Incorporation of Features in Multistatic Active Sonar Tracking

Evan Hanusa

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2012

Reading Committee:

Maya R. Gupta, Chair

David W. Krout

John D. Sahr

Program Authorized to Offer Degree:
UW Electrical Engineering

University of Washington

Abstract

Incorporation of Features in Multistatic Active Sonar Tracking

Evan Hanusa

Chair of the Supervisory Committee:
Professor Maya R. Gupta
Department of Electrical Engineering

This document contains the majority of the research on multistatic feature-aided tracking that I have done in my graduate student career at the University of Washington. It contains an overview of sonar and the measurements that an active sonar system generates. It also gives an overview of a tracker based on joint probabilistic data association (JPDA) which is the basis for the research on integrating features into tracking. Several methods for integrating features are compared: integration into JPDA itself at two places, integration into track management, and simply rejecting contacts that appear to be clutter. The methods were tested on the TNO benchmark dataset, showing that integrating the features into track management performed the best, resulting in increased accuracy, fewer "spurs" coming off of target tracks, and decreased track fragmentation.

In addition, the use of tracking information to improve classification was explored. By using a tracker to predict the aspect of the target at the current time, contacts can be classified based on their aspect-dependent features, target strength and Doppler. The results of this were interesting for two reasons: a high average accuracy can be obtained by using the aspect estimate along with the uncertainty from the prediction, and that only using the prediction (no uncertainty) always performed worse than using no information at all.

This document also describes the development of two preprocessing techniques (posterior distribution preprocessing and likelihood-based clustering) that allow the combination of measurements that come from different sources, which can then be tracked by a standard

JPDA-based tracker. This is especially key for multistatic sonar, as the preprocessing techniques allow a tracker to track very dim targets ($P_d \approx 0.1$) in high clutter scenarios (44 clutter contacts per receiver).

The posterior distribution preprocessing technique is extremely flexible and can fuse extremely different types of measurements (IR and HD video data, imaging sonar and HD video data, multistatic sonar). It allows for the appropriate modeling of the measurement noise, resulting in a system that can be applied to many types of data. In addition, this work describes how the preprocessing step can be modified to incorporate any additional feature data.

The likelihood-based clustering technique works well on multistatic sonar data, and allows for the incorporation of any features when calculating the similarity between contacts. This is especially useful for aspect-dependent features, such as Doppler or amplitude. The clustering step is followed by a fusion step that allows for the estimation of target heading or velocity if the appropriate features are used. Using the preprocessing step results in a tracking system that has improved performance, especially on dim targets in a large amount of clutter.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ACTIVE SONAR: Active sonar requires a transmitter and receiver. The transmitter emits a waveform, and then the receiver detects the same waveform when it is re-emitted by objects in the detection region. Allows for measuring range, bearing, Doppler (if an appropriate waveform is used) and other features.

BISTATIC SONAR: A sonar geometry where the transmitter and receiver are at different locations.

BLANKING REGION: An elliptical region inside of which targets are undetectable due to the direct blast of the transmitted waveform.

CONTINUOUS WAVE PING: A transmitted waveform that maintains a single frequency. This waveform allows for measuring the Doppler shift, but has worse range resolution. Abbreviated as CW.

CYLINDRICAL SPREADING: Transmission losses due to cylindrical spreading occur in an environment where there is a top and bottom boundary layer. Losses from a to b are approximated by $10 \log_{10} d_{ab}$.

FREQUENCY MODULATED PING: A transmitted waveform that has a changing frequency (often a linear chirp). This type of waveform has improved range resolution, but a Doppler shift cannot be measured. Abbreviated as FM.

KINEMATIC MEASUREMENT: A measurement which is due to kinematic features of the target (position, velocity, etc).

MONOSTATIC SONAR: A sonar geometry where the transmitter and receiver are colocated.

MULTISTATIC SONAR: A field of transmitters and receivers that have overlapping detection regions. A single transmitted ping can be observed by multiple receivers.

NON KINEMATIC MEASUREMENT: A feature of the target which is not (necessarily) a function of the target's state.

PASSIVE SONAR: No transmitter is used, only a receiver. Using an array can allow for estimating bearing, but ranging is not possible with a single receiver.

PROBABILITY OF DETECTION: The probability that a target will be detected. A function of geometry, distance, and the aspect-dependent target strength. Abbreviated as P_d.

ACKNOWLEDGMENTS

I would like to acknowledge the people that made this research possible. Laura, for listening when I complained about my research and suggesting new ideas. All of my professors at Rose-Hulman who somehow convinced me that I liked engineering so much that I should do 6 more years of school. All the students and faculty of the SSLI lab, especially Alex Stupakov and Amy Lastuka, for getting me started in machine learning and graduate school...and keeping me sane when it wasn't going too well. Maya Gupta and the students in her lab, especially Sergey Feldman, Eric Garcia, James Chen and Hyrum Anderson. David Krout, William Mortensen, Jack McLaughlin and everyone else at the APL who has helped with my research. Lastly, ONR for funding the vast majority of my research time the past 5 years.

DEDICATION

To my friends and family.

Chapter 1

MULTISTATIC FEATURE-AIDED TRACKING

The study of underwater sound has been ongoing for centuries. As early as 1490, Leonardo da Vinci noted that a long tube, placed in water would allow a person to observe ships at a great distance [3]. This was clearly a very crude detection scheme and did not allow for information about where the ship making the sound was (either in distance or direction). Interestingly enough, this idea has a simple extension; a human listener can estimate the angle of arrival placing two separated tubes in the water. This approach was used until World War I. Several of these tubes could be used, forming a large passive multireceiver sonar system that was able to perform rudimentary bearing and range estimation [4]. Sonar development moved quickly in the inter-war period and by the start of World War II, many US ships had equipment on board for echo ranging (active sonar) and underwater listening (passive sonar).

While development and research in sonar was clearly driven by military purposes, the technology has become very popular in many civilian and business uses. Down and side directed sonar is used to map bodies of water and create maps allowing for safer sea travel. Commercial and recreational fishers benefit from sonar-based fish finders and identifiers. As fossil fuels are becoming more scarce, many companies are using extremely intense sound sources to detect and measure the amount of oil and natural gas that are present below the ocean.

1.1 Sonar

The term sonar originated as a portmanteau of the word "sound" and the acronym "RADAR". Later, the phrase *Sound Navigation and Ranging* was fit to the word as the acoustic analog for radar [5]. Similarity between the two areas doesn't end there, and many of the techniques in this thesis are both inspired by and applicable to radar.

Sonar can be split into two main divisions: *active* and *passive* sonar. In passive sonar, hydrophones (or arrays of hydrophones) are placed in the water and simply "listen" for targets to emit sound. Assuming a hydrophone array is used, beamforming can be done to detect the angle of arrival and equivalently, the bearing towards the target. However, no range information is available, so the object's actual location is only constrained to a "slice" in Cartesian coordinates, as illustrated in Figure 1.1. If several arrays are in the water at different locations and are all able to detect the object, the bearing measurements can be combined to greatly improve the estimate of position, as described in Figure 1.2.

Passive sonar has an advantage of being simple, however it has a few obvious drawbacks.

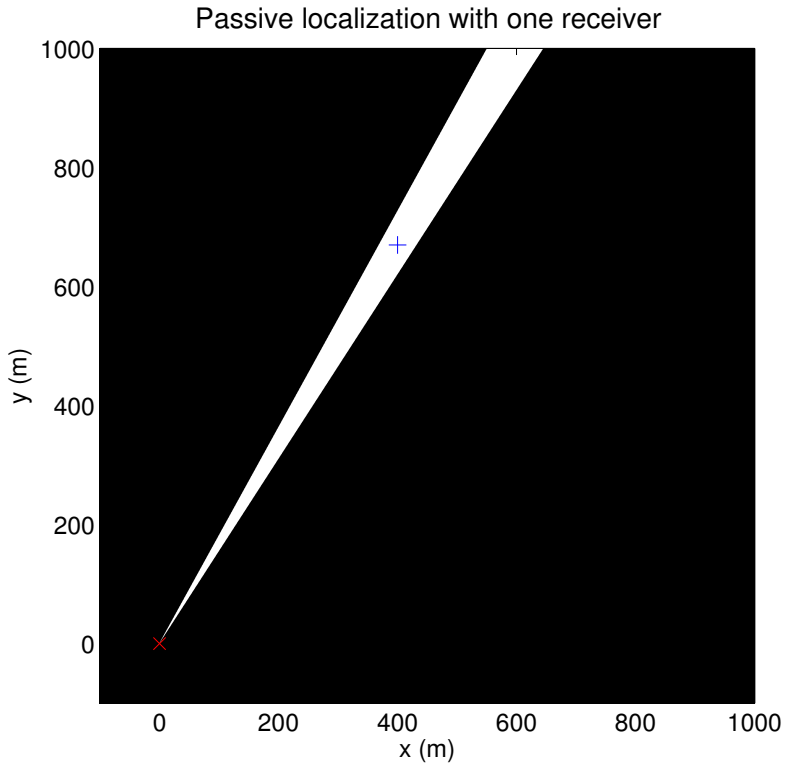

Figure 1.1: Object localization with one passive receiver. The receiver is indicated by a red 'x', the object by a blue '+'. The receiver determines the angle of arrival using a beamformer, restricting the object's possible location to a slice (shaded white) that is the width of the beamformer's bearing bin. In this case, a bin size of 4° is assumed. Note that the width of the slice gets larger as the distance from the object to the receiver.

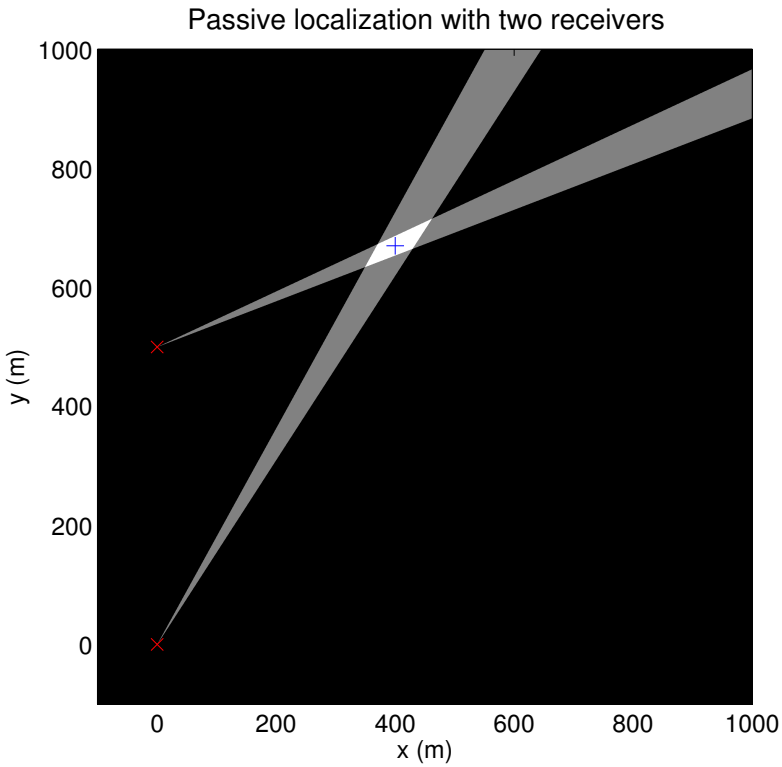

Figure 1.2: When a second receiver is added, the object must lie in the white area that is the overlapping region of the two receiver's slices. Additional receivers could be added to improve localization.

Passive sonar requires multiple receivers to determine range information. In addition, if the object you are interested in detecting is not making noise, you cannot find it using passive sonar. This brings us to active sonar, the second main type of sonar. In active sonar, there is both a transmitter and receiver. The transmitter emits a "ping" of sound into the water, which travels as a pressure wave in all directions. When this pressure wave encounters an object, it is re-emitted by that object. This reflection travels again in all directions and can be sensed by the receiver's hydrophone. By measuring the amount of time between transmission and reception of the ping, the range to the target can be estimated. The receiver will consist of an array of hydrophones, allowing for the detection of angle of arrival (and bearing to the target). By combining the estimates of bearing and range, the target

can be localized.

An active sonar system where the transmitter and receiver are located at the same location is referred to as a *monostatic* system. Figure 1.3 illustrates a monostatic sonar geometry. Often times, it is advantageous to separate the transmitter and receiver, which is referred to as a *bistatic* system, illustrated in Figure 1.4. In a bistatic geometry, the same principles are at work, but there is an additional elliptical *blanking region* in which objects cannot be detected. This blanking region is due to the transmitted waveform having a non-zero duration. During waveform transmission, nothing other than the *direct blast* of the transmitted sound is able to be detected, so receivers are turned off in order to protect the sensitive electronics.

Recall that when the transmitted pressure wave arrives at an object of interest, it is re-emitted in *all* directions. This is a main motivation for *multistatic* sonar. Several receivers are placed at different locations in the area of interest, and each can possibly detect the re-emitted pressure wave. In this thesis, any sonar geometry that has multiple receivers (and possibly transmitters) is referred to as a multistatic field. In the multistatic field, each of the transmitter-receiver pairs can be viewed as a bistatic pair and therefore have measurements of bearing and bistatic range. As mentioned before, active sonar results in measurements of bearing and bistatic range. In addition, if the transmitted waveform is chosen appropriately, the Doppler shift of the waveform due to the target's motion can be measured. Chapter 3 discusses the measurements in detail.

1.2 Tracking

Tracking can be viewed as the problem of estimating $\hat{X}_{t=1:n}$, the past and current state vectors of the object given measurement vectors $z_{t=1:n}$. The state vector of an object typically consists of position, velocity, and potentially higher-order derivatives. This problem is complicated by the existence of noise on the measurements z_t, measurements that do not correspond to the object being tracked (clutter), the existence of multiple objects, and the possibility that the object is not observed at any moment in time. The measurement vector typically contains one or more of the state variables (or a known function thereof) at the current time as well as other features of the object. Feature-aided tracking focuses on the

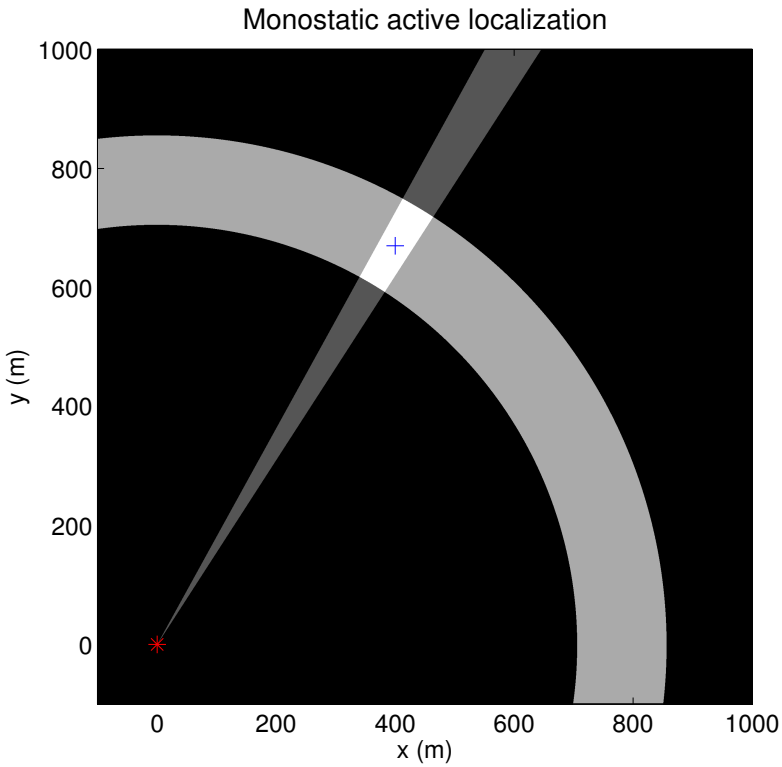

Figure 1.3: Object localization with monostatic active sonar. The transmitter/receiver is indicated by a red '*', the object by a blue '+'. The receiver does beamforming to determine the angle of arrival, which constrains the object's location to the dark gray slice. In addition, the time of the return determines the range, constraining the object's location to the lighter gray ring. By combining the two measurements, the object can be localized to the small white region.

use of these other features to improve tracking performance.

Depending on the problem domain and the type of additional features, they can potentially be used to overcome some of the complications mentioned above. An application of object tracking is in the observation of cells under a microscope, a problem characterized by many closely spaced objects. The non-kinematic features such as the shape or color of the cell can be used to differentiate between different types of cells in close proximity. In radar applications, comparing features of the object returns to known classes allows for the use of class-specific kinematic models. In the low frequency active sonar tracking domain, which

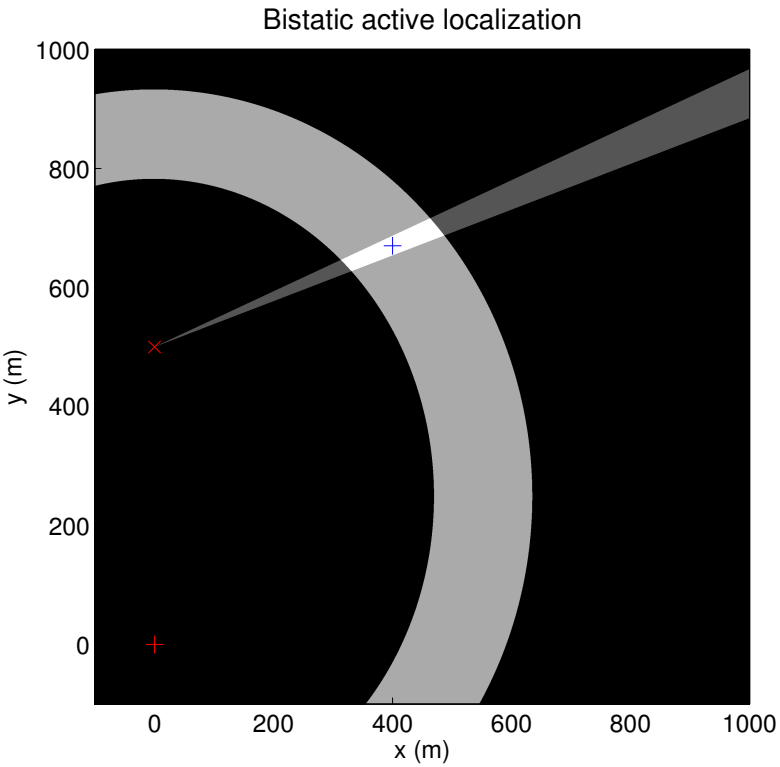

Figure 1.4: Object localization with bistatic active sonar. The transmitter is marked with a red 'x' and the receiver with a red '+'. In this geometry, the range measurement constrains the object's location to an ellipse-shaped ring, rather than a perfect circle.

is often characterized by high clutter and low probability of detection, additional features of the return time series could be used in a classifier to discard clutter or infer additional target characteristics.

There are many approaches to tracking in general and also to integrating features into tracking, and they will be discussed in Chapter 2.

1.3 Applications of this Work

The overall goal of this thesis is to show that integrating features into multistatic tracking is a useful thing to do. In the process, several specific contributions and applications have emerged:

1. *Integration of Classification into Joint Probabilistic Data Association (JPDA)*: Features derived from the return are used to classify contacts as targets or clutter, and the classification results are integrated at several steps of a JPDA-based tracker [6].

2. *Integration of Tracking Information in Target Classification*: Using the the track state to estimate target heading allows for improved classification of contacts, especially if uncertainty is appropriately handled [7].

3. *Posterior Distribution Preprocessing of Multistatic Sonar Data and Integrating features*: A framework which allows for the fusion of kinematic measurements and non-kinematic features for multistatic sonar data. This framework also allows fusion of very different types of measurements, for example High Definition (HD) video and imaging sonar data. [8, 9, 10, 11, 12].

4. *Clustering, Fusing, and Classification of Contacts using Likelihood-Based Similarities*: A extension of the posterior distribution preprocessing technique with a contact clustering step that uses a novel likelihood-based similarity to improve tracking results.[13, 14, 15].

The end result of the research is a contact preprocessing and tracking framework that performs well on multistatic data, allows for integration of features, and can handle data that comes from different sensing modalities.

Chapter 2

RELATED WORK

2.1 Feature-Aided Tracking

In addition to the bearing and range measurements that are obtained in active sonar, there exist other features that can be available to incorporate into a tracking framework. One of the earliest works to do so was in the use of target amplitude. The target amplitude discussed here is the signal to noise ratio (SNR) of the received power of a target and the environmental noise. Lerro and Bar-Shalom modified the PDA filter to include amplitude [16]. The assumption made in the modification, (referred to as PDAFAI), is that the amplitudes of targets will be distributed differently than the amplitudes of contacts that are due to clutter or background noise. The use of amplitude information in PDAFAI is shown to improve tracking performance, even as SNR decreases.

In general, the target amplitude is a function of the range of the target, the environment, and the target strength (TS). This target strength is aspect dependent, and an example target strength curve is shown in Figure 2.1 [3].

This aspect dependence is intuitive: most targets are not radially symmetric and therefore the strength of the return should be a function of the aspect. Recently, Pitton and Fox investigated the use of this aspect-dependent target strength curve to modify PDAFAI tracking [17]. The velocity from an existing track's state space is used to create a distribution over aspect. This distribution is combined with the target strength curve to improve the PDAFAI algorithm. Krout and Morrison applied the PDAFAIwTS algorithm to tracking datasets and showed that it can give improvements in more difficult tracking (lower SNR) scenarios [18].

Another use of the target strength curve is to improve track initiation by only starting a track when a "specular" hit occurs [19, 20, 21, 22, 23]. Specular refers to the case when the target is hit broadside by the transmitted pulse. This occurs at 90° and 270° in Figure 2.1.

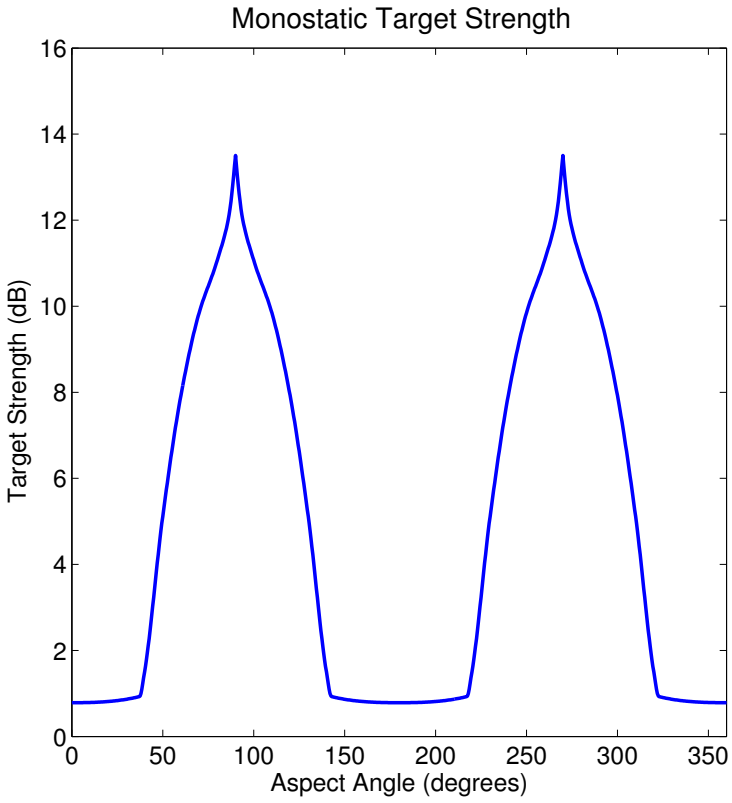

Figure 2.1: Example monostatic target strength from the PACsim dataset. The target strength is much higher at broadside (90° and 270°), which can be a useful way to differentiate a target from clutter.

The amplitude of the return is much higher than at non-specular angles, so it is simple to detect these returns and infer the heading of the target. Because the target's heading is now known, the track's initial state can be known with higher confidence than other initiation techniques. Additionally, contacts at previous time steps can be associated with the track by going backwards through the data. This method could be applied to any feature that uniquely characterizes an aspect.

In a recent extension of the specular cuing approach, tracks build up a history of the amplitudes associated with the contacts that have been assigned to them. The amplitudes are then compared to a template, allowing for the classification of tracks as target or clutter

[24].

Both PDAFAI and PDAFAIwTS start from the amplitude information which is the result of a signal processing chain that detects contacts in the time domain return signal. This information is available in most tracking datasets, and can be simulated if not. However, the underlying time domain signal may contain even more information. Consider the plot of a (simulated) time domain signal vs aspect in Figure 2.2.

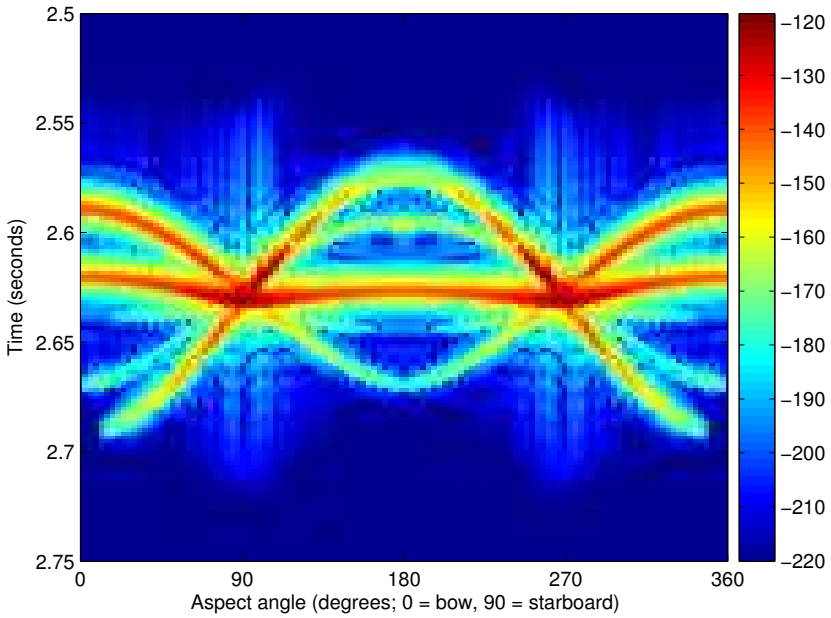

Figure 2.2: Clean return time series envelope (dB) vs. aspect angle for BASIS-3D model in monostatic geometry [1]. Each vertical slice is the time-domain impulse response of the target at an aspect. Blue indicates that little energy is received at a time, and red indicates a large amount of energy is received. This figure was generated by William Mortensen (APL-UW) using SST [2].

In this figure, time is increasing along the y-axis, aspect varies along the x-axis. Blue indicates little energy is received at that time, and red indicates a high amount. The location and spacing of these will be dependent on features of the target. If you take a single vertical slice, it is the impulse response of the target at that aspect angle. One can imagine several possible aspect-dependent features that can be derived from this signal.

The existence of these non-kinematic features is not unique to active sonar, and significant work has been done to use them in tracking in the radar domain. Agate and Sullivan used the similarity between return data and the returns of known classes, and used this information to modify the kinematic associations [25]. Bar-Shalom et al. use classification to modify the association cost function in an auction algorithm [26]. Hong et al. show that the use of features to identify targets can improve the performance of JPDA in a HRR radar system [27, 28]. Salmond and Parr augment the tracking state vector with an estimate of target length [29]. The target length is estimated from a measurement of target extent. Chang and Fung use frequency-domain features and other radar-specific features to improve an MHT system by augmenting the track's states space [30].

One specific feature, the Doppler shift of the signal has been used in a variety of works. Early use of Doppler involved the use of a mean-squared error fitting technique to track a CW-radiating object [31]. In more recent works, Doppler is used in a passive sonar setting to improve the localization and tracking of objects [32, 33, 34]. Wang et al. use Doppler measurements to improve the initial estimate of target velocity, showing both improvement in tracking performance and a decrease in necessary computation [35]. Other recent work has shown that moving receivers can use a signal processing approach to localize a stationary transmitter if the entire time-domain signal can be transmitter [36]. LaCour uses Doppler measurements in a Bayesian tracking framework, with the goal of tracking an object across multiple time indices [37].

One common feature that is being used in surface ship tracking is the Automatic Identification System (IAS) data stream. This data includes information about the type of ship being tracked. However, it is a relatively low rate system and can be subject to intermittent coverage (or intentional errors). It can be used in conjunction with radar tracking to improve tracking performance [38].

Features can also be used to detect maneuvers in targets. The detection of a maneuver allows the tracker to switch to a model that is more appropriate to the target's motion, resulting in improved performance on a maneuvering target [39].

2.2 Multistatic Tracking

Recently, there has been a large interest in multistatic tracking, due in part to the deployment of large multistatic sonar fields in the ocean. In addition, there is a large amount of interest in using passive radar with many transmitters to observe planes and/or ground traffic [40, 41].

One approach to multistatic sonar tracking is to use multiple stages of *maximum likelihood multiple hypothesis tracking* (ML-MHT) [42, 43]. In this approach, the first stage runs on all the contacts that are observed for a single ping. Contacts that are likely to have originate from an object, rather than clutter, are passed to a second level of ML-MHT. The first level has the effect of reducing the number of contacts presented to the MHT tracker. This allows the second level to use a larger depth and detect targets that may not have been trackable in a single-layer system.

Additional research has been done to apply a more standard MHT tracker to multistatic data [44, 45]. Due to the large amount of contact data that comes from multistatic fields, a standard MHT tracker requires more data preprocessing and/or clever approaches. In these MHT approaches, thresholding is done on features of the data based on the known clutter and target distributions. This allows the MHT tracker to detect targets that match the expected distributions, but can cause issues if different targets are used. For example, if a fast target is added, an upper threshold on the Doppler measurement could reject all contacts from this target.

Unfortunately, many existing trackers do not perform well on dim targets in multistatic geometries when there are a large number of transmitters or receivers. There has been some research (including some presented later in this work) on preprocessing multistatic data to make it more amenable to tracking using existing trackers.

2.2.1 Dataset Preprocessing

As current systems are being pushed to their limits, there has been research into preprocessing the multistatic data down to more manageable data. This reduces the number of contacts presented to the tracker, allowing for more complex tracking algorithms, however

it clearly has the potential to reject valuable contacts.

Recently, there has been a focus on using more sophisticated approaches to initially reject some contacts, then combine the remaining contacts into fused measurements that can be used in tracking. Several of the methods in this thesis take this approach (Chapters 9, 10).

