Doctoral theses at NTNU, 2019:4

Olofsson, Jonatan

On Multi-UAS Sea Ice Monitoring

Doctoral Thesis

Olofsson, Jonatan

ISBN 978-82-326-3606-8 (printed version)
ISBN 978-82-326-3607-5 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2019:4

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU

NTNU
Norwegian University of
Science and Technology

Olofsson, Jonatan

On Multi-UAS Sea Ice Monitoring

Thesis for the degree of Philosophiae Doctor

Trondheim, January 2019

Norwegian University of Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-326-3606-8 (printed version)
ISBN 978-82-326-3607-5 (electronic version)
ISSN 1503-8181

ITK-report: 2018-20-W

Doctoral theses at NTNU, 2019:4

MILJØMERKET
241 TRYKKSAK 731

Printed by Skipnes Kommunikasjon as

Summary

Work in the polar regions of our planet is unavoidably linked with hazards such as drift ice. Increased presence, fueled by economic interests in the Arctic, has for several decades called for research in the field of ice management. The field deals with the detection, tracking and forecasting of ice, but also the physical actions taken to avoid collisions. Managing ice is of great importance to polar ventures but predicting ice movement has proven difficult, concluding that observations are essential for tracking.

Ice management is applicable not only for stationary installations, but has also been studied for the protection of ship routes in the Arctic. A moving object may be not be able to rely solely on satellite imagery, as these generally have limited availability and coverage, with sample times in the order of days. Recent studies have introduced the use of Unmanned Aerial Systems (UAS's) as a supplement to local sensors such as ship- or ground-based radar.

This thesis aims to provide a detailed insight into the design of a scalable sea ice tracking system with components ranging, from data collection, classification and tracking, to the feedback of previous tracking results into the formation of new paths for multiple UAS agents to collect new observations. Beyond applying the UAS as a sensor platform, examples are also given in this thesis of the use of machine learning algorithms and background modeling to extract and classify sea ice detections from Synthetic Aperture Radar (SAR) and ground-based radar.

In the business of tracking ice objects in the Arctic, objects are tracked over a geographically vast area. While each observation covers only a relatively limited area, a complete system needs to handle large number of ice objects. Commonly, Multiple Target Tracking (MTT) algorithms scale poorly with target numbers, which poses a problem to the large scale of our scenario. This thesis goes into detail on selected approaches for enabling large-scale MTT. First, we introduce spatial indexing for a fast partitioning scheme for the Multiple Hypothesis Tracker (MHT) and Labeled Multi-Bernoulli (LMB) algorithms. Further, we propose a novel formulation of the LMB filter designed to simplify its implementation. The resulting filters are detailed, implemented and applied to sea ice tracking scenarios.

To utilize the information available from the tracker, we explore two applications in particular. First, we study probabilistic modeling of current and wind velocities based on tracker data and Gaussian fields. Second, we propose the use of the Probability Hypothesis Density (PHD) for informed planning of UAS flight paths. The PHD — the density of expected number of tracked objects — can be efficiently extracted from the same data structures used in e.g. the LMB filter, and is thus used to form a common "language" between the two algorithm families of target trackers and path planners. A proof of concept multi-agent path planner is developed and published open-source along with implementations of the MHT and LMB filters.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Philosophiae Doctor (PhD) at Norwegian University of Science and Technology (NTNU). The research presented in this thesis has been conducted at the Department of Engineering Cybernetics (ITK) at NTNU, and the work has been supervised by Professor Thor Inge Fossen, as well as co-supervised by Professor Tor Arne Johansen and Associate Professor Gustaf Hendeby from Linköping University (LIU). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 642153, MarineUAS, as well as the Research Council of Norway through the Centres of Excellence funding scheme, grant number 223254 — NTNU-AMOS. The thesis contains modified Copernicus Sentinel data captured 2015–2016. The Terrestrial Radar Interferometer (TRI) and UAS datasets have been provided by the Norut Northern Research Institute in Tromsø.

Acknowledgements

First and foremost, I would like to express my gratitude towards my supervisor Thor I. Fossen, and my co-supervisors Tor Arne Johansen and Gustaf Hendeby. The trust and help I have received has been humbling, and the possibilities truly life-changing. Further, I would like to thank Fredrik Gustafsson at LIU for establishing the connections which led to my PhD opportunity in Trondheim.

During the past years, I have had many appreciated colleagues and office-mates in both Trondheim and Linköping. My family situation may not have allowed me to spend as much time with everyone as I would have wanted, but I have always been warmly welcomed at both universities. Special thanks goes out to Svante Gunnarsson and Martin Enqvist at LIU who kindly provided a second office at LIU and made sure I felt included.

I have further much appreciated the academic collaboration with Clas Veibäck in Linköping, as well as Andreas Lindahl Flåten and Edmund Brekke at NTNU. I have also had the pleasure of working together with Norut in Tromsø — Tom Rune Lauknes, Agnar Sivertsen and Rune Storvold and others have been much appreciated colleagues and friends on my visits.

My PhD position has been a part of the MarineUAS Innovative Training Network, and I have thoroughly enjoyed working together with the other fellows, Christopher Dahlin Rodin, Anthony Hovenburg, Leopoldo Rodriguez Salazar, Fotios Balampanis, Du Ho Duc, Praveen Jain, Juan Braga, Bahareh Sabetghadam, Nguyen Tuan Hung, Alojz Gomola, Margarida Costa Faria, Fabio Andrade, Mads Bornebusch and my much appreciated friend and travel-mate, Andreas Wenz. Further, the training network would not have been possible without all the supervisors who made our workshops and meet-ups possible. I would also like to thank all the administrative staff who have helped me during the years, and special thanks to the UAS pilots Pål Kvaløy and Lars Semb.

Throughout my PhD, I have had the opportunity to travel with extraordinary people, including but not limited to Andreas Wenz, Andreas Lindahl Flåten, Erik Wilthil, Artur Zolich, Frederik Stendahl Leira, Kim Lynge Sørensen, Kasper Trolle Borup, Bardhyl Hajdini as well as all the representatives at FUSION2017 from NTNU and LIU. A special mention is also given to the 2017 PhD group in the AT-834 course in Longyearbyen, Svalbard: Per R. Leikanger, Ida Lemström, Dennis Monteban, Brian Murray, Runa A. Skarbø

In the final stretch, I have been gracefully helped with proofreading of this thesis. My sincere thanks for this service goes out to Gustaf Hendeby, Zoran Sjanic, Angela Fontan, Clas Veibäck, Per Boström, Martin Skoglund and Kristoffer Bergman.

Finally, I would like to thank my friends and my large and very much appreciated family for the encouragement and support not just throughout my PhD but in my entire life.

Anna, I love you dearly! Thank you for keeping us afloat! (and for all our dives together, for that matter)

December 2018
Jonatan Olofsson

On the 10th of April 1912 Johan Carlsson, an uncle of my grandmother's, boarded a ship in England to head for a new life the in the New World. He left from Liverpool, as he had been delayed in his transport to England and thus unable to board the ship he had originally planned for: the *RMS Titanic*, which left Southampton on the same day. Five days later, the *RMS Titanic* struck an iceberg in the North Atlantic and went down with an estimated 1, 517 casualties. This disaster triggered modern conventions of maritime safety, as well as the active monitoring of sea ice conditions, which is studied in this thesis.

Contents

List of Figures

List of Tables

Acronyms

AUV Autonomous Underwater Vehicle

BLUE Best Linear Unbiased Estimator

CBMeMBer Cardinality-Balanced Multiple-Target Multi-Bernoulli

CDF Cumulative Distribution Function

CID Corrected Inverse Distance

CIRFA Norwegian Centre for Integrated Remote Sensing and Forecasting for Arctic Operations

CPHD Cardinalized Probability Hypothesis Density

CPP Coverage Path Planning

ECEF Earth-Centered, Earth-Fixed

EKF Extended Kalman Filter

EM electromagnetic

ESA European Space Agency

EXP Exponential

FISST Finite Set Statistics

FOSS Free and Open Source Software

FOV Field-of-View

δ-GLMB δ-Generalized Labeled Multi-Bernoulli

GNN Global Nearest Neighbour

GOSPA Generalized Optimal Sub-Pattern Assignment

GPRI GAMMA Portable Radar Interferometer

GRD Ground Range Detected

IIP International Ice Patrol

IPP Informed Path Planning

IR infrared

ITK Department of Engineering Cybernetics

JPDA Joint Probabilistic Data Association

KD-TREE K-Dimensional tree

LAP Linear Assignment Problem

LIU Linköping University

LL Latitude-Longitude

LLA Latitude-Longitude-Altitude

LMB Labeled Multi-Bernoulli

LSD-TREE Local Split Decision tree

MeMBer Multiple-Target Multi-Bernoulli

MHT Multiple Hypothesis Tracker

MTT Multiple Target Tracking

NCV Nearly Constant Velocity

NE North-East

NED North-East-Down

NLL Negative Log Likelihood

NTNU Norwegian University of Science and Technology

OSPA Optimal Sub-Pattern Assignment

PDF Probability Density Function

PHD Probability Hypothesis Density

PSSM Probabilistic State-Space Model

RBF Radial Basis Function

RCS Radar Cross-Section

RFS Random Finite Set

RPAS Remotely Piloted Aerial System

RQ Rational Quadratic

RRT Rapidly-exploring Random Tree

RRT* Rapidly-exploring Random Tree*

R-TREE Rectangle tree

SAR Synthetic Aperture Radar

SE Squared Exponential

SLC Single Look Complex

SNR Signal-to-Noise Ratio

STT Single-Target Tracking

SVM Support Vector Machine

TOMHT Track-Oriented Multiple Hypothesis Tracker

TRI Terrestrial Radar Interferometer

UAS Unmanned Aerial System

WMO World Meteorological Organization

Notation

Mathematical Style

Notation	Meaning
a	Generic scalar
$\boldsymbol{a}$	Generic vector
$\boldsymbol{A}$	Generic matrix
$\mathcal{A}$	Generic set

Set Notation

Notation	Meaning
$\mathbb{R}^{a}$	Space of real column vectors of dimension a
$\mathbb{R}^{a \times b}$	Space of real matrices of dimension $a \times b$
$\mathbb{N}$	Set of natural numbers
$\mathcal{F}(\mathcal{X})$	This operator exhaustively iterates the collection of all subsets of set $\mathcal{X}$
$b \in [a, \ldots, c]$	$a \in \mathbb{N}, a \leq b \leq c$
$\lvert\mathcal{A}\rvert$	Size of set $\mathcal{A}$
$h^{\mathcal{X}}$	Multi-object exponential notation, such that $h^{\mathcal{X}} \triangleq \prod_{x \in \mathcal{X}} h(x)$ or $h^{\mathcal{X}} \triangleq \prod_{x \in \mathcal{X}} h_x$. $h^{\emptyset} = 1$ by convention
$\delta_{\mathcal{Y}}(\mathcal{X})$	Kronecker delta-function, used to select summands relevant to exactly the set $\mathcal{Y}$; $\delta_{\mathcal{Y}}(\mathcal{X}) \triangleq \begin{cases} 1, & \text{if } \mathcal{X} = \mathcal{Y}, \\ 0, & \text{otherwise,} \end{cases}$
$1_{\mathcal{Y}}(\mathcal{X})$	This defines the inclusion function, such that $1_{\mathcal{Y}}(\mathcal{X}) \triangleq \begin{cases} 1, & \text{if } X \subseteq Y, \\ 0, & \text{otherwise,} \end{cases}$ This is used to select summands relevant to a set Y of which X is a subset. If X is singular, $X = \{x\}$, the notation $1_Y(x)$ is used

Probability Notation

Notation	Meaning
$p\left(\boldsymbol{a}\right)$	Probability Density Function (PDF) of $\boldsymbol{a}$
$p\left(\boldsymbol{a}\vert\boldsymbol{b}\right)$	PDF of $\boldsymbol{a}$, conditioned on $\boldsymbol{b}$
$\boldsymbol{a} \sim p\left(\boldsymbol{a}\right)$	$\boldsymbol{a}$ is sampled from the distribution $p\left(\boldsymbol{a}\right)$
$\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Pi}\right)$	Multivariate Gaussian PDF with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Pi}$
$\mathcal{N}\left(\boldsymbol{a}\vert\boldsymbol{\mu}, \boldsymbol{\Pi}\right)$	Multivariate Gaussian PDF for vector $\boldsymbol{a}$, with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Pi}$
$\hat{\boldsymbol{a}}$	Estimate of $\boldsymbol{a}$
$\mathbb{E}\left[\boldsymbol{a}\right]$	Expected value of $\boldsymbol{a}$
$\operatorname{cov}\left(\boldsymbol{a}\right)$	Covariance (matrix) of $\boldsymbol{a}$

Vector & Matrix Notation

Notation	Meaning
$\boldsymbol{0}$	Column vector of zeros
$\boldsymbol{I}_a$	The identity matrix in $\mathbb{R}^{a \times a}$, with ones in the diagonal, zeros elsewhere
$\boldsymbol{a}^\intercal$ or $\boldsymbol{A}^\intercal$	Transpose of vector $\boldsymbol{a}$ or matrix $\boldsymbol{A}$
$\boldsymbol{a}^*$ or $\boldsymbol{A}^*$	Complex conjugate transpose of vector $\boldsymbol{a}$ or matrix $\boldsymbol{A}$
$\delta_{\boldsymbol{a}}\left(\boldsymbol{x}\right)$	Dirac delta-function; $\delta_{\boldsymbol{a}}\left(\boldsymbol{x}\right) \triangleq \begin{cases} 1, & \text{if } \boldsymbol{x} = \boldsymbol{b}, \\ 0, & \text{otherwise,} \end{cases}$
$\vert\boldsymbol{a}\vert$	2-norm of $\boldsymbol{a}$
$\vert\boldsymbol{A}\vert$	Determinant of $\boldsymbol{A}$
$\boldsymbol{A}_{ab}$	Element in matrix $\boldsymbol{A}$ at row a and column b
rows($\boldsymbol{A}$) and cols($\boldsymbol{A}$)	Number of rows and columns, respectively, of matrix (or vector) $\boldsymbol{A}$

Other Notation

Notation	Meaning
$\lvert a \rvert$	Absolute value of a
$a \vert b$	Variable a is conditioned on b
$\left\langle f, g \right\rangle$	Inner product of functions f and g, such that $\left\langle f, g \right\rangle \triangleq \int f(x)\, g(x)\, dx$
$\frac{\partial x}{\partial y}$	The (partial) derivative of x, with respect to y
$\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right\rvert_{\boldsymbol{x}=\boldsymbol{a}}$	The Jacobian matrix of all partial derivatives of $\boldsymbol{f}(\boldsymbol{x})$, with respect to each component of $\boldsymbol{x}$, evaluated at $\boldsymbol{x} = \boldsymbol{a}$. A matrix element on row i and column j corresponds to the partial derivative $\left.\frac{\partial f(x)_i}{\partial x_j}\right\rvert_{x=a}$
$A^{\Theta}_{\boldsymbol{z}\leftrightarrow\ell}$	Assignment indicator function, defined as $A^{\Theta}_{\boldsymbol{z}\leftrightarrow\ell} \triangleq \begin{cases} 1, & \text{if } \Theta \text{ assigns label } \ell \text{ to report } \boldsymbol{z}, \\ 0, & \text{otherwise.} \end{cases}$
$\mathcal{O}(\cdot)$	Big O notation, describing the limiting behaviour of the complexity of an operation

Symbols

k	Time index
K	End time/planning horizon
r	Bernoulli probability parameter
ϑ	Assignment index
φ	Track index
r	Report index
i	Row index
j	Column index
ℓ	Target
κ	Planner time index
ξ	Extraneous report
ζ	Cluster index
ζ	Cluster
i	Observation index
α	Agent index
a	Agent
$\boldsymbol{f}$	Prediction function
$\boldsymbol{F}$	Prediction matrix
$\boldsymbol{h}$	Measurement function
$\boldsymbol{H}$	Measurement matrix
$\boldsymbol{K}$	Kalman gain
$\boldsymbol{S}$	Innovation covariance

f Coordinate transform

$\boldsymbol{f}_*$ Gaussian field evaluation values

$\bar{\boldsymbol{f}}_*$ Gaussian field evaluation estimate

$\boldsymbol{K}$ Gaussian field covariance

$\boldsymbol{v}$ Gaussian field verification vector

Υ Gaussian field verification covariance

f Gaussian field function

γ Gaussian field evaluation score

ν Gaussian field verification vector index

$\mathcal{X}_s$ Gaussian field sample points

$\boldsymbol{y}_s$ Gaussian field measurements

$\boldsymbol{x}$ Generic state

σ_v Velocity covariance

$\boldsymbol{P}$ Filter covariance

$\hat{\boldsymbol{x}}$ State estimate

$\boldsymbol{z}$ Measurement report

$\hat{\boldsymbol{z}}$ Expected (estimated) report

$\mathcal{Z}$ Set of reports, or a scan

$\mathcal{X}$ Set of states

$\mathcal{P}$ Path set

χ Random set

c Path planning cost

β Path planning budget

b Path planning birth model probability hypothesis density

ϖ Probability of first observation

ρ Reward function

R Reward function integral

$N_{\mathcal{X}}$ Number of targets

N_{φ} Number of tracks

$N_{\mathcal{Z}}$ Number of reports

N_x Number of states

N_s Number of sampes

$N_{\mathcal{P}}$ Number of paths

N_d Number of dimensions

N_z Number of measured states

N_{ζ} Number of clusters

N_p Number of particles, or components

$N_{\mathcal{M}}$ Number of mixtures

N_h Number of hypotheses

N_{mb} Number of Multi-Bernoulli Components

w Weight

$\boldsymbol{\mu}$ Gaussian mean parameter

$\boldsymbol{\Pi}$ Gaussian covariance parameter

$\mathcal{M}$ Gaussian mixture component set

ΔT_k Time delta

v Probability Hypothesis Density

V Integrated probability hypothesis density

$\mathcal{L}$ Labels

m Mass

$\boldsymbol{B}$ Axis-Aligned Bounding-box

$\boldsymbol{a}$ Acceleration

$\dot{\boldsymbol{x}}$ Velocity vector

$\boldsymbol{w}$ Model noise

$\boldsymbol{v}$ Measurement noise

$\boldsymbol{Q}$ Model covariance

$\boldsymbol{R}$ Measurement covariance

$\vec{f}$ Force vector

$\Omega\left(\vartheta\right)$ Parent hypothesis mapping

$\boldsymbol{\iota}$ Innovation

γ_{ε} Covariance elliptic bound standard deviation

θ Assignment

Θ Hypothesis

$\mathcal{I}$ Label subset

$\boldsymbol{C}$ Cost matrix

σ Radar cross-section

$\mathscr{A}$ Set of agents

S Field-of-View

$\boldsymbol{k}_p$ SAR Pauli representation

S SAR scattering

$\boldsymbol{k}$ SAR scattering vector

$\boldsymbol{S}$ SAR scattering matrix

$\boldsymbol{C}_c$ SAR covariance matrix

$\boldsymbol{T}$ SAR coherency matrix

1 Introduction

The countries of the European Union share an interest in the waters to which it borders. The care for marine resources and natural environment is becoming an increasingly pressing issue, while economic interests at the same time call for cost efficiency and increased exploitation. Monitoring the coastal regions of the European Union is a vast operation which in recent years have started to benefit from the advances in technology of unmanned systems. Collaborative technology utilizing buoys, autonomous ships, underwater vehicles, satellites and unmanned aerial systems is quickly becoming a reality, enabling new possibilities in e.g. marine transport. For example, by mitigating dangers posed by drift ice in the Arctic transportation route north of Russia, the Northern Sea Route, it could be possible to use new shipping routes, effectively halving the distance between Europe and Asia.

Predicting ice movements have proved difficult however, and today's practice of manned flights by e.g. the Canadian Ice Service are expensive and — compared to potential autonomous solutions — ineffective. In this thesis, we aim to explore the groundworks of a full scale sea ice tracking system and propose methods to make use of the possibilities of the new technologies and close the loop by exploiting the information collected by all the system's agents. As we seek to track a very large amount of targets, we also explore ways of increasing the scalability of Multiple Target Tracking (MTT) algorithms.

1.1 Research Applications

To exemplify the potential application of the research presented in this thesis, two initial scenarios may be studied:

Scenario 1.1: Protection of Arctic Ship Routes

Shipping routes in the Arctic is inherently associated with risk of sea ice collisions. A globally available early warning system would be able to mitigate those risks and increase the safety of Arctic transports. Satellite imagery is available with Arctic coverage. However, with satellite observation as the only source of information, such a system would only be able to issue very broad warnings.

In this scenario, visualized in Figure 1.1, an Unmanned Aerial System (UAS) could be carried on-board a container ship and be launched to maintain a highly updated regionally detailed view to complement the overview satellites are able to provide. It would also be able to actively look for ice objects that may have impact on the safety of the shipping route. With detections sent to a global service, ships can take advantage of data collected from nearby ships.

The varying scales and the moving area of interest puts requirements on the partitionability and scalability of the algorithms used for tracking the ice. This would also apply to any active route planning employed to optimize the paths for active agents searching for sea ice hazards.

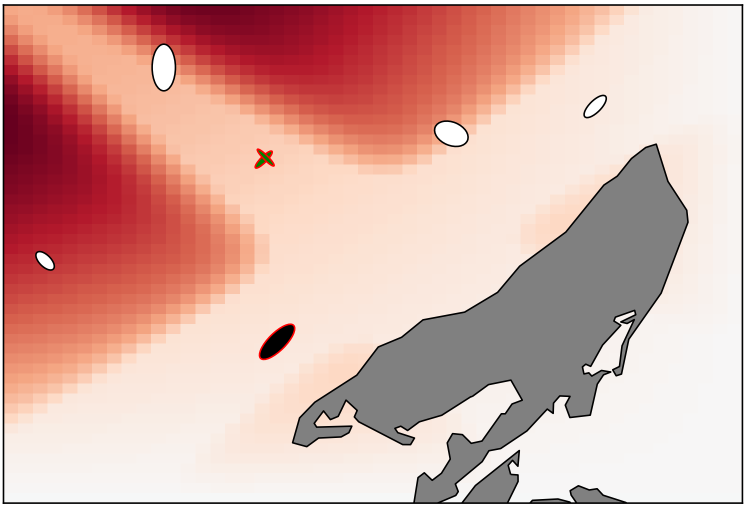


Figure 1.1: Illustration of the ship route protection scenario, where the black ellipse symbolizes the position of a container ship, and the white ellipses symbolizes tracked ice objects. The crossed ellipses symbolizes a UAS that has been launched from the ship to maintain overview and look for ice in the area. The more red in the background, the more important it is considered for continued surveillance

The idea of a global early warning system based on the combination of all available sensors warrants the following questions:

Q1	*How can we track large amounts of sea ice globally?*

Q2	*How can we combine large scale (global) and small-scale (local) tracking?*

Q3	*How can we make sure only the relevant parts of the tracker is updated with new measurements?*

Today, it is reasonable to assume that a map is available of all static landmasses in the region of interest. Satellite imagery provides a good foundation for ice monitoring, but can provide neither the high resolution which is desirable for detecting smaller objects, nor the sample rate required for reliable tracking. Other applicable sensors include stationary marine radars, which may provide high resolution measurements with high sample rate, but is restricted to the line-of-sight of the sensor.

Note that detected objects may be associated with motion models and properties such as

- position, velocity, mass;
- uncertainty of the measurements; and
- risk / assessed damage of impact.

All these properties changes over time, and the only way to find out how is to take new measurements.

Scenario 1.2: Protection of Static Assets in Sea Infested Areas

Another fueling factor for presence in the Arctic waters is the availability of oil in the region. Like all other Arctic operations it is exposed to the hazards of sea ice, but in the sensitive Arctic environment it has the potential of catastrophic impact in case of an incident. It is therefore of utmost importance in such operations to maintain monitoring of the surrounding sea ice conditions.

In this scenario, a system is studied which tracks and detects moving drift sea ice around a stationary protected asset. A visualisation of the scenario is provided in Figure 1.2.

Radar sensors around the installation will provide an overview of the region, but for closer inspection a sensor may need to be brought closer. However, local observations entails that only a limited part of the region can be observed with limited time and resources, and there is a need to prioritize which area to observe. From the available data about wind and currents, we assume in the scenario that drift ice is most likely to approach from the north-east, and so dispatch a UAS in that direction to gather more data.

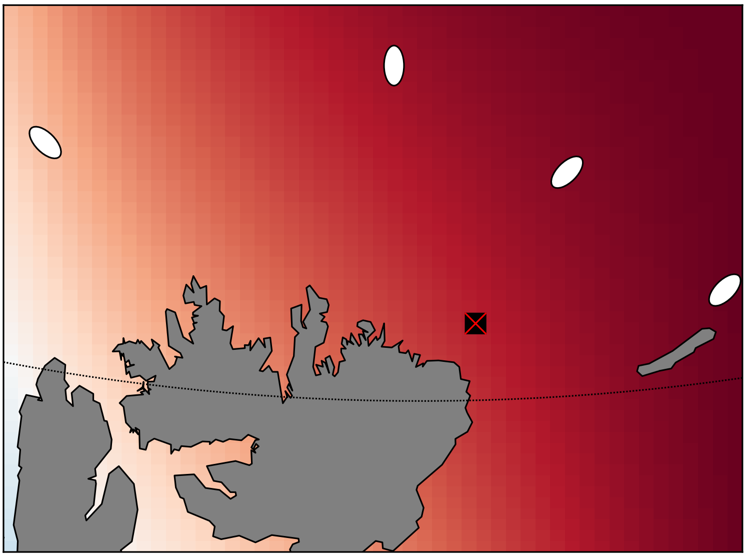


Figure 1.2: Static rig protection. In this image, the cross indicates the position of e.g. an oil rig, and the white ellipses tracked ice objects. The more red in the background, the more important it is considered for continued surveillance

The need to prioritize time and resources, especially with regards to a limited set of UAS agents with limited flight time, raises the following questions:

Q4	*How can we maximize the relevance of future acquired data?*

Q5	*How is the route affected if a hazard is found? How is the route affected if a hazard is **not** found?*

Q6	*How is the expected value of exploration affected by the modeled motion of objects?*

These questions can be studied as an optimisation problem with the goal of retrieving a search pattern — optimal in some sense — with restrictions such as end position (landing site), start position (current location) and (remaining) length of flight. Further, during the flight, any number of objects may be found and require recalculation of the proposed flight path.

1.2 Objectives

With an increasing range of sensors and data available to gather information on sea ice, we have studied the application of sensor fusion to this collected intelligence. Theory and practical implementation have been pursued for a system for probabilistic data fusion and decision support for ice management.

Given the questions stated in Section 1.1, the main objectives we wish to address in this thesis are to:

I. Research scalable multi-target, multi-sensor methods for global tracking of ice objects.

II. Explore the use of machine learning to make use of the information in the tracking data.

III. Improve the predictive capabilities through integrated mapping of currents.

IV. Develop a statistical representation of the geographical distribution of exploration value.

V. Optimize information acquisition through the exploitation of tracker data and statistical models.

1.3 Publications

The following list of publications contribute towards the purpose of this thesis, grouped by type and listed in chronological order of publication.

For all publications included in this thesis, the thesis author is the principal author.

Conference Papers

Paper A: Olofsson, J., Brekke, E., Fossen, T. I., and Johansen, T. A. (2017a). Spatially indexed clustering for scalable tracking of remotely sensed drift ice. In *IEEE Aerospace Conference Proceedings*, Big Sky, MT, USA. © 2017 IEEE.

This paper, forming the base for Chapter 5, explores the application of spatial indexing to the Multiple Hypothesis Tracker (MHT) algorithm. The author contributed with the idea, writing of the article, implementation and application examples. ¶

Paper B: Olofsson, J., Brekke, E., and Johansen, T. A. (2017b). Cooperative remote sensing of ice using a spatially indexed labeled multi-Bernoulli filter. In *International Conference Unmanned Aircraft Systems (ICUAS)*, Miami, USA. © 2017 IEEE.

This paper introduces the PYTHON Labeled Multi-Bernoulli (LMB) implementation and discusses how spatial indexing can be applied in the clustering process. Further, it presents a simulated scenario with combined satellite and UAS measurements of sea ice. The author contributed with the idea, writing of the article, algorithm implementation and simulated application examples. ¶

Paper C: Olofsson, J., Veibäck, C., and Hendeby, G. (2017d). Sea ice tracking with a spatially indexed labeled multi-Bernoulli filter. In *20th International Conference on Information Fusion (FUSION)*, Xi'an, China. © 2017 IEEE.

The contributions of this paper are twofold. First, we introduce a reformulation of the LMB filter, better suited for implementation. This is condensed into a simple algorithm which simplifies the calculation of the filter's probabilities. Second, the filter is applied to a sea ice tracking scenario using a dataset sampled by a Terrestrial Radar Interferometer (TRI) sensor stationed by Kongsfjorden, Svalbard. The author contributed with the idea, writing of the article and the algorithm implementations. ¶

Paper D: Olofsson, J., Flåten, A. L., Veibäck, C., and Lauknes, T. R. (2017c). Gaussian field current estimation from drift sea ice tracking with the labeled multi-Bernoulli filter. In *Proceedings of OCEANS 2017 MTS/IEEE*, Anchorage, Alaska, USA. © 2017 IEEE.

Here, we extend the results of Paper C, applying the result of the tracking scenario to form a velocity map of the tracked region, using Gaussian Fields. The results from this paper are presented in Chapter 7. The author contributed with the idea, writing of the article, implementation and application examples. ¶

Paper E: Olofsson, J., Veibäck, C., Hendeby, G., and Johansen, T. A. (2017e). Outline of a system for integrated adaptive ice tracking and multi-agent path planning. *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems, RED-UAS 2017*. doi: 10.1109/RED-UAS.2017.8101636. © 2017 IEEE.

This paper outlines the principles and components used to formulate the work that has resulted in this thesis. The article presents combined results of previous papers, and present an overview of how they connect in the bigger picture. The author contributed with the idea, writing of the article, implementation and application examples. ¶

Journal Articles

Paper F: Olofsson, J., Hendeby, G., Lauknes, T. R., and Johansen, T. A. (2018). Multi-Agent Informed Path-Planning Using the Probability Hypothesis Density. *Autonomous Robots*, (Submitted, Aug 2018). © 2017 IEEE.2018 IEEE

In this paper, we propose a multi-agent metric for path planning, using the Probability Hypothesis Density (PHD) to optimize the expected number of detected targets along the route. The paper derives the metric and applies it to propose paths for multiple agents, in a setting based of two datasets recorded by TRI and UAS respectively. The author contributed with the idea, writing of the article, implementation and application examples. ¶

Additionally, the following paper has been co-authored by the author of this thesis, although it is not included in the thesis:

- Veibäck, C., Olofsson, J., and Hendeby, G. (2018). Learning Target Dynamics While Tracking Using Gaussian Processes. *Transactions on Aerospace and Electronic Systems*, (Under review, June 2018)

Open-Sourced Software

For the papers above, four Free and Open Source Software (FOSS) libraries were developed and published by the author of this thesis.

MHT A PYTHON library demonstrating the integration of spatial indexing and the MHT algorithm.
The library is available at https://github.com/jonatanolofsson/mht

PYLMB This PYTHON library is a reference implementation for the reformulation of the LMB filter proposed in Chapter 6.
The library is available at https://github.com/jonatanolofsson/lmb

CLMB This is a C++ library with PYTHON bindings developed as a successor for the LMB library above, optimized for computational efficiency and improved parallelization.
The library is available at https://github.com/jonatanolofsson/clmb

PPL In Chapter 8, a multi-agent planner based on the PHD is proposed. This PYTHON library provides the implementation of the Monte Carlo/genetic optimization routine used for evaluation.
The library is available at https://github.com/jonatanolofsson/ppl

1.4 Structure of the Thesis and Main Contributions

The thesis is structured into three main parts.

In Part I, we present the background to the thesis along with an overview of our proposed structure for a system for large scale tracking of sea ice in Chapter 2. The major components of this system have been the foundation for the research presented in this thesis and are, as such, reflected in the structure of the subsequent chapters of the thesis. In Chapter 3, we also present the equipment and data sets which have been used to evaluate the research, as well as elaborate on the properties of sea ice relevant to its detection and tracking.

The contributions of Chapter 2 include:

- an overview of a modern scalable design for sea ice tracking.

The contributions of Chapter 3 include:

- a study of the use of machine learning segmentation for the extraction of detections from Synthetic Aperture Radar (SAR) satellite imagery.

In Part II, we present our research on scaling MTT algorithms by proposing data structures and optimizations which enable our applications to track thousands of targets and more. In Chapter 4 we lay the groundwork by presenting the concepts of MTT and spatial indexing, the combination of which is explored in Chapter 5 and Chapter 6. In Chapter 5 the discussion is focused on the application of spatial indexing to the MHT algorithm, whereas Chapter 6 introduces the spatially indexed LMB filter, along with other proposed improvements to the filter. Details to FOSS implementations of said algorithms and improvements are also given in their respective chapter.

The contributions of Chapter 5 include:

- a detailed description of a FOSS MHT implementation which uses spatial indexing;
- a comparative study of the scalability improvements from clustering and spatial indexing; and
- an application of the tracking algorithm to real satellite imagery with added simulated movements.

The contributions of Chapter 6 include:

- a novel simplifying reformulation of the LMB filter better suited for implementation;
- a simulated example of the collaboration of multiple types of sensors, with different sampling rates and Field-of-View (FOV);

- a tracking scenario from TRI data; and
- a description of the open-source algorithm implementation in PYTHON and C++ respectively.

In Part III, we present two utilizations of the statistics that can be extracted from the target tracking described in Part II. Chapter 7 describes the application of Gaussian Fields to the target track data, showing how velocity estimates can be used to predict the motion profiles of other sea ice objects. This can be used to improve the estimates e.g. through improved instantiation of new targets.

For the purpose of collecting more information about the area of interest, in Chapter 8 we propose how the measure of the PHD — a central concept in the emerging class of MTT algorithms of which the LMB filter is a part of — can be employed as a common language in the interface between target tracking and path planning. Easily extracted from the statistics kept by e.g. the LMB tracker, the chapter describes how the PHD can be used to close the information gathering loop by proposing flight paths for the very agents responsible for collecting the data used by the tracker.

The contributions of Chapter 7 include:

- a description of Gaussian Fields and its application to velocity field mapping from tracking data; and
- a velocity field mapping application from TRI data.

The contributions of Chapter 8 include:

- a novel PHD based reward function for Informed Path Planning (IPP) for the static and dynamic cases;
- a decoupled approximative reward function that allows for computationally efficient multi-agent IPP;
- a suggestion of how to solve the obtained optimization problem by sampling full path proposals; and
- a description of how to apply the method in two applications, which is also used to evaluate the method.

Finally concluding comments are given in Chapter 9.

Part I

Background

2 Sea Ice Tracking System Design

This chapter presents a background and an outline of the system whose components are presented throughout this thesis. The outline is based on the work in

Paper E: Olofsson, J., Veibäck, C., Hendeby, G., and Johansen, T. A. (2017e). Outline of a system for integrated adaptive ice tracking and multi-agent path planning. *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems, RED-UAS 2017*. doi: 10.1109/RED-UAS.2017.8101636. © 2017 IEEE.

The contributions of this chapter include:

- an overview of a modern scalable design for sea ice tracking.

Drift sea ice awareness is essential to enable safe operations in Arctic regions. Recent years have seen the rise of Unmanned Aerial System (UAS) as a platform for geoobservation, and so too for the tracking of sea ice. Being a mobile platform, the research on UAS path planning is extensive and usually involves an objective-function to minimize. For the purpose of observation however, the objective-function typically changes as observations are made along the path. Further, the general problem involves multiple UAS and — in the case of sea ice tracking — vast geographical areas.

In this chapter we discuss the architectural outline of a system capable of fusing data from multiple sources — UAS's and others — as well as incorporating that data for both path planning, sea ice movement prediction and target initialization. The system contains tracking of sea ice objects, situation map logic and is expandable, as discussed, with path planning capabilities for closing the loop by optimizing paths based on previous observations for the purpose of continued information acquisition.

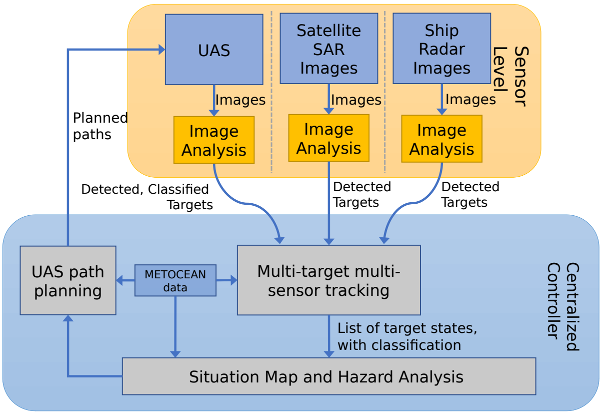


Figure 2.1: Outline of a system for global sea ice monitoring and management

2.1 Introduction

Drift sea ice is a major limiting factor in regions with Arctic conditions, often restricting the operational season to a few months of the year (Hnatiuk, 1983). Through planning and organization however, it is possible to extend the season and to perform safe operations in Arctic waters. Ice management is the field encompassing all activities to reduce or avoid impact from ice features (Eik, 2008). In general terms, this includes detection, tracking and forecasting of ice features but also the threat evaluation and the physical management of breaking or towing ice objects (Eik, 2008). Traditionally, ship-mounted radars and visual detection from e.g. manned flights have been used to detect potential threats, to aid in the manual decision process to launch countermeasures (Wright, 2000; Eik, 2008). Also, in severely ice-plagued operations, ice-breakers have been used to pre-emptively break down large upstream ice floes to manageable pieces (Moran et al., 2006).

Recent years have seen increased availability of new types of sensor platforms — autonomous unmanned systems under [Autonomous Underwater Vehicle (AUV)] and over [Unmanned Aerial System (UAS)] water (Haugen, 2014) — capable of aiding in the situation awareness of the operation. Not only are these new classes of platforms capable of operating with greater availability than previous sensors, but potentially also at a reduced cost.

With a limited Field-of-View (FOV), sensors rely on moving to cover larger areas. Whereas traditionally sensors have been attached to man-controlled vessels, the development of autonomy for these sensors involve the automated planning of new routes (Albert et al., 2017; Leira et al., 2017; Olofsson et al., 2018) which the carrier should follow autonomously to acquire relevant sensor data. As data is acquired, the planning relies on the feedback of findings into the consideration of route re-planning. This has been studied for single-UAS optimal control in e.g. Leira et al. (2017). In Leira et al. (2017), the proposed framework is an optimal-control scheme generating UAS intermediate-level guidance, whereas the proposed approach in this thesis is to provide high-level agent-generic paths which, for each agent, can be converted locally to low-level control signals. Further, ice tracking and path planning using occupancy grids have been explored in e.g. Hals and Skjønhaug (2017); Langeveld (2017).

In this chapter we wish to focus on the bigger picture and put each of the central subsystems of a modern scalable tracking system in context. In Figure 2.1, an outline is presented with the overview of a system for the continuous integration of sensor input and situation awareness into the optimization problem of assigning sensor paths which maximize the utility of each sensor. As reflected in Figure 2.1, the general tracking problem formulation is that of multi-agent, multi-target tracking. On a sensor level, expanded upon in Section 2.2, many sensors of varying types can contribute to the full image of the situation.

In our proposed architecture, the data from each sensor is fed into a central controller — see Section 2.3 — which summarizes the information for both user presentation and optimization of paths for future data acquisition. For this purpose, we also discuss — in Section 2.3.2 — a set of factors available for consideration in a path planning optimization problem.

2.2 Sensor Level

A wide variety of sensors can be of use in the process of detecting and tracking sea ice, and a comprehensive summary can be found in Haugen (2014). Sensors with varying FOV are mounted to a carrier — a movable sensor platform, such as a ship or a UAS. One way to categorize the sensors and sensor platforms is as

stationary sensors with known, fixed position and FOV, such as ground-based radar;

traceable sensors with known but not fixed position and FOV, albeit not controllable, such as satellites; and

controllable sensors with known position and FOV, with the ability to be commanded to new locations, such as UAS's.

Whereas all above types of sensors are compatible with the system in Figure 2.1, the feedback from the system can only be applied to controllable sensors.

Each sensor has different ways of measuring the presence of ice, and a sensor platform can be equipped with several types of sensors simultaneously. In the example of Figure 2.2, detections are extracted from infrared imagery (Leira, 2017), although more ways are presented in Chapter 3. Each type of detection has unique characteristics in terms of e.g. noise and error covariance (Gustafsson, 2018). For the purpose of forming a generic situation awareness map, it may be beneficial to standardize the reporting format for measurements — for example by transformation to geodetic coordinates — although for some sensors this may lead to loss of information as e.g. error statistics may be distorted by transformations (Gustafsson, 2018).

Other considerations for the sensor–controller interface involve the inclusion of other properties of the ice, such as — as available in imaging sensors — shape-describing Hu moments (Hu, 1962) of each detection. These added properties might be used to improve association of measurements from consecutive sightings.

