i - w_j) \log \frac{w_i}{w_j}, \tag{3.41}$$

where D_{KL} represents the KL distance between the two von Mises components (see Appendix A.2) [82, 127]

$$D_{\mathrm{KL}}(p_i, p_j) = \log \frac{I_0(\kappa_j)}{I_0(\kappa_i)} + A(\kappa_i)(\kappa_i - \kappa_j \cos(\mu_i - \mu_j)). \tag{3.42}$$

Note that the KL distance is a generalized distance functional, which is not symmetric nor it satisfies the triangle inequality, but it is positive definite, i.e. $D_{\mathrm{KL}}(p_i, p_j) \geq 0$ and $D_{\mathrm{KL}}(p_i, p_j) = 0$ if and only if $p_i = p_j$. However, with symmetrization we have assured that $D_{\mathrm{KL}}(p_i, p_j) = D_{\mathrm{KL}}(p_j, p_i)$.

Once we have selected the components to be merged, e.g. $w_i p(\theta; \mu_i, \kappa_i)$ and $w_j p(\theta; \mu_j, \kappa_j)$, we calculate the new component parameters, w^*, μ^* and κ^*, via following equations which in effect are optimal in the KL distance sense (see Appendix A.3) [127]

$$\tan \mu^* = \frac{w_i A(\kappa_i) \sin \mu_i + w_j A(\kappa_j) \sin \mu_j}{w_i A(\kappa_i) \cos \mu_i + w_j A(\kappa_j) \cos \mu_j} \tag{3.43}$$

$$w^{*2} A^2(\kappa^*) = w_i^2 A^2(\kappa_i) + w_j^2 A^2(\kappa_j) + 2 w_i w_j A(\kappa_i) A(\kappa_j) \cos(\mu_i - \mu_j), \tag{3.44}$$

where $w^* = w_i + w_j$. We can see that computing κ^* involves an implicit equation (3.44) that is a ratio of Bessel functions and that it cannot be solved analytically. In [63] the following approximation was proposed. If we divide (3.44) by w^{*2}, take the square root, and denote the right-hand side of the resulting equation with r, then the approximate solution to κ^* is

$$\kappa^* = \frac{2r - r^3}{1 - r^2}. \tag{3.45}$$

This approximation could be further refined by using (3.45) as a starting point in iterative numerical procedures for solving $A(\kappa^*) - r = 0$. Note the close relation of (3.45) and (2.10). The rest of the modifications are minor, and the pseudocode is given in Algorithm 3.

3.5.5 *Entropy of the von Mises mixture*

In tracking applications it is often very practical, if not necessary, to have a measure of uncertainty of the tracked state. While the uncertainty of unimodal distributions is characterized by their respective moments, for multimodal distributions the same is not that straightforward. Therefore, for the latter case, entropy is usually utilized for uncertainty calculation and as a practical feature in track management [21, 139].

Entropy of a mixture of von Mises-Fisher distributions, a probability distribution on a sphere, can be found in [140]. However, reducing the dimension of the result in [140] in order to derive an expression for entropy of a mixture of von Mises distributions is not a straightforward task, and therefore we derive a closed-form solution in this section.

A measure of entropy can take many analytical forms. Shannon entropy of a mixture of distributions cannot be expressed in closed-form, while Rényi entropies usually offer a more suitable framework for analytical calculations [141]. Therefore, to calculate entropy of the von Mises mixture, we used the Rényi entropy, which of order α is defined as follows [142]:

$$H_\alpha(\theta) = \frac{1}{1 - \alpha} \log \int p^\alpha(\theta) \, \mathrm{d}\theta, \tag{3.46}$$

Algorithm 3: Reduction of the von Mises mixture

Require: Components parameters $\mathcal{P} = \{\mu_i, \kappa_i, w_i\}_{i=1}^{NM}$

Ensure: Reduced component parameters $\mathcal{Q} = \left\{\mu_j^*, \kappa_j^*, w_j^*\right\}_{j=1}^{N}$

```
# Order set P ascending by weights
P ← {P : w_i ≤ w_j, i < j, i, j ∈ {1, 2, ..., |P|}}
while |P| > N do
    for i = 2 : |P| do
        d(i) ← D_sKL(w_1 p_1, w_i p_i)
    end for
    j ← arg min_{i∈2,3,...,|P|} d(i)
    # Remove components 1 and j
    P ← P ∖ {μ_i, κ_i, w_i}_{i=1,j}
    μ* ← calculate via (3.43)
    κ* ← calculate via (3.45)
    w* ← w_1 + w_j
    # Insert the merged component by weight
    P ← P ∪ {μ*, κ*, w*}
end while
Q ← P
```

where $1 \leq \alpha < \infty$. In the limit $\alpha \to 1$ Rényi entropy becomes Shannon entropy. The quadratic Rényi entropy of a von Mises mixture has the following form (detailed derivation can be found in Appendix A.4)

$$H_2(\theta_t) = -\log \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \frac{I_0(\kappa_{ij})}{2\pi I_0(\kappa_i) I_0(\kappa_j)}, \tag{3.47}$$

where $w_{ij} = w_i w_j$ and κ_{ij} is given by (3.31). Note that we have lost explicit dependence on θ_t. But on closer inspection, we can see that the state is implicitly included in κ_{ij} through the difference $\Delta\mu = \mu_i - \mu_j$. We can also utilize the symmetry $\kappa_{ij} = \kappa_{ji}$ in order to reduce the number of terms in the double sum in (3.47)

$$H_2(\theta_t) = -\log \frac{1}{2\pi} \left[\sum_{i=1}^{N} \frac{I_0(2\kappa_i)}{I_0^2(\kappa_i)} + 2 \sum_{\substack{i,j=1 \\ i<j}}^{N} \frac{I_0(\kappa_{ij})}{I_0(\kappa_i) I_0(\kappa_j)} \right]. \tag{3.48}$$

3.5.6 *Experiments*

In this section we investigate the application of the von Mises mixture algorithm in a bearing-only tracking scenario. Two experiments were conducted. The first, synthetic experiment analyzes two scenarios: a speaker making a full circle and a turn-take scenario. Furthermore, the proposed algorithm is compared to the tracking method based on the PF. In the second, real-world data experiment the mixture algorithm is tested also in a speaker making a full circle and a turn-take scenario.

SYNTHETIC DATA. We have simulated two trajectories of a maneuvering object in 2D, where the dynamics of the system were described by a jump-state Markov model [143]. The second trajectory had a rapid change in the bearing value to simulate a turn-taking scenario in order to test the capability of the algorithms to keep up with the track in such situations. For an example, this might occur when one speaker stops talking and the other continues, or the currently talking speaker stops, moves around the robot and then continues talking again. Note that the application of the described speaker localization algorithm is to detect and track the currently active speaker, and not to to detect and track multiple concurrently talking speakers and keep separate tracks for each one.

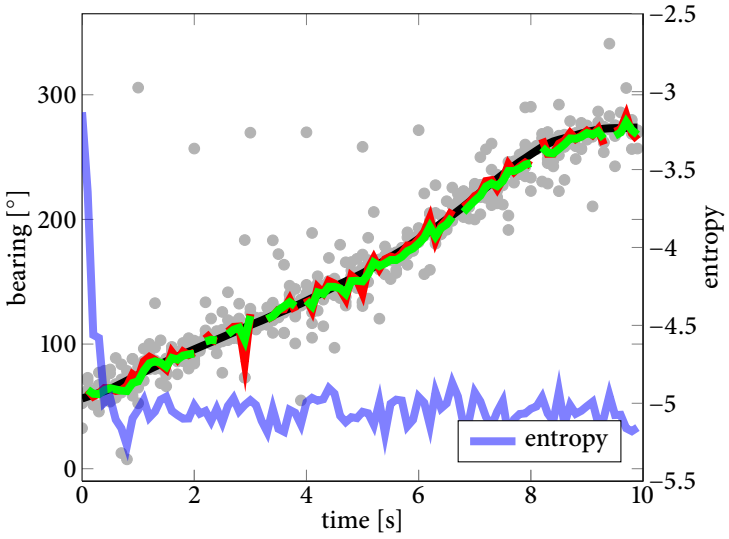
In order to make the simulation as realistic as possible (i) measurements were corrupted with von Mises noise of $\kappa = 70$ to model measurement noise, (ii) outliers where added with probability $P_O = 0.3$, i.e. close to 30% of measurements at random locations were corrupted with von Mises noise of $\kappa = 5$, and (iii) detection probability was $P_D = 0.9$, i.e. close to 10% of measurements at random locations were discarded.

For the von Mises mixture estimator, we used 12 components with mean directions uniformly spread over 0 to 2π, the process model was a single von Mises PDF (just the process noise), while the likelihood consisted of 12 components. The state was always represented with 12 components but concentration parameters changed at each iteration.

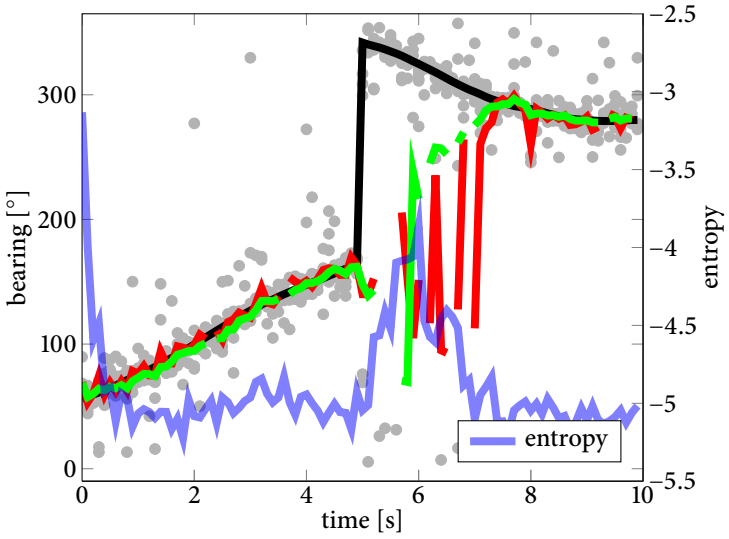
The PF was implemented as described in Section 3.4, where the likelihood also consisted of 12 von Mises PDFs, the state was represented with 360 particles, and the process model was a Langevin motion model [125]. Instead of resampling we used a variant of regularization [144], where we placed a von Mises kernel on each particle instead of a Gaussian distribution and drew new particles from such a multimodal distribution.

The results of the bearing estimation of both trajectories with the mixture of von Mises PDFs and with the PF along with corresponding entropies are shown in Fig. 3.13. For the first trajectory, Fig. 3.13a, we can see that both estimators have similar performance—RMSE was 2.7° and 2.8° for the von Mises mixture estimator and PF, respectively. The second trajectory, Fig. 3.13b, depicts the turn-taking scenario. We can see that again both show similar performance, and were a bit reluctant at the beginning to switch to a new bearing value. Concerning the entropies, at the beginning the entropy is largest since the distribution is close to uniform. As the filter is updated with measurements the entropy drops. We can also see the result of the turn-take at 5–7 s in the second trajectory where the entropy rose due to discrepancy between the believed state and measurements. Of course, both could be tuned to respond faster to rapid changes by decreasing κ of the transition PDF or by increasing κ of the measurement likelihood, but this would be at the cost of higher sensitivity to outliers. The former parameter tuning depends on the characteristics of the sensor measurements—if we expect large percentage of outliers, then we should make the estimator more inert, and vice-versa.

The number of parameters required for the state representation was smaller in the case of the mixture filter. We used 12 kernels, i.e. 36 parameters including the means, concentration parameters, and weights, while for the particle filter we used 360 particles, i.e. 360 parameters after regularization (due to equal particle weights).



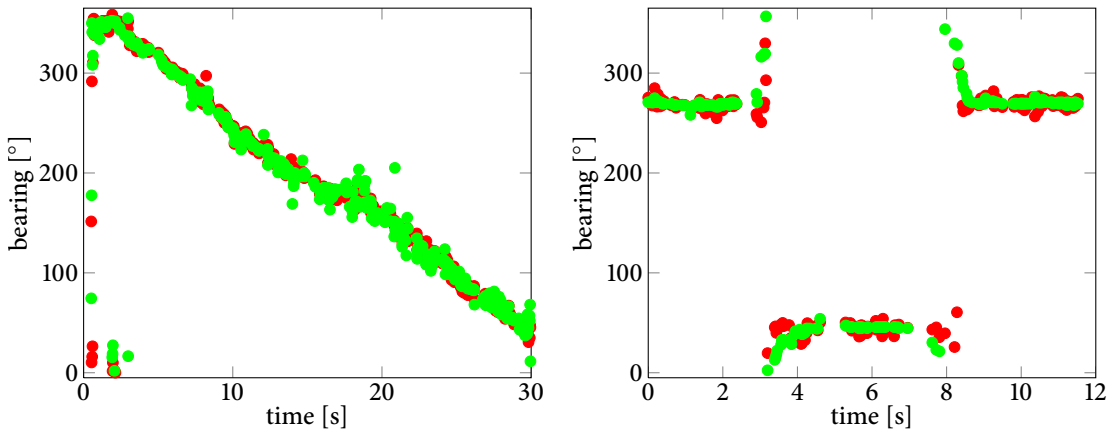
(a) The first trajectory



(b) The second trajectory

Figure 3.13: Bearing estimation for the two simulated trajectories. Gray dots represent sensor measurements, while lines in green and red represent the PF and von Mises mixture filter, respectively. The black line is the true trajectory while blue line is the entropy of the mixture filter.

REAL-WORLD DATA. The data was recorded with a microphone array consisting of four omnidirectional microphones placed on a Pioneer 3DX robot as depicted in Fig 2.2. The recordings were made with sampling frequency $F_s = 48$ kHz and frame length $L = 1024$ samples in a classroom which has dimensions of 7 m × 7 m × 3.2 m, parquet wooden flooring, and one side covered with windows. During the experiments, typical noise conditions were present, like computer noise and air ventilation. Figure 3.14a shows the results of real-world tracking of a single speaker making a full circle around the microphone array, while Fig. 3.14b shows a turn-take scenario. Speakers were, at an approximate distance of 2 m, reading sentences from the IEEE sentence database [145].



(a) Speaker making a full circle

(b) Turn-take scenario

Figure 3.14: Real-world data tracking of speakers with the PF (green) and mixture filter (red). We can notice some outliers due to uniformity of the prior distribution at the initialization and corrupted measurements caused by difficult acoustic conditions (reverberation).

3.6 ACTIVE SPEAKER LOCALIZATION

In this section we present a solution to estimating both the bearing and the range by fusing bearing-only measurement with the known motion of the robot. Although this problem has been studied for few decades, it still receives attention due to emerging new filtering methods. In [20] three different filters were compared for the task, while in [21] various methods for tracking and decentralized sensor fusion were studied, including bearing-only scenarios. In [113], relative localization is performed from a pair of moving microphones, based on a multiple-hypothesis square-root unscented Kalman filter. The filtering scheme uses time delays estimated from the sensed audio signals, together with information on the sensor's velocities to perform a consistent source localization. Results show that the strategy, together with a suitable sensor motion, allows to break front-back ambiguity and get accurate range information.

In this section, active speaker localization is performed with two microphones mounted on a spherical head by particle filtering [73]. The underlying state space equation describing the evolution of the source position in the head frame is defined in both cartesian and polar coordinates. A pseudo-likelihood function proposed in [100] of the source bearing as the measurement model, which captures both the interaural phase difference (IPD) and interaural level difference (ILD) between the binaural signals. Since the pseudo-likelihood has no analytic expression and is only given for a discrete set of candidate bearings, the fitting of circular distributions to the discrete pseudo-likelihood is discussed in order to enhance its resolution for the purpose of estimation. Incidentally, this can give further ground for possible analytical filtering schemes. Two distributions are presented and compared for the task: namely the VM distribution and the wrapped Cauchy (WC) distribution. Furthermore, we compare two particle filtering schemes on experimental data—one using the raw discrete pseudo-likelihood, and the other based on the fitted circular distribution. As aforementioned, both fuse the known head velocities with binaural data in order to infer the speaker location.

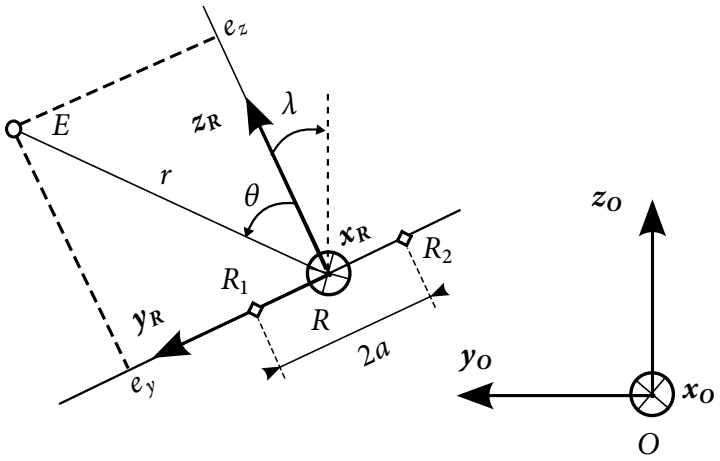


Figure 3.15: Considered localization problem

3.6.1 *Kinematics and state space equation*

A pointwise sound emitter E and a binaural sensor lie on a common plane parallel to the ground. The two receivers equipping the sensor are denoted by R_1 and R_2. A frame $\mathcal{F}_R : (R, \boldsymbol{x}_R, \boldsymbol{y}_R, \boldsymbol{z}_R)$ is rigidly linked to the sensor, with R the midpoint of the line segment $[R_1R_2]$, $\boldsymbol{y}_R$ the vector $\frac{\boldsymbol{RR}_1}{|\boldsymbol{RR}_1|}$ and $\boldsymbol{x}_R$ the downward vertical vector. The frame $\mathcal{F}_E : (E, \boldsymbol{x}_O, \boldsymbol{y}_O, \boldsymbol{z}_O)$ attached to the source is parallel to the world reference frame $\mathcal{F}_O : (O, \boldsymbol{x}_O, \boldsymbol{y}_O, \boldsymbol{z}_O)$, with $\boldsymbol{x}_O = \boldsymbol{x}_R$ (see Fig. 3.15). The source is assumed motionless with respect to the world frame, while the sensor is endowed with two translational and one rotational degrees-of-freedom (velocities v_{Ry}, v_{Rz} of $\mathcal{F}_R$ with respect to $\mathcal{F}_O$ expressed along axes $\boldsymbol{y}_R, \boldsymbol{z}_R$; rotation velocity ω of $\mathcal{F}_R$ with respect to $\mathcal{F}_O$ around $\boldsymbol{x}_O = \boldsymbol{x}_R$). Assuming v_{Ry}, v_{Rz}, ω are known, the aim is to localize the emitter ($\mathcal{F}_E$) with respect to the binaural sensor ($\mathcal{F}_R$) on the basis of the sensed data at R_1, R_2. The audio sensor location with respect to $\mathcal{F}_O$ is not required and the localization of the mobile base is not performed. The relative attitude of $\mathcal{F}_R$ with respect to $\mathcal{F}_E$ can be described, when v_{Ry}, v_{Rz}, ω are zero-order held at the sampling period T_s, by the discrete-time deterministic state space equation [101, 113][2]

$$\boldsymbol{x}_t = F\boldsymbol{x}_{t-1} + G_1\boldsymbol{u}_{1,t}, \text{ with}$$

$$F = \begin{bmatrix} \cos(\omega_{t-1}T_s) & \sin(\omega_{t-1}T_s) & 0 \\ -\sin(\omega_{t-1}T_s) & \cos(\omega_{t-1}T_s) & 0 \\ 0 & 0 & 1 \end{bmatrix}, G_1 = -\begin{bmatrix} \frac{\sin(\omega_{t-1}T_s)}{\omega_{t-1}} & \frac{1-\cos(\omega_{t-1}T_s)}{\omega_{t-1}} & 0 \\ \frac{\cos(\omega_{t-1}T_s)-1}{\omega_{t-1}} & \frac{\sin(\omega_{t-1}T_s)}{\omega_{t-1}} & 0 \\ 0 & 0 & T_s \end{bmatrix},$$

Therein, the state vector $\boldsymbol{x} = [e_y, e_z, \lambda]^{\mathrm{T}}$ gathers the entries e_y and e_z (the $\boldsymbol{y}_R$ and $\boldsymbol{z}_R$ component of E in $\mathcal{F}_R$) and the orientation angle λ. The sensor velocities constituting $\boldsymbol{u}_1 = [v_{Ry}, v_{Rz}, \omega]^{\mathrm{T}}$ are supposed known. When parameterizing the problem in terms of polar coordinates rather than Cartesian, i.e. when using the variables $\theta = \mathrm{atan2}(e_y, e_z)$,

[2] Since the object is assumed to be static the motion model does not entail object's velocities like the Langevin or the constant velocity model

$r = \sqrt{e_y^2 + e_z^2}$, the state space equation comes as

$$r_t = \sqrt{r_{t-1}^2 + \boldsymbol{u}_t^{\mathrm{T}} G^{\mathrm{T}} G \boldsymbol{u}_t + 2 r_{t-1} [\sin\theta_{t-1}, \cos\theta_{t-1}] G^{\mathrm{T}} \boldsymbol{u}_t} \tag{3.49}$$

$$\theta_t = \operatorname{atan2}\left(r_{t-1}\sin(\theta_{t-1} + \omega_{t-1}T_s) + \boldsymbol{g}_1\boldsymbol{u}_t, r_{t-1}\cos(\theta_{t-1} + \omega_{t-1}T_s) + \boldsymbol{g}_2\boldsymbol{u}_t\right)$$

$$\lambda_t = \lambda_{t-1} - \omega_{t-1}T_s, \tag{3.50}$$

with $\boldsymbol{u} = [v_{R_y}, v_{R_z}]^{\mathrm{T}}$, G the square matrix made up with the first two rows and columns of G_1, $\boldsymbol{g}_1$ (resp. $\boldsymbol{g}_2$) the first (resp. second) row of G. To model uncertainty in the relative motion, a random white Gaussian noise of known statistics is added to (3.49).

3.6.2 *Acoustic model, measurement vector, pseudo-likelihood*

Consider first a static world where the sensor is motionless. We assume that the source lies in the farfield (i.e. the source range $r = |\overline{RE}|$ is sufficiently high compared to the microphones interspace $2a$ so that the source wavefronts can be considered as planar in the vicinity of the microphone pair). We model the signals y_1, y_2 monitored at R_1, R_2 in the presence of additive noise as follows

$$\begin{cases} y_1(\tau) = s(\tau) + n_1(\tau) \\ y_2(\tau) = (s * h_\theta)(\tau) + n_2(\tau), \end{cases} \tag{3.51}$$

where the signal s (i.e. the contribution of the emitter at R_1) and the noises n_1, n_2 are real, band-limited, individually and jointly stationary random processes, and $*$ denotes convolution. The deterministic impulse response h_θ between R_1, R_2, is parameterized by θ, and captures free-field propagation of the emitted signal as well as head scattering. H_θ, the Fourier transform of h_θ, is supposed known for every θ within a discrete set of values (say, it has been learned from calibration, or is known theoretically). The process $\boldsymbol{y}(\tau) = [y_1(\tau), y_2(\tau)]^{\mathrm{T}}$ is observed over N adjacent non-overlapping rectangular T/N-width time windows. Denote $\boldsymbol{y}_n$ the observation of $\boldsymbol{y}$ over the n^{th} window. A data vector $\boldsymbol{Z}$ is made up by stacking the values of [101]

$$\boldsymbol{Y}_{n,k} = \sqrt{\frac{N}{T}} \int_{\mathbb{R}} \boldsymbol{y}_n(\tau) e^{-2i\pi k \frac{N}{T}\tau} d\tau, \;\; n = 1, \ldots, N \tag{3.52}$$

at $k = k_1, \ldots, k_B$, the B frequency indexes within the bandwidth of s. $\boldsymbol{Z}$ is hence defined as $\boldsymbol{Z} = [\boldsymbol{Y}_{k_1}^{\mathrm{T}}, \ldots, \boldsymbol{Y}_{k_B}^{\mathrm{T}}]^{\mathrm{T}}$, with $\boldsymbol{Y}_k = [\boldsymbol{Y}_{1,k}^{\mathrm{T}}, \ldots, \boldsymbol{Y}_{N,k}^{\mathrm{T}}]^{\mathrm{T}}$. Assume now that s, n_1, n_2 are zero-mean jointly Gaussian and that n_1, n_2 are identically distributed, uncorrelated with each other and with s. Then, under general mild conditions on the power spectra of s, n_1, n_2 and on H_θ, the maximum likelihood estimate of θ can be obtained, given a sample $\boldsymbol{z}$ of $\boldsymbol{Z}$, by maximizing the following criterion [100, 101], hereafter referred to as the *pseudo log-likelihood function*

$$J(\boldsymbol{z} \,|\, \theta) = c_2 - N \sum_{k=k_1}^{k_B} \left(\ln|P_{\boldsymbol{\theta},k} \hat{C}_k P_{\boldsymbol{\theta},k} + \hat{\sigma}_{\boldsymbol{\theta},k}^2 P_{\boldsymbol{\theta},k}^{\perp}| \right), \tag{3.53}$$

with $c_2 = -2NB(\ln(\pi)+1)$, $\hat{C}_k = \frac{1}{N}\sum_n \boldsymbol{Y}_{n,k}\boldsymbol{Y}_{n,k}^{\dagger}$, $P_{\boldsymbol{\theta},k} = \boldsymbol{V}_{\boldsymbol{\theta},k}(\boldsymbol{V}_{\boldsymbol{\theta},k}^{\dagger}\boldsymbol{V}_{\boldsymbol{\theta},k})^{-1}\boldsymbol{V}_{\boldsymbol{\theta},k}^{\dagger}$, $P_{\boldsymbol{\theta},k}^{\perp} = \mathbb{I}_2 - P_{\boldsymbol{\theta},k}$, $\boldsymbol{V}_{\boldsymbol{\theta},k} = [1, H_{\boldsymbol{\theta},k}]'$, $\hat{\sigma}_{\boldsymbol{\theta},k}^2 = \operatorname{tr}(P_{\boldsymbol{\theta},k}^{\perp}\hat{C}_k)$. Therein, $(\,.\,)^{\dagger}$,$(\,.\,)^{\perp}$, $|\,.\,|$, $\operatorname{tr}(\,.\,)$ respectively stand for Hermitian transpose, orthogonal complement, determinant and trace; $\boldsymbol{y}_{n,k}$ denotes a sample

of $\boldsymbol{Y}_{n,k}$, and the sample covariance matrix $\hat{C}$ is assumed full rank. For details regarding the pseudo log-likelihood function please confer [100, 101].

Consider now a real world where the sensor moves and where the source signal and environment noise are possibly nonstationary. All the fundamental hypotheses leading to (3.51)–(3.52)–(3.53) are consequently violated. Nevertheless, the problem can still be handled if, at each process time index t, the data vector $\boldsymbol{z}_t$ is made up from audio data collected over a time window matched to t, sufficiently short so that, along this window, the sensor motion is negligible and the recorded signals can be regarded as finite-time samples of stationary processes. Hence, at each time index t, the pseudo likelihood of θ_t with respect to $\boldsymbol{z}_t$, denoted $p(\boldsymbol{z}_t \,|\, \theta_t)$, can be output and will henceforth be used in a Bayesian filtering scheme in Section 3.6.5. Importantly, $p(\boldsymbol{z}_t \,|\, \theta_t)$ has in the general case no analytic expression. Its numerical values are just given for a discrete set of tested azimuths and are obtained as a result of the maximization of (3.53). This precludes the use of Bayesian filtering schemes requiring an analytic form of the pseudo likelihood, e.g. Gaussian mixture filters, unless an analytic function is fitted to the discrete values. Alternatively, with particle filters, the pseudo likelihood in its discrete form can be utilized as a sensor model. However, low bearing resolution can affect the particle filter performance and consistency, and it may be useful to fit some distribution to the discrete pseudo likelihood. Section 3.6.3 is thus dedicated to the fitting of von Mises and wrapped Cauchy mixtures models to the discrete pseudo likelihood.

3.6.3 *Fitting the circular distributions*

In this section we present two solutions to fitting the pseudo likelihood function, namely fitting with the VM distribution and with the WC distribution. The fact that circular distributions intrinsically take noneucledian properties of the angular data into account proves useful in the fitting problem since a circular distribution with mean close to 2π continues contributing significantly to points larger than 0. This will be of great importance of the optimization step that will be presented shortly. Furthermore, here we do not require the component weights to sum up to one, since the pseudo likelihood function itself is not a valid probability distribution.

Since the VM distribution was introduced in Section 2.3.1, in this section we shall focus on the WC distribution. The WC distribution is a distribution that is wrapped on a circle. Given a distribution on the line we can wrap it around the circumference of a circle with unit radius. If a random variable θ is defined on a line, then the corresponding random variable of the wrapped distribution is $\theta_w = \theta \pmod{2\pi}$. Furthermore, if θ has a PDF p, then the corresponding wrapped pdf p_w is defined as $p_w(\theta) = \sum_{k=-\infty}^{k=\infty} p(\theta + 2k\pi)$ [46], from which we can note practical issues when dealing with the infinite number of terms in the summation. However, it can be shown that the Cauchy distribution on the line

$$p(\theta; \mu, a) = \frac{1}{\pi} \frac{a}{a^2 + (\theta - \mu)^2}, \tag{3.54}$$

where $-\infty < \mu < \infty$ and $a > 0$ has an interesting property that its wrapped counterpart, due

to certain geometric series expansion property, reduces to [46]

$$p(\theta;\mu,\rho) = \frac{1}{2\pi}\frac{1-\rho^2}{1+\rho^2-2\rho\cos(\theta-\mu)}, \tag{3.55}$$

where μ is the mean direction and ρ is called the mean resultant length. When $\rho \to 0$ the WC tends to uniform distribution, while if $\rho \to \infty$ the distributions becomes concentrated at point μ.

Naturally, the pseudo likelihood function will suffer from front-back ambiguity since we utilize a binaural setup. Hence, our sensor model will contain at least two distinct modes on the interval 0 to 2π and for this reason we chose to model the likelihood as a mixture of distributions. If we denote with $\mathcal{X}$ a set of distributions parameters, then an N component mixture can be defined as $p(\theta;\mathcal{X}) = \sum_{i=1}^{N} w_i p(\theta;\mathcal{X}_i)$, where the set $\mathcal{X}$ consists of $\{\mu_i,\kappa_i\}_{i=1}^{N}$ for the VM distribution and $\{\mu_i,\rho_i\}_{i=1}^{N}$ for the WC distribution.

3.6.4 *Evaluation of the fitting performance*

The fitting of a mixture of distributions to the pseudo likelihood function, denoted as $\hat{p}(\theta)$, comes down to solving the following optimization problem

$$\min_{w,\mathcal{X}} \quad \sum_{i=1}^{N} \left(w_i p(\theta;\mathcal{X}_i) - \hat{p}(\theta)\right)^2$$

with the constraints $0 \le w_i \le 1$ and $0 \le \mathcal{X}_i \le \mathcal{B}$ for $i = 1,\dots,N$, and where the upper bound $\mathcal{B}$ depends on the parameter and the distribution. For both distributions the upper bound of the mean directions μ is $\mathcal{B} = 2\pi$, while for the VM distribution the upper bound was $\mathcal{B} = 2000$ for the concentration parameter, and for the WC distribution $\mathcal{B} = 1$ for the mean resultant length.

Concerning the number of the mixture components, all the results were obtained with $N = 4$. Initial conditions for the mean directions were determined by searching recursively for N most dominant peaks in the vein of [26] where the authors searched for the number of active speakers. Once the dominant peak is found, an area around it is removed and the search continues until the predetermined number of modes is found. Since in the pseudo likelihood function we expect two peaks to be dominant we set the initial conditions for the first two dominant peaks to be $\kappa = 1500$ or $\rho = 0.9$, while for the rest we set $\kappa = 10$ or $\rho = 0.1$. The weights are initially set to $w_i = 0.5, \forall i$.

In Fig. 3.16 we can see the result of fitting[3] for a single relatively difficult frame when the speaker was close to the end-fire position of the array and the two dominant modes started overlapping. Empirically we noticed that this is the more difficult case for the WC distribution and that often the two distinct nodes tend to be fitted with a single component in between them. Overall, the whole dataset consisted of four experiments with a talking speaker as the source. The average RMSE of fitting for the speaker scenario was $1.6 \cdot 10^{-3}$ for the VM mixture and $3.7 \cdot 10^{-3}$ for the WC mixture, respectively. Given that, for the rest of the section we have chosen to work with the VM mixture since it provided better fitting in terms of the average RMSE.

[3] The fitting was solved by non-linear least squares method implemented in Matlab

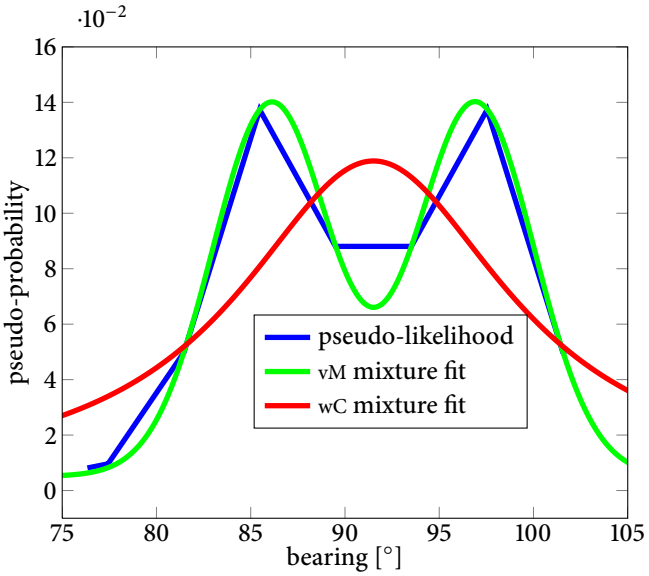


Figure 3.16: Fitting the pseudo likelihood for a single frame with a VM and a WC mixture

3.6.5 *Speaker localization in 2D*

In this section we utilize the PF for estimating the location of the speaker. Note that here the state of the speaker is different compared to previous sections. Whereas θ_t denoted the speaker state—the bearing—in this section $\boldsymbol{x}_t$ denotes the speaker state—the (e_y, e_z) coordinates with respect to $\mathcal{F}_R$.

Recall that $\{\boldsymbol{x}^p, w^p\}_{p=1}^P$ depicts the random measure that characterizes the posterior state PDF $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t})$, where each particle in the set $\{\boldsymbol{x}^p\}_{p=1}^P$ is associated to the respective weight in $\{w^p\}_{p=1}^P$. The weights satisfy $\sum_p w^p = 1$, so that $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t})$ can be approximated as [73, 74]

$$p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}) \approx \sum_{p=1}^{P} w_t^p \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^p), \tag{3.56}$$

with $\delta(.)$ the Dirac delta measure. Since for any recursive particle filter, the significant weights tend to concentrate on a limited set of particles after few iterations, a resampling step is inserted, which consists in turning $\{\boldsymbol{x}_t^p, w_t^p\}_{p=1}^P$ into the equivalent evenly weighted set $\{\boldsymbol{x}_t^{*p}, \frac{1}{P}\}_{p=1}^P$ by independently sampling (with replacement) $\boldsymbol{x}_t^{*p}$ according to $P(\boldsymbol{x}_t^{*p} = \boldsymbol{x}_t^p) = w_t^p$. In the SIR scheme [73] the importance function matches the prior dynamics $p(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1})$, i.e. each particle $\boldsymbol{x}_t^p$ at time t is drawn from its predecessor $\boldsymbol{x}_{t-1}^p$ at time $t-1$ according to the proposal density $\boldsymbol{x}_t^p \sim p(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1}^p)$. Then, its weight is updated by evaluating its likelihood $p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t^p)$ prior to setting

$$w_t^p \propto w_{t-1}^p p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t^p), \tag{3.57}$$

where $p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t)$ represents the sensor model, i.e. the fitted VM mixture:

$$p(\boldsymbol{z}_t \,|\, \boldsymbol{x}_t) = \sum_{i=1}^{N} w_i \frac{1}{2\pi I_0(\kappa_i)} \exp\left[\kappa_i \cos(\boldsymbol{x}_t - \boldsymbol{z}_{t,i})\right]. \tag{3.58}$$

Once the random measure approximating the posterior PDF of the state is computed, the

posterior mean and posterior covariance can be estimated via

$$\begin{aligned}\hat{\boldsymbol{x}}_t &= E[\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}] \approx \sum_{p=1}^{P} w_t^p \boldsymbol{x}_t^p \\ \hat{\boldsymbol{P}}_t &= E[(\boldsymbol{x}_t - E[\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}])(\boldsymbol{x}_t - E[\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}])^{\mathrm{T}} \,|\, \boldsymbol{z}_{1:t}] \\ &\approx \sum_{p=1}^{P} w_t^p (\boldsymbol{x}_t^p - \hat{\boldsymbol{x}}_t)(\boldsymbol{x}_t^p - \hat{\boldsymbol{x}}_t)^{\mathrm{T}}.\end{aligned} \tag{3.59}$$

This is needed since we will be analyzing the consistency of the filter from multiple runs. To avoid a loss of diversity in the particle cloud, the resampling step was applied only when the number of effective weights $P_{\text{eff}} = 1/\sum_p (w^p)^2$ was less than a given threshold, e.g. 33% of the total number of particles P.