Georgescu et al. have also been working on several iterations of a predetection fusion framework [46, 47, 48]. In this framework, the (optional) first step is to reject contacts based on expected target and clutter statistics. Specifically, looking at the ratio of Doppler and amplitude measurements gives an intuitive and well-performing classifier. A target that has a high Doppler shift is moving towards (or away) from the receiver. In this case, the amplitude will be low (due to the aspect-dependent nature of the target strength). If the Doppler shift is low, then the expected amplitude is high. Next, particles generated using contact error statistics are used to determine which contacts should be combined using a "sifting" process that quantizes particles to a grid. If a certain grid space has enough particles, then measurements which overlap in that grid are fused.

Later, they used a random finite set Markov chain Monte Carlo approach to improve performance as well as get an estimate of measurement covariance [49]. This approach is more computationally complex, but has improved performance.

Chapter 3

SONAR MEASUREMENTS

3.1 Kinematic Measurements

The first type of measurements that are available in active sonar are "kinematic" measurements. Kinematic measurements are defined here as measurements that are a function only of the kinematic state of the target (position and/or velocity). These measurements are calculated at the receiver from the received time series.

Receivers consist of an array of hydrophones that allow for the determination of the angle of arrival of sound in the water. This angle of arrival determines the *bearing*, b to the object that reflected or generated the sound. Here, bearing is assumed to be measured clockwise from receiver's north. Due to finite receiver array size, processing limitations, and other error sources, measurements are corrupted with noise. Here, the noise is assumed to be zero mean additive Gaussian, however the methods presented in this work are distribution-agnostic. With a single measurement of bearing and a noise distribution, the contact's location is constrained to a pie-shaped slice in Cartesian coordinates. For convenience in later chapters, let $B(x, y, r)$ be a function that maps a state $X = [x, y]$ to a measurement of bearing,

$$B(x, y, r) = \tan^{-1} \frac{x - r_x}{y - r_y}, \tag{3.1}$$

where receiver r is located at $[r_x, r_y]$.

In addition to a bearing measurement, the receiver also measures the time difference between the transmitted ping's direct blast and the received echo. This time difference can be used in conjunction with the speed of sound in water to determine the bistatic range, d, to the object that generated the echo. The accuracy of this measurement is largely a function of the transmitted waveform. In general, FM waveforms result in lower error and CW waveforms have larger errors (but allow for sensing of Doppler shift). A single measurement of bistatic range with noise constrains an object's position to an ellipse with

foci at the transmitter and receiver locations. Next, let $D(x, y, r, t)$ be a function that maps a state $X = [x, y]$ to a measurement of bistatic range,

$$D(x, y, r, t) = d_{tc} + d_{rc}, \tag{3.2}$$

where d_{tc} and d_{rc} are the distance from the transmitter and receiver to the contact location $c = [x, y]$.

By combining the measurements of bearing and bistatic range, a contact can be localized to a single position in 2-D Cartesian space. Figures 3.1 and 3.2 show the bearing and range measurements for monostatic and bistatic geometries, respectively. However, the error is in the measurement domain, and is receiver dependent. Mapping the error into the Cartesian domain results in a banana-shaped measurement error surface.

If the transmitted waveform is Doppler sensitive (CW is the simplest example), the Doppler shift of the waveform due to target motion can be measured. The Doppler shift is a change in frequency of the received signal - if the target is decreasing in bistatic range, the Doppler shift will be positive. With the measurement of bistatic range, bearing, and Doppler, the Doppler shift can be used to estimate the speed, $|v|$ and heading, θ_v of the object of interest. The bistatic Doppler shift can be calculated as the projection of the target's velocity onto the line that bisects the bistatic angle,

$$\Delta(x, y, |v|, \theta_v, r, t) = |v| \cos \theta_e, \tag{3.3}$$

where

$$\cos \theta_e = \cos \theta_{TB} \cos \frac{\beta}{2}. \tag{3.4}$$

Figure 3.3 describes these angles graphically.

All of these angles and kinematic features are measured from the receiver - in a multistatic system, there will be multiple receivers, which I will index by i. This results in vastly different measurements for the same target state for different receivers.

3.1.1 Noisy Measurements and Posterior Distributions

The measurements introduced in the previous sections are a result of processing the time-domain return data, including beamforming, clustering, and a variety of other steps. These

measurements always have some error on them, and modeling that error appropriately is important. For later chapters, it is helpful to be able to visualize the posterior distribution over position $\mathbf{x}$ given a measurement z, $P(\mathbf{x}|z)$. In this notation, $\mathbf{x}$ is the realization of the random state vector, $X = [x, y]$. For ease of explanation and visualization, zero mean additive Gaussian noise will be assumed, but the concept applies to any additive noise. For a single measurement of bearing b, the posterior distribution is $P(\mathbf{x}|z = b) = \gamma_1^{-1}\mathcal{N}(B(x, y, r); b, \sigma_b^2)$. In the cartesian space this looks like Figure 3.4.

If only a measurement of bistatic range, d, is available, the posterior distribution is $P(\mathbf{x}|z = d) = \gamma_2^{-1}\mathcal{N}(D(x, y, r, t); d, \sigma_d^2)$. In Cartesian space, this looks like Figure 3.5.

In an active system, both measurements are available, and the posterior distribution

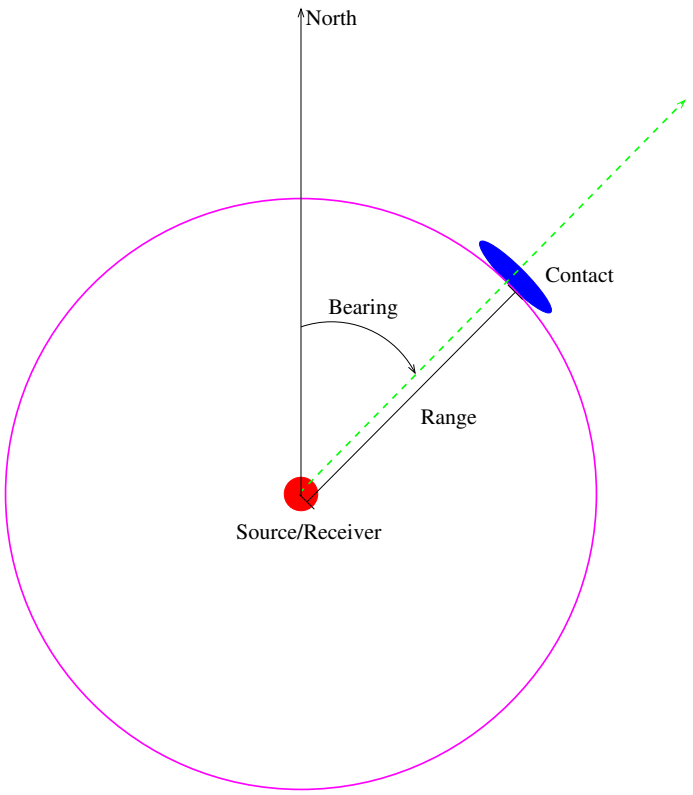

Figure 3.1: Contact localization using monostatic active sonar. Bearing is measured clockwise from receiver north and combined with the range to localize the target to a single point.

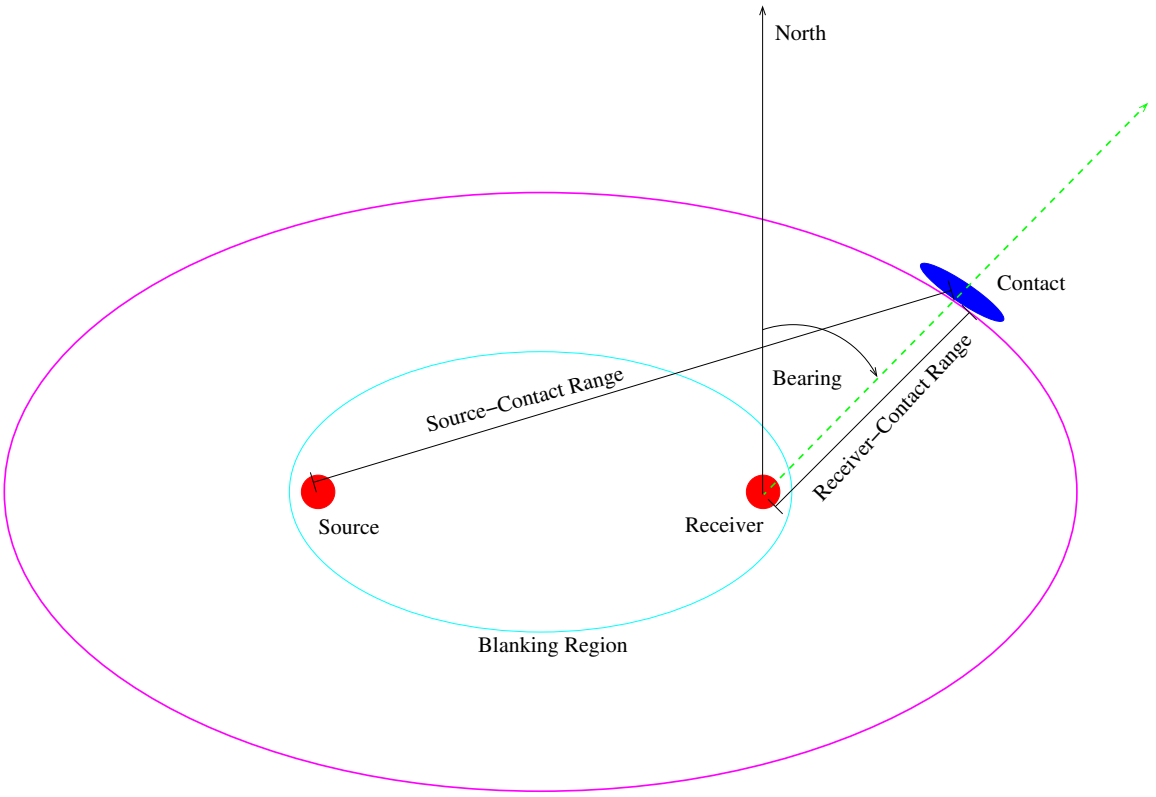

Figure 3.2: Contact localization using bistatic sonar. Bistatic range is the sum of the distance from the source to the contact, d_{tc} and the contact to the receiver d_{rc}. Bistatic sonar also has a blanking region (the cyan ellipse) in which objects cannot be detected.

is $P(\mathbf{x}|z = [b, d]) = \gamma_3^{-1}\mathcal{N}(D(x, y, r, t); d, \sigma_d^2)\mathcal{N}(B(x, y, r); b, \sigma_b^2)$. This looks like a banana-shaped posterior distribution in the Cartesian space (Figure 3.6). These plots were generated using the error statistics from the Metron dataset. Note that the banana-shaped distribution covers a range of nearly 20km.

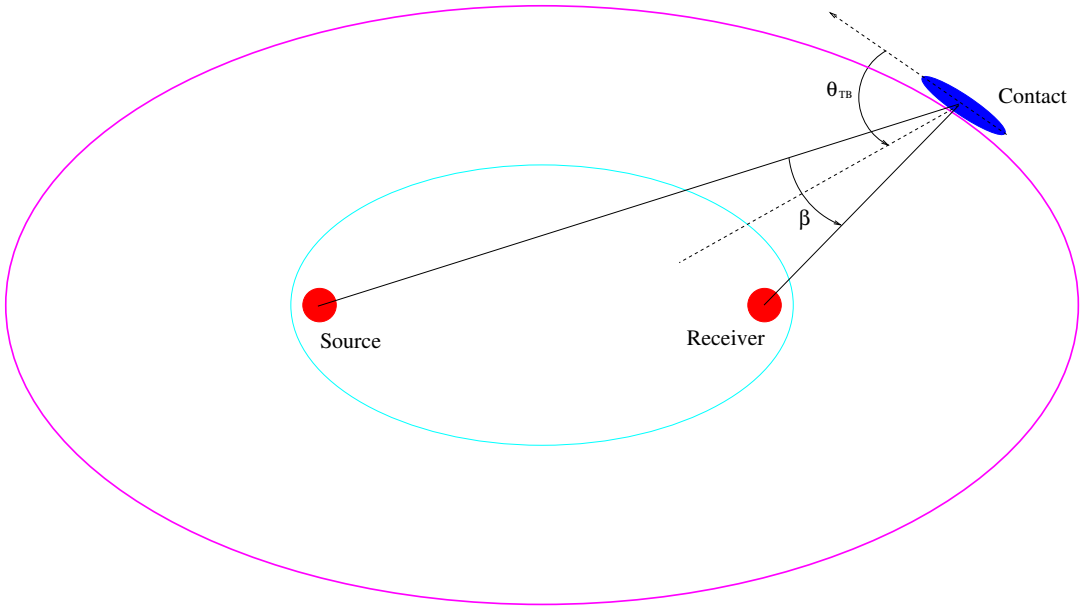

Figure 3.3: Angles in a bistatic sonar geometry. The bistatic angle, β, is the angle formed by the transmitter, contact location, and the receiver. The angle between the contact's heading and the bisector of the bistatic angle is θ_{TB}.

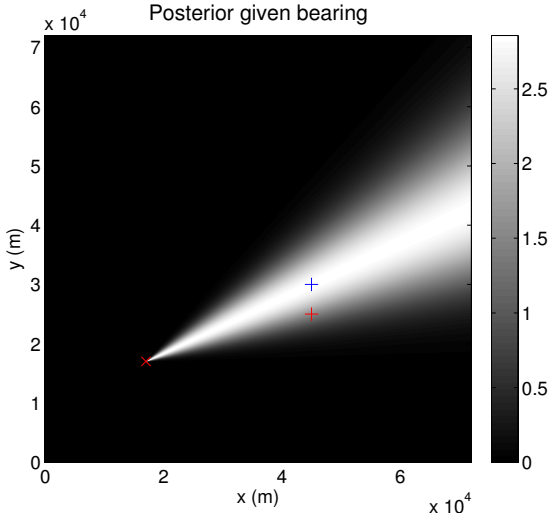

Figure 3.4: Posterior distribution over position, $X = [x, y]$ given a measurement of bearing b. The blue 'x' is the target location, the red 'x' is the receiver and the red '+' is the transmitter.

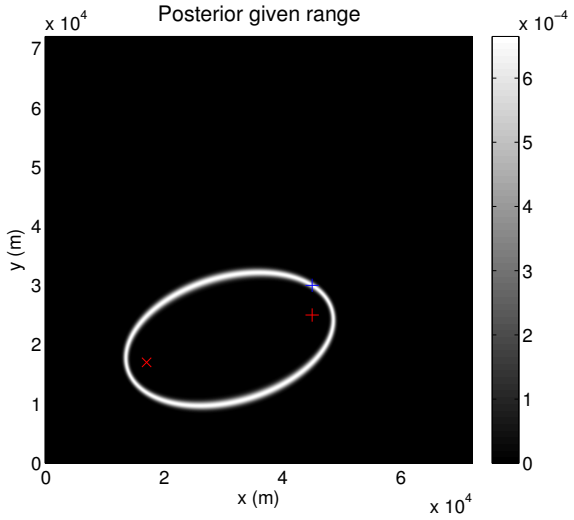

Figure 3.5: Posterior distribution over position given a measurement of range d. The blue 'x' is the target location, the red 'x' is the receiver and the red '+' is the transmitter.

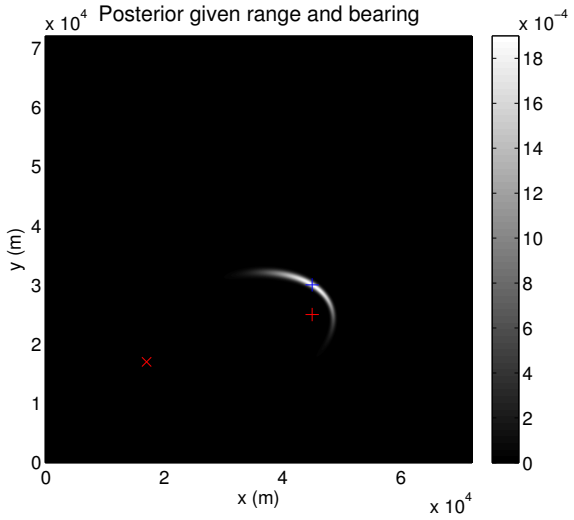

Figure 3.6: Posterior distribution over position given a measurement of range d and bearing b. The blue 'x' is the target location, the red 'x' is the receiver and the red '+' is the transmitter. Note the banana shape.

3.2 Amplitude

In addition to these kinematic features, there are many features that are not only functions of the target's state. The amplitude of the target's return is one such feature. It depends on the aspect-dependent target strength, which is a function of the target's heading and sensor geometry, as well as any environmental factors. The amplitude of a target return is predicted by the sonar equation [50, 3].

The received amplitude is a function of the source level (SL), the aspect-dependent target strength ($TS(\cos\theta_e)$), the transmission loss between the source/receiver and contact location (TL), the ambient noise (AN), the reverberation level (RL), the array gain (AG), the processing gain (PG) and detection threshold (DT).

$$SNR = SL + TS(\cos\theta_e) - (TL_{SC} + TL_{CR}) - (AN \oplus RL) + AG + PG - DT + \eta. \quad (3.5)$$

The sonar equation can be expanded to include other terms, but the datasets used in this work do not model or use them. These terms, and any measurement errors are assumed to be modeled in the noise, η. The transmission losses are modeled by cylindrical spreading, $TL_{AB} = 10\log_{10} d_{AB}$, where d_{AB} is the distance between locations A and B.

The target strength of the return is a function of θ_e as defined in (3.4). In general, objects of interest have a target strength that is highly dependent on aspect - they have a much higher target strength at broadside $(90°, 270°)$, and a lower target strength as the aspect approaches end-on $(0°, 180°)$. Figure 2.1 shows the target strength function used for the PACSim dataset.

3.3 Other Features

The previously mentioned features are the main focus of this work. However, this is definitely not an exhaustive list. In the initial experiments presented in Chapter 6, time domain data was simulated for the contacts in the dataset [6]. The time domain data was compared to expected target and clutter returns and then classified as target or clutter. This was only one approach to using the time domain data, and current research is looking at the development of new features that are discriminative and noise robust [51, 52, 53].

Chapter 4

TARGET TRACKING

Tracking is useful in a variety of sensing modalities and for a variety of reasons. It allows for recognition of suspicious behavior in video, estimating the state of a target in sonar and radar, and projecting possible future states of targets or values of interest. In addition, it allows for the rejection of clutter that is inconsistent with the expected motion of objects of interest.

Tracking can be broken into several building blocks that all interact with each other: data association, track management, and state prediction. There are many tracking and filtering algorithms that can combine two or more of these blocks. Many of these algorithms are described in Section 2.2

In this work, I focus on a tracker that uses JPDA for data association, an extended Kalman filter for state prediction, and SPRT for track management.

4.1 Notation

The algorithms in this chapter are unfortunately a bit notation-heavy, so Table 4.1 defines the terms used.

4.2 Kalman Filtering

Tracking results in this work use an extended Kalman filter for state prediction. A Kalman filter is optimal when you know the statistics of the measurement and process noise and that these noises are Gaussian. These assumptions result in a simple recursive form, but these assumptions are rarely correct. The extended Kalman filter (EKF) is an approximation that can be applied when the state transitions and/or measurement functions are not linear, but can be assumed to be locally linear (and differentiable). Here we duplicate the recursion of the extended Kalman filter.

	Kalman Filtering Notation
$X(t)$	The state variable at time t. In general includes position and time derivatives
$\hat{X}(t\|t-1)$	The predicted state variable at time t, based on the state at the previous time step
$\mathbf{P}(t)$	The state uncertainty at time t
$\mathbf{P}(t\|t-1)$	The predicted uncertainty at time t, based on the state at the previous time step
$z(t)$	The measurement observed at time t
$f(\cdot)$	The state transition function
$h(\cdot)$	The measurement function. Transforms the state variable to the measurement space
	JPDA Notation
z_{ij}	The i^{th} track's j^{th} measurement
g_{ij}	The similarity between the predicted location and measurement z_{ij}
λ	The extraneous return density
P_d	Probability of detection
P_g	The probability that the true contact lies in the gate
β_{ij}	The weight assigned to contact z_ij
$L(H_k)$	The likelihood of an association hypothesis
N_m	The number of tracks without a contact associated with them
N_d	The number of tracks with a contact associated with them

Table 4.1: Table of key tracking notation.

$$\begin{aligned} X(t) &= f(X(t-1), u(t-1)) + w(t-1) \\ z(t) &= h(X(t)) + v(t), \end{aligned}$$

where X is the state vector, $f(\cdot)$ and $h(\cdot)$ are differentiable functions, $w(t-1)$ is the process noise, $z(t)$ is the measurement, and $v(t)$ is the measurement noise. The measurement and process noises are assumed to be zero mean with known covariances, $\mathbf{R}(t)$ and $\mathbf{Q}(t)$ respectively. We assume that the control input, $u(t-1)$ is not present. The current state

is predicted using the motion model, $f(\cdot)$.

$$\hat{X}(t|t-1) = f(\hat{X}(t-1)). \tag{4.1}$$

The predicted covariance estimate, $\mathbf{P}(t|t-1)$ is updated using the Jacobian of the motion model, $\mathbf{F}(t-1) = \frac{\delta f}{\delta X}$,

$$\mathbf{P}(t|t-1) = \mathbf{F}(t-1)\mathbf{P}(t-1)\mathbf{F}^{\intercal}(t-1) + \mathbf{Q}(t-1). \tag{4.2}$$

These predicted estimates are updated using the measurement, $z(t)$, that has been associated with the track (the data association step is discussed in the next section). First, the residual, $\hat{y}(t)$ is calculated,

$$\tilde{y}(t) = z(t) - h(\hat{X}(t|t-1)). \tag{4.3}$$

This can be thought of as the distance between the measurement and the predicted location. Next, the residual covariance $\mathbf{S}(t)$ is calculated,

$$\mathbf{S}(t) = \mathbf{H}(t)\mathbf{P}(t|t-1)\mathbf{H}^{\intercal}(t) + \mathbf{R}(t), \tag{4.4}$$

where $\mathbf{H}(t)$ is the Jacobian of $h(\cdot)$, $\frac{\delta h}{\delta X}$, evaluated at $\hat{X}(t|t-1)$. The near-optimal (near-optimal due to the linearization estimate) Kalman gain, $\mathbf{K}(t)$ is calculated,

$$\mathbf{K}(t) = \mathbf{P}(t|t-1)\mathbf{H}^{\intercal}(t)\mathbf{S}^{-1}(t).$$

The Kalman gain is used to update both the state estimate and estimated covariance,

$$\begin{aligned} \hat{X}(t) &= \hat{X}(t|t-1) + \mathbf{K}(t)\tilde{y}(t) \\ \mathbf{P}(t) &= (\mathbf{I} - \mathbf{K}(t)\mathbf{H}(t))\mathbf{P}(t|t-1), \end{aligned} \tag{4.5}$$

where $\mathbf{I}$ is an identity matrix of the appropriate size.

For specific implementations in this work, a nearly constant velocity model with the state vector $X = [x\ y\ x'\ y']$ is used, resulting in a linear state transition matrix $\mathbf{F}$. Specific transition and measurement matrices $\mathbf{F}$ and $\mathbf{H}$ (and the underlying non linear $h(\cdot)$) will be described in each chapter as they are used.

Recall the residual calculation in (4.3) requires a measurement to be associated with a track. This association step is simple with a single contact and a single track, however can

become much more complex in a scenario where there are a lot of false contacts, multiple tracks, and tracks that do not have a contact associated with them. The term "data association" refers to the process of resolving these ambiguities which are present in real tracking data.

4.3 Data Association

There are a variety of data association techniques, many of which are briefly mentioned in Chapter 2, but here we focus on JPDA. To properly explain JPDA, we first start with a simpler association technique, PDA.

Using the prediction step (4.1), the current position of each track $\hat{X}_i(t|t-1)$ is estimated. This is done for all N tracks, $i = 1..N$. Next, contacts which are too far from the predicted location $\hat{X}_i(t|t-1)$ are rejected from consideration (for that track only). This step, called gating, is necessary to make the (J)PDA algorithms computationally feasible. The definition of "too far" for the gating step is variable for specific tracking problems, we use an elliptical gate whose shape and size is dependent on the predicted covariance matrix (4.2) scaled by a constant. Other common approaches include k-nearest-neighbor (only look at the k closest contacts) and rectangular gating [54].

4.3.1 Probabilistic Data Association

Once contacts are gated, the residuals are calculated for each of the M contacts z_j in track i's gate:

$$\tilde{y}_{ij} = z_j - \mathbf{H}\hat{X}_i(t|t-1).$$

The distance, d_{ij}^2 is calculated for each of the residuals using the tracks residual covariance $\mathbf{S}_i$ from (4.4):

$$d_{ij}^2 = \tilde{y}_{ij}^{\mathsf{T}}\mathbf{S}_i^{-1}\tilde{y}_{ij}.$$

The distances are converted into similarities, assuming a Gaussian distribution:

$$g_{ij} = \frac{e^{-d_{ij}^2/2}}{(2\pi)\sqrt{|\mathbf{S}_i|}}. \tag{4.6}$$

A term is needed to account for the probability that there was a missed detection,

$$q = \lambda(1 - P_d P_g), \tag{4.7}$$

where λ is the false alarm rate and P_g is the probability that the target return will be in the gate (usually assumed to be =1). The similarities are normalized and the probability of a missed detection is taken into account to calculate the association weights:

$$\begin{aligned} \beta_{ij} &= \frac{P_d g_{ij}}{q + \sum_{j=1}^{N} P_d g_{ij}} \\ \beta_{i0} &= \frac{q}{q + \sum_{j=1}^{N} P_d g_{ij}}, \end{aligned}$$

The association weights, β_{ij} are used to obtain the final estimate, $\tilde{y}_i$ for the Kalman filter update (4.6).

$$\tilde{y}_i = \sum_{j=1}^{N} \beta_{ij} \tilde{y}_{ij} + \beta_{i0} \hat{X}_i(t|t-1). \tag{4.8}$$

4.3.2 Joint Probabilistic Data Association

JPDA is an extension of PDA to account for crossing targets and overlapping gating regions. JPDA calculates the association probabilities jointly for all non-independent targets. Figure 4.1 is a simple cartoon of a scenario in which JPDA will perform better than PDA. The two existing tracks are indicated by the red and blue filled circles with their respective gates indicated by solid lines. There are two possible contacts, the filled triangles. In PDA, the two tracks would be evaluated separately, and the blue track would update by placing a high weight on the black contact (because it is close to the predicted location) and a low weight on the magenta contact. Next, the red track would update using only the black contact.

In JPDA, the association step is done jointly. The result of JPDA will be the blue track having a high weight on the magenta contact and a low weight for the black contact. The red track will have a high weight on the black contact (as well as a small probability of a missed detection). The blue track will be updated more towards the "top" of the cartoon and the red track will stay lower, resulting in improved tracking performance.

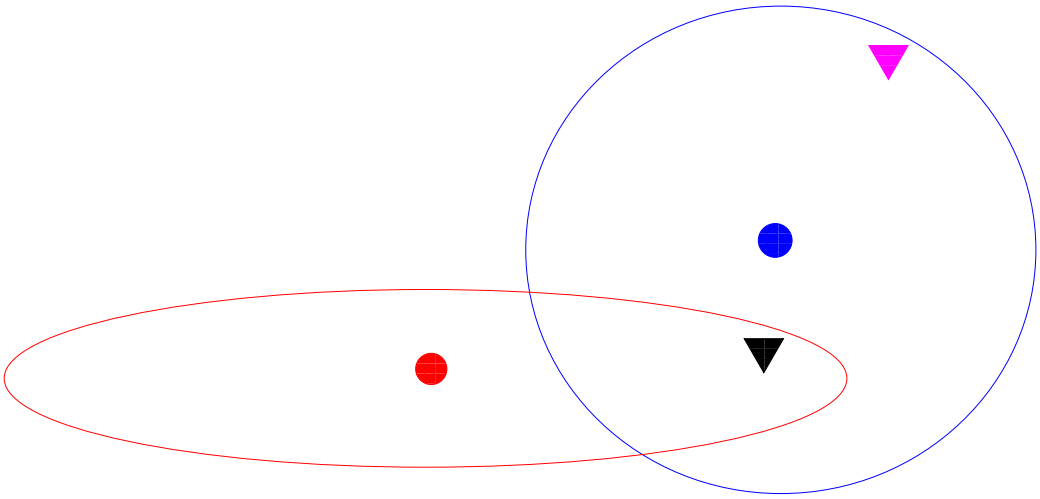

Figure 4.1: Simple association scenario where JPDA would provide better results than PDA. The red and blue filled circles are existing tracks, surrounded by their respective gates indicated by solid lines. There are two contacts near both targets, indicated by the colored triangles.

This is done by the creation of an association hypothesis, H_k for each of the possible track-contact associations (including missed detections). JPDA diverges from PDA after the measurement similarities are calculated in (4.6). For each hypothesis, the likelihood of the hypothesis, $L(H_k)$ is calculated using the similarities ($\{g_{ij}\}$):

$$L(H_k) = (1 - P_d)^{N_m} P_d^{N_d} \lambda^{(1+N_d)} \prod_{g_{ij} \in H_k} g_{ij}, \tag{4.9}$$

where $g_{ij} \in H_k$ is the set of all contacts which are associated with a target in the specific hypothesis H_k. N_m is the number of tracks without an associated contact, N_d is the number of tracks with an associated contact, and λ is the false alarm rate.

The probability of each hypothesis, $P(H_k)$ is calculated for each of the K hypotheses H_k is calculated by normalizing:

$$P(H_k) = \frac{L(H_k)}{\sum_{q=1}^{Q} L(H_q)}. \tag{4.10}$$

The association probability β_{ij} for contact j to track i is the sum of all the probabilities that correspond to any hypothesis that associates contact j with track i. For notational

convenience, let A_{ij} be the set of all hypotheses that associate contact j with track i, then β_{ij} is defined:

$$\begin{aligned} \beta_{ij} &= \sum_{H_k \in A_{ij}} P(H_k), \\ \beta_{i0} &= 1 - \sum_{j=1}^{N} \beta_{ij}. \end{aligned} \tag{4.11}$$

Finally, the association probabilities are used to obtain the final estimate $\tilde{y}_i$, for the Kalman filter update (4.6):

$$\tilde{y}_i = \sum_{j=1}^{N} \beta_{ij} \tilde{y}_{ij} + \beta_{i0} \hat{X}_i(t|t-1), \tag{4.12}$$

noting that this is the same update as the PDA update in (4.8).