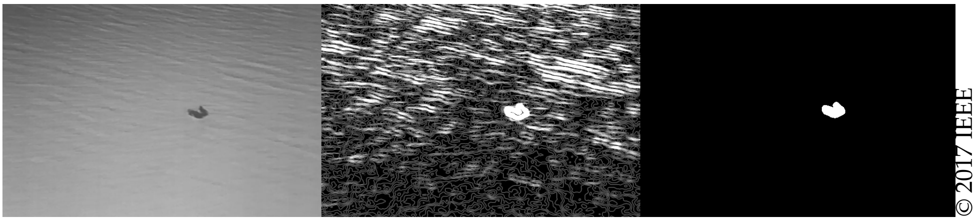
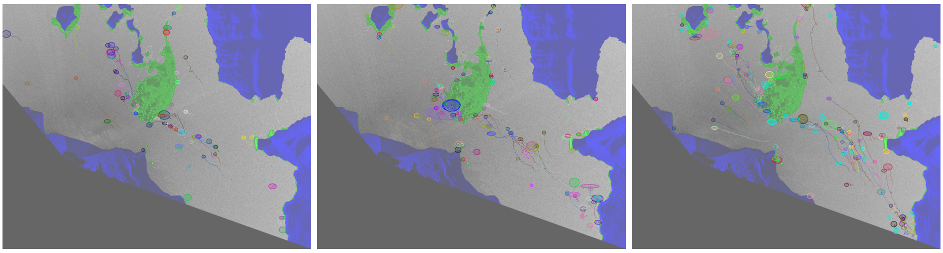


Figure 2.2: Detections of ice extracted from infrared imagery from UAS-carried cameras. Left: Raw thermal image; Middle: Enhanced features; Right: Segmented detected sea ice (Leira, 2017). Image courtesy of Frederik S. Leira.



(a) Tracks after 2 h

(b) Tracks after 4 h

(c) Tracks after 7 h

Figure 2.3: Drift sea ice tracks over time, showing the land mask in blue and stationary detections in green. Tracks and targets retain an individual randomly assigned color over time.

2.3 Centralized Controller

In the proposed architecture, from Figure 2.1, sensor reports are fed to a central controller, responsible for data fusion, situation map and high-level path planning. While its purpose is introduced here, the chapters in Part II and Part III especially focuses on the possibilities of this subsystem.

2.3.1 Data Fusion

The first role of this controller is the fusing of the information from all sensor sources into a combined map of ice objects and ice coverage. The geographical bounds of this map are not necessarily defined, but can span the entire globe with varying resolution. This means consideration has to be taken to the scalability of the map and to the algorithms employed, and in particular to the possibility of performing local updates to a global filter. At the heart of this fusion is a Multiple Target Tracking (MTT) algorithm, where tracks are maintained of each observed ice object. Sea ice coverage mapping can be performed using occupancy grids (Leira, 2017) whereas identifiable individual objects are tracked with sparse target tracking (Haugen, 2014; Olofsson et al., 2017b,d,a). In the former case, scalability can be achieved through variable resolution, whereas in the latter, objects can be stored in efficient data structures (Olofsson et al., 2017a,b) to limit the number of objects involved in each update. MTT and improvements to the scalability of MTT algorithms is discussed in Part II. An example result of the data fusion, from Olofsson et al. (2017d), is found in Figure 2.3 and detailed upon in Chapter 6.

Each choice of map coordinate system will have its advantages and disadvantages due to nonlinearities and discontinuities. As ice tracking is constrained to the water surface, our coordinate system of choice for global tracking is by Latitude-Longitude-Altitude (LLA). Since the problem is constrained to the sea surface, this naturally devolves to a two-dimensional Latitude-Longitude (LL) system. The details on tracker coordinate transformations, and how they have been applied in implementations, are given in Section 4.7.

2.3.2 Situation Map

Another role of the central controller is to form what we call a situation map. The map of tracked objects in Section 2.3.1 is one of the components of this map; other examples include

- sensor locations;
- traffic information;
- weather conditions; and
- information quality.

The main objective of the proposed system is to provide good situational awareness of where the relevant ice objects in the region are, as well as where there are no ice objects. To achieve this the system should reward behaviours resulting in large amounts of high-quality information.

The tracks from the centralized tracking algorithm naturally provide information on where the sea ice is, as well as a metric of the quality of that information — quantified as the inverse of the estimation uncertainty/covariance at each point. This could directly be used to build an information quality map. However, a disadvantage of only considering the resulting tracks from sparse tracking is that no information is available to discern whether an area is empty or simply unobserved — a *lack* of detections in a region also provides information on the situation that should be incorporated into the map. One aspect that is handled by the tracking algorithm naturally is that information quality of track information degrades over time without further observations.

Another option to the information quality estimated by the tracking algorithm — which mainly focuses on information gathered on the tracked targets — is to directly consider the full, dense, map of where information has been collected by the sensors. This is done in Skoglar (2012) where the information in each point over a grid is estimated using a large number of extended information filters. This can be rather costly for large high-resolution grids, but intrinsically handles the difference of no detections versus no observations. The choice of a scalar comparable metric of the information amount is a relevant issue for both approaches, also discussed in Skoglar (2012). Relevant choices include e.g. the trace of the information matrix, as a function of map coordinates. This would be used to simplify the evaluation of the quality of information at each point.

Weather and oceanic conditions (METOCEAN data) can be incorporated into the situation map in order to allow the motion planner to, for example, account for winds, and to avoid hazardous situations for the vehicles. In Arctic conditions, icing and severe winds is a real problem which need to be managed either through mitigation, as in e.g. Sørensen et al. (2015), or through temporary cancellation of flight operations.

In addition to the awareness of the physical properties in an area, the situation map should also take into account user priorities. This can be provided either by an operator or by some other source, such as traffic information. For example, it is more useful to have high-quality information in regions with heavy traffic than in those in lack of human activity. Besides constructing the information used by the subsequent agent path planner, the same data can be used to create other situation maps, tuned for example for presentation to operators.

2.3.3 Path Planning

Since UAS's are mobile sensor platforms they need to be controlled. Control of an aircraft is a multi-level control problem, and whereas low-level control is assumed to be handled locally by each agent, the situation awareness of the centralized controller can be used to provide each UAS with a high-level path to follow. This path could in the simple case be a pre-determined route to scout along, but in the more advanced case, feedback control for the data acquisition is possible.

A desirable objective of the path planning is to provide good situational awareness of the relevant ice objects in the region, as well as where there is open water. To achieve this the objective function should reward paths resulting in larger amounts of information of high user relevance. A model of this quantity, given the path, can be obtained using the information quality map discussed in Section 2.3.2.

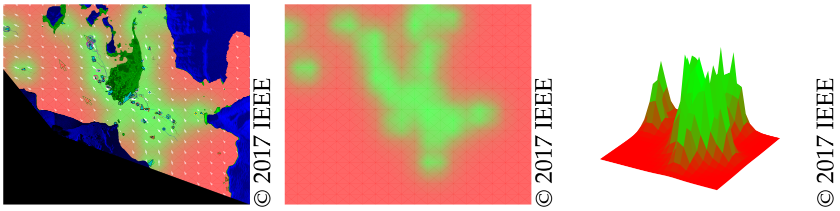
Formulations and solutions to problems of this form are discussed in Skoglar (2012), including constraints such as dynamics of the vehicles and field of view. A concern with the solver algorithm used in Skoglar (2012), namely receding horizon, is that it quickly becomes very costly. Mitigating solutions often include finding sub-optimal paths by reducing the time horizon. Other path planning methods, such as Multiple-shooting methods (Bock and Plitt, 1984), Rapidly-exploring Random Tree (RRT) (LaValle, 1998, 2006) and Monte Carlo (Hollinger and Sukhatme, 2014; Olofsson et al., 2018) can also be considered. For Chapter 8, a Monte Carlo-based planner was implemented and used to demonstrate multi-agent Informed Path Planning (IPP).

A path planner generally generates paths for its agents by trying to optimize an objective function, and one approach is to leverage a combined function of the maps discussed in Section 2.3.2 by weighted summation. Conceptually, the local value can then be calculated as

$$m(\boldsymbol{x}) = \sum_i w_i m_i(\boldsymbol{x}). \tag{2.1}$$

for maps $m_i(\boldsymbol{x})$ (each of which are functions of respective inputs, such as previous observations) and their weights.

This general metric yields a two-dimensional landscape which can be navigated to find an optimal path. For example, an applied metric — shown in Figure 2.4 — can be constructed to minimize the uncertainty (as



(a) Gaussian fields estimate with both the velocity field in an image as well as its estimated model covariance (ranging from red (high) to green (low)).

(b) The information map is determined by the trace of the Gaussian field velocity covariance in each point.

(c) The data in Figure 2.4b can also be interpreted as a 3D landscape to be navigated (z-axis is inverted to show areas of low information as valleys).

Figure 2.4: Information metric example using Gaussian field covariance

determined by the trace of the covariance) of expected velocity estimates in a Gaussian field, constructed by the tracker velocity estimates. Another metric — the maximization of the expected observed Probability Hypothesis Density (PHD) — is proposed in Chapter 8. Note that the execution of the plan to follow the path will yield information which possibly warrants re-planning.

Further, note that the summation in (2.1) implies that to form a two-dimensional map $\{m(x)\}_{x\in\mathcal{X}}$ over region $\mathcal{X}$, information is only needed from the individual maps $\{m_i(x)\}_{x\in\mathcal{X}}$, i.e. the cut-outs from the same region of each map. This means that not only can we partition the map into independent parts for storage and evaluation, it also trivial to for example assign different areas to different groups of agents.

Besides optimizing the objective function, the optimization routine should also take into account other limitations of the problem, such as the maximum total (remaining) range of an agent. In the path planning, multiple agents can be planned for simultaneously, to collaborate towards a common goal.

In Chapter 8, a path-planner is proposed which is specifically constructed to optimize the expected number of detections throughout its flight. This optimization is based on the metric of the observed PHD, which is derived from a staple metric of an emerging class of MTT algorithms, with algorithms such as the PHD filter (Mahler and Zajic, 2001) and the Labeled

Multi-Bernoulli (LMB) filter presented in Chapter 6.

2.4 Conclusions

In this chapter we have discussed the high-level design of an ice management system capable of tracking ice and feeding back the information it collects to the optimization of paths for future data acquisition. The principle is based on a feedback loop with information from tracker-generated high-level paths as the interface to the moving agents. The overview in Figure 2.1 also serves to outline the structure of this thesis.

Section 2.2 introduced the three types of sensors used in this thesis — stationary, traceable and controllable. These are represented in the continued thesis by the Terrestrial Radar Interferometer (TRI), Synthetic Aperture Radar (SAR) sensors and the UAS platform, respectively.

The requirements of a central controller module capable of fusing data from all these sources were discussed in Section 2.3, with particular focus on sensor fusion capabilities, a situation awareness map and the capability of utilizing the data from the tracker to propose future actions and flight paths.

Importantly, all the algorithms of the proposed system can be partitioned to deal with only relevant separable parts of each problem. This is essential to scaling their applicability up to the challenge of handling the large amounts of sea ice in the geographical expanse that is the Arctic.

Detecting Ice in the Ocean

The central application discussed in this thesis is the tracking of sea ice and utilizations of said tracking. This chapter introduces properties of ice relevant for detection and tracking, as well as the main sensors and datasets used in the applications of later chapters.

The contributions of this chapter include:

- a study of the use of machine learning segmentation for the extraction of detections from Synthetic Aperture Radar (SAR) satellite imagery.

As indicated in Figure 3.1, and relating to the system outline presented in Chapter 2, this chapter covers the main parts of the sensor level, detailing both sensor platforms, sensing technologies and the analysis of the output to extract detections of sea ice targets.

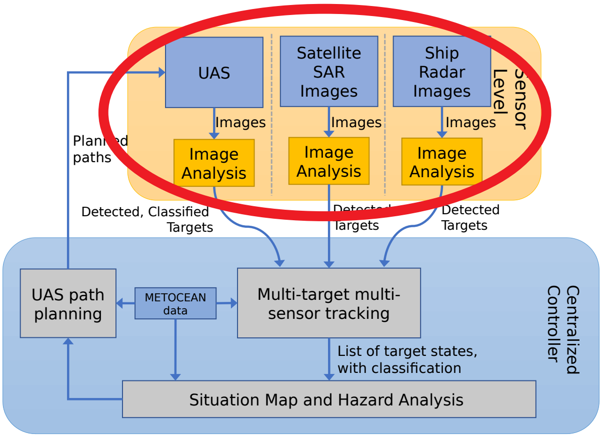


Figure 3.1: Ice-monitoring sensors in context of the system from Chapter 2

3.1 Introduction

Sea ice tracking has been carried out in polar regions since the aftermath of the infamous sinking of the *RMS Titanic* in 1912. The incident led, as part of the implementation of the *International Convention for the Safety of Life at Sea*, to the foundation of the International Ice Patrol (IIP). This organization has since monitored and reported on the ice conditions in the Atlantic and Arctic oceans.

In the early 20th century, almost none of the modern technological sensing abilities of today were available. Today, satellite data can provide a general overview, although manned flights and manual observations still represent the primary reconnaissance work of the IIP.

Ice can be detected using a number of sensors and sensor platforms. In this chapter, we summarize the satellite-carried SAR, the ground-based Terrestrial Radar Interferometer (TRI) and electro-optical cameras carried by a Unmanned Aerial System (UAS) platform. An overview including other sensors available for sea ice monitoring is given in e.g. Haugen (2014) and Lubin and Massom (2006).

Following this introduction, the chapter continues — in Section 3.2 — with an introduction to sea ice and, in Section 3.3, ice management. A background on radar imagery is then given in Section 3.4 before the sensors and datasets used in this thesis are presented in Section 3.5.

Section 3.6 then goes on to introduce detection and classification as exemplified from the perspective of the SAR sensor. The principles, however, extends to the other sensors described in this chapter as well. In Section 3.7, a study is conducted using SAR data provided by the European Space Agency (ESA) to test and demonstrate the extraction of ice objects from both Single Look Complex (SLC) and Ground Range Detected (GRD) products, using methods from the field of machine learning (Bishop, 2006). Conclusions from the chapter are offered in Section 3.8.

For the recording of most of the datasets used in this thesis, operations were carried out on Svalbard, at the Kongsfjorden test area in Ny-Ålesund, seen in Figure 3.2. One of the major contributors to sea ice in the fjord is the Kronebreen glacier, shown in Figure 3.3.

Figure 3.2: The village of Ny-Ålesund, with one of the nearby glaciers in the background

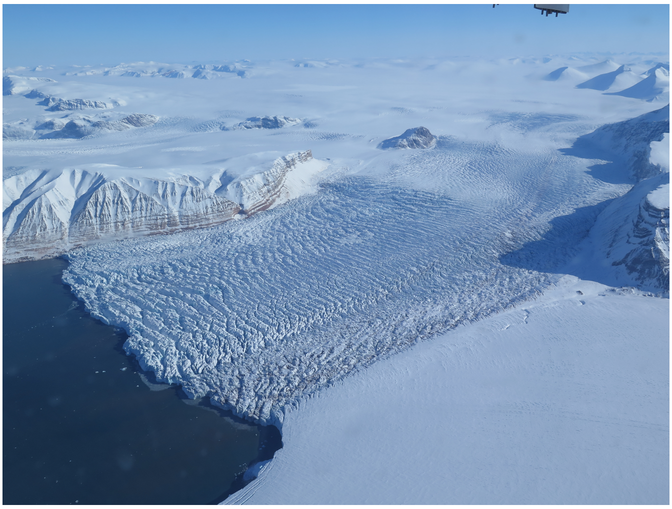


Figure 3.3: The Kronebreen glacier is one of the major contributors of ice of land origin to the Kongsfjorden test area

3.2 Sea Ice

Floating ice can be found in all bodies of water in the polar regions of our planet. As the life of an ice object progresses, it evolves through freezing, melting and snow buildup as well as mechanical wear and tear. Further, a free-floating object will — as the center of mass changes and waves push it around — tumble and may completely change appearance depending on its three-dimensional orientation. For tracking, as discussed later in this thesis, this limits the capabilities of e.g. recognizing previously known objects by their shape. In this section, we give a brief background on the classes of ice which are our main interest for this thesis. For a more thorough treatment of the topic, we refer to e.g. Leppäranta (2011).

3.2.1 Ice Classification

There are several ways to categorize ice — by March 2014, the standard for sea ice nomenclature, WMO (2014), contained 220 terms and definitions in 13 sections. The major classifications include by age (e.g. first-year ice, multi-year ice depending on how many annual melts it has survived) and by origin. The World Meteorological Organization (WMO) defines the following types of floating ice (WMO, 1970, 2014):

sea ice Any form of ice found at sea which has originated from the freezing of sea water.

ice of land origin Ice formed on land or in an ice shelf, found floating in water. The concept includes ice that is stranded or grounded.

lake ice Ice formed on a lake, regardless of observed location.

river ice Ice formed on a river, regardless of observed location.

WMO (1970) further defines discrimination of ice based on size,

iceberg A massive piece of ice of greatly varying shape, protruding more than 5 m above sea-level, which has broken away from a glacier, and which may be afloat or aground. Icebergs may be described as tabular, dome-shaped, sloping, pinnacled, dry-docked, blocky, weathered or glacier bergs in addition to having a size qualifier.

ice island A large piece of floating ice protruding about 5 m above sea-level, which has broken a way from an Arctic ice shelf, having a thickness of 30 m to 50 m and an area of from a few thousand square meters to 500 km^2 or more.

bergy bit A large piece of floating glacier ice, generally showing less than 5 m above sea-level but more than 1 m and normally about 100 m^2 to 300 m^2 in area.

growler A piece of ice smaller than a bergy bit and floating less than 1 m above the sea surface, a growler generally appears white but sometimes transparent or blue-green in color. Extending less than 1 m above the sea surface and normally occupying an area of about 20 m^2, growlers are difficult to distinguish when surrounded by sea ice or in high sea state

Another common classification is based on its position,

coastal ice is ice that is fixed to the coast, even during the tidal cycle.

fast ice shifts vertically with the tide, but is locked to the coast.

shear zone is the transition zone between fast and drift ice (Leppäranta, 2011).

drift ice moves with wind and currents.

marginal ice zone constitutes the transition into open water, and is affected by the waves of the ocean.

The properties and distributions of floating ice are well covered by e.g. Leppäranta (2011). In regions partially covered with ice, we define the ice concentration as the fraction (in percent) of the area that is covered by ice.

Depending on the scenario, each of the ice classes presents different problems to polar operations but in this thesis, the focus is mainly on ice in open ocean or the marginal ice zone.

3.2.2 Drift Ice Motion

The motion of ice objects floating in open water — unaffected by the forces involved when icebergs collide or otherwise interact — is governed by a number of forces which act up the sea ice itself and its surrounding body of water. The resulting velocity is, characteristically, in the order of 0.01 $\mathrm{m\,s^{-1}}$ to 1 $\mathrm{m\,s^{-1}}$ (Leppäranta, 2011). In Figure 3.4, a cross-section of an iceberg is sketched with some definitions of parts of the object most affected by forces — the sail (above sea-level) and the keel (below sea-level).

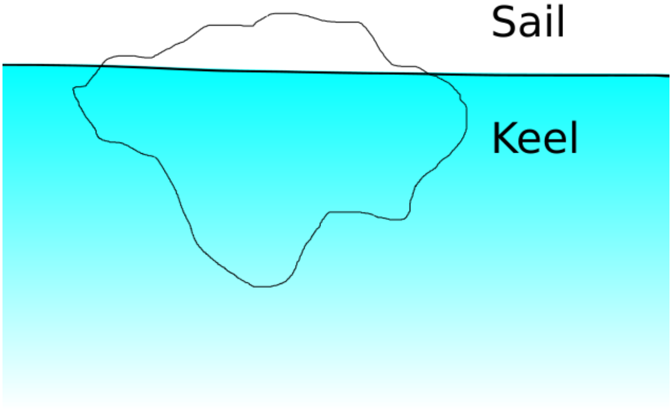


Figure 3.4: Ice in the water

Formulations of Newton's second law of motion, $m\boldsymbol{a} = \sum \vec{f}$, as applied to ice drift, differs mainly in the choice of notation and the choice of forces to include. Using the formulation of e.g. Eik (2009); Andersson (2018); Mountain (1980); Johannessen et al. (1999) we identify the following expression:

$$m\boldsymbol{a} = \vec{f}_{cor} + \vec{f}_a + \vec{f}_c + \vec{f}_r + \vec{f}_p \tag{3.1}$$

where $\boldsymbol{a}$ is the acceleration, $\vec{f}_{cor}$ is the Coriolis force, $\vec{f}_a$ is the air drag force, $\vec{f}_c$ is the water (current) drag force, $\vec{f}_r$ is the wave radiation force and $\vec{f}_p$ is the pressure gradient force. m is the total ice mass, which is the sum of the ice mass (m_0) and the added mass due to the surrounding water field. This is often written as $m = m_0\,(1 + C_m)$; C_m is usually taken to be 0.5 (Eik, 2009).

In particular; with drag coefficients ($C_{(\cdot)}$), medium densities ($\rho_{(\cdot)}$) and velocities ($\dot{\boldsymbol{x}}_{(\cdot)}$) for $a$ir and $w$ater respectively, as well as keel surface area $\boldsymbol{A}_k$ and sail surface area $\boldsymbol{A}_s$, we define the forces

$$\vec{f}_c = \frac{1}{2}\rho_w C_w \boldsymbol{A}_k \left|\dot{\boldsymbol{x}}_c - \dot{\boldsymbol{x}}\right| \left(\dot{\boldsymbol{x}}_c - \dot{\boldsymbol{x}}\right), \tag{3.2}$$

$$\vec{f}_a = \frac{1}{2}\rho_a C_a \boldsymbol{A}_s \left|\dot{\boldsymbol{x}}_a - \dot{\boldsymbol{x}}\right| \left(\dot{\boldsymbol{x}}_a - \dot{\boldsymbol{x}}\right). \tag{3.3}$$

where $\dot{\boldsymbol{x}}$ is the velocity vector of the ice object.

For the purpose of this thesis, we defer the detailed discussion of each force (and the omitted force definitions) to Eik (2009); Andersson (2018) or Leppäranta (2011) with the following comments

- the main two forces acting upon the ice object are the wind and the current;
- the Coriolis, wave radiation and pressure gradient forces are, for our purposes, negligible in comparison to air and current forces (Leppäranta, 2011);
- the wave radiation force can, for waves parallel with the wind, be included in the $\vec{f}_a$ parameter;
- both the major forces are functions of velocity difference to the speed of water and air respectively.
- the uncertainty of the parameters — especially in weight and three-dimensional shape — will in many cases render the uncertainty of the simulations beyond what is acceptable for creating relevant predictions (Eik, 2010);
- parameters are subject to change as the shape or orientation of the ice changes; and
- with large uncertainties in the parameters, the benefits of an advanced (physical) motion model diminish, and the model can be replaced by a simpler one with fewer parameters — such as the Nearly Constant Velocity (NCV) model, detailed in Chapter 4.

Regardless of the choice of model, predictions need to be supplemented with observations for accurate tracking, for all but short time horizons.

3.3 Ice Management

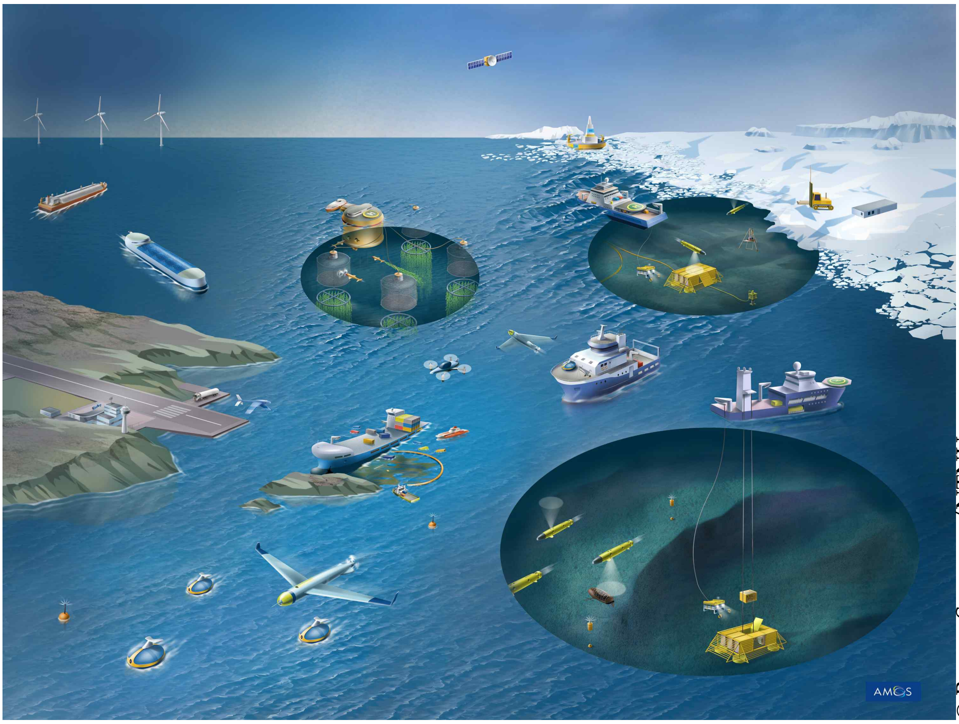
Following the definition from (Eik, 2010):

> ***Ice management*** *is the sum of all activities where the objective is to reduce or avoid actions from any kind of ice features. This will include, but is not limited to:*

- *Detection, tracking and forecasting of sea ice, ice ridges and icebergs;*
- *Threat evaluation;*
- *Physical ice management such as ice breaking and iceberg towing; and*
- *Procedures for disconnection of offshore structures applied in search for or production of hydrocarbons.*

Ice management can further be split into passive and active ice management. Passive ice management is mainly due to actions taken in the constructions of an installation, such as the use of tilted foundation surfaces to break oncoming ice (Wright, 2000; Marchenko, 2010) or the use of steel pile barriers (Evers and Weihrauch, 2004). In contrast, active ice management requires online monitoring and response to detected threats and will, in general, include marine vessels with ice breaking- or ice towing capabilities on active duty to handle detected threats.

Ice management operations have been successfully carried out for several decades (Eik, 2010), mainly driven by the search and extraction of oil in Arctic regions. Until recently, the main sources of data has come from satellites and the icebreakers performing the physical ice breaking and towing. Due to the variability of the ice threats, no single source will be sufficient to detect all hazards. Thus, as concluded by Eik (2010) and illustrated in Figure 3.5, future operations are likely to include a range of sensors and sensor platforms, such as Autonomous Underwater Vehicles (AUVS), sonar, ice drift buoys and UAS's. The decreased cost and increased availability has spurred an increased movement of research towards the application of UAS's in the Arctic (Eik, 2010; Haugen, 2014; Leira, 2017) as well as of the complications involved (Sørensen, 2016).



© Bjarne Stenberg/NTNU

Figure 3.5: Ice management in the future is likely to be a connected setup of many agents for sea ice detection and many stakeholders of their collected data. Sensors include satellites, AUVs, UAS's, marine radar etc. Stakeholders include marine transport, oil rigs and earth scientists.

3.4 Radar Imagery

Two of the three sensors employed to record the datasets used in this thesis are based on radar technology. Therefore, we here provide a brief introduction to the sensors, and to the radar measurement principles, as well as a discussion on the radar properties of the intended target — sea ice.

The technical core of the radar is an instrument which measures electromagnetic (EM) reflectivity of an observed object by active illumination with EM waves of a set interval of wavelengths. In each measurement, a region is illuminated with an EM pulse, and the response is recorded as a sampled EM signal. From this, measurements of the amplitude of a reflected signal as well as its phase, based on the signal return delay (Richards, 2009), can be extracted. These two numerical entities can be interpreted as complex valued measurements. The sampled signal is the sum of all reflections on the illuminated surface as well as unwanted noise and interference phenomena — *speckle*. The raw measurements thus have to undergo significant signal processing to make the relevant information accessible as imagery (Richards, 2009).

The reflectivity of a material is related to its complex dielectric constant; a lower real part gives a higher penetration depth and lower reflectivity. A target's ability to reflect radar signals back to the receiver is measured as the Radar Cross-Section (RCS) σ. This measure describes how much power of the incoming signal is extracted by the target. It is measured in square meters, although it is not easily related to any physical area of the target (Richards, 2009). The RCS is a strong function of the incidence angle (Haykin, 1994). Various methods exist to deal with normalization of this data, as studied in e.g. Mladenova et al. (2011). The radar cross-section of sea ice can, within the diffuse scattering region, be well modeled by an affine model as a function of the incidence angle (Haykin, 1994, p. 589).

A polarimetric radar measures the EM backscatter in both polarization directions — horizontal (H) and vertical (V) as seen from the sensor. Since the signal can be transmitted in either of those directions and measured in either or both, four possible modes of measurements are possible: HH, HV, VV, VH. Here, the first letter indicates the polarization of the transmitted wave, and the other that of the received ditto. The two most used modes of polarimetric operation are quad-pol — measuring all four of these — and dual-pol — where only two are measured. Commonly in the latter case, one polarization is transmitted and both are measured, resulting in either HH–HV or VV–VH measurements.

Measurements from two main categories of radar were considered in this thesis; a fixed-aperture ground-based radar and satellite-borne SAR.

3.4.1 Satellite-borne SAR

Satellite imagery enables observation of large areas with high detail. However, as the satellites follow their predefined orbits, the revisit time to the same area is in the order of days. The observed regions — as seen in Figures 3.6–3.7 — can contain a large number of ice objects worth tracking. The main satellite imaging technique used for sea ice tracking is the SAR. Its prevalence is perhaps most importantly due to its ability to see through

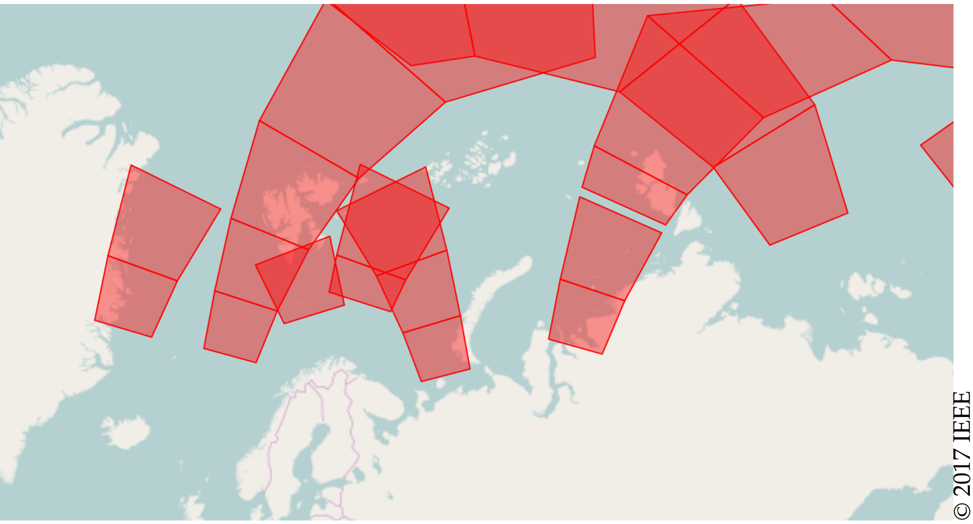


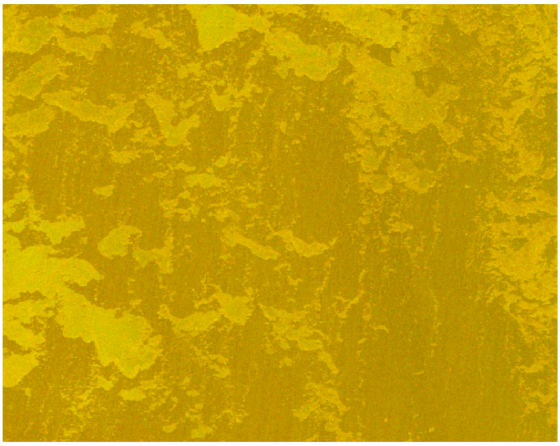
Figure 3.6: Sample Sentinel 1 satellite observation areas in the Arctic. Each square represents a sampled area distributed as one image.

clouds and deliver consistent imagery regardless of weather conditions or daylight (Richards, 2009).

The sensor, as illustrated in Figure 3.8, is often carried by a satellite or airplane and the measurements are transmitted to a recipient on the ground. A key in the SAR principle is that it uses the forward linear motion, illustrated in Figure 3.9, of its platform to synthesize an apparently long antenna through repeated sampling. This also leads to the remarkable result that, at the cost of a more advanced post-processing than fixed-aperture radar, the attainable resolution is independent of operating altitude and even increases with smaller antennae (Richards, 2009). This makes spaceborne operation acceptable.

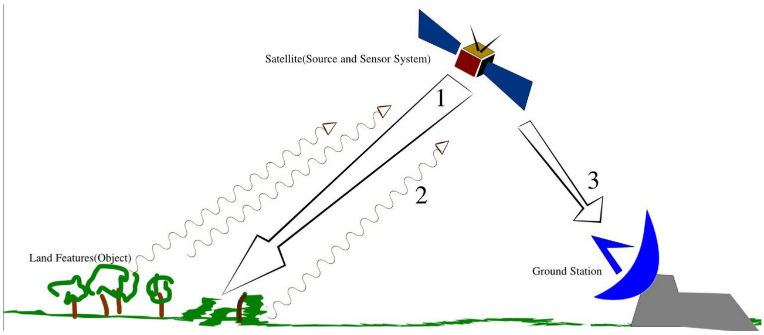
3.4.2 Ground-based Radar

Ground-based radar has, compared to satellite- or aircraft-borne radar, a capability of repeated observations over large spans of time. Further, it is possible to further improve Signal-to-Noise Ratio (SNR) as multiple observations can be carried out with little delay from exactly the same position (Werner et al., 2012).



© 2017 IEEE

Figure 3.7: Region with ice coverage from SAR imagery

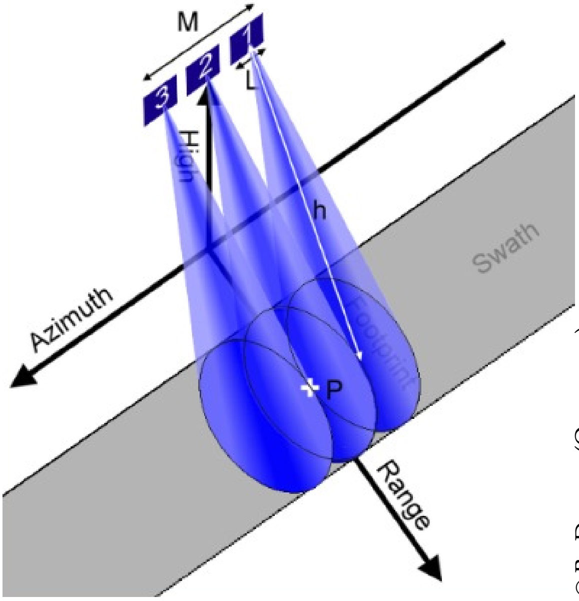


© By Arkarjun (Own work) [CC BY-SA 3.0], via Wikimedia Commons

Figure 3.8: SAR measurement principle

3.4.3 Radar Properties of Sea Ice

Sea ice is an inhomogeneous radar target with variations of the complex dielectric constant on all spatial scales. Nevertheless, characteristic signatures have been observed which allow for classification of the different stages of sea ice formation (Haykin, 1994). With only limited sensing penetration depth, the radar sensor foremost provides data about the EM scattering at the ice surface. This is subject to significant seasonal variations, including seasonal melt which can lead to flooding and pooling difficult to discern from free water (Haykin, 1994).



© By Dantor (Own work)
[CC-BY-SA-3.0], via Wikimedia Commons

Figure 3.9: Satellite direction of motion and sampling

In general, water reflection is dominated by Bragg- and diffuse surface scattering, yielding backscatter to the radar sensor (Thenkabail, 2015) which is generally weak, but much depending on wind/sea roughness (Arkett et al., 2006; Johannessen et al., 2007). Correspondingly, sea ice scattering properties vary greatly with ice type and season (Onstott, 1992), but are often characterized by a stronger reflection due to its crystalline structure and volumetric scattering (Haykin, 1994), despite a much lower dielectric constant than the surrounding water (Hallikainen, 1992). Its lower dielectric constant also allows for deeper penetration by the EM waves, enabling sub-surface ice properties to be sensed to some extent (Onstott and Shuchman, 2004). Different radar wavelengths excel at detecting different ice properties, due in part to different penetration depth (Dierking and Pedersen, 2011). Accurate separation of rough water and certain ice types is not always possible, but research suggests it can be improved with the combination of co- and cross-polarized data (Arkett et al., 2006).

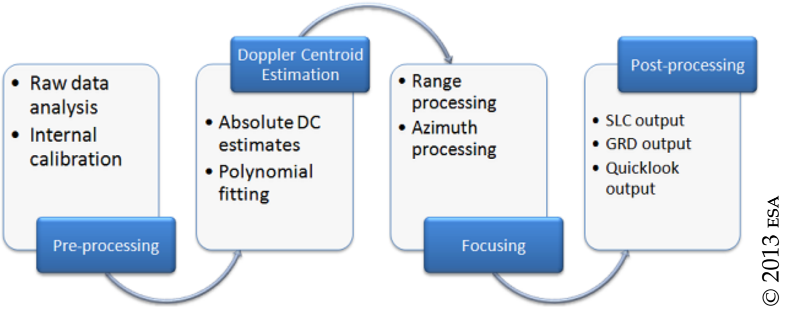


Figure 3.10: SAR level-1 processing steps (ESA, 2013)

3.5 Sensors for Sea Ice Detection

This section introduces the sensors and their respective datasets that were used in this thesis — the SAR, the TRI and the infrared (IR) imagery captured using the UAS platform.

3.5.1 Sentinel 1 SAR

The principles of SAR image acquisition were introduced in Section 3.4. For convenience of use, the provider of SAR imagery processes the raw reflection measurements provided by the sensor. Here, we briefly outline the ESA process, using their terminology; the raw (*level-0 data*) is transmitted to the ground for processing, analysis, distribution. The initial steps of the data processing are shared by most applications of the data, and the result of these are therefore provided as a set of *level-1* products. These steps are summarized in Figure 3.10 and detailed in ESA (2013); Thain (2014).

The level-1 products from the SENTINEL 1 satellite are distributed in two data formats: Single Look Complex (SLC) and Ground Range Detected (GRD). As the name implies, the SLC data is an image with the complex property of the data preserved. Unlike GRD data, it has not been projected to the earth surface but is calibrated and focused using the orbit data from the satellite (ESA, 2013). GRD data is, in contrast, projected onto the WGS-84 earth ellipsoid (ESA, 2013) and averaged — *multi-looked* — to provide approximately square pixels with reduced noise. The phase information is lost in the GRD processing.

SAR images taken with the SENTINEL 1 satellite pair are made available by ESA online at https://scihub.copernicus.eu/dhus.

3.5.2 GAMMA Portable Radar Interferometer

In April 2016, the partners of the Norwegian Centre for Integrated Remote Sensing and Forecasting for Arctic Operations (CIRFA) conducted a field campaign in Kongsfjorden on Svalbard, where ice breaks off the nearby glaciers and flows towards the ocean. One of the collected datasets is from a GAMMA Portable Radar Interferometer (GPRI)[1] (Werner et al., 2012), a fixed-aperture ground-based TRI sensor shown in Figure 3.11. This sensor primarily monitors sea ice drift in the fjord. An illustration of the Kongsfjorden test area is given in Figure 3.11, with the surrounding glaciers and other features of the region. It also shows the TRI sensor setup and location. An example image is shown in Figure 3.12, giving a general overview of the area. In a long-term monitoring effort, a second similar dataset was gathered in spring 2018.

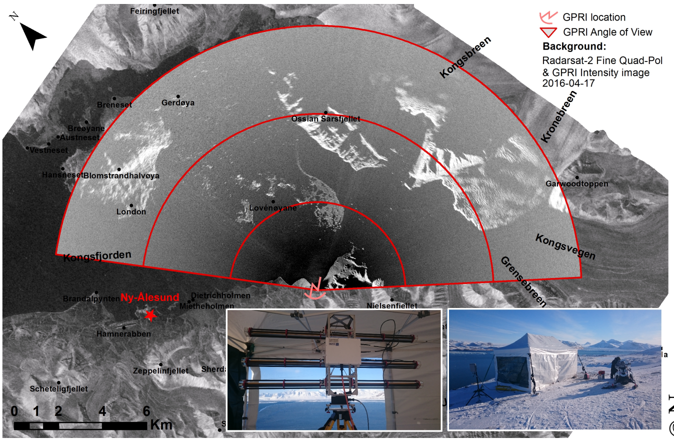
Images in the first dataset was sampled with a 3 min interval, whereas the second dataset was gathered with images taken in 15 min intervals.

A image of the geocoded data is shown in Figure 3.12, overlaid with the sensor position and an illustration of the nature of the untransformed measurements. The radar is located at the central red dot and images are detected in a range-angle coordinate system. Unlike the SAR images, the TRI images are available at a controllable — and much higher — rate, simplifying the use of correlations between consecutive images. As such, it is possible to extract ice objects and have them tracked continuously, as demonstrated e.g. in Chapter 6.

