NTNU

Norwegian University of
Science and Technology

Harbor Surveillance with a K-best, Track Terminating, Hypothesis-Oriented MHT

Jesper Pedersen

Master of Science in Cybernetics and Robotics
Submission date: July 2018
Supervisor: Edmund Førland Brekke, ITK
Co-supervisor: Egil Eide, IES

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

A high activity harbor environment presents several problems for tracking systems. The observable area may be small, targets may be closely spaced, move in formation and disappear close to where other appears. Therefore, to perform harbor surveillance one requires a tracking algorithm capable of both robust track initiation, maintenance and termination.

To deal with this, the Hypothesis-Oriented Multiple Hypothesis Tracker (HO-MHT) of Reid is extended to also model the event of targets ceasing to exist. The result is similar to that of Kurien, but is formulated such that the K-Best hypotheses can be generated in polynomial time by the algorithm of Murty. Important design aspects of the algorithm are explained, including track and hypothesis tree data structures, together with additional complexity reduction techniques aside from K-best generation.

The surveillance system is to be used for the autonomous ferry project in Trondheim, Norway, and the objective is to gather statistics for the area the ferry is going to operate in. This is important to both evaluate the impact of the ferry on the existing traffic, and to aid in the development of a collision avoidance system.

A high performing radar detection system is developed, removing 40 % of the clutter, and under 1 % of the true target detections. This includes a filter to remove "multiples" generated by large boats. Measurement and plant noise parameters for a DWNA-model are estimated for the sensor and the targets that operate in the area. Both the optimal parameters of the detection system and the target model is obtained by evaluating it against an approximate ground truth.

Two Graphical User Interfaces (GUI) have been made. One facilitates the creation of a ground truth. The other is made specifically to analyze the results of the tracking system and the MHT. It is possible to examine individual clusters and its hypotheses through time. The tracks of a hypothesis are shown on top of a satellite image along with detection data.

Sammendrag

Et havnemiljø med høy aktivitet kan gi flere problemer for et tracking-system. Det observerbare området er gjerne lite, målene kan være tett fordelt, bevege seg i formasjon og forsvinne nær der andre viser seg. Derfor krever havneovervåking en tracking-algoritme som er i stand til robust track-initiering, -vedlikehold og -avslutning.

For å håndtere dette, er Reids Hypothesis-Oriented Multiple Hypothesis Tracker (HO-MHT) utvidet til å også modellere inn at mål kan slutte å eksistere. Resultatet ligner det til Kurien, men er formulert slik at de K-beste hypotesene kan genereres i polynomisk tid med algoritmen til Murty. Viktige designaspekter av algoritmen er forklart, inkludert track- og hypotese-trestrukturer, sammen med flere kompleksitetsreduksjonsteknikker i tilegg til generering av de K-beste hypotesene.

Overvåkningssystemet skal brukes i det autonome fergeprosjektet i Trondheim, og målet er å samle statistikk for det området fergen skal operere i. Dette er viktig for både å evaluere fergens påvirkning på eksisterende trafikk, og å bistå i utviklingen av et system for å unngå kollisjoner.

Et radardeteksjonssystem med høy ytelse er utviklet, og fjerner 40% av falske deteksjonene og under 1 % av de sanne. Dette inkluderer et filter for å fjerne "multipler" generert av store båter. Måling og modelstøy for en "DWNA"-modell er estimert for sensoren og målene som opererer i området. Både de optimale parametrene til deteksjonssystemet og bevegelsesmodellen til målene er oppnådd ved å evaluere de mot en tilnærming til faktiske tracks. Et grafisk brukergrensesnitt (GUI) har blitt laget for å både legge til rette for opprettelsen og øke presisjonen til disse trackene.

En annen GUI er laget for å analysere resultatene til tracking-systemet. Det inkluderer muligheter til å undersøke individuelle "clusters" og dens hypoteser for forskjellige tidspunkt. Trackene til en hypotese vises over et satellittbilde sammen med deteksjonsdata og ved siden av en kameravisning.

Preface

The work described in this thesis is carried out in the Department of Engineering Cybernetics at the Norwegian University of Science and Technology. It concludes a 5-year program to obtain a Master of Science degree in Cybernetics and Robotics. All work was done in the spring of 2017, the ultimate semester of the program.

The report is written assuming that the reader has a background in cybernetics, including knowledge of statistics, the Kalman Filter and data structures. Some knowledge of basic image processing is also expected.

I would like to thank my academic supervisors Edmund Førland Brekke and Egil Eide for their excellent assistance with the thesis. Their interest in my academic learning and the project itself has given me motivation, while their expertise knowledge have helped me in the right direction.

Trondheim, 2018-07-02

Jesper Pedersen

Contents

List of Figures

List of Tables

Acronyms

CPHD Cardinalized PHD.

FISST Finite Set Statistics.

HO-MHT Hypothesis-Oriented MHT.

JPDAF Joint Probalistic Data Association Filter.

MDA Multi-Dimensional Assignment.
MHT Multiple Hypothesis Tracking.
MTT Multi-Target Tracking.

PDAF Probabilistic Data Association Filter.
PHD Probability Hypothesis Density.

RFS Random Finite Sets.

TO-MHT Track-Oriented MHT.

1 | Introduction

This thesis has two main contributions. The primary task was to gather statistics for the autonomous ferry project in Trondheim, Norway. In the process, an extension of the Hypothesis Oriented Multiple Hypothesis Tracker (HOMHT) has been made, which models in track termination, and contributes independently of the original problem.

This chapter is an introduction to the primary task, while chapter two will serve as an introduction to the MHT with Track Termination. For further overview of the structure of the thesis, the reader is referred to the outline at the end of this chapter.

1.1 Introduction to the Autonomous Ferry Project

Researchers at NTNU are currently developing an autonomous ferry designed for carrying pedestrians and cyclists across the canal between Ravnkloa and Brattøra in the city of Trondheim, Norway.

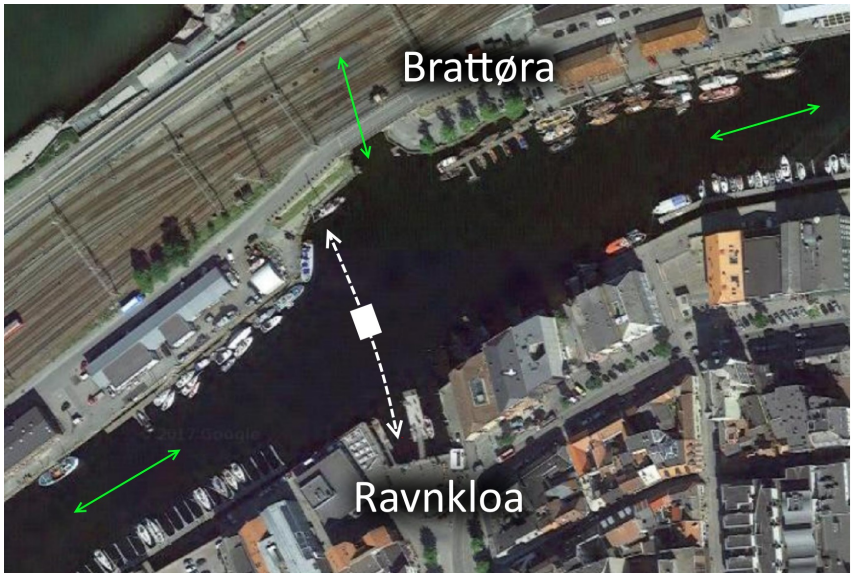


Figure 1.1: The canal between Ravnkloa and Brattøra. Green arrows mark entry/exits for the canal intersection.

One is in need of connecting the two districts closer together. Ravnkloa is part of the city center of Trondheim, close to different stores, restaurants and other attractions. Brattøra was formerly an industrial area with a dock. Later, the industrial activity was ended and the area was forgotten for a few years. Now, the former dock has been replaced with cafés, a hotel is placed nearby, and the area is undergoing a change. The closest bridge to pass the canal is 300 meters away, and therefore one was in need of a new way to get pedestrians over the canal to connect Brattøra and the city center closer.

However, a new bridge was considered too much of an impact on the area and the boat traffic in the canal. Therefore, the idea of an autonomous ferry was presented which will serve pedestrians "on demand", in much similarity to an elevator.

The canal is heavily used by marine vehicles, ranging from large boats to jet skis and canoes. It is important that the ferry interferes as little as possible with the original boat traffic. Therefore, one is need of gaining a better understanding of the traffic in the canal. This includes everything from the number of boats at different times, what kind of boats that tends to be present, where they are heading, and their typical velocities.

Since most of these boats do not use the Automatic Identification System (AIS), it is necessary to monitor the traffic using shore-bound exteroceptive sensors such as radar and camera to build up such statistics. Further ahead in the future, shore-bound sensors will also be used to aid the Collision Avoidance (COLAV) system of the autonomous ferry. For this purpose one is in need of tracking statistics, like measurement errors, target speed and acceleration, distribution of clutter and also how targets tend to resolve collision avoidance situations.

Therefore, a surveillance station with a marine radar and a camera has been placed at the shore of Brattøra. It is capable of unsupervised collection of large amounts of data. However, analyzing large amounts by hand is tedious and one is in need of a tracking system to automatically gather information.

To gather the statistics needed, the tracking system must be able to initiate, maintain and terminate tracks. The observable area is expected to be crowded at times, with targets at close proximity, which are known to present difficulties for a tracking system.

1.2 Previous Work

During the fall of 2017, the author worked on the same project. The surveillance station was set up, and the camera was assessed to be of little use for tracking, but good for validating radar data and tracking results. Due to lack of easy power access, only small amounts of data was collected. In addition, an error was made leading to a sampling time of 20 seconds for the radar data.

This made it hard to assess the possibilities of an effective tracking system.

Still, work was done in the detection system of the radar, where large amounts of false detections were removed by using image masks. These techniques are carried on from the original problem, and will be described further on. A Probabilistic Data Association Filter (PDAF) was implemented and used as a tracking algorithm. It was concluded that it was insufficient, due to the lack of estimating the correct number of targets and track coalescence. Targets were seen to have close proximity in the canal and to move together in groups. This both led to the track coalescence and it was also seen that a great amount of detections from targets were being merged. The previous project was coded in *MATLAB*, but it proved insufficient in speed, memory use and code flexibility, and therefore, for this project, Python has been chosen.

1.3 Problem Formulation

The project involved the following tasks:

1. Extend the single-target tracker from the specialization project to a multi-target tracker, with particular focus on challenges relating to merged measurements and track coalescence.

2. Test the tracking system on progressively longer sequences of radar scans in order to qualify its capabilities as the core of data interpretation in the harbor surveillance system.

3. Generate statistics on the number of vessels, their entry points and their exit points.

4. Generate statistics on paths/trajectories of vessels in the channel. Investigate the possibilities of using these statistics for long-term prediction of vessel trajectories.

5. Develop an interface that allows the user to quickly get access to camera data of the scene according to a variety of queries.

For reference, there were additional tasks on this list, as one was uncertain on the comprehensiveness of the tasks and the time necessary to complete them. They are expected to be tasks if the project is continued.

6. Develop methods for identifying situations where interaction between vehicles, e.g. COLAV maneuvers, find place.

7. Include automatic detection in camera data, fusion between camera and radar as well as automatic classification of vessels.

1.4 Contribution

The main contributions of the project are listed below:

- A sophisticated radar detection system has been made, removing large amounts of false detections and only a few true ones.
- A K-Best HO-MHT with track termination has been formulated, running in polynomial time. The algorithm is implemented in Python, using low memory and having capabilities of analyzing large amounts of data.
- A GUI has been made for creating a ground truth. It has been used to create 40 hours of ground truth data.
- A GUI has been made for examining tracking results of the system, specifically designed for MHT with track termination.
- Evaluation and statistical tools for evaluating tracking performance has been implemented, including the OSPA metric.
- Measurement and plant noise parameters for a Discrete White Noise Acceleration-model have been analyzed and decided upon to give a smallest possible track gate. Parameters for an optimal target state initialization has also been found.

1.5 Outline

The thesis is organized in the following matter. Chapter 2 gives an introduction and overview of the tracking problem. It also presents some of the algorithms that were investigated and considered to be used for the surveillance. The discussion of the algorithms serves as the basis for Chapter 3 where the MHT with Track Deletion is presented. Along with the derivation of the algorithm, a solution for generating the K-best hypotheses is given, as well as a clustering scheme, pruning techniques and a suggested implementation.

Chapter 4 presents the underlying part of the tracking system, including radar settings used and the detection system. All parameters in the tracking system are discussed in relation to this particular case. Nonetheless, the discussion of the parameters concerning the MHT are general enough to be applicable for other cases as well. In chapter 5 the three GUIs that have been created are presented. Chapter 6 presents the results of the thesis. This includes a ground truth for a subset of the data, created "by hand". It is used for both identifying parameters and for reviewing the performance of the MHT. Lastly, the MHT is used on the rest of dataset to gather track data and resulting statistics. Chapter 6 discusses the results, while Chapter 7 concludes on the project as a whole and suggests on further work.

2 | The Tracking Problem

The aim of this section is to present for the reader the general concepts and problems concerning multi-target tracking. Also, it will define some of the terminology to be used further in this thesis. At last, some established tracking methods will be briefly discussed.

2.1 Objective

In general, the objective of tracking is to observe a physical space, attempting to identify targets of interest, and estimate their position and motion. However, tracking is a vast term and there may be other objectives in together with these, depending on the purpose of the tracking. Two common purposes are collision avoidance and surveillance.

A target of interest may be airplanes, boats or people, but may also include birds or similar. In general, a target may be anything in the physical space that stands out from its surroundings. A common way to observe a physical space is with exteroceptive sensors, and this is the type of sensor assumed in tracking literature.

2.2 Exteroceptive sensors

Exteroceptive sensors observe the space around them, either passively (camera, etc.) or actively (radars, sonars, LIDARs). The objective of the sensor and its surrounding system is to deliver measurements assumed to originate from targets. The following pipeline is typical for a exteroceptive sensor system.

1. Observe energy
2. Signal processing
3. Thresholding
4. Extraction of measurements

The last two steps are often referred to as detection and is a process heavily

dependent upon both the sensor, the observable area, targets and the tracking algorithm to be used.

Usually, an exteroceptive sensor system obtains measurements in batches, and it is assumed that the measurements are obtained at approximately the same time. A set of measurements obtained at the same time is called a scan. Each measurement includes information about the position of the assumed target, but some sensors may also be able to supply additional information about velocity, shape or other attributes in addition to the location.

One of the main problems presented by exteroceptive sensors is the origin uncertainty of measurements. There are three possible origins of a measurement:

- No target. Either internal (sensor) or external (environmental) noise could lead to a measurement not belonging to a target. The measurement is often referred to as a false measurement or "clutter".
- One target. This is the ideal case, and is referred to as a resolved measurement.
- Several targets. Limited resolution of the sensor and/or noise may lead to several targets only producing one single measurement. Referred to as a merged measurement.

Furthermore, usually it is not only desired to know if a measurement originated from a target, but also from which target it originated from. A measurement could both originate from some previously seen target, and one could therefore associate a measurement with a set of previous ones, or it could originate from a new target appearing in the observable area of the sensor.

In addition, if a target is present in the observable space around the sensor, one has two possible errors concerning this target:

- A measurement is not obtained of the target. This is usually referred to as a missed detection of the target.
- The target produces several measurements. For some sensors with high resolution it may be expected that a target produces several measurements, and it may also be desired to be able to estimate target size. However, when this is the case, it is often trivial to know that a group of measurements all belong to the same target, for instance due to the close proximity of the measurements. Anyhow, there is a possibility that one target produces several measurements (or groups of measurements), which appears to originate from more than one target. This is referred to as split measurements.

Therefore, a measurement may originate from no target, one or several, while a target may produce no measurement, one or many. The ideal case is of course a one-to-one measurement-to-target relationship.

However, the ideal case is not trivial, since the problem of knowing from which target the measurement originated from still persists. This problem is greatly

affected by the time step of the sensor. A time step that approaches zero would deem the problem trivial, as measurements could easily be associated with the closest one in the previous time step.

In addition, an error in the measurement itself is present. Assuming one knows the origin of every measurement, an error between the position of the measurement and the true target location is expected, because of noise and accuracy limitations of the sensor.

2.3 Tracking Algorithms

To overcome the problem of the origin uncertainty and missed detections, it is a necessity to use scans from one or multiple sensors over time.

Usually the term "tracking" refers to what is considered classical tracking-algorithms. Classical tracking algorithms are able to rely on only position measurements to perform the tracking, although many are able to use additional detection information as velocity to improve on it. This differs from "visual tracking"-algorithms where color information and a detailed shape are the main features used to associate a measurement with a previous one. Further on this thesis, classical tracking algorithms are the ones being discussed. Interestingly, these are also proven to be useful in visual tracking scenarios [5].

The concept of a track is of concern, as it is a concept loosely defined in literature. Some may define a track as a set of measurements assumed to belong to the same target, but this is not intuitive for algorithms that does not assign measurements explicitly to targets. Therefore, defining a track in terms of target state estimates is more appropriate. Also, a set of track estimates may be acquired, but some or all of the estimates may be far off from actually corresponding to the path of an actual target. Therefore, the following definition is made:

A track is an ordered sequence of discrete-time state estimates for what is estimated to be a target.

Also, some additional terminology relating to a track are defined.

- *True track*: The actual track of a target in the observable area of the sensor.
- *Track error*: The error between position estimates and the true target location.
- *Track loss*: Acceptable estimates are not acquired for the whole time the target is present in the observable area.
- *False and partly false track*: All, or some of the estimates of the track are present when a target is not.
- *Correct track*: There are no track loss, no false track estimates and the

track error is within reasonable measure for all estimates.

Not all tracking algorithms constructs tracks, but rather give a set of point estimates for targets at each timestep. Specifically, this is the case for some of the FISST based algorithms, which will be discussed in the Section 2.4.

Tracking algorithms are either single-target or multi-target algorithms. As indicated by the names, single-target algorithms assume only one target to be present at the same time, while for multi-target there may be several.

Further, tracking algorithms can be categorized into single-frame and multi-frame ones. Single-frame algorithms decides on a single "tracking result" at every timestep, while multi-frame algorithms stores several outcomes, postponing its decision on what is the correct one. The different outcomes of a multi-frame algorithm usually have a score or probability, which gets updated as additional scans are received.

2.3.1 Performance Metrics for Multi-Target Tracking

The performance of tracking algorithms may be evaluated by examining results. However, a more systematic way is to use a metric to evaluate them. A metric measures the error between the tracking algorithm output and a ground truth, and is particularly useful when comparing different algorithms.

Cardinality

A simple measure of performance is the *cardinality* of the set of target estimates at each iteration, i.e. how well the algorithm estimates the number of targets, $\bar{N}_k$. In this case one does not consider the actual value of the estimates. Since this is simply a deviation between two numbers, the cardinality of the tracker and the ground truth, various metrics may be used, typical ones are Mean Absolute Error (MAE) and root-mean-square error (RMSE). The latter penalizes outliers more heavily than the former, and is given for a set of scans $k \in [0, T]$ as

$$e_{card}^{RMSE} = \sqrt{\frac{1}{T}\sum_{k=1}^{T}(\bar{N}_k - N_k^{true})^2} \tag{2.1}$$

OSPA

The Optimal Subpattern Assignment (OSPA)-metric penalizes both cardinality errors and errors in state estimates [22]. At a certain timestep, the set of target state estimates are compared to the set of true target state estimates.

The metric finds the best match of estimated targets to the true estimates. If the sets are of different cardinality, it is penalized and the best matching subset of the larger set is found. The error of the states are penalized with a distance metric, for instance the euclidean distance. However, if the distance value is above a threshold, called the cut-off distance, this is used as the distance value. Practically, the cut-off distance decides the error in which one decides that two target states can not be associated, and the difference between the states are penalized as much as a cardinality error. At a certain timestep the metric is given as

$$\bar{d}_p^{(c)}(X,Y) = \begin{cases} \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n-m) \right) \right)^{\frac{1}{p}} & \text{if } m \leq n \\ \bar{d}_p^{(c)}(Y,X) & \text{if } m > n \\ 0 & \text{if } m = n = 0 \end{cases} \tag{2.2}$$

where $d^{(c)} = \min(c, d(\mathbf{x}, \mathbf{y}))$, c is the cutoff distance. X and Y are the set of target estimates for the ground truth and the tracking algorithm output, with cardinality m and n, respectively. p is similar to that of a p-order metric, and from a practical point of view, it will penalize outliers more if its increased. Π_n is the set of all permutations of Y in m.

For a whole dataset, the mean of the OSPA values for each timestep may be used. This is reasonable as it already penalizes outliers with the p parameter.

A limitation of OSPA is that it does not consider tracks, but only the set of state estimates at a single timestep. Therefore, if two tracks get mixed up, meaning that all estimates of one track after a time k are acceptable estimates for the other track and vice versa, the OSPA will not penalize it.

2.4 Multi-Target Tracking Algorithms

Multi-Target Tracking (MTT) algorithms can roughly be categorized into two different approaches to solving the tracking problem: Data association based approaches and approaches based on FISST. These two approaches will be presented and a selection of algorithms will be discussed, mostly in terms of advantages and disadvantages.