Consequently, particle filtering can be implemented even if a closed-form measurement model is not available, in that the particle likelihoods just need to be evaluated. In our case, the sensor model comes as the pseudo likelihood digitized with a resolution of 4°. However, we assert that the fitting utilized constitutes a form of interpolation which yields better resolution. So, we henceforth compare the performance of the PF which directly utilizes the discrete pseudo likelihood against the PF utilizing the fitted VM mixture. Importantly, fitting with a VM mixture would be a prerequisite if the tracking was performed in the vein of [56].

3.6.6 *Experiments*

Experiments were conducted at the premises of the University of Pierre and Marie Curie by Alban Portello, Sylvain Argentieri and Bruno Gas. The room was acoustically prepared, equipped with 3D pyramidal pattern studio foams placed on the roof and on the walls. Two surface microphones were mounted at the antipodes of a 8.9 cm radius plastic rigid sphere, itself place on a tripod. The two microphones outputs were synchronously acquired

Figure 3.17: Experimental setup: plastic sphere and speaker tripods in the acoustic room. Infrared cameras were measuring the ground-true positions.

at 44.1 kHz. The sphere tripod was moved manually with a wheeled cart while the source, a loudspeaker placed at the same height as the microphones, was emitting voice recordings from a French radio programme. The true source and sensor positions were acquired at 200 Hz with a motion capture system, providing a less than 1 mm position error. For that purpose, small infrared active markers were placed on the sphere and the loudspeaker, and their signals were beamed to three infrared camera units placed at the corners of the room.

For the considered case of a rigid sphere, H_θ is shown to have the following analytic expression [101, 146]

$$H_\theta(f) = \frac{\psi_{\frac{\pi}{2}+\theta}(f)}{\psi_{-\frac{\pi}{2}-\theta}(f)}, \text{ with} \tag{3.60}$$

$$\psi_\alpha(f) \triangleq \frac{1}{\left(\frac{2\pi f a}{c}\right)^2} \sum_{m=1}^{\infty} \frac{(-i)^{m-1}(2m+1)P_m(\cos\alpha)}{h'_m\left(\frac{2\pi f a}{c}\right)}.$$

Therein, ψ_β is the normalized head related transfer function to the microphone at angle β—with respect to boresight—on the sphere, where α stands for the angle between the source bearing and the direction to the considered microphone, P_m is the Legendre polynomial of degree m, h_m is the mth-order spherical Hankel function and h'_m is its first derivative. This expression was thus used in the pseudo likelihood computation. In practice, the infinite sum in (3.60) is approximated by a finite sum, the minimum order required to make the approximation reasonable depending on the maximum frequency considered. To avoid cumbersome computation during localization, H_θ was precomputed and stored offline for a discrete set of bearings.

In order to assess the performance of the PFs, we ran 50 Monte-Carlo runs on the sensed binaural data using either the discrete pseudo likelihood or the VM fitted pseudo likelihood. The runs were performed on four scenarios with different trajectories of the sensor, out of which one scenario included an intermittent sound source. In Fig. 3.18 we can see the results of range estimation for the four cases, while Fig. 3.19 shows the estimation of the bearing. By analyzing the results we can see that on average the PF with the VM fitted likelihood gave smaller error in terms of the range estimation although the performance in the bearing was similar for both PFs. The explanation lies in the fact that estimating the range from bearing-only measurements benefited from having an analytical likelihood compared to the 4° resolution of the discrete pseudo likelihood.

Then, for each entry of the posterior mean output by the filter, a minimum-width confidence interval for a moment matched Gaussian distribution of the estimation error was drawn (from the posterior particle set) which should approximately enclose the genuine hidden state vector with 99% probability. By analyzing the obtained plots concerning the range estimation error, we can see that the present implementation of the PF was not consistent over all the runs, since the true range is outside of the filter's $\pm 3\sigma$ interval calculated from the estimated covariance matrix and that bias is present, which would indicate that the particle filter diverged at several instances of Monte-Carlo (MC) runs. This problem could be alleviated by utilizing a higher number of particles and/or more elaborate initialization and maneuvering strategies (cf. [147] for a deeper study of the problem).

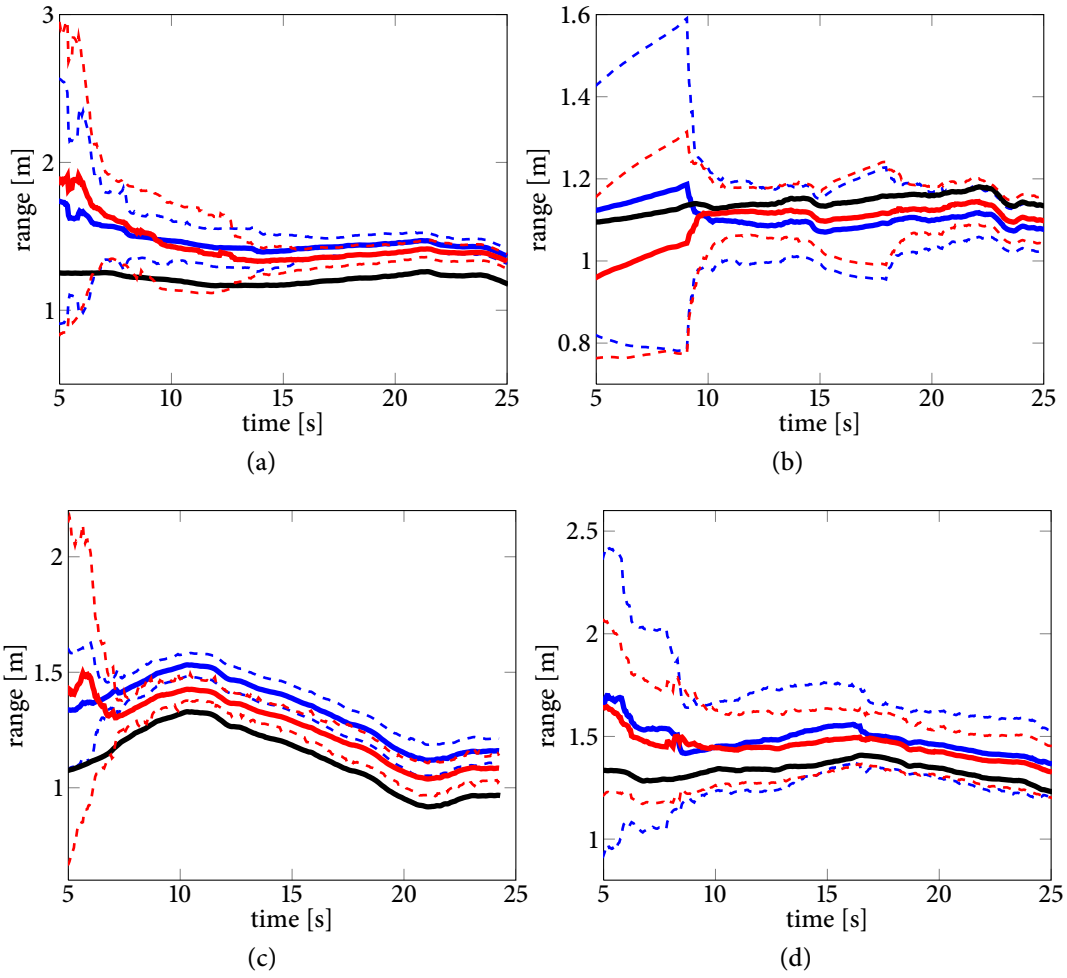


Figure 3.18: Mean value of range estimates (solid) and pertaining three standard deviations (dashed) of 50 Monte-Carlo runs of the PF with pseudo likelihood (blue), VM fitted pseudo likelihood (red) and true range (black) for four different data sets

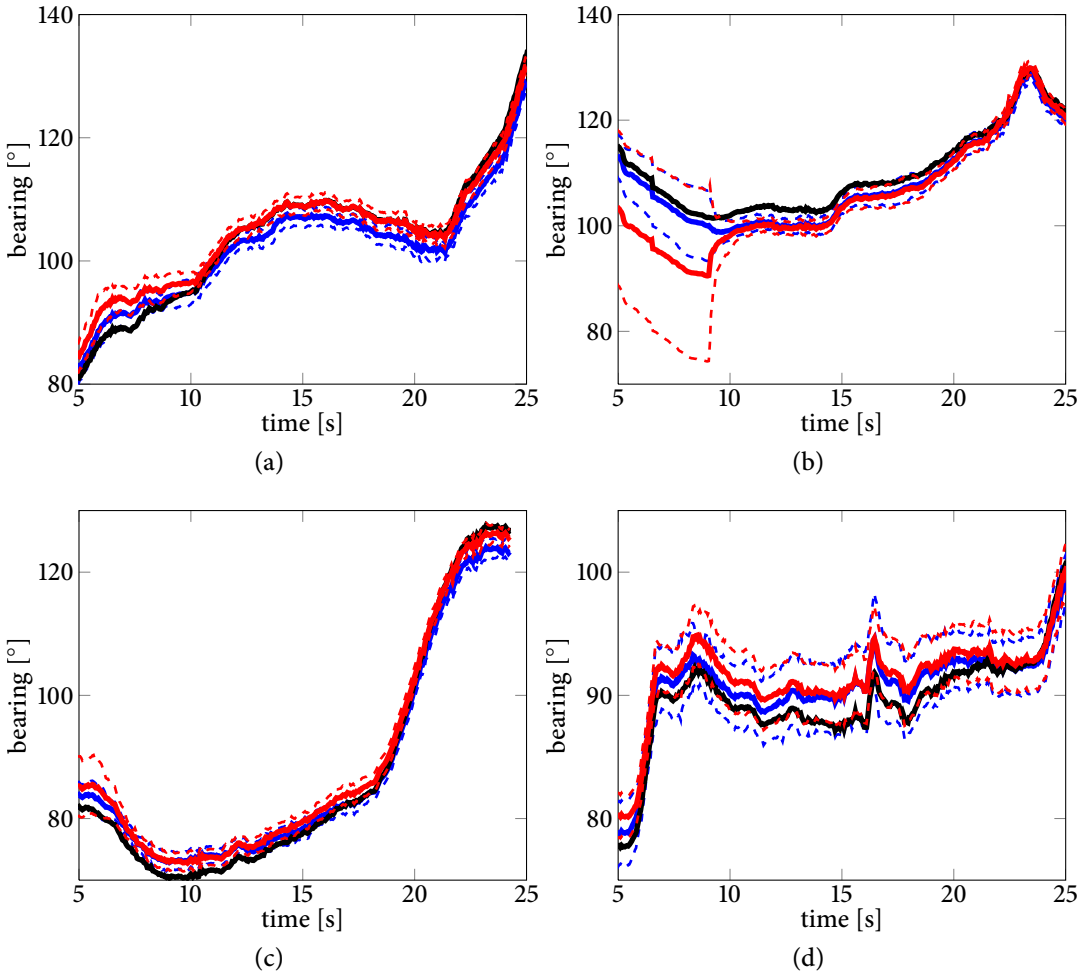


Figure 3.19: Mean value of bearing estimates (solid) and pertaining three standard deviations (dashed) of 50 Monte-Carlo runs of the PF with pseudo likelihood (blue), VM fitted pseudo likelihood (red) and true bearing (black) for four different data sets

3.7 SUMMARY

In this chapter we have first presented a novel approach to speaker localization and microphone array measurement modeling. Novelty of the proposed approach is in the method based on a convex combination von Mises distributions the for direction of arrival analysis and for derivation of an adequate bearing estimation method. The algorithm solves the front-back ambiguity, a unique bearing value is calculated from the posterior distribution, and the PF is utilized for the tracking task. Moreover, a voice activity detector can easily be integrated to the time difference of arrival estimation, and operation under adverse noisy conditions is guaranteed up to the performance of the voice activity detector itself. The algorithm accuracy and precision was tested in real-time with a reliable ground truth method based on leg-tracking with a laser range finder.

Furthermore, two most common microphone array geometries were meticulously analyzed and compared theoretically based on error sensitivity to time difference of arrival estimation and the robustness to microphone occlusion. The analysis and experiments showed square array having several advantages over the Y array configuration, but from a practical point of view these two configurations have similar performances.

Ensuingly, we have proposed a method for speaker tracking that was based on von Mises mixtures. The method included calculating the convolution of two von Mises PDFs for the prediction stage, the product of two von Mises PDFs for the update stage, the component reduction of the mixture to prevent exponential growth of the component number, and Rényi quadratic entropy for uncertainty tracking of such a multimodal state representation. The proposed approach was tested and compared to the PF in a speaker tracking scenario on a synthetic data experiment and on real-world recordings. The results supported the proposed approach and showed similar performance to the particle filter, with the benefit of smaller number of parameters for state representation and complete support on the state space.

Although the algorithm was presented on the problem of speaker tracking with a microphone array, the potential field of interest is by no means limited to this application. The proposed approach can be utilized in any tracking scenario which involves bearing-only measurements. Furthermore, the chapter highlights the merits of using a von Mises distribution for directional data, which does not receive that much attention due to pervasive use of the Gaussian distribution. One of the potential expected practical significances lies in systems where the communication bandwidth is limited, e.g. in decentralized architectures when different robots need to communicate the a posteriori distributions.

At the end of the chapter, a solution for the problem of active speaker localization with a head mounted binaural microphone sensor was presented. The solution was based on calculating a discrete pseudo likelihood function in speaker bearing based on the geometrical properties of the spherical head. The resulting likelihood was fitted with a mixture of circular distributions, namely the VM and WC distributions, whose comparison showed better results for the case of the VM distribution. A PF was utilized with the direct and VM fitted pseudo likelihood in order to estimate the location of the speaker. We performed an experimental evaluation on four different data sets with accurate ground-truth, due to an active motion capture system, whose analysis showed that on average from 50 Monte Carlo

runs both algorithms localized the speaker successfully, while the estimate with the VM fitted likelihood showed better accuracy in range and equal or better accuracy in bearing. However, a careful analysis revealed that at times the algorithms were inconsistent deviation-wise and that robust variants of the PF could be utilized, but which were eschewed in the experiments in order to guarantee the true posterior statistics.

4

Voice Activity Detection

The chapter presents a novel approach for voice activity detection. The main idea behind the presented approach is to use, next to the likelihood ratio of a statistical model-based voice activity detector, a set of informative distinct features in order to, via a supervised learning algorithm, enhance the detection performance. Firstly, we present three different statistical model-based voice activity detection algorithms in an unifying and consistent manner, by incorporating noise spectrum and the a priori signal-to-noise ratio estimation to their respective frameworks. Furthermore, the likelihood ratio of the best performing statistical model-based detector together with 70 other various features is meticulously analyzed with an input variable selection algorithm based on partial mutual information. The resulting analysis produced a 13 element reduced input vector which when compared to the full input vector did not undermine the detector performance. The evaluation is performed on a speech corpus consisting of recordings made by six different speakers, which were corrupted with three different types of noises and noise levels. In the end, three different supervised learning algorithms are tested for the task, namely, support vector machine, Boost, and artificial neural networks. The experimental analysis was performed by 10-fold cross-validation due to which threshold averaged receiver operating characteristics curves were constructed. Also, the area under the curve score and Matthew's correlation coefficient are calculated for both the three supervised learning classifiers and the statistical model-based voice activity detector. The results show that the classifier with the reduced input vector significantly outperformes the standalone detector based on the likelihood ratio, and that among the three classifiers, Boost showed the most consistent performance.

4.1 Introduction

Voice activity detection is a technique in speech processing by which presence of speech is detected in a given signal frame. This problem can be seen as a dual hypothesis problem, where a signal frame is classified as either containing speech or containing noise. In a voice activity detector (VAD), the absence of speech usually presumes presence of noise only. This system is not only of great importance for many applications, like mobile telephony, internet telephony, hearing aid devices, but also for robotics if speech oriented systems are utilized like speaker localization, speech and speaker recognition. For most of the stated research problems, it is indispensable to save on bandwidth resources by coding noise with

significantly less bits, while for others it is mandatory to completely ignore frames with noise.

A VAD must provide a robust and reliable decision procedure in varying acoustical conditions. This task gets quite formidable with the varying level and type of background noise. Approaches to voice activity detection mostly differ in the type of the extracted features and in the decision models used to reach a speech/non-speech decision based on those features. A lot of attention was given to statistical model-based VADs, in which certain probabilistic properties are assumed on the coefficients of the DFT. For an example, in [148] they are assumed to have Gaussian distribution and this approach was further developed in [149–155]. Furthermore, special attention was given to derivation of various noise robust features and decision rules in [156–158]. Concerning supervised learning approaches, they have been utilized in various sound processing scenarios, e.g. music classification [159], general audio signal classification (music, news, sports etc.) [160], speech inteligibility quantification [161] etc. Supervised learning based voice activity detection approaches have so far been mostly focused on applying support vector machine (SVM) by treating as features: a priori SNR, a posteriori SNR and/or statistical model-based likelihood ratio [162, 163], mel frequency cepstral coefficients (MFCCs) [164], sub-band and long-term SNR [165, 166], or features used in the standard G.729B [167, 168]. Furthermore, a recent work [169] presented a novel unsupervised learning approach called support-vector-regression-based maximum margin clustering which was also tested in a voice activity detection scenario and showed comparable performance to supervised approach based on support vector machine method.

Methods presented in this chapter survey the statistical model-based VADs and supervised learning approaches to VAD and builds on upon the aforementioned related works with the following main contributions. Three statistical model-based VADs are compared, namely detectors based on the Gaussian distribution (GD), generalized Gaussian distribution (GGD) and Rayleigh and Rice distribution (RRD) (for which we derive the likelihood ratio (LR)). Subsequently, a method for input variable analysis based on partial mutual information algorithm in the context of voice activity detection is introduced. This method systematically classifies features on those that should be included and those that could be omitted from the input set, which we find extremely important when extending input spaces of supervised learning algorithms. Moreover, we extend the input space with distinct features under the hypothesis (which is tested) that this will improve the performance of VADs. While most of the features in the related works are variants on the SNR estimation (a priori, a posteriori, predicted, sub-band and long-term), with two exceptions—one which used only MFCC [164] and other which is based on features from G.729B [167], in the chapter we extended this feature space by using information from the SNR estimation in the form of a statistical-based LR by modeling the distribution of the spectral envelope, along with several distinct features like magnitudes of some of the DFT coefficients, spectral flux, spectral centroid and bandwidth, power-normalized cepstral coefficients, MFCCs etc. Furthermore, for the classification task we present a systematic quantitative evaluation of the following three supervised learning algorithms: Boost, artificial neural networks (ANNs) and SVM, while all the related work papers on VAD utilize only SVM. The algorithms were tested and compared under varying noise conditions, namely three types of noises and three different SNRs, and showed similar performance with a slight advantage in the direction of the Boost

classifier. Some results and novelties in this chapter have been proposed and presented in [170], where the RRD VAD is introduced and its comparison to GD and GGD based VADs is performed, and in [171] where the partial mutual information input variable analysis is performed and the supervised learning approaches are proposed and analyzed.

Although a detector can be considered as a binary classifier, for clarity throughout the thesis we use the term detector to denote the statistical model-based detector based on the likelihood ratio, while the term classifier or supervised learning based VAD denotes the SVM, Boost and ANN classifiers.

4.2 STATISTICAL MODEL-BASED DETECTORS

These VADs rely on statistical modeling of the DFT coefficients. All the statistical model-based VADs assume a two hypotheses scenario. Since speech is degraded with uncorrelated additive noise, the two hypotheses are as follows

$$\begin{aligned} H_0 &: \text{speech absent} \Rightarrow \mathbf{X} = \mathbf{N} \\ H_1 &: \text{speech present} \Rightarrow \mathbf{X} = \mathbf{N} + \mathbf{S}, \end{aligned} \tag{4.1}$$

where the DFT coefficients of a K-point DFT of the noisy speech, noise, and clean speech are denoted as $\mathbf{X} = [X_0, X_1, \ldots, X_{K-1}]^{\mathrm{T}}$, $\mathbf{N} = [N_0, N_1, \ldots, N_{K-1}]^{\mathrm{T}}$ and $\mathbf{S} = [S_0, S_1, \ldots, S_{K-1}]^{\mathrm{T}}$, respectively.

The form of the PDF of $\mathbf{X}$ conditioned on the hypotheses, i.e. $p(\mathbf{X}\,|\,H_0)$ and $p(\mathbf{X}\,|\,H_1)$, depends on the distribution used to model each DFT coefficient. After the PDFs $p(\mathbf{X}\,|\,H_0)$ and $p(\mathbf{X}\,|\,H_1)$ are determined, usually a likelihood ratio on all the DFT coefficient indices k is calculated

$$\Lambda_k = \frac{p(X_k\,|\,H_1)}{p(X_k\,|\,H_0)}, \tag{4.2}$$

where Λ_k becomes a vector of length K. This information is then used to calculate geometric mean which is then compared to a certain threshold in order to reach a final decision in favor of either the hypothesis H_0 or H_1

$$\log \Lambda = \frac{1}{K} \sum_{k=1}^{K} \log \Lambda_k \underset{H_0}{\overset{H_1}{\gtrless}} \eta. \tag{4.3}$$

4.2.1 *Gaussian distribution*

This VAD was first proposed by [148], where the DFT coefficients are asymptotically independent and zero-mean complex Gaussian random variables. Let us look at the DFT of the clean speech signal. In the complex Gaussian speech model, both the real and the imaginary parts of the DFT, $S_k = S_{R,k} + jS_{I,k}$, are independent zero-mean Gaussian random variables, each with a variance of $\lambda_{s,k}/2$. The PDFs of the coefficients are

$$p(S_{R,k}) = \frac{1}{\sqrt{\pi \lambda_{s,k}}} \exp\left\{ -\frac{S_{R,k}^2}{\lambda_{s,k}} \right\} \tag{4.4}$$

$$p(S_{I,k}) = \frac{1}{\sqrt{\pi \lambda_{s,k}}} \exp\left\{ -\frac{S_{I,k}^2}{\lambda_{s,k}} \right\}. \tag{4.5}$$

Since real and imaginary coefficients are independent, joint PDF can be written in the following form

$$p(S_k) = p(S_{R,k})p(S_{I,k}) = \frac{1}{\pi\lambda_{s,k}} \exp\left(-\frac{S_{R,k}^2 + S_{I,k}^2}{\lambda_{s,k}}\right) = \frac{1}{\pi\lambda_{s,k}} \exp\left(-\frac{|S_k|^2}{\lambda_{s,k}}\right). \tag{4.6}$$

Similar derivation can be done for the PDF of the noise coefficients.

When both speech and noise are present, we have for each coefficient a sum of independent Gaussian variables, thus resulting with a PDF of variance $\lambda_{x,k} = \lambda_{n,k} + \lambda_{s,k}$. Hence, the conditional PDFs of X_k on hypotheses H_0 and H_1 are as follows:

$$p(X_k \,|\, H_0) = \frac{1}{\pi\lambda_{n,k}} \exp\left\{-\frac{|X_k|^2}{\lambda_{n,k}}\right\} \tag{4.7}$$

$$p(X_k \,|\, H_1) = \frac{1}{\pi(\lambda_{n,k} + \lambda_{s,k})} \exp\left\{-\frac{|X_k|^2}{\lambda_{n,k} + \lambda_{s,k}}\right\}. \tag{4.8}$$

Under the Gaussian distribution model, the LR is simply calculated as ratio of (4.8) and (4.7)

$$\Lambda_k^{\mathrm{GD}} = \frac{p(X_k \,|\, H_1)}{p(X_k \,|\, H_0)} = \frac{1}{1+\xi_k} \exp\left\{\frac{\gamma_k \xi_k}{1+\xi_k}\right\}, \tag{4.9}$$

where $\xi_k = \lambda_{s,k}/\lambda_{n,k}$ is the a priori SNR, and $\gamma_k = |X_k|^2/\lambda_{n,k}$ is the a posteriori SNR. The algorithms for estimation of these values are presented in Section 4.2.4.

4.2.2 *Generalized Gaussian distribution*

In [151] statistical model-based VAD was improved by incorporating a complex Laplacian model. The analysis in the latter paper showed that the Laplacian provides a better model of the distribution of noisy speech spectra than the Gaussian model. Furthermore, VAD based on GGD, which includes the Gaussian and Laplacian model as special cases, was proposed in [152], where it was also experimentally verified that VAD based on GGD outperforms the VAD based on the Laplacian model. Following the same train of thought as in Section 4.2.1, joint GGD of the DFT coefficients for clean speech signal is given by

$$p(S_k) = \frac{\nu^2 \alpha^2(\nu)}{4\lambda_{s,k}\Gamma^2(1/\nu)} \cdot \exp\left\{-\alpha^{\nu}(\nu)\left[\left|\frac{S_{R,k}}{\sqrt{\lambda_{s,k}}}\right|^{\nu} + \left|\frac{S_{I,k}}{\sqrt{\lambda_{s,k}}}\right|^{\nu}\right]\right\}, \tag{4.10}$$

with

$$\alpha(\nu) = \sqrt{\frac{\Gamma(3/\nu)}{\Gamma(1/\nu)}}, \tag{4.11}$$

where $\Gamma(\cdot)$ denotes the Gamma function, and ν denotes parameter controlling the distribution shape. For $\nu = 1$ and $\nu = 2$ the GGD becomes the Laplacian and Gaussian density, respectively.

The shape parameter ν needs to be continuously estimated. By letting m_1 and m_2 be the first and the second moment of $|X_k|$ (cf. [152]), ν can be estimated by solving the following equation

$$\hat{\nu} = F^{-1}\left(\frac{m_1}{m_2}\right), \tag{4.12}$$

where

$$F(x) = \frac{\Gamma(2/x)}{\sqrt{\Gamma(1/x)\Gamma(3/x)}}. \tag{4.13}$$

The (4.12) is the inverse of (4.13) and can be solved by precomputing a lookup table.

From the previous discussion we can write the distribution of X_k conditioned on the hypotheses H_0 and H_1 as follows

$$p(X_k \,|\, H_0) = \frac{\nu_{n,k}^2 \alpha^2(\nu_{n,k})}{4\lambda_{n,k}\Gamma^2(1/\nu_{n,k})} \exp\left\{ - \alpha^{\nu_{n,k}}(\nu_{n,k}) \cdot \left[\left| \frac{X_{R,k}}{\sqrt{\lambda_{n,k}}} \right|^{\nu_{n,k}} + \left| \frac{X_{I,k}}{\sqrt{\lambda_{n,k}}} \right|^{\nu_{n,k}} \right] \right\} \tag{4.14}$$

$$p(X_k \,|\, H_1) = \frac{\nu_{s,k}^2 \alpha^2(\nu_{s,k})}{4(\lambda_{s,k}+\lambda_{n,k})\Gamma^2(1/\nu_{s,k})} \exp\left\{ - \alpha^{\nu_{n,k}}(\nu_{n,k}) \right. \tag{4.15}$$

$$\left. \cdot \left[\left| \frac{X_{R,k}}{\sqrt{\lambda_{s,k}+\lambda_{n,k}}} \right|^{\nu_{n,k}} + \left| \frac{X_{I,k}}{\sqrt{\lambda_{s,k}+\lambda_{n,k}}} \right|^{\nu_{n,k}} \right] \right\}, \tag{4.16}$$

where $\nu_{n,k}$ and $\nu_{s,k}$ are shape parameters related to H_0 and H_1 of noisy speech on frequency bin k, respectively. In order to compute these parameters, the corresponding $(m_{1,k}^n, m_{2,k}^n)$ and $(m_{1,k}^s, m_{2,k}^s)$ are calculated recursively from $|X_k|$ as proposed in [152].

Finally, we can write the LR for the GGD model

$$\begin{aligned} \Lambda_k^{\mathrm{GGD}} = \frac{1}{1+\xi_k} \cdot \frac{\nu_{s,k}^2 \alpha^2(\nu_{s,k})\Gamma^2(1/\nu_{n,k})}{\nu_{n,k}^2 \alpha^2(\nu_{n,k})\Gamma^2(1/\nu_{s,k})} \exp\left\{ - \alpha^{\nu_{s,k}}(\nu_{s,k}) \left[\frac{|X_{R,k}|^{\nu_{s,k}} + |X_{I,k}|^{\nu_{s,k}}}{\left(\sqrt{\lambda_{n,k}(1+\xi_k)}\right)^{\nu_{s,k}}} \right] \right. \\ \left. + \alpha^{\nu_{n,k}}(\nu_{n,k}) \left[\frac{|X_{R,k}|^{\nu_{n,k}} + |X_{I,k}|^{\nu_{n,k}}}{\left(\sqrt{\lambda_{n,k}}\right)^{\nu_{n,k}}} \right] \right\}. \end{aligned} \tag{4.17}$$

4.2.3 *Rayleigh and Rice distribution*

In the approach proposed by [172], derived from [173], the DFT coefficients are still modelled as having a Gaussian distribution, but instead of using their joint distribution, the distribution of the signal envelope is used. The envelope of a signal, $|X_k| = \sqrt{X_{R,k}^2 + X_{I,k}^2}$, is actually the euclidean norm of the real and imaginary coefficients. Therefore, instead of looking at the distribution of the coefficients, the distribution of the signal envelope is analysed.

Under hypothesis H_0 the signal is only noise, which means that the DFT coefficients are both independent, zero-mean Gaussian variables with variance $\lambda_{n,k}/2 = \mathrm{E}[|N_k|^2]$. Under that assumption, the PDF of the euclidean distance of such DFT coefficients is a Rayleigh distribution

$$p(X_k \,|\, H_0) = \frac{2|X_k|}{\lambda_{n,k}} \exp\left\{ -\frac{|X_k|^2}{\lambda_{n,k}} \right\}. \tag{4.18}$$

Under hypothesis H_1, the envelope is the euclidean norm of two independent, non-zero-mean Gaussian variables. Such PDF is a Rician

$$\begin{aligned} p(X_k \,|\, H_1) &= \frac{2|X_k|}{\lambda_{n,k}} \exp\left\{-\frac{1}{\lambda_{n,k}}\left(|X_k|^2 + |A_k|^2\right)\right\} \cdot I_0\left\{\frac{2|A_k||X_k|}{\lambda_{n,k}}\right\} \\ &= \frac{2|X_k|}{\lambda_{n,k}} \exp\left\{-\frac{|X_k|^2}{\lambda_{n,k}} - \xi_k\right\} \cdot I_0\left\{2\sqrt{\xi_k \frac{|X_k|^2}{\lambda_{n,k}}}\right\}, \end{aligned} \tag{4.19}$$

where A_k is the amplitude of the clean speech spectrum, $\xi_k = |A_k|^2/\lambda_{n,k}$ is the a priori SNR and $I_0(\cdot)$ is the modified Bessel function of the first kind and order zero. In [172] this VAD was implemented by calculating the a posteriori probability $p(H_1 \,|\, X_k)$ of voice activity from (4.18) and (4.19) via the Bayes formula. Since in this thesis the a priori SNR estimation, presented in Section 4.2.4, for all frequency bins is implemented, we are proposing the LR instead of the a posteriori probability $p(H_1|X_k)$.

Finally, we derive the LR for RRD model

$$\Lambda_k^{\text{RRD}} = \exp\{-\xi_k\}\, I_0\left\{2\sqrt{\xi_k \gamma_k}\right\}. \tag{4.20}$$

4.2.4 *Noise spectrum estimation*

We can see from previous sections that all VADs require estimation of the noise spectrum $\lambda_{n,k}$ and the a priori SNR ξ_k. First we shall address the estimation of $\lambda_{n,k}$ and then the estimation of ξ_k.

In most VADs the noise spectrum estimation is done in a way to assume that in the first several frames only noise is present and for that time $\lambda_{n,k}$ is estimated by time averaging the spectrum of the recorded signal. Then, the VAD itself is used to discriminate between frames where speech is present and where only noise is present. When only noise is detected, $\lambda_{n,k}$ is again estimated in a time-averaging fashion. In this thesis the MCRA algorithm, proposed by [129] and [130], is used since it performs well in varying noise situations and it allows estimation from all frames, and not just the ones where no speech is detected.

MINIMA-CONTROLLED RECURSIVE AVERAGING. As stated earlier, a common technique for noise spectrum estimation is to apply temporal recursive smoothing during the frames when only noise is present. Now, we have the following hypotheses

$$\begin{aligned} H_0 &: \lambda_{n,k}(l+1) = a_n \lambda_{n,k}(l) + (1-a_n)|X_k(l)|^2 \\ H_1 &: \lambda_{n,k}(l+1) = \lambda_{n,k}(l), \end{aligned} \tag{4.21}$$

where $0 < a_n < 1$ is a smoothing parameter.

Let $p_{s,k}(l) = p(H_1 \,|\, X_k(l))$ denote the conditional speech presence probability at time frame l. Hence, we can write (4.21) as follows

$$\begin{aligned} \lambda_{n,k}(l+1) &= \lambda_{n,k}(l) p_{s,k}(l) + \left[a_n \lambda_{n,k}(l) + (1-a_n)|X_k(l)|^2\right](1 - p_{s,k}(l)) \\ &= \tilde{a}_{n,k}(l)\lambda_{n,k}(l) + (1 - \tilde{a}_{n,k}(l))|X_k(l)|^2, \end{aligned} \tag{4.22}$$

where

$$\tilde{a}_{n,k}(l) = a_n + (1-a_n) p_{s,k}(l) \tag{4.23}$$

is a time-varying smoothing parameter. We can see that the noise spectrum is estimated by averaging past power spectral values, using a smoothing parameter that is adjusted by the speech presence probability $p_{s,k}(l)$. In order to determine $p_{s,k}(l)$, speech absence is calculated by looking at the ratio of the local energy of the noisy signal and its minimum within a certain time frame.

Firstly, the squared magnitude of the spectrum is defined

$$S_{f,k}(l) = |X_k(l)|^2, \tag{4.24}$$

which could be smoothed in the frequency domain, but we have omitted this step due to the increase it brings to computational complexity. However, we do smooth the spectrum in the time domain

$$S_k(l) = \alpha_s S_k(l-1) + (1-\alpha_s) S_{f,k}(l), \tag{4.25}$$

where $0 < \alpha_s < 1$ is a smoothing parameter. The minimum of the local energy of the noisy signal is calculated by first initializing the minimum and temporary local variable: $S_{\min,k}(0) = S_k(0)$ and $S_{\text{tmp},k}(0) = S_k(0)$, respectively. Then, the minimum value of the squared amplitude spectrum is tracked in time

$$S_{\min,k}(l) = \min\{S_{\min,k}(l-1), S_k(l)\} \tag{4.26}$$

$$S_{\text{tmp},k}(l) = \min\{S_{\text{tmp},k}(l-1), S_k(l)\}. \tag{4.27}$$

Whenever the number of frames reaches an arbitrarily chosen M, the temporary variable is initialized by:

$$S_{\min,k}(l) = \min\{S_{\text{tmp},k}(l-1), S_k(l)\} \tag{4.28}$$

$$S_{\text{tmp},k}(l) = S_k(l). \tag{4.29}$$

We can see that the parameter M determines the scope of the local minima search, and that the temporary variable insures that the minimum will be adapted to a change in the noise level.

For calculating the conditional speech presence probability $p_{s,k}(l)$ a decision rule based on the ratio of the local energy of the noisy signal and its minimum, $S_{r,k}(l) = S_k(l)/S_{\min,k}(l)$, is needed

$$S_{r,k}(l) \underset{H_0}{\overset{H_1}{\gtrless}} \delta. \tag{4.30}$$

In [129] the following estimator for $p_{s,k}(l)$ was proposed

$$p_{s,k}(l) = \alpha_p p_{s,k}(l-1) + (1-\alpha_p) I_k(l), \tag{4.31}$$

where $0 < \alpha_p < 1$ is a smoothing parameter, and $I_k(l)$ is an indicator function for the result in (4.30), i.e. $I_k(l) = 1, \forall k$ if $S_{r,k}(l) > \delta$ and $I_k(l) = 0, \forall k$ if $S_{r,k}(l) < \delta$. At this point, we have calculated all the variables needed for the estimation of the noise spectrum via (4.22).

DECISION-DIRECTED A PRIORI SNR ESTIMATION. The DD estimation approach for the estimation of ξ_k, the a priori SNR, was proposed in [128]. Firstly, the Wiener gain is introduced as the following ratio

$$\zeta_k = \frac{\xi_k}{\xi_k + 1}. \tag{4.32}$$

Now, we can define the estimator for ξ_k

$$\xi_k(l) = \alpha_a \zeta_k(l-1)^2 \gamma_k(l-1) + (1-\alpha_a)\max\{\gamma_k(l) - 1, 0\}, \tag{4.33}$$

where $0 < \alpha_a < 1$ is a smoothing parameter.

The noise spectrum $\lambda_{n,k}$ and the a priori SNR ξ_k are continuously updated via the MCRA and DD methods, respectively, and are afterwards used in the statistical model-based VAD algorithms.

4.2.5 *Speech corpus and metrics for voice activity detection evaluation*

In order to analyze the supervised learning based VAD algorithms and performance thereof, we used the NOIZEUS speech corpus by [174]. Although the corpus was originally created for testing speech enhancement algorithms, we used it for the following reasons: (i) the recordings are of high quality and were made in a sound-proof booth, (ii) it offers eight different types of noises from AURORA database by [175] which corrupt the original recordings at four different SNR levels, (iii) the recordings were made by six different speakers—three male and three female, (iv) it uses the ieee sentence database which contains phonetically-balanced sentences with relatively low word-context predictability, and (v) the corpus is available to researchers free of charge. The percentage of the speech segments amounted to 61.28%, which is as twice as high as compared to [148], and [151], but less than 5% higher than in the cases of [152] and [155]. The recordings were sampled at the rate of 25 kHz and were later downsampled to 8 kHz. The total length of all the recordings was 80.04 s, which offered, with 50% overlap and frame length of $L = 256$, in total 5000 frames for detection. However, in order to test the performance and train the classifier for different types of noises and noise levels, we have added to the clean speech also versions corrupted with babble (SNR 15 dB, 10 dB, 5 dB), car (SNR 15 dB, 10 dB, 5 dB) and white Gaussian noise (SNR 20 dB, 15 dB, 10 dB). In total, this gave us 50000 frames for evaluation.