For ease of use, the raw data from each TRI image was transformed to geocoded Cartesian images with 5m × 5m resolution. The series of images were then processed using the OpenCV MOG2 background subtractor (Zivkovic, 2004) and detections were extracted from foreground objects with a minimum size of 10 connected pixels.

Two types of ice are considered; *i*) large regions of stationary sea ice with high ratio that can be segmented independently for each scan; and *ii*) drift ice with low SNR that requires pre-processing over several scans for detection. A land mask was applied to the images to ensure that detections were only obtained in water regions.

[1] https://www.gamma-rs.ch/rud/microwave-hardware/gpri.html

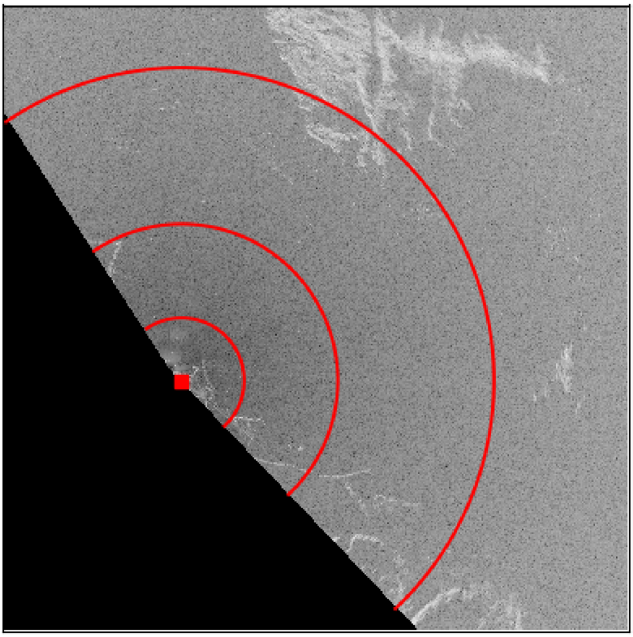


© Norut

Figure 3.11: Illustration of the Kongsfjorden test area, its surroundings and the setup of the TRI sensor

Detection of Stationary Sea Ice Large areas of stationary ice are visible in the water, in particular in proximity to land. However, due to speckle noise and varying intensity over the image, a simple threshold results in poor performance. To improve the SNR, a sequence of standard image segmentation methods (Gonzalez and Woods, 2008) were applied (Olofsson et al., 2017d) to average, filter and extract areas considered stationary over an extended period of time.

Detection of Drift Ice Drift ice is generally small and often difficult to distinguish in the speckle noise. In the datasets, a background model was estimated in water regions, modeling each pixel as a mixture of Gaussian distributions (Kaewtrakulpong and Bowden, 2001; Stauffer and Grimson, 1999). A simplified expectation-maximization method (Dempster et al., 1977) was then used to continuously estimate the means and covariances in the model over time. For an incoming scan, all pixels that are significantly different from their background models are segmented as foreground. This results in many false detections, which was mitigated by extracting connected components of a minimum RCS of $150\,\mathrm{m}^2$. The reports are then obtained as the centroid of each connected component.



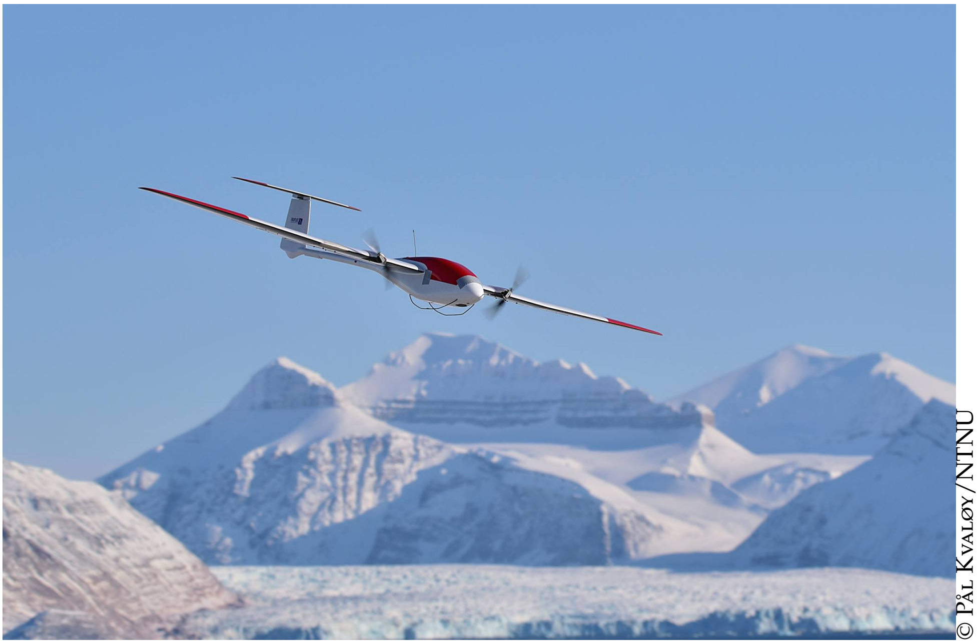
© NORUT

Figure 3.12: Example of a radar scan from the TRI sensor, with brighter areas of sea ice and an overlaid illustration of the radial nature of its raw measurements

3.5.3 Unmanned Aerial Systems

While UAS's have been imagined and constructed since ancient Greece and China (Valavanis, 2007), the field of study took its first major leaps during the first and second world war, illuminating the dominant role that the military has played in its developments. The digital age has brought new opportunities and a whole new level of autonomy. While a common alliteration for UAS is that they can operate missions too "Dull, Dirty and Dangerous" for human beings, recent years have seen their use in a growing range of civilian applications, such as traffic monitoring and rescue scenarios.

The Arctic has proved a harsh frontier for UAS's, with a cold operating environment — for both the machines and their operators — and with often high wind speeds. Among other problems, this can lead to aerodynamical problems due to icing (Sørensen, 2016; Hann et al., 2017), which is attributed as the main cause for the loss of UAS's in cold regions.



© Pål Kvaløy/NTNU

Figure 3.13: The CryoWing Scout, developed by Norut was flown to collect imagery of sea ice in Kongsfjorden

The UAS datasets used in this thesis have been collected in collaboration with Norut in Tromsø, Norway. The datasets were collected in Kongsfjorden Ny-Ålesund, Svalbard, during a campaign in April–May 2016. The Kongsfjorden test area, the drone launch site and its relation to the location of the TRI dataset is illustrated in Figure 3.15.

The datasets were collected using the *CryoWing Scout* drone — shown in Figure 3.13 and Figure 3.14 — developed by Norut. The CryoWing Scout has a maximum take-off weight of 9 kg, and a wingspan of 2.5 m. It has a range of appropriately 120 km and a two-hours flight-time. Apart from telemetry sensors and radio equipment for communication, the UAS was also equipped with a downwards facing infrared camera, with images taken every third second of the flight. Together with speed of flight, Field-of-View (FOV) and mission altitude, this entails that a detected object is within FOV for an approximate maximum of three frames. The object detection algorithm is performed online and the results stored in an SQL database from which the flight can be replayed. Apart from optical imagery, the recorded flight data includes position data and other telemetry, as well as FOV. The captured

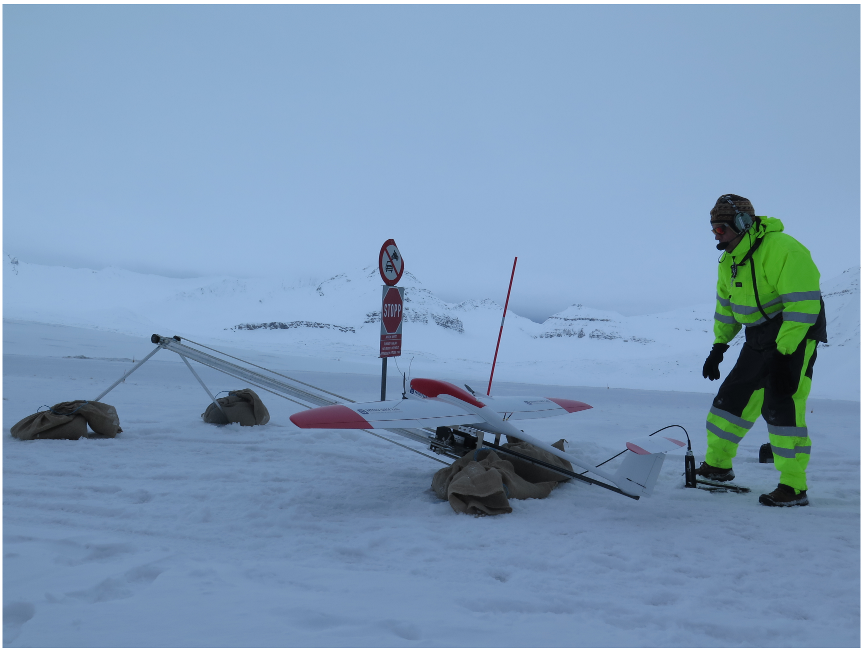
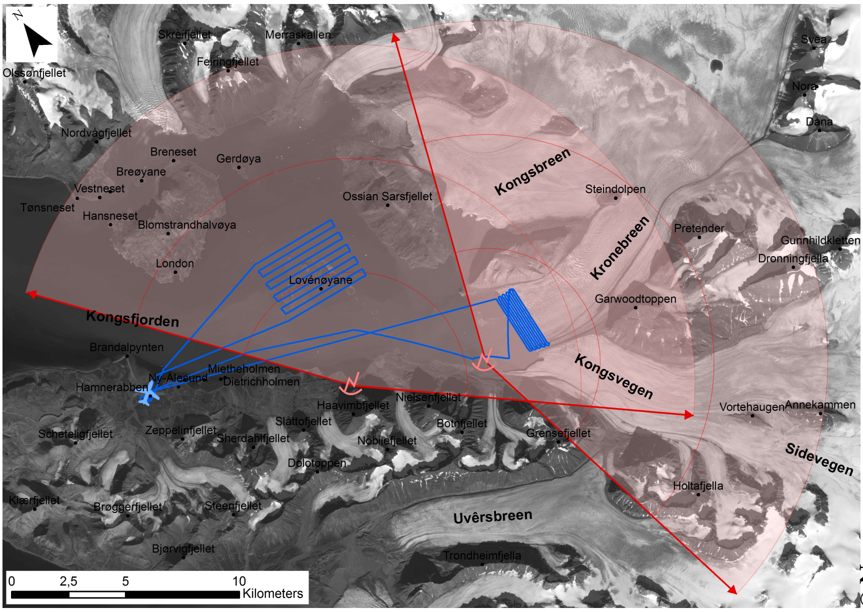


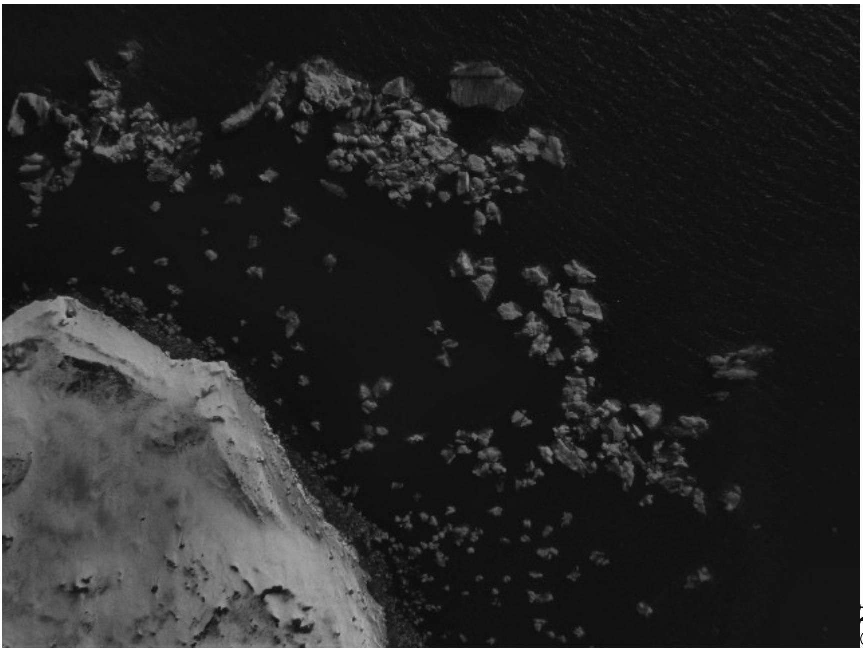
Figure 3.14: Preparation for launch of the UAS on Svalbard

imagery is exemplified in Figure 3.16, and the detection procedure — as detailed by Leira (2017) — is summarized in Figure 3.17.



© NORUT

Figure 3.15: Illustration of the Kongsfjorden test area, the drone launch site (at drone) and its relation to the TRI data capture (red semicircles) and surrounding area



© NORUT

Figure 3.16: Example image from one of the captured UAS datasets

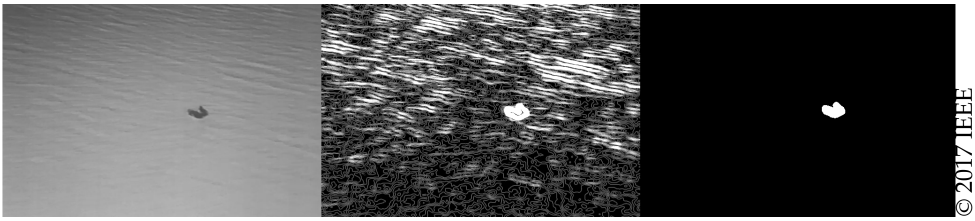


Figure 3.17: Detections of ice extracted from infrared imagery from UAS-carried cameras. Left: Raw thermal image; Middle: Enhanced features; Right: Segmented detected sea ice (Leira, 2017). Image courtesy of Frederik S. Leira.

3.6 Detection and Classification

A fundament to this thesis is the availability of detections of sea ice. This section presents relevant background, to Section 3.7 in particular, to how images can be processed to extract and classify detections from the available raw imagery. The processing is centered around the calculation of *features*, numerical metrics to help distinguish each sought class. At the level presented in this section, classification is performed individually for each pixel and each processed image.

The family of classification algorithms may be categorized in four categories from their requirements of training data and knowledge of feature probability distributions, as shown in Table 3.1 with example algorithms.

In the following, the k-means[2], clustering algorithm as well as the Support Vector Machine (SVM) algorithm are briefly presented, along with a mention of features suitable for Bayesian classification. The topic of Bayesian classification has been studied further in e.g. Zakhvatkina and Bychkova (2015).

3.6.1 k-means Clustering

The k-means algorithm is an iterative clustering method, which makes hard classification assignments to exactly K classes. The algorithm is based

[2] This k, and the index k used in this section, is unrelated to time-index k, used in the rest of the thesis

Table 3.1: Example of classification algorithms, categorized by requirements

		Training data?	
		Yes	No
Known pdf?	Yes	Bayesian classifier	Clustering
	No	svm, Neural Networks	?

around the minimization of the *distortion measure*

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{n,k} \left| \boldsymbol{x}_n - \boldsymbol{\mu}_k \right| \tag{3.4a}$$

for feature vectors $\boldsymbol{x}_n$ and class mean $\boldsymbol{\mu}_k$ for class k, through the selection of

$$\left\{ r_{n,k} : r_{n,k} \in \{0,1\}, \sum_{k=1}^{K} r_{n,k} = 1 \quad \forall n \in [1, \ldots, N] \right\}. \tag{3.4b}$$

The iteration scheme is otherwise equivalent to that of the Expectation Maximization algorithm (Bishop, 2006) and yields a set of multidimensional centroids. For any given point, the closest (generally in the Euclidean sense (Bishop, 2006)) centroid represents the point's class assignment. Briefly; starting from a (random) initial set of centroids, the following two steps are repeated until convergence:

E) Each datapoint is assigned to its nearest centroid,

M) Given the set of assigned datapoints, each centroid is moved to the set's mean.

As a preprocessing step, the classified feature set should be whitened (Coates and Ng, 2012).

While the k-means algorithm is easily applied, the result is sometimes considered indiscriminatory, because of its naïve assumption of nearest-neighbor class assignment. Therefore, it is often applied to initiate more advanced classification algorithms (Bishop, 2006), such as a svm.

3.6.2 Support Vector Machines

The Support Vector Machine (SVM) algorithm is, at its core, a linear classifier which uses optimization techniques to find the linear (hyper)plane which separates the feature space with the largest possible margin (Bishop, 2006). The intuition here is that the wide margin should minimize the risk for future misclassification. In its basic formulation, the SVM is a two-class classifier, but extensions exist to the multi-class case (Bishop, 2006).

The separating plane can be expressed as

$$\boldsymbol{w}^{\intercal} \boldsymbol{\phi}(\boldsymbol{x}) + b = 0 \tag{3.5}$$

where $\boldsymbol{\phi}(\boldsymbol{x})$ denotes the feature vector given the state $\boldsymbol{x}$, $\boldsymbol{w}$ the feature weights, and b the plane constant.

With the margin given by the perpendicular distance to the closest point in the dataset, the training consists of solving

$$\underset{\boldsymbol{w},b}{\arg\max} \left\{ \frac{1}{|\boldsymbol{w}|} \min_{n} \left[c_n \left(\boldsymbol{w}^{\intercal} \boldsymbol{\phi}(\boldsymbol{x}_n) + b \right) \right] \right\} \tag{3.6}$$

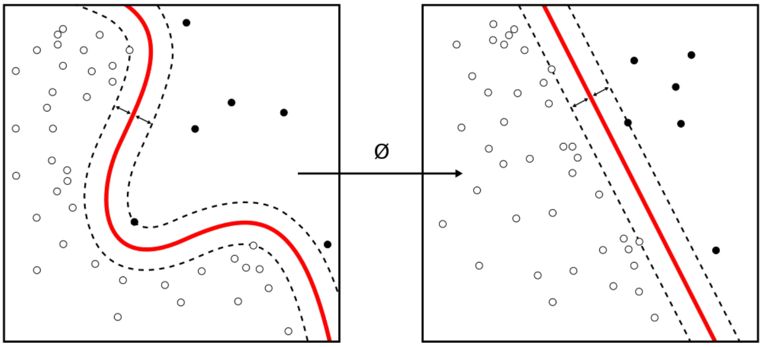
where c_n denotes training sample n and $t_n \in \{-1, 1\}$ specifies its class. The optimization, detailed in e.g. Bishop (2006), yields that only a very limited set of the training samples — the ones closest to the margin — will affect the classification of new data. This set is called the Support Vectors.

While the discrimination is linear by default, the distance measure used in the algorithm is often transformed using *kernel functions* to create non-linear decision boundaries, as in Figure 3.18. By the selection of kernel functions, features can be mapped to a space where the classes are linearly separable, and the standard formulation applies with the modified distance measure (Fletcher, 2009). Notably, kernel functions allow a mapping to a higher, even infinite, dimensional space while also reducing computational complexity (Ng, 2015). Common kernels are listed in Table 3.2.

The SVM is a decision machine, meaning that once trained, it will produce hard classifications, similar to the k-means algorithm (Bishop, 2006). A related algorithm, based on a Bayesian formulation, is the Relevance Vector Machine which is described in e.g. Bishop (2006) but not further discussed here.

Table 3.2: Common kernels

Kernel (hyperparameters)	**Expression**
Linear	$\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$
Polynomial (γ, r, d)	$\left(\gamma \langle \boldsymbol{x}, \boldsymbol{x}' \rangle + r\right)^{d}$
RBF (γ)	$\exp\left(-\gamma \lvert \boldsymbol{x} - \boldsymbol{x}' \rvert^{2}\right)$
Sigmoid (γ, r)	$\tanh\left(\gamma \langle \boldsymbol{x}, \boldsymbol{x}' \rangle + r\right)$



© By Alisneaky, svg version by Zirguezi [CC BY-SA 4.0], via Wikimedia Commons

Figure 3.18: Linear decision boundaries can still be applied to nonlinear problems using kernel functions to transform the problem into linear space.

3.6.3 SAR Classification Features

Classification algorithms depend on numerical metrics — features — which separate the different classes by expressing distinguishing information. Common features include pixel intensity and, in the case of polarimetric radar, the relation of measurements between the different polarizations. Relevantly, cross-polarized data — HV and VH, respectively — has been proposed to be relevant to the distinction between sea ice and open water (Scheuchl et al., 2004; Dierking and Pedersen, 2011).

As objects of significant size occupy multiple pixels, one may also consider the corresponding metrics from neighboring pixels, as neighboring pixels often share the same class.

In polarimetric SLC imagery, where the complex representation of the pixels preserve the phase of the return pulse, several alternative decompositions are also available (Richards, 2009; Cloude and Pettier, 1996), often derived from the complex scattering matrix $\boldsymbol{S}$. This matrix relates the target's incident and scattered EM fields, and is formed from the complex measurements (Anfinsen et al., 2007; Oliver and Quegan, 2004) $S_{\mathcal{R}\mathcal{T}}$ for received polarization $\mathcal{R} \in \{H, V\}$ and transmitted polarization $\mathcal{T} \in \{H, V\}$:

$$\boldsymbol{S} = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix}. \tag{3.7}$$

Extracting the terms, and assuming $S_{HV} = S_{VH}$, it is common to form the scattering vector, $\boldsymbol{k}$, or its *Pauli representation*, $\boldsymbol{k}_p$:

$$\boldsymbol{k} = \begin{bmatrix} S_{HH}, & \sqrt{2}S_{HV}, & S_{VV} \end{bmatrix}^{\mathsf{T}}, \tag{3.8}$$

$$\boldsymbol{k}_p = \frac{1}{\sqrt{2}} \begin{bmatrix} S_{HH} + S_{VV} \\ S_{HH} - S_{VV} \\ 2S_{HV} \end{bmatrix}. \tag{3.9}$$

Inherent speckle noise in the image data advocates the study of the (multilooked) covariance of these vectors, as locally averaged in a pixel neighborhood to form the covariance matrix $\boldsymbol{C}_c$ and the coherency matrix $\boldsymbol{T}$;

$$\boldsymbol{C}_c = \mathbb{E}\left[\boldsymbol{k}\boldsymbol{k}^*\right], \tag{3.10}$$

$$\boldsymbol{T} = \mathbb{E}\left[\boldsymbol{k}_p\boldsymbol{k}_p{}^*\right], \tag{3.11}$$

where * denotes the complex conjugate transpose.

Different methods of analysis of these matrices have been proposed. A commonly employed decomposition is the so called Entropy/Anisotropy-/Alpha ($H/A/\alpha$) decomposition — see e.g. Cloude et al. (2002); Richards (2009) — in which the properties of the coherency matrix' eigenvalues are used to form the features that give the decomposition its name. This feature space has proved useful for scattering type separation (Richards, 2009), and has also been used for initialization and automation of other classification methods, such as the Complex Wishart Classifier (Lee et al., 1999; Anfinsen et al., 2007).

The Wishart classifier, and variants thereof, assumes a complex Gaussian distribution to the elements of the scattering matrix, yielding a complex Wishart distribution of the coherency (Anfinsen et al., 2007) or covariance (Doulgeris et al., 2011) matrix. Properties derived thereof may be used e.g. in a modified k-means classifier to obtain a family of spectral Wishart classifiers (Anfinsen et al., 2007).

While the above equations assume the full quad-pol S-matrix to be available, extensions exist for dual-pol data, which is more generally available in remote areas typical to ice surveillance (Cloude, 2007).

3.6.4 Point Representation of Non-Point Objects

The k-means and the SVM algorithms described above can be used to provide pixel-wise classification of sea ice objects, contrasted by surrounding water. To distinguish and represent separate objects, it is necessary to cluster the classified pixels e.g. based on pixel connectivity. This process is called labeling, see e.g. Kong and Rosenfeld (1996).

Once labeled, the extent and position of each object can be represented by any sufficient statistic. Variants include fitting a Gaussian, or simply representing its position by the mean of the associated pixels' positions. While in some cases over-simplistic — e.g. it sometimes results in mean positions outside the actual object — it has the advantages of being simple and invariant to scale.

3.7 SAR Ice Extraction Study

In this section, we apply two machine learning algorithms to extract sea ice object detections from SAR images. While the study presented here is performed using SAR data, a similar process could be applied to other imagery, such as that from ground-based radar or optical cameras.

3.7.1 Method

The analysis was performed from two different types of SAR data — GRD and SLC. The two types carry different types of data, which yield separate sets of available features. Given that those features have been extracted however, the subsequent algorithms for segmentation and classification are independent of data type. Thus, as a delimitation, only the GRD image was used for the evaluation of classification algorithm below.

The images used in the analysis, Figures 3.19–3.20, were captured on 2016-03-11 (GRD) and 2015-10-12 (SLC), respectively[3]. The significant dominance of yellow in both images shows a likeness between the HH (red) and HV (green) bands, which is consistent with the conclusions of other studies (Arkett et al., 2006). The origins of these datasets are detailed in Section 3.5.1.

Feature Extraction

The sample images were exported as GeoTIFF rasters from the original SAFE[4] format using the ESA SNAP tool for Sentinel 1 data analysis. Further processing and classification was performed in a PYTHON environment.

The following properties were exported for the GRD image:

- HH amplitude;
- HV amplitude;
- latitude;
- longitude; and
- incidence angle.

The following properties were exported for the SLC image:

- HH complex amplitude;
- HV complex amplitude;
- latitude;
- longitude;
- incidence angle; and
- entropy/alpha decomposition (Cloude et al., 2002).

[3] Full names of the datasets are:

GRD: *S1A_EW_GRDM_1SDH_20160311T082543_20160311T082643_010319_00F469_A99F*,

SLC: *S1A_IW_SLC__1SDH_20151012T033833_20151012T033846_008114_00B612_C893*.

[4] Standard Archive Format for Europe: http://earth.esa.int/SAFE/

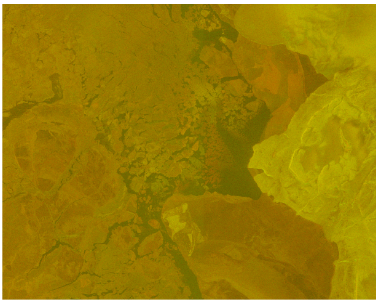


Figure 3.19: Sample GRD image used for sea ice extraction. Red = HH; Green = HV

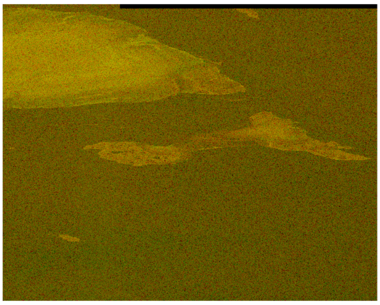


Figure 3.20: Sample SLC image used for sea ice extraction. Red = |HH|; Green = |HV|

The PYTHON packages *scipy* and *scikit-learn* provided implementations of the *k*-means and the SVM algorithm, respectively, while the *gdal* package was used to access the exported rasters.

Simplifying Assumptions

Lacking ground truth, three simplifying assumptions were made

- every pixel is either water, or ice (of any type);
- the incidence angle can be considered constant in the image subset; and
- the lowest intensity pixels are water.

The first assumption is justified by the selection of a relevant image, whereas the second is justified by the choice of a narrow subset of the full dataset. The true variation of incidence angle in the data is less than 4.1°. The third assumption is in fact not necessarily a good one, as the backscatter of open water is much depending on the wind and may be difficult to discern from e.g. new ice (Johannessen et al., 2007). Nevertheless, it is a reasonable assumption for our examples, and the separation of classes from the image data.

Clustering and Labeling

In the GRD case, only the multilooked pixel intensities are available as features, to be combined in any number of ways. Here, the logarithmic HH &

HV amplitudes of each individual pixel were used, as well as those same metrics from neighboring pixels. To reduce image noise, the image was convoluted with a 3-by-3 smoothing kernel.

Two different clustering algorithms were employed: k-means and SVM. The k-means algorithm was run with two and three centroids respectively, with initialization in the feature space (2-norm) min, max and — in the triple-centroid case — mean. While the k-means algorithm requires little initialization, the SVM needs a training set to form its decision boundaries. Therefore, a random subset of k-means-classified feature vectors was sampled to form the SVM training set. A training-set size of 100 samples was concluded to be of sufficient size. The SVM algorithm was evaluated by the comparison of different kernels and settings, when applying the trained classifier to the full image. Since no ground truth was available, only visual confirmation could be applied.

Since the pixel-wise classification may produce "holes" in the detected object — either actual holes or through misclassification — a hole filling algorithm was applied prior to the labeling. 4-connected labeling was used to cluster neighboring pixels classified as ice.

3.7.2 Results

Data from both GRD and SLC imagery was processed for feature extraction. In the GRD case, the pixel intensity is the main source of information. The "rawer" nature of the SLC image allows for more complex features to be extracted, e.g. through $H/A/\alpha$-decomposition as in Figure 3.21. This comes with the added cost of the decomposition, and very large datasets to deal with. Being the more informative and accessible dataset available, the GRD image was selected for further classification.

Unless otherwise noted, classification plots are displayed with separate colors for each separate detected cluster of ice (as determined by the labeling algorithm); open water is displayed with white, and a red cross is used to mark the mean position of each detection. The different feature sets that were used are described in Table 3.3.

k-means Clustering

The first algorithm to be applied was k-means. As displayed in Figure 3.22a, the naïve approach to classify "ice" vs. "no ice" clearly fails to appreciate the

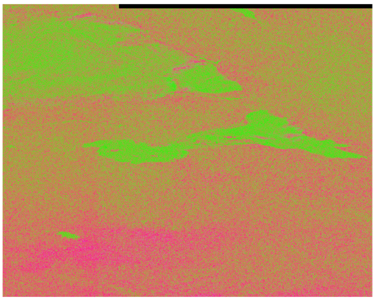


Figure 3.21: SLC dual-pol $H/A/\alpha$ decomposition. Red = H; Green = A; Blue = α

Table 3.3: List of feature sets

ID	Description
A	HH, HV. HH, HV of edge-connected neighbors
B	HH, HV
C	HH
D	HV

Table 3.4: Number of detections for different k-means variants/feature sets

K	Feature set	Targets Found
2	A	2052
3	A	907
3	B	1221
3	C	539
3	D	2727

different types of ice, leading open water to be put in the same class as the less reflective ice regions. This motivates the addition of a third centroid to represent low-scattering ice, as displayed in Figure 3.22b. The classification results for different feature sets/settings are displayed in Table 3.4.

SVM Clustering

The SVM algorithm was initiated using a random set from the triple-centroid k-means classification, where the two classes representing different ice types were grouped together in a single SVM class. The result from the comparison of different kernel methods are displayed in Table 3.5 and Fig-

Table 3.5: Number of detections for different SVM variants

Kernel	**Ft.set**	**Targets found**
Linear	A	855
Linear	B	1038
Linear	C	839
Linear	D	2199
Poly, $(\gamma, d, r) = (1, 2, 0)$	A	1038
Poly, $(\gamma, d, r) = (1, 2, 0)$	B	1213
Poly, $(\gamma, d, r) = (1, 2, 0)$	C	1412
Poly, $(\gamma, d, r) = (1, 2, 0)$	D	2041
Poly, $(\gamma, d, r) = (1, 5, 0)$	A	955
Poly, $(\gamma, d, r) = (1, 5, 0)$	B	1740
Poly, $(\gamma, d, r) = (1, 5, 0)$	C	976
Poly, $(\gamma, d, r) = (1, 5, 0)$	D	2179
Poly, $(\gamma, d, r) = (3, 3, 0)$	A	947
Poly, $(\gamma, d, r) = (3, 3, 0)$	B	1368
Poly, $(\gamma, d, r) = (3, 3, 0)$	C	1413
Poly, $(\gamma, d, r) = (3, 3, 0)$	D	2135
RBF, $\gamma = 1$	A	911
RBF, $\gamma = 1$	B	1145
RBF, $\gamma = 1$	C	964
RBF, $\gamma = 1$	D	2079
RBF, $\gamma = 5$	A	1730
RBF, $\gamma = 5$	B	1136
RBF, $\gamma = 5$	C	1553
RBF, $\gamma = 5$	D	2125

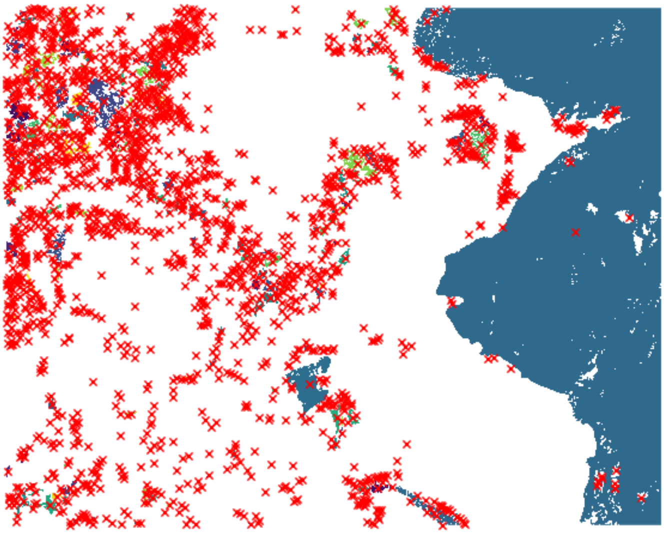
ure 3.23.

Some brief comments;

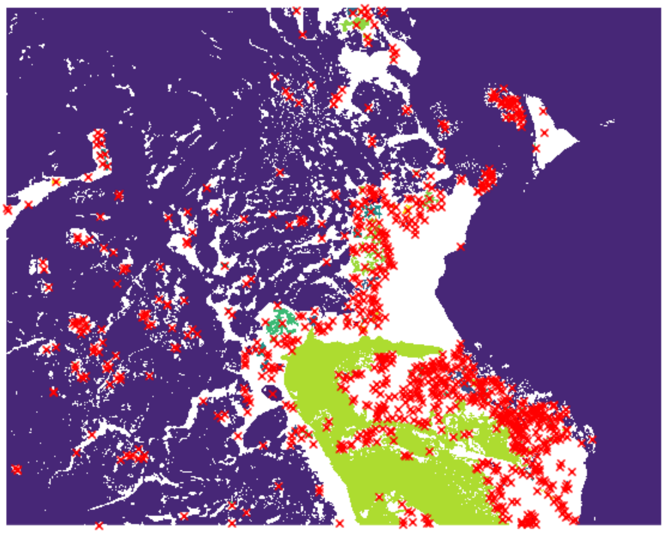
- Studying the decision boundaries in Figure 3.23; in the HH/HV domain, the ice is not not clearly separable from the open water — the intensity limits seem quite arbitrary in view of the data.
- In Figure 3.23e, it is evident that two of the most intense pixels are misclassified as open water, due to the mismatch of a radial kernel and the nature of the classes.
- Despite the lack of separability in the HH/HV domain, visual inspection of the data in Figure 3.23a and Figure 3.23c reveals a seemingly

adequate accuracy.

- In Table 3.5; the inclusion of neighboring pixels clearly have a smoothing effect, leading to fewer (mis?)detections. Also, there is a distinct difference between the detections from HH and HV data respectively. This would seem to imply a certain unreliability in the use of HV data as a detector.

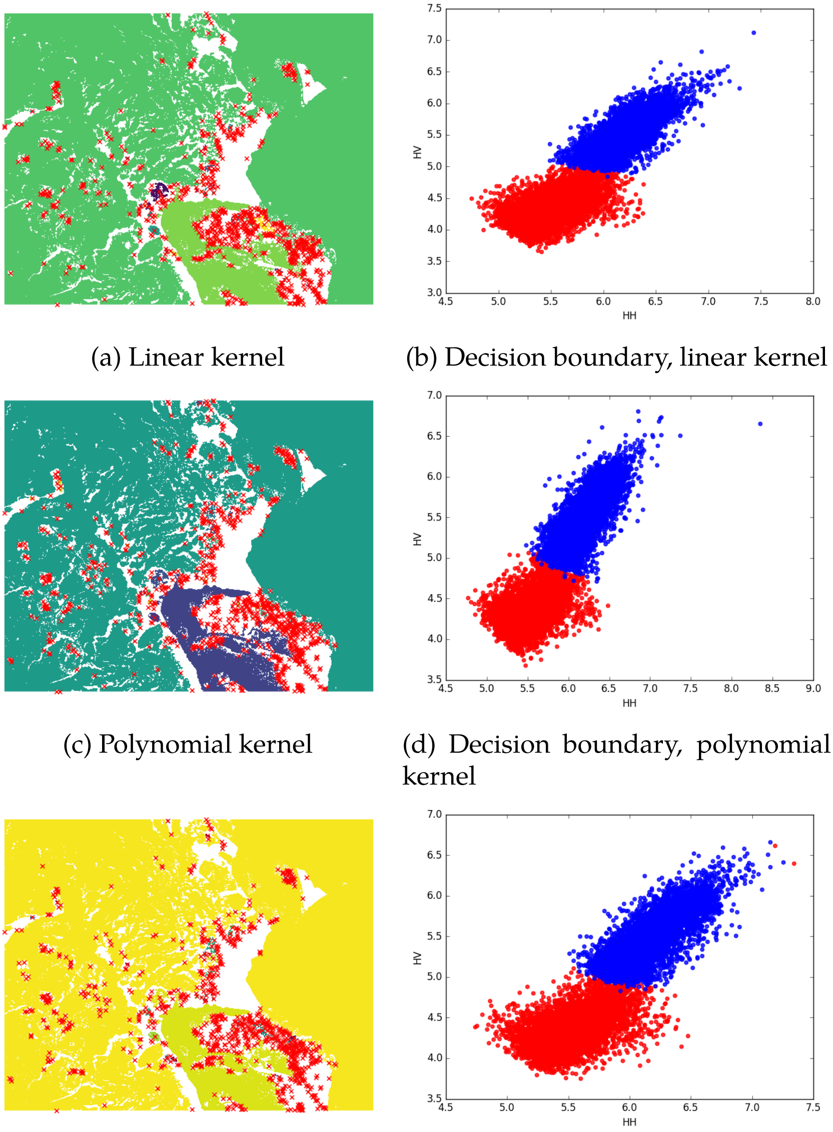


(a) 2 centroids



(b) 3 centroids

Figure 3.22: *k*-means classification. Each cluster of ice is displayed with a unique color. White is the background (water).



(a) Linear kernel

(b) Decision boundary, linear kernel

(c) Polynomial kernel

(d) Decision boundary, polynomial kernel

(e) RBF kernel

(f) Decision boundary, RBF kernel

Figure 3.23: SVM classification with various kernels. Each cluster of ice is displayed with a unique color. White is the background (water).

3.8 Conclusions

While the major currents of the ocean are well studied, the complexity at a small scale is far too great to practically consider in real time applications today. Accurate prediction of ice movements further require exact knowledge of e.g. shape parameters of each ice object, interaction with other objects etc., making pure simulations futile. Hence, and by the principle of Ockham's razor, a simpler model is needed. This is found in the NCV model, a standard model for linear motion. Without an exact model, observations are necessary.

Three of the major sensors used for sea ice monitoring is presented in this chapter: satellite-borne radar, ground (or ship-) based radar and — as an emerging new class — UAS-carried electro-optical sensors. The harsh conditions of the Arctic present a challenging environment far beyond what most research normally consider suitable for UAS use, which makes the field of research even more interesting.

In the continued thesis, detections from each of the sensors are used in tracking problems. The extraction of detections from the raw sensor data varies slightly. In this chapter, a representative example of how extractions can be made from imagery was presented, for the example of SAR imagery but with a generic procedure. In many cases, such as was discussed for the TRI data, model-based imaging techniques such as background modeling can be employed to improve the SNR ratio.

Retrieving ground truth data is a recurrent problem in Multiple Target Tracking (MTT), and no less so when observing remote areas in the Arctic. This makes the extracted data from Section 3.7 approximate at best. The separability of the detection problem from the tracking problem ensures, however, that this has little impact on the continued use of the data for evaluation of the tracking algorithms, as long as detections are made consistently. Of course, this does not apply to actual use of the data, such as in a commercial application.

Part II

Scalable Multi-Target Tracking

4 Concepts in Target Tracking

This chapter introduces some of the concepts of target tracking, as background to the following chapters in this part. The content is primarily based on pieces previously published in

Paper A: Olofsson, J., Brekke, E., Fossen, T. I., and Johansen, T. A. (2017a). Spatially indexed clustering for scalable tracking of remotely sensed drift ice. In *IEEE Aerospace Conference Proceedings*, Big Sky, MT, USA. © 2017 IEEE.

Paper B: Olofsson, J., Brekke, E., and Johansen, T. A. (2017b). Cooperative remote sensing of ice using a spatially indexed labeled multi-Bernoulli filter. In *International Conference Unmanned Aircraft Systems (ICUAS)*, Miami, USA. © 2017 IEEE.

Paper C: Olofsson, J., Veibäck, C., and Hendeby, G. (2017d). Sea ice tracking with a spatially indexed labeled multi-Bernoulli filter. In *20th International Conference on Information Fusion (FUSION)*, Xi'an, China. © 2017 IEEE.

As outlined in Figure 4.1, the tracking of sea ice has a central role in the context of the system described in Chapter 2. This forms the basis for the awareness not only of the individual ice objects, but also of joint characteristics such as velocity fields, and of the statistics used to find regions with particular risk of hazards. To enable global sea ice tracking, it is essential that the algorithms can be partitioned with respect to each individually observed area.