2.4.1 Data Association Filters

The data association based MTT-algorithms associates specific measurements to targets, and usese a filter to update its state. The most popular algorithms

are the Joint Probalistic Data Association Filter (JPDAF) and the two main variations of Multiple Hypothesis Tracking (MHT).

JPDAF

The JPDAF assumes targets are known and its state are initialized. When updating a target state all measurements inside a validation region of the predicted position of the state are considered. This region are referred to as a track gate. It is assumed that the target generated at most one measurement, however no measurement is picked explicitly, and rather the gated measurements are combined in a single statistically most probable update. The update takes into account nearby tracks and the statistical distribution of misdetections and clutter.

A drawback to JPDAF is that it assumes targets exists and does not include track initiation or track termination. However, by using a scheme like M/N-initiation/termination in parallel with the filter these capabilities may be achieved [1]. M/N initiation creates a track when two consecutive measurements, not inside the validation region of any existing tracks, are in close vicinity. The resulting track is maintained by a separate JPDAF to not interfere with the probability of confirmed tracks. If the track successfully gates measurements in at least M of N time steps, it is declared a confirmed track and moved to the JPDAF for confirmed tracks. If not, it is removed. M/N-termination works in similar fashion. If a confirmed track gates a measurement in at least M out of the N last time steps it is kept, if not it is terminated. The M/N values can be set to assure only reasonable tracks are confirmed, but this may be unsuited for targets that are present in the observable area for only a short time. In addition, if a target appears inside the validation region of a confirmed target, the measurement is not used by the scheme. One may remove this requirement, but then one would end up with duplicate tracks interfering with each other, which would be even worse. This leads to M/N-initiation not being suited for problems where targets appear close to each other.

Another drawback to JPDAF is its approximation in combining all measurements in its update. As will be shown when discussing MHT, it is possible to update a track with each measurements separately and later decide upon which actually was the correct one. This will lead to more accurate estimation, though the accuracy of JPDAF may well be sufficient for many applications.

Track coalescence is another problem of JPDAF. If two targets are in close proximity of each other they could validate and use the same measurements to update its state. If two targets remain close to each other for some time, this will eventually lead to the tracks being merged, and to errors in estimation [7]. There exists improvements to JPDAF that attempts to deal with this problem [3][9].

MHT

Multiple Hypothesis Tracking are in general used to describe two similar multi-frame data association algorithms, the hypothesis-oriented MHT (Hypothesis-Oriented MHT (HO-MHT)) and the track-oriented MHT (Track-Oriented MHT (TO-MHT)). There exists an algorithm called Probabilistic MHT, but it will not be considered here. The approach of HO-MHT and TO-MHT is to explicitly define all possible associations between targets and measurements, resulting in a set of mutually exclusive outcomes called hypotheses. The idea is that while a hypothesis may be unlikely at first, as new scans are received it may end up being the most likely.

Hypothesis Orientated MHT

Singer, Sea and Housewright first introduced the idea of propagating multiple hypotheses for a single target with clutter present [24]. However, Reid first developed a complete algorithm and framework, also assuming multiple targets could be present [21]. This framework were originally called just MHT, but are later referred to as HO-MHT.

The algorithm enumerates all possible data-associations of measurements, and a specific enumeration is called a hypothesis. A measurement is either classified as clutter, classified as a new target or assigned to a single existing target implied by the previous hypothesis. By this fashion, hypotheses are created recursively by enumerating all data-associations for the measurements in a scan, for all previous hypothesis. The probability of a hypothesis is also evaluated recursively as the joint probability of the parent hypothesis and the specific enumeration of the current scan. It is assumed that clutter and new born targets occur according to a Poisson point process with constant intensity, and that they are uniformly distributed over the observable region. Also, it is assumed that targets generate at most one measurement per scan and are detected with a constant probability of detection.

HO-MHT does not include track termination as the event of a target ceasing to exist is not modelled. The result is that, with time, the most likely hypothesis will become the one assuming that all measurements are clutter. The hypotheses that assume a target to be present will model the target as misdetected for every timestep after the target cease to exist. This makes the other hypotheses, which assumes no new targets, become increasingly probable. An extension to HO-MHT that includes the event of targets ceasing to exist will be presented in Chapter 3.

Merged and split measurements are not accounted for explicitly by the algorithm. However, neither case should be an issue if they are not too frequent. In the case of a merged measurement, the measurement can only be assigned to one of the targets that produced it and the rest of the targets will simply be modelled as misdetected. If merged measurements of the targets persists this

could lead to track loss [10]. One should consider decreasing the probability of detection to account for the misdetections caused by merged measurements. In the case of a split measurement, one of the measurements in the split will be assigned to the target, and the other measurements of the split will be classified as clutter. Here, one should consider increasing the clutter density if split measurements are frequent.

MHT has a reputation of being too expensive in both memory and computation. The number of hypotheses increase exponentially with the number of measurements, and it is necessary to prune unlikely hypotheses to make the algorithm run for even small problems. However, it has been shown that the K-best hypotheses can be generated using the algorithm of Murty and runs in polynomial time $\mathcal{O}(K^2(m+n)^3)$, where m and n are the number of measurements in the scan and the number existing targets in the hypothesis respectively [17]. Techniques like clustering and additional pruning strategies could further decrease the computation time.

In terms of tracking performance MHT is considered to be one of the best, although thorough studies to back this up is hard to find. It is shown to be superior to JPDA even with quite extensive pruning [8]. In general its performance will largely depend on the amount of pruning, as one would expect it to be optimal if all hypotheses are generated.

Track Oriented MHT

Track Oriented MHT was created as a more lightweight alternative to HO-MHT. The idea was to take advantage of the number of tracks in HO-MHT being far less than the number of hypotheses.

In Track-Oriented MHT targets are assumed to be known and its tracks initialized. At every scan, each track are split into different track-hypotheses for each of the measurements in the validation region of the track, and in addition a track-hypothesis for the possibility that the target was misdetected. The result is that an initialized target and its track serves as the root node of a tree, and as scans are received the tree expands. Each leaf node represents an alternate track for the initialized target, and is assigned a score. Track nodes grow exponentially with measurements and pruning needs to be applied in similar fashion to HO-MHT.

A hypothesis, in a similar sense to the hypothesis of HO-MHT, is a set of compatible tracks from each of the track trees. Two tracks are compatible if they do not share a measurement. The score of the hypothesis is the sum of the score of all its track. It is possible to enumerate all hypotheses, but it will suffer from the same combinational explosion as HO-MHT. However, while HO-MHT is dependant on generating a set of hypotheses to progress its data, all data is kept in the track trees in TO-MHT, and since one usually is interested in only the highest scoring hypothesis, one only needs to find that one. The problem of finding the highest scoring one is a Multi-Dimensional

Assignment (MDA) problem. The problem is NP-hard, but can be solved by using Lagrangian Relaxation [20].

A drawback to TO-MHT is that it does not include track initiation and needs to use a track initiation scheme as for JPDAF. Track termination may be implemented by pruning away low-scoring tracks. One may also be interested in more than one hypothesis when tracking, especially in a collision avoidance scenario, where all hypotheses that are sufficiently likely should be taken into account.

There exists debate to whether TO-MHT actually is faster than HO-MHT. It will largely depend on what is faster, the solving of the MDA-problem in TO-MHT or the generation of the K-best hypotheses in HO-MHT [2].

2.4.2 FISST algorithms

FISST takes a different approach to the tracking problem. It avoids explicit data association and rather treats the collections of target states and measurements as Random Finite Sets (RFS). This allows one to express a Bayes-optimal solution to the full multi-target tracking problem, known as the multi-target Bayes Filter. The solution is found with a single prediction equation and a single update equation, but it requires set integrations, and is therefore computationally intractable [16]. The Probability Hypothesis Density (PHD) filter [15] and the Cardinalized PHD (CPHD) filter [14] are developed as first moment approximation of the multi-target Bayes recursion. Adaptations and extension to the above methods exists, in addition to other algorithms based on FISST, but they will not be considered further.

In terms of performance, studies show that PHD and CPHD is better than JPDAF [19]. As for MHT no thorough comparisons exists. A study concludes with PHD outperforming TO-MHT [18], but the simulation scheme used seems somewhat sparse.

The PHD and CPHD filter runs in $\mathcal{O}(mn)$ and $\mathcal{O}(m^3n)$ respectively, where m and n are the number of measurements in the current scan and the currently existing targets [13]. Comparing that to HO-MHT with K-best generation, PHD and CPHD is far superior in terms of time complexity.

A drawback to the basic PHD and CPHD is that the output of the algorithm are target state points (called particles) and not tracks. This makes the suitable for purposes like collision avoidance, but less suitable for gathering the statistics required for this task. However, there exists adaptations that use some form of clustering of the particles to construct tracks. One is the popular SMC-PHD [25], another is one by Bar-Shalom[12].

3 | K-best Multiple Hypothesis Tracking with Track Deletion

In the first section a Bayesian multi-frame tracking algorithm is presented. It accounts for new targets, false measurements, misdetections of targets and the event of a target ceasing to exist. It has an approach similar to the works of Reid in what is now known as Hypothesis-Oriented MHT (HO-MHT). An algorithm for generating the K best hypotheses in polynomial time by using Murtys method is also presented, making it suitable for real-time applications. Further on, additional complexity reduction techniques are given, and ultimately, main topics on the implementation of the algorithm are discussed.

3.1 The Algorithm

3.1.1 Assumptions

First, assumptions about the tracking problem are made. A linear Gaussian model of the form is assumed for a single target state, $\mathbf{x}_k$, and a measurement, $\mathbf{z}_k$, associated with the target:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q}_k) \tag{3.1}$$
$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{R}_k) \tag{3.2}$$
$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0) \tag{3.3}$$
$$\tag{3.4}$$

where $\mathbf{F}_k$ is known as the transition matrix, and $\mathbf{H_k}$ the measurement matrix. The plant noise, $\mathbf{v}_k$, and the measurement noise $\mathbf{w}_k$, are assumed to be white. Also, the initial state, and the noise vectors at each step, $\{\mathbf{x}_0, \mathbf{v}_1, ..., \mathbf{v_k}, \mathbf{w}_1, ..., \mathbf{w}_k\}$, are all assumed to be mutually independent.

Given these assumptions, the Kalman Filter is an optimal solution to esti-

mating target states. As will become important further on, one also have the likelihood of a measurement associated with a target [4].

$$p(\mathbf{z}_k) = \mathcal{N}(\mathbf{z}_k;\, \mathbf{H}_k\bar{\mathbf{x}}_k, \mathbf{B}_k) \tag{3.5}$$

$$\mathbf{B}_k = \mathbf{H}_k\bar{\mathbf{P}}_k(\mathbf{H}_k)^T + \mathbf{R}_k \tag{3.6}$$

where $\bar{\mathbf{x}}_k$ is the predicted state estimate for the target at time k. $\mathbf{B}_k$ is the innovation covariance for the measurement and the target and $\bar{\mathbf{P}}_k$ is the predicted state error covariance of the target.

Further, the following assumptions are made for the rest of the problem:

A.1 *Point target*: Any target generates at most one measurement per scan.

A.2 *No merged measurements*: Any measurement originates from at most one target.

A.3 *Clutter and New targets*: Both clutter measurements and new targets occur according to a homogeneous Poisson point process with intensity β_C and β_N respectively, and are therfore uniformly distributed over the surveillance region.

A.4 *Target detection and existence*: A target either generates a measurement with constant probability P_D, cease to exist with constant probability P_X, or exists, but are misdetected, with probability $P_O = (1 - P_D - P_X)$.

3.1.2 Association Hypotheses

The approach to solving the tracking problem is to construct a set of hypotheses that account for all possible data-association of measurements and all possible target existences.

Assume at a time k, one has a set of measurements received at time k and a set of targets assumed to exist at $k-1$. Then, by following the assumptions, there exists a mapping and classification of measurements and existing targets that:

- Map one subset of the measurements to a subset of existing targets at $k-1$, implying continued existence of the targets for the next iteration,
- classify one subset of the measurements as new targets, implying the existence of new targets for the next iteration,
- classify the remaining measurements as clutter,
- and classify all previously existing targets, not assigned a measurement, as either existing or not existing.

A variation of this procedure, meaning a particular mapping and classification of measurements and targets, will now be referred to as a hypothesis innova-

tion, or just innovation. A hypothesis at time k consists of a set of innovations at each time step up to k. The complete set of hypotheses may be created recursively by creating every possible innovation for every hypotheses from the previous timestep. It should be clear that the number of hypotheses grows exponentially with the number of measurements.

A hypothesis will consist of two sets of tracks. One set of tracks for targets that still exists, and one set of tracks for targets that are terminated. A single track consists of a set of measurements assigned to the same target. No measurement in a track originated at the same timestep, and therefore the states of the target can be obtained by a Kalman Filter. Also, all tracks of a hypothesis are disjoint, a measurement is only in one track. Measurements not contained in any of the tracks of a hypothesis are measurements classified as clutter.

Two hypotheses at time k may have the same track, and may also have the same set of existing tracks or the same set of terminated tracks, but not both of them. Therefore, the complete set of tracks for hypotheses are disjoint.

Since a track consists of at maximum one measurement at each timestep, target estimates may be obtained by using a Kalman filter. Therefore, a hypothesis contains a set of estimates of targets that exists, which will prove useful when hypotheses are generated recursively.

3.1.3 Probability of a hypothesis

Let $Z_k = \{\mathbf{z}_{k,1}, \mathbf{z}_{k,2}, ..., \mathbf{z}_{k,M_k}\}$ denote the set of measurements at time k, and $Z^k = \{Z_1, Z_2, ..., Z_k\}$ denote all sets of measurements up to and including time k. Let Ω_i^k be the i-th hypothesis at scan k, consisting of a set of hypothesis innovations for all timesteps up to and including time k. Also, let $\Omega_{p(i)}^{k-1}$ denote the parent hypothesis of Ω_i^k, and ψ_k^i be the corresponding innovation that led to it. Then, one has the following relations between a parent hypothesis, a particular innovation and the resulting hypothesis. Also, one has a similar relation for the set of all measurements.

$$\begin{aligned} \Omega_i^k &= \psi_k^i \cup \Omega_{p(i)}^{k-1} \\ Z^k &= Z_k \cup Z^{k-1} \\ P(\Omega_i^k) &= P(\psi_k^i, \Omega_{p(i)}^{k-1}) \\ P(Z^k) &= P(Z_k, Z^{k-1}) \end{aligned}$$

Then the probability of a hypothesis given all measurement results in the following relation:

$$P_i^k := P(\Omega_i^k \mid Z^k) = P(\psi_k^i, \Omega_{p(i)}^{k-1} \mid Z_k, Z^{k-1}) \tag{3.7}$$

By using Bayes theorem and the chain rule, one obtains a recursive relationship between the hypothesis, the assignment and the parent hypotheses.

$$P(\Omega_i^k \mid Z^k) = \frac{1}{P(Z^k)} P(Z_k \mid \psi_k^i, \Omega_{p(i)}^{k-1}, Z^{k-1})\, P(\psi_k^i \mid \Omega_{p(i)}^{k-1}, Z^{k-1})\, P(\Omega_{p(i)}^{k-1} \mid Z^{k-1}) \tag{3.8}$$

$$P_i^k = \frac{1}{c_{k,1}} P(Z_k \mid \psi_k^i, \Omega_{p(i)}^{k-1}, Z^{k-1})\, P(\psi_k^i \mid \Omega_{p(i)}^{k-1}, Z^{k-1})\, P_{p(i)}^{k-1} \tag{3.9}$$

Since, the probability of all measurements, $P(Z^k)$, are equal for all hypotheses at time k we can set this a normalizing constant, $c_{k,1}$. The third term on the RHS is known from the previous iteration. Then there remains two terms on the RHS that need to be derived.

We start off with the second term. From $\Omega_{p(i)}^{k-1}$ one has the number of existing targets, N_E, at $k-1$. Then, depending on the particular ψ_k^i one will have:

- N_D targets that are detected and assigned a measurement.
- N_N measurements that are classified as new targets.
- $N_C = M_k - N_D - N_N$ measurements that are classified as clutter.
- N_X targets that are terminated.
- $N_O = N_E - N_D - N_X$ that were not detected, but still exists (occluded).

From assumption A.3 the number of clutter and the number of new targets are Poisson distributed. And from assumption A.4 the number of detected, misdetected and terminated targets are multinomially distributed. Therefore the probability of the number of elements in the subset are given as.

$$\begin{aligned} P(N_{k,i} \mid \Omega_{p(i)}^{k-1}, Z^{k-1}) &= P(N_D, N_N, N_C, N_X, N_O \mid \Omega_{p(i)}^{k-1}, Z^{k-1}) \\ &= \frac{N_E!}{N_D!\, N_X!\, N_O!} P_D^{N_D} P_X^{N_X} P_O^{N_O} \\ &\quad e^{-V\beta_N} \frac{(V\beta_N)^{N_N}}{N_N!} e^{-V\beta_C} \frac{(V\beta_C)^{N_C}}{N_C!} \end{aligned} \tag{3.10}$$

Where V is the volume (or area) of the surveillance region. By using that exponential terms are equal for all hypotheses, we combine and simplify the expression to:

$$P(N_{k,i} \mid \Omega_{p(i)}^{k-1}, Z^{k-1}) = \frac{1}{c_{k,2}} \frac{N_E!}{N_D!\, N_X!\, N_O!} P_D^{N_D} P_X^{N_X} P_O^{N_O} V^{N_N+N_C} \frac{\beta_N^{N_N} \beta_C^{N_C}}{N_N! N_C!} \tag{3.11}$$

Given the number in the different subsets, there exists various combinations the subsets can be drawn from the whole set of measurements and targets. Since, all target probabilities and also clutter and new target intensities are constant, the probability of all configurations are equal. Therefore, the probability of a particular configuration can be found by dividing it by the number of all possible configurations.

$$\begin{aligned} P(C_{k,i} \,|\, N_{k,i}, \Omega_{p(i)}^{k-1}, Z^{k-1}) &= \frac{1}{\binom{M_k}{N_D}\binom{M_k - N_D}{N_C}\binom{M_k - N_D - N_C}{N_N}\binom{N_E - N_D}{N_X}\binom{N_E - N_D - N_X}{N_O}} \\ &= \frac{N_D!\, N_N!\, N_C!\, N_X!\, N_O!}{M_k!\,(N_E - N_D)!} \end{aligned} \tag{3.12}$$

In addition, given the configuration, there are different ways of assigning the N_D measurements to the N_E targets. The probability of them are equal as well, given the constant probability of detection.

$$P(A_{k,i} \,|\, C_{k,i}, N_{k,i}, \Omega_{p(i)}^{k-1}, Z^{k-1}) = \frac{1}{\frac{N_E!}{(N_E - N_D)!}} = \frac{(N_E - N_D)!}{N_E!} \tag{3.13}$$

The probability of a particular innovation is the joint probability of the three events described above.

$$\begin{aligned} P(\psi_k^i \,|\, \Omega_{p(i)}^{k-1}, Z^{k-1}) &= P(A_{k,i}, C_{k,i}, N_{k,i} \,|\, \Omega_{p(i)}^{k-1}, Z^{k-1}) \\ &= P(A_{k,i} \,|\, C_{k,i}, N_{k,i}, \Omega_{p(i)}^{k-1}, Z^{k-1}) \cdot \\ &\quad P(C_{k,i} \,|\, N_{k,i}, \Omega_{p(i)}^{k-1}, Z^{k-1}) \cdot \\ &\quad P(N_{k,i} \,|\, \Omega_{p(i)}^{k-1}, Z^{k-1}) \end{aligned} \tag{3.14}$$

Moving on to the first term in equation 3.9, we have from assumption A.3 that measurements that are clutter or new targets are uniformly distributed over the surveillance region. Also, from equation 3.5 we have the likelihood of an assigned measurement given that it is assigned to a target. Classification and assignment of measurements are given by ψ_k^i, and therefore we have:

$$P(Z_k \,|\, \psi_k^i, \Omega_{p(i)}^{k-1}, Z^{k-1}) = \prod_{j=1}^{M_k} \begin{cases} 1/V & \text{if } z_j \text{ is clutter or a new target} \\ \mathcal{N}(z_{k,j} - H\bar{x}_{k,l}, B_{k,jl}) & \text{if } z_j \text{ is assigned to target } l \end{cases} \tag{3.15}$$

If one, for ease of notation, assume the N_D first measurements are the measurements assigned to detected targets, where measurement j is assigned to target $a(j)$ one can rewrite the expression as:

$$P(Z_k \mid \psi_k^i, \Omega_{p(i)}^{k-1}, Z^{k-1}) = \frac{1}{V^{N_C+N_N}} \prod_{j=1}^{N_D} \mathcal{N}(z_{k,j} - H\bar{x}_{k,a(j)}, B_{k,a(j)}) \qquad (3.16)$$

Finally, by inserting (3.11 - 3.13) into 3.14, and then the resulting equation together with 3.16 into 3.9 we obtain the final expression for the probability of a hypothesis:

$$P_i^k = \frac{1}{c_k} \beta_C^{N_C} \beta_N^{N_N} P_D^{N_D} P_X^{N_X} P_O^{N_O} \prod_{j=1}^{N_D} [\mathcal{N}(z_{k,j} - H\bar{x}_{k,a(j)}, B_{k,a(j)})] P_{p(i)}^{k-1} \qquad (3.17)$$

The normalizing constant, c_k, can be found by taking the sum of all hypotheses at time k. The initial case, when $k = 1$, then $P_{p(i)}^0 = 1$. Note that the expression equals that of Reid when $N_X = 0$ and $P_X = 0$ [21]. A comparison with a similar expression of Kurien [11] will be given in Section 3.1.4.