Usually, in order to test and train the algorithms, the speech segments are hand-labeled. However, in the present chapter we used signal energy calculated via Parseval's theorem as the indicator of speech presence, which enabled automatic frame labeling. We find this approach justifiable in the case of the NOIZEUS corpus, since the clean recordings were made in a sound-proof booth resulting with the speech-absent frames having energy a thousand times lower than the weakest speech frame.

The evaluation metrics we used are based on the standard elements of the confusion matrix: true positive (TP)—voice classified as voice, true negative (TN)—silence classified as silence, false positive (FP)—silence classified as voice, false negative (FN)—voice classified as silence. We also used speech detection rate (SDR)—percentage of speech frames classified as speech, and false alarm rate (FAR)—percentage of noise frames classified as speech. The

former and latter are calculated as follows

$$\text{SDR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FAR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{4.34}$$

These two rates are actually used in order to draw a receiver operating characteristics (ROC) curve. An ROC curve is a two-dimensional depiction of classifier performance. Usually, the curves are produced by graphing pairs of SDR and FAR values as a function of changes in the threshold value. To compare different classifiers it is practical to reduce the information in the ROC curve to a single scalar value. A common method is to evaluate the area under the receiver operating characteristics curve (AUC). For an example, since both the SDR and FAR take values in the range of $[0, \ldots, 1]$, for a perfect classifier the AUC value would be 1, since it is able to make a perfect SDR without any false alarms. A completely random classifier would have AUC value of 0.5, since the ROC curve would be a diagonal line in the SDR–FAR space. This would be equivalent to predicting based on fair coin tosses. More on the ROC curves and metrics for evaluation of classifiers can be found in [176, 177].

Another balanced measure of classification performance with respect to all elements is the Matthew's correlation coefficient (MCC) which we chose as additional metric for performance comparison. It is calculated as follows [176]

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FN})(\text{TP} + \text{FP})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}. \tag{4.35}$$

The MCC is always between −1 and +1, where −1 indicates total disagreement and +1 indicates total agreement. The MCC is zero for completely random predictions. If two variables are independent, then their MCC is zero. The converse in general is not true.

4.2.6 *Experimental comparison of statistical model-based VADs*

The results of statistical model-based VADs are shown in Fig. 4.1. By analyzing Figures 4.1a to 4.1b we can see that in the lower SNR scenarios the GGD and RRD mostly outperform the GD VAD. On the contrary, in Fig. 4.1d under very low SNR, the GD and RRD VAD show similar performance, and basically better results than the GGD VAD. We can see that with the changing SNR and noise type, the performance of the VADs relative to each other changes. But still, from Fig. 4.1, we can conclude that the RRD VAD shows equal or better performance than the other VADs in all four scenarios, and that preliminarily it seems as the best choice.

As suggested in [177], we generate results from several test subsets and average these results in order to obtain a measure of variance. The ROC curves can be either averaged vertically—by fixing FAR and averaging over SDR, or by the threshold—for each threshold value an SDR–FAR pair is found and their values are averaged thus yielding both vertical and horizontal variance. The test set for this experiment was constructed by concatenating the clean signal with its corrupted versions thus, with frame length of $L = 256$ samples, yielding 50000 examples for evaluation. In the present experiments we used 10-fold cross-validation procedure and threshold averaging. In Fig. 4.2 we can see the results of the experiment. Each point in the ROC curve also depicts a horizontal and vertical error bars which correspond to a value of three standard deviations. Moreover, in the legend we can also see the AUC score along with one standard deviation. By analyzing Fig. 4.2 we can assert that none of

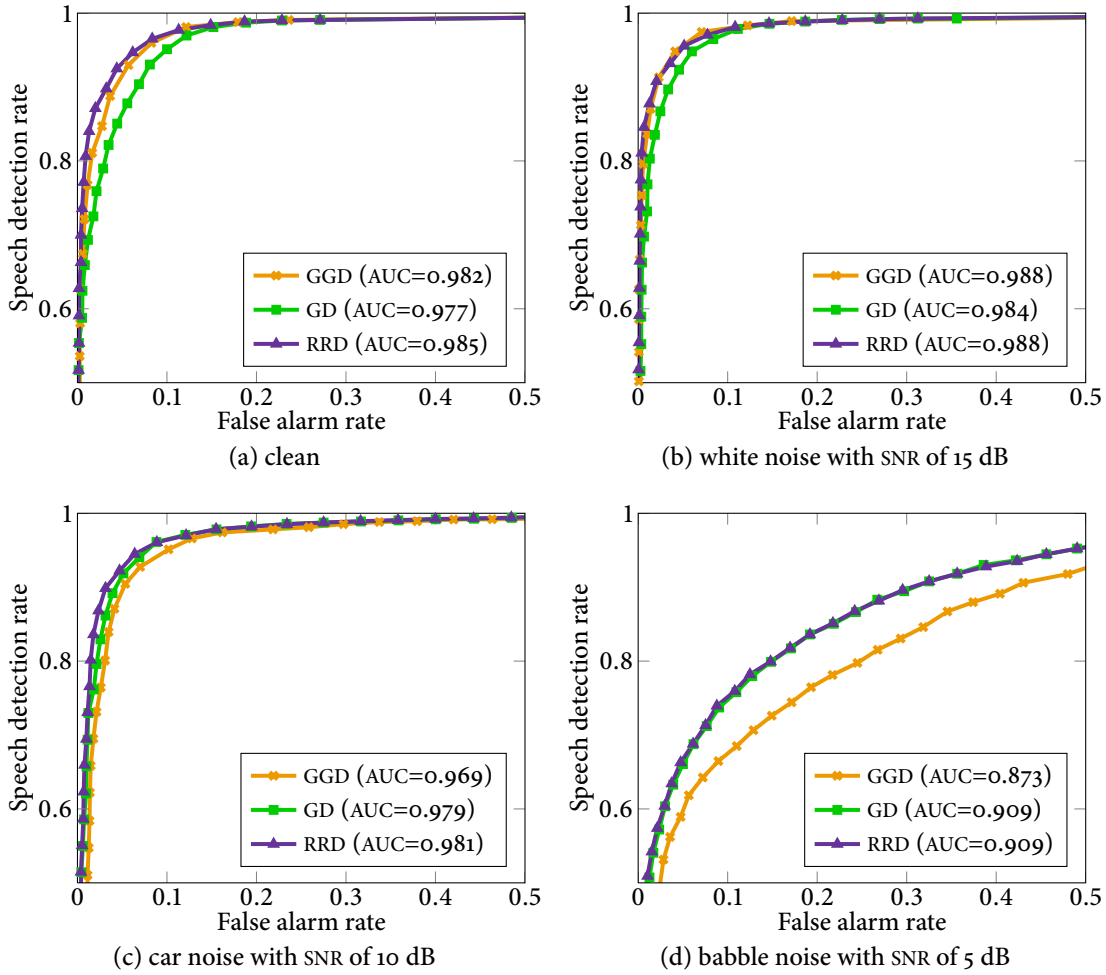


Figure 4.1: ROC curves for the three voice activity detectors. Each figure represents a different type of noise and a different SNR level.

the detectors exhibited large deviations and thus they all performed consistently on all the subsets, and that the RRD VAD, on average, had the best performance.

Another important parameter that should be analyzed is the computational demand, since we can see that (4.9), (4.17), and (4.20) differ in complexity. The execution times of all the VADs (without the MCRA and the DD SNR estimation), was measured for Matlab implementations on an Intel Core2Quad processor with 2.33 GHz frequency (only one core was used). The results were as follows: the GGD, RRD, and GD VAD had the execution times of 9.70 ms, 0.37 ms, and 0.21 ms, respectively. The reason behind the much higher computational complexity of the GGD VAD lies in the need to evaluate (4.12). Without this step, the GGD VAD takes on average 0.90 ms, which is still twice as much as the RRD VAD. However, a faster time varying estimate of the shape parameter ν could be utilized (cf. [178]) to lower the computational complexity.

The reader should note at this point, that in the thesis we have implemented the VADs somewhat differently than when they were first proposed in [148, 152, 172]. Mostly, the difference is in the noise spectrum and the a priori SNR estimation, and in the case of the RRD VAD, in the introduction of the LR for that model. Furthermore, the algorithms did exhibit some variance in performance with respect to changes in some of the smoothing

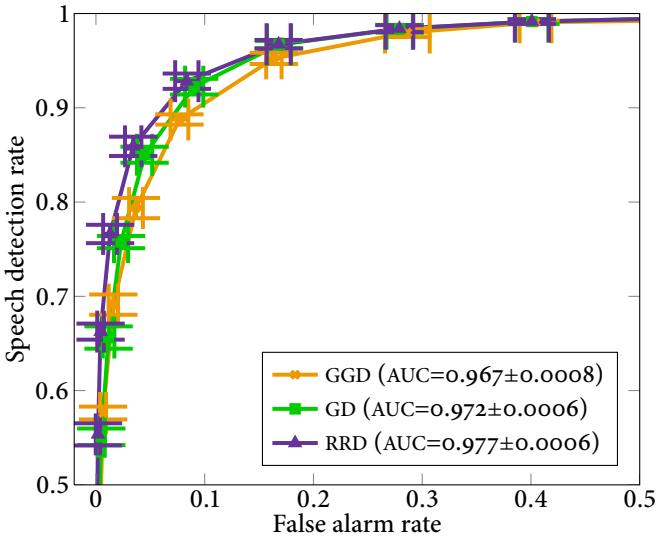


Figure 4.2: Threshold averaged ROC curves with AUC scores

parameters, but however this did not cause a change in relative performance of the detectors.

4.3 Supervised learning based voice activity detection

Before we start with classification, we need to choose input variables, i.e. features, upon which the classifiers will make decision and which, in effect, will be combined to form a strong classifier. We already mentioned that LR is one of the features, but we hypothesize that by adding other features we could improve the clasiffication results.

4.3.1 *Input variable selection via partial mutual information*

The partial mutual information (PMI) based input variable selection (IVS) algorithm used in [179, 180] overcomes two main issues that limit the applicability of many IVS techniques. Those are the underlying assumption of linearity and redundancy within the available data. The way that PMI IVS works is that it first selects the most informative input variable, then it searches for the next most informative variable but by taking into account information already received from the previously selected variable. This process continues until an introduction of an additional input variable increases the mean squared error of the prediction, i.e. the square of the expected value minus the label, or PMI drops below a certain threshold. Hereafter, we present the mathematical background of the PMI IVS.

Assuming y is a classification outcome, i.e. signal frame label, x is a currently considered input variable (feature), and $\mathbf{z}$ is a set of previously selected variables, partial mutual information in x about y given $\mathbf{z}$ is formulated as follows

$$\mathrm{PMI} = \iint p_{u,v}(u,v) \log \frac{p_{u,v}(u,v)}{p_u(u)p_v(v)} \mathrm{d}u\mathrm{d}v \,, \tag{4.36}$$

where $u = y - \mathrm{E}[y\,|\,\mathbf{z}]$, $v = x - \mathrm{E}[x\,|\,\mathbf{z}]$, and $\mathrm{E}[\,.\,]$ is the expectation operator.

In order to obtain probability density functions for PMI from the data, we used kernel

density estimators (KDEs). Hence, in order to calculate $\mathrm{E}[x\,|\,\mathbf{z}]$ we used the following KDE

$$\hat{p}(x,\mathbf{z}) = \frac{1}{n}\frac{1}{\left(\sqrt{2\pi}h\right)^{d}\sqrt{|\Sigma|}}\sum_{i=1}^{n}\exp-\frac{\|[x\ \mathbf{z}]^T-[x_i\ \mathbf{z}_i]^T\|_{\Sigma}}{2h^2}, \tag{4.37}$$

where $\|[x\ \mathbf{z}]^T-[x_i\ \mathbf{z}_i]^T\|_{\Sigma} = ([x\ \mathbf{z}]-[x_i\ \mathbf{z}_i])\,\Sigma^{-1}([x\ \mathbf{z}]^T-[x_i\ \mathbf{z}_i]^T)$ is the Mahalanobis distance, and h is the kernel bandwidth, for which we used the Gaussian reference bandwidth throughout this chapter

$$h = \left(\frac{4}{d+2}\right)^{\frac{1}{d+4}} n^{-\frac{1}{d+4}}\,, \tag{4.38}$$

where d is the dimension of the multivariate variable set, and n is the sample size.

Note that for $\mathrm{E}[x\,|\,\mathbf{z}]$ we need $\hat{p}(x\,|\,\mathbf{z})$. If we take

$$\Sigma = \begin{bmatrix}\Sigma_{xx} & \Sigma_{xz}\\ \Sigma_{zx} & \Sigma_{zz}\end{bmatrix}, \tag{4.39}$$

we get

$$\hat{p}(x\,|\,\mathbf{z}) = \frac{1}{n}\frac{1}{\left(\sqrt{2\pi}h\right)^{d}\sqrt{|\bar{\Sigma}|}}\sum_{i=1}^{n}\exp-\frac{\|x^T-\bar{x_i}^T\|_{\bar{\Sigma}}}{2h^2}\,, \tag{4.40}$$

where $\bar{\Sigma} = \Sigma_{xx}-\Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}$ and $\bar{x_i} = x_i + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z}-\mathbf{z}_i)$. Finally,

$$\mathrm{E}[x\,|\,\mathbf{z}] = \sum_{i=1}^{n} w_i\left[x_i + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z}-\mathbf{z}_i)\right]\,, \tag{4.41}$$

where each sample is weighted by its weighting factor introduced in [179]

$$w_i = \frac{\exp\left(-\frac{\|\mathbf{z}^T-\mathbf{z}_i^T\|_{\Sigma_{zz}}}{2h^2}\right)}{\sum_{j=1}^{n}\exp\left(-\frac{\|\mathbf{z}^T-\mathbf{z}_j^T\|_{\Sigma_{zz}}}{2h^2}\right)}\,. \tag{4.42}$$

The pseudocode of IVS based on PMI utilized in this thesis is given in Algorithm 4.

4.3.2 *Feature space*

In the ensuing paragraphs we present features that form the potential input variable set. Each of them was analyzed as a standalone detector and as a candidate for the reduced input vector by the PMI IVS.

Magnitude of the DFT coefficients. A K-point transform was used to analyze the spectrum of the recorded frames. The magnitude of the first 32 coefficients of the transform were used as a feature for the classifier.

Zero-crossing rate. The zero-crossing rate (ZCR) of a signal is the rate of sign changes along the signal. It is defined as follows

$$\begin{aligned} f_{ZCR} &= \sum_{i=2}^{L} Z_i,\\ \text{where } Z_i &= \begin{cases} 1, & \text{if } \operatorname{sign}\{x(i)\}-\operatorname{sign}\{x(i-1)\}\neq 0\\ 0, & \text{otherwise.}\end{cases}\end{aligned} \tag{4.43}$$

Algorithm 4: Input variable selection based on partial mutual information.

Require: Sets of considered variables $\mathcal{X} = \{x_1, x_2, \dots\}$ and labels $\mathcal{Y} = \{y_1, y_2, \dots\}$
Ensure: Set of chosen input variables $\mathcal{Z} = \{z_1, z_2, \dots\}$

```
# Write function mse( . ) that calculates mean-squared error
Initialize Z ← ∅
Initialize u_MSE ← ∞
while X ≠ ∅ do
    Construct an estimator E[y | z]
    Calculate u ← y − E[y | z]
    if u_MSE < mse(u) then
        # Remove previously added x from Z
        Z ← Z \ x_last
        exit
    end if
    u_MSE ← mse(u)
    for all x ∈ X do
        Construct an estimator E[x | z]
        Calculate v ← x − E[x | z]
        Determine the PMI I(v, u)
    end for
    # Determine x, i.e. v, which maximizes I(v, u)
    x_s = arg max_x I(v, u)
    if I(v, u) < I_min then
        exit
    end if
    Z ← Z ∪ x_s
end while
```

Human voice consists of voiced and unvoiced sounds. Voiced sounds have higher ZCR value than the unvoiced sounds do. Therefore, it is a reasonable assumption that ZCR of either voiced or unvoiced parts of speech will be different than the ZCR of noise in the silent periods.

Spectral flux. The spectral flux (SF) measures how quickly the spectrum of the signal is changing. It is calculated by comparing the power spectrum of the current frame with the power spectrum of the previous frame

$$f_{SF} = \left| \sum_{k=1}^{K} \left(|X_k(l)|^2 - |X_k(l-1)|^2 \right) \right|. \tag{4.44}$$

Speech changes quickly between voiced and unvoiced parts, thus resulting with high SF values.

Spectral rolloff. The spectral rolloff (SR) is defined as the a-quantile of the total energy in $|X_k|^2$. It is a frequency under which a fraction of the total energy is found. If K is the length

of the signal DFT, then SR can be defined as

$$f_{SR} = \max_{y} \left\{ y : a > \frac{\sum_{k=1}^{y} |X_k|^2}{\sum_{k=1}^{K} |X_k|^2} \right\}. \tag{4.45}$$

Spectral rolloff was calculated at six quantiles equally spaced in $[0, 1]$.

Mel-frequency cepstral coefficients. Mel-frequency analysis is a technique inspired by human sound perception. The human ear acts as a filter and concentrates only on specific spectral components. The filters are non-uniformly spaced on a frequency scale, and their density is higher in the low frequency regions. The MFCCs are calculated in several steps: (i) the magnitude spectrum $|X_k|$ is filtered with a bank of non-uniformly spaced overlapping triangular filters, (ii) the logarithm is taken, and (iii) the MFCC are obtained by computing the discrete cosine transform of the result. In [181] where authors consider a voice conversion system, MFCC feature is identified as a feature that does not consider any particular speech model, i.e. feature that is useful for general voice activity detection, without considering any speaker in particular.

Power-normalized cepstral coefficients. In [182, 183] a feature extraction algorithm called power normalized cepstral coefficients (PMCCs) was proposed, which instead of log nonlinearity like MFCC uses power-law nonlinearity and a gammatone filterbank. In [182] it was shown to outperform MFCC, among others, in speech recognition accuracy. After adapting the algorithm proposed in [182] to our scenario, we have used the first thirteen PMCCs which were the result of a 20 element gammatone prefiltering.

Spectral centroid. The spectral centroid (SC) is a statistic that measures where most of the power of a speech segment is spectrally located. It is defined as follows

$$f_{SC} = \frac{\sum_{k=1}^{K} k|X_k|^2}{\sum_{k=1}^{K} |X_k|^2}. \tag{4.46}$$

Spectral bandwidth. The spectral bandwidth (SBW) describes spreading of the spectral components with respect to the spectral centroid

$$f_{SBW} = \sqrt{\frac{\sum_{k=1}^{K} (k - f_{SC})^2 |X_k|^2}{\sum_{k=1}^{K} |X_k|^2}}. \tag{4.47}$$

Feature aggregation. In total the following features were aggregated: 1 LR, 32 DFT magnitude coefficients, 1 ZCR, 1 SF, 6 SR quantiles, 15 mel-frequency cepstral coefficients, 13 power normalized cepstral coefficients, 1 SC and 1 SBW. Thus, we had a feature vector of 71 for input variable analysis. Similar approach was used in [159, 184] for music classification.

4.3.3 *Individual feature performance and IVS results*

Each of the presented features can be considered as a detector in itself, whose performance might indicate the suitability of being an element in the input vector. As an intuitive preliminary analysis, we utilized the ROC curves, i.e. the related AUC score, of each feature evaluated on the whole data set at once. Table 4.1 shows the AUC for all the features presented in the current section. We can see that the LR has the highest score, followed by the first PMCC, SF, the first MFCC coefficient, while the third and ninth PMCC have the lowest score.

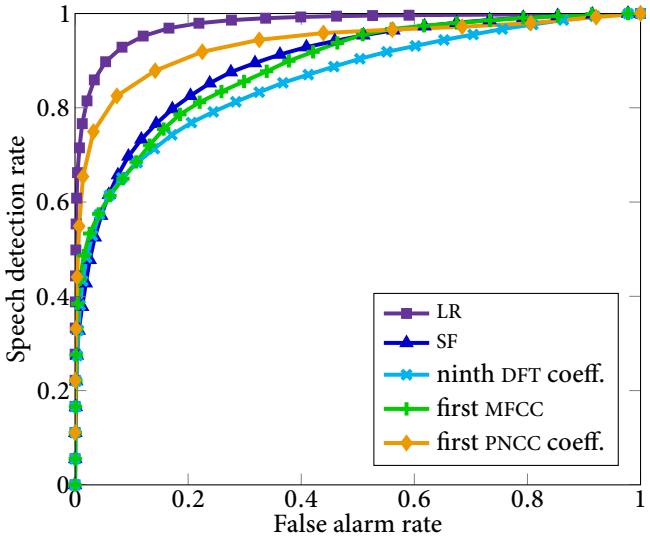


Figure 4.3: ROC curves for the five features with the highest AUC score

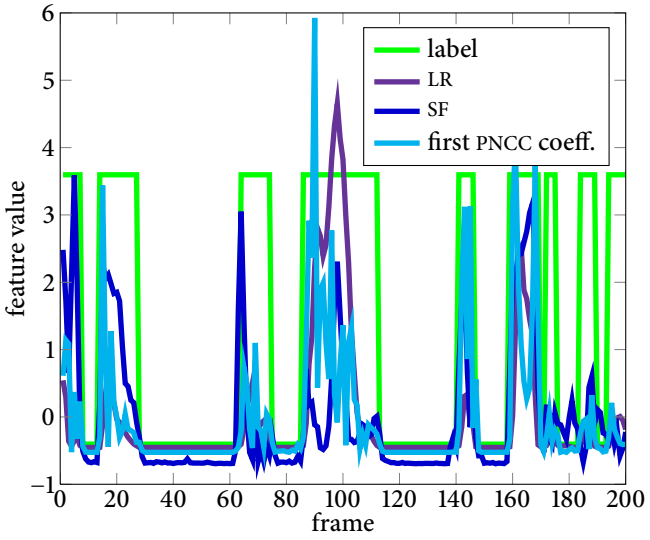


Figure 4.4: Feature values for a random segment of 200 frames corrupted with babble noise (15 dB SNR)

Furthermore, ROC curves for five features with the highest AUC score are depicted in Fig. 4.3, while the values of three features with the highest AUC score along with the label for 200 frames are depicted in Fig. 4.4.

Due to high memory requirements the analysis based on partial mutual information was carried out on the set consisting of the clean signal, and its versions corrupted with babble (SNR 10 dB), car (SNR 10 dB), and white Gaussian noise (SNR 15 dB) separately. The analysis on each set was stopped once the addition of another feature caused increase in the mean squared error. Based on the results we kept those features that were chosen in at least two sets: the LR, DFT indexes 7, 8, 9, 11, the 1st and 2nd SR, the 1st MFCC, SC, SBW, and 1st, 2nd and 3rd PMCC. It is interesting to note that the PMI algorithm chose the 3rd PMCC as a good feature, although it has by far the lowest AUC score than many other features. However, the PMI chooses features which bring additional information when all the information from other features is taken into account, meaning that in certain scenarios the 3rd PMCC

Table 4.1: AUC score of all the features

	Feature	AUC		Feature	AUC		Feature	AUC
1.	LR	0.978	25.	31st DFT	0.708	49.	3rd DFT	0.632
2.	1st PMCC	0.936	26.	4thSR	0.708	50.	1stSR	0.624
3.	SF	0.895	27.	22nd DFT	0.708	51.	12th MFCC	0.622
4.	1st MFCC	0.888	28.	32nd DFT	0.706	52.	4th DFT	0.619
5.	9th DFT	0.861	29.	23rd DFT	0.706	53.	4th MFCC	0.609
6.	15th DFT	0.815	30.	21st DFT	0.704	54.	14th MFCC	0.609
7.	8th DFT	0.810	31.	20th DFT	0.702	55.	4th PMCC	0.603
8.	6th MFCC	0.809	32.	19th DFT	0.702	56.	6th DFT	0.602
9.	16th DFT	0.805	33.	30th DFT	0.700	57.	9th MFCC	0.601
10.	14th DFT	0.793	34.	24th DFT	0.694	58.	7th PMCC	0.597
11.	10th DFT	0.786	35.	2nd MFCC	0.692	59.	3rd MFCC	0.586
12.	17th DFT	0.767	36.	5th MFCC	0.686	60.	8th MFCC	0.583
13.	13th DFT	0.765	37.	29th DFT	0.680	61.	13th MFCC	0.566
14.	12th DFT	0.751	38.	25th DFT	0.663	62.	10th PMCC	0.564
15.	11th DFT	0.747	39.	1st DFT	0.661	63.	11th PMCC	0.561
16.	7th MFCC	0.743	40.	28th DFT	0.660	64.	13th PMCC	0.554
17.	3rd SR	0.739	41.	2nd PMCC	0.658	65.	15th MFCC	0.548
18.	ZCR	0.731	42.	6th PMCC	0.655	66.	5th PMCC	0.545
19.	18th DFT	0.726	43.	2nd DFT	0.655	67.	10th MFCC	0.541
20.	2nd SR	0.725	44.	7th DFT	0.652	68.	8th PMCC	0.519
21.	SBW	0.722	45.	27th DFT	0.648	69.	12th PMCC	0.518
22.	5thSR	0.720	46.	11th MFCC	0.647	70.	3rd PMCC	0.511
23.	6thSR	0.719	47.	26th DFT	0.644	71.	9th PMCC	0.505
24.	SC	0.713	48.	5th DFT	0.637			

contributed to correct classification. In total this amounts to 13 features forming a reduced vector of input variables, which is an 82% decrease in the size of the feature vector.

Although from Fig. 4.3 we can see that the LR as a standalone detector outperforms other features, we conjecture and shall test (i) that a trained classifier based on LR and other features should outperform a statistical model-based detector based on LR, and (ii) that a detector with carefully chosen reduced input vector should not significantly underperform the detector based on a full feature vector. We shall test these hypotheses on 50000 learning examples and by meticulous analysis with ROC curves and the AUC metric.

4.3.4 *Evaluation of the supervised learning VAD algorithms*

In this chapter we utilized and compared three supervised learning algorithms: SVM, Boost, and ANN, which were to classify if a signal frame contains speech or not based on the full and the reduced feature set generated by the algorithm in Section 4.3.1. The three have different approaches to learning and all have their advantages, and we shall briefly introduce each in the following paragraphs. But it is important to notice at this point that the goal of this section is not to provide a detailed tutorial in either of the classifiers, but to analyze and

compare the performance of the three for the specific purpose of voice activity detection based on various features and not in general. For training and testing the three learning algorithms we used the OpenCV library [185].

Essentially, SVM [186, 187] is a learning algorithm that constructs a hyperplane or a set of hyperplanes which define boundaries for the data to be discriminated. The data, most often, are not linearly separable and this problem is addressed by SVM in a way that non-linearly maps the input vector with a kernel function to a high-dimensional feature space. The SVM can also be used in regression tasks, but in this section we use them in the context of a binary classifier. An introduction to the theory behind SVM and some practical insights can be found in [188]. In the thesis we used C-support vector classification and radial basis function (RBF) as the kernel function.

The main idea behind boosting algorithms is to use many simple detectors which should have performance a bit better than 50% at least (i.e. better than random guessing)—these are called weak classifiers—and combine them to obtain highly accurate classifier—usually called a strong classifier. In its original form, Boost handles binary classification problems only, although there are extensions to handle multi-class and even multi-label classification problems [189]. In the thesis, a variant of the Boost algorithm proposed in [190] called Real Boost is used [191].

The ANNs are a product of the desire to imitate the workings of the biological brain. They involve a network of simple processing elements (artificial neurons) which can exhibit complex global behavior. One of the most important properties of ANNs is the ability to approximate any continuous function up to a given precision. They have been extensively used in both classification and regression tasks and more on the ANNs can be found in [192]. In the thesis we utilize a static multilayer perceptron network (MLP) with a sigmoid activation function, a single hidden layer with 5 neurons, while the network parameters are learned using the resilient propagation (RPROP) algorithm [193].

In the sequel we analyze the performance of the classifiers. The data was constructed by concatenating the clean signal with its corrupted versions thus, with frame length of $L = 256$ samples, yielding 50000 examples for evaluation. For the full input vector we had 71 features, while the reduced input vector consisted of 13 features. Prior to the learning process, all the features were scaled in a way to have a zero mean value and standard deviation of one.

The evaluation was performed by K-fold cross-validation. Essentially, the original dataset was partitioned randomly into K subsets of equal size. Of the K subsets, one was retained for testing the classifier while the other $K - 1$ subsets were used for training. The cross-validation process was repeated K times thus yielding K results which were used for drawing the average ROC curves. As discussed in [177], by drawing just an ROC curve of different classifiers and seeing which one dominates to assess the performance might be misleading, since we do not have a measure of variance. Therefore, it is suggested to generate results from several test subsets, by a cross-validation or bootstrap method, and average these results in order to obtain a measure of variance. The ROC curves can be either averaged vertically by fixing FAR and averaging over SDR, or by the threshold, where for each threshold value an SDR–FAR pair is found and their values are averaged thus yielding both vertical and horizontal variance. In this section we used 10-fold cross-validation and threshold averaging for evaluation of the VAD algorithms.

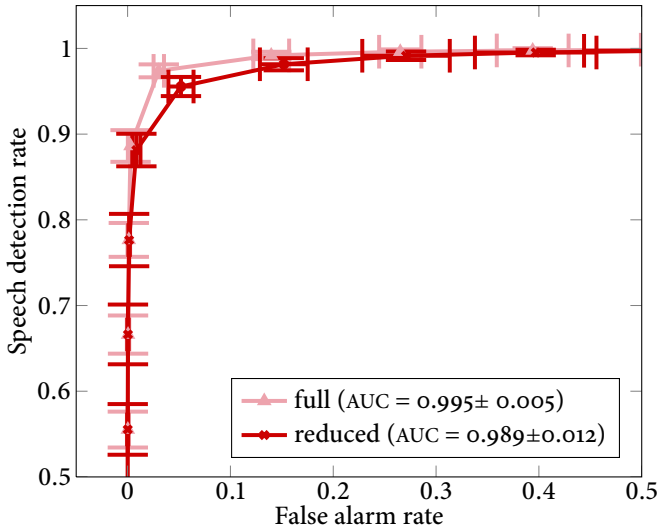


Figure 4.5: Averaged ROC curves for the SVM classifier with the full and reduced input vector

Firstly, we compared intra-classifier performance, i.e. performance of each classifier working with either the full or the reduced input vector. Henceforth, all the figures depicting ROC curves have for each point a confidence interval of three standard deviations included, along with the AUC score and three standard deviations thereof. These deviations indicate just how consistent the classifier performance was with respect to different cross-validation sets. Figure 4.5 shows the averaged ROC curves and their AUC score for the SVM, from which we can see that the classifier with the reduced feature set did not significantly underperform compared to the classifier trained on the full feature set. In Fig. 4.6 we can see a bit different result for the Boost classifier. In this case the classifier showed practically equal performance both in the mean and standard deviation when being trained on the full and the reduced input set. Finally, Fig. 4.7 shows the averaged ROC curves and their AUC score for the ANN. It performed slightly better in the mean and standard deviation of the AUC score with the full input vector, but overall exhibited larger deviations than any of the other two classifiers. This means that it did not perform as consistently over all the subsets.

To conclude the intra-classifier analysis, we can assert that the results supported our second hypothesis from the Section 4.3.1: neither of the classifiers significantly underperformed when being trained on the reduced input vector formed by a careful IVS. Henceforth, we shall only include in the analysis the classifiers trained on the reduced input vector.

For the inter-classifier performance we also included the statistical model-based detector presented in Section 4.2.3 which too was evaluated by K-fold cross-validation. Since it does not require training it was simply tested on the same K subsets and these results were averaged. Figure 4.8 shows ROC curves for the three supervised learning classifiers and the RRD detector based on LR, from which we can see that the supervised learning approach with several additional features can significantly increase the performance of a detector. Moreover, judging from the AUC scores shown in Fig. 4.8 we can assert that the Boost classifier slightly outperforms the other classifiers, since it has the largest AUC mean value and the smallest AUC standard deviation. Furthermore, by inspecting Figures 4.5, 4.6, and 4.7 we can also see that Boost overall exhibited smaller deviations in the ROC curves, which further tips the balance in Boost's favor.

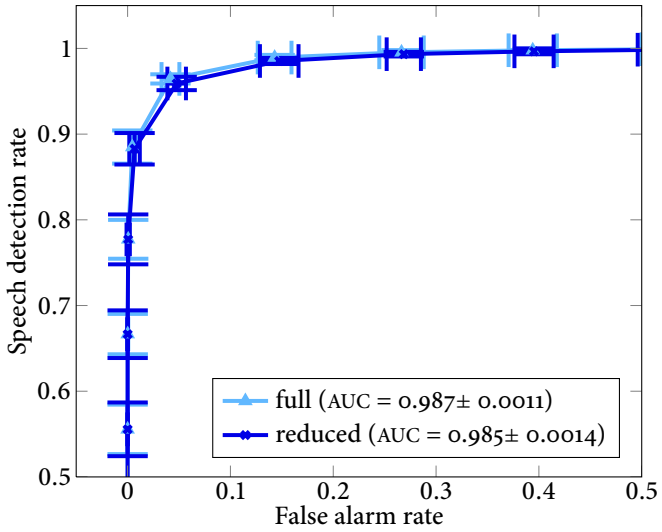


Figure 4.6: Averaged ROC curves for the Boost classifier with the full and reduced input vector

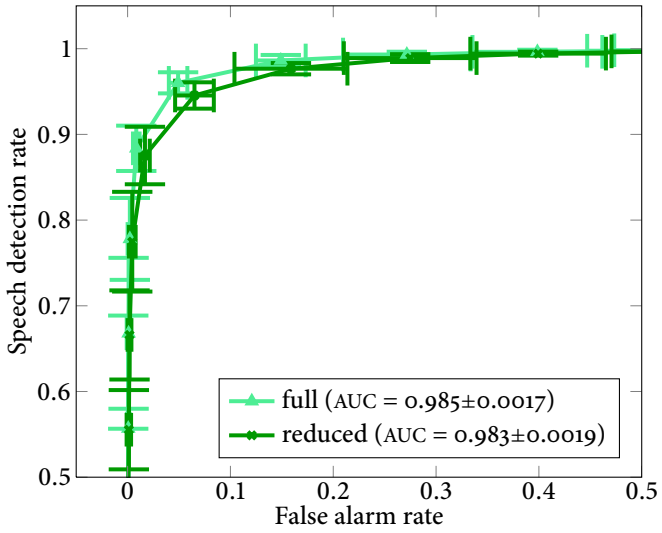


Figure 4.7: Averaged ROC curves for the ANN classifier with the full and reduced input vector

Table 4.2: Averaged statistical scores of the trained classifier performance

		SDR [%]	FAR [%]	ERR [%]	MCC±3σ_{MCC}
SVM	full	96.73	2.26	5.53	0.944±0.0141
	red	94.47	3.71	9.24	0.906±0.0183
Boost	full	95.79	3.35	7.56	0.923±0.0132
	red	95.10	3.75	8.65	0.912±0.0150
ANN	full	95.23	3.90	8.67	0.912±0.0189
	red	93.43	5.05	11.62	0.882±0.0309

During the K-fold cross-validation we also monitored the performance of the trained classifiers for each subset by calculating the SDR, FAR, and MCC presented in Section 4.2.5. Since all the classifiers were trained to output a value between −1, for non-speech, and 1, for

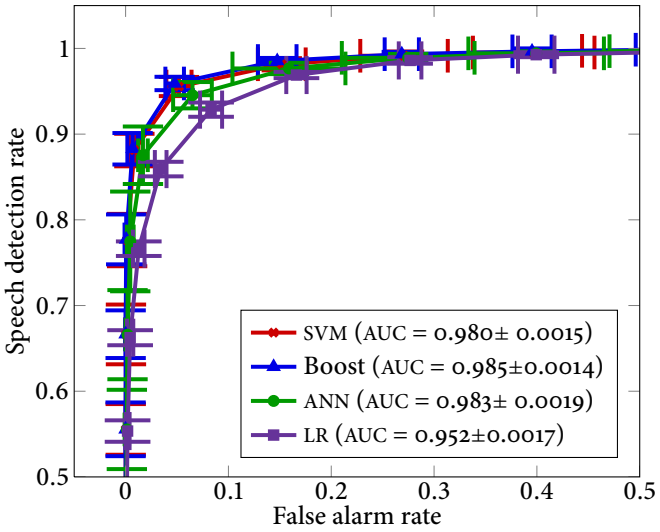


Figure 4.8: Averaged ROC curves for all the classifiers with the reduced input vector and the detector based solely on the LR

speech frames, we set the threshold to zero, thus all the frames with score larger or equal to zero were classified as containing speech, while the other were classified as non-speech frames. This essentially would correspond to only a single point in the ROC curve graph, but it is very practical since it provides a tangible sense of performance for a single threshold value. The average of these statistical scores for the aforementioned 10 subsets is shown in Table 4.2, where we also provide error rate (ERR), ERR = (100 - SDR) + FAR, since it is often used in other works.

To conclude the inter-classifier performance, from the above presented results we can see that the classifiers significantly outperformed the statistical model-based detector, and that due to having the highest AUC score with the smallest standard deviation, and exhibiting no significant deviations anywhere in the ROC curve, the Boost algorithm had the advantage over the other algorithms for this specific application of speech activity detection based on various features. Therefore, we can assert that the results supported our first hypothesis from Section 4.3.1 that a trained classifier based on LR and other features should outperform a statistical model-based detector based on LR.