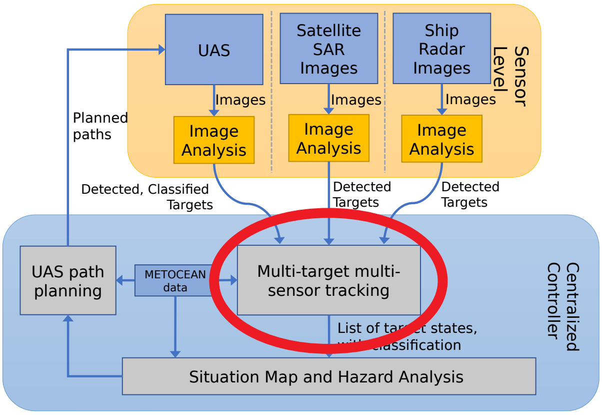


Figure 4.1: Multiple Target Tracking in system context

4.1 Introduction

The modern field of target tracking has evolved from signal filtering theory pioneered during World War II (Wiener, 1965). Through a series of evolutionary steps it has seen the development of the indisputably significant Kalman filter (Kalman, 1960) and later the Particle Filter (Gordon et al., 1993). Further, it has extended into the exploration of scenarios with sensors that can detect multiple sources but not distinguish their identities, giving rise to the subfield of Multiple Target Tracking (MTT).

Commonly, an MTT algorithm does not only have to deal with the tracking of targets, but also with the presence of false reports, new targets and disappearing (“dying”) targets. An MTT algorithm also has to deal with the assignment problem, the problem of deducing which report comes from which target. In fact, with no uncertainty in the assignment, the MTT problem naturally collapses into multiple Single-Target Tracking (STT) problems. Considering that any report could either be false, be a new target, or stem from any of the previously known targets, this is a combinatorially expanding problem. Attempting all possible assignments is therefore prohibitively computationally expensive, and approximations range from taking the closest match to each target (Bar-Shalom, 1987) to generating a limited number of best assignments ordered by probability of correctness (Murty, 1968).

In Section 4.2, the foundations of tracking is summarized for the single target case. Section 4.3 then introduces Random Finite Sets (RFS's) and the measure of the Probability Hypothesis Densities (PHDS) — concepts which have been popularized in the MTT field only in the past decades (Mahler and Zajic, 2001) and are central in the Labeled Multi-Bernoulli (LMB) algorithm used in Chapter 6. In Section 4.4, the multi-target case is properly introduced by expanding upon the assignment problem and details on the algorithms used to generate valid assignment hypotheses.

In the following chapters, a technique called spatial indexing is applied to both the Multiple Hypothesis Tracker (MHT) and LMB algorithms. This is a technique which can be used for quickly finding intersecting squares — here used to find which clusters and targets are affected by sensor updates. The base for the application of spatial indexing is the clustering of the tracking problem into smaller problems, presented in Section 4.5. In Section 4.6 we review algorithms for spatial indexing — Rectangle trees (R-TREES) in particular — and how it can be applied to the target tracking algorithms to index the tracker storage such that the relevant potential targets can be loaded quickly upon the arrival of new data.

Global tracking of ice objects requires the tracker to handle a range of coordinate systems. These include both the ice's geodetic — latitude/longitude — positions, as well as the local Cartesian coordinate systems in which prediction- and measurement updates are performed. Section 4.7 presents the coordinate systems and transformations considered in the implementations developed for the purpose of this thesis. Finally, conclusions are offered in Section 4.8.

4.2 Single Target Tracking

At the core of most MTT algorithms is a set of STT filtering problems. In fact, the underlying STT algorithm for the tracking of each individual target in an MTT can generally be exchanged for other ones or can even — technically — be independently chosen for each target. This entails that advances in the field of STT are easily translatable and applicable in MTT.

This section presents a selection of STT filtering algorithms which have been used in this thesis, starting with the general Bayesian formulation in Section 4.2.1. In the sections that follow, we detail the algorithms and their corresponding assumptions to each of the filtering algorithms used in this thesis — the Extended Kalman Filter (EKF) in Section 4.2.2 and the particle filter in Section 4.2.3. The Gaussian mixture representation is introduced in Section 4.2.4 both for its application as a method for STT filtering, but also as a general representation of continuous numerical distributions, such as a PHD.

Implementation of said algorithms entails further considerations, such as numerical issues, filter restarting, distribution degeneration and memory management, although none of these issues are discussed here.

4.2.1 Bayesian State Estimation

The statistics that describe the estimated state of a tracked target can be represented by a Probability Density Function (PDF). This PDF is often described by a state-space model, with the target state variables $\boldsymbol{x} \in \mathbb{R}^n$ — the essential parameters which we want to estimate. These include the parameters needed to describe the PDF but also those aiding in the modeling of the evolution of the PDF, and the measurement model.

The filtering problem then consists of following the evolution of the target with the goal to continually improve the quality of the estimated PDF, $p(\boldsymbol{x})$, to follow the "true" state distribution as closely as possible. This is performed with the aid of measurements of related properties, $\boldsymbol{z}$. While a true state generally is continuous in time, the estimate thereof is generally estimated at discrete points in time, t_k, $k \in [0, \dots, K]$.

Example 4.1: Target state and measurements

A target tracked in a two-dimensional Euclidean xy-plane is often parametrized by its position and velocity in each dimension, giving the following state vector (for dimensional positions p_x, p_y, and their derivatives):

$$\boldsymbol{x} = \begin{bmatrix} p_x, & p_y, & v_x, & v_y \end{bmatrix}^{\mathsf{T}} \tag{4.1}$$

The goal for the target state estimation algorithm — also known as a filter — is to estimate those parameters as accurately as possible.

As is the case with all sensors described in Section 3.5, many types of sensors employed in tracking of external objects provide images of various types to be interpreted. Knowledge of where the image was taken allows for the extraction of positions, which can be taken as measurements. This yields measurements of the type

$$\boldsymbol{z} = \begin{bmatrix} p_x, & p_y \end{bmatrix}^{\mathsf{T}} \tag{4.2}$$

For a state $\boldsymbol{x}$, with components as in (4.1), we can create a model, $\boldsymbol{h}(\boldsymbol{x})$, of what the measurement would be if a target with said state were to be observed:

$$\boldsymbol{z} = \boldsymbol{h}\left(\boldsymbol{x}\right) = \boldsymbol{H}_k \boldsymbol{x} \tag{4.3}$$

$$\boldsymbol{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.4}$$

In Bayesian state estimation, efforts are focused on finding a tractable solution to the general Bayesian recursion (Mahler and Zajic, 2001). That is, we are interested in finding the following distributions:

$$\textbf{Predicted prior:} \quad \boldsymbol{x}_k|\mathcal{Z}_{1:k-1} \sim p\left(\boldsymbol{x}_k|\mathcal{Z}_{1:k-1}\right), \tag{4.5a}$$

$$\textbf{Corrected posterior:} \quad \boldsymbol{x}_k|\mathcal{Z}_{1:k} \sim p\left(\boldsymbol{x}_k|\mathcal{Z}_{1:k}\right). \tag{4.5b}$$

Here and below, $\boldsymbol{z}_k$ is the measurement at time k, and $\mathcal{Z}_{1:k'}$ the set of all measurements registered up until, and including, time k'. Likewise, $\mathcal{X}_{0:k'}$ is the set of target states $\boldsymbol{x}_k$ until time k'.

To solve (4.5), a general Probabilistic State-Space Model (PSSM) further defines the following distributions (Sarkka, 2013):

$$\textbf{Transition model:} \quad \boldsymbol{x}_k|\boldsymbol{x}_{k-1} \sim p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}\right), \tag{4.6a}$$

$$\textbf{Measurement model:} \quad \boldsymbol{z}_k|\boldsymbol{x}_k \sim p\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right), \tag{4.6b}$$

$$\textbf{Initial state:} \quad \boldsymbol{x}_0 \sim p\left(\boldsymbol{x}_0\right). \tag{4.6c}$$

An initial state distribution guess, $p\left(\boldsymbol{x}_0\right)$, can e.g. be inferred from the first measurement, or from prior information. Note that in the formulation of (4.6a) the Markovian property — $p\left(\boldsymbol{x}_k|\mathcal{X}_{0:k-1}\right) = p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}\right)$ — has been implicitly used, implying that all relevant information for the evolution of the model can be induced from the previous state alone. While the general Bayesian update equations below do not require this property, it is often assumed in practice.

The distributions of the PSSM are updated through the Bayesian recursion:

Prediction update

$$p\left(\boldsymbol{x}_k|\mathcal{Z}_{1:k-1}\right) = \int p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}\right) p\left(\boldsymbol{x}_{k-1}|\mathcal{Z}_{1:k-1}\right) d\boldsymbol{x}_{k-1} \tag{4.7a}$$

Measurement update

$$p\left(\boldsymbol{x}_k \middle| \mathcal{Z}_{1:k}\right) = \frac{p\left(\boldsymbol{z}_k \middle| \boldsymbol{x}_k\right) p\left(\boldsymbol{x}_k \middle| \mathcal{Z}_{1:k-1}\right)}{p\left(\boldsymbol{z}_k \middle| \mathcal{Z}_{1:k-1}\right)}. \tag{4.7b}$$

The raw formulation of (4.7) is rarely used directly, but are the expressions from which all other related algorithms can be derived by applying a specific set of assumptions. As discussed in Chapter 6, even the MTT filter can be formulated using the same equations, albeit with an extended meaning to the variables.

When applying Bayesian state estimation, the PDFS of (4.6) are generally chosen such that the choice of distribution of $\boldsymbol{x}$ is preserved through the updates, i.e only the parameters of the distribution are updated. To model this update, we define the state transition function $\boldsymbol{f}$, as

$$\boldsymbol{x}_k = \boldsymbol{f}\left(\boldsymbol{x}_{k-1}\right) + \boldsymbol{w}_k, \tag{4.8}$$

where $\boldsymbol{w}_k$ is introduced as additive noise describing the uncertainty in the prediction model. The equations of (4.7) do not explicitly require linear additive noise (it may enter nonlinearly through $\boldsymbol{f}$), although it is a common assumption to which we restrict the discussion. In the measurement update (4.7a), the choice of measurement model $p\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right)$ determines how measurement noise enters the estimate. Similarly to (4.8), it is often defined with additive noise, yielding the measurement model

$$\boldsymbol{z}_k = \boldsymbol{h}\left(\boldsymbol{x}_k\right) + \boldsymbol{v}_k. \tag{4.9}$$

The noise variables $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are often modeled as zero-mean Gaussian distributions (see Definition 4.1):

$$\boldsymbol{w}_k \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{Q}_k\right), \tag{4.10a}$$

$$\boldsymbol{v}_k \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{R}_k\right), \tag{4.10b}$$

where $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are positive definite, symmetric covariance matrices of dimensions $N_x \times N_x$ and $N_z \times N_z$, respectively.

In this thesis, the motion of each individual sea ice object is modeled according to the Nearly Constant Velocity (NCV) model, as defined in Definition 4.2.

Definition 4.1 (*Gaussian Distribution*).
The Gaussian distribution PDF, parametrized by its mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Pi}$ is given by

$$\mathcal{N}\left(x|\boldsymbol{\mu}, \boldsymbol{\Pi}\right) = |2\pi\boldsymbol{\Pi}|^{-1/2} \exp\left[-\frac{1}{2}\left(x-\boldsymbol{\mu}\right)^{\mathsf{T}} \boldsymbol{\Pi}^{-1}\left(x-\boldsymbol{\mu}\right)\right]. \tag{4.11}$$

□

Definition 4.2. Nearly Constant Velocity (NCV) model
The NCV model (Li and Jilkov, 2003), defines the state transition function,

$$\boldsymbol{x}_k = \boldsymbol{f}_k\left(\boldsymbol{x}_{k-1}\right) + \boldsymbol{w}_k = \boldsymbol{F}_k \boldsymbol{x}_{k-1} + \boldsymbol{w}_k, \tag{4.12a}$$

$$\boldsymbol{F}_k = \begin{bmatrix} 1 & 0 & \Delta T_k & 0 \\ 0 & 1 & 0 & \Delta T_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.12b}$$

$$\Delta T_k = t_k - t_{k-1}. \tag{4.12c}$$

This model also entails the transition model noise covariance, for velocity covariance σ_v^2, to be

$$\boldsymbol{Q}_k = {\sigma_v}^2 \begin{bmatrix} \Delta T_k^3/3 & 0 & \Delta T_k^2/2 & 0 \\ 0 & \Delta T_k^3/3 & 0 & \Delta T_k^2/2 \\ \Delta T_k^2/2 & 0 & \Delta T_k & 0 \\ 0 & \Delta T_k^2/2 & 0 & \Delta T_k \end{bmatrix}. \tag{4.13a}$$

□

4.2.2 The Extended Kalman Filter

The single most prevalent method for Bayesian state estimation is the family of Kalman filters, introduced by Kalman (1960).

The derivation of the linear Kalman filter (Kalman, 1960) assumes a linear system, although the theory can be extended to nonlinear systems through the assumption that the system can be "sufficiently" described by a linearization in a large-enough neighbourhood around the current state. The linear Kalman filter is an exact solution to the recursion of (4.7) in the case of linear transition and measurement models, with additive zero-mean Gaussian noise. Even for non-Gaussian noise (albeit still zero-mean), the linear Kalman filter is proved to be the Best Linear Unbiased Estimator (BLUE) (Anderson and Moore, 1979), although the generalization — the EKF — provides no such guarantees. The equations for the EKF are given below.

A detailed description of the algorithm can be found in e.g. Jazwinski (1970); Smith et al. (1962).

Below, we use the notation that the subscript $k|k'$ for the variables $\hat{\boldsymbol{x}}$ and $\boldsymbol{P}$ denotes the parameter's value at time k, given measurement until time k'. If the times are equal, i.e. from the measurement update, only one is typed out, as the subscript k'.

Prediction update

$$p\left(\boldsymbol{x}_k|\mathcal{Z}_{1:k-1}\right) = \mathcal{N}\left(\boldsymbol{x}_k|\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1}\right) \tag{4.14a}$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k-1}\right) \tag{4.14b}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k\boldsymbol{P}_{k-1}\boldsymbol{F}_k{}^{\mathsf{T}} + \boldsymbol{Q}_k \tag{4.14c}$$

$$\boldsymbol{F}_k = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}} \tag{4.14d}$$

Measurement update

$$p\left(\boldsymbol{x}_k|\mathcal{Z}_{1:k}\right) = \mathcal{N}\left(\boldsymbol{x}_k|\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}\right) \tag{4.15a}$$

$$\hat{\boldsymbol{z}}_k = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_{k|k-1}\right) \tag{4.15b}$$

$$\boldsymbol{\iota}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_k \tag{4.15c}$$

$$\boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\mathsf{T}} + \boldsymbol{R}_k \tag{4.15d}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k\boldsymbol{S}_k^{-1} \tag{4.15e}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\boldsymbol{\iota}_k \tag{4.15f}$$

$$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{P}_{k|k-1} \tag{4.15g}$$

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k-1}} \tag{4.15h}$$

In (4.15), $\boldsymbol{\iota}_k$ is referred to as the innovation, which in the linear Gaussian case is distributed according to:

$$\boldsymbol{\iota}_k \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{S}_k\right). \tag{4.16}$$

4.2.3 The Particle Filter

A more generic representation of a PDF can be constructed using a weighted summation of particles — discrete points in state space, each with an individual weight after their probability of correctly corresponding to the true state. For a set of N_p particles:

$$p\left(\boldsymbol{x}_k\right) \approx \sum_{i=1}^{N_p} w_k^i \delta_{\hat{\boldsymbol{x}}_k^i}\left(\boldsymbol{x}_k\right), \tag{4.17a}$$

$$\left\{\left(w_k^i, \hat{\boldsymbol{x}}_k^i\right) \quad : \quad i \in \left[1, \ldots, N_p\right]\right\}, \tag{4.17b}$$

$$\sum_{i=1}^{N_p} w_k^i = 1. \tag{4.17c}$$

Here, $\delta_{\hat{\boldsymbol{x}}_k^i}\left(\boldsymbol{x}\right)$ is the Dirac delta function — zero everywhere but at $\boldsymbol{x} = \hat{\boldsymbol{x}}_k^i$ where it integrates to 1. The mean and covariance of the particle distribution can be calculated as:

$$\hat{\boldsymbol{x}}_k = \sum_{i=1}^{N_p} w_k^i \hat{\boldsymbol{x}}_k^i, \tag{4.18a}$$

$$\operatorname{cov}\left(\boldsymbol{x}_k\right) = \sum_{i=1}^{N_p} w_k^i \left(\left(\hat{\boldsymbol{x}}_k^i - \hat{\boldsymbol{x}}_k\right)\left(\hat{\boldsymbol{x}}_k^i - \hat{\boldsymbol{x}}_k\right)^{\mathsf{T}}\right). \tag{4.18b}$$

Note however that unlike the Gaussian distribution assumed in the Kalman filter, a particle distribution can approximate any distribution, even multi-modal ones where the terms mean and covariance are less relevant.

There are numerous ways to use the particle distribution in Bayesian filtering. The one presented here, is sometimes known as the *Bootstrap filter*. The initialization, prediction, and measurement update is defined as follows:

Initialization

$$\hat{\boldsymbol{x}}_0^i \sim p\left(\boldsymbol{x}_0\right), \, w_k^i = \frac{1}{N_p} \quad \forall i \in \left[1, \ldots, N_p\right] \tag{4.19}$$

Prediction update

$$\hat{\boldsymbol{x}}_k^i \sim p\left(\boldsymbol{x}_k | \hat{\boldsymbol{x}}_{k-1}^i\right) \quad \forall i \in \left[1, \ldots, N_p\right] \tag{4.20}$$

Measurement update

$$w_k^i \propto w_{k-1}^i p\left(z_k | \hat{x}_{k|k-1}^i\right) \quad \forall i \in \left[1, \ldots, N_p\right], \tag{4.21a}$$

$$\sum\nolimits_{i=0}^{N_p} w_k^i = 1. \tag{4.21b}$$

For practical purposes, as first proposed by Gordon et al. (1993), a resampling step (Hol et al., 2006; Gustafsson, 2018) needs to be added to the algorithm to maintain sample diversity. This algorithm reselects particles from the particle distribution based on their relevance. Different resampling strategies are discussed in, e.g., Hol et al. (2006).

4.2.4 Gaussian Mixtures

A Gaussian mixture is a numerical representation similar to the particle distribution, but instead of the discrete point values of particles, Gaussian mixtures form a continuous measure evaluated as the sum of Gaussian components:

$$\mathcal{M} = \left\{\left(w_k^i, \hat{x}_k^i, \boldsymbol{P}_k^i\right) \quad : \quad i \in \left[1, \ldots, N_p\right]\right\} \tag{4.22a}$$

$$p\left(x_k\right) = \mathcal{GM}\left(x_k | \mathcal{M}\right) = \sum_{i=1}^{N_p} w_k^i \mathcal{N}\left(x_k | \hat{x}_k^i, \boldsymbol{P}_k^i\right), \tag{4.22b}$$

$$\sum_{i=1}^{N_p} w_k^i = 1, \quad w_k^i \geq 0 \, \forall i \in \left[1, \ldots, N_p\right]. \tag{4.22c}$$

For filtering purposes, each weighted Gaussian component can be handled in independent EKFS, with the weights being adjusted in the measurement update according to (4.21) (Anderson and Moore, 1979). The mean and covariance of the mixture can be calculated as:

$$\hat{x}_k = \sum_{i=1}^{N_p} w_k^i \hat{x}_k^i, \tag{4.23a}$$

$$\operatorname{cov}\left(x_k\right) = \sum_{i=1}^{N_p} w_k^i \left(\boldsymbol{P}_k^i + \left(\hat{x}_k^i - \hat{x}_k\right)\left(\hat{x}_k^i - \hat{x}_k\right)^{\mathsf{T}}\right). \tag{4.23b}$$

For further treatment of Gaussian mixtures for STT, see e.g. Anderson and Moore (1979); Wills et al. (2017).

By removing or appropriately adjusting the constraint of $\sum_{i=1}^{N_p} w_k^i = 1$, the representation can be used to approximate any continuous function, such as the PHD of a tracking problem (Vo and Ma, 2006). In essence, the PHD is a weighted sum of PDFS (Mahler, 2007b) and notably, the weighted sum of Gaussian mixtures is again a Gaussian mixture,

$$\sum_{j=1}^{N_{\mathcal{M}}} \omega_j \mathcal{GM}\left(\boldsymbol{x}|\mathcal{M}_j\right) = \mathcal{GM}\left(\boldsymbol{x}\left|\bigcup_{j}^{N_{\mathcal{M}}} \left\{\left(\omega_j w_k^i,\ \hat{\boldsymbol{x}}_k^i,\ \boldsymbol{P}_k^i\right) \quad : \quad i \in \left[1,\ldots,N_{p,j}\right]\right\}\right.\right), \tag{4.24}$$

for mixture weights ω_j and mixture size $N_{p,j}$ for mixture j of the sum of $N_{\mathcal{M}}$ mixtures.

4.3 Random Finite Sets and Probability Hypothesis Densities

As proposed by Mahler (Mahler and Zajic, 2001), the variables of target tracking — and MTT in particular — can be expressed as Random Finite Sets (RFS's) — sets which not only contain objects with statistical properties but are themselves of randomly distributed cardinality. In an RFS, each potential element is included in the set with a given probability. Specifically, a Bernoulli RFS is a random set which is empty with probability $1-r$, and with probability r a singleton. For an element $\boldsymbol{x}$ with an associated distribution $p\left(\boldsymbol{x}\right)$, the PDF for the Bernoulli RFS χ is given by (Mahler, 2007b)

$$\pi\left(\chi\right) = \begin{cases} 1-r, & \text{if } \chi = \emptyset, \\ r \cdot p\left(\boldsymbol{x}\right), & \text{if } \chi = \left\{\boldsymbol{x}\right\}. \end{cases} \tag{4.25}$$

A multi-Bernoulli RFS is the result of the union of N_{mb} independently Bernoulli-distributed RFS's $\chi^{(i)}$, given by $\chi = \bigcup_{i=1}^{N_{mb}} \chi^{(i)}$. Consequently, the multi-Bernoulli RFS can be parametrized by the set $\left\{\left(r^{(i)}, p^{(i)}\right)\right\}_{i=1}^{N_{mb}}$. For each member of χ, $p\left(\boldsymbol{x}\right)$ can be updated as individual, although possibly interdependent, targets according to the single-target interpretation of (4.7).

The Labeled Multi-Bernoulli (LMB) RFS is obtained by augmenting each Bernoulli RFS with a unique label, $\ell \in \mathcal{L}$, in the set of all labels, $\mathcal{L}$ — one label for each potential object in the set. The RFS can thus be described by the set

$$\left\{\left(r^{(\ell)}, p^{(\ell)}(x)\right)\right\}_{\ell\in\mathcal{L}}. \tag{4.26}$$

Using Finite Set Statistics (FISST) (Mahler, 2007b), the Bayesian filtering equations (4.7) can be generalized to the MTT case by reconceptualizing the meaning of the involved terms. Just as a single-target distribution can be described by its moments, so can a multi-target distribution. This introduces the question of the interpretation of a multi-target moment. The definition of the first moment employed in FISST is known as the Probability Hypothesis Density (PHD) (Winter and Stein, 1993) and corresponds in each point to the density of expected set objects at that point. This density, $v(\boldsymbol{x})$ is often integrated over a region S, giving the expected number of objects within that region:

$$V(S) = \mathbb{E}\left[|\chi \cap S|\right] = \int_S v(\boldsymbol{x}) d\boldsymbol{x}, \tag{4.27}$$

where χ is the RFS of all targets — stochastic in cardinality and content. This set may or may not be explicitly known, as the measure here is the set's expected cardinality, not contents. The intersection operator indicates the windowing of each target's PDF — limited by a given minimum probability — with the region of interest. The PHD of a LMB RFS can be calculated by

$$v(\boldsymbol{x}) = \sum_{\ell\in\mathcal{L}} r_\ell \cdot p_\ell(\boldsymbol{x}). \tag{4.28}$$

4.4 Hypothesis-based Multi-Target Tracking

The terminology often employed in common MTT literature, and consequently here, is based on the following definitions.

At any given time instance, a sensor delivers a *scan* (see Figure 4.2) — an unordered set of reports, all detected at the same instance. This list is exhaustive, i.e. it contains all reports from that sensor from that time[1]. However, the reports cannot be tied to a target identity with certainty. The scan set $\mathcal{Z}_k$ contains all reports (indexed by r), $z_{k,r}$ for time k whereas

[1] The scan can however be subdivided for each cluster in the MTT, as described in Section 4.5.

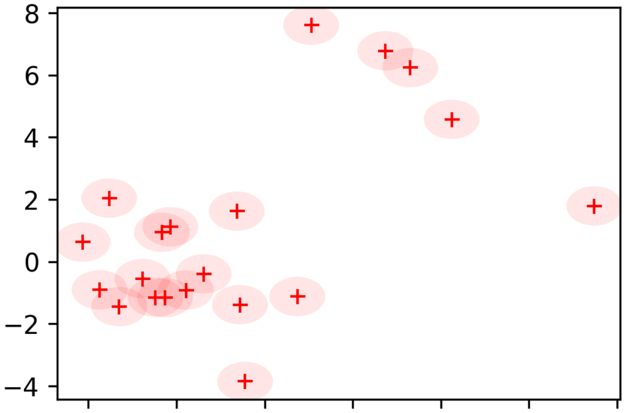


Figure 4.2: A scan is an unordered collection of measurements from the same time instance and sensor.

$$\mathcal{Z}_{1:k} = \mathcal{Z}_{1:k-1} \cup \{\mathcal{Z}_k\} \tag{4.29}$$

is the, recursively defined, collection of all scans up until and including time k. Note that in this syntax, time of observation is considered to be available for each $z_{k,r}$ through its subscripted time-index, meaning that each $\mathcal{Z}_k$ added to the collection can be subsequently individually extracted.

The purpose of an MTT algorithm is to estimate a list of tracked *targets*. In hypothesis-based MTT, each target is associated with a set of hypothetical *tracks* — STT filters, each of which is the result of a unique sequence of hypothetical assignments to the target. These assignments can indicate association with a specific report, but also missed detections or even assignments under alternative motion models (Kurien, 1990).

In this thesis, we make considerable use of *indicator functions* — binary functions to select between cases. For example, $A^{\theta}_{z\leftrightarrow\ell}$ is the assignment indicator function, defined as

$$A^{\theta}_{z\leftrightarrow\ell} \triangleq \begin{cases} 1, & \text{if assignment } \theta \text{ assigns target } \ell \text{ to report } z, \\ 0, & \text{otherwise.} \end{cases} \tag{4.30}$$

4.4.1 Assignments and Hypotheses

In hypothesis-based MTT, tracks commonly start with the event of a report that is not unambiguously originated from a previously known target. A track evolves as reports are assigned to it under the assumption that they are originated from the same target, and as the number of tracks grows, the number of possible assignments grows combinatorially. The problem of connecting the reports with targets is known as the Linear Assignment Problem (LAP) (Kuhn, 1955). Since the chances of correctly assigning all targets are slim, multiple hypothetical assignments are often considered, along with their likelihood of being correct. Given multiple hypothetical assignments, the hypothetical tracks of a target can be visualized as a tree (Kurien, 1990), as displayed in Figure 4.3 where each graph leaf track is reached via the "path" of assignment choices that lead to it. Each track is updated under the assumption that its history of hypothetical assignments is correct. A set of tracks as updated by assignments over time is an hypothesis, and the hypothesis containing the tracks up until the previous time step is called the parent hypothesis.

In an assignment, a report can either be associated with a track, or be considered *extraneous*, meaning that it is considered to be either from a new (previously unknown) target, or to be a false report. The two cases can generally not be distinguished with a single observation, and so are treated jointly. Such an assignment can be used to initiate a new target to be considered in the tracker (Bar-Shalom et al., 2007), awaiting further observations to confirm its actual existence.

With the definition employed in this thesis, an assignment

i) assigns all reports of a scan to either pre-existing or new tracks; and

ii) assigns a "missed", null assignment to any track which is present in the parent hypothesis but not associated with any of the scan's reports.

Since multiple assignments θ are formed at each timestep, we use index ϑ to designate a specific assignment or hypothesis.

Hypotheses, again as defined in this thesis, are sets containing tracks resulting from assignments from one or more of consecutive sets of scans. Hypotheses can either be generated anew for each incoming scan (Popp et al., 2001) or be formed as a continuation based on an existing hypothesis

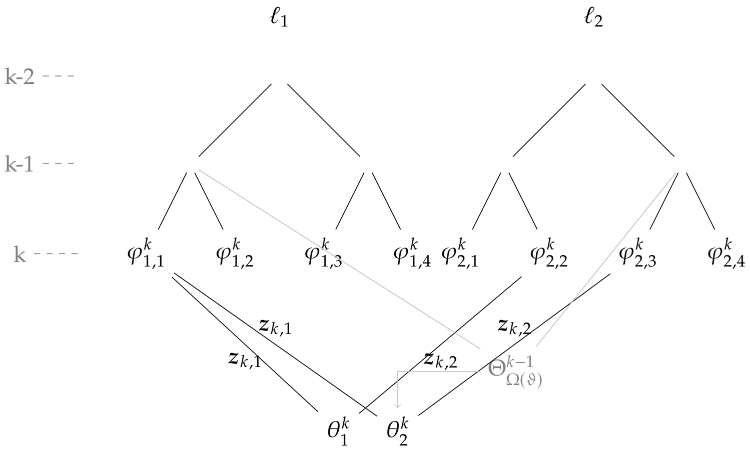


Figure 4.3: The different hypothetical tracks ($\varphi^k_{\ell,\cdot}$) of a target ℓ can be visualised as a tree (Kurien, 1990), with branches indicating that multiple options exist, e.g. for which report to assign to the target. Each vertical level indicates a timestep, and the leaves point to the latest hypothetical state estimate for each target. Global hypotheses Θ^ϑ_k are formed as combinations of compatible leaves. New hypotheses are formed by extending the hypotheses from the previous timestep, $\Theta^{k-1}_{\Omega(\vartheta)}$, with assignments of new reports, θ^k_ϑ.

from the previous timestep. The latter can be expressed mathematically by that each hypothesis Θ^k_ϑ from time k fulfills

$$\Theta^k_\vartheta = \Theta^{k-1}_{\Omega(\vartheta)} \cup \left\{\theta^k_\vartheta\right\}, \tag{4.31}$$

for an assignment θ^k_ϑ, and the mapping $\Omega(\vartheta)$ that links assignment θ^k_ϑ to the hypothesis of time $k-1$ from which it was extended, and thus describes the historical assignments which when generating the new assignment are assumed true.

Example 4.2: Conteptual Contents of a Single Assignment

A selected parent hypothesis contains two targets, each with a track selected among each target's hypothetical tracks — $\varphi^{k-1}_{1,1}$ and $\varphi^{k-1}_{2,1}$, respectively , where track $\varphi^{k-1}_{\ell,j}$ is the j'th of the target ℓ's hypothetical tracks at time $k-1$. A scan is received with a single report $z_{k,1}$ and a new assignment θ^k_ϑ is generated that

i) assigns report $z_{k,1}$ to track $\varphi_{1,1}$, but *ii*) finds no matching report to $\varphi_{2,1}$.

The resulting new assignment contains the following pairs of association:

$$\theta_{\vartheta}^{k} = \left\{ \left(z_{k,1},\, \varphi_{1,1}^{k} \right), \quad \left(\emptyset,\, \varphi_{2,1}^{k} \right) \right\},$$

where $\emptyset$ denotes the null assignment of a track without an associated report at time k.

Different reports assigned to the same target will each result in a new hypothetical track being created. If different hypothetical assignments assign the same report to different targets, the resulting tracks will be incompatible. Hypotheses of compatible tracks are called global hypotheses. Targets that do not share any assignments, in any hypothesis, at any point in history (or, approximately, as far back as we care) are independent, and can be treated in separate independent *clusters* — groups of targets which share ambiguous reports. The purpose is to reduce the computational complexity by limiting the size of the multi-hypothesis assignment problem.

It is assumed that each track can be modeled as a standard single-target state estimation problem, conditioned on the validity of its history of hypothetical assignments.

4.4.2 Hypothesis Generation

Since all incoming reports theoretically can be assigned to any existing track, the set of possible hypotheses grows exponentially with time. An effective way to limit the complexity of an hypothesis-based MTT algorithm is to focus on generating and evaluating only the most relevant assignments, such as the ones most likely to be true. The generation of the single best possible assignment is known as the LAP. The problem can be formulated by defining a cost matrix $\boldsymbol{C} \in \mathbb{R}^{n \times m}$, with matrix elements c_{ij} from row $i \in [1, \ldots, n]$ and column $j \in [1, \ldots, m]$:

$$\begin{aligned} &\min \sum_{i,j} c_{ij} s_{ij} \\ &\sum_{j} s_{ij} = 1, \quad \forall i, \quad \sum_{i} s_{ij} \leq 1, \quad \forall j \\ &s_{ij} \in \{0, 1\} \end{aligned} \tag{4.32}$$

The gist of this problem is to achieve the lowest total cost selecting, from an assignment cost matrix $\boldsymbol{C}$, exactly one value in each row and at most one value in each column. The resulting 1-valued elements s_{ij} is a binary selection of one unique column per row. In an implementation, this result can be represented either as a binary matrix, or as a vector of column indices — one for each row. One LAP, with separate $\boldsymbol{C}$ matrices, is set up for each prospective parent hypothesis. Thus, the solution of the assignment problem is conditioned on the assignments of the parent hypothesis that $\boldsymbol{C}$ is based upon.

Given what we put into the $\boldsymbol{C}$-matrix, the interpretation of the result differs, as shown by Examples 4.3 and 4.4.

Example 4.3: Assigning reports to tracks

Given reports $\{z_1, z_2\}$ and tracks $\{\varphi_1, \varphi_2\}$ we define the $\boldsymbol{C}$ matrix as

$$\boldsymbol{C} = \begin{pmatrix} {}^{z_1}\Lambda_{\varphi_1} - {}^{m}\Lambda_{\varphi_1} & {}^{z_1}\Lambda_{\varphi_2} - {}^{m}\Lambda_{\varphi_2} & {}^{z_1}\Lambda_{\xi} & \infty \\ {}^{z_2}\Lambda_{\varphi_1} - {}^{m}\Lambda_{\varphi_1} & {}^{z_2}\Lambda_{\varphi_2} - {}^{m}\Lambda_{\varphi_2} & \infty & {}^{z_2}\Lambda_{\xi} \end{pmatrix}.$$

Here, and as further defined in Chapter 5,

i) ${}^{z_i}\Lambda_{\varphi_j}$ is the cost of assigning report z_i to track φ_j

ii) ${}^{m}\Lambda_{\varphi_j}$ is the cost of considering the target of track φ_j missed; and

iii) ${}^{z_i}\Lambda_{ex}$ is the cost of considering the report z_i extraneous, i.e. either a new target, or a false detection.

An example solution $s_{12} = s_{24} = 1$ (all other $s_{ij} = 0$) would generate the hypothesis that report z_1 is associated with track φ_2, and report z_2 is considered extraneous.

Report–track assignment is used in the MHT implementation described in Chapter 5.

Example 4.4: Assigning targets to reports

Given reports $\{z_1, z_2\}$ and targets $\{\ell_1, \ell_2\}$ we define the $\boldsymbol{C}$ matrix as

$$\boldsymbol{C} = \begin{pmatrix} {}^{z_1}\Lambda_{\ell_1} & {}^{z_2}\Lambda_{\ell_1} & {}^{n}\Lambda_{\ell_1} & \infty & {}^{F}\Lambda_{\ell_1} & \infty \\ {}^{z_1}\Lambda_{\ell_2} & {}^{z_2}\Lambda_{\ell_2} & \infty & {}^{n}\Lambda_{\ell_2} & \infty & {}^{F}\Lambda_{\ell_2} \end{pmatrix},$$

where ${}^{z_j}\Lambda_{\ell_i}$ is the cost assigned to associating target ℓ_j to report z_j. ${}^{n}\Lambda_{\ell_i}$ and ${}^{F}\Lambda_{\ell_j}$ is the cost associated with assigning the target as non-associated or false, respectively. These constants are further discussed in Chapter 6.

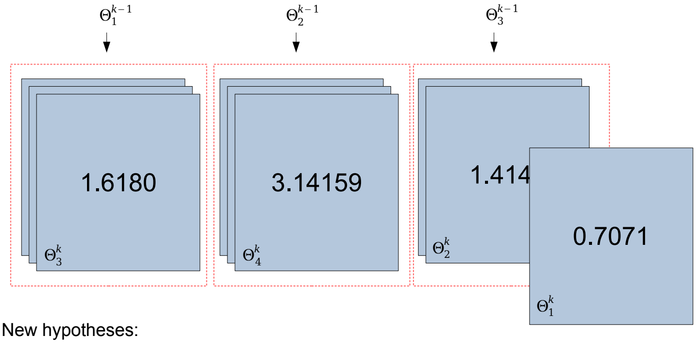
For example, solving (4.32) may yield that $s_{11} = s_{26} = 1$ (all other $s_{ij} = 0$), indicating that report z_1 is associated with target ℓ_1 and that target ℓ_2 is considered a false target.

Target–report assignment is used in the LMB implementation described in Chapter 6.

Each solution that fulfills the constraints put up in (4.32) corresponds to a hypothetical assignment, and the "cheapest" assignment is the best guess of assignment of the reports received for this given timestep, for the given parent hypothesis. Several algorithms exist to solve the LAP problem — the auction algorithm (Bertsekas, 1988), the Hungarian algorithm (Kuhn, 1955) and Jonker-Volgenant (Jonker and Volgenant, 1987) being notable mentions.

The task of finding the single best assignment was extended in an algorithm due to Murty (1968) to that of finding the N_h best assignments, for a given assignment cost matrix $\boldsymbol{C}$, i.e. for a single parent hypothesis. However it is also necessary to compare the assignments from different parents, as illustrated in Figure 4.4. Doing so, for each incoming scan, the N_h best assignment hypotheses can be selected without full enumeration, effectively limiting the exponential growth of assignment hypotheses. Note that removing tracks that are not selected for any of the N_h best global hypotheses decreases the need for post-update pruning techniques, such as reviewed by e.g. Blackman (2004).

The Murty algorithm, with the underlying LAP solver, is of polynomial complexity in the number of tracks with which the reports can be associated (Cox and Hingorani, 1996), and it is therefore of interest to limit the number of tracks considered for assignment. By applying *gating*, we can limit the number of tracks that are considered for association, and thus reduce the size of the $\boldsymbol{C}$-matrix to include only those.



New hypotheses:

Order	Score	Name	Parent
1	0.7071	Θ_1^k	Θ_3^{k-1}
2	1.414	Θ_2^k	Θ_3^{k-1}
3	1.6180	Θ_3^k	Θ_1^{k-1}
4	3.14159	Θ_4^k	Θ_2^{k-1}

Figure 4.4: Here, hypotheses from a given parent are illustrated as an ordered deck of hypothesis cards. To draw hypotheses from multiple parents (and thus multiple LAPS), a single iteration of Murty's algorithm is performed for each parent, generating the top cards of the decks. The best hypothesis of those is selected and removed from the deck, and a new Murty iteration for that LAP generates the deck's next top card. This is repeated to draw N_h hypotheses of decreasing probability.

4.5 Gating and Clustering

Clustering is the grouping of targets, such that no target in any cluster shares a report, by assignment in any hypothesis, with a target in another cluster (Kurien, 1990). As no information is shared, all clusters are independent of each other and can be updated separately. Theoretically, any report can be assigned to any track, but in practice many of the possible assignments are very unlikely. That is to say, tracks and reports which are unlikely to be associated are approximately independent. We therefore define a *gate* as a function of a report–track pair such that if the function is below a given threshold, the assignment need not be considered (Reid, 1979; Blackman, 2004). Targets whose tracks have at any point been assigned the same report share a connection, and connected targets belong to the same cluster — independent from the other targets, but not from each other.

If a report falls outside the gate for all tracks in a cluster, the report will not be associated with that cluster. If a report is associated with several clusters, these need to be merged such that each report is associated with only a single cluster (Reid, 1979). Note that this can have a "snowballing" effect, leading to seemingly unrelated targets belonging to the same cluster, as they are indirectly connected by ambiguous reports.

4.5.1 Association Probability Bounds and Overlap Gating

For reports and tracks with Gaussian distributions, the probability of association is proportional to the overlap integral of their respective distributions. This is proportional (Collins and Uhlmann, 1992) to the probability of the innovation, (4.16), itself a Gaussian.

As a general result, the area bounded by an iso-probability limit corresponds, for any Gaussian $\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Pi}\right)$, to an ellipse, which can be written as the set

$$\left\{\boldsymbol{x} \quad : \quad \left(\boldsymbol{x}-\boldsymbol{\mu}\right)^{\top}\boldsymbol{\Pi}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right) \leq \gamma_{\varepsilon}\right\}. \tag{4.33}$$

The covariance-normalized boundary-distance used, $d^2_{\text{Mah}} = \left(\boldsymbol{x}-\boldsymbol{\mu}\right)^{\top}\boldsymbol{\Pi}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)$, is known as the squared Mahalanobis distance (Mahalanobis, 1936). The squared Mahalanobis distance is, for a random Gaussian vector of size N_d, distributed according to the chi-squared distribution of N_d degrees of freedom. Hence, γ_{ε} can be selected for a desired gating probability. The size and orientation of the ellipse can be calculated from the covariance matrix $\boldsymbol{\Pi}$ and the error ellipsoid parameter γ_{ε} (Ribeiro, 2004).