4.4 Track Management

In the previous two sections, we assumed that there were already existing tracks that needed to be updated as measurements come in. In this section, we describe the bookkeeping that must be done to initiate, confirm and delete tracks. This is an important step in tracking and it can also be modified to include feature information as described in Chapter 6.

For track initiation, a single-point method is used. All contacts that are not associated with an existing target track are assigned to a new track. This method has the advantage of being simple, however it can suffer from large errors in the velocity estimate when the measurements from the next ping arrive. This can also create a large number of potential tracks, but it has not proven to be prohibitive in the research presented here. A common approach to reduce the number of false tracks is to use a higher threshold on a feature of the contacts (e.g. amplitude [45]). This will hopefully result in fewer clutter tracks without rejecting too many target contacts.

After a track has been initialized, one must decide if the track is due to clutter or target contacts and perform the appropriate action (discard or present the track to the user). This track maintenance step is important and adds another tuning step to a tracking algorithm. In this work, the *sequential probability ratio test* (SPRT) is used to manage tracks. For each

track, a score is maintained and updated every time a new set of measurements arrives:

$$LLR_i(t) = LLR_i(t-1) + \Delta L_K + \Delta L_S + \Delta L_{other}, \tag{4.13}$$

where ΔL_K is the update due to the kinematic information (how well the measurements match the predicted location), ΔL_S is the update due to any signal information, as well as an additive term for any other information, such as classification information (as presented in Chapter 6). The score is initialized to $LLR_i(0) = \log(P_d)$. If at any time step a track does not have a contact associated with it, the update is $\Delta L_{miss} = \log \frac{1-P_d}{1-P_{fa}}$. The user sets a deletion and confirmation threshold, depending on their desired false track rate, latency, and other needs. When a track score exceeds the confirmation threshold, it is considered confirmed until it falls below the deletion threshold, at which point it is deleted. It can be beneficial to use a different deletion threshold for confirmed and unconfirmed tracks to increase robustness of confirmed tracks to missed detections.

Other methods for track management exist, such as $M/N/K$ style logic. This well-known logic is quite simple. If a track has a contact associated with it for M of the last N pings, it is confirmed. If a track does not have a contact associated with it for the past K of N pings, it is deleted. This is a simple, yet well performing track management style, however it does not allow for the integration of external information in the same way that the SPRT update does (4.13), so it is not used in this work.

4.5 Tracking Metrics

Throughout this work, different metrics will be used to evaluate tracking performance. Most of these were defined for the *multistatic tracking working group* (MSTWG) to evaluate and compare different trackers on the same dataset [55]. The focus in this work is on the following metrics: *track fragmentation* (TFrag), *track localization error* (TLE), *track false alarm rate* (TFAR), and *track probability of detection* (TPd). Track fragmentation measures how many different tracks are needed to track a single target over the length of a scenario, and is ideally one (lower values are better). Track localization error measures the average distance from a target track to the true target's location and is ideally zero (lower values are better). Track false alarm rate measures how frequently a false track is declared and is

ideally zero (lower values are better). Track probability of detection measures what percent of the scenario a target was tracked and is ideally one (higher values are better).

The ability to declare a track as a true track is an important step in evaluation of these metrics, however it can be difficult. In simpler tracking scenarios, where contacts are used individually, one can declare a track to be a true track if it consists of greater than some fraction of true contacts. However, many of the results in this work use "fused" contacts, sometimes combining all the contacts on a single ping into completely new measurements. In these specific sections, the metrics are modified when possible and skipped when not possible.

Chapter 5

DATASETS

Several datasets were used to develop and evaluate the techniques presented in this work. The majority of them were created for the *Multistatic Tracking Working Group* (MSTWG). This is a working group in the International Society of Information Fusion, a diverse group of people from industry and academia that develop and compare multistatic tracking algorithms on common datasets. The datasets were created by members of the group and each provide unique challenges in tracking.

5.1 TNO Blind dataset overview

The first dataset I worked with is the TNO blind dataset. This dataset was developed for the MSTWG by researchers at TNO Defence, Security, and Safety (The Hague, The Netherlands). The dataset has three source-receiver pairs that move in the simulation area over the course of the scenario. Both CW and FM pulses are simulated, with Doppler information available for the CW pulses. Four targets are present throughout the simulation, each with characteristics that can cause difficulty in tracking. Figure 5.1 shows the four true target tracks and the source/receiver paths. Target 1 is following a sinusoidal path and is therefore constantly accelerating. Two of the targets have periods of time in which there are very few target contacts available. The existence of gaps in true contacts and the constantly accelerating target must be considered when tuning tracking parameters.

This dataset was used to compare the integration of feature data into tracking at different levels, as presented in Chapter 6. It turned out to be a relatively easy tracking simulation, however it allowed for the development of a JPDA-based tracker that has the flexibility to include classification information at multiple levels.

This dataset contains some registration errors, which will manifest as several parallel tracks in the tracking results. In addition, contacts in the dataset are not labeled as target

or clutter - making it somewhat difficult to evaluate tracking metrics. This lack of labels also presents problems for the creation of time series data for the contacts, this is discussed in Chapter 6.

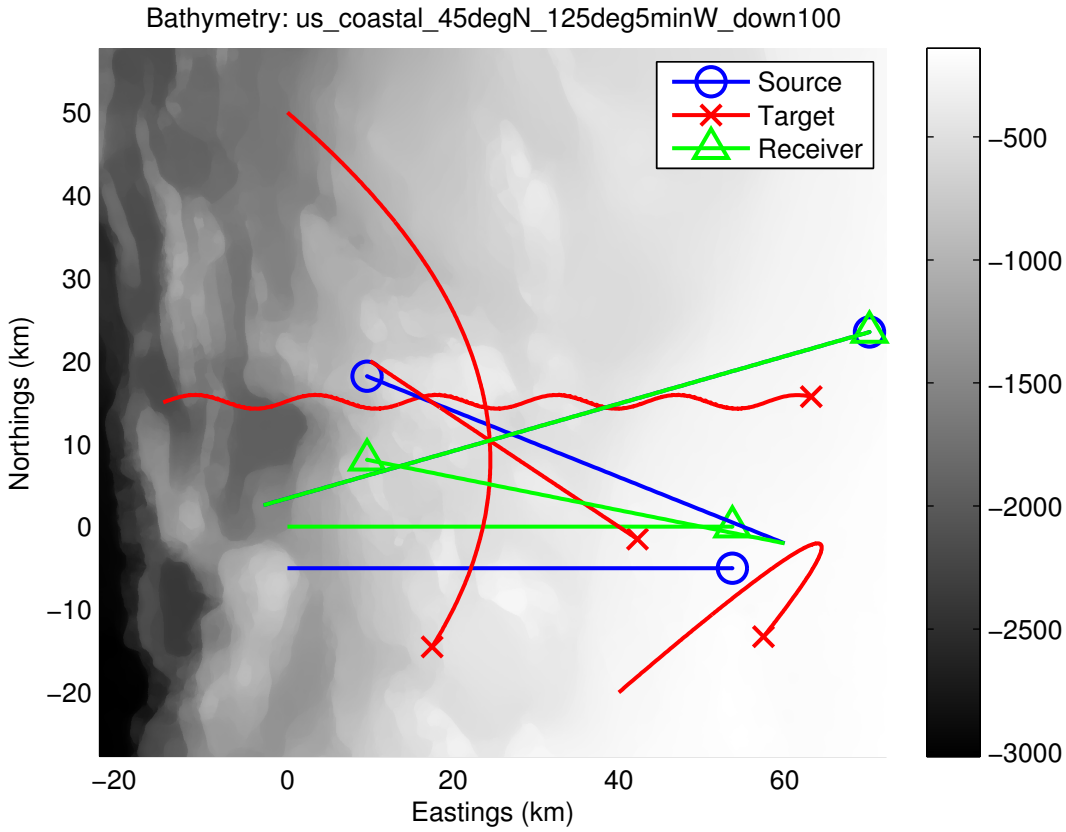

Figure 5.1: TNO dataset superimposed on the bathymetry used. The colorbar indicates the depth of the water in meters. The target paths are marked in red and end at the position marked with an 'x'. The source trajectories are marked in blue and end at the circles. The receiver trajectories are marked in green and end at the triangles.

5.2 Metron

The next dataset created for the MSTWG was the Metron dataset. It was created by Metron Inc as a first attempt at a multistatic dataset with a large number of transmitters and receivers. In general it is characterized by having very large measurement errors, low

probability of detection for targets and a higher clutter rate. The modeling is relatively simple, with aspect-dependent targets, simple cylindrical spreading and additive Gaussian noise on all the measurements. This dataset includes true target locations as well as labels on all the contacts, allowing for easier development and evaluation of new methods.

This was also the first dataset that had a large number of receivers (25) and transmitters (4). Having this many receivers accompanied by the low probability of detection for the targets resulted in our SPRT/JPDA-based tracker failing to work for several reasons: the track management did not work well when there were so many missed detections, the high clutter rate caused issues with JPDA runtime, and the measurement errors further exacerbated these issues.

To overcome the difficulties in the Metron dataset, I developed a preprocessing step that was added to the tracking framework. The preprocessing step combines all contacts on all receivers from a single transmitted ping. This allows a standard JPDA-based tracker to handle the multistatic data in the Metron dataset. It also fuses the measurement information across receivers, ideally decreasing measurement error. The preprocessing step and a newer clustering technique are presented in Chapters 9 and 10.

The Metron dataset is a simulated dataset with 4 scenarios. Each scenario contains multiple targets, four transmitters and 25 receivers. Figures 5.2- 5.5 shows the geometry of the sensor field and the paths of the targets for the different scenarios. The Metron dataset was generated using the following parameters:

- Sound speed 1500 m/s
- TDOA error: mean $0.0s$, standard deviation $0.4s$
- Bearing error: mean 0.0°, standard deviation 8.0°
- Bistatic Doppler error: mean 0.0 m/s, standard deviation 0.5 m/s
- Amplitude error: mean 0 dB, standard deviation 8 dB
- Source level 65 dB for CW pings, 69 dB for FM pings

- 200 pings, 180 second interval, alternating CW/FM
- Blanking region of 1 second, maximum return time of 90 seconds
- $P_d < 0.13$ for both types of pings. There are a few targets in scenario 4 with lower P_d.

The Metron dataset is quite difficult due to the large error on the bearing measurements and the large TDOA error on the FM pings. It also contains a large amount of clutter, with an average of 15.1 false contacts per receiver per ping. The measurement rate for the Metron dataset is also quite low, with a 180 second inter-ping interval.

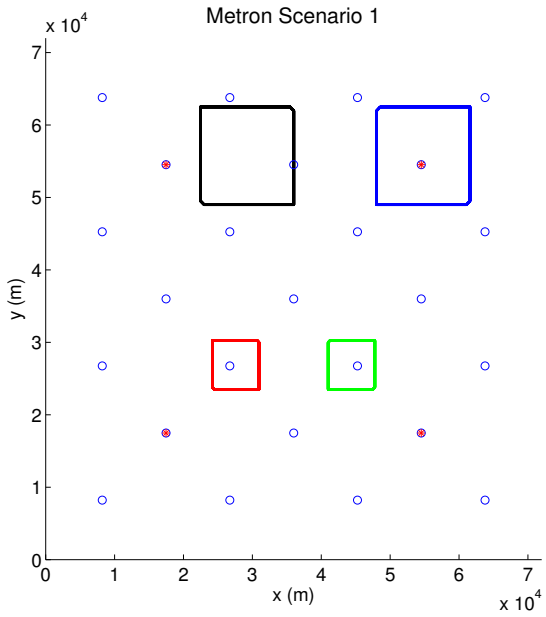

Figure 5.2: Metron Scenario 1. The four transmitters are marked with stars and the 25 receivers are marked with circles. There are four targets that travel in rectangular paths and repeat the loop 4 times through the course of the scenario.

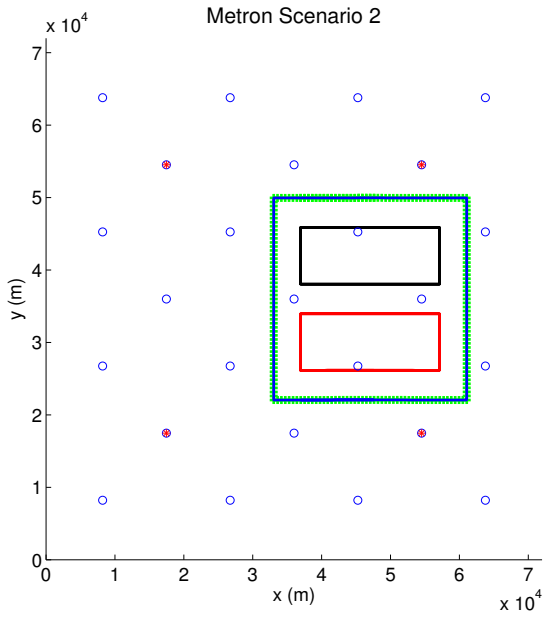

Figure 5.3: Metron Scenario 2. The sensor layout is the same. There are four targets that travel in rectangular paths. The blue and green targets travel on the same path, but are offset.

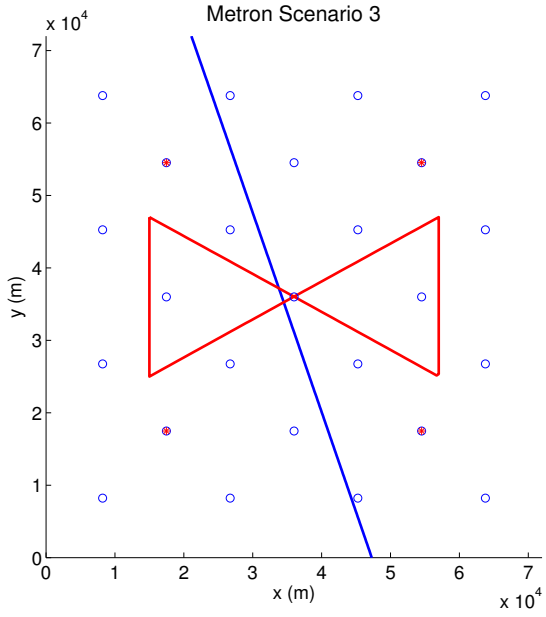

Figure 5.4: Metron Scenario 3. This scenario only has two targets.

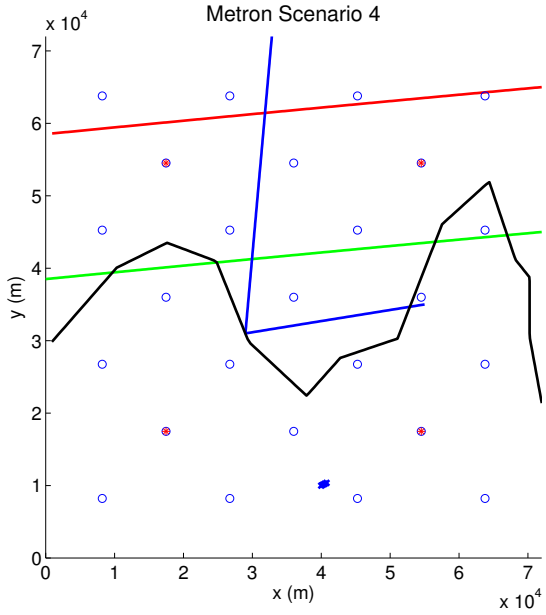

Figure 5.5: Metron Scenario 4. There are four moving targets. There are also four closely spaced fixed targets marked with a blue 'x'. The moving blue target extends further to the north of the simulation area.

5.3 PACSim Dataset

The next dataset used in this work is the Passive and Active Contact Simulator (PACSim). The PACSim tool was developed by Doug Grimmett and can model passive and active sonar in a high fidelity environment [56]. This dataset was simulated after the Metron dataset in an attempt to create data that had less noise on the measurements but still resulted in a challenge for tracking. The PACSim dataset was generated with the following parameters:

- Sound speed 1500 m/s
- TDOA error: mean 0.0s, standard deviation 0.1s for FM, 0.4s for CW
- Bearing error: mean 0.0°, standard deviation 4.0°
- Bistatic Doppler error: mean 0.0 knots, standard deviation 0.4 knots

- Amplitude error: mean 0 dB, standard deviation 8 dB
- 481 pings, 180 second interval, alternating CW/FM
- Two fixed clutter points for each scenario
- P_d varies by target and waveform (see Table 5.1)

The PACsim dataset has three scenarios of varying difficulty, with targets that have P_d that vary greatly by ping type and target. Table 5.1 shows the P_d by target, scenario, and ping type. For Scenarios A and B, the clutter rate is 20.9 false contacts per receiver per ping for CW pings and 23.4 for FM pings. Scenario C has approximately twice as much clutter, with 41.6 for CW pings and 47.0 for FM pings. This, in combination with the low probability of detection makes Scenario C a much harder task. Figures 5.6-5.8 shows the sonar field geometry for all the scenarios and the targets for Scenario B.

	Target	1	2	3	4	5	6
A	CW	0.21	0				
A	FM	0.47	0.44				
B	CW	0.14	0.15	0.10	0.01	0	0
B	FM	0.53	0.52	0.11	0.11	0.47	0.47
C	CW	0.08	0.02	0.09	0.01	0	0
C	FM	0.47	0.11	0.12	0.10	0.48	0.47

Table 5.1: Comparison of probability of detection across scenarios and transmitted waveform type.

The PACsim dataset includes a simulated feature for each contact. Clutter contact feature scores are distributed $\mathcal{N}(0, 10)$. Targets are assigned one of two classes and their feature scores are distributed $\mathcal{N}(\mu_i, 10)$, where $\mu_1 = 3, \mu_2 = 6$. The simulated feature provides interesting opportunities for incorporation of feature data into tracking, however,

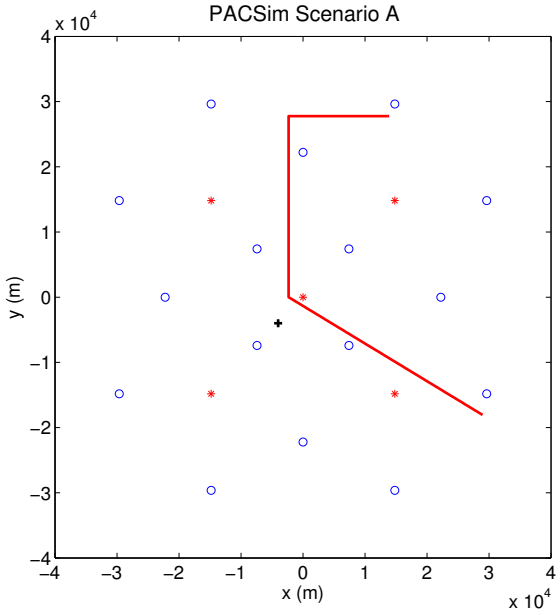

Figure 5.6: PACSim Scenario A. The blue circles are the receivers and the red stars are the sources. There is one moving target, marked in red and one fixed clutter point, marked with a black '+'.

the closeness of the distributions makes this difficult. In Section 10.4.2, we discuss how fusing contacts within a cluster allows for improved discrimination between classes.

The PACsim dataset uses a more complicated environmental model to estimate the amplitude of contact returns. This results in a simulation that is more realistic, however it is more difficult to estimate the expected SNR for a target at a certain state. The PACsim dataset's amplitude model is:

$$\begin{aligned} \hat{SNR} &= SL + TS(\cos\theta_e) - (TL_{sc} + TL_{cr}) - \\ &\quad (AN \oplus RL) + AG + PG - DT, \end{aligned}$$

where $\oplus$ denotes a sum in the power domain. Many of the parameters are unknown or difficult to model, so we model the amplitude as:

$$\hat{SNR} = TS(\cos\theta_e) - (TL_{sc} + TL_{cr}) + \eta. \tag{5.1}$$

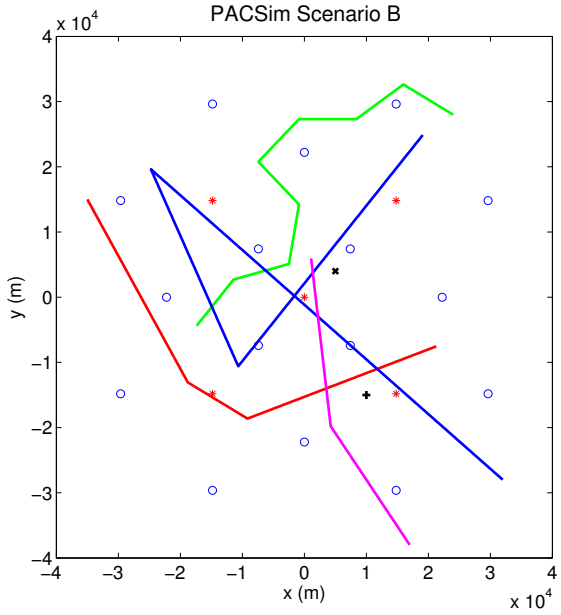

Figure 5.7: PACSim Scenario B. There are four moving targets, marked with colored lines. There are two fixed clutter points, marked with an 'x' and a '+'. The green and red targets are the easier, higher P_d targets.

where $\eta \sim \mathcal{N}(\mu_{amp}, \sigma_{amp})$. The transmission losses (TL_{sc} and TL_{cr}) are modeled by cylindrical spreading, $TL_{AB} = 10 \log_{10} d_{AB}$. The parameters μ_{amp}, σ_{amp} are estimated using the labeled target contacts from Scenario A of the PACsim dataset, with different values for CW and FM pings.

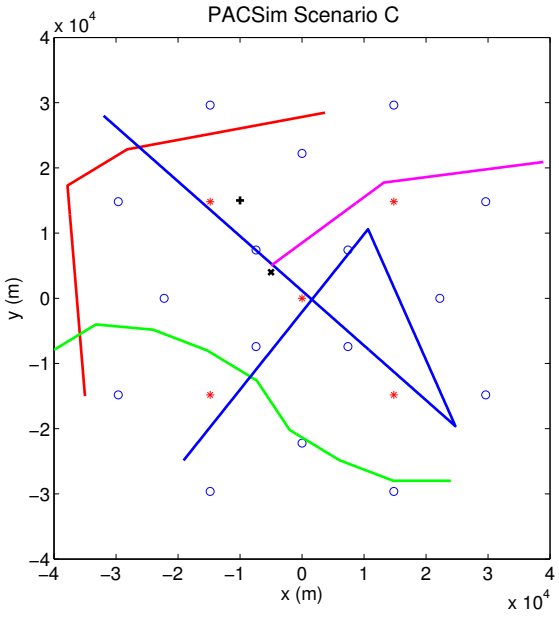

Figure 5.8: PACSim Scenario C. There are four moving targets, marked with colored lines. There are two fixed clutter points, marked with an 'x' and a '+'. The red target is the easier, higher P_d target.

Chapter 6

INTEGRATING FEATURES INTO A JPDA-BASED TRACKER

In this section, several methods of integrating classification into tracking are compared. In the related work section, I reviewed several of these techniques, and this section goes in depth into the various different levels of integrating this information.

6.1 Classification of Contacts using a Time Domain Template

To evaluate the tracker and classify contacts, it is necessary to know what contacts are true and which are clutter contacts. In the case of the TNO dataset, though, no contact labels are available; only the true target positions are given. To overcome this, Will Mortensen, a UWEE masters student and I used a simple though somewhat arbitrary heuristic to guess the labels of the contacts in each ping: for each target in the scenario at that ping time, the nearest contact within 600 meters of the target's true position is assumed to have been produced by that target. Thus, if no contacts are within 600 m of a target, then that target is assumed to have produced no contacts. If two or more targets are sufficiently near each other that multiple targets would be assigned to the same contact, we arbitrarily chose the highest-numbered target, though it would perhaps be better to choose the closest target. Figure 6.1 shows the results of this association step.

After the contacts were labeled, Will simulated time-domain data for them and created a classifier to create class conditional likelihoods. Further details can be found in the paper presented at Oceans 2010 Seattle [6].

6.2 Track Management

The class conditional likelihoods obtained from the classifier can be used to form a likelihood ratio

$$\alpha_j = \frac{P(c_T|z_j)}{P(c_0|z_j)} \tag{6.1}$$

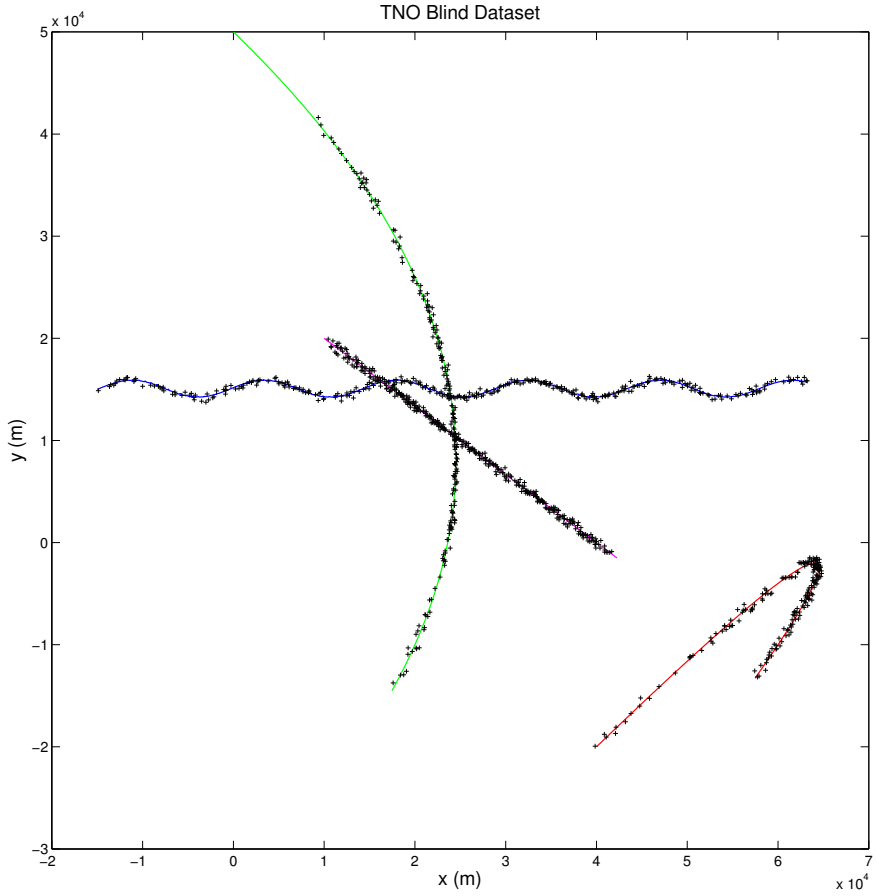

Figure 6.1: Targets in the TNO dataset with labeled true contacts. Note that the red and green targets have areas where there are very few associated "true" contacts.

for each of the contacts z_j, where $P(c_T|z_j)$ is the probability that contact j is a target and $P(c_0|z_j)$ is the probability that contact j is clutter.

The simplest approach is to remove any contacts whose likelihood ratio is less than a threshold. This method has the added benefit that it reduces the number of contacts that are passed to the data association stage of the tracker, reducing computational requirements. In situations where hard thresholding is undesirable or unnecessary, the likelihood ratio can be incorporated into the SPRT likelihood update. For each contact z_j associated with a

track, the track's score is modified with a classification update ΔL_{C_j} (Equation 4.13):

$$LLR_i(t) = LLR_i(t-1) + \Delta L_K + \Delta L_S + \Delta L_{C_j}, \tag{6.2}$$

where $\Delta L_{C_j} = \ln(\alpha_j)$. This method has the advantage of being less sensitive to misclassification. If a target is misclassified as clutter, the contact will not be removed completely and is still used to update the track state.

6.3 Data Association

JPDA has several levels at which classification information can be integrated to improve performance. In this section, the different levels and the advantages of each are discussed. Recall the similarities g_{ij} calculated in Equation 4.9. These are the initial similarities calculated only from the distance from the predicted location to the contacts within the gate. These similarities can be modified to include the class likelihood ratio α_j as follows:

$$\hat{g}_{ij} = \alpha_j g_{ij}. \tag{6.3}$$

The modified $\hat{g}_{ij}$ are input into Equation 4.9 in place of the standard g_{ij}.

The classification information can also be integrated later, after the association weights β_{ij} have been calculated in Equation 4.11. Recalling that β_{ij} is the weight that is applied to contact j for track i, the straightforward modification is as follows:

$$\begin{aligned} \hat{\beta}_{ij} &= \gamma^{-1} l_j \beta_{ij} \\ \gamma &= \sum_{j=1}^{N} l_j \beta_{ij}, \end{aligned} \tag{6.4}$$

where γ is normalizing the association weights to sum to one. After normalization, $\hat{\beta}_{ij}$ replaces β_{ij} in Equation 4.12.

These two methods are similar, but because there is a normalization step in Equation 4.10, they will result in different weights and therefore different performance.

6.4 Results

The metrics used to evaluate the different methods of integrating classification information into tracking are based on the metrics in [55]. TLE, TFrag, TPD and TFAR are used to

compare methods.

6.4.1 Tracking with Classification

Using the metrics described in Section 6.4, the effect of integrating classification information into track initiation is first evaluated using the TNO dataset. The three methods compared are SPRT track initiation with no classification information (SPRT, our baseline system), SPRT track initiation with hard thresholding of contacts (Hard), and SPRT track initiation with the track score being modified by $\Delta L_{C_j} = \ln(\alpha)$ at each association step (Soft). Table 6.1 compares these methods.

Figure 6.2 shows the confirmed tracks that result from using SPRT track initiation with no classification information, the baseline for comparisons. When comparing the Hard thresholding and the inclusion of the likelihood ratio into the track score to the baseline (referred to as Soft), the Soft method performs better than both the hard threshold and baseline. The largest improvements are a large increase in TPD for Target 2 and large decrease in TLE of Target 4. TFRAG is also decreased with both the hard thresholding and soft methods. In addition, the tracks which are confirmed tend to be of a higher quality with the soft method. This is visually evident in Figure 6.4. Note the result of the improved TPD of target 2 with the soft method.

The next method of classification integration is to modify the association probabilities in JPDA. The baseline method is JPDA with SPRT track initiation. Weighting the g_{ij} scores before the JPDA algorithm and weighting the β_{ij} terms are compared to the baseline in Table 6.2.

Weighting the β_{ij} terms is better than weighting the g_{ij} terms across almost all the metrics, the exception being TFRAG for track 3. Compared to the baseline, weighting the β_{ij} shows improvement in TLE for targets 1 and 2 and in TPD for target 4. There are some negative performance changes, notably in TFRAG for targets 1, 3, and 4, and in TLE for targets 3 and 4. This can also be seen in Figures 6.5 and 6.6 when compared to the baseline in Figure 6.2.