The hypothesis probability will often not be descriptive in the level of confidence that the hypothesis is the correct one. Hypotheses grow exponentially, and many hypotheses will be of negligible probability. However, these hypotheses may sum to a great deal of the probability. Therefore, as the number of hypotheses grows, the most likely hypotheses often decreases in probability. Therefore, the ratio between the most likely hypothesis and the second most likely may be seen as a better measure of confidence in that the hypothesis is the "correct" one.

In addition, there may be cases where the most likely hypotheses are similar, for instance if a target generates a double measurement (split measurement). The probability of the hypothesis that associates one of the measurements will be almost equal to the hypothesis associating the other, but still much lower than the hypothesis that would be created if the target did not generate a split measurement. However, the two hypotheses in the split case will still have close to the same ratio to other hypotheses as the hypothesis in the non-split case.

3.1.4 Comparison with Kurien

Kurien also gives an expression for a hypothesis in similarity to what is done here, by also expanding the framework of Reid to include a probability of track termination. It also includes the possibility of targets executing different maneuvers, together with leaving the likelihood of measurements undefined. However, if the number of clutter measurements is Poisson distributed over a region (as Kurien models), the likelihood of a measurement is by that model uniformly distributed over the same region.

If one removes the possibility of special target maneuvers ($m = 0, P_m = 0$), assume the likelihood of measurements are equal to that of here, and write the numbers in terms of the notation used in this thesis, one obtains

$$P_i^k = \frac{1}{c_k}\beta_C^{N_C}\beta_N^{N_N}(\hat{P}_D(1-\hat{P}_X))^{N_D}\hat{P}_X^{N_X}((1-\hat{P}_X)(1-\hat{P}_D))^{N_O}\cdot \prod_{j=1}^{N_D}[\mathcal{N}(z_{k,j}-H\bar{x}_{k,a(j)}, B_{k,a(j)})]P_{p(i)}^{k-1} \tag{3.18}$$

It equals Equation 3.17 by the following relation between probabilities of detection and termination by Kurien and the ones presented here:

$$\begin{aligned} P_D &= \hat{P}_D(1-\hat{P}_X) \\ P_X &= \hat{P}_X \\ P_O &= (1-\hat{P}_X)(1-\hat{P}_D) \end{aligned} \tag{3.19}$$

3.1.5 Target State Initialization

Generally, a single measurement does not provide enough information to fully initialize the states of the target in 3.3. Usually one only have information about the position, while target states contain states of motion as well. The initialization of $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$ will therefore depend on the particular model.

Reid presents a scheme for the Discrete-Time, White Noise Acceleration-model (DWNA-model)-model with measurements in cartesian coordinates. Position entries of $\hat{\mathbf{x}}_0$ are set equal to the measurement position and its variance. Velocity entries of $\hat{\mathbf{x}}_0$ are initialized to zero mean with variance equal to $(v_{max}/3)^2$, where v_{max} is the assumed maximum initial velocity of targets.

3.2 K-Best Hypothesis Generation

The number of hypotheses grows exponentially with the number of measurements and targets. While it is possible to prune away unlikely hypotheses at each time step (pruning will be discussed further on in section 3.3.2, just enumerating all hypotheses at a single timestep could give a very long computation time. At best it will deem the algorithm useless for real-time scenarios like collision avoidance. However, it will be shown that finding the K-best hypotheses can be done in polynomial time.

Finding an innovation from a parent hypothesis can be formulated as a weighted bipartite matching problem. The problem of finding the maximum probability innovation then reduces to the "assignment problem" from combinatorial optimization. This problem is solved in $\mathcal{O}(N^3)$, where N is the number of nodes,

by the "Hungarian Algorithm". However, it is usually wanted to find more than the maximum likelihood hypothesis, and thus, still possess the multi-frame property of the algorithm. The Murty algorithm solves the assignment problem K times, by iteratively finding the best solution and then removing it from the problem. By using the algorithm by Murty the K best solutions to the bipartite matching problem can be found in $\mathcal{O}(KN^3)$ [17], however as the problem size is reduced by every timestep, the average runtime is expected to be better as shown in [6].

We have that each measurement can either be assigned to an existing track, be classified as a new target or as clutter. At the same time we have that an existing track can either be detected and assigned a measurement, not be detected or become terminated. These can in some sense be seen as two overlapping matching problems, where the assignment of measurements to an existing track is common for both of them.

The problem of generating the best hypothesis from a parent is formulated as a bipartite matching problem, where the total score are given by the product of its arcs. It is defined by the following matrix, $\bar{C}$. Each entry in the matrix defines an arc between nodes in the graph. The names of the nodes are given on the top and left side, separated by solid lines.

	T_1	$\dots$	T_{N_E}	B_1	$\dots$	B_{M_k}	F_1	$\dots$	F_{M_k}	O_1	$\dots$	O_{N_E}
M_1	l_{11}	$\dots$	l_{1N_E}	β_N	$\dots$	0	β_C	$\dots$	0			
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$		0	
M_{M_k}	l_{M_k1}	$\dots$	$l_{M_kN_E}$	0	$\dots$	β_N	0	$\dots$	β_C			
X_1	P_X/P_O	$\dots$	0							1	$\dots$	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$		0			0		$\vdots$	$\ddots$	$\vdots$
X_{N_E}	0	$\dots$	P_X/P_O							0	$\dots$	1

(3.20)

where

$$l_{ij} = \frac{P_D}{P_O}\mathcal{N}(z_{k,i} - H\bar{x}_{k,j}, B_{k,ij}) \tag{3.21}$$

Then, by negative log-transforming each element of the matrix, the problem is reformulated as a sum of arcs instead of a product. Then, the K-best hypothesis, innovated from a particular parent, and their probability can be found by using Murty.

$$C = -\log(\bar{C} + \epsilon) \tag{3.22}$$

$$P_i^k = e^{-S_i} P_O^{N_E} P_{p(i)}^{k-1} \tag{3.23}$$

where S_i denotes the score of a particular solution of C, and ϵ is an arbitrarily small number, making the logarithm defined for all entries of $\bar{C}$. The matching translates into an innovation, ψ_k^i, by the following relations:

- $M_i \leftrightarrow T_j$: z_i is assigned to target j. It should be noted that this will always result in the matching $X_j \leftrightarrow O_j$ as only two matchings, $X_j \leftrightarrow T_j$ and $X_j \leftrightarrow O_j$, are possible for node X_j.
- $M_i \leftrightarrow B_i$: z_i is classified as a new target.
- $M_i \leftrightarrow F_i$: z_i is classified as clutter.
- $X_i \leftrightarrow T_i$: Target i is terminated.
- $X_i \leftrightarrow O_i$: Target i is detected if $M_j \leftrightarrow T_i$ and occluded if not.

There may be times where K valid innovations are not available, in which solutions are found where one or more of the mappings are between nodes with zero entries in $\bar{C}$. Those solutions are identified with $e^{-S_i} < \epsilon$, and one can stop the algorithm. A rule of thumb is to ensure $\epsilon < max(\bar{C}) * (M_k + N_E)$ to ensure that no valid solution can be below ϵ.

The resulting time complexity of generating the K-best hypotheses from a single parent is $\mathcal{O}(K(M_k + N_E)^3)$. The total runtime of generating all K^2 hypotheses from all K parents is then $\mathcal{O}(K^2(M_k + N_E)^3)$. Note that the N_E will vary from parent to parent. To keep a constant K hypotheses, a list of sorted, K hypotheses, is obtained in $\mathcal{O}(K^2)$ by building a max heap from the K^2 hypotheses and extracting K elements.

However, there exists a scheme that on average is expected to achieve a better runtime. By utilizing the fact that the K solutions to the Murty algorithm can be obtained iteratively, one can stop generating innovations from a parent if the probability of the resulting hypothesis is lower than the currently K worst hypothesis generated from all other parents, as the next innovations are guaranteed too have an even lower probability. The K-best hypotheses can be maintained in a *min*-heap sorted on probability. If there are more than K hypothesis in the heap, and the probability of a new hypothesis is better than the worse, it replaces it. At the end of the generation of all hypothesis there are at maximum K hypothesis in the heap. Then, the sorted list of hypothesis can be obtained by extracting all values of the min heap and reversing the resulting list.

As one innovates parent hypotheses with decreasing probability, one are increasingly likely to stop innovating hypotheses before K innovations are generated. The worst case is that all K hypotheses are innovated from the least likely parent hypothesis, in which case the runtime is still $O(K^2(M_k + max(N_E)))$.

For the above scheme, with a heap the worst case runtime is even worse with $O(T \log T)$, where $T = K^2(M_k + max(N_E))$.

3.3 Complexity reduction techniques

The number of hypotheses grows exponentially. In addition to only generating the K-best hypotheses there exists various techniques to reduce the complexity of the problem. Both clustering and pruning techniques will be presented, and discussed jointly as they are dependent.

3.3.1 Clustering

When tracking multiple targets, one often have a substantial distance between targets in the observable region. If two targets are far from each other, one remains with two approximately disjoint tracking problems, as the probability of assigning measurements generated by one target is close to zero for the other target, by equation 3.5.

Clustering is used to divide the entire set of global hypotheses into sets of local hypotheses (clusters) that do not interact with each other. This leads to a number of smaller tracking problems that can be solved independently from each other.

First, we introduce the validation region for a measurement and a predicted target estimate.

$$(\mathbf{z}_k - \mathbf{H}\bar{\mathbf{x}}_k)\mathbf{B}_k^{-1}(\mathbf{z}_k - \mathbf{H}\bar{\mathbf{x}}_k) \leq \text{chi2cdf}^{-1}(P_G) \tag{3.24}$$

where P_G is the confidence level of the validation test, and is often referred to as the "gate probability". It is usually in the interval $0.95 - 0.99$. chi2inv^{-1} is the inverse of the cumulative chi-squared distribution. As the variance in B_k is sampled from a set of normally distributed variables, it follows a chi-squared distribution. If a measurement and a target estimate satisfies 3.24, the measurement is "gated" by the target. If not, one removes the possibility of assigning the measurement to the target, with confidence P_G.

A cluster is defined by a set of mutually exclusive hypotheses. Each hypothesis in a cluster has a set of existing target estimates. If a measurement falls into the validation region of any of the existing target estimates of any of the hypothesis in a cluster, the measurement is associated with the cluster. If a measurement is not associated with any cluster, a new cluster is formed, leading to only two possible hypothesis in the cluster, that the measurement is clutter or a new target.

Last, if a measurement is associated with two or more clusters, the clusters are merged into a "supercluster". The resulting cluster consists of the joint hypothesis of the prior clusters. The number of hypothesis are the product of the number of hypotheses in the prior clusters. A joint hypothesis is created from two or more prior hypotheses by taking the union of the existing tracks, and the union of the terminated tracks. The probability is the product of the probabilities of the prior hypotheses. It can be shown that the resulting probability is the same as if the hypotheses were calculated jointly from the beginning if P_G is sufficiently large.

It should be noted that while clustering is optimal, it may not be as effective as one would expect. The reason for this is that one will have hypotheses that account for the possibility of tracks being occluded, in which the predicted estimate covariance, B_k, will grow large and lead to a big validation region. The respective hypotheses may be very unlikely. This leads to measurements being gated "easily" by clusters, and few new clusters will be created and/or clusters would often be merged. By pruning away very unlikely hypotheses one may prune away the problematic hypotheses and improve on the clustering.

With pruning and track termination one could end up with "dead clusters", in which none of the remaining hypotheses contain any existing tracks. This will lead to no gating of measurements, and the cluster can be removed from the tracking problem. Therefore, clustering does not only divide the problem into sub-problems only in terms of proximity, but also in time.

3.3.2 Pruning

Pruning is necessary to keep the number of hypotheses at a reasonable level, as briefly discussed in section 3.2, where K-best hypotheses generation was presented. In addition, pruning may lead to more effective clustering, as discussed in section 3.3.1. In addition to K-best, there exists other pruning techniques. Those will be presented and discussed.

Ratio pruning

One of the simplest pruning techniques is the ratio pruning. If the ratio between the most likely hypothesis and another one is greater than a threshold the latter may be removed. The reason for the ratio being used and not an absolute threshold on probability is used is because of that discussed late in Section 3.1.3.

$$\forall i: \ \frac{\max_j \left[P(\Omega_j^k)\right]}{P(\Omega_i^k)} > r_{prune} \Rightarrow \text{prune } \Omega_i^k \tag{3.25}$$

N-Scan

With N-Scan pruning the hypothesis at time $k - N$, with the greatest sum of child hypothesis probability at time k is kept, while all others are pruned away. If the hypotheses through time are visualized as a tree, this corresponds to removing all but one branch, N levels back. If $N = 0$, this corresponds to choosing the maximum likely hypothesis at time k, in which the algorithm reduces to a single-frame one.

Selective pruning

Both N-scan, K-best and ratio pruning filters out hypotheses based on only probability. A more sophisticated approach is to prune hypotheses depending on criteria other than just probability. For instance, one could keep hypotheses known to have initially low probability but usually increases as child hypotheses are generated. This could often be the case for new targets and deleted targets, as β_N and P_X usually are low compared to P_D, P_O and β_C.

Comparison

The problem with K-best and ratio pruning is that they may both prune away hypotheses prematurely. This could often be the case for hypotheses that deletes or initiates one or more targets, since their initial probability usually start off low, but increases through time, if they are correct.

An example is made for a target that generates its last measurement at time $k-1$. No measurement is generated at k. since P_X is low compared to P_O, the probability of the hypothesis accounting for the deletion is low. However, for the other hypotheses, where the target still exists, their child hypotheses will be multiplied by P_O for each subsequent time step where no target measurement is generated. The hypothesis where it was deleted will remain constant, and eventually become the most likely. If K or the ratio is too low, one may prune away the deletion hypothesis, which eventually shows to be the correct one. This both illustrates the strength of a multi-frame algorithm, and also the dangers of pruning. The ratio threshold should at least be greater than P_O/P_X and β_C/β_N.

"Selective pruning" may be used to avoid problems presented by the above example, but it requires more intimate knowledge of the problem and is harder to implement.

N-scan proves to be effective in for the above example, as it keeps a greater set of "probability diversity" at the current timestep, since the criteria on removing hypotheses is set back in time. If the target in the above example has been observed for a long time and is very likely to be present, the hypothesis is "confirmed" back in time, removing all other child nodes at time k, allowing

the low deletion hypothesis to be generated/kept. However, the case may be that back in time, the hypothesis that a target is present is not the most likely, which may be the case for a new target, where some timesteps/measurements are needed for the track too be likely. Then N-scan would prune away the track and classify the measurements as clutter. This illustrates it is important to keep N large.

3.4 Implementation and Data Structure

Multiple Hypothesis Tracking is both expensive in terms of computation time and memory. It is therefore in need of careful implementation and efficient data structures to both avoid recomputing and redundant memory use. First, an overview of the algorithm will be presented, then a more in depth explanation of the data structures used.

3.4.1 Overview

1. Receive measurements, Z_k
2. Clustering and likelihood calculation.
3. *For each cluster:*
   (a) Generate K-best hypotheses
   (b) Prune Hypotheses
   (c) Update cluster track nodes
   (d) Remove cluster if dead and save to file.
4. Create new clusters from measurements which are not gated.
5. Predict leaves of the target track tree.

3.4.2 Data Structures

The hypotheses of the algorithm are stored in a way similar to a tree data structure. If clustering is not used, a hypothesis has a single parent, and several child nodes. However, when one uses clustering, one or more clusters may be merged and a hypotheses could have more than one parent. Therefore, the hypothesis tree is in general not a "pure" tree structure, but rather several trees, which "occasionally" are merged. This is illustrated in Figure 3.1. A hypothesis structure is created when a new cluster is created. It will not have a single root as there are always two hypotheses when a cluster is created (the measurement is clutter or it is a new target).

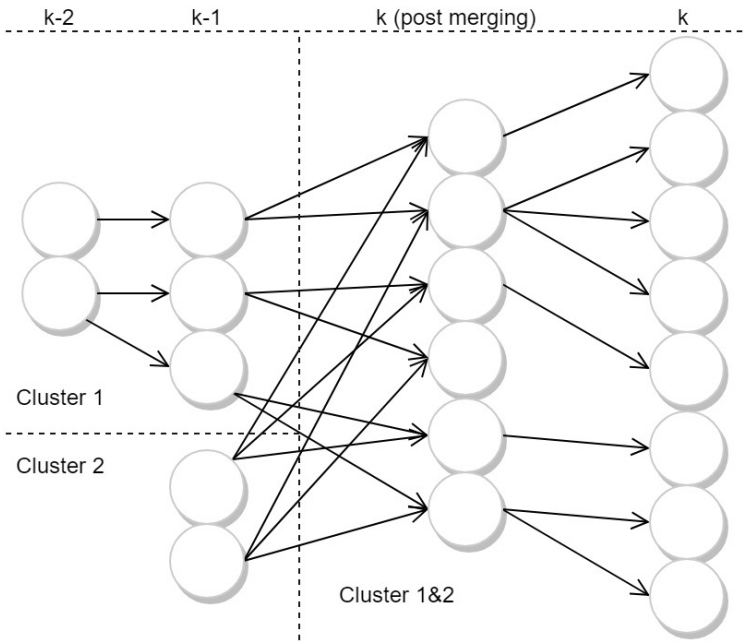


Figure 3.1: Example of the hypothesis data structure with a cluster merge.

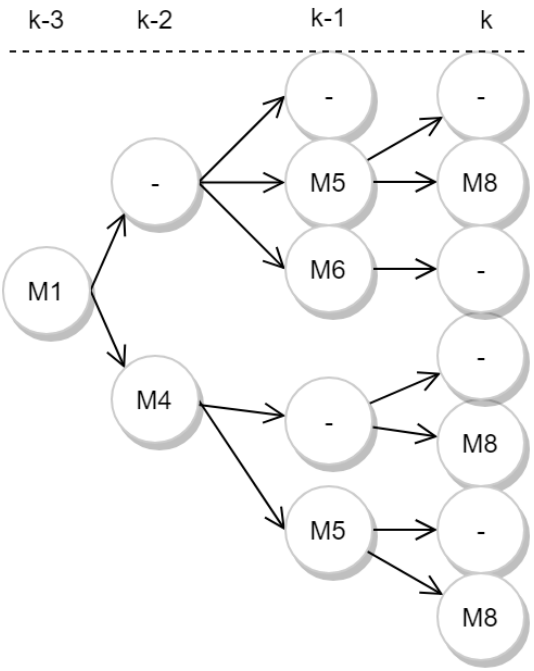


Figure 3.2: Example of a track tree. Measurement indices are shown inside the nodes. "-" denotes a node where there are no measurement.

Multiple track trees are maintained in addition to the hypothesis tree. The reason for this choice is that hypotheses may share the same tracks between them. A measurement will create a new track tree as one will have the possibility of it being a new target. Also, the same measurement will be used to

expand other track trees by associating it to previously existing targets. The track tree is a pure tree data structure, as a node may only have a single parent and a single root.

As a track tree is created and expanded, state estimates (and its error covariance) are updated. The track tree is also expanded with a trivial step for the possibility of targets not being assigned a measurement. This will be referred to as innovating the track with an "empty" measurement. Note that all track nodes at the same depth of a track tree have target estimates for the same timestep.

A hypothesis node have a list of pointers to nodes of the track trees, and the complete tracks of a hypothesis can be found by traversing the tree backwards from the node which the hypothesis points to. There are one list of pointers for existing targets and one for terminated targets. When a hypothesis is innovated, the dead track list is copied and previously existing tracks that gets terminated are added to the list. Each track that keeps their existence are updated with a new track node since the track is extended. The extension of a track node is done by querying a map structure contained in the track node. The map uses measurements as keys and the data is the child nodes corresponding to the measurements. If the child node does not exist upon querying, it is created and added to the map.

A cluster structure keeps both a list of all leaves for its hypothesis tree, and a list for all leaves of all its track trees. The leaves of the hypothesis tree are used for fast access when innovating hypotheses, while the track leaves are used for fast access when predicting leaf node states and when clustering. In addition, the cluster structure has a list of measurements that are gated in the clustering.

The LHS of Equation 3.24, which is computed when gating measurements, is also computed when calculating the measurement assignment likelihood, defined in Equation 3.5. Therefore, gating and measurement likelihood computation is done at the same time and the result of the likelihood is stored in a map structure in the track node. This saves computation time when creating the cost matrix in Equation 3.20 as several hypotheses have the same tracks. An overview of the clustering and likelihood calculation process is shown in Algorithm 1.