Because of its geometrical interpretation, gating through a limit of the association probability is referred to as ellipsoidal gating. For a report–track pair (both with Gaussian distribution), the corresponding ellipse is

$$\left\{\boldsymbol{z} \quad : \quad \left(\boldsymbol{z}-\hat{\boldsymbol{z}}_{\varphi}\right)^{\top}\left(\bar{\boldsymbol{P}}+\boldsymbol{R}\right)^{-1}\left(\boldsymbol{z}-\hat{\boldsymbol{z}}_{\varphi}\right) \leq \gamma_{\varepsilon}\right\}. \tag{4.34}$$

where $\hat{\boldsymbol{z}}_{\varphi} = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_{\varphi}\right)$ for track estimate $\hat{\boldsymbol{x}}_{\varphi}$ of track φ; $\bar{\boldsymbol{P}} = \boldsymbol{H}\boldsymbol{P}_{\varphi}\boldsymbol{H}_k^{\top}$, all with time indices dropped. If a report $\boldsymbol{z}$ fulfills the criterion (i.e. is within the ellipse), it is positively gated. Collins and Uhlmann (1992) shows that a necessary condition for (4.34) is that

$$\exists \boldsymbol{x} : \quad \begin{cases} \left(\boldsymbol{x}-\hat{\boldsymbol{z}}_{\varphi}\right)^{\top}\bar{\boldsymbol{P}}^{-1}\left(\boldsymbol{x}-\hat{\boldsymbol{z}}_{\varphi}\right) \leq \gamma_{\varepsilon} \\ \left(\boldsymbol{x}-\boldsymbol{z}\right)^{\top}\boldsymbol{R}^{-1}\left(\boldsymbol{x}-\boldsymbol{z}\right) \leq \gamma_{\varepsilon} \end{cases}. \tag{4.35}$$

That is the ellipses, inscribed respectively by the covariances of the state estimate and the report, overlap. This motivates the use of intersections as a method for gating. Further, since the ellipsoids of the tracks and reports are independent, the gate equation in (4.34) does not need to be recomputed for each association pair.

To further simplify the gating, we find the axis-aligned bounding box of the ellipse in (4.33), yielding, for track φ and a given γ_ε,

$$\boldsymbol{B}^{\varphi}_{\gamma_\varepsilon} = \begin{bmatrix} p^{\varphi}_{x-}, & p^{\varphi}_{x+}, & p^{\varphi}_{y-}, & p^{\varphi}_{y+} \end{bmatrix}. \tag{4.36}$$

for minimum ($-$) and maximum ($+$) box corner positions in the x and y dimension, respectively, as defined in Lemma 4.1. Unless needed, the γ_ε index may be dropped.

Lemma 4.1 (Axis-Aligned Bounding Boxes). *For a two-dimensional Gaussian distribution $\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Pi}\right)$, the axis-aligned bounding-box for the ellipse inscribed by the iso-probability limit for an error ellipse limit of γ_ε is given by*

$$\boldsymbol{B}^{\varphi}_{\gamma_\varepsilon} = \begin{bmatrix} p^{\varphi}_{x-}, & p^{\varphi}_{x+}, & p^{\varphi}_{y-}, & p^{\varphi}_{y+} \end{bmatrix}. \tag{4.37}$$

where (for $d \in \{x, y\}$)

$$p_d = \mu_d \pm \sqrt{a_d^2 + b_d^2}. \tag{4.38a}$$

$$\boldsymbol{a} = \sqrt{\gamma_\varepsilon \lambda_1}\frac{e_1}{|e_1|} = \left[a_x,\, a_y\right]^\intercal, \qquad \boldsymbol{b} = \sqrt{\gamma_\varepsilon \lambda_2}\frac{e_2}{|e_2|} = \left[b_x,\, b_y\right]^\intercal \tag{4.38b}$$

for the eigenvalues $\{\lambda_1, \lambda_2\}$ and eigenvectors $\{e_1, e_2\}$ of the covariance matrix $\boldsymbol{\Pi}$ (Ribeiro, 2004).

Proof: Consider the parametric ellipse equation:

$$\boldsymbol{\varepsilon}\left(\omega\right) = \boldsymbol{\mu} + \boldsymbol{a}\cos\left(\omega\right) + \boldsymbol{b}\sin\left(\omega\right) \tag{4.39}$$

where ω is the angle parameter. The ellipse is related to the Gaussian distribution through the mean $\boldsymbol{\mu}$, and the vectors $\boldsymbol{a} = \left[a_x,\, a_y\right]^\intercal$, $\boldsymbol{b} = \left[b_x,\, b_y\right]^\intercal$ that describe the ellipse axes of symmetry — the eigenvectors of the covariance:

$$\boldsymbol{a} = \sqrt{\gamma_\varepsilon \lambda_1}\frac{e_1}{|e_1|}, \qquad \boldsymbol{b} = \sqrt{\gamma_\varepsilon \lambda_2}\frac{e_2}{|e_2|} \tag{4.40}$$

for the eigenvalues $\{\lambda_1, \lambda_2\}$ and eigenvectors $\{e_1, e_2\}$ of the covariance matrix $\boldsymbol{\Pi}$ (Ribeiro, 2004).

The axis-aligned bounding box of the ellipse is defined between the minimum and maximum extent in each dimension. Thus, setting the derivative of (4.39) to zero in each dimension provides the minimum/maximum in each respective dimension, below for the x dimension;

$$\begin{aligned} &-\sqrt{1-\cos^2(\omega)}a_x + \cos(\omega) b_x = 0 \implies \\ &\cos(\omega) = \frac{a_x}{\sqrt{a_x^2+b_x^2}}, \quad \sin(\omega) = \frac{b_x}{\sqrt{a_x^2+b_x^2}} \end{aligned} \tag{4.41}$$

Finally, insertion into (4.39) yields

$$p_x = \mu_x \pm \frac{a_x^2+b_x^2}{\sqrt{a_x^2+b_x^2}} = \mu_x \pm \sqrt{a_x^2+b_x^2} \tag{4.42}$$

(4.41)-(4.42) is repeated for the y dimension to find the values for the bounding box $\boldsymbol{B}$,

$$\boldsymbol{B}_{\gamma_\varepsilon} = \left[p_{x-}, \quad p_{x+}, \quad p_{y-}, \quad p_{y+}\right]. \tag{4.43}$$

□

Since the association probability at the square bounds of the ellipses in (4.35) is lower than or equal to that at the ellipsoid edge, the intersection of their bounding boxes is a conservative estimate of (4.35): if the bounding boxes do not overlap, the ellipses do not either.

By gating based on ellipse or bounding-box overlap instead of the Mahalanobis distance of the innovation, the bounds for each report and target can be precomputed. Thus, instead of creating — for $N_{\mathcal{Z}}$ reports and N_φ tracks — $N_{\mathcal{Z}} \times N_\varphi$ gates (including matrix inversions for each pair), the complexity is reduced to creating $N_{\mathcal{Z}} + N_\varphi$ gates, and performing a $\mathcal{O}\left(N_{\mathcal{Z}} \cdot \log N_\zeta\right)$ search for matching all reports to any of N_ζ clusters, given an efficient indexation of clusters, as described in Section 4.6.

The concept of overlap gating may be extended to other unimodal distributions, as well as mixtures and clusters by using the minimum bounding box of their respective components. If wanted, a finer, e.g. elliptic, gating can be performed in a second pass to potentially further reduce the considered associations.

For a cluster ζ, containing tracks φ, we define the axis-aligned bounding box, $\boldsymbol{B}_\zeta$ as the minimum axis-aligned box to contain the bounding boxes of all its tracks (see Figure 4.5),

$$\boldsymbol{B}^\zeta = [\min_{\varphi\in\zeta} p_{x-}^\varphi, \quad \max_{\varphi\in\zeta} p_{x+}^\varphi, \quad \min_{\varphi\in\zeta} p_{y-}^\varphi, \quad \max_{\varphi\in\zeta} p_{y+}^\varphi]. \tag{4.44}$$

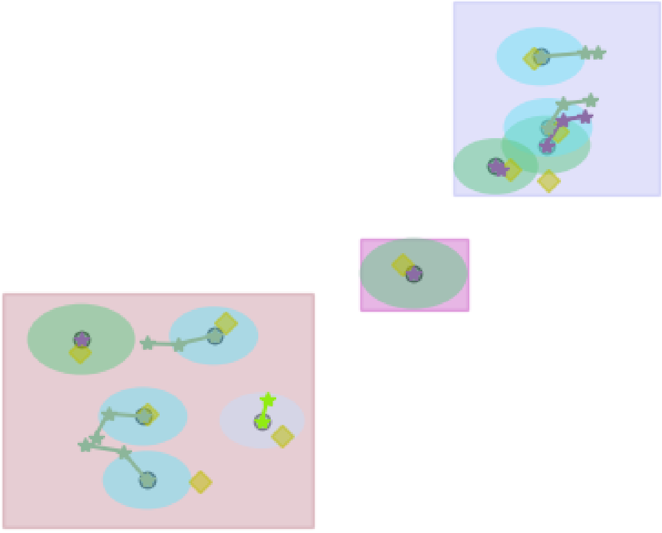


Figure 4.5: Cluster bounding boxes

Thus, a report bounding box may — but will not necessarily — intersect any of a cluster's tracks if and only if it also intersects the containing bounding box. Analogously, the same applies to single-target mixture distributions and their components.

4.5.2 Cluster Separation

From the matching of reports to tracks, a connection graph (visualized in Figure 4.6) can be formed by connecting pairs of targets which both are a potential match for a common report. The resulting graph will contain one or more groups of connected components which represent the targets that must be kept in the same cluster. Algorithms for finding graph connected components are studied in e.g. Tarjan (1972); Pearce (2005).

Hence, for each closed group of connected targets, the set of all targets

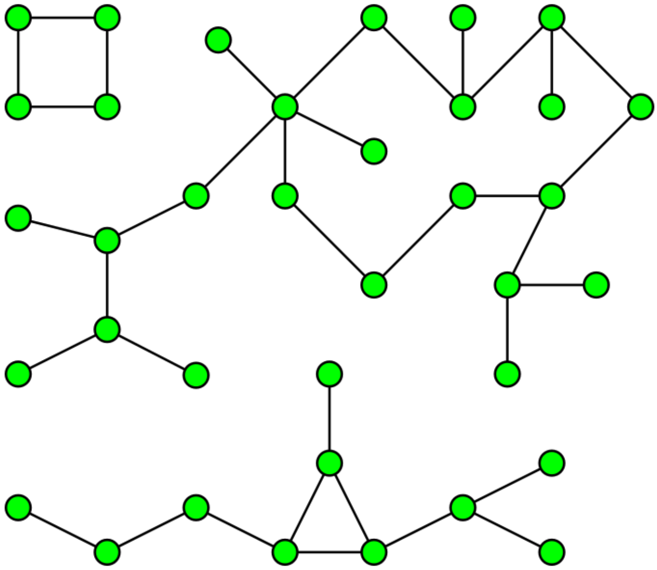


Figure 4.6: All nodes reachable from a node form its connected components. In clustering, a link represents a shared report between targets.

$\mathcal{L}_k$ hypothesized at time k is correspondingly partitioned into disjoint sets,

$$\mathcal{L}_k = \bigcup_{\zeta=1}^{N_\zeta} \mathcal{L}_k^{(\zeta)} \tag{4.45}$$

with $\mathcal{L}_k^{(\zeta)} \cap \mathcal{L}_k^{(m)} = \emptyset$ for $\zeta \neq m$. Similarly, the set of reports in the scan $\mathcal{Z}_k$ can be partitioned into the corresponding clusters

$$\mathcal{Z}_k = \mathcal{Z}_k^{(0)} \cup \bigcup_{\zeta=1}^{N_\zeta} \mathcal{Z}_k^{(\zeta)} \tag{4.46}$$

with $\mathcal{Z}_k^{(\zeta)} \cap \mathcal{Z}_k^{(m)} = \emptyset$ for $\zeta \neq m$, and $\mathcal{Z}_k^{(0)}$ is the set of measurements not associated with any previously known target.

A notable difference between the MHT filter in Chapter 5 and the LMB filter in Chapter 6 is that the targets in the LMB filter are not connected to each other through restrictions of track history compatibility — each target is assumed independent following each update, and consists of only a single merged track. This means that for an incoming scan it can easily be determined, on a target-level, which targets that fall in the sensor Field-of-View (FOV) and thus will be affected by the measurement update, simply by finding which of the target gates intersect with the FOV. Similarly, the possible associations between reports is a trivial matching process between the gate of each report and that of each target. Conversely, the clusters in MHT persist over time, adding complexity to the clustering process. Separate clusters whose targets come to share reports must be joined into a single larger cluster and ambiguities which are indubitably resolved can cause clusters to split into smaller ones. The smaller the cluster, the cheaper the updates.

Axis-aligned bounding boxes are well suited for fast intersection lookups, especially when stored in structures suitable for spatial indexing such as the R-TREE (Guttman, 1984). When in database storage, targets and clusters lie dormant, meaning that access to exact calculations — especially involving their distributions — are unavailable without loading them into memory, which is an expensive operation. However, select pre-computed numbers can be stored along the object in the database, accessible for database search. Overlap gating through intersection of pre-computed axis-aligned boxes requires only the four numbers in (4.36) and can thus be made available in the database lookup. This means that by storing the targets and clusters in a database with their bounding-boxes, the affected targets of each scan can efficiently be selected, leaving all other targets unloaded from the database.

4.6 Spatial Indexing for Matching and Storage

In global tracking of objects, observations are generally performed in bounded areas at a time, such as when a satellite image is processed, or an image from an Unmanned Aerial System (UAS) flight is processed. As the processed data contain no information about the area outside the boundary, it is warranted to ensure that the processing algorithms can be partitioned online to update only the relevant parts. This online partitioning is a natural match to spatial indexing — the organisation of spatial objects into data structures which can be efficiently queried for e.g. intersections and overlap. Common indexing techniques include

- K-Dimensional trees (KD-TREES) (Bentley, 1975),

- Local Split Decision trees (LSD-TREES) (Henrich et al., 1989),

- Rectangle trees (R-TREES) (Guttman, 1984; Sellis et al., 1987).

In this section, a method is motivated and described for the use of probability bounding boxes for screening of possible report–cluster associations, as well as for report–track gating.

4.6.1 Online Partitioning of MTT based on Field-Of-View

The ability of position-reporting sensors to observe targets is limited to targets within the sensors *FOV*. A sensor FOV is exemplified in Figure 4.7. We note the following:

Negative data Absence of reports is relevant to the filter. Thus, clusters within the FOV without report assignments need to be included in the update.

New clusters Because of the assumption that any target yield only a single report, reports with no associations to prior targets (in any hypothesis) will each form a new cluster, even if they would all overlap a potential, not yet detected, target.

Bordering clusters Tracks with a non-zero intersection with the FOV have a non-zero probability of being within the active area, and must be considered. This means their entire cluster should be updated. In such clusters, occlusion of tracks outside the FOV is obtained with track-individual probabilities of detection.

Outside information No information, negative or positive, is available on clusters outside the FOV. Thus, no measurement update should be performed on such clusters. This is essential to the scalability of the tracker.

Given the sensor's current FOV, we thus wish to extract from the filter only the clusters and/or targets which intersect with the FOV. For axis-aligned bounding boxes a and b, overlap between can be determined by the following logical expression (with $+-$ denoting the boxes' maximum and minimum in the x and y dimension respectively);

$$\begin{aligned}\text{overlap}(a,b) = \quad & (a_{x+} > b_{x-}) \wedge (a_{x-} < b_{x-}) \\ & \wedge (a_{y+} > b_{y-}) \wedge (a_{y-} < b_{y+}).\end{aligned} \tag{4.47}$$

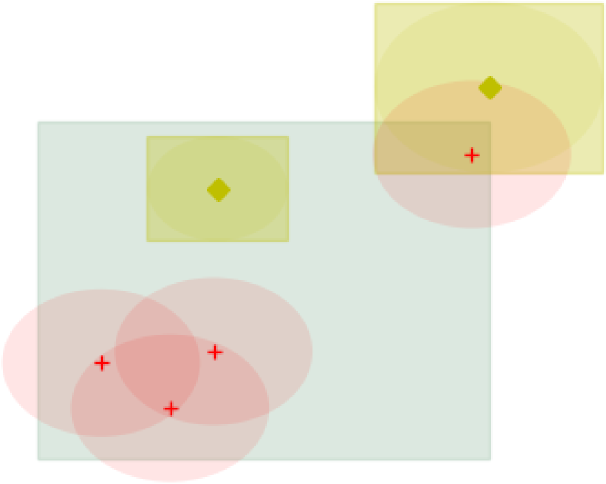


Figure 4.7: FOV example. Previous single-track clusters in yellow; reports in red; FOV in green. Note that the report in the upper right might originate from a target with a mean estimate outside the FOV, which must still be included in the update. Each of the three lower left reports will initiate separate new clusters.

A naïve method to exclude clusters and tracks outside the sensor FOV would be to perform the test of (4.47) on all clusters and tracks in the filter, for each report. However, as we — at least for coarse selection — approximate FOVs and bounding boxes as axis-aligned, intersections can be efficiently computed trough spatial indexing of the cluster and track storage.

4.6.2 Spatial Indexing with R-Trees

The R-TREE, short for rectangle tree, was introduced by Guttman (1984), and has formed the basis for several extensions. One of those is the R*-TREE,

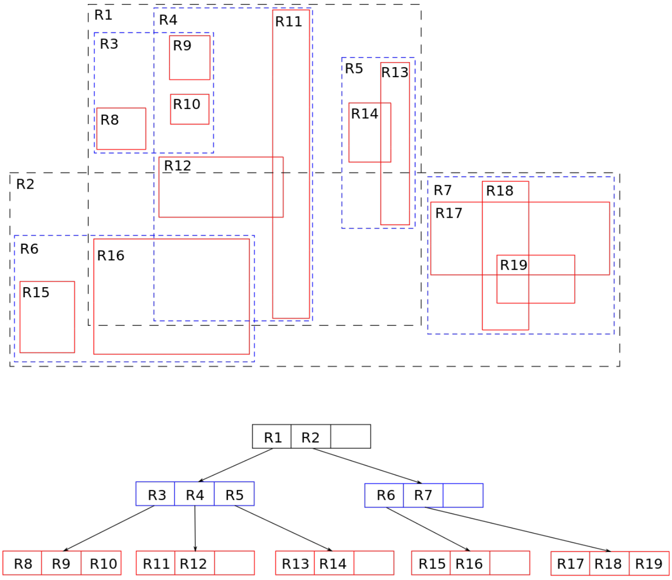


Figure 4.8: R-TREE example. Objects are stored in the leaf nodes R8–R19, and indexed using the non-leaf entries R1–R7.

which is used e.g. in the SQLITE database[2].

The basic idea of R-TREES is exemplified in Figure 4.8, where the objects are stored in the records R8–R19 of the tree. The R-TREE satisfies the following properties (Guttman, 1984):

1. All leaves contain between m and M records unless it is the root.

2. All non-leaf nodes contain between m and M children, unless it is the root.

3. The root node has at least two children, unless it is a leaf.

4. A leaf record contains the minimum bounding box of the object, and a pointer to the stored object.

5. A non-leaf entry contains the smallest rectangle which contains all rectangles of the child node, and a pointer to said child.

6. All leaves are on the same level.

[2] https://www.sqlite.org/rtree.html

When a node becomes under-/over-full during deletion/insertion, the tree is rebalanced (Guttman, 1984) to maintain the properties above. Different versions differ in the choice of heuristics in the choice of new nodes when rebalancing the tree (Beckmann et al., 1990). Specialized implementations also exists e.g. for indexing based on geodetic distance (Schubert et al., 2013).

Searching the tree for intersections is then performed recursively (Guttman, 1984), recursing only into nodes whose bounding box intersect with the query. Searching an R-TREE can be optimized to reach $\mathcal{O}\left(\log N\right)$ complexity (Göbel, 2007) for N nodes, reducing the search for report–cluster matches to a potential $\mathcal{O}\left(N_{\mathcal{Z}} \log N_{\zeta}\right)$ or report–track matches to $\mathcal{O}\left(N_{\mathcal{Z}} \log N_{\varphi}\right)$. Combinations of indexing is also possible, e.g. through separate indexation of clusters and tracks.

4.7 Coordinate Systems

To facilitate storage and handling of ice objects on a global scale, multiple coordinate frames of reference are used. Primarily, three coordinate systems are used in this thesis and its associated implementations:

- Latitude-Longitude-Altitude (LLA) or, in 2D, Latitude-Longitude (LL);
- Earth-Centered, Earth-Fixed (ECEF); and
- North-East-Down (NED), or in 2D, North-East (NE).

The above coordinate systems are visualized in Figure 4.9 (with "up" being shown as the negative of "down" for clarity).

4.7.1 Transformations

Coordinates expressed in one coordinate system can be — linearly or nonlinearly — transformed to another via transformation functions (from frame a to frame b):

$$\boldsymbol{x}^b = f^{ba}\left(\boldsymbol{x}^a\right).$$

The transformation between ECEF and NED (and the reverse) is an affine transformation — a translation and a rotation. To move to LLA (or LL, by additionally discarding the altitude), the transform is nonlinear however,

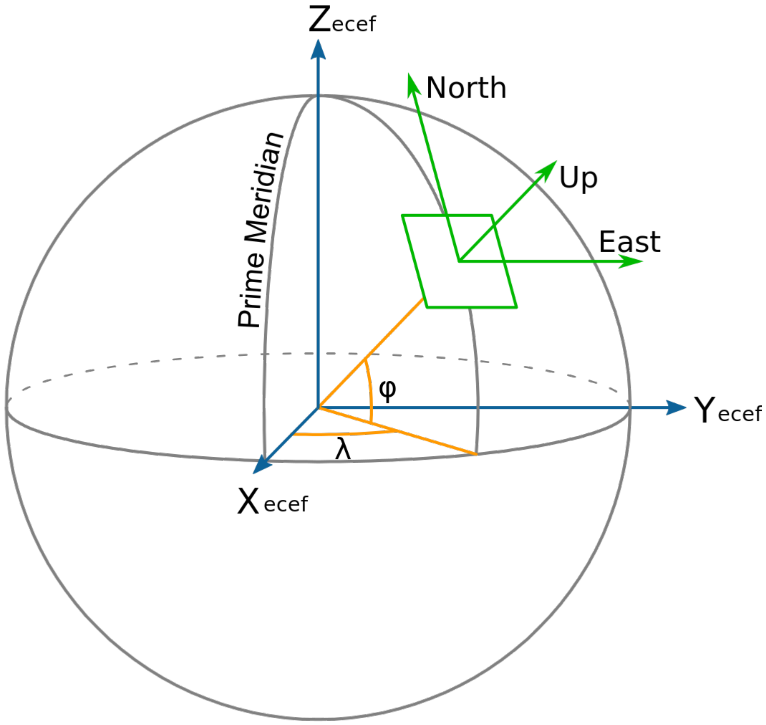


Figure 4.9: Coordinate frames used in the thesis. “down” in the NED system is the negative of the shown “up”; ϕ and λ is the latitude and longitude, respectively.

and poses a larger problem. The transformation from LLA to ECEF follows trigonometric equations outlined in Hofmann-Wellenhof et al. (1997), whereas the more intricate ECEF-to-LLA transform has been a subject for optimization research, but can be performed e.g. as per Olson (1996). Transformations between NED and LLA is performed via the ECEF frame, forming the chain LL↔LLA↔ECEF↔NED↔NE.

4.7.2 Storage and Updates

In implementations presented in this thesis, the targets and/or clusters are stored in LL coordinates. This, as can be seen in Figure 4.11, is not an Euclidean system, making it a less suitable coordinate frame for handling tracking. Instead, all measurement and prediction updates are performed in a local Euclidean NE coordinate frame. This is made possible by the

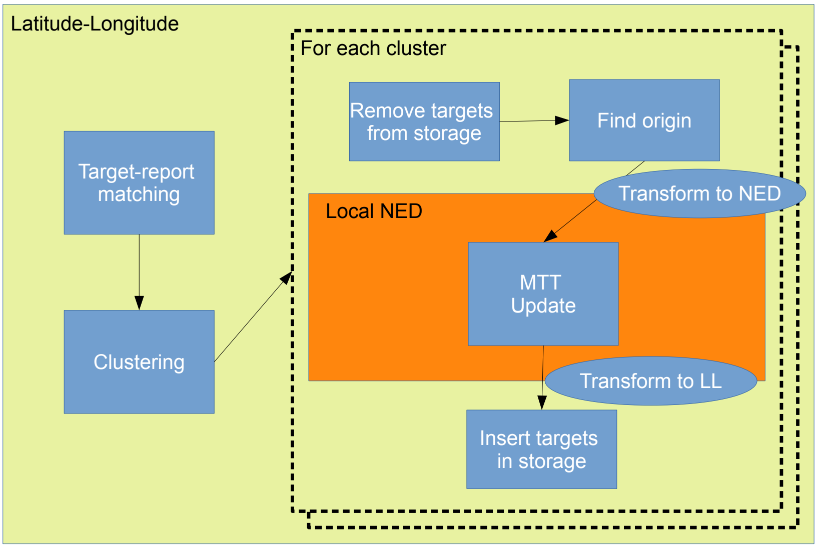


Figure 4.10: Cluster coordinate transformations

process described in Figure 4.10. The transformation sequence begins when reports — in the LL coordinate frame — are received and matched against existing targets, also in the LL frame as they are stored. Having extracted the matches, the clusters are formed and can be treated independently and in parallel. The cluster measurement update starts by finding a suitable origin, $\boldsymbol{x}_o$ around which to perform the transform to Euclidean space. Relevant choices include the center of the region, and — as used here — the cluster's center of mass,

$$\boldsymbol{x}_o = \sum_{\ell \in \mathcal{L}^{(\zeta)}} \omega_\ell \mathbb{E}\left[\boldsymbol{x}_\ell\right]. \tag{4.48}$$

Having determined a suitable origin, the clusters' reports and targets are transformed to a NE frame fixed there, and a standard MTT measurement update can be performed in Euclidean space. The prediction update is performed analogously, although for algorithms where the targets are considered independent between updates — such as the LMB filter — each target forms their own cluster, making the search for an origin trivial.

Spatially indexed objects that move must be handled appropriately, as

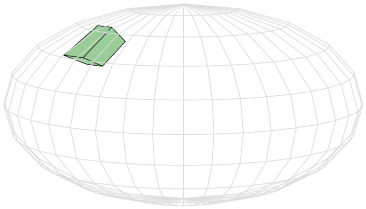


Figure 4.11: Global storage of targets is done in LL coordinates, indexed as rectangular sections in the LL frame.

their search index will change after the update. A common way to solve this is to remove the object from the storage data structure prior to update, and the re-insert it after the update has been performed.

After the update, the targets are returned to the LL frame and subsequently stored in the database, until the next update.

4.8 Conclusions

This chapter introduced the essential concepts which are used in the continued thesis to explore the field of MTT. This started with a basic overview of STT, which is still an essential base when the problem formulation is extended to the MTT case. In the problems associated with MTT, particular focus was given to the association (in Section 4.4) and clustering (in Section 4.5) which forms the bases for some of the algorithm improvements studied later in this part. These are inherently difficult problems to which many solutions have been proposed and discussed, yet still allows for further optimizations. Often, the solution to these problems are implementation specific, and is dependent e.g. on the exact definitions used of the terms assignments, hypotheses, targets, tracks and clusters. One take on the optimization of clustering is presented in Chapter 5 through the combination of the MHT algorithm and the R-TREE spatial indexing technique.

Hypothesis-based MTT is further put to use in Chapter 6, where a novel reformulation is proposed to the LMB algorithm, in which the results of the

Murty algorithm is mapped into a matrix representation to simplify the implementation of the algorithm.

Further, FISST, RFS's and the PHD were introduced in Section 4.3, as they form not only the foundation for the LMB filter of Chapter 6, but also further lends their applicability, as proposed in Chapter 8, to the formation of flight plans for controllable UAS agents.

5 Spatially Indexed Multiple Hypothesis Tracking

In this chapter, we explore the Multiple Hypothesis Tracker (MHT) algorithm, and describe the implementation of a tracker with spatially indexed cluster association, in efforts to apply it to the large-scale application of sea ice tracking, as performed by the tracking module in the sytem overview of Figure 5.1. The contents of this chapter were first published in

Paper A: Olofsson, J., Brekke, E., Fossen, T. I., and Johansen, T. A. (2017a). Spatially indexed clustering for scalable tracking of remotely sensed drift ice. In *IEEE Aerospace Conference Proceedings*, Big Sky, MT, USA. © 2017 IEEE.

The contributions of this chapter include:

- a detailed description of a Free and Open Source Software (FOSS) MHT implementation which uses spatial indexing;
- a comparative study of the scalability improvements from spatial indexing; and
- an application of the algorithm for sea ice tracking, using real Synthetic Aperture Radar (SAR) imagery with added simulated motion.

For operations in the Arctic, drift ice can be a major hazard. To be able to mitigate this, it is essential to know the position of viable threats. Many sensors can be employed, such as satellite SAR, marine radar, air surveillance et cetera. At the core of the fusion of this sensor data is a Multiple Target Tracking (MTT) problem. This problem is studied in this chapter through the implementation and application of the MHT algorithm.

A major limiting factor of the applicability of MHT is scalability. A common method of handling the scaling is clustering, which separates the MHT filter into smaller independent parts. However, with growing scale, the association of sensor data to the "right" cluster can become resource intensive in itself. A method is explored, based on rectangular lower probability bounds, to efficiently index the clusters and compartmentalize the measurement update of the MHT. The method uses the bounding box of a lower probability bound of tracks and reports respectively, to

perform an intersection lookup against the sensor Field-of-View (FOV), efficiently selecting clusters of relevance.

The method, as well as the MHT algorithm, has been implemented and published online under an open-source license. In this chapter, the implementation is described and tested on simulated data for statistics. Further, it is tested against data extracted from the polarimetric classification of ice using satellite imagery of the Arctic. Results show that computational speed improvements can be achieved compared to the linear complexity of a naïve search or, to a lesser extent, standard database lookups.

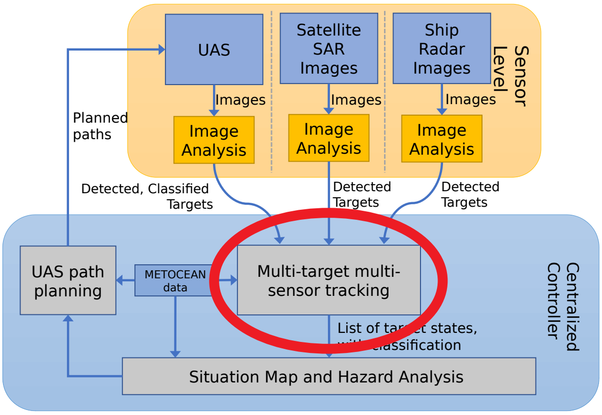


Figure 5.1: Multiple Target Tracking in system context

5.1 Introduction

The MHT tracker, as the name implies, makes heavy use of the concept of hypotheses. The background on this was presented in Section 4.4. There too the concept of gating and clustering was introduced, the foundation for the algorithm improvements we explore in this chapter.

The MHT algorithm was first introduced by Reid (1979), and many variants and improvements have been proposed since. It is drawn from the intuition that the MTT problem is effectively reduced to multiple single-target tracking problems, given a perfect association of each measurement to the two cases of either

- an existing target track; or
- a new target/a false measurement (both results from extraneous reports).

Recognizing that perfect associations are idealistic, multiple hypothetical associations are kept and maintained to defer the decision on which hypothesis is "best" until more data is available. However, the number of possible track associations increases exponentially with time, posing a fundamental limitation which must be overcome with approximative implementations.

The original MHT formulation was based on exhaustive enumeration of possible measurement associations, subjected to extensive pruning to retain only the most relevant association hypotheses. This has been the blueprint for multiple variants (Blackman, 2004), each of which has its individual approach to limit the otherwise exponential growth of possible hypotheses.

The Track-Oriented Multiple Hypothesis Tracker (TOMHT), introduced in Kurien (1990), contrasted the hypothesis-oriented approach in Reid (1979) by replacing the exponentially growing *hypothesis relation matrix* with target tree graphs to represent the different associations. This was also further improved by Cox and Hingorani (1996), who introduced the application of Murty's algorithm (Murty, 1968) for efficient generation of relevant hypotheses, providing a way to limit the number of generated hypotheses.

Generally, the main difference between variants of the MHT algorithm are their approach to bookkeeping of hypotheses and the sampling of new hypotheses. Another distinction can be made between batch sampling implementations, such as the m-best S-D Assignment proposed in Popp et al. (2001), and recursive implementations which retain and build upon the hypotheses from previous iterations, e.g. Reid (1979); Kurien (1990). The implementation described and used in this chapter uses recursive generation of hypotheses and track-oriented bookkeeping.

In many applications, the MHT algorithm suffers from scalability issues (Blackman, 2004; Reid, 1979), though this can sometimes be mitigated e.g. through the use of clustering (Roy et al., 1997), which subdivides the filter into clusters which can be updated independently. Clustering is generally coupled with the use of *gating*, by which a limit of association probability is used to limit the amount of considered associations and keep the number of targets in each cluster down. Through clustering and gating, it is possible to limit an observation's impact on the full filter to a local region around the sensor field of view, and determine for each cluster if it will need to be updated or not.

The application of spatial indexing to MTT was studied by Collins and Uhlmann (1992). Specifically, report–target associations were considered, and its application exemplified using the Joint Probabilistic Data Association (JPDA) filter. Their study concludes that spatial indexing offers an efficient way of coarsely selecting possible report–target associations. In this context, this chapter focuses on the application of spatial indexing to the MHT algorithm, where instead report–track and report–cluster assignments are considered. Specifically, we apply spatial indexing on the report–cluster assignment problem, to reduce a global coverage filter to a local subfilter.

The MHT algorithm is described in Section 5.2. The implementation is discussed in Section 5.3, and the obtained results presented in Section 5.4.

5.2 The Multiple Hypothesis Tracker

The core of the MHT algorithm is the likelihood of each hypothesis, first derived in Reid (1979):

$$\begin{aligned} p\left(\Theta^k_\vartheta | \mathcal{Z}_{1:k}\right) &= p\left(\theta^k_\vartheta, \Theta^{k-1}_{\Omega(\vartheta)} | \mathcal{Z}_k, \mathcal{Z}_{1:k-1}\right) \\ &\propto p\left(\mathcal{Z}_k | \theta^k_\vartheta, \Theta^{k-1}_{\Omega(\vartheta)}, \mathcal{Z}_{1:k-1}\right) p\left(\theta^k_\vartheta | \Theta^{k-1}_{\Omega(\vartheta)}, \mathcal{Z}_{1:k-1}\right) p\left(\Theta^{k-1}_{\Omega(\vartheta)} | \mathcal{Z}_{1:k-1}\right). \end{aligned} \tag{5.1}$$

In this expression, the last term is the likelihood of the parent hypothesis from the previous timestep, the value of which can be expressed as

$$p\left(\Theta^{k-1}_{\Omega(\vartheta)} | \mathcal{Z}_{1:k-1}\right) = e^{-\Lambda_{\Omega(\vartheta)}} \tag{5.2}$$

using the parent hypothesis' Negative Log Likelihood (NLL) score $\Lambda_{\Omega(\vartheta)}$ — $\Omega(\vartheta)$ again being the mapping to the parent hypothesis of hypothesis ϑ of time k. The two remaining terms in (5.1) represent the likelihood of the new assignment, which can be rewritten as follows, assuming Poisson-distributed (with parameter λ_ξ) extraneous reports,

$$
\begin{aligned}
p\left(\mathcal{Z}_k|\theta^k_\vartheta, \Theta^{k-1}_{\Omega(\vartheta)}, \mathcal{Z}_{1:k-1}\right) p\left(\theta^k_\vartheta|\Theta^{k-1}_{\Omega(\vartheta)}, \mathcal{Z}_{1:k-1}\right)& \\
\propto \prod_{z\in\mathcal{Z}_k} \left(\boldsymbol{\iota}_{\varphi(z)}\left(\boldsymbol{z}\right)\right)^{a_z}\left(\lambda_\xi\right)^{1-a_z} \cdot \prod_{\varphi\in\theta^{k-1}_{\Omega(\vartheta)}} \left(p^\varphi_D\right)^{a_\varphi}\left(1-p^\varphi_D\right)^{1-a_\varphi}& \\
= \prod_{\varphi\in\theta^k_l} \left(\boldsymbol{\iota}_\varphi\left(\boldsymbol{z}_\varphi\right)p^\varphi_D\right)^{a_\varphi}\left(1-p^\varphi_D\right)^{m_\varphi}\left(\lambda_\xi\right)^{\xi_\varphi}&
\end{aligned}
\tag{5.3}
$$

where a_φ, m_φ and ξ_φ are mutually exclusive indicator functions for assigned, missed and new tracks (from potential new targets) φ respectively, as assigned by assignment θ^k_ϑ. Analogously, a_z is the indicator function for assigned measurements. p^φ_D is the probability of detection for track φ, whereas $\boldsymbol{\iota}_\varphi\left(\boldsymbol{z}_\varphi\right)$ is the innovation probability of the association between track φ and its assigned report $\boldsymbol{z}_\varphi$. The factors in (5.3) can, too, can be expressed using the NLL, Λ_φ,

$$
e^{-\Lambda^k_\varphi} \triangleq \begin{cases} \boldsymbol{\iota}_\varphi\left(\boldsymbol{z}_\varphi\right)p^\varphi_D & \text{for associated tracks} \\ 1-p^\varphi_D & \text{for missed tracks} \\ \lambda_\xi & \text{for new tracks} \end{cases}
\tag{5.4}
$$

Unless beneficial in the context, the time index k is considered implicit below and dropped. Combining (5.1)–(5.3) yields that

$$
p\left(\Theta^k_\vartheta|\mathcal{Z}_{1:k}\right) \propto e^{-\Lambda_{\Theta^{k-1}_{\Omega(\vartheta)}}} \prod_{\varphi\in\theta^k_l} e^{-\Lambda_\varphi}.
\tag{5.5}
$$

Conveniently, this product is easily converted to a sum by using the (negative) logarithm. This results in a sum over the scores, Λ_φ, that each track head in the hypothesis individually contributes,

$$
\Lambda_{\Theta^k_\vartheta} = \Lambda_{\Theta^{k-1}_{\Omega(\vartheta)}} + \Lambda_{\theta^k_\vartheta} = \Lambda_{\Theta^{k-1}_{\Omega(\vartheta)}} + \sum_{\varphi\in\theta^k_\vartheta} \Lambda_\varphi.
\tag{5.6}
$$

The total score of a track can be calculated recursively from the assignment scores given at each historical assignment to the track. Hence, (5.6) can be calculated as the sum of the total track score of the hypothesis' tracks:

$$
\Lambda_{\Theta^k_\vartheta} = \sum_{\varphi\in\Theta^k_\vartheta} \left[\Lambda^k_\varphi + \Lambda^{1:k-1}_\varphi\right]
\tag{5.7}
$$

for $\Lambda_{\varphi}^{1:k-1} = \sum_{k'=1}^{k-1} \Lambda_{\varphi}^{k'}$ over the track history. Hence, in an implementation, each track need only maintain the running sum and its own score to enable the calculation of the score of a hypothesis containing a given set of tracks.