These experiments were designed so as to find a LR model that will show the best results [170], which we would then extend with features meticulously analyzed with PMI IVS and encompass it all in a supervised learning framework which showed the best and most consistent performance. Furthermore, the corpus that we used is freely available to all researchers [174] which will enable direct comparison of detection algorithms in the future. Comparison of our results to works which utilized a supervised learning approach [162–167] is not straightforward due to utilization of a different speech corpus, graphical result representation (no score presented) or non-direct metric (word accuracy rate in speech recognition). However, some do provide ERR score for different noise levels and types which we will use for crude comparison with our results. For an example, in [162] the best ERR was 5.38% and 13.47% for vehicle and office noise, respectively, while [163] reports 9.4% and 20.9% for vehicle and babble noise, respectively. In [167] authors report ERR from 7.83% to 41.39% for different test sequences. The authors in [164] report a score

named equal error rate for which equality 1-SDR=FAR holds. For three different datasets they report equal error rate of 8.0%, 13.1%, and 19.0% for an SVM trained on MFCC. Comparing these results with Table 4.2 we can see that our results do not deviate and are in the rank of their performance. However, since different datasets were used in these papers, a direct comparison is not possible.

4.4 SUMMARY

In this chapter we have presented three different statistical model-based voice activity detection algorithms in an unifying and consistent manner, by incorporating noise spectrum and the a priori signal-to-noise ratio estimation to their respective frameworks. Furthermore, we introduced the LR for the Rayleigh and Rice distribution based detector. The decision framework was based on a statistical hypothesis ratio measure, and its geometric mean over all the DFT coefficient indices. The algorithms were tested on the NOIZEUS speech corpus which consisted of clean recordings, and its versions corrupted with three types of noises and three different SNRs. The performance analysis was conducted using threshold averaged ROC curves and AUC score. Based of the aforementioned parameters, and the computational complexity, we concluded that the VAD based on Rayleigh and Rice distribution showed the best performance on average and is the most suitable statistical model-based VAD among the tested algorithms.

Furthermore, we have introduced in total 70 additional features which were combined with the RRD based VAD to form an input vector for the supervised learning classifiers. The input vector was extensively analyzed by a partial mutual information algorithm in order to single out the most informative features and by AUC score analysis to test the capability of each feature to serve as a VAD. The results yielded a 13 element reduced input vector. We have focused on SVM, Boost and ANN classifiers, whose performances were mutually compared both with the full and the reduced input vector. The algorithms were also tested on the NOIZEUS speech corpus. The performance evaluation was based on a 10-fold cross-validation and compared on threshold averaged ROC curves, AUC score and MCC. Firstly, the results showed that the performance was not undermined by utilizing the vector with the reduced number of features. Secondly, although the statistical model-based VAD by itself is a much better detector than any of the other utilized features, a combination of the latter and the former in the form of a trained classifier produced a VAD with significantly better performance. Finally, inter-classifier analysis showed similar performance of the three, with a slight advantage in the direction of the Boost classifier, since it had the highest AUC score and the smallest variability in the threshold averaged ROC curves, indicating a consistent performance over all the test subsets.

The presented approach consisting of aggregating various features, performing input variable selection by a partial mutual information algorithm whereat a reduced input vector is created, and training a classifier for voice activity detection, is quite generic. It can be used on any combination of features and, indeed, is not limited just to voice activity detection. In order to further increase the VAD performance or tailor it to specific scenarios, a cascaded classifier architectures could be utilized.

5

Detection and tracking in omnidirectional images

Equipping mobile robots with an omnidirectional camera and consequently endowing them with an entire view of the scene is very advantageous in numerous applications as all information about the surrounding scene is stored in a single image frame. In the given context, this chapter is concerned with detection, tracking and following of a moving object with an omnidirectional camera equipped mobile robot. The camera calibration and image formation is based on the spherical unified projection model thus yielding a representation of the omnidirectional image on the unit sphere. The detection of the moving object is performed by calculating a sparse optical flow in the image and then lifting the flow vectors on the unit sphere where they are discriminated as dynamic or static by analytically calculating the distance of the terminal vector point to a great circle arc. The flow vectors are then clustered and the center of gravity is calculated to form the sensor measurement. Furthermore, the tracking is posed as a Bayesian estimation problem on the unit sphere and the solution based on the von Mises-Fisher distribution is utilized. Visual servoing is performed for the object following task where the control law calculation is based on the projection of a point on the unit sphere. In conclusion, experimental results obtained by a camera with a fish-eye lens mounted on a differential drive mobile robot are presented and discussed. Majority of the research presented in this chapter was conducted at the Lagadic research group of INRIA Rennes-Bretagne Atlantique.

5.1 Introduction

Omnidirectional cameras by their definition provide a 360° view of the surrounding scene and as such pose themselves as a powerful tool in robot's vision system. The enhanced field of view can be obtained by using several synchronized panoramic cameras, a combination of a camera and a mirror, or a camera with a wide-angle lens. The amount of information in such a single image reinforces robot's abilities in interpreting and adequately acting and reacting in the environment. The sensor has been utilized in mobile robotics in a variety of applications: visual odometry, navigation, structure-from-motion, visual servoing, and moving object tracking to name but a few. This chapter is concentrated on the moving object detection, tracking and following with an omnidirectional camera equipped mobile robot.

Detection and tracking of moving objects with a camera mounted on a mobile robot is a task inconvenienced by the simultaneous ego-motion of the robot and the motion of the objects. With perspective cameras the problem is approached in [194] by calculating

the optical flow and optimizing the bilinear transformation to warp the image between the consecutive frames, after which the images are differentiated and motion is detected. Then the particle filter is used to track the moving objects in the image and a laser range finder is used to infer about the location in 3D. In [195] the detection was based on monocular scene reconstruction and affine transformation of a triangle mesh in order to perform the image warping. The tracking of the moving object and the scene reconstruction was performed using the extended Kalman filter. In [24] and [196] an omnidirectional image was first unwrapped to a panoramic image using a cylindrical projection, where the optical flow was calculated. In the former a synthetic optical flow is generated by estimating the position of the focii of expansion and contraction and the calculated flow is compared to the generated one, while the latter estimates an affine transform on square subsegments to warp the image and perform the differentiation. In [25] the omnidirectional image is segmented in a set of perspective images and detection is done in the vein of [194], while the tracking is based on the particle filter. To perform the following a control law based on a minimization of an ad hoc following error is calculated.

In this chapter, excepting the low-level image processing, we propose a method for moving object detection, tracking and following entirely based on processing on the unit sphere thus taking into account the specific geometry and making it as general as possible for omnidirectional systems. The moving object detection is based on analytically calculating the distance of a point on the sphere to an arc on the sphere. The tracking is performed in an analytical Bayesian prediction-correction manner where the underlying distribution is a spherical distribution, namely the von Mises-Fisher distribution. In the end, the object following is based on visual servoing [197] where the control law is calculated from an interaction matrix derived for a projection of a point on the unit sphere. The experiments were carried out at INRIA Rennes-Bretagne Atlantique in Rennes, France in the laboratory of the Lagadic group on a Pioneer 3DX differential drive mobile platform that we equipped with an omnidirectional camera composed out of a Point Grey Dragonfly2 camera and an Omnitech Robotics fish-eye lens. The task of the robot was to detect a moving object in the image, track it and use the visual servoing control law to keep the detected direction of the object at a specific location in the image, thus effectively following the object. Some of the novelties and results from this chapter were presented in [198] where the optical flow field segmentation is analyzed, in [22] where the detection is tacked for the first time by iteratively back-projecting the detected features from different heights in the world, and in [61] where the approach based on processing on the unit sphere is proposed along with the tracking based on the VMF distribution and following based on visual servoing.

5.2 UNIFIED PROJECTION MODEL AND CAMERA CALIBRATION

The unified projection model describes the image formation in catadioptric systems with a unique effective viewpoint, which includes the appropriate combinations of the mirror—parabolic, hyperbolic, elliptic, planar—and the lens—orthographic or perspective. A theoretical derivation of complete single-lens single-mirror catadioptric sensors characterized by a unique effective viewpoint was introduced in [199], while the unified projection model was introduced and studied in [43, 44]. There exists several methods for calibrating such

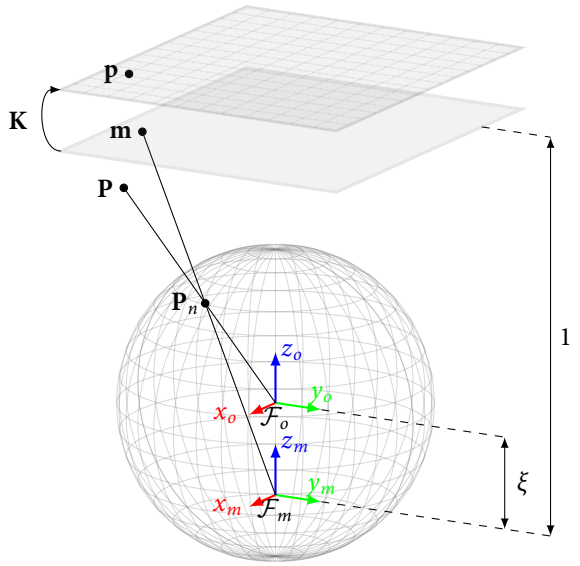


Figure 5.1: Illustration of the unified image formation

camera systems and a recent comparison can be found in [200]. Although the model and the calibration methods were developed for systems with a unique effective viewpoint, in practice they have been shown to be valid for dioptric systems with a fish-eye lens [45]. In this thesis we have chosen to use the calibration method based on planar grids proposed in [201] since it provides an analytical way of calculating the point on the sphere from pixel coordinates. In the vein of [201] we restate the model here since we find it important for understanding the methods proposed in this chapter.

The model consists of two consecutive projections: the spherical and the perspective. Consider a point in space $\mathbf{P}$ and the frame $\mathcal{F}_o : (x_o, y_o, z_o)$ attached to the origin of the unit sphere as shown in Fig. 5.1. First, $\mathbf{P}$ is projected to the surface of the sphere, which amounts to normalizing the points coordinates. Then, the normalized point $\mathbf{P}_n$ is perspectively projected from the coordinate system $\mathcal{F}_m : (x_m, y_m, z_m)$ to the point $\mathbf{m}$ on the normalized plane.

The model considers two main sources of distortion: imperfection of the lens shape that are modeled by radial distortion and improper lens and camera assembly that generate both the radial and tangential errors. Five parameters are used to model the distortion

$$f(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + k_5\rho^6, \tag{5.1}$$

where $\rho = \sqrt{x^2 + y^2}$ with (x, y) being coordinates of the point $\mathbf{m}$ in the normalized plane. To model the tangential distortion the following expression is used

$$dx = \begin{bmatrix} 2k_3xy + k_4(\rho^2 + 2x^2) \\ k_3(\rho^2 + 2y^2) + 2k_4xy \end{bmatrix}. \tag{5.2}$$

After the distortion modeling the point in the image $\mathbf{p}$ is obtained by calculating $\mathbf{p} = \boldsymbol{K}\mathbf{m}$, where $\boldsymbol{K}$ is a 3×3 matrix containing the camera intrinsic parameters. The matrix $\boldsymbol{K}$, the distortion parameters, and ξ are obtained by the calibration procedure [201]. In this thesis we used a standard perspective camera with a fish-eye lens, and the best calibration results

were obtained by utilizing just the $\boldsymbol{K}$ and ξ, i.e. without distortion modeling. In the sequel we assume that our omnidirectional camera is calibrated, which consequently enables us to apply the inverse projection (lifting) of the point in the image to a point on the unit sphere [201]

$$\mathbf{m} = \boldsymbol{K}^{-1}\mathbf{p}, \quad \mathbf{P}_n = \begin{bmatrix} \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{bmatrix}, \tag{5.3}$$

where, as before, the x and y are coordinates in the normalized plane of the point $\mathbf{m}$.

5.3 DETECTING MOVING OBJECTS

The main goal of our vision system is to detect moving objects in the omnidirectional image while the robot itself moves. However, this proves to be a daunting task since we have motion in the image induced both by the moving objects and the ego-motion of the robot. We have approached this problem in one of our previous works [22] by estimating the sparse optical flow in the image and by using the robot's odometry to discriminate between the flow vectors induced by the ego-motion (static features) from those induced by the moving objects (dynamic features). The detection part in this chapter continues in the similar vein, but with several important distinctions—after the optical flow is calculated in the image, the higher-level processing is done on the sphere and vector discrimination is performed analytically as opposed to by iteratively projecting points from different heights.

More concretely, we calculate the optical flow in the image using the sparse iterative version of the Lucas-Kanade algorithm in pyramids [202] implemented in the OpenCV library [185]. Furthermore, both the initial point (feature position in the previous frame) and the terminal point (feature position in the current frame) of the optical flow vector are lifted to the unit sphere for further processing. An extension and improvement would be to calculate also the optical flow and the low-level image processing on the unit sphere since it has been shown in [203, 204] to yield better results than operators derived for perspective images.

5.3.1 *Unit sphere-based motion detection*

With the optical flow calculated, we need to devise a procedure for finding the optical flow vectors caused by the moving objects. Consider Fig. 5.2 where we have depicted a sphere with $\mathcal{F}_p : (x_p, y_p, z_p)$ coordinate system in the origin—representing the image in the previous frame, henceforth referred to as the previous sphere—and a second sphere with $\mathcal{F}_c : (x_c, y_c, z_c)$ coordinate system in the origin—representing the image in the current frame, henceforth referred to as the current sphere. We assume that the displacement between the previous and the current sphere, i.e. $\mathcal{F}_p$ and $\mathcal{F}_c$, is known (in practice calculated from odometry measurements) and described by ${}^c\boldsymbol{R}_p$ and ${}^c\boldsymbol{t}_p$ accounting for the rotation and the translation, respectively. Furthermore, the point ${}^p\mathbf{P}$ in $\mathcal{F}_p$ represents a lifted point detected in the previous image whose matched point in the current image, the lifted point ${}^c\mathbf{P}_m$ in $\mathcal{F}_c$, has been determined by the optical flow algorithm. To determine whether this optical flow vector was induced by the moving object or the ego-motion, we will first

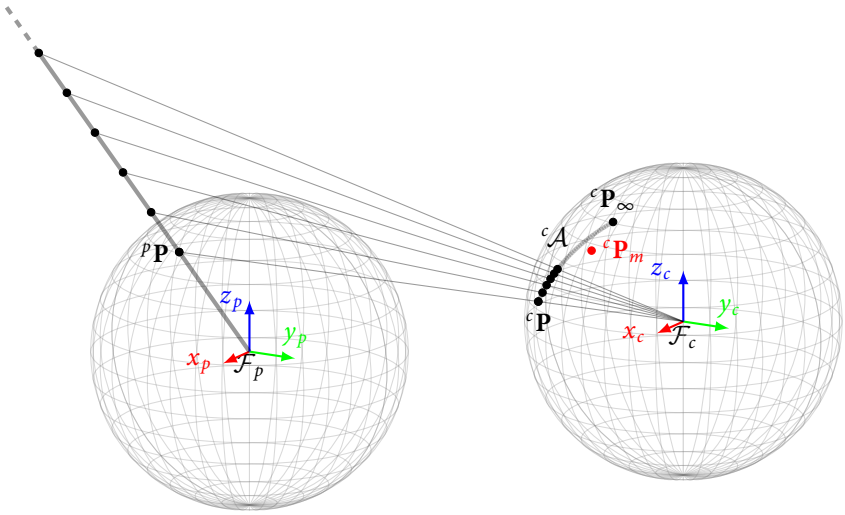


Figure 5.2: Hypothetical location on the sphere in the current frame of the feature on the sphere in the previous frame

hypothesize that the flow was due to ego-motion, and then if the condition is not met we will classify it as being caused by the moving object.

In order to achieve this task, we need to know where to expect a static feature from the previous sphere, like ${}^{p}\mathbf{P}$, on the current sphere (of course without any information about the depth of the feature). By looking at Fig. 5.2 we can assert that the point ${}^{p}\mathbf{P}$ is projection of a feature in the environment somewhere along the optical ray defined by the previous sphere's origin of $\mathcal{F}_p$ and the point ${}^{p}\mathbf{P}$. Furthermore, projection of the point ${}^{p}\mathbf{P}$ onto the current sphere is the point ${}^{c}\mathbf{P}$, and if we continued along the ray in $\mathcal{F}_p$ we can see in Fig. 5.2 where the points would project to on the current sphere. The point on the ray in the infinity projects to ${}^{c}\mathbf{P}_{\infty} = {}^{c}\boldsymbol{R}_{p}{}^{p}\mathbf{P}$, i.e. as if the point ${}^{p}\mathbf{P}$ did not move at all except for the rotation. Given the previous analysis we can conclude that a projection of a point like ${}^{p}\mathbf{P}$, representing a static feature on the previous sphere, should theoretically lie somewhere along the arc ${}^{c}\mathcal{A}$ of the great circle ${}^{c}\mathcal{C}$ [1] defined by points ${}^{c}\mathbf{P}$ and ${}^{c}\mathbf{P}_{\infty}$. To conclude, we will classify an optical flow vector as induced by ego-motion if its matched point on the current sphere ${}^{c}\mathbf{P}_{m}$ lies close to the aforementioned great circle arc ${}^{c}\mathcal{A}$. Naturally, this approach cannot detect objects moving along the optical ray, but this is an unlikely event since it is not possible to ensure such a scenario for all points belonging to a rigid object.

In spherical geometry the closest distance between two points on the sphere is the so-called great circle distance and for unit spheres it can be directly calculated as

$$d({}^{c}\mathbf{P}, {}^{c}\mathbf{P}_{\infty}) = \arccos({}^{c}\mathbf{P} \cdot {}^{c}\mathbf{P}_{\infty}), \tag{5.4}$$

where $(\cdot)$ represents the scalar product. Equation (5.4) is simply the angle between the two unit vectors and incidentally the length of the arc ${}^{c}\mathcal{A}$ in Fig. 5.2. In order to calculate the distance of ${}^{c}\mathbf{P}_{m}$ to ${}^{c}\mathcal{A}$, we first need to determine a point ${}^{c}\mathbf{Q}_{m}$ on the great circle ${}^{c}\mathcal{C}$ which is closest to ${}^{c}\mathbf{P}_{m}$ [205]. We solve this by projecting ${}^{c}\mathbf{P}_{m}$ to the plane defined by ${}^{c}\mathbf{P}$ and ${}^{c}\mathbf{P}_{\infty}$

[1] Intersection of the sphere and a plane which passes through the center point of the sphere

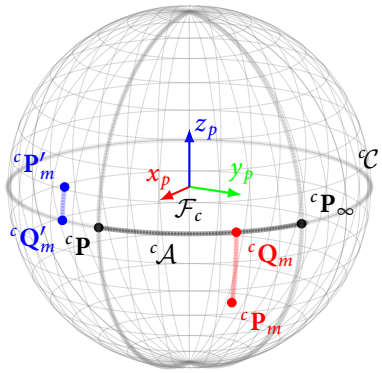


Figure 5.3: Projection of points ${}^c\mathbf{P}_m$ and ${}^c\mathbf{P}'_m$ to the great circle ${}^c\mathcal{C}$ and checking if they lie in the lune of the great arc ${}^c\mathcal{A}$

and then normalizing it to obtain a unit vector

$$\mathbf{P}' = {}^c\mathbf{P}_m - ({}^c\mathbf{P}_m \cdot \mathbf{n})\,\mathbf{n}$$
$${}^c\mathbf{Q}_m = \frac{\mathbf{P}'}{|\mathbf{P}'|},$$

where $\mathbf{n} = {}^c\mathbf{P} \times {}^c\mathbf{P}_\infty$. At this stage we have two possible positions of the point ${}^c\mathbf{Q}_m$: it either lies on ${}^c\mathcal{A}$, or outside of it but on ${}^c\mathcal{C}$ (points ${}^c\mathbf{Q}_m$ and ${}^c\mathbf{Q}'_m$, respectively, in Fig 5.3). The former case is true if the point ${}^c\mathbf{P}_m$ lies in the lune[2] of ${}^c\mathcal{A}$ which we verify by testing the following condition [205]

$$\begin{aligned} ({}^c\mathbf{P} \times {}^c\mathbf{Q}_m) \cdot ({}^c\mathbf{Q}_m \times {}^c\mathbf{P}_\infty) > 0 \quad \text{and} \\ ({}^c\mathbf{P} \times {}^c\mathbf{Q}_m) \cdot ({}^c\mathbf{P} \times {}^c\mathbf{P}_\infty) > 0, \end{aligned} \tag{5.5}$$

where $(\times)$ represents the vector product. Thus if ${}^c\mathbf{Q}_m$ lies on ${}^c\mathcal{A}$ the distance of the point ${}^c\mathbf{P}_m$ to the arc ${}^c\mathcal{A}$ is calculated as $d({}^c\mathbf{P}_m, {}^c\mathbf{Q}_m)$, otherwise as $\min\{d({}^c\mathbf{P}_m, {}^c\mathbf{P}), d({}^c\mathbf{P}_m, {}^c\mathbf{P}_\infty)\}$. If the robot does not move or just rotates then condition (5.5) is evaluated as false and $d({}^c\mathbf{P}, {}^c\mathbf{P}_m)$ is calculated.

The detection performance depends strongly on the measured displacement between two consecutive images. In the thesis we have utilized wheels' odometry for the task, but this can be further refined by fusion with other sensors, like inertial measurement unit, or by using the laser range finder and estimating the displacement by scan matching, localization, or relying purely on the image and use visual odometry.

Once we have selected optical flow vectors that we consider to be caused by moving objects, we still need to make sense of that particular set. There still may exist vectors that are segmented out due to errors in the optical flow calculation or erroneous discrimination caused by odometry based displacement calculation. For example, if we have a single vector classified as static, while most of its neighbors are classified as caused by a moving object, we can safely assume that it is an error. The other extreme case is a vector coming from a moving object while its neighbors are classified as static—in this case more often than not

[2] Area on a sphere bounded by two half great circles. In Fig. 5.3 the two great circle passing through ${}^c\mathbf{P}$ and ${}^c\mathbf{P}_\infty$.

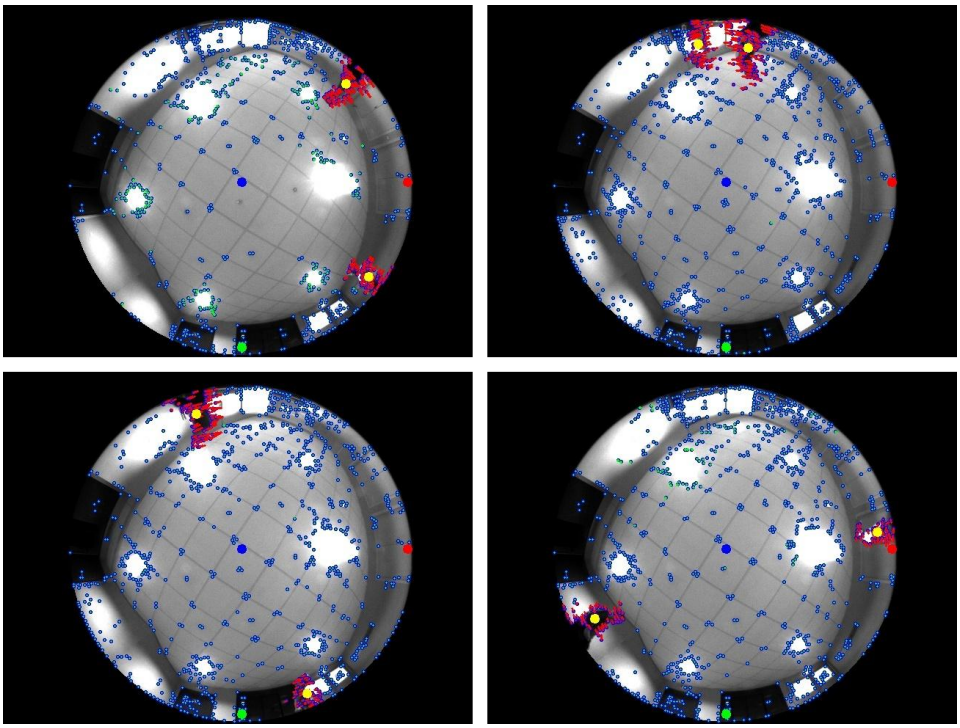


Figure 5.4: Snapshots of the detection experiment—two objects circling around the static robot. Upper left image is the earliest in time, while the lower right is the latest in time. Blue dots represent detected features while the green lines represent the optical flow from ego-motion and red lines represent the optical flow caused by moving objects.

the vectors have a significantly larger modulo than its neighbors, thus most likely being a bad optical flow calculation.

After the aforementioned filtering stage, the vectors are partitioned in equivalence classes using disjoint set data structure and union find algorithm. To partition the vectors we need a predicate to tell us if the two vectors are in the same group or not. We state that two vectors belong to the same group if they have similar modulo, elevation and azimuth (note that the vectors are compared after being lifted to the sphere). Afterwards, by examining the sets, all of them having number of vectors less or equal to 10 are considered insignificant and are removed from further consideration. The remaining groups of vectors are treated as representing moving objects in the scene and their center of gravity is calculated. This center is then a vector on the unit sphere which we henceforth treat as our sensor measurement. Several snapshots of the detection experiments can be seen in Fig. 5.4 and Fig. 5.5. Camera's coordinate system rotated for $\pi/2$ from the robot's system is depicted by red, green and blue points which represent the projections of the tips of the coordinate axes in the sphere. The yellow point represents the center of gravity of the clustered flow vectors. The next question that we address in the subsequent section is how to track, i.e. filter, moving objects on a unit sphere?

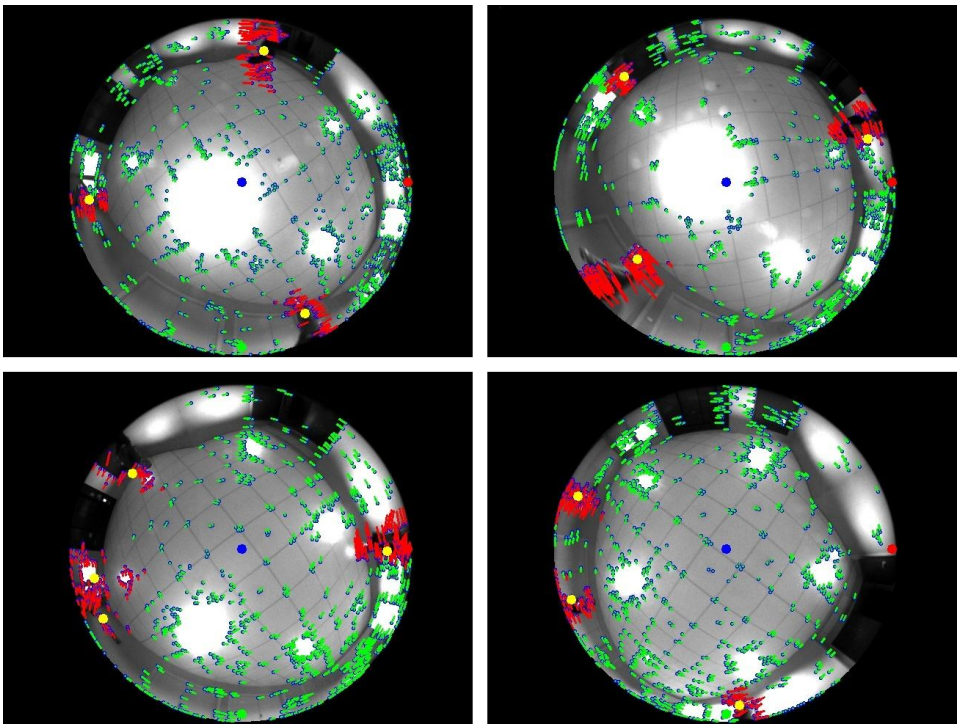


Figure 5.5: Snapshots of the detection experiment—three objects circling around the moving robot. Upper left image is the earliest in time, while the lower right is the latest in time. We can see an outlier in the third image in time where a group of flow vectors was wrongly classified as dynamic and a fourth cluster was created.

5.4 TRACKING ON THE UNIT SPHERE

At this stage we are working with vectors on the unit sphere which represent the direction of the detected moving objects. We propose at this point to advance by statistically modeling the measured direction, i.e. to pose a probabilistic model of the sensor measurement, by using the VMF distribution presented in Section 2.3.2. Here we restate the PDF for completeness

$$p(\boldsymbol{x};\boldsymbol{\mu},\kappa) = \frac{\kappa}{4\pi \sinh \kappa} \exp\left(\kappa \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{x}\right), \tag{5.6}$$

where $\boldsymbol{\mu}$ is the mean direction and κ is the concentration parameter.

Furthermore, by having probabilistically modeled the measurement, we proceed further by posing the problem as an estimation on a sphere thus devising a Bayesian state estimator (tracker) based solely on the VMF distribution [206]. A Bayesian estimation procedure of the a posteriori PDF consists of two steps: prediction and update [72], which in this case entails representing the state to be estimated $\boldsymbol{x}_t$ at time t as the VMF distribution and successively predicting and updating this distribution. As presented in Section 2.4 the prediction step involves calculating the PDF via the total probability theorem (2.30). In this case we do not have a strict state evolution model, but we choose to add process noise governed by a centered VMF in the prediction stage which amounts to convolving our posterior at time $t-1$ with the VMF distribution representing the process noise. Given two VMF distributions, $p(\boldsymbol{x};\boldsymbol{\mu},\kappa_i)$ and $p(\boldsymbol{x};\boldsymbol{\mu},\kappa_j)$, the result of the convolution does not produce another VMF distribution. However, the result of this operation can be well approximated by a VMF with

a suitably chosen value of the resulting κ [46]

$$\kappa_{ij} = A^{-1}(A(\kappa_i)A(\kappa_j)), \quad A(\kappa) = \frac{1}{\tanh\kappa} - \frac{1}{\kappa}. \tag{5.7}$$

Consequently, after the prediction step our state represented by a single VMF will have unchanged mean direction but newly calculated concentration parameter via (5.7).

In the update step, the posterior at time t is calculated via the Bayes theorem (2.28). In our case the sensor model $p(\boldsymbol{z}_t \mid \boldsymbol{x}_t)$ will be represented by a VMF as discussed at the beginning of the current section, while the predicted state $p(\boldsymbol{x}_t \mid \mathbf{z}_{1:t-1})$ will be the result of the previously discussed convolution. Given two VMF distributions, $p(\boldsymbol{x}; \boldsymbol{\mu}_i, \kappa_i)$ and $p(\boldsymbol{x}; \boldsymbol{\mu}_j, \kappa_j)$, the result of the update step is a VMF with the following parameters (see Appendix A.5) [206]

$$\begin{aligned} \kappa_{ij} &= \sqrt{\kappa_i^2 + \kappa_j^2 + 2\kappa_i\kappa_j(\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j)} \\ \boldsymbol{\mu}_{ij} &= \frac{\kappa_i\boldsymbol{\mu}_i + \kappa_j\boldsymbol{\mu}_j}{\kappa_{ij}}. \end{aligned} \tag{5.8}$$

These two steps, governed by (5.7) and (5.8), will cyclically produce the estimate of the direction of the moving object. Methods for practical calculation of some of the aforementioned equations can be found in [66].

In this chapter we focus on tracking a single object and if there are multiple moving objects detected, then only the closest measurement is considered in the update step. Fig. 5.6 depicts the measured and estimated direction azimuth and elevation of the moving object from an experiment in which an object circled around the mobile robot. From the figure we can see that the elevation measurements are more noisy than the azimuth measurements, and that, nevertheless, the filter manages to smoothly track the moving object. This is important since we need smooth estimates for the control task. Future research will aim at exploring tracking of all the detected moving objects which would involve solving the data association and the track management problem [76].

5.5 Following via visual servoing

Having solutions for moving object detection and tracking, we can now advance to solving the problem of object following. The idea is to keep the tracked moving object at the specific (user-defined) location in the omnidirectional image. In this thesis we have utilized a differential drive mobile robot where the linear and the angular velocity are controlled, hence our task at hand is to calculate the control law that will drive the error between the desired and the estimated direction to zero. To solve this problem we propose to utilize a visual servoing technique based on projection of a point on the unit sphere.

Visual servoing refers to the use of computer vision data to control the motion of a robot [197]. The vision data may be acquired from a camera that is mounted directly on a robot manipulator or on a mobile robot, in which case motion of the robot induces camera motion. The goal of vision-based control systems is to minimize an error $\boldsymbol{e}(t)$ defined as

$$\boldsymbol{e}(t) = \boldsymbol{s} - \boldsymbol{s}^* \tag{5.9}$$

where $\boldsymbol{s}$ represent a set of k measured visual features and $\boldsymbol{s}^*$ are the desired value of the features. Usually, in visual servoing the control law is designed as a velocity controller

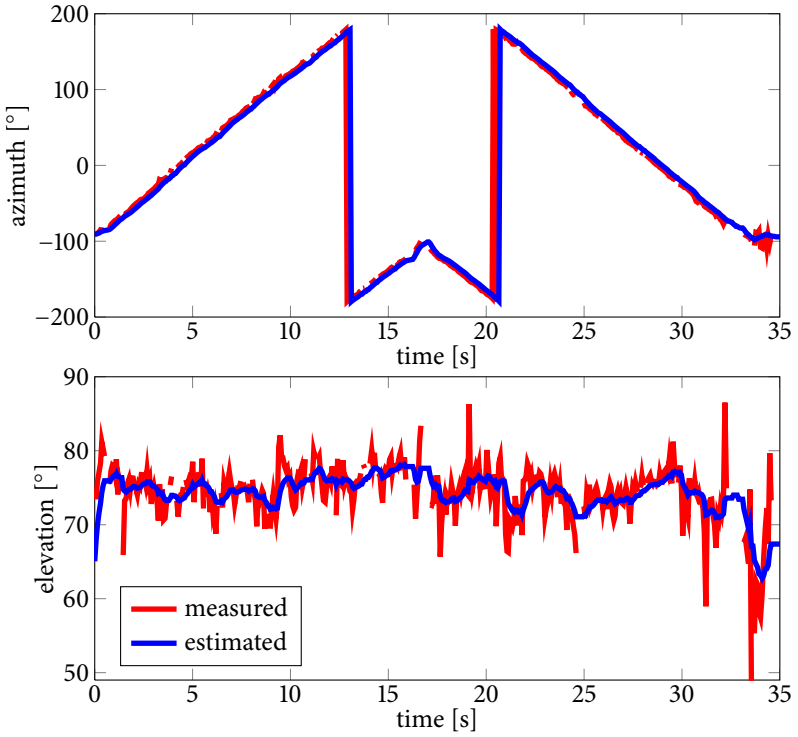


Figure 5.6: Measured and estimated azimuth and elevation of the moving object direction

(we assume motion control with six degrees of freedom) which necessitates a relationship between time variation of $\boldsymbol{s}$ and the camera velocity. If we denote the camera velocity as $\boldsymbol{v} = (v, \omega)$, where v is the linear velocity and ω is the angular velocity of the camera frame, then the sought relationship is given by

$$\dot{\boldsymbol{s}} = \boldsymbol{L}_s \boldsymbol{v}, \tag{5.10}$$

where $\boldsymbol{L}_s \in \mathbb{R}^{k\times 6}$ is called the interaction matrix. If we consider $\boldsymbol{v}$ as an input to the robot controller, our control law can then be calculated as follows [197]

$$\boldsymbol{v} = -\lambda \boldsymbol{L}_s^{\dagger} \boldsymbol{e}, \tag{5.11}$$

where $\boldsymbol{L}_s^{\dagger} \in \mathbb{R}^{6\times k}$ is the Moore-Penrose inverse of the matrix $\boldsymbol{L}_s$ and λ is a positive gain. In practice the interaction matrix or its inverse are often approximated or estimated, thus the control law becomes in fact

$$\boldsymbol{v} = -\lambda \widehat{\boldsymbol{L}}_s^{\dagger} \boldsymbol{e} = -\lambda \widehat{\boldsymbol{L}}_s^{\dagger} (\boldsymbol{s} - \boldsymbol{s}^*). \tag{5.12}$$

The feature $\boldsymbol{s}$ can be designed in numerous ways [197], but in this thesis we utilize a projection of a world point to the omnidirectional image, thus making this an image-based visual servo problem.

To represent the visual feature $\mathbf{s}$ we use a cylindrical coordinate system in the spherical image to represent the projection of the estimated direction of the moving object

$$\rho = \sqrt{s_x^2 + s_y^2}, \quad \theta = \arctan \frac{s_y}{s_x}. \tag{5.13}$$

For this case the relationship between $\dot{\boldsymbol{s}}$ and $\boldsymbol{v}$, i.e. the interaction matrix, is given by [207]

$$\boldsymbol{L}_s = \begin{bmatrix} \frac{-\cos\theta}{P_z} & \frac{-\sin\theta}{P_z} & \frac{\rho}{P_z} & (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \\ \frac{\sin\theta}{\rho P_z} & \frac{-\cos\theta}{\rho P_z} & 0 & \frac{\cos\theta}{\rho} & \frac{\sin\theta}{\rho} & -1 \end{bmatrix}, \tag{5.14}$$

where P_z is the z coordinate of the moving object (not on the sphere but in the environment).