Algorithm 1 Clustering and Likelihood Computation

```
1: C: List of clusters at k-1
2: Z: List of measurements at k
3: γ: Gate threshold. chi2cdf^{-1}(P_G)
4: Initialize empty list of unassociated measurements, UZ
5: for each z in Z do
6:    Initialize empty list of associated clusters, AC
7:    for each cluster in C do
8:       associated ← False
9:       for each track_node in cluster.track_nodes do
10:         gate_value ← calcGateValue(z, track_node)
11:         if gate_value < γ then
12:            likelihood ← calcLikelihood(track_node, gate_value)
13:            track_node.storeResult(z, likelihood)
14:            if not associated then
15:               associated ← True
16:               AC.append(c)
17:            end if
18:         end if
19:      end for
20:   end for
21:   if AC.length == 0 then
22:      UZ.append(z)
23:   else if AC.length == 1 then
24:      AC[0].gated_measurements.append(z)
25:   else
26:      new_cluster ← MergeClusters(AC)
27:      new_cluster.gated_measurements.append(z)
28:      C.remove(AC)
29:      C.append(new_cluster)
30:   end if
31: end for
32: return UZ
```

4 | The Tracking System

This section will specify the complete setup for the tracking system. The algorithm to be used is the MHT with Track Deletion and clustering, as presented in Chapter 3.

4.1 Radar

4.1.1 Location

The surveillance post is located on the north-east shore of the canal. The exact location is at (lat: 63.435183, lng: 10.392984) shown in Figure 4.1. Both radar and camera vision is quite good for the middle of the canal, but somewhat limited at the west and east entry/exit points.

The radar is positioned at about 7 meters above sea level, and is therefore able to detect some targets at the same line of sight, depending on the height and range to the targets. The camera is positioned about a meter lower.

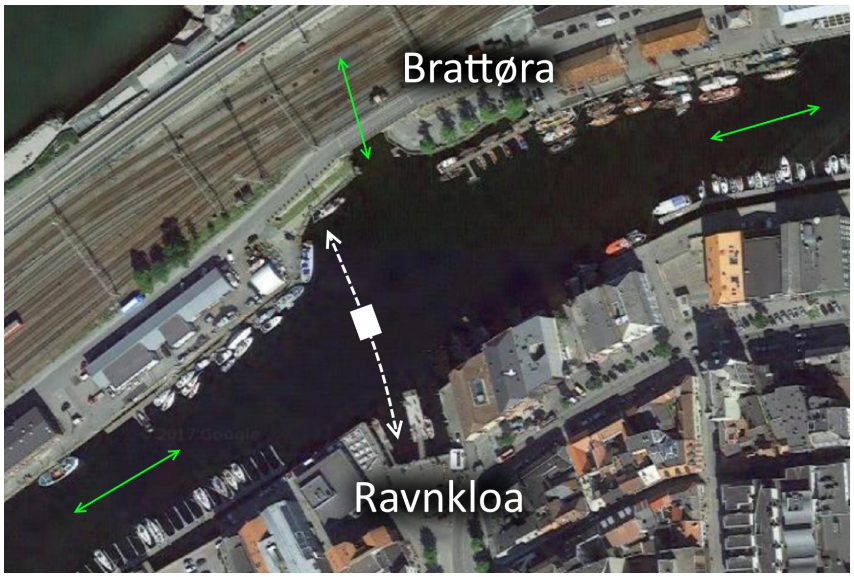


Figure 4.1: The location of the platform. Radar range in red, and the area of where boats are too be tracked in green. The main entry/exit points are marked in turquoise.

4.1.2 Radar

The radar used is the *Simrad Broadband 4G*. This is a Frequency-Modulated Continuous-Wave (FM-CW) radar. The available speeds are 24, 36 and 48 RPM, giving sampling times of respectively 2.5, 1.66 and 1.25 seconds. However, due to an error in the software provided for acquiring data, the sampling time had to be set as an integer, giving a lowest possible sampling time of 5 seconds.

There are settings available on the radar, shown in Table 4.1, in together with the values. Some settings like the gain, Sensitivity Time Control (STC) and Fast Time Constant (FTC) are standard for all modern radars. However, some of the settings specific for this radar were not described in its manual[23], and therefore lacks a description in the table.

Setting	Value	Description
Range	175	
Gain	Auto	Manual disables DCR
Sea	Harbor	Harbor/Sea. "Sea" disables DCR.
Rain	0	...
FTC	0	Used to reduce the effect of rain.
Sidelobe	Auto	
STC	Moderate	Used to avoid saturation at close range.
Fast scan	High	Off/Medium/High = 24/32/48 RPM
Interference rejection (IR)	High	
Local IR	High	Removes interference from other sources.
Beam Sharpening	High	Improves angular accuracy.
Target Stretch	Off	To be used for long ranges.
Target Boost	High	

Table 4.1: Table showing the settings of the radar.

The gain of the radar is set to "Auto", in which the radar software itself determines appropriate gain. Also, this allows for the radar to use "Directional Clutter Rejection" (DCR), where it is assumed that the gain is set differently for different angles. It was experienced that setting the gain to a manual setting gave worse results as more targets were misdetected.

Since information on the settings were not well documented, they were individually adjusted and its effects examined. Only "Interference Rejection" and "Local IR" showed a positive effect as it was changed from its default settings ("Medium"). Increasing it to "High" decreased the amount and fluctuation of clutter detections near the edge of the canal.

Output of the radar is a 1024x1024 "png"-image, giving a regular grid of cells marking detections. Since a radar obtains measurements in reference to a polar coordinate system, a projection from polar coordinates to Cartesian ones are already made by the radar:

$$z = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \end{bmatrix} \tag{4.1}$$

4.2 Detection

The raw radar output goes through a set of filters to remove false detections. Two mask images are first applied to remove detections from the raw radar output. Then the cell detections are clustered. If the area of the cluster is greater than a threshold, the centroid of the cluster is used as a measurement. In addition a filter was made to remove "multiple"-detections.

4.2.1 Rotation and Range

First, a 183 degree rotation is performed to represent the radar output in respect to a North East Down (NED)-frame. The value of the rotation is found by comparing the raw radar output to a satellite overlay. The rotation is applied by using a rotation matrix on the raw radar image and interpolating with nearest neighbour.

A difference between the range setting in the radar and the actual range was found. The specific factor was found in similar fashion to finding the rotation. When the radar setting was $175m$ the actual range was found to be $200m$.

$$\frac{\text{actual range}}{\text{radar setting}} = \frac{200}{175} = 1.143 \tag{4.2}$$

4.2.2 Mask Filters

Two image masks are made to filter out highly unlikely cells that are marked as a detection.

An image mask is simply a 2D-array where each element is either true or false. The output of the radar is also a 2D-array where each cell (element) is either true or false for a detection. If the masks are true for cells that should be filtered out, the filtered radar image is obtained by:

$$D_1 = D_0 \wedge \neg M_{combined} \qquad M_{combined} = M_{satellite} \vee M_{average} \tag{4.3}$$

Satellite Mask

A great deal of the detections in the raw radar output corresponds with land, upon there are no targets. By using a satellite overlay an image mask is made "by hand" to filter out these detections.

In addition, it was discovered in the preceding report [cite] that a "deadzone" was present on the east end of the canal, due to the radar being obscured. Only clutter occurred in the zone and no targets were detected. Therefore, this zone is also included in the mask. The resulting mask is shown in Figure 6.1.

Average Mask

In addition to the satellite image mask, another one was made to account for false detections appearing in the canal at the same location frequently. Frequent detections were seen to be located at the shore of the canal.

A mask to ignore detections at cells occurring p times or more in a set of samples is described mathematically as:

$$M_1(p; x, y) = \left(\frac{1}{N} \sum_{i=1}^{N} I_i(x, y) \right) \geq p, \quad I_i(x, y) = \{0, 1\} \tag{4.4}$$

where $I_i(x, y)$ is a radar sample and N is the number of images, equal to the number of samples. This resembles the simplest of "background subtraction"-techniques often used for camera detection.

After a mask has been created, the *findContours* method from *OpenCV* is used to locate connected components. In the method, the border points (contour) of a connected component is found. It uses 8-connectivity, meaning two pixels are adjacent if next or diagonal to each other. Each connected component contour are then filled, which leaves no holes in the connected components.

4.2.3 Clustering using Connected Components

All targets of interest are occupying more than one cell. Therefore, connected component analysis are performed on the detection image to create one single measurement value per target.

The contours of the connected components are found using the *findContours* method from *OpenCV*. In the method, the border points (contour) of a connected component is found. It uses 8-connectivity, meaning two pixels are adjacent if next or diagonal to each other.. Then, the centroid and the area of each component are found. The centroid is in pixel-coordinates. It is translated and scaled to give the measurement position in reference to a NED-frame centered at the radar location.

4.2.4 Size Filter

Some detections are too small in size to be originating from targets. Therefore, detections with area less than A_{min} are removed. To determine the threshold, sensitivity analysis will be performed. It is important to decrease the amount of clutter, but not too many of the detections.

4.2.5 Multiple Filter

By examining the raw data, "multiples" was seen to occur when large boats appeared close to the radar.

The "multiple"-phenomenon is illustrated in 4.2. A radar transmits energy, which is reflected by a surface and received by the antenna. The distance to

the surface is calculated by using the time from the transmit to the receive. However, the received energy of the antenna can reflect off the antenna, reflect off the same surface and get received again. If the received energy is large enough it will lead to a false detection at the same angle, but double the distance of the true one. This phenomenon may happen more than once, leading to several false detections at two, three, four, etc, distance of the correct detection.

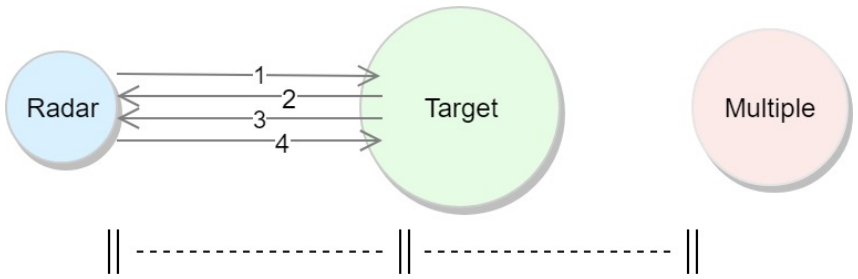


Figure 4.2: Illustrating the "multiple"-phenomenon of a radar. A second order multiple. shown.

If multiples occur at several consequent timesteps they could resemble the path of a target and lead to false tracks.

A multiple usually appears directly behind a target. Therefore, if a detection is within the angular region covered by another detection, closer to the radar, it is a candidate to being a multiple. Usually, a multiple will cover a smaller angular region than the true detection since only the center of the target reflects enough energy to generate a multiple. The very edges of the target will only reflect enough energy to generate the true one, but not enough to generate a detection for the multiple.

A multiple will occur at a distance multiple of the true detection, and one could in theory examine if the distance to a detection from the radar is about a multiple (2, 3, 4 etc.) of another. However, it is difficult to find the appropriate reference point, because the exact location of where the energy that leads to the multiple reflects off, is hard to identify. This will be illustrated in Section [results: multiple].

Therefore, a simplification was made and a detection is marked as a multiple if a it is completely obscured by a detection of size larger than a threshold $A_{multiple}$. This is reasonable since only large targets will generate multiples. Also, a large target will limit the view of the radar, leading to it being impossible to actually detect targets behind it.

The filter is implemented with the following procedure:

1. In the set of detections, see if any has area greater or equal to $A_{multiple}$. They are candidates for generating multiples.
2. For each detection: Calculate the angle of each contour point in reference to the radar, and find the minimum and maximum values. Also, calculate

the distance from each detection centroid to the radar.

3. Remove a detection if its minimum angle is greater and its maximum angle is less than a candidate, and the ratio of distance between the detection and the candidate is greater the threshold $r_{multiple}$.

4.3 Tracking Model

The state of a target is given as $x = [N, V_N, E, V_E]$, where N, E, V_N and V_E are the north and east positions and velocities of the target in a NED reference frame centered at the radar/camera location.

The popular DWNA-model is used to model the target motion, where acceleration is modeled as process noise. It assumes as little as possible of target maneuvers. The state transition matrix and the plant noise covariance as defined in Equation 3.1 are given as:

$$F = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad Q = \sigma_v^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & 0 & 0 \\ \frac{\Delta t^3}{2} & \Delta t^2 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ 0 & 0 & \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} \tag{4.5}$$

where Δt is the sample time and σ_v^2 is the continuous acceleration noise variance. The model assumes acceleration is constant in a single sample interval, $[x_k, x_k + 1)$.

The output from the detection is the position in the same reference frame as the state, leading to the following measurement mapping.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad R = \sigma_w^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.6}$$

where σ_w is the standard deviation of the *discrete* measurement noise.

State Initialization

The MHT with Track Termination requires the target state to be initialized with a single measurement. The measurement only supplies information of the position of the target and not its speed. Therefore, the scheme proposed in Section 3.1.5 is used, which requires determining the maximum speed of the targets, v_{max}. The initial state and its covariance is given as

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} z_{i,0} & 0 & z_{i,1} & 0 \end{bmatrix}^T \tag{4.7}$$

$$\hat{\mathbf{P}}_0 = \begin{bmatrix} \sigma_w^2 & 0 & 0 & 0 \\ 0 & \frac{v_{max}}{3}^2 & 0 & 0 \\ 0 & 0 & \sigma_w^2 & 0 \\ 0 & 0 & 0 & \frac{v_{max}}{3}^2 \end{bmatrix} \tag{4.8}$$

for measurement i with noise covariance σ_w.

4.4 Tracking parameters

The MHT with Track Deletion and the DWNA-model requires several parameters to be set. In this section they will be discussed, and the schemes for estimating them will be described.

The following parameters need to be defined:

- σ_w: The standard deviation of the discrete measurement noise.
- σ_v: The standard deviation of the continuous acceleration noise.
- P_G: The probability of a target measurement falling into the validation region of the predicted estimate of the target.
- v_{max}: The assumed maximum velocity of targets.
- β_N and β_C: The density of new targets and the density of clutter.
- P_D, P_O and P_X: The probability of detecting, not detecting and terminating a target, respectively.
- $N_{scan}, r_p rune, K_{best}$: Parameters for different pruning strategies.

4.4.1 Measurement Noise and Error

As mentioned in Section 2.2, limited accuracy creates error in the measurements. However, the error is expected to be small for resolved measurements, as the centroid would is likely to be a good estimate. However, when measurements are merged or split, the error is expected to be greater, especially for the latter.

For split measurements the true location of the target is likely to be somewhere in between the centroid of each of the measurements. The likely location of a target can be found by the average of centroids, weighted by area, giving the centroid of the combined mass of the split measurements. The errors of each of the measurements can be found by the distance to the likely location of the target.

$$\bar{z}_i = \frac{\sum_{j=1}^{n_i} z_{i,j} A_j}{\sum_{j=1}^{n_i} A_j} \tag{4.9}$$

$$e_{i,j} = ||\bar{z}_i - z_{i,j}||_2 \tag{4.10}$$

The largest detection of the split is expected to have less error than the smaller ones as it follows from the preceding equation. Therefore it is likely to be chosen as the target measurement by the MHT, at least if a target has about constant speed. Therefore, for the error of split detections one could consider only the larger of the measurements in the split, as the other ones would be classified as clutter.

Targets of greater size are expected to have a larger error than small ones. Therefore, one could vary the measurement error deviation in terms of the size of the detection. However, this could lead to unwanted results as it could favor assigning targets to smaller measurements as their error of innovation, B_k (3.14), would be less than for larger ones. Clutter are usually small in size. Also, for split measurements of unequal size, this could favor the smaller, "less correct", ones.

Therefore, the measurement standard deviation is set as a constant. It will be defined in terms of the split measurements error, which is the least error measurement in the group of measurements for a split detection. The worst case error is expected to be seen when the largest target generates two equally sized measurements.

4.4.2 Acceleration Noise and Gate Probability

It is expected that most targets will have close to constant speed, but some targets like jet-skis and fast boats may change their speed quickly. These boats are important to track as they are highly relevant for a collision avoidance scenario.

Usually, targets that have a high acceleration are small in size. However, as size is not estimated, σ_v is set constant.

The plant noise covariance will affect the probability of assigning a measurement to a target (Equation 3.5), as well as the size of track gates (Equation 3.24), since it affects the innovation covariance, B_k. The innovation covariance can be expressed in terms of plant noise.

$$\mathbf{B}_k = \mathbf{H}\bar{\mathbf{P}}_k\mathbf{H}^T + \mathbf{R} \tag{4.11}$$

$$= \mathbf{H}(\mathbf{F}\hat{\mathbf{P}}_{k-1}\mathbf{F}^{\mathbf{T}} + \mathbf{Q})\mathbf{H}^T + \mathbf{R} \tag{4.12}$$

Since the measurement covariance, $\mathbf{R}$, is constant, $\mathbf{B}_k$ will only vary on the updated estimate covariance for the track at the previous timestep, $\mathbf{P}_{k-1}$, and

the plant noise, $\mathbf{Q}$. Note, that $\mathbf{P}_{k-1}$ also depends on the plant noise for all timesteps from the beginning of the track and up to $k-1$.

Since most targets are expected to have a low acceleration, the acceleration variance should be set as low as possible. Especially in the case of closely spaced targets and high clutter density this is important since one wants to decrease the likeliness of assigning a target to some close by, but wrong measurement. However, to be able to track high acceleration targets, one must ensure they are at least gated.

Therefore, the gate probability is set high, $P_G = 0.99$, and the noise covariance are to be set as low as possible.

The continuous acceleration noise vector, $w(t)$, for one dimension, is given as

$$\hat{x}_{k+1} = F\hat{x}_k + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} w(t) \tag{4.13}$$

By using an approximation of a ground truth, one can obtain samples of the acceleration in both dimensions to estimate the plant noise covariance.

4.4.3 The Maximum Speed of Targets

The same argument is made for the maximum speed of targets, v_{max}, as for the previous section with σ_v. It is set as low as possible, but such that all initial tracks gates its next measurement. Samples of the speed and the initial speed of will be obtained from a ground truth, in which v_{max} will be set according to the maximum value.

4.4.4 Clutter and New Target Densities

The densities of clutter and new targets can be expressed in terms of the expected number of clutter and the expected number of new targets per scan, by calculating the size of the observable area.

$$\beta_C = \frac{\alpha_C}{V} \qquad \beta_N = \frac{\alpha_N}{V} \tag{4.14}$$

where α_C and α_N are the number of clutter and new targets per scan, respectively. The size of the observable area can be found by counting non-masked cells of the combined satellite and average mask, $M_{combined}$, and multiplying it by the area of a cell.

The expected number of clutter and new targets per scan can be estimated by establishing a ground truth for a set of data, and counting the number of clutter measurements, and the number of tracks.

$$\alpha_C = \frac{N_C}{L} \qquad \alpha_N = \frac{N_T}{L} \tag{4.15}$$

where N_C and N_T is the number of clutter measurements and the number of tracks in the dataset with L scans.

One may want to tune these parameters to increase the robustness of the MHT. If the clutter density is increased, while the new target density is decreased, one decreases the likeliness of a new track. This may result in less false tracks, but may come at the expense of loss of true tracks.

4.4.5 Target Probabilities

The probability of detection, misdetection and deletion can also be estimated from a ground truth, by examining each track for the number of detections and the track length, and then using the average values for all tracks.

$$\bar{P}_{D,\,i} = \frac{N_{detections,\,i}}{L_{track,\,i}} \qquad \bar{P}_{X,\,i} = \frac{1}{L_{track,\,i}} \qquad \bar{P}_{O,\,i} = 1 - \bar{P}_{D,\,i} - \bar{P}_{X,\,i} \tag{4.16}$$

However, the target probabilities is expected to greatly vary depending on the target, and the average may not be suited. Usually, targets of greater size will have a higher probability of detection than smaller targets, as they are easier to detect by the radar. The actual observable area could also be affected by the size, as there could be areas where only large targets are detected. Therefore, size can affect the probability of deletion as well. Slow targets will also have a lower probability of deletion as they stay in the observable area for a longer time.

Targets that are easily detected are the "easy" cases for the MHT. Since the target generates measurements regularly there will be low uncertainty in state estimates, leading to a low innovation covariance, and the likelihood of assigning measurements to it will be high. A low innovation covariance will also lead to a small track gate and less likelihood of gating clutter. A lower probability of detection than the actual probability for the target can be set, as the likelihood of detecting the target and assigning it a measurement is greater than the alternative, which is classifying the measurement as clutter. However, this may not be the applicable with a high clutter density, but this is not expected to be the case in this scenario.

A too high probability of deletion would too easy terminate tracks that are misdetected for consequent timesteps, and rather give birth to a new target, as it is detected again. A case is illustrated in 4.3, along with the resulting probabilities. The opposite case of P_X being very low, is that the deletion hypothesis may easily be pruned away. Also, the loss of a whole track loss

is possible if it is short, as the probability will make a track less likely than classifying all measurements as clutter.

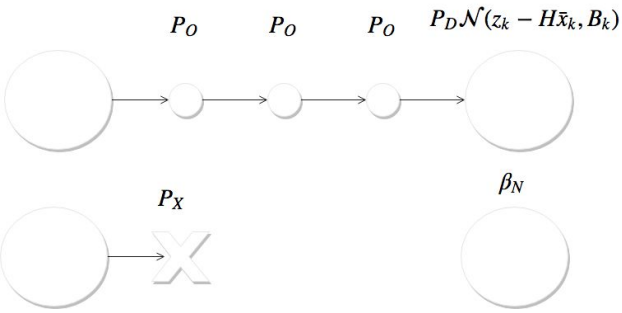


Figure 4.3: A case of a target being misdetected three times. The alternative hypotheses that are the target is deleted and a new is born is shown underneath.