Metric	SPRT	Hard	Soft
TPD			
1	0.92	0.92	0.92
2	0.56	0.42	0.79
3	0.87	0.87	0.87
4	0.91	0.97	0.91
TLE			
1	1205	1207	1112
2	678	645	613
3	1287	1084	1088
4	947	707	584
TFRAG			
1	6	5	3
2	4	2	3
3	4	3	3
4	7	4	3
TFAR	0	0.0139	0.002

Table 6.1: Comparison of methods of integrating classification information into track initiation.

6.5 Conclusions

The design and implementation of a framework for simulation of contact time series in realistic underwater environments is a useful tool for future research. It allows the modification of existing contact-based tracking datasets to include time series data that can be used to develop classification techniques. These techniques are useful on their own, but many methods exist for integrating the classification results into tracking algorithms. Several of these methods have been compared on the TNO Blind dataset, with mixed results. The

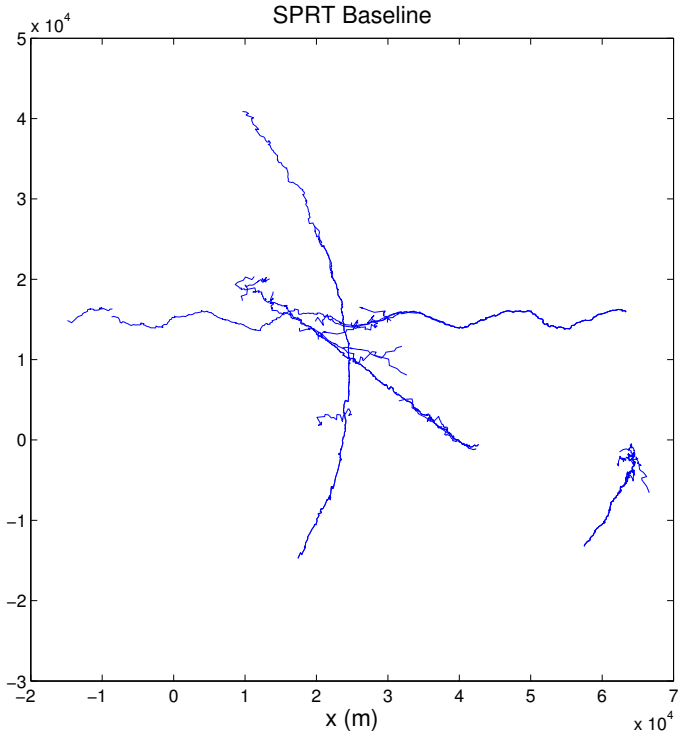

Figure 6.2: Confirmed tracks resulting from SPRT track initiation.

TNO dataset is a relatively easy tracking dataset, as can be seen by the Baseline SPRT results. Still, improvements were seen for some of the targets in TLE, TFRAG, and TPD, especially using the Soft method to adjust the track scores in the SPRT. These results are from the integration of a very simple binary classifier based on the time series of the returns.

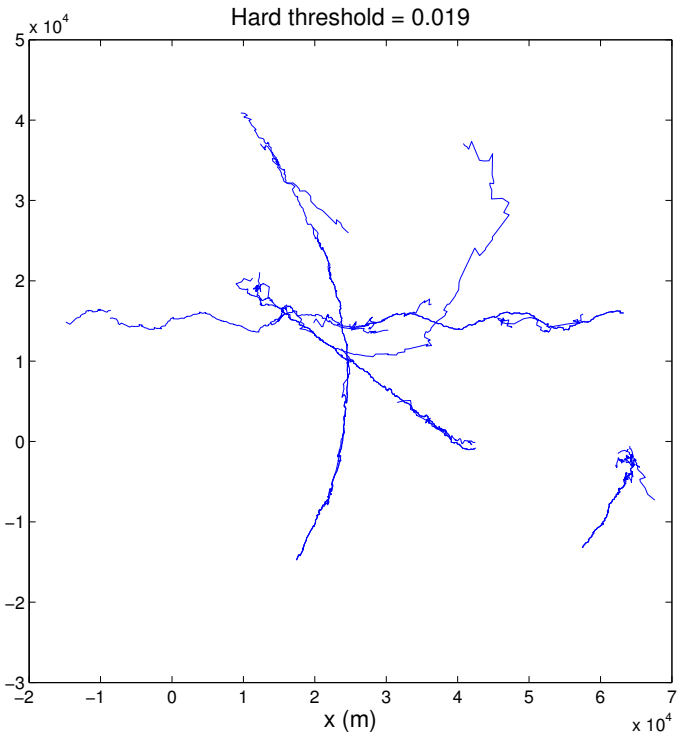

Figure 6.3: Confirmed tracks from SPRT track initiation. All contacts are thresholded such that any contacts j with $l_j < 0.0192$ are discarded.

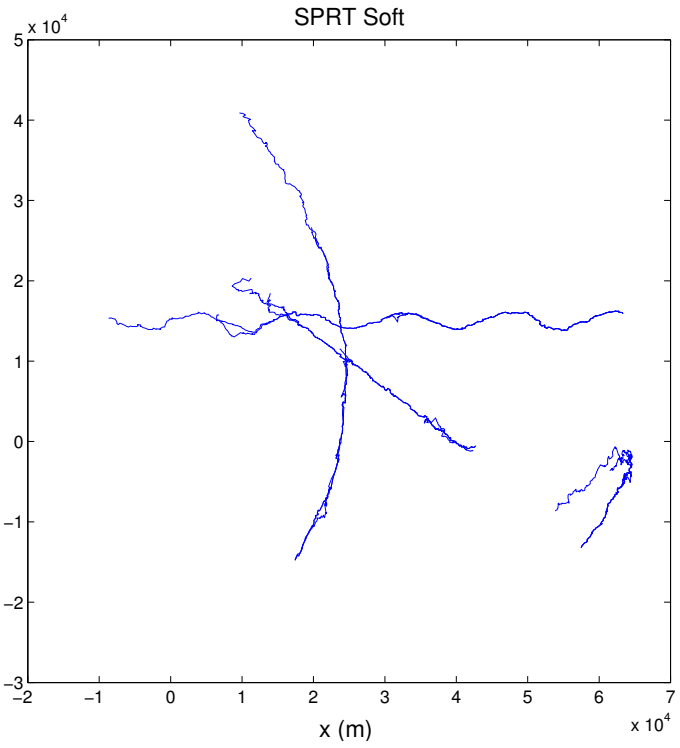

Figure 6.4: Confirmed tracks from SPRT track initiation. The SPRT track score update includes $\Delta C = log(l_j)$.

Metric	Baseline	g_{ij}	β_{ij}
TPD			
1	0.92	0.92	0.92
2	0.56	0.54	0.56
3	0.87	0.87	0.86
4	0.91	0.97	0.97
TLE			
1	1205	1474	1190
2	678	676	504
3	1278	1367	1328
4	947	1262	1004
TFRAG			
1	6	10	8
2	4	3	3
3	4	6	7
4	7	10	9
TFAR	0	0	0.002

Table 6.2: Comparison of methods of integrating classification into JPDA weighting.

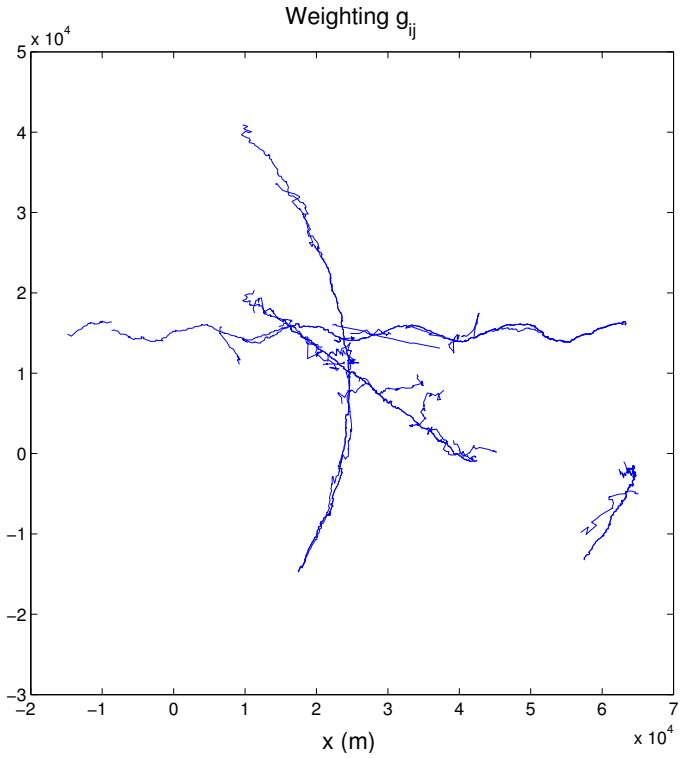

Figure 6.5: Confirmed tracks from weighting the JPDA similarities g_{ij}.

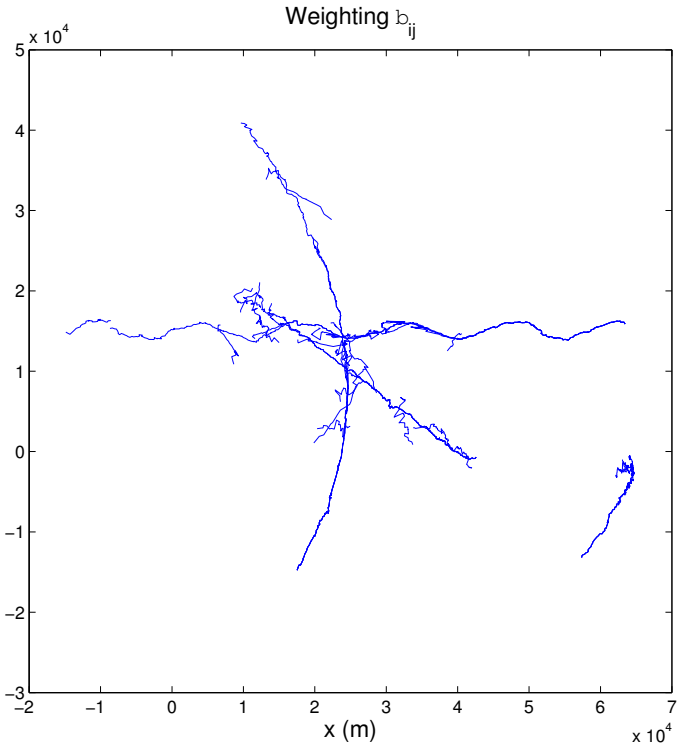

Figure 6.6: Confirmed tracks from weighting the JPDA association weights β_{ij}.

Chapter 7

IMPROVING CLASSIFICATION OF CONTACTS BY INCORPORATING TRACKING INFORMATION

In the previous chapter, I overviewed several different methods for integrating classification information into a JPDA based tracker. Here, I look at the other side of the coin, the integration of tracking information into classification. This work was presented at the 2012 Oceans Conference in Kona, Hawaii.

7.1 Motivation

As described in Chapter 3, many features of a contact's return are aspect-dependent. The aspect-dependent nature of the features requires appropriate consideration when designing a classifier. In the previous chapter, this was done explicitly by comparing the return's time series to a template at all possible aspect angles. In terms of classification, this can be viewed as using a uniform prior on the target's heading. In this section, the track state from a Kalman filter (and its uncertainty) is used to improve classification of contacts based on their target strength and Doppler measurements.

7.2 Classification and Incorporating Uncertainty

The measurements used to classify contacts in this section (Doppler and amplitude) have uncertainty included with them. In this case, it is simply additive Gaussian noise. In addition to the measurement error, there is also uncertainty associated with the estimate of target aspect used to predict the feature values. Accounting for the various sources of uncertainty is key to obtaining good classification performance.

7.2.1 Classification

A maximum likelihood approach is taken to classification:

$$\hat{c} = \arg\max_{c \in C} p(z|c), \tag{7.1}$$

where $\hat{c}$ is the predicted class, C is the set of all classes, z are the features available, and c is a single class. Using the law of total probability and assuming a distribution over velocity $v = \{|v|, \theta_v\}$ of the target,

$$\hat{c} = \arg\max_{c \in C} \iint p(z|c, \theta_v, |v|) p(\theta_v, |v|)\, d|v| d\theta_v. \tag{7.2}$$

This work focuses on obtaining an estimate of $p(|v|, \theta_v)$ from the tracker, and incorporating that into the classification rule. Using the Doppler and target strength features described in Chapter 3,

$$\hat{c} = \arg\max_{c \in C} \iint p(SNR, \delta|c, \theta_v, |v|) p(\theta_v, |v|)\, d|v| d\theta_v. \tag{7.3}$$

It is assumed that the error on the target strength (SNR) and Doppler (δ) measurements is independent:

$$\hat{c} = \arg\max_{c \in C} \iint p(SNR|c, \theta_v, |v|) p(\delta|c, \theta_v, |v|) p(\theta_v, |v|) d|v| d\theta_v. \tag{7.4}$$

Recalling that the target strength is only a function of aspect, θ_v:

$$\hat{c} = \arg\max_{c \in C} \int p(SNR|c, \theta_v) \int p(\delta|c, \theta_v, |v|) p(\theta_v, |v|)\, d|v| d\theta_v. \tag{7.5}$$

This maximum likelihood rule is the basis for the comparison of several methods for estimating $p(\theta_v, |v|)$ on the simulations described in Section 7.3.

7.3 Simulation Design

To evalate the performance of this type of classification, three datasets are simulated and the performance of the maximum likelihood classifier is compared.

7.3.1 Tracking Scenarios

In each scenario, a single target is present and "measurements" are taken of its bearing b, range d, and Doppler, δ every $\Delta t = 180s$. The measurements are corrupted with additive

Gaussian white noise with $\sigma_b = 4^\circ$, $\sigma_d = 600m$, $\sigma_\delta = 0.4m/s$. The mean for all the measurement noise is zero. The sonar equation in (5.1) is used to calculate an amplitude feature, which is corrupted with additive Gaussian white noise with $\sigma_{SNR} = 8dB$. Four clutter contacts are simulated at each time step, with the assumption that they are located inside the gate of the tracker; that is, they would be candidates for the track update at the current time step. The Doppler shift of the clutter is distributed normally with mean zero and $\sigma = 1.286m/s$. The target strength of the clutter is distributed normally with a mean of $6dB$ and $\sigma = 2dB$. The simulated Doppler shift and target strength are then corrupted with AWGN of the same statistics as the simulated target contacts.

The scenarios are designed to be relatively easy from a "tracking" viewpoint - the goal is to evaluate the usefulness of integrating the estimate of velocity from track state. To this end, the first scenario is shown in Figure 7.1. This scenario contains a single target which moves at a velocity of $5m/s$ at a constant heading. Because of the geometry, the target is either heading directly towards the receiver (the blue circle) or away from the receiver. This results in a high Doppler shift for most of the scenario, and a low target strength.

In the second scenario, the target takes a circular path around the receiver with a constant magnitude velocity, $|v| = 5m/s$. The aspect angle will be close to 90° for most of the scenario, so the Doppler shift will be low, and the target strength will be high. Figure 7.2 shows the second scenario.

The third scenario consists of a target which travels towards the receiver, and then makes a slow turn away from the receiver. In this scenario, the aspect angle varies greatly, so the Doppler and target strength will vary accross the scenario. Figure 7.3 shows the path taken in the third scenario.

7.3.2 Aspect Information

The derivation of the maximum likelihood classifier in Section 7.2.1 requires a prior $p(\theta_v, |v|)$. To obtain this prior, we run an extended Kalman filter (EKF) on the known true contacts. This was done to remove a source of potential error - the data association problem. The goal was to show how appropriately including uncertainty in the classification step can improve

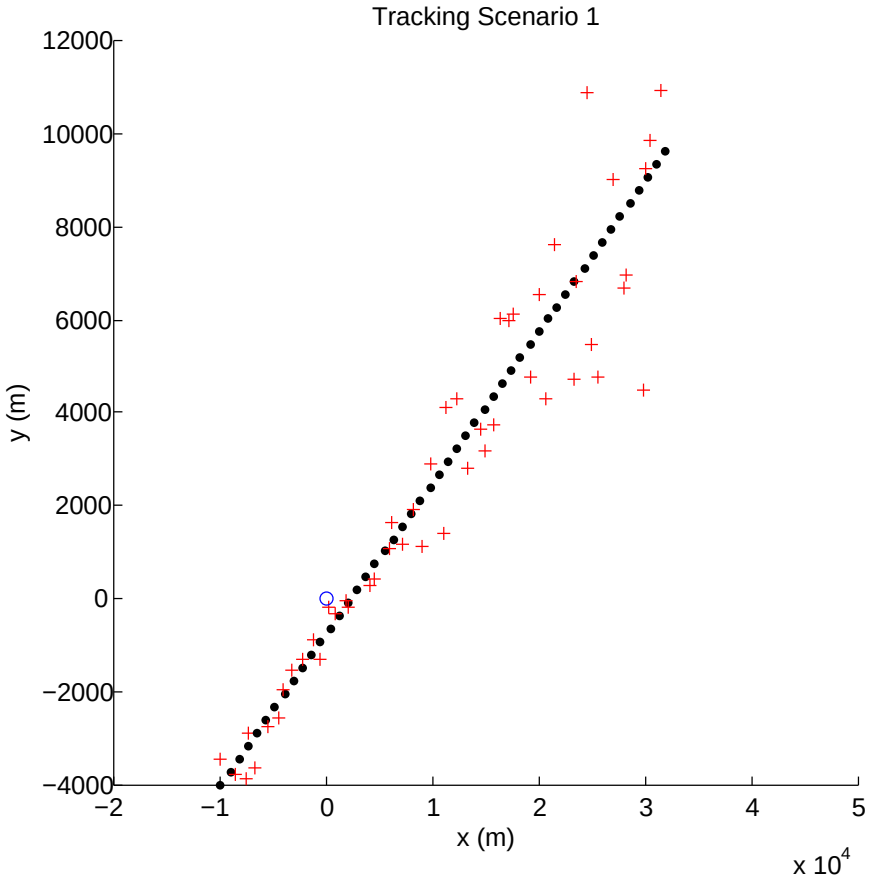

Figure 7.1: The first scenario. Black dots are the true location, red crosses are the measurements corrupted by noise, and the blue circle is the transmitter/receiver.

performance, not evaluate a data association problem. The EKF uses the nearly constant velocity model with state vector $X = [x, y, dx, dy]$ [54]. At each time step, the EKF is used to predict the velocity from the previous state. In Section 7.4, we compare the use of two estimates of the velocity: using only the mean value from the state vector and using the mean and covariance of the state vector. These methods are also compared to a uniform prior and an "oracle" which knows the exact true velocity.

Recalling that the velocity state of the EKF is in rectangular coordinates, this leads to an additional required step of transformation to polar coordinates. For a small set of dis-

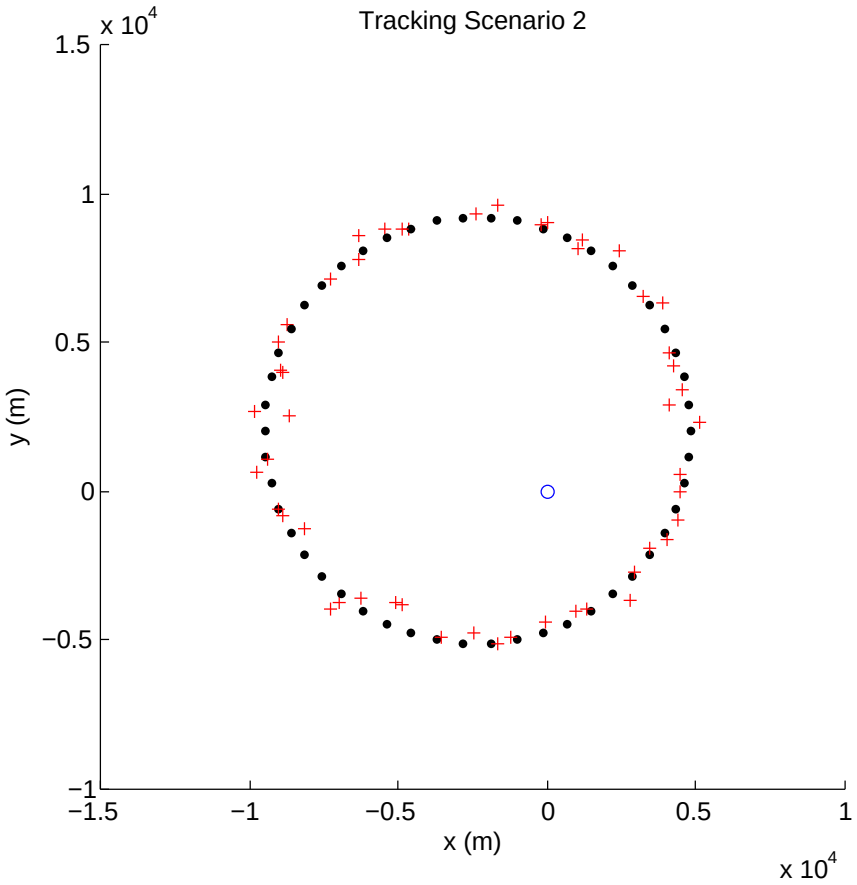

Figure 7.2: The second scenario. Black dots are the true location, red crosses are the measurements corrupted by noise, and the blue circle is the transmitter/receiver.

tributions, it is straightforward to map $p(dx, dy)$ to $p(|v|, \theta_v)$. For example, when $p(dx, dy)$ is jointly Gaussian with zero mean, uncorrelated, and have the same variance, $p(|v|)$ is Rayleigh and $p(\theta_v)$ is uniform over the interval $[0, 2\pi)$ [57]. However, in general, there is no simple mapping from $p(dx, dy)$ to $p(|v|, \theta_v)$. To overcome this, we sample $p(dx, dy)$ uniformly in $|v|$ and θ_v. This changes the integrals in (7.5) to discrete sums and allows for the evaluation numerically.

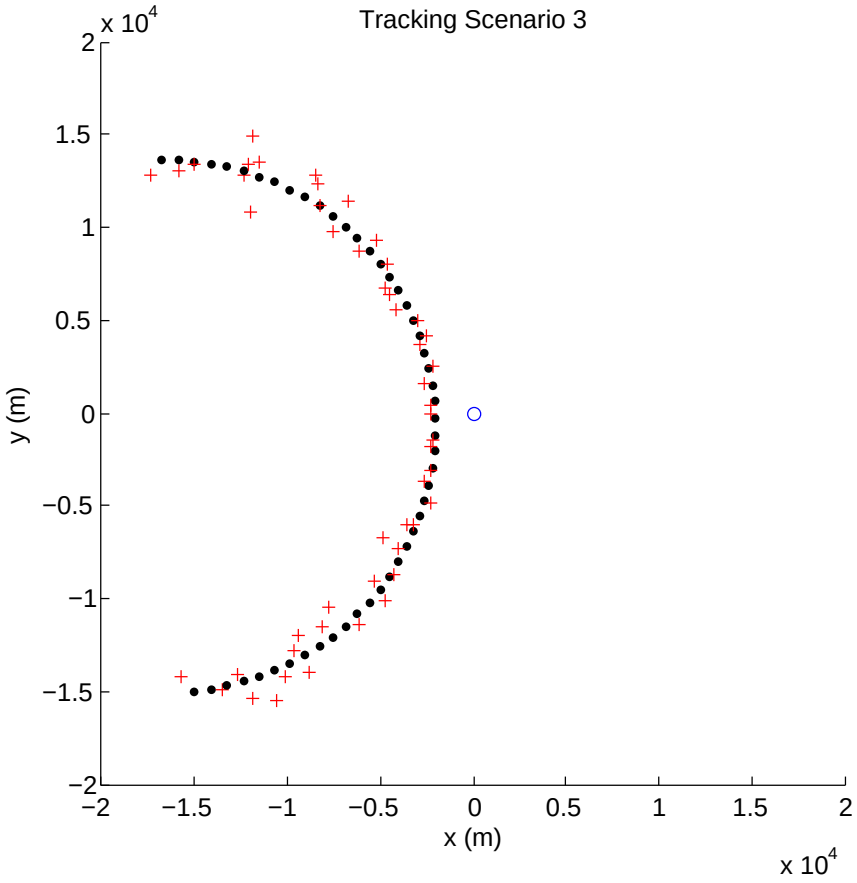

Figure 7.3: The third scenario. Black dots are the true location, red crosses are the measurements corrupted by noise, and the blue circle is the transmitter/receiver.

7.4 Results

Using the known clutter and target distributions from Section 7.3, the classification rule from Section 7.2.1 can be refined. For the clutter class c_0 there is no dependence on θ_v or $|v|$:

$$p(SNR, \delta|c_0) = \mathcal{N}(SNR; 6, \sigma^2_{SNR0})\mathcal{N}(\delta; 0, \sigma^2_{\delta 0}), \tag{7.6}$$

where $\sigma_{SNR0} = 6dB + 8dB$ and $\sigma_{\delta 0} = 2dB + 4dB$. In both of these cases, the first term is the error due to the distribution on the simulated mean of the feature, and the second term

is the error due to measurement noise. For the target class, c_1,

$$p(SNR, \delta|c_1) = \sum_{\theta_v} \sum_{|v|} \mathcal{N}(SNR; SNR_{true}(\theta_v), \sigma^2_{SNR1}) \mathcal{N}(\delta; |v|cos(\theta_v), \sigma^2_{\delta 1}) p(\theta_v, |v|), \quad (7.7)$$

where $SNR_{true}(\theta_v)$ is the target strength from the known target strength curve, $\sigma_{SNR1} = 8dB$, and $\sigma_{\delta 1} = 0.4m/s$. We compare the results for four different priors on $p(\theta_v, |v|)$: an *oracle* $p(\theta_v, |v|) = \delta(\theta_{true}, |v_{true}|)$, sampling from a *uniform* prior $\theta_v \sim U(0, 2\pi)$ and $|v| \sim U(0, 8)$, using only the *mean* from the EKF $p(\theta_v, |v|) = \delta(\theta_{est}, |v_{est}|)$, and sampling from the *Gaussian* which represents the EKF's velocity state.

Tables 7.1 to 7.3 show the accuracy of the classifier on the three scenarios using different priors and features. Results are averaged over 1000 Monte Carlo simulations for each scenario. For each class, the accuracy is defined as the ratio of number of contacts correctly classified to the total number of contacts. The accuracy reported in Tables 7.1-7.3 is the average of the accuracy of the clutter class and the target class. This is done to stop the large number of clutter contacts from distorting the accuracy figures.

Across scenarios, the oracle performs the best with almost all combinations of features. This is expected, and could be considered a rough maximum bound for performance - if your tracker could perfectly predict the target's aspect and velocity with no uncertainty, this would be the value that you use. Additionally, using the both the mean and covariance of the predicted velocity (labelled *Gaussian* in the tables) always performs better than simply using the mean. This is simply because the mean estimate is rarely going to be exactly correct. Figure 7.4 shows the velocity estimates from scenario one - the mean of the estimated velocity can differ significantly from the true value. In the other scenarios, this difference will be larger due to the constantly changing velocity.

The Doppler measurement is very sensitive to this error, because the standard deviation of the measurement error is only $0.4m/s$. By appropriately accounting for the uncertainty in the prediction, results can be improved. Lastly, using the *Gaussian* estimate from the tracker always results in a higher accuracy than using the uniform distribution across all three scenarios and feature sets.

An interesting result is that in the scenarios which contain more contacts with high magnitude Doppler measurements, the performance is significantly better. This is due to

the greater separation between the mean of the clutter Doppler distribution ($0m/s$) and the Doppler shift due to the target's motion. In scenarios one and three, there are long stretches of time where the target is travelling almost completely towards the transmitter, whereas in scenario two, the Doppler shift is much lower across the whole track.

	Oracle	Uniform	Mean	Gaussian
Target Strength	0.539	0.531	0.530	0.531
Doppler	0.987	0.558	0.453	0.986
Both	0.988	0.948	0.454	0.981

Table 7.1: Accuracy of the maximum likelihood classifier with different priors on Scenario 1. The accuracy reported the average of the accuracy on the target and clutter classes.

	Oracle	Uniform	Mean	Gaussian
Target Strength	0.618	0.600	0.597	0.596
Doppler	0.803	0.500	0.495	0.503
Both	0.826	0.521	0.521	0.682

Table 7.2: Accuracy of the maximum likelihood classifier with different priors on Scenario 2. The accuracy reported the average of the accuracy on the target and clutter classes.

7.5 Conclusions

The results presented here suggest that using an estimate of velocity from the track state is beneficial for contact classification. In the simulations, the estimate of the targets heading and velocity were used as prior information for a maximum likelihood-based classifier. Including uncertainty from the track state's covariance matrix also resulted in increased performance.

This research had a couple of very interesting results that helped to shape the work presented later in this dissertation. First, in Scenarios 1 and 3, classification a uniform prior

	Oracle	Uniform	Mean	Gaussian
Target Strength	0.536	0.533	0.526	0.526
Doppler	0.979	0.559	0.417	0.955
Both	0.980	0.942	0.414	0.961

Table 7.3: Accuracy of the maximum likelihood classifier with different priors on Scenario 3. The accuracy reported the average of the accuracy on the target and clutter classes.

and both target strength and amplitude features results in surprisingly good performance. This suggests that there can be a large amount of information in those two features - and that the two features must be consistent. This led to work on using bistatic Doppler for localization [13] and also for the comparison and clustering of contacts presented in Chapter 10.