For differential drive robots the convention is to set the robot's coordinate system such that the linear velocity v is in the positive direction of the x axis, while the angular velocity ω is defined positive counter-clockwise with respect to the z axis. Thus for this case we need to choose the appropriate column of (5.14) which yields interaction matrix of the following form

$$\boldsymbol{L}_s = \begin{bmatrix} \frac{-\cos\theta}{P_z} & 0 \\ \frac{\sin\theta}{\rho P_z} & -1 \end{bmatrix}. \tag{5.15}$$

Then, the calculation of the control law proceeds as follows [197]

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = -\lambda \widehat{\boldsymbol{L}}_s^{\dagger} \begin{bmatrix} \rho - \rho^* \\ \theta - \theta^* \end{bmatrix}, \tag{5.16}$$

where ρ^* and θ^* are the desired values and $\widehat{\boldsymbol{L}}_s^{\dagger}$ is the pseudoinverse of the estimated interaction matrix—the coordinate P_z is not known and in this chapter we set it to an arbitrary value. The gain λ is adaptively calculated according to the following law

$$\lambda(e(t)) = a\exp(-be(t)) + c, \tag{5.17}$$

where $e(t)$ is the error of the control task (calculated as the great circle distance between the estimated and the desired position), $a = \lambda(0) - \lambda(\infty), b = \lambda'(0)/a, c = \lambda(\infty)$ with $\lambda(0) = 0.5, \lambda(\infty) = 0.05, \lambda'(0) = 0.5$.

In [208] a visual servoing control law in spherical coordinates for omnidirectional images was presented. We have noticed that for our specific task with the differential drive robot it exhibits an additional singularity in the control law. Namely, in the cylindrical coordinate system as in [207] the control law has a singularity in $\rho = 0$—estimated direction in the middle of the image where bearing is undefined (in our case practically unlikely)—and $\theta = \pm\pi/2$—the values of the bearing (possible in our case if the estimated direction gets too far away from the desired one; in that case we saturate the control signal to a reasonable maximal value). However, the control law in spherical coordinates as in [208] has additional singularity when the elevation of the estimated direction is equal to $\pm\pi/2$—the vector lies on the sphere's equator (fairly often in our case). This was the reasoning due to which we chose to work in the cylindrical coordinate system.

5.5.1 *Experiments*

A snapshot of the experiment is shown in Fig. 5.7. The violet point is the desired direction of the moving object, while the magenta point represents the estimated direction of the

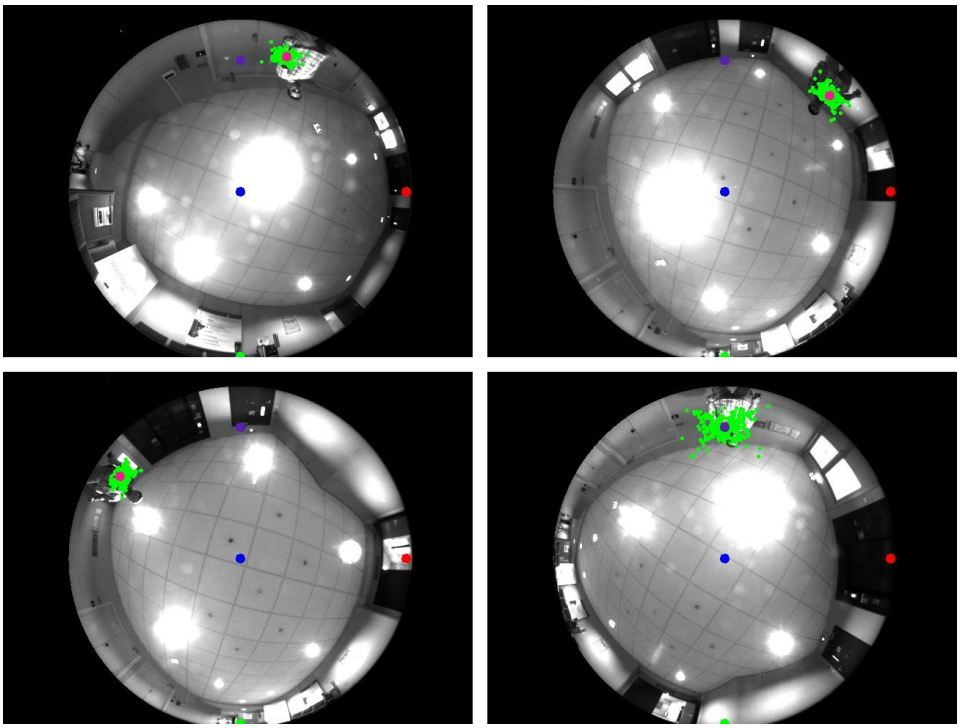


Figure 5.7: Snapshots of the experiment—an object moving away from the desired direction. Upper left image is the earliest in time, while the lower right is the latest in time.

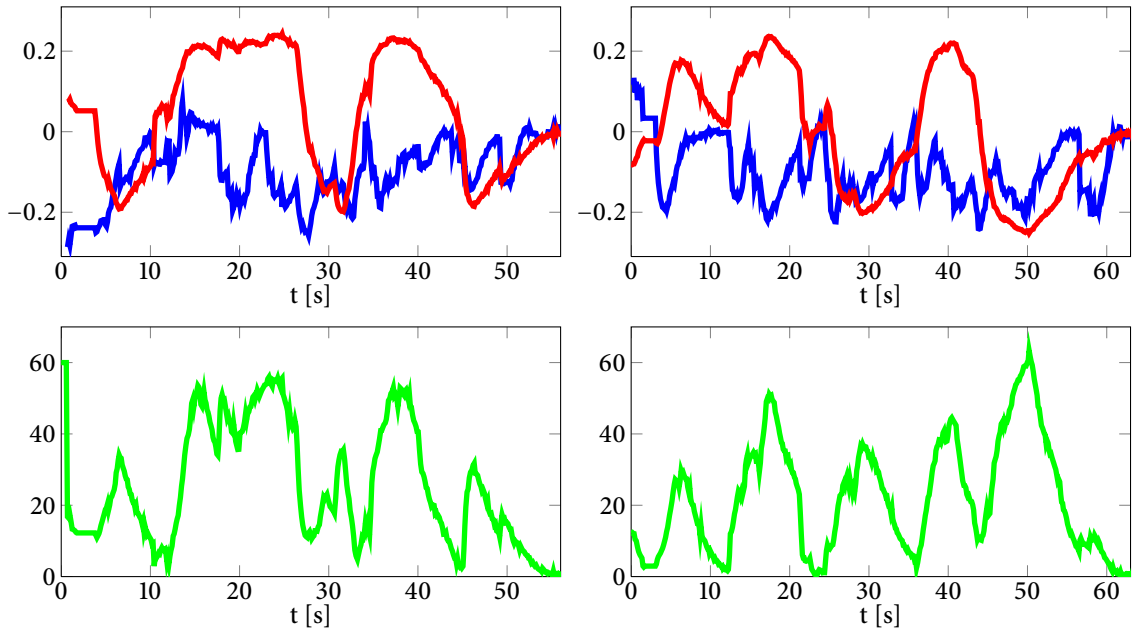


Figure 5.8: Command velocities and error (great circle distance from the desired to the estimated direction)

moving object. The green points surrounding the estimated direction are samples from the a posteriori VMF representing the current state of the object.

In Fig. 5.8 we have depicted for two experiments the linear and angular velocity commands along with the visual servoing task error, which was calculated as the great circle distance between the desired and the estimated direction of the object via (5.4). The object

was moving in such way as to first distance itself from the desired position and then waited until the robot closed the distance by reducing the servoing task error to zero. This motion pattern was repeated several times during the experiments and the result can be clearly seen in the errors depicted in Fig. 5.8. If there were other moving objects in the scene, their measurements were regarded as false alarms and only the closest measurement to the estimated direction was taken into account. Concerning the control velocities we can see that the angular velocity followed closely the behavior of the error—when the error is greatest so is the velocity command, sometimes changing the sign of the command (when the robot would correct for the error the object would move away in the direction from which the robot came) and sometimes keeping the same sign (when the robot would correct the error coming from one direction the object would move away in the opposite). The linear velocity command appears to be more noisy than its angular counterpart. This is due to the fact that our visual servo control law corrects the error based on the projection of a single point—the center of gravity of the clustered optical flow vector. We have no information about the shape nor the height of the object which makes it difficult to guarantee that the robot would position itself relative to the object at a specific distance. However, in practice during the experiments we have noticed that most often the tracked center of gravity of the segmented cluster would not deviate much thus making it possible to define a point in the image for which the robot would reasonably close the distance to the object. Naturally, none of the aforementioned problems were experienced with the orientation.

5.6 SUMMARY

In this chapter we have presented a method based on processing on the unit sphere for moving object detection, tracking and following with an omnidirectional camera mounted on a mobile robot. The spherical projection model coupled with displacement information from motor encoders was used to segment out vectors that do not belong to the static scene around the mobile robot. This was achieved by calculating the great circle distance of the terminal point of the optical flow vector on the sphere to an arc representing the hypothetical locations of the projection of the initial point of the optical flow vector. With movement segmented in the image, clusters of similar optical flow vectors were created based on their similarity in the magnitude, azimuth and elevation angles. With the moving object segmented out on the sphere, its center of gravity was probabilistically structured in order to be included in the tracking framework based on the Bayesian estimation on the sphere with the von Mises-Fisher distribution. This procedure entailed calculating the convolution and product of two von Mises-Fisher distributions where the resulting mean direction was considered as the estimated direction of the moving object. Given the estimated position a visual servo control law based on a projection of a world point to the sphere was calculated which in turn made the robot follow the moving object. Experimental results obtained with a camera and fish-eye lens mounted on a differential drive platform were presented and discussed.

6

Sensor fusion for object tracking

The problem of Bayesian sensor fusion for moving object tracking is studied in this chapter. The prospects of utilizing measurements from several sensors to infer about a system state are manyfold, from increased estimate accuracy to more reliable and robust estimates due to several sensors measurements of the same phenomenon, possibly, based on different features. Sensor measurements may be combined, or fused, at a variety of levels; from the raw data level to the state vector level, or at the decision level. In this chapter we mainly focus on the Bayesian fusion at the likelihood and state vector level. Firstly, we analyze two groups of data fusion methods: centralized independent likelihood fusion, where the sensors report only their measurements to the fusion center, and hierarchical fusion, where each sensor runs its own local estimate which is then communicated to the fusion center along with the corresponding uncertainty. The analysis is performed for a single moving object scenario and we compare the prospects of utilizing both approaches, and present explicit solutions in the forms of extended information filter, unscented information filter and particle filter. Furthermore, we also propose a solution for fusion of arbitrary filters and test it on a hierarchical fusion example of the extended information and the particle filter. The methods are tested on a synthetic data experiment of tracking a dynamic object with several sensors of different accuracies by analyzing the quadratic Rényi entropy and root-mean-square error.

Secondly, we study the problem of tracking an arbitrary number of people with multiple heterogeneous sensors from a mobile robot. To solve the data association problem, instead of using the optimal multiple-hypothesis tracking with complex hypothesis branching, we choose the computationally simpler joint probabilistic data association filter since we are interested only in local observations by a mobile robot for people detection, tracking, and avoidance. However, the joint probabilistic data association filter assumes a constant and known number of objects in the scene, and therefore, we use an entropy based track management scheme. The benefits of the approach are that all the required data come from a running filter, and that it can be readily utilized for an arbitrary type of filter, as long as such a strong mathematical principle like entropy is tractable for the underlying distribution. The algorithm is implemented for the case of the Kalman filter, and the performance is verified in experiments where we used a laser range sensor, a microphone array and an RGB-D camera.

6.1 INTRODUCTION

The prospects of utilizing measurements from several sensors to infer about a system state are manyfold. To begin with, the use of multiple sensors results in increased sensor measurement accuracy, and moreover, additional sensors will never reduce the performance of the optimal estimator [209]. However, in order to ensure this performance, special care must be taken when choosing the process model [210]. Furthermore, system reliability increases with additional sensors, since the system itself becomes more resilient to sensor failure [211]. Therefore, by combining data from multiple sensors, and perhaps related information from associated databases, we can achieve improved accuracies and more specific inferences than using only a single sensor [212, 213].

Sensor measurements may be combined, or fused, at a variety of levels; from the raw data level to a state vector level, or at the decision level [212]. Raw sensor data can be directly combined if the sensor data are commensurate (i.e., if the sensors are measuring the same physical phenomena), while if the sensor data are noncommensurate, then the sensor data, i.e. sensor information, must be fused at a feature/state vector level or decision level.

Information from multiple sensors can be classified as redundant or complementary [211]. *Redundant information* is provided from multiple sensors (or a single sensor over time) when each sensor perceives the same feature in the environment. On the other hand, *complementary information* from multiple sensors enables the system to perceive features impossible to perceive by using just a single sensor. But what is in common for both classifications, is that all the sensors are used to somehow infer about a system state. It is important to note that complementary sensors do not have to necessarily provide information about the full system state. Some sensors, like omnidirectional cameras and microphone arrays, measure angle and not the range of the detected objects, while laser range scanners and depth cameras can give measurements in 2D or 3D. Moreover, some sensors can provide measurements at higher rates than others, thus making sensor fusion an even more challenging problem.

One way of approaching the problem of sensor fusion is at the likelihood level. Basically, each sensor measurement is modeled as a Gaussian random variable and the resulting fused distribution is also Gaussian with the new fused mean and covariance. In [214], the fused moments are calculated by optimizing a weighted sum of Gaussian random variables so as to minimize the volume of the fused uncertainty ellipsoid. The resulting moments are equal to as if they were obtained by calculating the product of Gaussian distributions. Similar results were obtained in [215] where the fused moments are calculated by estimating the moments of a product of Gaussians via maximum likelihood approach. Both of these methods do not take any past measurements into account, and if tracking is needed then different approach needs to be utilized.

If the system is linear and the system state is modeled as Gaussian, then multisensor fusion can be performed with the decentralised Kalman filter (DKF) proposed in [216]. The DKF enables us to fuse not only the measurements, but also the local independent Kalman filters. The inverse covariance form is utilized, thus resulting in additive fusion equations, which can further be elegantly translated to the information filter form as shown in [217]. For the case of non-linear systems the extended information filter (EIF) or the

unscented information filter (UIF) [218] can be utilized. Another approach, proposed in [219], is to define for each sensor system, a separate and specific Gaussian probability distribution and to fuse them using covariance intersection method [220]. If the underlying distribution characterizing the system is not Gaussian and possibly non-linear, then usually PF are utilized. In [143] a distributed particle filtering algorithm is proposed where each sensor maintains a particle filter and the information is propagated in sensor network in the form of partial likelihood functions. The last sensor then back-propagates the final importance distribution so that a new set of particles is generated at each sensor using the final distribution. The standard particle filter algorithm was decentralized in [221] by communicating and fusing only the most informative subsets of samples. It was applied on mobile robots playing the game of laser tag. In [222] a speaker tracking system was implemented by using a camera and a microphone array. Each sensor estimate was modeled as a Gaussian distribution in order to obtain overall likelihood function. The fusion was performed by a global particle filter which used the sum of the former Gaussians as the proposal distribution and their product as the likelihood function for calculating the weights of particles.

In order to perform fusion between decentralized tracking filters, we have to take into account the common information that the distributions might share. This usually entails a product and a division of particle sets and a solution for consistent fusion was proposed in [223, 224]. An overview of decentralized fusion methods and non-Gaussian estimation techniques can be found in [21, 225]. In [139] we implicitly used centralized independent likelihood fusion via joint probabilistic data association filter in the problem of multi-target tracking with multiple sensors on a mobile robot and some of the results are presented also in this chapter.

6.2 BAYESIAN SENSOR FUSION

The goal of the sensor fusion is to estimate the system state $\boldsymbol{x}_t$ at time t based on all previous control inputs $\boldsymbol{u}_{1:t}$, and all previous sensor measurements from all the m available sensors $\boldsymbol{z}_{1:t}^{1:m}$. Note that the problem differs from previous discussions that we now have multiple sensors indicated by the superscript $(\,.\,)^{1:m}$. In other words, from a probabilistic perspective, we need to estimate the posterior distribution $p(\boldsymbol{x}_t \,|\, \boldsymbol{u}_{1:t}, \boldsymbol{z}_{1:t}^1, \boldsymbol{z}_{1:t}^2, \ldots, \boldsymbol{z}_{1:t}^m) = p(\boldsymbol{x}_t \,|\, \boldsymbol{u}_{1:t}, \boldsymbol{z}_{1:t}^{1:m})$. By applying the Bayes theorem, we can reformulate the problem as follows (for convenience we drop the condition on $\boldsymbol{u}_{1:t}$ since in tracking this is usually not known) [225]

$$\begin{aligned} p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}^{1:m}) &= p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_t^{1:m}, \boldsymbol{z}_{1:t-1}^{1:m}) \\ &= \frac{p(\boldsymbol{z}_t^{1:m} \,|\, \boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}^{1:m}) p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m})}{p(\boldsymbol{z}_t^{1:m} \,|\, \boldsymbol{z}_{1:t-1}^{1:m})}. \end{aligned} \tag{6.1}$$

Furthermore, we assume that (i) given the state $\boldsymbol{x}_t$ the measurement at the i^{th} sensor is independent of the measurements obtained from other sensors, and (ii) that the current state $\boldsymbol{x}_t$ includes all the required information to evaluate the likelihood meaning that we can drop the conditional dependency of the current measurement of the i^{th} sensor $\boldsymbol{z}_t^i$ on all

the previous measurements of all the sensors $\boldsymbol{z}_{1:t-1}^{1:m}$

$$p(\boldsymbol{z}_t^{1:m} \,|\, \boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}^{1:m}) = \prod_{i=1}^{m} p(\boldsymbol{z}_t^i \mid \boldsymbol{x}_t, \boldsymbol{z}_{1:t-1}^{1:m}) = \prod_{i=1}^{m} p(\boldsymbol{z}_t^i \mid \boldsymbol{x}_t). \tag{6.2}$$

At this point, we can proceed further in three different directions: (i) centralized independent likelihood fusion, (ii) hierarchical fusion without feedback and (iii) hierarchical fusion with feedback. If each sensor reports only its measurement modeled in a probabilistic manner, i.e. likelihood or the sensor model, then this leads us to the first solution, in which we have a global estimate of the system state updated by fusing only the likelihoods communicated from each sensor

$$p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}^{1:m}) \propto p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m}) \prod_{i=1}^{m} p(\boldsymbol{z}_t^i \mid \boldsymbol{x}_t), \tag{6.3}$$

where $p(\boldsymbol{z}_t^{1:m} \,|\, \boldsymbol{z}_{1:t-1}^{1:m})$ is omitted since it only accounts for the normalization of the calculated posterior. This is an example of centralized independent likelihood fusion.

Now, the second solution amounts to each sensor modality estimating its own local system state based only on its local observations. These local posterior estimates are then fused on a global level. Since all sensors operate without having any knowledge of other sensor measurements, at each sensor i we have $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}^i)$ as the local posterior. By inspecting (6.3) we can see that we need to 'extract' the likelihood $p(\boldsymbol{z}_t^i \,|\, \boldsymbol{x}_t)$ from the local posterior. By using a similar procedure as in (6.1) we can derive the expression for the needed likelihood

$$p(\boldsymbol{z}_t^i \,|\, \boldsymbol{x}_t) \propto \frac{p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}^i)}{p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^i)}. \tag{6.4}$$

This leads us to the following expression

$$p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}^{1:m}) \propto p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m}) \prod_{i=1}^{m} \frac{p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}^i)}{p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^i)}. \tag{6.5}$$

This is an example of hierarchical fusion without feedback which suggests that if we want to fuse a global prediction based on all the sensors $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m})$ with local independent sensor posteriors $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}^i)$, we need to first remove the local prediction $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^i)$, i.e. the local prior knowledge, by a division. This is logical since we already have all the prior knowledge in the global prediction $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m})$ and are only interested in acquiring new knowledge arising from new measurements. If the local predictions $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^i)$ shared common or very similar prior information which was not removed during the fusion, each of them would implicitly count through $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}^i)$ with each multiplication, thus resulting in a posterior being too confident, or swayed, by all the multiply-counted prior information.

For the third solution we have the global prediction based on all the measurements communicated back to each sensor i to serve as a new local prior which will then be updated only with the local measurement $\boldsymbol{z}_t^i$. Therefore, at each sensor i we have $p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m}, \boldsymbol{z}_t^i)$ as the local posterior from which we will need to 'extract' the likelihood $p(\boldsymbol{z}_t^i \,|\, \boldsymbol{x}_t)$. Again, by following a similar procedure as in (6.1) we calculate the needed expression for the likelihood

$$p(\boldsymbol{z}_t^i \,|\, \boldsymbol{x}_t) \propto \frac{p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m}, \boldsymbol{z}_t^i)}{p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m})}, \tag{6.6}$$

which leads us to the following equation for the *hierarchical fusion with feedback*

$$p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t}^{1:m}) \propto p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m}) \prod_{i=1}^{m} \frac{p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m}, \boldsymbol{z}_t^i)}{p(\boldsymbol{x}_t \,|\, \boldsymbol{z}_{1:t-1}^{1:m})}. \tag{6.7}$$

Each approach has its benefits. The centralized independent likelihood fusion is quite elegant since we only need to communicate the likelihoods to the fusion center, thus requiring only that each likelihood represents each sensor measurements faithfully. The hierarchical approach without feedback requires each sensor to run its own local estimate independent of other sensors, while the hierarchical fusion with feedback takes one step further and communicates the global fused posterior back to each sensor to serve as the next prior in the local estimation process. In this way each sensor benefits by having the same global prior, even in situations when the sensor itself has no measurements. This approach is closely related to decentralized systems where we could have several independent agents exchanging estimations in an unstructured or arbitrary network, but without central fusion processor. Although decentralization has many advantages [216, 217], it requires dealing with delayed and asequent observations, and filtering of previously exchanged common information which is a much broader topic and shall not be studied in this thesis since we are concerned with tracking from a single mobile platform. Therefore, we shall concentrate on the centralized independent likelihood fusion and the hierarchical fusion without feedback, since we want to explore the effects of fusion of sensor modalities which share no common information.

6.2.1 *Kinematics and state space equation of the tracked object*

In this chapter we use a fairly general piecewise constant white acceleration model in order to describe the system behavior [76]. The system state is defined as a vector $\boldsymbol{x}_t = [x_t, \dot{x}_t, y_t, \dot{y}_t]$, where (x_t, y_t) are the Cartesian coordinates, while $\dot{x}_t$ and $\dot{y}_t$ represent their respective velocities in the x, y-plane. The model itself is given by[1]

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{A}_t \boldsymbol{x}_{t-1} + \boldsymbol{G}\boldsymbol{v}_t \\ &= \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_{t-1} + \begin{bmatrix} \frac{\Delta T^2}{2} & 0 \\ \Delta T & 0 \\ 0 & \frac{\Delta T^2}{2} \\ 0 & \Delta T \end{bmatrix} \boldsymbol{v}_t, \end{aligned} \tag{6.8}$$

where ΔT is the sampling period, $\boldsymbol{v}_t$ describes the uncertainty in the evolution of the system state with the associated process noise covariance matrix $\boldsymbol{Q}_t$.

For the measurement model, we assume that the sensors measure both range and bearing, thus yielding a non-linear measurement equation

$$\boldsymbol{z}_t = \boldsymbol{h}(\boldsymbol{x}_t) + \boldsymbol{n}_t = \begin{bmatrix} \sqrt{x_t^2 + y_t^2} \\ \arctan\left(\frac{y_t}{x_t}\right) \end{bmatrix} + \boldsymbol{n}_t, \tag{6.9}$$

[1] In this section we are using the piecewise constant velocity model since it enables us to conveniently address the problem of asynchronous data arrival and its effect on calculating the process noise in the prediction step

where $\boldsymbol{z}_t$ is the sensor measurement, $\boldsymbol{n}_t$ describes the uncertainty in the measurement with the associated measurement covariance matrix $\boldsymbol{R}_t$. Naturally, both the process and measurement noise are assumed to be normal, zero-mean, white and themselves uncorrelated as in the Section 2.4.1.

6.3 CENTRALIZED SENSOR FUSION

6.3.1 *Extended information filter*

With transition and observation equations defined with (6.8) and (6.9), respectively, for the Kalman filter the a priori predicted values of the system state and covariance are calculated as follows

$$\boldsymbol{x}_{t|t-1} = \boldsymbol{A}_t \boldsymbol{x}_{t-1|t-1} \tag{6.10}$$

$$\boldsymbol{P}_{t|t-1} = \boldsymbol{A}_t \boldsymbol{P}_{t-1|t-1} \boldsymbol{A}_t^T + \boldsymbol{G}\boldsymbol{Q}_t \boldsymbol{G}^T. \tag{6.11}$$

Instead of continuing with the Kalman filter update equations, we shall now revert to its equivalent information filter form, whose advantages in sensor fusion will become apparent soon.

The information matrix $\boldsymbol{Y}_{t|t}$ and the information vector $\boldsymbol{y}_{t|t}$ are defined as follows [72]

$$\boldsymbol{Y}_{t|t} = \boldsymbol{P}_{t|t}^{-1}, \quad \boldsymbol{y}_{t|t} = \boldsymbol{P}_{t|t}^{-1} \boldsymbol{x}_{t|t}. \tag{6.12}$$

The prediction equations for the information matrix and information vector are then

$$\boldsymbol{Y}_{t|t-1} = \left[\boldsymbol{A}_t \boldsymbol{Y}_{t-1|t-1}^{-1} \boldsymbol{A}_t^{\mathrm{T}} + \boldsymbol{Q}_t\right]^{-1} \tag{6.13}$$

$$\boldsymbol{y}_{t|t-1} = \boldsymbol{Y}_{t|t-1} \left[\boldsymbol{A}_t \boldsymbol{Y}_{t-1|t-1}^{-1} \boldsymbol{y}_{t-1|t-1} + \boldsymbol{B}\boldsymbol{u}_t\right]. \tag{6.14}$$

If we define the information associated with the observation taken at time t as

$$\boldsymbol{i}_t = \boldsymbol{H}_t^{\mathrm{T}} \boldsymbol{R}_t^{-1} \left(\boldsymbol{v}_t + \boldsymbol{H}_t \boldsymbol{x}_{t|t-1}\right), \quad \boldsymbol{I}_t = \boldsymbol{H}_t^{\mathrm{T}} \boldsymbol{R}_t^{-1} \boldsymbol{H}_t, \tag{6.15}$$

where $\boldsymbol{v}_t = \boldsymbol{z}_t - \boldsymbol{h}(\boldsymbol{x}_{t|t-1})$ is the innovation vector and $\boldsymbol{H}_t$ is the observation matrix, we can write the update stage of the information filter as

$$\boldsymbol{y}_{t|t} = \boldsymbol{y}_{t|t-1} + \boldsymbol{i}_t, \quad \boldsymbol{Y}_{t|t} = \boldsymbol{Y}_{t|t-1} + \boldsymbol{I}_t. \tag{6.16}$$

From (6.16) we can see that the update stage of the information filter is additive. In fact, this very property of the information filter is the main reason for its utility in multisensor fusion.

If we have m sensors, then for each sensor i we can define an observation equation

$$\boldsymbol{z}_t^i = \boldsymbol{H}_t^i \boldsymbol{x}_t + \boldsymbol{n}_t^i, \quad i = 1, \ldots, m, \tag{6.17}$$

with the corresponding observation matrix $\boldsymbol{H}_t^i$. Since the measurement model can be linearized about the predicted state vector, the observation matrix may be introduced

$$\boldsymbol{H}_t^i = \left.\frac{\partial \boldsymbol{h}^i(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{t|t-1}}. \tag{6.18}$$

For the measurement model (6.9) the observation matrix takes the following form

$$\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{x}{\sqrt{x^2+y^2}} & 0 & \frac{y}{\sqrt{x^2+y^2}} & 0 \\ -\frac{y}{x^2+y^2} & 0 & \frac{x}{x^2+y^2} & 0 \end{bmatrix}. \tag{6.19}$$

In the standard Kalman filter notation, contributions from multiple sensors cannot be additively combined, since, although the sensor measurements given the system state are themselves independent, the innovations are correlated through common information from the prediction stage in (6.10). However, in the information form, the terms $\boldsymbol{i}_t^i$ from each sensor are uncorrelated, thus resulting with additive update stage with contributions from each sensor [216, 217]

$$\boldsymbol{y}_{t|t} = \boldsymbol{y}_{t|t-1} + \sum_{i=1}^{m} \boldsymbol{i}_t^i \tag{6.20}$$

$$\boldsymbol{Y}_{t|t} = \boldsymbol{Y}_{t|t-1} + \sum_{i=1}^{m} \boldsymbol{I}_t^i, \tag{6.21}$$

where now $\boldsymbol{y}_{t|t}$ and $\boldsymbol{Y}_{t|t}$ represent the central fused information vector and information matrix. The central fused estimate of the system state may be found via $\boldsymbol{x}_{t|t} = \boldsymbol{Y}_{t|t}^{-1}\boldsymbol{y}_{t|t}$.

By inspecting (6.20) and (6.15), we can see that during fusion each sensor measurement is weighted by its corresponding variance. In essence, this approach is similar to the product of Gaussians, except that it does take past values into account through $\mathbf{y}_{t|t-1}$, which we can see from (6.12) that it is just the predicted global system state weighted by the corresponding global predicted variance.

The previous approach to sensor fusion was derived in [216] following the work in [226], and was termed DKF. The main idea was to offer a flexible method for decomposing the linear Kalman filter into autonomous local processors associated with each sensor modality. However, so far we have presented only the means for fusing multiple sensor measurements. If we want to fuse estimates form several running filters each adjoined to a sensor (for which the DKF was initially derived for), then we have to further extend the fusion approach.

6.3.2 *Unscented information filter*

In this section the unscented version [227, 228] of the information filter is utilized for centralized sensor fusion. Unlike EIF which approximates the non-linear function by a Taylor series expansions, the UIF deterministically generates sigma points and uses them to estimate the mean and the covariance. Therefore, for an n dimensional system we need to generate $2n+1$ sigma points $\boldsymbol{\mathcal{X}}_{j,t-1}$ by

$$\begin{aligned} \boldsymbol{\mathcal{X}}_{0,t-1|t-1} &= \boldsymbol{x}_{t-1|t-1} \\ \boldsymbol{\mathcal{X}}_{j,t-1|t-1} &= \boldsymbol{x}_{t-1|t-1} + \left(\sqrt{(n+\lambda)\,\boldsymbol{P}_{t-1|t-1}}\right)_j \\ \boldsymbol{\mathcal{X}}_{j,t-1|t-1} &= \boldsymbol{x}_{t-1|t-1} - \left(\sqrt{(n+\lambda)\,\boldsymbol{P}_{t-1|t-1}}\right)_j, \end{aligned} \tag{6.22}$$

where $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter with $0 \leq \alpha \leq 1$ and κ usually chosen by the heuristic $n + \kappa = 3$, and $\left(\sqrt{(n+\lambda)\,\boldsymbol{P}_{t-1|t-1}}\right)_j$ is the j^{th} column of the square root matrix of the multiplied covariance matrix.

The corresponding weights for recovering the mean and the covariance are calculated as follows

$$\begin{aligned} w_0^{(l)} &= \lambda/(n+\lambda) \\ w_j^{(l)} &= 1/\left[2(n+\lambda)\right] \\ w_0^{(c)} &= \lambda/(n+\lambda) + (1-\alpha^2+\beta) \\ w_j^{(c)} &= 1/\left[2(n+\lambda)\right], \end{aligned} \tag{6.23}$$

where the parameter β is for encoding additional higher order effects. If the underlying distribution is a Gaussian, then $\beta = 2$ is the optimal choice.

The information prediction equations are

$$\boldsymbol{y}_{t|t-1} = \boldsymbol{Y}_{t|t-1} \sum_{j=0}^{2n} w_j^{(l)} \boldsymbol{\mathcal{X}}_{j,t|t-1} \tag{6.24}$$

$$\boldsymbol{Y}_{t|t-1} = \boldsymbol{P}_{t|t-1}^{-1}, \tag{6.25}$$

where $\boldsymbol{\mathcal{X}}_{j,t|t-1}$ are predicted sigma points calculated by the process model (6.8), and the predicted covariance matrix is computed by

$$\boldsymbol{P}_{t|t-1} = \sum_{j=0}^{2n} w_j^{(c)} \left[\boldsymbol{\mathcal{X}}_{j,t|t-1} - \boldsymbol{x}_{t|t-1}\right]\left[\boldsymbol{\mathcal{X}}_{j,t|t-1} - \boldsymbol{x}_{t|t-1}\right]^T + \boldsymbol{G}\boldsymbol{Q}_t\boldsymbol{G}^T. \tag{6.26}$$

In order to present the UIF update equations, let us first define a pseudo measurement matrix $\boldsymbol{\mathcal{H}}_t$ as [218]

$$\boldsymbol{\mathcal{H}}_t^T = \boldsymbol{P}_{t|t-1}^{-1} \boldsymbol{P}_{t|t-1}^{\mathcal{X},\mathcal{Z}}, \tag{6.27}$$

where the cross-covariance matrix is calculated by

$$\boldsymbol{P}_{t|t-1}^{\mathcal{X},\mathcal{Z}} = \sum_{j=1}^{2n} w_j^{(c)} \left[\boldsymbol{\mathcal{X}}_{j,t|t-1} - \boldsymbol{x}_{t|t-1}\right]\left[\boldsymbol{\mathcal{Z}}_{j,t|t-1} - \boldsymbol{z}_{t|t-1}\right]^T, \tag{6.28}$$

where $\boldsymbol{\mathcal{Z}}_{j,t|t-1} = \boldsymbol{h}\left(\boldsymbol{\mathcal{X}}_{j,t|t-1}\right)$ are observation sigma points, and the predicted measurement vector is obtained by $\boldsymbol{z}_{t|t-1} = \sum_{j=0}^{2n} w_j^{(l)} \boldsymbol{\mathcal{Z}}_{j,t|t-1}$. Then, in terms of pseudo-measurement matrix, information contribution for sensor i can be expressed as[2]

$$\boldsymbol{i}_t^i = \boldsymbol{\mathcal{H}}_{i,t}^T \boldsymbol{R}_{i,t}^{-1} \left[\boldsymbol{z}_t^i - \boldsymbol{z}_{t|t-1} + \boldsymbol{\mathcal{H}}_{i,t}\boldsymbol{x}_{t|t-1}\right] \tag{6.29}$$

$$\boldsymbol{I}_t^i = \boldsymbol{\mathcal{H}}_{i,t}^T \boldsymbol{R}_{i,t}^{-1} \boldsymbol{\mathcal{H}}_{i,t}. \tag{6.30}$$

Now, the measurements are fused just as in the case of EIF, through (6.20) and (6.21).

[2] Here we use index i in matrices $\boldsymbol{\mathcal{H}}_{i,t}$ and $\boldsymbol{R}_{i,t}$ to denote the sensor i in the subscript instead of superscript in order to more clearly denote the transpose and the inverse operators.

6.3.3 *Particle filter*

In the previous sections we have focused mainly on filters which assume unimodal (Gaussian) distribution over the system state. In many applications this assumption may not be adequate and more versatile representations may be needed. In this section we present methods for sensor fusion via particle filters which due to their specific representation of density need additional tools to calculate the update equations.

Let $\{\boldsymbol{x}^p, w^p\}_{p=1}^P$ denote a random measure that characterizes the posterior PDF $p(\boldsymbol{x})$, where $\{\boldsymbol{x}^p,\ p = 1,\dots,P\}$ is a set of particles with associated weights $\{w^p,\ p = 1,\dots,P\}$. The weights are normalised so that $\sum_p w^p = 1$. Then, the posterior density, as discussed previously in Section 2.4.2, can be approximated as [73, 74]

$$p(\boldsymbol{x}_t) \approx \sum_{p=1}^{P} w_t^p \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^p), \tag{6.31}$$

where P is the number of particles and $\delta(.)$ is the Dirac delta measure.

In the centralized solution all the sensor modalities report only their measurements (likelihoods), which corresponds to estimating the posterior via (6.3). After similar derivation to the one in [73] we obtain the expression for weights calculation

$$w(\boldsymbol{x}_t^p) \propto w(\boldsymbol{x}_{t-1}^p) \frac{p(\boldsymbol{x}_t^p \mid \boldsymbol{x}_{t-1}^p)}{q(\boldsymbol{x}_t^p \mid \boldsymbol{x}_{t-1}^p, \boldsymbol{z}_t^{1:m})} \prod_{i=1}^{m} p(\boldsymbol{z}_t^i \mid \boldsymbol{x}_t^p), \tag{6.32}$$

where $q(\,.\,)$ denotes the proposal density. If we choose the prior as the proposal density, $q(\boldsymbol{x}_t^p \mid \boldsymbol{x}_{t-1}^p, \boldsymbol{z}_t^{1:m}) = p(\boldsymbol{x}_t^p \mid \boldsymbol{x}_{t-1}^p)$, then weights are calculated from the following expression

$$w(\boldsymbol{x}_t) \propto w(\boldsymbol{x}_{t-1}^p) \prod_{i=1}^{m} p(\boldsymbol{z}_t^i \mid \boldsymbol{x}_t^p). \tag{6.33}$$

Once the weights are calculated we can estimate the state as follows

$$\hat{\boldsymbol{x}}_{t|t} = E[\boldsymbol{x}_t \mid \boldsymbol{z}_t] \approx \frac{1}{P} \sum_{p=1}^{P} w(\boldsymbol{x}_t^p) \boldsymbol{x}_t^p. \tag{6.34}$$

The resampling of the particles is done at each iteration via the SIR algorithm [73]. Concerning the prediction stage of the filter, we use the model (6.8) to predict the state of each particle.