Therefore, as MHT is likely to be robust to a lower probability of detection than the reality for easily detected targets, P_O should be increased from $\bar{P}_O$, while decreasing P_D and P_X, to increase the robustness against false tracks.

4.4.6 Pruning Parameters

The pruning parameters are important to be set sufficiently high to be able to generate the "correct" hypotheses at the end of the tracking. The "correct" hypothesis is defined to be the most probable hypothesis if one did not prune at all. Pruning to "hard" could lead to one not generating the correct hypothesis or its parent.

The parameters N_{scan} and K_{best} control the depth and width of the hypothesis tree. They are very much dependant on each other in finding the values needed to, at least most of the time, generate the best hypothesis.

Without N-Scan ($N_{scan} = \text{inf}$), one could end up with severe errors in the tracking. At a time step one could generate top hypotheses that have very similar probability, e.g both being likely. If the outcome is similar as well, their child hypotheses will have similar probabilities as well. Then, if this happens for several timesteps, one would eventually "fill up" the set of K-best hypotheses. This is a typical case for targets moving together. The result is that hypotheses with low probability events, like target deletion and birth, will not be generated and tracks would neither be created or deleted, leading to severe errors in the tracking. With $N - scan$ one would eventually "settle" on these similar hypotheses back in time, allowing one to have more diverse hypotheses for the recent time.

A consequence of the dependency is that if K_{best} is just sufficient to generate the correct hypotheses for a value of N_{scan}, one must increase it if one increases N_{scan}.

The ratio pruning parameter (Equation 3.25), r_{prune}, must be set sufficiently high such that the ratios between hypotheses that lead to the correct one and the most likely hypotheses at each of the time steps, are less or equal to r_{prune}. The ratio pruning will be set quite high and just used to remove the extremely unlikely hypotheses, to reduce the computation time.

5 | Code and Graphical User Interfaces

In this section, the code will be presented. Only an overview and the essential parts will be presented, due to its extensiveness. The focus will be on the three different Graphical User Interface (GUI)s that have been made.

All the code has been implemented using Python, except for the algorithm by Murty, coded in *C*, with a Python interface, by Jonatan Olofsson. The Python libraries used are *OpenCV*, *Numpy*, *Scipy*, *PyQt* and *Matplotlib*. Otherwise, all code is made from scratch.

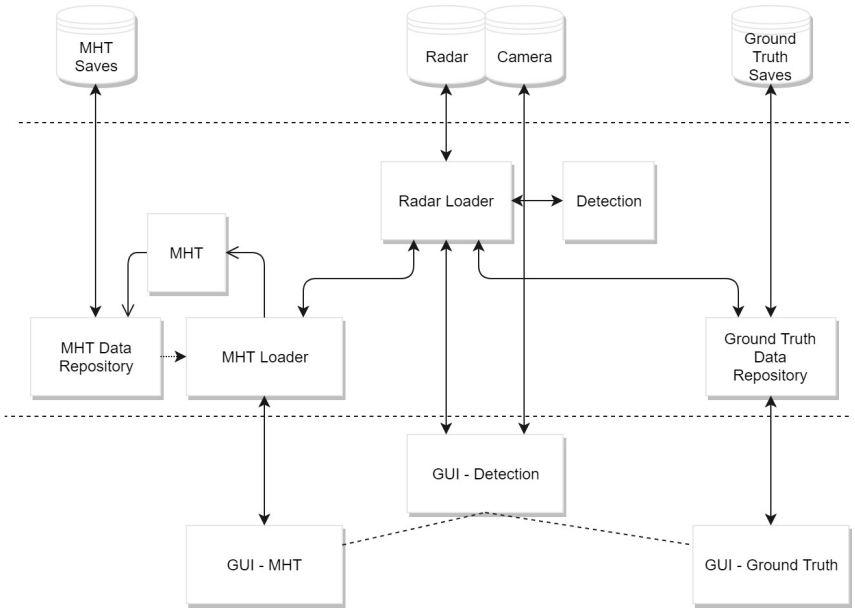


Figure 5.1: The data flow for the three different views of the GUI

5.1 Detection GUI

The detection view was made for browsing radar and camera output, and examining the results of the detection.

The radar output is shown on top of a satellite image of the area, in addition to

showing the satellite mask and average mask used in the detection. In addition it has the following features:

- Browse to scan index.
- Jump to next or previous scan that has a measurement.
- Navigation using arrow keys.
- Show time and date of scan.
- Disable camera to speed up loading.
- Enable camera enhancing. Improves camera image with adaptive histogram normalization (slows down browsing).

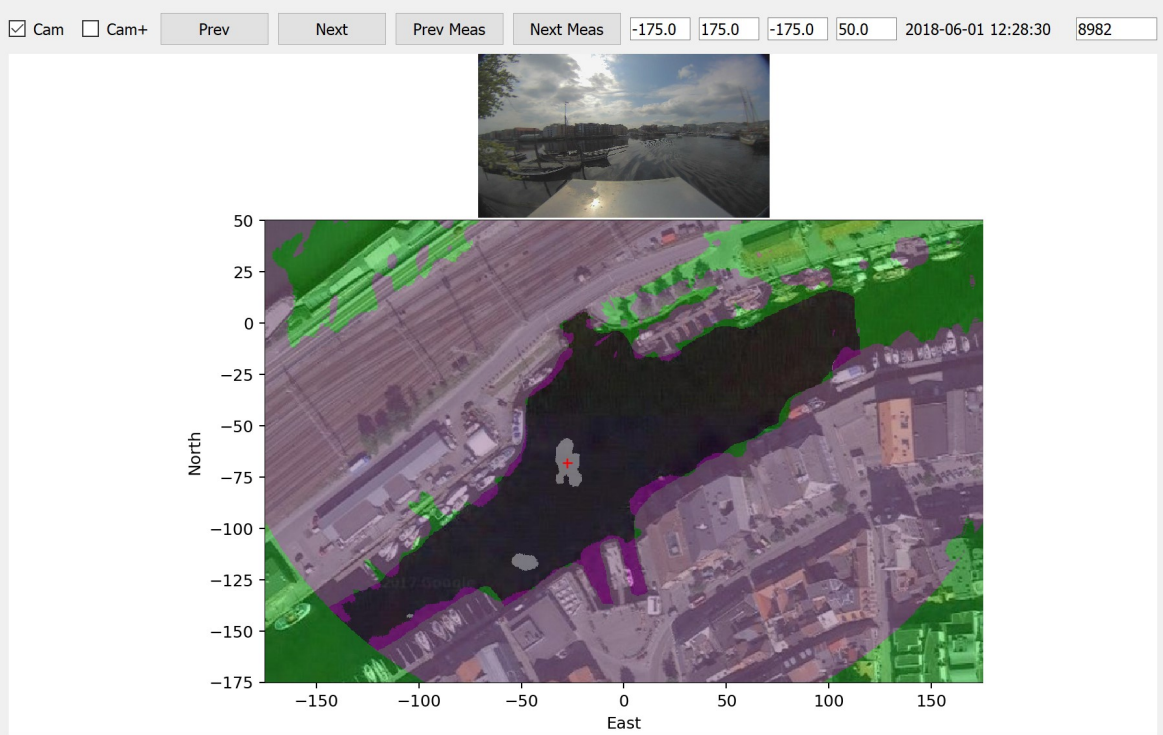


Figure 5.2: The detection GUI

The GUI is made using *PyQt*, while the actual view of the radar and camera is a *Matplotlib*-plot. Because of the slow rendering of the latter, work has been done to improve the speed, like caching of the radar background.

5.2 MHT GUI

This GUI builds on the detection one, and is made to analyze MHT results, and to aid in debugging the MHT code when it was created.

The view is centered around the list of clusters. The list shows information about each cluster, like the number of targets in the top hypothesis. This

makes it easy to navigate the results of the MHT. For instance if the ratio is low, this suggests an uncertain result and one may want to examine it more closely.

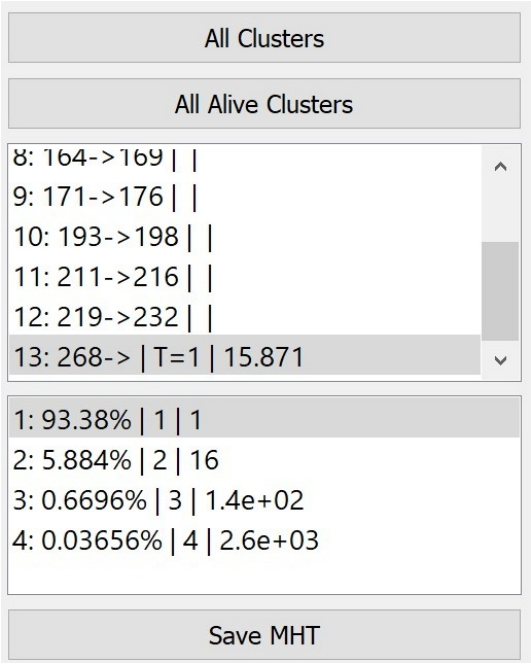


Figure 5.3: The sidebar of the MHT GUI. The top list are the clusters, while the bottom are the hypotheses for the cluster selected.

One or several clusters can be selected, showing the most likely hypotheses of each. This is useful when there are several clusters present at the same timestep. If a specific cluster is selected the GUI brings up its sorted list of hypotheses, with the probability of each. Then, one can select other hypotheses than the most likely one, showing its tracks instead.

Debugging of K_{best} and $r_{pruning}$ is made easy. Each hypothesis shows its maximum K-value, K_{max}, and its maximum ratio-value $r_{prune,max}$. The maximum K-value of a hypothesis is the worst ranking of it, or one of its parents, in its respective timestep, while the maximum ratio-value is similar, only for the ratio between the hypothesis and the most likely one.

Similar to the detection view one can navigate the scans. In addition, one can select a specific MHT timestep that can be greater than the timestep of the scan, thereby viewing the tracks for that scan based on the hypothesis at the MHT timestep. Also, one can set a track cutoff timestep, which makes the view not show any track estimates before that timestep. This is implemented because there may be many and long tracks which could overlap and create a chaotic view of the tracking results.

The GUI requests the *MHT-loader* for the MHT-data at a given timestep. If the MHT data is present it is given. If not, it instructs the *MHT* to process the scans from where it was last loaded up until the given timestep. This makes it

possible to analyze data on the go, not needing to process the complete dataset first. Also, one has the possibly to save the MHT-data and load it another time.

5.3 Ground Truth GUI

This view is made to create a ground truth for the radar data.

At a scan, a list of measurements is presented where each can be selected to create a new track. Also, the current set of alive tracks are shown, which can be assigned one or several of the measurements in the scan. If several are selected it indicates that the measurements are split. As well, a track can be assigned no measurement and also be set as deleted.

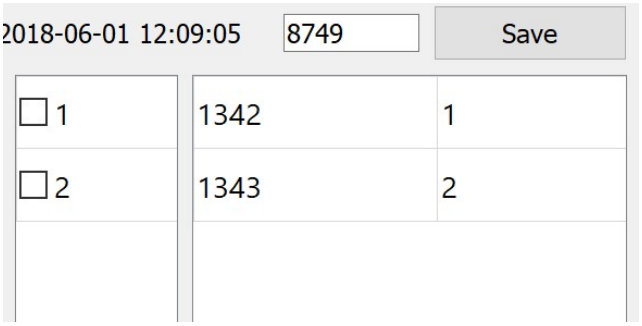


Figure 5.4: The sidebar of the Ground Truth GUI. To the left, measurements can be selected to start a new track. To the right are two tracks which can be assigned a measurement. Here, they are assigned measurement 1 and 2 respectively.

The current set of alive tracks are shown, and the view is updated if the assignment of a track is changed. One may also go back and forth between scans and alter the ground truth. This makes it easy to create the ground truth for one track at the time. Lastly, the ground truth data can be saved at anytime, to either create a checkpoint or alter previous data.

5.4 Radar and Track View

Here, a description of what is visible in the center view of the GUIs.

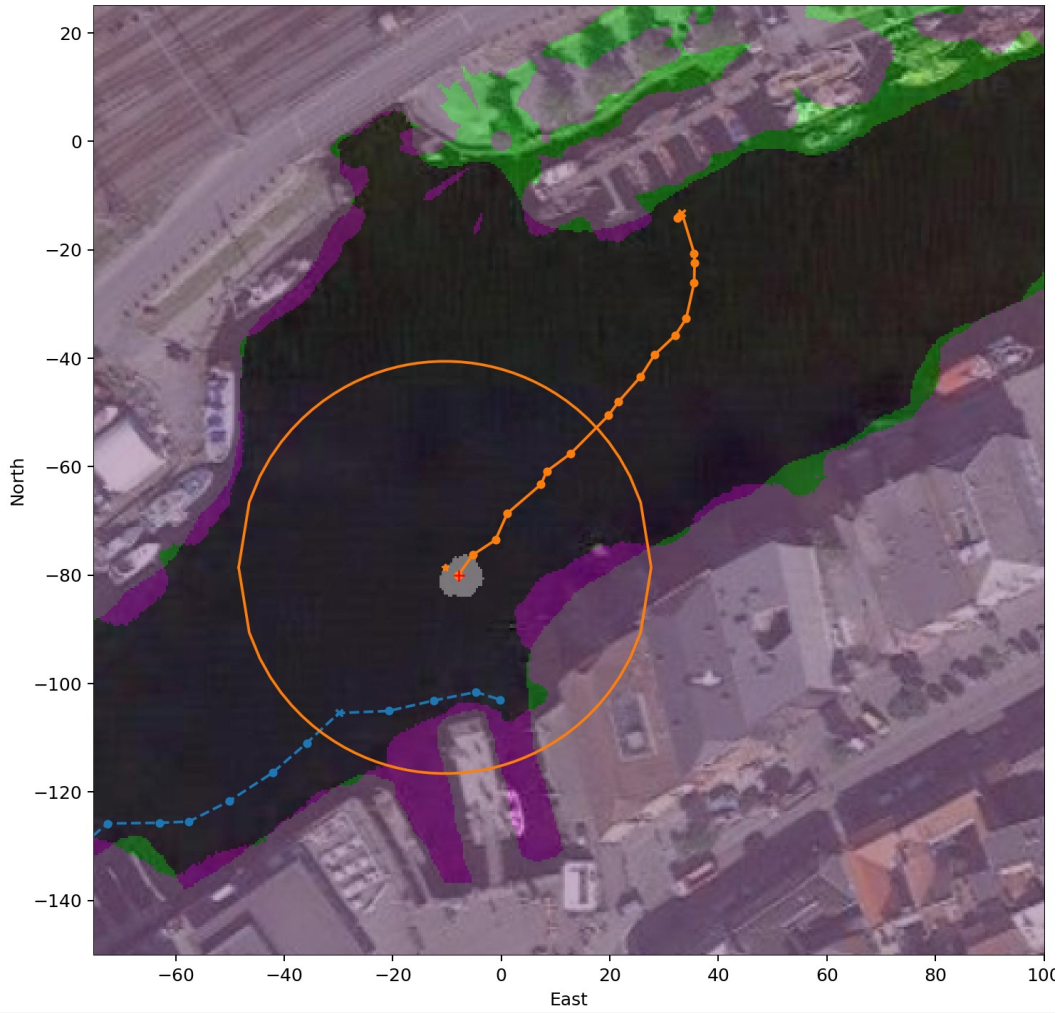


Figure 5.5: The Track View. Two tracks are shown. One is terminated, while the other alive.

Detection view:

- *Green*: Satellite mask.
- *Purple*: Average mask.
- *White*: Detection cells (without filtering).
- *Red cross*: Detection centroid.

Track view:

- Red plus sign: Centroid of a measurement.
- Line: Alive track.
- Stippled line: Terminated track
- Filled circle: An estimate and the target was detected.
- X: An estimate and the target was not detected.
- Big circle: Track gate of previous estimate.

6 | Results

6.1 Experimental Data

Data was collected from 00:00 Friday, June 1st to 23:55 Thursday, June 7th, resulting in seven days of data. Samples were taken from the radar and the camera every 5 seconds, giving 120960 samples of each. The data uses 47.7 GB of disk space, averaging at 394 KB per combined radar and camera sample.

6.2 Satellite and Average Mask

The satellite mask created for the area is shown in 6.1.

Figure 6.1: The Satellite Mask (in green) shown on top of a satellite image of the area

An average mask were created by using all samples in dataset. The p-value was set by examining different values. The final p-value was set to be 0.01.

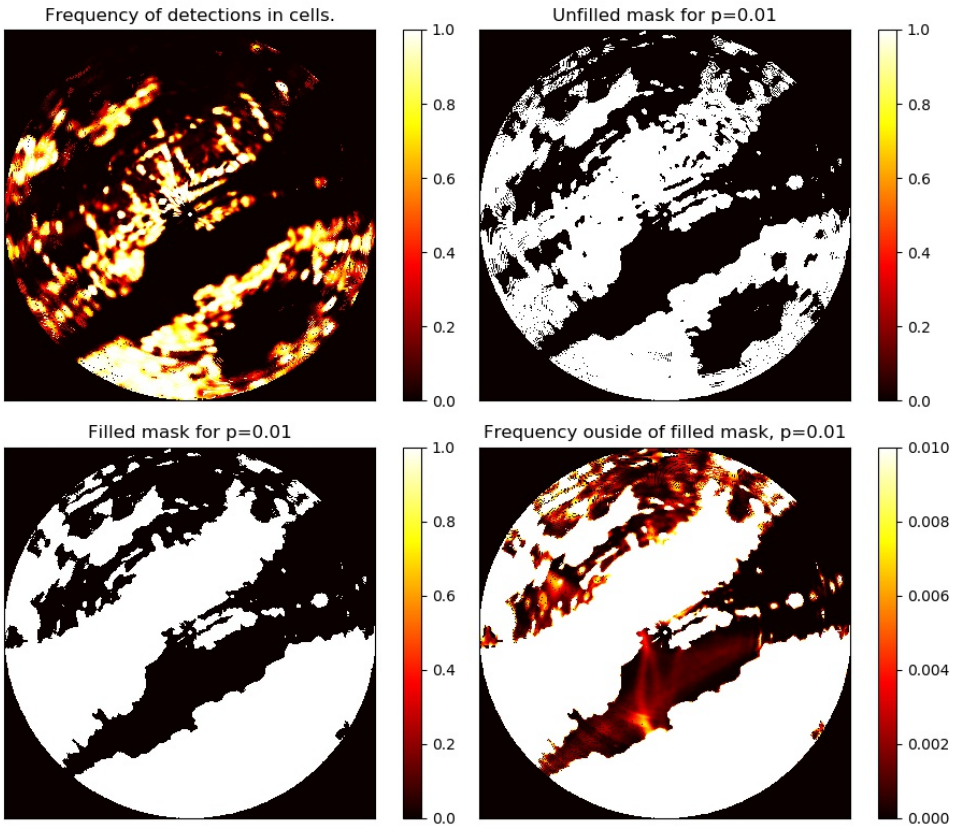


Figure 6.2: Plots illustrating the process of finding the p value for the average mask.

The combined Satellite and Average Mask for $p = 0.01$ is shown in the following figure. The combined mask is used throughout all following experiments.

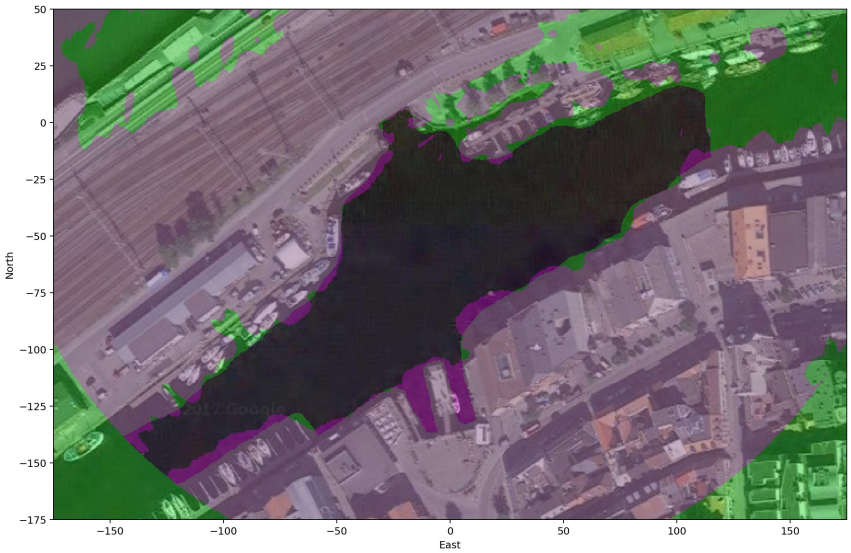


Figure 6.3: The Satellite (green), and the Average Mask (purple) for $p = 0.01$. The black area in the image remains as the observable area.

The observable area was found by counting the non-zero pixels (false pixels) as described in Section 4.4.4, and then scaling.

$$A_{obs} = 14117\,m^2 \tag{6.1}$$

6.3 Ground Truth

By using the GUI presented in Section 5.3, a ground truth for target-measurement assignments were made. It is made for a subset of the data, from 00:00 June 1st to 16:00 June 2nd, resulting in 40 hours of ground truth data. Importantly, the ground truth dataset is well distributed in target activity. It is assumed to be two periods with low activity (morning/night) and two periods with high activity (mid-day Friday and mid-day Saturday). Figure 6.11 later confirms this. Also, the dataset has samples from days of the week assumed to have high activity. Along with good weather when data was sampled, this is expected to present difficult cases for the MHT as the number of targets are high. However, because of the good weather, the water was calm, and a low amount of clutter originating from waves is therefore expected.