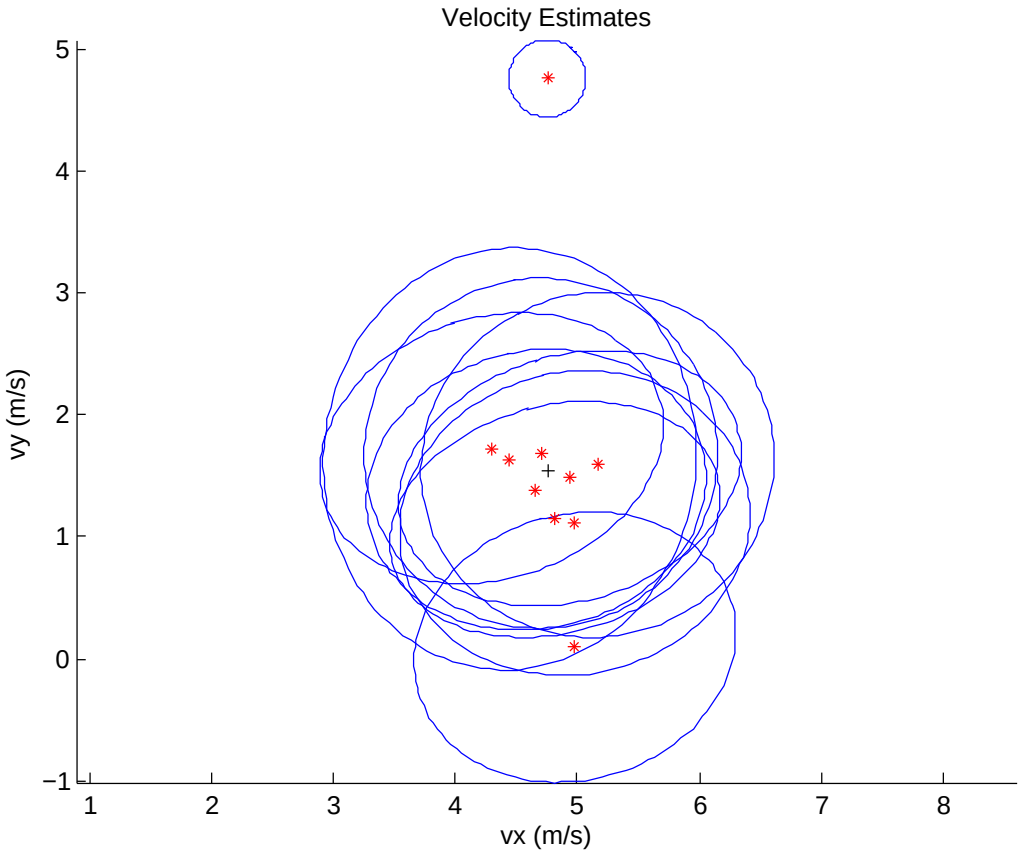

Figure 7.4: Velocity estimates from the first scenario. Red stars are the mean estimate, the black cross is the true value, and the blue ellipses are the 1-sigma uncertainty ellipses.

Chapter 8

TRACKING MULTISTATIC SONAR DATA

Members of the MSTWG have been developing tracking algorithms for many years and had developed tracking methods that performed well on many simulated and real datasets. In 2009, Metron released their dataset, which was a simulated dataset with 5 transmitters and 25 receivers (described in detail in Chapter 5). This dataset contained a much larger amount of data and promptly broke the JPDA-based tracker described in earlier chapters. The reasons for this are many-fold and include the poor scaling of JPDA to many contacts, the difficulty of the dataset (low P_d and high measurement error), and poor track management performance.

These issues were not unique to the tracker presented up to this point, much of the MSTWG's work presented at Fusion 2010 also had poor performance in one way or another [8, 58, 44]. This to the development and refinement of a flexible preprocessing technique that allows for the fusion of measurements from multiple receivers into a form that is amenable to JPDA-based tracking.

8.1 Properties of Multistatic Sonar

A multistatic sonar field consists of many transmitters and/or receivers deployed over an area of interest. This sort of sensor field is becoming more popular as the cost of sensors decreases, wireless communications become ubiquitous, and targets become better at evading detection.

Multistatic sonar fields have several advantages over a monostatic or bistatic setup. The field is less sensitive to a single sensor failing because detection regions from other sensors may overlap with the failed sensor. If there is a region that is of specific interest, more resources (either physical receivers or transmitter power) can be used to improve detection probability. Additionally, in areas where detection regions overlap, it is likely that multiple

returns from the target will be available. Due to the spatial diversity of the field, each of the receivers will see a different aspect of the target, allowing for improved classification and target tracking.

As with everything else, these advantages come with downsides as well. Assuming a false alarm rate that is independent across receivers, the clutter rate will be increased by a factor equal to the number of receivers. Steps must be taken to either make the tracker more efficient or reject clutter contacts before they are passed to the tracker. Measurements are taken at the receiver and are functions of the specific geometry and the receiver's measurement accuracy. This results in measurements that are difficult to compare across receivers and that have wildly different error characteristics when mapped to a common $[x, y]$ space.

Chapter 9

POSTERIOR DISTRIBUTION PREPROCESSING AND INCORPORATING FEATURE DATA

Tracking multistatic data can cause a multitude of problems for some standard trackers. The combination of the large number of receivers, large error on the measurements, and higher clutter rate made tracking performance very poor. To attempt to overcome these difficulties, a preprocessing technique was developed to appropriately combine contacts.

9.1 Motivation

In its raw form, the data from the Metron dataset was untrackable by the JPDA-based tracker presented earlier. An ideal solution would be able to reject clutter contacts and combine the target contacts to reduce the measurement error. The *posterior distribution preprocessing* step presented here attempts to do both these things: contacts that are due to noise (clutter) will not align with each other and cancel out - conversely, contacts that originate from a target will be consistent and align, resulting in a high probability. Here I present a preprocessing step that allows for the integration of features as well as combination of arbitrary measurements.

9.2 Posterior Distribution Preprocessing

We found in initial testing of the Metron and PACsim datasets that using the PDA, PDAFAI, and the PDAFAIwTS algorithms did not produce very good results. There were many false tracks, so many that even if the true targets were tracked, they were indistinguishable. This is due to the large number of false contacts and the low probability of detection. Recently, it has become common to apply a pre-tracking contact fusion step[59, 47, 8]. The contact fusion step decreases the amount of clutter presented to the association step, which results in decreased runtime and memory usage. In this work, we expand on the posterior

distribution preprocessing technique described below[8], with the addition of a term for feature information.

The posterior distribution over state $X = [x, y]$ given the j^{th} bearing measurement, b_{ij}, and j^{th} bistatic range measurement, d_{ij} for the i^{th} receivers can be written $P(\mathbf{x}|b_{ij}, d_{ij})$. The error statistics for the i^{th} receiver are used in the formulation. Let I_i be an indicator variable which indicates whether or not the true contact was actually detected.

$$P\left(\mathbf{x}|b_{ij}, d_{ij}, I_i = 1\right) = \left(2\pi\sigma_b\sigma_r\right)^{-1} \exp\left(\frac{-\left(B_i(x, y, r_i) - b_{ij}\right)^2}{2\sigma_b^2} + \frac{-\left(D_i(x, y, r_i, t) - d_{ij}\right)^2}{2\sigma_d^2}\right),$$

where $B_i(x, y, r_i)$ is a function which maps state $\mathbf{x} = [x, y]$ to bearing for i^{th} receiver, $D_i(x, y, r_i, t)$ is the bistatic range at $\mathbf{x}$ for the i^{th} receiver, σ_b^2 is the bearing variance, and σ_d^2 is the bistatic range variance.

For each receiver i, the posterior distribution given that the true contact was detected is calculated by summing over all j contacts for each receiver $(\bar{b}_i, \bar{d}_i)$, with normalizing constant γ,

$$P\left(\mathbf{x}|\bar{b}_i, \bar{d}_i, I_i = 1\right) = \gamma \sum_{j=1}^{n} P\left(\mathbf{x}|b_{ij}, d_{ij}, I_i = 1\right). \tag{9.1}$$

Each receiver's probability of detection surface $P_{di}(x, y)$ is the probability that a contact at $\mathbf{x}$ will be detected. The total probability of a contact being at $\mathbf{x}$ is calculated in Equation (9.2), where $P(\mathbf{x}|I_i = 0)$ is the probability that a contact is at $\mathbf{x}$ and not detected. As a starting value, choose a small constant for $P(\mathbf{x}|I_i = 0)$ (10^{-25}, for example). This value could be tweaked to be location dependent in a scenario where additional information is known (bathymetry, for example).

$$P\left(\mathbf{x}|\bar{b}_i, \bar{d}_i\right) = P\left(\mathbf{x}|\bar{b}_i, \bar{d}_i, I_i = 1\right) P_{di}\left(\mathbf{x}\right) + P(\mathbf{x}|I_i = 0)\left(1 - P_{di}\left(\mathbf{x}\right)\right). \tag{9.2}$$

The complete posterior distribution is calculated:

$$P\left(\mathbf{x}|\bar{b}_{1..m}, \bar{d}_{1..m}\right) = \prod_{i=1}^{m} P\left(\mathbf{x}|\bar{b}_i, \bar{d}_i\right).$$

The top 50 local maxima of the posterior distribution are then calculated and sent to the JPDA tracking algorithm. These maxima are simply found through exhaustive search.

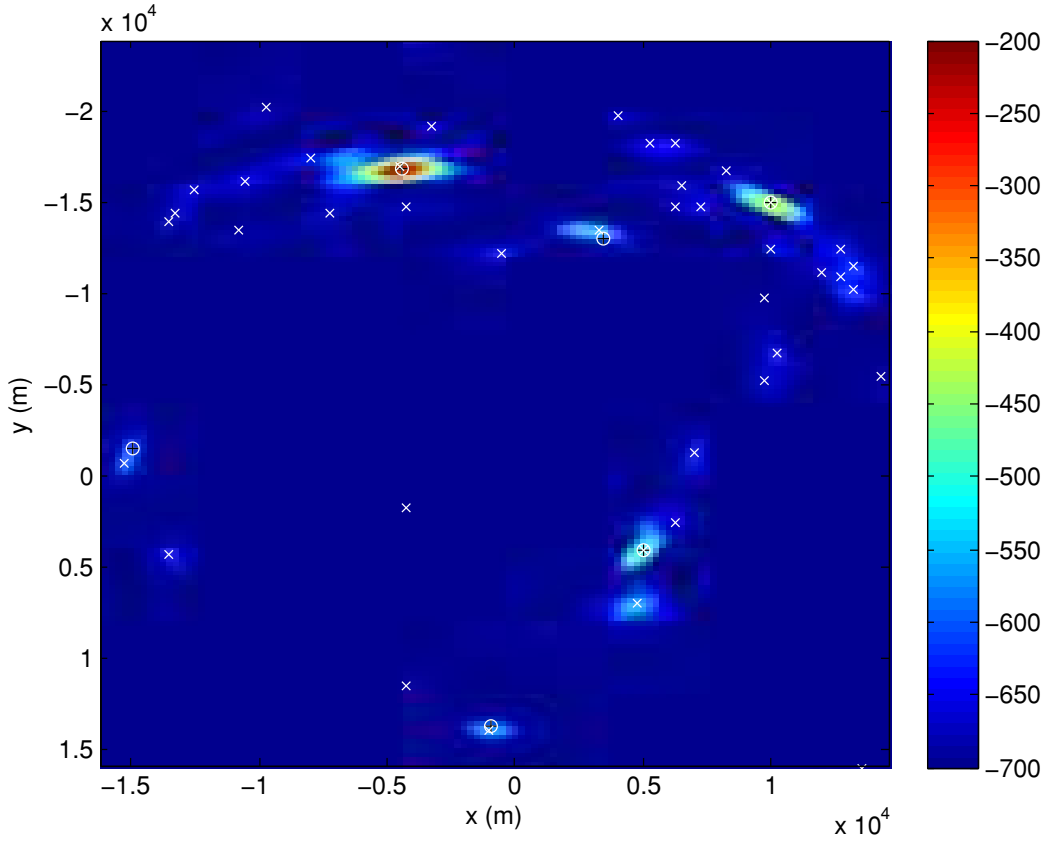

Figure 9.1: The log of the posterior distribution for ping #387 of PACSim Scenario B. True target locations are marked with black pluses inside white circles. The white 'x' symbols mark the top 50 maximum of the posterior distribution.

It is noted that there are other techniques for approximating the maxima of a distribution. The authors are currently exploring some of these techniques [60], including a Gaussian mixture approximation. As stated earlier, choosing to use 50 maxima is somewhat arbitrary, however when more maxima were chosen the tracking results were not significantly different and resulted in greatly increased computational requirements. When fewer than 50 were chosen the results were negatively affected some of the time. An example of the posterior distribution is shown in Figure 9.1. The log of the distribution is plotted for ease of viewing.

The 50 maxima are marked with black plus signs. Note that the posterior distribution is high near all the true target locations (black plus signs in white circles), and a local maximum of the surface is often close to the true target location.

9.3 Inclusion of Feature data

The posterior distribution preprocessing step described above focuses on the use of position measurements to combine contact data. In this work, we extend it to include any feature data, z_{ij}. For each contact, the class-conditional likelihood ratio, α_{ij} is calculated:

$$\alpha_{ij} = \frac{P(c_T|z_j)}{P(c_0|z_j)},$$

where c_T is the target class and c_0 is the clutter class.

The class conditional likelihood ratio is then used to scale the posterior distributions for each contact in equation (9.1):

$$\tilde{P}\left(\mathbf{x}|\bar{b}_i, \bar{r}_i, I_i = 1\right) = \gamma' \sum_{j=1}^{n} \alpha_{ij} P\left(\mathbf{x}|b_{ij}, r_{ij}\right),$$

where γ' normalizes the posterior distribution.

In cases where there are multiple classes of interest, the class conditional likelihood ratio is calculated for each of the possible classes and then either the maximum or mean of the values can be used. In Scenarios B and C, neither method was better than the other.

9.4 Results

9.4.1 Metron Data Set

The Metron Dataset was the motivation for this preprocessing technique, yet it still only performed marginally on it. The measurement errors were too large and the scan rate too low to yield any good tracking results. Some trackers, which work on a "batch" of data are able to track some targets in the Metron dataset [48]. Batch trackers have the advantage of looking at a long window of data and attempting to fit tracks to the data. Tracks are fit under the assumption that a target moves in a straight line over a long period of time. This allows for the detection of very difficult targets, but results in a high latency and large computation complexity.

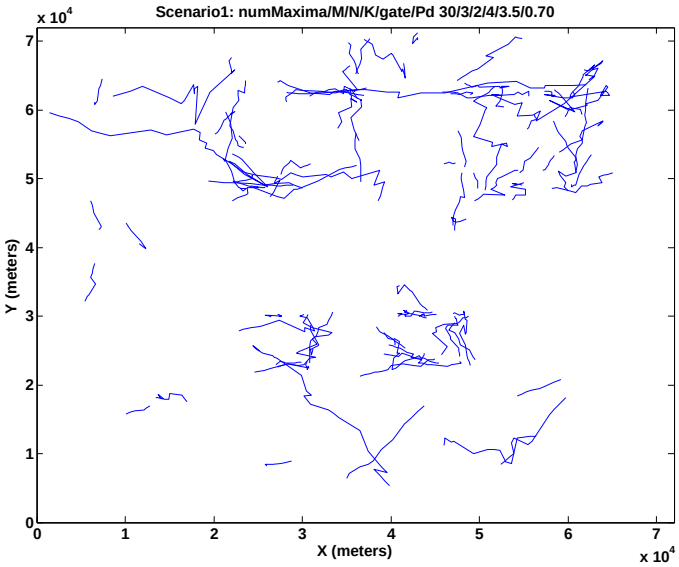

Figure 9.2: Tracking results for scenario 1 of the Metron dataset.

9.4.2 PACSim Data Set

The PACSim dataset is the latest dataset from the MSTWG and is unique because it contains a simulated feature measurement. This section discusses the results of integrating the feature into the preprocessing technique.

Blind Tracking Results

The truth for Scenario A was known *a priori* and was used to provide feedback on how well the tracking algorithms were performing on a baseline scenario and allow for tuning of tracking parameters. The truth for Scenarios B and C was provided after initial results were

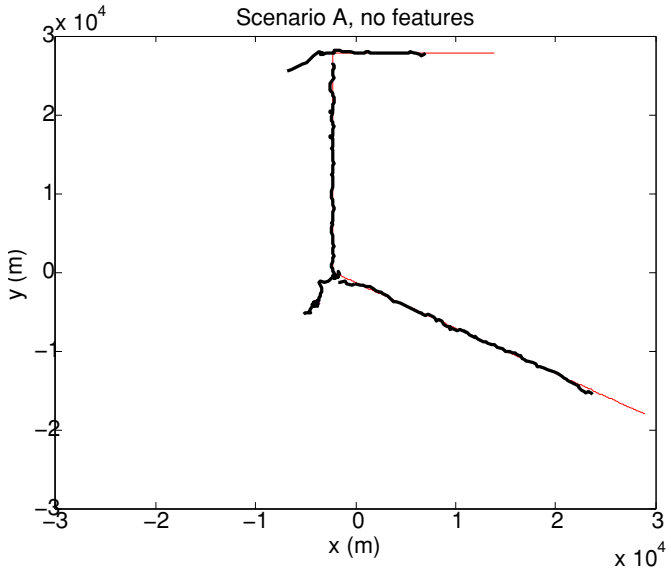

Figure 9.3: Tracking results for Scenario A of the PACsim data set without feature data. The black lines represent all the confirmed tracks for the entire duration of the simulation. The red line is the true target track.

presented, and was not used to tune any tracking parameters in the reporting of the results in this section. The parameters tuned were: the number of maxima found in the posterior distribution, the fineness of the grid at which the posterior distribution is estimated, the initial state and covariance estimates for the extended Kalman Filter, the process noise for the extended Kalman Filter, the track confirmation threshold and the track deletion threshold.

The MSTWG metrics calculated for Scenarios A, B, and C are based on the metrics in [61] developed for the MSTWG. The posterior distribution preprocessing step provides a different set of contacts than the originals that do not include the associated truth flags. For this reason, the only metrics reported are: track fragmentation (Frag), track probability of detection (TPD), track localization error (TLE) and track false alarm rate (TFAR).

Figures 9.3 (feature data is not used) and 9.4 (feature data is used) show the tracking results for Scenario A of the PACsim data set using the JPDA algorithm with the

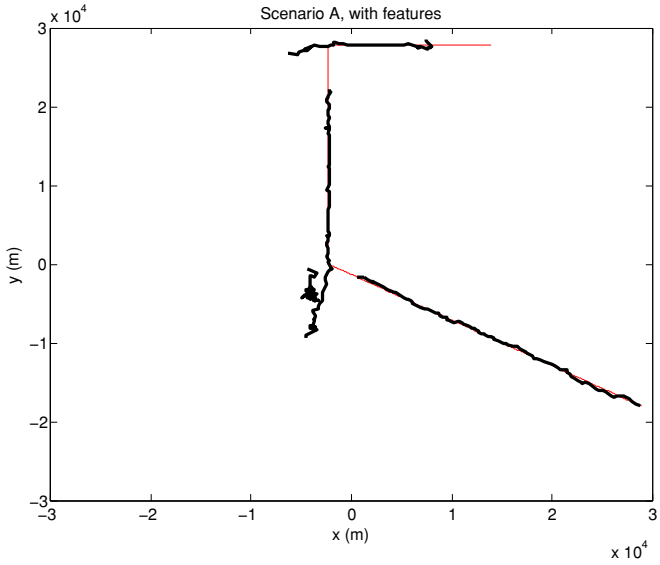

Figure 9.4: Tracking results for Scenario A of the PACsim data set with feature data. The black lines represent all the confirmed tracks for the entire duration of the simulation. The red line is the true target track.

pre-processing. These results were obtained using the following parameters: confirmation threshold = 15, deletion threshold = -3, deletion window = 5, gate = 3, EKF process noise scale = 4×10^{-4}. These parameters were tuned on Scenario A and used for both Scenario B and C. The target is tracked reasonably well although the confirmed track (red line) is not continuous through the corners. Features for target contacts are drawn one of two Gaussian distributions with mean of either 6 or 3 and standard deviation of 10. Features for clutter contacts are drawn from a Gaussian distribution with mean 0 and standard deviation of 10. The feature data does affect the tracking results, but for Scenario A it is difficult to make any conclusions.

Figures 9.5 through 9.8 are the tracking results for Scenarios B and C. Plots are included for simulations with and without feature data included in the tracking algorithms. The tracking results for Scenario B showed two targets and two fixed clutter points. Including the feature data improved the tracking results, which can be clearly seen in comparing

Figures 9.5 and Figures 9.6. The two targets that are tracked have fewer fragments and tracking probability of detections is improved. This will be more clearly seen in the following section where metrics are reported.

Scenario C proved very difficult to track using the parameters tuned on Scenario A. As expected, the fixed clutter points are appropriately tracked, however none of the other confirmed tracks are from a true target. This is due to the increased clutter rate in Scenario C as well as the much lower probability of detection for both the CW and FM pings, as included in Table 5.1.

Metrics

The metrics and plots presented for Scenario B are from simulation runs before the truth data was known and the tracking parameters were not changed. The metrics were calculated based on the methods described above. Table 9.1 shows the per-target metric results for Scenario A. Table 9.2 shows the metric results for the single targets in Scenario B. Scenario C is not included in this section due to there being no correct target tracks. Metrics for Scenario C will be presented for the *Optimized* tracking results in the next section.

For targets 1 and 2 in Scenario B, including feature data results in increased TPD and decreased Fragmentation. For target 2, the inclusion of feature data decreased the TLE. In this scenario, adding feature information is clearly helpful.

Target	1	1F	2	2F
TPD	0.757	0.738	0.986	0.944
TLE	211	212	132	137
Frag	4	4	2	2

Table 9.1: Metric results for the target and fixed clutter in Scenario A. “F” indicates results when features were included in the preprocessing step.

Target	1	1F	2	2F
TPD	0.812	0.830	0.536	0.701
TLE	235	253	508	383
Frag	4	2	4	3

Table 9.2: Metric results for targets in Scenario B. "F" indicates results when features were included in the preprocessing step.

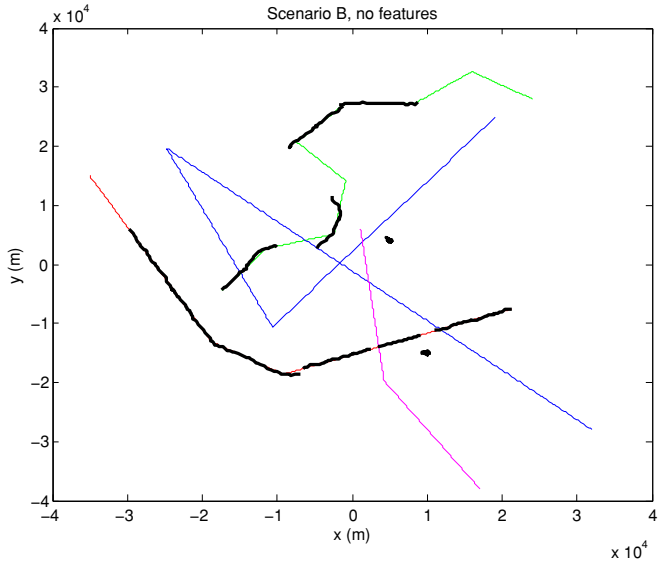

Figure 9.5: Tracking results for Scenario B of the PACsim data set without feature data. The black lines represent all the confirmed tracks for the entire duration of the simulation. The colored lines are the target paths. The results presented here are from a tracker that was only tuned on Scenario A.

Optimized Tracking Results

After the truth was revealed, a quick analysis of the data showed that the CW pings in Scenarios B and C were of limited use. Recall Table 5.1, a comparison of the probability of detection for all of the contacts broken down by ping type. The extremely low probability of detection for all the targets motivated the discarding of all the CW pings and tracking

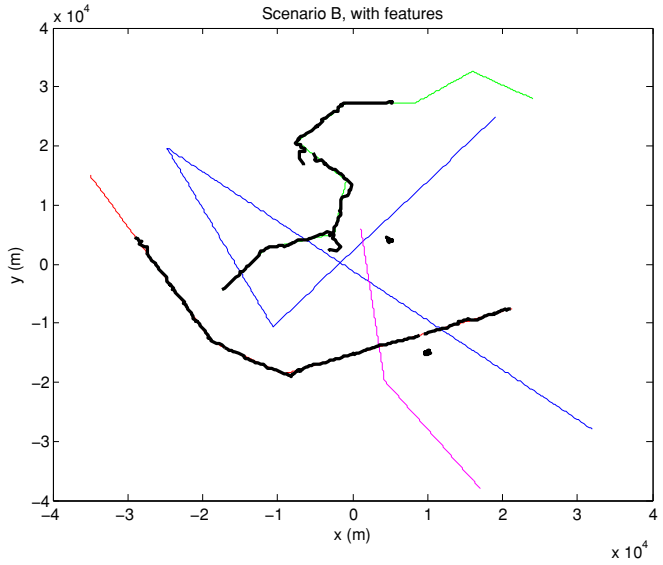

Figure 9.6: Tracking results for Scenario B of the PACsim data set with feature data. The black lines represent all the confirmed tracks for the entire duration of the simulation. The colored lines are the target paths. The results presented here are from a tracker that was only tuned on Scenario A.

only on the FM data. To further improve results on Scenario C, all contacts which had an SNR measurement of less than 25dB were discarded. Results showed a large improvement in tracking of targets 1 and 2 in Scenario B and target 1 in Scenario C. Figures 9.9 and 9.10 illustrate the difference in tracking results and Table 9.3 shows the metrics for the three tracked targets.

Target	B1	B2	C1
TPD	0.942	1	0.94
TLE	126	178	520
Frag	1	1	1

Table 9.3: Metric results for targets in Scenarios B and C after discarding CW pings.

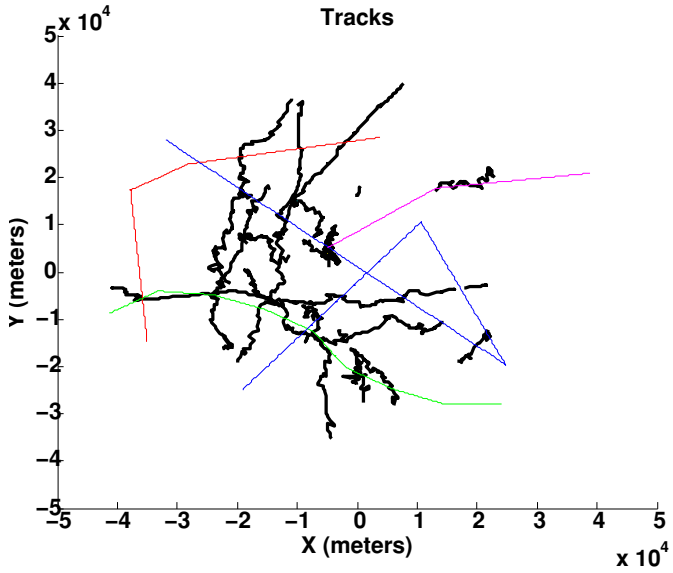

Figure 9.7: Tracking results for Scenario C of the PACsim data set without feature data. The black lines represent all the confirmed tracks for the entire duration of the simulation. The colored lines are the target paths. The results presented here are from a tracker that was only tuned on Scenario A.

9.4.3 Posterior Distribution Preprocessing for Data Fusion

In addition to being a useful approach for multistatic sonar data fusion, this approach also works for fusing different sensing modalities. It has been applied to combine electro-optical and infrared sensors to track drifting surface objects [12]. It has also been used to track objects using imaging sonar alone and also to fuse imaging sonar with HD video [10, 11].

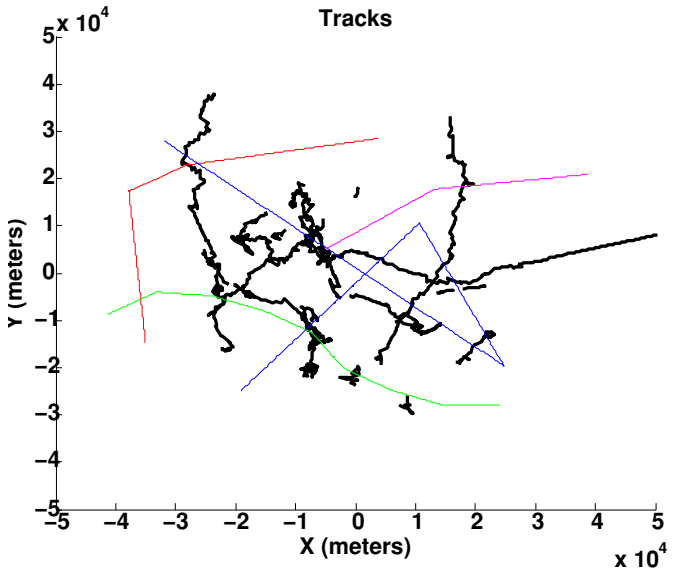

Figure 9.8: Tracking results for Scenario C of the PACsim data set with feature data. The black lines represent all the confirmed tracks for the entire duration of the simulation. The colored lines are the target paths. The results presented here are from a tracker that was only tuned on Scenario A.

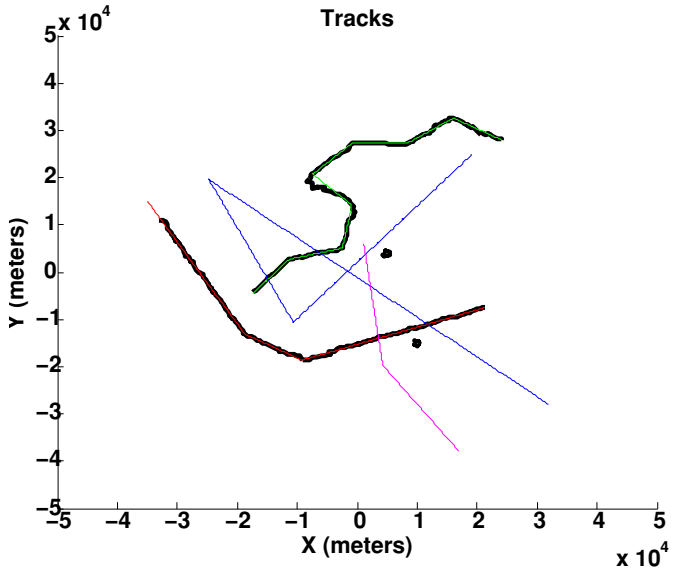

Figure 9.9: Tracking results for Scenario B after optimizing using the truth data. The black lines represent all the confirmed tracks for the entire duration of the simulation. The parameters for tracking were tuned on Scenario B.

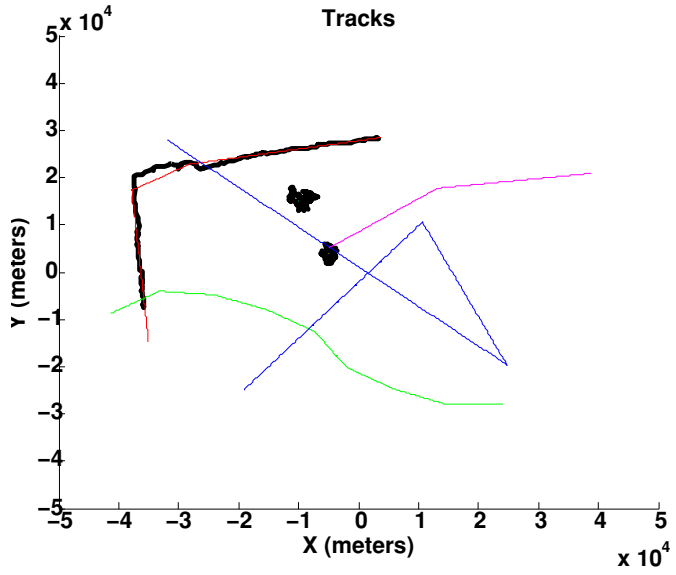

Figure 9.10: Tracking results for Scenario C after optimizing using the truth data. The black lines represent all the confirmed tracks for the entire duration of the simulation. The parameters for tracking were tuned on Scenario C.