6.4 HIERARCHICAL SENSOR FUSION

6.4.1 *Information filter*

In this example each sensor runs its own local instance of the EIF—prediction through (6.10) and (6.11), and update through (6.16). Furthermore, all sensor modalities utilise the same process model (6.8). The central processor, on the other hand, also runs its own instance of EIF—prediction through (6.10) and (6.11) with the same process model (6.8) as the sensors utilise, but the global update, i.e. fusion, should be performed in the following manner [217]

$$\boldsymbol{y}_{t|t} = \boldsymbol{y}_{t|t-1} + \sum_{i=1}^{m} \left[\boldsymbol{y}_{i,t|t} - \boldsymbol{y}_{i,t|t-1} \right] \tag{6.35}$$

$$\boldsymbol{Y}_{t|t} = \boldsymbol{Y}_{t|t-1} + \sum_{i=1}^{m} \left[\boldsymbol{Y}_{i,t|t} - \boldsymbol{Y}_{i,t|t-1} \right]. \tag{6.36}$$

We can see that the sensor modalities only have to communicate the difference between the updated, $\boldsymbol{y}_{i,t|t}$, and the predicted, $\boldsymbol{y}_{i,t|t-1}$, information vector. The same applies for the update of the information matrix. This ensures that only the new information is used for fusion.

Hierarchical sensor fusion with UIF is performed in a similar manner as with EIF. Both sensor modalities run their own local, independent, and autonomous UIF and report their estimates to the central fusion processor. The central processor runs a global UIF, and performs the global update, i.e. fusion, through (6.35) and (6.36).

6.4.2 *Particle filter*

In this hierarchical solution with particle filters each sensor modality runs its own local independent particle filter, which needs to be fused with the global particle filter. This corresponds to estimating the posterior via (6.5). Therefore, the importance weights are given by

$$w(\boldsymbol{x}_t^p) \propto w(\boldsymbol{x}_{t-1}^p) \frac{p(\boldsymbol{x}_t^p|\boldsymbol{x}_{t-1}^p)}{q(\boldsymbol{x}_t^p|\boldsymbol{x}_{t-1}^p, \boldsymbol{z}_t^{1:m})} \prod_{i=1}^{m} \frac{p(\boldsymbol{x}_t^p|\boldsymbol{z}_{1:t}^i)}{p(\boldsymbol{x}_t^p|\boldsymbol{z}_{1:t-1}^i)}. \tag{6.37}$$

If we again choose the global prior as the proposal density, $q(\boldsymbol{x}_t^p|\boldsymbol{x}_{t-1}^p, \boldsymbol{z}_t^{1:m}) = p(\boldsymbol{x}_t^p|\boldsymbol{x}_{t-1}^p)$, then weights are calculated from the following expression

$$w(\boldsymbol{x}_t^p) \propto w(\boldsymbol{x}_{t-1}^p) \prod_{i=1}^{m} \frac{p(\boldsymbol{x}_t^p|\boldsymbol{z}_{1:t}^i)}{p(\boldsymbol{x}_t^p|\boldsymbol{z}_{1:t-1}^i)}. \tag{6.38}$$

If all the weights from all the m distributions were on the same support space, then explicit multiplication of weights would be possible (locally at each filter this might be the case). But since most weights are assigned to an infinitesimally small point mass, direct multiplication is not applicable. To solve this problem we need a way to estimate the density function from a particle set. One such method is the Parzen window method [229] which involves placing a kernel function on top of each sample and then evaluating the density as a sum of the kernels—a similar procedure to the one utilized in Section 4.3.1. We continue this approach as proposed in [21, 144], and convert each sample to a kernel

$$K_h(\boldsymbol{x}_t) = h^n K(\boldsymbol{x}_t), \tag{6.39}$$

where $K(.)$ is the particle set covariance, and $h > 0$ is the scaling parameter. For the kernel, we choose:

$$h = \left(\frac{4}{n+2}\right)^e P^{-e}, \tag{6.40}$$

where $e = \frac{1}{n+4}$, and P is the number of particles. At this point, the estimated density function is described as a sum of Gaussian kernels

$$\hat{p}\left(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}^i\right) = \sum_{p=1}^{P} \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_t^p, 2K_h(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}^i)), \tag{6.41}$$

and an illustration of the process is depicted in Fig. 6.1. In [223, 224] the authors propose how to utilize this function estimation for particle set multiplication and division.

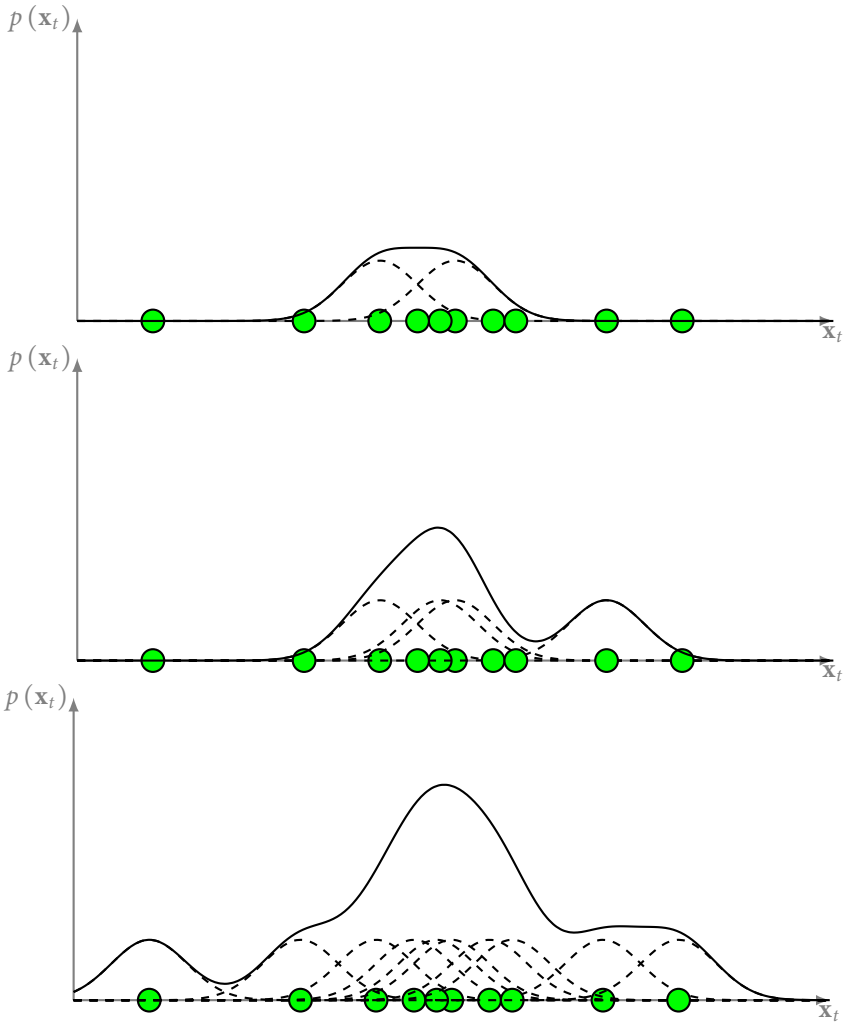


Figure 6.1: An illustration of building up a kernel density estimate from a particle set

6.4.3 *Fusion of arbitrary filters*

In the previous sections we have addressed centralized and hierarchical fusion of the EIFs, UIF and PFs. But what if wee need to fuse a combination of these filters? For an example, an EIF and a PF? In this section we propose a solution to such a problem.

To answer the question, we first need to choose the global filter which will actually keep the global track and fuse the local filters. In most cases we will utilize the filter which has better or higher modeling capabilities. For an example, if one of the filters is a PF, then we might choose also a PF for the global filter, since it is capable of handling both non-linearities and multimodal distributions. This reasoning stems from the fact that if we are using a more versatile filter for local estimation, there must have been a good reason for such a choice, and the global filter should be equally versatile. However, this might not always be the case and there may be situations in which a less versatile and computationally less complex filter could be applied for fusion. Therefore, in this section we shall analyze both of the aforementioned situations, i.e. fusion of a local EIF and PF with a hierarchical EIF, and the fusion of local EIF and PF with a hierarchical PF.

For the case of fusion with the hierarchical EIF we have equations defined in Section 6.4.1, from which we can see that we need to calculate the difference of the information vectors

and matrices of the local updated and predicted states. For the case of the local EIF this is straightforward, while for the case of the local PF we propose to calculate the covariance of the particle set first

$$\begin{aligned}\hat{\boldsymbol{P}}_{t|t} &= E[(\boldsymbol{x}_t - E[\boldsymbol{x}_t])(\boldsymbol{x}_t - E[\boldsymbol{x}_t])^{\mathrm{T}}|\boldsymbol{z}_t] \\ &\approx \frac{1}{P}\sum_{p=1}^{P} w(\boldsymbol{x}_t^p)(\boldsymbol{x}_{t|t}^p - \boldsymbol{x}_{t|t})(\boldsymbol{x}_{t|t}^p - \boldsymbol{x}_{t|t})^{\mathrm{T}},\end{aligned} \tag{6.42}$$

which can then be used to calculate the information variables via (6.12). However, care must be taken with this approach since we are summing up the information of the particle set to a single unimodal distribution. Once having analogously calculated the information variables for the prediction, we can readily fuse the local PF and EIF with the hierarchical EIF via (6.35) and (6.36).

For the case of fusion with the hierarchical PF, we have presented fusion equations in Section 6.4.2, from which we can see that in order to calculate the weights of the hierarchical PF we need to explicitly evaluate the prior and the posterior density of the local EIF and PF. Since EIF assumes a Gaussian distribution, the updated density will have the following form

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}^i) = \frac{1}{2\pi\sqrt{|\boldsymbol{P}_{t|t}|}} \cdot \exp\left\{-\frac{1}{2}\left[\boldsymbol{x}_t - \boldsymbol{x}_{t|t}\right]^{\mathrm{T}} \boldsymbol{P}_{t|t}^{-1}\left[\boldsymbol{x}_t - \boldsymbol{x}_{t|t}\right]\right\}. \tag{6.43}$$

A similar expression can be obtained for the prediction $\boldsymbol{x}_{t|t-1}$ and $\boldsymbol{P}_{t|t-1}$. Furthermore, in order to be able to divide the prior and the posterior density of the particle filter we will need to resort to the kernel density estimation method presented in Section 6.4.2. Hence, the updated and the predicted densities will have the form defined in (6.41).

All this results with the following expression for the calculation of the weights $\boldsymbol{w}(\boldsymbol{x}_t^q)$ of the global PF which relies on the expressions derived for hierarchical sensor fusion (6.5) and on the calculation of the hierarchical particle filter weights (6.38)[3]

$$w(\boldsymbol{x}_t^q) \propto w(\boldsymbol{x}_{t-1}^q) \cdot \frac{\sum_{p=1}^{P}\mathcal{N}(\boldsymbol{x}_t^q; \boldsymbol{x}_t^p, 2K_h(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}^i))}{\sum_{p=1}^{P}\mathcal{N}(\boldsymbol{x}_t^q; \boldsymbol{x}_t^p, 2K_h(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}^i))} \cdot \frac{\mathcal{N}(\boldsymbol{x}_t^q; \boldsymbol{x}_{t|t}, \boldsymbol{P}_{t|t})}{\mathcal{N}(\boldsymbol{x}_t^q; \boldsymbol{x}_{t|t-1}, \boldsymbol{P}_{t|t-1})}. \tag{6.44}$$

6.4.4 *Asynchronous fusion*

In the analysis thus far, we have assumed that all the measurements/estimates arrive synchronously to the fusion center. In most real world applications this might not be the case. So, the question is, how should the fusion be calculated if the measurements/estimates arrive asynchronously? Is there a difference for the centralized and hierarchical case?

Let us assume at this point that all the sensors send their measurements/estimates in fixed, but different time intervals. For an example, if we have three sensors, two might report each 25 ms, while the third might report each 60 ms. In such a case, only the fusion center has to change in order to accommodate asynchronous arrivals, since from a local sensor's point of view, nothing has actually changed.

[3] Note that the particular particle in the hierarchical PF is now denoted with q instead of p in order to leave the p to denote the particles in the local PF

By inspecting (6.3) and (6.5), we see that for the sheer aspect of fusion itself, we only need to change m, the number of sensors that we are fusing at a certain point. But there is also one more very subtle change that needs to be addressed. When we use (6.8) for state prediction we assume that the object undergoes a constant acceleration during a given sampling period, which makes it inappropriate for asynchronous fusion where the prediction and update occurs in practically arbitrary time intervals [76]. This, in effect, changes the way we must calculate the prediction of the state, and the solution is to switch to discretized continuous white noise acceleration model.

Basically, the state prediction equation remains the same, only the process noise covariance matrix needs to be evaluated differently [76]

$$\begin{aligned}\tilde{\boldsymbol{Q}}_t = E\left[\boldsymbol{v}_t \boldsymbol{v}_t^{\mathrm{T}}\right] &= \tilde{q} \int_0^{\Delta T} \begin{bmatrix} \Delta T - \tau & 0 \\ 1 & 0 \\ 0 & \Delta T - \tau \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta T - \tau & 1 & 0 & 0 \\ 0 & 0 & \Delta T - \tau & 1 \end{bmatrix} d\tau \\ &= \tilde{q} \begin{bmatrix} \frac{1}{3}\Delta T^3 & \frac{1}{2}\Delta T^2 & 0 & 0 \\ \frac{1}{2}\Delta T^2 & \Delta T & 0 & 0 \\ 0 & 0 & \frac{1}{3}\Delta T^3 & \frac{1}{2}\Delta T^2 \\ 0 & 0 & \frac{1}{2}\Delta T^2 & \Delta T \end{bmatrix},\end{aligned} \tag{6.45}$$

where $\tilde{q}$ is the continuous-time process noise intensity assumed to be a constant. Recommendations on how to choose $\tilde{q}$ can be found in [76]. This implicitly means that $\boldsymbol{G}\boldsymbol{Q}_t\boldsymbol{G}_t^{\mathrm{T}} \leftarrow \tilde{\boldsymbol{Q}}_t$ in (6.11).

The result above suggests that regardless of the type of fusion, centralized or hierarchical, we only need to correctly calculate the process noise covariance if the measurements/estimates arrive in asynchronous, but locally fixed, time intervals. If the measurements arrive out-of-sequence then a different approach must utilized, confer [230] for a solution in a mulitple hypothesis tracker framework.

6.5 EVALUATION

In this section we test the sensor fusion methods on the problem of object tracking with multiple sensors. For the purpose of simulating a moving object we used a nearly coordinated turn rate model with large process noise [76] (see Appendix A.6).This model differs intentionally from the model used in prediction which is defined in Section 6.2.1 since it is possible that true dynamics of the object are unknown (consider the problem of people tracking). The tracked object is observed at all time by two sensors, one being more precise than the other. The measurements of the first and the second sensor are both corrupted with white Gaussian noise. Figure 6.2 shows the simulated trajectory and the measurements of the sensors. Hereafter, we assume that only one object is being tracked and that all the sensor measurements arrive synchronously. For monitoring tracker performance we utilize entropy and RMSE.

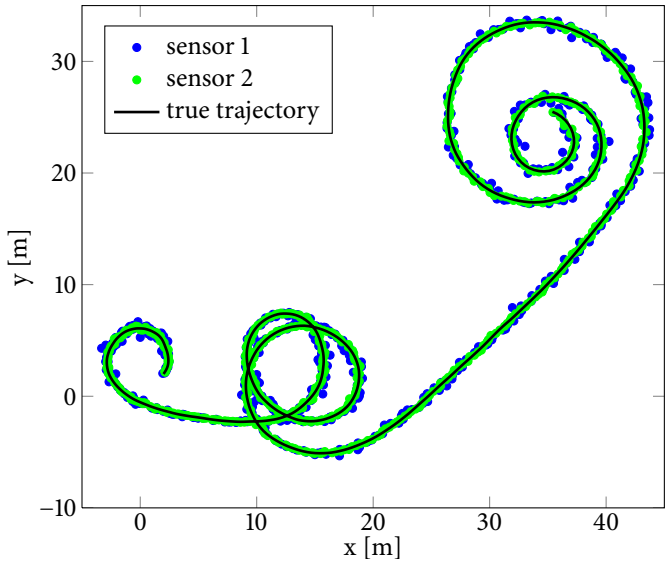


Figure 6.2: Simulated trajectory of a moving object and measurements of two sensors with different noise parameters

6.5.1 *Entropy and RMSE*

We utilize entropy $H(\boldsymbol{x}_t)$ as a measure of the tracker performance. Entropy is a very useful measure of informativeness, and therefore we use it to track 'confidence' of the tracker in its estimates. This way we can analyze how sensor fusion affects the tracker's informativeness. Ideally, by including more sensors, even the less precise ones, we should experience an increase in informativeness, i.e. a decrease in entropy.

A measure of entropy can take many analytical forms. Shannon entropy can be difficult to analytically work with, e.g. Shannon entropy of a mixture of distributions cannot be expressed in closed-form, and therefore we chose to work with Rényi entropy which usually offers a more suitable framework for analytical calculations [141]. Rényi quadratic entropy was introduced in Section 3.5.5 and here we present its form for a random variable $\boldsymbol{x}_t$ with a Gaussian distribution

$$H_2(\boldsymbol{x}_t) = \frac{n}{2}\log 4\pi + \frac{1}{2}\log|\boldsymbol{P}_t|, \tag{6.46}$$

where n is the state dimension and the entropy is proportional to the logarithm of the determinant of the covariance $\boldsymbol{P}_t$.

Entropy calculation of continuous random variables is based on the probability density functions of these variables. In order to calculate entropy of a particle filter, which rather represents the density and not the function, we need a non-parametric method to estimate the PDF. As in Section 6.4.2 we will utilize the Parzen window method [229] which estimates the density as a sum of Gaussian kernels for which an analytical solution for the quadratic Rényi entropy exists [231]

$$H_2(\boldsymbol{x}_t) = -\log\frac{1}{P^2}\sum_{i=1}^{P}\sum_{j=1}^{P}\mathcal{N}(\boldsymbol{x}_t^i - \boldsymbol{x}_t^j; 0, 2K_h(\boldsymbol{x}_t)). \tag{6.47}$$

Table 6.1: Evaluation results of the sensor fusion object tracking on synthetic data

	RMSE position [m] (velocity [m/s])		
	EIF	UIF	PF
Centralized	0.69 (1.59)	0.46 (0.85)	0.10 (0.59)
Sensor 1	1.10 (2.13)	0.23 (0.70)	0.25 (0.98)
Sensor 2	0.77 (1.72)	0.22 (0.68)	0.10 (0.66)
Hierarchical	0.70 (1.61)	0.22 (0.69)	0.13 (0.54)
Arbitrary	local EIF	local PF	fused EIF
	0.10 (0.66)	0.11 (0.66)	0.09 (0.59)
	local EIF	local PF	fused PF
	0.10 (0.66)	0.11 (0.66)	0.11 (0.60)

The RMSE is calculated both for the position and velocity as follows

$$\begin{aligned} e_{\text{pos}} &= \sqrt{\frac{1}{T}\sum_{k=1}^{T}(\hat{x}_t - x_t)^2 + (\hat{y}_t - y_t)^2} \\ e_{\text{vel}} &= \sqrt{\frac{1}{T}\sum_{k=1}^{T}(\hat{\dot{x}}_t - \dot{x}_t)^2 + (\hat{\dot{y}}_t - \dot{y}_t)^2}, \end{aligned} \tag{6.48}$$

where T is the simulation length, $(\hat{x}_t, \hat{y}_t)$ are estimated coordinates and (x_t, y_t) are true coordinates at time index k, while $(\hat{\dot{x}}_t, \hat{\dot{y}}_t)$ are the estimated velocities and $(\dot{x}_t, \dot{y}_t)$ are true velocities at time index k.

6.5.2 *Comparative analysis*

In this section we will present the entropy and the RMSE for the cases of centralized, hierarchical fusion and for the examples of fusing an EIF and a PF through a global EIF and PF. Table 6.1 shows the results of the RMSE in the position and in the velocity. In centralized fusion all the measurements from the sensors were communicated to the fusion node which in turn ran an estimator and fused the measurements via (6.20) and (6.21), (6.29) and (6.30), and (6.33) for the cases of EIF, UIF, and PF, respectively. In the case of hierarchical fusion each sensor ran its own local estimator, which communicated its estimate to the fusion node, which then via (6.35) and (6.36), and (6.38) fused the local estimates. Figs. 6.3, 6.4 and 6.5 show the entropy of sensor 1, sensor 2, and the fused EIF,UIF and PF, respectively. For all the examples of hierarchical fusion we can notice a pattern in which the fused estimator had similar RMSE as the more precise sensor, but smaller entropy than any of local sensor estimators indicating a reduction in uncertainty as can be seen in Figs. 6.3, 6.4, and 6.5. This result showed that although we fused a very precise sensor with a less precise one, the resulting estimator did in fact have a benefit in form of a reduced uncertainty.

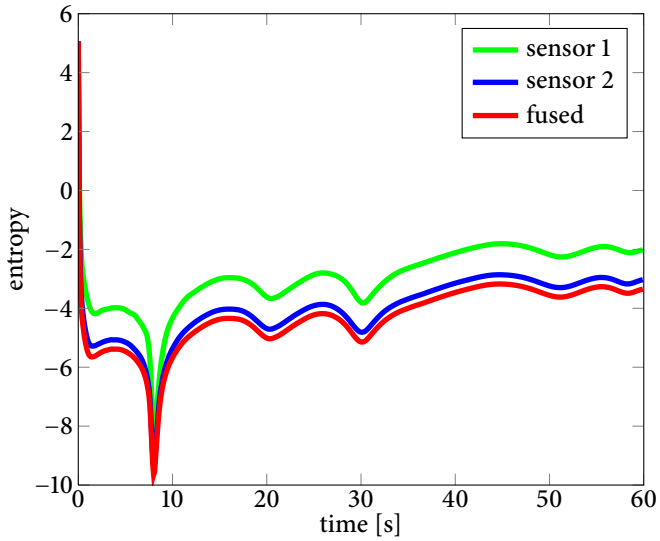


Figure 6.3: Entropy of the EIF tracker with the first sensor, with the second sensor, and the entropy of the fused hierarchical EIF tracker

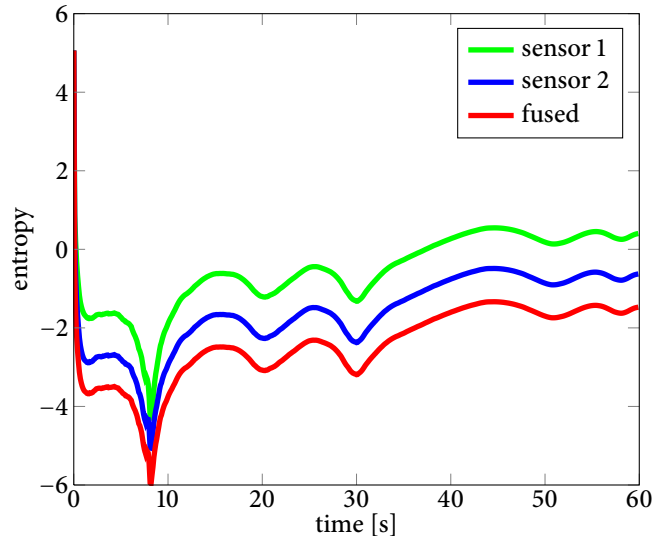


Figure 6.4: Entropy of the UIF with the first sensor, with the second sensor, and the entropy of the fused hierarchical UIF

6.6 MULTIPLE OBJECT TRACKING AND SENSOR FUSION BY A MOBILE ROBOT

A large body of work exists on tracking moving objects with mobile robots. As discussed in [232] two major approaches can be identified, both defined by the sensors. The first approach stems form the field of computer vision and implies a camera as a major sensor, while the second utilizes LRS whose measurements are similar to those of radars and sonars. Since the field of tracking and surveillance (where radars and sonars are commonly used), was well established before the mobile robotics, a lot of results [233, 234] from that field were applied to the problem of people tracking with an LRS. The LRS approach can be further subdivided according to data association techniques into deterministic and probabilistic [135, 235–237] approaches. Additionally, these two sensors can also be used conjointly. For example, in [238], the nearest neighbour approach and unscented Kalman filter are used for tracking

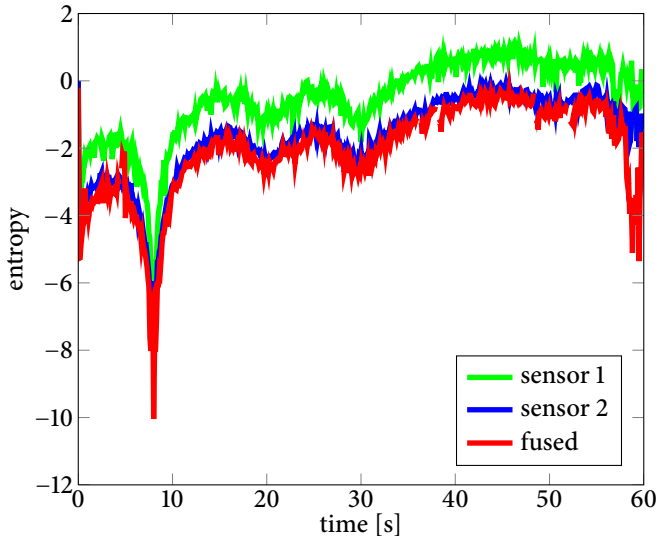


Figure 6.5: Entropy of the PF with the first sensor, with the second sensor, and the entropy of the fused hierarchical PF

people with a laser and a camera, while in [219] the authors used euclidean distance and covariance intersection method for fusing laser, sonar and camera measurements.

When considering multitarget tracking, data association is the fundamental problem. A detailed overview of probabilistic data association techniques is given in [239]. Our previous work [135] was heavily influenced by [235, 236], where the authors use the joint probabilistic data association filter (JPDAF) to solve the data association problem. In [240] the JPDAF is extended to handle multiple data sources (sensors). Such a rigorous approach is questioned when looking at the JPDAF seminal paper [234], since the target-sensor geometry indicates that three sonar sensors were used to obtain the measurements. Since in our case the data acquisition happens asynchronously across sensors, we prefer the approach in [234]. The idea is as follows. When the new sensory inputs arrive, predictions about track states are made, and then the JPDAF is used to solve the data association problem. Finally, the track states are updated according to the association probabilities, where the final steps use the likelihood function of the reporting sensor, and that is the only thing required by the JPDAF to handle the multisensor case.

Another approach to probabilistic data association is the multiple hypothesis tracker (MHT) developed in the seminal paper [233]. It is an optimal solution to the data association problem, unlike the JPDAF. As discussed in [233], the JPDAF is a special case of the MHT, in which only one hypothesis remains after data processing. To be clear, the reference is made to [241], which is the initial derivation of the JPDAF.

In [233] the multisensor problem for two different generic types of sensors is solved. It is accomplished by describing sensors with their detection and false-alarm statistics. Thanks to such approach, we can use any type of a sensor, provided we have its probabilistic description. The downside of the MHT is in its high memory and processing requirements (which grow exponentially with the number of tracks). However, an efficient implementation of the MHT is discussed in [239] and some recent applications are presented in [242, 243].

Instead of using the optimal MHT with complex hypothesis branching, we choose the simpler, although not optimal, JPDAF as it is a very convenient solution for people tracking

by a mobile robot for its local navigation [244]. However, the JPDAF assumes a constant and known number of objects in the scene, and to solve this drawback an entropy based track management algorithm is used. The approach is tested for the case of multiple people tracking with the Kalman filter and heterogeneous sensors.

6.6.1 *Kalman JPDAF*

We consider initialized tracks at time t described by a set of continuous random variables $\boldsymbol{X}_t = \left\{\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \ldots, \boldsymbol{x}_t^{T_k}\right\}$, where T_k denotes the number of tracks. At time t we receive a set $\boldsymbol{Z}_t^m = \left\{\boldsymbol{z}_t^{m,1}, \boldsymbol{z}_t^{m,2}, \ldots, \boldsymbol{z}_t^{m,T_j}\right\}$ of measurements from sensor m, where T_j denotes the number of measurements. Note that a single sensors can give us multiple measurements at a single time instant.

In this section we use the general constant velocity model for motion in 2D plane (6.8) presented in Section 6.2.1. Prediction is calculated using the standard Kalman filter equations

$$\begin{aligned} \boldsymbol{x}_{t|t-1}^k &= \boldsymbol{A}\boldsymbol{x}_{t-1}^k\,, \\ \boldsymbol{P}_{t|t-1}^k &= \boldsymbol{A}\boldsymbol{P}_{t-1}^k\boldsymbol{A}^T + \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T, \end{aligned} \tag{6.49}$$

where $\boldsymbol{x}_t^k$ denotes estimated state at track k at time instant t. If our measurements arrive asynchronously the process noise needs to be taken into account as presented in Section 6.4.4. The innovation vector is calculated via

$$\boldsymbol{v}_t^{k,j} = \boldsymbol{z}_t^{m,j} - \boldsymbol{H}^m\boldsymbol{x}_{t|t-1}^k, \tag{6.50}$$

and its covariance matrix is given by

$$\boldsymbol{S}_t^{k,j} = \boldsymbol{H}^m\boldsymbol{P}_{t|t-1}^k\boldsymbol{H}^{m^T} + \boldsymbol{R}^m. \tag{6.51}$$

The innovation vector and covariance matrix can be used for measurement gating. Since $\boldsymbol{v}_t^{k,j^T}\boldsymbol{S}_t^{k,j^{-1}}\boldsymbol{v}_t^{k,j}$ has χ^2 distribution, by using tables we can select upper limit which includes valid measurements with, e.g., 99% probability.

Update is done by using all the validated measurements, i.e. weighted innovation is used for the state update

$$\begin{aligned} \boldsymbol{v}_t^k &= \sum_{j=1}^{T_j}\beta_t^{k,j}\boldsymbol{v}_t^{k,j} \\ \boldsymbol{x}_{t|t}^k &= \boldsymbol{x}_{t|t-1}^k + \boldsymbol{K}_t\boldsymbol{v}_t^k\,. \end{aligned} \tag{6.52}$$

Given $\beta_t^k = 1 - \sum_{j=1}^{T_j}\beta_t^{k,j}$ and $\boldsymbol{P}_{\boldsymbol{v}_t^k} = \sum_{j=1}^{T_j}\beta_t^{k,j}\boldsymbol{v}_t^{k,j}\boldsymbol{v}_t^{k,j^T} - \boldsymbol{v}_t^k\boldsymbol{v}_t^{k^T}$ the covariance update is calculated as in [245]

$$\boldsymbol{P}_{t|t}^k = \beta_t^k\boldsymbol{P}_{t|t-1}^k + (1-\beta_t^k)\left[\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H}^m\right]\boldsymbol{P}_{t|t-1}^k + \boldsymbol{K}_t\boldsymbol{P}_{\boldsymbol{v}_t^k}\boldsymbol{K}_t^T\,. \tag{6.53}$$

An important implementation note is that instead of the standard Kalman filter covariance update $\left[\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H}^m\right]\boldsymbol{P}_{t|t-1}^k$ we use Joseph's stabilized form $\left[\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H}^m\right]\boldsymbol{P}_{t|t-1}^k\left[\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H}^m\right]^T + \boldsymbol{K}_t\boldsymbol{R}^m\boldsymbol{K}_t^T$, since the standard form caused numerical problems. For details regarding the calculation of posterior probabilities of association events β_t^j, i.e. that measurement j belongs to object t, please confer [139, 234, 235].

6.6.2 *Track management*

When tracking multiple targets, track management is practically as important as the association itself. A solution for the Kalman filter, described in [245], is based on a logarithmic hypothesis ratio and innovation matrix. In [235] a Bayesian estimator of the number of objects for an LRS is proposed. This approach requires learning the probability of how many features are observed under a presumed number of objects in the perceptual field of the sensor, while the tracking performance is monitored by an average of the sum of unnormalized sample weights of the particle filter.

In this section we use entropy measure as a feature in track management. If such a strong mathematical principle is tractable for the underlying probability distribution, then it can be readily utilized for track management independently of the filtering approach. Furthermore, all the information required for the entropy calculation is already available in the running filter, for the Gaussian distribution confer (6.46), and as it will be presented, threshold setting is quite convenient. Although Shannon entropy could have been utilized also for the case of the Gaussian distribution, we have opted for the quadratic Rényi entropy since it can also be evaluated in the case of the particle filter when the posterior density is estimated by kernel density estimation methods [21, 139, 144].

The track management logic is as follows. When the tracks are initialized, they are considered tentative and the initial entropy is stored. When the entropy of a tentative track drops for 50%—it is a confirmed track. If and when the entropy gets 20% larger than the initial entropy—the track is deleted. This logic reflect the fact that if the entropy is rising, we are becoming less and less confident that the track is informative.

6.6.3 *Experimental setup and results*

The experiments were conducted with a Pioneer 3DX robot. The laser sensor was the Sick LMS 200 model, while the microphone array is of our design. Furthermore, we also used the Kinect time-of-flight camera with a face recognition algorithm based on [247] to yield a set of measurements in 3D. In the experiment two people were walking in an intersecting trajectory in front of the robot (a snapshot of the experiment is shown in Fig. 6.6). The results are shown in Fig. 6.7 from which we can see that the first person (blue line) started at $(-1.2, 2.3)$ m and finished at $(0.9, 2.3)$ m, while the second person (green line) started at $(0.7, 0)$ m and finished at $(0.6, 0)$ m. The first person was in the FOV of all the three sensors and was talking throughout the experiment, while the second person entered LRS FOV at a later time, kept quiet and was facing the robot only in the second half of the trajectory. At 15 s the second object got occluded by the first, which caused an increase in entropy, while at 30 s the second object occluded the first shortly before exiting the scene. The false alarms were caused by tiles on the wall and leg-like features in the room (chairs and tables). Tracks were correctly initialized and maintained, despite the large number of false alarms. The second track was deleted short-after the second person left the LRS FOV.

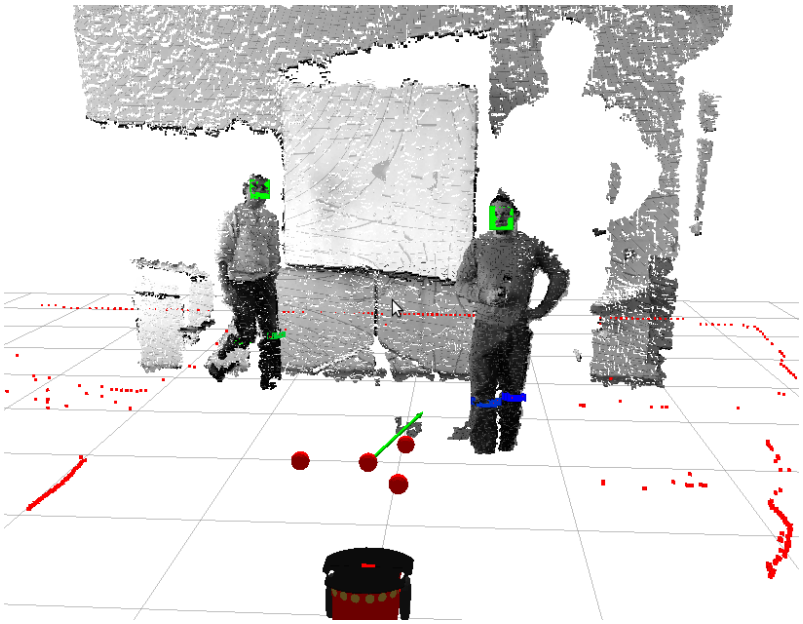
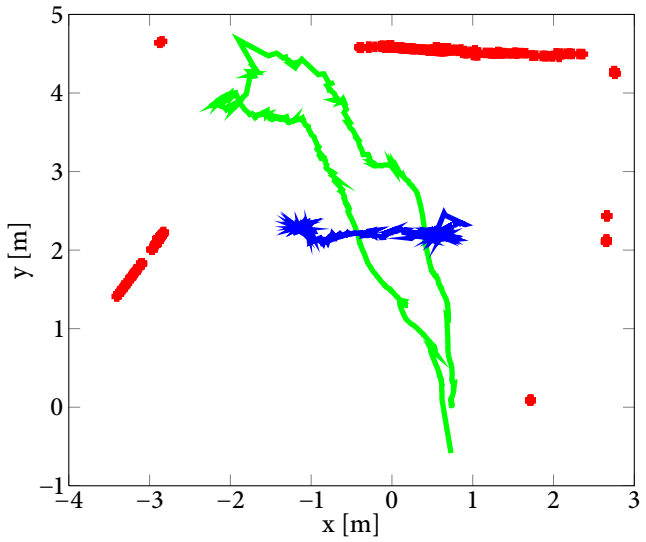
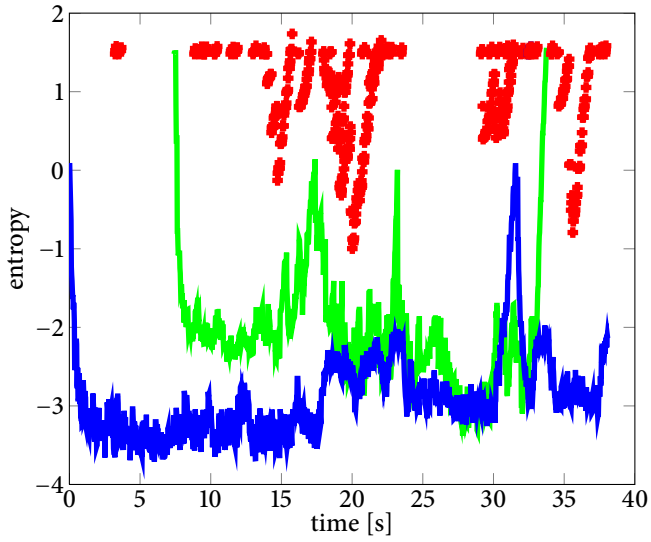


Figure 6.6: A snapshot of the data acquisition and signal processing for the experiments. The measurements were classified and collected based on our work in [55, 135, 246], with only the signal processing stage done, i.e. no tracking was performed on the sensor level.