In contrast to the MHT, the targets of the ground truth were allowed to be assigned more than one measurement, and also a single measurement was allowed to be assigned to more than one track. In other words, the ground truth were created with the possibility of both merged and split measurements. The camera was used to verify actual target presence.

The ground truth was created with both the mask filters enabled, in addition to the area filter with a, presumably, low threshold of $20\,pixel^2 \approx 3\,m^2$.

6.3.1 Ground Truth Track Estimates

The ground truth only has measurement associations. Therefore, schemes are used to approximate the true target location.

- Resolved: Centroid
- Merged: Centroid
- Split: The centroid of the combined area of the detections (Equation 4.9).

By examining the contours and the paths of the targets, the resolved and split measurements schemes seemed to give a good fit of what seemed to be the true target location. For merged measurements this was not case. However, since there are few merged measurements in the dataset it is expected to not affect the proceeding results appreciably.

6.3.2 Ground Truth Statistics

Measurements and Tracks

An overview of the number of resolved, merged, split and clutter measurements are given in the following table. "Clutter MHT" is the number of clutter measurements if only one of the measurements in a split is classified as a target measurement, and the rest is classified as clutter. This is the number "seen" by the MHT, as it assumes split measurements are not present.

Type	Number	Share [%]
Resolved	6224	57.2
Split	523	4.81
Merged	36	0.33
Clutter	4096	37.7
Total	10879	100
Clutter MHT	4096	40.0

Table 6.1: The different types of detections and their respective numbers in the ground truth.

	Avg	Min	Max
P_D [%]	91.0	62.5	98.5
P_O [%]	3.59	0.00	32.14
P_X [%]	5.40	0.75	20.00
Track Length [scans]	23.4	5.0	134
Track Length [time]	1m 57 s	0m 25s	11m 10s

Table 6.2: Statistics for the 292 tracks in the ground truth.

By using the size of the observable area and the number of tracks and number of MHT-clutter, we obtain:

Clutter per scan	0.1514
New Targets per scan	0.01014
Clutter intensity β_C	1.07e-5
New Target intensity β_N	7.18e-7

Table 6.3: Clutter and target intensity from the ground truth.

Clutter/Detection ratio, location. Start/End of tracks, location.

6.4 Detection

Here, effects the different parameters of the size filter and the multiple filter will be examined. The ground truth will be used as a reference. The number of true detections and clutter are the ones expected to be used by the MHT. Therefore, for a split measurement only the largest of the detections are classified as a true detection, while the rest is classified as clutter.

6.4.1 Performance Measures

The size filter and the multiple filter are both binary classifiers, classifying a detection either as a true detection or clutter. One has the following relation to binary classification.

- True Positive (TP): Target detection kept.
- False Positive (FP): Clutter kept.
- False Negative (FN): Target detection removed.
- True Negative (TN): Clutter removed.

In addition, one has the following metrics:

$$sensitivity = \frac{TP}{TP + FN} \tag{6.2}$$

$$specificity = \frac{TN}{TN + FP} \tag{6.3}$$

These are also referred to as true positive rate and false negative rate, or recall and precision, respectively. Preferably, the sensitivity and specificity approaches one. In a ROC-curve the specificity are plotted against $1 - specificity$ for a range of parameter values.

6.4.2 Size Filter

One wants to find the optimal A_{min} for the size filter.

Figure 6.4 shows the ROC-curve. Increasing A_{min} decreases the sensitivity, but increases the specificity, as expected.

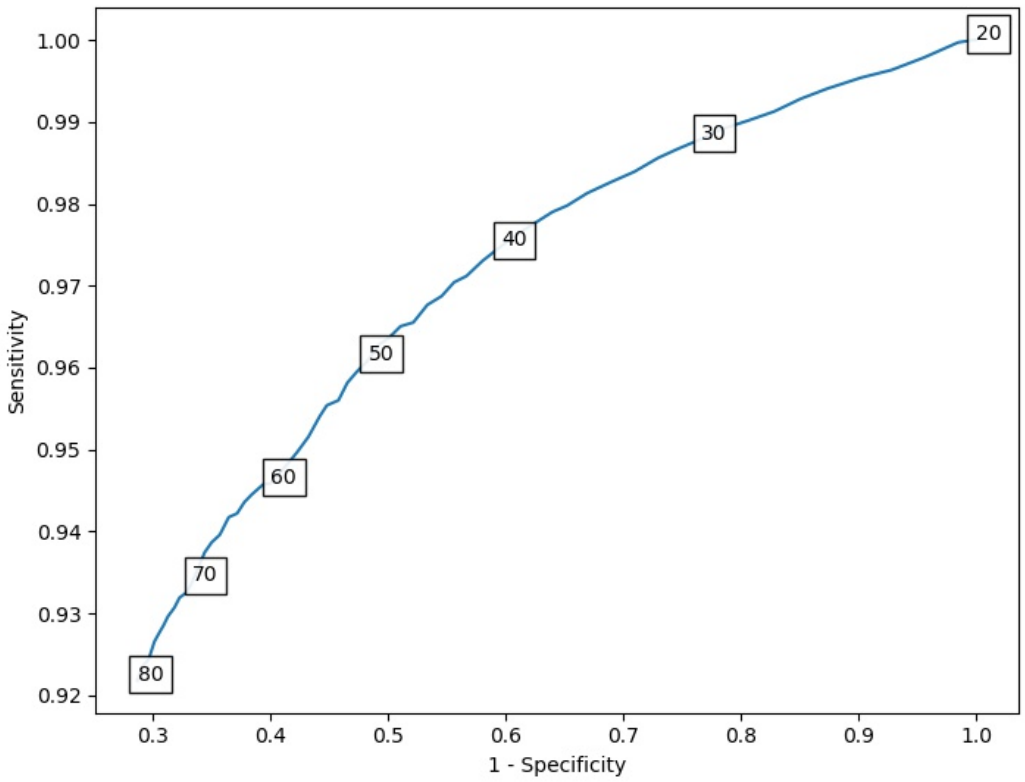


Figure 6.4: ROC-curve for different values of $A_{min}[pixel^2]$

Tracks with small targets may already have a low probability of detection. Therefore, the effects of the size filter is examined on the ground truth tracks. If the size filter removes detections at beginning or ends of a track, the track is clipped to a shorter track.

Figure 6.5 shows worst case metrics for the tracks. The measures used are the worst probability of detection of all tracks and the shortest track length. Also, the number of tracks that are completely removed by the filter is shown. For $A_{min} \in [20, 34]\, pixel^2$ the worst probability of detection is unchanged. As A_{min} is further increased the P_D drops all the way to 0.5. In the end, all detections of a track is filtered out. The track affected was identified and its detections are shown in Figure 6.6.

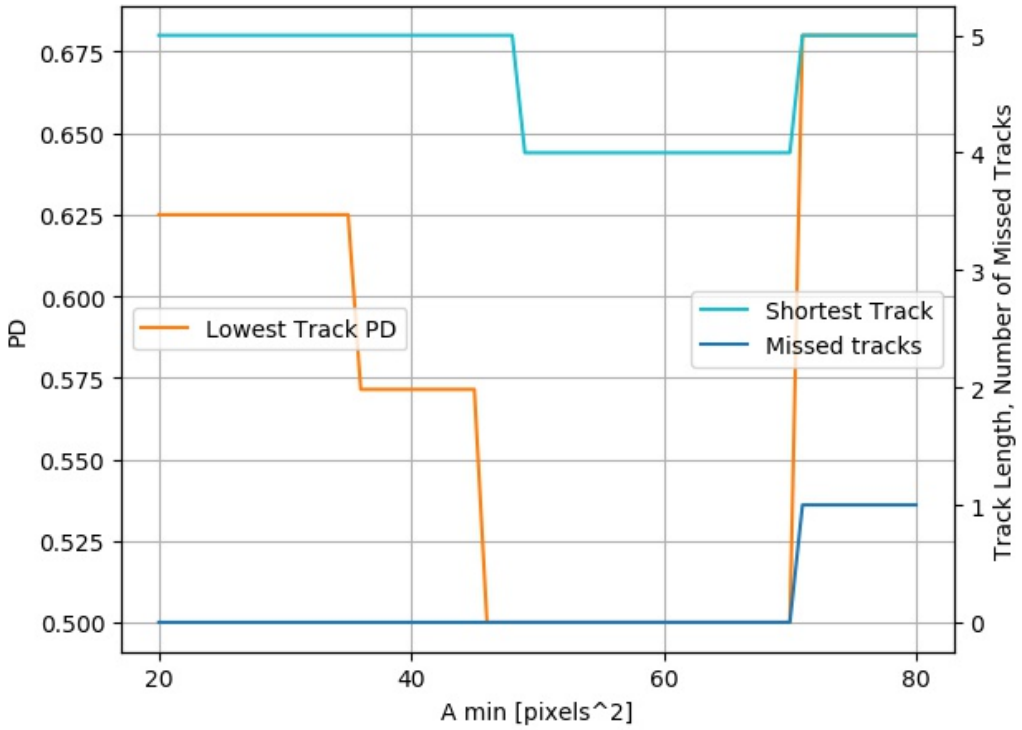


Figure 6.5: Examining the effects of the size filter threshold on tracks.

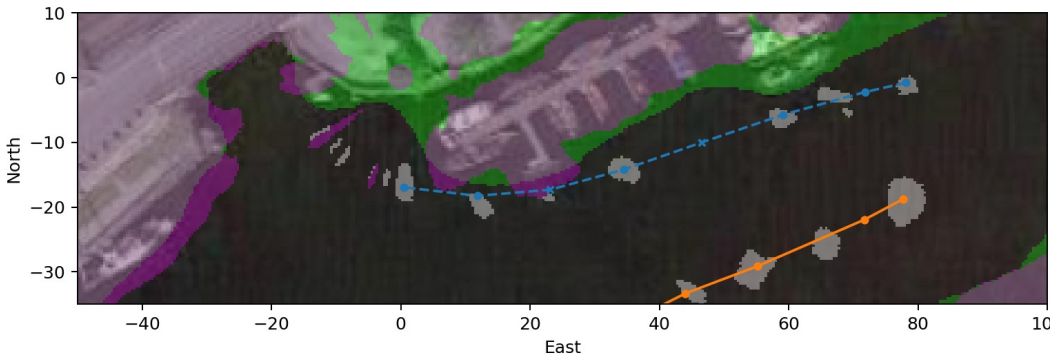


Figure 6.6: The target with the lowest P_D for $A_{min} \geq 35\,pixel^2$. All seven detections (in 8 timesteps) are shown, together with the track in blue.

Figure 6.7: The target with the lowest P_D for $A_{min} \geq 35\,pixel^2$

6.4.3 Multiple Filter

For the multiple filter, the thresholds $A_{multiple}$ and $r_{multiple}$ need to be found. The effects are examined for all scans with two or more detections.

Figure 6.8 shows the ROC-curve. The reader are advised to notice the scale of the plot. For all combinations, the filter shows high sensitivity, while greater variation is seen for specificity. Regardless of $r_{multiple}$, $A_{multiple} = 800[pixel^2]$ seems like the optimal value, where about 0.5% of the true detections are removed. $r_{multiple} = 1.0$ is the optimal range, removing 24% of the clutter.

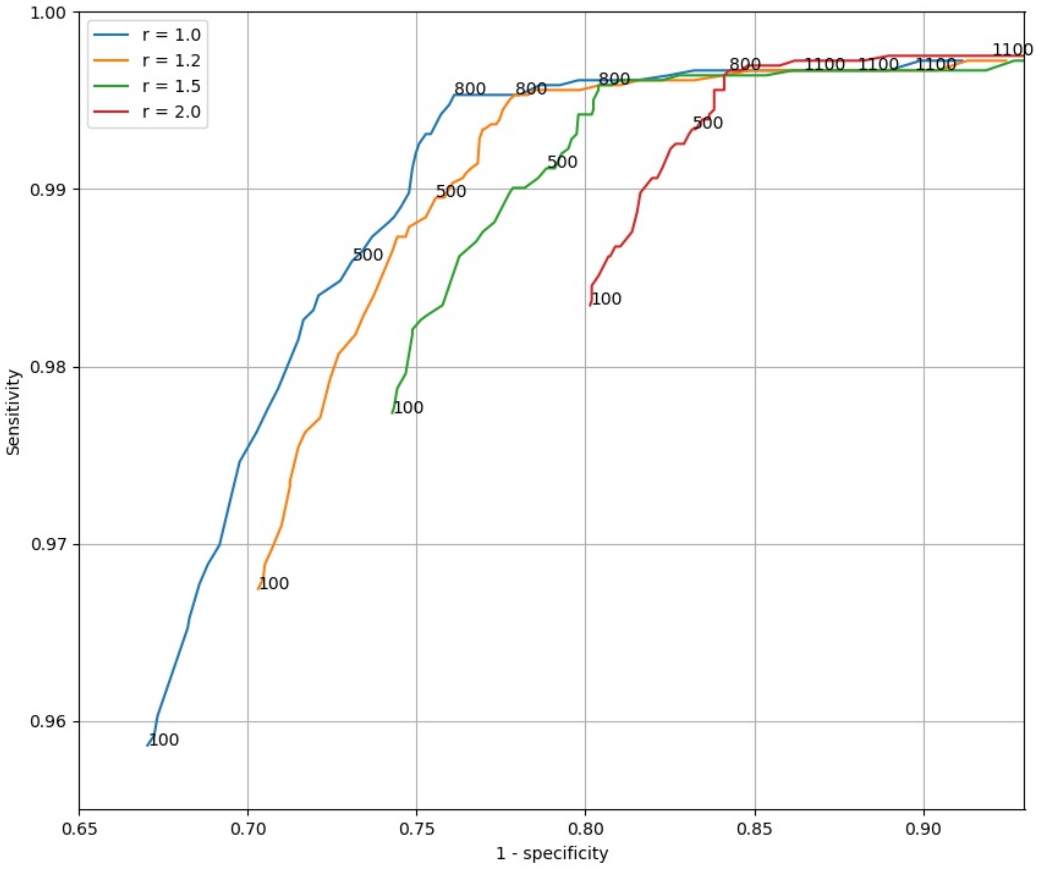


Figure 6.8: ROC-curves for different values of $A_{multiple}$ and $r_{multiple}$. $A_{multiple}$ values are in $pixel^2$ and are shown as text labels on the lines.

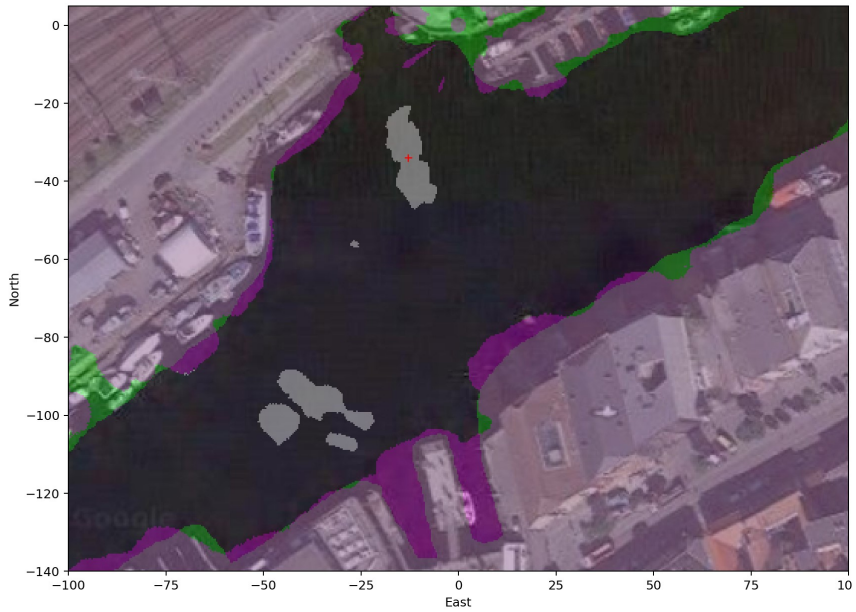


Figure 6.9: Example of multiples being filtered out. The true target detection is shown with a red cross. The multiples are in the bottom left.

6.4.4 Statistics After Filtering

Presented are the statistics for tracks and clutter density after the size and the multiple filter is applied on the ground truth. The parameters used are $A_{min} = 34$, $A_{multiple} = 800$ and $r_{multiple} = 1.0$. The total number of clutter (seen by MHT) is 2566, giving a 41 % decrease. This is done without affecting the probability of detection, neither the average or worst cases. The most notable effect is that the average and maximum track length are slightly shorter.

Type	Number	Share [%]	Change [%]
Resolved	6172	68.07	-0.84
Split	523	5.769	0.00
Merged	36	0.397	0.00
Clutter	2335	25.76	-43.0
Total	9066	100	-16.8
Clutter MHT	2566	28.30	-41.1

Table 6.4: The different types of detections and their respective numbers in the ground truth post filtering with area and multiple filter. The change relative to the ground truth is shown.

	Avg	Min	Max
P_D [%]	91.0	62.5	98.4
P_O [%]	3.49	0.00	32.14
P_X [%]	5.46	0.76	20.00
Track Length [scans]	23.2	5.0	131
Track Length [time]	1m 56 s	0m 25s	10m 55s

Table 6.5: Statistics for the 292 tracks in the ground truth post filtering with area and multiple filter.

Clutter per scan	0.0891
New Targets per scan	0.01014
Clutter intensity, β_C	6.31e-6
New Target intensity β_N	7.18e-7

Table 6.6: Clutter and target intensity from the ground truth after filtering with area and multiple filter.

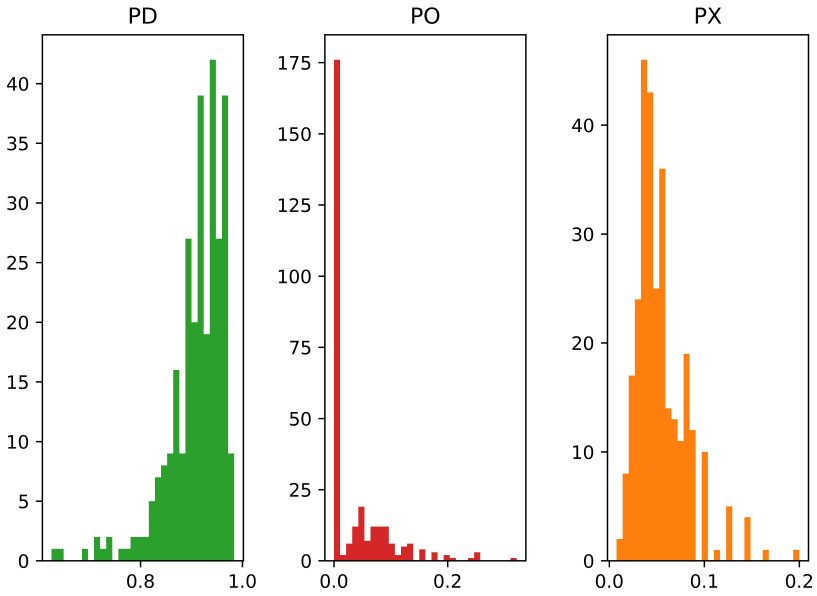


Figure 6.10: The distribution of target probabilities after detection filtering.

Figure 6.11 shows the clutter and target per scan, using a running average. The running average is close to the average track length. It can be seen that the two correlate, and also that they have a large variation. This may suggest that the clutter density should be higher than its mean to reject clutter for the cases when its the most present.

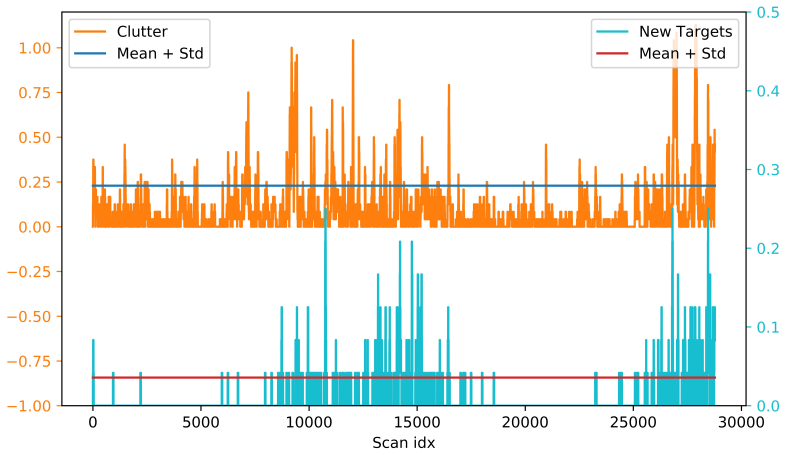


Figure 6.11: The number of clutter and the number of new targets per scan (after filtering). A running average of 2 minutes ($N = 24$) is used to smooth out the data.

6.5 Estimation of Measurement, Plant Noise and Maximum Speed

6.5.1 Measurement Noise

The measurement noise is estimated according to Equation 4.10, using all split measurement groups from the ground truth. A histogram of the errors are seen in Figure 6.12. A value of $\sigma_w = 8.5/3 = 2.833$ was chosen, as it fit the distribution of the errors in the histogram well.