Chapter 10

CLUSTERING CONTACTS USING LIKELIHOOD-BASED SIMILARITES

Combining contacts using the preprocessing technique performed well, but still left room for improvement. High probability tracks were able to be tracked, but the more difficult dim targets were still untrackable. If it were possible to know what contacts were from which target, it should be possible to improve accuracy from fusing contacts. To this end, a clustering approach is applied to the data.

In order to cluster contacts, most clustering algorithms require a measure of similarity between contacts. This similarity measure must be designed so that contacts that originate from the same target are very similar and are dissimilar from contacts which do not originate from that target. For clustering contacts, this means that we must appropriately account for the measurement error distributions.

Figure 3.6 shows the posterior distribution given a single measurement of bearing and bistatic range with the Metron dataset measurement error statistics ($\sigma_b = 8^\circ, \sigma_r = 600m$). This figure illustrates why using Euclidean distance between contacts does not work: a contact that is close in Euclidean distance but far in bistatic range can be much less similar than a contact that is far in Euclidean distance, but on the bistatic range ellipse. For this reason, we use the likelihood to define a similarity as detailed below. We published preliminary results with this approach [15, 14].

Given two measurements of position, z_1, z_2 from two receivers at positions, r_1, r_2, the posterior distribution is

$$P(\mathbf{x}|z_1, z_2, r_1, r_2, t),$$

where $\mathbf{x}$ is a position $[x, y]$ and t is the transmitter location. By Bayes' rule,

$$P(\mathbf{x}|\cdot) = \gamma^{-1} P(z_1, z_2|r_1, r_2, t, \mathbf{x}) P(\mathbf{x}|r_1, r_2, t),$$

where $P(z_1, z_2|r_1, r_2, t, \mathbf{x})$ is the *likelihood* of the measurements and γ is a normalizing

constant,

$$\gamma = P(z_1, z_2 | r_1, r_2, t).$$

Assuming the measurements from each receiver are independent,

$$P(\mathbf{x}|\cdot) = \gamma^{-1} P(z_1|r_1, t, \mathbf{x}) P(z_2|r_2, t, \mathbf{x}) P(\mathbf{x}|r_1, r_2, t). \tag{10.1}$$

In a bistatic sonar, there will be a blanking region for each receiver,

$$P(\mathbf{x}|r_1, r_2, t) = 0 \ \forall \mathbf{x} \ (D_1(x, y, r_1, t) < d_{blank} \ or \ D_2(x, y, r_2, t) < d_{blank})$$

where $D_i(x, y, r_i, t)$ is the mapping from location $\mathbf{x} = [x, y]$ to the bistatic range for receiver r_i and transmitter t. Assuming the measurements of bearing and range are independent,

$$P(z_i|r_i, t, \mathbf{x}) = P(b_i|B_i(x, y, r_i)) P(d_i|D_i(x, y, r_i, t)), \tag{10.2}$$

The similarity is defined here as the max over all possible contact states, $\mathbf{x} = [x, y]$

$$S(z_1, z_2) = \max_{\mathbf{x}} P(\mathbf{x}|z_1, z_2, r_1, r_2, tx). \tag{10.3}$$

This similarity formulation allows for appropriately modeling any additive error in the measurement ($[b, d]$). In the case that Doppler (δ) or amplitude (SNR) measurements are available, the state space X is extended to include velocity. The velocity is parametrized by magnitude and heading, $[|v|, \theta_v]$. The additional features are incorporated by modifying (10.2),

$$\begin{aligned} P(z_i|r_i, t, \mathbf{x}) \quad = \quad & P(b_i|B_i(x, y, r_i)) \times \\ & P(d_i|D_i(x, y, r_i, t)) \times \\ & P(\delta_i|\Delta_i(x, y, |v|, \theta_v, r_i, t)) \times \\ & P(SNR_i|\hat{SNR}_i(x, y, \theta_v, r_i, t)), \end{aligned}$$

where $\Delta_i(\cdot)$ is the Doppler shift of a contact with state $\mathbf{x}$ and $\hat{SNR}_i(\cdot)$ is the predicted SNR of a contact with state $\mathbf{x}$.

If other features are available, they can be incorporated in the same way.

10.1 Estimating Pairwise Similarity

It is pretty simple to see that two contacts are easily compared when detected by the same receiver, simply because their bistatic range and bearing measurements can be directly compared. However, calculating the similarity between two contacts detected by different receivers is made significantly more difficult by the lack of a mapping between measurements of bearing, bistatic range and bistatic Doppler for receivers at different locations. To overcome this difficulty, we compare contact similarity in the $\mathbf{x} = [x, y, |v|, \theta_v]$ state space.

In the low frequency active sonar domain, bistatic ranges often extend to hundreds of kilometers and this combined with potentially large errors on the bearing measurement necessitates comparing contacts that are quite distant in Euclidean space. As a result, a naive sampling-based approach requires a large number of samples to accurately estimate the similarity. If one wants to compare features that are dependent on target motion (amplitude or Doppler), the state space must be augmented to include target heading and/or target velocity. This augments the state space only by one or two dimensions, however this has a very large impact on the number of samples needed to accurately estimate the similarity. This increase in number of samples results in a much higher computational load, both in CPU time and memory requirements.

10.1.1 Sampling-Based Similarity Estimation

To overcome the accuracy vs speed tradeoff present in the naive grid sampled approach, I propose the following iterative algorithm for finding the maximum of the posterior distribution: $P(\mathbf{x}|\cdot) = P(\mathbf{x}|z_i, z_j, r_i, r_j, t)$.

The initial set of points to evaluate, Ψ_0, is chosen by sampling $N_p/4$ points from the measurement error distribution, $P(z_m|r_n, t)$ for each $m, n \in i, j$. If a velocity state is used, sample N_v points from a uniform distribution in heading ($[0, 2\pi)$) and speed($[0, v_{max}]$) for each point in Ψ_l. Select a tolerance ϵ and a number of iterations c_{max} after which to exit if the similarity hasn't increased. Initialize the counter c to zero.

1. While $(c < c_{max})$, select $\hat{\mathbf{x}}_l = \underset{\mathbf{x} \in \Psi_{l-1}}{\arg\max}\, P(\mathbf{x}|\cdot)$

2. If $(P(\hat{\mathbf{x}}_l|\cdot) - P(\hat{\mathbf{x}}_{l-1}|\cdot)) > \epsilon$, $c = 0$, $\hat{S}(z_i, z_j) = P(\hat{\mathbf{x}}_l|\cdot)$,

3. Else $c = c + 1$, $\hat{\mathbf{x}}_l = \hat{\mathbf{x}}_{l-l}$.

4. Select Ψ_l by sampling $N/2$ points from $P(h(\hat{\mathbf{x}}_{l-1})|r_n, t)$ for each $n \in i, j$, where $h(\mathbf{x})$ is a function which maps $\mathbf{x}$ to the measurement domain for r_n.

10.2 Clustering

The result of the pairwise similarity calculations is a similarity matrix that can be used by many different types of clustering algorithms. Different types of clustering algorithms perform best on different types of data and result in different types of clusters, but in here we focus on agglomerative clustering methods [62].

10.2.1 Agglomerative Clustering Methods

Agglomerative clustering begins with all N contacts a cluster of size one. At each step, the two clusters that are the most "similar" are combined into a single cluster. In section 10.1, we defined the likelihood-based similarity between two contacts, however, agglomerative clustering requires the calculation of similarity between clusters. For this work, the complete linkage metric will be used to calculate the similarity between clusters [63]. Complete linkage defines the similarity between clusters K_m and K_n as the similarity between the two **least** similar contacts in the clusters,

$$S_{CL}(K_i, K_j) = \min_{i \in K_m, j \in K_n} s_{ij}.$$

As a contrast, single linkage defines the similarity between clusters K_m and K_n as the similarity between the two **most** similar contacts in the clusters,

$$S_{SL}(K_m, K_n) = \max_{i \in K_m, j \in K_n} s_{ij},$$

where s_{ij} is the similarity between contact i of cluster K_m and contact j of cluster K_n.

Lastly, group average defines the similarity between clusters K_m and K_n as the average similarity between all contacts in the clusters,

$$S_{GA}(K_m, K_n) = \frac{1}{N_m N_n} \sum_{i \in K_m} \sum_{j \in K_n} s_{ij}.$$

Using the single linkage metric can often result in "chaining," the clustering of a set of measurements that are each only similar to a single other measurement in the group [64]. This often means that contacts in a cluster may be spread out over a large area and suggests that single linkage may not be the optimal choice of similarity between clusters.

The complete linkage metric suffers from the opposite effect: two clusters that are extremely similar except for a pair of contacts that are dissimilar will not be combined, resulting in smaller, localized clusters. Fortunately, this is a desired effect for clustering contacts. This will stop clutter contacts from being clustered along with target contacts.

One important consideration with agglomerative clustering is when to stop combining clusters. One method is to cluster to a certain number of clusters - this is not appropriate for clustering contacts because there is no way to know how many targets there are, or how clutter contacts will cluster. In this work, a threshold is used to determine when clustering should stop. If all clusters are less similar to each other than the threshold, clustering is terminated. To chose this threshold, similarities were calculated using PACSim Scenario A, and then a similarity threshold that was less than 99.5% of all target similarities was chosen. This threshold varies depending on waveform type and what additional features were used for clustering.

10.3 Cluster Fusion

After clustering, each cluster will contain a set of contacts, each with varying error statistics. We propose that the contact measurements be fused into a single measurement. This fusion step has several benefits: the combination of several measurements allows for improved localization, the number of measurements presented to the tracker is decreased, and it is possible to estimate speed and/or heading if Doppler and amplitude measurements are available.

10.3.1 Cartesian Coordinate Fusion

A simple method of fusing contacts in a cluster is to simply take the average of the measurements mapped to the X state space:

$$\hat{X}_l = N_l^{-1} \sum_{j=1}^{N_l} g_i(z_j),$$

where N_l is the number of contacts in cluster l, $g_i(z_j)$ is the j^{th} contact mapped to the X state space. The variance of the points in the cluster is used as the measurement error.

This method has the advantage of being very simple and quick to calculate, however it does so by ignoring the measurement error distributions.

10.3.2 Maximum Likelihood Fusion

For cluster fusion, we propose the use of a maximum likelihood approach that closely follows the likelihood-based similarity approach. Given a cluster of size N_i, we extend (10.1) to the following:

$$P(\mathbf{x}|\bar{z}, \bar{r}, t) = \gamma'^{-1} \prod_{j=1}^{N_l} \left[P(z_j|\mathbf{x}, r_j, t) + \epsilon\right] P(\mathbf{x}|\bar{r}, t), \tag{10.4}$$

where $\bar{z}$ and $\bar{r}$ are all the measurements in the cluster and the receivers that detected them. The ϵ term is required to stop an incorrectly clustered point from zeroing out the posterior distribution everywhere. The cluster is either fit with a single Gaussian or simply replaced with the maximum likelihood point:

$$\hat{X} = \arg\max_X P(\mathbf{x}|\bar{z}, \bar{r}, t).$$

This method has the advantage over the simpler Euclidean mean fusion in that it lets you get an estimate of heading if you are using any aspect-dependent features. If the measurements include Doppler shift, you can also obtain an estimate of speed.

10.3.3 Comparison of Fusion Methods

The performance of the two different fusion methods using different features is evaluated on the Metron and PACSim datasets. To evaluate the fusion methods, the labeled true

contacts are combined using each method, then the Euclidean distance between the fused estimate and the true contact position is measured. This is averaged over the scenario for each target. When Doppler and/or amplitude are used, an estimate of heading and speed are available. For these, the average error is reported.

Figure 10.1 shows the results of fusing the measurements for the Metron dataset. The likelihood-based fusion is better than the Euclidean mean fusion for all targets in all scenarios. There is not much difference between FM and CW pings for this dataset.

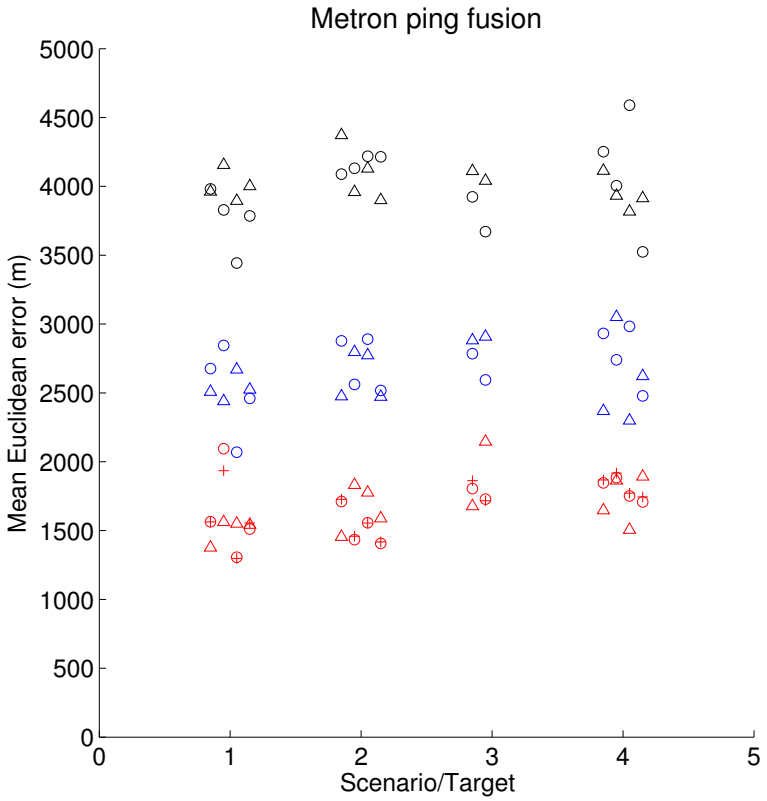

Figure 10.1: Mean Euclidean error for Metron dataset Scenarios. The black circles and triangles are the unfused data (CW and FM pings, respectively). The blue circles and triangles are the results of taking the mean of the measurements mapped to the Cartesian space. The red circles, triangles, and plus symbols are the results of the maximum likelihood fusion technique. The plus symbols are the CW pings, using Doppler information in the fusion step.

Figures 10.2 and 10.3 show the results of fusing the CW and FM pings of the PACSim dataset, using different feature sets. For all targets except two in the CW pings, the maximum likelihood fusion technique outperforms the mean in the Cartesian space. One very interesting effect of the fusion step is that even though the unfused error is not very different between the CW and FM pings, the error of the fused contacts is much lower for the FM pings than the CW pings.

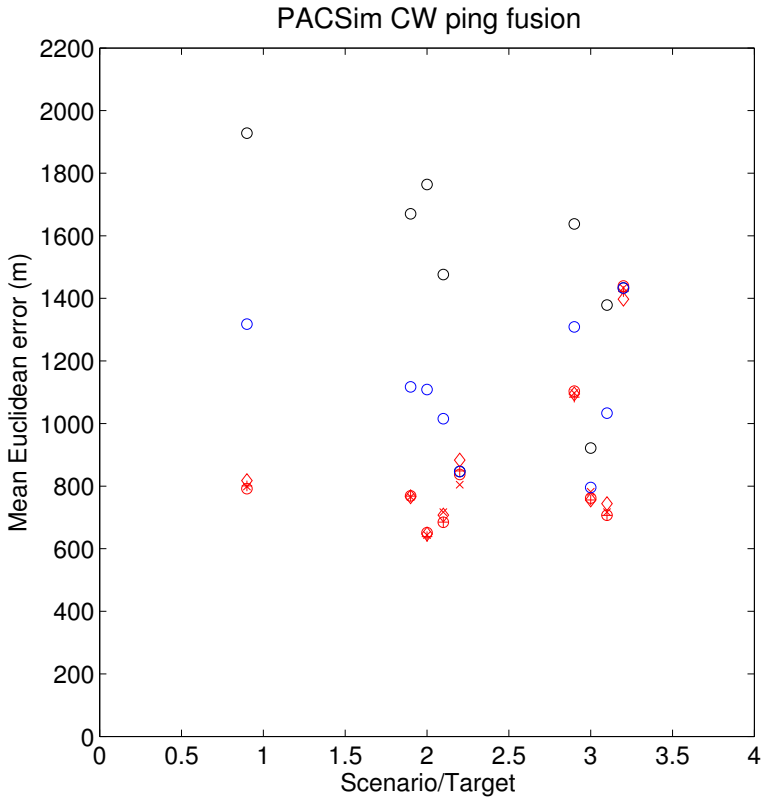

Figure 10.2: Mean Euclidean error for CW pings of the PACSim dataset Scenarios. The black circles are the unfused data. The blue circles are the results of taking the mean of the measurements mapped to the Cartesian space. The red symbols are the results of the maximum likelihood fusion technique. The red circles are results using only position, the plus symbols are results using Doppler and position, the 'x' symbols are results using amplitude and position, and the diamonds are results using position, Doppler, and amplitude.

For pings that have Doppler and/or amplitude features, the maximum likelihood fusion

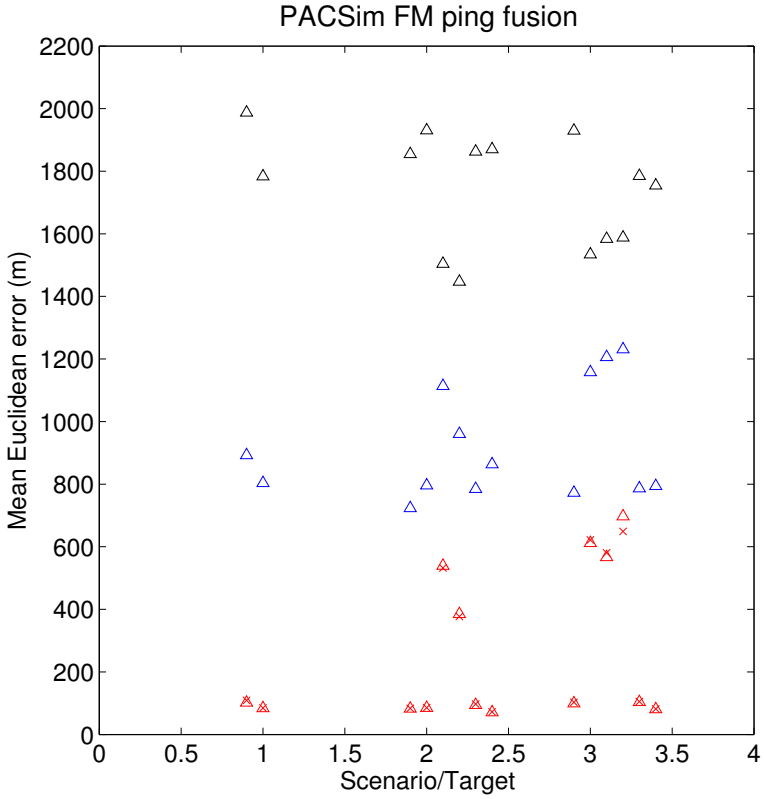

Figure 10.3: Mean Euclidean error for FM pings of the PACSim dataset Scenarios. The triangles are the unfused data. The blue triangles are the results of taking the mean of the measurements mapped to the Cartesian space. The red triangles and 'x' symbols are the results of the maximum likelihood fusion technique. The triangles are only using position and the 'x' symbols use amplitude and position in the fusion step.

step can also produce an estimate of heading and speed (if Doppler is available). Table 10.1 presents the estimation results for the Metron dataset. Estimation error is also presented for the PACsim dataset in Figure 10.4.

10.3.4 Results of Contact Fusion

In all cases except two targets in the Metron dataset, the likelihood-based fusion outperformed the simple mean in Cartesian coordinates. Inclusion of features had a small impact on the position estimate error, however including both Doppler and amplitude in the fusion

Target	1	2	3	4
Speed (m/s)				
1	0.727	0.203	-0.069	-0.230
2	0.025	-0.381	-0.321	-0.193
3	-0.054	-0.524		
4	-0.237	-0.567	-0.432	-0.629
Heading (deg)				
1	-3.87	-0.33	-1.74	-4.44
2	1.72	2.68	-0.64	-7.74
3	-1.78	1.78		
4	-5.15	3.73	16.10	1.96

Table 10.1: Heading and speed average estimate error for the Metron dataset.

steps allows for estimating the object's velocity as well. In the PACSim dataset, FM pings have a much lower bistatic range error, and fusing FM estimates results in greatly reduced position error. In addition, FM pings have a much higher P_d for the targets, resulting in more measurements being fused. In the Metron dataset, CW and FM pings are very similar, but even after fusion have errors of more than $1km$. This makes tracking even the fused estimates extremely difficult.

10.4 Cluster Classification

After contacts are clustered and fused, further processing can be done to classify clusters of contacts. The main goal of the classification step is to classify a cluster as containing target or clutter contacts. The ability to reject clutter clusters would improve tracking performance by reducing the number of possible false associations, as well as reducing the total association hypotheses. The latter is especially critical in JPDA and MHT-style trackers, where the number of possible associations has a large impact on speed [54].

If more sophisticated classification can classify a cluster as coming from a certain type of

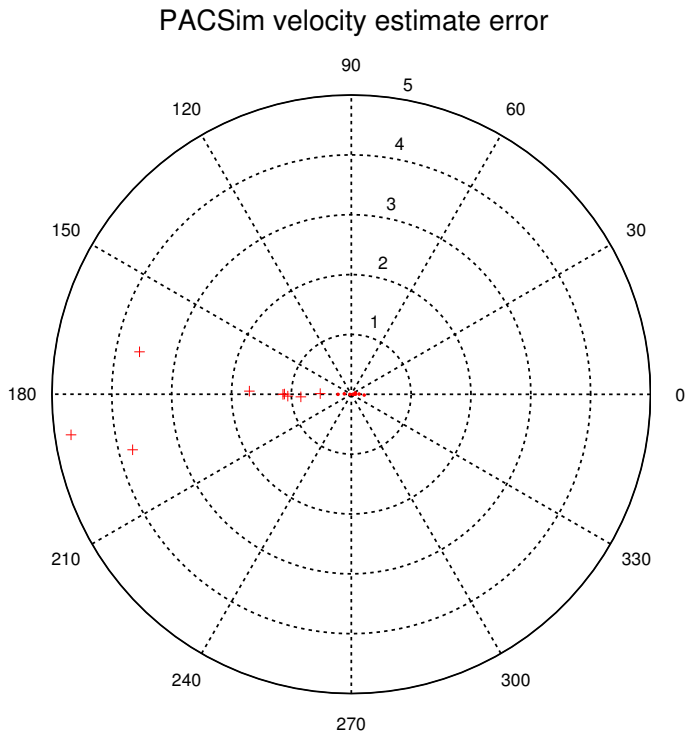

Figure 10.4: Velocity estimation error for the PACSim dataset. The red plus symbols are the average estimation error using only Doppler, the red dots are the estimation error using Doppler and amplitude.

target, this information can be used in tracking as well. The classification score can be used in the track state, allowing for improved tracking of close targets of different types [65]. It can also be incorporated in the track score, which can result in improved track management [6, 54]. Additionally, it can be used to select a motion model which is more appropriate to the target type.

10.4.1 Classification via Contact Features

The individual contacts contained in each cluster often have features associated with them (amplitude, Doppler, etc) that can be used to classify the cluster. Recall that the contacts in the cluster are each observed by a different receiver, resulting in a different aspect for each contact. Aspect-dependent features of the contacts in the cluster can be compared across

aspect to ensure they are consistent with the aspect-dependent nature of a target of interest. A prime example of this would be if one of the contacts has a very large amplitude due to a specular reflection, any other measurements of amplitude must be aligned with the target strength curve. If contacts are not correctly clustered, there would be little agreement with the target strength curve.

For non-aspect dependent features, there is a benefit as well. Assuming that a contact feature is normally distributed with standard deviation σ_f, averaging that feature for all of the contacts of a cluster of size N will result in a new feature that is normally distributed with standard deviation $\frac{\sigma_f}{\sqrt{N}}$. In the case of the PACsim dataset, there are three contact classes (clutter, target type 1, target type 2) and a class-dependent feature that is drawn from one of three Gaussian distributions. The separation between each of the Gaussians is small relative to the standard deviations ($\sigma_f = 10$ and each of the means is separated by 3). Classifying using only the original feature will result in a high Bayes error rate (0.44 when $\Delta\mu = 3$ and 0.38 when $\Delta\mu = 6$), but by averaging the features, classification can be improved.

10.4.2 Classification via Cluster Features

Features of the cluster itself can also be used to classify the cluster. A simple rule is to threshold clusters on the number of contacts that they contain. The motivation behind this rule is that targets will be observed by multiple receivers while clutter contacts will be distributed uniformly in each receiver's measurement space. Larger clusters will be due to multiple measurements of a target. This rule has the downside that it will discard contacts that are due to a low-P_d target. This is a large problem in Scenarios B and C of the PACSim dataset, where there are several contacts with $P_d \approx 0.1$. With 16 receivers, the expected number contacts from the target is only ≈ 1.6. Rejecting clusters of even size 2 and smaller would discard most of the contacts from these targets.

An improved classification rule for clusters is to use the maximum value from the fused cluster's posterior distribution (10.4). The posterior distribution is a product of multiple distributions and the maximum possible value will decrease with the number of contacts in

the cluster. To scale the distributions to the same range, for a cluster of size N, we use the N^{th} root of the maximum of (10.4) as a feature.

10.4.3 Feature Distributions

Distributions of the proposed features are presented here for Scenario 1 of the Metron dataset and Scenario C of the PACSim dataset. Scenario 1 was chosen for the Metron dataset because it was presented as a truthed scenario. This allowed for the calculation of histograms for target and clutter clusters separately. For the PACSim dataset, Scenario C was chosen because it was the most difficult scenario. It has several targets with extremely low P_d as well as a much higher clutter rate.

Results are separated by ping type (CW or FM) and cluster type (clutter or target). Target clusters are defined as clusters that have at least one true target contact in them. In the case of the metron dataset, there is not a large difference between FM and CW pings, but they are separated due to the slight difference in P_d for the two types of pings. For CW pings, results are also presented for similarity calculation and clustering using Doppler. For the Metron dataset, target type is not known, so the amplitude (and therefore target strength) feature is not used in clustering.

For the PACSim dataset, FM pings have a much lower bistatic range measurement error than CW pings ($\sigma_d = 150m$, rather than $600m$). In addition, the TS curve is known for all targets, so it is used in both FM and CW pings.

Metron Scenario 1

Figures 10.5-10.11 show histograms of several features calculated for Scenario 1 of the Metron dataset. As predicted, thresholding cluster size is a useful step for classification. Rejecting clusters with a size of one removes a large percent of clutter clusters (80.5%), but maintains most of the target clusters (98.6%).

The next feature presented is the N^{th} root of the maximum from the cluster fusion step. When only position is used, there is a very large spike at the high end of the normalized maximum histogram for the clutter clusters. This is due to the large number of single-

contact clutter clusters. Rejecting contacts in the maximum bin would have similar effects to rejecting the single-contact clusters, but this feature is not discrete, which can be better suited for certain types of classifiers.

The final feature presented for the metron dataset is the magnitude of the velocity at the maximum of (10.4). This feature is only available for CW pings when the fusion step uses the Doppler measurements. The histograms are very similar - the most obvious difference is a "bump" in the histogram near 6 m/s.

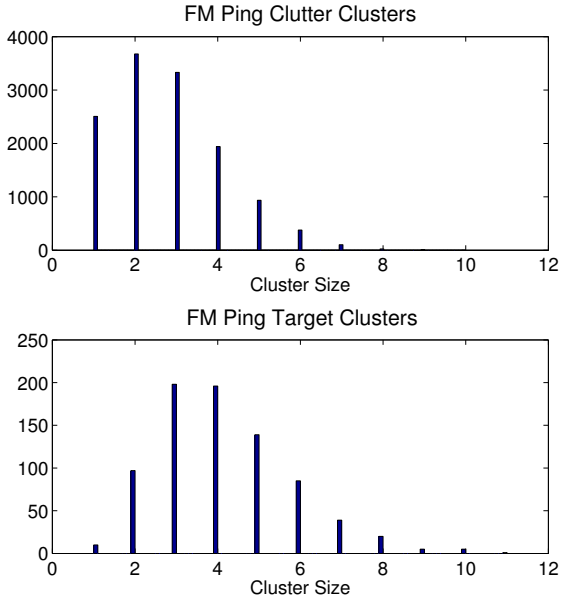

Figure 10.5: Histogram of cluster size for FM pings of Metron scenario 1. Similarities are calculated using position measurements only. Target clusters are defined as clusters that have at least one true target contact in them.

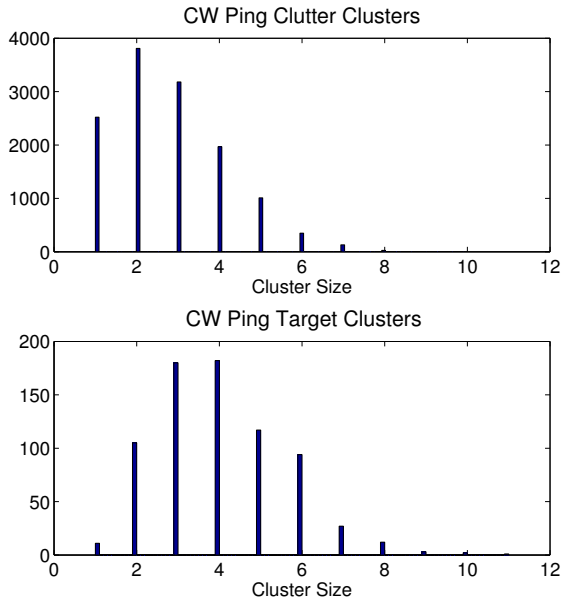

Figure 10.6: Histogram of cluster size for CW pings of Metron scenario 1. Similarities are calculated using position measurements only.