(a) People trajectories



(b) KF entropies

Figure 6.7: Experimental results for the KF—estimated (solid) track states, and tentative but not confirmed tracks (red + marker)

6.7 SUMMARY

In this chapter we have first presented Bayesian methods for sensor fusion which were divided in two groups based on the information that each sensor modality reported: a centralized independent likelihood fusion where each sensor only reported its measurement, and hierarchical fusion where each sensor ran its filter and reported its own estimate along with the uncertainty. The solution for sensor fusion in the former case was an elegant multiplication of local sensor likelihoods and the central (global) prior, while the solution for the latter case was a bit more involved and required a division of the local posterior and prior in order to extract only the new information which was then multiplied with the central (global) prior. The aforementioned approaches were given concrete expressions in the form of the EIF, UIF and PF. The experiments were conducted on synthetic data modeling a situation of dynamic object tracking with several sensor modalities.

The benefits of proper sensor fusion was demonstrated by depicting entropy of the trackers. We have shown that the fused estimates have lower entropy than the most precise sensor, even when being fused with a more imprecise sensors. Furthermore, we also discussed the problem of arbitrary sensor fusion, i.e. situations in which one sensor tracks the object with one type of a filter, e.g. a EIF, while the other tracks the object with a PF. We proposed a solution and demonstrated the approach by fusing local EIF and PF with a global PF and a global EIF. All the previous results were based on the assumption that the measurements/estimates arrive synchronously to the fusion center. Furthermore, we also discussed a solution with necessary modifications in the case of asynchronous fusion, which mostly pertained to the correct system prediction, i.e. calculation of the process noise covariance matrix.

Ensuingly, we addressed the problem of tracking multiple objects with multiple heterogeneous sensors—specifically an LRS, a microphone array, and an RGB-D camera. The integration of multiple sensors is solved by asynchronously updating the tracking filters as new data arrives. We solved the data association problem by applying the JPDAF, which is a suboptimal zero-scan derivation of the MHT, but which in effect assumes a known number of objects. To circumvent this assumption, an entropy based track management scheme was used, and demonstrated its performance for the Kalman filter in an experiment where two people were tracked by a mobile robot. The results showed that the proposed algorithm is capable of maintaining a viable number of filters with correct association and accurate tracking.

7

Conclusions and outlook

Navigating among highly dynamic objects is a daily routine for humans, but still presents a significant challenge for autonomous mobile robots. Naturally, this stems from the fact that humans are equipped with a combination of sophisticated sensors and with processing capabilities still unmet by artificial systems. However, this does not imply that mobile robots cannot be endowed with similar capabilities, albeit by a different approach. Therefore, the aim of the thesis was to set a building block that will help in solving the aforementioned problem. This building block, as the title implies, is the detection and tracking of objects in motion by omnidirectional sensors of a mobile robot. As stated earlier, this task is important for autonomous mobile robots, especially if they are to become a part of our daily lives.

The thesis started first with a general background which described the utilized sensors in the thesis and the main mathematical tools. The sensors were the microphone array and the omnidirectional camera which, as it has been shown, yield direction-only measurements thus making them perfect candidates for application of methods from the field of directional statistics. The microphone array, as it was used in the thesis, measures the location of the speaker as a direction in 2D plane, i.e. a point on the unit circle. Due to the nature of the measurement derivation our sensor model ended up being a multimodal mixture of probability density functions. Since we were dealing with random angular variables, it was only natural to model the measurements with a circular distribution, a distribution on the unit circle—the von Mises distribution. However, this was only the first step in fully solving the speaker localization problem. Since our measurement model, the likelihood, was non-Gaussian and multimodal, we needed to resort to versatile Bayesian estimation methods and the first applied solution was the particle filter. The solution proved to be effective and was tested in a series of carefully designed experiments to validate the proposed approach. But the next question was, could the finite sample-based approach be circumvented in a way? Can we stay in the domain of analytical solutions somehow? We found the solution in the form of Bayesian mixture filters which was derived for the case of the von Mises distribution. As we could have seen this constituted calculation of the convolution and the product of two von Mises distributions to solve the prediction and the update part of Bayesian estimators. However, both the particle filter and the von Mises mixture filter estimate only the bearing of the speaker, and not the range. Thus our next research was focused in the direction of active speaker localization, where the information about the displacement of the mobile robot was fused with the bearing measurements in order to

estimate both the bearing and the range of the speaker.

In order to ensure robust speaker localization we needed a method that would reliably detect the presence of speech in the signal frames, thus prevent misguided measurements of noise-only frames. The problem of voice activity detection was approached firstly via the frameworks of statistical model-based detectors. This constituted modeling the distribution of the discrete Fourier transform coefficients under two distinct hypotheses: (i) the signal frame consists of noise and speech and (ii) the signal frame consists of noise only, and then calculating the likelihood ratio thereof. Three different statistical model-based detectors were compared for the task and the detector that modeled the distribution of the signal envelope with the Rayleigh and Rice distribution, under the two previously mentioned hypotheses, respectively, showed the best performance. The voice activity detection algorithm was further enhanced by combining the likelihood ratio of the Rayleigh and Rice distribution with 70 other spectral and temporal features. It was conjecture that this would bring an increase, via a supervised learning algorithm, in detection performance. But in order to avoid blind aggregation of the input space and account for possible correlations among the input variables, partial mutual information algorithm was employed for the task. The result was a reduced input vector consisting of 13 variables and thereafter three supervised learning algorithms were tested and compared, namely the support vector machine, Boost and artificial neural networks. Finally, the results showed that the proposed approach yielded better results than the standalone statistical model-based detector and among the three supervised learning algorithms the Boost proved to be most appropriate.

The second omnidirectional sensor that was used in the thesis is the omnidirectional camera—a rather general concept that can be achieved by several mirror-lens and camera combinations. In the thesis a perspective camera with a fish-eye lens was used to yield an omnidirectional image. The main reason that directional statistics proved utile in moving object tracking in omnidirectional images was due to the spherical unified projection model. This model describes the formation of images in omnidirectional systems and as such provides a way to represent the image on the unit sphere. Since our task at hand was to detect motion and perform tracking in such images we proposed methods that mostly relied on the processing on the unit sphere. Hence, the detection was performed by calculating the optical flow in the image and lifting the optical flow vectors to the sphere, after which they were classified as either being induced by the ego-motion of the robot or by moving objects in the surrounding scene. Once the moving object flow clusters were detected they could serve as a measurement on the unit sphere. Since we were dealing with random spherical measurements it was only natural to model the measurements with a distribution on the unit sphere—the von Mises-Fisher distribution. With such an approach the foundations for Bayesian estimation on the sphere were set up and the moving object was tracked in a similar manner as the speaker was tracked with the microphone array—prediction and update were solved by calculating the convolution and the product of two von Mises-Fisher distributions (except in this case we used a single density, and not a mixture). Thereafter, the estimated position was used to represent a feature in image-based visual servoing in order to devise a control law that would make the robot follow the moving object. The control law was calculated by utilizing an interaction matrix that connected the camera velocities with the velocities of image features. The algorithm was based on a projection of a point to

the unit sphere and was calculated in cylindrical coordinates.

When working with several sensors on a mobile robot, the next question is how to fuse multiple, possibly, heterogeneous sensors? In the thesis we have analyzed the sensor fusion problem via the prism of moving object tracking. Firstly, we looked at two fusion principles: centralized fusion in which sensors report only their probabilistically structured measurements, and hierarchical fusion where each sensor runs its own local filter and reports the estimates instead of just the measurements. We have presented the solutions to the two fusion principles in the form of the extended information filter, the unscented information filter, and the particle filter. Furthermore, we have analyzed the problem of fusing arbitrary filters like the extended information filter and the particle filter, Finally, we presented a solution to multiple people tracking by mobile robot equipped with a laser range sensor, microphone array and an RGB-D camera. The solution was based on the join probabilistic data association filter and entropy-based track management.

Naturally, some of the presented methods in the thesis can still be further improved or extended. The speaker tracking method that was presented can be applied to any bearing-only scenario and is not limited to just the presented problem. More elaborate methods can be applied at the signal processing stage that would detect multiple talking speakers and thus necessitate extension of the von Mises mixture tracking to handle multiple targets. This would entail solving the data association and track management problems. The voice activity detection could also be improved further by taking into account that we have multi-channel recordings thanks to the microphone array as opposed to performing detection based on a single channel signal. Concerning the omnidirectional camera, the detection stage is highly dependent on the calculated mobile robot displacement between the images. Thus, further enhancement would comprise of fusing the robot odometry with other sensors, like an inertial measurement unit, or with image-based methods like visual odometry or even SLAM. Just like in the case of speaker tracking, the von Mises-Fisher based tracker was developed for single target tracking and as such can be further developed to accommodate multiple target scenarios.

A

Appendix

A.1 PRODUCT OF VON MISES DISTRIBUTIONS

In Section 3.5 in order to evaluate the update step we needed to calculate the product of two von Mises distributions. Let us assume that the two multiplicands are $p(x;\mu_i,\kappa_i)$ and $p(x;\mu_j,\kappa_j)$. Then their product yields

$$p(x;\mu_i,\kappa_i)p(x;\mu_j,\kappa_j) = \frac{1}{4\pi^2 I_0(\kappa_i)I_0(\kappa_j)} \exp\{\kappa_i\cos(x-\mu_i)+\kappa_j\cos(x-\mu_j)\}. \tag{A.1}$$

Intuitively, next thing that we have to do is manipulate the argument in the exponential function in order to produce a von Mises like expression. Given that, we use substitution $\xi = x-\mu_i$ in the exponent argument and apply trigonometric equality of the cosine of a sum of angles

$$\begin{aligned}\kappa_i\cos(x-\mu_i)+\kappa_j\cos(x-\mu_j) &= \kappa_i\cos\xi+\kappa_j\cos(\xi+\mu_i-\mu_j)\\ &= \kappa_i\cos\xi+\kappa_j\cos(\xi+\Delta\mu)\\ &= \kappa_i\cos\xi+\kappa_j(\cos\xi\cos\Delta\mu-\sin\xi\sin\Delta\mu)\\ &= (\kappa_i+\kappa_j\cos\Delta\mu)\cos\xi+(-\kappa_j\sin\Delta\mu)\sin\xi,\end{aligned} \tag{A.2}$$

where $\Delta\mu = \mu_i-\mu_j$. By utilizing the following equality

$$\alpha\cos\xi+\beta\sin\xi = \sqrt{\alpha^2+\beta^2}\cos(\xi-\arctan\frac{\alpha}{\beta}) \tag{A.3}$$

we obtain

$$\begin{aligned}\kappa_i\cos(x-\mu_i)+\kappa_j\cos(x-\mu_j) = \cos\left\{x-\left[\mu_i+\arctan\left(\frac{-\sin\Delta\mu}{\frac{\kappa_i}{\kappa_j}+\cos\Delta\mu}\right)\right]\right\}\\ \cdot\sqrt{\kappa_i^2+\kappa_j^2+2\kappa_i\kappa_j\cos\Delta\mu}\end{aligned} \tag{A.4}$$

By using the substitutions

$$\begin{aligned}\mu_{ij} &= \mu_i+\arctan\left(\frac{-\sin\Delta\mu}{\frac{\kappa_i}{\kappa_j}+\cos\Delta\mu}\right)\\ \kappa_{ij} &= \sqrt{\kappa_i^2+\kappa_j^2+2\kappa_i\kappa_j\cos\Delta\mu}\end{aligned} \tag{A.5}$$

we finally arrive to the expression for the product of two von Mises distributions

$$\begin{aligned} p(x;\mu_i,\kappa_i)p(x;\mu_j,\kappa_j) &= \frac{1}{4\pi^2 I_0(\kappa_i)I_0(\kappa_j)} \exp\{\kappa_{ij}\cos(x-\mu_{ij})\} \\ &= \frac{I_0(\kappa_{ij})}{2\pi I_0(\kappa_i)I_0(\kappa_j)} \frac{1}{2\pi I_0(\kappa_{ij})} \exp\{\kappa_{ij}\cos(x-\mu_{ij})\}, \end{aligned} \tag{A.6}$$

from which we can see that product of two von Mises distributions is an unnormalized von Mises distribution by the factor

$$\frac{I_0(\kappa_{ij})}{2\pi I_0(\kappa_i)I_0(\kappa_j)}. \tag{A.7}$$

The normalizing factor in the Bayes rule which integrates (A.6) would account for proper normalization and we would obtain a correct von Mises density. By inspecting (A.7) we can see that it is strikingly similar to the convolution of two von Mises densities from the prediction part of the filter (3.26). Indeed, by integrating (A.1) we are convolving the two distributions in x. By taking into account that (3.26) can been approximated by (3.27) we can apply the same logic to (A.7) which yields

$$\frac{I_0(\kappa_{ij})}{2\pi I_0(\kappa_i)I_0(\kappa_j)} \approx \frac{1}{2\pi I_0(\tilde{\kappa})} \exp\{\tilde{\kappa}\cos(\mu_i-\mu_j)\}, \tag{A.8}$$

where $\tilde{\kappa} = A^{-1}(A(\kappa_i)A(\kappa_j))$. Thus when we have a von Mises mixture for a prior and a likelihood, each result of a product of $p_i = w_i p(x;\mu_i,\kappa_i)$ and $p_j = w_j p(x;\mu_j,\kappa_j)$ would have the weight $w_i w_j$ scaled by (A.7). We can see that this would make the weight of the product a function of the factor mean directions and concentration parameters through (A.7), which when approximated by (A.8) can be interpreted as a sort of von Mises 'innovation'. Similar behavior is exhibited by the Gaussian distribution [137].

A.2 KULLBACK-LEIBLER DISTANCE BETWEEN VON MISES DENSITIES

For component number reduction in von Mises mixtures in Section 3.5 we needed to derive a distance measure that we would calculate in order to choose the closest components in the mixture which we would then merge. We used the scaled symmetrized KL which accounts for components not being valid densities, but their scaled versions. In essence this distance requires calculation of the KL between true von Mises densities, $p(x;\mu_p,\kappa_p)$ and $q = p(x;\mu_q,\kappa_q)$, and to calculate it we shall resort to the following formula

$$D_{\mathrm{KL}}(p,q) = B_F(\theta_q,\theta_p), \tag{A.9}$$

where θ_p and θ_q are natural parameters of p and q, respectively, and B_F denotes Bregman divergence generated by the convex log-normalizing function F [248]

$$B_F(\theta_1,\theta_2) = F(\theta_1) - F(\theta_2) - \nabla F(\theta_2)\cdot(\theta_1-\theta_2). \tag{A.10}$$

The result in (A.10) is valuable in the sense that it translates the KL distance between exponential family densities to the Bregman divergence between the respective natural parameters, but in reversed order. This will facilitate the derivation of the KL distance between von Mises

densities. Recall that for the von Mises distribution the natural parameter θ, the minimum sufficient statistics $T(x)$, and the log normalizer $F(\theta)$ are given by

$$\begin{aligned} \theta &= (\theta_1, \theta_2) = (\kappa \cos \mu, \kappa \sin \mu) \\ T(x) &= (\cos(x), \sin(x)) \\ F(\theta) &= \log(2\pi I_0(\sqrt{\theta_1^2 + \theta_2^2})) = \log(2\pi I_0(\kappa)). \end{aligned} \tag{A.11}$$

Before we calculate the KL distance we need to first evaluate the gradient of the log normalizer

$$\nabla F(\theta) = \left(\frac{\partial}{\partial \theta_1} F(\theta), \frac{\partial}{\partial \theta_2} F(\theta) \right). \tag{A.12}$$

By taking into account that $\frac{\mathrm{d}}{\mathrm{d}\kappa} I_0(\kappa) = I_1(\kappa)$ we shall evaluate only the gradient only for θ_1 since the problem is symmetrical for θ_2

$$\begin{aligned} \frac{\partial}{\partial \theta_1} F(\theta) &= \frac{2\pi I_1(\sqrt{\theta_1^2 + \theta_2^2})}{2\pi I_0(\sqrt{\theta_1^2 + \theta_2^2})} \frac{1}{2} \frac{1}{\sqrt{\theta_1^2 + \theta_2^2}} 2\theta_1 \\ &= \frac{I_1(\kappa)}{I_0(\kappa)} \frac{\kappa \cos \mu}{\kappa} \\ &= A(\kappa) \cos \mu. \end{aligned} \tag{A.13}$$

Thus, we obtain the expression for the gradient of the log normalizer

$$\nabla F(\theta) = A(\kappa)(\cos \mu, \sin \mu). \tag{A.14}$$

Now we can return to calculating the KL distance via the Bregman divergence A.10

$$\begin{aligned} B_F(\theta_q, \theta_p) &= F(\theta_q) - F(\theta_p) - \nabla F(\theta_p) \cdot (\theta_q - \theta_p) \\ &= \log \frac{I_0(\kappa_q)}{I_0(\kappa_p)} - A(\kappa_p) \left(\cos \mu_p, \sin \mu_p \right) \\ &\cdot \left(\kappa_q \cos \mu_q - \kappa_p \cos \mu_p, \kappa_q \sin \mu_q - \kappa_p \sin \mu_p \right) \\ &= \log \frac{I_0(\kappa_q)}{I_0(\kappa_p)} - A(\kappa_p) \\ &\cdot (-\kappa_p \cos^2 \mu_p - \kappa_p \sin^2 \mu_p + \kappa_q \cos \mu_q \cos \mu_p + \kappa_q \sin \mu_q \sin \mu_p) \\ &= \log \frac{I_0(\kappa_q)}{I_0(\kappa_p)} - A(\kappa_p)(-\kappa_p + \kappa_q \cos(\mu_p - \mu_q)). \end{aligned} \tag{A.15}$$

Finally, the formula for the KL distance between two von Mises densities is given by

$$D_{\mathrm{KL}}(p, q) = \log \frac{I_0(\kappa_q)}{I_0(\kappa_p)} + A(\kappa_p)(\kappa_p - \kappa_q \cos(\mu_p - \mu_q)). \tag{A.16}$$

A.3 VON MISES COMPONENT MERGING

Once we have selected components in the von Mises mixture that we would like to merge in Section 3.5, the next question is how to actually calculate the components of the resulting density $p(x; \mu^*, \kappa^*)$? The optimal procedure in the KL sense is to preserve the expected

value of the sufficient statistics [127]. Thus if we would like to merge M components the formula is as follows

$$w^* \nabla F(\theta^*) = \sum_{i=1}^{M} w_i \nabla F(\theta_i), \tag{A.17}$$

where $w^* = \sum_i w_i$. Since our problem in Section 3.5 is structured so that we always merge two components, e.g. $p(x; \mu_i, \kappa_i)$ and $p(x; \mu_j, \kappa_j)$ by plugging the von Mises densities in (A.17) we obtain the following

$$w^* A(\kappa^*)(\cos\mu^*, \sin\mu^*) = w_i A(\kappa_i)(\cos\mu_i, \sin\mu_i) + w_j A(\kappa_j)(\cos\mu_j, \sin\mu_j) \tag{A.18}$$

from which we can set a system of two equations

$$w^* A(\kappa^*) \cos\mu^* = w_i A(\kappa_i) \cos\mu_i + w_j A(\kappa_j) \cos\mu_j \tag{A.19}$$

$$w^* A(\kappa^*) \sin\mu^* = w_i A(\kappa_i) \sin\mu_i + w_j A(\kappa_j) \sin\mu_j. \tag{A.20}$$

By dividing (A.20) with (A.19) we obtain the expression for calculating the merged mean direction

$$\tan\mu^* = \frac{w_i A(\kappa_i) \sin\mu_i + w_j A(\kappa_j) \sin\mu_j}{w_i A(\kappa_i) \cos\mu_i + w_j A(\kappa_j) \cos\mu_j}. \tag{A.21}$$

By squaring (A.20) and (A.19) and summing up the result we get the expression for calculating the merged concentration parameter

$$\begin{aligned} w^{*2} A^2(\kappa^*) &= w_i^2 A^2(\kappa_i) + w_j^2 A^2(\kappa_j) + 2 w_i w_j A(\kappa_i) A(\kappa_j)(\cos\mu_i \cos\mu_j + \sin\mu_i \sin\mu_j) \\ &= w_i^2 A^2(\kappa_i) + w_j^2 A^2(\kappa_j) + 2 w_i w_j A(\kappa_i) A(\kappa_j) \cos(\mu_i - \mu_j). \end{aligned} \tag{A.22}$$

Note that for determining κ^* from (A.22) we need to resort to numerical methods or approximations as suggested in Section 3.5.

A.4 Von Mises mixture quadratic Rényi entropy calculation

Here we present detailed derivation of the quadratic Rényi entropy of a von Mises mixture. It is utilized in Chapter 3 for tracking the uncertainty of speaker tracking with a von Mises mixture. Rényi entropy of order 2 is calculated via [142]

$$H_2(\theta) = -\log \int p^2(\theta)\, \mathrm{d}\theta. \tag{A.23}$$

For a von Mises mixture it is derived as follows

$$\begin{aligned} H_2(\theta_t) &= -\log \int_0^{2\pi} p^2(\theta_t \,|\, \boldsymbol{z}_{1:k})\, \mathrm{d}\theta_t \\ &= -\log \int_0^{2\pi} \left(\sum_{i=1}^{N} \frac{w_i \exp\left[\kappa_i \cos(\theta_t - \mu_i)\right]}{2\pi I_0(\kappa)} \right)^2 \mathrm{d}\theta_t \\ &= -\log \int_0^{2\pi} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{w_i \exp\left[\kappa_i \cos(\theta_t - \mu_i)\right]}{2\pi I_0(\kappa_i)} \cdot \frac{w_j \exp\left[\kappa_j \cos(\theta_t - \mu_j)\right]}{2\pi I_0(\kappa_j)} \mathrm{d}\theta_t \\ &= -\log \int_0^{2\pi} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{w_{ij} \exp\left[\kappa_{ij} \cos(\theta_t - \mu_{ij})\right]}{4\pi^2 I_0(\kappa_i) I_0(\kappa_j)} \mathrm{d}\theta_t, \end{aligned} \tag{A.24}$$

where $w_{ij} = w_i w_j$, and μ_{ij} and κ_{ij} are given by (3.30) and (3.31), respectively. By rearranging the sums and the integral we arrive to the final expression for the quadratic Rényi entropy

$$\begin{aligned} H_2(\theta_t) &= -\log \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{w_{ij}}{2\pi I_0(\kappa_i) I_0(\kappa_j)} \cdot \frac{1}{2\pi} \int_0^{2\pi} \exp\left[\kappa_{ij}\cos(\theta_t - \mu_{ij})\right] \mathrm{d}\theta_t \\ &= -\log \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} \frac{I_0(\kappa_{ij})}{2\pi I_0(\kappa_i) I_0(\kappa_j)}. \end{aligned} \tag{A.25}$$

Note that in the last step we have lost explicit dependence on θ_t. But on closer inspection, we can see that the state is implicitly included in κ_{ij} through the difference $\Delta\mu = \mu_i - \mu_j$. We can also utilise the symmetry $\kappa_{ij} = \kappa_{ji}$ in order to reduce the number of terms in the double sum in (A.25)

$$H_2(\theta_t) = -\log \frac{1}{2\pi} \left[\sum_{i=1}^{N} \frac{I_0(2\kappa_i)}{I_0^2(\kappa_i)} + 2 \sum_{\substack{i,j=1 \\ i<j}}^{N} \frac{I_0(\kappa_{ij})}{I_0(\kappa_i) I_0(\kappa_j)} \right]. \tag{A.26}$$

A.5 PRODUCT OF VON MISES-FISHER DISTRIBUTIONS

In Section 5.4 we needed to calculate the product of von Mises-Fisher distributions in order to evaluate the Bayesian update step in the estimation process on the unit sphere. Let us assume that we have two VMF densities $p(\boldsymbol{x}; \boldsymbol{\mu}_i, \kappa_i)$ and $p(\boldsymbol{x}; \boldsymbol{\mu}_j, \kappa_j)$, then their product is proportional to

$$\begin{aligned} p(\boldsymbol{x}; \boldsymbol{\mu}_i, \kappa_i) p(\boldsymbol{x}; \boldsymbol{\mu}_j, \kappa_j) &\propto \exp\{\kappa_i \boldsymbol{\mu}_i^{\mathrm{T}} \boldsymbol{x} + \kappa_j \boldsymbol{\mu}_j^{\mathrm{T}} \boldsymbol{x}\} \\ &= \exp\{(\kappa_i \boldsymbol{\mu}_i + \kappa_j \boldsymbol{\mu}_j)^{\mathrm{T}} \boldsymbol{x}\}. \end{aligned} \tag{A.27}$$

In order to obtain a VMF-like argument in the exponent we will normalize the vector $(\kappa_i \boldsymbol{\mu}_i + \kappa_j \boldsymbol{\mu}_j)$

$$p(\boldsymbol{x}; \boldsymbol{\mu}_i, \kappa_i) p(\boldsymbol{x}; \boldsymbol{\mu}_j, \kappa_j) \propto \exp\{\|\kappa_i \boldsymbol{\mu}_i + \kappa_j \boldsymbol{\mu}_j\| \frac{(\kappa_i \boldsymbol{\mu}_i + \kappa_j \boldsymbol{\mu}_j)^{\mathrm{T}}}{\|\kappa_i \boldsymbol{\mu}_i + \kappa_j \boldsymbol{\mu}_j\|} \boldsymbol{x}\} = \exp\{\kappa_{ij} \boldsymbol{\mu}_{ij}^{\mathrm{T}} \boldsymbol{x}\}, \tag{A.28}$$

where

$$\begin{aligned} \kappa_{ij} &= \|\kappa_i \boldsymbol{\mu}_i + \kappa_j \boldsymbol{\mu}_j\| = \sqrt{\kappa_i^2 + \kappa_j^2 + 2\kappa_i \kappa_j (\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j)} \\ \boldsymbol{\mu}_{ij} &= \frac{\kappa_i \boldsymbol{\mu}_i + \kappa_j \boldsymbol{\mu}_j}{\|\kappa_i \boldsymbol{\mu}_i + \kappa_j \boldsymbol{\mu}_j\|} = \frac{\kappa_i \boldsymbol{\mu}_i + \kappa_j \boldsymbol{\mu}_j}{\kappa_{ij}}. \end{aligned} \tag{A.29}$$

Thus, after the normalization which brings the Bayes rule, we obtain the final expression

$$p(\boldsymbol{x}; \boldsymbol{\mu}_{ij}, \kappa_{ij}) = \frac{\kappa_{ij}}{4\pi \sinh \kappa_{ij}} \exp\{\kappa_{ij} \boldsymbol{\mu}_{ij}^{\mathrm{T}} \boldsymbol{x}\}. \tag{A.30}$$

A.6 THE NEARLY COORDINATED TURN MODEL

The dynamics of the simulated moving object in Section 6.5 were governed by the nearly coordinated turn model given by [76]

$$\begin{aligned}\boldsymbol{x}_k &= \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{G}\boldsymbol{v}_k \\ &= \begin{bmatrix} 1 & \sin\omega\Delta T/\omega & 0 & -(1-\cos\omega\Delta T)/\omega & 0 \\ 0 & \cos\omega\Delta T & 0 & -\sin\omega\Delta T & 0 \\ 0 & (1-\cos\omega\Delta T)/\omega & 1 & \sin\omega\Delta T/\omega & 0 \\ 0 & \sin\omega\Delta T & 0 & \cos\omega\Delta T & 1 \end{bmatrix} \boldsymbol{x}_{k-1} + \begin{bmatrix} \frac{\Delta T^2}{2} & 0 & 0 \\ \Delta T & 0 & 0 \\ 0 & \frac{\Delta T^2}{2} & 0 \\ 0 & \Delta T & 0 \\ 0 & 0 & \Delta T \end{bmatrix} \boldsymbol{v}_k,\end{aligned} \tag{A.31}$$

where the state vector from the Section 6.2.1 was augmented with the turn rate $\omega = 0.5$ rad/s, and the process noise was simulated with $\boldsymbol{v}_k \sim \mathcal{N}_{3\times 1}(0, 0.5)$.

BIBLIOGRAPHY

[1] M. Deans and M. Hebert, "Experimental comparison of techniques for localization and mapping using a bearing-only sensor", in *Seventh International Symposioum on Experimental Robotics*, 2000.

[2] M. Deans, "Bearings-only localization and mapping", PhD thesis, Carnegie Mellon University, 2005, p. 160.

[3] H. Huang, "Bearing-only SLAM. A vision-based navigation system for autonomous robots", PhD thesis, Queensland University of Technology, 2008, p. 175.

[4] T. Bailey, "Constrained initialisation for bearing-only SLAM", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2003, pp. 1966–1971.

[5] K. Choi, J. Park, Y.-H. Kim, and H.-K. Lee, "Monocular SLAM with undelayed initialization for an indoor robot", *Robotics and Autonomous Systems*, vol. 60, no. 6, pp. 841–851, 2012.

[6] A. Boberg, A. N. Bishop, and P. Jensfelt, "Robocentric mapping and localization in modified spherical coordinates with bearing measurements", in *International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2009, pp. 139–144.

[7] J.-S. Hu, C.-Y. Chan, C.-K. Wang, and C.-C. Wang, "Simultaneous localization of mobile robot and multiple sound sources using microphone array", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 29–34, 2009.

[8] C. Gamallo, C. Regueiro, P. Quintía, and M. Mucientes, "Omnivision-based KLD-Monte Carlo localization", *Robotics and Autonomous Systems*, vol. 58, no. 3, pp. 295–305, 2010.

[9] I. Shimshoni, "On mobile robot localization from landmark bearings", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 971–976, 2002.

[10] I. Loevsky and I. Shimshoni, "Reliable and efficient landmark-based localization for mobile robots", *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 520–528, 2010.

[11] M. Beinhofer, J. Müller, and W. Burgard, "Effective landmark placement for accurate and reliable mobile robot navigation", *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1060–1069, 2012.

[12] L. Montesano, "Detection and tracking of moving objects from a mobile platform. Application to navigation and multi-robot localization", PhD thesis, Universidad de Zaragoza, 2006, p. 243.

[13] P. Giguere, I. Rekleitis, and M. Latulippe, "I see you, you see me: Cooperative localization through bearing-only mutually observing robots", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[14] R. Sharma, "Bearing-only cooperative localization and path-Pplanning of ground and aerial robots", PhD thesis, Brigham Young University, 2011, p. 139.

[15] R. Sharma, R. W. Beard, C. N. Taylor, and S. Quebe, "Graph-based observability analysis of bearing-only cooperative localization", *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 522–529, 2012.

[16] T. Krajník, J. Faigl, V. Vonásek, K. Košnar, M. Kulich, and L. Přeučil, "Simple yet stable bearing-only navigation", *Journal of Field Robotics*, vol. 27, no. 5, pp. 511–533, 2010.

[17] C. Pradalier and R. Siegwart, "A Bearing-only 2D/3D-homing method under a visual servoing framework", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4062–4067.

[18] V. Aidala and S. Hammel, "Utilization of modified polar coordinates for bearings-only tracking", *IEEE Transactions on Automatic Control*, vol. AC-28, no. 3, pp. 283–194, 1983.

[19] N. Peach, "Bearings-only tracking using a set of range-parameterised extended Kalman filters", *IEEE Proceedings – Control Theory and Applications*, vol. 142, no. 1, p. 73, 1995.

[20] B. La Scala and M. Morelande, "An analysis of the single sensor bearings-only tracking problem", in *International Conference on Information Fusion*, 2008, pp. 1–6.

[21] L.-L. Ong, "Non-gaussian representations for decentralised Bayesian estimation", PhD thesis, The University of Sydney, 2007, p. 202.

[22] D. Herceg, I. Marković, and I. Petrović, "Real-time detection of moving objects by a mobile robot with an omnidirectional camera", in *International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2011, pp. 289–294.

[23] H. Koyasu, J. Miura, and Y. Shirai, "Recognizing moving obstacles for robot navigation using real-time omnidirectional stereo vision real-time omnidirectional stereo", *Journal of Robotics and Mechatronics*, vol. 14, no. 2, pp. 147–156, 2002.

[24] J. Kim and Y. Suga, "An omnidirectional vision-based moving obstacle detection in mobile robot", *International Journal of Control, Automation, and Systems*, vol. 5, no. 6, pp. 663–673, 2007.

[25] M. Kobilarov, G. Sukhatme, J. Hyams, and P. Batavia, "People tracking and following with mobile robot using an omnidirectional camera and a laser", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 557–562.

[26] J.-M. Valin, F. Michaud, and J. Rouat, "Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering", *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 216–228, 2007.

[27] Y. Sasaki, Y. Kagami, S. Thompson, and H. Mizoguchi, "Sound localization and separation for mobile robot teleoperation by tri-concentric microphone array", *Digital Human Symposium*, 2009.

[28] J. Ferreira, C. Pinho, and J. Dias, "Implementation and calibration of a Bayesian binaural system for 3D localisation", in *IEEE International Conference on Robotics and Biomimetics*, 2009, pp. 1722–1727.

[29] V. Trifa, G. Cheng, A. Koene, and J. Morén, "Real-time acoustic source localization in noisy environments for human-robot multimodal interaction", *IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, pp. 393–398, 2007.

[30] A. Badali, J.-M. Valin, F. Michaud, and P. Aarabi, "Evaluating real-time audio localization algorithms for artificial audition in robotics", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 2033–2038.

[31] B. Kwon, G. Kim, and Y. Park, "Sound source localization methods with considering of microphone placement in robot platform", *IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, pp. 127–130, 2007.

[32] A. Portello, P. Danès, and S. Argentieri, "Active binaural localization of intermittent moving sources in the presence of false measurements", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 3294–3299.

[33] K. Nakamura and K. Nakadai, "Real-time Super-resolution Sound Source Localization for Robots", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 7–12.

[34] K. Okutani, T. Yoshida, K. Nakamura, and K. Nakadai, "Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 3288–3293.

[35] M. Basiri, F. Schill, P. Lima, and D. Floreano, "Robust acoustic source localization of emergency signals from micro air vehicles", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4737–4742.

[36] F. Perrodin, J. Nikolic, J. Busset, and R. Siegwart, "Design and calibration of large microphone arrays for robotic applications", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4596–4601.

[37] Y. Sasaki, M. Kabasawa, S. Thompson, S. Kagami, and K. Oro, "Spherical microphone array for spatial sound localization for a mobile robot", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 713–718.

[38] K. Nakadai, H. Nakajima, K. Yamada, Y. Hasegawa, T. Nakamura, and H. Tsujino, "Sound source tracking with directivity pattern estimation using a 64 channel microphone array", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 1690–1696.

[39] J. Passerieux and D. van Cappel, "Optimal observer maneuver for bearings-only tracking", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 777–788, 1998.

[40] J.-P. Le Cadre and S. Laurent-Michel, "Optimizing the receiver maneuvers for bearings-only tracking", *Automatica*, vol. 35, no. 4, pp. 591–606, 1999.

[41] S. Singh, B.-N. Vo, A. Doucet, and R. Evans, "Stochastic approximation for optimal observer trajectory planning", in *IEEE Conference on Decision and Control*, 2003, pp. 6313–6318.

[42] F. Grondin, D. Létourneau, F. Ferland, V. Rousseau, and F. Michaud, "The ManyEars open framework", *Autonomous Robots*, vol. 34, no. 3, pp. 217–232, 2013.

[43] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications", in *European Conference on Computer Vision (ECCV)*, 2000, pp. 445–461.

[44] P. J. Barreto and H. Araújo, "Issues on the geometry of central catadioptric image formation", in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 422–427.

[45] X. Ying and Z. Hu, "Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model?", in *European Conference on Computer Vision (ECCV)*, 2004, pp. 442–455.

[46] K. V. Mardia and P. E. Jupp, *Directional Statistics*. New York: Wiley, 1999, p. 350.

[47] N. I. I. Fisher, *Statistical Analysis of Circular Data*. Cambridge University Press, 1993, p. 296.

[48] N. I. Fisher, T. Lewis, and B. J. J. Embleton, *Statistical Analysis of Spherical Data*. Cambridge University Press, 1993, p. 329.

[49] S. R. Jammalamadaka and A. Sengupta, *Topics in Circular Statistics*. World Scientific, 2001, p. 322.

[50] R. von Mises, "Uber die 'Ganzzahligkeit' der Atomgewicht und Verwandte Fragen", *Physikalische Zeitschrift*, vol. 19, pp. 490–500, 1918.

[51] H. Shatkay and L. Kaelbling, "Learning geometrically-constrained hidden markov models for robot navigation: bridging the topological-geometrical gap", *Journal of Artifical Intelligence Research*, vol. 16, pp. 167–207, 2002.

[52] N. Roy, G. Gordon, and S. Thrun, "Finding approximate POMDP solutions through belief compression", *Journal of Artificial Intelligence Research*, vol. 23, pp. 1–40, 2005.

[53] D. Nikovski, "State-aggregation algorithms for learning probabilistic models for robot control", PhD thesis, Carnegie Mellon University, 2002, p. 165.

[54] I. Marković and I. Petrović, "Applying von Mises distribution to microphone array probabilistic sensor modelling", in *Proceedings for the Joint Conference of ISR 2010 (41st Internationel Symposium on Robotics) und ROBOTIK 2010 (6th German Conference on Robotics)*, München, Germany, 2010, pp. 21–27.

[55] I. Marković and I. Petrović, "Speaker localization and tracking with a microphone array on a mobile robot using von Mises distribution and particle filtering", *Robotics and Autonomous Systems*, vol. 58, no. 11, pp. 1185–1196, 2010.