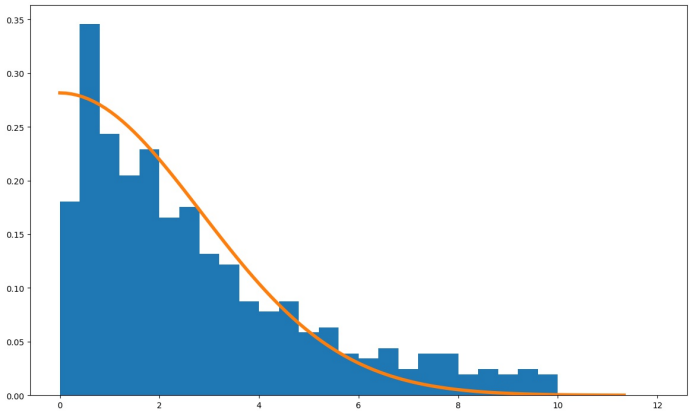


Figure 6.12: Normalized histogram of the split detection error for all split measurement groups in the ground truth. The normal distribution with $\sigma = 2.833$ is shown as an overlay.

6.5.2 Maximum Initial Speed

The initial speed is calculated for all tracks that are detected for their first two timesteps. Figure 6.13 shows the speed samples. The maximum speed of all samples was $5.63\,m/s$.

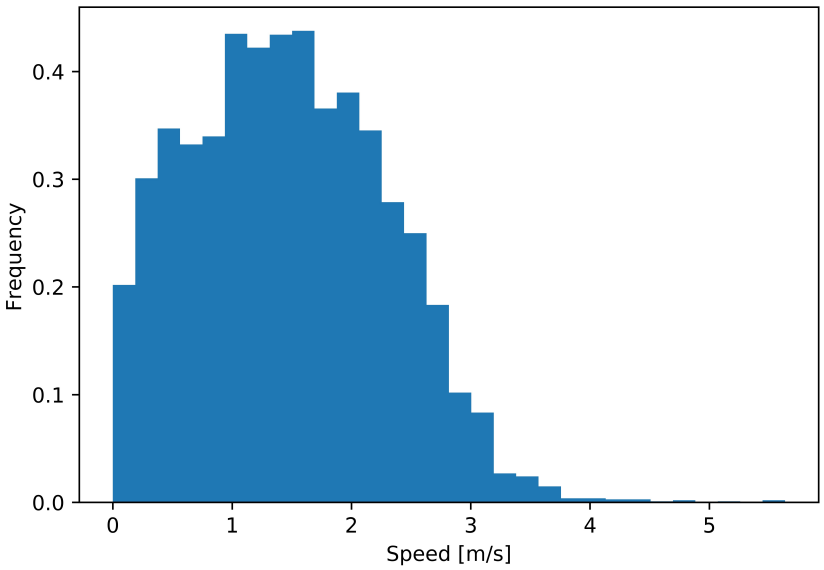


Figure 6.13: Histogram showing the distribution of speed samples from ground truth.

By setting $v_{max} = 5.63$, $\sigma_v = 0.6$ and $\sigma_w = 2.833$ the maximum speeding target was confirmed to be gated. Figure 6.14 shows this. The σ_v used is the one found in the next section.

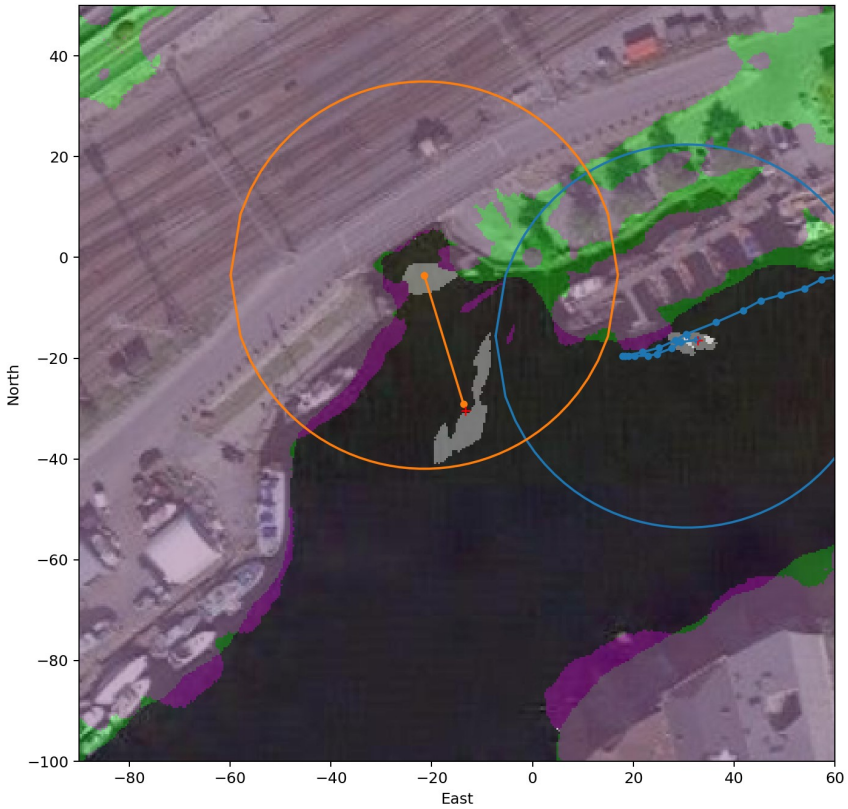


Figure 6.14: The target with the maximum speed in the ground truth (orange). The track gate is shown for $P_G = 0.99$, $\sigma_w = 2.833$, $\sigma_v = 0.6$.

6.5.3 Plant Noise

Acceleration samples will be impacted by measurement noise, with small errors in measurements potentially leading to great noise in the samples. Since the ground truth measurements are not really free from noise, the acceleration samples are obtained from filtered estimates. $\sigma_w = 2.833$ is set constant throughout this section, and $v_{max} = 5.63$. All samples are obtained with Equation 4.13, and the samples of each dimension are combined.

First, $\sigma_v = 2.0$. This is a high value, which smooths out the estimates only slightly to remove the outliers caused by noise. The highest acceleration sample is found and the resulting track along with the track gate is shown in Figure 6.15.

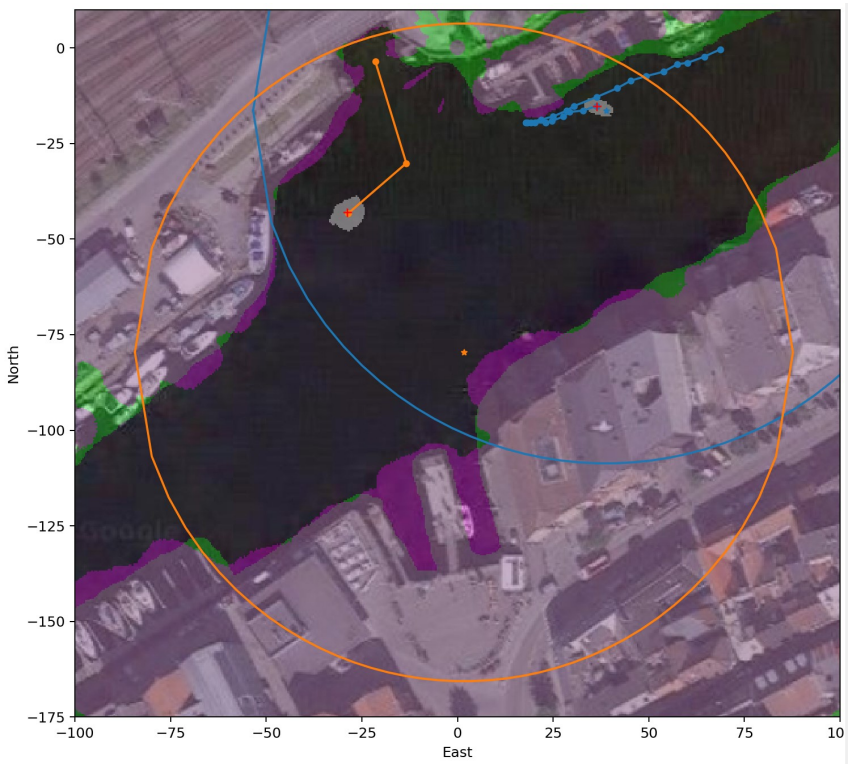


Figure 6.15: The track gate for the highest acceleration maneuver. Estimates filtered with $\sigma_v = 2.0$

By iteratively lowering σ_v from 2.0, a value of 0.6 was found to give a track gate that just gates the highest accelerating target. The resulting track gate is shown in Figure 6.16. A normal distribution with $\sigma_v = 0.6$ is shown on top of the distribution of the samples obtained with $\sigma_v = 1.0$ in Figure 6.17 and Figure 6.18. It can be seen that it does not fit the distribution of samples well, but covers the outliers (like the maximum acceleration target). The distribution of samples when filtering with $\sigma_v = 0.6$ is shown in Figure 6.19.

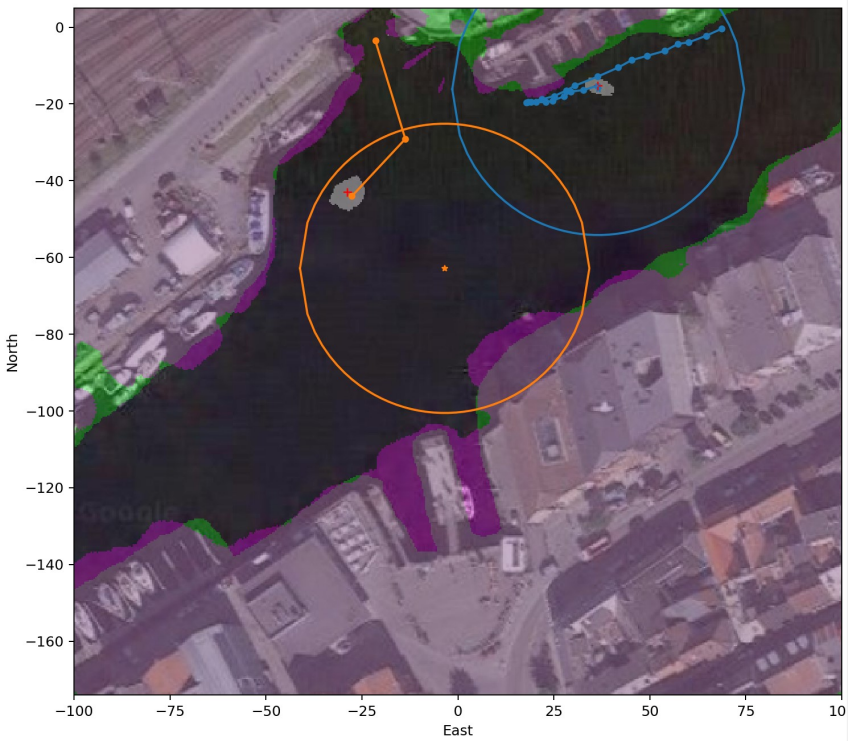


Figure 6.16: The track gate for the highest acceleration maneuver. Estimates filtered with $\sigma_v = 0.6$

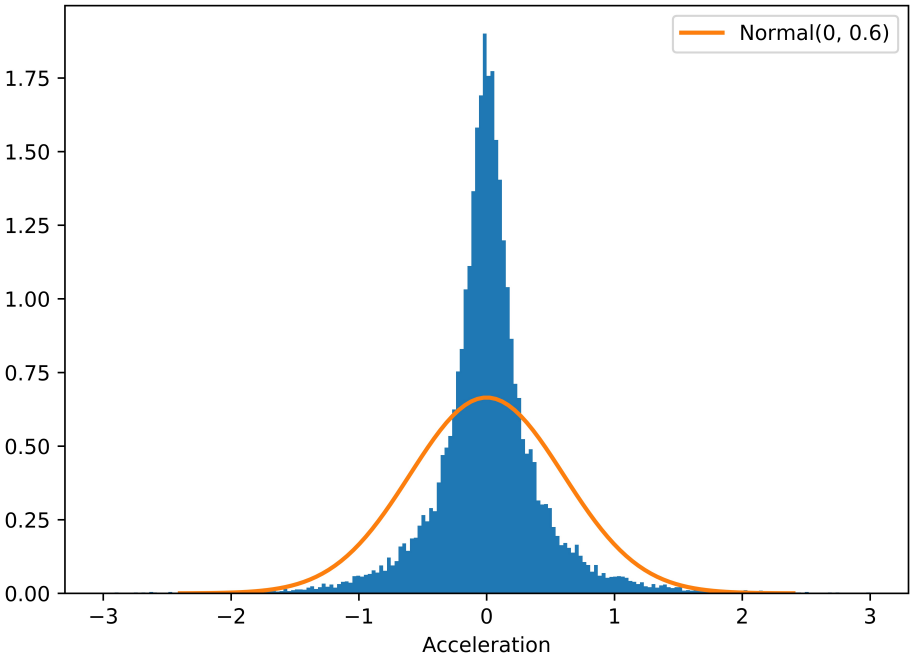


Figure 6.17: The distribution of acceleration samples for $\sigma_v = 2.0$. The distribution is normalized such that the area under the bars equals 1.

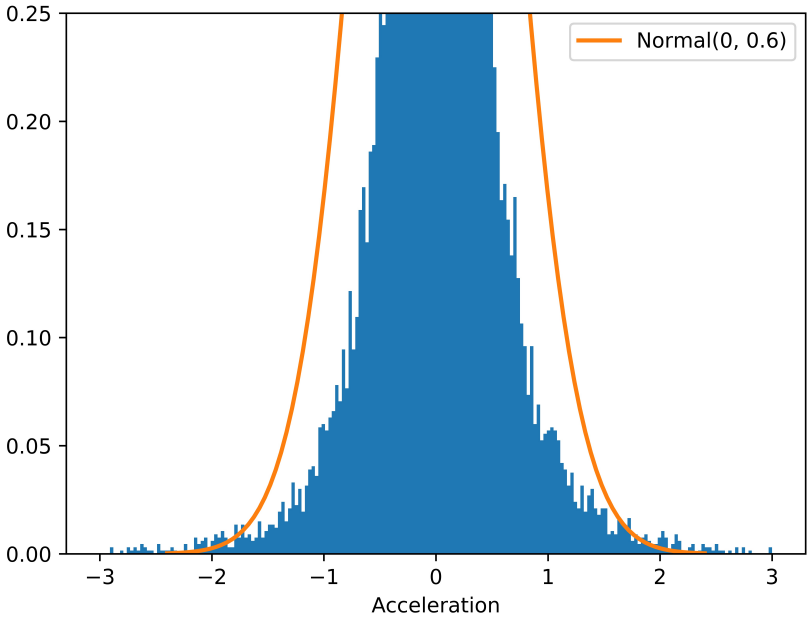


Figure 6.18: The distribution of acceleration samples for $\sigma_v = 2.0$, with clipped y-axis. The distribution is normalized such that the area under the bars equals 1.

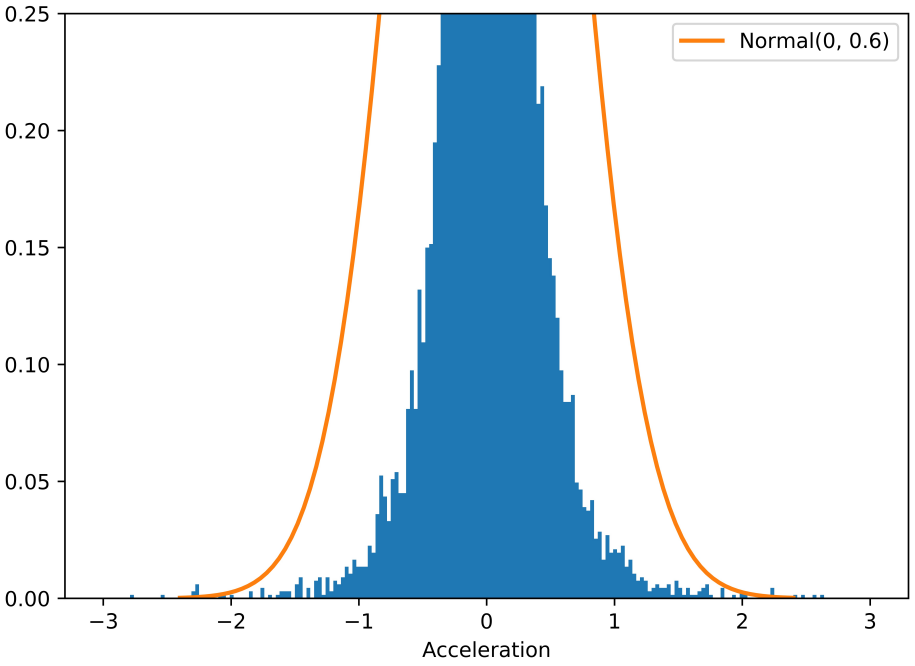


Figure 6.19: The distribution of acceleration samples for $\sigma_v = 0.6$, with clipped y-axis. The distribution is normalized such that the area under the bars equals 1.

6.6 MHT evaluated on Ground Truth

To use the MHT to analyze the rest of the dataset, one needs to find the parameters best suited. Also, one is in need of getting an intuition of how well the algorithm performs compared to the ground truth to be able to establish a level of confidence in the results for the rest of the dataset. The OSPA metric will be used to evaluate the algorithm for different set of parameter values. Also the RMSE of the cardinality and the total number of estimated tracks will be used to establish a level of confidence.

6.6.1 Ground Truth Track Processing

The ground truth tracks are created with less detection filtering applied (no multiple and low area threshold). Therefore they are clipped as described in Section 6.4.2 to give an equal detection basis for the MHT and the ground truth.

Also, tracks are smoothed with the Rauch–Tung–Striebel (RTS) method to give better estimates for the ground truth when there are missed detections in the track. The measurement and plant noise and the initial state covariance is equal to that to be used by the MHT.

6.6.2 Post-Processing

By examining some preliminary results of the MHT, many false tracks, with no speed, appeared near shore due to clutter. Therefore, tracks are removed if all position estimates are within a radius of the mean of the estimates. By examination this threshold was set to $5\,m$.

6.6.3 Trial Setup

Some of the parameters are set constant for all trials. These are the ones identified in the previous sections.

- $\sigma_v = 0.6$
- $\sigma_w = 2.833$
- $v_{max} = 5.76$
- $P_G = 0.99$
- $A_{min} = 34 pixel^2$
- $A_{multiple} = 800 pixel^2$
- $r_{range} = 1.0$

The average estimated values from the ground truth will be used as a starting point for the parameters that are to be tuned. The maximum hypothesis at the very last timestep is used for providing the set of estimates for all timesteps for

the RMSE cardinality and OSPA. This is reasonable since this is an evaluation of the post-analyzing capabilities. The OSPA cut-off distance is set to $5m$, with the p-order set to 2. The mean of the OSPA-values for each timestep are used as the OSPA-value for the whole dataset.

6.6.4 Performance - Metrics

Label	P_D [%]	P_X [%]	β_C	β_N	N_{scan}	K_{best}	r_{prune}	RMSE[e-3]	OSPA[e-3]	N_t (/292)	K_{worst}	r_{worst}
Avg values	90.0	5.46	0.089	0.010	7	750	1e15	164	348	330	450	5174
Avg values (low N)	90.0	5.46	0.089	0.010	5	750	1e15	164	346	330	127	5174
Avg values (high K)	90.0	5.46	0.089	0.010	7	1250	1e15	164	348	328	271	5174
1.5x Clutter (high K)	90.0	5.46	0.135	0.010	7	1250	1e15	174	370	309	1009	942
2.0x Clutter	90.0	5.46	0.178	0.010	7	750	1e15	174	342	278	627	1e6
2.0x Clutter (low N)	90.0	5.46	0.178	0.010	5	750	1e15	180	351	279	738	1e6
High P_O	70.0	2.50	0.089	0.010	7	750	1e15	182	382	319	650	942
High P_O (low N)	70.0	2.50	0.089	0.010	7	750	1e15	188	391	329	693	1331
Mix	70.0	5.00	0.135	0.010	5	750	1e10	177	376	292	811	1331

Table 6.7: Performance of the MHT on ground truth for different parameters. $N = N_{scan}$ and $K = K_{best}$ for the labels.

Generally, with $N_{scan} = 7$ and $K_{best} = 750$ one see that the K_{worst} is close to K_{best}, which suggests the correct hypothesis may have been pruned/not been generated. Therefore, both N_{scan} was lowered and K_{best} increased. Note that sometimes the OSPA score is lower for lower N_{scan} and K_{best}, with the same tracking parameters.