An equivalent formulation of (5.3), useful in the formulation of the Linear Assignment Problem (LAP) is

$$\begin{aligned}
&\prod_{\varphi\in\theta_l^k} \left(\boldsymbol{\iota}_{\varphi}\left(\boldsymbol{z}_{\varphi}\right) p_D^{\varphi}\right)^{a_{\varphi}} \left(1-p_D^{\varphi}\right)^{m_{\varphi}} \left(\lambda_{\xi}\right)^{\xi_{\varphi}} \\
&= \prod_{\varphi\in\theta^{k-1}_{\Omega(\vartheta)}} \left(1-p_D^{\varphi}\right) \prod_{\varphi\in\theta_l^k} \left(\frac{\boldsymbol{\iota}_{T_i}\left(\boldsymbol{z}_i\right) p_D^{\varphi}}{\left(1-p_D^{\varphi}\right)}\right)^{a_{\varphi}} \left(\lambda_{\xi}\right)^{1-a_{\varphi}}.
\end{aligned} \tag{5.8}$$

which yields the NLL sum

$$\Lambda_{\Theta_{\vartheta}^k} = \Lambda_{\Theta^{k-1}_{\Omega(\vartheta)}} + \sum_{\varphi\in\theta^{k-1}_{\Omega(\vartheta)}} {}^{m}\Lambda_{\varphi} + \sum_{\varphi\in\theta_{\vartheta}^k} \bar{\Lambda}_{\varphi} \tag{5.9}$$

with the following definitions:

$$\begin{aligned}
{}^{z}\Lambda_{\varphi} &= -\ln\left(\boldsymbol{\iota}_{\varphi}\left(\boldsymbol{z}\right) p_D^{\varphi}\right) \\
{}^{m}\Lambda_{\varphi} &= -\ln\left(1-p_D^{\varphi}\right) \\
\Lambda_{\xi} &= -\ln\left(\lambda_{\xi}\right) \\
\bar{\Lambda}_{\varphi} &= \begin{cases} {}^{z}\Lambda_{\varphi} - {}^{m}\Lambda_{\varphi} & \text{for associated (not missed) tracks} \\ \Lambda_{\xi} & \text{for extraneous tracks} \end{cases} .
\end{aligned}$$

To generate hypotheses with this formulation of the MHT equations, a cost matrix is created for each parent hypothesis in the cluster (Danchick and Newnam, 2006). This uses the NLLs of the track assignments outlined in (5.9), resulting in the matrix exemplified in (5.10) for a two-track hypothesis and a two-report scan:

$$\boldsymbol{C} = \begin{pmatrix} {}^{z_1}\Lambda_{\varphi_1} - {}^{m}\Lambda_{\varphi_1} & {}^{z_1}\Lambda_{\varphi_2} - {}^{m}\Lambda_{\varphi_2} & {}^{z_1}\Lambda_{\xi} & \infty \\ {}^{z_2}\Lambda_{\varphi_1} - {}^{m}\Lambda_{\varphi_1} & {}^{z_2}\Lambda_{\varphi_2} - {}^{m}\Lambda_{\varphi_2} & \infty & {}^{z_2}\Lambda_{\xi}. \end{pmatrix} \tag{5.10}$$

In the Gaussian case of an underlying Kalman filter Single-Target Tracking (STT), the NLL of the innovation probability density $\boldsymbol{\iota}_{\varphi}\left(\boldsymbol{z}\right)$ is calculated through the Gaussian equation of (4.11):

$$-\ln\left(\boldsymbol{\iota}_{\varphi}\left(\boldsymbol{z}\right)\right) = \frac{1}{2}\left(\boldsymbol{z}-\hat{\boldsymbol{z}}_{\varphi}\right)^{\intercal} S^{-1}\left(\boldsymbol{z}-\hat{\boldsymbol{z}}_{\varphi}\right) + \ln\left|2\pi S\right|. \tag{5.11}$$

5.2.1 Filter Output Presentation

As discussed in Crouse et al. (2011), there are several ways to present the data from an MHT tracker, including e.g. weighted averages of the hypotheses (Bottlik and Blackman, 1989), and the most likely hypothesis:

$$\Theta^k_{\mathrm{ml}} = \underset{\Theta^k_\vartheta \in \Theta^k}{\arg\max}\, p\left(\Theta^k_\vartheta | \mathcal{Z}_{1:k}\right) \tag{5.12a}$$

$$p_{\mathrm{ml}}\left(\mathcal{X}_k | \mathcal{Z}_{1:k}\right) \triangleq p\left(\mathcal{X}_k\left(\Theta^k_{ml}\right) | \Theta^k_{ml}, \mathcal{Z}_{1:k}\right). \tag{5.12b}$$

where $\mathcal{X}_k(\Theta)$ are the states of the targets given hypothesis Θ at time k.

While Crouse et al. (2011) proposes the application of improved methods of presentation, (5.12) remains the simplest and was thus selected as the first to be implemented in the implementation described in Section 5.3.

5.3 Implementation

The MHT algorithm presented in this chapter has been implemented in the PYTHON programming language. The implementation has been made public under a FOSS license at https://github.com/jonatanolofsson/mht.

Much of the complexity of an MHT implementation stems from the bookkeeping necessary to handle tracks, hypotheses, targets etc. Considerations include, but are not limited to,

Track deletion A track can be selected for removal from a hypothesis based on multiple grounds. Common choices are due to high track cost (Blackman, 2004) or repeated missed detections (Kim et al., 2015);

Target deletion As soon as a target has no remaining track in any hypothesis, a target can be removed;

Cluster deletion A cluster with no targets left can be removed;

Cluster management In the MHT algorithm, clusters persist over time, and they need to be merged, split, and maintained, as described in Section 5.3.1.

A few high-level details regarding the implementation;

Object hierarchy The implementation is object oriented, preserving the natural hierarchy of a *filter with clusters with targets with tracks*. Clusters also have hypotheses which are linked to the tracks of the targets. Thus, equivalent tracks are created only once and not for each hypothesis.

Track representation Each track is represented as an object. Since only tracks from the last time step are included in the active hypotheses, old tracks can be discarded as their relevant statistics are included in the new track heads, assuming Markovian tracking.

Hypothesis representation The history of hypotheses are contained in the tracks to which they refer. thus, once a hypothesis has been used to generate new hypotheses, it can be discarded. Hence, it is only necessary to store old hypotheses for a single time step.

Cluster management After the cluster splitting process, described in Section 4.5, clusters are stored in an R*-TREE indexed SQLITE database, away from the active algorithm. As a new scan comes in, the affected clusters are selected and loaded into memory.

Cluster independence Since unaffected clusters are entirely left out of the update, all parameters need to be specified on a per-cluster basis. For example, there is no limit to the global number of hypotheses, only to the number of hypotheses per cluster.

5.3.1 Cluster Management

Clustering as an addition to MHT was proposed already in the original MHT paper by Reid (1979), and has been subsequently extended with improved clustering algorithms, e.g. in Kurien (1990). As new scans arrive, new ambiguities may arise which makes previously independent clusters to have to be merged. Conversely, as ambiguities are resolved over time, clusters need to be split to maintain the benefits of clustering. Below are discussed some of the intricacies involved in these operations.

Cluster Merging

As a report may only be associated with a single cluster, any reports that are potentially associated with tracks in different clusters will trigger a merge of those clusters. The gating criterion described above is used to limit the associations, as hypothetically any report could be associated with any track. Since the number of hypotheses in the merged cluster is the product of the number of hypotheses in each included cluster, it may be necessary to select the N_h best merged hypotheses to maintain a tractable number of hypotheses.

Cluster Splitting

The implementation developed for this thesis, exploits that this problem can be formulated as a graph search for *connected components* (Tarjan, 1972; Pearce, 2005).

For each target-ambiguous report the resulting (incompatible) tracks are grouped in a set, which is added to a list of all target-ambiguous assignments. As new scans arrive and new hypotheses form, each set of tracks in the list is replaced by the joint set of the children of the set's tracks, as these too are incompatible. When all tracks of the set belong to the same target, i.e. all hypotheses agree on a single report–target association, the original ambiguity has been resolved and the set can be removed from the list.

Each hypothesis' set of tracks can be transformed into the set of targets which are associated with the respective track. By connecting all targets which appear together in those sets, a graph can be formed which connects all targets which, directly or indirectly, are connected through ambiguities. The resulting graph will contain one or more groups of connected components which represent the targets which thus must be kept in the same cluster. If the resulting graph contains more than one such group, the cluster can be split for each of these.

5.3.2 Report–cluster Association

In the aforementioned association problem, incoming reports are sought to be associated with pre-existing tracks, or considered extraneous. Since the scan may contain reports from several different clusters, reports may first be associated with clusters to maintain the separability of cluster updates. Association gates can, as described in Section 4.6, be used to uniquely assign each report to either *i*) a pre-existing cluster (possibly merged from multiple ambiguous clusters); or *ii*) a new cluster, if a report falls outside the gates of all existing clusters.

To investigate different report–cluster association schemes, the following three were implemented:

Naïve All clusters are loaded into memory, one by one, and their intersection with the FOV is calculated in the PYTHON-code.

Database Cluster bounding-boxes are pre-stored in the database and their intersections with the sensor FOV is calculated during the database lookup, without loading objects into memory.

R-tree Similarly to the database matching scheme, cluster bounding boxes are stored in the database, but using an Rectangle tree (R-TREE) implementation for potentially improved lookup-speeds.

5.4 Results

Two studies were performed using the Spatially Indexed MHT implementation. In Section 5.4.1, a performance comparison is carried out for the properties of the filter that benefits from a spatially indexed database for cluster storage. Second, the tracker was applied to satellite sea ice detections, extracted from the SAR sensor of the SENTINEL 1A satellite from the European Space Agency (ESA).

5.4.1 Spatial Indexing Performance

To verify the performance gain from spatial indexing, a test case was set up with a uniform random field of objects over a large area of which only a fraction was observed. The setup is displayed in Figure 5.2. Timing analysis was then performed, comparing naïve cluster filtering to that achievable using an R-TREE database. In the analysis, the MHT filter was initialized with a given size of random clusters, and the process of finding clusters intersecting with the sensor FOV was isolated and timed. The timing results are presented in Figure 5.3. Elaborating on the last data point; 39810 intersections were queried, with a comparative timing improvement of 841%. The timing of the R-TREE association method was also compared to that of a standard database lookup, see Figure 5.4, where the bounding boxes of the clusters were stored in an SQLITE database for faster lookup. Notably, the performance gain of the R-TREE algorithm is far more modest, mainly due to the reduced cost of loading clusters into memory. Note that

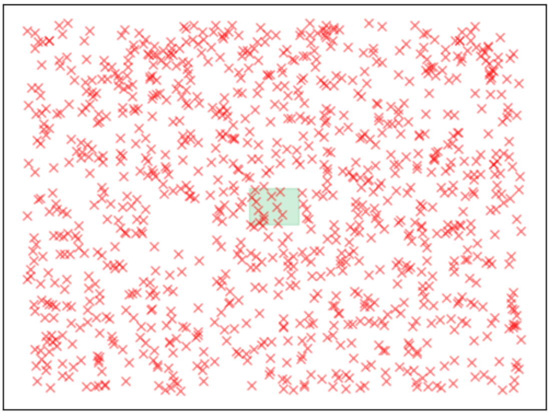


Figure 5.2: Setup for spatial indexing time analysis. The sensor FOV (green) was queried for intersection with clusters (red) for possible associations.

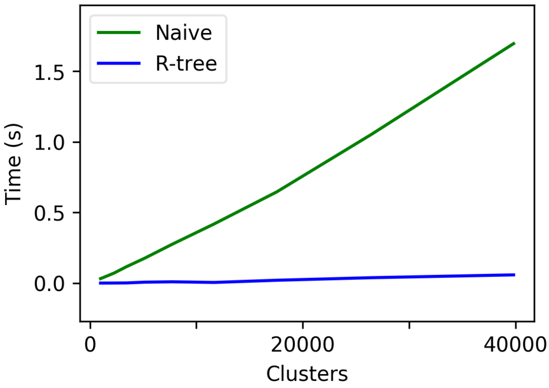


Figure 5.3: Compared results between naïve and R-TREE cluster matching

this latter comparison is between two methods using pre-computed square bounds, so significant differences are not to be expected except for very large datasets.

5.4.2 SAR Sea Ice Tracking

The algorithm was also evaluated on sample data extracted from the SAR image in Figure 3.7, a HH+HV[1] polarized image from the SENTINEL 1A satellite. As described in Sections 3.4.1 and 3.5.1, polarimetric classifi-

[1] Horizontally polarized illumination, horizontal and vertical reception.

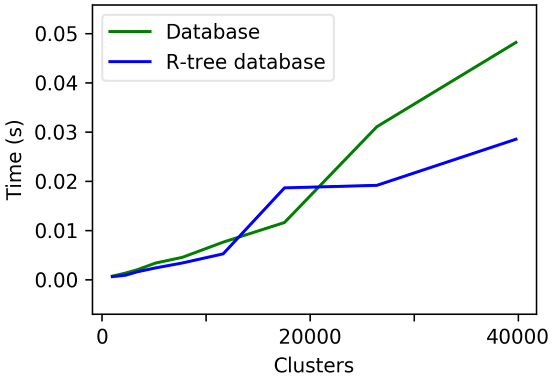


Figure 5.4: Compared results between database and R-TREE cluster matching

Table 5.1: MHT tuning parameters

Position prediction covariance:	$1 \times 10^{-5\circ}$
Initial velocity covariance:	1e−5 1×10^{-5} °/time
Report disturbance:	$1 \times 10^{-5\circ}$
Sensor noise:	$1 \times 10^{-5\circ}$
Max number of hypotheses / cluster:	100
Minimum normalized hypothesis score:	4

cation (Richards, 2009; Haykin, 1994) was used to extract and position individual ice objects from the image, see Figure 5.5. The estimated positions, in degrees latitude/longitude, were used to initialize the filter, and augmented with random velocities ($\dot{x} \sim \mathcal{N}(\mathbf{0}, 1\mathrm{e}{-6} \cdot \boldsymbol{I}_2)$ °/time) to generate further data. Subsets of the data were extracted to simulated sensor FOVS and used to study the performance for larger datasets. Further, an additional 400000 clusters were instantiated outside the sensor FOV to demonstrate the effect of spatial indexing. No clutter or target death was included. The globally most likely set of tracks are visualized in Figure 5.6. Tuning parameters include those summarized in Table 5.1.

Tracker performance is summarized in Figure 5.7, showing

a) Number of clusters in the tracker over time, before and after cluster splitting.

b) Estimated, and true, number of visible targets.

c) Average and maximum number of targets per cluster.

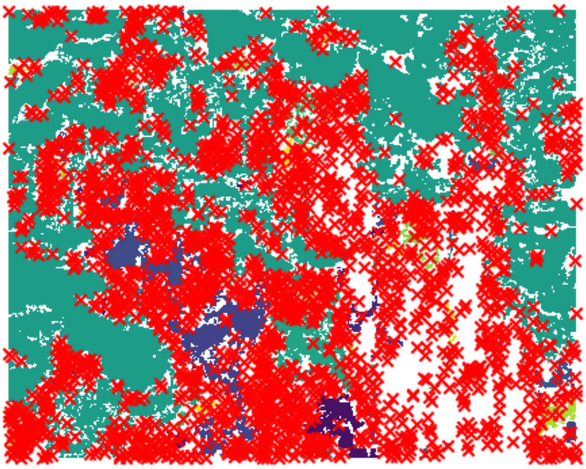


Figure 5.5: Individual ice objects were extracted from polarimetric satellite imagery. Each object (most of which are too small to see) is plotted with distinct colors, whereas red dots indicate the centerpoint of each object.

d) Average, and maximum number of hypotheses per cluster. This is limited by the score limit on included hypotheses.

e) Average number of true sources per track in the most likely hypothesis. This is a measure of the quality of the hypothesis' level of correct associations. In the case that all targets are visible in each timestep, and if all reports in all tracks come from the same source, this will have a value of one (and all associations will be correct). A value of two indicate that tracks have, on average, one misassociation.

f) Average track length. If all true targets remain alive and visible, this will increase linearly with time, as all tracks have a history back to the first timestep.

g) Step time in seconds. This is the total amount of time spent to generate associations and hypotheses for the scan(s) received at each time step.

h) The step time improvement achieved using R-TREE versus linear database search. Over time, there is a performance increase of about 4 % by using the R-TREE index over the database search.

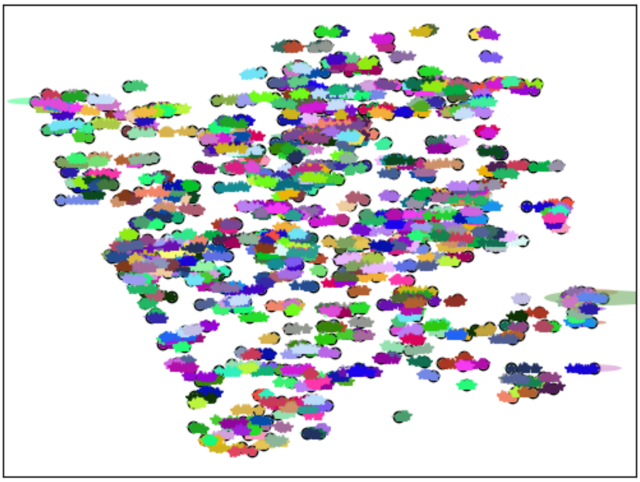


Figure 5.6: Most likely tracks of tracked ice objects. Ice tracking may result in dense object clouds which are difficult to cluster.

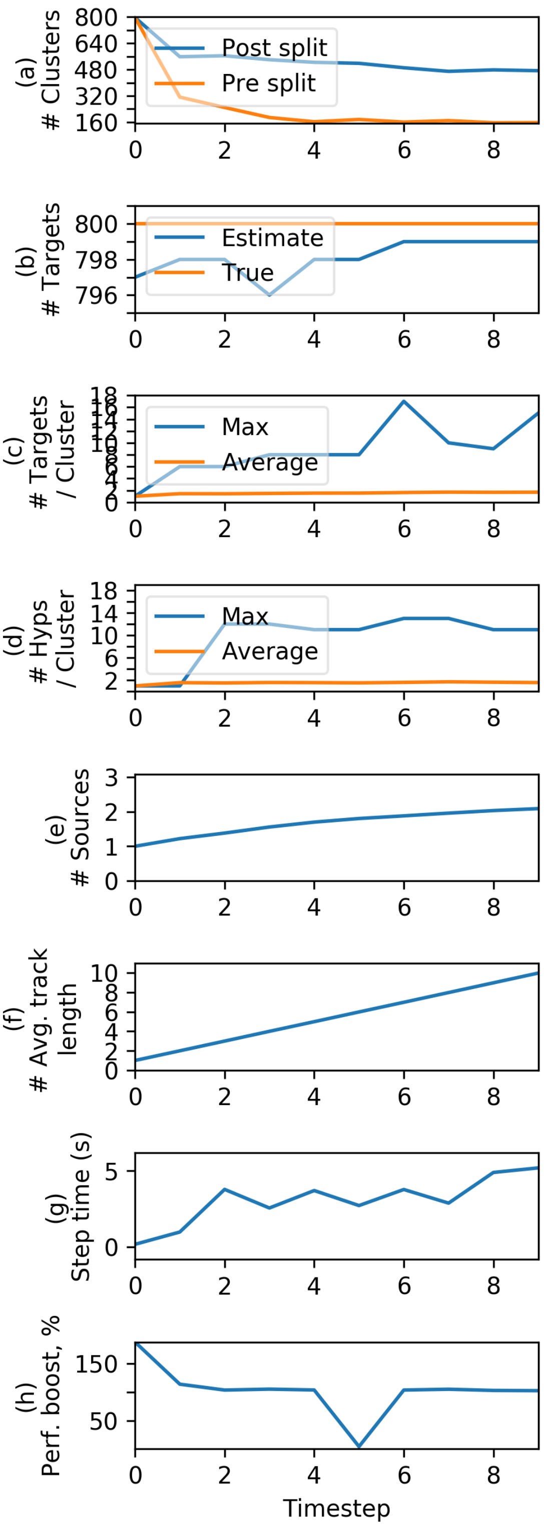


Figure 5.7: Statistics of the ice object tracking sequence

5.5 Conclusions

In this chapter, spatially indexed clustering was explored to determine its benefit when implementing MHT on a massive scale. Results show that the linear search time of the in-memory naïve cluster matching is significantly reduced, by several orders of magnitude, when exchanged for the R-tree approach, but that modern database lookups can achieve comparable results. In particular, when comparing to execution time of the full MHT algorithm, the gain with the choice of R-tree over the standard database cluster match is modest — however there is no obvious drawback to this choice. A major enabler for the fast cluster-match lookups is the overlap gating of (4.35), applied to the report–cluster association. Without this result, gates would need to be individually calculated for each report–track pair, entailing a higher calculation load as well as loading clusters into memory. A high level of partionability of the algorithm seems to be the big part of the speed gain, not the exact details on how the clusters are stored.

Notably, the low sample rate and transmission speed of SAR imagery allow the update to run for quite some time and still be usable in "real time", though the value of the tracking of individual objects will decrease with time. The large amount of data may however also call for a large amount of hypotheses.

As noted in Figure 5.6, ice tracking may result in dense object clouds which are interconnected through ambiguous reports. This effect can be mitigated through the use of MHT pruning parameters (e.g. limited number of hypotheses or association likelihood). As the spatial indexing in the MHT implementation operates on clusters, a large amount of clustering is crucial for the performance increase provided by spatial indexing.

The principle of spatial indexing could easily be extendible to other filtering techniques than MHT, either on a cluster, target or track level. Other considered extensions include the integration of the current filter with more real-world data and sensors, including an automated process for satellite image processing of real-time acquisitions.

6 Scaling the Labeled Multi-Bernoulli Filter

In this chapter, we explore the applicability of the Labeled Multi-Bernoulli (LMB) algorithm to large-scale Multiple Target Tracking (MTT) as performed by the tracking module in the sytem overview of Figure 6.1. The contents of this chapter is primarily based on the following papers:

Paper B: Olofsson, J., Brekke, E., and Johansen, T. A. (2017b). Cooperative remote sensing of ice using a spatially indexed labeled multi-Bernoulli filter. In *International Conference Unmanned Aircraft Systems (ICUAS)*, Miami, USA. © 2017 IEEE.

Paper C: Olofsson, J., Veibäck, C., and Hendeby, G. (2017d). Sea ice tracking with a spatially indexed labeled multi-Bernoulli filter. In *20th International Conference on Information Fusion (FUSION)*, Xi'an, China. © 2017 IEEE.

The contributions of this chapter include:

- a novel simplifying reformulation of the LMB filter better suited for implementation;
- a simulated example of the collaboration of multiple types of sensors, with different sampling rates and Field-of-View (FOV);
- a tracking scenario from Terrestrial Radar Interferometer (TRI) data; and
- a description of the open-source algorithm implementation in PYTHON and C++ respectively.

In polar region operations, drift ice positioning and tracking is useful for both scientific and safety reasons. At its core is a MTT problem in which currents and winds make motion modeling difficult. One recent algorithm in the MTT field is the LMB filter. In particular, a proposed reformulation of the LMB equations exposes a structure which is exploited to propose a compact algorithm for the generation of the filter's posterior distribution. Further, spatial indexing is applied to the clustering process of the filter, allowing efficient separation of the filter into smaller, independent parts with lesser total complexity than that of an unclustered filter.

Many types of sensors can be employed to generate detections of sea ice. In this chapter, a recorded dataset from a TRI is used to demonstrate the application of the Spatially Indexed LMB filter to estimate the currents of an observed area in Kongsfjorden, Svalbard.

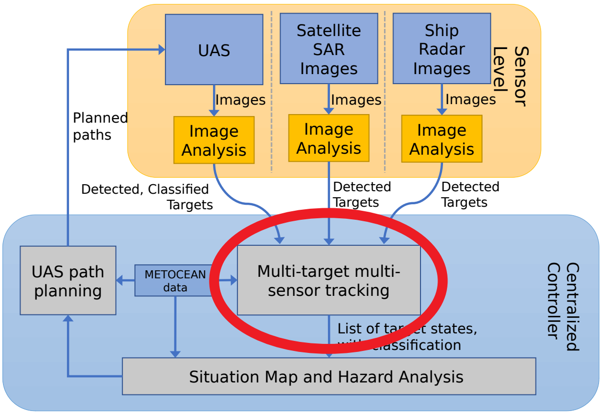


Figure 6.1: Multiple Target Tracking in system context

6.1 Introduction

Tracking of sea ice movements is an essential part of ice management. Multiple types of sensors have been studied for this application — such as satellite-carried Synthetic Aperture Radar (SAR) (Olofsson et al., 2017a), Unmanned Aerial Systems (UAS's) (Johansen and Perez, 2016; Haugen, 2014; Leira, 2017) and, as studied in this chapter, TRI (Voytenko et al., 2015).

At the core of the tracking of individual sea ice objects lies the problem of MTT. For this purpose we study the use of the LMB filter (Reuter et al., 2014; Williams, 2015) — presented in Section 6.2 — and also make a contribution with the first main result of the chapter: a reformulation, in Section 6.2.3, of the LMB equations to expose a structure which lends itself to an efficient algorithm for the filter implementation.

In Section 6.3, we present the contribution of a Free and Open Source Software (FOSS) implementation of the Spatially Indexed Labeled Multi-Bernoulli filter, which was implemented to demonstrate the aforementioned proposed algorithm. The implementation is verified through simulated examples in Section 6.4.

The implementation is then applied, in the chapter's second main contribution, to a TRI dataset provided by the research institute Norut. The application and dataset are described with results and conclusions in Sections 6.5 and 6.6, respectively.

6.2 Labeled Multi-Bernoulli Filter

The LMB filter is defined in the framework of Finite Set Statistics (FISST) (Reuter et al., 2014), of which the Random Finite Set (RFS) is an integral part. An introduction to RFS's was given in Section 4.3. In this section we present the LMB algorithm, and proposes a reformulation of the equations and an algorithm that follows for the calculation of the filter posterior distribution.

6.2.1 LMB Filter Recursion

The LMB filter follows the classical predict/correct filter recursion, each step outlined below.

LMB Prediction

Given an LMB Probability Density Function (PDF) $\pi\left(\mathcal{X}\right)$, the prediction step of the LMB filter and the updated distribution, $\pi_+\left(\mathcal{X}\right)$, is obtained by the application of the standard prediction update of a Bayesian filter, the Chapman-Kolmogorov equation[1],

$$\pi_+\left(\mathcal{X}_+\right) = \int f\left(\mathcal{X}_+\right)\pi\left(\mathcal{X}\right)\delta\mathcal{X}, \tag{6.1}$$

extended for an RFS as defined in Vo and Vo (2013). This gives the following set of surviving and new-born targets (Reuter et al., 2014),

$$\pi_+ = \left\{\left(r_{+,s}^{(\ell)}, p_{+,s}^{(\ell)}\right)\right\}_{\ell\in\mathcal{L}} \cup \left\{\left(r_B^{(\ell)}, p_B^{(\ell)}\right)\right\}_{\ell\in\mathcal{B}} \tag{6.2}$$

[1] Note that the syntax here differs slightly from (4.7a) to match existing LMB literature

where

$$r_{+,s}^{(\ell)} = \eta_s(\ell)\, r^{(\ell)}, \tag{6.3}$$

$$p_{+,s}^{(\ell)} = \frac{\left\langle p_s(\cdot,\ell)\, f(\boldsymbol{x}|\cdot,\ell), p(\cdot|\ell)\right\rangle}{\eta_s(\ell)}, \tag{6.4}$$

$$\eta_s(\ell) = \left\langle p_s(\cdot,\ell), p(\cdot,\ell)\right\rangle, \tag{6.5}$$

$p_s(\cdot,\ell)$ is the distribution of target survival probability and $f(\boldsymbol{x}|\cdot,\ell)$ is the transition density. The set $\left\{\left(r_B^{(\ell)}, p_B^{(\ell)}\right)\right\}_{\ell\in\mathcal{B}}$ is given by the birth model, further discussed in Section 6.2.2.

LMB Measurement Update

Drawn from the update of the δ-Generalized Labeled Multi-Bernoulli (δ-GLMB) (Vo and Vo, 2013), the LMB measurement update is derived in Reuter et al. (2014).

In general, as noted in Reuter et al. (2014), the LMB distribution is not closed under the Bayesian filter measurement update. This means that when inserting an LMB distribution into the measurement update equations, the algebraic result cannot be fully represented by an LMB distribution. However, the resulting δ-GLMB distribution — which is able to represent multiple disjoint hypotheses — may be approximated as in (6.7a) with an LMB PDF through the collapse of its hypotheses, weighted by their probabilities. That is, the measurement updates the set

$$\pi_+ = \left\{\left(r_+^{(\ell)}, p_+^{(\ell)}\right)\right\}_{\ell\in\mathcal{L}_+} \tag{6.6}$$

by the following approximation, for N_ζ clusters:

$$\pi(\cdot|\mathcal{Z}) \approx \left\{\left(r^{(\ell)}, p^{(\ell)}\right)\right\}_{\ell\in\mathcal{L}_+} = \bigcup_{\zeta=1}^{N_\zeta} \left\{\left(r^{(\ell,\zeta)}, p^{(\ell,\zeta)}\right)\right\}_{\ell\in\mathcal{L}_+^{(\zeta)}}, \tag{6.7a}$$

in which parameters are given by

$$r^{(\ell,\zeta)} = \sum_{(\mathcal{I}_+,\theta)\in\mathcal{F}\left(\mathcal{L}_+^{(\zeta)}\right)\times\Theta_{\mathcal{I}_+}} w^{(\mathcal{I}_+,\theta)}\left(\mathcal{Z}^{(\zeta)}\right) 1_{\mathcal{I}_+}(\ell), \tag{6.7b}$$

$$p^{(\ell,\zeta)}(\boldsymbol{x}) = \frac{1}{r^{(\ell,\zeta)}} \sum_{(\mathcal{I}_+,\theta)\in\mathcal{F}\left(\mathcal{L}_+^{(\zeta)}\right)\times\Theta_{\mathcal{I}_+}} w^{(\mathcal{I}_+,\theta)}\left(\mathcal{Z}^{(\zeta)}\right) \times 1_{\mathcal{I}_+}(\ell)\, p^{(\theta)}\left(\boldsymbol{x},\ell|\mathcal{Z}^{(\zeta)}\right), \tag{6.7c}$$

where $\Theta_{\mathcal{I}_+}$ is the space of hypothesis mappings of tracks, with assignment mappings $\theta : \mathcal{I}_+ \rightarrow \left[0, \ldots, \left|\mathcal{Z}^{(\zeta)}\right|\right]$, such that $\theta(\ell) = \theta(\ell') > 0$ implies that $\ell = \ell'$, i.e., the mapping is unique for all values except those mapped to zero, i.e. tracks with no associated measurement (Reuter et al., 2014). Also, for $\mathcal{I}_+ \subseteq \mathcal{L}_+^{(\zeta)}$

$$w^{(\mathcal{I}_+,\theta)}\left(\mathcal{Z}^{(\zeta)}\right) \propto w_{+,\zeta}^{(\mathcal{I}_+)}\left[\eta_{\mathcal{Z}^{(\zeta)}}^{(\theta)}\right]^{\mathcal{I}_+}, \tag{6.8a}$$

$$w_{+,\zeta}^{(\mathcal{I}_+)} = \prod_{\ell \in \mathcal{L}_+^{(\zeta)} - \mathcal{I}_+}\left(1 - r_+^{(\ell)}\right)\prod_{\ell' \in \mathcal{I}_+} r_+^{(\ell)}, \tag{6.8b}$$

$$\eta_{\mathcal{Z}^{(\zeta)}}^{(\theta)}(\ell) = \left\langle p_+^{(\ell,\zeta)}(x), \psi_{\mathcal{Z}^{(\zeta)}}(\cdot, \ell; \theta)\right\rangle, \tag{6.8c}$$

$$\psi_{\mathcal{Z}^{(\zeta)}}(x, \ell; \theta) = \begin{cases} \frac{p_D(x,\ell)p_G g\left(z_{\theta(\ell)}|x,\ell\right)}{\kappa\left(z_{\theta(\ell)}\right)}, & \theta(\ell) \neq z_\emptyset, \\ q_{D,G}(x, \ell), & \theta(\ell) = z_\emptyset, \end{cases} \tag{6.8d}$$

$$q_{D,G}(x, \ell) = 1 - p_D(x, \ell)\, p_G, \tag{6.8e}$$

$$p^{(\theta)}\left(x, \ell|\mathcal{Z}^{(\zeta)}\right) = \frac{p_+^{(\ell,\zeta)}(x)\,\psi_{\mathcal{Z}^{(\zeta)}}(x, \ell; \theta)}{\eta_{\mathcal{Z}^{(\zeta)}}^{(\theta)}(\ell)}, \tag{6.8f}$$

where p_G is the gating probability, $g\left(z_{\theta(\ell)}|x, \ell\right)$ is the likelihood and $\kappa\left(z_{\theta(\ell)}\right)$ is the (Poisson) clutter intensity (Reuter et al., 2014). To exhaustively iterate all hypothetical report–target associations, the operator $\mathcal{F}(\mathcal{A})$ is used to denote the collection of all subsets of set $\mathcal{A}$.

Note that in general MTT, if different hypothetical associations assign the same report to different targets, the resulting tracks will generally be incompatible. Through the approximations of the LMB filter, assignment compatibility is only considered in the hypothesis generation, and is then lost in the summations in (6.7b)–(6.7c). This also simplifies clustering for the LMB algorithm, as no historical incompatibilities are considered.

Originally, the LMB filter was derived using a δ-GLMB reconstruction from the LMB distribution (Reuter et al., 2014). While this is an intermediate representation in the theoretical derivation of the filter, its construction in implementation is not necessary to reach the collapsed LMB representation of (6.7a).

To calculate the weight from (6.8a) for a hypothesis we start by making the distinction between associated and non-associated targets by splitting the hypothesis label set $\mathcal{I}_+$:

$$\mathcal{I}_+^a = \{\ell : \quad \theta(\ell) \neq z_\emptyset\}_{\ell \in \mathcal{I}_+}, \tag{6.9a}$$

$$\mathcal{I}_+^n = \{\ell : \quad \theta(\ell) = z_\emptyset\}_{\ell \in \mathcal{I}_+}, \tag{6.9b}$$

(implying $\mathcal{I}_+ = \mathcal{I}_+^a \cup \mathcal{I}_+^n$ and $\mathcal{I}_+^a \cap \mathcal{I}_+^n = \emptyset$). We can then rewrite (6.8a)–(6.8e) as

$$\begin{aligned} w^{(\mathcal{I}_+,\theta)}\left(\mathcal{Z}^{(\zeta)}\right) \propto & w_{+,\zeta}^{(\mathcal{I}_+)}\left[\eta_{\mathcal{Z}^{(\zeta)}}^{(\theta)}\right]^{\mathcal{I}_+} \\ = & \prod_{\ell \in \mathcal{L}_+^{(\zeta)} - \mathcal{I}_+} \left(1 - r_+^{(\ell)}\right) \\ & \times \prod_{\ell' \in \mathcal{I}_+^a} r_+^{(\ell')} \eta_{\mathcal{Z}^{(\zeta)}}^{(\theta)}(\ell') \prod_{\ell'' \in \mathcal{I}_+^n} r_+^{(\ell'')} \eta_{\mathcal{Z}^{(\zeta)}}^{(\theta)}(\ell''), \end{aligned} \tag{6.10}$$

This product can be efficiently expressed using the Negative Log Likelihoods (NLLS), Λ_ℓ;

$$e^{-\Lambda_\ell} = \begin{cases} 1 - r_+^{(\ell)}, & \text{if } \ell \in \mathcal{L}_+^{(\zeta)} - \mathcal{I}_+, \\ r_+^{(\ell)} \eta_{\mathcal{Z}^{(\zeta)}}^{(\theta,a)}(\ell), & \text{if } \ell \in \mathcal{I}_+^a, \\ r_+^{(\ell)} \eta_{\mathcal{Z}^{(\zeta)}}^{(\theta,n)}(\ell), & \text{if } \ell \in \mathcal{I}_+^n, \end{cases} \tag{6.11a}$$

yielding

$$w^{(\mathcal{I}_+,\theta)}\left(\mathcal{Z}^{(\zeta)}\right) \propto \exp\left(-\sum_{\ell \in \mathcal{L}_+^{(\zeta)}} \Lambda_\ell\right). \tag{6.11b}$$

Each hypothesis $(\mathcal{I}_+, \theta)$ is generated for each cluster in order of decresing probability using Murty's algorithm (Murty, 1968; Miller et al., 1997), to create a truncated approximation of the full sums of (6.7b), (6.7c). The truncation is achieved through the termination of Murty's algorithm based on either a maximum number of drawn hypotheses, or a minimum hypothesis probability.

In the context of LMB, the cost matrix $\boldsymbol{C}$ for Murty's algorithm is created for each target cluster, from the NLLS of the track assignments outlined in (6.11a), resulting in a matrix exemplified by

$$\boldsymbol{C} = \begin{pmatrix} {}^{z_1}\Lambda_{\ell_1} & {}^{z_2}\Lambda_{\ell_1} & {}^{n}\Lambda_{\ell_1} & \infty & {}^{F}\Lambda_{\ell_1} & \infty \\ {}^{z_1}\Lambda_{\ell_2} & {}^{z_2}\Lambda_{\ell_2} & \infty & {}^{n}\Lambda_{\ell_2} & \infty & {}^{F}\Lambda_{\ell_2} \end{pmatrix}, \tag{6.12}$$

for a two-track hypothesis and a two-report scan (with costs for each target being associated, non-associated and false targets, respectively).

6.2.2 Adaptive Birth Model

To include new targets in the tracker, the LMB filter relies on a birth distribution. Different birth-models has been discussed in e.g. Williams (2015), but here, following (Reuter et al., 2014), the selected birth model for time $k+1$ is based on the reports of time k:

$$\pi_{B,k+1} = \left\{\left(r_B^{(\ell)}, p_B^{(\ell)}\right)\right\}_{\ell \in \mathcal{B}_k} \tag{6.13}$$

for new labels in $\mathcal{B}_k$ generated for each report in $\mathcal{Z}_k$.

The existence probabilities of new targets in this model are proportional to the probability of the report not being associated with any previously known target. For a report, the association probability is given by

$$r_{U,k}(z) = \sum_{(\mathcal{I}_+,\theta)\in\mathcal{F}\left(\mathcal{L}_+^{(\zeta)}\right)\times\Theta_{\mathcal{I}_+}} w^{(\mathcal{I}_+,\theta)}\left(\mathcal{Z}^{(\zeta)}\right) 1_\theta(z). \tag{6.14}$$

Given an expected number of new targets in each scan, $\lambda_{B,k+1}$ — the existence probability of new targets — is thus given by

$$r_{B,k+1}(z) = \min\left(r_B^{\max}, \frac{(1 - r_{U,k}(z)) \cdot \lambda_{B,k+1}}{\sum_{z'\in\mathcal{Z}_k} 1 - r_{U,k}(z')}\right). \tag{6.15}$$

Note that, for $\lambda_{B,k+1} > 1$, the existence probability may need to be limited by the min()-clause to a maximum value of $r_B^{\max} \leq 1$.

6.2.3 Reformulation

The classic LMB filter formulation of Reuter et al. (2014) carries a heritage from the δ-GLMB implementation, leading to the necessity of the artificial reconstruction of the δ-GLMB distribution using the k-shortest-path algorithm in the measurement update stage. Here we propose a formulation that does not require the δ-GLMB reconstruction due to its immediate collapse into the LMB-distribution approximation.

In any hypothetical association, a target may be assigned as either associated to a specific report, missed, or assumed non-existent. In the following, being missed is defined as being associated with the null report, $z_\emptyset$. However, unlike for standard reports, any number of targets may be assigned to $z_\emptyset$. This warrants the following definition:

$$\mathcal{Z}^\dagger = \mathcal{Z} \cup \{z_\emptyset\}. \tag{6.16}$$

With this definition, we note that $p^{(\theta)}(\boldsymbol{x},\ell|\mathcal{Z})$ in (6.7c), with the cluster index omitted for notational convenience, belongs to a limited set:

$$p^{(\theta)}(\boldsymbol{x},\ell|\mathcal{Z}) \in \left\{p^{(\ell)}(\boldsymbol{x}|z)\right\}_{z\in\mathcal{Z}^{\dagger}}. \tag{6.17}$$

Thus, the sums in (6.7b)–(6.7c) may be partitioned as follows, abbreviating with $w^{\theta} = w^{(\mathcal{I}_{+},\theta)}\left(\mathcal{Z}^{(\zeta)}\right)$ and denoting the inner sums as ${}^{z}w_{\ell}$. Again, as defined in the notation, $A^{\theta}_{z\leftrightarrow\ell}$ is the indicator function for hypothesis θ assigning report z (or $z_{\emptyset}$) to label ℓ (also implying the inclusion function $1_{\mathcal{I}_{+}}(\ell)$).

$$\begin{aligned} r^{(\ell)} &= \sum_{z\in\mathcal{Z}^{\dagger}}\left[\sum_{(\mathcal{I}_{+},\theta)\in\mathcal{F}(\mathcal{L}_{+})\times\Theta_{\mathcal{I}_{+}}} w^{\theta}A^{\theta}_{z\leftrightarrow\ell}\right] \\ &= \sum_{z\in\mathcal{Z}^{\dagger}} {}^{z}w_{\ell}, \end{aligned} \tag{6.18}$$

$$\begin{aligned} p^{(\ell)}(\boldsymbol{x}) &= \frac{1}{r^{(\ell)}}\sum_{z\in\mathcal{Z}^{\dagger}}\left[\sum_{(\mathcal{I}_{+},\theta)\in\mathcal{F}(\mathcal{L}_{+})\times\Theta_{\mathcal{I}_{+}}} w^{\theta}A^{\theta}_{z\leftrightarrow\ell}\right]p^{(\ell)}(\boldsymbol{x}|z) \\ &= \frac{1}{r^{(\ell)}}\sum_{z\in\mathcal{Z}^{\dagger}} {}^{z}w_{\ell}\,p^{(\ell)}(\boldsymbol{x}|z). \end{aligned} \tag{6.19}$$

We see that ${}^{z}w_{\ell}$ corresponds to *the sum of weights of all hypotheses that assign report z to label ℓ*. The outer sum includes all reports (and the null report), altogether covering the same summands as the original sums.