[56] I. Marković and I. Petrović, "Bearing-only tracking with a mixture of von Mises distributions", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 707–712.

[57] P. Langevin, "Magnétisme et théorie des électrons", *Annales de chimie et de physique*, vol. 5, pp. 71–127, 1905.

[58] J. Knuth and P. Barooah, "Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[59] K. Bouyarmane, A. Escande, F. Lamiraux, and A. Kheddar, "Potential field guide for humanoid multicontacts acyclic motion planning", in *International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1165–1170.

[60] R. Detry, C. H. Ek, M. Madry, and D. Kragic, "Learning a dictionary of prototypical grasp-predicting parts from grasping experience", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[61] I. Marković, F. Chaumette, and I. Petrović, "Moving object detection, tracking and following using an omnidirectional camera on a mobile robot", in *International Conference on Robotics and Automation (ICRA)*, 2014.

[62] F. Nielsen and V. Garcia, "Statistical exponential families: A digest with flash cards", *Computing Research Repository*, vol. abs/0911.4, 2009.

[63] A. Banerjee, "Clustering on the unit hypersphere using von Mises-Fisher distributions", *Journal of Machine Learning Research*, vol. 6, pp. 1–39, 2005.

[64] A. Tanabe, K. Fukumizu, S. Oba, T. Takenouchi, and S. Ishii, "Parameter estimation for von Mises–Fisher distributions", *Computational Statistics*, vol. 22, no. 1, pp. 145–157, 2007.

[65] D. Amos, "A portable package for Bessel functions of a complex argument and nonnegative order", *Transaction on Mathematical Software*, 1986.

[66] W. Jakob, "Numerically stable sampling of the von Mises-Fisher distribution on S^2 (and other tricks)", Interactive Geometry Lab, ETH Zürich, Tech. Rep., 2012, p. 6.

[67] A. Wood, "Simulation of the von Mises-Fisher distribution", *Communications in statistics – simulation and computation*, vol. 23, no. 1, pp. 157–164, 1994.

[68] S. Sra, "A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of Is(x)", *Computational Statistics*, vol. 27, no. 1, pp. 177–190, Feb. 2011.

[69] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993, p. 625.

[70] T. Bayes, "An essay towards solving a problem in the doctrine of chances", *Philosophical Transactions of the Royal Society of London*, vol. 53, no. 0, pp. 370–418, 1763.

[71] E. T. Jaynes, *Probability Theory: The Logic of Science*. Cambridge University Press, 2003, p. 727.

[72] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2006, p. 645.

[73] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking", *Signal Processing*, vol. 50, no. 2, pp. 174–188, 2001.

[74] A. Doucet, N. de Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods", in *Sequential Monte Carlo Methods in Practice*, A Doucet, N de Freitas, and N Gordon, Eds., Springer-Verlag, 2001, ch. 1, pp. 3–14.

[75] D. Fox, "Adapting the sample size in particle filter through KLD-sampling", *International Journal of Robotics Research*, vol. 22, no. 12, pp. 985–1003, 2003.

[76] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications To Tracking and Navigation*. John Wiley & Sons, Inc., 2001, vol. 9, p. 558.

[77] S. Musick and J. Greenewald, "Comparison of particle method and finite difference nonlinear filters for low SNR target tracking", in *Fourth Annual Conference on Information Fusion*, 2001.

[78] H. Lambert, F. Daum, and J. Weatherwax, "A split-step solution of the Fokker–Planck equation for the conditional density", in *Conference on Signals, Systems and Computers*, 2006, pp. 2014–2018.

[79] D. Salmond, "Mixture reduction algorithms for point and extended object tracking in clutter", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 2, pp. 667–686, 2009.

[80] M. West, "Approximating posterior distributions by mixtures", *Journal of Royal Statistical Society, Series B*, vol. 55, no. 2, pp. 409–442, 1993.

[81] J. L. Williams and P. S. Maybeck, "Cost-function-based Gaussian mixture reduction for target tracking", in *Proceedings of the Sixth International Conference of Information Fusion*, 2003, pp. 338–342.

[82] S. Kullback, *Information Theory and Statistics*. New York: Dover Publications, 1997.

[83] J. Goldberger and H. Aronowitz, "A distance measure between GMMs based on the unscented transform and its application to speaker recognition", in *Proceedings of Interspeech*, 2005, pp. 1985–1989.

[84] A. Runnalls, "Kullback-Leibler approach to Gaussian mixture reduction", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 989–999, 2007.

[85] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman divergences", *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[86] F. Nielsen, "Closed-form information-theoretic divergences for statistical mixtures", in *International conference on Pattern Recognition (ICPR)*, 2012, pp. 1723–1726.

[87] O. Schwander and F. Nielsen, "Learning mixtures by simplifying kernel density estimators", in *Matrix Information Geometry*, F. Nielsen and R. Bhatia, Eds., Springer Berlin Heidelberg, 2013, pp. 403–426.

[88] V. Garcia and F. Nielsen, "Simplification and hierarchical representations of mixtures of exponential families", *Signal Processing*, vol. 90, no. 12, pp. 3197–3212, Dec. 2010.

[89] P. Dognin, J. Hershey, V. Goel, and P. Olsen, "Restructuring exponential family mixture models", in *Proceedings of Interspeech*, 2010, pp. 62–65.

[90] J. DiBiase, H. Silverman, and M. Brandstein, "Robust localization in reverberant rooms", in *Microphone Arrays: Signal Processing Techniques and Applications*, Springer, 2001.

[91] B. Mungamuru and P. Aarabi, "Enhanced sound localization", *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, vol. 34, no. 3, pp. 1526–40, 2004.

[92] M. Fréchette, D. Létorneau, J.-M. Valin, and F. Michaud, "Integration of sound source localization and separation to improve dialogue management on a robot", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2358–2363.

[93] Y. Sasaki, N. Hatao, K. Yoshii, and S. Kagami, "Nested iGMM recognition and multiple hypothesis tracking of moving sound sources for mobile robot audition", in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Invited Session on Robot Audition (IROS)*, 2013, pp. 3930–3936.

[94] E. Di Caludio and R. Parisi, "Multi source localization strategies", in *Microphone Arrays: Signal Processing Techniques and Applications*, Springer, 2001.

[95] S. Argentieri, P. Danès, and P. Souères, "Prototyping filter-sum beamformers for sound source localization in mobile robotics", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 51–56.

[96] P. Danès and J. Bonnal, "Information-theoretic detection of broadband sources in a coherent beamspace MUSIC scheme", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 1976–1981.

[97] K. Nakamura, R. Gomez, and K. Nakadai, "Real-time super-resolution three-dimensional sound source localization for robots", in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Invited Session on Robot Audition (IROS)*, 2013, pp. 3949–3954.

[98] K. Nakadai, K. Hidai, H. Okuno, and H. Kitano, "Real-time multiple speaker tracking by multi-modal integration for mobile robots", *Eurospeech - Scandinavia*, 2001.

[99] J. Merimaa, "Analysis, synthesis, and perception of spatial sound - binaural localization modeling and multichannel loudspeaker reproduction", PhD thesis, Helsinki University of Technology, 2006.

[100] A. Portello, P. Danès, and S. Argentieri, "HRTF-based source azimuth estimation and activity detection from a binaural sensor", in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Invited Session on Robot Audition (IROS)*, 2013.

[101] A. Portello, "Active binaural localization of sound sources in humanoid robotics", PhD thesis, University of Toulouse, 2013.

[102] C. Vina, S. Argentieri, and M. Rébillat, "A spherical cross-channel algorithm for binaural sound localization", in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Invited Session on Robot Audition (IROS)*, 2013, pp. 2921–2926.

[103] K. Youssef, S. Argentieri, and J.-L. Zarader, "A learning-based approach to robust binaural sound localization", in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Invited Session on Robot Audition (IROS)*, 2013, pp. 2927–2932.

[104] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-24, no. 4, pp. 320–327, 1976.

[105] M. Brandstein, J. Adcock, and H. Silverman, "A closed form location estimator for use with room environment microphone arrays", *IEEE Transactions on Speech and Audion Processing*, vol. 5, no. 1, pp. 45–56, 1997.

[106] Y. Chan and K. Cho, "A simple and efficient estimator for hyperbolic location", *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, 1994.

[107] R. Bucher and D. Misra, "A synthesizable VHDL model of the exact solution for three-dimensional hyperbolic positioning system", *VLSI Design*, pp. 507–520, 2002.

[108] K. Dogançay and A. Hashemi-Sakhtsari, "Target tracking by time difference of arrival using recursive smoothing", *Signal Processing*, vol. 85, no. 4, pp. 667–679, 2005.

[109] I. Marković and I. Petrović, "Speaker localization and tracking in mobile robot environment using a microphone array", in *International Symposium on Robotics (ISR)*, J. Basañez, Luis; Suárez, Raúl ; Rosell, Ed., Barcelona: Asociación Española de Robótica y Automatización Tecnologías de la Producción, 2009, pp. 283–288.

[110] T. Nishiura, M. Nakamura, A. Lee, H. Saruwatari, and K. Shikano, "Talker tracking display on autonomous mobile robot with a moving microphone array", in *International Conference on Auditory Display*, 2002, pp. 1–4.

[111] J. Murray, H. Erwin, and S. Wermter, "A recurrent neural network for sound-source motion tracking and prediction", *IEEE International Joint Conference on Nerual Networks*, pp. 2232–2236, 2005.

[112] J. Valin, F. Michaud, J. Rouat, and D. Létourneau, "Robust sound source localization using a microphone array on a mobile robot", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003, pp. 1228–1233.

[113] A. Portello, P. Danès, and S. Argentieri, "Acoustic models and Kalman filtering strategies for active binaural sound localization", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 137–142.

[114] I. Marković, A. Portello, P. Danès, and I. Petrović, "Active speaker localization with circular likelihoods and bootstrap filtering", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 2914–2920.

[115] F. Ribeiro, C. Zhang, and D. A. Florêncio, "Using reverberation to improve range and elevation discrimination for small array sound source localization", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 7, pp. 1781–1792, 2010.

[116] I. Dokmanic and M. Vetterli, "Room helps: acoustic localization with finite elements", in *IEEE Internation Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2617–2620.

[117] C. Ishi, J. Even, and N. Hagita, "Using multiple microphone arrays and reflections for 3D localization of sound sources", in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Invited Session on Robot Audition (IROS)*, 2013, pp. 3937–3942.

[118] N. Kallakuri, J. Even, Y. Morales, C. Ishi, and N. Hagita, "Using sound reflections to detect moving entities out of the field of view", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 5201–5206.

[119] I. Dokmanic, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 30, pp. 12 186–91, Jul. 2013.

[120] M. I. Mandel and D. P. W. Ellis, "A probability model for interaural phase difference", in *ISCA Workshop Statist. Percept. Audio Process (SAPA)*, 2006, pp. 1–6.

[121] M. I. Mandel, R. J. Weiss, and D. P. W. Ellis, "Model-based expectation-maximization source separation and localization", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 2, pp. 382–394, 2010.

[122] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*. Springer, 2001.

[123] J. Chen, J. Benesty, and Y. Huang, "Time delay estimation in room acoustic environments: an overview", *EURASIP Journal on Applied Signal Processing*, vol. 2006, no. 1, pp. 170–189, 2006.

[124] E. Lehmann and A. Johansson, "Particle filter with integrated voice activity detection for acoustic source tracking", *EURASIP Journal on Advances in Signal Processing*, vol. 2007, p. 11, 2007.

[125] J. Vermaak and A. Blake, "Nonlinear filtering for speaker tracking in noisy and reverberant environments", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2001, pp. 3021 –3024.

[126] D. Ward, E. Lehmann, and R. Williamson, "Particle filtering algorithms for tracking an acoustic source in a reverberant environment", *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 826–836, Nov. 2003.

[127] M. Bukal, I. Marković, and I. Petrović, "Composite distance based approach to von Mises mixture reduction", *Information Fusion*, 2014.

[128] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[129] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments", *Signal Processing*, vol. 81, pp. 283–288, 2001.

[130] I. Cohen, "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging", *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 466–475, 2003.

[131] Y. Ephraim and I. Cohen, "Recent Advancements in Speech Enhancement", in *The Electrical Engineering Handbook, Circuits, Signals, and Speech and Image Processing*, R. Dorf, Ed., Boca Raton, FL: CRC Press, 2006, pp. 15–12–15–26.

[132] J. Huang, N. Ohnishi, and N. Sugie, "Sound localization in reverberant environment based on the model of the precedence effect", *IEEE Transactions on Instrumentation and Measurement*, vol. 46, no. 4, pp. 842–846, 1997.

[133] J. Huang, N. Ohnishi, X. Guo, and N. Sugie, "Echo avoidance in a computational model of the precedence effect", *Speech Communication*, vol. 27, no. 3-4, pp. 223–233, 1999.

[134] K. Donohue, J. Hannemann, and H. Dietz, "Performance of phase transform for detecting sound sources with microphone arrays in reverberant and noisy environmnents", *Signal Processing*, vol. 87, no. 7, pp. 1677–1691, 2007.

[135] S. Jurić-Kavelj, M. Seder, and I. Petrović, "Tracking multiple moving objects using adaptive sample-based joint probabilistic data association filter", in *International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS)*, 2008, pp. 99–104.

[136] R. Murray and Y. Morgenstern, "Cue combination on the circle and the sphere", *Journal of Vision*, vol. 10, no. 11, pp. 1–11, 2010.

[137] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums", *Automatica*, vol. 7, no. 1, pp. 465–479, 1971.

[138] S. Amari, "Alpha-divergence is unique, belonging to both f-divergence and bregman divergence classes", *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 4925–4931, 2009.

[139] S. Jurić-Kavelj, I. Marković, and I. Petrović, "People tracking with heterogeneous sensors using JPDAF with entropy based track management", in *European Conference on Mobile Robots (ECMR)*, 2011, pp. 31–36.

[140] T. McGraw, B. Vemuri, B. Yezierski, and T. Mareci, "Von Mises-Fisher mixture model of the diffusion ODF", in *IEEE International Symposium on Biomedical Imaging: Nano to Macro*, 2006, pp. 65–68.

[141] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 2006, p. 776.

[142] A Rényi, *Probability Theory*. London: North-Holland, 1970.

[143] M. Coates, "Distributed particle filters for sensor networks", in *Information Processing in Sensor Networks*, Springer, 2004, pp. 99–107.

[144] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularised particle filters", in *Sequential Monte Carlo Methods in Practice*, A Doucet, N de Freitas, and N Gordon, Eds., Springer-Verlag, 2001, pp. 247–272.

[145] IEEE Subcommittee, "IEEE Recommended Practice for Speech Quality Measurements", *IEEE Transactions on Audio and Electroacoustics*, vol. 17, no. 3, pp. 225–246, 1969.

[146] R. Duda and W. Martens, "Range dependence of the response of a spherical head model", *Journal of the Acoustical Society of America*, vol. 104, no. November, pp. 3048–3058, 1998.

[147] T. Bréhard and J.-P. Le Cadre, "Hierarchical particle filter for bearings-only tracking", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1567–1585, 2007.

[148] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection", *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[149] Y. Cho, K. Al-Naimi, and A. Kondoz, "Improved voice activity detection based on a smoothed statistical likelihood ratio", in *International Conference on Acoustics, Speech and Signal Processing (ICASP)*, 2001, pp. 737–740.

[150] D. K. Kim and J.-H. Chang, "A subspace approach based on embedded prewhitening for voice activity detection.", *The Journal of the Acoustical Society of America*, vol. 130, no. 5, EL304–10, Nov. 2011.

[151] J.-H. Chang and N. S. Kim, "Voice activity detection based on complex laplacian model", *Electronics Letters*, vol. 39, no. 7, p. 632, 2003.

[152] J.-H. Chang, J. Shin, and N. Kim, "Voice activity detector employing generalised gaussian distribution", *Electronics Letters*, vol. 40, no. 24, pp. 25–26, 2004.

[153] J. Ramírez, J. C. Segura, J. M. Górriz, and L. García, "Improved voice activity detection using contextual multiple hypothesis testing for robust speech recognition", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2177–2189, 2007.

[154] J. Ramírez, J. Górriz, and J. Segura, "Statistical voice activity detection based on integrated bispectrum likelihood ratio tests", *Journal of the Acoustical Society of America*, vol. 121, no. 5, pp. 2946–2958, 2007.

[155] J. Górriz, J. Ramírez, E. Lang, C. Puntonet, and I. Turias, "Improved likelihood ratio test based voice activity detector applied to speech recognition", *Speech Communication*, vol. 52, no. 7-8, pp. 664–677, Jul. 2010.

[156] K.-H. Woo, T.-Y. Yang, K.-Y. Park, and C. Lee, "Robust voice activity detection algorithm for estimating noise spectrum", *Electronics Letters*, vol. 36, no. 2, pp. 180–181, 2000.

[157] Q. Li, J. Zheng, A. Tsai, and Q. Zhou, "Robust endpoint detection and energy normalization for real-time speech and speaker recognition", *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 146–157, 2002.

[158] M. Marzinzik and B. Kollmeier, "Speech pause detection for noise spectrum estimation by tracking power envelope dynamics", *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 6, pp. 341–351, 2002.

[159] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and ADABOOST for music classification", *Machine Learning*, vol. 65, no. 2-3, pp. 473–484, Jun. 2006.

[160] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using AANN and GMM", *Applied Soft Computing*, vol. 11, no. 1, pp. 716–723, Jan. 2011.

[161] F. Li and T. Cox, "A neural network model for speech intelligibility quantification", *Applied Soft Computing*, vol. 7, no. 1, pp. 145–155, Jan. 2007.

[162] J. Shin, J.-H. Chang, and N. Kim, "Voice activity detection based on statistical models and machine learning approaches", *Computer Speech & Language*, vol. 24, no. 3, pp. 515–530, Jul. 2010.

[163] Q.-H. Jo, J.-H. Chang, J. Shin, and N. Kim, "Statistical model-based voice activity detection using support vector machine", *IET Signal Processing*, vol. 3, no. 3, p. 205, 2009.

[164] T. Kinnunen, E. Chernenko, M. Tuononen, P. Fränti, and H. Li, "Voice activity detection using MFCC features and support vector machine", in *International Conference on Speech and Computer (SPECOM)*, 2007, pp. 556–561.

[165] J. Ramírez, P. Yélamos, J. Górriz, and J. Segura, "SVM-based speech endpoint detection using contextual speech features", *Electronics Letters*, vol. 42, no. 7, pp. 426–428, 2006.

[166] J. Ramirez, P. Yélamos, J. Górriz, J. Segura, and L. García, "Speech / non-speech discrimination combining advanced feature extraction and SVM learning", in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2006, pp. 1662–1665.

[167] D. Enqing, L. Guizhong, Z. Yatong, and Z. Xiaodi, "Applying support vector machines to voice activity detection", in *International Conference on Signal Processing*, 2002, pp. 1124 –1127.

[168] ITU-T, "A Silence Compression Scheme for G.729 Optimized for Terminals Conforming to Recommendation v70. ITU-T Rec. G. 729, Annex B", International Telecommunication Union, Tech. Rep., 1996.

[169] X.-L. Zhang and J. Wu, "Linearithmic time sparse and convex maximum margin clustering", *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, vol. 42, no. 6, pp. 1669–1692, May 2012.

[170] I. Marković and I. Petrović, "Comparison of statistical model-based voice activity detectors for mobile robot speech applications", in *IFAC Symposioum on Robotic Control (SYROCO)*, 2012, pp. 39–44.

[171] I. Marković, S. Jurić-Kavelj, and I. Petrović, "Partial mutual information based input variable selection for supervised learning approaches to voice activity detection", *Applied Soft Computing*, vol. 2013, no. 13, pp. 4383–4391, 2013.

[172] E. Mumolo, M. Nolich, and G. Verchelli, "Algorithms for acoustic localization based on microphone array in service robotics", *Robotics and Autonomous Systems*, vol. 42, no. 2, pp. 69–88, 2003.

[173] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 137–145, 1980.

[174] Y. Hu and P. Loizou, "Subjective comparison and evaluation of speech enhancement algorithms.", *Speech Communication*, vol. 49, no. 7, pp. 588–601, 2007.

[175] D. Pearce and H.-G. Hirsch, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions", in *ISCA ITRW ASR2000*, 2000, pp. 29–32.

[176] P. Baldi, S. Brunak, Y. Chauvin, C. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: an overview", *Bioinformatics Review*, vol. 16, no. 5, pp. 412–424, 2000.

[177] T. Fawcett, "ROC graphs: notes and practical considerations for researchers", HP Labs Tech Report, Tech. Rep., 2004, pp. 1–38.

[178] R. Krupiński and J. Purczyński, "Approximated fast estimator for the shape parameter of Generalized Gaussian distribution", *Signal Processing*, vol. 86, no. 2, pp. 205–211, 2006.

[179] A. Sharma, "Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 — A strategy for system predictor identification", *Journal of Hydrology*, vol. 239, no. 1-4, pp. 232–239, 2000.

[180] R. May, H. Maier, G. Dandy, and T. Fernando, "Non-linear variable selection for artificial neural networks using partial mutual information", *Environmental Modelling and Software*, vol. 23, no. 10-11, pp. 1312–1326, 2008.

[181] R. Laskar, D. Chakrabarty, F. Talukdar, R. Sreenivasa, and K. Banerjee, "Comparing ANN and GMM in a voice conversion framework", *Applied Soft Computing*, vol. 12, no. 11, pp. 3332–3342, 2012.

[182] C. Kim and R. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition", *IEEE Transactions on Audio, Speech and Language Processing*, Mar. 2012.

[183] C. Kim and R. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recoginition", in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, 2012, pp. 4101–4104.

[184] T. Diethe and J. Shawe-Taylor, "Linear programming boosting for the classification of musical genre", in *Neural Information Processing Systems (NIPS)*, Whistler, Canada, 2007.

[185] G. Bradski, "The OpenCV library", *Dr. Dobb's Journal of Software Tools*, 2000.

[186] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifers", in *Annual ACM Workshop on COL*, 1992, pp. 144–152.

[187] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[188] C.-C. Chang and C.-J. Lin, "LIBSVM : A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, 2:27:1–27:27, 2011.

[189] R. Shapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions", *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[190] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[191] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting", Stanford University, Tech. Rep. 12, Sep. 1998, p. 36.

[192] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*. PWS Publishing Company, 1996.

[193] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm", in *IEEE International Conference on Neural Networks*, Ieee, 1993, pp. 586–591.

[194] B. Jung and G. Sukhatme, "Real-time motion tracking from a mobile robot", *International Journal of Social Robotics*, vol. 2, no. 1, pp. 63–78, 2009.

[195] E. Einhorn, M. Filzhuth, C. Schröter, and H.-M. Gross, "Monocular detection and estimation of moving obstacles for robot navigation", in *European Conference on Mobile Robots (ECMR)*, 2011, pp. 121–126.

[196] C.-M. Oh, Y.-C. Lee, D.-Y. Kim, and C.-W. Lee, "Moving object detection in omnidirectional vision-based mobile robot", *Annual Conference on IEEE Industrial Electronics Society (IECON)*, pp. 4232–4235, 2012.

[197] F. Chaumette and S. Hutchinson, "Visual Servoing and Visual Tracking", in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Springer, 2008, ch. Visual Ser, pp. 563–583.

[198] H. Šamija, I. Marković, and I. Petrović, "Optical flow field segmentation in an omnidirectional camera image based on known camera motion", in *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2011, pp. 165–169.

[199] S. Baker and S. Nayar, "A theory of catadioptric image formation", in *International Conference on Computer Vision (ICCV)*, 1998, pp. 35–42.

[200] L. Puig, J. Bermúdez, P. Sturm, and J. Guerrero, "Calibration of omnidirectional cameras in practice: A comparison of methods", *Computer Vision and Image Understanding*, vol. 116, no. 1, pp. 120–137, Jan. 2012.

[201] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3945–3950.

[202] J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm", Intel Corporation, Microsoft Research Labs, Tech. Rep. 2, 2000, pp. 1–9.

[203] C. Demonceaux and P. Vasseur, "Omnidirectional image processing using geodesic metric", in *IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 221–224.

[204] H. Hadj-Abdelkader, E. Malis, and P. Rives, "Spherical image processing for accurate visual odometry with omnidirectional cameras", in *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS)*, 2008.

[205] A. Brink, *Speeding up the computation of similarity measures based on Minkowski addition in 3D*, 2004.

[206] A. Chiuso and G. Picci, "Visual tracking of points as estimation on the unit sphere", *The Confluence of Vision and Control, Lecture Notes in Control and Information Sciences*, vol. 237, pp. 90–105, 1998.

[207] R. Fomena and F. Chaumette, "Improvements on visual servoing from spherical targets using a spherical projection model", *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 874–886, Aug. 2009.

[208] P. Corke, "Spherical image-based visual servo and structure estimation", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 5550–5555.

[209] J. Richardson and K. Marsh, "Fusion of multisensor data", *The International Journal of Robotics Research*, vol. 7, no. 6, pp. 78–96, 1988.

[210] G. Watson and W. Blair, "Tracking maneuvering targets with multiple sensors using the interacting multiple model algorithm", in *Signal and Data Processing of Small Targets*, vol. 1954, 1993, pp. 438–447.

[211] R. Luo and M. Kay, "Multisensor integration and fusion in intelligent systems", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 901–931, 1989.

[212] D. Hall and J. Llinas, "An introduction to multisensor data fusion", *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[213] J.-H. Choi, Y.-W. Park, J.-B. Song, and I.-S. Kweon, "Localization using GPS and VISION aided INS with an image database and a network of a ground-based reference station in outdoor environments", *International Journal of Control, Automation and Systems*, vol. 9, no. 4, pp. 716–725, 2011.

[214] Y. Nakamura and Y. Xu, "Geometrical fusion method for multi-sensor robotic systems", in *IEEE International Conference on Robotics and Automation (ICRA)*, 1989, pp. 668–673.

[215] R. Luo, M.-H. Lin, and R. Scherp, "Dynamic multi-sensor data fusion system for intelligent robots", *IEEE Journal of Robotics and Automation*, vol. 4, no. 4, pp. 386–396, 1988.

[216] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance", *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 20–44, 1993.

[217] E. Nettleton, H. F. Durrant-Whyte, and S. Sukkarieh, "A robust architecture for decentralised data fusion", in *International Conference on Advanced Robotics (ICAR)*, 2003.

[218] D.-J. Lee, "Nonlinear estimation and multiple sensor fusion using unscented information filtering", *IEEE Signal Processing Letters*, vol. 15, pp. 861–864, 2008.

[219] C. Martin, E. Schaffernicht, A. Scheidig, and H.-M. Gross, "Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking", *Robotics and Autonomous Systems*, vol. 54, no. 9, pp. 721–728, 2006.

[220] S. Julier and J. Uhlmann, "A Non-divergent estimation algorithm in the presence of unknown correlations", in *Proceedings of the 1997 American Control Conference*, American Autom. Control Council, 1997, pp. 2369–2373.

[221] M. Rosencrantz, G. Gordon, and S. Thrun, "Decentralized sensor fusion with distributed particle filters", in *Conference on Uncertainty in AI (UAI)*, Acapulco, Mexico, 2003.

[222] Y. Chen and Y. Rui, "Real-time speaker tracking using particle filter sensor fusion", *Proceedings of the IEEE*, vol. 92, no. 3, pp. 485–494, 2004.

[223] L.-L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh, and H. F. Durrant-Whyte, "Consistent methods for decentralised data fusion using particle filters", in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2006, pp. 85–91.

[224] L.-L. Ong, T. Bailey, H. Durrant-Whyte, and B. Upcroft, "Decentralised particle filtering for multiple target tracking in wireless sensor networks", in *International Conference on Information Fusion*, 2008, pp. 1–8.

[225] H. F. Durrant-Whyte and T. C. Henderson, "Multisensor Data Fusion", in *Handbook of Robotics*, Springer, 2008, pp. 585–610.

[226] H. Hashemipour, S. Roy, and A. Laub, "Decentrailzed structures for paraller Kalman filtering", *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, 1988.

[227] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems", in *International Symposium on Aerospace/Defense Sensing, Simulate and Control*, vol. 54, Orlando, FL, 1997.

[228] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators", *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.

[229] E. Parzen, "On estimation of a probability density function and mode", *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[230] I.-H. Seo and T.-L. Song, "Out-of-sequence-measurement processing for probabilistic multiple hypothesis tracker with measurement reordering", *International Journal of Control, Automation and Systems*, vol. 8, no. 2, pp. 301–307, 2010.

[231] K. Torkkola, "Feature extraction by non-parametric mutual information maximization", *Journal of Machine Learning Research*, vol. 3, no. 7-8, I. Guyon and A. Elisseeff, Eds., pp. 1415–1438, 2003.

[232] E. Prassler, J. Scholz, and A. Elfes, "Tracking mulitple moving objects for real-time robot navigation", *Journal of Autonomous Robots, Special Issue on Perception for Mobile Agents*, vol. 8, 2008.

[233] D. Reid, "An algorithm for tracking multiple targets", *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[234] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association", *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, Jul. 1983.

[235] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters", *The International Journal of Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003.

[236] A. Almeida, J. Almeida, and R. Araújo, "Real-time tracking of multiple moving objects using particle filters and probabilistic data association", *AUTOMATIKA Journal*, vol. 46, no. 1-2, pp. 39–48, 2005.

[237] A. Kraussling and D. Schulz, "Tracking extended targets—A switching algorithm versus the SJPDAF", in *International Conference on Information Fusion*, 2009, pp. 1–8.

[238] N. Bellotto and H. Hu, "Vision and laser data fusion for tracking people with a mobile robot", in *IEEE International Conference on Robotics and Biomimetics*, Ieee, 2006, pp. 7–12.

[239] I. J. Cox, "A review of statistical data association techniques for motion correspondence", *International Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[240] S. Neil and L. Pao, "Multisensor fusion algorithms for tracking", in *American Control Conference*, vol. 2, 1993, pp. 1–5.

[241] Y. Bar-Shalom, "Extension of the probabilistic data association filter to multi-target environment", in *Symposium on Nonlinear Estimation*, 1974, pp. 16–21.

[242] K. O. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities", in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2008, pp. 1710–1715.

[243] M. Luber and K. O. Arras, "Spatially grounded multi-hypothesis tracking of people", in *International Conference on Robotics and Automation (ICRA), Workshop on People Detection and Tracking*, 2009.

[244] M. Seder and I. Petrović, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 1986–1991.

[245] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House Publishers, 1999.

[246] S. Jurić-Kavelj and I. Petrović, "Experimental comparison of AdaBoost algorithms applied on leg detection with different range sensor setups", in *International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*, 2010, pp. 267–272.

[247] P. Viola and M. Jones, "Robust real-time face detection", *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[248] L. M. Bregman, "The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming", *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200–217, 1967.

CURRICULUM VITAE

IVAN MARKOVIĆ was born in Osijek, Croatia in 1985. He finished math-oriented high school in Osijek in 2003 and received B.Sc. degree in electrical engineering from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia in 2008. His research interests include estimation theory, robot audition and omnidirectional vision.

In 2008 he was employed by S.C.A.N. Ltd. where he worked on developing control algorithms and supervision interfaces for terminal automation systems. In 2009 he joined FER and the Department of Control and Computer Engineering as a research assistant funded by the Ministry of Science, Education and Sport, Republic of Croatia where he enrolled PhD studies the same year. In 2009 and 2012 he was a member of the National Organizing Committee of two international conferences and since 2009 he is the coordinator of the technical editing team of the journal "Automatika – Journal for Control, Measurement, Electronics, Computing and Communications." In collaboration with Siemens Croatia Inc. he has been a trainer for TIA BASIS and TIA SERV2 training programmes. He has actively participated in the following projects: "Control of mobile robots and vehicles in unknown dynamic environments", Ministry of Science, Education and Sport of the Republic of Croatia (2007-2013) and "ACROSS – Centre of Research Excellence for Advanced Cooperative Systems, European FP7 project, (2011-2014). In 2013 he was a visiting researcher at INRIA Rennes-Bretagne Atlantique, Rennes, France. Besides scientific work, he has also been a teaching assistant at several undergraduate and graduate courses.

He is a graduate student member of IEEE and KoREMA. During the studies, he was receiving scholarship for exceptional students from the Ministry of Science, Education and Sports of the Republic of Croatia. As a recognition for successful studying, he received the "Josip Lončar" award in the academic year 2005/2006 from FER, and for the overall success the "INETEC award" from the Institute of Nuclear Technology. In the sequel the list of papers published at the time of the writing of the doctoral thesis is presented.

PUBLICATIONS

JOURNAL PUBLICATIONS:

1. M. Bukal, I. Marković, and I. Petrović. Composite distance based approach to von Mises mixture reduction. *Information Fusion*, doi: 10.1016/j.inffus.2014.01.003

2. I. Marković, S. Jurić-Kavelj, and I. Petrović. Partial mutual information based input variable selection for supervised learning approaches to voice activity detection. *Applied Soft Computing*, 13(13):4383–4391, 2013.

3. I. Marković and I. Petrović. Speaker localization and tracking with a microphone array on a mobile robot using von Mises distribution and particle filtering. *Robotics and Autonomous Systems*, 58(11):1185–1196, November 2010.

CONFERENCE PUBLICATIONS:

1. I. Marković, F. Chaumette, and I. Petrović. Moving object detection, tracking and following using an omnidirectional camera on a mobile robot. In *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014*.

2. I. Marković, A. Portello, P. Danès, and I. Petrović. Active speaker localization with circular likelihoods and bootstrap filtering. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Invited Session on Robot Audition*, pages 2914–2920, Tokyo, Japan, 2013.

3. I. Marković and I. Petrović. Bearing-only tracking with a mixture of von Mises distributions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Invited Session on Robot Audition*, pages 707–712, Vila Moura, Portugal, 2012.

4. I. Marković and I. Petrović. Comparison of statistical model-based voice activity detectors for mobile robot speech applications. In *IFAC Symposioum on Robotic Control (SYROCO)*, pages 39–44, Dubrovnik, Croatia 2012.

5. D. Herceg, I. Marković, and I. Petrović. Real-time detection of moving objects by a mobile robot with an omnidirectional camera. In *International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 289–294, Dubrovnik, Croatia, 2011.

6. S. Jurić-Kavelj, I. Marković, and I. Petrović. People tracking with heterogeneous sensors using JPDAF with entropy based track management. In *European Conference on Mobile Robots (ECMR)*, pages 31–36, Örebro, Sweden, 2011.

7. H. Šamija, I. Marković, and I. Petrović. Optical flow field segmentation in an omnidirectional camera image based on known camera motion. In *34th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 165–169, Opatija, Croatia, 2011.

8. I. Marković and I. Petrović. Applying von Mises distribution to microphone array probabilistic sensor modelling. In *Proceedings for the Joint Conference of ISR 2010 (41st Internationel Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 21–27, München, Germany, 2010.

9. I. Marković and I. Petrović. Speaker localization and tracking in mobile robot environment using a microphone array. In *International Symposium on Robotics (ISR)*, pages 283–288, Barcelona, Spain, 2009.

RESEARCH STUDIES:

1. I. Petrović, J. Matuško, S. Jurić-Kavelj, I. Marković, V. Petrović, N. Perić. Mathematical models for short-term prediction of hourly electrical energy consumption for distribution areas. In Croatian, pages 92, 2011.

ŽIVOTOPIS

IVAN MARKOVIĆ je rođen 1985. godine u Osijeku. Treću gimnaziju u Osijeku završio je 2003. godine te je diplomirao u polju elektrotehnike na Sveučilištu u Zagrebu, Fakultetu elektrotehnike i računarstva (FER) 2008. godine. Njegovi istraživački interesi uključuju teoriju estimacije, obradbu govora za robotske primjene i svesmjernu viziju.

Po završetku studija zapošljava se u tvrtki S.C.A.N. d.o.o. gdje radi na razvijanju algoritama upravljanja i nadzora u sustavima automatizacije naftnih terminala. U 2009. godini zapošljava se na FER-u na Zavodu za automatiku i računalno inženjerstvo u zvanju znanstvenog novaka financiranog od strane Ministarstva znanosti, obrazovanja i sporta te iste godine upisuje i doktorske studije. U 2009. i 2012. godini bio je član organizacijkog odbora dviju međunarodnih znanstvenih konferencija, a od 2009. godine koordinator je tehničkog uredništva časopisa „Automatika – časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije." U suradnji s tvrtkom Siemens Hrvatska d.d. predavač je na tečajevima TIA BASIS i TIA SERV2. Aktivno je sudjelovao na znanstvenim projektima „Upravljanje mobilnim robotima i vozilima u nepoznatim i dinamičkim okruženjima," Ministarstva znanosti, obrazovanja i sporta (2007-2013) i „ACROSS – znanstveni centar izvrsnosti za napredne i kooperativne sustave", EU FP7 projekt (2011-2014). U 2013. godini bio je gostujući istraživač na ustanovi INRIA Rennes-Bretagne Atlantique u Rennesu, Francuska. Uz znanstveni rad asistent je i na nekoliko dodiplomskih i diplomskih predmeta.

Član je udruge IEEE i KoREMA. Tijekom studija primao je stipendiju za nadarene studente Ministarstva znanosti, obrazovanja i sporta Republike Hrvatske. Također, dobitnik je nagrade „Josip Lončar" Fakulteta elektrotehnike i računarstva za postignut uspjeh tijekom 2005./2006. akademske godine te za cjelokupan studij dobitnik je „INETEC nagrade" Instituta za nuklearnu tehnologiju.

COLOPHON

This document was typeset and inspired by the typographical look-and-feel `classicthesis` developed by André Miede, which was based on Robert Bringhurst's book on typography "The Elements of Typographic Style", and by the `FERBook` developed by Jadranko Matuško.