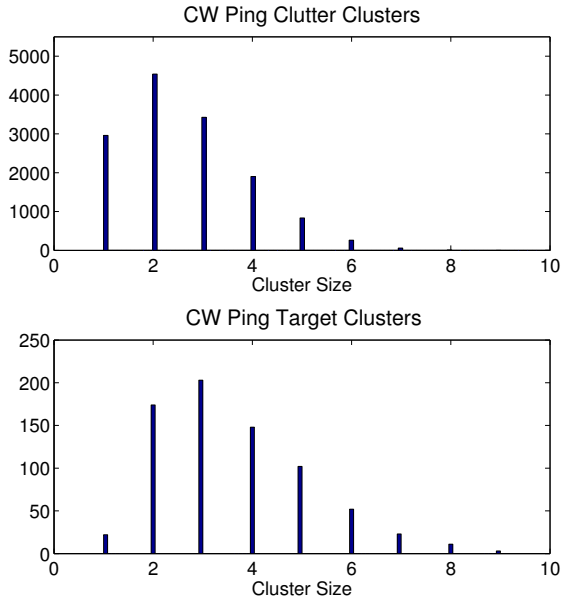

Figure 10.7: Histogram of cluster size for CW pings of Metron scenario 1. Similarities are calculated using position and Doppler measurements.

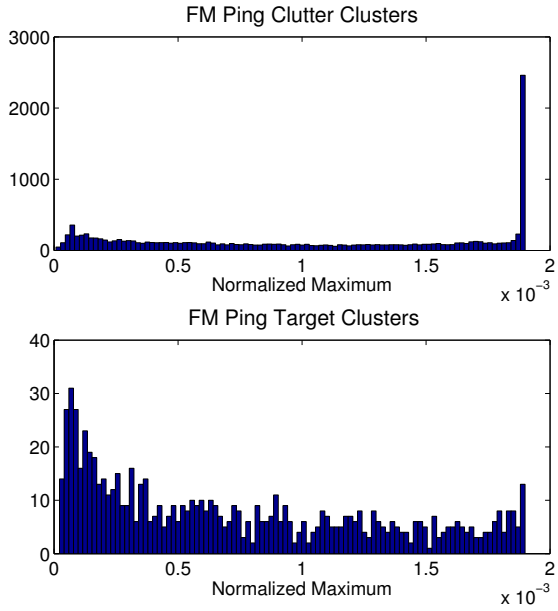

Figure 10.8: Normalized maximum of fused cluster for FM pings of Metron Scenario1. Similarities are calculated using position measurements only.

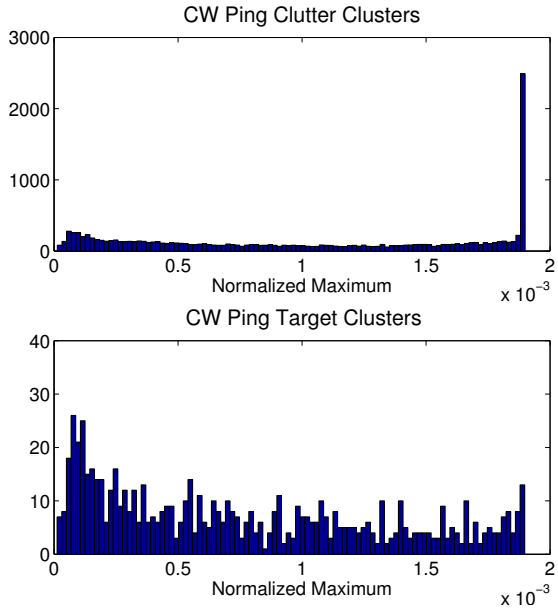

Figure 10.9: Normalized maximum of fused cluster for CW pings of Metron Scenario1. Similarities are calculated using position measurements only.

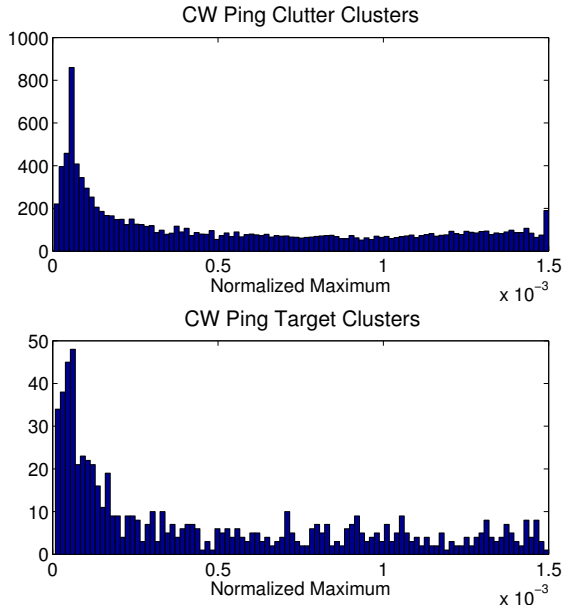

Figure 10.10: Normalized maximum of fused cluster for CW pings of Metron Scenario1. Similarities are calculated using position and Doppler measurements.

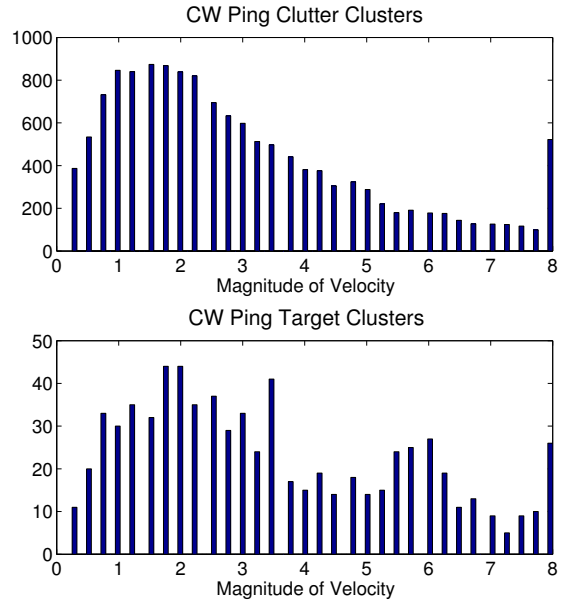

Figure 10.11: Histogram of velocity estimate for CW pings of Metron scenario 1. Similarities are calculated using position and Doppler measurements.

PACSim Scenario C

The PACSim dataset has a single target strength model, so it is usable to calculate similarities and fuse contacts. It turns out to be very useful in both of these steps. It makes the velocity feature in figure 10.31 much more discriminative between target and clutter. In addition, The PACSim dataset includes a simulated feature score to help identify contacts as target or clutter. Unfortunately, it has a high standard deviation (10) relative to the class separation (3). The feature score is averaged for each cluster, which results in a new feature that has a distribution with a standard deviation of $\frac{\sigma_f}{\sqrt{N}}$ where N is the size of the cluster. This will make it a more discriminative feature.

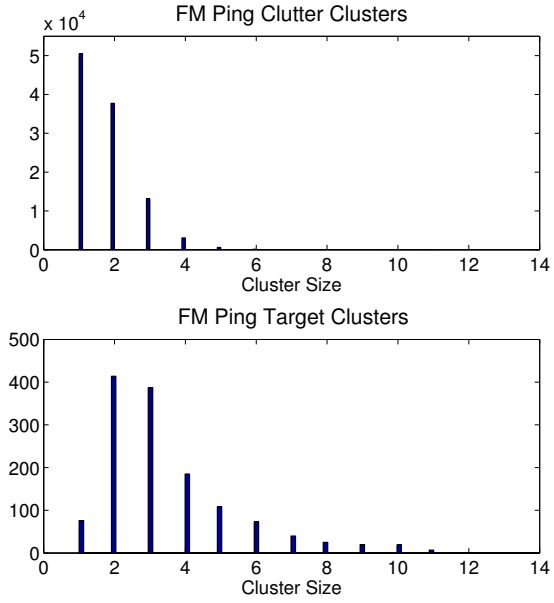

Figure 10.12: Histogram of cluster size for FM pings of PACSim scenario C. Similarities are calculated using position measurements only.

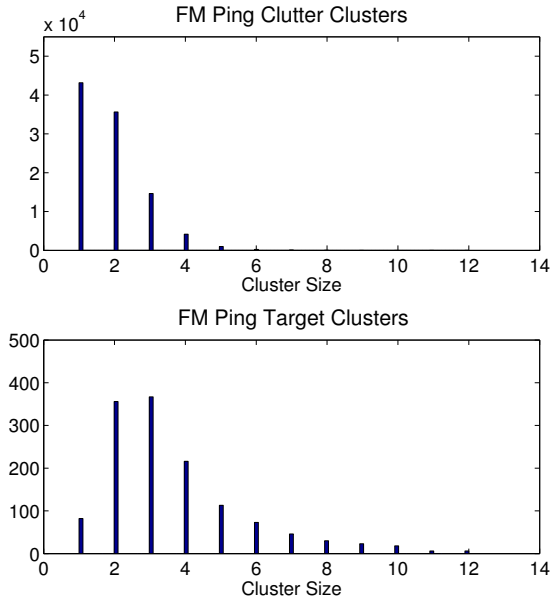

Figure 10.13: Histogram of cluster size for FM pings of PACSim scenario C. Similarities are calculated using position and amplitude measurements.

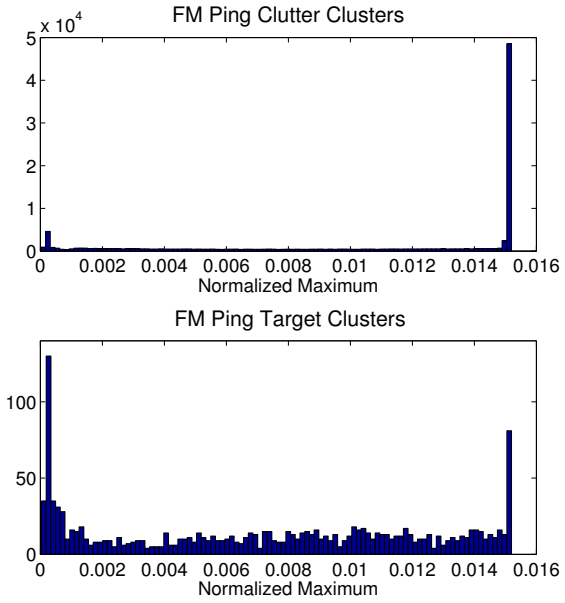

Figure 10.14: Histogram of normalized maximum from fusion for FM pings of PACSim scenario C. Similarities are calculated using position only.

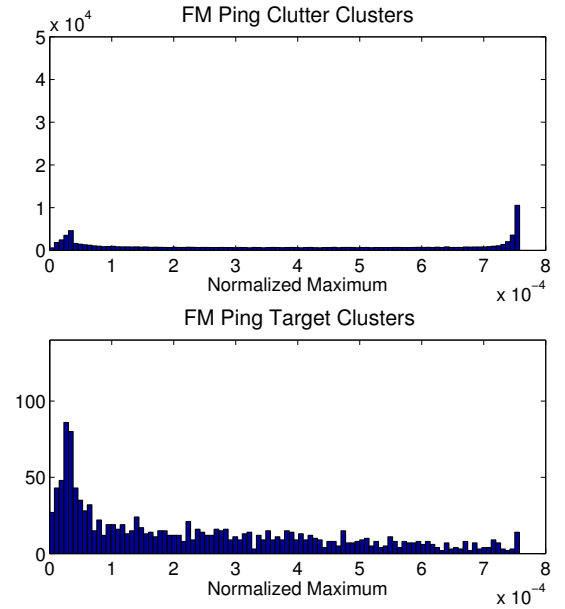

Figure 10.15: Histogram of normalized maximum from fusion for FM pings of PACSim scenario C. Similarities are calculated using position and amplitude measurements.

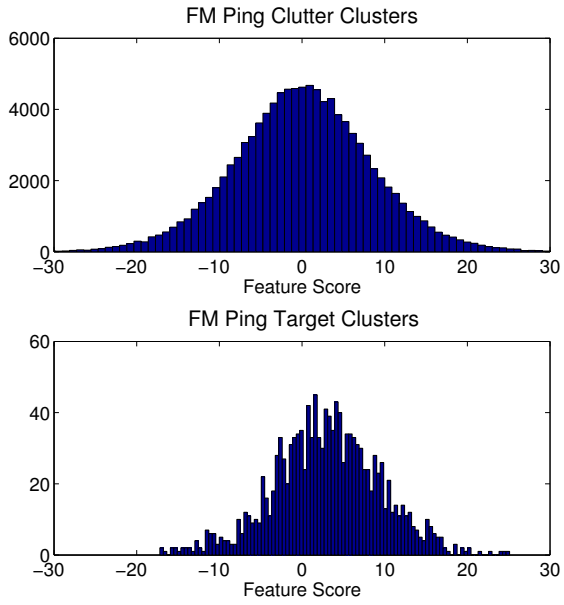

Figure 10.16: Histogram of averaged feature score for FM pings of PACSim scenario C. Similarities are calculated using position only.

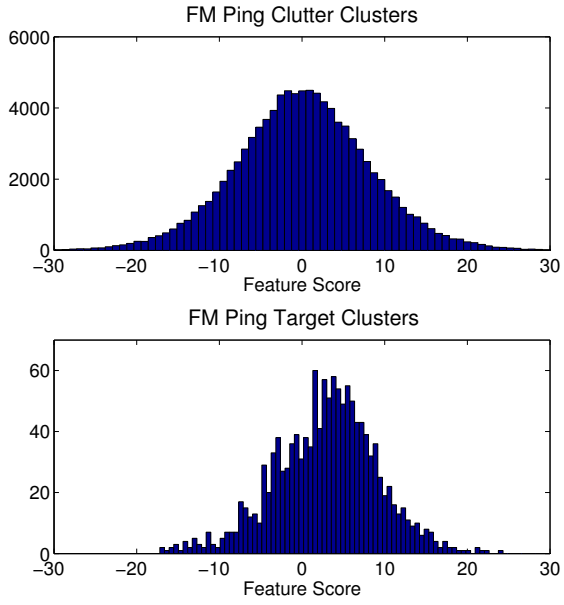

Figure 10.17: Histogram of averaged feature score for FM pings of PACSim scenario C. Similarities are calculated using position and amplitude measurements.

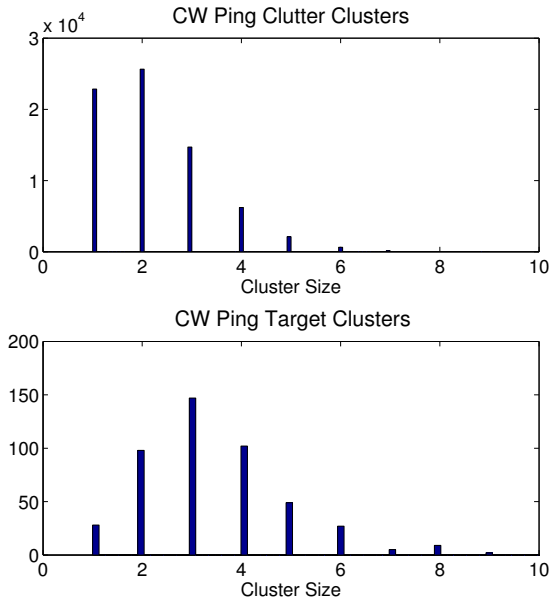

Figure 10.18: Histogram of cluster size for CW pings of PACSim scenario C. Similarities are calculated using position measurements only.

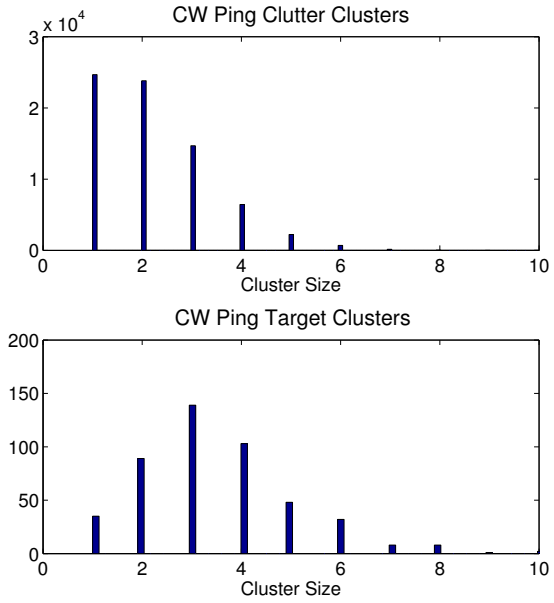

Figure 10.19: Histogram of cluster size for CW pings of PACSim scenario C. Similarities are calculated using position and amplitude measurements.

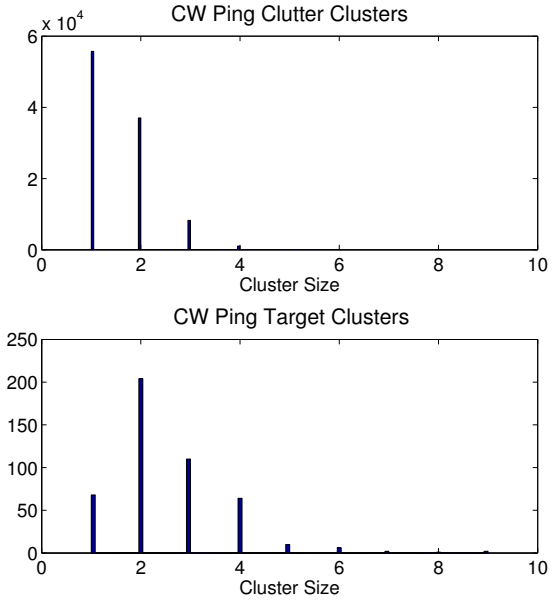

Figure 10.20: Histogram of cluster size for CW pings of PACSim scenario C. Similarities are calculated using position and Doppler measurements.

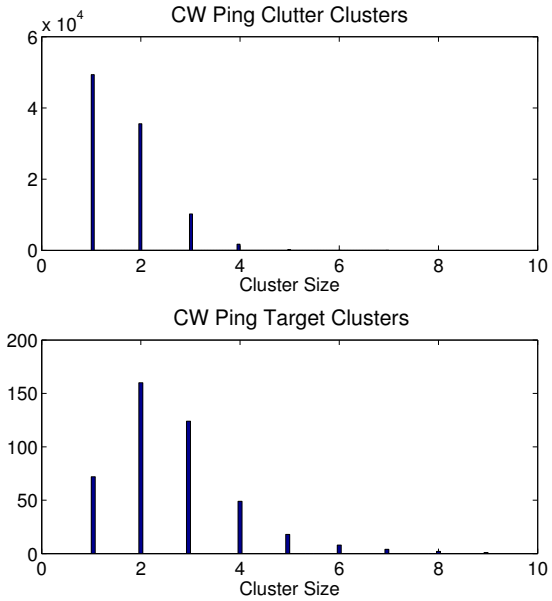

Figure 10.21: Histogram of cluster size for CW pings of PACSim scenario C. Similarities are calculated using position, Doppler and amplitude measurements.

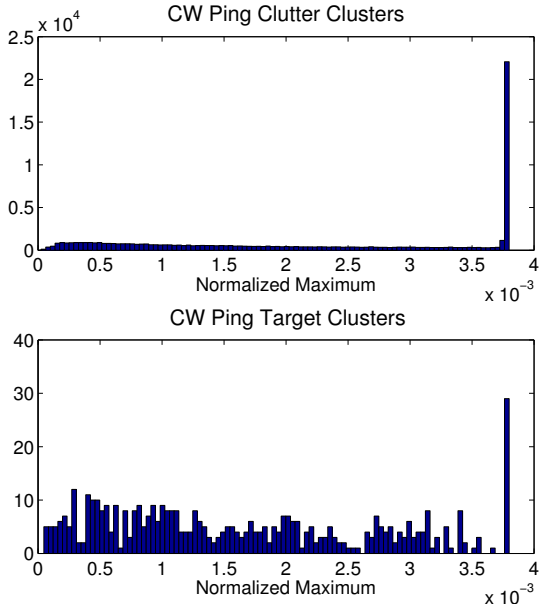

Figure 10.22: Histogram of normalized maximum from fusion for CW pings of PACSim scenario C. Similarities are calculated using position only.

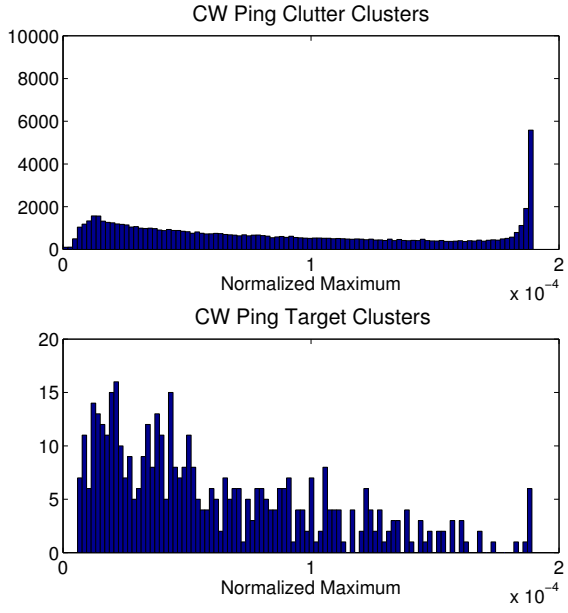

Figure 10.23: Histogram of normalized maximum from fusion for CW pings of PACSim scenario C. Similarities are calculated using position and amplitude measurements.

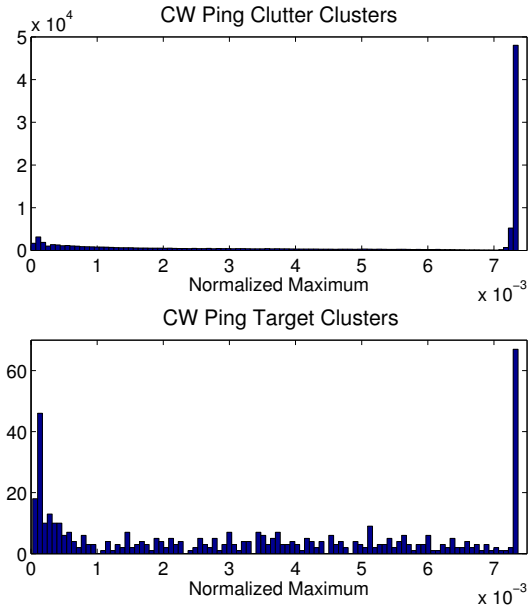

Figure 10.24: Histogram of normalized maximum from fusion for CW pings of PACSim scenario C. Similarities are calculated using position and Doppler measurements.

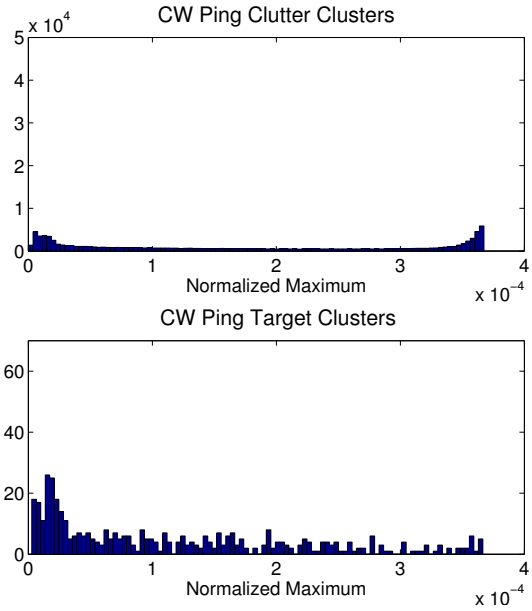

Figure 10.25: Histogram of normalized maximum from fusion for CW pings of PACSim scenario C. Similarities are calculated using position, Doppler, and amplitude measurements.

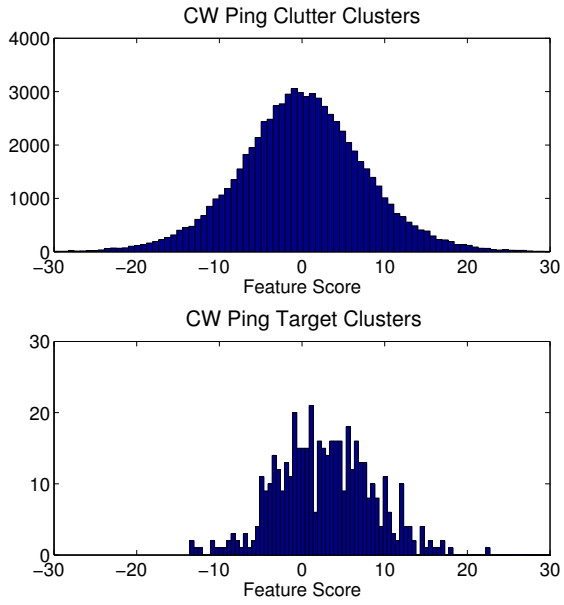

Figure 10.26: Histogram of averaged feature score for CW pings of PACSim scenario C. Similarities are calculated using position only.

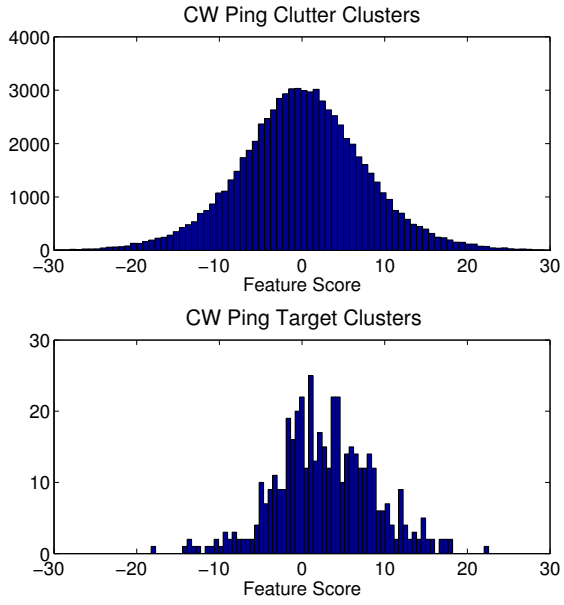

Figure 10.27: Histogram of averaged feature score for CW pings of PACSim scenario C. Similarities are calculated using position and amplitude.

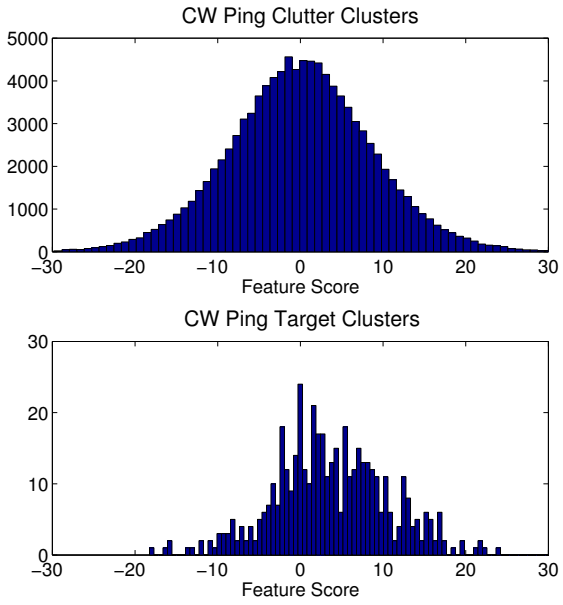

Figure 10.28: Histogram of averaged feature score for CW pings of PACSim scenario C. Similarities are calculated using position and Doppler measurements.

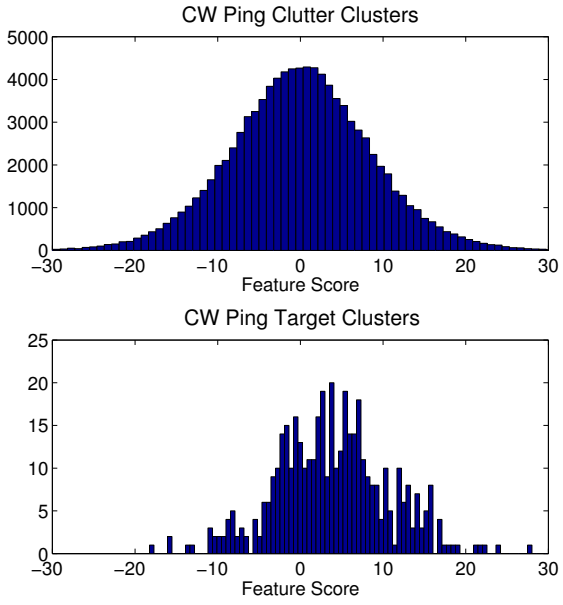

Figure 10.29: Histogram of averaged feature score for CW pings of PACSim scenario C. Similarities are calculated using position, Doppler, and amplitude measurements.

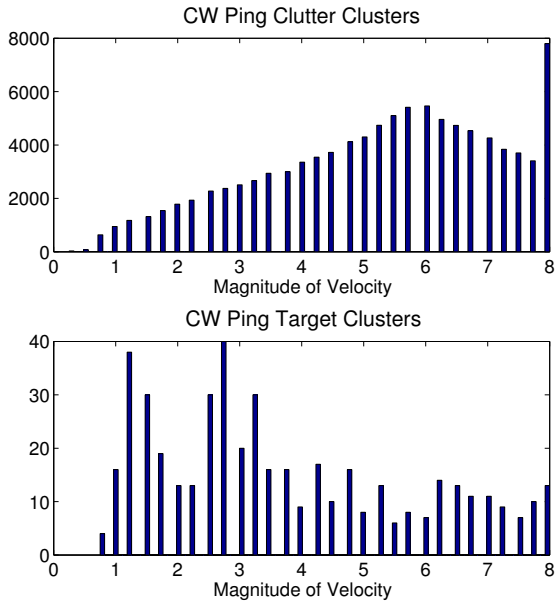

Figure 10.30: Histogram of velocity magnitude for CW pings of PACSim scenario C. Similarities are calculated using position and Doppler measurements.

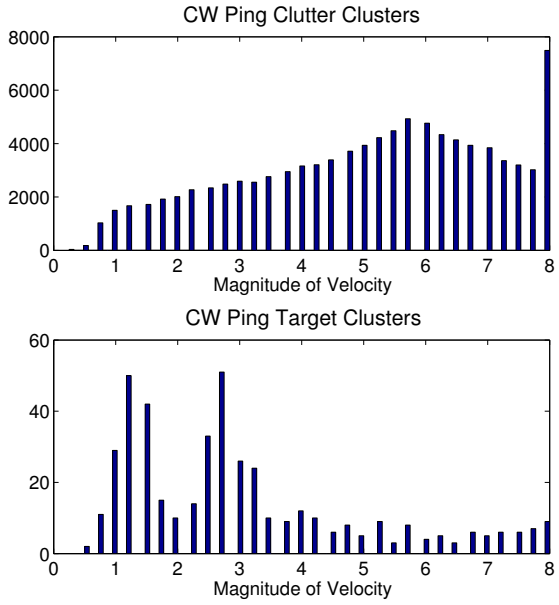

Figure 10.31: Histogram of velocity magnitude for CW pings of PACSim scenario C. Similarities are calculated using position, Doppler, and amplitude measurements.