Further, the birth model of (6.14) may be rewritten as follows (for time index k):

$$r_{U,k}(z) = \sum_{\ell\in\mathcal{L}_{+}^{(\zeta)}}\left[\sum_{(\mathcal{I}_{+},\theta)\in\mathcal{F}\left(\mathcal{L}_{+}^{(\zeta)}\right)\times\Theta_{\mathcal{I}_{+}}} w^{\theta}A^{\theta}_{z\leftrightarrow\ell}\right] \tag{6.20}$$

$$= \sum_{\ell\in\mathcal{L}_{+}^{(\zeta)}} {}^{z}w_{\ell}. \tag{6.21}$$

To exploit this reformulation, consider a cluster of $N_{\mathcal{X}}$ targets and $N_{\mathcal{Z}}$ reports, and a matrix $\boldsymbol{W} \in \mathbb{R}^{N_{\mathcal{X}}\times(N_{\mathcal{Z}}+2)}$. Further, consider a hypothesis assignment mapping $R_{\theta}(i)$ to be used for mapping each row index of $\boldsymbol{W}$ (corresponding to a target) to a column index (corresponding to an assignment). Hence, for all known targets (rows), $R_{\theta}(i)$

1. maps associated targets to its report's integer position in an ordered enumeration of the reports;
2. maps missed targets to the integer index $N_{\mathcal{Z}} + 1$; and
3. maps false targets to the integer index $N_{\mathcal{Z}} + 2$.

Using this mapping, Algorithm 1 works on the assignments θ and the hypothesis score w^{θ} (of (6.11b)), drawn from each iteration of Murty's algorithm, to readily form the relevant sums of the LMB algorithm, without the reconstruction of the δ-GLMB of the original formulation. Note that the addition to the matrix in the algorithm adds the value w^{θ} once to each row in the column specified by $R_{\theta}(i)$.

Algorithm 1 Weight matrix calculation

```
$\boldsymbol{W} \leftarrow N_{\mathcal{X}} \times (N_{\mathcal{Z}} + 2)$ zero matrix.
$s \leftarrow 0$
for $(w^{\theta}, \theta) \in \text{murty}(\boldsymbol{C})$ do
    $\boldsymbol{W}[i, R_{\theta}(i)] \leftarrow \boldsymbol{W}[i, R_{\theta}(i)] + w^{\theta}, \ \forall i \in [1, \ldots, N_{\mathcal{X}}]$
    $s \leftarrow s + w^{\theta}$
end for
$\boldsymbol{W} \leftarrow \frac{\boldsymbol{W}}{s}$
```

The result is exemplified in (6.22) for a problem of three targets and two reports. Recalling the definitions of $r^{(\ell)}$ and $r_{U,k}(z)$, from (6.18) and (6.20) respectively, we see that these correspond to the column (excluding the "False" column) and row sum respectively, as illustrated in (6.22).

$$\begin{array}{rc} & \qquad\qquad\qquad\qquad r^{(\ell)} \\ \boldsymbol{W} = & \begin{pmatrix} {}^{z_1}w_{\ell_1} & {}^{z_2}w_{\ell_1} & {}^{\emptyset}w_{\ell_1} & {}^{F}w_{\ell_1} \\ {}^{z_1}w_{\ell_2} & {}^{z_2}w_{\ell_2} & {}^{\emptyset}w_{\ell_2} & {}^{F}w_{\ell_2} \\ {}^{z_1}w_{\ell_3} & {}^{z_2}w_{\ell_3} & {}^{\emptyset}w_{\ell_3} & {}^{F}w_{\ell_3} \end{pmatrix}. \\ & \qquad\qquad\qquad r_{U,k}(z) \end{array} \tag{6.22}$$

Thus, with this proposed reformulation and algorithm, it is possible to easily extract the existence probabilities. Moreover, through (6.19), this formulation clarifies that the target PDFS may be attained simply through the weighted sum of $N_{\mathcal{Z}} + 1$ PDFS, instead of one per hypothesis. Note that in an implementation, the last column is unnecessary.

The proposed algorithm reduces the complexity of the algorithm by the complete removal of the, at best, pseudo-polynomially complex δ-GLMB reconstruction step. Additionally, the more complex multi-hypothesis δ-GLMB update is replaced with a single iteration over hypotheses generated by the Murty algorithm, as in Algorithm 1.

6.3 Implementation

Two implementations of the LMB filter were published in connection with the articles this chapter build upon.

6.3.1 Python Implementation

This section details the PYTHON language implementation of the Spatially Indexed LMB filter implemented for this thesis. The implementation has been made public under a FOSS license at `https://github.com/jonatanolofsson/lmb`.

A few remarks regarding the implementation;

Particle target tracking For the single-target distributions, the implementation is based on particle filter distributions, where the general equations of (6.8c)–(6.8f) are specialized as in Vo et al. (2014).

Parallelization Due to restrictions in PYTHON parallelization, clusters are currently updated sequentially. However, particle updates are vectorized using the *numpy* package.

Target storage and indexing In the implementation, targets are serialized post-update and stored in an SQLITE database. The database is Rectangle tree (R-TREE) indexed based on the axis-aligned bounding boxes of each target PDF. This allows for a fast extraction of relevant targets in the initial gating process, allowing large parts of the filter to remain dormant through measurement updates.

Rectangular gating The implementation makes use of rectangular gating through the minimum bounding box of reports and targets. As a first stage, all targets within the sensor FOV's minimum axis-aligned bounding-box are loaded from the database. Second, the bounding-box of each target is tried for intersection with the bounding-box of each report to establish the clusters, in accordance with Section 4.5.2. Notably, these bounding-boxes need not be axis-aligned.

6.3.2 C++/Python Implementation

The C++ LMB filter implementation has several features in common with the PYTHON one. For example, like the PYTHON implementation the C++ LMB implementation uses rectangular gating and R-TREE-indexed storage. However, the C++ implementation has a significantly improved design in terms of e.g. memory management, parallelization and speed. Other features distinguishing this implementation includes:

Gaussian mixture target tracking The filtering of individual targets uses Gaussian mixtures. This includes maintenance of the distribution such as pruning of components, where low-weighted Gaussian components are dropped either because of low probability weight, or due to a maximum number of components.

Parallelization The implementation is parallelized using the *OpenMP* API. This can be disabled through compile flags.

Geodetic target storage The implementation uses the geodetic target storage scheme described in Section 4.7.

Performance indicators The implementation includes methods for performance evaluation of tracking performance. In particular, the Optimal Sub-Pattern Assignment (OSPA) (Schuhmacher et al., 2008) and the Generalized Optimal Sub-Pattern Assignment (GOSPA) (Rahmathullah et al., 2017) algorithms are implemented.

PYTHON bindings The implementation includes PYTHON bindings, leveraging the *pybind11* library.

The implementation has been made public under a FOSS license at https://github.com/jonatanolofsson/clmb.

6.4 Simulated Examples

In this section we study the implementation of the Spatially Indexed LMB filter through a linear simulated multi-target tracking scenario. Next we present a simulated sea ice tracking scenario with the collaboration between two UAS agents and a satellite. Both simulations in this section use the PYTHON filter implementation.

6.4.1 Crossing Tracks

This scenario, detailed in Figure 6.2, demonstrates the filter's ability to detect and track objects through a crossing by considering the multiple likely association possibilities. It also illustrates the track-keeping abilities of the LMB filter and the clustering feature of the implementation.

We see, in Figure 6.2:

a) The crossing tracks scenario in which in total of five targets are tracked through collision courses. To simplify visual interpretation, each object was given a velocity of 1 in the x-direction, to match the time dimension of the following plots. Notably, all tracks retain the correct association throughout the simulation, as indicated by the consistent color of each straight track. As targets approach each other and come to share ambiguous reports, it becomes apparent that the LMB filter, like the Joint Probabilistic Data Association (JPDA) (Blom and Bloem, 2000), suffers from track coalescence — the tracks gravitate towards each other when nearby, as associations with "the other" report becomes more plausible.

b) Target cardinality, for both true, estimated and verified targets. A target is considered verified if its existence probability exceeds 0.7.

c) Number of clusters which the algorithm separates. Note that due to hypothetical new-born targets, this can in fact exceed the number of estimated targets.

d) Number of hypotheses used. In sequences where multiple associations are possible, more hypotheses are generated before iteration termination due to low hypothesis probability. In particular, note the peak between $t = 5$ and $t = 10$ corresponding to the period where the two initial targets cross tracks.

6.4.2 Collaborative Sea Ice Tracking

In preparation of sea ice tracking field-tests, a scenario was devised to emulate the anticipated data. In this scenario, shown in Figure 6.3, sea ice objects are caught in a vortex which is initially observed in full by satellite imagery. In the interest of protecting a fixed installation — in the center of the figure — in the following sequence the sea ice in the vortex is partially observed by two independent UAS agents which report wirelessly to the central filter. The central filter then continuously fuses the data received from each sensor into a joint estimate of the sea ice flow field. The observation is also assisted by a second incoming satellite image processed at timestep 75 which span the entire area of interest.

The detections from each UAS are drawn from a FOV which in the UAS-local frame of reference corresponds to the bounding box

$$S_{\mathrm{UAS}} = \begin{bmatrix} -60, & 60, & -20, & 60 \end{bmatrix} \tag{6.23}$$

in each UAS's frame of reference at each timestep.

As sample values, detection probability of UAS reports are set to 0.99, and for the satellite reports to 0.8. 20 icebergs are simulated. As it is unlikely that a verified ice object disappears during the short time frame of the UAS flight, each track has a survival probability (in the prediction step) of 0.999, uniformly. In practice, disappearing objects are instead likely to be removed by the measurement update step, as they will no longer yield reports likely to be associated to the target.

The tracks of the UAS agents are a combination of a loitering UAS with a specific area to guard, and a second UAS track with the intention of more broadly exploring the area.

In Figure 6.3, the ice objects (in black) are plotted with their velocity vectors, together with the (colored) filter estimates. UAS tracks are drawn red. In Figure 6.4, the final timestep's output of the filter is displayed as a Gaussian field.

Notably, whereas position information is available from the fully covering satellite imagery, velocity may only be observed through repeated observations and associations. New targets are initiated with a large variance in their velocity estimate, which results in a large variety of velocities considered within the individual filters. Although only the mean velocity estimates are displayed in the plots for clarity, this means a lot of options for future associations will be considered as the uncertainty in velocity is translated to uncertainty in position over time.

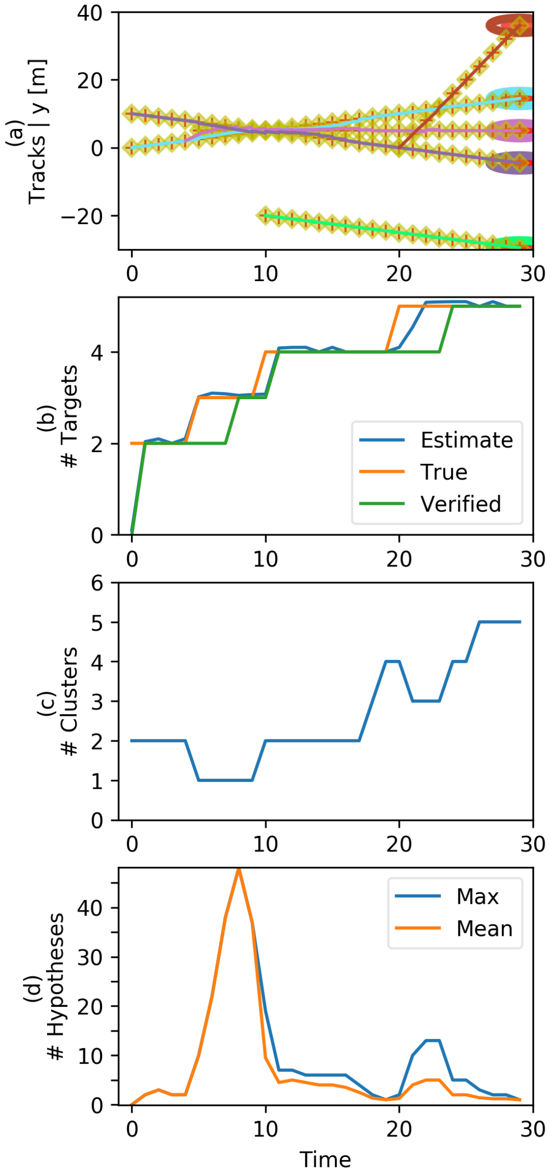


Figure 6.2: Crossing tracks LMB scenario

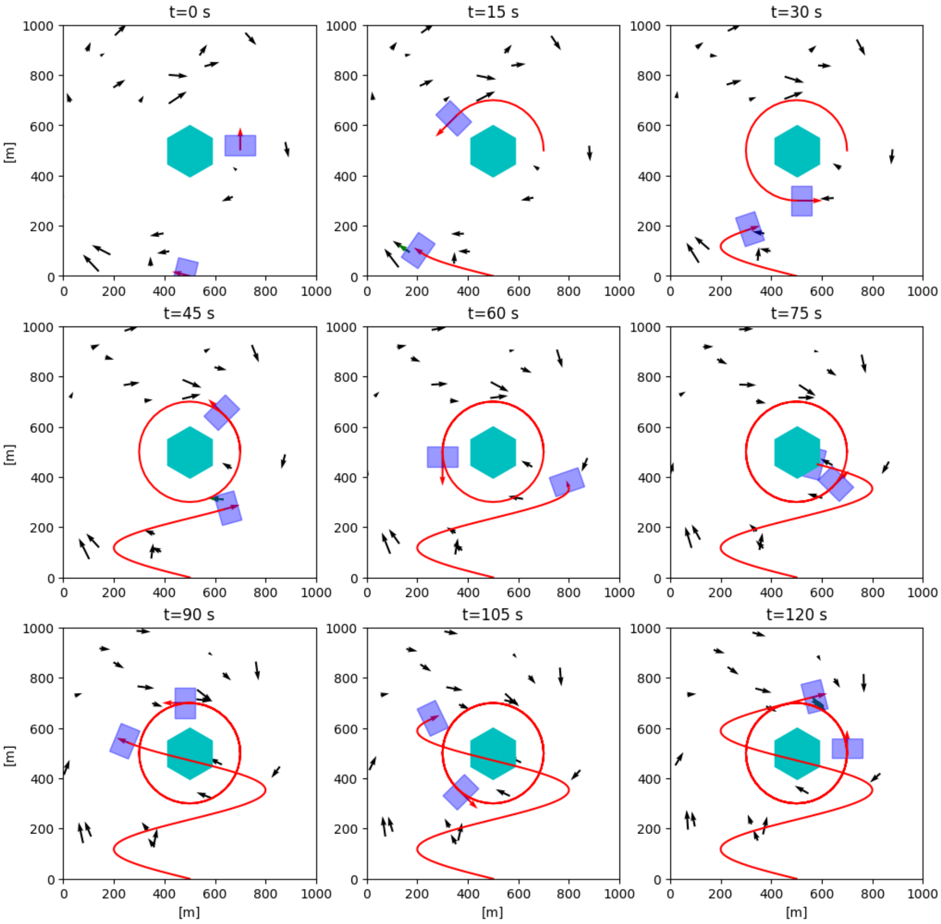


Figure 6.3: The sea ice tracking scenario is based on the collaboration of two UAS's to loiter and explore respectively, assisted by satellite radar measurements. 20 icebergs are simulated in this scenario. The (black) simulated icebergs are here shown along with the (colored) estimates resulting from the joint observations.

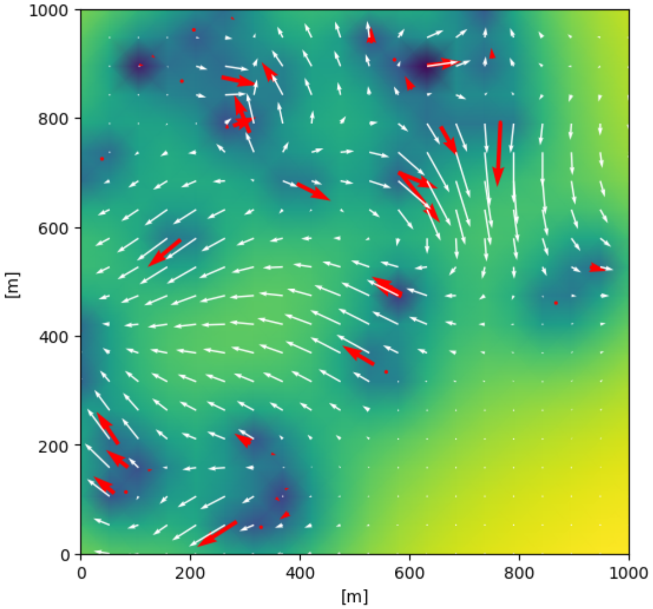


Figure 6.4: The tracker estimates may be used as an input for current and wind estimates (track estimates shown in red).

6.5 Sea Ice Tracking Application

In this application, we use the first dataset presented in Section 3.5.2 to track the ice movements over time using the proposed LMB algorithm. We also give a basic current estimation example based on the data extracted from the tracker. The application uses the PYTHON LMB implementation.

The drift ice is modeled using a Nearly Constant Velocity (NCV) model subject to zero-mean white-noise acceleration. The states in this model are position and velocity in two dimensions. The sensor is modeled to directly measure drift ice positions in the image, with zero-mean Gaussian noise. Uniformly across the scanned region, each detection is assigned a covariance of $12.2^2\,\mathrm{m}^2 \cdot \mathcal{I}_2$ to form reports with Gaussian distributions. The sampling time of the motion model is 180 s, and the motion model covariance parameter is chosen in both dimensions as $1.7 \times 10^{-5}\,\mathrm{m}^2/\mathrm{s}^3$.

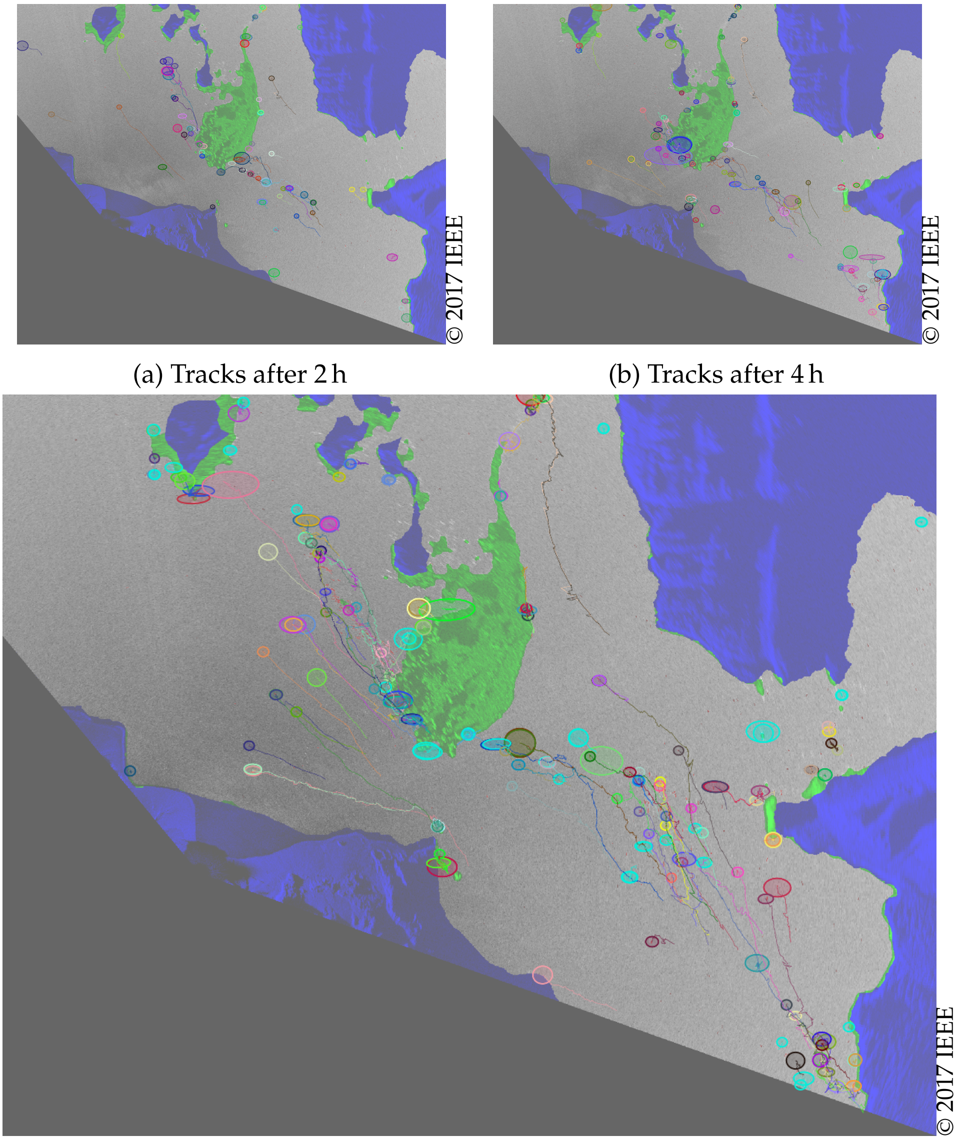
The sea ice movements were tracked over a period of seven hours, with scans delivered every three minutes. In Figure 6.5, we see the tracks of tracked ice objects build up over time.

The stationary sea ice, shown in green, changes only slightly over the course of the experiment, suggesting it would remain largely undetected if treated as drift ice. The detections of drift ice, shown in red, suffer many false alarms, but the LMB filter manages to confirm the targets, shown as ellipses, and maintain their tracks, shown as lines, over large stretches of water.

Looking at the estimated tracks from the LMB filter, there is a trend for the drift ice to move in the north-west direction. This is further corroborated by Figure 6.6 where the currents are estimated from the velocity estimates of the targets.

As can be seen in Figure 6.5, each scan consists of hundreds of reports, approximately 150 to 400, and about 50 to 190 targets are being maintained over time by the LMB filter. However, the low sampling rate of the TRI allows the filter to run in real-time, despite such conditions.

Apart from tracking individual sea ice objects for collision avoidance, the movements of drift ice were also be used to estimate the water currents in the region over time, as shown in Figure 6.6. A Gaussian kernel smoother (Bishop, 2006) was applied to a grid over the water region. The position and time of the drift ice, obtained from the tracks, are used to estimate the local velocity, which depends on the current. The length scale is chosen at 750 m for the distance component and 1 h for the time component. The grid spacing used is 600 m in the water. Note that the method can easily be adapted to an online method by considering only causal measurements. Note too that, unlike the method of Gaussian Fields used in Chapter 7, this particular method does not account for uncertainties in the track estimates.



(a) Tracks after 2 h

(b) Tracks after 4 h

(c) Tracks after 7 h

Figure 6.5: Drift sea ice tracks over time, showing the land mask in blue and stationary detections in green. Tracks and targets retain an individual randomly assigned color over time.

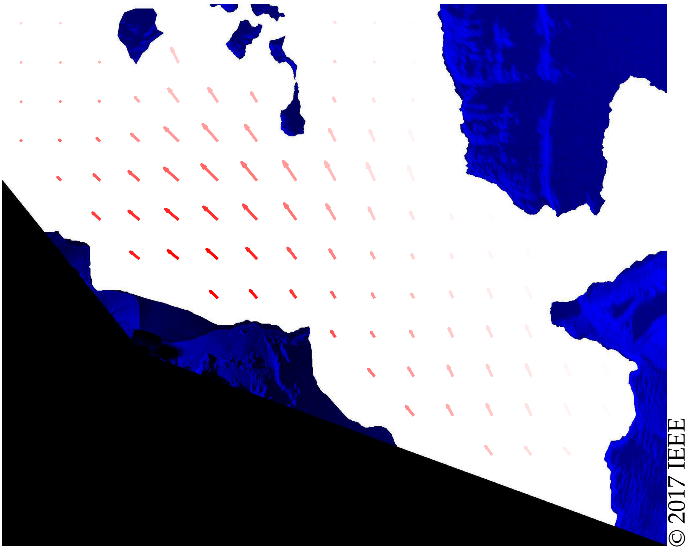


Figure 6.6: Estimation of currents in the water region. The opacity of each arrow is indicative of the amount of data available close to the arrow.

6.6 Conclusions

In this chapter we presented a simplifying reformulation of the LMB filter, which highlights that it largely can be reduced to a summation of hypothesis weights according to the proposed algorithm. This is exploited in an algorithm which simultaneously calculates both the update weights and the weights for the adaptive birth model. Further, we presented its implementation in which also the proposed application of spatial indexing aids in the online formation of target clusters. This clustering is critical for the scalability of the LMB algorithm, and for the continuation of the studied application of tracking a large number of ice objects.

The current information we extract from this tracking scenario could potentially be used for prediction and modeling. However, this might cause information looping which would need to be studied further prior to a practical application. With the possible exception of existence probabilities, the information carried in the LMB filter is essentially the same as other MTT-filters — some of which might be preferred for application-specific advantages in specific cases. In related experiments, Global Nearest Neighbour (GNN) tracking has been used to verify, with good results, the LMB solution studied in this chapter. Note that GNN roughly corresponds to limiting the LMB algorithm to use only the single best hypothesis, and as such is expected to perform faster but less robust to incorrect associations.

For the specific task of current estimation (without positioning) one may also compare the results with techniques such as optical flow. This is, however, outside the scope of this thesis.

Part III

Tracker Information Utilization

Gaussian Field Current Estimation 7

In this chapter, a method is presented to construct a current/wind velocity field map estimate. This map is an example of the data that can be extracted from the tracking module to establish situation awareness maps, as indicated in Figure 7.1. The chapter is primarily based on:

Paper D: Olofsson, J., Flåten, A. L., Veibäck, C., and Lauknes, T. R. (2017c). Gaussian field current estimation from drift sea ice tracking with the labeled multi-Bernoulli filter. In *Proceedings of OCEANS 2017 MTS/IEEE*, Anchorage, Alaska, USA. © 2017 IEEE.

The contributions of this chapter include:

- a description of Gaussian fields and its application to velocity field mapping from tracking data; and
- a velocity field mapping application from Terrestrial Radar Interferometer (TRI) data.

In polar region operations, drift sea ice positioning and tracking is useful for both scientific and safety reasons. Modeling ice movements has proven difficult, not least due to the lack of information of currents and winds of high enough resolution. Thus, observations of drift ice is essential to an up-to-date ice-tracking estimate.

As an inverse problem, it is possible to extract current and wind estimates from the tracked objects of a Multiple Target Tracking (MTT) filter. By inserting the track estimates into a Gaussian field, we obtain a two-dimensional current estimate over a region of interest.

The algorithm is applied to a TRI dataset from Kongsfjorden, Svalbard, to show the practical application of the current estimation.

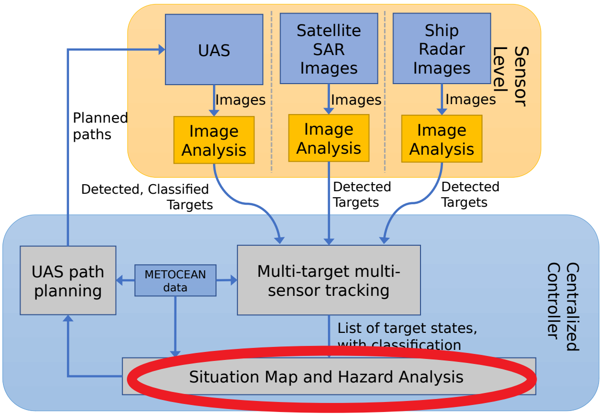


Figure 7.1: System overview: Situation mapping

7.1 Introduction

Any target tracking problem is, ultimately, set up with the purpose of making use of its output, the estimated tracks and states of the estimated target list. In this chapter, we apply Gaussian fields for the creation of a velocity map based on the tracking results presented in Section 6.5. The results are based on a TRI dataset provided by Norut, described in Section 3.5.2.

The main forces involved in sea ice drift is presented in Section 3.2.2, and (3.1) in particular. As discussed in Section 3.2.2, the main forces to act upon the ice object is that of wind and water, both functions of the velocity difference between the object and the air or currents, respectively. This means that the theoretical terminal velocity is given by

$$\frac{1}{2}\rho_w C_w \boldsymbol{A}_k \left|\dot{\boldsymbol{x}}_c - \dot{\boldsymbol{x}}\right| (\dot{\boldsymbol{x}}_c - \dot{\boldsymbol{x}}) + \frac{1}{2}\rho_a C_a \boldsymbol{A}_s \left|\dot{\boldsymbol{x}}_a - \dot{\boldsymbol{x}}\right| (\dot{\boldsymbol{x}}_a - \dot{\boldsymbol{x}}) = 0, \qquad (7.1)$$

for drag coefficients ($C_{(\cdot)}$), medium densities ($\rho_{(\cdot)}$) and velocities ($\dot{\boldsymbol{x}}_{(\cdot)}$) for $a$ir and $w$ater respectively, as well as keel surface area $\boldsymbol{A}_k$ and sail surface area $\boldsymbol{A}_s$.

Thus, under the assumption of similar shape properties $\boldsymbol{A}_k$ and $\boldsymbol{A}_s$, similar ice objects will reach the same terminal velocity. This motivates the

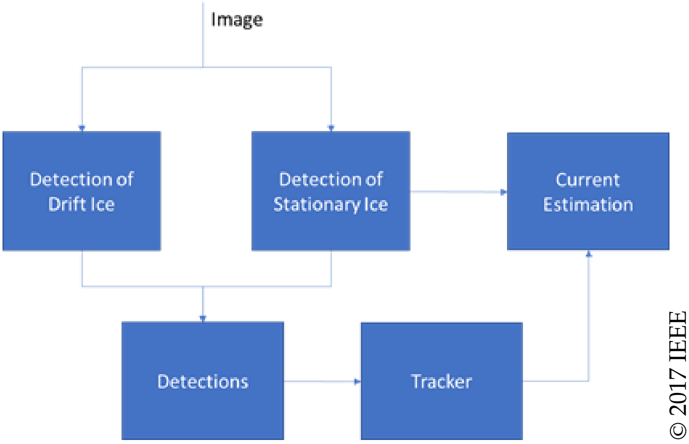


Figure 7.2: Tracking system outline. Scans are reported from multi-target sensing sensors and tracked in an LMB filter. This information is then extrapolated through a Gaussian field current estimation model and evaluated and displayed to the user.

study of velocity field maps based on historical observations, in order to model the velocities of new detections.

The Labeled Multi-Bernoulli (LMB) filter application presented in Section 6.5 estimates positions and velocities of the detected drift sea ice in the scene. In this chapter, the data extracted from the tracker is used to estimate a Gaussian velocity field. Since the tracked objects are passively drifting, this field can be interpreted as the result of currents and/or winds in the vicinity of each object. The estimated field can be used for user presentation, but potentially also for motion modeling and prediction. This flow of data for the tracking system and current estimation is outlined in Figure 7.2.

In Section 7.2, the theory of Gaussian fields is briefly presented and then applied in Section 7.3 where the tuning and results are presented.

7.2 Gaussian Fields

Gaussian fields is the extension of Gaussian processes (Rasmussen and Williams, 2006) into multi-dimensional space, and the standard equations of Gaussian processes are straightforwardly applied by simply extending the state vectors and covariances accordingly.

Gaussian fields are used to estimate a vector field function as point-wise Gaussian distributions at given points. As such it yields, for each point, an estimated function value but also an estimated measure of the uncertainty of this estimate — its covariance. In essence, for a given point of interest, a Gaussian process/field uses the function values of other points and their associated covariances to create a weighted estimate of the function value at the point of interest.

In its simplest form, each dimension of the vector field is separable into independent Gaussian processes although in general, covariance between the dimensions must be taken into account if the dimensions are not fully statistically independent. The field measurements and estimates, as formulated below, are all assumed to be zero mean, although a mean is easily removed beforehand and later re-added.

The evaluation of a Gaussian field is based on the points, $\mathcal{X}_S$, at which the field has been originally sampled. In an application, the value of a function is sampled at those points — and normalized by removing the mean — to form the measurements $\boldsymbol{y}_S$. This enables the evaluation of an expected value at points of interest in $\mathcal{X}_{S*}$, at which we wish to evaluate the estimated function. The equation, from which the concept of Gaussian processes and fields are derived, is the joint Gaussian:

$$\begin{bmatrix} \boldsymbol{y}_S \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^\mathsf{T} & \boldsymbol{K}_{**} \end{bmatrix}\right), \tag{7.2}$$

where $\boldsymbol{f}_*$ is a vector of the (concatenated) vector-field values at the points of interest, and $\boldsymbol{K} = \operatorname{cov}(\boldsymbol{y}_S, \boldsymbol{y}_S)$, $\boldsymbol{K}_* = \operatorname{cov}(\boldsymbol{y}_S, \boldsymbol{f}_*)$, $\boldsymbol{K}_{**} = \operatorname{cov}(\boldsymbol{f}_*, \boldsymbol{f}_*)$ (Rasmussen and Williams, 2006). This leads, for the points in $\mathcal{X}_{S*}$, to the predictive equation

$$\boldsymbol{f}_* | \mathcal{X}_S, \boldsymbol{y}_S, \mathcal{X}_{S*} \sim \mathcal{N}\left(\bar{\boldsymbol{f}}_*, \operatorname{cov}(\boldsymbol{f}_*)\right) \tag{7.3a}$$

where

$$\bar{\boldsymbol{f}}_* \triangleq \boldsymbol{K}_*^\mathsf{T} \boldsymbol{K}^{-1} \boldsymbol{y}_S, \tag{7.3b}$$

$$\operatorname{cov}(\boldsymbol{f}_*) = \boldsymbol{K}_{**} - \boldsymbol{K}_*^\mathsf{T} \boldsymbol{K}^{-1} \boldsymbol{K}_*. \tag{7.3c}$$

Hence, $\bar{\boldsymbol{f}}_*$ contains the mean of the estimated vectors at the points in $\mathcal{X}_{S*}$, and $\operatorname{cov}(\boldsymbol{f}_*)$ contains their joint covariance. In the application presented in this chapter, we consider the velocity estimates as the measurements $\boldsymbol{y}_S$, taken at the estimated position of each target track — $\mathcal{X}_S$.

For Gaussian processes, the matrices $\boldsymbol{K}_{*}$ and $\boldsymbol{K}_{**}$ are predominantly generated by standard covariance-generating functions — *kernels*. These are functions of point pairs which form valid covariance matrices (Duvenaud, 2014), each element (i,j) being

$$\boldsymbol{K}_{i,j} = k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \operatorname{cov}\left(f\left(\boldsymbol{x}_i\right), f\left(\boldsymbol{x}_j\right)\right),$$

where f is the estimate of the underlying function, such that

$$f\left(\boldsymbol{x}\right) \sim \mathcal{GP}\left(m\left(\boldsymbol{x}\right), k\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right) \tag{7.4}$$

for a specified mean function m. In the N_d-dimensional case, $\boldsymbol{K}$ is instead represented as a block matrix with $N_d \times N_d$-dimensional blocks at positions (i,j).

As an example kernel, the Squared Exponential (SE) kernel for scalar points x, x' is defined in one dimension as

$$k_{\mathrm{SE}}\left(x, x'\right) = \sigma^2 \exp\left(-\frac{\left(x - x'\right)^2}{2\ell^2}\right), \tag{7.5}$$

parametrized by the length hyperparameter l and standard covariance σ. This can be extended to the multi-dimensional case

$$\boldsymbol{k}_{\mathrm{SE}}\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \boldsymbol{P} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^{\top} \boldsymbol{L}^{-1}\left(\boldsymbol{x} - \boldsymbol{x}'\right)\right), \tag{7.6}$$

with $\boldsymbol{P}$ being the covariance matrix at $\boldsymbol{x}$ and $\boldsymbol{L}$ being the scaling and rotation of the bell-shaped attenuation when moving away from $\boldsymbol{x}$. Other kernels may be extended analogously.

Note that the points are not necessarily in metric space — kernels may be defined in any dimension. For example, a kernel may be defined for the time dimension to represent a time dependency with a forgetting factor.

While kernels, and combination of kernels (Duvenaud, 2014), is the standard way of forming the submatrices of (7.2), the predictive equations of (7.3a) are valid for all valid covariance matrices. For example — and relevantly — the uncertainty of the filtered velocity estimates of the MTT filter may be incorporated in the $\boldsymbol{K}$ and $\boldsymbol{K}_{*}$ matrices. For the LMB velocity covariance, this corresponds to the case of noisy measurements in the Gaussian process (Rasmussen and Williams, 2006). Another noise source is the uncertainty in the position of the target from which the velocity measurement is taken. However, a detailed handling of this is less straightforward and thus here only taken into account indirectly through the general kernel.

7.3 Velocity Field Mapping in TRI Data

This section describes the application of Gaussian fields to a TRI dataset collected by the Norwegian research institute Norut at Kongsfjorden, Svalbard, see Section 3.5.2.

From the dataset, detections were extracted and tracked with the LMB filter from Chapter 6 to generate tracks with estimated velocities and positions. The estimated target velocities are used to form a Gaussian field of velocities, observed at the targets' estimated positions.

A variation of kernel choices and kernel parametrizations were evaluated against a score for comparing the predictive qualities of the model. A verification set was formed by separating out tracks from the tracker, forming disjoint training- and verification sets. To calculate the score, the model was evaluated at the points of the verification set, and compared to the verification track estimate. Given a velocity prediction and a verification vector — $\bar{v}_\nu$ and v_ν, respectively, with associated covariances $\Upsilon_{\bar{v}_\nu}$ and Υ_{v_ν} — the optimal prediction is obtained when $\bar{v}_\nu = v_\nu$. A description of the distribution difference is the innovation between the two which in the Gaussian case, assuming independence, is formed by the Gaussian distribution

$$\mathcal{N}\left(\bar{v}_\nu - v_\nu | 0, \Upsilon_{\bar{v}_\nu} + \Upsilon_{v_\nu}\right) = \mathcal{N}\left(\tilde{v}_\nu | \vec{0}, \Upsilon_\nu\right). \tag{7.7}$$

A score function may be devised from the Negative Log Likelihood (NLL) of the innovation:

$$\gamma_\nu = \frac{1}{2}\left(\ln|\Upsilon_\nu| + \tilde{v}_\nu^\intercal \Upsilon_\nu{}^{-1} \tilde{v}_\nu + 2\ln 2\pi\right). \tag{7.8}$$

The total score was created for each frame — repeatedly in Monte Carlo fashion — by putting aside 25 % of the detections for the verification dataset and averaging the score they receive when compared to the predictions from the Gaussian field.

For the predictive modeling of the iceberg motion, the kernels considered in particular are presented in Table 7.1.

Additionally, the kernels were combined with a CID kernel over the time dimension. In these cases, historical data from the LMB tracks were used for additional data.

A range of kernels were tested — with and without time kernel — for a variation of physically reasonable hyperparameter settings. The results are

Table 7.1: Kernels considered in the velocity field mapping (for $r = |\boldsymbol{x} - \boldsymbol{x}'|$, hyperparameter l)

Exponential (exp)	$e^{-r/l}$
Squared Exponential (se)	$e^{-(r/l)^2}$
Corrected Inverse Distance (cid)	$\left(1 + \frac{r}{l}\right)^{-1}$
Rational Quadratic (rq)	$\left(1 + \frac{r^2}{l^2}\right)^{-1}$

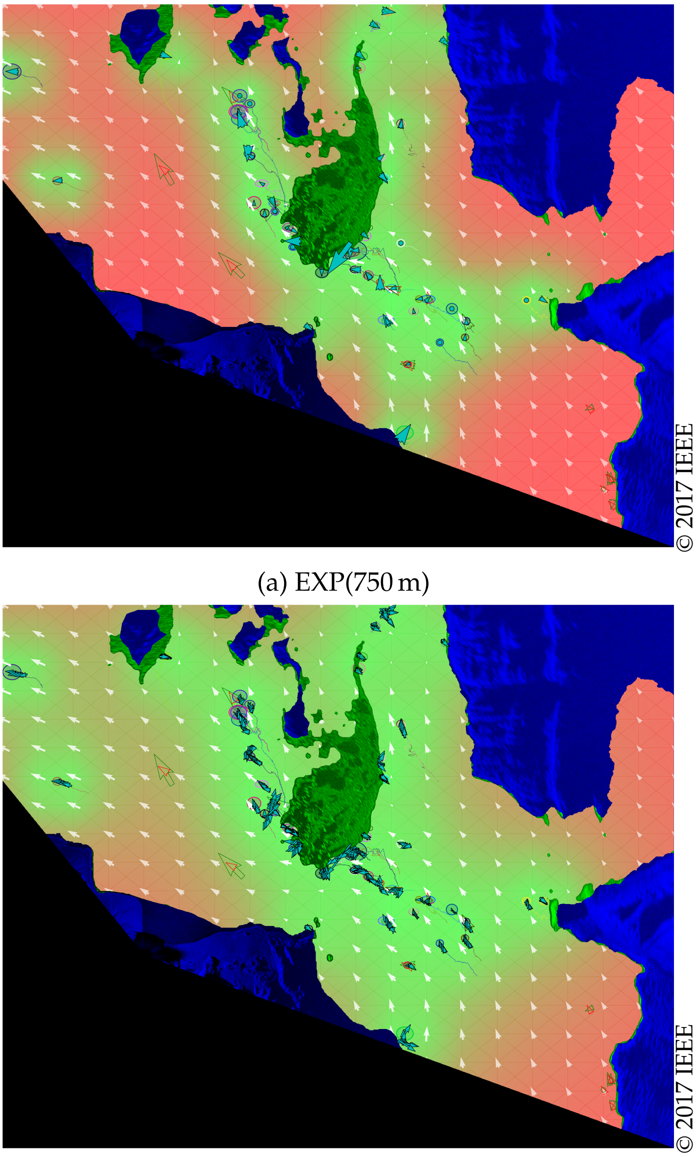
Table 7.2: Score chart for a selected sample of kernels and hyperparameters. A lower score indicates a better match, and a lower covariance indicate consistent performance. Kernel distance hyperparameters are given in meters and minutes respectively. With multiple similar scores, the comparison is inconclusive for a single best setting.