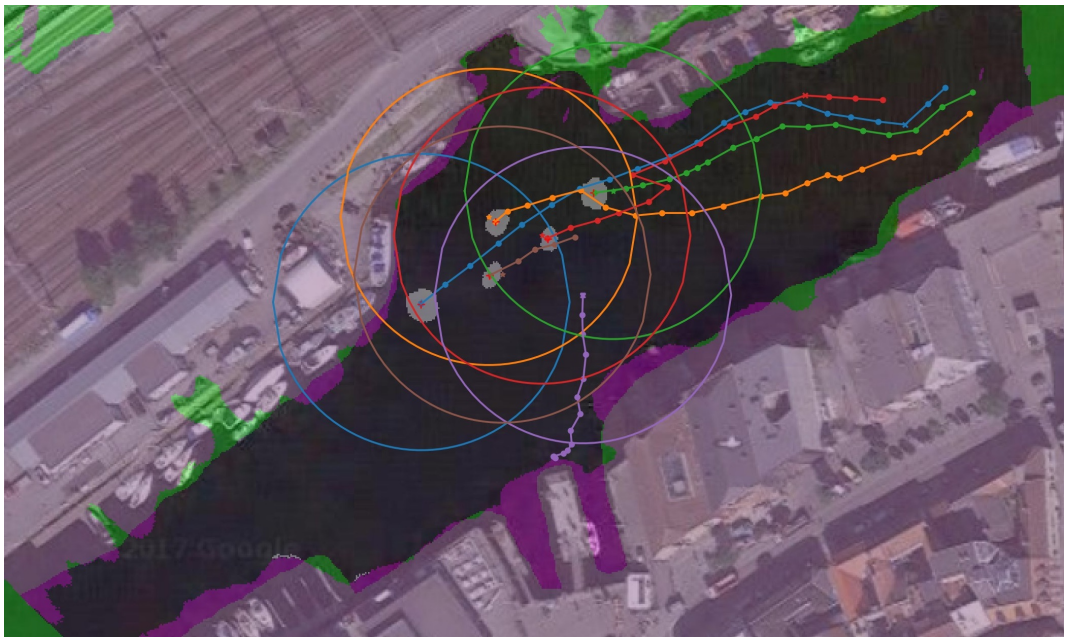
A problem with the Murty Implementation caused the program to freeze, and is the reason only a few trials were conducted. The error was reproduced for the same set of parameters and scans, and after some debugging seen to occur when solving one of the linear assignment problems in the Murty algorithm. It is believed to be a bug of one of the loops or the "goto"-statements used. The cost matrices that caused the problem were not found to have a pattern or be different to "working" ones, and it is therefore assumed they are not the problem. There was not time before the deadline of the thesis to find the bug and correct it, though it was attempted.

6.6.5 Performance - Example

A particular example is highlighted for the MHT, to show the effects of track termination. The parameters used are those of "Avg values (low N)". Eight targets are in and out of the observable area in a total of 4 minutes and 25 seconds (53 scans). At a particular instance, seven of them are present. The MHT correctly estimated the number of targets that were present, though two of the tracks suffer from track loss. However, this is not due to the MHT, but merged measurements of the detection for several scans. Low amounts of clutter are present during the time, except for some split measurements of the last two targets.

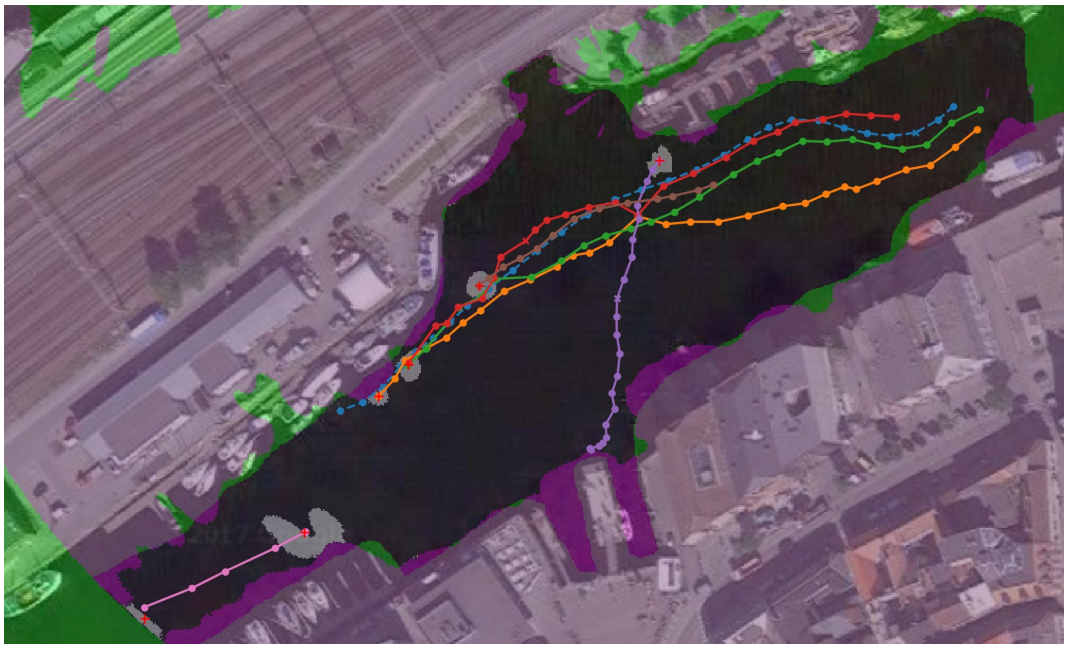
Four (actually five) kayaks have entered the canal. Two of them are merged (largest detection), and has been since entering.

Another target crosses. Now, all kayaks have been detected, and a track initiated for the last one.

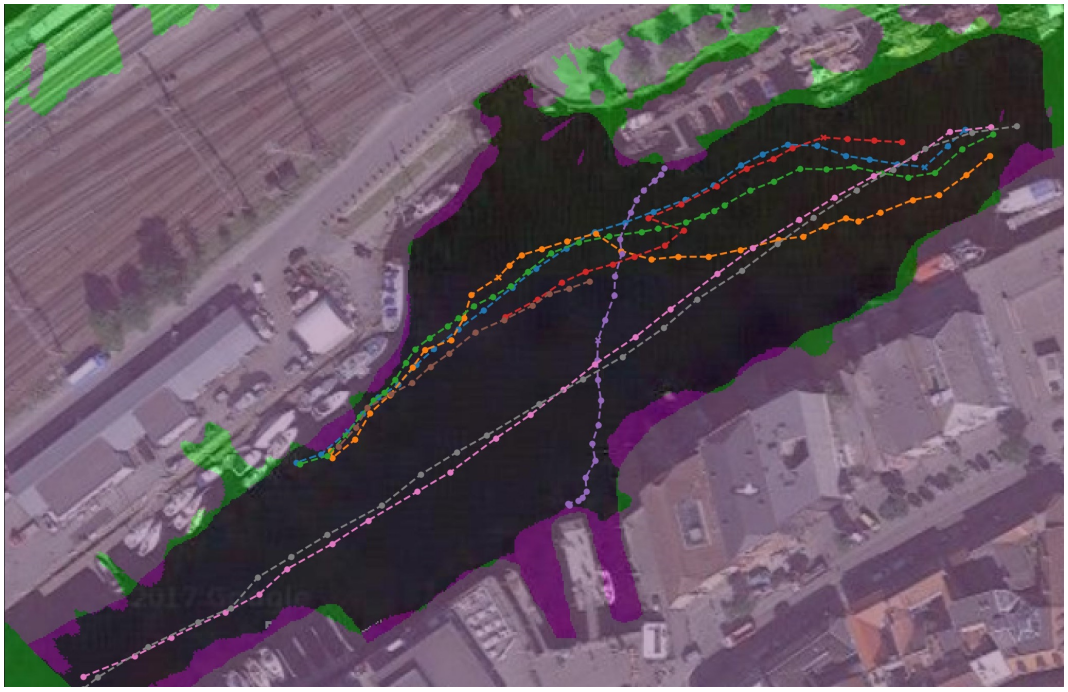


Same image as the previous one, but with track gates (for the previous timestep). The purple target has been misdetected.

The red kayak has been terminated (prematurely), due to two of the kayaks being merged again. The blue and brown has moved out of the observable area and has been terminated. Two new targets enter the canal at the left.



The ground truth for the same instance as the previous figure.



The resulting eight tracks after all targets have left the observable area.

Figure 6.22: Example of the MHT for a large amount of targets in the canal.

7 | Discussion

7.1 Filtering

The filtering with the multiple and the area filter gave a big improvement on the radar sensor system. The number of clutter measurements for the MHT was reduced with 41.1%, while only reducing the number of resolved measurements with 0.84%. Importantly, the reduction in resolved measurements did not lower the probability of detection, neither the average or the worst case.

The average track length was reduced, which suggests that the majority of the resolved measurements that were removed by the filtering was at the beginning or end of the tracks. This is reasonable, since when targets enter the observable area, they may be partly inside the area and partly outside, leading to small measurements, which then are removed by the area filter.

Since the worst case track length is not affected this will not be a problem for the MHT, as most targets are still observed for a long time. Also, if the time step is lowered from $5s$ to $1.25s$ in future experiments, this will increase the track length. The worst case length will increase from 5 to about 40 scans.

7.1.1 Multiple Filter

$A_{multiple} \geq 800$ removed close to no detections. This confirms the assumption that targets above a certain size completely obscures the radar view and no true detections are made behind them. $r_{multiple} = 1.0$ gave the best value for specificity, thereby removing more clutter than the higher values. This is interesting since multiples in general occurs for $r > 2$. However, as discussed in Section 4.2.5, the centroid of the true target which is used as a reference point may be off from where the reflection actually hits. Therefore one would expect $1.5 < r_{multiple} < 2.0$ to be somewhat reasonable for detecting multiples even with a wrong point of reference. Still $r_{multiple} = 1.0$ is low, and it signifies that other phenomenons also take place for large boats. A possible cause may be that some power is reflected off the sea surface and the target (or vice versa), which would give a slightly longer time for the power to be received and thus give a false detection right behind the true one.

7.2 Noise Estimation and Maximum Initial Speed

Split measurements were expected to represent the worst case measurement errors as discussed in Section 4.4.1. Figure 6.12 confirms the assumption, where some of the samples have errors of about $10\,m$. However, one can see that for most of the samples the error is low. The distribution of the samples resembles a normal distribution with mean equal to zero, though slightly underrepresented for values between 0 and 0.5. However, one could assume most resolved measurements are in this range.

As discussed in Section 4.4.2 most targets were expected to have low acceleration. Figure 6.17 also showed that this was the case. The samples resembles a normal distribution, but are too heavily represented around the mean of zero. However, as it was decided to be desirable to find a value that not necessarily represented the distribution of plant noise the best, but led to the maximum accelerating target just being gated, the value of $\sigma_v = 0.6$ was chosen and shown to just gate the target with $P_G = 0.99$. To account for the non-gaussian acceleration distribution, multi-model approaches may be utilized as for 3.1.4.

The target state initiation scheme proved to be a good one. By setting v_{max} equal to the highest initial velocity of the ground truth samples, the track gate of the target with the highest initial velocity just gated the next measurement.

7.3 Tracking parameters

Table 6.7 shows that different tracking parameters gives different performance for the MHT, both in terms of just cardinality and OSPA.

However, some of the results were not as expected. The expectation was that lowering P_D and P_X would improve performance of the tracking as described in Section 4.4.5, since one would improve tracking results for targets of low detectability. Figure 6.10 shows that there actually are few targets with low detectability, and it may be the case that lowering P_D will improve their performance, but worsen it for the ones with high detectability, giving an overall worse OSPA score.

Using the average values gave among the best OSPA-scores. However, it can be seen that the number of tracks estimated is too high, suggesting many false tracks.

Generally, the difference between $N_{scan} = 5$ and $N_{scan} = 7$ is small. In some cases a lower N_{scan} value gives better results for the same parameters ("Avg Values", "2x Clutter"). One reason for this is that the lower N_scan prunes away the "correct" hypothesis, but that this "correct" hypothesis is not the best in terms of OSPA. An other reason is that the correct hypothesis is not generated with the higher N_scan for the reasons discussed in Section 4.4.6.

This is underlined by the results that show that N_{scan} in general gives higher K_{worse}.

As expected, increasing the clutter density reduces the number of tracks. For clutter density higher than the average value this leads to less estimated tracks than there actually is. For $1.5x$ the clutter density, the estimated number of tracks is close to the correct number, but the high OSPA-score and the high K_{worst} suggests that the best hypothesis for some timesteps may not have been generated.

More experiments should be conducted to better determine the effects of the parameters. Also, the cases where the tracking is poor and the OSPA-score is bad should be identified and examined with the GUI to get a better understanding of what goes wrong.

7.4 Tracking Performance

In Section 6.6.5 it was shown that the system could handle a high number of targets in the area. The targets both appeared late due to merged measurements and disappeared closely to where other targets was appearing or misdetected. The system still estimated the correct number of targets and had optimal performance given the detection output.

The number of estimated targets for all trials in Table 6.7 are close to the true number, varying between being to low and too high, implying that the MTH is close to getting it right, and are likely to do so with additional tuning.

All tracks were examined for "Avg Values (low N-scan)" by using the GUI. In general, the algorithm performed well. A case where it often failed was for targets being misdetected for consecutive time steps, causing the MHT to terminate the track and then initiate it again. Also, tracks were too easily initiated, especially when consecutive split measurements were generated by targets. This led to a parallel false target being created in addition to the real one. It is suggested to increase either the clutter intensity, decrease the new target intensity and/or lower the probability of deletion.

8 | Closing Remarks

8.1 Conclusion

A complete surveillance system has been created. It includes a detection system and a new K-Best HO-MHT with Track Termination. In addition, two GUIs has been created to aid in finding tracking parameters and to examine tracking results.

The Track Terminating HO-MHT builds on the results of Reid [21]. It models that targets not only can be detected and misdetected, but also cease to exist with constant probabilities. The result is similar to that of Kurien, though the derivation is slightly different [**ref:kurien**]. However, it builds on top of Kurien by providing a polynomial time solution to generating the K-Best hypotheses at each timestep by using the algorithm of Murty.

The MHT has been implemented in Python, with track and hypotheses tree structures to improve on both computation time and memory consumption. A GUI has been made to examine the results and for tuning parameters. One also has the possibility to save and load computed data.

Ground truth tracks have been created for 40 hours of real radar data by using a GUI developed. It has been used to aid in determining parameters for both the detection system and the tracking system.

The detection system from the previous project of the author has been more thoroughly examined and improved upon, removing 41.1% of clutter, and only 0.84% of the true target detections. The improvements include two filters, one for removing detections based on the area of the detection contour and one to remove "multiples" generated by large boats. The optimal parameters of the filters have been found by examining ROC-curves. In addition, it has been asserted that the area filter does not decrease the worst case probability of detection, which was shown to be for a small target.

Measurement and plant noise parameters for a DWNA-model have been estimated using ground truth data. Also, the maximum initial speed of targets has been found for use in target state initialization. Both the plant noise and the initial speed parameter has been shown to give minimally sized track gates for their respective worst case scenarios, that being the maximum acceleration

maneuver and the maximum initial speed.

As a final remark, a solid foundation has been laid to surveille the area around the autonomous ferry. That is both in terms of the tracking system and the tools to analyze it. With a polynomial time track termination MHT, realtime possibilities of the system could also be possible.

8.2 Further Work

The main problem of the current project is an error in the Murty implementation. This, and limited time, led to a thorough examination of the intensity and probability parameters of the MHT lacking. Also, a more thorough verification of the MHT and its capabilities are needed. It will be interesting to see its performance for a timestep of $1.25s$

In addition, some further work is listed which can prove itself useful:

- Use the MHT and the data to obtain additional statistics including typical entry/exit points, COLAV maneuvers and similar.
- Determining good values for the N-scan pruning and the K-best generation, both in terms of performance and time consumption.
- Evaluate other methods than the MHT and compare their performance. Especially FISST based methods are interesting.
- It is believed by the author that the MHT can be reformulated in terms of an inhomogenous Poisson point process and variable track probabilities, in which thorough statistics of the area and targets can be used to improve on the tracking.

Bibliography

[1] Y. Bar-Shalom and X.R. Li. *Multitarget-multisensor Tracking: Principles and Techniques*. Yaakov Bar-Shalom, 1995. ISBN: 9780964831209.

[2] Samuel S Blackman. "Multiple hypothesis tracking for multiple target tracking". In: *IEEE Aerospace and Electronic Systems Magazine* 19.1 (2004), pp. 5–18.

[3] Henk AP Blom and Edwin A Bloem. "Probabilistic data association avoiding track coalescence". In: *IEEE Transactions on Automatic Control* 45.2 (2000), pp. 247–259.

[4] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*. CourseSmart Series. Wiley, 2012. ISBN: 9780470609699.

[5] I. J. Cox and S. L. Hingorani. "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.2 (Feb. 1996), pp. 138–150. ISSN: 0162-8828. DOI: 10.1109/34.481539.

[6] I. J. Cox et al. "A comparison of two algorithms for determining ranked assignments with application to multitarget tracking and motion correspondence". In: *IEEE Transactions on Aerospace and Electronic Systems* 33.1 (Jan. 1997), pp. 295–301. ISSN: 0018-9251. DOI: 10.1109/7.570789.

[7] David F. Crousea et al. *The JPDAF in practical systems: computation and snakeoil*. 2010. DOI: 10.1117/12.848895. URL: https://doi.org/10.1117/12.848895.

[8] M. de Feo et al. "IMMJPDA versus MHT and Kalman filter with NN correlation: performance comparison". In: *IEE Proceedings - Radar, Sonar and Navigation* 144.2 (Apr. 1997), pp. 49–56. ISSN: 1350-2395. DOI: 10.1049/ip-rsn:19970976.

[9] G. Gavriloaia, A. Sperila, and A. Stoica. "An Ad-Hoc Method for Avoiding Tracks Coalescence in Pdaf for Tracks Fusion". In: *TELSIKS 2005 - 2005 uth International Conference on Telecommunication in Modern-Satellite, Cable and Broadcasting Services*. Sept. 2005. DOI: 10.1109/TELSKS.2005.1572180.

[10] W. Koch and G. Van Keuk. "Multiple hypothesis track maintenance with possibly unresolved measurements". In: *IEEE Transactions on Aerospace and Electronic Systems* 33.3 (July 1997), pp. 883–892. ISSN: 0018-9251. DOI: 10.1109/7.599263.

[11] Thomas Kurien. "Issues in the design of practical multitarget tracking algortihms". In: *Multitarget-multisensor tracking: advanced applications*. Ed. by Yaakov Bar-Shalom. 1990. Chap. 3, pp. 43–83.

[12] L. Lin, Y. Bar-Shalom, and T. Kirubarajan. "Track labeling and PHD filter for multitarget tracking". In: *IEEE Transactions on Aerospace and Electronic Systems* 42.3 (July 2006), pp. 778–795. ISSN: 0018-9251. DOI: 10.1109/TAES.2006.248213.

[13] Ronald Maher. *A survey of PHD filter and CPHD filter implementations*. 2007. DOI: 10.1117/12.721125. URL: https://doi.org/10.1117/12.721125.

[14] R. Mahler. "PHD filters of higher order in target number". In: *IEEE Transactions on Aerospace and Electronic Systems* 43.4 (Oct. 2007), pp. 1523–1543. ISSN: 0018-9251. DOI: 10.1109/TAES.2007.4441756.

[15] R. P. S. Mahler. "Multitarget Bayes filtering via first-order multitarget moments". In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (Oct. 2003), pp. 1152–1178. ISSN: 0018-9251. DOI: 10.1109/TAES.2003.1261119.

[16] Ronald PS Mahler. *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[17] Katta G. Murty. "An Algorithm for Ranking all the Assignments in Order of Increasing Cost". In: *Operations Research* 16.3 (1968), pp. 682–687. ISSN: 0030364X, 15265463.

[18] Kusha Panta et al. "Probability hypothesis density filter versus multiple hypothesis tracking". In: *Signal processing, sensor fusion, and target recognition XIII*. Vol. 5429. International Society for Optics and Photonics. 2004, pp. 284–296.

[19] S. A. Pasha et al. "A Gaussian Mixture PHD Filter for Jump Markov System Models". In: *IEEE Transactions on Aerospace and Electronic Systems* 45.3 (July 2009), pp. 919–936. ISSN: 0018-9251. DOI: 10.1109/TAES.2009.5259174.

[20] R. L. Popp, K. R. Pattipati, and Y. Bar-Shalom. "m-best S-D assignment algorithm with application to multitarget tracking". In: *IEEE Transactions on Aerospace and Electronic Systems* 37.1 (Jan. 2001), pp. 22–39. ISSN: 0018-9251. DOI: 10.1109/7.913665.

[21] Donald Reid. "An algorithm for tracking multiple targets". In: *IEEE transactions on Automatic Control* 24.6 (1979), pp. 843–854.

[22] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. "A consistent metric for performance evaluation of multi-object filters". In: *IEEE Transactions on Signal Processing* 56.8 (2008), pp. 3447–3457.

[23] Navico - Simrad. *Broadband 4G^{TM} Radar Installation Guide*. URL: http://www.navico-commercial.com/Root/SimradProSeries_docs/RADAR_3G_4G_IG_988-10113-003_EN_W.pdf.

[24] R Singer, R Sea, and K Housewright. "Derivation and evaluation of improved tracking filter for use in dense multitarget environments". In: *IEEE Transactions on Information Theory* 20.4 (1974), pp. 423–432.

[25] B. N. Vo, S. Singh, and A. Doucet. "Sequential Monte Carlo methods for multitarget filtering with random finite sets". In: *IEEE Transactions on Aerospace and Electronic Systems* 41.4 (Oct. 2005), pp. 1224–1245. ISSN: 0018-9251. DOI: 10.1109/TAES.2005.1561884.