10.5 Tracking

After contacts have been clustered, clusters fused, and clusters classified, the fused measurements can be tracked using a the JPDA-based tracker presented earlier. Each feature is thresholded at a score that maintains 90 percent of the target clusters. The results on the Metron dataset are poor, but this is expected given the properties of the dataset and the results presented in this chapter. Even after clustering and fusion of the true contacts alone, the average error in the Cartesian space is around $1500m$. Exacerbating this high error is the high inter-ping interval ($180s$). This means that in Scenario 1, targets only travel on one side of the square for about 12 pings, then make a sharp turn. For this reason, tracking was only successful on parts of Scenario 4 of the Metron dataset. This scenario contains long straight paths and the cluster-fuse-reject-track framework presented in this chapter is capable of tracking the targets. While this doesn't sound impressive, recall that the baseline for tracking on the Metron dataset with the JPDA-based tracker was no results. Figure 10.32 shows the confirmed tracks for Scenario 4.

Clustering for preprocessing is much more successful on the PACSim dataset. This is due to several factors: targets tend to move in smoother paths over longer periods of time, the inter ping interval is only $60s$, and the measurements are much more accurate, especially after clustering and fusion. Figures 10.33-10.35 show the confirmed tracks for Scenarios A-C of the PACSim dataset. There are a few spurious tracks in Scenario C due to the high clutter rate. These could potentially be removed with some post-tracking classification.

Comparing these results to the results using posterior distribution preprocessing shows a great improvement, especially on the low-P_d targets in Scenarios B and C. Tracking metrics for PACSim tracking with clustering are presented in Tables 10.2-10.4. The metrics are compared to two state of the art trackers presented in [66]. The JPDA tracker with clustering as a preprocessing step outperforms or ties JIPDA and SSPFE trackers in the TPD and TFRAG metrics for all the targets in all the datasets. It does suffer from increased TLE. This is due in part to the inability of the nearly constant velocity extended Kalman filter to follow the target around sharp turns. This could be alleviated with a multiple model tracker, however that is not in the scope of this work.

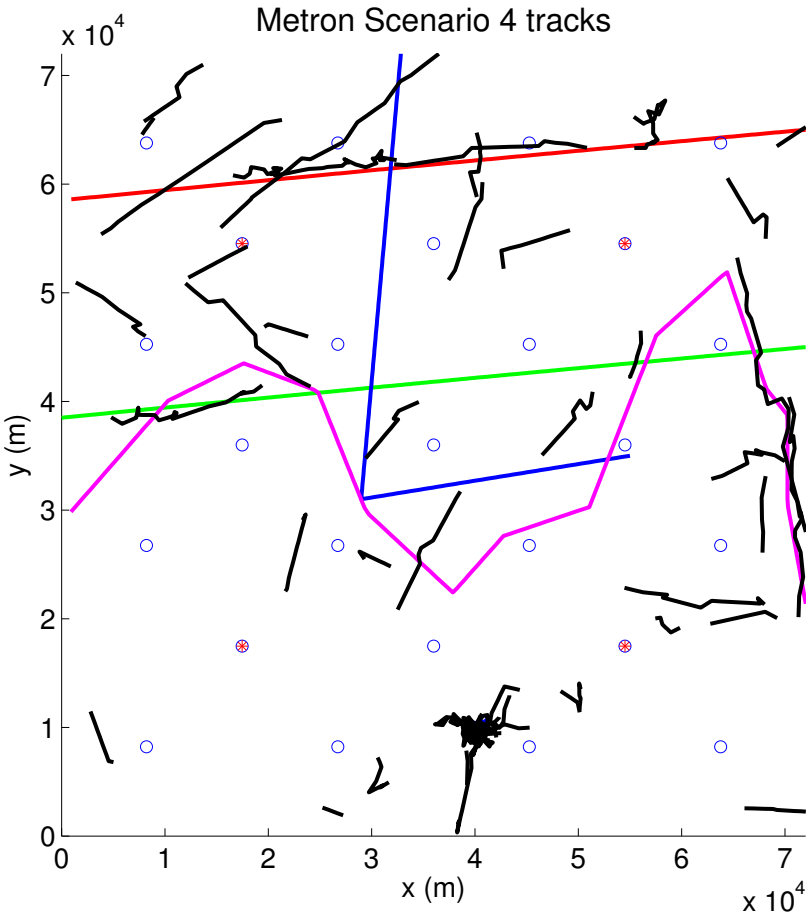

Figure 10.32: Confirmed tracks for Metron Scenario 4.

The biggest difference in performance is apparent on the low-P_d targets in Scenario B. The JPDA tracker with clustering as a preprocessing step greatly outperforms the other trackers for targets 2-4. In addition, the JIPDA and SSPFE perform very poorly (JIPDA is unable to complete) on Scenario C, due to the larger number of false contacts. No metrics were reported for SSPFE due to the large number of false tracks.

10.6 Conclusion

Fusing contacts before tracking shows great promise when it is possible to determine which contacts originate from an object of interest. On the Metron dataset, average error in

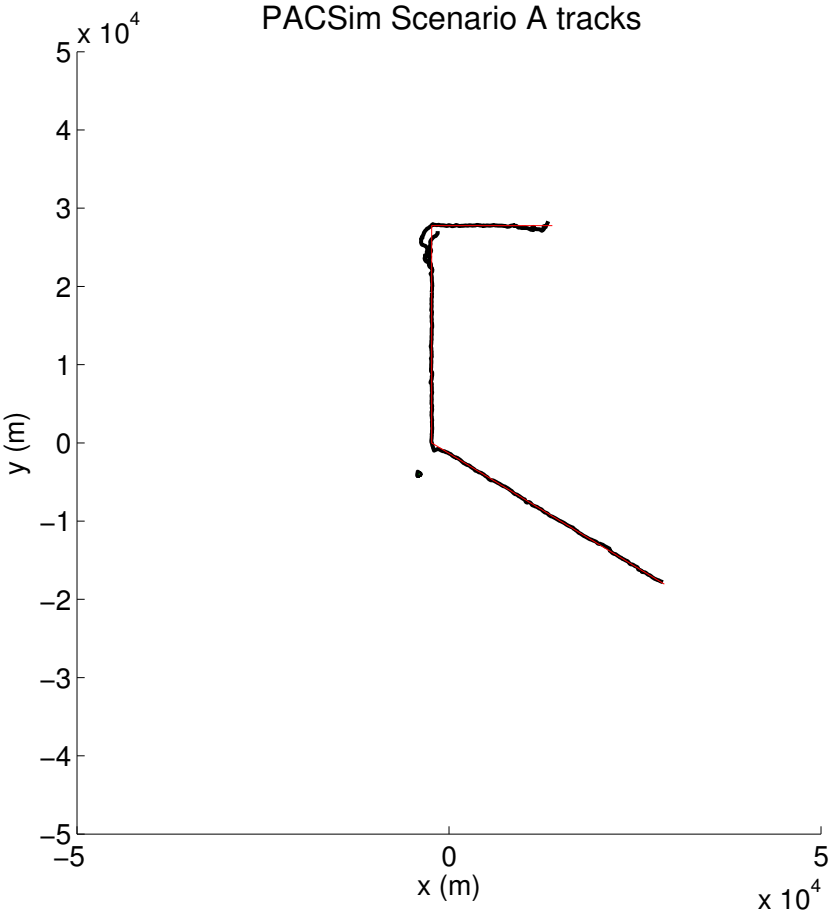

Figure 10.33: Confirmed tracks for PACSim Scenario A.

Cartesian space is reduced by a factor of more than 2.5. For the PACSim dataset CW pings, the average error is reduced by a factor of more than 2 for most targets. On CW pings, average error drops from more than $1.8km$ to less than $100m$ for most targets. In addition, fusing Doppler and amplitude measurements for CW pings allows for accurate estimation of heading and speed.

This all assumes that you have the correct labels for fusion - the likelihood-based similarity presented here is robust to measurement errors and allows for the incorporation of any features, aspect-dependent or not. Clustering using the similarity allows for grouping similar contacts for fusion and tracking.

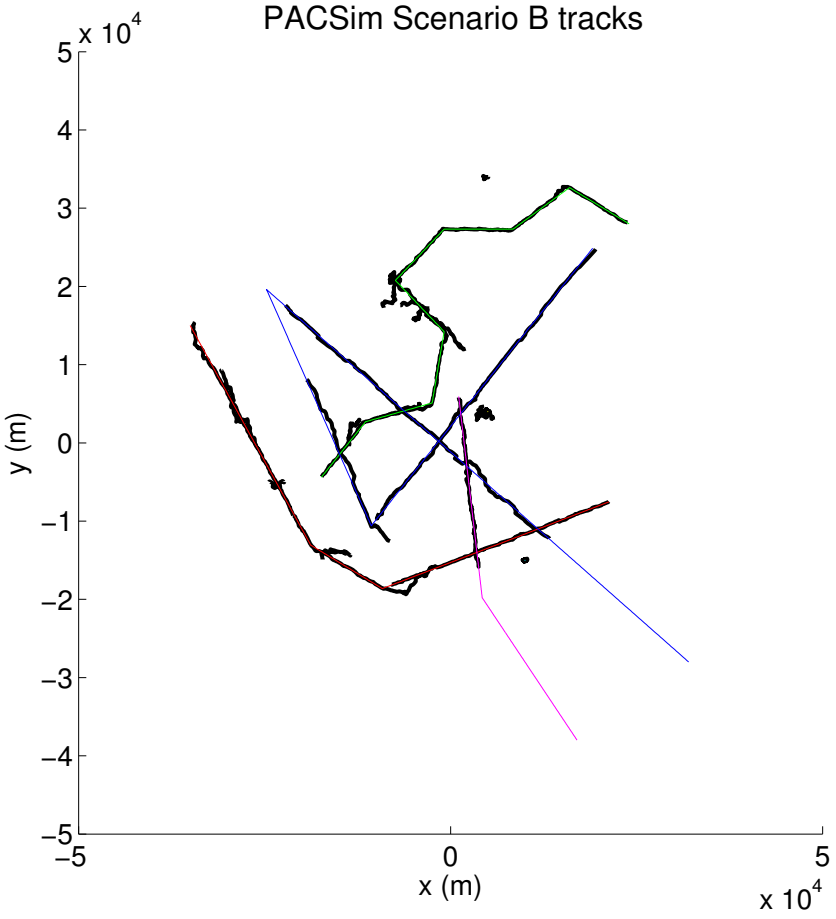

Figure 10.34: Confirmed tracks for PACSim Scenario B.

This process results in good performance on the PACSim dataset, and is able to track targets with a P_d of approximately 0.1 in high clutter (more than 40 per receiver per ping) rate environments.

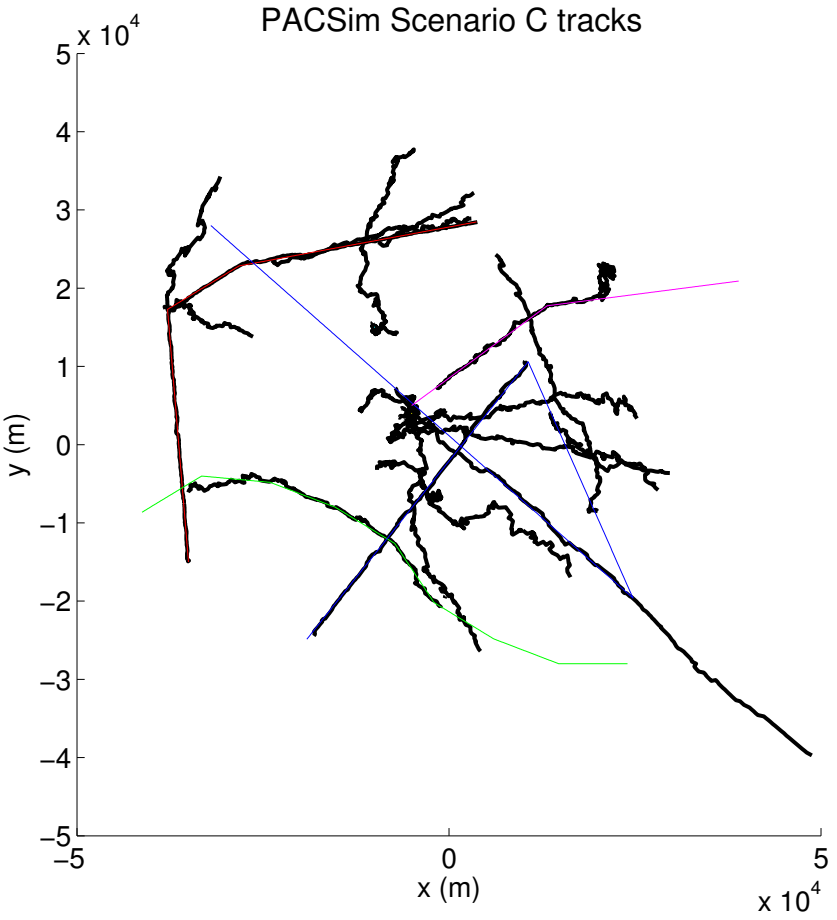

Figure 10.35: Confirmed tracks for PACSim Scenario C.

Metric	Target	Clustering	JIPDA	SSPFE
TPD	1	479	480	480
TLE	1	166.9	66.36	89.19
TFrag	1	1	1	1
TFAR		1	2	1

Table 10.2: Metric results for targets in Scenario A. For compatibility with results provided for JIPDA and SSPFE, TPD is the total number of pings that the target was tracked for and TFAR is the total number of false tracks during tracking. On this easy scenario, the trackers perform about the same on all metrics except the TLE. JIPDA and SSPFE both outperform JPDA with a clustering preprocessing step.

Metric	Target	Clustering	JIPDA	SSPFE
TPD	1	480	478	480
	2	479	476	475
	3	357	245	242
	4	205	80	124
TLE	1	171.7	63.83	95.02
	2	286.3	66.56	99.65
	3	1225.5	143.48	258.82
	4	201.4	101.68	234.47
TFrag	1	1	1	1
	2	2	1	2
	3	3	5	5
	4	1	5	6
TFAR		9	7	5

Table 10.3: Metric results for targets in Scenario B. For compatibility with results provided for JIPDA and SSPFE, TPD is the total number of pings that the target was tracked for and TFAR is the total number of false tracks during tracking. Clustering preprocessing performs the best in the TPD for all targets. It also performs best in TFrag. JIPDA and SSPFE perform better in TLE for almost all targets.

Metric	Target	Clustering
TPD	1	479
	2	267
	3	293
	4	290
TLE	1	272.6
	2	475.9
	3	1039.6
	4	369.9
TFrag	1	2
	2	1
	3	2
	4	1
TFAR		12

Table 10.4: Metric results for targets in Scenario C. For compatibility with results provided for JIPDA and SSPFE, TPD is the total number of pings that the target was tracked for and TFAR is the total number of false tracks during tracking. Metrics were not reported for JIPDA or SSPFE. TPD and TFRAG performance is again very good. The larger number of false alarms is due to the greatly increased clutter rate in PACSim C.

Chapter 11

CONCLUSIONS

In simple tracking scenarios, adding features to tracking at the appropriate stage can improve performance across the metrics. For the TNO dataset, incorporating the features into the track management portion of a JPDA-based tracker showed the best performance improvement, as presented in Chapter 6. Taking the complementary approach is true as well - incorporating information from the tracker into classification can improve classification performance, if uncertainty is modeled appropriately as presented in Chapter 7.

These less complex approaches improve tracking on data that does not have a large amount of data coming from multiple sources like a multistatic sonar field does. In a multistatic tracking scenario, a traditional tracker can choke at various stages. Track management often fails when there can be many updates without a true contact. High bearing and bistatic range errors combine at long distances to make measurements extremely unreliable. Clutter rate increases linearly with the number of receivers. It is possible to modify the tracker, threshold out unlikely contacts, and devote massive compute time to overcome these issues, but it is also possible to apply a flexible posterior distribution preprocessing approach to the data that distills the large number of contacts to a probability surface. This surface can incorporate features in addition to measurements from multiple sensing modalities (high frequency imaging sonar and HD video, for example). The preprocessing step allows for tracking of high P_d targets, as presented in Chapter 9.

This posterior distribution preprocessing step, combined with earlier work on position estimation using multistatic Doppler gave way to a clustering approach that clusters contacts that are likely to have come from the same target. Similarity between contacts is calculated using a likelihood-based approach that appropriately models the measurement error and allows for incorporation of any features, aspect dependent or not. After clustering, contacts are fused using a likelihood-based approach that is a simple extension of the similarity

calculation and results in greatly reduced localization error. If Doppler, amplitude or other aspect-dependent features are available, the fusion step can also result in an estimate of the object's speed and/or heading. These fused clusters can be classified based on fused features of the contacts within the cluster (for example, velocity estimates) or features of the cluster itself (for example, cluster size). The fused and classified contacts can then be tracked, resulting in a tracker than performs well on low P_d (≈ 0.1) targets in high clutter rate (> 40 false contacts per receiver per ping) scenarios. Depending on the metric, the performance of the JPDA tracker with the clustering preprocessing step outperforms two state of the art trackers. It has much better track probability of detection and track fragmentation characteristics, however it does have increased track localization error. This is due to two factors: because the tracker is able to follow the target in difficult conditions, it is likely coasting, resulting in increased error. In addition, the underlying EKF used for prediction does not handle sharp turns due to the nearly constant velocity assumption. Incorporating a switching motion model would help to decrease the tracking error near the corners.

All of these research steps combine into a tracker that is built on a clustering, fusion, classification framework that can handle a feature measurement that is dependent or independent of the object's kinematic state.

11.0.1 Future Work

Though performance on the PACSim dataset is quite good, several additional steps could potentially improve performance. A post-tracking classification approach, similar to the SPECSweb approach could reject additional false tracks. The tracks would build up a number of samples of the available features, then they can be classified at a later time.

In addition, cluster classification was done using simple thresholding on features. A more complex classifier could provide improved clutter rejection. It would be important to make sure that the classifier is trained on representative data - the results on PACSim made it clear that Scenario A was much easier than Scenarios B or C.

One consistent issue with the tracker in the PACSim and Metron datasets is the inability of the tracker to hold targets as they go around very sharp turns. This was due to the

nearly constant velocity model used for the Kalman filter. It locks on to a target moving in a straight line and expects all future measurements to follow that straight line. The switch to multiple motion models and including a turn model could greatly improve a real tracking system's performance.

The velocity estimates from cluster fusion are currently being used as a classification feature, however they could also be used to improve track initiation and/or association. This could potentially help the tracker stay with a target around the sharp turns that are made in the PACSim dataset. It would also allow for improved track initiation, due to an initial estimate of velocity being available.

The environmental model used to generate the PACSim dataset was quite complex, however in this work it is approximated using an extreme simplification of the sonar equation. A more complicated model could help improve the similarity and fusion steps, however, environmental modeling takes large amounts of compute power, and would only be feasible in a production tracking system.

BIBLIOGRAPHY

[1] Robert F. Gragg. The BASIS-3D acoustic target strength model. Technical Report TR-MAR-12-2008, The Technical Cooperation Program, September 2008.

[2] Robert P. Goddard. The Sonar Simulation Toolset. *U.S. Navy Journal of Underwater Acoustics*, 54(1):193–232, 2004.

[3] R.J. Urick. *Principles of underwater sound*, volume 3. McGraw-Hill New York, 1975.

[4] H.C. Hayes. Detection of submarines. *Proceedings of the American Philosophical Society*, 59(1):1–47, 1920.

[5] L. Batchelder. When sonar was called submarine signalling. *The Journal of the Acoustical Society of America*, 31(6):832–832, 1959.

[6] E. Hanusa, W.H. Mortensen, D.W. Krout, and J. McLaughlin. Improving target tracking performance by incorporating classification information. In *OCEANS 2010*, pages 1–7. IEEE, 2010.

[7] E. Hanusa and D. Krout. Improving Contact Classification Using an Estimate of Aspect from Track State. In *Oceans 2011*. IEEE, September 2011.

[8] D. Krout and E. Hanusa. Likelihood Surface Preprocessing with the JPDA Algorithm: Metron Data Set. In *Information Fusion (FUSION), 2010 Proceedings of the the 13th International Conference on*, July 2010.

[9] E. Hanusa and D.W. Krout. Posterior distribution preprocessing with the jpda algorithm: Pacsim data set. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2593–2599. IEEE, 2012.

[10] D.W. Krout, G. Okopal, and E. Hanusa. Video data and sonar data: Real world data fusion example. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–5. IEEE, 2011.

[11] D.W. Krout, W. Kooiman, G. Okopal, and E. Hanusa. Object tracking with imaging sonar. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2400–2405. IEEE, 2012.

[12] D.W. Krout, G. Okopal, A. Jessup, and E. Hanusa. Tracking drifting surface objects with aerial infrared and electro-optical sensors. In *OCEANS 2012*, 2012.

[13] E. Hanusa, D. Krout, and M. R. Gupta. Estimation of Position from Multistatic Doppler Measurements. *Information Fusion (FUSION), 2010 13th International Conference on*, July 2010.

[14] E. Hanusa, M.R. Gupta, and D.W. Krout. Contact clustering and classification using likelihood-based similarities. In *OCEANS 2012*. IEEE, 2012.

[15] E. Hanusa, D. Krout, and M.R. Gupta. Clutter Rejection by Clustering Likelihood-Based Similarities. In *Information Fusion (FUSION), 2011 14th International Conference on*. IEEE, 2011.

[16] D. Lerro and Y. Bar-Shalom. Automated tracking with target amplitude information. In *American Control Conference, 1990*, pages 2875 –2880, May 1990.

[17] J.W. Pitton and W.L.J. Fox. Incorporating target strength into environmentally-adaptive sonar tracking. *OCEANS 2007-Europe*, pages 1–5, 2007.

[18] D.W. Krout and D. Morrison. PDAFAI vs. PDAFAIwTS: TNO blind dataset and SEABAR'07. In *Information Fusion (FUSION), 2009 Proceedings of the 12th International Conference on*, pages 1845–1850. IEEE, 2009.

[19] D. Grimmett. Reduction of False Alarm Rate in Distributed Multistatic Sonar Systems through Detection Cueing. In *OCEANS 2007-Europe*, pages 1–6. IEEE, 2007.

[20] D. Grimmett. Multistatic target tracking using specular cue initiation and directed data retrieval. In *Information Fusion (FUSION) 2008. 11th International Conference on*, pages 1–8. IEEE, 2008.

[21] D. Grimmett. SPECSweb multistatic tracking on a truth-blind simulated scenario of the MSTWG. In *Information Fusion (FUSION), 2009. 12th International Conference on*, pages 1568–1575. IEEE, 2009.

[22] R. Ricks, D. Grimmett, and C. Wakayama. Passive acoustic tracking for cueing a multistatic active acoustic tracking system. In *OCEANS, 2012-Yeosu*, pages 1–7. IEEE, 2012.

[23] C. Wakayama, D. Grimmett, and R. Ricks. Active multistatic track initiation cued by passive acoustic detection. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2577–2584. IEEE, 2012.

[24] D. Grimmett, C. Wakayama, and R. Plate. Multistatic post-track classification using a target strength function. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2339–2346. IEEE, 2012.

[25] C.S. Agate and K.J. Sullivan. Signature-aided tracking using association hypotheses. In *Proceedings of SPIE, Conference on Signal Processing, Sensor Fusion, and Target Recognition XI*, volume 4729, pages 44–55, 2002.

[26] Y. Bar-Shalom, T. Kirubarajan, and C. Gokberk. Tracking with classification-aided multiframe data association. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(3):868–878, 2005.

[27] L. Hong, S. Cong, M.T. Pronobis, and S. Scott. Wavelets feature aided tracking (WFAT) using GMTI/HRR data. *Signal Processing*, 83(12):2683–2690, 2003.

[28] L. Hong, N. Cui, M. Pronobis, and S. Scott. Local motion feature aided ground moving target tracking with GMTI and HRR measurements. *IEEE Transactions on Automatic Control*, 50(1):127–133, 2005.

[29] D.J. Salmond and M.C. Parr. Track maintenance using measurements of target extent. In *Radar, Sonar and Navigation, IEE Proceedings-*, volume 150, pages 389–395. IET, 2003.

[30] K.C. Chang and R. Fung. Target identification with Bayesian networks in a multiple hypothesis tracking system. *Optical Engineering*, 36:684, 1997.

[31] E. Weinstein and N. Levanon. Passive array tracking of a continuous wave transmitting projectile. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-16(5):721 –726, Sept. 1980.

[32] Y.T. Chan and J.J. Towers. Passive localization from Doppler shifted frequency measurements. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 40(10):1465–1468, 1991.

[33] Y.T. Chan and J.J. Towers. Sequential localization of a radiating source by Doppler-shifted frequency measurements. *Aerospace and Electronic Systems, IEEE Transactions on*, 28(4):1084 –1090, Oct 1992.

[34] Y.T. Chan and S.W. Rudnicki. Bearings-only and Doppler-bearing tracking using instrumental variables. *Aerospace and Electronic Systems, IEEE Transactions on*, 28(4):1076 –1083, Oct 1992.

[35] X. Wang, D. Musicki, R. Ellem, and F. Fletcher. Enhanced Multi-Target Tracking with Doppler Measurements. *Information, Decision and Control, 2007. IDC'07*, pages 53–58, 2007.

[36] A. Amar and A.J. Weiss. Localization of narrowband radio emitters based on Doppler frequency shifts. *Signal Processing, IEEE Transactions on*, 56(11):5500 –5508, Nov. 2008.

[37] B. R. La Cour. Bayesian sensor registration for multistatic active sonar. *Conference Proceedings, IEEE OCEANS '05 EUROPE, 20-23 June 2005, Brest, France*, pages 131–136.

[38] C. Carthel, S. Coraluppi, K. Bryan, and G. Arcieri. Wide-area feature-aided tracking with intermittent multi-sensor data. In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, 2010.

[39] J. Fan, Y. Zhu, S. Fan, H. Fan, and Q. Fu. Feature aided switching model set approach for maneuvering target tracking. *Progress In Electromagnetics Research B*, 45:251–268, 2012.

[40] M. Daun, W. Koch, and R. Klemm. Tracking of ground targets with bistatic airborne radar. In *Radar Conference, 2008. RADAR'08. IEEE*, pages 1–6. IEEE, 2008.

[41] W. Koch. On bayesian tracking and data fusion: A tutorial introduction with examples. *Aerospace and Electronic Systems Magazine, IEEE*, 25(7):29–52, 2010.

[42] S. Coraluppi and C. Carthel. An ML-MHT approach to tracking dim targets in large sensor networks. In *Information Fusion (FUSION), 2010. 13th International Conference on*, July 2010.

[43] S. Coraluppi and C. Carthel. A hierarchical mht approach to esm-radar fusion. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 677–683. IEEE, 2012.

[44] G.R. Mellema. Multistatic tracking of dim targets in large sonobuoy fields using the metron blind data set. In *OCEANS 2011*, pages 1–9. IEEE, 2011.

[45] G.R. Mellema. Feature-assisted multistatic tracking using the pacsim data set. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2608–2615. IEEE, 2012.

[46] R. Georgescu, P. Willett, S. Marano, and V. Matta. Predetection fusion in large sensor networks with unknown target locations. *Submitted to ISIF Journal of Advances in Information Fusion*, 2010.

[47] R. Georgescu and P. Willett. Predetection fusion with doppler measurements and amplitude information. *Oceanic Engineering, IEEE Journal of*, 37(1):56–65, 2012.

[48] R. Georgescu and P. Willett. The GM-CPHD Tracker Applied to Real and Realistic Multistatic Sonar Data Sets. *Oceanic Engineering, IEEE Journal of*, 37(2):220–235, 2012.

[49] R. Georgescu and P. Willett. Random finite set markov chain monte carlo predetection fusion. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8. IEEE, 2011.

[50] CS Clay and H. Medwin. *Acoustical oceanography: Principles and applications(Book)*. 1977.

[51] N. Allen, P.C. Hines, V.W. Young, and D.A. Caldwell. Study on the human ability to aurally discriminate between target echoes and environmental clutter in recordings of incoherent broadband sonar. *The Journal of the Acoustical Society of America*, 119(5):3395–3395, 2006.

[52] S. Philips, J. Pitton, and L. Atlas. Perceptual feature identification for active sonar echoes. In *OCEANS 2006*, pages 1–6. IEEE, 2006.

[53] N. Allen, P.C. Hines, and V.W. Young. Performances of human listeners and an automatic aural classifier in discriminating between sonar target echoes and clutter. *The Journal of the Acoustical Society of America*, 130:1287, 2011.

[54] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, 1999.

[55] S. Coraluppi, D. Grimmett, and P. de Theije. Benchmark evaluation of multistatic trackers. In *Information Fusion (FUSION), 2006. 9th International Conference on Information Fusion*, pages 1–7, 2006.

[56] D. Grimmett. MSTWG - PACsim dataset . SPAWAR Systems Center Pacific, Released to the MSTWG, Powerpoint, July 2011.

[57] Y. Viniotis. *Probability and random processes for electrical engineers*. WCB/McGraw-Hill, 1998.

[58] D. Grimmett. SPECSweb tracking on a large, simulated multistatic field of the MSTWG. In *Information Fusion (FUSION), 2010. 13th International Conference on*, pages 1–8. IEEE, 2010.

[59] R. Georgescu, P. Willett, and S. Schoenecker. GM-CPHD and ML-PDA Applied to the Metron Multi-Static Sonar Dataset. In *The 13th International Conference on Information Fusion*, July 2010.

[60] D. L. Hall. *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Norwood, MA, 1992.

[61] S. Coraluppi, D. Grimmett, and P. de Theije. Benchmark Evaluation of Multistatic Trackers. In *The 9th International Conference on Information Fusion*, July 2006.

[62] R. Arora, M.R. Gupta, A. Kapila, and M. Fazel. Clustering by left-stochastic matrix factorization. In *Proc. International Conference on Machine Learning (ICML)*, 2011.

[63] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[64] C.D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[65] J. Folkesson, J. Leonard, J. Leederkerken, and R. Williams. Feature tracking for underwater navigation using sonar. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3678–3684. IEEE, 2007.

[66] F. Fletcher and S. Arulampalam. A comparison of existence-based multitarget trackers for multistatic sonar. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2362–2369. IEEE, 2012.