Kernel	Score	Score cov.	Relative, %
exp(400)	5.09175	0.0189955	101.105
exp(750)	5.10378	0.0181352	101.344
exp(1200)	5.12499	0.0369751	101.765
se(200)	6.10111	23.3351	121.147
cid(100)	5.07517	0.0204076	100.775
cid(400)	5.07999	0.0274274	100.871
cid(900)	5.11524	0.0307042	101.571
rq(200)	5.57251	0.578188	110.651
expt(400, 30)	5.05754	0.0236918	100.425
expt(750, 30)	5.07125	0.0203695	100.698
expt(1200, 30)	5.07709	0.0217448	100.813
cidt(100, 30)	5.03612	0.0145595	100
cidt(400, 30)	5.05708	0.0216197	100.416
cidt(900, 30)	5.0955	0.0224501	101.179

summarized in Table 7.2 and exemplified in Figure 7.3 (kernels with a time factor are suffixed by T).

Using two example kernels, the images in Figure 7.3 exemplify the resulting velocity vector field attained from the Gaussian field. The blue area represents the masked land, the green area the stationary ice and the blue arrows the ice objects used for training the Gaussian field. The verification set and their projected estimates are green and red arrows respectively. The background — ranging from red (high) to green (low) — is determined by the trace of the Gaussian field velocity covariance in each point, thus representing the inverse of the level of information available at each point.

The best matches to the verification set are obtained through the use of an exponential or a CID kernel, although for this specific dataset, similar scores are obtainable from different parameter settings. In our simulations, the rapid decline in correlation with the SE kernel was so severe that it caused numerical problems with an l hyperparameter outside the tested range. Thus, only values up to 200 m were tested.



(a) EXP(750 m)

(b) CIDT(1000 m, 90 min)

Figure 7.3: Examples of the resulting velocity vector field attained from the Gaussian field The blue area represents the masked land, the green area the stationary ice and the blue arrows the ice objects used for training the Gaussian field. The verification set and their projected estimates are green and red arrows respectively. The background — ranging from red (high) to green (low) — is determined by the trace of the Gaussian field velocity covariance in each point.

7.4 Conclusions

This chapter has presented a follow-up for the Olofsson et al. (2017d) paper — exploring the extension of MTT state estimation into Gaussian field prediction modeling — and presented an abbreviated introduction to the application of Gaussian fields as a method of modeling ice motion over an observed region, based on the input of tracked ice objects.

Based on the available observations, the Gaussian field provides a continuous representation of the predicted object velocities of an area. Due to the lack of precise information about for example ice object size and weight, it is difficult to draw precise predictive conclusions about other, or future, ice objects as the forces will act differently depending on object size and other physical properties. One remedy for this may be to include information about e.g. shape-describing Hu moments (Hu, 1962) and use this information in the kernels as measures of inter-object proximity, showing that similar objects move alike. The steady-state assumption that corresponds to assuming similar speeds of nearby ice-objects appears, however, to work reasonably well in practice.

Since it is its main input, the performance of the Gaussian field model strongly depends on the quality of the tracker. Further tuning, testing and verification of both the tracker and the Gaussian field model is still required to attain a general result which confidently describes the scenario. One potential improvement, for scalability as well as improved results, would be to create a more local model of the velocity mean. Currently the mean is shared throughout the entire dataset. Relevantly, a major limitation for using Gaussian fields with large datasets is the computational burden of inverting large matrices. This can partially be remedied through the use of gating — a process in which only the points which most affect the result are selected to create a considerably smaller matrix to invert, at the same time yielding more local results for each point. This gating could be naturally facilitated with the spatially indexed storage in the LMB implementation used here.

The application of Gaussian models are often automated through the optimization of the kernel hyperparameters using e.g. Monte Carlo optimization. This is of course relevant here, although must be combined with the manual addition of the experience and understanding of physically relevant hyperparameter intervals. It also requires datasets of significant size, which is not yet available for this particular application.

With parameter tuning, very similar results can be obtained with different kernels. The similarities in scores also indicates that in this dataset, the velocities are generally not too far from the mean. Thus, rather than conclude a specific best kernel choice for this application, we chose to focus on the general process of combining the multi-target tracking with Gaussian fields to attain a velocity field model over the observed region.

Proposed future work include two primary aspects: the feedback of the Gaussian field to the LMB filter for ice motion prediction, and its use for the planning of information acquisition. In the first case, the velocity model obtained in the Gaussian field can be utilized e.g. in the initialization of new targets in the tracker, providing an improved model for initiating new targets from single detections where velocity data is otherwise unavailable. This is explored further in Veibäck et al. (2018).

In the second case, we can see the Gaussian field covariance measure as an inverse metric of information. This metric can be employed in an optimization routine to plan the route of one or more moving sensor agents, to maximize the information gain.

8

Informed Path Planning

The content of this chapter is based upon the following paper:

Paper F: Olofsson, J., Hendeby, G., Lauknes, T. R., and Johansen, T. A. (2018). Multi-Agent Informed Path-Planning Using the Probability Hypothesis Density. *Autonomous Robots*, (Submitted, Aug 2018). 2018 IEEE

This paper is awaiting publication and is not included in NTNU Open

9 Conclusions

The goal of this thesis has been to present the research performed on the requirements of a modern MTT system capable of scaling to the challenge of tracking sea ice on a global scale, and to explore the utility of the data in the system. This has, in particular, been explored from the perspective of using the UAS platform as a viable supplement to other available sensors. To help formalize the requirements on such a system, we posed the following questions in the introduction:

Questions

Q1. How can we track large amounts of sea ice globally?

Q2. How can we combine large scale (global) and small-scale (local) tracking?

Q3. How can we make sure only the relevant parts of the tracker is updated with new measurements?

Q4. How can we maximize the relevance of future acquired data?

Q5. How is the route affected if a hazard is found? How is the route affected if a hazard is **not** found?

Q6. How is the expected value of exploration affected by the modeled motion of objects?

These questions, in the backdrop of combining the MTT field and the UAS platform, led to the formations of the plans for a system capable of the full loop presented in Figure 9.1, and the formulations of the objectives from the introduction, restated below for convenience.

Objectives

I. Research scalable multi-target, multi-sensor methods for global tracking of ice objects.

II. Explore the use of machine learning to make use of the information in the tracking data.

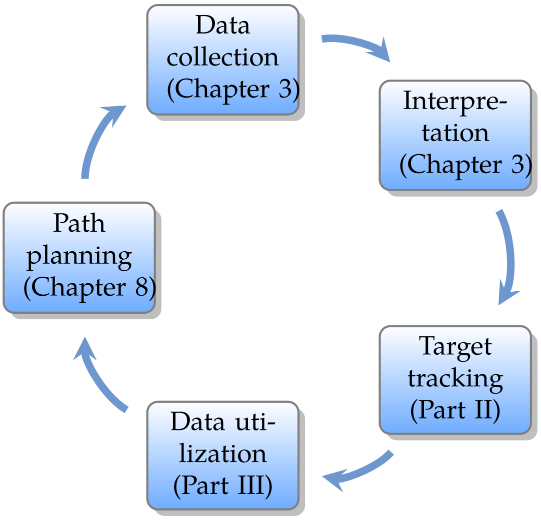

Data collection (Chapter 3)
Interpre-tation (Chapter 3)
Target tracking (Part II)
Data uti-lization (Part III)
Path planning (Chapter 8)

Figure 9.1: Circle of data collection and utilization

III. Improve the predictive capabilities through integrated mapping of currents.

IV. Develop a statistical representation of the geographical distribution of exploration value.

V. Optimize information acquisition through the exploitation of tracker data and statistical models.

We propose in this thesis a modular design of a system for global sea ice tracking, visualized in Figure 9.2. Contributions have been made to all principal components of this system in accordance with the objectives, such as

- exploration, improvement, implementations and application of two methods for MTT (Objective I);
- description of data extraction methods for Synthetic Aperture Radar (SAR) and TRI datasets (Objective II);
- current mapping from sea ice tracking data (Objective III); and
- a PHD based scheme for multi-agent IPP (Objectives IV and V).

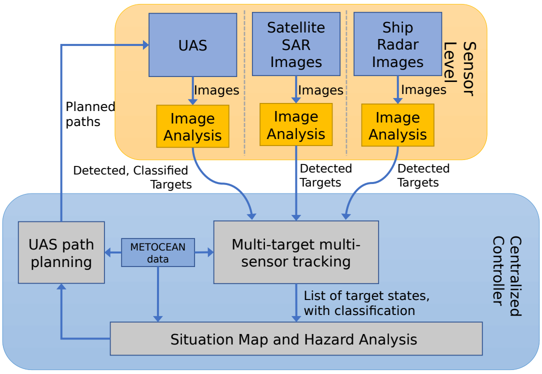


Figure 9.2: System overview

Machine learning was applied to extract detections of sea ice. This was exemplified using SAR data, and through model-based foreground extraction for detecting moving ice in TRI images. The data extraction served to demonstrate the application of Objective II in the research, and the fundamental role the detections play in target tracking.

Spatial indexing was considered as one of the schemes pursuing Objective I, which condenses the first three of the questions above. Spatial indexing was applied to both the MHT and the LMB algorithm. The performance gain of spatial indexing was studied and the algorithms were applied to large datasets — such as detections extracted from satellite imagery — demonstrating its applicability and scalability. Ultimately, the performance of the MHT filter was considered insufficient, and the research went on to focus on the LMB filter.

The LMB algorithm was studied in detail and improved through a, to our knowledge, novel formulation of the LMB filter which significantly simplifies its implementation.

Two utilities of the MTT tracking data were demonstrated in Part III. The current/wind velocity field estimation of Chapter 7 — demostrating the research carried out based on Objective III — could e.g. potentially be employed to improve the initialization of new targets in the MTT.

In Chapter 8, the concept of the PHD was brought, from the field of modern MTT, into the field of IPP. This was presented as an option for a common interfacing language between the tracker — that holds the collected observations — and the path planning algorithms that optimizes the value of continued data collection for controllable agents. This covers Objectives IV and V, which formalizes the three last questions in the list above.

The proposed path planning scheme is, specifically, based on the expected observed PHD, and it was applied to generate paths for multiple controllable UAS agents. A proof of concept Monte Carlo path planner, using full path sample proposals, was developed and applied to the aforementioned TRI dataset, as well as to a dataset collected with the UAS platform described in Section 3.5.3. In the context of Figure 9.1, this module is that which in the end brings it all together and closes the loop.

The implementations presented in this thesis are available online at https://github.com/jonatanolofsson/. Significant effort has gone into the implementation of each of the algorithms used in this thesis, and the encouragement to publish it as FOSS has been duly appreciated. The implementations show proof of concept for all the proposed algorithm improvements discussed in this thesis.

A spatially indexed MHT filter was presented in Chapter 5, followed by an LMB implementation in Chapter 6. The LMB algorithm was implemented first as a PYTHON module, then as an improved version in C++ with PYTHON bindings. The PYTHON implementation was used in the application of tracking drift sea ice in a dataset from a TRI. This highlights a successful cooperation with Norut, which further emanated in the applications presented in Chapters 7 and 8. The C++ LMB implementation was similarly employed in the applications of Chapter 8, where it interfaced with the Monte Carlo path planner implementation presented in the same chapter.

Each application presented in this thesis has been with real data, all but the satellite imagery collected in situ.

Given the presented implementations, what mainly remains for a real-life full demonstration is to connect the existing components and tune filtering parameters to suite the available sensors. For example, a global tracker could, theoretically, be instantiated through an effort to connect the European Space Agency (ESA) SENTINEL 1 data API — currently freely available online — to a module for data extraction — as per Section 3.7. This could then feed into a tracker module based on the proposed spatially indexed LMB algorithm and that would — when appropriately tuned — constitute a full demonstration of global sea ice tracking. In theory, nothing would then hinder the addition of local observations into the same LMB module as performed by e.g. UAS agents.

In fact, discussions like the above have taken place in the collaboration with Norut, in the context of including the implemented tracker into the visualization tools they already develop. The same interface would even allow the display of proposed optimized paths for the UAS pilots to consider when performing their flights.

Other future work that has been considered is for the target tracking to include estimation of parameters such as ice mass, shape etc. Potentially, this could improve the interpretation of how the currents and winds affect the motion of the object, and enable more advanced Single-Target Tracking (STT) motion models. Notably, any improvements of the underlying STT algorithm would improve MTT performance. This includes improved motion modeling as well as improved modeling of the measurement noise. In particular, the distortion introduced in the transformation of noise distributions has been largely ignored in this thesis, mentioned only in the context of the range/angle detections of the TRI.

A proper information loop study may be needed for the inclusion of velocity field estimates into the motion modeling of the tracker. In theory, a circular dependency is introduced which may have the effect of falsely reinforcing erroneous estimates. However, it may still work well in practice, or require only feasible changes to do so. Only using the estimated velocity to initialize newly detected targets would not introduce the same information loop.

The concept of the unobserved PHD introduced in Chapter 8 may also be studied further, in particular with respect to to the inclusion of motion modeling. While the basic time update was presented in the chapter, it remains to be properly implemented and utilized. Further, other path planning algorithms could also be explored for their applicability to the format of the proposed metric.

Bibliography

Albert, A., Leira, F. S., and Imsland, L. (2017). UAV Path Planning using MILP with Experiments. *Modeling, Identification and Control*, 38(1): 21–32. doi: 10.4173/mic.2017.1.3.

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*, volume 16. Prentice-Hall, Inc., Englewood Cliffs, New Jersey. doi: 10.1109/TSMC.1982.4308806.

Andersson, L. E. (2018). *Short-term Iceberg Drift Estimation and Prediction*. PhD thesis, Norwegian University of Science and Technology.

Anfinsen, S. N., Jenssen, R., and Eltoft, T. (2007). Spectral clustering of polarimetric SAR data with Wishart-derived distance measures. In *PolInSAR 2007: Proceedings of the 3rd International Workshop on Science and Applications of SAR Polarimetry and Polarimetric Interferometry*, Frascati, Italy.

Arkett, M., Flett, D., De Abreu, R., and Gillespie, C. (2006). Sea ice type and open water discrimination for operational Ice monitoring with RADARSAT-2. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1631–1634, Anchorage, Alaska, USA. doi: 10.1109/IGARSS.2006.421.

Bar-Shalom, Y. (1987). *Tracking and Data Association*. Academic Press Professional, Inc., San Diego, CA, USA.

Bar-Shalom, Y., Blackman, S. S., and Fitzgerald, R. J. (2007). Dimensionless Score Function for Multiple Hypothesis Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1): 392–400. doi: 10.1109/TAES.2007.357141.

Beckmann, N., Schneider, R., Seeger, B., Kriegel, H.-P., Schneider, R., and Seeger, B. (1990). The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. *SIGMOD Record*, 19(2): 322–331. doi: 10.1145/93605.98741.

Bentley, J. L. (1975). Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*, 18(9): 509–517. doi: 10.1145/361002.361007.

Bertsekas, D. P. (1988). The Auction Algorithm: A Distributed Relaxation Method for the Assignment Problem. *Annals of Operations Research*, 14(1): 105–123. doi: 10.1007/BF02186476.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. doi: 10.1117/1.2819119.

Blackman, S. S. (2004). Multiple Hypothesis Tracking for Multiple Target Tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1 January).

Blom, H. A. P. and Bloem, E. A. (2000). Probabilistic Data Association Avoiding Track Coalescence. *IEEE Transactions on Automatic Control*, 45(2): 247–259.

Bock, H. G. and Plitt, K. J. (1984). A multiple shooting algorithm for direct solution of optimal control Problems. In *Proceedings of the 9th IFAC world congress*, pp. 243–247, Budapest, Hungary. Pergamon Press.

Bottlik, I. P. and Blackman, S. S. (1989). Coordinated Presentation of Multiple Hypotheses in Multitarget Tracking. In *SPIE 1096, Signal and Data Processing of Small Targets*, pp. 152–159. doi: 10.1117/12.960350.

Brekke, E. and Chitre, M. (2018). Relationship between Finite Set Statistics and the Multiple Hypothesis Tracker. *IEEE Transactions on Aerospace and Electronic Systems*, 9251: 1–15. doi: 10.1109/TAES.2018.2805178.

Chekuri, C. and Pál, M. (2005). A recursive greedy algorithm for walks in directed graphs. In *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, volume 2005, pp. 245–253, Pittsburgh, PA, USA. doi: 10.1109/SFCS.2005.9.

Chen, S. G. (2014). Reduced recursive inclusion-exclusion principle for the probability of union events. In *IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 5–7, Bandar Sunway, Malaysia. doi: 10.1109/IEEM.2014.7058590.

Cloude, S. R. (2007). The Dual Polarisation Entropy / Alpha Decomposition. *Image Rochester NY*, pp. 1–6.

Cloude, S. R. and Pettier, E. (1996). A Review of Target Decomposition Theorems in Radar Polarimetry. *IEEE Transactions on Geoscience and Remote Sensing*, 34(2): 498–518. doi: 10.1109/36.485127.

Cloude, S. R., Pottier, E., and Boerner, W. M. (2002). Unsupervised image classification using the entropy/alpha/anisotropy method in radar polarimetry. In *NASA-JPL, AIRSAR-02 Workshop*, volume 44, pp. 04–06, Pasadena, CA, USA.

Coates, A. and Ng, A. Y. (2012). Learning Feature Representations with K-Means. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7700 LECTU: 561–580. doi: 10.1007/978-3-642-35289-8-30.

Collins, J. and Uhlmann, J. (1992). Efficient Gating in Data Association with Multivariate Gaussian Distributed States. *IEEE Transactions on Aerospace and Electronic Systems*, 28(3): 909 – 916. doi: 10.1109/7.256316.

Cox, I. J. and Hingorani, S. L. (1996). Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and its Evaluation for the Purpose of Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2): 138–150. doi: 10.1109/34.481539.

Crouse, D. F., Willett, P., and Bar-shalom, Y. (2011). Developing a Real-Time Track Display That Operators Do Not Hate. *IEEE Transactions on Signal Processing*, 59(7): 3441–3447.

Danchick, R. and Newnam, G. E. (2006). Reformulating Reid's MHT Method with Generalised Murty K-best Ranked Linear Assignment Algorithm. *IEE Proceedings - Radar, Sonar and Navigation*, 153(1): 13–22. doi: 10.1049/ip-rsn:20050041.

Dempster, A., Laird, N., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B Methodological*, 39(1): 1–38. doi: http://dx.doi.org/10.2307/2984875.

Dierking, W. and Pedersen, L. T. (2011). Monitoring Sea Ice using ENVISAT ASAR – A New Era Starting 10 Years Ago. Technical report, Alfred Wegener Institute for Polar and Marine Research.

Doulgeris, A. P., Anfinsen, S. N., and Eltoft, T. (2011). Automated K-Wishart clustering of POLSAR images. In *5th International Workshop on Science and Applications of SAR Polarimetry and Polarimetric Interferometry, PolinSAR 2011*, Frascati, Italy.

Duvenaud, D. (2014). *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge.

Eik, K. J. (2008). Review of Experiences within Ice and Iceberg Management. *Journal of Navigation*, 61(04): 557. doi: 10.1017/S0373463308004839.

Eik, K. J. (2009). Iceberg Drift Modelling and Validation of Applied Metocean Hindcast Data. *Cold Regions Science and Technology*, 57(2-3): 67–90. doi: 10.1016/j.coldregions.2009.02.009.

Eik, K. J. (2010). *Ice Management in Arctic Offshore Operations and Field Developments*. PhD thesis, Norwegian University of Science and Technology.

ESA (2013). *Sentinel-1 User Handbook*. European Space Agency.

Evers, K. U. and Weihrauch, A. (2004). Design and model testing of ice barriers for protection of offshore structures in shallow waters during winter. In *Proceedings of the 17th IAHR Symposium on Ice*, number June, pp. 21–25, St. Petersburg, Russia.

Fiorelli, E., Leonard, N. E., Bhatta, P., Paley, D. A., Bachmayer, R., and Fratantoni, D. M. (2006). Multi-AUV Control and Adaptive Sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering*, 31(4): 935–948. doi: 10.1109/JOE.2006.880429.

Fletcher, T. (2009). Support Vector Machines Explained. Technical report, University College London.

Galceran, E. and Carreras, M. (2013). A Survey on Coverage Path Planning for Robotics. *Robotics and Autonomous Systems*, 61(12): 1258–1276. doi: 10.1016/j.robot.2013.09.004.

Göbel, R. (2007). Towards Logarithmic Search Time Complexity for R-Trees. *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, pp. 201–206. doi: 10.1007/978-1-4020-6268-1-37.

Gonzalez, R. C. and Woods, R. E. (2008). *Digital Image Processing*. Pearson Prentice Hall, Upper Saddle River, NJ, USA, third edition.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings F - Radar and Signal Processing*, 140(2): 107. doi: 10.1049/ip-f-2.1993.0015.

Granström, K. and Lundquist (2012). *Extended Target Tracking Using PHD Filters*. PhD thesis, Linköping University, Linköping.

Gustafsson, F. (2018). *Statistical Sensor Fusion*. Studentlitteratur, 3 edition.

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, pp. 47–57, Boston, Massachusetts, USA. doi: 10.1145/602259.602266.

Hallikainen, M. (1992). Review of the microwave dielectric and extinction properties of sea ice and snow. In *IGARSS '92 International Geoscience and Remote Sensing Symposium*, volume 2, pp. 961–965, Houston, Texa. doi: 10.1109/IGARSS.1992.578309.

Hals, M. and Skjønhaug, M. (2017). *Optimization of Coordinated Path Planning for Autonomous Vehicles in Ice Management*. M.sc. thesis, Norwegian University of Science and Technology.

Hann, R., Wenz, A., Gryte, K., and Johansen, T. A. (2017). Impact of atmospheric icing on UAV aerodynamic performance. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems, RED-UAS 2017*, pp. 66–71, Linköping, Sweden. doi: 10.1109/RED-UAS.2017.8101645.

Haugen, J. (2014). *Autonomous Aerial Ice Observation*. PhD thesis, Norwegian University of Science and Technology.

Haykin, S. (1994). *Remote Sensing of Sea Ice and Icebergs*. A Wiley-Interscience publication. Wiley.

Henrich, A., Six, H.-W., and Widmayer, P. (1989). The LSD tree: spatial access to multidimensional point and non-point objects. In *The 5th Very Large Data Bases Conference*, pp. 45–54, Amsterdam, The Netherlands.

Hnatiuk, J. (1983). Exploration Methods in the Canadian Arctic. *Cold Regions Science and Technology*, pp. 181–193.

Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. (1997). *Global Positioning System: Theory and Practice*. Springer-Verlag.

Hol, J. D., Schön, T. B., and Gustafsson, F. (2006). On resampling algorithms for particle filters. In *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK. IEEE. doi: 10.1109/nsspw.2006.4378824.

Hollinger, G. A. and Sukhatme, G. S. (2014). Sampling-based Robotic Information Gathering Algorithms. *International Journal of Robotics Research*, 33(9): 1271–1287. doi: 10.1177/0278364914533443.

Hu, M.-K. (1962). Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8: 179–187. doi: 10.1109/TIT.1962.1057692.

IAMSAR (2010). *IAMSAR Manual: International Aeronautical and Maritime Search and Rescue Manual*. IMO.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. Mathematics in Science and Engineering. Elsevier Science.

Johannessen, K., Løset, S., and P., S. (1999). Simulation of iceberg drift. In *Proceedings 15th International conference on Port and Ocean Engineering under Arctic Conditions, POAC'99*, pp. 97–105.

Johannessen, O. M., Alexandrov, V. Y., Frolov, I. I. Y., Sandven, S., Pettersson, L. L. H. L., Bobylev, L. L. P., Kloster, K., Smirnov, V. G. V., Mironov, Y. Y. U., and Babich, N. N. G. N. (2007). *Remote Sensing of Sea Ice in the Northern Sea Route*. Springer. doi: 10.1007/978-3-540-48840-8.

Johansen, T. A. and Perez, T. (2016). Unmanned aerial surveillance system for hazard collision avoidance in autonomous shipping. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, Washington DC, USA.

Jonker, R. and Volgenant, A. (1987). A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems. *Computing*, 38(4): 325–340. doi: 10.1007/BF02278710.

Kaewtrakulpong, P. and Bowden, R. (2001). An Improved Adaptive Background Mixture Model for Real- time Tracking with Shadow Detection. *Advanced Video Based Surveillance Systems*, pp. 1–5. doi: 10.1.1.12.3705.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1): 35. doi: 10.1115/1.3662552.

Kim, C., Li, F., Ciptadi, A., and Rehg, J. M. (2015). Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4696–4704, Santiago, Chile. doi: 10.1074/JBC.274.42.30033.(51).

Kong, T. Y. and Rosenfeld, A. (1996). *Topological Algorithms for Digital Image Processing*. Machine Intelligence and Pattern Recognition. Elsevier Science.

Krause, A. and Guestrin, C. (2011). Submodularity and its Applications in Optimized Information Gathering. *ACM Transactions on Intelligent Systems and Technology*, 2(4): 1–20. doi: 10.1145/1989734.1989736.

Kuhn, H. W. (1955). The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2: 83–97.

Kurien, T. (1990). Issues in the Design of Practical Multitarget Tracking Algorithms. In Bar-Shalom, Y., editor, *Multitarget-Multisensor Tracking: Advanced Applications*, pp. 43–83. Artech House.

Langeveld, S. M. (2017). *Unmanned Aerial Vehicle Mission Planning for Combined Iceberg Detection and Tracking Missions*. Msc. thesis, Norwegian University of Science and Technology.

LaValle, S. M. (1998). Rapidly-Exploring Random Trees: A New Tool for Path Planning. Technical report, Department of Computer Science, Iowa State University.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, New York, NY, USA.

Lee, J.-S., Grünes, M. R., Ainsworth, T. L., Du, L. J., Schuler, D. L., and Cloude, S. R. (1999). Unsupervised Classification Using Polarimetric Decomposition and the Complex Wishart Classifier. *IEEE Transactions on Geoscience and Remote Sensing*, 37(5 pt 1): 2249–2258. doi: 10.1109/36.789621.

Leira, F. S. (2017). *Object Detection and Tracking With UAVs*. PhD thesis, Norwegian University of Science and Technology.

Leira, F. S., Johansen, T. A., and Fossen, T. I. (2017). A UAV ice tracking framework for autonomous sea ice management. In *International Conference Unmanned Aircraft Systems (ICUAS)*, Miami, USA.

Leppäranta, M. (2011). *The Drift of Sea Ice*. Geophysical Sciences. Springer, 2nd edition.

Li, X.-R. and Jilkov, V. (2003). Survey of Maneuvering Target Tracking. Part I: Dynamic Models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4): 1333–1364. doi: 10.1109/TAES.2003.1261132.

Lubin, D. and Massom, R. (2006). *Polar Remote Sensing, Volume I: Atmosphere and Oceans*. Springer Berlin Heidelberg. doi: 10.1007/3-540-30785-0.

Mahalanobis, P. C. (1936). On the Generalized Distance in Statistics. *Proceedings of the National Institute of Sciences of India*, pp. 49–55.

Mahler, R. P. S. (2003). Multitarget Bayes Filtering via First-Order Multitarget Moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4): 1152–1178. doi: 10.1109/TAES.2003.1261119.

Mahler, R. P. S. (2006). A theory of PHD filters of higher order in target number. In *Proceedings of SPIE*, number 6235, May 2006, pp. 1–12, Orlando, FL, USA. doi: 10.1117/12.667083.

Mahler, R. P. S. (2007a). PHD Filters of Higher Order in Target Number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4): 1523–1543. doi: 10.1109/TAES.2007.4441756.

Mahler, R. P. S. (2007b). *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., Norwood, MA, USA.

Mahler, R. P. S. and Zajic, T. (2001). Multitarget filtering using a multitarget first-order moment statistic. *Proceedings of SPIE*, 4380(August 2001). doi: 10.1117/12.436947.

Marchenko, A. (2010). Modelling of ice piling up near offshore structures Aleksey. In *20th IAHR International Symposium on Ice*, Lahti, Finland.

Miller, M. L., Stone, H. S., and Cox, I. J. (1997). Optimizing Murty's Ranked Assignment Method. *IEEE Transactions on Aerospace and Electronic Systems*, 33(3): 851–862. doi: 10.1109/7.599256.

Mitchell, M. (1996). An Introduction to Genetic Algorithms. *Computers & Mathematics with Applications*, 32(6): 133. doi: 10.1016/S0898-1221(96)90227-8.

Mladenova, I. E., Jackson, T. J., Bindlish, R., and Hensley, S. (2011). Incidence Angle Normalization of Backscatter Data. *IEEE Transactions on Geoscience and Remote Sensing*, 51(3): 1791–1804. doi: 10.1109/TGRS.2012.2205264.

Moran, K., Backman, J., and Farrell, J. W. (2006). Deepwater drilling in the Arctic Ocean ′ s permanent sea ice. In *Proceedings of the Integrated Ocean Drilling Program*, volume 302, pp. 1–13. doi: 10.2204/iodp.proc.302.106.2006.

Mountain, D. G. (1980). On Predicting Iceberg Drift. *Cold Regions Science and Technology*, 1: 273–282.

Murty, K. G. (1968). An Algorithm for Ranking all the Assignments in Order of Increasing Cost. *Operations Research*, 16(3): 682–687. doi: 10.1287/opre.16.3.682.

Ng, A. Y. (2015). CS229 Lecture notes: Support Vector Machines.

Oliver, C. and Quegan, S. (2004). *Understanding Synthetic Aperture Radar Images*. SciTech radar and defense series. SciTech Publ.

Olofsson, J., Brekke, E., Fossen, T. I., and Johansen, T. A. (2017a). Spatially indexed clustering for scalable tracking of remotely sensed drift ice. In *IEEE Aerospace Conference Proceedings*, Big Sky, MT, USA.

Olofsson, J., Brekke, E., and Johansen, T. A. (2017b). Cooperative remote sensing of ice using a spatially indexed labeled multi-Bernoulli filter. In *International Conference Unmanned Aircraft Systems (ICUAS)*, Miami, USA.

Olofsson, J., Flåten, A. L., Veibäck, C., and Lauknes, T. R. (2017c). Gaussian field current estimation from drift sea ice tracking with the labeled multi-Bernoulli filter. In *Proceedings of OCEANS 2017 MTS/IEEE*, Anchorage, Alaska, USA.

Olofsson, J., Hendeby, G., Lauknes, T. R., and Johansen, T. A. (2018). Multi-Agent Informed Path-Planning Using the Probability Hypothesis Density. *Autonomous Robots*, (Submitted, Aug 2018).

Olofsson, J., Veibäck, C., and Hendeby, G. (2017d). Sea ice tracking with a spatially indexed labeled multi-Bernoulli filter. In *20th International Conference on Information Fusion (FUSION)*, Xi'an, China.

Olofsson, J., Veibäck, C., Hendeby, G., and Johansen, T. A. (2017e). Outline of a system for integrated adaptive ice tracking and multi-agent path planning. *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems, RED-UAS 2017*. doi: 10.1109/RED-UAS.2017.8101636.

Olson, D. K. (1996). Converting Earth-Centered, Earth-Fixed Coordinates to Geodetic Coordinates. *IEEE Transactions on Aerospace and Electronic Systems*, 32(1): 473–476. doi: 10.1109/7.481290.

Onstott, R. G. (1992). SAR and Scatterometer Signatures of Sea Ice. In *Microwave Remote Sensing of Sea Ice*, pp. 73–104. American Geophysical Union. doi: 10.1029/gm068p0073.

Onstott, R. G. and Shuchman, R. a. (2004). SAR Measurements of Sea Ice. *SAR Marine User's Manual*, pp. 81–115.

Pasha, A., Vo, B.-N., Tuan, H. D., and Mat, W.-K. (2006). Closed form PHD filtering for linear jump Markov models. In *International Conference on Information Fusion (FUSION)*, Florence, Italy. doi: 10.1109/I-CIF.2006.301593.

Pearce, D. J. (2005). An Improved Algorithm for Finding the Strongly Connected Components of a Directed Graph. Technical report, Victoria University, Wellington, NZ.

Popp, R. L., Pattipati, K. R., and Bar-Shalom, Y. (2001). m-best S-D Assignment Algorithm with Application to Multitarget Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1): 22–39. doi: 10.1109/7.913665.

Rahmathullah, A. S., García-Fernández, Á. F., and Svensson, L. (2017). Generalized optimal sub-pattern assignment metric. In *20th International Conference on Information Fusion (FUSION)*, Xi'an, China. doi: 978-0-9964-5270-0.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Reid, D. B. (1979). An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6): 843–854. doi: 10.1109/CDC.1978.268125.

Reuter, S., Vo, B.-T., Vo, B.-N., and Dietmayer, K. (2014). The Labeled Multi-Bernoulli Filter. *IEEE Transactions on Signal Processing*, 62(12): 3246–3260. doi: 10.1109/LSP.2015.2414274.

Ribeiro, M. I. (2004). Gaussian Probability Density Functions : Properties and Error Characterization. Technical Report February, Institute for Systems and Robotics, Lisboa, Portugal.

Richards, J. A. (2009). *Remote Sensing with Imaging Radar*. Signals and Communication Technology. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-02020-9.

Roy, J., Duclos-Hindie, N., and Dessureault, D. (1997). Efficient Cluster Management Algorithm for Multiple-Hypothesis Tracking. *Signal and Data Processing of Small Targets*, 3163: 301–313. doi: 10.1117/12.279526.

Sarkka, S. (2013). Bayesian Filtering and Smoothing. *Cambridge University Press*, p. 254. doi: 10.1017/CBO9781139344203.

Scheuchl, B., Caves, R., Flett, D., De Abreu, R., Arkett, M., and Cumming, I. (2004). The potential of cross-polarization information for operational sea ice monitoring. In *Proceedings of the 2004 Envisat & ERS Symposium*, Salzburg, Austria.

Schubert, E., Zimek, A., and Kriegel, H.-P. (2013). Geodetic distance queries on r-trees for indexing geographic data. In *Proceedings of the 13th international conference on Advances in Spatial and Temporal Databases*, pp. 146–164, Munich, Germany.

Schuhmacher, D., Vo, B.-N., and Vo, B.-T. (2008). A Consistent Metric for Performance Evaluation of Multi-Object Filters. *IEEE Transactions on Signal Processing*, 56(8 I): 3447–3457. doi: 10.1109/TSP.2008.920469.

Sellis, T., Roussopoulos, N., and Faloutsos, C. (1987). The R+-Tree: A dynamic index for multi-dimensional objects. In *Proceedings of the 13th International Conference on Very Large Data Bases*, pp. 507–518, Brighton, England. doi: 645914.671636.

Sidenbladh, H. (2003). Multi-target particle filtering for the probability hypothesis density. In *Proceedings of the 6th International Conference of Information Fusion*, volume 2, pp. 800–806, Queensland Australia. doi: 10.1109/ICIF.2003.177321.

Singh, A., Krause, A., Guestrin, C., and Kaiser, W. J. (2009). Efficient Informative Sensing using Multiple Robots. *Journal of Artificial Intelligence Research*, 34: 707–755. doi: 10.1613/jair.2674.

Skoglar, P. (2012). *Tracking and Planning for Surveillance Applications*. PhD thesis, Linköping University.

Smith, G. L., Schmidt, S. F., McGee, L. A., Aeronautics, U. S. N., and Administration, S. (1962). *Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle*. NASA technical report. National Aeronautics and Space Administration.

Sørensen, K. L. (2016). *Autonomous Icing Protection Solution for Small Unmanned Aircraft*. PhD thesis, Norwegian University of Science and Technology.

Sørensen, K. L., Helland, A. S., and Johansen, T. A. (2015). Carbon nanomaterial-based wing temperature control system for in-flight anti-icing and de-icing of unmanned aerial vehicles. *IEEE Aerospace Conference Proceedings*, 2015-June. doi: 10.1109/AERO.2015.7119206.

Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149*, volume 2, pp. 246–252. doi: 10.1109/CVPR.1999.784637.

Streit, R. (2017). How I learned to stop worrying about a thousand and one filters and love analytic combinatorics. In *IEEE Aerospace Conference Proceedings*, Big Sky, MT, USA. doi: 10.1109/AERO.2017.7943644.

Tarjan, R. (1972). Depth-first Search and Linear Graph Algorithms. *SIAM Journal on Computing*, 1(2): 146–160.

Thain, C. (2014). Sentinel-1 Level 1 Detailed Algorithm Definition. Technical report.

Thenkabail, P. S. (2015). *Remote Sensing of Water Resources, Disasters, and Urban Studies*. Remote Sensing Handbook. CRC Press.

Valavanis, K. P. (2007). *Advances in Unmanned Aerial Vehicles*. Springer Netherlands. doi: 10.1007/978-1-4020-6114-1.

Veibäck, C., Olofsson, J., and Hendeby, G. (2018). Learning Target Dynamics While Tracking Using Gaussian Processes. *Transactions on Aerospace and Electronic Systems*, (Under review, June 2018).

Vo, B.-N. and Ma, W. K. (2006). The Gaussian Mixture Probability Hypothesis Density Filter. *IEEE Transactions on Signal Processing*, 54(11): 4091–4104. doi: 10.1109/TSP.2006.881190.

Vo, B.-N., Vo, B.-T., and Phung, D. (2014). Labeled Random Finite Sets and the Bayes Multi-Target Tracking Filter. *IEEE Transactions on Signal Processing*, 62(24): 6554–6567. doi: 10.1109/TSP.2014.2364014.

Vo, B.-T. and Vo, B.-N. (2013). Labeled Random Finite Sets and Multi-Object Conjugate Priors. *IEEE Transactions on Signal Processing*, 61(13): 3460–3475. doi: 10.1109/TSP.2013.2259822.

Vo, B.-T., Vo, B.-N., and Cantoni, A. (2006). The cardinalized probability hypothesis density filter for linear Gaussian multi-target models. In *2006 40th Annual Conference on Information Sciences and Systems*, pp. 681–686, Princeton, New Jersey, USA. doi: 10.1109/CISS.2006.286554.

Vo, B.-T., Vo, B.-N., and Cantoni, A. (2007). On multi-Bernoulli approximations to the Bayes multi-target filter. In *Proceedings of the 10th International Conference on Information Fusion, FUSION 2007*, Québec, Canada.

Vo, B.-T., Vo, B.-N., and Cantoni, A. (2009). The Cardinality Balanced Multi-Target Multi-Bernoulli Filter and its Implementations. *IEEE Transactions on Signal Processing*, 57(2): 409–423. doi: 10.1109/TSP.2008.2007924.

Voytenko, D., Dixon, T. H., Luther, M. E., Lembke, C., Howat, I. M., and de la Peña, S. (2015). Observations of Inertial Currents in a Lagoon in Southeastern Iceland using Terrestrial Radar Interferometry and Automated Iceberg Tracking. *Computers and Geosciences*, 82: 23–30. doi: 10.1016/j.cageo.2015.05.012.

Werner, C., Wiesmann, A., Strozzi, T., Kos, A., Caduff, R., and Wegmuller, U. (2012). The GPRI multi-mode differential interferometric radar for ground-based operations. In *9th European Conference on Synthetic Aperture Radar*, pp. 304–307, Nuremberg, Germany.

Wiener, N. (1965). *Cybernetics, Second Edition: Or the Control and Communication in the Animal and the Machine*. The MIT Press.

Williams, J. L. (2015). Marginal Multi-Bernoulli Filters: RFS Derivation of MHT, JIPDA, and Association-based MeMBer. *IEEE Transactions on Aerospace and Electronic Systems*, 51(3): 1664–1687. doi: 10.1109/TAES.2015.130550.

Wills, A. G., Hendriks, J., Renton, C., and Ninness, B. (2017). A Bayesian Filtering Algorithm for Gaussian Mixture Models. *ArXiv e-prints*, pp. 1–17.

Winter, C. L. and Stein, M. C. (1993). An Additive Theory of Probabilistic Evidence Accrual. *Advances in Applied Mathematics*.

WMO (1970). *WMO Sea-ice Nomenclature: Terminology, Codes, Illustrated Glossary and Symbols*. Number v. 4 in WMO sea-ice nomenclature. Secretariat of the World Meteorological Organization.

WMO (2014). Sea Ice Nomenclature. Technical Report WMO-259, Joint Technical Commission for Oceanography and Marine Meteorology.

Wright, B. (2000). Full Scale Experience with Kulluk Stationkeeping Operations in Pack Ice (With Reference to Grand Banks Developments). Technical report, National Research Council of Canada. doi: 10.4224/12327366.

Zakhvatkina, N. Y. and Bychkova, I. A. (2015). Bayesian Classification of the Ice Cover of the Arctic Seas. *Izvestiya, Atmospheric and Oceanic Physics*, 51(9): 883–888. doi: 10.1134/S0001433815090212.

Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition*, Cambridge, UK. doi: 10.1109/ICPR.2004.1333992.